

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DIRETORIA DE PESQUISA E PÓS GRADUAÇÃO  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
CURSO DE ESPECIALIZAÇÃO EM SISTEMAS EMBARCADOS PARA A  
INDÚSTRIA AUTOMOTIVA

VICTOR VOLOCHTCHUK DE ARAUJO

**CIRCUITO SIMULADOR DE MOVIMENTAÇÃO  
VEICULAR COM JOYSTICK**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA  
2020

VICTOR VOLOCHTCHUK DE ARAUJO

**CIRCUITO SIMULADOR DE MOVIMENTAÇÃO  
VEICULAR COM JOYSTICK**

Monografia de Especialização, apresentada ao Curso de Especialização em Sistemas Embarcados para Indústria Automotiva, do Departamento Acadêmico de Eletrônica – DAELN, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Dr. Guilherme Alceu Schneider

CURITIBA  
2020



Ministério da Educação  
Universidade Tecnológica Federal do Paraná  
Câmpus Curitiba

Diretoria de Pesquisa e Pós-Graduação  
Departamento Acadêmico de Eletrônica  
Curso de Especialização em Sistemas Embarcados para Indústria  
Automotiva



---

## TERMO DE APROVAÇÃO

CIRCUITO SIMULADOR DE MOVIMENTÇÃO VEICULAR COM JOYSTICK

por

VICTOR VOLOCHTCHUK DE ARAUJO

Esta monografia foi apresentada em 04 de Maio de 2020 como requisito parcial para a obtenção do título de Especialista em Sistemas Embarcados para Indústria Automotiva. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Dr. Guilherme Alceu Schneider  
Orientador

---

Prof. Dr. Kleber Kendy Horikawa Nabas  
Membro titular

---

Prof. M. Sc. Omero Francisco Bertol  
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

## **AGRADECIMENTOS**

Primeiramente gostaria de agradecer a Deus. Agradeço ao meu orientador Guilherme Alceu Schneider por aceitar conduzir o meu trabalho de pesquisa. A todos os meus professores do Curso de Especialização em Sistemas Embarcados para a Indústria Automotiva da Universidade tecnológica Federal do Paraná pela excelência da qualidade técnica de cada um. Aos meus pais Valter Soares de Araujo e Elisabete Volochtchuk de Araujo que sempre estiveram ao meu lado me apoiando ao longo de toda minha trajetória. Também à minha esposa Ana Vitória Lachowski Volochtchuk pela compreensão e paciência demonstrada durante o período do projeto.

## RESUMO

ARAÚJO, Victor Volochchuk de. **Circuito Simulador de Movimentação Veicular com Joystick**. 2020. 35 f. Monografia de Especialização (Especialização em Sistemas Embarcados para a Indústria Automotiva) - Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2020.

Este trabalho apresenta um sistema simplificado para a simulação de um veículo elétrico com o uso de um *joystick*. O trabalho foi elaborado pensando em disponibilizar uma forma para que pessoas que tenham limitações motoras possam usufruir autonomamente de um veículo. Para esse projeto foi utilizada a comunicação com barramento CAN com a finalidade de trazer o experimento para próximo do mundo automotivo, uma vez que este tipo de comunicação é amplamente utilizada em veículos que estão a disposição da população. Para a realização do projeto foram utilizados 2 placas Arduino *Leonardo* devido a praticidade de montagem geral do circuito necessário, a facilidade para a programação do sistema, bem como a familiaridade que o autor tem com o material. Em conjunto foram utilizados módulos MCP2515, responsáveis por gerenciar o barramento CAN e enviar/receber as mensagens através do barramento, e o módulo ponte H L298N, responsável pelo acionamento do motor elétrico. Como sensor e atuadores temos um *joystick*, que é o responsável por enviar os sinais para a movimentação, um servo motor SG90, para simular o sistema de direção e um motor elétrico para simular o motor propriamente dito. Como resultado o autor verificou que o sistema funcionou como esperado, fazendo com que o sistema de direção e de motor fossem acionados através de um *joystick* presente no outro lado do barramento CAN em relação aos atuadores. O autor propôs como ações futuras uma maior proximidade de empresas para possível implementação industrial, primeiramente em veículos elétricos e após, juntar essa tecnologia e desenvolver em veículos a combustão.

**Palavras-chave:** CAN. Veículo elétrico. Simulador de movimentação. Acessibilidade. Veículo com *Joystick*.

## ABSTRACT

ARAUJO, Victor Volochtchuk de. **Vehicle Movement Simulation Circuit with Joystick**. 2020. 35 f. Monografia de Especialização (Especialização em Sistemas Embarcados para a Indústria Automotiva) - Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2020.

This work presents a simplified system for the simulation of an electric vehicle using a joystick. The work was designed with the intention of providing a way for people with motor limitations to use autonomously a vehicle. For this project, communication with CAN bus was used in order to bring the experiment closer to the automotive world, since this type of communication is widely used in vehicles that are available to the population. For the realization of the project, 2 Arduino Leonardo boards were used due to the practicality of general assembly of the necessary circuit, the facility for programming the system, as well as the familiarity that the author has with the material. MCP2515 modules were used together, responsible for managing the CAN bus and sending / receiving messages through the bus and the L298N H-bridge module, responsible for driving the electric motor. As a sensor and actuators we have a joystick, which is responsible for sending the signals for movement and a SG90 servo motor, to simulate the steering system and an electric motor to simulate the motor itself. As a result, the author verified that the system worked as expected, causing the steering and engine activations through a joystick present on the other side of the CAN bus, in relation to the actuators. The author proposed as future actions a greater proximity of companies for possible industrial implementation, first in electric vehicles and after joining this technology and developing in combustion vehicles.

**Keywords:** CAN. electric vehicle. Moving simulator. accessibility. Vehicle with joystick.

## LISTA DE FIGURAS

Figura 1 – Custos de eletrônica em um veículo . . . . .	9
Figura 2 – Benefícios do barramento CAN . . . . .	10
Figura 3 – Barramento CAN veicular . . . . .	10
Figura 4 – <i>Cockpit</i> de um Airbus A-380 . . . . .	11
Figura 5 – Topologia do sistema . . . . .	12
Figura 6 – <i>Joystick</i> KY-023 . . . . .	14
Figura 7 – Resistores para o eixo x . . . . .	14
Figura 8 – Servo motor SG90 . . . . .	15
Figura 9 – Motor elétrico . . . . .	15
Figura 10 – Fonte regulável . . . . .	16
Figura 11 – Arduino usado no projeto . . . . .	17
Figura 12 – Placa emissora . . . . .	18
Figura 13 – Placa receptora . . . . .	18
Figura 14 – MCP2515 . . . . .	18
Figura 15 – L298N . . . . .	20
Figura 16 – Lógica de ativação . . . . .	20
Figura 17 – Controle de velocidade . . . . .	21
Figura 18 – Esquemático da topologia . . . . .	21
Figura 19 – Circuito montado . . . . .	21
Figura 20 – <i>Bits</i> CAN . . . . .	23
Figura 21 – Mensagem CAN . . . . .	23
Figura 22 – Estrutura de uma mensagem CAN . . . . .	23
Figura 23 – <i>Joystick</i> eixo x posição neutra . . . . .	26
Figura 24 – <i>Joystick</i> eixo x posição intermediária a direita . . . . .	27
Figura 25 – <i>Joystick</i> eixo x posição final a direita . . . . .	27
Figura 26 – Tempos de leitura serial . . . . .	28
Figura 27 – <i>Joystick</i> eixo x posição intermediária a esquerda . . . . .	32
Figura 28 – <i>Joystick</i> eixo y posição intermediária reverso . . . . .	32
Figura 29 – <i>Joystick</i> eixo x posição final a esquerda . . . . .	33
Figura 30 – <i>Josytick</i> eixo y posição intermediária a frente . . . . .	33
Figura 31 – <i>Josytick</i> eixo y posição final a frente . . . . .	33
Figura 32 – <i>Joystick</i> eixo y posição final reverso . . . . .	34

## LISTA DE ABREVIATURAS E SIGLAS

ABS	<i>Anti Blocking System</i>
CAN	<i>Controller Area Network</i>
CS	<i>Chip Select</i>
DLC	<i>Data Length Code</i>
ENA	<i>Enable</i>
GND	<i>Ground</i>
ICSP	<i>In-Circuit Serial Programming</i>
ISO	<i>International Organization for Standardization</i>
MISO	<i>Master In Slave Out</i>
MOSI	<i>Master Out Slave In</i>
Mbps	<i>Megabyte por segundo</i>
NI	<i>National Instruments</i>
OSI	<i>Open Systems Interconnection</i>
PNE	<i>Pessoa com Necessidades Especiais</i>
PWM	<i>Pulse Width Modulation</i>
RPM	<i>Rotação por minuto</i>
SCK	<i>Serial Clock</i>
s.d.	<i>Sem data</i>
SPI	<i>Serial Peripheral Interface</i>
VCC	<i>Voltage Continuous Current</i>

## LISTA DE SÍMBOLOS

mA      miliamperes

V      Volts

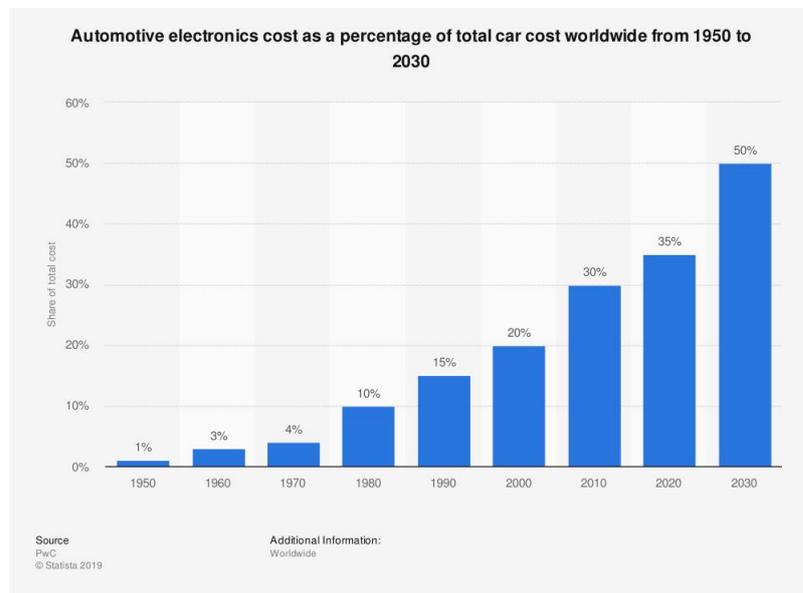
## SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>9</b>
1.1	Motivação	10
1.2	Principais Decisões do Projeto	11
1.3	Topologia do Sistema	11
<b>2</b>	<b>Hardware</b>	<b>13</b>
2.1	Sensores	13
2.2	Atuadores	14
2.2.1	Servomotor <i>Tower Pro</i> SG90	14
2.2.2	Motor elétrico	15
2.3	Controladores	16
2.3.1	Arduino <i>Leonardo</i>	16
2.3.2	MCP2515	17
2.3.3	L298N	19
<b>3</b>	<b>Comunicação e Software</b>	<b>22</b>
3.1	Comunicação CAN	22
3.2	Software	24
3.2.1	1.Emissor	24
3.2.2	2.Receptor	25
3.3	Experimentos e Discussões	26
<b>4</b>	<b>Conclusão</b>	<b>29</b>
	<b>Referências</b>	<b>30</b>
	<b>ANEXO A Imagens dos testes</b>	<b>32</b>

## 1.INTRODUÇÃO

A indústria automotiva mundial está em constante crescimento e evolução, como se notam as diferenças em estética, segurança e conforto entre os veículos dos anos 90 e os veículos atuais. Ocorreram mudanças na forma de produção, na propulsão e, principalmente, em número de módulos eletrônicos presentes nos automóveis. Como é possível ver na [Figura 1](#), os custos de componentes eletrônicos em relação ao preço de um automóvel alcançaram 30% em 2010 em comparação a 15% em 1990. É possível verificar que este crescimento ocorreu em uma escala acentuada e continua crescendo, com o horizonte demonstrado para os veículos no futuro próximo, no qual estão incluídos os conceitos de veículos conectados e autônomos.

Figura 1 – Custos de eletrônica em um veículo

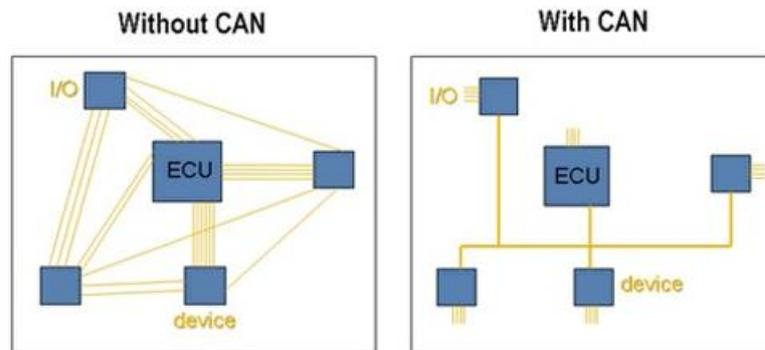


Fonte: [Statita \(2019\)](#).

Com o aumento dos módulos eletrônicos embarcados no veículo e com a finalidade de gerenciar e facilitar a comunicação módulos/sensores e atuadores foi proposto organizá-los em um barramento de comunicação, conhecido como CAN (*Controlled Area Network*). Como pode-se ver na [Figura 2](#), a comunicação é facilitada ao se centralizar as informações

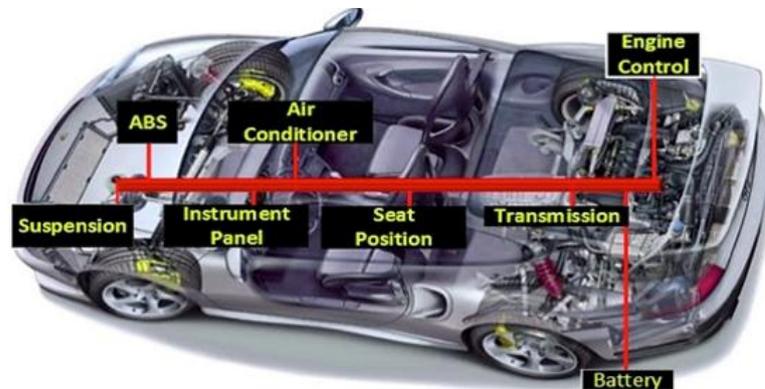
em um único barramento, eliminando grande quantidade do cabeamento interno facilitando a organização da arquitetura eletro-eletrônica do veículo. Na [Figura 3](#) é possível verificar um exemplo de como o barramento está no veículo interligando os módulos (Motor, habitáculo, ABS, *airbag*, dentre outros) a esse barramento.

Figura 2 – Benefícios do barramento CAN



Fonte: [NI \(2019\)](#).

Figura 3 – Barramento CAN veicular



Fonte: [All-About-Circuits \(2019\)](#).

## 1.1.MOTIVAÇÃO

A indústria automotiva está na vanguarda das inovações, porém algumas pessoas da sociedade não tem a possibilidade de usufruir do veículo de forma autônoma, isso devido ao fato de terem alguma limitação motora ou amputação em braços e/ou pernas, conhecidas como Pessoas com Necessidades Especiais (PNE's). Deste modo, com a utilização da atual tecnologia presente em automóveis, foi proposta a montagem de um sistema de simulação de movimentação veicular utilizando um *joystick* para comandar o sistema. A inspiração

foi a cabine de controle dos aviões da marca Airbus, como é possível ver na [Figura 4](#), que apresenta o *cockpit* de um Airbus A380 com os seus *joysticks* de comando.

Figura 4 – *Cockpit* de um Airbus A-380



Fonte: [Airbus \(2014\)](#).

Esse sistema possibilitará que as PNE's consigam controlar um veículo por si só com o uso do *joystick*, necessitando somente de uma mão, ou pequena adaptação para que o veículo se mova como desejar.

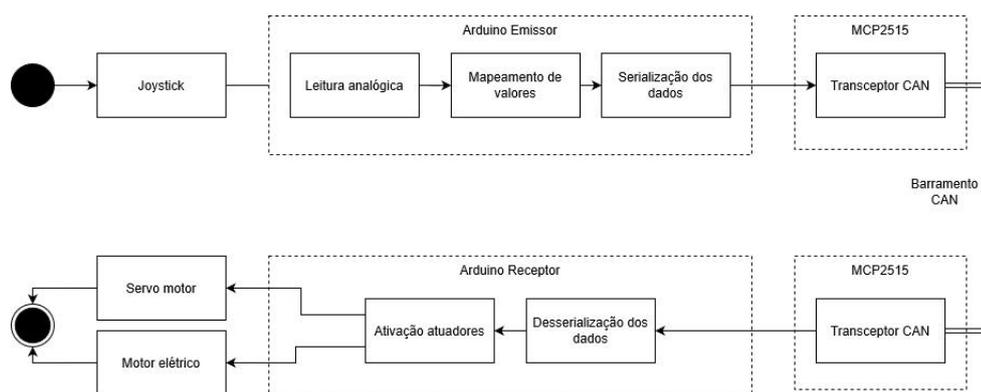
## 1.2.PRINCIPAIS DECISÕES DO PROJETO

Para a realização desse projeto foram tomadas algumas decisões levando em conta a praticidade, a usabilidade e a proximidade de um cenário real. Foi decido isolar a aplicação para uma escala menor, somente utilizando os sinais do sensor e os atuadores necessários para verificar a funcionalidade do projeto. De tal forma foram escolhidos componentes já prontos, placas e controladores, capazes de realizar a transmissão CAN para que a simulação mesmo em escala reduzida fosse próxima do que acontece em um veículo.

## 1.3.TOPOLOGIA DO SISTEMA

A [Figura 5](#) ilustra a topologia do sistema que é dividida em blocos que se interconectam. Assim os dados podem ser tratados da maneira desejada.

Figura 5 – Topologia do sistema



Fonte: Autoria própria.

## 2.HARDWARE

O sistema proposto pode ser dividido em hardware e software, esta seção apresentará o *hardware* utilizado (sensores, atuadores e controladores) para que o projeto fosse desenvolvido.

### 2.1.SENSORES

Para esse projeto foi utilizado um único sensor que é responsável por indicar o eixo de movimento e a graduação do movimento das rodas. Foi utilizado o *joystick* KY-023 (Figura 6). Esse *joystick* é composto por 5 pinos de conexão, que tem a seguinte função de acordo com [Datasheets-pdf \(2017\)](#):

- Pino 1: Botão estilo *push botton*;
- Pino 2: Resistor do eixo y;
- Pino 3: Resistor do eixo x;
- Pino 4: Entrada de alimentação;
- Pino 5: Entrada massa elétrica.

Nos pinos dos resistores dos eixos x e y estão ligados 2 resistores variáveis, como mostrado na Figura 7. Na posição neutra o valor da tensão de entrada, normalmente entre 3,3V e 5V, pode ser alterada com a movimentação da alavanca para fora da posição de repouso. Haverá uma variação de tensão entre os terminais de saída e de massa elétrica. Essa variação ocorre no *range* de 0V a 5V.

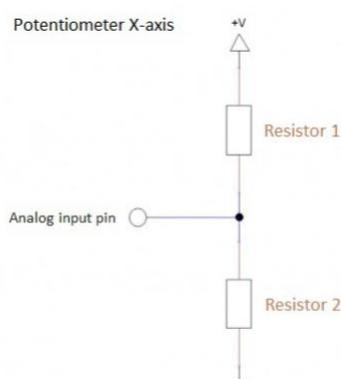
No atual projeto foram utilizadas as saídas do eixo x, responsável pela movimentação das rodas, e do eixo y, responsável pela ativação do motor. Esses pinos foram então conectados a placa Arduino *Leonardo* para que fossem feitas as leituras dos valores e a manipulação dos sinais, como é melhor explicado na seção 2.3.

Figura 6 – Joystick KY-023



Fonte: [Datasheets-pdf \(2017\)](#).

Figura 7 – Resistores para o eixo x



Fonte: [Datasheets-pdf \(2017\)](#).

## 2.2. ATUADORES

Os motores utilizados para este protótipo foram um Servo motor *Tower Pro SG90* e Motor elétrico *Akyiama AK360/25PL12S3500S*.

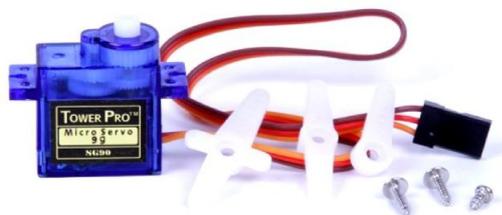
### 2.2.1. SERVOMOTOR *TOWER PRO SG90*

A [Figura 8](#) apresenta um servo motor que foi usado com a finalidade de simular o movimento de um sistema de direção, atuando como o sistema de auxílio de direção acoplado à cremalheira, porém sem o uso da barra de direção.

O servomotor SG90 é constituído por três entradas que têm a seguinte função de acordo com [Imperial-Colege-London \(s.d\)](#):

- Vermelho: Alimentação 5V

Figura 8 – Servo motor SG90



Fonte: [Imperial-Colege-London](#) (s.d.).

- Marrom: Massa elétrica
- Laranja: Sinal PWM para ativação do motor

O sinal que contém a informação da angulação requerida pelo *joystick* é recebido através do barramento CAN, o controlador envia o sinal PWM, através de um dos pinos PWM, para que o servomotor se mova para a posição requerida.

### 2.2.2.MOTOR ELÉTRICO

O motor elétrico empregado foi o motor *Akyiama* AK360/25PL12S3500S ([Figura 9](#)). Este motor trabalha com alimentação de 12V nominal com consumo de 40mA e rotação de 3500RPM com o eixo livre de acordo com [Robocore](#) (s.d.).

Figura 9 – Motor elétrico



Fonte: [Robocore](#) (s.d.).

Esse motor foi escolhido pela simplicidade de uso e instalação, necessitando para seu completo uso de uma fonte de alimentação 12V. Assim foi utilizada uma fonte regulável, como a da [Figura 10](#), e um *driver* de acionamento de motor que será mencionado na seção [2.3](#).

O funcionamento desse motor se dá de forma similar ao servo motor citado anteriormente. Os sinais do eixo y do comando *joystick* são enviados através do barramento

Figura 10 – Fonte regulável



Fonte: Autoria própria.

CAN, esses sinais são compostos pela graduação de velocidade requerida e o sentido de rotação do motor.

### 2.3.CONTROLADORES

Para a realização deste trabalho foram utilizados os seguintes controladores:

- 2 placas Arduino *Leonardo*;
- 2 módulos MCP2515;
- 1 *driver* ponte h L298N.

#### 2.3.1.ARDUINO *LEONARDO*

Essa placa foi escolhida por ter as funções necessárias para o funcionamento do sistema simulador, a saber:

- comunicação SPI necessário para programar o *driver* MCP2515;
- entradas analógicas para leitura dos valores do *joystick*;
- saídas PWM para ativação dos atuadores;
- saídas digitais para ativar o sentido de rotação do motor elétrico.

Além disso, existe a facilidade de implementação do código do sistema, facilidade de encontrar dispositivos compatíveis com o sistema e o baixo custo para o desenvolvimento do sistema simulador. No sistema desenvolvido nesse trabalho foram usadas duas placas, uma é responsável pela leitura do *joystick* e atua como emissor das mensagens para o barramento CAN, e a outra placa é responsável por realizar o ativamento dos atuadores e é o receptor das mensagens CAN no modelo construído. A placa escolhida é semelhante a apresentada na figura [Figura 11](#).

Figura 11 – Arduino usado no projeto



Fonte: [Bau-Da-Eletronica](#) (s.d.).

Foram utilizados os pinos do ICSP do Arduino para que fosse possível realizar a comunicação CAN, utilizando para isso duas placas MCP2515 conectadas a esses pinos.

Levando-se em conta a placa emissora, além de conectar os pinos do ICSP, foram conectados, também, os pinos 10, como ativador do *chip* escravo para a comunicação serial, A4 e A5, responsáveis pelas leituras analógicas do *joystick* e também os pinos de alimentação e massa para alimentação do *joystick*.

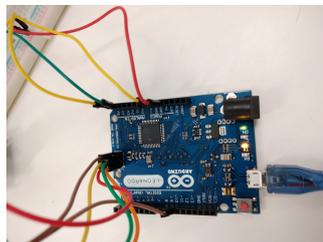
Para a placa receptora foram utilizados além dos pinos de alimentação e comunicação serial, o pino 6, para acionamento do servo motor, o pino 3, para acionamento do motor elétrico e os pinos 5 e 2 para indicar o sentido de rotação do motor.

Assim as duas placas ficaram conectadas como mostra o esquemático apresentado em [Figura 18](#) e em detalhes a [Figura 12](#) para a placa emissora e [Figura 13](#) para a placa receptora.

### 2.3.2.MCP2515

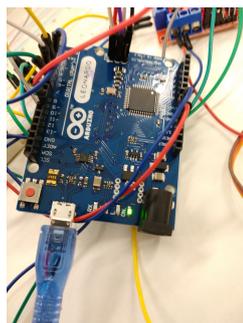
Para trazer o protótipo mais próximo do universo automotivo real, e também por ter utilizado esse dispositivo em disciplina ministrada durante o curso, o controlador escolhido foi o MCP2515, como o mostrado na [Figura 14](#).

Figura 12 – Placa emissora



Fonte: Autoria própria.

Figura 13 – Placa receptora



Fonte: Autoria própria.

Figura 14 – MCP2515



Fonte: [Filipeflop](#) (s.d.).

Esse controlador comporta o protocolo CAN versão 2.0B com velocidade de comunicação de até 1Mbps, funciona com alimentação de 5V e já possui, embarcado no módulo, o resistor de 120 ohms necessário para o casamento de impedância requisitado para o barramento de comunicação. Além de ter o controlador MCP2515 há na placa, o transceptor CAN TJA1050 que é responsável pela comunicação com o barramento CAN.

Na configuração destes controladores foram usados os seguintes pinos:

- VCC ao pino de alimentação do arduino;
- GND ao pino de massa do arduino;
- CS ao pino 10 do arduino - pino responsável pela ativação da placa na comunicação serial;
- SO ao pino 1 do ICSP (MISO) - pino que envia mensagens para o arduino;
- SI ao pino 4 do ICSP (MOSI) - pino que envia mensagens para o MCP2515;
- SCK ao pino 3 do ICSP (SCK) - pino responsável pelo *clock* para sincronização das mensagens.

Os dados lidos no arduino emissor são serializados e enviados através do pino MOSI para a placa MCP2515, essa mensagem serializada é transcrita de acordo com os padrões diferenciais da rede CAN. Essa mensagem é enviada pelo par de fios CAN até o módulo MCP2515 da placa receptor, onde está o transceptor irá desempacotar a mensagem e enviá-la através do pino MISO para que seja usado.

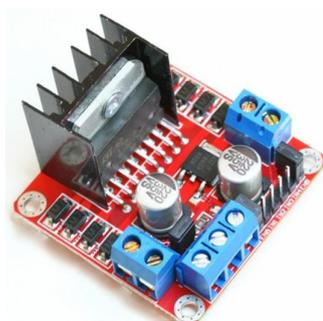
### 2.3.3.L298N

Para a ativação do motor elétrico de 12V com variação de sentido e velocidade fez-se necessário a utilização de um módulo ponte H, sendo escolhido, por familiaridade, o módulo com o controlador L298N ([Figura 15](#)).

Esse módulo é alimentado externamente com auxílio da fonte apresentada em [2.2](#) nos pinos +12V e GND. O sentido de rotação é regulado ao setar valor lógico 1 em um dos pinos de entrada (*in 1* ou *in 2*) e valor lógico 0 no pino de entrada complementar.

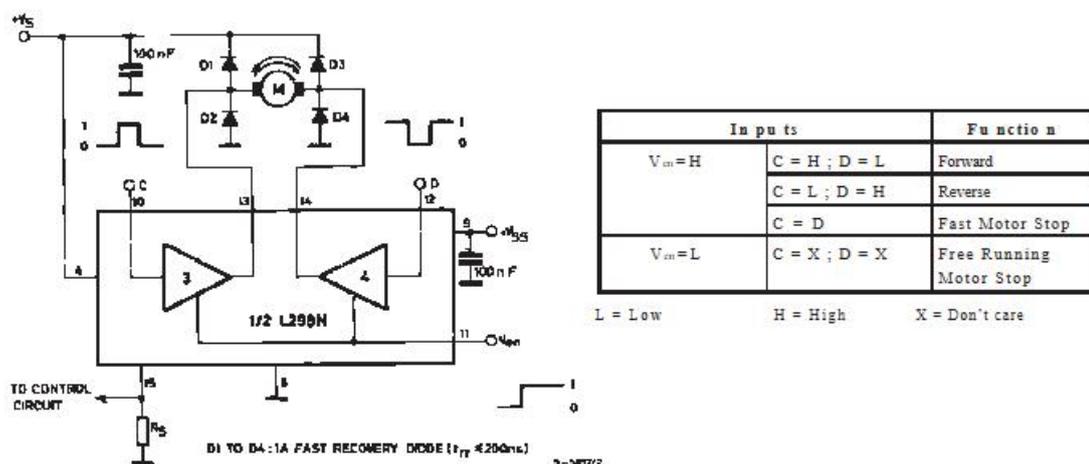
Como mencionado em [2.3.1](#) foram ligados os pinos *in 1* e *in 2* aos pinos 3 e 5 do Arduino, logo esses pinos são os responsáveis pelo sentido de rotação do motor elétrico.

Figura 15 – L298N



Fonte: [Bau-Eletronica \(s.d.\)](#).

Figura 16 – Lógica de ativação

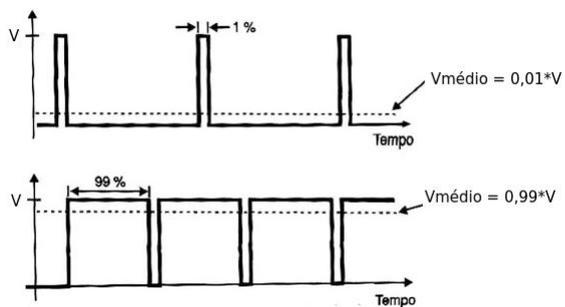


Fonte: [Alldatasheet \(2000\)](#).

Já o pino 3 do arduino foi ligado ao pino ENA do *driver* L298N, desse modo é possível controlar a velocidade de rotação do motor pela ativação do sinal PWM. Essa ativação PWM controla a velocidade do motor, isso porque quanto maior o *duty cycle* (porcentagem de tempo com sinal em nível lógico alto) maior será a tensão entregue ao motor, logo maior será a velocidade de rotação do motor e vice-versa ([Figura 17](#)).

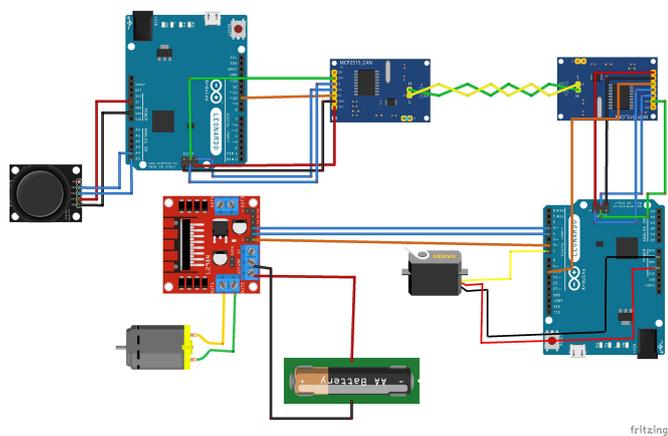
Ao final o sistema montado ficou similar ao do esquemático apresentado em [Figura 18](#), como é possível ver no circuito após montagem, que é apresentado em [Figura 19](#).

Figura 17 – Controle de velocidade



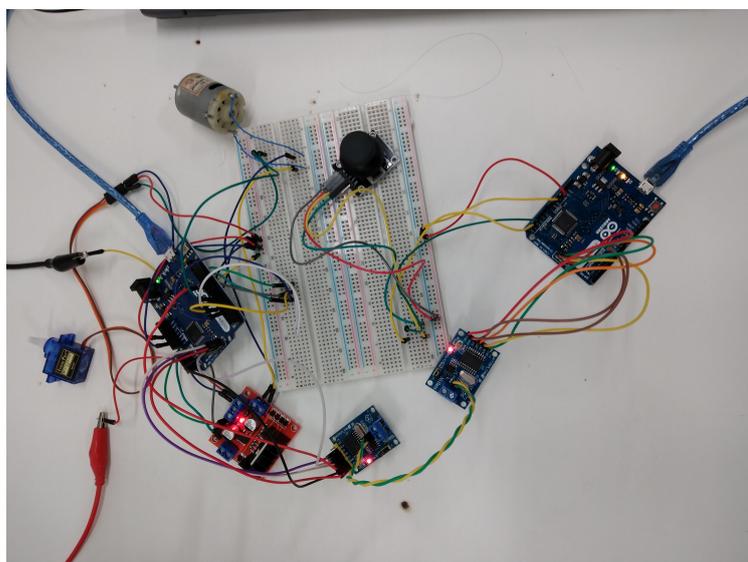
Fonte: Filipeflop (2019).

Figura 18 – Esquemático da topologia



Fonte: Autoria própria.

Figura 19 – Circuito montado



Fonte: Autoria própria.

### 3.COMUNICAÇÃO E SOFTWARE

A partir da montagem do *hardware* concluída, pôde-se concentrar os esforços na parte de comunicação CAN e no tratamento dos dados por via de *software*.

Este capítulo abordará como funciona a comunicação através do barramento CAN, também serão abordados como que os dados foram obtidos e manipulados para que o protótipo funcionasse bem como os resultados obtidos durante os ensaios.

#### 3.1.COMUNICAÇÃO CAN

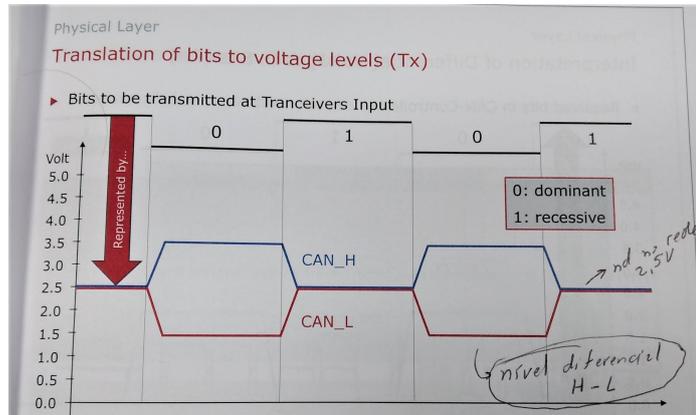
Com o aumento do número de componentes elétricos em um veículo foi necessário uma organização estrutural com a finalidade de simplificação da arquitetura eletro-eletrônica do veículo. Atualmente, em um barramento, já é possível encontrar mais de 70 módulos inter conectados. Isso eliminou o número de erros, o tamanho e peso do chicote elétrico, uma vez que cada comunicação era realizada ponto a ponto, ou seja cada função tinha um fio específico para cada troca de informação e hoje isso é minimizável.

O protocolo CAN está inserido nos níveis 1 e 2 no padrão de referência ISO-OSI, que diz respeito a ter a camada física, no caso os fios de comunicação e a camada de comunicação de dados, ou seja, paginação, endereçamento, acesso ao barramento, sincronização e proteção dos dados.

A comunicação CAN se dá através de uma sequência de bits como cita [Vector \(2016\)](#). A representação desses bits ocorre como demonstrado em [Figura 20](#).

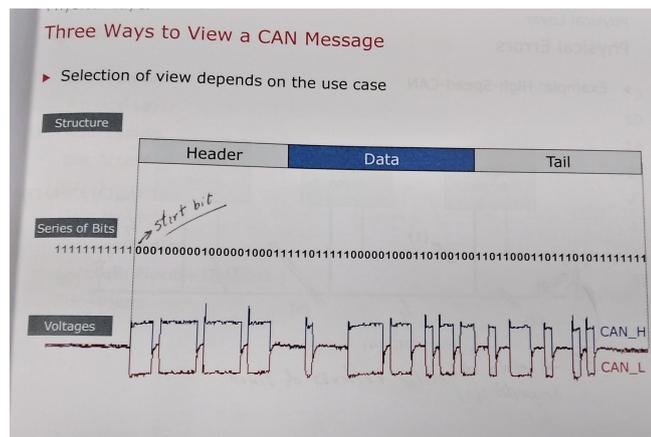
Na [Figura 20](#) é possível ver que quando o bit é 0, existe a tensão diferencial no par CAN High e CAN Low de aproximadamente 2V entre os fios, pois em CAN High teremos aproximadamente 3,5V e em CAN Low aproximadamente 1,5V. Já quando o bit é 1, não há diferença entre o par de fios e ambos terão aproximadamente 2,5V. De tal forma, ao fim, uma mensagem pode ser verificada no osciloscópio como mostra a [Figura 21](#) e como mostra a

Figura 20 – Bits CAN



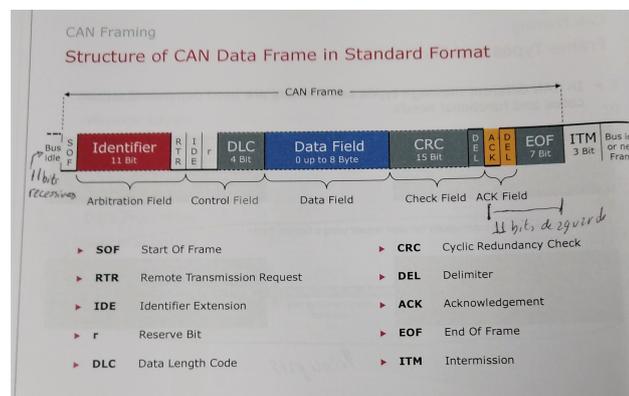
Fonte: Vector (2016).

Figura 21 – Mensagem CAN



Fonte: Vector (2016).

Figura 22 – Estrutura de uma mensagem CAN



Fonte: Vector (2016).

Uma mensagem CAN é constituído da forma apresentada na [Figura 22](#). Dentre esses pacotes os mais relevantes são o identificador, composto por 11 *bits* sendo quem carrega a informação de identificação e prioridade da mensagem. Essa prioridade é responsável por escalonar a ordem de acesso das mensagens no barramento, sendo que ela decresce com o aumento do valor. Assim mensagens com identificador mais baixo tem maior prioridade. O DLC é o campo responsável por indicar qual o tamanho do campo de dados, possui 4 *bits* que indicam quantos *bytes* o campo de dados possui, assim o campo de dados pode ter de 0 a 8 *bytes*. O campo de dados é onde os dados da mensagem estão contidos.

Nesse projeto foram atribuídos, por somente haverem dois nós na comunicação, o identificador 0xA25 com DLC de 3 *bytes*. A partir desses conhecimentos passou-se a programação do software para que fosse possível a execução da tarefa proposta.

## 3.2.SOFTWARE

O código para a aplicação pode ser dividido em código feito para a placa emissora e código para a placa receptora. Como base para a codificação final foram levados em conta códigos exemplos, a saber:

- leitura de *joystick* e ativação de servo motor [Filipeflop \(2014\)](#);
- leitura de *joystick* e ativação de motor elétrico [How-To-Mechatronics \(s.d.\)](#);
- Diálogo CAN entre duas placas Arduino usando MCP2515 [circuitdigest \(2019\)](#).

### 3.2.1.1.EMISSOR

A placa emissora, além da função de enviar a mensagem CAN tem a função de fazer a aquisição dos sinais do *joystick* e realizar o mapeamento do sinal.

Os valores dos sinais do *joystick* são obtidos através da leitura analógica das portas, armazenando esses valores em duas variáveis do tipo inteiro (`val_x` e `val_y`). Com esses valores são feitas manipulações para cada eixo. O eixo x (relativo ao movimento do servo motor) o sinal do *joystick* que tem sua variação de 0 a 1024 é mapeado dentro dos limites de angulação do servo motor, ou seja é colocado nos limites ente 1 e 180 e armazenada na variável inteira `valor_x`.

O eixo y (relativo ao acionamento do motor elétrico), foi criado primeiramente um intervalo referente a posição de repouso do *joystick*, isso devido a aspectos construtivos que podem fazer variar o nível de tensão dessa posição, assim esse intervalo foi definido como entre 470 e 550. Se estiver entre esses valores, os valores de sentido (referente ao sentido de rotação) e valor\_y (referente a atuação) recebem zero, caso o valor de val\_y seja menor que 470, a variável sentido recebe o valor 2 (indicando rotação para trás) e a variável valor\_y recebe o mapeamento de 470 a 0 em 0 a 255, isso ocorre porque o motor deverá acelerar com o decréscimo desse sinal, quando em posição reversa. Finalmente quando o valor de val\_y for maior que 550, a variável sentido receberá o valor 1 (indicando rotação para frente) e a variável valor\_y recebe o mapeamento de 550 a 1023 em 0 a 255, para que a aceleração ocorra com o aumentar do valor do joystick.

Para o funcionamento da mensagem CAN primeiramente foram adicionadas as bibliotecas para comunicação SPI (SPI.h) e para o módulo MCP2515 (mcp2515.h) disponíveis em [circuitdigest \(2019\)](#). A comunicação serial SPI é iniciada e logo são colocados os parâmetros de inicialização do barramento CAN para o módulo MCP2515. Com o barramento inicializado foi definido o identificador da mensagem CAN, nesse caso 0xA25, o comprimento da mensagem, definido em 3 bytes para esse projeto e os valores dos eixos e sentido de rotação são armazenados nos respectivos bytes da mensagem, após isso a mensagem é enviada para o barramento.

### 3.2.2.2. RECEPTOR

A placa receptora recebe a mensagem emitida, decodifica os dados inseridos no barramento e faz a ativação dos atuadores de acordo com esses dados.

Para a função de decodificação são utilizadas as bibliotecas de comunicação SPI e do módulo MCP2515. A ativação da comunicação SPI e do barramento CAN ocorre da mesma do envio das mensagens. No receptor o módulo MCP2515 monitora o barramento e verifica a ocorrência da mensagem. Quando a mensagem está disponível os dados de cada posição, pré definida no envio, são armazenados em variáveis do tipo inteiro eixo\_x, eixo\_y e sentido.

Para a função de ativação dos atuadores foi utilizada a biblioteca VarSpeedServo.h disponível em [Filipeflop \(2014\)](#). Com a inserção do dado eixo\_x é feita a movimentação do servo até a posição correta. para o acionamento do motor foram utilizados os dados eixo\_y

e sentido. O motor no instante de inicialização é colocado como parado, colocando as portas de acionamento do *driver* ambas em nível lógico 0. Dependendo do valor do sentido de rotação (1 ou 2) as portas de entrada do *driver* são ativadas fazendo o motor girar para o sentido requisitado, com a velocidade definida pelo dado eixo\_y. Caso a variável esteja em nível lógico 0 o motor mantém-se parado.

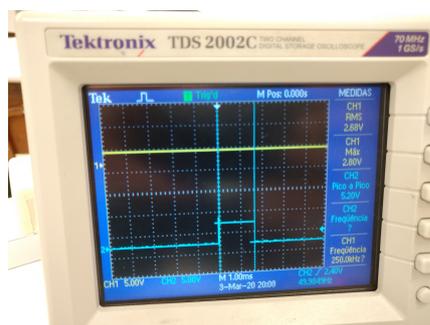
### 3.3.EXPERIMENTOS E DISCUSSÕES

No trabalho foi proposto realizar um sistema que simulasse, em pequena escala, o processo de movimentação de um veículo elétrico sendo comandado através de um *joystick*.

Após a montagem do sistema, e a programação no ambiente de desenvolvimento Arduino, foram verificados os sinais do *joystick*, do sinal de ativação do servo motor, do sinal de ativação e de velocidade do motor elétrico, bem como verificou-se a mensagem CAN.

Com o sistema protótipo da Figura 19 foram feitas comparações dos sinais joystick eixo x com sinal de atuação do servo motor e joystick eixo y com o sinal de atuação do motor elétrico em posições intermediárias (Figura 24, Figura 27, Figura 30 e Figura 28), no final do curso (Figura 25, Figura 29, Figura 31 e Figura 32) e na posição neutra (Figura 23). Como é possível observar nas imagens do osciloscópio o tempo do sinal em 5V varia com a movimentação do *joystick*.

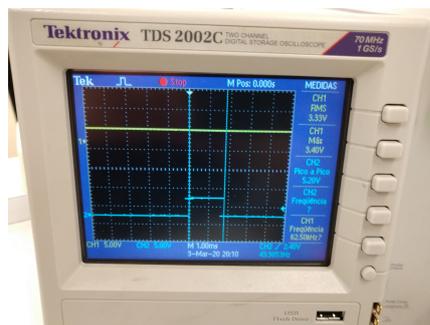
Figura 23 – *Joystick* eixo x posição neutra



Fonte: Autoria própria.

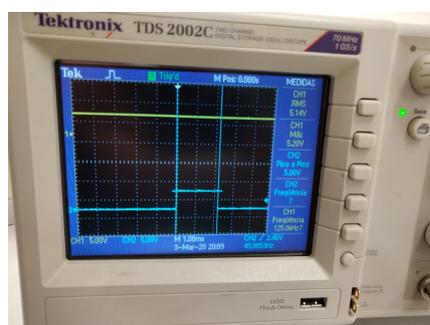
Foi observado que a variação de tensão do *joystick* escolhido não é linear até o fim do curso, logo os valores de velocidade e de direção podem sofrer algumas descontinuidades devido a variação rápida do valor de tensão do *joystick*, assim não apresentando muitas posições intermediárias.

Figura 24 – Joystick eixo x posição intermediária a direita



Fonte: Autoria própria.

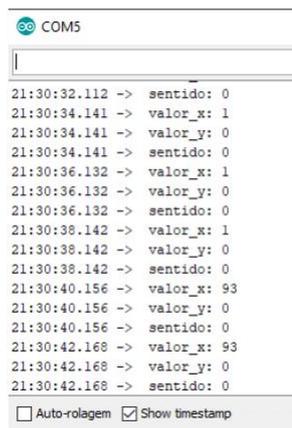
Figura 25 – Joystick eixo x posição final a direita



Fonte: Autoria própria.

Também foi observado que o sistema Arduino algumas vezes apresentava demora em fazer a leitura do barramento CAN, isso devido ao processo que o dado necessitava ser submetido, serialização, empacotamento, transcrição, interpretação, desempacotamento, desserialização, muitas vezes isso acarretava um atraso de 1 segundo entre as leituras, como mostrado na [Figura 26](#). Esse desafio foi contornado e o sistema então respondeu adequadamente logo que acionado.

Figura 26 – Tempos de leitura serial



```
COM5
21:30:32.112 -> sentido: 0
21:30:34.141 -> valor_x: 1
21:30:34.141 -> valor_y: 0
21:30:34.141 -> sentido: 0
21:30:36.132 -> valor_x: 1
21:30:36.132 -> valor_y: 0
21:30:36.132 -> sentido: 0
21:30:38.142 -> valor_x: 1
21:30:38.142 -> valor_y: 0
21:30:38.142 -> sentido: 0
21:30:40.156 -> valor_x: 93
21:30:40.156 -> valor_y: 0
21:30:40.156 -> sentido: 0
21:30:42.168 -> valor_x: 93
21:30:42.168 -> valor_y: 0
21:30:42.168 -> sentido: 0
 Auto-rolagem  Show timestamp
```

Fonte: Autoria própria.

## 4.CONCLUSÃO

Ao fim desse projeto ficou evidente que é possível realizar a movimentação de um carro elétrico usando um *joystick*, mesmo que em pequena escala foi possível provar que um comando *joystick* atrelado a rede CAN consegue acionar um motor elétrico e controlar um sistema de direção.

Somente foi montado um sistema protótipo de bancada, não colocando o em um veículo, pois o objetivo foi observar o funcionamento profícuo do sistema fazendo a leitura de um *joystick* e acionando os motores, não necessitando de um veículo propriamente dito.

O sistema apresentou algumas pequenas dificuldades como a não linearidade do *joystick* escolhido e alguns atraso no tratamento dos dados. Contudo como elemento didático que demonstra a possibilidade para o mundo automotivo do uso dessa inovação, que o protótipo tem resultados a contento.

## REFERÊNCIAS

AIRBUS. 2014. *Site* da construtora de aviões Airbus. Disponível em: <<https://www.airbus.com/aircraft/passenger-aircraft/cockpits.html>>. Acesso em: 29 fev. 2020. 11

ALL-ABOUT-CIRCUITS. 2019. *Site* com conceitos sobre barramento CAN. Disponível em: <<https://www.allaboutcircuits.com/technical-articles/introduction-to-can-controller-area-network/>>. Acesso em: 25 fev. 2020. 10

ALLDATASHEET. 2000. *Datasheet* para o L298N. Disponível em: <<https://www.alldatasheet.com/datasheet-pdf/pdf/22440/STMICROELECTRONICS/L298N.html>>. Acesso em: 02 mar. 2020. 20

BAU-DA-ELETRONICA. s.d. *Site* de venda de componentes eletrônicos. Disponível em: <<https://www.baudaeletronica.com.br/placa-leonardo-r3.html>>. Acesso em: 02 mar. 2020. 17

BAU-ELETRONICA. s.d. *Site* de venda de componentes eletrônicos. Disponível em: <<https://www.baudaeletronica.com.br/driver-motor-ponte-h-l298n.html>>. Acesso em: 02 mar. 2020. 20

CIRCUITDIGEST. 2019. *Site* com informações referente a comunicação CAN com Arduino. Disponível em: <<https://circuitdigest.com/microcontroller-projects/arduino-can-tutorial-interfacing-mcp2515-can-bus-module-with-arduino>>. Acesso em: 10 fev. 2020. 24, 25

DATASHEETS-PDF. 2017. *Datasheet* para o joystick KY-023. Disponível em: <<https://datasheetspdf.com/pdf/1402034/Joy-IT/KY-023/1>>. 13, 14

FILIPEFLOP. 2014. *Site* que demonstra o controle de servo motor. Disponível em: <<https://www.filipeflop.com/blog/controlando-micro-servo-9g-usando-joystick/>>. Acesso em: 10 fev. 2020. 24, 25

FILIFEFLOR. 2019. Conceito de PWM com Arduino. Disponível em: <<https://www.filifefflor.com/blog/control-motor-dc-pwm-arduino-uno/>>. Acesso em: 02 mar. 2020. 21

FILIFEFLOR. s.d. *Site* de vendas de módulos e conceitos para arduino. Disponível em: <<https://www.filifefflor.com/produto/modulo-can-bus-mcp2515-tja1050/#tab-wc-simulador-parcelas>>. Acesso em: 02 mar. 2020. 18

HOW-TO-MECHATRONICS. s.d. *Site* que demonstra o controle de motor elétrico. Disponível em: <<https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>>. Acesso em: 10 fev. 2020. 24

IMPERIAL-COLEGE-LONDON. s.d. *Datasheet* para o servomotor SG90. Disponível em: <[http://www.ee.ic.ac.uk/pcheung/teaching/DE1\\_EE/stores/sg90\\_datasheet.pdf](http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf)>. Acesso em: 01 mar. 2020. 14, 15

NI. 2019. *Site* da *National Instruments* apresentando protocolo CAN. Disponível em: <<https://www.ni.com/pt-br/innovations/white-papers/06/controller-area-network--can--overview.html>>. Acesso em: 25 fev. 2020. 10

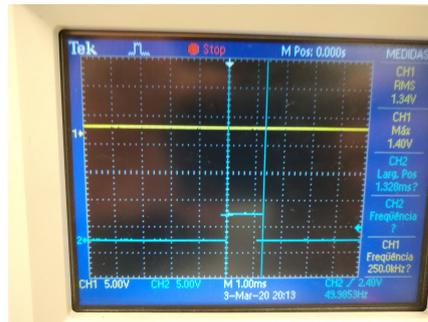
ROBOCORE. s.d. *Datasheet* para o motor elétrico. Disponível em: <[https://s3-sa-east-1.amazonaws.com/robocore-lojavirtual/963/DataSheet\\_AK360.pdf](https://s3-sa-east-1.amazonaws.com/robocore-lojavirtual/963/DataSheet_AK360.pdf)>. Acesso em: 01 mar. 2020. 15

STATITA. 2019. *Site* com gráfico da evolução do custo de veículo. Disponível em: <<https://www.statista.com/statistics/277931/automotive-electronics-cost-as-a-share-of-total-car-cost-worldwide/>>. Acesso em: 25 fev. 2020. 9

VECTOR. 2016. *CAN Controller Area Network - Fundamentals*. 22, 23

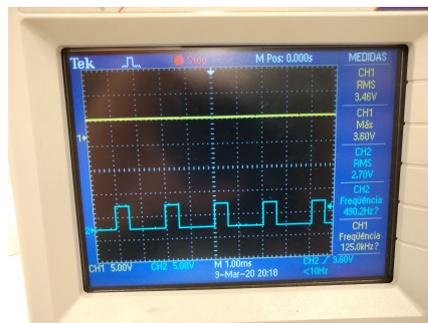
## A.IMAGENS DOS TESTES

Figura 27 – *Joystick* eixo x posição intermediária a esquerda

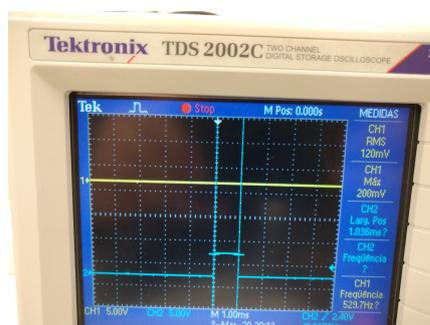


Fonte: Autoria própria.

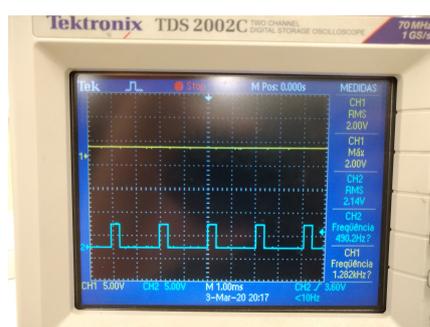
Figura 28 – *Joystick* eixo y posição intermediária reverso



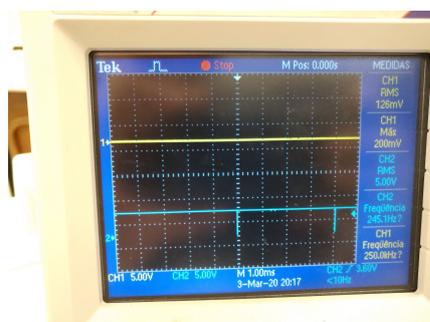
Fonte: Autoria própria.

Figura 29 – *Joystick* eixo x posição final a esquerda

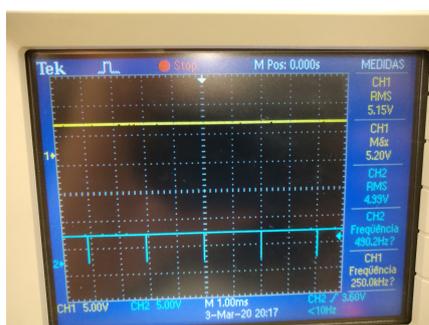
Fonte: Autoria própria.

Figura 30 – *Joystick* eixo y posição intermediária a frente

Fonte: Autoria própria.

Figura 31 – *Joystick* eixo y posição final a frente

Fonte: Autoria própria.

Figura 32 – *Joystick* eixo y posição final reverso

Fonte: Autoria pr&amp;#x225;pria.