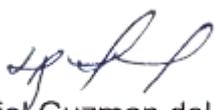


UNIVERSIDADE DO ESTADO DO AMAZONAS – UEA
ESCOLA SUPERIOR DE TECNOLOGIA – EST
CURSO DE ENGENHARIA ELÉTRICA

EWERTON CASSIANO XAVIER

**DESENVOLVIMENTO DE SISTEMA DE TROCA DE MARCHAS VIA
BORBOLETA PARA CARROS FÓRMULA SAE**


Daniel Guzman del Rio
Orientador. 23/12/2021

MANAUS

2021

EWERTON CASSIANO XAVIER

**DESENVOLVIMENTO DE SISTEMA DE TROCA DE MARCHAS VIA
BORBOLETA PARA CARROS FÓRMULA SAE**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Daniel Guzman del Rio

MANAUS

2021

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia - EST

Reitor:

Cleinaldo de Almeida Costa

Vice-Reitor:

Cleto Cavalcante de Souza Leal

Diretor da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo

Coordenador do Curso de Engenharia Elétrica:

Israel Gondres Torné

Banca Avaliadora composta por:

Prof. Dr. Daniel Guzmán del Río

Prof. Dr. Jozias Parente de Oliveira

Prof. Dr. Edgard Luciano Oliveira da Silva

Data da defesa: 29/12/2021.

(Orientador)

CIP – Catalogação na Publicação

Xavier, Ewerton Cassiano

Desenvolvimento de Sistema de Troca de Marchas via Borboleta para Carros Fórmula SAE / Ewerton Cassiano Xavier; [orientado por] Daniel Gúzman del Río. – Manaus: 2021.

59 p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica). Universidade do Estado do Amazonas, 2021.

1. Troca de marchas, 2. Software Embarcado, 3. Acionamentos Pneumáticos, 4. Otimização de Processos. I. Rio, Daniel Guzman del Río.

EWERTON CASSIANO XAVIER

**DESENVOLVIMENTO DE SISTEMA DE TROCA DE MARCHAS VIA
BORBOLETA PARA CARROS FÓRMULA SAE**

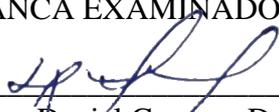
Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletricista.

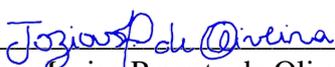
Nota obtida: 9.03 (Nove pontos e três centésimos)

Aprovada em 29 / 12 / 2021.

Área de concentração: Software Embarcado

BANCA EXAMINADORA


Orientador: Daniel Guzman Del Río, Dr.


Avaliador: Jozias Parente de Oliveira, Dr.


Avaliador: Edgard Luciano Oliveira da Silva, Dr.

Manaus 2021

AGRADECIMENTOS

Agradeço primeiramente a Deus, pelo dom da vida e conhecimento.

Agradeço ao meu pai, Carlos Xavier, e minha mãe, Mariza Cassiano, por todos os ensinamentos e oportunidades que me proporcionaram ao longo dos anos.

Agradeço ao meu orientador, professor Daniel Guzman, por ter aceitado o convite.

Agradeço aos meus amigos Victor Dias e João Valério por todo o suporte durante os anos de equipe Viper Racing.

Por fim, agradeço aos meus amigos James Monteiro, Kaiky Shinji, Matheus Nery, Italo Santos, Kassio Lima, Samuel Marinho e demais colegas por todo o apoio durante a faculdade.

RESUMO

Os carros que participam da competição Fórmula SAE são reconhecidos pela agilidade em curvas e alta aceleração partindo do repouso, sendo tais características desenvolvidas através de um longo trabalho entre as várias áreas que compõem o veículo. Uma delas, apesar de menos aparente, contribui de forma significativa para um melhor desempenho, sendo este o sistema de troca de marchas. Este trabalho tem como objetivo o desenvolvimento de um sistema de troca de marchas simplificado, visando melhorias de caráter ergonômico para o piloto e de desempenho para a equipe que adotar a estrutura definida ao longo do texto, tendo em mente o menor custo possível e facilidade de implementação/replicação do sistema. Para atingir este fim, a pesquisa ocorreu através do desenvolvimento de código responsável pelo processo de troca de marchas e testes funcionais por meio da plataforma de simulação de *hardware* e *firmware* Proteus, contendo transistores de efeito de campo, relés de estado sólido e plataforma microcontrolada (Arduino). Como resultado, o sistema apresentou funcionamento esperado durante todas as simulações efetuadas.

Palavras-chave: Troca de marchas, Software Embarcado, Acionamentos Pneumáticos, Otimização de Processos

ABSTRACT

Cars that attend Formula SAE competition are recognized by the agility in curves and high acceleration from stand position, being such characteristics developed along an extensive work among many areas that comprises the vehicle. One of them, even though not outstanding, contributes significantly for a better performance, being the gear shifting system. This work has the goal of developing a simplified gear shifting system, aiming improvements related to pilot ergonomics and performance of the teams that adopt the structure established along the content, always having in mind the lowest cost possible and implementation/replication system easiness. To achieve this goal, the research happened through the development of a code responsible for gear changing process (paddle shift) and functional tests at hardware and firmware simulation platform called Proteus, comprising metal oxide field effect transistors, solid state relays and microcontrolled platform (Arduino). As a result, the system presented expected operation in all simulations made.

Keywords: Paddle Shift, Embedded Software, Pneumatic Trigger, Process Optimization.

LISTA DE FIGURAS

FIGURA 1 - LOGOTIPO DA SAE.....	6
FIGURA 2 - FOTO DA EQUIPE TU GRAZ DURANTE O ENDURANCE DA COMPETIÇÃO INTERNACIONAL DA SAE, EM MICHIGAN.....	7
FIGURA 3 - VOLANTE DE VEÍCULO FÓRMULA COM BORBOLETA EM SUA PARTE TRASEIRA.....	8
FIGURA 4 - MODELO DE BORBOLETA DE VOLANTE	9
FIGURA 5 - APLICAÇÃO DA ECU MOTEC EM SISTEMA DE TROCA DE MARCHAS ELÉTRICO.....	10
FIGURA 6 - ECU AUTRONIC COM CONTROLE DE ROTAÇÃO DO MOTOR E CORTE DE INJEÇÃO/IGNIÇÃO	10
FIGURA 7 - ESTRUTURA DO SISTEMA DE TROCA DE MARCHAS UTILIZANDO BORBOLETAS DA MOTEC - M1	12
FIGURA 8 - ITENS OBRIGATÓRIOS/OPCIONAIS NO SISTEMA DE TROCA DE MARCHAS DA MOTEC M1.	12
FIGURA 9 - INFORMAÇÕES UTILIZADAS PARA DEFINIR O TEMPO DE MARCHA DAS TROCAS	13
FIGURA 10 - ESQUEMA DE POLARIZAÇÃO DO RELÉ RESPONSÁVEL POR ALIMENTAR A BOMBA DE AR DO SISTEMA PNEUMÁTICO	13
FIGURA 11 - EXEMPLO DE POSICIONAMENTO DO ATUADOR PNEUMÁTICO EM UM MOTOR	14
FIGURA 12 - DASHBOARD UTILIZADO PELA EQUIPE DA UNIVERSIDADE DE DELAWARE PARA OBTENÇÃO E EXIBIÇÃO DE DADOS AO PILOTO.....	15
FIGURA 13 - KIT BÁSICO DA QUICKSHIFTER PARA TROCA DE MARCHAS EM MOTOCICLETAS	16
FIGURA 14 - SISTEMA HIDRÁULICO DE TROCA DE MARCHA	17
FIGURA 15 - CÂMBIO MANUAL	18
FIGURA 16 - ESQUEMA DE UM CONJUNTO DE EMBREAGEM.....	19
FIGURA 17 - AMPLIFICADORES: (A) CONTROLADOS POR CORRENTE E (B) CONTROLADOS POR TENSÃO.....	19
FIGURA 18 - FORMAÇÃO DO CANAL NO MOSFET TIPO DEPLEÇÃO DE CANAL N	20
FIGURA 19 - MODOS DE OPERAÇÃO DO MOSFET TIPO DEPLEÇÃO DE CANAL N21	

FIGURA 20 - FORMAÇÃO DO CANAL NO MOSFET TIPO INTENSIFICAÇÃO DE CANAL N	21
FIGURA 21 - CURVAS CARACTERÍSTICAS DE DRENO DE UM MOSFET TIPO INTENSIFICAÇÃO DE CANAL N COM $V_T = 2V$ E $K = 0,278 \times 10^{-3} A/V^2$	22
FIGURA 22 - REPRESENTAÇÃO DE ARDUINO UNO.....	23
FIGURA 23 - IMAGEM DA IDE DO ARDUINO.....	24
FIGURA 24 - SÍMBOLO DA LINGUAGEM C	25
FIGURA 25 - ESQUEMA PARCIAL DE CIRCUITO DO SISTEMA DE TROCA DE MARCHAS CONSTRUÍDO NO SOFTWARE PROTEUS	27
FIGURA 26 - CIRCUITO POLARIZADOR DE VÁLVULA SOLENOIDE.....	27
FIGURA 27 - SSR MODELO 100DD 5-220VDC	28
FIGURA 28 - IMAGEM DA VÁLVULA SOLENOIDE 3V210-08-12V	29
FIGURA 29 - FORÇAS MÍNIMAS APLICADAS PARA TROCA DE MARCHA NO MOTOR DA EQUIPE UCF.....	29
FIGURA 30 - ATUADOR PNEUMÁTICO DE DUPLA AÇÃO VORTEX 20 MM X 60 MM	30
FIGURA 31 - INICIALIZAÇÃO DE PINOS DIGITAIS ATRAVÉS DO MÉTODO SETUP() DO ARDUINO.....	33
FIGURA 32 - BILL OF MATERIAL PARA CONSTRUÇÃO BÁSICA DO CIRCUITO DE TROCA DE MARCHAS	35
FIGURA 33 - REPRESENTAÇÃO DE UM CIRCUITO PUSH-DOWN PARA POLARIZAÇÃO DOS PINOS DIGITAIS DO ARDUINO.....	36
FIGURA 34 - CIRCUITO RESPONSÁVEL POR POLARIZAR O RELÉ DE ESTADO SÓLIDO	37
FIGURA 35 - CONFIGURAÇÃO DO ARDUINO 328P UTILIZADO NA SIMULAÇÃO DA PLATAFORMA PROTEUS	38
FIGURA 36 - CIRCUITO PARA VERIFICAÇÃO DE ESTADO ATUAL DA EMBREAGEM, SUBIDA E DESCIDA DE MARCHA.....	38
FIGURA 37 - SINALIZADOR DE FUNCIONAMENTO DA EMBREAGEM.....	39
FIGURA 38 - EMBREAGEM E ATUADOR EM POSIÇÃO DE REDUÇÃO DA MARCHA (REPRESENTAÇÃO) ATIVADOS	39
FIGURA 39 - CIRCUITO DE ALIMENTAÇÃO DO ATUADOR RESPONSÁVEL POR REDUZIR A MARCHA EM ESTADO ATIVO	40

LISTA DE ABREVIATURAS E SIGLAS

CAD – *Computer Aided Design*;

ECU – *Electronic Control Unit*;

FSAE – *Formula of Society of Automotive Engineering*;

IDE – *Integrated Development Environment*;

LED – *Light Emitting Diode*;

MOSFET – *Metal Oxide Semiconductor Field Effect Transistor*

PCB – *Printed Circuit Board*;

SAE – *Society of Automotive Engineering* (Em português, Sociedade de Engenharia Automotiva)

Sumário

INTRODUÇÃO	5
1 REFERENCIAL TEÓRICO	6
1.1 SAE – SOCIETY OF AUTOMOTIVE ENGINEERING	6
1.2 SAE – COMPETIÇÃO.....	6
1.3 CÂMBIO BORBOLETA (PADDLE SHIFT).....	8
1.4 SISTEMAS EXISTENTES NO MERCADO	9
1.4.1 Circuitos com MoTec M400 ECU	9
1.4.2 ECU Autronic	10
1.4.3 MoTeC M1 PaddleShift	11
1.4.4 Sistema de Aquisição de Dados	14
1.4.5 Sistema QUICKSHIFTER de troca de marcha	15
1.4.6 Sistema hidráulico de troca de marchas	16
1.5 DETECÇÃO CORRETA DE MARCHA	17
1.6 TIPOS DE CAIXA DE TRANSMISSÃO	17
1.6.1 Embreagem.....	18
1.6.2 Eixos e engrenagens	19
1.7 TRANSISTORES DE EFEITO DE CAMPO – MOSFETs.....	19
1.8 PLATAFORMA ARDUINO.....	22
1.9 IDE ARDUINO	23
1.10 MICROCONTROLADOR ATMEGA328P	24
1.11 LINGUAGEM C	25
2 METODOLOGIA	26
2.1 ESPECIFICAÇÃO DE COMPONENTES E SOFTWARES UTILIZADOS NO PROJETO	26
2.1.1 Plataforma PROTEUS.....	26
2.1.2 Relé de Estado Sólido – Fotek 100DD 5-220VDC	28
2.1.3 Válvula Direcional 3V210-08-12V	28
2.1.4 Atuador pneumático de dupla ação VORTEX 20x60mm.....	29
3 IMPLEMENTAÇÃO	31
3.1 ORÇAMENTO DE COMPONENTES	35
4 RESULTADOS E DISCUSSÃO	36

CONCLUSÃO.....	41
REFERÊNCIAS BIBLIOGRÁFICAS	43
APÊNDICE A – CÓDIGO PARA TROCA DE MARCHAS	46
APÊNDICE B – FLUXOGRAMAS DO SISTEMA (SUBIR MARCHA)	50
APÊNDICE C – FLUXOGRAMAS DO SISTEMA (NEUTRO).....	51

INTRODUÇÃO

Competições automobilísticas são reconhecidas pela alta velocidade e capacidade de aceleração, não sendo diferente para os modelos de carros participantes da competição Fórmula *Society of Automotive Engineering* (SAE), sendo esse campeonato realizado para desenvolvimento de protótipos de carros modelo Fórmula 1 (F1), em menor escala, e seu posterior teste em diversas condições, como forma de verificar o seu funcionamento como um todo, seja em caráter estático, através de apresentações de custos e design, ou dinâmica, testes em pista, e assim permitir o desenvolvimento de profissionais para categorias superiores no mundo automobilístico, como *Stock Car*, *Formula Porsche* e outros. De modo a alcançar os atributos citados, um melhor aproveitamento da rotação do motor se faz necessário, sendo este alcançado por meio de uma troca de marchas mais eficiente, possibilitando um melhor aproveitamento da faixa de rotação excelente do motor, através de um novo modelo de câmbio de marcha, posicionado no volante, proporcionando ao mesmo tempo, menor emissão de gases poluentes. Em relação ao conforto, pilotos de carros *Fórmula SAE* (FSAE) apresentam desgaste após longos períodos de teste, sendo o principal fator relacionado ao esforço em trocas de marcha.

Com base nesses pontos, este trabalho apresentará o desenvolvimento de um sistema de troca de marchas via borboleta (*paddle shift*) para carros modelo Formula SAE, passando por pontos como definição de fluxograma para a construção do código de controle do processo de trocas e verificação de funcionamento em software para testes de Hardware e Software (Proteus).

Visando fluidez na leitura, a pesquisa é ordenada em 4 capítulos, além das referências, conforme abaixo:

Capítulo 1 – Referencial Teórico: Exibir os conceitos-chave relacionados ao tema, como: ideia por trás da competição, tipos de sistema de troca de marcha disponíveis no mercado.

Capítulo 2 – Metodologia: apresentação dos componentes e softwares utilizados na implementação.

Capítulo 3 – Implementação: Apresenta os passos adotados na construção do código da aplicação, explicitando os caminhos com base no fluxograma do sistema.

Capítulo 4 – Resultados e Discussão: Apresentação dos testes e validações do código desenvolvido no capítulo 3, a fim de produzir informações relevantes para a conclusão.

1 REFERENCIAL TEÓRICO

1.1 SAE – SOCIETY OF AUTOMOTIVE ENGINEERING

É a entidade responsável por coordenar o evento Fórmula SAE, cujo objetivo é o avanço do conhecimento relacionado à mobilidade e a busca de soluções que beneficiem e melhorem a vida da humanidade, por meio da conexão de profissionais e estudantes de Engenharia, de acordo com o seu site oficial.

Fundada em 1905, com o intuito de promover conhecimento público acerca do automóvel como novo modelo de transporte, além de uma entidade que proporcionasse proteção a patentes, criação de problemas técnicos comuns e o desenvolvimento de padrões de engenharia emergiam rapidamente, e muitos engenheiros da área possuíam o desejo de trocar ideias gratuitamente, de modo a expandir as suas bases técnicas individuais.

Figura 1 - Logotipo da SAE



Fonte: *SAE International* (2020).

Em 1991, foi criada a SAE Brasil, representante da *SAE International* no Brasil. Atualmente, contam com mais de 15 mil participantes anualmente em seus eventos, simpósios e demais atividades. A figura 1 exibe a logo oficial da *SAE International*.

1.2 SAE – COMPETIÇÃO

Criada inicialmente para suprir a necessidade de engenheiros automotivos nas competições de alto nível, como Formula Indy, em 1981, nos Estados Unidos, e denominada *Mini-Indy*, a competição cresceu e, em 2004, iniciou-se a competição Formula SAE Brasil, contando com 1030 participantes em 2018, de acordo com o site oficial.

Dispondo de eventos estáticos e dinâmicos, propicia as mais variadas experiências para as equipes participantes de seu final de semana. Entre as provas estáticas, são encontradas: Custos, Apresentação e Design. As provas dinâmicas, em ordem de execução, são: Aceleração, *Skid-pad*, *Autocross* e *Enduro*.

Durante as atividades estáticas, as equipes são responsáveis por exibir os aspectos técnicos dos carros, de modo a evidenciar aos juízes da competição a viabilidade de fabricação de 2 carros, ao menos, por uma linha de produção fictícia, com o menor custo possível. Na prova de Design, a estética e facilidade de manufatura são os critérios adotados para a pontuação.

Em se tratando de provas dinâmicas, a aceleração consiste na realização de um percurso de 75 metros, com o carro partindo do repouso. *Skid-pad* é a segunda atividade da competição dinâmica, consistindo na realização de 4 voltas em trecho fechado em formato de oito, sendo uma volta em cada círculo por piloto. *Autocross* é a penúltima atividade, ocorrendo em um circuito fechado com aproximadamente 800 metros de extensão. A figura 2 exibe uma das provas realizadas na competição. Por fim, tem-se o enduro, prova de maior pontuação, o qual testa o projeto de modo completo durante um percurso de 22 km, onde cada piloto percorre 11 voltas. A prova de eficiência está ligada a esta última, sendo medido o consumo de combustível e dividido pela distância de prova, àqueles times que completaram, no mínimo, 11 voltas. Conquista pontuação máxima a equipe que completar o trecho em menor tempo e, os demais, ganham pontos proporcionalmente ao primeiro colocado. A distribuição de pontos, de acordo com a sequência da competição, com base em critérios definidos pela própria organização, corresponde a:

- 1 – 100 pontos;
- 2 – 75 pontos;
- 3 – 125 pontos;
- 4 – 275 pontos;
- 5 – 125 pontos;

Figura 2 - Foto da Equipe TU Graz durante o Endurance da competição internacional da SAE, em Michigan



Fonte: SAE International (2020).

1.3 CÂMBIO BORBOLETA (PADDLE SHIFT)

Segundo Oliveira (2018), este é um sistema de troca de marchas que permite o uso de alavancas fixadas no volante, similares às asas de uma borboleta. Iniciado em 1989, nas Ferraris 640 pilotadas por Nigel Mansell e Berger, em um sistema de transmissão semiautomática. Tal sistema permitia maior controlabilidade do carro, pelo fato de não retirarem as mãos do volante, além de proporcionar menor fadiga.

É composto por “asas” ou pedais em cada extremidade do volante, sendo a posicionada no lado direito responsável por subir/elevar a marcha, enquanto a da esquerda, descer/reduzir.

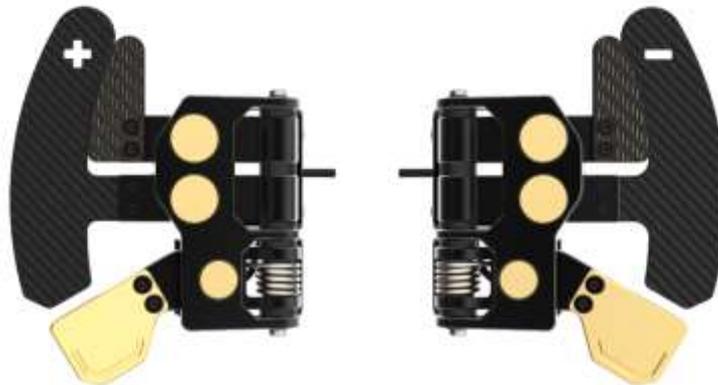
Entre os ganhos proporcionados pela utilização do sistema, podem ser citados a maior confiabilidade da caixa de transmissão, pois, durante a troca de marchas em câmbio manual, existe a mudança forçada de engrenagens da transmissão, podendo reduzir a vida útil ou, durante o movimento, não trocar a marcha e gerar um estanque no motor. Em sistemas de troca via borboleta, o acionamento automático da embreagem durante a mudança de marcha permite transição mais suave de engrenagens. A figura 3 apresenta um modelo de volante utilizado por uma das equipes de Formula 1 na temporada de 2019 e, a figura 4, exibe um modelo de borboletas adotadas em volantes.

Figura 3 - Volante de veículo Fórmula com borboleta em sua parte traseira



Fonte: Racing Hall of Fame Collection (2019).

Figura 4 - Modelo de borboleta de volante



Fonte: BSIM Racing (2018)

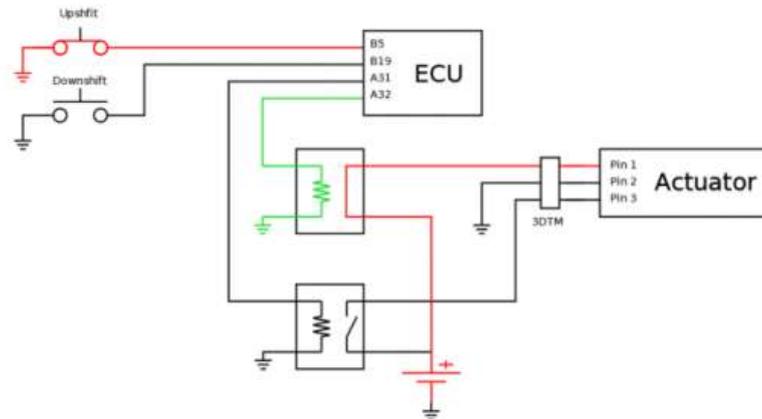
1.4 SISTEMAS EXISTENTES NO MERCADO

1.4.1 Circuitos com MoTec M400 ECU

Uma das aplicações mais conhecidas do mercado é referente à ECU MoTeC M400, apresentada na figura 5, a qual substitui inteiramente a ECU do veículo original (moto ou carro). Nesta configuração, é possível configurar a Unidade de Controle para o recebimento de sinal de acordo com a escolha do piloto (descer/subir marcha) e uma saída/resposta é fornecida. Ainda referente à figura 5, a equipe decidiu pelo uso de válvulas solenoide polarizadas pela ECU, cuja função é controlar o estado de um atuador elétrico de 12V. Conforme será apresentado à frente no presente trabalho, este modelo não depende de fontes pneumáticas/hidráulicas, sendo 100% alimentado pela bateria do veículo, a qual é recarregada pelo alternador, reduzindo, assim, a complexidade do sistema. Apesar de apresentar um custo de componentes alto, devido à ECU personalizada, com custo inicial na faixa de 2870 dólares, em pesquisa realizada no dia 27 de dezembro, sendo possível o controle de 4 velas de ignição, e a possibilidade de inserção de sensores referentes à abertura do *'throttle'* (passagem de ar do motor), motor de passo de controle do estado em marcha neutra e controle do eixo de comando do motor. Além dos controles citados anteriormente, é possível verificar o estado do sensor lambda, o qual é posicionado no escapamento e tem o objetivo de determinar o estado da mistura ar/combustível provenientes dos pistões.

Versões mais completas desta unidade de controle podem atingir o patamar de US\$ 4.700,00 dólares, sendo destinadas a carros customizados e com maior possibilidade de controle um maior número de sensores do carro/veículo.

Figura 5 - Aplicação da ECU MoTeC em sistema de troca de marchas elétrico



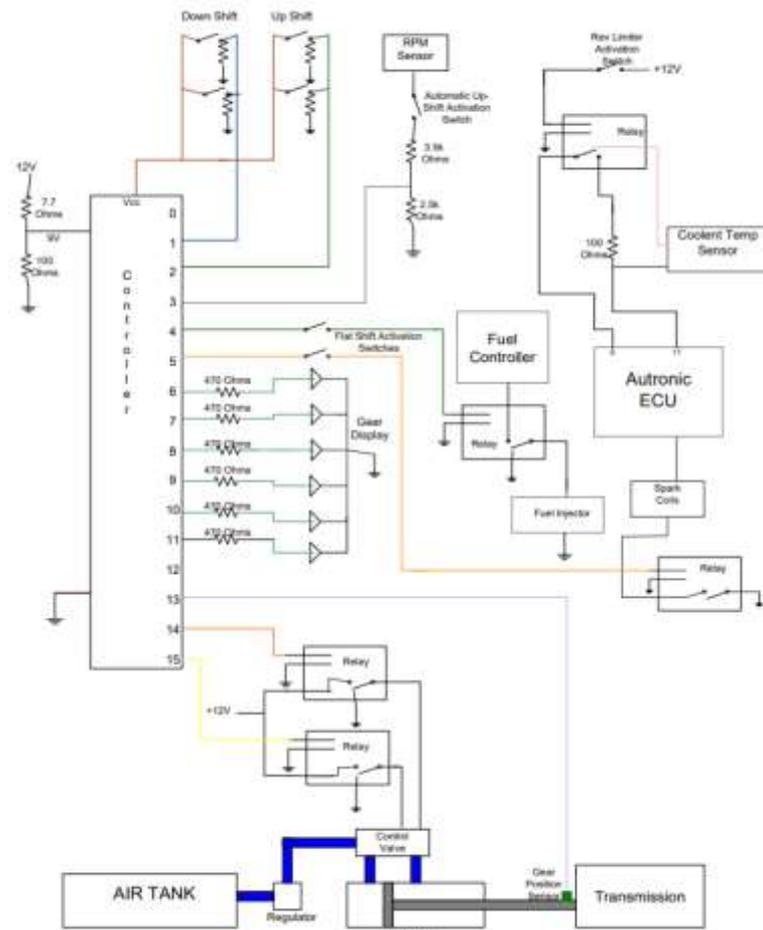
Fonte: PITTMAN (2015)

1.4.2 ECU Autronic

O sistema exibido na figura 6 apresenta uma ECU Autronic, a qual possui ideia de funcionamento semelhante à da MoTeC, porém, não controlando os sinais de entrada provenientes do piloto, mas sim, informações referentes à operação do motor, como número de rotações e a temperatura do líquido de arrefecimento do sistema, em outras palavras, líquido de refrigeração do motor, sendo estes dois necessários para o caso em que o piloto opte por uma troca de marchas automática em uma faixa de rotação ideal.

As informações de troca de marcha, para o sistema não automático, continuam sendo fornecidas pelo piloto através dos “paddles” (pedais), porém, os sinais são processados por um controlador à parte. Caso uma das saídas para os atuadores fique em estado/nível alto, entra em ação um dos grandes diferenciais deste sistema, a ativação das chaves de ‘flat shifting’, as quais consistem no desligamento temporário dos sistemas de injeção e ignição do motor, através do desligamento de seus respectivos relés de polarização, com o intuito de reduzir a sua rotação e permitir a troca de marchas sem a necessidade de atuação de embreagem e remoção do pé no pedal do acelerador, aumentando a eficiência do processo e, ao mesmo tempo, evitando possíveis desgastes prematuros das engrenagens da caixa de marcha. Este sistema é eletropneumático, apresentando elementos como, tanques de ar, válvulas solenoides e relés de estado sólido, conforme será apresentado adiante. É um sistema com custo-benefício elevado, contendo apenas uma desvantagem referente à limitação do número de trocas através da capacidade do sistema (volume do cilindro de ar pressurizado).

Figura 6 - ECU Autronic com controle de rotação do motor e corte de injeção/ignição

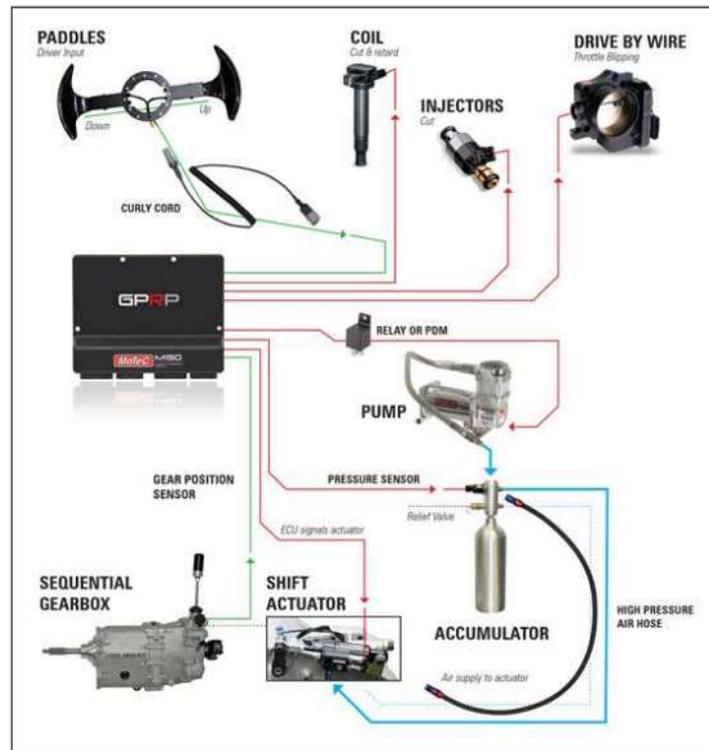


Fonte: PITTMAN (2015).

1.4.3 MoTeC M1 PaddleShift

De acordo com o manual de usuário fornecido pela fabricante MoTeC para o modelo M1 de seu sistema de troca de marchas via borboleta e apresentado em esquema na figura 7, são necessários os seguintes itens, contendo obrigatoriedade de sensores::

Figura 7 - Estrutura do sistema de troca de marchas utilizando borboletas da MoTeC - M1



Fonte: MoTeC (2017a).

Figura 8 - Itens obrigatórios/opcionais no sistema de troca de marchas da MoTeC M1.

Item	Importance	Use of item within GPRP
Up and Down Paddles	Mandatory	Initiate the up or down shift event
Up and Down Actuators	Mandatory	Move the gear change mechanism with compressed air on request
Paddle shift controller (GPRP)	Mandatory	Control the system
Barrel Position Sensor	Mandatory	Measure the gear position
Air pump	Mandatory	Create the air pressure for the shift
Air reservoir	Mandatory	Store the air pressure for the shift
Air Pressure Sensor	Mandatory	Control the pump state to maintain pressure
Clutch Switch	Mandatory	Allow the system to lock out certain changes without a clutch
Drive by Wire throttle	Preferable	Used for downshift throttle blipping
Wheel speeds	Preferable	Used for Engine speed matching
Paddle shift mode dial	Optional	Vary the system between auto, manual and fault modes
Brake State	Optional	Used in Auto shift mode only

Fonte: MoTeC (2017b).

De modo simplificado, a figura 8 informa aos usuários que o sistema é composto por injetores, velas de injeção, borboleta do motor (responsável por regular a passagem de ar para os pistões), bomba de ar pressurizado, relé de acionamento da bomba, acumulador, atuador pneumático e a caixa de marcha sequencial. As linhas verdes da figura correspondem aos sinais de entrada da ECU da MoTeC, sendo a requisição do piloto para mudança de marcha e a marcha

atual do carro (antes e após a tentativa de troca). As linhas em vermelho, no entanto, referem-se à saída, sendo as velas e injetores desativados durante a troca, a borboleta eletrônica do motor fechada momentaneamente e o sistema de ar ativado conforme programação estabelecida por interface gráfica previamente ao uso, conforme figura 9 a seguir:

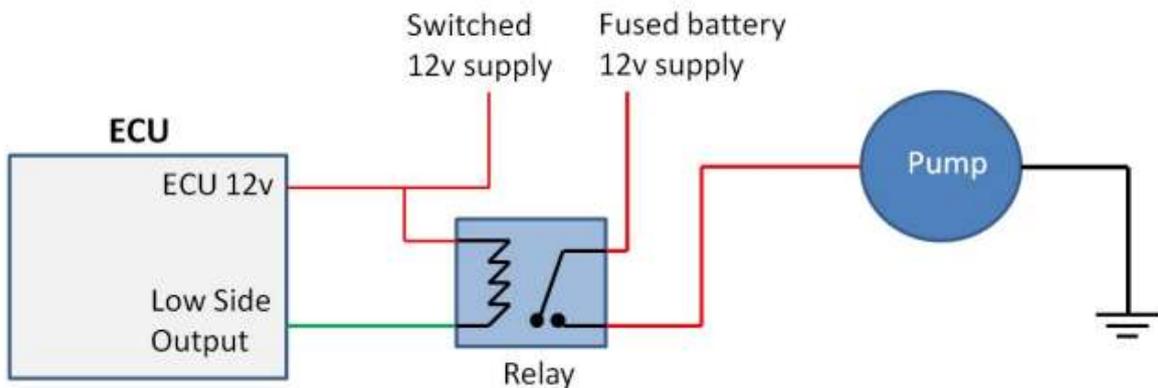
Figura 9 - Informações utilizadas para definir o tempo de marcha das trocas

	Default	Reverse	Neutral	First	Second	Third	Fourth	Fifth	Sixth	Seventh
Rearm	100	100	100	100	100	100	100	100	100	100
Recover	100	100	100	100	50	50	30	20	20	20
Post Shift	0	0	0	0	0	0	0	0	0	0
Shift	150	150	150	150	150	150	150	150	150	150
Pre Shift	0	0	0	0	0	0	0	0	0	0

Fonte: MoTeC (2017c).

Com respeito ao processo de ativação da bomba, todo o controle é realizado pela ECU, a qual disponibiliza ou não uma tensão de 12V com corrente capaz de ativar um relé e polarizar o motor apropriadamente, conforme figura 10:

Figura 10 - Esquema de polarização do relé responsável por alimentar a bomba de ar do sistema pneumático



Fonte: MoTeC (2017d).

Por fim, a figura 11 representa a instalação do atuador, sendo necessário suporte para a base e a pressurização do atuador, quando solicitada, permite a movimentação da alavanca de troca de marcha, destacada em preto, à esquerda.

Figura 11 - Exemplo de posicionamento do atuador pneumático em um motor



Fonte: MoTeC (2017e).

1.4.4 Sistema de Aquisição de Dados

Segundo Pittman (2015), o time de FSAE da Universidade de Delaware utilizou um sistema de aquisição de dados com base no Dashboard AIM PISTA MXL. Através deste sistema, informações diversas são fornecidas ao piloto, podendo ser citadas: Marcha atual, rotação do motor em escala, velocidade, tempo de volta, melhor volta, luzes indicadoras de troca de marcha e outras configurações acessíveis e ajustáveis através de botões ou painel utilizado pela equipe de Telemetria da equipe. Informações mais específicas, como tempo de melhor volta, são sincronizadas com o dashboard através da rede veicular CAN, com dados saindo da própria ECU do veículo, conforme figura 12.

Figura 12 - Dashboard utilizado pela equipe da Universidade de Delaware para obtenção e exibição de dados ao piloto



Fonte: PITTMAN (2015).

1.4.5 Sistema QUICKSHIFTER de troca de marcha

Uma opção de mercado com bastante destaque no cenário nacional é a fornecida pela empresa QUICKSHIFTER, com um conjunto de custo de até R\$ 2.500,00 reais em pesquisa de sites gerais de venda, é um kit completo utilizado principalmente em motos esportivas, mas que possui aplicação a carros FSAE. É um sistema bastante semelhante com os citados anteriormente, porém, de tecnologia nacional e preço bastante atrativo. Levando em conta a instalação em motos, a sua presença dispensa o pressionamento da alavanca de embreagem para troca de marchas, sendo apenas necessário a utilização do pé para movimentação do seletor, o qual aciona o atuador, que corta a injeção e ignição momentaneamente, a fim de permite uma troca segura. É um sistema puramente elétrico, sem necessidade de acionamentos pneumáticos ou hidráulicos. A seguir é apresentada, na figura 13, uma imagem contendo o modelo de entrada da marca:

Figura 13 - Kit básico da QuickShifter para troca de marchas em motocicletas



Fonte: QuickShifter (2021).

1.4.6 Sistema hidráulico de troca de marchas

É um sistema de troca de marchas rápido, quando comparado ao processo manual, mas, diferentemente dos métodos elétricos/eletropneumáticos, apresenta uma grande propensão a vazamentos de óleo, que é a base do sistema, podendo acarretar riscos de segurança ao piloto e veículo. Outro ponto de destaque é a complexidade da fiação e peso do sistema, pois é necessário um sistema pressurizado para que todo o conjunto funcione de maneira apropriada, necessitando de compressor e tanque de armazenamento, em alguns casos. Devido aos pontos citados acima, não possui a atenção de equipes da categoria, sendo mais voltado para casos particulares de substituição do sistema de marcha em H (comum em carros manuais). A seguir é exibida na figura 14 uma imagem referente a uma montagem realizada por uma equipe e citada no trabalho de Pittman (2015).

Figura 14 - Sistema hidráulico de troca de marcha



Fonte: PITTMAN (2015).

1.5 DETECÇÃO CORRETA DE MARCHA

Em sistemas mais avançados, a detecção de sucesso ou não na troca de marcha é realizada através do recebimento de sinal do sensor de marcha do veículo. A ideia se dá através dos seguintes passos:

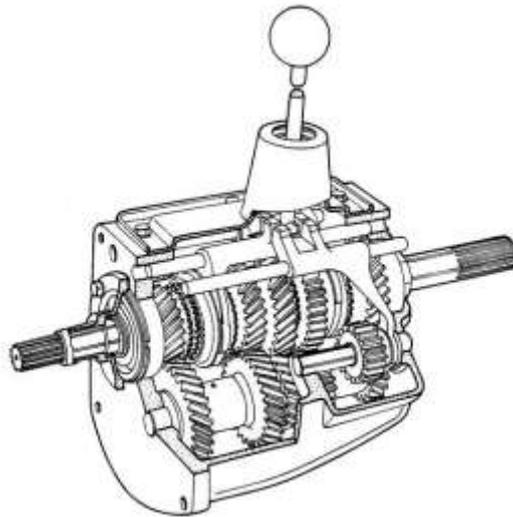
- O sistema entra em processo de troca de marcha;
- Após o término, a marcha atual é informada ao microcontrolador / ECU;
- Caso a marcha seja igual à esperada, a ação é considerada concluída, senão, uma nova tentativa é iniciada.

1.6 TIPOS DE CAIXA DE TRANSMISSÃO

Segundo Soares et al (2021), existem dois tipos de caixa de transmissão disponíveis no mercado, sendo a manual e a automática. A seguir será apresentada uma explanação breve acerca do primeiro modelo, pois é o enfoque deste trabalho. O câmbio manual é a principal e mais simples forma de transmissão automotiva. Seu funcionamento está associado as relações de engrenagens da caixa de velocidades, que serão as responsáveis pela multiplicação do torque. Além disso, sua principal característica é a troca de marcha manual em conjunto com o acionamento do pedal de embreagem (WEIHERMANN 2015).

Este sistema é o adotado em motocicletas, em específico ao modelo XJ6N2012, com capacidade de 600cc e funcionamento sequencial. A seguir, na figura 15, é exibida uma imagem de caixa de transmissão manual.

Figura 15 - Câmbio Manual

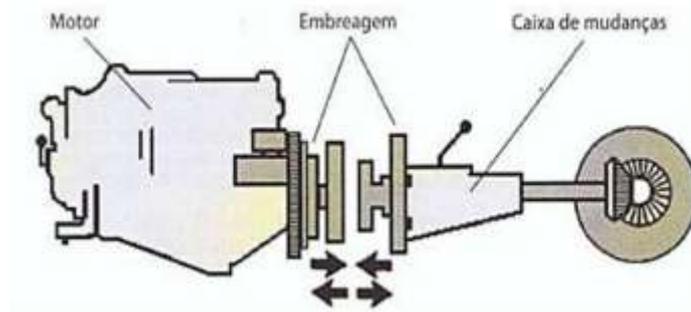


Fonte: EEEP (2014, p. 17).

1.6.1 Embreagem

Segundo Cavalcante (2021), a embreagem é um conjunto de discos que ligam o eixo virabrequim do motor ao eixo primário de transmissão da caixa de velocidades. Permite a separação entre o torque e a transmissão do motor durante uma troca de marchas, de forma a evitar desgastes prematuros nas engrenagens do sistema. Em carros, o seu acionamento se dá por meio do pedal esquerdo, enquanto que em motos, na alavanca esquerda do guidão. Abaixo é exibida na figura 16 uma imagem do sistema de embreagem em operação:

Figura 16 - Esquema de um conjunto de embreagem



Fonte: JUNIOR (2018).

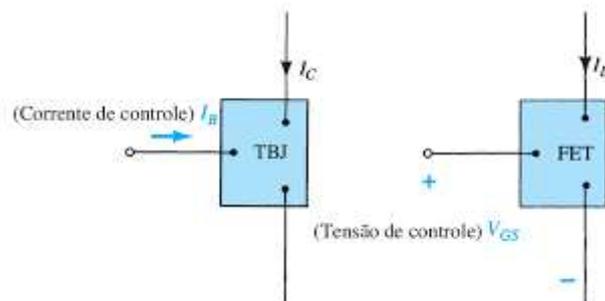
1.6.2 Eixos e engrenagens

Os eixos e engrenagens encontram-se dentro da caixa de velocidade. As engrenagens são responsáveis pela correta multiplicação do torque, sendo os elementos da relação de transmissão existente no sistema. Os eixos são os elementos que irão transmitir a rotação proveniente do motor com o auxílio dos pares de engrenagens.

1.7 TRANSISTORES DE EFEITO DE CAMPO – MOSFETs

O transistor de efeito de campo (FET, do inglês *field-effect transistor*) é um dispositivo de três terminais utilizado em várias aplicações que em muito se assemelham àquelas do transistor TBJ (BOYLESTAD, 2013). A principal diferença entre os dois tipos de transistor é o fato de que o TBJ é um dispositivo controlado por corrente, enquanto o FET, por tensão. A figura 17 apresenta as diferentes configurações de um transistor TBJ e FET.

Figura 17 - Amplificadores: (a) controlados por corrente e (b) controlados por tensão



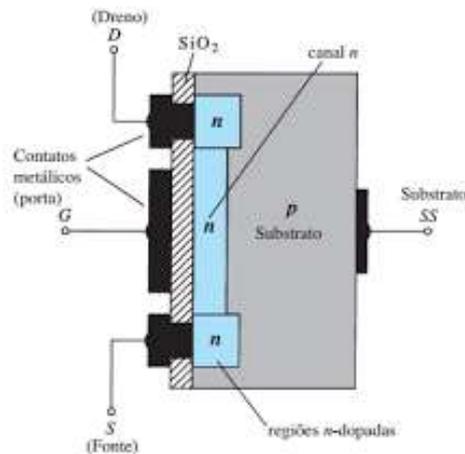
Fonte: BOYLESTAD (2013).

Uma das principais características do FET é a sua alta impedância de entrada (BOYLESTAD, 2013). Com valores que variam de $1M\Omega$ a várias centenas de megaohms, a impedância é muito maior que em dispositivos TBJ. Por outro lado, o TBJ possui sensibilidade maior a variações de sinal.

Para os MOSFETs, um dos modelos entre os transistores de efeito de campo e adotado no projeto, existem dois tipos: depleção e intensificação. Estes termos definem seus modos básicos de operação, e o nome MOSFET significa transistor de efeito de campo metal-óxido-semicondutor.

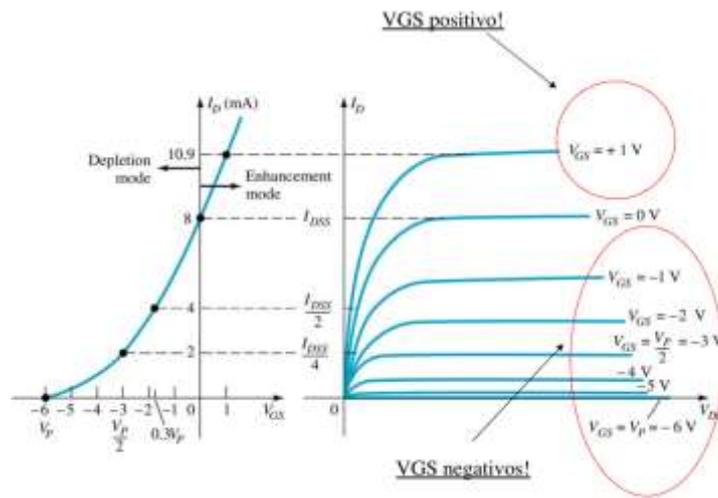
Apesar de possuírem estrutura semelhante, com a presença de terminais para o dreno (*drain*), porta (*gate*) e fonte (*source*), a principal diferença entre os dois modelos se encontra na presença ou não de canal. Para o de tipo depleção, dependendo da configuração, o canal já é existente. A figura 18 representa um MOSFET tipo n, onde, nesta configuração, caso seja injetada uma tensão dreno-fonte positiva, haverá fluxo de corrente através do canal e o dispositivo entra em operação. A sua região de corrente é somente atendida para valores negativos, normalmente acima de 5 V (negativo), responsável por “estrangular” o canal e interromper o fluxo de elétrons entre os terminais, conforme figura 19.

Figura 18 - Formação do canal no MOSFET tipo depleção de canal n



Fonte: BOYLESTAD (2013).

Figura 19 - Modos de operação do MOSFET tipo depleção de canal n



Fonte: BOYLESTAD (2013).

Enquanto isso, os MOSFETs de intensificação não possuem canal, sendo, assim, necessária uma tensão superior (em termos absolutos) ao nível de limiar do dispositivo para que entre em operação. Estes valores, em grande parte, iniciam em 2.5 a 3 V. Para tensões abaixo do nível de *threshold*, entra em corte. A seguir na figura 20 é exibida uma imagem em operação e, na figura 22, o seu respectivo gráfico para diferentes valores de polarização porta-fonte (V_{GS}).

Figura 20 - Formação do canal no MOSFET tipo intensificação de canal n

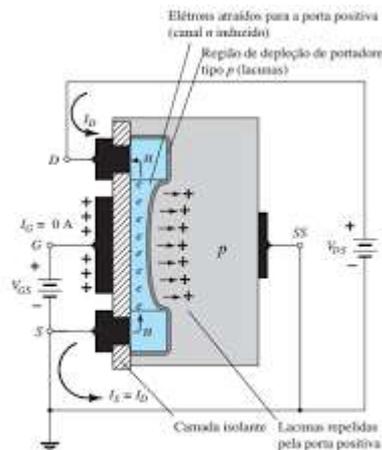
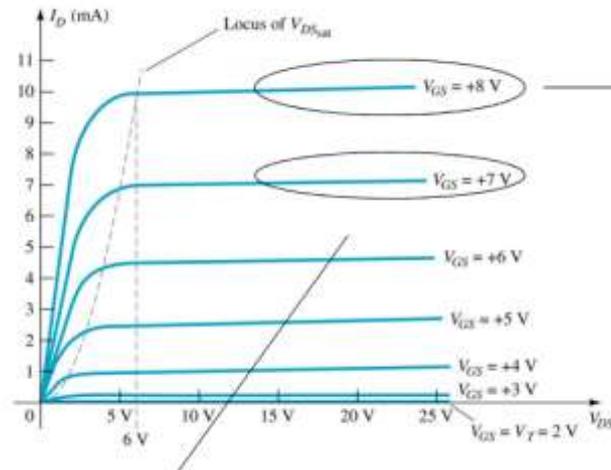


Figura 6.33 Formação do canal no MOSFET tipo intensificação de canal n.

Fonte: BOYLESTAD (2013, p.337).

Figura 21 - Curvas características de dreno de um MOSFET tipo intensificação de canal n com $V_t = 2\text{V}$ e $k = 0,278 \times 10^{-3} \text{ A/V}^2$



Fonte: BOYLESTAD (2013, p.338).

1.8 PLATAFORMA ARDUINO

Segundo o site do Arduino (2020), esta é uma ferramenta eletrônica baseada em *hardware* e *software* de fácil utilização. As placas de Arduino são capazes de ler entradas – luz ligada em um sensor, o pressionamento de um botão ou o recebimento de uma mensagem no Twitter – e o transforma em uma saída – ativando um motor, ligando um *Light Emitting Diode* (LED), publicando algo on-line. Você pode dizer à sua placa o que fazer através do envio de uma série de instruções ao microcontrolador em sua placa. Para fazê-lo, você usa a linguagem de programação do Arduino (baseada em Wiring) e o software Arduino (*Integrated Development Environment* - IDE), baseado em Processamento.

Ao longo dos anos, Arduino vem sendo o cérebro de milhares de objetos diários a instrumentos científicos complexos. Uma comunidade mundial de criadores – estudantes, pessoas que o fazem por hobby, artistas, programadores e programadores – têm se juntado nesta plataforma de fonte aberta e suas contribuições têm adicionado a uma incrível quantidade de conhecimento acessível que pode ser de grande ajuda para novatos e experientes também.

Nasceu no Instituto Ivrea de Design Interacional como uma ferramenta fácil para rápida prototipação, visando estudantes sem prévio conhecimento em programação eletrônica. Assim que alcançou uma comunidade maior, a placa de Arduino iniciou uma mudança para se adaptar a novas necessidades e desafios, diferenciando sua oferta de placas simples de 8-bits para produtos destinados a aplicações de IoT, “usáveis”, impressão 3D e ambientes embarcados. Todas as placas de Arduino são completamente de circuito aberto,

empoderando usuários a construí-los independentemente e eventualmente adaptá-los às suas necessidades particulares. O *software* também, é de fonte aberta, e está crescendo através da contribuição de usuários ao redor do mundo. A figura 22 exibe um Arduino UNO.

Figura 22 - Representação de Arduino UNO



Fonte: Arduino (2020).

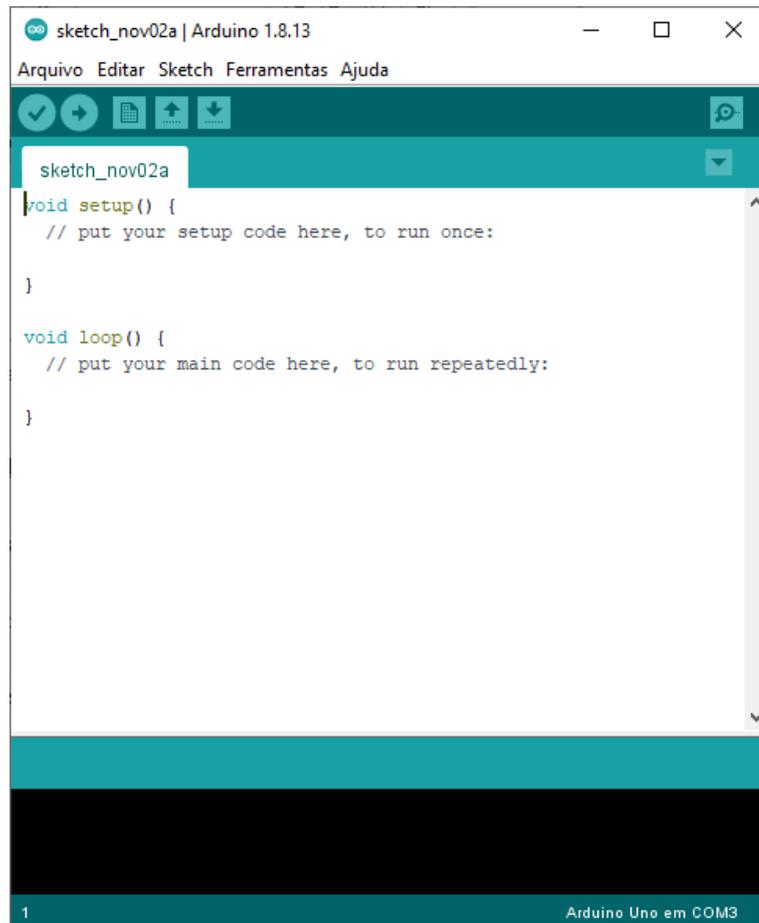
1.9 IDE ARDUINO

Utiliza comandos da linguagem Wiring, desenvolvida por Hernando Barragán, através da lista de comandos apresentados a seguir, os quais facilitam a atribuição de níveis lógicos para níveis digitais:

- `pinMode()`, `digitalRead()`, `digitalWrite()`, `analogRead()`, `analogWrite()`, `delay()` e `millis()`.

Abstrair os pinos do microcontrolador como número foi, sem dúvidas, uma grande decisão, possível porque a sintaxe foi definida antes da implementação em qualquer plataforma de *hardware*. Toda a nomeação de comandos da linguagem e sintaxe são resultado de um exaustivo processo de design que Barragán conduziu, incluindo teste de usuários com estudantes, observação, análise, ajuste e iteração, segundo Barragán (2020). A figura 23 apresenta a interface inicial da IDE do Arduino.

Figura 23 - Imagem da IDE do Arduino



Fonte: Autoria Própria (2021).

1.10 MICROCONTROLADOR ATMEGA328P

O ATmega328 é um microcontrolador com 14 pinos digitais de entrada/saída e velocidade de clock de 16 MHz. Segundo Pittman (2015), existem recursos suficientes no Atmega328 para adequadamente lidar com as demandas do sistema de troca. A razão primária é que este microcontrolador é fácil de usar em termos de programação. As placas do Arduino têm relativamente funções simples devido a suas bibliotecas bem mantidas e forte apoio da comunidade. Mas, quando comparada à sua contra-parte Texas Instruments, o Arduino Uno perde em termos de flexibilidade. O Atmega328 não permite o controle sobre a velocidade do clock como os microcontroladores da linha da Texas Instruments e não há modo de repouso (baixo consumo), o qual é muito importante, pois o sistema de troca de marchas será alimentado pela bateria do carro (PITTMAN, 2015). A seguir é exibida na tabela 1 as características do microcontrolador adotado para o trabalho:

Tabela 1 - Dados referentes ao microcontrolador Atmega328

Primary Features	Values
Clock	16 MHz
Flash Memory	32 KB
SRAM	2 KB
Operating Voltage	5V
Extended Temperature	-40°C - 85°C
Current Consumption	46.5 mA, active 1456 µA, sleep

Fonte: PITTMAN (2015).

1.11 LINGUAGEM C

Segundo Pinho (2020), a primeira versão de C foi criada por Dennis Ritchie em 1972 nos laboratórios Bell para ser incluído como um dos softwares a serem distribuídos juntamente com o sistema operacional Unix do computador PDP-11, na equipe certificada por Ken Thompson.

Linguagem de programação lançada após a não continuidade das linguagens ALGOL, CPL e BPCL, com o intuito de ser uma linguagem de programação que mantivesse contato com o computador real e proporcionar o desenvolvimento de códigos em área como comercial, científica e engenharia.

Apresentou documentação padrão na publicação *“The C Programming Language”*, considerada a base da linguagem C. Entre os pontos de destaque, está a capacidade de funcionar em diferentes modelos de computador, também denominada portabilidade, permitindo uma redução de custos devido a não necessidade de máquinas compatíveis a que gerou o código. Tal característica permitiu a sua expansão, apresentando grande aplicabilidade até os dias de hoje.

Figura 24 - Símbolo da Linguagem C



Fonte: Perito em PHP (2019).

2 METODOLOGIA

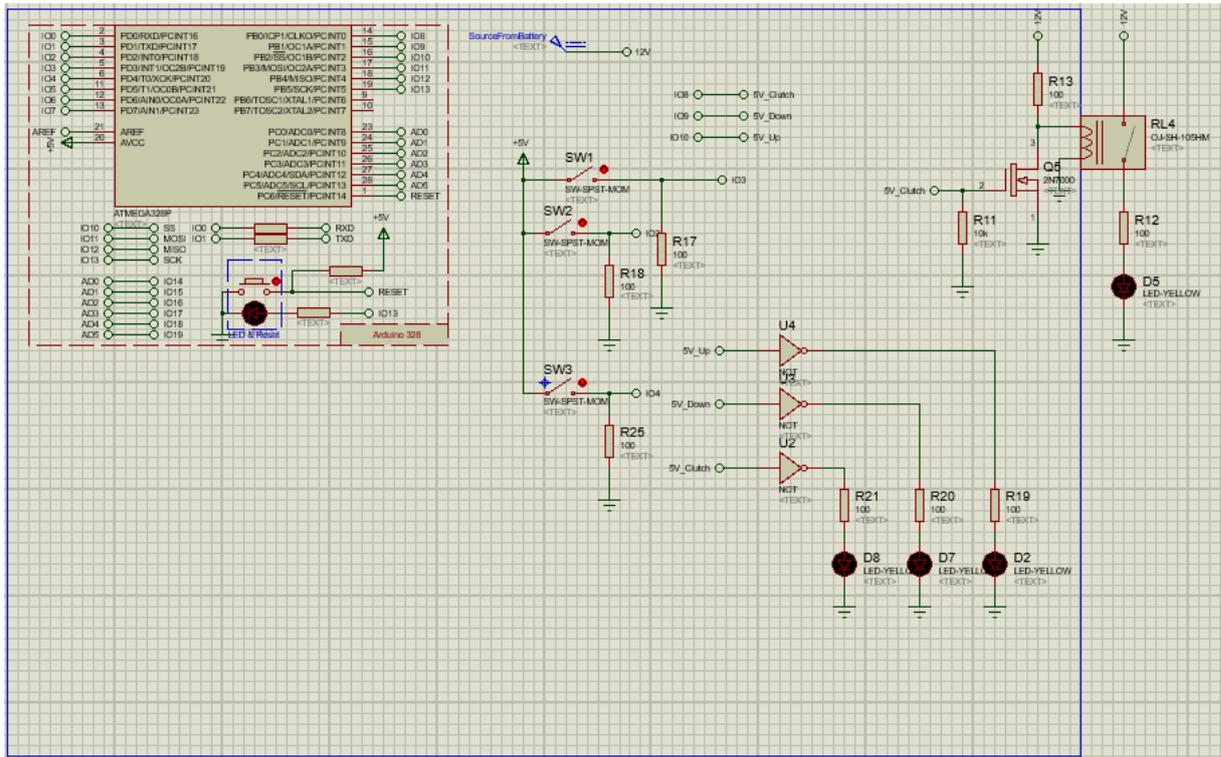
- Para o desenvolvimento do sistema, foram adotados os passos abaixo:
- Elaboração de diagrama de fluxo (fluxograma) para a identificação de etapas-chave a serem representadas na codificação do sistema.
- Codificação do sistema, com base nos itens identificados no fluxograma definido no item
- Testes em plataforma de desenvolvimento de software embarcado (Proteus), utilizando todos os componentes dimensionados para futuros testes práticos.
- Compra de materiais para teste prático.

2.1 ESPECIFICAÇÃO DE COMPONENTES E SOFTWARES UTILIZADOS NO PROJETO

2.1.1 *Plataforma PROTEUS*

Adotada para a construção dos esquemáticos contendo MOSFETs, Arduino e demais componentes do sistema. Permitiu uma visualização física da estrutura lógica estruturada no Arduino, sendo necessários alguns ajustes para atender à disposição com transistor de efeito de canal n. A seguir é apresentada na figura 23 uma imagem de parte do circuito projetado no *software*:

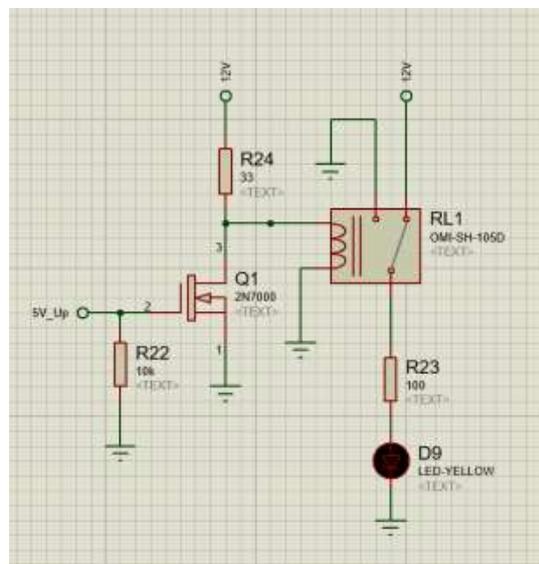
Figura 25 - Esquema parcial de circuito do sistema de troca de marchas construído no Software Proteus



Fonte: Autoria Própria (2021).

Como parte do dimensionamento deste sistema, a escolha do resistor de 33Ω se deu com base na tensão de operação/resistência interna do SSR escolhido (5V / 55Ω), trabalhando como um divisor de tensão. A adoção de um resistor e LED em série configuram uma representação da válvula solenoide e atuador pneumático, nesta ordem. Essa escolha ocorreu devido ao não suporte do Proteus a elementos pneumáticos. A figura abaixo exhibe a configuração.

Figura 26 - Circuito polarizador de válvula solenoide



Fonte: Autoria Própria (2021).

2.1.2 Relé de Estado Sólido – Fotek 100DD 5-220VDC

Este modelo de relé de estado sólido da marca Fotek tem fabricação em Taiwan, possui uma tensão de acionamento mínima de 3VDC e máxima de 32VDC e suporta cargas entre 5 e 220 VDC. Sua escolha foi baseada na disponibilidade do mercado, pois, em sua maioria, os relés de estado sólido são aplicados para cargas em corrente alternada, com tensão de operação entre 110VAC e 220VAC. Apesar da possibilidade de acionamento em uma tensão menor que a planejada, se encaixou às especificações do sistema. A figura 27 corresponde a um modelo de SSR utilizado.

Figura 27 - SSR modelo 100DD 5-220VDC



Fonte: Mercado Livre (2021a).

2.1.3 Válvula Direcional 3V210-08-12V

Modelo de válvulas produzido pela empresa Toral Comércio, sendo correspondente à um controlador de fluxo de ar de 3/2 vias, apresentando retorno por mola, e tensão de polarização de 12V. Outra característica relevante para este item é o seu grau de proteção IP65, correspondente a ondas de mar/jatos fortes e totalmente protegido à poeira. É adotado no sistema de troca de marchas para controlar o fluxo de ar aos atuadores pneumáticos ligados à alavanca de marcha e embreagem, respectivamente. A seguir é apresentada na figura 28 uma imagem obtida de um site de revenda:

Figura 28 - Imagem da válvula solenoide 3V210-08-12V



Fonte: Mercado Livre (2021b).

2.1.4 Atuador pneumático de dupla ação VORTEX 20x60mm

Atuador pneumático escolhido com base nos critérios adotados por (PITTMAN, 2015), levando em conta as mesmas forças aplicadas e extensão da alavanca de marcha, conforme a figura 29:

Figura 29 - Forças mínimas aplicadas para troca de marcha no motor da equipe UCF

FORCE TABLE FOR SHIFTING Up From	To	Shifting Up Force (lbs)
1	2	12
2	3	11
3	4	11
4	5	11
5	6	11

Fonte: Pittman (2015).

Por meio da figura 28, é possível observar que o pior cenário ocorre para uma força de 12 libras aplicada, o que equivale a 53,4N. Em operação normal, uma pressão de 100 psi (aproximadamente 0,689 MPa) é aplicada ao sistema, em um comprimento de haste médio de 40 mm. Com todos os parâmetros definidos, foi possível aplicar na fórmula da pressão aplicada em uma área, conforme passos abaixo:

$$P = \frac{F}{A}$$

$$0,689476 * 10^6 (MPa) = \frac{53,4 (N)}{A}$$

$$A = 7,74 * 10^{-5} m^2$$

Com a área definida, foi possível calcular o diâmetro mínimo para o atuador suportar as condições definidas no sistema através dos passos seguintes:

$$\pi * \left(\frac{\phi}{2}\right)^2 = 7,74 * 10^{-5} (m^2)$$

$$\frac{\phi}{2} = 4,965 * 10^{-3} (m^2)$$

$$\phi \cong 9,93 \text{ mm}$$

Assim, para que não haja quaisquer problemas, foi necessário orçar/comprar um equipamento que apresentasse um diâmetro mínimo de 10 mm, sendo atendido pelo modelo da fabricante VORTEX, com 20 mm de diâmetro e 60 mm de curso, sendo suficiente para movimentar a alavanca em, pelo menos, uma direção. A seguir é exibida na figura 30 uma imagem do modelo utilizado para o projeto, obtida através de site de compra/venda:

Figura 30 - Atuador pneumático de dupla ação VORTEX 20 mm x 60 mm



Fonte: Mercado Livre (2021c)

3 IMPLEMENTAÇÃO

A codificação do sistema foi iniciada com base nos seguintes pressupostos:

1. O carro realizará todo o processo de troca de marcha ativando o sistema de embreagem (tanto subida, quanto descida);
2. Durante o processo de troca de marcha, caso uma operação esteja sendo realizada, não é possível que outra seja efetuada enquanto a primeira não finalizar (não é permitida operação simultânea);
3. O tempo de troca de marcha para simulação em software equivaleu a 1.5 segundos, sendo divididos em: 500 milissegundos para ativação do relé de estado sólido da embreagem e 1 segundo para visualização da transição do sistema.
4. O piloto não precisa segurar a borboleta durante todo o período de mudança de marcha, devido a questões ergonômicas.
5. O motor possui capacidade para 6 marchas, não existindo ré, sendo o processo de trocas sequencial;
6. A marcha da moto somente pode ser enviada ou sair do neutro se uma das condições a seguir for atendida:
 - 6.1 A marcha sai da posição neutra para a primeira somente com o pressionamento da borboleta de descida de marcha, sendo negada qualquer outra tentativa.
 - 6.2 O motor só atinge a posição neutra se estiver na primeira marcha, com o acionamento do botão de neutro.

Com base nesses pontos, os passos a seguir exibem a declaração das variáveis e preparação dos pinos utilizados no Arduino, plataforma adotada para o controle e gerenciamento dos pulsos fornecidos pelo piloto.

Nas primeiras linhas do código exposto nos apêndices, são definidos os pinos digitais de entrada e saída do sistema, sendo nesta ordem:

1. “**downGear**”: pino digital do tipo entrada estabelecido para a porta 2. É responsável por receber o sinal de redução de marcha proveniente do piloto.
2. “**upGear**”: pino digital do tipo entrada estabelecido para a porta 3. Recebe o sinal proveniente do piloto para subir a marcha.
3. “**clutchOut**”: pino digital do tipo saída estabelecido para a porta 8. Envia um sinal elétrico responsável por ativar a embreagem do carro.
4. “**upGearOut**”: pino digital do tipo saída estabelecido para a porta 10. Envia um sinal elétrico responsável por ativar a alavanca de marcha e subir a marcha.

5. “**downGearOut**”: pino digital do tipo saída estabelecido para a porta 9. Envia um sinal elétrico responsável por ativar a alavanca de marcha e descer a marcha.

Além destas, 3 variáveis lógicas (booleanas) foram criadas a fim de controlar o estado de cada uma das entradas do sistema (subir a marcha, descer a marcha e ativar a embreagem). Todas são inicialmente falsas, denotando que a saída do sistema, em termos lógicos, está inativa.

Conforme citado no item 3 dos pressupostos para a codificação, as variáveis de tempo (temporizadores) foram estabelecidas em 500ms para o tempo de troca de embreagem, e 1s para a mudança efetiva de marcha no simulador.

Com base nos pressupostos 5 e 6, foram definidos os requisitos ou especificações base para o desenvolvimento correto e seguro da troca de marchas para ou saindo do neutro. Para que este fim pudesse ser alcançado, foram definidas três variáveis, as quais se encontram na figura acima, sendo estas:

gearCounter: É o contador de marchas do sistema. Nesta versão, não possui uma realimentação por sensor de troca de marcha, sendo apenas uma promessa de sucesso no processo.

neutralTime: Tempo definido para que a troca de marcha para o neutro seja realizada. Um tempo menor foi considerado devido ao funcionamento próximo à linearidade de um sistema pneumático pressurizado, de forma a garantir que a alavanca de marcha percorra apenas metade de seu curso (distância suficiente para garantir o posicionamento do neutro do motor).

type: Variável do tipo char utilizada para o controle de estado do sistema. Caso o piloto deseje sair da marcha neutra para a primeira, seu estado é definido como ‘u’, de *upshift*, senão, ‘d’, de *downshift*.

Após a declaração, deu-se início à preparação do ambiente, com a inicialização dos estados de cada pino: se entrada ou saída de sinal, e, caso saída, o seu nível em alto ou baixo. Em seguida, foi adotada a modularização (separação de códigos importantes da aplicação em funções bem definidas) para mais fácil legibilidade e reaproveitamento, caso seja necessária inserção de sensores do carro, por exemplo. A utilização do *Serial.begin(9600)*, na última linha da imagem a seguir, está relacionada à necessidade de debug durante a construção do código, como meio de garantir a execução do trecho de código definido. A figura 31 apresenta as configurações dos pinos utilizados na implementação do código:

Figura 31 - Inicialização de pinos digitais através do método setup() do Arduino

```

void setup() {
  // put your setup code here, to run once:
  pinMode(downGear, INPUT);
  pinMode(upGear, INPUT);
  pinMode(neutralButton, INPUT);
  pinMode(clutchOut, OUTPUT);
  pinMode(downGearOut, OUTPUT);
  pinMode(upGearOut, OUTPUT);

  digitalWrite(clutchOut, LOW);
  digitalWrite(downGearOut, LOW);
  digitalWrite(upGearOut, LOW);

  Serial.begin(9600);
}

```

Fonte: Autoria Própria (2021).

A ideia por trás de cada módulo é baseada nos estados alto e baixo, ligado e desligado. Caso o piloto deseje trocar a marcha, o método de ativar a embreagem é chamado, onde é verificado o seu estado original e, caso inativo, é trocado e vice-versa. O mesmo princípio é adotado para a subida e descida de marcha, sendo realizado o desativamento de cada um destes módulos conforme a lógica definida no escopo principal da aplicação, encontrado em *void loop()*.

Com base no diagrama exposto em anexo, a lógica por trás do *void loop()* é como segue:

Para a subida de marcha:

1. Caso o piloto tenha pressionado a alavanca de subida de marcha e a embreagem não tenha sido ativada (está verificando o primeiro sinal para a realização da troca) e não esteja sendo processado nenhuma descida de marcha e a marcha esteja entre a 1° e 5°, faça:
2. Troca o estado da variável que checa o estado do pressionamento do botão/borboleta de subida.
3. Troca o estado da booleana da embreagem para verdadeiro;
4. Realiza a chamada da função de ativação da embreagem com o estado da booleana;
5. Inicia a contagem do tempo;

6. Caso a embreagem e a variável de estado da subida de marcha estejam ativadas, verifica se o tempo de execução do Arduino menos o tempo de início da troca é maior ou igual ao tempo base para a ativação da embreagem (500ms);
7. Se verdade, desativa a variável lógica da embreagem (muda o estado simbolicamente, pois o método de desativação da saída não é chamado neste trecho do código) e chama a função responsável por subir a marcha e reinicia a contagem com o tempo de execução atual. **OBS:** Este processo é realizado para que o relé de estado sólido e as válvulas eletropneumáticas entrem em operação, pois possuem um tempo de atraso entre 50 e 80ms, para o pior caso.
8. Caso a variável de estado da embreagem esteja em nível baixo e ainda esteja no processo de subida de marcha, verifica se a diferença entre o tempo atual do microcontrolador e o novo tempo de referência é maior ou igual ao tempo efetivo de troca.
9. Se verdade, o processo de troca de marcha é finalizado através da mudança de estado da variável *isUpPressed* e chamada dos métodos de ativação da subida de marcha e embreagem, nesta ordem e mudança na referência de troca de marcha, caso seja necessário para uso posterior.

O processo de redução de marcha é semelhante ao de subida, porém, havendo a troca das chamadas do método *upshift()* para o *downshift()*.

Entre as diferenças dos passos de redução de marcha, podem ser citadas a verificação de marcha atual, garantindo que a marcha seja maior que a 1^o, além de que as variáveis de controle da embreagem e de acionamento da subida estejam em nível baixo. Após esta verificação, o fluxo normal é seguido, sendo chamado o método responsável pelo *downshift* após o tempo de acionamento da embreagem e, após o tempo definido, todo o sistema é posto em nível baixo, encerrando o processo de troca.

Devido à possibilidade de acessar a primeira marcha através do pressionamento do botão de descida de marcha (lógica da transmissão sequencial de motos), foi adicionada à estrutura a verificação de marcha sendo 0 e outros passos já citados.

Prosseguindo, é verificado o estado da variável de controle da embreagem e o estado atual de troca (descida), caso o tempo seja suficiente para acionar o atuador pneumático da embreagem e a marcha atual seja 0 (neutro), é chamado o método responsável pela troca no neutro e passado o tipo 'u' como parâmetro, representando uma subida de marcha.

3.1 ORÇAMENTO DE COMPONENTES

Após pesquisa em algumas plataformas de venda on-line e, levando em conta o acesso a componentes mais gerais localmente, como *pushbuttons* e MOSFETs, a lista de compras (*Bill of Materials*) simplificada para a construção de um teste parcial é composta dos itens apresentados na figura 32 a seguir:

Figura 32 - Bill of Material para construção básica do circuito de troca de marchas

Bill of Material para construção da versão básica do sistema de troca de marchas						
Material	Local de Compra	Preço Unitário (R\$)	Quantidade	Frete (R\$)	Preço total (R\$)	
Válvula Solenoide 3/2 Vias Retorno Mola Rosca 1/4 12v	Mercado Livre	R\$ 60,00	3	R\$ -	R\$ 180,00	
Cilindro Pneumático Mini 6432 Dupla Ação Ø20 X 60mm Curso	Mercado Livre	R\$ 117,71	2	R\$ -	R\$ 235,42	
Ssr-100dd Relé De Estado Sólido Dc 100a Marcas Diversas	Mercado Livre	R\$ 140,30	3	R\$ -	R\$ 420,90	
MOSFET tipo N - IRF640N	Tokio Componente	R\$ 5,00	12	R\$ -	R\$ 60,00	
Pushbuttons + Solda de Tungstênio	Luane Componente	R\$ 24,00	4	R\$ -	R\$ 96,00	
Total					R\$ 992,32	

Fonte: Autoria Própria (2021).

Em breves termos, as 3 válvulas solenoide foram definidas com base nas três saídas provenientes do Arduino (saídas para subir a marcha, descer a marcha e entrar no neutro), 2 cilindros pneumáticos, sendo 1 destinado para a troca de marchas (alavanca de marcha do motor) e outra, para a embreagem. 3 relés de estado sólidos, que estão associados a cada uma das válvulas solenoides. Em relação aos MOSFETs, são necessários apenas 3 para a configuração definida no Proteus, porém, na possibilidade de ter ocorrido algum teste prático alternativo (que não houve), 6 ou mais se fariam necessários. Os *pushbuttons* e soldas foram inseridos para evitar quaisquer tipos de mau contato, pois, as versões mais simples dos *pushbuttons* apresentam uma taxa de falha nos sinais razoavelmente elevada, o que poderia interferir na qualidade dos testes.

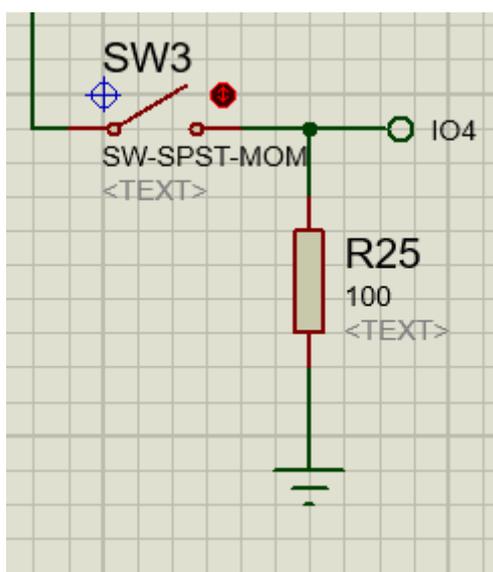
Conforme exibido na figura 32, apenas 3 válvulas solenoides foram obtidas, sendo requisitados pelos dois atuadores de dupla ação um mínimo de 4, ou seja, apenas 1 atuador pôde ser configurado para os testes.

4 RESULTADOS E DISCUSSÃO

Para a confirmação de funcionamento do código desenvolvido ao longo dos últimos meses, foi adotada a ferramenta de simulação Proteus, desenvolvido pela LabCenter com foco na construção de placas PCB e circuitos gerais desenvolvidos com base em Eletrônica Analógica/Digital. Sua adoção, ao invés do Multisim, se deu pela maior disponibilidade de components, desde resistores comerciais a switches bem específicos, por exemplo, além de suporte para Firmware, com o uso de Arduino, como no projeto atual.

Com base no código desenvolvido na seção 3 (Implementação) e tendo em vista a necessidade de garantir que o sinal de entrada do microcontrolador atrelado ao Arduino não ficasse com tensão flutuante (sem tensão de referência), foi adotado um push-down para permitir que o nível lógico alto chegasse ao pino digital do Arduino somente na situação de pressionamento do botão/alavanca, e, para os demais casos, nível lógico zero/baixo. A seguir é exibida a figura 33 representando esta estrutura:

Figura 33 - Representação de um circuito push-down para polarização dos pinos digitais do Arduino



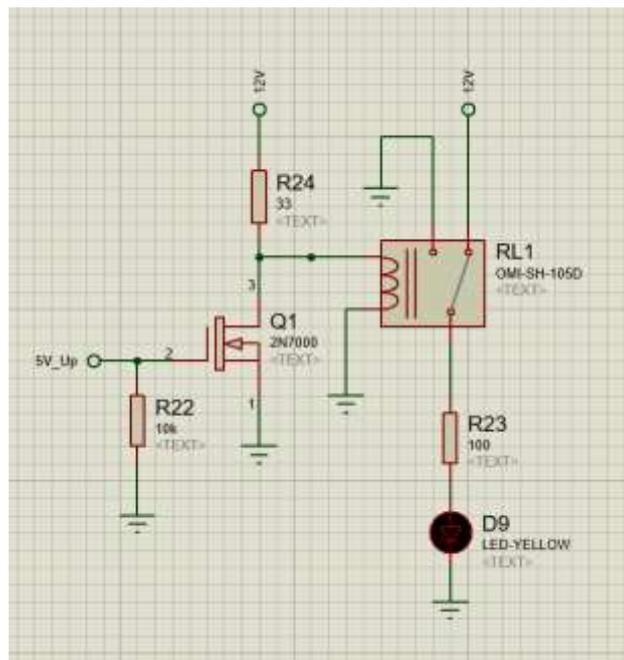
Fonte: Autoria Própria (2021).

Refrisando: Caso a chave esteja aberta, o resistor de 100 Ω polariza com nível lógico baixo, senão, 5V são fornecidos através da chave.

Conforme apresentado no tópico anterior, a respeito do código desenvolvido na IDE do Arduino, foram utilizados os mesmos pinos digitais para entrada e saída na plataforma. Caso a saída estivesse em nível alto, para a lógica original do sistema, um pulso era transmitido e recebido pela válvula solenoide, a qual direcionaria o fluxo de ar de acordo com a origem (para cima, caso o comando fosse para subir marcha e vice-versa). No entanto, devido ao uso de

MOSFETs tipo N, sendo, para a simulação, o modelo 2N7000, é necessário que uma tensão de polarização abaixo do *threshold* seja estabelecida (normalmente menor que 3V), para que o dispositivo entre na região de corte e a lógica opere conforme esperado. De modo resumido: nas configurações físicas do sistema, é necessário que as portas ligadas aos MOSFETs fiquem inicialmente em nível alto, a fim de deixar o dispositivo entre as regiões de triodo (linear) e saturação. A figura 34 exibe uma das configurações para polarização de válvulas solenoide:

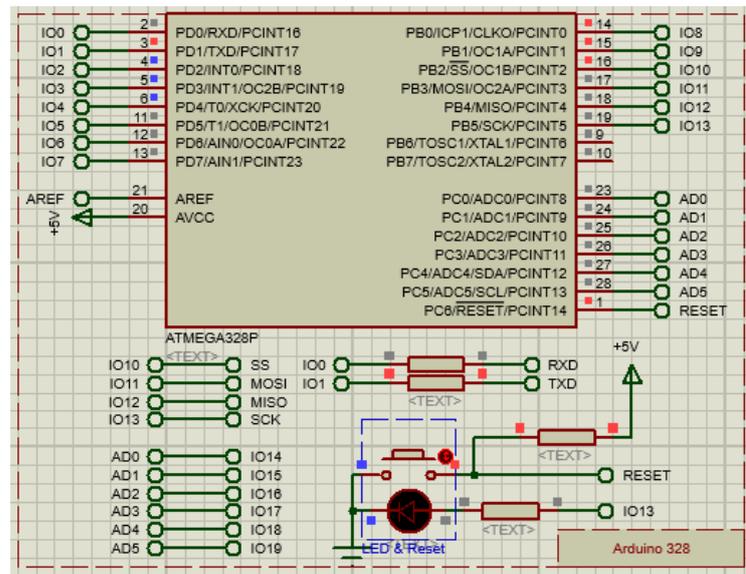
Figura 34 - Circuito responsável por polarizar o relé de estado sólido



Fonte: Autoria Própria (2021).

Conforme citado em tópicos anteriores, foi-se utilizado o Arduino Uno para implementação, mesmo que virtual, do sistema, devido à sua disponibilidade e simplicidade inicial da proposta após mudanças. A figura 35 apresenta o circuito de polarização deste, assim como os pinos digitais utilizados na aplicação e mencionados na seção de Implementação.

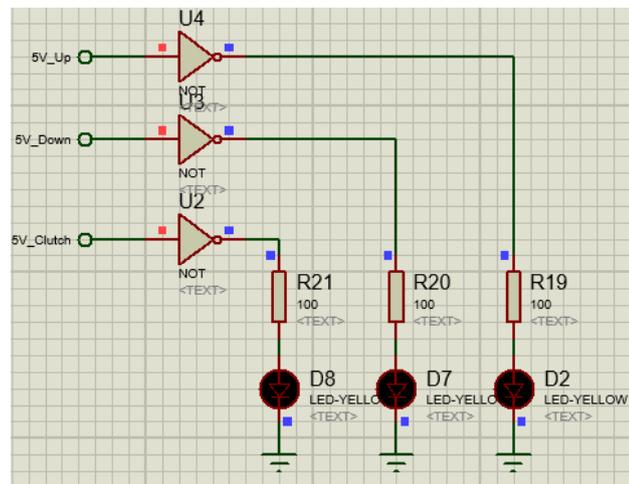
Figura 35 - Configuração do Arduino 328p utilizado na simulação da plataforma Proteus



Fonte: Autoria Própria (2021).

O circuito, em seu estado normal (não ativo), possui a característica de enviar corrente para o MOSFET tipo N utilizado na simulação, como forma de deixá-lo em estado de corte, de acordo com a sua curva característica apresentada nos referenciais teóricos deste trabalho. A figura 36 apresenta um esquema para visualização dos estados, sendo utilizados uma parte inversora (NOT), devido à adaptação do código ao MOSFET e diodos tipo LED.

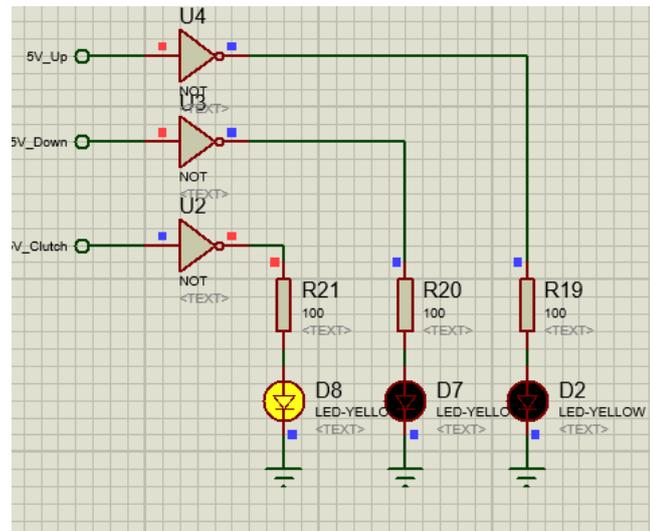
Figura 36 - Circuito para verificação de estado atual da embreagem, subida e descida de marcha



Fonte: Autoria Própria (2021).

Durante o processo de troca de marchas, esta versão do sistema necessita que a embreagem seja ativada, sendo a representação de seu estado atual exibido na figura 37:

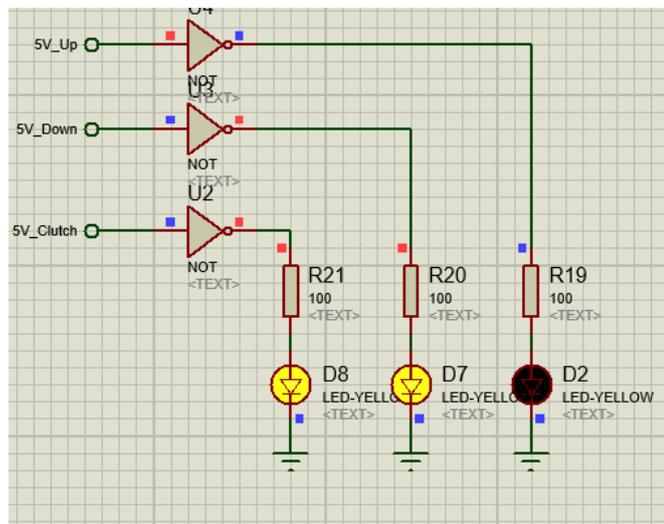
Figura 37 - Sinalizador de funcionamento da embreagem



Fonte: Autoria Própria (2021).

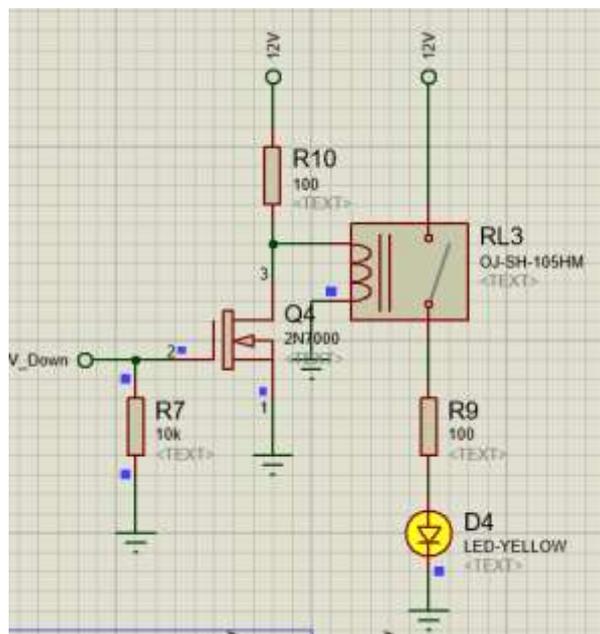
Em se tratando de saída do neutro ou descida de marchas, o segundo LED é ativado após certo tempo de operação da embreagem, onde atualmente depende de um valor pré-definido, podendo, em versão mais atualizada e de melhor *performance*, estar diretamente ligado ao sinal enviado pelo sensor de marcha do veículo. A figura 38 apresenta uma imagem dos atuadores da embreagem e descida de marcha em estado ativo:

Figura 38 - Embreagem e atuador em posição de redução da marcha (representação) ativados



Fonte: Autoria Própria (2021).

Figura 39 - Circuito de alimentação do atuador responsável por reduzir a marcha em estado ativo



Fonte: Autoria Própria (2021).

Para a figura 39, representada acima, o LED acionado representa a válvula solenoide polarizada, pois o seu circuito somente pode ser representado em plataforma com suporte a itens pneumáticos, como a fornecida pela FESTO. O relé de estado sólido foi utilizado na aplicação devido a menores níveis de ruídos, quando comparado ao relé mecânico, além de possuir um melhor controle, pois possui apenas componentes eletrônicos em sua estrutura. O modelo em questão é o OMI-SH-105D, o qual apresenta tensão de polarização de 5V, sendo uma chave aberta em seu lado direito (carga), caso o nível de tensão esteja abaixo deste limiar.

Devido a não disponibilidade de relés de estado sólido com tensão de polarização DC, foi encontrado através de pesquisa em sites de compra/venda um modelo com tensão de entrada variando entre 3 e 32V DC e intervalo de tensão de carga de 5 a 220 VDC. A marca em questão é *Fotek*, com fabricação no Taiwan e modelo 100DD 5-220VDC, conforme apresentado na seção de metodologias.

O atuador pneumático, alimentado pela válvula solenoide, é encaixado na alavanca de troca de marchas do motor, durante a concepção do projeto.

CONCLUSÃO

Este trabalho teve o intuito de desenvolver um sistema de troca de marchas com aplicação para veículos da categoria *Fórmula SAE*, apresentando como métricas: baixo custo, fácil manutenibilidade e alta confiabilidade, com base em um sistema pneumático de acionamento. Foi apresentado um referencial teórico abordando os principais itens adotados no desenvolvimento da pesquisa e construção dos protótipos, mesmo virtuais, sendo eles: Sistemas pneumáticos, componentes a serem utilizados e a plataforma microprocessada para gerenciamento dos sinais do sistema.

Como pilar inicial, desenvolveu-se o fluxograma do funcionamento de cada opção fornecida ao piloto (subir/descer marchas e entrar/sair do neutro), a fim de simplificar a estrutura que posteriormente seria desenvolvida. Em seguida, deu-se início ao desenvolvimento do código do sistema, com todas as validações para possíveis erros de operação por parte do piloto, como forma de garantir a sua integridade e permitir um funcionamento adequado, através da IDE do Arduino e uma placa Uno da mesma marca. E, por fim, realizou-se a construção do esquemático na plataforma *Proteus*, contendo Arduino, transistores de efeito de campo tipo N (MOSFETs), relés de estado sólido e indicadores de sinal (LEDs), a fim de confirmar a operação correta do sistema desenvolvido.

Em se tratando de orçamento dos componentes, uma lista simplificada, contendo apenas a base para testes de funcionalidade do sistema, teve um valor estipulado em R\$ 992,32 desconsiderando custos logísticos, ponto este que influenciou no andamento das atividades e somente foi finalizado (aquisição) às vésperas da entrega do presente trabalho.

Como pontos de melhoria, podem ser citados: inclusão de sensor de marcha do veículo como entrada no sistema, código adaptativo para o tempo de acionamento dos atuadores e uso de *flat shifting*. O primeiro possibilitaria a construção de um sistema em malha fechada com realimentação, podendo ser realizadas novas operações automaticamente, sem quaisquer intervenções por parte do piloto. O segundo se trata no desenvolvimento básico na estrutura do código para suporte a adaptações nos tempos de acionamento, como forma de alcançar uma margem mínima de tempo em ociosidade e permitir um desempenho máximo. O terceiro e último, equivale ao corte de injeção/ignição durante trocas de marcha, com o intuito de remover total/parcialmente o uso da embreagem, tornando-se, assim, um item opcional, diferentemente da estrutura atual proposta.

Por fim, os resultados através de plataforma de desenvolvimento de Software Embarcado (*Proteus*) se mostraram positivos, principalmente em função da resposta consistente com a operação solicitada, mesmo durante os fluxos de exceção, propiciando uma

confiabilidade no desempenho físico, em futuros testes de pista. Trabalhos futuros podem englobar testes práticos, visando coleta de dados para amostragem e detecção de possíveis falhas, de forma a melhorar a robustez do sistema

REFERÊNCIAS BIBLIOGRÁFICAS

ARDUINO. Arduino Uno Rev-3. Arduino. 2020. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acesso em: 02 de novembro de 2020.

ARDUINO. What is Arduino?. Arduino. 2020. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction/>>. Acesso em: 02 de novembro de 2020.

BARRAGÁN, Hernando. The Untold History of Arduino. GitHub. 2020. Disponível em: <<https://arduinohistory.github.io/>>. Acesso em: 02 de novembro de 2020.

BOYLESTAD, Robert L. Introdução à Análise de Circuitos. 10ª. Ed. Prentice Hall/Pearson, 2013.

BSIM RACING. Fanatec Introduces the Clubsport Steering Wheel F1 2018 – Podium Advanced Paddle Module. 2018. Disponível em: <<https://www.bsimracing.com/fanatec-introduces-the-clubsport-steering-wheel-f1-2018-podium-advanced-paddle-module/>>. Acesso em: 28 de set. 2020.

EEEP. Curso Técnico em Manutenção Automotiva. [S.l.]: Escola Estadual de Educação Profissional, 2014.

MERCADO LIVRE. SSR-100dd Relé de Estado Sólido DC 100a Marcas Diversas. 2021a. Disponível em: <https://produto.mercadolivre.com.br/MLB-1927731965-ssr-100dd-rele-de-estado-solido-dc-100a-marcas-diversas-_JM?quantity=3>. Acesso em: 19 de dezembro de 2021.

MERCADO LIVRE. Válvula Solenoide 3/2 Vias Retorno Mola Rosca ¼ 12V. 2021b. Disponível em: <https://produto.mercadolivre.com.br/MLB-1013050630-valvula-solenoide-32-vias-retorno-mola-rosca-14-12v-_JM?quantity=3>. Acesso em: 19 de dezembro de 2021.

MERCADO LIVRE. Cilindro Pneumático Mini 6432 Dupla Ação d20 x 60mm Curso. 2021c. Disponível em: <https://produto.mercadolivre.com.br/MLB-887830357-cilindro-pneumatico-mini-6432-dupla-aco-20-x-60mm-curso-_JM?quantity=2>. Acesso em: 19 de dezembro de 2021.

MoTeC. M1 GPRP Guide - Motec. 2017a. Disponível em: <<https://www.motec.com.au/forum/viewtopic.php?f=39&t=3235>>

MoTeC. M1 GPRP Guide - Motec. 2017b. Disponível em: <<https://www.motec.com.au/forum/viewtopic.php?f=39&t=3235>>

MoTeC. M1 GPRP Guide - Motec. 2017c. Disponível em: <<https://www.motec.com.au/forum/viewtopic.php?f=39&t=3235>>

MoTeC. M1 GPRP Guide - Motec. 2017d. Disponível em: <<https://www.motec.com.au/forum/viewtopic.php?f=39&t=3235>>

MoTeC. M1 GPRP Guide - Motec. 2017e. Disponível em: <<https://www.motec.com.au/forum/viewtopic.php?f=39&t=3235>>

OLIVEIRA, Ricardo. Paddle Shift: Entenda o que é. Notícias Automotivas. 2018?. Disponível em: <<https://www.noticiasautomotivas.com.br/paddle-shift-entenda-o-que-e/>>. Acesso em: 28 de setembro de 2020.

PERITO EM PHP. Principais funções da linguagem C e C++. 2019. Disponível em: <<https://peritoemphp.com/principais-funcoes-da-linguagem-c-e-c/>>. Acesso em: 27 de dezembro de 2021.

PINHO, Marcio Sarroglia. Histórico da Linguagem C. Pontifícia Universidade Católica - RS. 2020?. Disponível em: <<https://www.inf.pucrs.br/~pinho/LaproI/Historico/Historico.htm>>. Acesso em: 02 de novembro de 2020.

QUICKSHIFTER. Quickshifter – Modelo Standard. 2021. Disponível em: <<https://quickshifter.com.br/ver-produto.php?id=1&MODELO%20STANDARD>>. Acesso em: 19 de dezembro de 2021.

RACING HALL OF FAME COLLECTION. Max Verstappen Red Bull Racing Raced used F1 Steering Wheel. 2019?. Disponível em: <<https://www.racinghalloffamecollection.com/product/2018-max-verstappen-red-bull-racing-race-used-f1-steering-wheel-2/>>. Acesso em: 28 de set. 2020.

SAE BRASIL. Classificação Geral. 2019. Disponível em: <<http://saebrasil.org.br/wp-content/uploads/2020/03/Classifica%C3%A7%C3%A3o-Final-15a-Competi%C3%A7%C3%A3o-F%C3%B3rmula-SAE-BRASIL-PETROBRAS-2018-Combust%C3%A3o.pdf>> . Acesso em: 25 de setembro de 2020.

SAE BRASIL. Conheça a SAE no Brasil – SAE no Brasil. 2020. Disponível em: <<https://saebrasil.org.br/quem-somos/>>. Acesso em: 25 de setembro de 2020.

SAE BRASIL. Fórmula SAE. 2020. Disponível em: <<https://saebrasil.org.br/programas-estudantis/formula-sae-brasil/>> . Acesso em: 26 de setembro de 2020.

SAE INTERNATIONAL. About SAE International - About. 2020. Disponível em: <<https://www.sae.org/about>>. Acesso em: 23 de setembro de 2020.

SAE INTERNATIONAL. About SAE International – History. 2020. SAE. Disponível em: <<https://www.sae.org/about/history>> . Acesso em: 25 de setembro de 2020.

SOARES, C. G. C. .; DEL PINO, G. G. .; BEZAZI, A. .; LOPES, A. P. .; TORRES, A. R. .; KIELING, A. C. .; DO SANTOS, M. D. .; LLERENA , G. F. R. . Estudo comparativo entre transmissões automotivas automáticas e manuais: Comparative study between automatic and manual automotive transmissions. **Latin American Journal of Development**, [S. l.], v. 3, n. 5, p. 2801–2818, 2021. DOI: 10.46814/lajdv3n5-003. Disponível em: <https://latinamericanpublicacoes.com.br/ojs/index.php/jdev/article/view/731>. Acesso em: 19 dezembro 2021.

WEIHERMANN, Henrique Weber. Estudo sobre aplicação de transmissão continuamente variável para veículos de pequeno porte. Trabalho de Conclusão de Curso -Universidade Federal de Santa Catarina, Santa Catarina, Joinville, 2015

APÊNDICE A – CÓDIGO PARA TROCA DE MARCHAS

Código responsável pela troca de marchas

```

/*
 * CRIADO POR: Ewerton Cassiano Xavier
 * MÊS/ANO: 12/2021
 * Aplicação: Sistema de troca de marchas via borboleta para carros fórmula
SAE (TCC)
 * Orientador: Daniel Guzman del Rio
 *
 */
int downGear = 2;
int upGear = 3;
int neutralButton = 4;
int clutchOut = 8;
int downGearOut = 9;
int upGearOut = 10;

boolean isUpPressed = false;
boolean isDownPressed = false;
boolean isClutchOn = false;
boolean isNeutralOn = false;
int clutchTime = 500;
int shiftTime = clutchTime * 2;
unsigned long time = 0;

int gearCounter = 0;
int neutralTime = 80;
char type = 'u';

void setup() {
  // put your setup code here, to run once:
  pinMode(downGear, INPUT);
  pinMode(upGear, INPUT);
  pinMode(neutralButton, INPUT);
  pinMode(clutchOut, OUTPUT);
  pinMode(downGearOut, OUTPUT);
  pinMode(upGearOut, OUTPUT);

  digitalWrite(clutchOut, LOW);
  digitalWrite(downGearOut, LOW);
  digitalWrite(upGearOut, LOW);

  Serial.begin(9600);
}

void activateClutch (boolean state) {
  switch (state) {
    case true:
      digitalWrite(clutchOut, HIGH);
      break;
    default:
      digitalWrite(clutchOut, LOW);
      break;
  }
}

void upshift (boolean state) {
  switch (state) {

```

```

        case true:
            digitalWrite(upGearOut, HIGH);
            break;
        default:
            digitalWrite(upGearOut, LOW);
            break;
    }
}

void downshift (boolean state) {
    switch (state) {
        case true:
            digitalWrite(downGearOut, HIGH);
            break;
        default:
            digitalWrite(downGearOut, LOW);
            break;
    }
}

void neutralShift (boolean state, char type) {
    Serial.println(type);
    if(type == 'd') {
        switch(state) {
            case true:
                digitalWrite(upGearOut, HIGH);
                break;
            default:
                digitalWrite(upGearOut, LOW);
                break;
        }
    } else {
        switch(state) {
            case true:
                digitalWrite(downGearOut, HIGH);
                break;
            default:
                digitalWrite(downGearOut, LOW);
                break;
        }
    }
}

void loop() {
    // UPSHIFT CODE
    if(digitalRead(upGear) == HIGH and isClutchOn == false and isDownPressed ==
false and gearCounter < 6 and gearCounter >= 1) {
        isUpPressed = true;
        isClutchOn = true;
        activateClutch(isClutchOn);
        gearCounter = gearCounter + 1;
        Serial.println(gearCounter);
        time = millis();
    }
    if(isClutchOn == true and isUpPressed == true) {
        if((millis() - time) >= clutchTime) {
            isClutchOn = false;
            upshift(isUpPressed);
            time = millis();
        }
    }
}

```

```

}

if((isClutchOn == false) and (isUpPressed == true)) {
  if((millis() - time) > shiftTime) {
    isUpPressed = false;
    upshift(isUpPressed);
    activateClutch(isClutchOn);
    time = millis();
  }
}

// DOWNSHIFT CODE

if(digitalRead(downGear) == HIGH and isClutchOn == false and isUpPressed
== false and gearCounter > 1) {
  gearCounter = gearCounter - 1;
  Serial.println(gearCounter);
  isDownPressed = true;
  isClutchOn = true;
  activateClutch(isClutchOn);
  time = millis();
} else if (digitalRead(downGear) == HIGH and isClutchOn == false and
isUpPressed == false and gearCounter == 0) {
  isDownPressed = true;
  isClutchOn = true;
  activateClutch(isClutchOn);
  time = millis();
}

if(isClutchOn == true and isDownPressed == true) {
  if((millis() - time) > clutchTime) {
    isClutchOn = false;
    if(gearCounter == 0) {
      gearCounter = gearCounter + 1;
      Serial.println(gearCounter);
      neutralShift(isDownPressed, 'u');
    } else {
      type = 'a';
      downshift(isDownPressed);
    }
  }
  time = millis();
}

if((isClutchOn == false) and (isDownPressed == true)) {
  if((millis() - time) > shiftTime) {
    isDownPressed = false;
    if(gearCounter == 1 and type == 'u') {
      neutralShift(isDownPressed, 'u');
    } else {
      downshift(isDownPressed);
    }
  }
  activateClutch(isClutchOn);
  time = millis();
}

}

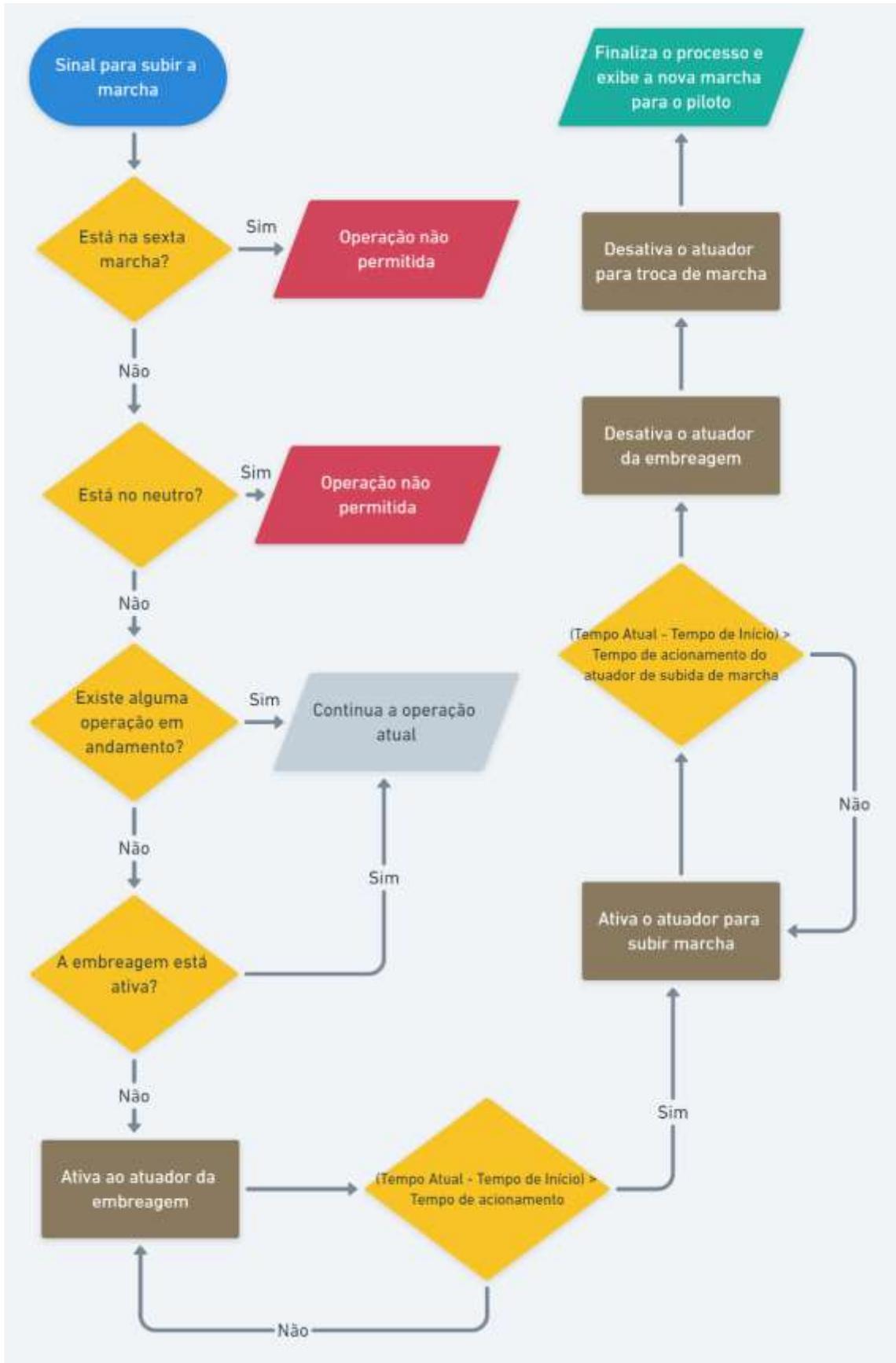
// NEUTRO

if(digitalRead(neutralButton) == HIGH and gearCounter == 1 and isClutchOn
== false) {
  isNeutralOn = true;
  isClutchOn = true;
}

```

```
gearCounter = 0;
  Serial.println(gearCounter);
  activateClutch(isClutchOn);
  type = 'd';
  time = millis();
}
if(isClutchOn == true and isNeutralOn == true) {
  if((millis() - time) > clutchTime) {
    isClutchOn = false;
    neutralShift(isNeutralOn, type);
    time = millis();
  }
}
if((isClutchOn == false) and (isNeutralOn == true)) {
  if((millis() - time) > neutralTime) {
    isNeutralOn = false;
    neutralShift(isNeutralOn, type);
    activateClutch(isClutchOn);
    time = millis();
  }
}
}
```

APÊNDICE B – FLUXOGRAMAS DO SISTEMA (SUBIR MARCHA)



APÊNDICE C – FLUXOGRAMAS DO SISTEMA (NEUTRO)

