

**CENTRO PAULA SOUZA**  
**FACULDADE DE TECNOLOGIA SANTO ANDRÉ**  
**Tecnologia em eletrônica automotiva**

**Deivison Pereira Dos Santos**  
**Moabe Lima De Souza Filho**  
**Thierry Felipe Santos Borges**

**DESENVOLVIMENTO DO CONCEITO DO SOFTWARE DE UMA**  
**TRANSMISSÃO AUTOMATIZADA**

**Santo André**

**2020**

**Deivison Pereira Dos Santos**

**Moabe Lima De Souza Filho**

**Thierry Felipe Santos Borges**

**DESENVOLVIMENTO DO CONCEITO DO SOFTWARE DE UMA  
TRANSMISSÃO AUTOMATIZADA**

Trabalho de Conclusão de Curso apresentado ao Curso Tecnológico em Eletrônica Automotiva da FATEC Santo André, orientado pelo Prof. Dr. Armando Antônio Maria Laganá e coorientado pelo Prof. Paulo Alexandre Pizará Hayashida como requisito parcial para obtenção do título de tecnólogos em Eletrônica Automotiva.

**Santo André**

**2020**

S237d

Santos, Deivison Pereira dos  
Desenvolvimento do conceito do software de uma transmissão automatizada / Deivison Pereira dos Santos, Moabe Lima de Souza Filho, Thierry Felipe Santos Borges. - Santo André, 2020. – 143f: il.

Trabalho de Conclusão de Curso – FATEC Santo André.  
Curso de Tecnologia em Eletrônica Automotiva, 2020.

Orientador: Prof. Armando Antônio Maria Laganá

1. Eletrônica. 2. Transmissão automatizada. 3. Eletroválvulas. 4. Câmbio. Sistema mecatrônico. 5. Dualogic. I. Souza Filho, Moabe Lima de. II. Borges, Thierry Felipe Santos III. Desenvolvimento do conceito do software de uma transmissão automatizada.

629.2

**Deivison Pereira Dos Santos**

**Moabe Lima De Souza Filho**

**Thierry Felipe Santos Borges**

**DESENVOLVIMENTO DO CONCEITO DO SOFTWARE DE UMA  
TRANSMISSÃO AUTOMATIZADA**

Trabalho de Conclusão de Curso  
apresentado a FATEC SANTO ANDRÉ como  
requisito parcial à obtenção de título de Tecnólogo  
em Eletrônica Automotiva.

**BANCA EXAMINADORA**

**Local: Fatec Santo André**

**Horário: 16:00**

**Data: 07/07/2020**

Prof. Dr. Armando Antônio Maria Laganá  
Presidente da Banca  
Fatec Santo André

Prof. Paulo Alexandre Pizará Hayashida  
Primeiro Membro da Banca  
Fatec Santo André

Prof. Fernando Garup Dalb  
Segundo Membro da Banca  
Fatec Santo André

**SANTO ANDRÉ**

**2020**

## **AGRADECIMENTO**

Primeiramente gostaríamos de agradecer a Deus, pois se não fosse por Sua permissão, não teríamos chegado neste momento. Agradecemos também à instituição FATEC Santo André e aos nossos professores Laganá e Paulo Hayashida que nos orientaram nesta caminhada. A cada amigo que nos apoiaram e nos prestaram toda assistência, nosso muito obrigado!

Não poderíamos passar por esse momento sem agradecer aos nossos familiares, os quais nos ajudaram e nos compreenderam nos momentos difíceis, e nos prestaram total suporte quando preciso, nos dando ânimo e motivação para prosseguirmos.

*“Só eu sei cada passo por mim dado nessa estrada esburacada que é a vida, passei coisas que até mesmo Deus duvida, fiquei triste, capiongo, aperreado, porém nunca me senti desmotivado, me agarrava sempre numa mão amiga, e de forças minha alma era munida pois do céu a voz de Deus dizia assim: ‘suba o queixo, meta os pés, confie em mim, vá pra luta que eu cuido das feridas’ “*

Bráulio Bessa

## RESUMO

O sistema de transmissão automatizada, ou semiautomático, se trata de basicamente um câmbio manual com um sistema mecatrônico o qual é responsável por realizar as trocas de marcha sem o pedal da embreagem, ou seja, é um aprimoramento dos sistemas manuais. Mesmo havendo benefícios, o câmbio semiautomático ainda possui uma falta de suavidade na hora de realizar as trocas de marchas, devido aos trancos que o motorista sente ao dirigir o automóvel. O projeto proposto é a construção de um sistema que controle as eletroválvulas da transmissão. Realizando testes dos sensores e eletroválvulas que fazem parte do sistema de transmissão para assim adquirir os dados necessários para o efetuar o seu controle. O trabalho apresentado utiliza como base o projeto “ESTUDO E DESENVOLVIMENTO DE UMA UNIDADE DE GERENCIAMENTO ELETRÔNICO QUE SIMULE A ATUAÇÃO NAS ELETROVÁLVULAS DA TRANSMISSÃO AUTOMATIZADA” (2017) realizado pelos graduados da FATEC SANTO ANDRÉ Gustavo Bento, Gustavo Oliveira e Willian Cesário.

Palavras chaves: Transmissão, Dualogic®, Eletroválvulas, automatizada.

## **ABSTRACT**

The automated or semi-automatic transmission system is basically a manual gearbox with a mechatronic system which is responsible for shifting without the clutch pedal, it is an improvement of the manual systems. Even with benefits, the semi-automatic gearbox still has a lack of smoothness when shifting, due to the locks the driver feels when driving the car. The proposed project is the construction of a system that controls the transmission solenoid valves. Performing tests of the sensors and solenoid valves that are part of the transmission system in order to acquire the necessary data to perform its control. The work presented is based on the project "STUDY AND DEVELOPMENT OF AN ELECTRONIC MANAGEMENT UNIT SIMULATING ACTION IN AUTOMATED TRANSMISSION ELECTROVALVES" (2017) conducted by FATEC graduates SANTO ANDRÉ Gustavo Bento, Gustavo Oliveira and Willian Cesário.

Keywords: Transmission, Dualogic®, Solenoid Valves, automated.



## LISTA DE ILUSTRAÇÕES

Figura 1 - Fluxo de massa e energia em um motor de combustão interna	22
Figura 2 - MCI, peças fixas	23
Figura 3 - MCI, peças móveis	24
Figura 4 - Ciclos e tempos de um MCI	25
Figura 5 - Torque aplicado ao abrir uma porta	26
Figura 6 - Sistema biela-manivela e ângulo forma pela biela e o eixo de manivelas	28
Figura 7 - Disposição das forças no momento da expansão dos gases	29
Figura 8 - Polias e correia	31
Figura 9 - Engrenagens	32
Figura 10 - Trem de força	33
Figura 11 - Conjunto Powertrain	34
Figura 12 - Conjunto de engate e seleção da transmissão manual	36
Figura 13 - Demonstração do funcionamento do câmbio manual	37
Figura 14 - Diferencial em corte e seus componentes internos	38
Figura 15 - Transmissão automática em corte	39
Figura 16 - Polias da transmissão CVT	40
Figura 17 - Transmissão automatizada em corte	41
Figura 18 - Transmissão Dualogic® - FIAT	42
Figura 19 - Conjunto eletro-hidráulico	44
Figura 20 - Reservatório do óleo do conjunto hidráulico	45
Figura 21 - Indicadores de mínimo e máximo do nível de óleo	46
Figura 22 - Eletrobomba de óleo	46
Figura 23 - Acumulador de óleo sob pressão	47
Figura 24 - Defletor de óleo	47
Figura 25 - Conjunto eletro-hidráulico	48
Figura 26 - Circuito de pressão hidráulica das válvulas EV1 e EV2	49
Figura 27 - Posição de montagem das Válvulas EV3 e EV4	50
Figura 28 - Circuito de pressão hidráulica das válvulas EV3 e EV4	50
Figura 29 - Posição de montagem da válvula EV0	51
Figura 30 - Gráfico da vazão/corrente conforme posição da válvula EV0	52
Figura 31 - Eixo de comando do câmbio	53

Figura 32 - Flange das eletroválvulas _____	53
Figura 33 - Atuador da embreagem _____	54
Figura 34 - Circuito hidráulico completo das eletroválvulas _____	55
Figura 35 - Joystick _____	56
Figura 36 - Diagrama em blocos da matriz de diodos _____	57
Figura 37 - Sistema das borboletas do volante _____	59
Figura 38 - Esquema de um sensor PLCD _____	60
Figura 39 - Ilustração dos campos magnéticos no sensor PLCD e gráfico das tensões (1) _____	61
Figura 40 - Ilustração dos campos magnéticos no sensor PLCD e gráfico das tensões (2) _____	62
Figura 41 - Esquema simplificado de um sensor capacitivo de placas paralelas ____	63
Figura 42 - Esquema de um sensor de relutância magnética _____	64
Figura 43 - tensão gerada pelo sensor de relutância magnética _____	64
Figura 44 - Diagrama de Blocos do MC33816 _____	67
Figura 45 - Exemplo de ligação esquemática no MC33816 _____	68
Figura 46 - BOB _____	69
Figura 47 - sinais da eletroválvula de seleção EV3 _____	71
Figura 48 - sinais da eletroválvula de seleção EV1 _____	72
Figura 49 - Sinais EV1 e EV2 1° e 3° marcha _____	73
Figura 50 - Sinais EV1 e EV2 3° e 5° marcha _____	74
Figura 51 - Sinais dos sensores do atuador de engate e seleção _____	75
Figura 52 - Configuração da curva do pulso _____	77
Figura 53 - Configuração dos registradores do Micro Code _____	78
Figura 54 – Endereço da Memória de Dados _____	79
Figura 55 – Ferramentas para configuração de parâmetros de trabalho _____	79
Figura 56 - Compilador SPIGen do MC33816 _____	80
Figura 57 - Controles das sequências das saídas do MC33816 _____	81
Figura 58 – Comparação da sequência das saídas do MC33816 com acionamento	82
Figura 59 - Analisador logico 24MHz de 8 canais _____	83
Figura 60 - Comunicação SPI do MC33816 _____	83
Figura 61 - Registradores do MC33816 _____	85
Figura 62 - Amplificador operacional em modo comparador _____	91
Figura 63 - Amplificador operacional em modo seguidor de tensão _____	92

Figura 64- Kit Dualogic® funcionando via Software _____	93
Figura 65: Teste do sensor de pressão _____	95
Figura 66 - Comunicação SPI entre PIC e MC _____	96

## LISTA DE TABELAS

Tabela 1- Vantagem e desvantagem das transmissões	43
Tabela 2 - Tabela da ordem lógica de acionamento das válvulas	55
Tabela 3 - Lógica do sensor de joystick	58
Tabela 4 - PIN OUT NCD	70
Tabela 5 - Tensões nos atuadores de engate e seleção	76
Tabela 6 - Pinos comunicação SPI	84

## LISTA DE ABREVIATURAS E SIGLAS

MCI – Motores a combustão interna

GNV - Gás natural veicular

PMI – Ponto morto inferior

PMS – Ponto morto superior

D - Diâmetro do cilindro[mm];

S - Curso do êmbolo[mm];

R - Raio de giro da árvore de manivelas[mm];

L - Comprimento total biela[mm];

Vc - Volume comprimido do cilindro[mm<sup>3</sup>];

Zk - Distância entre o centro da árvore de manivelas e o pino do êmbolo[mm];

$\alpha$  - Ângulo entre o raio de giro e a distância entre o centro da árvore de manivelas[°].

TDC - Top Dead Center/ponto morto superior

BDC - Bottom Dead Center/Ponto Morto Inferior

Fg - Força gerada pela expansão dos gases

Ft - Força de forma tangencia

Fb - Força biela

Fn - Força lateral do pistão

Fr - Força radial

CVT - *Continuously variable transmission* - Transmissão continuamente variável

ECU - *Engine Control Unit* (Unidade de controle automotivo)

A – Ampere

NCD – Nó do Câmbio Dualogic®

NA – Normalmente aberto

VS - Tensão de saída

TI – *Texas Instruments*

CI – Circuito integrado

RAM - Memória de acesso aleatório

BOB – Break Out Box

PWM - Modulação por largura de pulso - *Pulse width modulation*

## Sumário

1	INTRODUÇÃO.....	19
1.1	Justificativa.....	20
1.2	Objetivos .....	20
1.2.1	Objetivos gerais.....	20
1.2.2	Objetivos específicos.....	20
1.3	Estrutura e Organização.....	20
2	Revisão bibliográfica.....	22
2.1	Motores de combustão interna .....	22
2.1.1	Principais componentes dos motores de combustão interna .....	23
2.1.2	Ciclos e tempos de um motor de combustão interna .....	24
2.1.3	Torque .....	26
2.1.3.1	Torque gerado a partir do sistema biela-manivela .....	27
2.1.3.2	Força de inercia .....	30
2.2	Transmissão de torque.....	30
2.2.1	Transmissão por polias e correia .....	31
2.2.2	Transmissão por engrenagens.....	32
2.2.3	Fluxo de força no veículo .....	33
2.2.4	Relação de transmissão .....	33
2.3	Transmissão automotiva .....	34
2.3.1	Característica das transmissões .....	35
2.3.1.1	Transmissão manual.....	35
2.3.1.1.1	Embreagem.....	36
2.3.1.1.2	Relação de marcha .....	36
2.3.1.1.3	Diferencial .....	37
2.4	Transmissão automática e continuamente variável.....	39
2.4.1	Transmissão automática .....	39

2.4.2	CVT .....	40
2.5	Sistema DUALOGIC® .....	41
2.6	Vantagem e desvantagem.....	43
2.7	Grupo eletro hidráulico .....	44
2.7.1	Grupo de potência.....	45
2.7.1.1	Reservatório de óleo.....	45
2.7.1.2	Óleo hidráulico.....	45
2.7.1.3	Eletrobomba .....	46
2.7.1.4	Acumulador hidráulico .....	47
2.7.1.5	Defletor de óleo.....	47
2.7.2	Grupo das eletroválvulas.....	47
2.7.2.1	Eletroválvulas EV1 e EV2 .....	48
2.7.2.2	Eletroválvulas EV3 e EV4 .....	49
2.7.2.3	Eletroválvula EV0.....	51
2.8	Componentes de interface .....	53
2.8.1	Eixo de comando do câmbio .....	53
2.8.2	Flange intermediária.....	53
2.8.3	Atuador da embreagem.....	54
2.9	Circuito hidráulico e lógica de acionamento .....	54
2.9.1	Circuito hidráulico .....	54
2.9.1	Lógica de acionamento .....	55
2.10	Sensores .....	56
2.10.1	Sensor de alavanca de comando do câmbio ( <i>joystick</i> ).....	57
2.10.2	Borboletas do volante.....	58
2.10.3	Sensores de posição dos atuadores de seleção e engate .....	59
2.10.4	Sensor de posição da embreagem (PLCD).....	60
2.10.5	Sensor de pressão do óleo.....	62



2.10.6	Sensor de rotação da árvore primária .....	63
2.10.7	Interruptor <i>Sport</i> .....	65
2.10.8	Interruptor de freio .....	65
2.10.9	Sensor de velocidade do veículo .....	65
2.10.10	Interruptor da porta do condutor .....	65
2.11	MC33816 .....	65
3	Aquisições de dados .....	68
3.1	Testes dos solenoides .....	71
3.2	Testes do sensor de atuador de engate e seleção .....	75
3.3	Controlando o MC33816 com o kit de desenvolvimento KL25Z .....	76
3.3.1	Rodando o programa .....	80
3.4	Comunicação entre o PIC18F4550 e o MC33816 .....	82
3.5	Fluxogramas .....	86
3.5.1	Controle da bomba hidráulica e leitura do Joystick .....	86
3.5.2	Lógica de engate de marcha e verificação .....	88
3.5.3	Algoritmo de engate .....	90
3.6	Circuitos de leitura dos sensores .....	91
4	Resultados Obtidos .....	93
4.1	Testes de acionamento das eletroválvulas .....	93
4.2	Teste sensor de pressão .....	94
4.3	Comunicação entre as placas .....	96
5	Considerações Finais .....	97
6	Bibliografia .....	98
7	APÊNCICES .....	100
7.1	APÊNDICE A .....	100
7.2	APÊNDICE B .....	106
7.3	APÊNDICE C .....	115

7.4	APÊNDICE D .....	130
7.5	APÊNDICE E.....	131

## 1 INTRODUÇÃO

O sistema de transmissão é um dos mais importantes em um automóvel pois é ele o responsável por transmitir o torque produzido pelo motor para as rodas.

Hoje no mercado existem quatro tipos de sistemas de transmissão diferentes, sendo eles os manuais, automáticos, Transmissão continuamente variável (CVT - *Continuously variable transmission*) e por fim o sistema de transmissão automatizado o qual é o foco neste trabalho.

Cada montadora possui um sistema diferente sobre o câmbio automatizado como a Easytronic (Chevrolet), I-Motion (Volkswagen), Powershift (Ford), Easy-R (Renault), 2-Tronic (Peugeot), I-Shift (Honda), SMT (Toyota) e Dualogic® (Fiat) o qual será abordado neste trabalho.

O câmbio automatizado, conhecido como câmbio semiautomático ou robotizado, por possuir um “robô” que realiza as trocas de marcha, teve seu início pelo sistema Freechoice nas corridas de Formula 1 pela Ferrari em 1987 e só em 1994 teve o início da aplicação em carros de passeio (Quieroz, et al., 2015).

Segundo o engenheiro Carlos Henrique Ferreira que é assessor técnico da Fiat defende o uso da transmissão automatizada nos veículos de passeio.

Este sistema foi pensado para unir o conforto da função automática sem roubar potência do motor ou aumentar o consumo do carro. E por não roubar potência do motor, pode ser aplicado em motores menores. (Webmotors S.A., 2009).

O sistema semiautomático possui a função de troca de marchas manual e automática, assim o motorista pode escolher se o sistema eletrônico irá calcular e realizar as trocas automaticamente adotando a melhor estratégia, ou se ele mesmo realize as trocas caso preferir, lembrando sempre que nesses sistemas os carros não possuem o pedal da embreagem.

A transmissão automatizada Dualogic® possui algumas vantagens se comparada as outras, e como exemplo podemos citar um melhor rendimento de combustível, um maior conforto e segurança para o motorista e como esse sistema não possui um conversor de torque como o sistema automático é mais barato e mais leve em comparação ao automático.

## **1.1 Justificativa**

Mesmo possuindo vantagens, o conjunto automatizado, conforme escreveu Queiroz, Melo e Calabrez (2015), possui reclamações devido à falta de suavidade e os trancos no carro nas trocas de marcha.

Tendo em vista essas reclamações, o trabalho proposto realizará pesquisas e desenvolvimento para obter o melhor controle nos engates e desengates através das eletroválvulas do conjunto eletro-hidráulico.

## **1.2 Objetivos**

Para realizarmos esse trabalho, estabelecemos alguns objetivos a serem conquistados e dividimos eles entre objetivos gerais, que são os objetivos finais a serem alcançados, e os objetivos específicos os quais mostram algumas metas a serem conquistadas para obtermos o nosso objetivo final.

### **1.2.1 Objetivos gerais**

O principal objetivo neste trabalho é desenvolver o conceito de software do controle das eletroválvulas do sistema Dualogic® através do drive de controle MC33816, o qual cuidará exclusivamente das eletroválvulas realizando os engates e desengates.

### **1.2.2 Objetivos específicos**

Os focos nesse projeto são:

- Realizar a leitura dos sensores utilizados para o bom funcionamento do câmbio automatizado, para ser interpretado no microcontrolador PIC;
- Realizar a programação do MC33816;
- Estabelecer uma comunicação entre o PIC e o MC33816;
- Dar continuidade ao projeto tomado como base de Bento, Oliveira e Cesário (2017).

## **1.3 Estrutura e Organização**

O trabalho apresentado está organizado da seguinte forma: o primeiro capítulo retrata da motivação, justificativa e objetivos do projeto proposto. O segundo traz o conteúdo das referências bibliográficas que serão necessárias para o entendimento

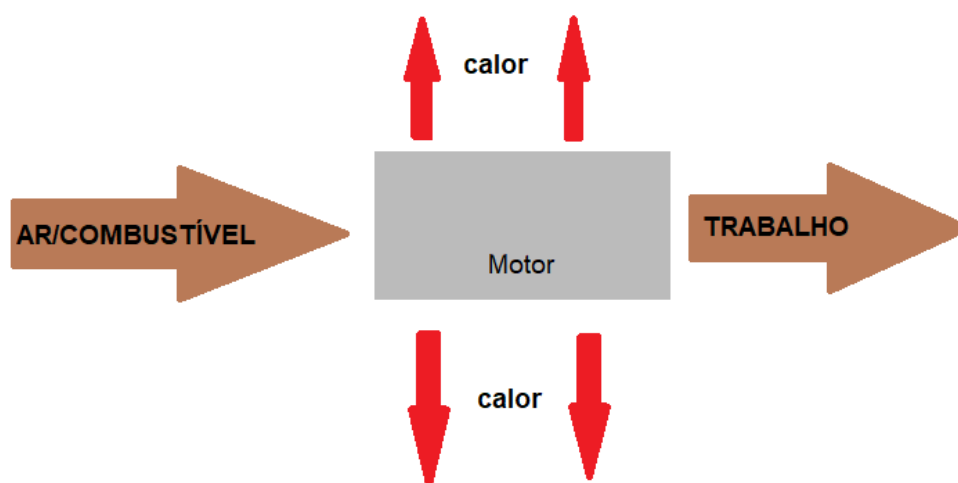
de geração de torque, tipos de transmissão, a transmissão Dualogic® juntamente com seus componentes e sistemas e o MC33816. O terceiro mostra todas as aquisições de dados dos componentes, sendo eles: sensores; eletroválvulas; união do hardware base; SPI, amostrando o fluxograma de controle da transmissão e detalhando o uso do programa de controle do MC33816. O quinto trata da conclusão que se teve com este trabalho. O sexto mostra as referências bibliográficas que foram tomadas como base ao decorrer de todo o projeto seguido. E por fim no capítulo final com os apêndices que em conjunto formam o algoritmo para o controle do Kit Dualogic®.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 Motores de combustão interna

Conforme Brunetti (2012, p.27) motores de combustão interna (MCI) convertem energia térmica (calor) em trabalho. O calor é obtido através da combustão proveniente da mistura ar/combustível (gasolina, etanol, gás natural veicular (GNV), diesel), ou seja, conversão de energia química em energia mecânica. Na figura 1 é ilustrado em um diagrama o que se introduz no motor (ar e combustível), e o que ele entregada (trabalho e calor).

Figura 1 - Fluxo de massa e energia em um motor de combustão interna



Fonte – Adaptado de Brunetti (2012)

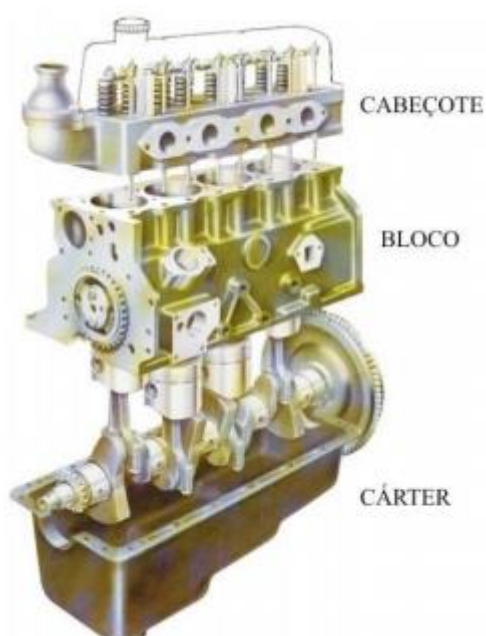
Motores de combustão interna podem ser classificados como: alternativos, rotativos e de impulso, para nosso projeto o foco será o motor alternativo. Esses motores são chamados de motores alternativos, pelo fato que internamente seus pistões realizam o movimento ascendente e descendente dentro do cilindro. Para atender os projetos e necessidades do mercado esses motores podem conter um ou mais, cilindros e pistões.

### 2.1.1 Principais componentes dos motores de combustão interna

Conforme descrito no material de treinamento da disciplina de Motores A Combustão Interna do professor Marco Aurélio Froes (Manual tecnologia automotiva Bosch, p.488), os motores são constituídos por um conjunto de peças fixas e móveis que permitem a transformação de energia química em mecânica. As peças fixas são divididas em três partes, conforme demonstrado na figura 2, sendo:

- Cabeçote, onde estão alojadas as câmaras de combustão, válvulas de admissão, escape e o comando de válvulas;
- Bloco do motor, estão inseridos os cilindros, pistões, bielas e a árvore de manivelas;
- Cárter, que é o reservatório ou recipiente onde está o fluido lubrificante do motor.

Figura 2 - MCI, peças fixas

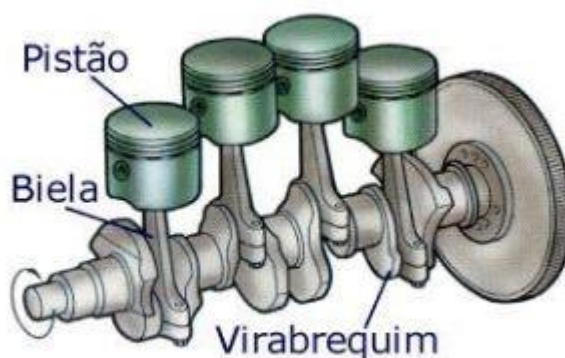


Fonte – Lopes, Bruno Bonfim. **Divisão do Motor, BLOG**. Disponível em: <http://brunobonfimlopes.blogspot.com/2012/02/divisao-do-motor-diesel.html>. Acesso em: 20 Mai. 2020, 23:35:30.

E peças móveis, mostrado na figura 3, sendo:

- Êmbolo, que a partir do movimento ascendente e descendente e a mistura ar/combustível, converte a energia química da mistura em mecânica;
- Bielas, o qual transmite a energia mecânica gerada pelo êmbolo para o eixo de manivelas linearmente;
- Árvore de manivelas, que em conjunto com a biela transforma o movimento linear em rotação.

Figura 3 - MCI, peças móveis



Fonte – Padovan, Luiz Atilio. **Estudos dos Motores de combustão interna**, 2011. Disponível em: <<https://docplayer.com.br/7360536-Estudo-dos-motores-de-combustao-interna.html>>. Acesso em: 15 abr. 2020, 21:35:30.

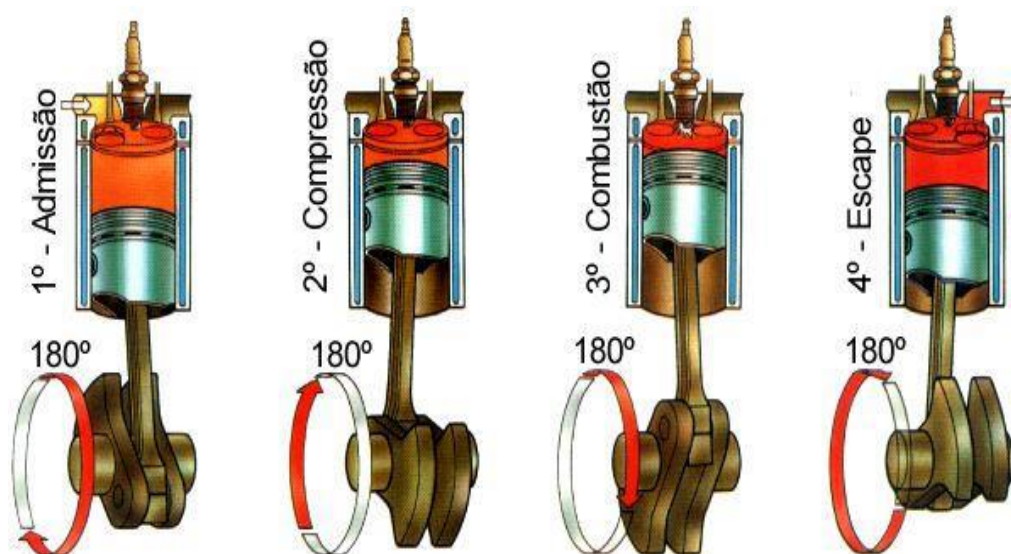
### 2.1.2 Ciclos e tempos de um motor de combustão interna

Tomando como referência Brunetti (2012, p. 36), os MCI dependem de duas voltas completas para que todos os conjuntos de êmbolos, bielas e árvore de manivelas inseridos no motor gerem trabalho. A figura 4 mostra que cada ciclo é provido de 360° ou seja, dois ciclos 720° duas voltas completas da árvore de manivelas, nesse momento são realizados os tempos de admissão, compressão, expansão e escapamento em todos os cilindros conforme demonstra o Quadro 1. O



MCI utilizado no veículo atualmente são chamados de motores quatro tempos, essa característica se dá pela quantidade de vezes que um único êmbolo realiza o movimento de deslocamento do ponto morto inferior (PMI) ao ponto morto superior (PMS) para realizar os quatro tempos, que é dado pelo sincronismo entre a árvore de manivelas e comando de válvulas.

Figura 4 - Ciclos e tempos de um MCI



Fonte - Brunetti (2012, p.36)

Tempo de admissão: a válvula de admissão se encontra aberta e o êmbolo está em movimento descendente, esse processo gera uma depressão fazendo com que os ar seja admitido para dentro do cilindro e em algumas aplicações junto com o ar é adicionado o combustível de forma gasosa para se obter uma mistura ar/combustível, porém nos motores com injeção direta é diferente.

Tempo de compressão: é nesse momento que as válvulas de admissão e escapamento estão fechadas e o êmbolo está em movimento ascendente, causando a compactação de moléculas de ar/combustível e como consequência um aumento da pressão no interior do cilindro.

Tempo de expansão: no MCI após graus do PMS é gerada uma centelha através do sistema de ignição, provocando a combustão, empurrando o êmbolo para PMI e, como descrito acima, esse é o tempo útil do motor onde é gerado trabalho.

Tempo de escape: o pistão é deslocado do PMI ao PMS já com a válvula de escapamento aberta e os gases que foram queimados pela combustão serão expelidos para o sistema de escapamento voltando para a atmosfera.

Quadro 1 - Ordem de ignição

Ordem de ignição					Fase
1	Combustão	Escape	Admissão	Compressão	180°
3	Compressão	Combustão	Escape	Admissão	180°
4	Admissão	Compressão	Combustão	Escape	180°
2	Escape	Admissão	Compressão	Combustão	180°

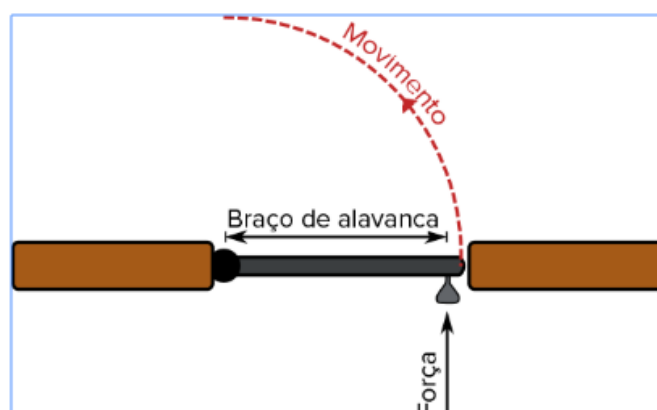
Fonte – Autor

### 2.1.3 Torque

Torque é a força aplicada em um eixo submetido a uma alavanca, ou seja, distância que é aplicada a força nesse determinado eixo. Normalmente a unidade de medida mais usual é a N/m<sup>2</sup> (Newton por metro quadrado).

Na figura 5, podemos visualizar melhor o conceito de torque. Ao abrir a porta, a maçaneta está a uma distância  $x$  do seu eixo de movimento, por isso quanto maior a distância da força aplicado do seu eixo, maior será a resultante de torque.

Figura 5 - Torque aplicado ao abrir uma porta



Fonte – Khan Academy, 2017. Disponível em: <<https://pt.khanacademy.org/science/physics/torque-angular-momentum/torque-tutorial/a/torque>>. Acesso em: 11 Dez. 2019, 15:21:45.

### 2.1.3.1 Torque gerado a partir do sistema biela-manivela

“Como sabemos, após a combustão ocorre aumento da temperatura e pressão dentro do cilindro. A pressão desloca o pistão do PMS para PMI havendo realização de trabalho mecânico” (Varella, 2010).

Como apresentado por Bento, Oliveira e Cesário (2017) a transmissão é responsável por transmitir o torque e potência gerados pelo motor para as rodas do veículo que por sua vez irão gerar o deslocamento do veículo. Isso pode ser realizado a partir da combustão no motor, o êmbolo recebe essa força gerada pela expansão da mistura ar/combustível transmitindo-a para a biela que forma uma alavanca dando origem a um movimento rotacional da árvore de manivelas que está conectada a transmissão.

A intensidade dessa força depende diretamente da capacidade volumétrica do cilindro do motor e o momento que é realizada a expansão dos gases que foram admitidos pelo motor, ou seja, o momento em que ocorreu um aumento de temperatura e pressão no cilindro e logo em seguida a liberação da centelha.

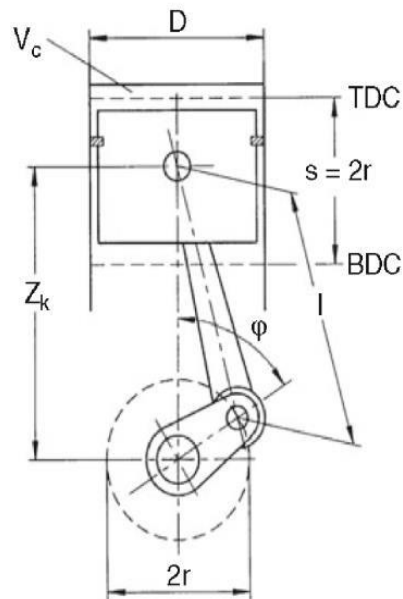
A figura 6 mostra as principais dimensões de um motor, onde:

- $D$ = Diâmetro do cilindro[mm];
- $S$ = Curso do êmbolo[mm];
- $V_c$ = Volume comprimido do cilindro[mm<sup>3</sup>];
- $Z_k$ = Distância entre o centro da árvore de manivelas e o pino do êmbolo[mm];
- $R$ = Raio de giro da árvore de manivelas[mm];
- $L$ = Comprimento total biela[mm];
- $\alpha$ = Ângulo entre o raio de giro e a distância entre o centro da árvore de manivelas[°].

E duas demarcações:

- TDC= *Top Dead Center* / ponto morto superior
- BDC= *Bottom Dead Center* / ponto morto inferior

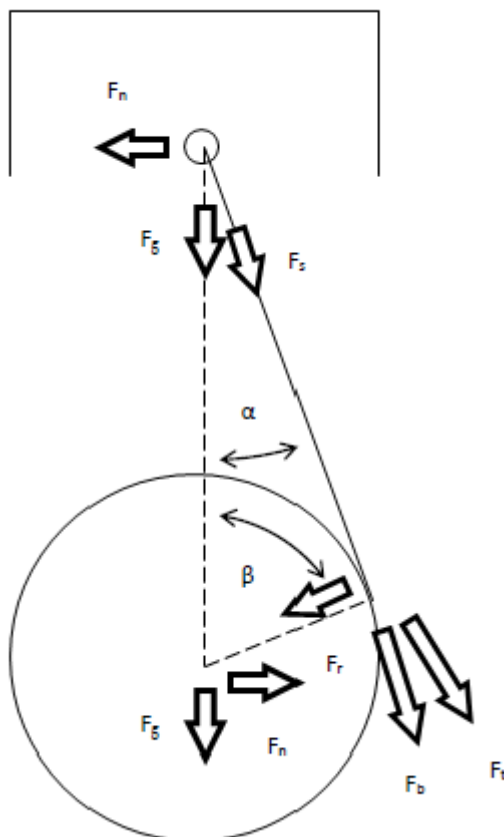
Figura 6 - Sistema biela-manivela e ângulo forma pela biela e o eixo de manivelas



Fonte - Miotto, et al. (2018)

Para Bosch (2014) o conjunto de pistão, biela e árvore de manivelas está exposto à força gerada pela expansão dos gases ( $F_g$ ), sendo que essa força age sobre o pistão (êmbolo) que por sua vez ocasiona uma resultante lateral que é a força aplicada pelo pistão na parede do cilindro e biela, neste momento a biela entrega essa força de forma tangencial ( $F_t$ ) a árvore de manivelas, entregando essa resultante para o conjunto de transmissão conforme ilustrado na figura 7.

Figura 7 - Disposição das forças no momento da expansão dos gases



Fonte - BOSCH (2014)

Nas equações de 1 a 4, podemos observar as fórmulas referentes as forças ilustradas na figura 7.

Equação 1 - Cálculo da Força biela ( $F_b$ )

$$F_b = \frac{F_g}{\cos \beta} \text{ [N]};$$

Equação 2 - Cálculo da Força lateral do pistão ( $F_n$ )

$$F_n = F_g \tan \beta \text{ [N]};$$

Equação 3 - Cálculo da Força radial ( $F_r$ )

$$F_r = F_g \left( \frac{\cos \alpha + \beta}{\cos \beta} \right) \text{ [N]};$$

Equação 4 - Cálculo da Força tangencial ( $F_t$ )

$$F_t = F_g \frac{\sin(\alpha+\beta)}{\cos \beta} [\text{N}];$$

### 2.1.3.2 Força de inercia

Como abordado por Bosch (2014, p. 457), essa força é gerada a partir das massas rotacionais da árvore de manivelas, como já abordado nesse trabalho em motores com múltiplos cilindros há um espaço de tempo em que se é aplicado força no conjunto pistão, biela e árvore de manivelas, ou seja, em um conjunto por vez. Essa propriedade pode ser evidenciada pela massa do pistão (apenas massa oscilante) e massa da árvore de manivelas (massa rotativa), sendo seus cálculos representados nas equações 5 e 6 respectivamente onde,  $m_k$  é a massa do pistão e  $m_w$  a massa da árvore de manivelas.

Equação 5 - Cálculo da Massa oscilante

$$m_o = \frac{m_{pl}}{3} + m_k;$$

Equação 6 - Cálculo da Massa rotativa

$$m_r = \frac{2m_{pi}}{3} + m_w;$$

## 2.2 Transmissão de torque

A partir do conteúdo mostrado no decorrer desse estudo, é possível visualizar de onde vem o torque que chega até as rodas do veículo permitindo o seu deslocamento. Agora será explicado como o torque gerado pelo motor é aproveitado pela transmissão.

“A transmissão de um automóvel tem a função de fornecer as forças de tração e impulsão necessárias para induzir o movimento” (Bosch, 2005 p. 736)

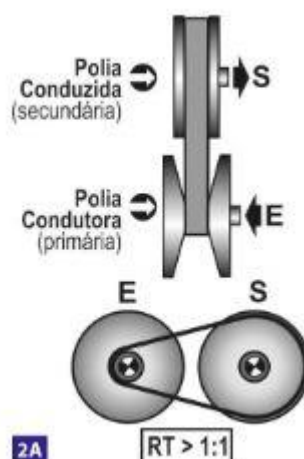
### 2.2.1 Transmissão por polias e correia

Como estudado por Bento, Oliveira e Cesário (2017, p. 35) um sistema de transmissão por polias é formado por duas polias (motora e movida) e uma correia, como demonstra a figura 8, ou seja, uma das polias irá receber o torque que é chamada de polia motora e transferir esse torque em forma de movimento através da correia para a polia conhecida como movida.

- Polia movida: são polias que recebem o movimento e a força;
- Polia motora: responsáveis por transmitir o movimento e a força para a polia movida por intermédio de uma correia.

Transmissões por polias são utilizadas para colocar em funcionamento os periféricos do motor de um veículo como: alternador, compressor do ar-condicionado, e bomba d'água. Este conceito também é usado no funcionamento de uma transmissão CVT que será abordada no decorrer deste estudo.

Figura 8 - Polias e correia



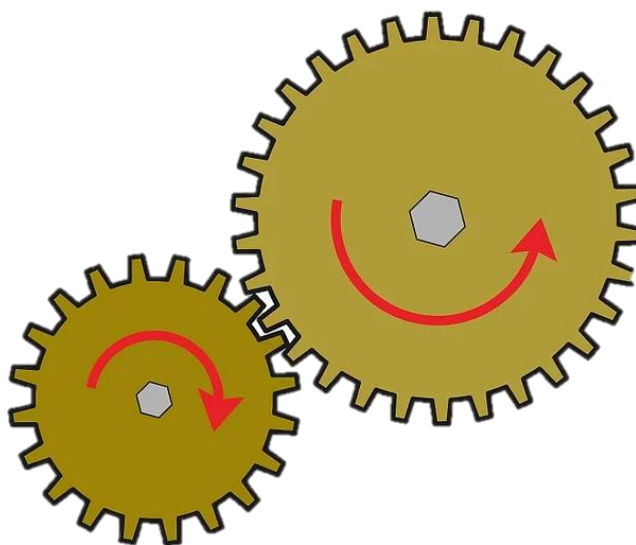
Fonte – Manavella, Humberto. **Diâmetro das Polias do sistema CVT, 2016**. Disponível em: <<https://www.oficinabrasil.com.br/noticia/tecnicas/principio-de-funcionamento-da-transmissao-continuamente-variavel-cvt>>. Acesso em: 10 jan. 2020, 10:27:06.

### 2.2.2 Transmissão por engrenagens

São utilizadas em diversos sistemas automotivos, como na transmissão manual, automática e automatizada, capazes de transmitir torque, aumentando ou diminuindo esse torque através de uma relação de dentes em cada uma das engrenagens. A relação é determinada pela distância do centro até o ponto de contato do “dente” da engrenagem.

Como mostrado no tópico transmissões por polias, utiliza-se o mesmo conceito de polias motora e movida para engrenagens motora e movida. A figura 9 traz a imagem das engrenagens em conjunto de diferentes tamanhos, gerando uma relação entre elas.

Figura 9 - Engrenagens



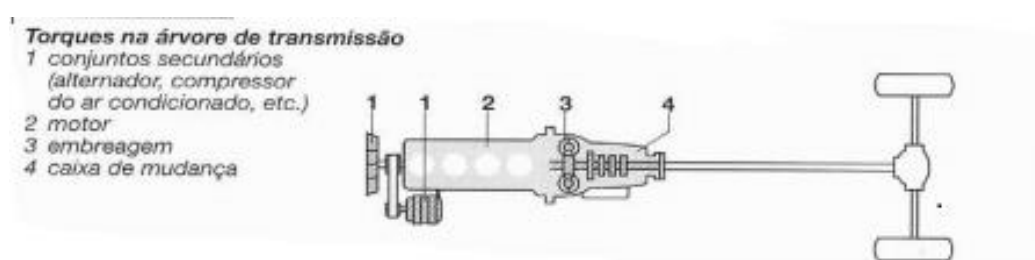
Fonte – wikiHow, **calculando a relação de duas engrenagens**. Disponível em: <<https://pt.wikihow.com/Determinar-a-Rela%C3%A7%C3%A3o-de-Transmiss%C3%A3o-de-Engrenagens>>. Acesso em: 10 out. 2019.



### 2.2.3 Fluxo de força no veículo

O torque gerado pelo motor é direcionado à transmissão por intermédio da fixação do volante do motor ou conversor de torque, que estão conectados ao eixo primário da transmissão e neles estão as engrenagens motoras que irão transferir para as engrenagens movidas, depois disso indo para o conjunto do diferencial e as rodas conforme a figura 10.

Figura 10 - Trem de força



Fonte - Manual de Tecnologia Automotiva Bosch, 2005, p. 592.

### 2.2.4 Relação de transmissão

A relação de transmissão é dada pela quantidade de dentes em cada engrenagem motora e movida, isso quer dizer que se na engrenagem motora tenho uma quantidade de dentes inferior do que a engrenagem movida precisarei de mais voltas na motora para realizar somente uma rotação na movida. A equação 7 que representa esse conceito e é composta por T1 (quantidade de dentes da engrenagem motora), T2 (quantidade de dentes da engrenagem movida) e Re (relação de transmissão).

Equação 7 - Relação de Transmissão

$$Re = \frac{T2}{T1}$$

Essa mesma fórmula pode ser usada para um conjunto com mais de duas engrenagens.

Usando a relação de engrenagens podemos encontrar a velocidade na engrenagem movida, levando em consideração que sabemos qual a velocidade da engrenagem motora que é a mesma rotação da árvore de manivelas pois o motor está

conectado diretamente a ela, fica fácil obter essa velocidade, através da equação 8, onde S1 e T1 são a rotação e o número de dentes da primeira engrenagem (motora) e S2 e T2 a rotação e número de dentes da segunda(movida).

Equação 8 - Expressão para cálculo da rotação da engrenagem movida

$$S1 \times T1 = S2 \times T2$$

Realizando uma simples manipulação das variáveis teremos na equação 9 a rotação na engrenagem movida.

Equação 9 - Velocidade na engrenagem movida

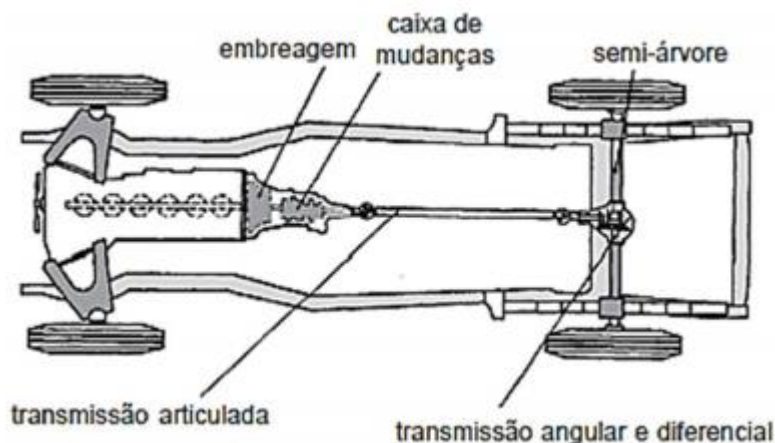
$$S2 = S1 \times \frac{T1}{T2}$$

### 2.3 Transmissão automotiva

O torque produzido no motor é transmitido para as rodas através da transmissão. O conjunto de transmissão ligado ao motor forma o conjunto denominado “*Powertrain*” ilustrado pela figura 11 (pp. Senai, 2003, SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL. Automobilística: sistemas).

No mercado atual existem alguns métodos de transmissões diferentes, como transmissão manual, semiautomática (automatizada), automática e transmissão CVT.

Figura 11 - Conjunto Powertrain



Fonte - Apostila do Senai - 2003, p.7

### **2.3.1 Característica das transmissões**

Uma transmissão manual utiliza a embreagem como meio de ligação entre transmissão e motor, e esta é dividida em vários componentes mecânicos que trabalham em sincronismo o qual convertem o torque gerado pelo motor para as rodas.

Já em transmissões automáticas utilizam conversor de torque que está mergulhado em óleo, ele consegue exercer a função da embreagem. A relação de marchas é promovida através dos conjuntos de planetárias.

O sistema CVT, por sua vez, utiliza de um sistema de polias variáveis que impõem o regime de marchas de acordo com a necessidade do condutor.

Para os sistemas automatizados, temos a mecânica de uma transmissão comum, sendo auxiliada por um conjunto mecatrônico controlado por uma Unidade de controle automotivo (ECU - Engine Control Unit). (Oliveira, 2017).

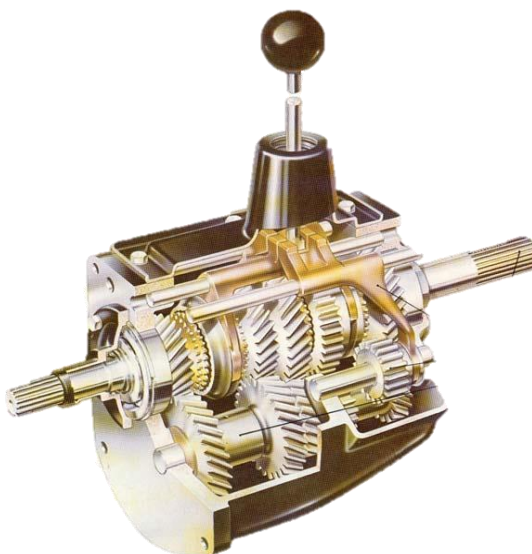
#### **2.3.1.1 Transmissão manual**

É um sistema que contém conjuntos de engrenagens que promovem a ampliação de torque ou velocidade recebidos pelo giro do motor.

Pela ilustração da figura 12 essa troca de relações é possível através de uma série de componentes mecânicos que trabalham em sincronismo com o motor do veículo, e a escolha destas relações é realizada pelo condutor. Sendo o componente que realiza o ligamento e desligamento entre caixa de marcha e o motor é denominado de embreagem

Quando o condutor movimenta a alavanca de câmbio para engatar alguma marcha, internamente a transmissão movimenta uma série de componentes que acoplam mecanicamente as engrenagem, e então o eixo de saída irá transmitir o torque e rotação respectivo a relação de marcha solicitado (Mattera, 2018).

Figura 12 - Conjunto de engate e seleção da transmissão manual



Fonte - Adaptado. Mauricio. **Caixa de câmbio, 2007**. Disponível em: <http://centroautomec.blogspot.com/2008/09/cmbio-caixa-de-marchas.html>. Acesso em: 16 fev. 2020, 16:25:44.

#### 2.3.1.1.1 Embreagem

Embreagem é um componente que fica disposto entre a transmissão e o motor, esta tem a função de fazer a ligação e separação progressiva entre ambos com a finalidade de proporcionar o engate na relação de marcha correspondente ao regime de rotação e torque do motor (Choleet, 2002).

Quando você pisa no pedal da embreagem, esses discos são afastados, rompendo essa conexão, permitindo, assim, a troca de marchas sem dificuldade. Ao soltar o pedal, os discos voltam a ser pressionados um contra o outro e as rodas passam a receber o movimento do motor. Alguns modelos usam assistência hidráulica, o que deixa o pedal extremamente macio e as trocas ainda mais suaves. (Nakata, 2019).

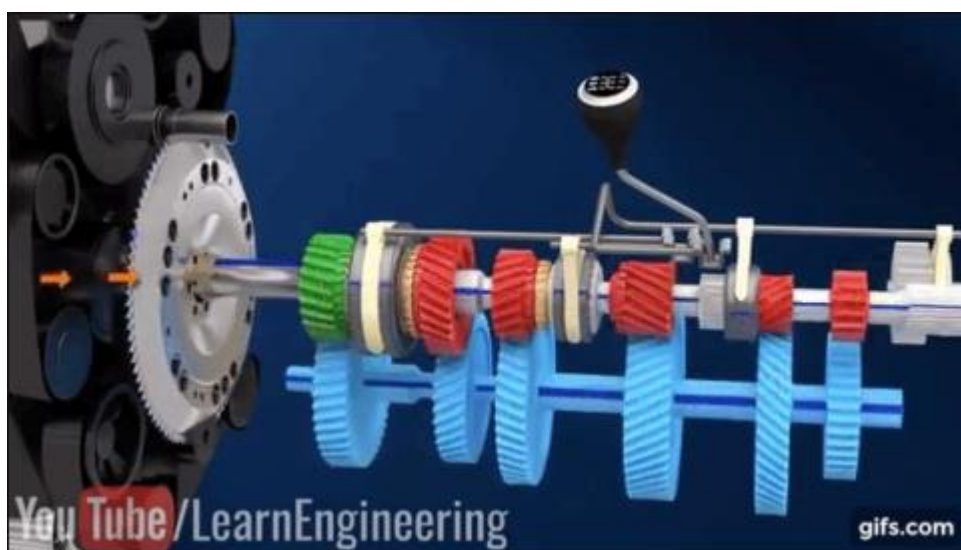
#### 2.3.1.1.2 Relação de marcha

A relação de marchas é a relação de transmissão que promove a redução ou amplificação da rotação de entrada da transmissão, aplicada ao câmbio do veículo temos conjuntos de engrenagens motoras que recebem o torque diretamente do motor através do acoplamento da embreagem no volante e do eixo primário.

A entrada da rotação que vem do motor, em relação a saída, que vai para o diferencial, pode ser alta ou baixa conforme ilustrado pela figura 13, respectivamente quando esta relação é maior que uma volta à amplificação de torque é gerado. Caso essa relação de marcha seja menor que um, a transmissão de velocidade será entregue às rodas. (Murta, 2012)

Em veículos comuns, a ré, primeira e segunda marchas têm uma relação de transmissão de torque, isso porque tem como função tirar o veículo do momento de inércia.

Figura 13 - Demonstração do funcionamento do câmbio manual

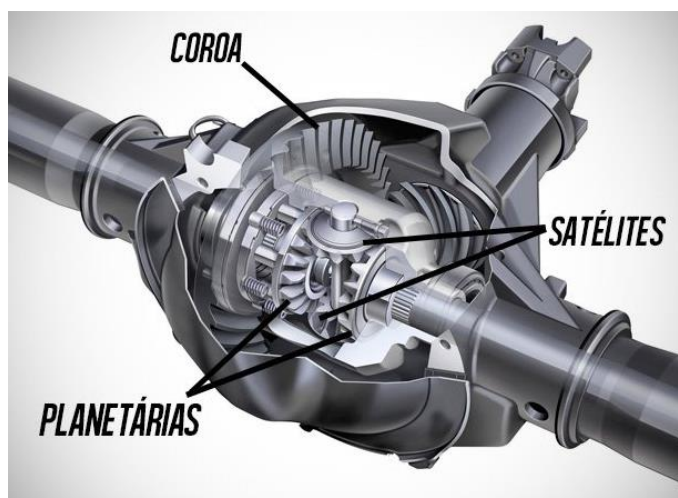


Fonte – Engenharia, Aprenda. **Embreagem, como funciona? 2020**. Disponível em: <[https://www.youtube.com/watch?v=\\_3p0IJM296Y](https://www.youtube.com/watch?v=_3p0IJM296Y)>. Acesso em: 21 Mai. 2020, 00:07:20.

### 2.3.1.1.3 Diferencial

Em conjunto de todas as transmissões temos o diferencial em sua montagem, este é projetado em conjunto com as relações de marchas da transmissão e este tem a função de transmitir o torque para as rodas, nele é instalado um conjunto planetário, este demonstrado na figura 14, que promove o giro das rodas de forma independente à outra, essa condição é necessária para a realização de curvas do veículo onde a roda do lado de fora à curva gire mais que a roda do lado de dentro da curva sem que ocorra o arraste da roda (Jaicar, 2017).

Figura 14 - Diferencial em corte e seus componentes internos



Fonte – Contesini, Leonardo. **Como funciona e para que serve diferencial em carros, 2017.**  
Disponível em: <<https://www.flatout.com.br/como-funciona-e-para-que-serve-o-diferencial-dos-carros/>>. Acesso em: 09 Fev. 2020, 15:42:03.

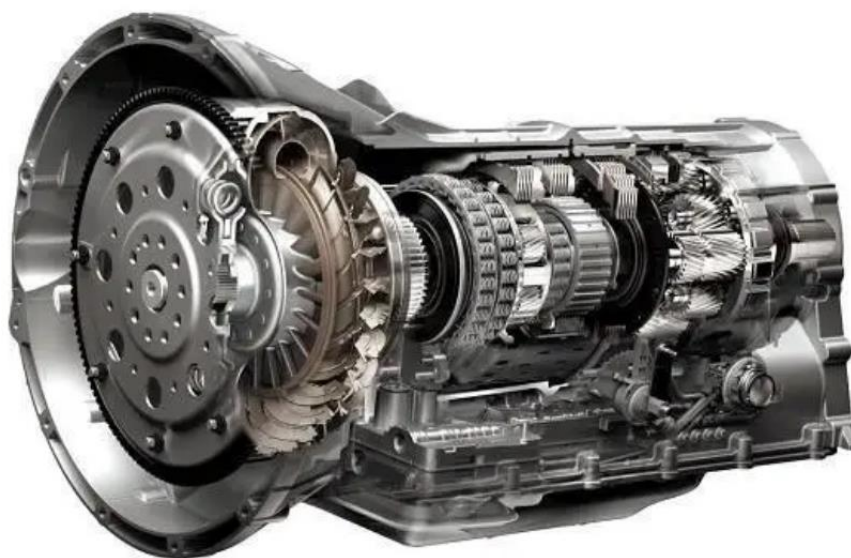
## 2.4 Transmissão automática e continuamente variável

Nesta seção iremos tratar das transmissões do tipo automática e CVT respectivamente, mostrando suas características, componentes, operação e funcionamento.

### 2.4.1 Transmissão automática

Transmissões automáticas são constituídas de diversos sistemas dentre eles mecânicos, hidráulicos e elétricos. A figura 15 ilustra a montagem e construção interna de uma transmissão automática cujos sistemas de controles hidráulicos são gerenciados eletronicamente pela unidade de controle da transmissão do veículo. A transmissão automática contém um sistema que remove a ação física de operação do pedal da embreagem, pois o conversor de torque, que é acoplado por fluido, provê essa função por atuar como um dispositivo de acionamento.

Figura 15 - Transmissão automática em corte



Fonte – Carro de Garagem. **Como funciona o câmbio automático?, 2020.** Disponível em: <https://www.carrodegaragem.com/como-funciona-cambio-automatico/>. Acesso em: 25 Jan. 2020, 09:54:21.

Os fatores como velocidade, rotação, posição do pedal do acelerador e carga influenciam no funcionamento desse modelo de transmissão, a unidade de controle eletrônica interpreta os sinais obtidos e calcula a melhor estratégia de operação da

transmissão. O sistema conta também um rigoroso protocolo de falhas, que mantém a segurança do sistema. (Francisco, 2017)

Transmissões automatizadas e automáticas entram para o grupo de sistemas de segurança, pois proporcionam um conforto ao condutor, evitando fadiga ou irritações que poderiam causar distrações e acarretar um acidente. Esses tipos de transmissões são capazes de fornecer tanto desempenho quanto economia a todas as faixas de operação do motor, sendo o diferencial para o consumidor final. (Bosch, 2005).

#### 2.4.2 CVT

A transmissão do tipo CVT tem em sua construção duas polias em um sistema continuamente variável e essas polias são interligadas por uma correia ilustrado da figura 16. A relação da área de contato das polias permitem que sejam geradas a relação de marcha. O sistema de transmissão CVT identifica a faixa ideal de rotação do motor e com suas polias variáveis altera a rotação de entrada para a de saída, assim provendo as relações de marchas contínuas visando sempre a melhor eficiência energética do motor (iCarros, 2016).

Figura 16 - Polias da transmissão CVT



Fonte – Mendonça, Douglas. **Câmbio CVT, automatizado ou automático: qual é o melhor, 2016.** Disponível em <<https://motorshow.com.br/cambio-cvt-automatizado-ou-automatgico-qual-e-o-melhor/>>.

Acesso em: 12 Set. 2019, 18:33:01.

Esta tem a função de troca de marcha de forma automática. Esse sistema auxilia nas trocas de marchas e gera economia de combustível, e é frequentemente



utilizado em veículos pesados pois possui vantagens deste seguimento, como alta eficiência na transmissão de força, baixo custo, facilidade nas trocas de marcha, instalação simples e segurança contra falhas na operação (Bosch, 2005).

A figura 17 traz a vista em corte de uma transmissão automatizada sendo possível ver seus componentes internos que são semelhantes a uma transmissão mecânica convencional.

Figura 17 - Transmissão automatizada em corte



Fonte – Almeida, Denis Freire de. **Câmbio Easytronic, 2008**. Disponível em:

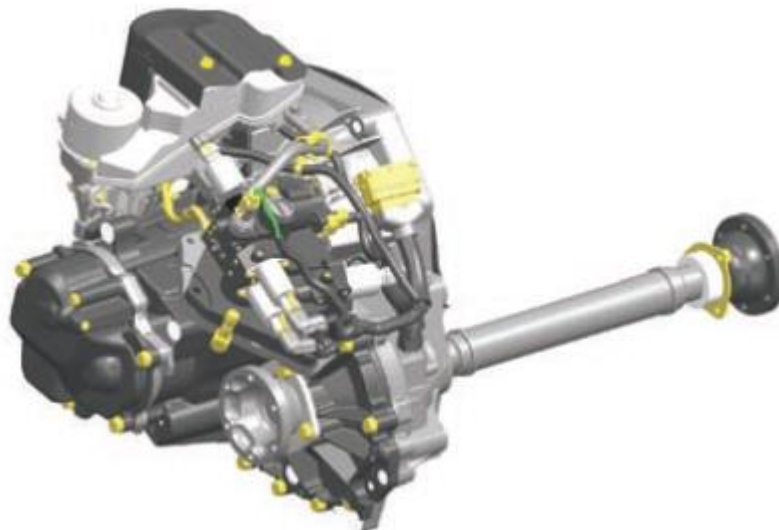
<<http://g1.globo.com/Noticias/Carros/0,,MUL581788-9658,00-BRASILEIROS+DESCOBREM+OS+CARROS+COM+CAMBIO+AUTOMATICO.html>>. Acesso em: 16 Fev. 2020, 14:26:32.

## 2.5 Sistema DUALOGIC®

A transmissão utilizada para o desenvolvimento do projeto é a Dualogic®, observado na figura 18, do FIAT Strada que é semelhante a transmissão *I-Motion*® utilizada pela Volkswagen. Essa irá realizar o acionamento da embreagem e a

mudança das marchas de forma solicitada pelo condutor ao acionar o *Joystick* ou de forma automática comanda pela ECU (FIAT, 2007).

Figura 18 - Transmissão Dualogic® - FIAT



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 06)

## 2.6 Vantagem e desvantagem

Na tabela 1 temos um comparativo entre os tipos de transmissões, na qual há prós (“+”) e contras (“-“) para utilização em cada requisito<sup>1</sup>.

Como resultado vimos que entre as transmissões comparadas, a automatizada esta é a que tem o melhor posicionamento nos quesitos preenchidos, pois possui quase as mesmas vantagens das demais e ao analisar suas desvantagens também tem uma posição melhor se comparado (FIAT, 2007)

Tabela 1- Vantagem e desvantagem das transmissões

<b>Características</b>	<b>Transmissão Mecânica</b>	<b>Transmissão Automática ou CVT</b>	<b>Transmissão Automatizada– Dualogic</b>
Peso	0	--	-
Eficiência Energética	0	-	0
Interrupção de torque na troca de marcha	0	++	+
Flexibilidade na estratégia de controle	0	++	++
Economia de combustível	0	--	+
Qualidade de troca de marcha	0	++	0
Sensação de esportividade	0	-	+
Custo do produto	0	--	-
Custo de manutenção	0	--	-

Fonte – Adaptado, Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 07)

A transmissão automatizada possui quase todos os benefícios da transmissões automáticas, porém sua construção tem como base a caixa de marchas manual, isso impacta diretamente no custo do projeto e no valor do produto final que é atribuído,

<sup>1</sup> A transmissão mecânica tem como a referência sendo “0”, a atribuição “++” significa maior vantagem e a atribuição “--” significa menor desvantagem em comparação com a referência.

sendo esse resultado uma transmissão com um funcionamento automatizado em um custo reduzido se comparado aos outros modelos de transmissão (FIAT, 2007).

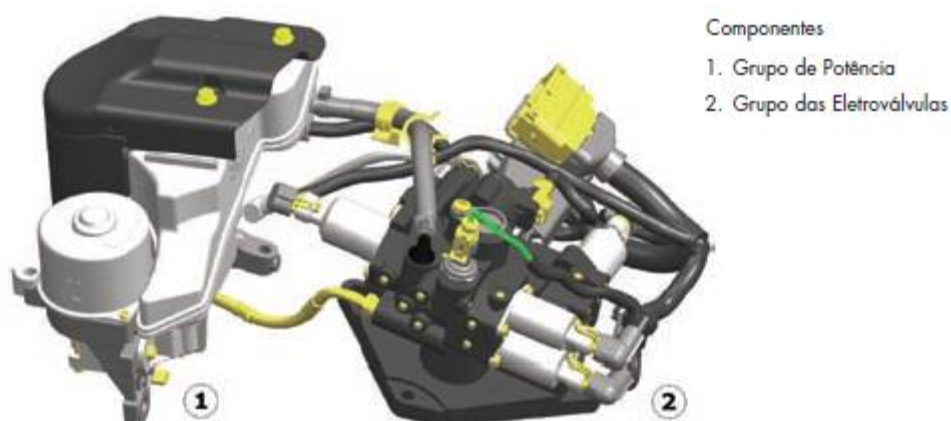
Esse sistema contém as seguintes funcionalidades:

- Função de condução em modo MANUAL ou AUTO;
- Sistema de segurança que previne erro ao acionamento incorreto da transmissão;
- Redução do consumo de combustível de 5% em modo AUTO (tráfego urbano);
- Redução da emissão de poluentes;
- Satisfação durante condução do veículo.

## 2.7 Grupo eletro hidráulico

Pode ser dividido em dois grupos, conforme demonstra a figura 19, sendo respectivamente grupo de potência, na qual opera o circuito de pressão permitindo o funcionamento do sistema, e grupo das eletroválvulas o qual é responsável por transformar a energia hidráulica em movimento através do grupo de pistões de seleção e engate das marchas. (FIAT, 2007)

Figura 19 - Conjunto eletro-hidráulico



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 08)

## 2.7.1 Grupo de potência

Este grupo tem a função de fornecer e manter a pressão hidráulica para o sistema, para que possa ser acionado a embreagem e realizar a mudança de marchas.

### 2.7.1.1 Reservatório de óleo

Possui a função de armazenar o óleo do sistema a baixa pressão, a figura 20 ilustra sua construção que é dividida nas seguintes partes:

- 1 – Tampa;
- 2 - Conexão de retorno de óleo;
- 3 - Conexão com a eletrobomba;
- 4 - Ressaltos para fixação do defletor de óleo;
- 5 – Reservatório.

Figura 20 - Reservatório do óleo do conjunto hidráulico



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 09)

### 2.7.1.2 Óleo hidráulico

Por trabalhar em sistema fechado, o óleo hidráulico não necessita de reabastecimento ou troca no seu período de vida, apenas em condições especiais em que é necessário realizar a substituição de componentes e, também é necessário repor o óleo nas faixas especificadas conforme visto na figura 21.

Característica do óleo:

- Nome comercial: *Tutela CS SPEED*
- Cor: amarela

Figura 21 - Indicadores de mínimo e máximo do nível de óleo



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 10)

### 2.7.1.3 Eletrobomba

Conforme a figura 22 a eletrobomba é composta por engrenagens movidas por um motor elétrico de corrente contínua, este é ativado quando a pressão do sistema está abaixo de 35 Bar e é desativado quando supera a pressão de 50 Bar (a 20°C) com vazão de 0,25cm<sup>3</sup>/rotação. Sua construção é dividida nas seguintes partes:

- 1 - Motor elétrico
- 2 - Bomba de engrenagens
- 3 - Conector elétrico do motor

Figura 22 - Eletrobomba de óleo



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 10)

#### 2.7.1.4 Acumulador hidráulico

Tem a função de manter uma reserva de energia hidráulica para o funcionamento dos atuadores quando a eletrobomba está desligada, demonstrado na figura 23 este item permite a troca de marcha sem que seja necessário ligar a eletrobomba.

Figura 23 - Acumulador de óleo sob pressão



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 11)

#### 2.7.1.5 Defletor de óleo

Ilustrado na figura 24, o defletor tem a função de prevenção contra incêndio caso o acumulador venha a se romper e esguichar óleo pressurizado para as partes quentes do motor, assim elevando a segurança contra incêndio.

Figura 24 - Defletor de óleo



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 11)

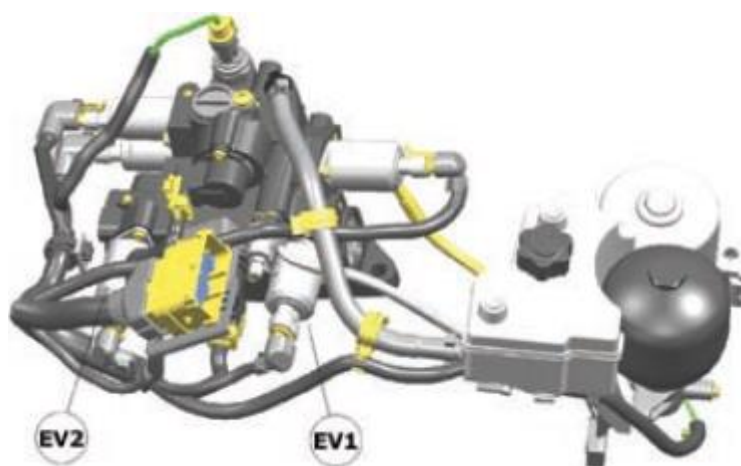
### 2.7.2 Grupo das eletroválvulas

Ao todo são cinco eletroválvulas que equipam o sistema, e essas estão divididas de modo a controlar a seleção e engate/desengate das machas, assim como o acionamento da embreagem.

### 2.7.2.1 Eletroválvulas EV1 e EV2

Responsáveis por controlar as pressões do óleo para as câmaras do atuador de engate/desengate das marchas. A figura 25 mostra o conjunto eletro-hidráulico evidenciando EV1 e EV2.

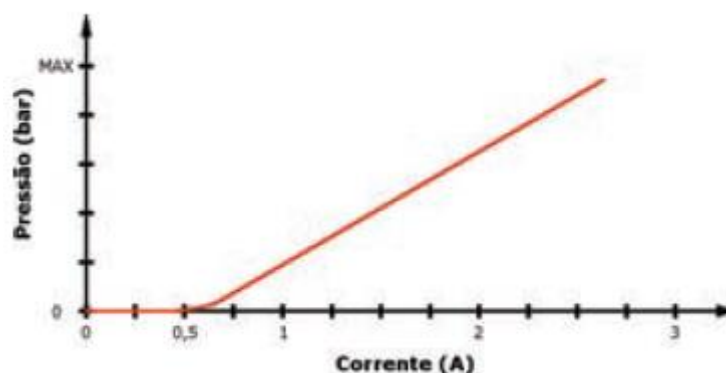
Figura 25 - Conjunto eletro-hidráulico



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 12)

Esse tipo de válvula é linear, ou seja, a pressão da saída da válvula depende do valor da corrente elétrica do acionamento conforme mostrado no gráfico 1. Deste modo é possível controlar a pressão hidráulica que chega até o atuador, assim controlando a força que é exercida para o engate.

Gráfico 1 - Curva da pressão x corrente



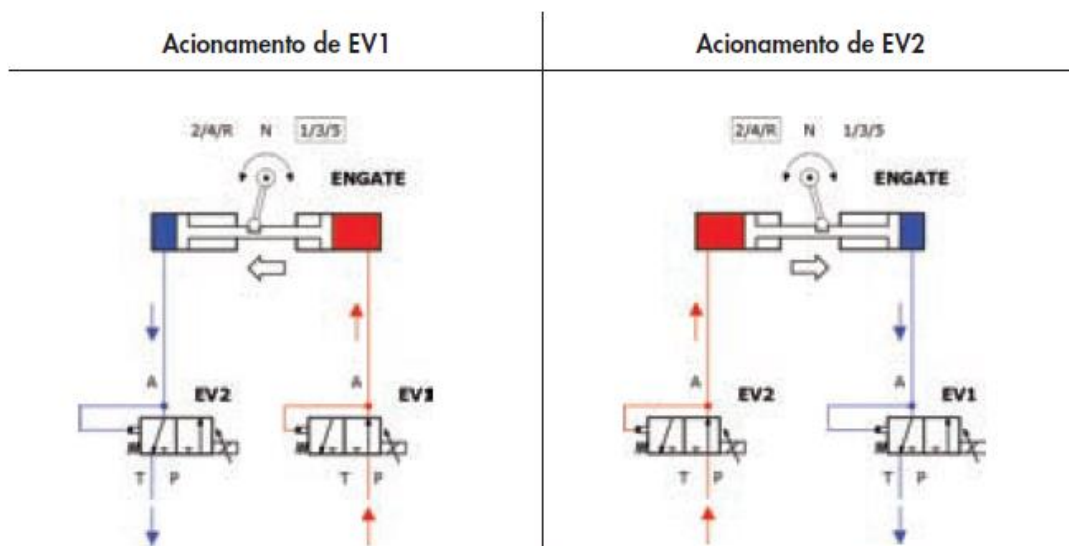
Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 12)

O resultado do acionamento das válvulas é representado pela figura 26. Ao acionar a válvula EV1 o óleo pressurizado direciona à câmara do atuado de engate



para a direita e isso resulta no engate das marchas ímpares, ao acionar a EV2 o óleo pressurizado direciona o atuador de engate para a esquerda resultado no engate das marchas pares e ré (FIAT, 2007).

Figura 26 - Circuito de pressão hidráulica das válvulas EV1 e EV2



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 13)

Essas válvulas trabalham com uma corrente de comando que varia entre 0 A e 2,5 A que é controlada pelo Nó do Câmbio Dualogic® (NCD), a vazão de trabalho pode chegar a uma máxima de 7 litros por minutos e chega até a 10 Bar de pressão de trabalho e a resistência da bobina é 2,5  $\Omega$  variando cerca de 6% a 20°C.

### 2.7.2.2 Eletroválvulas EV3 e EV4

Essas estão ilustradas na figura 27 e têm a função de controlar o fluxo de óleo para os atuadores de seleção. A construção dessa configuração de válvula é do tipo *ON/OFF*.

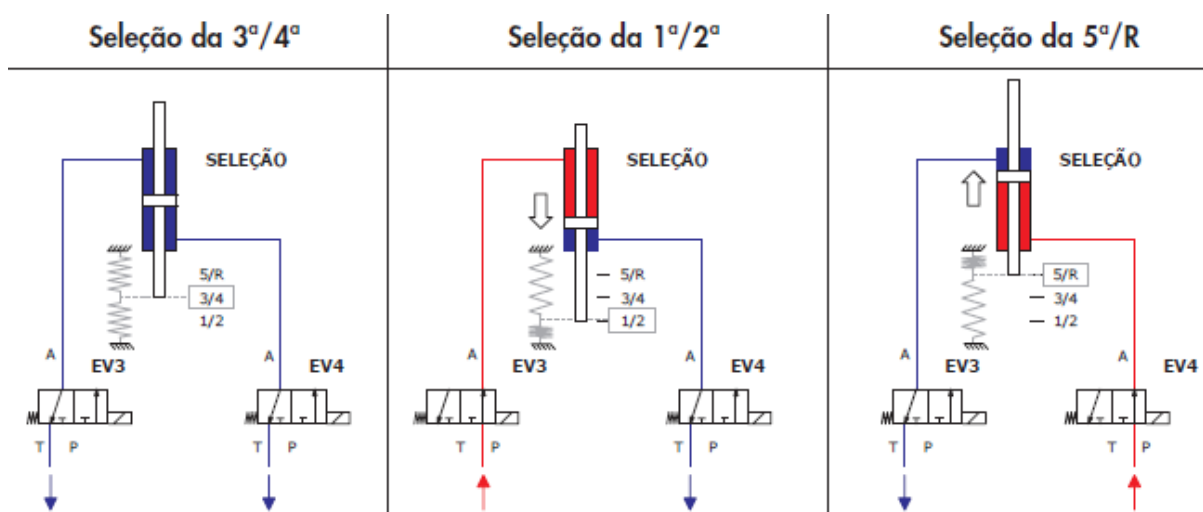
Figura 27 - Posição de montagem das Válvulas EV3 e EV4



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 13)

Conforme a figura 28 podemos verificar o funcionamento da alavanca de seleção pelo trabalho das válvulas EV3 e EV4 sendo que, se ambas estiverem desligadas a seleção fica na posição 3<sup>o</sup>/4<sup>o</sup> por auxílio de molas de retorno do eixo de comando da transmissão, se houver o acionamento da válvula EV3 o fluxo de óleo movimenta o êmbolo para baixo fazendo com que ocorra a seleção da 1<sup>o</sup>/2<sup>o</sup> marcha, e se houver o acionamento da válvula EV4 o fluxo de óleo movimenta o êmbolo para cima fazendo com que seja possível engatar a 5<sup>o</sup>/Ré (FIAT, 2007).

Figura 28 - Circuito de pressão hidráulica das válvulas EV3 e EV4

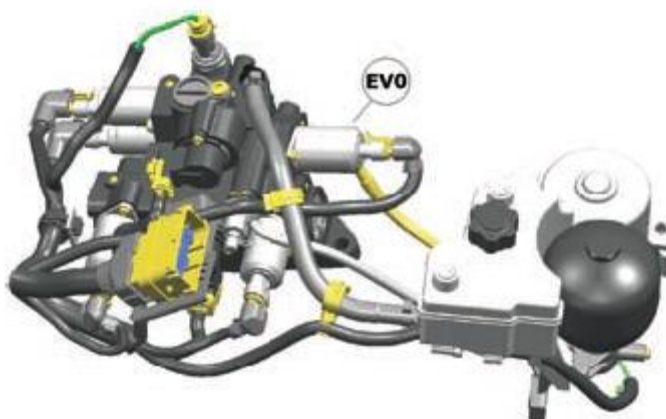


Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 14)

### 2.7.2.3 Eletroválvula EV0

Essa, cujo está ilustrada pela figura 29, tem a função de controlar o fluxo de óleo para o atuador da embreagem.

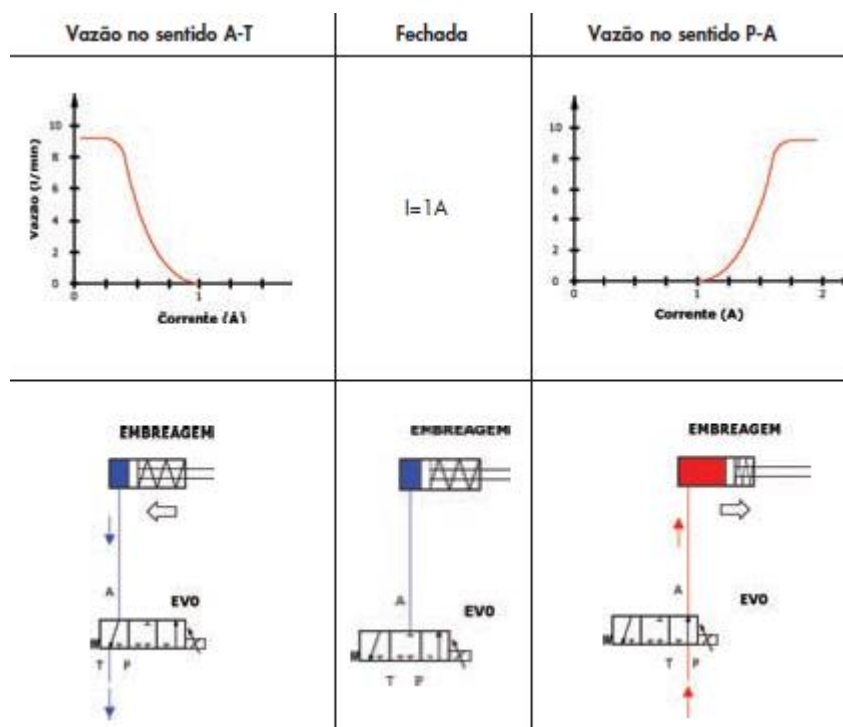
Figura 29 - Posição de montagem da válvula EV0



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 14)

Esta funciona pela variação da corrente que é aplicada podendo chegar a 2A, quando fechada ela direciona a pressão da linha para o atuador, e do atuador para o tanque. Como dito essa transição ocorre pela variação da corrente onde seu *offset* é 1A, quando menor a corrente, a válvula direciona a vazão no sentido do atuador para o tanque e quando é maior esta direciona a linha de pressão para o atuador conforme ilustrado na figura 30, esse sistema permite controlar com precisão a posição da embreagem.

Figura 30 - Gráfico da vazão/corrente conforme posição da válvula EVO



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 15)

## 2.8 Componentes de interface

### 2.8.1 Eixo de comando do câmbio

Este tem a função de transmitir a rotação dos cilindros hidráulicos para o trambulador, e é possível verificar sua construção pela figura 31.

Figura 31 - Eixo de comando do câmbio



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 16)

### 2.8.2 Flange intermediária

A flange ilustrada pela figura 32 tem a função de posicionar o grupo de eletroválvulas sobre a transmissão, posicionar o eixo de comando em relação ao grupo de eletroválvula e também tem a função de vedar a entrada de poeira, água e o vazamento de óleo, esta contém uma válvula em sua construção que permite o alívio da pressão dos gases dentro da transmissão para o meio externo (FIAT, 2007).

Figura 32 - Flange das eletroválvulas



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 16)

### 2.8.3 Atuador da embreagem

O atuador de embreagem, ilustrado na figura 33, é operado pela pressão de óleo que é fornecida, para identificação da posição que está o acionamento em sua carcaça está montado um sensor de posição.

Figura 33 - Atuador da embreagem



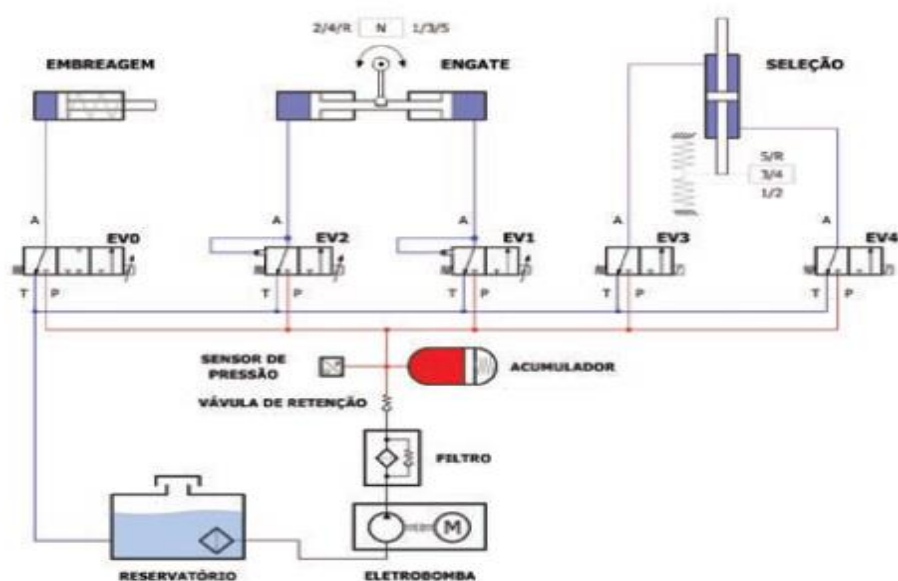
Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 17)

## 2.9 Circuito hidráulico e lógica de acionamento

### 2.9.1 Circuito hidráulico

A figura 34 ilustra o circuito eletro hidráulico do acionamento das válvulas e da embreagem, sendo a união dos circuitos hidráulicos pertinente de cada sistema, permitindo a análise global do sistema e possibilitando entendimento de como seria o funcionamento a partir da(s) eletroválvula(s) que está sendo acionada em conjunto dos demais componentes do sistema.

Figura 34 - Circuito hidráulico completo das eletroválvulas



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 18)

### 2.9.1 Lógica de acionamento

A tabela 2 mostra qual é a lógica de acionamento das válvulas da embreagem, seleção e engate, este pode ser analisado em conjunto da figura 34 pois mostra o resultado do acionamento das eletroválvulas na sequência descrita pela tabela 2, com esta será possível realizar as trocas de marcha na transmissão sem que ocorra o travamento das engrenagens e de forma lógica.

Tabela 2 - Tabela da ordem lógica de acionamento das válvulas

Marcha requerida	Eletroválvulas				
	EV0	EV1	EV2	EV3	EV4
1 <sup>a</sup>	A	C	---	B	---
2 <sup>a</sup>	A	---	C	B	---
3 <sup>a</sup>	A	B	---	---	---
4 <sup>a</sup>	A	---	B	---	---
5 <sup>a</sup>	A	C	---	---	B
R	A	---	C	---	B

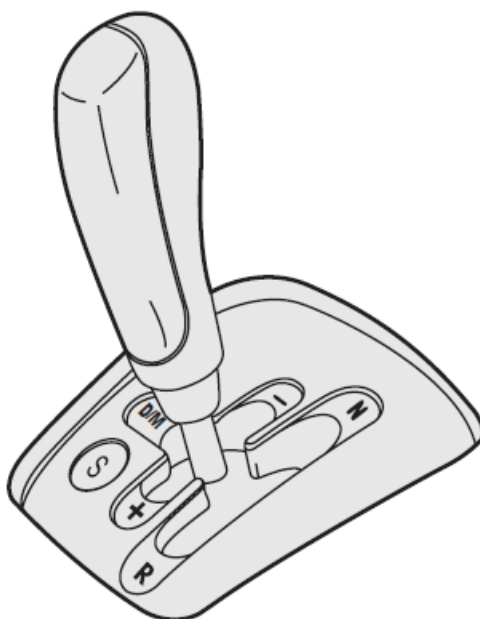
Fonte – Adaptado, Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 19)

## 2.10 Sensores

Sensores são dispositivos que transformam grandezas físicas (como por exemplo: temperatura, luminosidade, velocidade, sons e pressões) em sinais elétricos, os quais podem ser interpretados por um microcontrolador, como no nosso caso, e processado para se obter o valor da medição desejada.

Segundo o manual DUALOGIC® Diagnóstico (FIAT, 2007 p. 22), o sistema possui vários tipos de sensores como o de posição da alavanca de comando do câmbio, mais conhecido como *joystick* visto na figura 35, sensor da borboletas do volante, sensor de interruptor *SPORT*, rotação da árvore primária, posição do atuador de engate e seleção, posição de embreagem, pressão de óleo, interruptor do pedal de freio, sensor de velocidade do veículo e, por fim, interruptor da porta do condutor. Tais sensores são importantes para o bom funcionamento do NCD e serão abordados nos próximos tópicos.

Figura 35 - Joystick



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 06)



### 2.10.1 Sensor de alavanca de comando do câmbio (*joystick*)

Para realizarmos as trocas de marcha, é fundamental sabermos em qual posição o *joystick* se encontra, por isso na alavanca é instalado um circuito eletrônico que informa qual sua posição. O conjunto de alavancas possui um grupo de resistores e interruptores para informar ao NCD a posição que o motorista colocou o *joystick* (FIAT, 2007 p. 22). Tendo os seguintes comandos:

**UP (+):** engate sequencial das marchas durante a fase de aceleração

(1 → 2 → 3 → 4 → 5);

**DOWN (-):** engate sequencial das marchas durante a fase de desaceleração

(5 → 4 → 3 → 2 → 1);

**D/M:** alterna entre os modos AUTO e MANUAL;

**N:** coloca o câmbio em ponto-morto (neutro);

**R:** engrena a marcha a ré;

**BOTÃO S:** alterna entre os modos NORMAL e *SPORT*.

Tal sistema também é composto por sete sensores do tipo *Hall*, um para cada entrada da matriz lógica de diodos, e ao passar pela matriz lógica, teremos quatro saídas independentes demonstrado na figura 36, que nos informará qual a posição do *joystick* seguindo a seguinte lógica na tabela 3.

Junto ao *joystick* existe um ímã, no qual quando o *joystick* se movimenta produz um campo magnético o qual se é necessário para fazer a excitação dos sensores *hall* (FIAT, 2007 p. 22).

Figura 36 - Diagrama em blocos da matriz de diodos

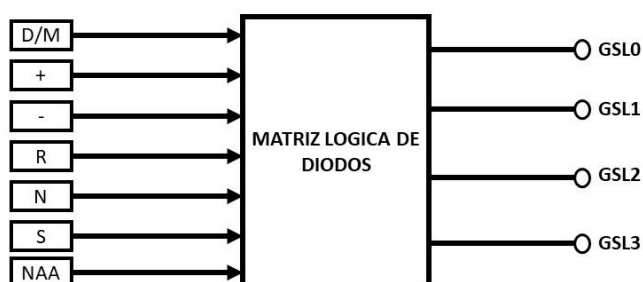


Tabela 3 - Lógica do sensor de *joystick*

	GSL3	GSL2	GSL1	GSL0
Falha	1	1	1	1
D	1	0	1	0
-	1	1	0	0
+	0	1	0	1
N	1	0	0	1
R	0	0	1	1
A/M	0	1	1	0
NAA	0	0	0	0

Fonte – Adaptado, Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 23)

Nível “1” = de 2,7V a 3,9V<sup>2</sup>

Nível “0” = de 0,7V a 2,1V<sup>3</sup>

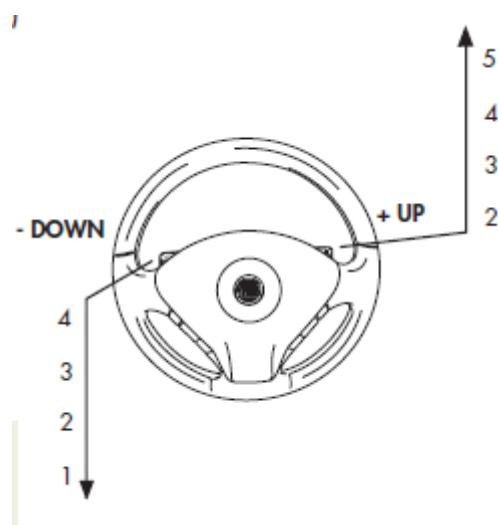
### 2.10.2 Borboletas do volante

As borboletas do volante são acessórios opcionais, que possuem seletores no lado esquerdo e direito do volante com sinais de “-” para representar o *down* e “+” para representar *up*, assim o motorista também pode optar pela troca de marcha no volante. A figura 37 demonstra onde estão localizadas as borboletas e as sequências de atuação delas.

<sup>2</sup> Se a tensão for acima de 3,9V, o sistema reconhece CA.

<sup>3</sup> Se a tensão for abaixo de 0,7V, o sistema reconhece CC GND

Figura 37 - Sistema das borboletas do volante



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p.23)

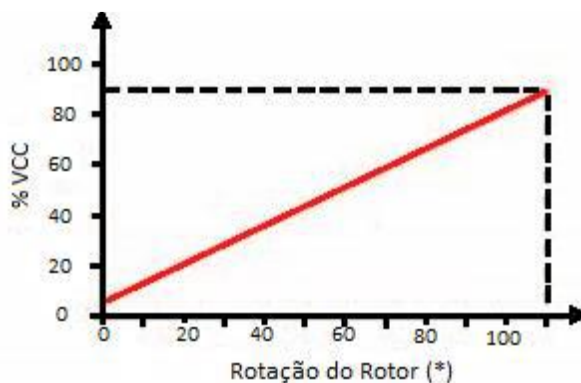
Se no caso ambos os sistemas de troca de marcha (*joystick* e borboleta) forem acionados juntos, o sistema sempre dará prioridade ao *joystick*.

### 2.10.3 Sensores de posição dos atuadores de seleção e engate

O sensor de posição dos atuadores de engate e seleção se trata de dois potenciômetros sem contato deslizante, ou como escreveu Florêncio et. al. (2013), também chamado de *full contactles*. Esse sensor utiliza um anel magnético o qual “aumenta a vida útil e a confiabilidade do sensor” (FIAT, 2007 p. 24).

Neste sensor, caso haja um aumento da rotação no rotor sua tensão aumenta proporcionalmente como exhibe o gráfico 2 e isso é devido a influência que o ímã exerce em um sensor de efeito *Hall* que nele se encontra o qual é responsável por gerar a tensão conforme o rotor gira. Dentro do sensor possui circuitos eletrônicos que são de fundamental importância para ler a tensão gerada pelo *Hall* e gerar a tensão de saída (FIAT, 2007).

Gráfico 2 - Sinal dos sensores de posição dos atuadores de seleção e engate



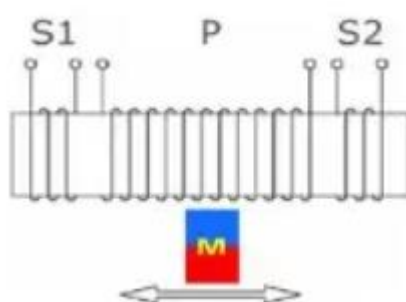
Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 25)

#### 2.10.4 Sensor de posição da embreagem (PLCD)

O sensor de posição de embreagem mede o deslocamento linear sem contato deslizante e com um ímã permanente.

Constituído basicamente por um três bobinas, uma central chamada de bobina primária, e duas nas extremidades denominadas de bobinas secundárias, e para o sistema funcionar se é necessário um ímã móvel conforme ilustrado na figura 38 (FIAT, 2007).

Figura 38 - Esquema de um sensor PLCD

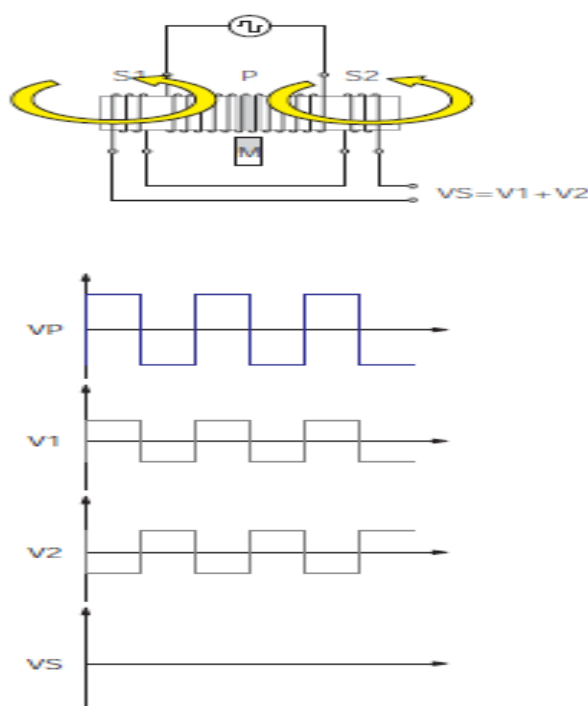


Fonte - Seminário: Automatizados (FIAT,2019)

Conforme explicado no manual do DUALOGIC®, o ímã permanente é responsável por “produzir uma região de saturação magnética no núcleo das bobinas, dividindo-o em duas regiões magnéticas”.

A principal função da bobina primária é produzir um campo magnético no núcleo do sensor para induzir também um campo magnético nas bobinas das extremidades, o qual se é formado as tensões  $V1$  e  $V2$  (defasadas em  $180^\circ$ ), que ao se somar as duas tensões temos tensão de saída ( $V_S$ ), por isso se é ligada uma tensão alternada sobre a primária. Quando o ímã está no centro, as bobinas secundárias produzem as mesmas tensões, logo  $V1 = V2$ , e como são defasadas em  $180^\circ$  a tensão de saída é igual a zero, conforme demonstra a figura 39.

Figura 39 - Ilustração dos campos magnéticos no sensor PLCD e gráfico das tensões (1)

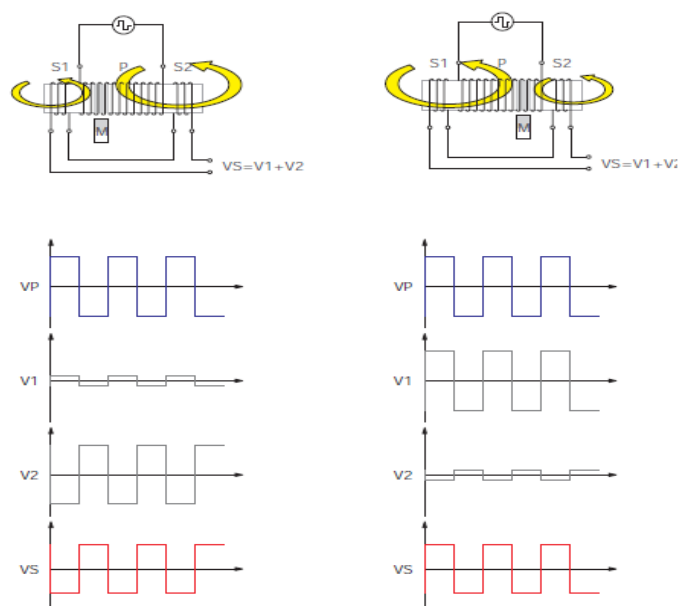


Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 26)

Assim quando, o ímã se locomove para a direita ou para a esquerda, as tensões  $V1$  e  $V2$  das bobinas secundárias mudam, o que influencia também na tensão de saída conforme mostra nos gráficos da figura 40.

“Se a tensão  $V_S$  tem a mesma fase que a tensão no primário ( $V_i$ ), significa que o deslocamento foi para um determinado lado. Já uma defasagem de  $180$  graus indica que o ímã foi deslocado para o outro lado.” (FIAT, 2007).

Figura 40 - Ilustração dos campos magnéticos no sensor PLCD e gráfico das tensões (2)



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 26)

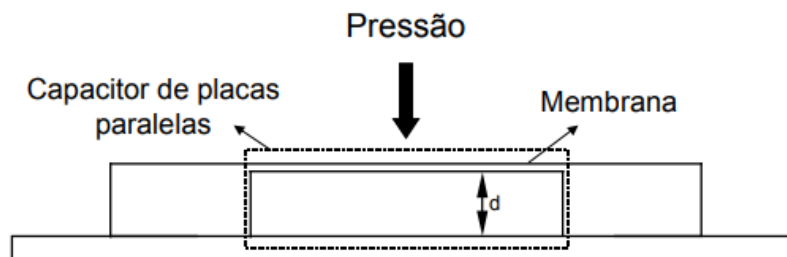
### 2.10.5 Sensor de pressão do óleo

Esse sensor está conectado na parte do grupo hidráulico do câmbio e é responsável por verificar e informar ao NCD qual a pressão em que o óleo das eletroválvulas está trabalhando.

Seu funcionamento consiste em um elemento sensível capacitivo que possui uma membrana, ou diafragma, que tem como finalidade substituir uma das placas de um capacitor. De acordo com Garcia (2006) ao se aplicar uma pressão, o diafragma se deforma e como consequência o valor da capacitância se altera, ou seja, a variação da capacitância está em função da pressão.

A figura 41, mostra o esquema simplificado de um sensor capacitivo de placas paralelas.

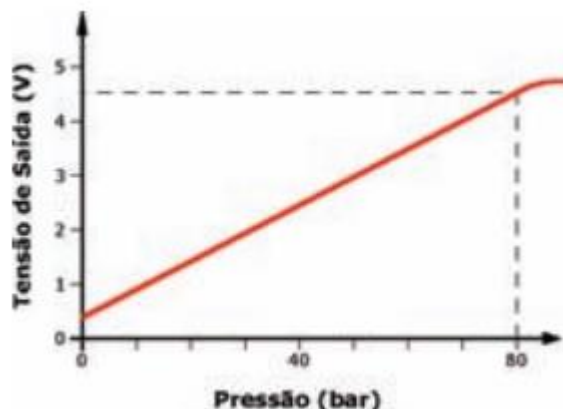
Figura 41 - Esquema simplificado de um sensor capacitivo de placas paralelas



Fonte - Garcia (2006)

Existem técnicas diferentes para realizar a leitura da capacitância, sendo as mais usuais a conversão em tensão e em frequência (Garcia, 2006). E conforme mostrado no gráfico 3, o nosso sensor possui componentes internos que transformam a capacitância em tensão.

Gráfico 3 - Curva da tensão de saída do sensor de pressão em função da pressão



Fonte – Câmbio Dualogic Diagnóstico®, Treinamento da Rede. Fiat (2007, p. 27)

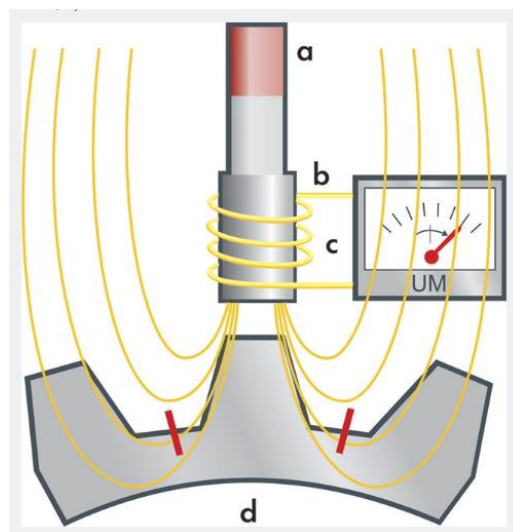
### 2.10.6 Sensor de rotação da árvore primária

O sensor de rotação da árvore primária, ou rotação da embreagem, está localizado diretamente no câmbio (FIAT, 2007 p. 28).

Trata-se de um sensor de relutância magnética que é constituído por um ímã permanente, um indutor e um núcleo de material ferromagnético. Para o funcionamento dele, é necessário que haja variação de fluxo magnético ilustrado na

figura 42. Sendo assim, no caso do sensor do DUALOGIC® é colocado na engrenagem de 2º marcha para que cada dente e espaçamento da engrenagem haja uma variação de fluxo magnético, gerando assim uma tensão segundo a figura 43, onde 'a' é o ímã, 'b' é o núcleo ferromagnético, 'c' é bobina e 'd' é a roda gerador de impulsos.

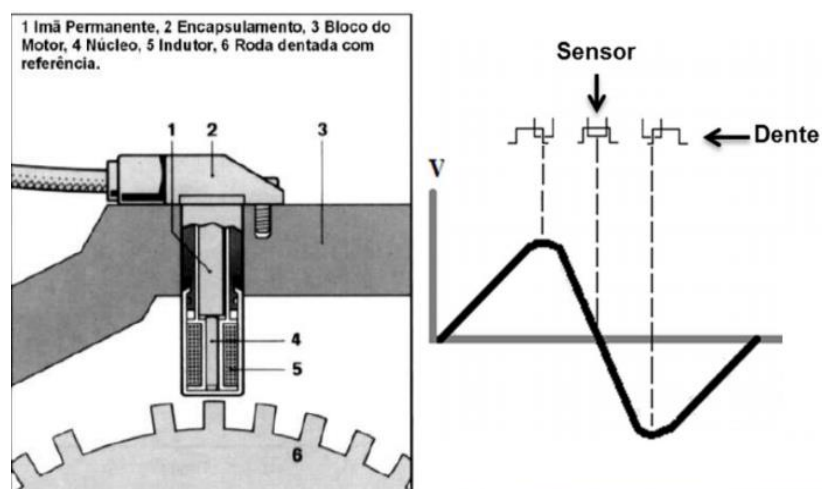
Figura 42 - Esquema de um sensor de relutância magnética



Fonte – Redação. **Capítulo4: Rodas Fônicas**. Disponível em:

<<http://www.noticiasdaoficinavw.com.br/v2/2013/06/capitulo-4-rodas-fonicas/>>. Acesso em: 16 Fev. 2020, 15:28:55.

Figura 43 - tensão gerada pelo sensor de relutância magnética



Fonte: Adaptado, Bosch (2005, p. 119)



Sendo assim, a frequência de saída do sensor, está diretamente ligada à rotação do eixo do câmbio.

### **2.10.7 Interruptor Sport**

Este sensor é montado no *joystick* e é responsável por informar ao NCD se o condutor solicitou o modo *sport*.

### **2.10.8 Interruptor de freio**

O NCD recebe informações via barramento CAN do interruptor de freio do nó *body computer* (NBC), o qual verifica se a informação do CAN é coerente com o interruptor de freio que possui um contato NA (normalmente aberto), assim essa informação é replicada para NCD via cabo elétrico permitindo a troca de marchas com o veículo parado e a estratégia da troca de marchas durante a desaceleração do veículo (FIAT, 2007 p. 28).

### **2.10.9 Sensor de velocidade do veículo**

Este sensor é instalado diretamente do câmbio e é responsável por informa ao módulo de controle da transmissão qual a velocidade do veículo para calcular o momento das trocas de marcha e adotar estratégias no acionamento da embreagem (FIAT, 2007 p. 29).

### **2.10.10 Interruptor da porta do condutor**

Este sensor verifica se a porta do motorista está aberta ou fechada para o NCD realizar o ligamento e desligamento da eletrobomba de óleo e para adotar estratégia do controle do câmbio quando o veículo estiver parado (FIAT, 2007 p. 29).

## **2.11 MC33816**

Existem alguns drives de controle de corrente, no mercado como por exemplo o DV110 da empresa *Texas Instruments* (TI), porém este circuito integrado (CI) controla apenas uma saída de um atuador, ou seja, necessitaríamos de cinco desses para fazermos o controle das solenoides, e não possui um sistema para se trabalhar em malha fechada para o controle da corrente.

Foi escolhido para o projeto o MC33816 que é um drive programável que tem como principal objetivo o controle de corrente de dispositivos de potência com alta

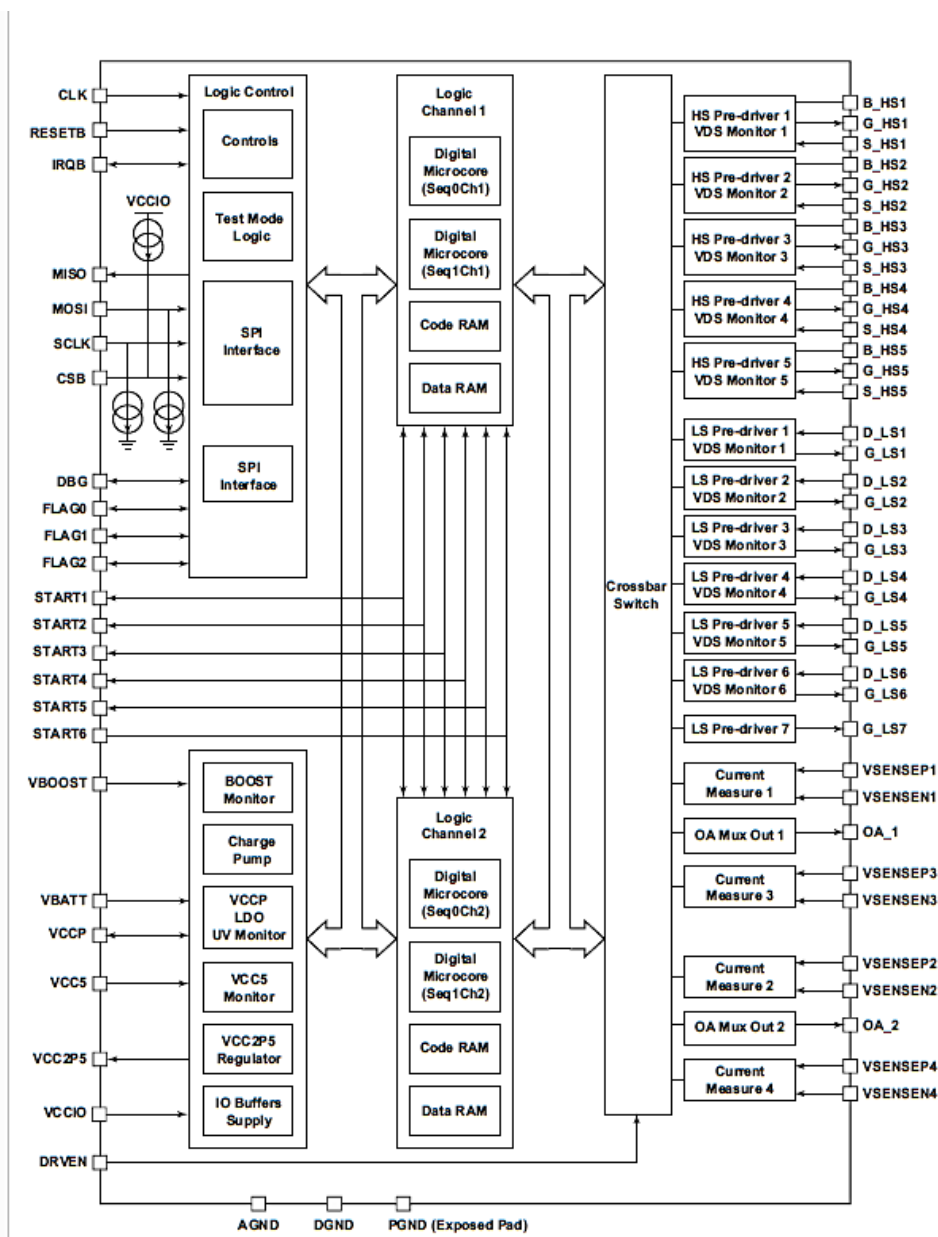
precisão, devido trabalhar em malha fechada. Na indústria é utilizado para controle de injeção direta a diesel e à gasolina e controles de eletroválvulas de transmissão.

Conforme mostra a figura 44, o dispositivo da empresa *FREESCALE* possui dois blocos de canais lógicos, os quais cada um possui uma memória de acesso aleatório (RAM) de programa, uma RAM de dados e 2 $\mu$  cores fazendo com que possam ser rodados até quatro microcódigos simultaneamente. Além de possuir, quatro blocos para medição de corrente, diagnóstico de falhas embutido, monitoramento da fonte de alimentação, monitoramento térmicos e outras funções.

Possui seis entradas STARTx o qual quando ativada inicia o ciclo de trabalho da saída respectiva.

Suas saídas de potência são associadas a cinco MOSFETs, canal N, externos para a parte alta da configuração (HS) e a sete MOSFETs, também do tipo canal N, externos para a parte baixa da potência (LS).

Figura 44 - Diagrama de Blocos do MC33816

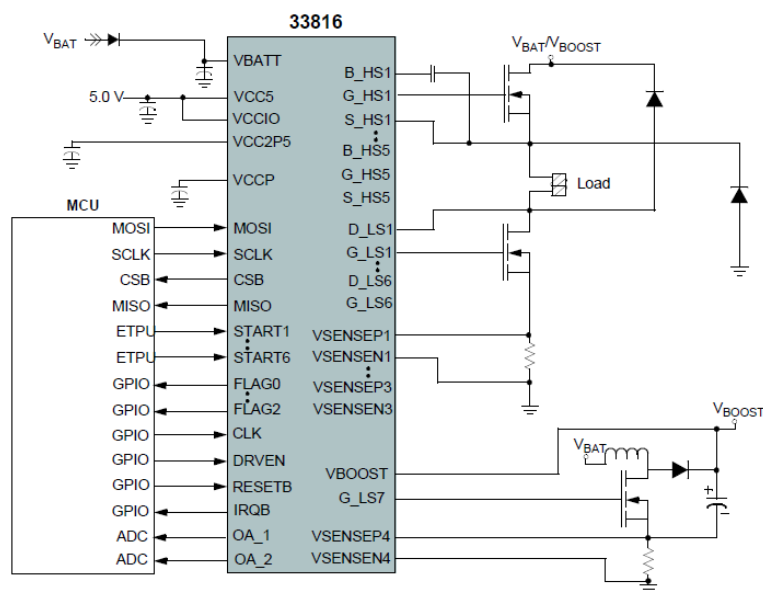


Fonte - Datasheet MC33816 (NXP, 2019)

Sua interface é realizada pelo protocolo *Serial Peripheral Interface* (SPI) 10 MegaBits por segundo (Mbps) de velocidade, e assim podemos ter uma comunicação com o(s) microcontrolador(es) do sistema de controle.

A figura 45, demonstra um exemplo de aplicação do MC33816 com as informações já citadas sobre ele. Está aplicação é mais comum do CI nos sistemas o qual se é aplicado, conforme dita seu manual, e é neste circuito que iremos tomar como base para construir o sistema de controle, pois já é um circuito que também é utilizado em seu kit de desenvolvimento.

Figura 45 - Exemplo de ligação esquemática no MC33816



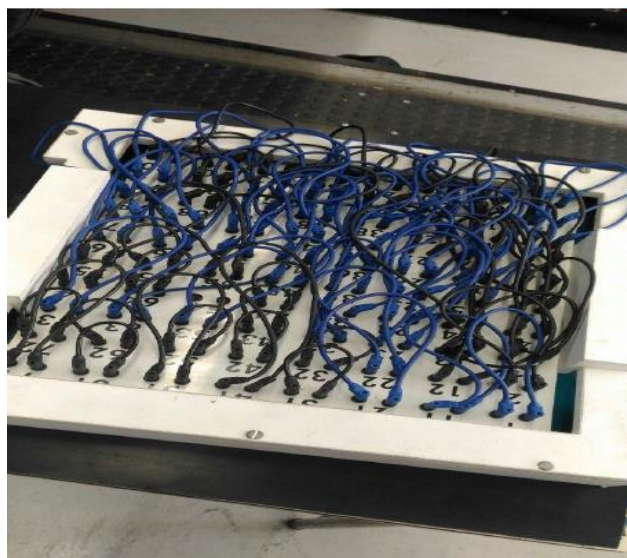
Fonte - Datasheet MC33816 (NXP, 2019)

### 3 AQUISIÇÕES DE DADOS

Para validarmos todas as informações adquiridas durante o processo de construção de base teórica bibliográfica, começamos a realizar alguns testes no carro STRADA da montadora FIAT, o qual possui um câmbio DUALOGIC® instalado.

Um dos materiais mais utilizados foi o *Break Out Box* (BOB), mostrado da figura 46. Assim podemos conectar a ECU nele e verificar todos seus sinais que nos são aproveitáveis.

Figura 46 - BOB



Fonte - Bento, et al. (2017)

Todos os pinos do BOB possuem uma numeração o qual são de acordo com a numeração dos 80 pinos da ECU, porém os usados para o controle do NCD são os que mostram a tabela 4.

Tabela 4 - PIN OUT NCD

n° Pino	Descrição
1	Massa de potencia
2	Massa de Potencia
3	Saída de comando para Eletroválvula 4
7	CAN L- conexão com o nó de controle do motor
19	CAN H - conexão com o nó de controle do motor
26	Sinal do sensor do joystick (GSL 0)
27	Entrada de alimentação +30 via CVM - F18-10 A
28	Entrada de alimentação sob chave via CVM - F16-7,5 A
29	Saída de comando para a Eletroválvula 3
31	Saída de comando para o relé da eletrobomba
32	Saída de comando para a Eletroválvula 1
33	CAN L – conexão com o nó do body computer
34	Saída para o sensor de posição da embreagem (bobina primária)
35	Entrada de sinal do sensor de posição da embreagem (bobina secundária)
36	Entrada do sinal do sensor de velocidade (Conexão com o nó do body computer)
38	Entrada do sensor de rotação da árvore primária (+)
39	Entrada de sinal do sensor de posição do atuador de engate
40	Entrada de sinal do sensor de pressão de óleo
42	Saída de comando para o relé de partida T-20
43	Saída de comando para a Eletroválvula 0
44	Saída de comando para a Eletroválvula 2
45	CAN H – conexão com o nó do body computer
46	Saída para o sensor de posição da embreagem (bobina primária)
47	Entrada de sinal do sensor de posição da embreagem (bobina secundária)
49	Linha K
50	Sinal do sensor de rotação da árvore primária (-)
51	Entrada de sinal do sensor de posição do atuador de seleção
65	Saída de alimentação negativa para o sensor do joystick e das borboletas do volante
66	Saída de alimentação negativa para o sensor de pressão de óleo e para os sensores dos atuadores de engate e de seleção.
67	Entrada de sinal do sensor do joystick (GSL 1)
68	Entrada de sinal do sensor do joystick (GSL 3)
69	Entrada de sinal do interruptor de freio (contato NA)
73	Saída de alimentação positiva para o sensor de pressão de óleo e para os sensores dos atuadores de engate e de seleção.
74	Entrada de sinal do sensor do joystick (GSL 2)
75	Entrada de sinal do sensor de mudança no volante (Borboletas do volante)
76	Entrada de sinal de partida. Vem do comutador de ignição
77	Entrada de sinal do botão SPORT
78	Entrada de sinal do sensor de porta aberta

Fonte – Adaptado. Manual DUALOGIC® (FIAT,2007)

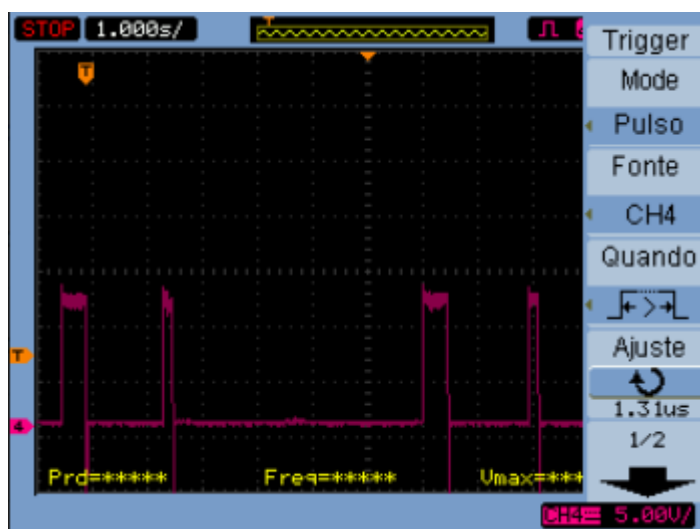
### 3.1 Testes dos solenoides

O primeiro teste a ser realizado pelo grupo foi o teste dos sinais provenientes das eletroválvulas diretamente no carro, assim validamos os sinais dos solenoides do tipo *on/off* e os sinais das que são proporcional a pressão.

Na figura 47 temos o sinal proveniente da EV3 (osciloscópio conectado ao pino 29 do BOB), uma das eletroválvulas de seleção do tipo *on/off*. Vemos quatro pulsos no osciloscópio. O primeiro pulso é equivalente ao engate da 1ª marcha, o segundo pulso é para desengate de 1ª marcha para a EV4 (pino 3) e EV2 (pino 44) realizar o engate da 2ª. O terceiro e quarto pulsos possuem a mesma ideologia, porém para a 3ª e 4ª marchas respectivas.

Como podemos observar os pulsos são de aproximadamente sua tensão de alimentação (12v). Para os engates das marchas a duração é de aproximadamente 500ms e para o desengate um tempo bem menor, cerca de  $\frac{1}{4}$  para o engate. Esta lógica é seguida para as EV3 e EV4.

Figura 47 - sinais da eletroválvula de seleção EV3

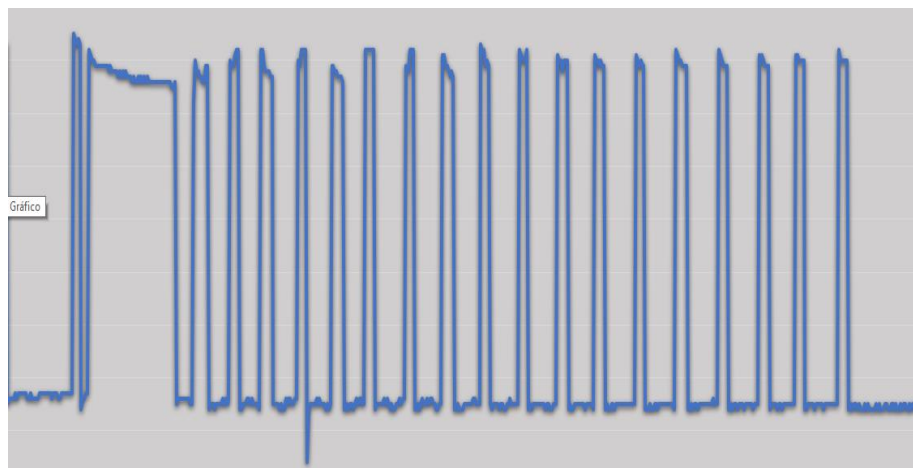


Fonte - Autor (2019)

A figura 48 demonstra um sinal de uma das eletroválvulas proporcionais a pressão (EV1). Neste caso podemos observar que seu acionamento é diferente da anterior, pois não é comandada apenas por um pulso, e sim por um sinal denominado modulação por largura de pulso (PWM - *Pulse width modulation*), ou seja, de acordo com a largura do pulso que envio a eletroválvula, ela terá mais ou menos corrente fazendo o controle de pressão do óleo admitida por ela.

Os pulsos também atingem aproximadamente a tensão de alimentação, e tem sua duração está diretamente ligada ou pulso das eletroválvulas de seleção, ou seja, enquanto houver pulso na eletroválvula de seleção haverá sinal nela também.

Figura 48 - sinais da eletroválvula de seleção EV1

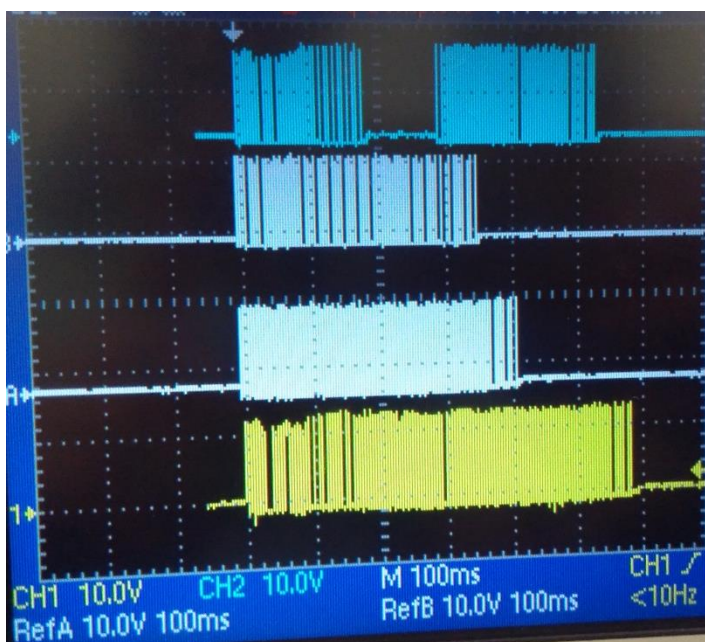


Fonte - Autor (2019)



Nos testes realizados posteriormente, tendo como foco os solenoides de engate/desengate (EV1 e EV2), apresentou tempos diferentes de engate para cada marcha. Na figura 49 observamos quatro sinais, o primeiro em cor azul temos a EV2 no engate de 3° marcha, o segundo sinal em branco EV2 em 1° marcha, o terceiro em branco temos EV1 (pino 32) em 1° marcha e por último em amarelo temos EV1 em 3° marcha.

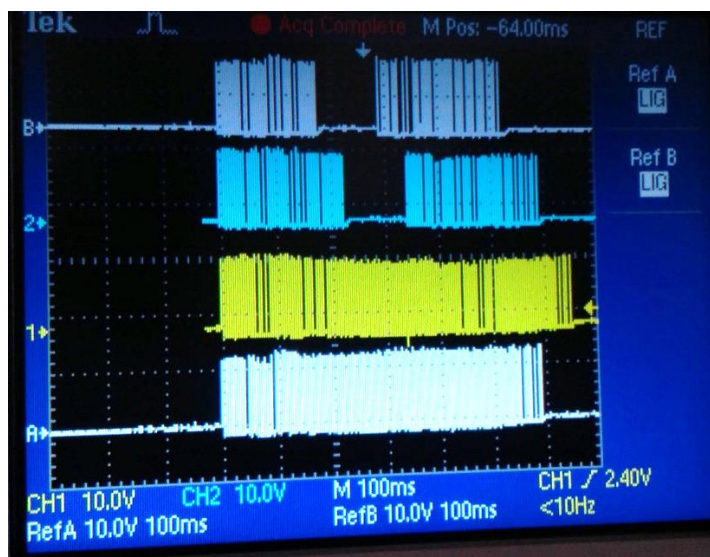
Figura 49 - Sinais EV1 e EV2 1° e 3° marcha



Fonte – Autor (2019)

Na figura 50 temos as mesmas eletroválvulas, porém com a 3° e 5° marcha, ou seja, o primeiro em branco é EV2 em 3° marcha, o segundo em azul EV2 em 5° marcha, em terceiro em amarelo EV1 em 5° marcha e por fim em branco EV1 em 3° marcha.

Figura 50 - Sinais EV1 e EV2 3° e 5° marcha



Fonte - Autor (2019)

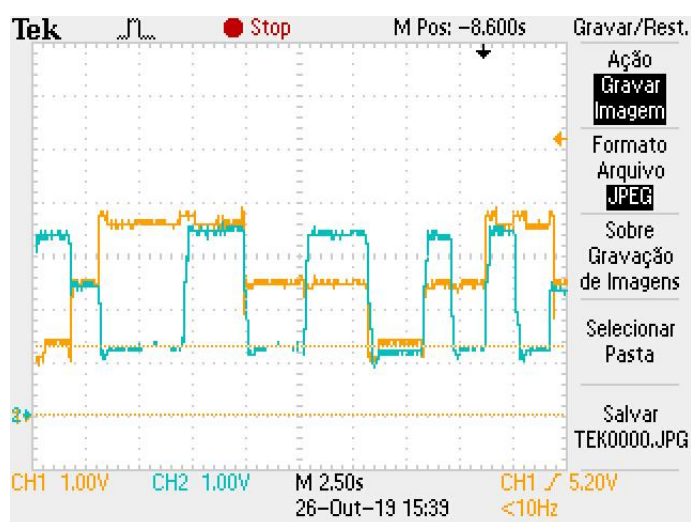
Pode-se notar a base de tempo para o engate de cada marcha não é constante. O teste mostrou que quanto maior a rotação do motor maior será o tempo para o engate da marcha selecionada.

### 3.2 Testes do sensor de atuador de engate e seleção

Para realizar esse teste foi-se necessário colocar o osciloscópio nos pinos 39 e 51 os quais são as entradas dos atuadores de engate e seleção respectivamente. Foi-se medido os sinais nesses pinos em posição R, N, 1°, 2°, 3° 4° e 5° marcha enquanto um dos integrantes do grupo realizava a aceleração, desaceleração e a troca de marcha.

A figura 51 ilustra os sinais provenientes nesses sensores em conjunto sendo que o sinal em azul é o sinal do sensor do atuador de engate e em amarelo o de seleção.

Figura 51 - Sinais dos sensores do atuador de engate e seleção



Fonte – Autor (2019)

Com essa aquisição de dados foi possível montar a tabela 5 que faz um resumo das informações de tensões em cada marcha selecionada.

Tabela 5 - Tensões nos atuadores de engate e seleção

Marcha	Tensão (V)	
	Engate	Seleção
R	3,5	1,4
N	2,5	2,5
1 <sup>a</sup>	1,4	3,5
2 <sup>a</sup>	3,5	3,5
3 <sup>a</sup>	1,4	2,5
4 <sup>a</sup>	3,5	2,5
5 <sup>a</sup>	1,4	1,4

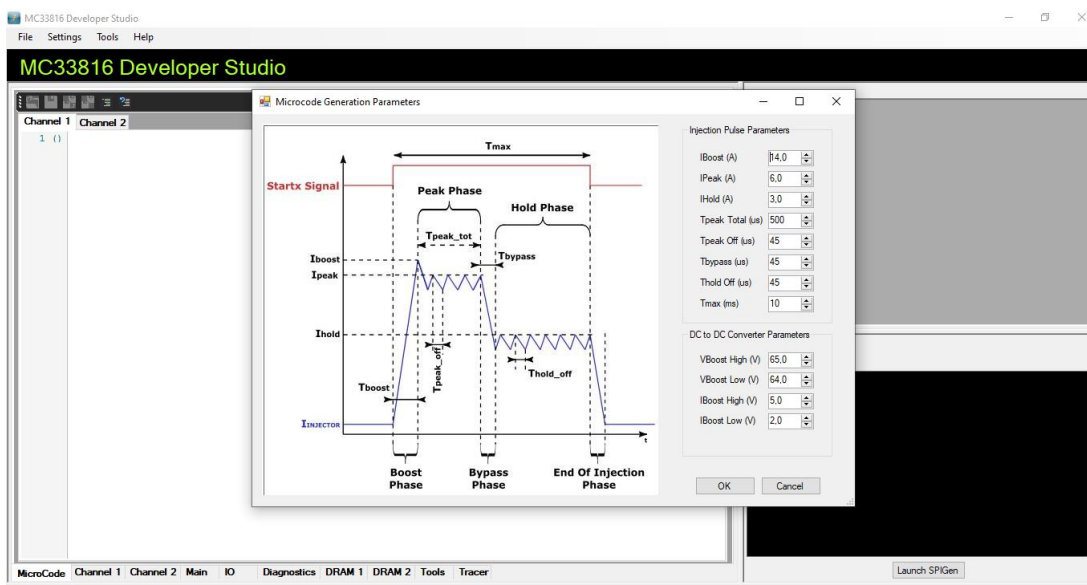
Fonte – Autor (2019)

### 3.3 Controlando o MC33816 com o kit de desenvolvimento KL25Z

Para o início do controle do drive MC33816 foi-se necessário baixar dois programas disponibilizados pela NXP. Um deles é o MC33816 DEV Studio, o qual nele será possível realizar as configurações e a programação do dispositivo, e o segundo programa necessário é o SPIGen, que é responsável por transformar o código gerado no MC33816 DEV Studio em comunicação SPI para *drive*.

Primeiramente temos que abrir o MC33816 DEV Studio, ir em “*file*”, criar um novo arquivo, nomear e escolher a pasta de destino, e assim que clicamos em “*create*” aparece uma tela com os parâmetros que você deseja configurar de corrente, tensão e tempo conforme mostra a figura 52.

Figura 52 - Configuração da curva do pulso

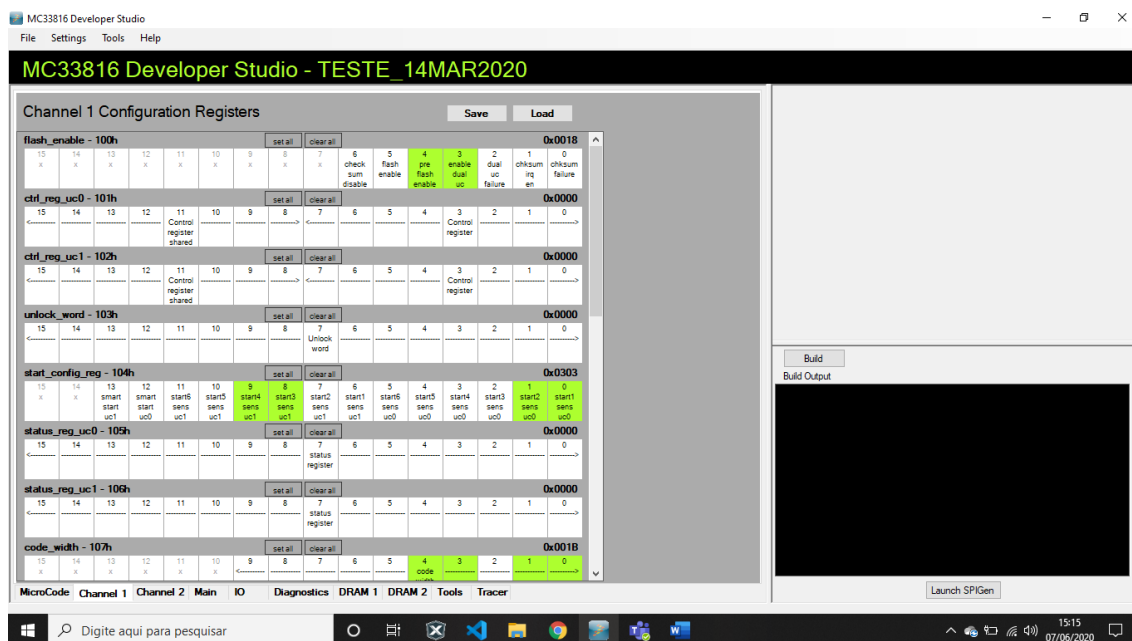


Fonte 1 - Autor (2020)

Assim que configurado os parâmetros desejados e clicado em “ok”, o programa gera um código base. Na parte superior da tela, demonstrado na figura 53 podemos ver o *channel 1* e *channel 2*, na parte de *MicroCode*, o qual o programa divide as programações da seguinte forma: micro código 1 será responsável pelo controle das eletroválvulas ou injetores ligados das entradas, e o micro código 2 que é responsável pela fonte DC-DC. E na parte inferior vemos dez abas conforme descrito.

A primeira aba é a do *MicroCode*, que foi citado no parágrafo anterior. A segunda e a terceira abas da parte inferior são os *Channel 1* e *Channel 2* respectivamente, essa parte é responsável por configurar os registradores necessários para a sua utilização.

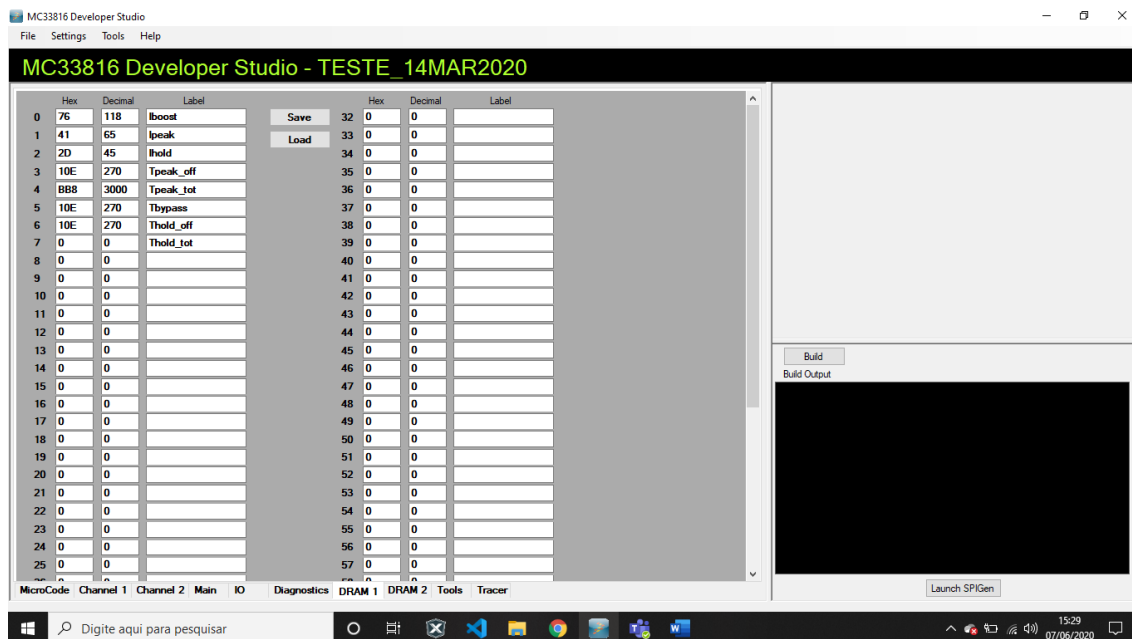
Figura 53 - Configuração dos registradores do Micro Code



Fonte 2 - Autor (2020)

A aba seguinte, denomina-se “main”. Nessa parte configura-se os registradores de uso geral como por exemplo a pré escala do *clock*, as *flags* e as interrupções. A quinta aba chama-se “IO”, o qual configuramos as entradas e saídas que iremos utilizar no MC33816. A próxima aba é a de *Diagnostics*, aqui são os registradores que informam quando há algum erro de sobre tensão, ou outro erro configurado para ser informado. A sétima e a oitava abas são a DRAM1 e DRAM2 respectivamente, que são as Memórias de Dados dos micro core 1 e micro core 2 conforme demonstrado na figura 54, o qual mostra o endereço da memória, o valor armazenado no sistema decimal e hexadecimal e o nome da variável atribuída aquele endereço.

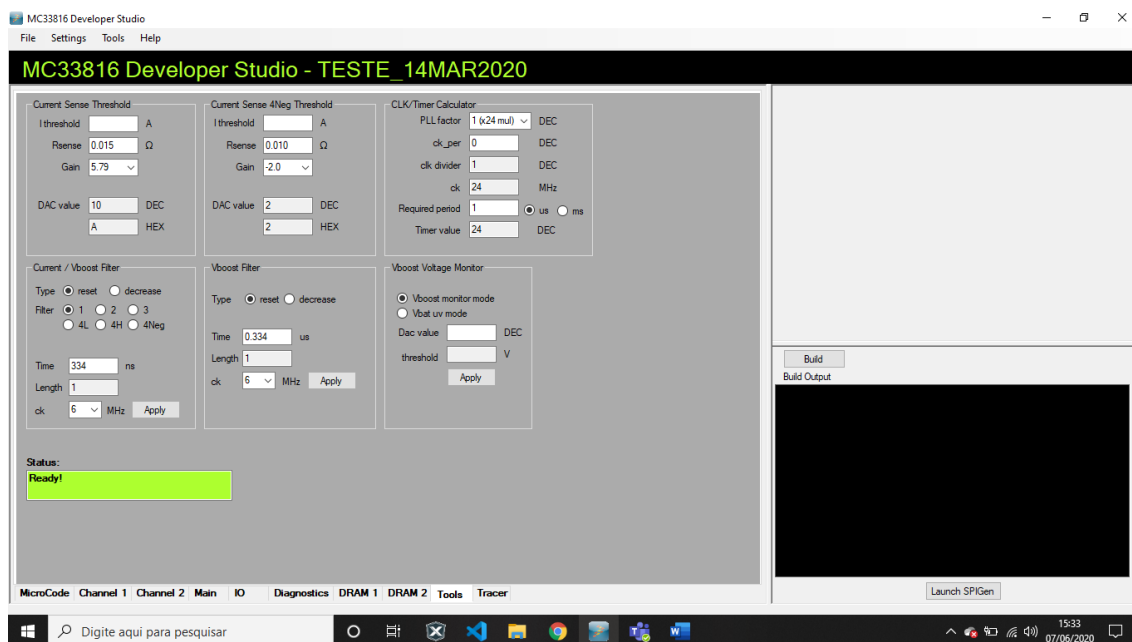
Figura 54 – Endereço da Memória de Dados



Fonte 3 - Autor (2020)

A figura 55, mostra a aba *Tools*, o qual temos as configurações para o sensor de corrente, configuração do *clock* e de das tenções de para monitoramento.

Figura 55 – Ferramentas para configuração de parâmetros de trabalho



Fonte 4 - Autor (2020)

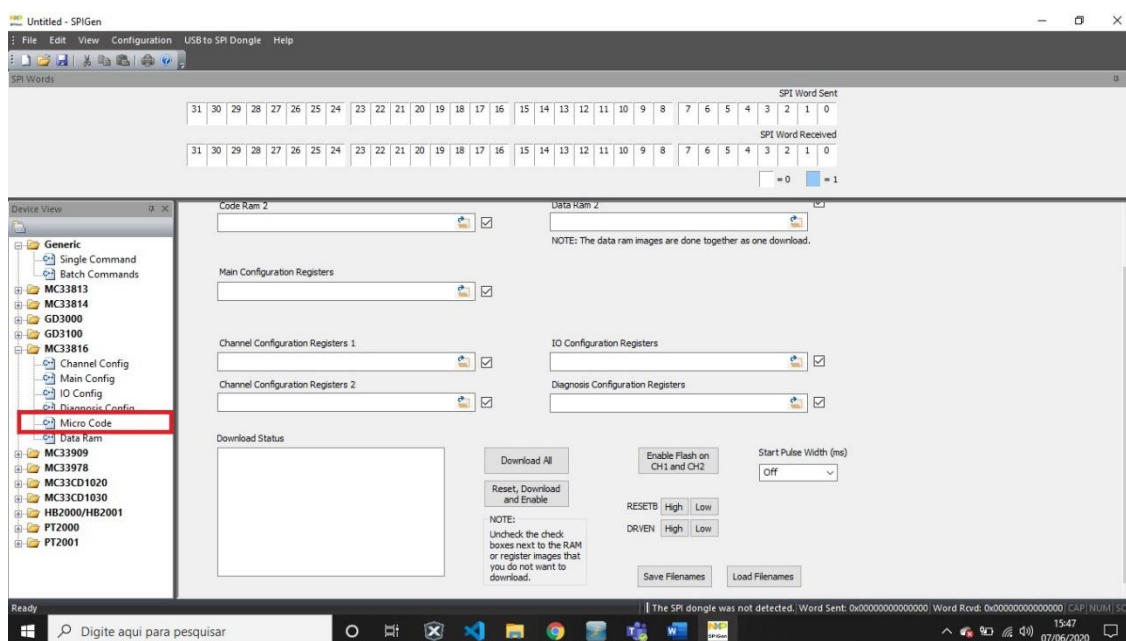
A aba seguinte o *Tracer*, são outras configurações que não utilizaremos neste momento.

### 3.3.1 Rodando o programa

Assim que realizarmos a criação do programa, e ter gerado o programa base, devemos apenas verificar se está tudo certo com os “.” onde necessário, pois a linguagem *assembly* não aceita “,” como separador decimal de números, como na hora de configurar os ganhos dos amplificadores operacionais no micro core 1.

Após ter ajustado, clicamos em “*build*” para copilar, e depois em “*Launch SPIGen*”, que abrirá e carregará o arquivo no SPIGen para realizarmos a comunicação SPI com o nosso dispositivo com o USB já conectado na placa. A figura 56 mostra a tela do SPIGen na pasta MC33816 na parte *micro code*, pois é aqui onde carregaremos os nossos arquivos gerados pelo MC33816 DEV Studio.

Figura 56 - Compilador SPIGen do MC33816





Para carregar todos os arquivos clicamos em “*Load FileNames*”, vamos até a pasta onde salvamos e procuramos o arquivo nomeado de SPIGenMC33816Files.

Assim que abrirmos esse arquivo, as partes “*Code RAM 1*”, “*Code RAM 2*” e as demais serão preenchidas em seus devidos lugares. Agora basta apenas clicar em “*Reset*”, “*Download and Enable*” para todos os parâmetros e programações serem enviadas pelo protocolo SPI para o MC33816.

Na parte *star pulse width (ms)* configuramos a largura do pulso do “*STARTx*” para podermos realizar o controle na parte.

Para simularmos os pulsos de STARTx, vamos à parte *Generic* e observamos duas opções. A primeira é o “*Single Command*”, a qual enviamos um comando de cada vez ao MC, e a segunda é o “*Batch Commands*”, o qual nos possibilita enviar uma sequência desejada de comandos. A figura 57 e a figura 58, mostra respectivamente a tela da parte “*Single Command*” e “*Batch Command*”, os quais temos 9 opções de botões que são, *Data 0*, *Data 1*, *Data 2*, *Data 3*, *Data 4*, *Control 0*, *Control 1*, *Control 2* e *Control 3*, que podemos identificar como *Data 0* e *Data 1* como *RESETB* e *DRVEN*, de *Data 2* a *Control 2* interpretamos como *START1* a *START 6*, e por fim o *Control 3* Como *IRQB*.

Ao colocar um atuador nas saídas respectivas, podemos ver a saída operando

Figura 57 - Controles das sequências das saídas do MC33816

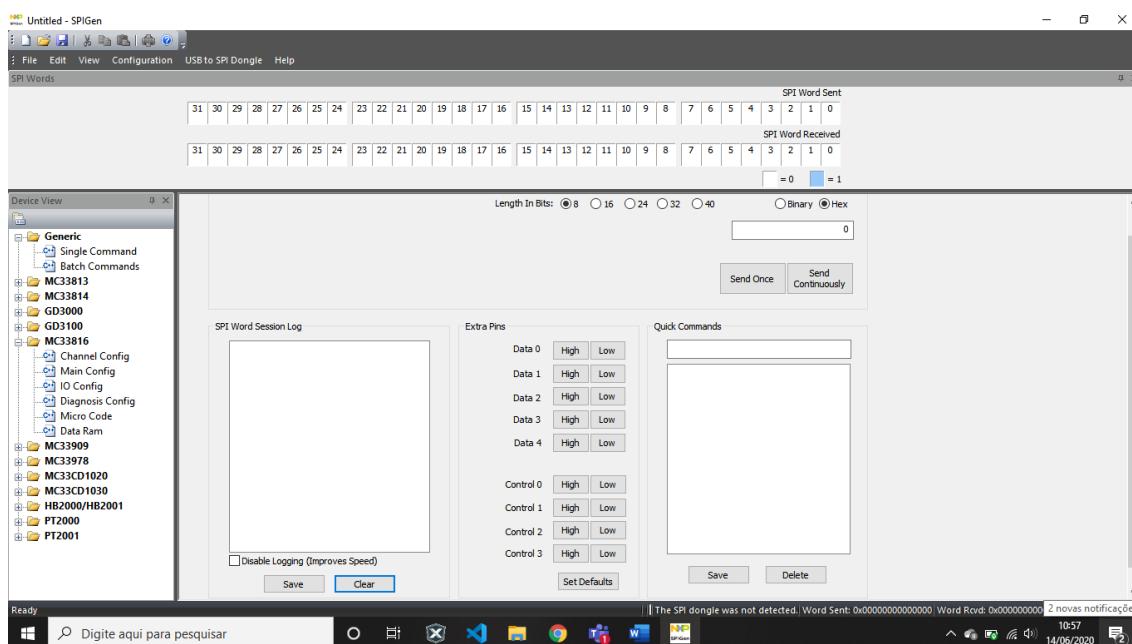
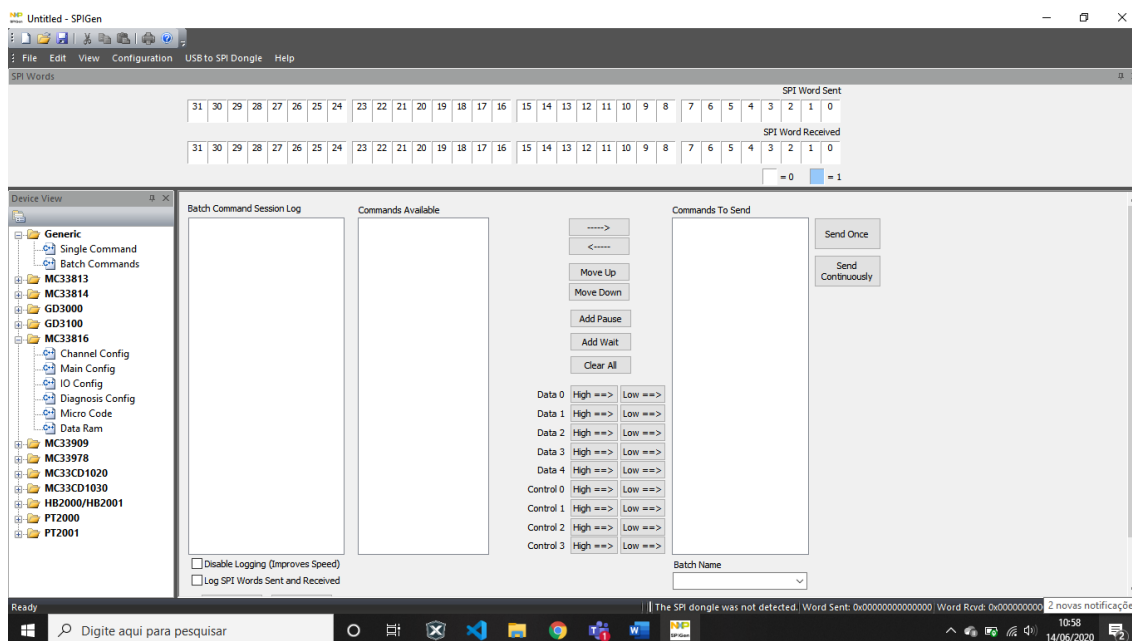


Figura 58 – Comparação da sequência das saídas do MC33816 com acionamento



Fonte 7 - Autor (2020)

### 3.4 Comunicação entre o PIC18F4550 e o MC33816

Após feito a comunicação entre a placa do kit de desenvolvimento KL25Z, agora partimos para a comunicação entre o PIC18F4550 da placa padrão FATEC e o MC33816.

Para ajudar a entender alguns comandos de escrita e leitura, foi conectado junto ao kit de desenvolvimento KL25Z um analisador lógico 24MHz de 8 canais, demonstrado na figura 59. Com os canais do analisador conectado no CLK, MISO, MOSI e CS<sup>4</sup> da placa podemos enviar um comando pelo programa SPIGen e verificar os valores enviados.

---

<sup>4</sup> CLK: *Clock* - Sincronismo de tempo entre mestre e escravo – MISO: *Master Input Slave Output* – Informação flui do escravo para mestre – MOSI: *Master Output Slave Input* – Informação flui do mestre para o escravo – CS: *Slave Select* – Seleção de momento da comunicação entre o mestre e o escravo.

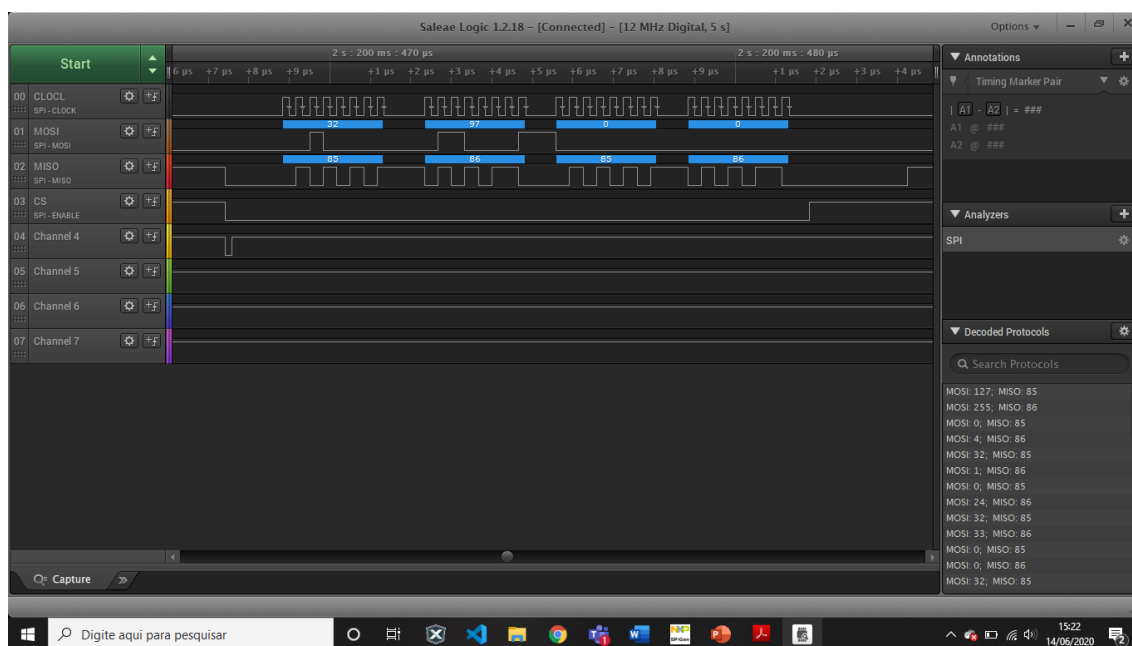
Figura 59 - Analisador logico 24MHz de 8 canais



Fonte 8 - Auto Core Robótica. Disponível em: <<https://www.autocorerobotica.com.br/analizador-logico-usb-8-canais-24mhz>>. Acesso em 17 Junho 2020, 01:32.

Ao enviamos alguns comandos podemos verificar na tela do programa *Logic* 1.2.18 , programa específico para o analisador, os dados enviados e recebidos na comunicação SPI conforme demonstra a figura 60, mostrando um exemplo de dado apurado pelo analisador, com a primeira linha o *CLOCK*, a segunda o *MOSI* do MC33816, a terceira linha o *MISO* do MC33816 e na quarta linha tenho o *CS*.

Figura 60 - Comunicação SPI do MC33816



Fonte 9 - Autor (2020)

Analisando alguns dados foi possível ver que ao iniciar a configuração de um determinado conjunto de registradores, por exemplo os registradores do *MAIN*, existe um conjunto de dados iniciais de 4 *bytes* que inicia a comunicação, isso para cada conjunto de registradores, e ao final possui uma sequência de 8 *bytes* que finalizam a comunicação por completo. Com esses dados montamos a lógica de programação no programa MPLABX para o PIC conforme demonstrado no Apêndice A, e realizamos a ligação entre o MC33816 e o PIC conforme anotado na tabela 6.

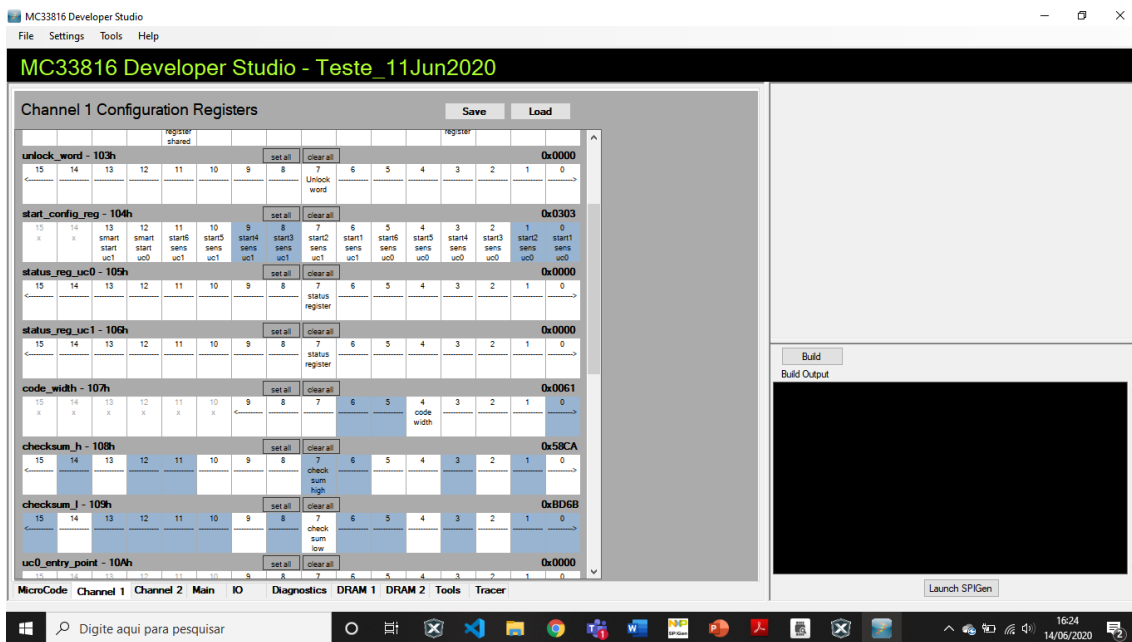
Tabela 6 - Pinos comunicação SPI

MC33816	PIC
CLOCK	PORTB.RB1
MISO -MC88816	PORTB.RB0
MOSI – MC33816	PORTC.RC7
CHIP SELECT	PORTB.RB3

Fonte 10 - Autor (2020)

Para configurar cada endereço de cada registrador foi realizado anotado os nomes e endereços de cada registrador do MC33816 conforme demonstrado no Apêndice B e olhando o programa do MC33816 DEV STUDIO, verificamos os valores de cada endereço e colocamos no função de configuração de cada endereço mostrado no Apêndice C que são as definições do projeto. A figura 61 mostra a tela do MC88816 DEV STUDIO dos registradores do *MAIN* como exemplo.

Figura 61 - Registradores do MC33816

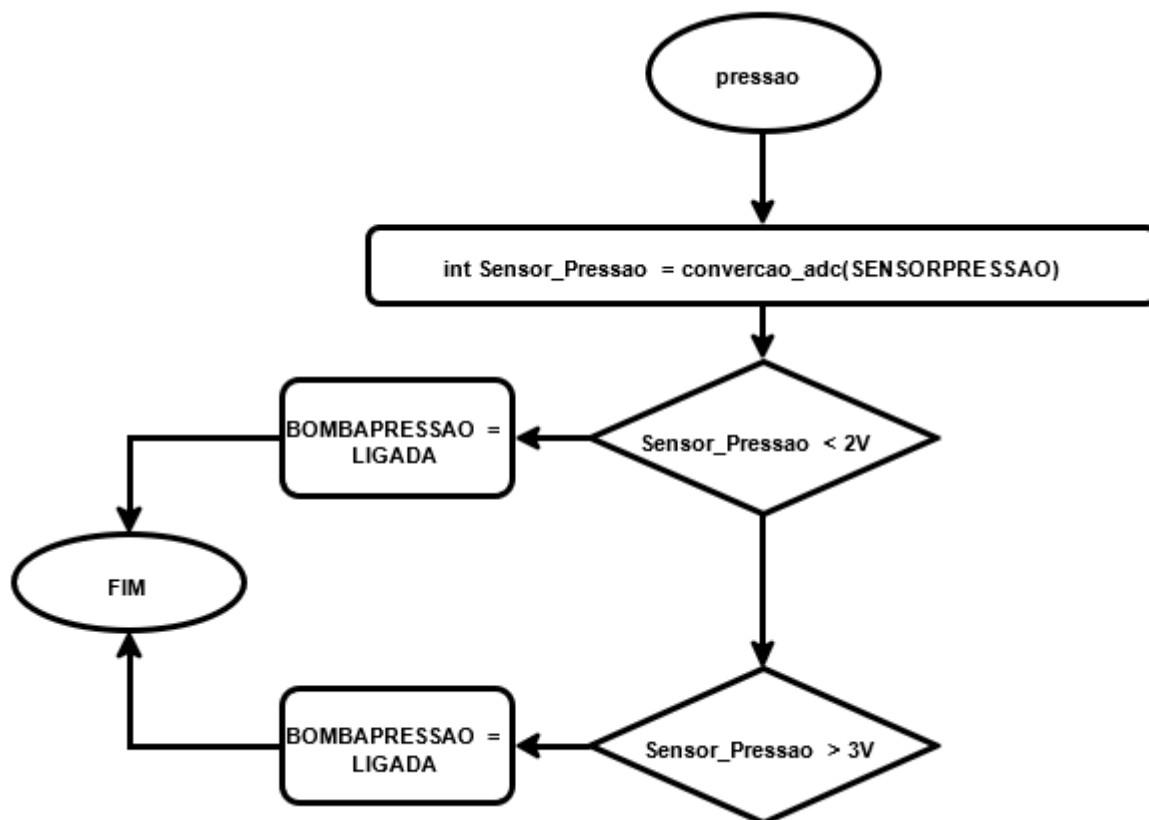


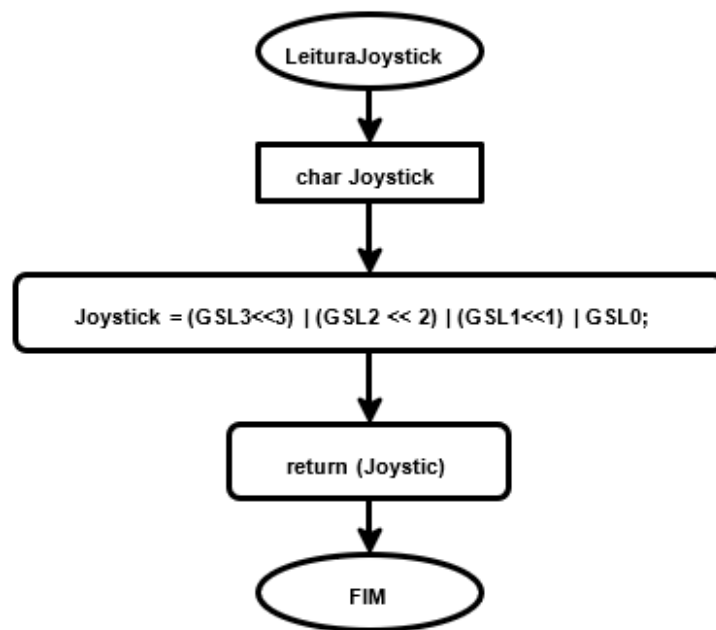
Fonte 11 - Autor (2020)

### 3.5 Fluxogramas

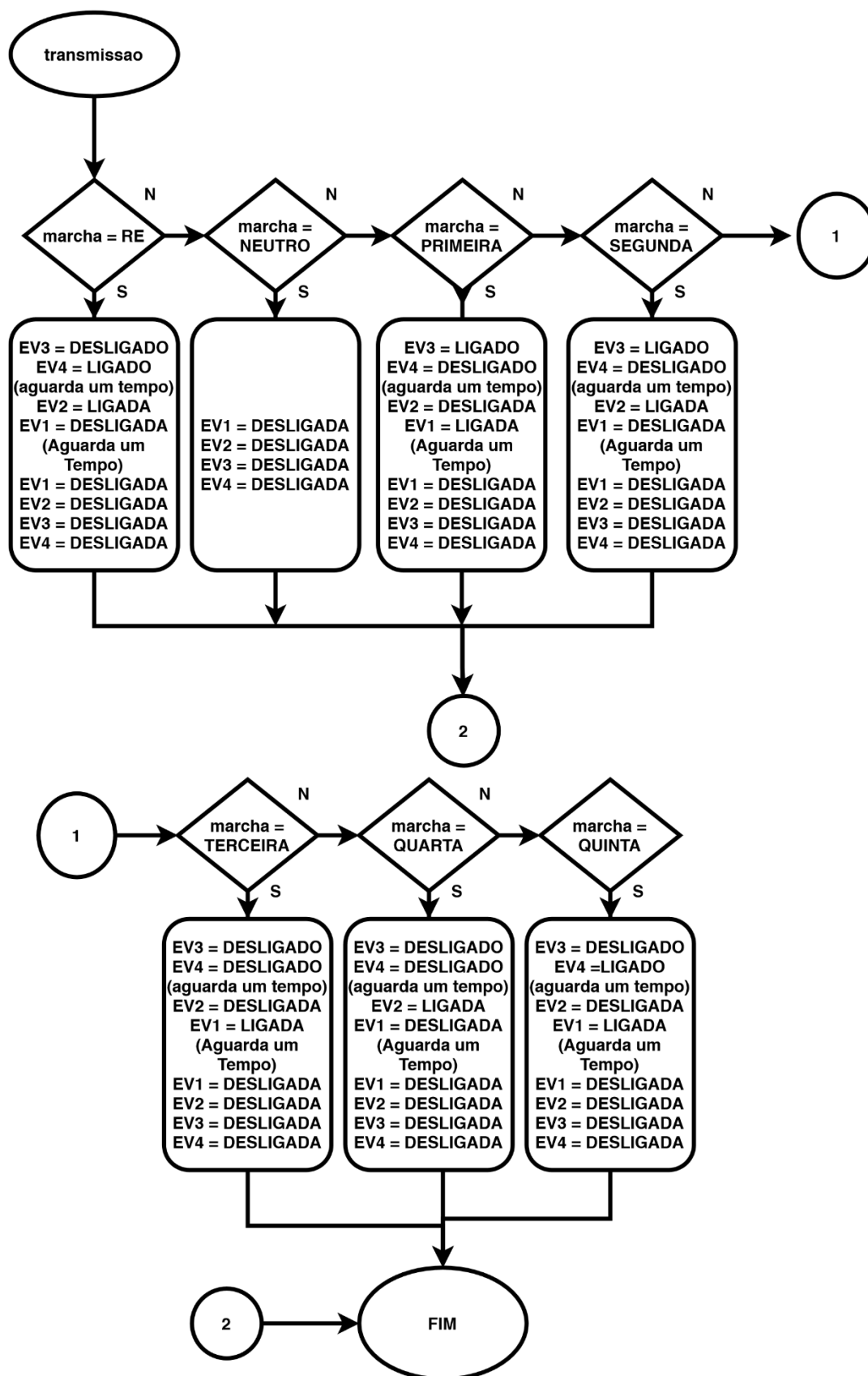
Este tópico mostrara as lógicas utilizadas para desenvolver os programas bases de controle

#### 3.5.1 Controle da bomba hidráulica e leitura do Joystick

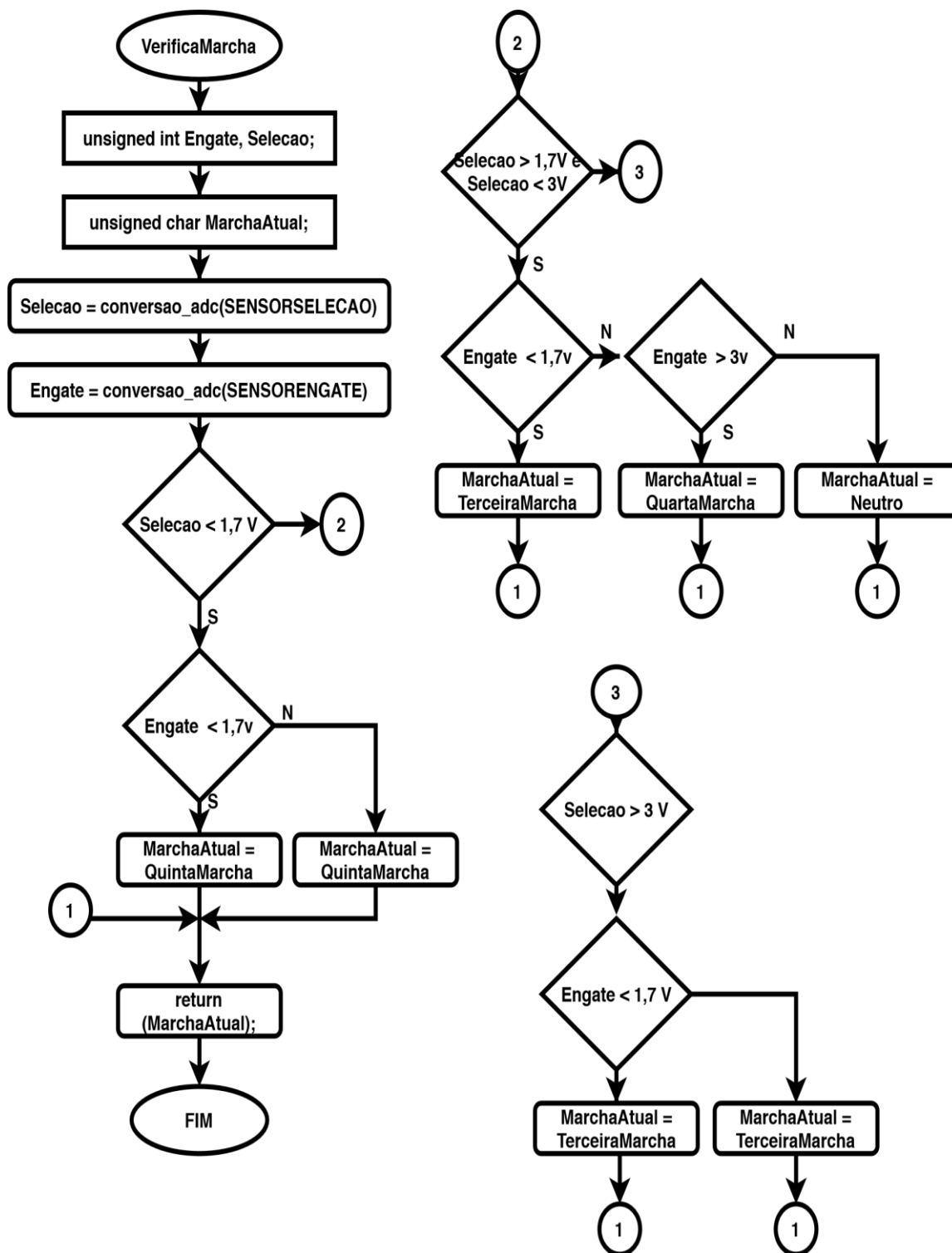




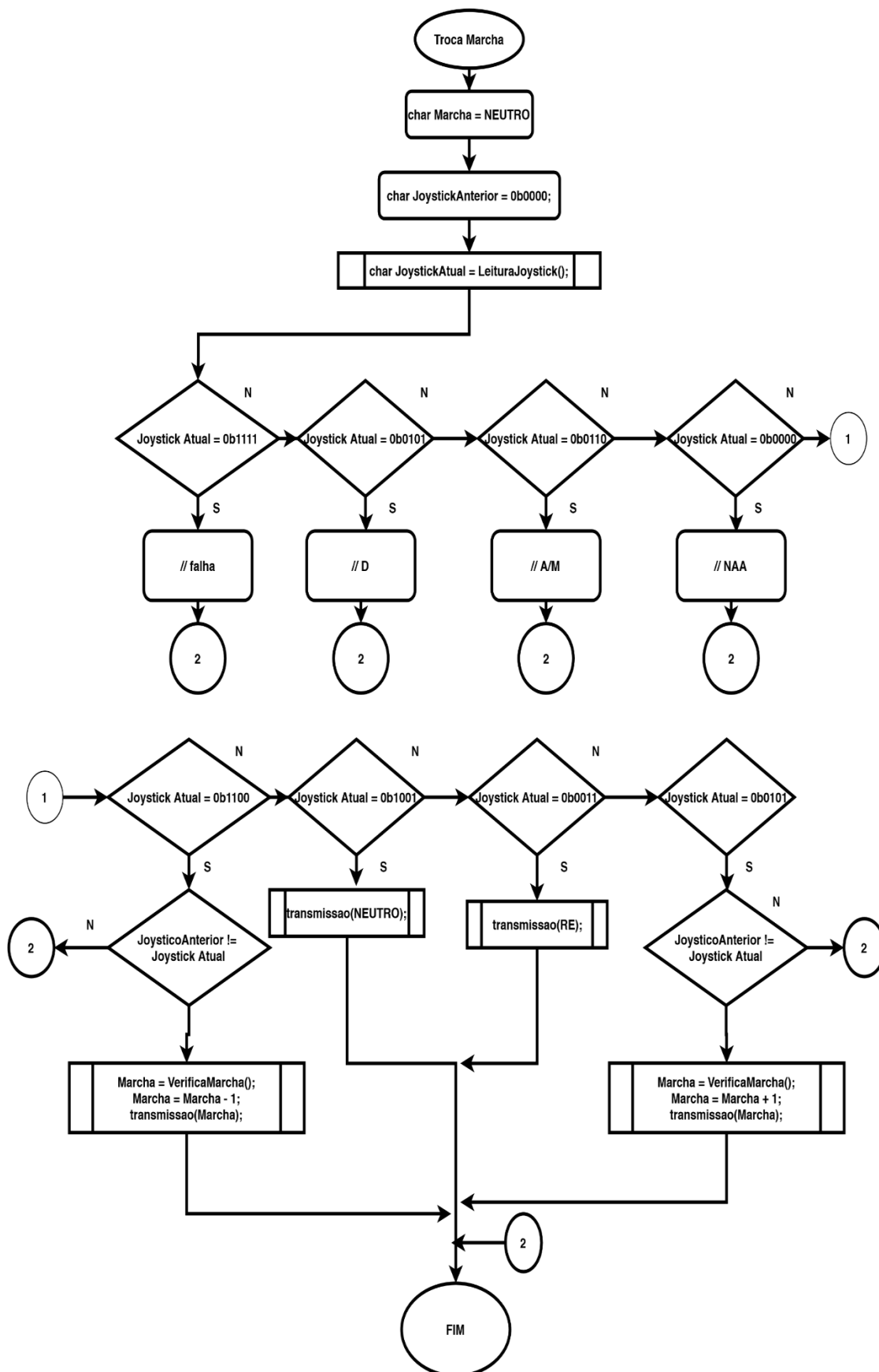
## 3.5.2 Lógica de engate de marcha e verificação







### 3.5.3 Algoritmo de engate

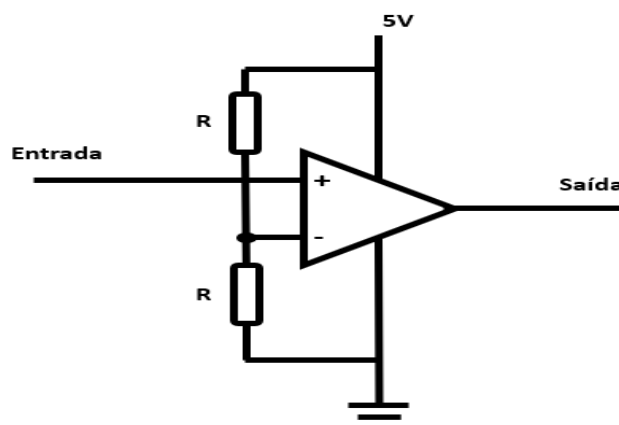


### 3.6 Circuitos de leitura dos sensores

Para a leitura dos sensores é necessário fazer circuitos de adaptação para proteção e melhor interpretação do microcontrolador.

Para a leitura do sensor de posição do *Joystick* é necessário e recomendável realizar um circuito com o amplificador operacional em modo comparador, pois ele se trata de nível lógico *ON/OFF*. A figura 62 demonstra o circuito comparador de tensão de uma maneira genérica, onde na entrada negativa tem-se um divisor de tensão de  $\frac{1}{2}$  VCC pois se a entrada estiver abaixo de 2,5V a saída do amplificador saturará negativamente, ou para o GND, e caso a entrada estiver acima de 2,5V o amplificador saturará positivamente, no caso 5V. As lógicas desses sensores serão interpretadas pelo PIC segundo mostra o Apêndice E, no menu principal do programa.

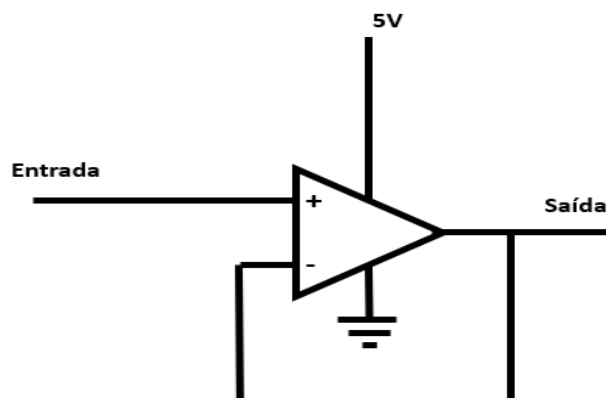
Figura 62 - Amplificador operacional em modo comparador



Fonte 12 - Adaptador. Boni, Djones. Comparadores Shmitt-trigger. Disponível em: <<https://livroseletronica.com/posts/teoria/comparadores-schmitt-trigger/>>. Acesso em 17 de Junho de 2020, 01:54.

Já para os circuitos dos sensores analógicos, sensor de engate, sensor de seleção e sensor de pressão, utilizaremos o circuito seguidor de tensão, ilustrado na figura 63, apenas para termos uma isolação entre as duas partes. Os sinais desses sensores serão lidos através do conversor ADC do PIC programado como o Apêndice D.

Figura 63 - Amplificador operacional em modo seguidor de tensão



Fonte 13 - Fonte 13 -Adaptado. Aprender sobre eletrônicos. Disponível em: <http://www.learningaboutelectronics.com/Artigos/Seguidor-de-tensao.php>. Acesso em: 17 de Junho de 2020, 02:45.

## 4 RESULTADOS OBTIDOS

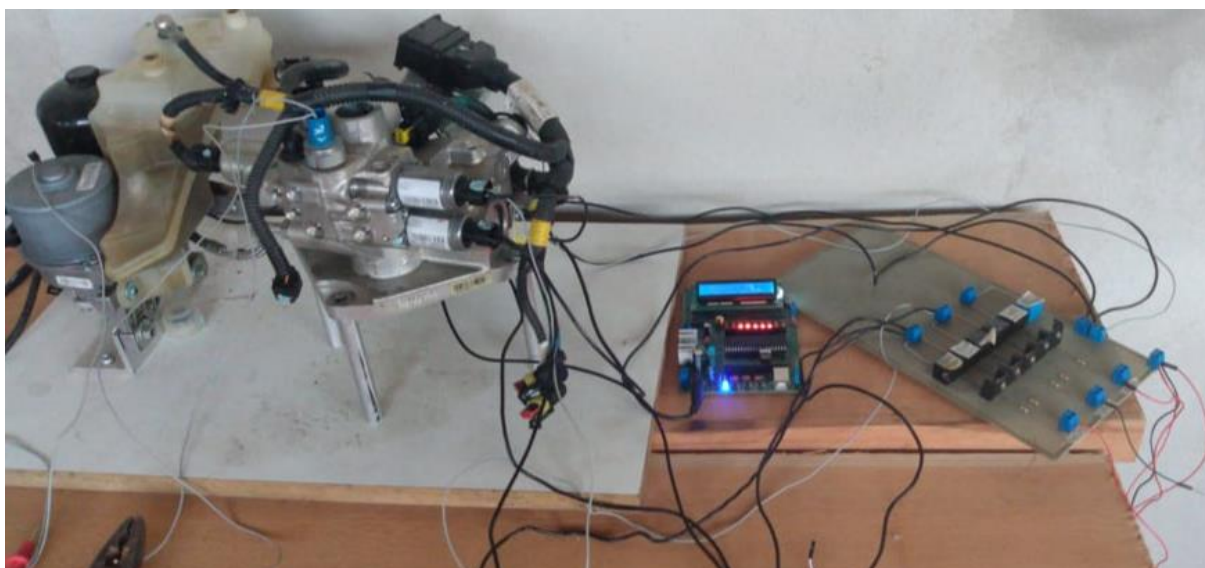
Neste tópico, trataremos de alguns resultados obtidos durante alguns testes realizados. Os testes foram realizados com todos os materiais que possuíamos em nossas residências durante a quarentena realizada na instituição de ensino devido a Pandemia de COVID-19 no ano de desenvolvimento do projeto.

### 4.1 Testes de acionamento das eletroválvulas

Para realizar os testes de bancada com o kit Duallogic® fornecido pela FATEC e a placa de potência com relés desenvolvida inicialmente pelo grupo do Bento, et al.(2017, pag. 99-100), foram necessárias duas fontes de energia contínua que em uma ligação em paralelo resultavam em 30 A de corrente, e esse valor é necessário devido ao alto consumo de corrente da bomba hidráulica do kit e eletroválvulas.

Contudo, com os testes foi possível verificar se a placa de potência anteriormente projetada estava em funcionamento e se as eletroválvulas do kit de bancada estavam acionando, demonstrado na figura 52, com isso identificamos que o sensor de posição do atuador de engate e seleção está avariado não fornecendo a saída com as informações de que marcha está sendo engatada no momento. Porém era possível ouvir as solenoides sendo acionadas quando se havia pressão.

Figura 64- Kit Duallogic® funcionando via Software



Fonte - Autor (2020)

Algumas alterações sugeridas para o kit é reparo do chicote que está em curto e substituição do sensor de posição do atuador de engate para o funcionamento pleno,

pois sem ele não foi possível verificar se todas as eletroválvulas desejadas haviam sido acionadas.

#### **4.2 Teste sensor de pressão**

Com o teste do sensor de pressão foi possível estabelecer uma pressão segura de trabalho para o sistema levando em consideração as especificações de operação do kit, em que as eletroválvulas só poderão ser acionadas havendo um pressão mínima no sistema. Caso exista uma pressão abaixo de 10 Bar as eletroválvulas não entram em funcionamento.

Com essa informação, foi definido durante o software de teste as pressões mínimas e máximas de acionamento e desacionamento da bomba de pressão.

A figura 65 mostra um momento em que o multímetros estava conectado ao sensor de pressão enquanto o software de acionamento e desacionamento era rodado no PIC.

Figura 65: Teste do sensor de pressão

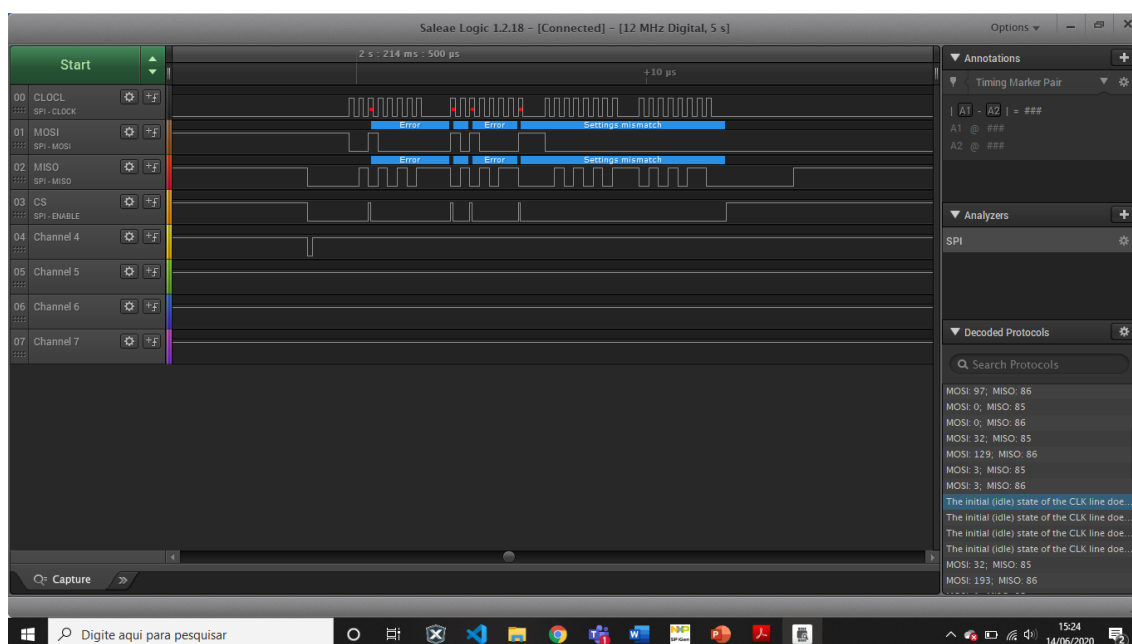


Fonte: Autor (2020)

### 4.3 Comunicação entre as placas

Ao realizarmos o envio de dados pelo PIC ao MC, observamos que os dados realmente são enviados e armazenados, porém em alguns momentos da comunicação há uma falha no *Chip Select* mostrados na figura 66 devido a ligação entre ambas as placas, pois não tínhamos todos os componentes necessários e foi-se colocado os fios de comunicação em paralelo com a placa original.

Figura 66 - Comunicação SPI entre PIC e MC



Fonte 14 - Autor (2020)



## 5 CONSIDERAÇÕES FINAIS

Como conclusão deste trabalho vemos que foi possível alcançar os objetivos propostos no início da pesquisa. Desenvolvemos o software inicial do controle da transmissão Dualogic®, estabelecendo os conceitos básicos para coleta de dados necessários para o controle da transmissão, deixando claro o funcionamento do sistema e subsistemas que promovem o funcionamento da transmissão como um todo. Devido a pandemia existente no período em que o trabalho foi executado houve a necessidade de alterar o cronograma projetado para o desenvolvimento e discorrer do trabalho, mas sucesso as propostas disponíveis foram finalizadas.

Durante o desenvolvimento foi usado o Kit Hidráulico da transmissão Dualogic® e placa de acionamento das eletroválvulas projetada por Bento, et al. como ferramenta para validação do funcionamento do software. Toda a programação foi desenvolvida na Placa ECU FATEC realizando o engate sequencial de todas as marchas, autonomia de acionamento da bomba eletro-hidráulica bem como interpretação das marchas que foram engatas.

A comunicação SPI entre a ECU Fatec e o MC33816 foi estabelecida com sucesso, podendo ser realizada a escrita das configurações dos registradores do MC33816, para este resultado foi usado o Kit de desenvolvimento KL25Z.

Como proposta futura é sugerido realizar a modelagem e controle da embreagem da transmissão, adequar esse novo gerenciamento no software já criado e unificar o software criado junto ao MC33816 com o Kit Dualogic, este que não foi realizado devido a pandemia já citada. Também é sugerido que seja aprofundado o conhecimento do gerenciamento da camada superficial *assembly* do próprio software do Kit NXP, realizar a manipulação e criação de instruções para o gerenciamento da transmissão via comando SPI, criar o gerenciamento da transmissão via comando do PIC com mapas, comunicação CAN com o veículo para uso de dados reais durante o funcionamento do motor. Realizar a análise e adequação de toda compatibilidade eletromagnética para não haver problemas durante o funcionamento dos sistemas.

Este trabalho nos permitiu aprimoramento o conhecimento acadêmico, mostrar as dificuldade e necessidades que a criação de um projeto/sistema tem, e, o impacto que possuir essas competências geram no desenvolvimento de qualquer trabalho.

## 6 BIBLIOGRAFIA

**Bento, Gustavo, Oliveira, Gustavo e Cesário, Willian. 2017.** *ESTUDO E DESENVOLVIMENTO DE UMA UNIDADE DE GERENCIAMENTO ELETRÔNICO QUE SIMULE A ATUAÇÃO NAS ELETROVÁLVULAS DA TRANSMISSÃO AUTOMATIZADA DUALOGIC*. Santo André : s.n., 2017. p. 72.

**Bosch, Robert. 2005.** *Manual de tecnologia automotiva*. São Paulo : Edgard Blucher, 2005.

**Brunetti, Franco. 2012.** *Motores de Combustão interna*. 2012.

**Choleet, M. H. 2002.** *Mecânica de automóveis. Mecânico de Automóveis*. 2002.

**FIAT. 2007.** *Cambio DUALOGIC - Diagnóstico*. 2007.

— . **2007.** *CÂMBIO DUALOGIC - DIAGNÓSTICO*. 2007.

**Florêncio, Rodrigo Kraft, et al. 2013.** *TECNOLOGIA DE SENSORIAMENTO 3D HALL APLICADA EM SISTEMA ELETRÔNICO DE COMANDO AUTOMATIZADO DA TRANSMISSÃO É UMA SOLUÇÃO QUE COLOCA O BRASIL NA VANGUARDA TECNOLÓGICA*. São Paulo : s.n., 2013.

**Francisco, Portal São.** Portal São Francisco. *Portal São Francisco*. [Online] [Citado em: 06 de 11 de 2019.] <https://www.portalsaofrancisco.com.br/mecanica/transmissao-automatica>.

**Garcia, Vitor. 2006.** *SENSOR DE PRESSÃO MICROELETRÔNICO BASEADO NO EFEITO PIEZOMOS*. Campinas - SP : s.n., 2006.

**iCarros. 2016.** iCarros. [Online] 05 de 01 de 2016. [Citado em: 03 de 11 de 2019.]

**Jaicar. 2017.** JaiCar AutoPeças. [Online] 15 de 02 de 2017. [Citado em: 17 de 10 de 2019.]

**Mattera, Liam. 2018.** AutoVídeos. [Online] 2018. [Citado em: 17 de 10 de 2019.] <https://autovideos.com.br/como-funciona-cambio-manual-carro-aprenda-tudo-videos-ilustrados/>.

**Miotto, Guilherme e Silva Filho, Helio. 2018.** *ESTUDO DE CASO REFERENTE AO GANHO DE PERFORMANCE A PARTIR DO SISTEMA BIELA-MANIVELA*. Santo André : s.n., 2018.

**Murta, Márcio. 2012.** FullPower. *Full Power*. [Online] Full Power, 01 de Outubro de 2012. [Citado em: 07 de Março de 2020.] <https://www.revistafullpower.com.br/tecnica-cambios/>.

**Nakata. 2019.** Blog Nakata. [Online] 12 de 08 de 2019. [Citado em: 17 de 10 de 2019.]

**Oficina Santa Luiza. 2013.** Oficina Santa Luiza. *Auto-Peça Mecânica Santa Luiza*. [Online] 04 de Novembro de 2013. [Citado em: 05 de Dezembro de 2019.] <http://oficinasantaluzia.com.br/como-funcionam-as-transmissoes-manuais/>.

**Oliveira, Ricardo de. 2017.** Notícia automotiva. *Notícia automotiva*. [Online] 2017. [Citado em: 17 de 10 de 2019.] <https://www.noticiasautomotivas.com.br/diferenca-entre-cambio-automatico-e-automatizado/>.

**Quieroz, Caio, Melo, Elaine e Calabrez, Felipe Henrique. 2015.** *Revisão dos Sistemas de Transmissão Automotiva*. Santo André : s.n., 2015.

**SOBRENOME, NOME. ANO. TITULO LIVRO. CIDADE : EDITORA, ANO.**

**Varella, Carlos Alberto. 2010.** *Estimativa da potência dos motores*. Rio de Janeiro : s.n., 2010.

**Webmotors S.A. 2009.** Webmotors. *webmotors.com.br*. [Online] 17 de Dezembro de 2009. [Citado em: 24 de Novembro de 2019.] <https://www.webmotors.com.br/wm1/dinheiro-e-economia/guerra-dos-cambios-automaticos-e-automatizados>.

## 7 APÊNCICES

### 7.1 APÊNDICE A

```
#include <xc.h>

#include "Prototipos.h"

#include "Definicoes.h"

#include "Config.h"

#ifndef SPI_C
#define SPI_C

// nome da funcao: config_SPI (void)

//

// Objetivo: Realizar as configurações nos registradores

// necessarios parautilizar o protocolo SPI

//

// Entrada: NULL

// Saida: NULL

void config_SPI(void) {

    ADCON0 = 0x00;

    //configura os pinos de SPI como digital

    ADCON1 = 0b00001100;

    //modo SPI DESABILITADO
```

```
SSPCON1bits.SSPEN = 0;

//FOSC/16 = frequencia de 3 MHz
SSPCON1bits.SSPM0 = 1;
SSPCON1bits.SSPM1 = 0;
SSPCON1bits.SSPM2 = 0;
SSPCON1bits.SSPM3 = 0;
SSPCON1bits.CKP = 1;

//No modo master não é utilizado esse pino
SSPCON1bits.CKP = 0;

//Pega o dado no meio do clock
SSPSTATbits.SMP = 0;

//transmissao de dados de alto para baixo
SSPSTATbits.CKE = 0;

//Flag zerada
SSPIF = 0;

//modo SPI Habilitado
SSPCON1bits.SSPEN = 1;
}

// nome da funcao: escrita_SPI (int endereco, int dado)
//
// Objetivo: Realizar a escrita de um dado desejado
// em um determinado endereço referente
//
```

```
// Entrada: int endereco, int dado - endereço de escrita e dado desejado
```

```
// Saida: NULL
```

```
void escrita_SPI (int endereco, int dado) {  
    //Coloca CS para 0, habilita comunicação  
    CHIP_SELECT = 0;  
    //ajusta o endereço com o comando de leitura - parte alta  
    SSPBUF = ((endereco >> 3) & 0X003F);  
    __delay_us (3);  
    //ajusta o endereço com o comando de leitura - parte baixa  
    SSPBUF = (((endereco << 5) & 0x00FF) | 0x0001);  
    __delay_us (3);  
    // envia a parte alta do dado  
    SSPBUF = ((dado >> 8)& 0x00FF);  
    __delay_us(3);  
    // envia a parte baixa do dado  
    SSPBUF = (dado & 0xFF);  
    __delay_us (3);  
    //Coloca CS para 1, desabilita comunicação  
    CHIP_SELECT = 1;  
}
```

```
//nome da funcao: InitEscritaSPI(void)
```

```
//
```

```
// Objetivo: Envia comandos iniciais para o MC33816
```

```
//
```

```
// Entrada: NULL

// Saida: NULL

void InitEscritaSPI(void) {

    //Coloca CS para 0, habilita comunicação
    CHIP_SELECT = 0;

    // envia primeira parte do comando
    SSPBUF = 127;
    __delay_us (3);

    // envia segunda parte do comando
    SSPBUF = 255;
    __delay_us (3);

    // envia terceira parte do comando
    SSPBUF = 0;
    __delay_us (3);

    // envia quarta parte do comando
    SSPBUF = 4;
    __delay_us (12);

    //Coloca CS para 1, desabilita comunicação
    CHIP_SELECT = 1;
    __delay_ms (10);
}

// nome da funcao: FinalizaEscritaSPI(void)

// Objetivo: Envia comandos finais para a comunicação com o MC33816

//
```

```
// Entrada: NULL

// Saida: NULL

void FinalizaEscritaSPI(void) {

    //Coloca CS para 0, habilita comunicação
    CHIP_SELECT = 0;

    // envia primeira parte do comando
    SSPBUF = 32;
    __delay_us (3);

    // envia segunda parte do comando
    SSPBUF = 1;
    __delay_us (3);

    // envia terceira parte do comando
    SSPBUF = 0;
    __delay_us (3);

    // envia quarta parte do comando
    SSPBUF = 24;
    __delay_us (20);

    //Coloca CS para 1, desabilita comunicação
    CHIP_SELECT = 1;
    __delay_ms (20);

    //Coloca CS para 0, habilita comunicação
    CHIP_SELECT = 0;

    // envia quinta parte do comando
    SSPBUF = 36;
    __delay_us (3);
```



```
// envia sexta parte do comando
SSPBUF = 1;
__delay_us (3);
// envia setima parte do comando
SSPBUF = 0;
__delay_us (3);
// envia oitava parte do comando
SSPBUF = 24;
__delay_us (20);
//Coloca CS para 1, desabilita comunicação
CHIP_SELECT = 1;
__delay_ms (20);
}
#endif
```

## 7.2 APÊNDICE B

```
#ifndef DEFINICOES_H

#define    DEFINICOES_H

/*Definições dos esdereços do Ch1 Config Registers*/

#define END_FLASH_ENABLE_CH1          0x100
#define END_CTRL_REG_UC0_CH1         0x101
#define END_CTRL_REG_UC1_CH1         0x102
#define END_UNLOCK_WORD_CH1          0x103
#define END_START_CONFIG_REG_CH1     0X104
#define END_STATUS_REG_UC0_CH1       0x105
#define END_STATUS_REG_UC1_CH1       0x106
#define END_CODE_WIDTH_CH1           0x107
#define END_CHECSUM_H_CH1             0x108
#define END_CHECSUM_L_CH1            0x109
#define END_UC0_ENTRY_POINT_CH1      0x10A
#define END_UC1_ENTRY_POINT_CH1      0x10B
#define END_DIAG_ROUTINE_ADDR_CH1    0x10C
#define END_DRIVER_DISABLED_ROUTINE_ADDR_CH1 0x10D
#define END_SW_INTERRUPT_ROUTINE_ADDR_CH1 0x10E
#define END_UC0_IRQ_STATUS_CH1       0x10F
#define END_UC1_IRQ_STATUS_CH1       0x110
#define END_COUNTER_34_PRESCALER_CH1 0x111
#define END_CONTROL_REGISTER_SPLIT_CH1 0X112
```

/\*Definições dos endereços do Ch2 Config Registers\*/

```
#define END_FLASH_ENABLE_CH2          0x120
#define END_CTRL_REG_UC0_CH2          0x121
#define END_CTRL_REG_UC1_CH2          0x122
#define END_UNLOCK_WORD_CH2           0x123
#define END_START_CONFIG_REG_CH2      0x124
#define END_STATUS_REG_UC0_CH2        0x125
#define END_STATUS_REG_UC1_CH2        0x126
#define END_CODE_WIDTH_CH2            0x127
#define END_CHECSUM_H_CH2              0x128
#define END_CHECSUM_L_CH2              0x129
#define END_UC0_ENTRY_POINT_CH2        0x12A
#define END_UC1_ENTRY_POINT_CH2        0x12B
#define END_DIAG_ROUTINE_ADDR_CH2      0x12C
#define END_DRIVER_DISABLED_ROUTINE_ADDR_CH2  0x12D
#define END_SW_INTERRUPT_ROUTINE_ADDR_CH2  0x12E
#define END_UC0_IRQ_STATUS_CH2         0x12F
#define END_UC1_IRQ_STATUS_CH2         0x130
#define END_COUNTER_34_PRESCALER_CH2   0x131
#define END_CONTROL_REGISTER_SPLIT_CH2 0x132
```

/\*Definições dos endereços do Main Config Registers\*/

```
#define END_CLOCK_PRESCALER            0x1C0
#define END_FLAGS_DIRECTION            0x1C1
```

```

#define END_FLAGS_POLARITY          0x1C2
#define END_FLAGS_SOURCE            0x1C3
#define END_OFFSET_COMPENSATION_PRESCALER  0x1C4
#define END_DRIVER_CONFIG           0x1C5
#define END_PLL_CONFIG              0x1C6
#define END_BACKUP_CLOCK_STATUS      0x1C7
#define END_SPI_CONFIG              0x1C8
#define END_TRACE_START              0x1CA
#define END_TRACE_STOP               0x1CB
#define END_TRACE_CONFIG             0x1CC
#define END_DEVICE_LOCK              0x1CD
#define END_RESET_BEHAVIOUR         0x1CE
#define END_DEVICES_UNLOCK           0x1CF
#define END_GLOBAL_RESET_CODE_PART1    0x1D0
#define END_GLOBAL_RESET_CODE_PART2    0x1D1
#define END_DRIVER_STATUS            0x1D2
#define END_SPI_ERROR                0x1D3
#define END_INTERRUPT_REGISTER        0x1D4
#define END_IDENTIFIER_REVISION       0x1D5
#define END_RESET_SOURCE             0x1D6
#define END_BIST_INTERFACE           0x1DC

/*Definições dos esdereços do IO Config Registers*/

#define END_FBK_SENS_UC0_CH1          0x180
#define END_FBK_SENS_UC1_CH1          0x181

```

```

#define END_FBK_SENS_UC0_CH2          0x182
#define END_FBK_SENS_UC1_CH2          0x183
#define END_OUT_ACC_UC0_CH1           0x184
#define END_OUT_ACC_UC1_CH1           0x185
#define END_OUT_ACC_UC0_CH2           0x186
#define END_OUT_ACC_UC1_CH2           0x187
#define END_CUR_BLOCK_ACCESS_PART1    0x188
#define END_CUR_BLOCK_ACCESS_PART2    0x189
#define END_VDS_THRESHOLDS_HS         0x18A
#define END_VSRC_THRESHOLDS_HS        0x18B
#define END_VDS_THRESHOLDS_LS_PART1   0x18C
#define END_VDS_THRESHOLDS_LS_PART2   0x18D
#define END_HS_SLEWRATE                0x18E
#define END_LS_SLEWRATE                0x18F
#define END_OFFSET_COMPENSATION1       0x190
#define END_OFFSET_COMPENSATION2       0x191
#define END_OFFSET_COMPENSATION3       0x192
#define END_OFFSET_COMPENSATION4       0x193
#define END_ADC1_RESULT                0x194
#define END_ADC2_RESULT                0x195
#define END_ADC3_RESULT                0x196
#define END_ADC4_RESULT                0x197
#define END_CURRENT_FILTER12           0x198
#define END_CURRENT_FILTER34L          0x199
#define END_CURRENT_FILTER4H4NEG       0x19A
#define END_BOOST_DAC                   0x19B

```

```
#define END_BOOST_DAC_ACCESS          0x19C
#define END_BOST_FILTER               0x19D
#define END_DAC1_VALUE                0x19E
#define END_DAC2_VALUE                0x19F
#define END_DAC3_VALUE                0x1A0
#define END_DAC4L_VALUE               0x1A1
#define END_DAC4H_VALUE               0x1A2
#define END_DAC4NEG_VALUE              0x1A3
#define END_BIAS_CONFIG               0x1A4
#define END_BOOTSTRAP_CHARGED         0x1A5
#define END_HS12_LS_ACT                0x1A6
#define END_HS34_LS_ACT                0x1A7
#define END_HS5_LS_ACT                0x1A8
#define END_DAC_SETTLING_TIME         0x1A9
#define END_OA_OUT1_CONFIG            0x1AA
#define END_OA_OUT2_CONFIG            0x1AB
#define END_MEAS_FUNC1                0x1AC
#define END_MEAS_FUNC2                0x1AD
#define END_VDS_THRESHOLD_HS_PART1    0x1AE
#define END_VDS_THRESHOLD_HS_PART2    0x1AF
#define END_VSRC_THRESHOLD_HS_PART1    0x1B0
#define END_VSRC_THRESHOLD_HS_PART2    0x1B1
#define END_VDS_THRESHOLD_LS_PART1    0x1B2
#define END_VDS_THRESHOLD_LS_PART2    0x1B3
#define END_BATT_RESULT                0x1B4
```

/\*Definições dos endereços do Diagnostic Config Registers\*/

```
#define END_LS1_DIAG_CONFIG1      0x140
#define END_LS1_DIAG_CONFIG2      0x141
#define END_LS1_OUTPUT_CONFIG     0x142
#define END_LS2_DIAG_CONFIG1      0x143
#define END_LS2_DIAG_CONFIG2      0x144
#define END_LS2_OUTPUT_CONFIG     0x145
#define END_LS3_DIAG_CONFIG1      0x146
#define END_LS3_DIAG_CONFIG2      0x147
#define END_LS3_OUTPUT_CONFIG     0x148
#define END_LS4_DIAG_CONFIG1      0x149
#define END_LS4_DIAG_CONFIG2      0x14A
#define END_LS4_OUTPUT_CONFIG     0x14B
#define END_LS5_DIAG_CONFIG1      0x14C
#define END_LS5_DIAG_CONFIG2      0x14D
#define END_LS5_OUTPUT_CONFIG     0x14E
#define END_LS6_DIAG_CONFIG1      0x14F
#define END_LS6_DIAG_CONFIG2      0x150
#define END_LS6_OUTPUT_CONFIG     0x151
#define END_LS7_OUTPUT_CONFIG     0x152
#define END_HS1_DIAG_CONFIG1      0x153
#define END_HS1_DIAG_CONFIG2      0x154
#define END_HS1_OUTPUT_CONFIG     0x155
#define END_HS2_DIAG_CONFIG1      0x156
#define END_HS2_DIAG_CONFIG2      0x157
```

```
#define END_HS2_OUTPUT_CONFIG          0x158
#define END_HS3_DIAG_CONFIG1          0x159
#define END_HS3_DIAG_CONFIG2          0x15A
#define END_HS3_OUTPUT_CONFIG         0x15B
#define END_HS4_DIAG_CONFIG1          0x15C
#define END_HS4_DIAG_CONFIG2          0x15D
#define END_HS4_OUTPUT_CONFIG         0x15E
#define END_HS5_DIAG_CONFIG1          0x15F
#define END_HS5_DIAG_CONFIG2          0x160
#define END_HS5_OUTPUT_CONFIG         0x161
#define END_ERR_UC0CH1_PART1          0x162
#define END_ERR_UC0CH1_PART2          0x163
#define END_ERR_UC1CH1_PART1          0x164
#define END_ERR_UC1CH1_PART2          0x165
#define END_ERR_UC0CH2_PART1          0x166
#define END_ERR_UC0CH2_PART2          0x167
#define END_ERR_UC1CH2_PART1          0x168
#define END_ERR_UC1CH2_PART2          0x169
#define END_FW_EXTERNAL_REQUEST        0x16A
#define END_DIAGNOSTICS_OPTION         0x16B

// Canais analogicos dos sensores

#define SENSORPRESSAO 0
#define SENSORENGATE 1
#define SENSORSELECAO 2
```



```
// Marchas para a funcao transmissao
```

```
#define RE 0
```

```
#define NEUTRO 1
```

```
#define PRIMEIRA_MARCHA 2
```

```
#define SEGUNDA_MARCHA 3
```

```
#define TERCEIRA_MARCHA 4
```

```
#define QUARTA_MARCHA 5
```

```
#define QUINTA_MARCHA 6
```

```
//definições gerais
```

```
#define LIGADO 1
```

```
#define DESLIGADO 0
```

```
// definicoes dos sensores e atuadores
```

```
#define GSL0 PORTAbits.RA3
```

```
#define GSL1 PORTAbits.RA4
```

```
#define GSL2 PORTAbits.RA5
```

```
#define GSL3 PORTEbits.RE1
```

```
#define EV1 PORTBbits.RB7
```

```
#define EV2 PORTBbits.RB6
```

```
#define EV3 PORTBbits.RB5
```

```
#define EV4    PORTBbits.RB4
```

```
#define BOMBAPRESSAO PORTBbits.RB2
```

```
#define CHIP_SELECT    LATBbits.LATB3
```

```
#endif
```

### 7.3 APÊNDICE C

```
#ifndef _CONFIGURAMC33816_H
#define _CONFIGURAMC33816_H

#include "Definicoes.h"
#include "Prototipos.h"
#include "Config.h"

// nome da funcao: Config_Registrador_Ch1(void)
// Objetivo: Configurar os registradores do CH1
//
// Entrada: NULL
// Saida: NULL
void Config_Registrador_Ch1(void)
{
    InitEscritaSPI();
    escrita_SPI(END_FLASH_ENABLE_CH1,0x0018);
    __delay_ms(7);
    escrita_SPI(END_CTRL_REG_UC0_CH1,0x0000);
    __delay_ms(7);
    escrita_SPI(END_CTRL_REG_UC1_CH1,0x0000);
    __delay_ms(7);
    escrita_SPI(END_UNLOCK_WORD_CH1,0x0000);
    __delay_ms(7);
    escrita_SPI(END_START_CONFIG_REG_CH1,0x0303);
```

```
__delay_ms(7);
escrita_SPI(END_STATUS_REG_UC0_CH1,0x0000);
__delay_ms(7);
escrita_SPI(END_STATUS_REG_UC1_CH1,0x0000);
__delay_ms(7);
escrita_SPI(END_CODE_WIDTH_CH1,0x0061);
__delay_ms(7);
escrita_SPI(END_CHECSUM_H_CH1,0x58CA);
__delay_ms(7);
escrita_SPI(END_CHECSUM_L_CH1,0xBD6B);
__delay_ms(7);
escrita_SPI(END_UC0_ENTRY_POINT_CH1,0x0000);
__delay_ms(7);
escrita_SPI(END_UC1_ENTRY_POINT_CH1,0x0031);
__delay_ms(7);
escrita_SPI(END_DIAG_ROUTINE_ADDR_CH1,0x0000);
__delay_ms(7);
escrita_SPI(END_DRIVER_DISABLED_ROUTINE_ADDR_CH1,0x0000);
__delay_ms(7);
escrita_SPI(END_SW_INTERRUPT_ROUTINE_ADDR_CH1,0x0000);
__delay_ms(7);
escrita_SPI(END_UC0_IRQ_STATUS_CH1,0x3FF);
__delay_ms(7);
escrita_SPI(END_UC1_IRQ_STATUS_CH1,0x3F);
__delay_ms(7);
escrita_SPI(END_COUNTER_34_PRESCALER_CH1,0x0000);
```

```
    __delay_ms(7);  
    escrita_SPI(END_CONTROL_REGISTER_SPLIT_CH1,0x0000);  
    __delay_ms (100);  
}
```

```
// nome da funcao: Config_Registrador_Ch2(void)
```

```
// Objetivo: Configurar os registradores do CH2
```

```
//
```

```
// Entrada: NULL
```

```
// Saida: NULL
```

```
void Config_Registrador_Ch2(void)
```

```
{  
    InitEscritaSPI();  
    escrita_SPI(END_FLASH_ENABLE_CH2,0x0018);  
    __delay_ms(7);  
    escrita_SPI(END_CTRL_REG_UC0_CH2,0x0000);  
    __delay_ms(7);  
    escrita_SPI(END_CTRL_REG_UC1_CH2,0x0020);  
    __delay_ms(7);  
    escrita_SPI(END_UNLOCK_WORD_CH2,0x0000);  
    __delay_ms(7);  
    escrita_SPI(END_START_CONFIG_REG_CH2,0x0C00);  
    __delay_ms(7);  
    escrita_SPI(END_STATUS_REG_UC0_CH2,0x0000);  
    __delay_ms(7);  
}
```

```
escrita_SPI(END_STATUS_REG_UC1_CH2,0x0000);
__delay_ms(7);
escrita_SPI(END_CODE_WIDTH_CH2,0x002A);
__delay_ms(7);
escrita_SPI(END_CHECSUM_H_CH2,0x5906);
__delay_ms(7);
escrita_SPI(END_CHECSUM_L_CH2,0x81FD);
__delay_ms(7);
escrita_SPI(END_UC0_ENTRY_POINT_CH2,0x0000);
__delay_ms(7);
escrita_SPI(END_UC1_ENTRY_POINT_CH2,0x0010);
__delay_ms(7);
escrita_SPI(END_DIAG_ROUTINE_ADDR_CH2,0x0000);
__delay_ms(7);
escrita_SPI(END_DRIVER_DISABLED_ROUTINE_ADDR_CH2,0x0000);
__delay_ms(7);
escrita_SPI(END_SW_INTERRUPT_ROUTINE_ADDR_CH2,0x0000);
__delay_ms(7);
escrita_SPI(END_UC0_IRQ_STATUS_CH2,0x03FF);
__delay_ms(7);
escrita_SPI(END_UC1_IRQ_STATUS_CH2,0x03FF);
__delay_ms(7);
escrita_SPI(END_COUNTER_34_PRESCALER_CH2,0x0000);
__delay_ms(7);
escrita_SPI(END_CONTROL_REGISTER_SPLIT_CH2,0x0000);
__delay_ms(100);
```

```
}

// nome da funcao: Config_Main_Register(void)
// Objetivo: Configurar os registradores do Main
//
// Entrada: NULL
// Saida: NULL

void Config_Main_Register(void)
{
    InitEscritaSPI();
    escrita_SPI(END_CLOCK_PRESCALER,0x0003);
    __delay_ms(7);
    escrita_SPI(END_FLAGS_DIRECTION,0x1FC0);
    __delay_ms(7);
    escrita_SPI(END_FLAGS_POLARITY,0x0000);
    __delay_ms(7);
    escrita_SPI(END_FLAGS_SOURCE,0x1000);
    __delay_ms(7);
    escrita_SPI(END_OFFSET_COMPENSATION_PRESCALER,0x0017);
    __delay_ms(7);
    escrita_SPI(END_DRIVER_CONFIG,0x2000);
    __delay_ms(7);
    escrita_SPI(END_PLL_CONFIG,0x0001);
    __delay_ms(7);
    escrita_SPI(END_BACKUP_CLOCK_STATUS,0x0000);
```

```
__delay_ms(7);
escrita_SPI(END_SPI_CONFIG,0x001F);
__delay_ms(7);
escrita_SPI(END_TRACE_START,0x0000);
__delay_ms(7);
escrita_SPI(END_TRACE_STOP,0x0000);
__delay_ms(7);
escrita_SPI(END_TRACE_CONFIG,0x0000);
__delay_ms(7);
escrita_SPI(END_DEVICE_LOCK,0x0000);
__delay_ms(7);
escrita_SPI(END_RESET_BEHAVIOUR,0x0000);
__delay_ms(7);
escrita_SPI(END_DEVICES_UNLOCK,0x0000);
__delay_ms(7);
escrita_SPI(END_GLOBAL_RESET_CODE_PART1,0x0000);
__delay_ms(7);
escrita_SPI(END_GLOBAL_RESET_CODE_PART2,0x0000);
__delay_ms(7);
escrita_SPI(END_DRIVER_STATUS,0x0000);
__delay_ms(7);
escrita_SPI(END_SPI_ERROR,0x0000);
__delay_ms(7);
escrita_SPI(END_INTERRUPT_REGISTER,0x0000);
__delay_ms(7);
escrita_SPI(END_IDENTIFIER_REVISION,0x9C00);
```



```
    __delay_ms(7);
    escrita_SPI(END_RESET_SOURCE,0x0000);
    __delay_ms(7);
    escrita_SPI(END_BIST_INTERFACE,0x0000);
    __delay_ms(100);
}

// nome da funcao: Config_IO_Register(void)
// Objetivo: Configurar os registradores de I/O
//
// Entrada: NULL
// Saida: NULL

void Config_IO_Register(void)
{
    InitEscritaSPI();
    escrita_SPI(END_FBK_SENS_UC0_CH1,0x0000);
    __delay_ms(7);
    escrita_SPI(END_FBK_SENS_UC1_CH1,0x0000);
    __delay_ms(7);
    escrita_SPI(END_FBK_SENS_UC0_CH2,0x0000);
    __delay_ms(7);
    escrita_SPI(END_FBK_SENS_UC1_CH2,0x0000);
    __delay_ms(7);
    escrita_SPI(END_OUT_ACC_UC0_CH1,0x0063);
    __delay_ms(7);
}
```

```
escrita_SPI(END_OUT_ACC_UC1_CH1,0x018C);
__delay_ms(7);
escrita_SPI(END_OUT_ACC_UC0_CH2,0x0800);
__delay_ms(7);
escrita_SPI(END_OUT_ACC_UC1_CH2,0x0610);
__delay_ms(7);
escrita_SPI(END_CUR_BLOCK_ACCESS_PART1,0x0041);
__delay_ms(7);
escrita_SPI(END_CUR_BLOCK_ACCESS_PART2,0x0098);
__delay_ms(7);
escrita_SPI(END_VDS_THRESHOLDS_HS,0x0000);
__delay_ms(7);
escrita_SPI(END_VSRC_THRESHOLDS_HS,0x0000);
__delay_ms(7);
escrita_SPI(END_VDS_THRESHOLDS_LS_PART1,0x0000);
__delay_ms(7);
escrita_SPI(END_VDS_THRESHOLDS_LS_PART2,0x0000);
__delay_ms(7);
escrita_SPI(END_HS_SLEWRATE,0x0000);
__delay_ms(7);
escrita_SPI(END_LS_SLEWRATE,0x0000);
__delay_ms(7);
escrita_SPI(END_OFFSET_COMPENSATION1,0x0000);
__delay_ms(7);
escrita_SPI(END_OFFSET_COMPENSATION2,0x0000);
__delay_ms(7);
```

```
escrita_SPI(END_OFFSET_COMPENSATION3,0x0012);
__delay_ms(7);
escrita_SPI(END_OFFSET_COMPENSATION4,0x0012);
__delay_ms(7);
escrita_SPI(END_ADC1_RESULT,0x0012);
__delay_ms(7);
escrita_SPI(END_ADC2_RESULT,0x0000);
__delay_ms(7);
escrita_SPI(END_ADC3_RESULT,0x0015);
__delay_ms(7);
escrita_SPI(END_ADC4_RESULT,0x0015);
__delay_ms(7);
escrita_SPI(END_CURRENT_FILTER12,0x0000);
__delay_ms(7);
escrita_SPI(END_CURRENT_FILTER34L,0x0000);
__delay_ms(7);
escrita_SPI(END_CURRENT_FILTER4H4NEG,0x0000);
__delay_ms(7);
escrita_SPI(END_BOOST_DAC,0x0000);
__delay_ms(7);
escrita_SPI(END_BOOST_DAC_ACCESS,0x0000);
__delay_ms(7);
escrita_SPI(END_BOST_FILTER,0x0080);
__delay_ms(7);
escrita_SPI(END_DAC1_VALUE,0x0080);
__delay_ms(7);
```

```
escrita_SPI(END_DAC2_VALUE,0x0080);
__delay_ms(7);
escrita_SPI(END_DAC3_VALUE,0x0001);
__delay_ms(7);
escrita_SPI(END_DAC4L_VALUE,0x0081);
__delay_ms(7);
escrita_SPI(END_DAC4H_VALUE,0x0041);
__delay_ms(7);
escrita_SPI(END_DAC4NEG_VALUE,0x0008);
__delay_ms(7);
escrita_SPI(END_BIAS_CONFIG,0x0004);
__delay_ms(7);
escrita_SPI(END_BOOTSTRAP_CHARGED,0x0001);
__delay_ms(7);
escrita_SPI(END_HS12_LS_ACT,0x008C);
__delay_ms(7);
escrita_SPI(END_HS34_LS_ACT,0x0000);
__delay_ms(7);
escrita_SPI(END_HS5_LS_ACT,0x0020);
__delay_ms(7);
escrita_SPI(END_DAC_SETTLING_TIME,0x0020);
__delay_ms(7);
escrita_SPI(END_OA_OUT1_CONFIG,0x002E);
__delay_ms(7);
escrita_SPI(END_OA_OUT2_CONFIG,0x0000);
__delay_ms(7);
```

```
escrita_SPI(END_MEAS_FUNC1,0x0000);
__delay_ms(7);
escrita_SPI(END_MEAS_FUNC2,0x0000);
__delay_ms(7);
escrita_SPI(END_VDS_THRESHOLD_HS_PART1,0x0000);
__delay_ms(7);
escrita_SPI(END_VDS_THRESHOLD_HS_PART2,0x0000);
__delay_ms(7);
escrita_SPI(END_VSRC_THRESHOLD_HS_PART1,0x0000);
__delay_ms(7);
escrita_SPI(END_VSRC_THRESHOLD_HS_PART2,0x0000);
__delay_ms(7);
escrita_SPI(END_VDS_THRESHOLD_LS_PART1,0x0000);
__delay_ms(7);
escrita_SPI(END_VDS_THRESHOLD_LS_PART2,0x0000);
__delay_ms(7);
escrita_SPI(END_BATT_RESULT,0x0000);
__delay_ms (100);
}

// nome da funcao: Config_Diagnostic_Register(void)
// Objetivo: Configurar os registradores de diagnostico
//
// Entrada: NULL
// Saida: NULL
```

```
void Config_Diagnostic_Register(void)
{
    InitEscritaSPI ();
    escrita_SPI(END_LS1_DIAG_CONFIG1,0x1DDA);
    __delay_ms (7);
    escrita_SPI(END_LS1_DIAG_CONFIG2,0x0009);
    __delay_ms(7);
    escrita_SPI(END_LS1_OUTPUT_CONFIG,0x001E);
    __delay_ms(7);
    escrita_SPI(END_LS2_DIAG_CONFIG1,0x1DDA);
    __delay_ms(7);
    escrita_SPI(END_LS2_DIAG_CONFIG2,0x0009);
    __delay_ms(7);
    escrita_SPI(END_LS2_OUTPUT_CONFIG,0x001E);
    __delay_ms(7);
    escrita_SPI(END_LS3_DIAG_CONFIG1,0x028E);
    __delay_ms(7);
    escrita_SPI(END_LS3_DIAG_CONFIG2,0x0009);
    __delay_ms(7);
    escrita_SPI(END_LS3_OUTPUT_CONFIG,0x001E);
    __delay_ms(7);
    escrita_SPI(END_LS4_DIAG_CONFIG1,0x028E);
    __delay_ms(7);
    escrita_SPI(END_LS4_DIAG_CONFIG2,0x0009);
    __delay_ms(7);
    escrita_SPI(END_LS4_OUTPUT_CONFIG,0x001E);
```

```
__delay_ms(7);
escrita_SPI(END_LS5_DIAG_CONFIG1,0x028E);
__delay_ms(7);
escrita_SPI(END_LS5_DIAG_CONFIG2,0x0009);
__delay_ms(7);
escrita_SPI(END_LS5_OUTPUT_CONFIG,0x001E);
__delay_ms(7);
escrita_SPI(END_LS6_DIAG_CONFIG1,0x028E);
__delay_ms(7);
escrita_SPI(END_LS6_DIAG_CONFIG2,0x0009);
__delay_ms(7);
escrita_SPI(END_LS6_OUTPUT_CONFIG,0x001E);
__delay_ms(7);
escrita_SPI(END_LS7_OUTPUT_CONFIG,0x001E);
__delay_ms(7);
escrita_SPI(END_HS1_DIAG_CONFIG1,0x1DDA);
__delay_ms(7);
escrita_SPI(END_HS1_DIAG_CONFIG2,0x0069);
__delay_ms(7);
escrita_SPI(END_HS1_OUTPUT_CONFIG,0x001E);
__delay_ms(7);
escrita_SPI(END_HS2_DIAG_CONFIG1,0x1DDA);
__delay_ms(7);
escrita_SPI(END_HS2_DIAG_CONFIG2,0x0069);
__delay_ms(7);
escrita_SPI(END_HS2_OUTPUT_CONFIG,0x001E);
```

```
__delay_ms(7);
escrita_SPI(END_HS3_DIAG_CONFIG1,0x028E);
__delay_ms(7);
escrita_SPI(END_HS3_DIAG_CONFIG2,0x0069);
__delay_ms(7);
escrita_SPI(END_HS3_OUTPUT_CONFIG,0x001E);
__delay_ms(7);
escrita_SPI(END_HS4_DIAG_CONFIG1,0x028E);
__delay_ms(7);
escrita_SPI(END_HS4_DIAG_CONFIG2,0x0069);
__delay_ms(7);
escrita_SPI(END_HS4_OUTPUT_CONFIG,0x001E);
__delay_ms(7);
escrita_SPI(END_HS5_DIAG_CONFIG1,0x0000);
__delay_ms(7);
escrita_SPI(END_HS5_DIAG_CONFIG2,0x0000);
__delay_ms(7);
escrita_SPI(END_HS5_OUTPUT_CONFIG,0x001E);
__delay_ms(7);
escrita_SPI(END_ERR_UC0CH1_PART1,0x0000);
__delay_ms(7);
escrita_SPI(END_ERR_UC0CH1_PART2,0x0000);
__delay_ms(7);
escrita_SPI(END_ERR_UC1CH1_PART1,0x0000);
__delay_ms(7);
escrita_SPI(END_ERR_UC1CH1_PART2,0x0000);
```



```
    __delay_ms(7);  
    escrita_SPI(END_ERR_UC0CH2_PART1,0x0000);  
    __delay_ms(7);  
    escrita_SPI(END_ERR_UC0CH2_PART2,0x0000);  
    __delay_ms(7);  
    escrita_SPI(END_ERR_UC1CH2_PART1,0x0000);  
    __delay_ms(7);  
    escrita_SPI(END_ERR_UC1CH2_PART2,0x0000);  
    __delay_ms(7);  
    escrita_SPI(END_FW_EXTERNAL_REQUEST,0x0000);  
    __delay_ms(7);  
    escrita_SPI(END_DIAGNOSTICS_OPTION,0x0001);  
    __delay_ms (100);  
}  
  
#endif
```

## 7.4 APÊNDICE D

```
#include "Prototipos.h"
#include "Definicoes.h"
#include "Config.h"

/*Recebe o canal ao qual deve ser feita a conversão adc e retorna o valor
convertido*/
unsigned int converte_adc (unsigned char porta)
{
    unsigned char x = porta;
    unsigned int resultado;

    ADCON0bits.CHS2 = (x >> 2) & 0x01;
    ADCON0bits.CHS1 = (x >> 1) & 0x01;
    ADCON0bits.CHS0 = (x) & 0x01;
    ADCON0bits.GO = 1;

    while(ADCON0bits.GO == 1);
    resultado = (ADRESH << 8) | ADRESL;
    return resultado;
}
```

## 7.5 APÊNDICE E

```
#include <xc.h>
#include "displayLCD.h"
#include "Config.h"
#include "Definicoes.h"
#include "Prototipos.h"

unsigned int tempo_engate = 0;

// configura interrupção a cada 1ms
void interrupt isr(void)
{
    if (TMR0IF && T0IF)
    {
        T0IF = 0;
        INTCONbits.TMR0IF = 0;
        TMR0 = 67;

        if (tempo_engate > 0) tempo_engate--;
    }
}

// Configura Timer 0
void ConfigTimer0(void)
```

```

{
    T0CON = 0b11000101;
    SPPCON = 0b00000000;
    SPPCFG = 0b00000000;

    TMR0 = 67;
}

// Configura entradas e saidas do PIC
void init_hardware(void)
{
    TRISAbits.RA0 = 1;    // SENSORPRESSAO (AN0) configurado como
entrada
    TRISAbits.RA1 = 1;    // SENSORENGATE (AN1) configurado como entrada
    TRISAbits.RA2 = 1;    // SENSORSELECAO (AN1) configurado como
entrada

    TRISBbits.RB7 = 0;    // EV1 configurado como saida
    TRISBbits.RB6 = 0;    // EV2 configurado como saida
    TRISBbits.RB5 = 0;    // EV3 configurado como saida
    TRISBbits.RB4 = 0;    // EV4 configurado como saida
    TRISBbits.RB3 = 0;    // CHIP_SELECT configurado como saida
    TRISBbits.RB2 = 0;    // Bomba de Pressao como saida
    TRISBbits.RB1 = 0;    // SCK configurado como saida
    TRISBbits.RB0 = 1;    // MOSI configurado como saida

```

```
TRISCbits.RC7 = 0;    // MISO configurado como saida

TRISAbits.RA3 = 1;    // GSL0 configurado como saida
TRISAbits.RA4 = 1;    // GSL1 configurado como saida
TRISAbits.RA5 = 1;    // GSL2 configurado como saida
TRISEbits.RE1 = 1;    // GSL3 configurado como saida

}

// Inicializa o ADC
void Init_ADC (void)
{

    ADCON1 = 0b00001100;

    ADCON0bits.ADON = 1;// habilita conversor
    ADCON2bits.ADFM = 1;
    ADCON2bits.ADCS0 = 0;
    ADCON2bits.ADCS1 = 1;
    ADCON2bits.ADCS2 = 1;
    ADCON2bits.ACQT0 = 0;
    ADCON2bits.ACQT1 = 1;
    ADCON2bits.ACQT2 = 1;

}
```

```
// Realiza a leitura dos pinos coectados ao Joystick
char LeituraJoystick(void)
{
    char Joystick;

    Joystick = (GSL3<<3) | (GSL2 << 2) | (GSL1<<1) | GSL0;
    return (Joystick);
}

// Verifica qual a posicao do joystick para
// efetuar a troca de marcha

void TrocaMarcha(void)
{
    static char Marcha = NEUTRO;
    static char JoystickAnterior = 0b0000;
    char JoystickAtual = LeituraJoystick();
    switch (JoystickAtual)
    {
        case 0b1111 :
            // Falha
            break;

        case 0b1010 :
            // D
            break;
```

case 0b1100:

```
// -  
  
//Verifica se a Posicao do Joystick Mudou  
if (JoystickAnterior! = JoystickAtual)  
{  
  
    // Verifica qual a marcha, soma +1 para  
    //chamar a funcao transmissao que realiza  
    //a troca de marcha  
    Marcha = VerificaMarcha ();  
    Marcha = Marcha - 1;  
    transmissao(Marcha);  
}  
  
break;
```

case 0b0101 :

```
// +  
  
//Verifica se a Posicao do Joystick Mudou  
if (JoystickAnterior! = JoystickAtual)  
{  
  
    // Verifica qual a marcha, soma +1 para  
    //chamar a funcao transmissao que realiza  
    //a troca de marcha  
    Marcha = VerificaMarcha ();  
    Marcha = Marcha + 1;  
    transmissao(Marcha);  
}
```

```
        break;

    case 0b1001 :
        // N
        transmissao(NEUTRO);
        break;

    case 0b0011 :
        // R
        transmissao(RE);
        break;

    case 0b0110 :
        // A/M
        break;

    case 0b0000 :
        // NAA
        break;

    default :
        break;
}

JoystickAnterior = JoystickAtual;
}
```



```
// Verifica qual a marcha atual do sistema de transmissão
char VerificaMarcha(void)
{
    int Engate;
    int Selecao;
    char MarchaAtual;
    Engate = converte_adc (SENSORENGATE);
    Selecao = converte_adc (SENSORSELECAO);
    // Se o sensor de selecao for menor que 1,7v
    if (Selecao < 348)
    {
        // Se o sensor de Engate for menor que 1,7v
        if (Engate < 348)
        {
            // O sistema está em quinta marcha
            MarchaAtual = QUINTA_MARCHA;
        }
        else
        {
            // se for maior que 1,7 o Sistema está em Ré
            MarchaAtual = RE;
        }
    }
    //Se o Sensor de Selecao estiver entre 1,7v e 3v
    else
    if ((Selecao > 348) && (Selecao < 614))
    {
```

```
// Se sensor de engate for menor que 1,7v
if (Engate < 384)
{
    MarchaAtual = TERCEIRA_MARCHA;
} else
//Se sensor de engate for maior que 3v
if (Engate > 614)
{
    MarchaAtual = QUARTA_MARCHA;
}
// Se estiver entre 1,7v e 3v
else
{
    MarchaAtual = NEUTRO;
}
}
else
//Se o Sensor de Selecao for maior que 3v
if (Selecao > 614)
{
    //Se o sensor de selecao for menor que 1,7v
    if (Engate < 384)
    {
        MarchaAtual = PRIMEIRA_MARCHA;
    } // se o sensor de selecao for maior que 1,7v
    else
```

```
{
    MarchaAtual = SEGUNDA_MARCHA;
}
}

//retorna a macha atual
return (MarchaAtual);
}

// funcao que realiza o comando de cada eletrovalvula em cada marcha
void transmissao(char Marcha)
{
    switch(Marcha)
    {
        case RE:
            EV4 = LIGADO;
            EV3 = DESLIGADO;
            tempo_engate = TEMPOACIONAVALVULAS;
            while (tempo_engate <= 0);
            EV2 = LIGADO;
            EV1 = DESLIGADO;
            tempo_engate = TEMPODESACIONAVALVULAS;
            while (tempo_engate <= 0);
            EV4 = DESLIGADO;
            EV3 = DESLIGADO;
            EV2 = DESLIGADO;
            EV1 = DESLIGADO;
```

```
break;
```

```
case NEUTRO:
```

```
    EV4 = DESLIGADO;
```

```
    EV3 = DESLIGADO;
```

```
    EV2 = DESLIGADO;
```

```
    EV1 = DESLIGADO;
```

```
break;
```

```
case PRIMEIRA_MARCHA:
```

```
    EV4 = DESLIGADO;
```

```
    EV3 = LIGADO;
```

```
    tempo_engate = TEMPOACIONAVALVULAS;
```

```
    while (tempo_engate <= 0);
```

```
    EV2 = DESLIGADO;
```

```
    EV1 = LIGADO;
```

```
    tempo_engate = TEMPODESACIONAVALVULAS;
```

```
    while (tempo_engate <= 0);
```

```
    EV4 = DESLIGADO;
```

```
    EV3 = DESLIGADO;
```

```
    EV2 = DESLIGADO;
```

```
    EV1 = DESLIGADO;
```

```
break;
```

case SEGUNDA\_MARCHA:

EV4 = DESLIGADO;

EV3 = LIGADO;

tempo\_engate = TEMPOACIONAVALVULAS;

while (tempo\_engate <= 0);

EV2 = LIGADO;

EV1 = DESLIGADO;

tempo\_engate = TEMPODESACIONAVALVULAS;

while (tempo\_engate <= 0);

EV4 = DESLIGADO;

EV3 = DESLIGADO;

EV2 = DESLIGADO;

EV1 = DESLIGADO;

break;

case TERCEIRA\_MARCHA:

EV4 = DESLIGADO;

EV3 = DESLIGADO;

tempo\_engate = TEMPOACIONAVALVULAS;

while (tempo\_engate <= 0);

EV2 = DESLIGADO;

EV1 = LIGADO;

tempo\_engate = TEMPODESACIONAVALVULAS;

```
while (tempo_engate <= 0);
```

```
EV4 = DESLIGADO;
```

```
EV3 = DESLIGADO;
```

```
EV2 = DESLIGADO;
```

```
EV1 = DESLIGADO;
```

```
break;
```

```
case CUARTA_MARCHA:
```

```
EV4 = DESLIGADO;
```

```
EV3 = DESLIGADO;
```

```
tempo_engate = TEMPOACIONAVALVULAS;
```

```
while (tempo_engate <= 0);
```

```
EV2 = LIGADO;
```

```
EV1 = DESLIGADO;
```

```
tempo_engate = TEMPODESACIONAVALVULAS;
```

```
while (tempo_engate <= 0);
```

```
EV4 = DESLIGADO;
```

```
EV3 = DESLIGADO;
```

```
EV2 = DESLIGADO;
```

```
EV1 = DESLIGADO;
```

```
break;
```

```
case QUINTA_MARCHA:
```

```
    EV4 = LIGADO;
    EV3 = DESLIGADO;
    tempo_engate = TEMPOACIONAVALVULAS;
    while (tempo_engate <= 0);
    EV2 = DESLIGADO;
    EV1 = LIGADO;
    tempo_engate = TEMPODESACIONAVALVULAS;
    while (tempo_engate <= 0);
    EV4 = DESLIGADO;
    EV3 = DESLIGADO;
    EV2 = DESLIGADO;
    EV1 = DESLIGADO;

    break;
case 7 :
    Marcha = QUINTA_MARCHA;
    break;
}
}
```