

**Dissertação de Mestrado**

**PROJETO E DESENVOLVIMENTO DE UM  
SISTEMA DE CONTROLE ELETRÔNICO DE  
INJEÇÃO DIRETA DE GNV EM VEÍCULOS  
MOVIDOS A GASOLINA/ÁLCOOL**

**Filipe Silva Pereira Carneiro**

Brasília, dezembro de 2011

**UNIVERSIDADE DE BRASÍLIA**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia

## DISSERTAÇÃO DE MESTRADO

# PROJETO E DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE ELETRÔNICO DE INJEÇÃO DIRETA DE GNV EM VEÍCULOS MOVIDOS A GASOLINA/ÁLCOOL

**Filipe Silva Pereira Carneiro**

Relatório submetido como requisito parcial para obtenção do grau de Mestre em Sistemas Mecatrônicos

### **Banca Examinadora**

Prof. Carlos Humberto Llanos Quintero, UnB/ENM  
(orientador)

Prof. Alessandro Borges de Sousa Oliveira,  
UnB/FGA

Prof. Rudi Henri van Els, UnB/FGA

---

---

---

## Dedicatória(s)

*Dedico esse trabalho ao meu orientador, que sempre acreditou nesse projeto, mesmo quando eu duvidava. Também aos meus colegas de mestrado, que tanto me ajudaram, e sem os quais não teria sido possível trilhar a difícil caminhada de se dividir entre o mestrado, o trabalho, e todas as outras atividades.*

*Filipe Silva Pereira Carneiro*

## **Agradecimentos**

*Agradeço ao apoio que me foi dado pelas duas empresas nas quais trabalhei durante o mestrado, Ztecnologia e Autotrac, com uma compreensão, nem sempre presente, de que o mestrado é importante para o crescimento do profissional e não se trata apenas de um título acadêmico.*

*Também ao meu orientador Prof. Carlos Humberto Llanos, que aprendi a respeitar e admirar como pessoa, e o qual considero um amigo: obrigado pela sua paciência e persistência comigo e com o projeto, por tornar possível esse desafio de conciliar as vidas profissional e acadêmica, que talvez tenha sido o maior que já enfrentei.*

*Filipe Silva Pereira Carneiro*

---

## RESUMO

O projeto descrito neste documento apresenta uma primeira abordagem ao desafio de se utilizar injeção de GNV (Gás Natural Veicular) em veículos movidos originalmente movidos à gasolina/álcool, utilizando para isso Lógica Nebulosa a fim de evitar-se cálculos complexos e a necessidade de armazenamento de mapas de correção. A estratégia empregada no projeto utiliza o tempo de injeção lido da UCE (Unidade de Controle Eletrônico) original de gasolina/álcool como base para determinar o tempo de injeção de GNV, e define uma correção baseada em Lógica Nebulosa, com sensores e o próprio tempo de injeção original como entradas e a correção sobre o tempo de injeção como saída. O desenvolvimento do projeto aconteceu em três grandes áreas: *hardware*, *firmware* e estratégia de injeção, todos sendo tratados neste documento. O *hardware* e o *firmware* foram especificados, projetados, desenvolvidos e testados, com bons resultados em todos os testes, provando sua capacidade em desempenhar as tarefas para as quais foram desenvolvidos: ler os sensores MAP, de rotação, de pressão de gás e de sonda lambda, ler o tempo de injeção original do veículo, emular os sensores MAP e de sonda lambda e atuar nos bicos injetores de GNV. A base de regras da Lógica Nebulosa foi criada a partir da observação de uma UCE de GNV comercial (previamente instalada no veículo), com o objetivo de fazer com que o projeto desenvolvido se comportasse o mais próximo possível a essa ECU. A partir de logs de campo da ECU comercial, a estratégia desenvolvida foi testada computacionalmente e esses testes apresentaram bons resultados. Resultados de campo, entretanto, mostraram a necessidade de aprofundamento do problema para que sejam promovidos alguns ajustes na Lógica Nebulosa. Os resultados atuais indicam que a base de regras deve ser desenvolvida por meio de testes de campo ao invés de simulações computacionais, como foi feito. Finalmente, a plataforma desenvolvida (*hardware* e *firmware*) foi capaz de manter o motor em funcionamento quando o tempo de injeção de GNV foi calculado a partir de uma correção fixa sobre tempo de injeção da ECU original. Portanto, a plataforma foi desenvolvida com sucesso, o que permite a experimentação de diferentes estratégias para cálculo do tempo de injeção de GNV (como foi feito com o controlador baseado em Lógica Nebulosa)

*Palavras chave: GNV, Injeção Eletrônica de Combustível, Lógica Nebulosa, Hardware, Firmware.*

---

## ABSTRACT

The project described in this document presents a first approach to the challenge of using NGV (Natural Gas Vehicle) injection in gasoline/ethanol vehicles, by using fuzzy logic to avoid complex calculations and the need to store correction maps. The strategy employed in the project adopts the injection time read from the original ECU (Electronic Control Unit) of gasoline/alcohol as the first approach to determine the NGV injection time, and it defines a correction based on a fuzzy logic, with sensors and the original injection time itself as inputs and the correction as output. The designed project involved the design in three major areas: hardware, firmware and injection strategy, all of them being addressed by this document. The hardware and firmware have been specified, designed, developed and tested, with good results in all tests, proving their capacity to perform the tasks they have been designed for: reading MAP, rotation, gas pressure and lambda oxygen sensors, reading the original injection time, emulating MAP and lambda oxygen sensors and actuating in the NGV injector. The rule base of the fuzzy logic was created from the observation of a commercial NGV ECU (previously installed in the vehicle) in order to make the designed project to behave as close as possible to this ECU. From field logs of the commercial ECU, the designed strategy has been tested computationally, and these tests have shown good results. The field tests, however, have showed the need to delve into the problem in order to perform some adjusts in the fuzzy logic. The current results point out the idea of designing the rule base from field testing instead of the computational simulations as have been done. Finally, the developed platform (hardware and firmware) was capable to maintain the engine running when the injection time of NGV was calculated through a fixed correction in the injection time of the original ECU. Therefore, the platform has been successfully developed, which allows the experimentation of different strategies for calculating the NGV injection time (as performed with the developed fuzzy logic controller).

*Keywords: NVG, Electronic Fuel Injection, Fuzzy Logic, Hardware, Firmware.*

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	MOTIVAÇÃO	1
1.2	ABORDAGEM UTILIZADA PARA O PROJETO	2
1.3	OBJETIVOS DO PROJETO	3
1.3.1	Objetivo geral	3
1.3.2	Objetivos específicos	3
1.4	ASPECTOS METODOLÓGICOS UTILIZADOS NO TRABALHO	4
1.5	RESULTADOS OBTIDOS DO TRABALHO	4
1.6	ESTRUTURA DA DISSERTAÇÃO	5
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>6</b>
2.1	REVISÃO BIBLIOGRÁFICA	6
2.2	INJEÇÃO E IGNIÇÃO DE MOTORES DE CICLO OTTO	7
2.3	ASPECTOS TECNOLÓGICOS DA INJEÇÃO E IGNIÇÃO ELETRÔNICAS	8
2.3.1	Os principais Sensores de um sistema de injeção/ignição eletrônica	9
2.3.2	Principais atuadores controlados por um sistema de injeção/ignição eletrônica	10
2.3.3	A UCE (Unidade de Comando Eletrônico)	12
2.3.4	Relação ar-combustível e os principais gases emitidos por motores a combustão	13
2.4	GNV E CONSIDERAÇÕES EM RELAÇÃO AO SEU USO	14
2.4.1	Um breve histórico do GNV no Brasil [37] [38]	14
2.4.2	Modificações requeridas para o uso de GNV	16
2.4.3	Kits disponíveis no mercado	17
2.4.4	Vantagens do GNV [48]	18
2.4.5	Desvantagens do GNV	18
2.4.5.1	PROBLEMAS ENCONTRADOS NA CONVERSÃO PARA GNV	18
2.4.5.2	SOLUÇÕES PARA OS PROBLEMAS DE CONVERSÃO PARA GNV	18
2.5	ASPECTOS BÁSICOS DE LÓGICA NEBULOSA ( <i>FUZZY LOGIC</i> )	19
2.5.1	Conjunto Nebuloso	19
2.5.2	Funções de Pertinência	20
2.5.2.1	Funções Direitas	20
2.5.2.2	Funções Esquerdas	21
2.5.2.3	Funções Triangulares	22
2.5.2.4	Funções Trapezoidais	22
2.5.2.5	Funções Gaussianas	23
2.5.2.6	Funções Sino Generalizadas	23
2.5.3	Variáveis lingüísticas	24
2.5.4	Base de Conhecimento	24
2.5.5	Sistemas difusos, fuzzificação e defuzzificação	25
2.5.5.1	Fuzzificação	25
2.5.5.2	Aplicação de Regras	25
2.5.5.3	Combinação de Conseqüentes	26
2.5.5.4	Defuzzificação	26
2.5.6	Método Sugeno	26
2.6	CONCLUSÕES DO CAPÍTULO	27
<b>3</b>	<b>METODOLOGIA USADA PARA O PROJETO DO CONTROLADOR</b>	<b>28</b>
3.1	O PROJETO PASSO A PASSO	28
3.2	DESCRIÇÃO FUNCIONAL DO PRODUTO	28
3.2.1	Breve descrição do produto	28
3.2.2	Funções	29
3.3	ESTUDO DE UM KIT GNV COMERCIAL	30
3.3.1	Objetivos	30
3.3.2	Sistema de GNV utilizado	30
3.3.3	Método de Medição	32
3.3.3.1	Sinais Medidos	32
3.3.3.2	Dispositivo de Aquisição dos Sinais	33
3.3.3.3	Tratamento Analógico dos Sinais	33
3.3.3.4	Aquisição digital dos Sinais	39
3.3.4	Cálculo dos parâmetros dos dados lidos	40
3.3.4.1	Sinal de Gás	41
3.3.4.2	Bico Gás 1	41
3.3.4.3	Pressão da Flauta	44
3.3.4.4	Rotação	45
3.3.4.5	Bico Original – UCE 1	46

3.3.4.6	Sonda Lambda.....	47
3.3.4.7	MAP .....	48
3.3.4.8	Bico Original – Bico 1.....	50
3.3.4.9	Temperatura do Redutor de Pressão.....	50
<b>4</b>	<b>O PROJETO DO <i>HARDWARE</i> DO CONTROLADOR .....</b>	<b>52</b>
4.1	Separação dos Circuitos Digitais e Analógicos dos Circuitos de Potência.....	52
4.1.1	Disposição do SINAL_GAS.....	52
4.2	PCI – Circuitos digitais e analógicos .....	52
4.2.1	Especificações da PCI – Circuitos digitais e analógicos .....	53
4.2.2	Proteções de Entrada .....	54
4.2.3	Fonte de Alimentação .....	54
4.2.4	CPU .....	56
4.2.5	Leitura e Emulação do Sensor MAP .....	59
4.2.6	Leitura e Emulação do sensor de Sonda Lambda .....	61
4.2.7	Leitura do sensor de pressão de flauta.....	64
4.2.8	Leitura e Comutação dos Bicos Injetores originais .....	66
4.2.9	Sensor de Rotação .....	68
4.2.10	Sensor de Rotação Alternativo .....	68
4.2.11	Sinal de acionamento dos Bicos .....	70
4.2.12	Saídas Genéricas .....	71
4.3	PCI – Circuitos de potência .....	73
4.3.1	Especificações do <i>hardware</i> .....	73
4.3.2	Circuito de Acionamento dos Bicos .....	73
4.3.3	Circuito de geração do SINAL_GAS .....	74
<b>5</b>	<b>ESTRATÉGIA DE FUNCIONAMENTO DO CONTROLADOR .....</b>	<b>75</b>
5.1	Descrição da estratégia de funcionamento .....	75
5.2	Procedimentos seguidos .....	75
5.2.1	Parâmetro de Avaliação ( <i>erro</i> ).....	76
5.2.2	Determinação da Correção Fixa ( <i>k</i> ).....	76
5.2.3	Determinação dos Conjuntos Nebulosos e suas Funções de Pertinência.....	76
5.2.4	Determinação dos Predecessores.....	80
5.2.5	Criação da Base de Regras .....	81
5.3	Conclusões.....	85
<b>6</b>	<b>O PROJETO DO <i>FIRMWARE</i> DO CONTROLADOR.....</b>	<b>86</b>
6.1	Conceitos que Guiaram o Desenvolvimento .....	86
6.2	Estruturação do Código.....	86
6.3	Bibliotecas desenvolvidas .....	87
6.3.1	libDevDrivers.....	87
6.3.2	libCommon.....	89
6.3.3	libFuzzy.....	89
6.3.4	libATMega32.....	90
6.4	Escalonamento de Tarefas.....	92
6.5	A Aplicação .....	94
6.6	Funções Criadas .....	95
<b>7</b>	<b>TESTES E RESULTADOS ALCANÇADOS.....</b>	<b>96</b>
7.1	Testes de Bancada.....	96
7.1.1	Fonte de Alimentação .....	96
7.1.2	Leitura dos sensores de Pressão da Flauta, Sensor MAP e Sonda Lambda .....	98
7.1.3	Emulação dos sensores MAP e de Sonda Lambda.....	99
7.1.4	Leitura dos Bicos Injetores originais .....	100
7.1.5	Sensor de Rotação .....	101
7.1.6	Acionamento do Bico de Gás.....	102
7.2	Testes de Campo .....	103
7.2.1	Metodologia de Testes.....	103
7.2.2	Resultados.....	106
<b>8</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>108</b>
<b>9</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>109</b>
<b>Anexo I</b>	<b>Protótipos das Funções e Macros Desenvolvidas .....</b>	<b>112</b>
<b>1</b>	<b>MCUDrivers.....</b>	<b>112</b>
1.1	ADC.h.....	112
1.2	I2C.h.....	113
1.3	IO.h.....	114
1.4	TIMER0.h .....	115
1.5	TIMER1.h .....	116
1.6	TIMER2.h .....	116
1.7	UART.h.....	117
<b>2</b>	<b>Common.....</b>	<b>118</b>
2.1	STRING.h.....	118



2.2	CLOCK.h.....	119
2.3	COMMONOP.h.....	119
<b>3</b>	<b>DEVDrivers.....</b>	<b>121</b>
3.1	BICOS.h.....	121
3.2	CONSOLE.h.....	122
3.3	EEPROM.h.....	123

# LISTA DE FIGURAS

Figura 1: Número de conversões anuais para GNV [1].	1
Figura 2: Diagrama de Blocos da abordagem utilizada.	3
Figura 3: Etapas do ciclo OTTO [28]	8
Figura 4: Sistema de injeção monoponto [31]	9
Figura 5: Sistema de injeção multiponto [31]	9
Figura 6: Vista em corte de um bico injetor [32].	11
Figura 7: Vista em corte de uma vela de ignição [33].	11
Figura 8: Vista em corte de uma bobina de ignição [33].	12
Figura 9: Diagrama de blocos de um sistema de injeção/ignição eletrônica	12
Figura 10: Consumo de Energia Primária por Tipo de Energético no mundo – [42] – e no Brasil – [43].	15
Figura 11: Gráfico dos preços dos combustíveis e do N° de conversões [1]	16
Figura 12: Modificações em um carro convertido para uso de GNV [45]	17
Figura 13: Diferença entre um conjunto clássico e um conjunto nebuloso	20
Figura 14: Exemplo de função de pertinência direita	21
Figura 15: Exemplo de função de pertinência esquerda	21
Figura 16: Exemplo de função de pertinência triangular	22
Figura 17: Exemplo de função de pertinência trapezoidal	23
Figura 18: Exemplos de funções de pertinência gaussianas	23
Figura 19: Exemplos de funções de pertinência gaussianas	24
Figura 20: Exemplo de combinação de consequentes segundo Mamdani.	26
Figura 21: Exemplo de cálculo em um sistema de inferência do tipo Sugeno	27
Figura 22: Sinal de Gás quando veículo está rodando no combustível original	34
Figura 23: Sinal de Gás quando veículo está rodando no GNV.	34
Figura 24: Sinal dos quatro bicos injetores (Bico Gás 1, 2, 3 e 4).	34
Figura 25: Sinal de um dos bicos injetores - visão aproximada	35
Figura 26: Sinal de Pressão na Flauta.	35
Figura 27: Sinal de Rotação - visão macro	35
Figura 28: Sinal de Rotação - visão aproximada	36
Figura 29: Bico Original - UCE	36
Figura 30: Sinal Sonda Lambda	36
Figura 31: Sinal do Sensor MAP	37
Figura 32: Sinal Bico Original - Bico, rodando no GNV. Visão macro	37
Figura 33: Sinal Bico Original - Bico, rodando no GNV. Visão aproximada	37
Figura 34: Sinal de Temperatura do Redutor de Pressão	38
Figura 35: Circuito para sinais que necessitam de amplificação	39
Figura 36: Circuito para sinais que não necessitam nem de amplificação nem de atenuação	39
Figura 37: Circuito para sinais que necessitam de atenuação	39
Figura 38: sistema de aquisição dos sinais	40
Figura 39: Forma de onda do sinal Bico Gás 1 - vários ciclos de injeção	41
Figura 40: Forma de onda do sinal Bico Gás 1 - um único ciclo de injeção	41
Figura 41: Tempo de <i>hold</i>	42
Figura 42: Período de chaveamento durante o <i>hold</i>	43
Figura 43: Tempo em alta durante o <i>hold</i>	43
Figura 44: Perfil de corrente no bico injetor de GNV (em rosa) – repare que a corrente durante o <i>hold</i> estabiliza em aproximadamente 34% da corrente máxima durante o <i>peak</i> .	43
Figura 45: Pressão da Flauta	44
Figura 46: FFT da Pressão da Flauta	45

Figura 47: Sinal de Pressão da Flauta filtrado	45
Figura 48: Sinal de Rotação	46
Figura 49: Tempo de Injeção de gasolina	47
Figura 50: Sensor de Sonda Lambda	47
Figura 51: FFT do sensor de Sonda Lambda	48
Figura 52: Sensor de Sonda Lambda filtrado	48
Figura 53: Sensor MAP	49
Figura 54: FFT do Sensor MAP	49
Figura 55: Sensor de Sonda Lambda filtrado	50
Figura 56: Bico Original - Bico 1	50
Figura 57: Sinal de Gás e Temperatura do Redutor de Pressão	51
Figura 58: Sinal de Gás e Temperatura do Redutor de Pressão – Ampliação	51
Figura 59: Load Dump	54
Figura 60: Esquemático do circuito de proteção de entrada.	54
Figura 61: Esquemático da Fonte.	56
Figura 62: Esquemático da CPU	59
Figura 63: Circuito de leitura, emulação e comutação do Sensor MAP	59
Figura 64: Circuito de leitura, emulação e comutação do sensor de Sonda Lambda	62
Figura 65: Circuito de leitura do sensor de Pressão da Flauta	64
Figura 66: Circuito de leitura de sinal de injeção e congelamento dos bicos injetores.	66
Figura 67: Circuito de leitura do Sensor de Rotação	68
Figura 68: Circuito alternativo do Sensor de Rotação	69
Figura 69: Circuito de criação do sinal de controle do circuito de acionamento dos bicos injetores de gás.	70
Figura 70: Saída genérica	72
Figura 71: Circuito de Acionamento dos Bicos	74
Figura 72: Circuito de geração do SINAL_GAS	74
Figura 73: Estratégia de funcionamento baseada no cálculo de tempo de injeção da centralina original.	75
Figura 74: Funções de Pertinência da Pressão da Flauta	77
Figura 75: Funções de Pertinência do Sensor MAP	78
Figura 76: Funções de Pertinência do Sensor de Rotação	78
Figura 77: Funções de Pertinência do Sensor de Temperatura	79
Figura 78: Funções de Pertinência do Sensor de Sonda Lambda	79
Figura 79: Funções de Pertinência do Sensor de Pressão da Flauta	80
Figura 80: Comparação entre os tempos de injeção de gasolina, de gás do <i>kit</i> comercial e o calculado pelo sistema de inferência nebuloso	85
Figura 81: Estruturação de Código	87
Figura 82: Máquinas de estados de injeção	88
Figura 83: Fluxograma da Aplicação	95
Figura 84: Foto do ambiente de testes de bancada.	96
Figura 85: Teste do Acionamento do Bico de Gás	102
Figura 86: Teste da Corrente de Acionamento do Bico de Gás	102
Figura 87: Circuito para congelamento do acionamento dos bicos por parte do kit comercial	103
Figura 88: Foto com o detalhe da conexão para leitura dos sinais dos sensores e da injeção de combustível original.	104
Figura 89: Conexão que permite ao sistema em teste interromper o sinal de acionamento dos bicos de gás.	104
Figura 90: Placa para leitura do console	105
Figura 91: Visão geral do sistema testado inserido no veículo de teste, 1ª foto.	105
Figura 92: Visão geral do sistema testado inserido no veículo de teste, 2ª foto.	105

Figura 93: Visão geral do sistema testado inserido no veículo de teste, 3ª foto.

106

Figura 94: Sistema sendo comandado a partir do computador.

106

# LISTA DE TABELAS E CÓDIGOS

Tabela 1: Relações estequiométricas dos principais combustíveis [34].	13
Tabela 2: Principais gases poluentes	13
Tabela 3: Sinais do <i>kit</i> GNV comercial	30
Tabela 4: Faixas de tensão dos sinais medidos	33
Tabela 5: Arquivos com resultados de medição	40
Tabela 6: Resumo da faixa de valores de cada sinal	77
Tabela 7: Funções de Pertinência da Pressão da Flauta	77
Tabela 8: Funções de Pertinência do Sensor MAP	77
Tabela 9: Funções de Pertinência do Sensor de Rotação	78
Tabela 10: Funções de Pertinência do Sensor de Temperatura	78
Tabela 11: Funções de Pertinência do Sensor de Sonda Lambda	79
Tabela 12: Funções de Pertinência do Sensor de Pressão da Flauta	79
Tabela 13: Base de Regras para Rotação, Pressão da Flauta e $TI_{Gasolina}$	83
Tabela 14: Base de Regras para Sonda Lambda, Pressão da Flauta e $TI_{Gasolina}$	84
Tabela 15: Resumo das saídas dos Conjuntos Nebulosos de Rotação	84
Tabela 16: Resumo das saídas dos Conjuntos Nebulosos de Sonda Lambda	84
Tabela 17: Resumo das saídas dos Conjuntos Nebulosos de Pressão da Flauta	84
Tabela 18: Resumo das saídas dos Conjuntos Nebulosos de $TI_{Gasolina}$	85
Tabela 19: Escalonamento das tarefas	94
Tabela 20: Regulação de linha da fonte de alimentação	97
Tabela 21: Regulação de carga da fonte de alimentação	98
Tabela 22: Resultados do teste de leitura dos sensores de Pressão da Flauta, Sensor MAP e Sonda Lambda	99
Tabela 23: Resultado do Teste de Emulação do Sensor MAP	100
Tabela 24: Resultado do Teste de Emulação do Sensor de Sonda Lambda	100
Tabela 25: Resultado dos testes de leitura do tempo de injeção	101
Tabela 26: Resultado dos testes do Sensor de Rotação	101



# 1 INTRODUÇÃO

## 1.1 MOTIVAÇÃO

O automóvel movido a gasolina ou álcool é um dos principais meios de transporte no Brasil e no mundo e a poluição causada por esses automóveis sempre foi motivo de preocupação, principalmente nos dias atuais, com a crescente conscientização da importância e da urgência em se preservar o meio ambiente. Outro ponto que vem se tornando crítico relaciona-se aos preços praticados pelos postos de gasolina, cada dia mais elevados, devido, em grande parte, à falta de um combustível que possa competir e forçar uma queda de preço.

Nesse contexto, obtiveram-se nos últimos anos avanços tecnológicos que permitiram introduzir técnicas sofisticadas nos automóveis visando dois objetivos: (1) aumentar o rendimento dos motores (reduzindo o consumo de combustível) e (2) diminuir a emissão de poluentes. Na busca por esses objetivos, aparecem os primeiros carros comerciais que se utilizam de injeção eletrônica na década de 90. A injeção eletrônica veio substituir o carburador na tarefa de dosar a quantidade de combustível da mistura ar-combustível. Por se tratar de um processo controlado eletronicamente, a injeção eletrônica é muito mais eficiente que a carburação, diminuindo o consumo de combustível e a emissão de poluentes e aumentando a potência. Nas seções 2.2 e 2.3 serão introduzidos com mais detalhe os conceitos básicos dos motores de combustão de ciclo Otto e os fundamentos dos sistemas de injeção eletrônica, respectivamente.

Ainda com foco na solução dos problemas descritos acima, o uso de GNV aparece como alternativa concreta em relação à gasolina e ao álcool. Embora o GNV não seja a única alternativa e existam esforços em outras direções (como por exemplo, estudos envolvendo energia solar, energia elétrica, biocombustíveis e hidrogênio, para citar alguns) as tecnologias obtidas são ainda pouco competitivas do ponto de vista de eficiência tecnológica e/ou pelos seus aspectos econômicos, pelo menos no Brasil. Neste contexto, o GNV tem-se mostrado uma alternativa barata e eficaz, com a frota de carros convertidos a esse combustível chegando a 1.661.919 em novembro de 2010, segundo o IBP (Instituto Brasileiro de Petróleo, Gás e Biocombustíveis) [1].

No entanto, segundo o mesmo Instituto (IBP), o setor de GNV vinha apresentando uma queda no número de conversões anuais desde o ano de 2007, quando o número de conversões caiu de 272.610 para 187.040 [1] (vide Figura 1), a principal causa dessa queda na procura por gás natural advém de uma crise de confiança nesse combustível, devido à ocupação pelo exército boliviano dos campos de petróleo e gás desse país ([2] e [3]). Porém, esse quadro deve se reverter com a descoberta de novas jazidas de gás natural dentro do Brasil, no contexto do Pré-sal. De acordo com a edição de 06 de Abril de 2011 do jornal *Brasil Econômico* [2], o volume de gás disponível (que soma gás nacional, gás importado da Bolívia e das unidades de gás natural liquefeito - GNL) deve dobrar até 2020 devido às descobertas no pré-sal, indo de 90.000.000 m<sup>3</sup> para algo entre 180.000.000 e 200.000.000 m<sup>3</sup>.

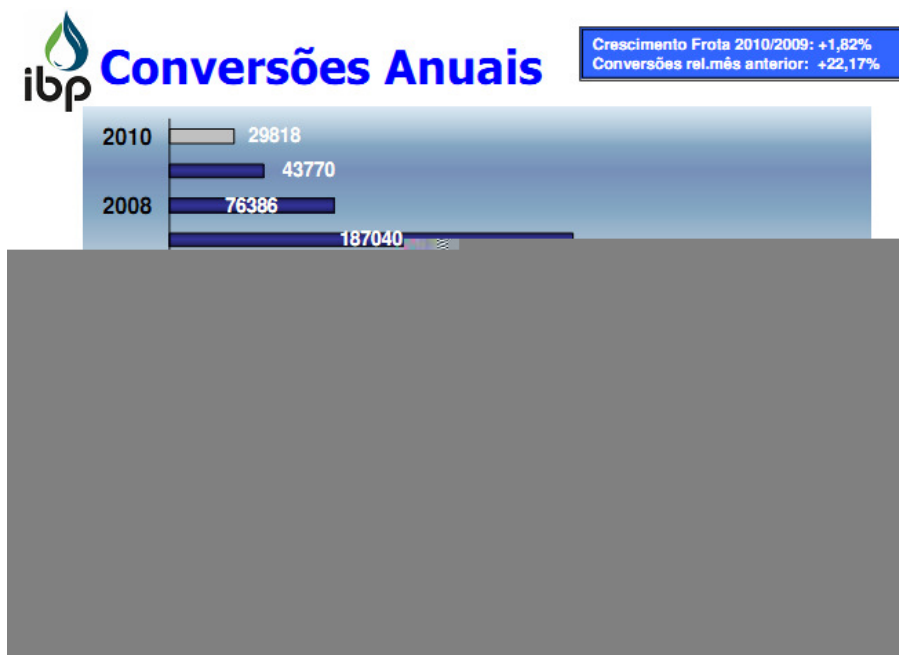


Figura 1: Número de conversões anuais para GNV [1].

Outro fator importante para o crescimento do setor GNV deve ser o aumento dos preços dos combustíveis Etanol (álcool) e Gasolina. Segundo a ANP (Agência Nacional do Petróleo), os preços médios do mês de Abril de 2011 dos três combustíveis foi: Etanol – R\$ 2,345; Gasolina – R\$ 2,825 e GNV – R\$ 1,599 [4]. Utilizando esses valores na ferramenta online disponibilizada pela SCGAS (Companhia de Gás de Santa Catarina) [5], obtemos os valores de custo por km: Etanol – R\$ 0,22; Gasolina – R\$ 0,19 e GNV – R\$0,09, isso representa uma economia de 52,63% em relação à gasolina e 59,09% em relação ao álcool.

Em um sistema GNV, é importante controlar a vazão de gás que entrará no coletor de admissão do motor, assim como controlar o sistema de ignição do carro. Existem alguns *kits* de conversão comercializados hoje que fazem essa função, mas os mais utilizados são importados e utilizam um *hardware* caro, o que os faz ter um alto custo.

A principal motivação deste trabalho é criar a central eletrônica de um *kit* de conversão para GNV baseada em uma estratégia mais simples de funcionamento (explicada brevemente na seção 1.2 e com mais detalhes na seção 5.1) e que, portanto, seja possível com um *hardware* mais barato. Adicionalmente, objetiva-se fazer uma experimentação do uso de uma lógica nebulosa para controle de injeção. Por fim, vai se tratar de um produto nacional o que, conseqüentemente, irá reduzir seu preço de venda no Brasil.

Os trabalhos propostos nesta dissertação situam-se dentro do contexto do projeto GN-JET (FINEP-Petrobrás-UnB), onde se propõe fazer avanços na direção de nacionalizar a tecnologia de *kits* GNV.

## 1.2 ABORDAGEM UTILIZADA PARA O PROJETO

De uma maneira simplificada, a central eletrônica original do carro é responsável, entre outras funções, por ler os sinais dos sensores, processá-los e gerar o sinal de acionamento de bicos originais (ver seção 2.3). Quando um carro é convertido para GNV com um *kit* de 5ª geração (ver seção 2.4.3), bicos de injeção de GNV são instalados, e com eles uma central eletrônica dedicada ao sistema GNV, responsável por comandar os novos bicos. O sistema original não é prejudicado em nada, e os bicos de injeção de combustível original continuam funcionais, o usuário do veículo convertido tem acesso a uma chave seletora de combustível instalada como parte do *kit*, a partir da qual pode optar pelo uso do combustível original ou do GNV.

Em uma estratégia convencional, o sinal de acionamento dos bicos de injeção de GNV é calculado a partir dos sinais dos sensores do carro, seguindo a mesma estratégia e refazendo alguns dos cálculos já realizados pela central eletrônica original.

O principal ponto que pretende diferenciar do sistema proposto neste projeto dos demais *kits* disponíveis no mercado é a abordagem de se aproveitar o processamento realizado pela central eletrônica original do carro dos sinais provenientes dos sensores. Dessa forma, no lugar de se refazer um complexo e oneroso cálculo de tempo de injeção a partir dos sensores, o projeto proposto irá ler o tempo de injeção já calculado pela central eletrônica original do carro (responsável pela injeção de gasolina e/ou álcool) e promover uma correção simples para adequá-lo ao GNV. Essa correção será feita baseada na leitura de alguns dos sensores com um sistema de controle baseado em lógica nebulosa, o que deve poupar a memória necessária para o armazenamento de mapas de correção e, conseqüentemente, resultar em um sistema de mais baixo custo.

Essa abordagem pode ser sumarizada nos seguintes tópicos:

- No funcionamento normal, a Central Eletrônica Original processa os sinais dos sensores e gera o sinal de acionamento dos bicos originais;
- O sistema proposto irá interceptar esse sinal a fim de processá-lo em um sistema de controle simples baseado em lógica nebulosa, usando para isso uma parte dos sinais dos sensores, e gerar o sinal de acionamento de bicos de gás;
- Por fim, o sistema proposto também será responsável por “congelar” o sinal de acionamento de bicos originais quando o seletor de modo original/gás estiver na posição gás.

A Figura 2 ilustra a abordagem utilizada (vide seção 5 para mais detalhes).



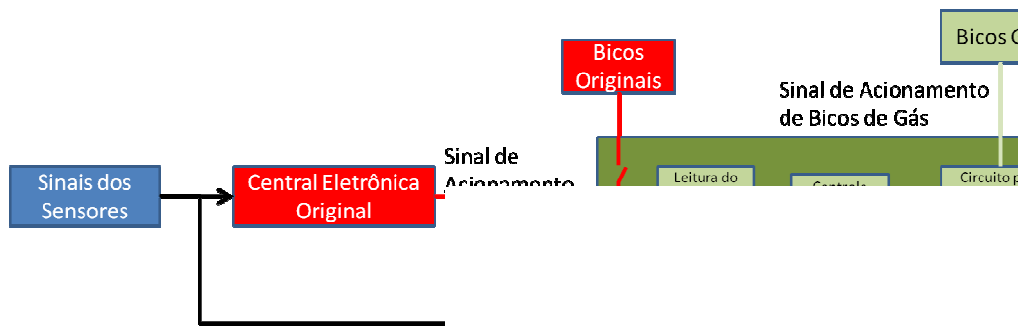


Figura 2: Diagrama de Blocos da abordagem utilizada.

## 1.3 OBJETIVOS DO PROJETO

Esta seção descreve tanto o objetivo gerais quanto os objetivos específicos do projeto.

### 1.3.1 Objetivo geral

Este trabalho faz parte do projeto GNJET (chamada pública MCT/FINEP, Ação Tripartite) que surgiu de uma parceria entre a Petrobrás, a UnB e a FINEP, tendo como um dos seus objetivos a tecnologia de injeção de GNV.

O sistema apresentado tem o objetivo de apresentar uma primeira abordagem ao desenvolvimento de um sistema de injeção de GNV em veículos movidos originalmente à gasolina/álcool. O projeto desenvolve uma proposta mais simples e de menor custo que as centrais eletrônicas comercializadas, sem que o desempenho seja prejudicado.

### 1.3.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- criar a especificação funcional do sistema a ser desenvolvido. Nessa etapa o objetivo é definir com a maior clareza possível, evitando indefinições ou tomadas de decisões ao acaso, para resultar em falhas por falta de planejamento. (seção 3.2)
- Fazer um estudo aprofundado de kits GNV amplamente utilizados no mercado para a definição de estratégias para atacar o problema do desenvolvimento de um sistema de injeção de gás. Para este caso, será empregado o sistema de instrumentação virtual LabView, assim como placas de aquisição de sinais DAQ's [6] (seção 3.3);
- Especificar o *hardware* necessário para leitura e processamento dos sinais de injeção da central eletrônica original do veículo, bem como o *hardware* necessário para acionamento dos bicos de gás. O *hardware* deve ser de baixo custo, envolvendo preferencialmente componentes facilmente obtidos no mercado nacional (vide Seções 4.2.1 e 4.3.1);
- Desenvolver o *hardware* da central eletrônica do kit, baseado em um microprocessador AVR, caracterizando uma placa de baixa complexidade e baixo custo de desenvolvimento;
- Desenvolver um algoritmo para o processamento dos sinais de injeção da central eletrônica dos sensores do veículo baseado em lógica nebulosa. O sinal gerado deve ser enviado para os injetores de gás (seção 5).
- Planejar a estruturação do código (*firmware*) a ser criado, visando boas práticas de programação e facilidade em evolução e manutenção do código.
- criar uma biblioteca de *firmware* para a placa microprocessada, envolvendo o acesso ao microcontrolador, acesso a memória EEPROM, leitura dos bicos injetores de gás serial (seção 6.3).
- criar uma metodologia de estruturação de código, com escalonamento de tarefas e testes.

*hardware* (vide seção 6.4);

- i) Criar o código da estratégia de funcionamento no *firmware* (seção 6.5).
- j) Criar uma metodologia de teste para o sistema desenvolvido, envolvendo testes de bancada (no laboratório, seção 7.1) e testes de campo, em um veículo de passeio convertido para GNV com um *kit* de 5ª geração, tanto empregando correções fixas no tempo de injeção quanto empregando a utilização da lógica nebulosa desenvolvida (seção 7.2);

## 1.4 ASPECTOS METODOLÓGICOS UTILIZADOS NO TRABALHO

A primeira etapa para o desenvolvimento do projeto foi um estudo bibliográfico e prático de diversos aspectos relacionados ao sistema a ser desenvolvido: injeção eletrônica, lógica nebulosa e GNV. Posteriormente, foram desenvolvidos o *hardware* e a biblioteca de *firmware*, que foram utilizados como plataforma para o desenvolvimento do sistema. Por fim, foram testadas diversas possibilidades de bases de regras nebulosas para controle do tempo de injeção de GNV, utilizando como parâmetro de validação a semelhança do sinal gerado com o sinal captado em campo de um *kit* GNV consagrado.

As etapas compridas podem ser resumidas em:

- a) Estudo de sistemas GNV (resumido na seção 2.4.3);
- b) Visitas técnicas para troca de idéias com engenheiros projetistas da SEVA, empresa que comercializava produtos da marca DEVEL;
- c) Adaptação de um veículo para uso de GNV com um *kit* de 5ª geração, a ser utilizado para os testes de campo do projeto;
- d) Uso de ferramentas de instrumentação virtual para coletar informações dos sinais eletrônicos do carro convertido a GNV (seção 3.3);
- e) Teste de partes do sistema eletrônico em *proto-board*;
- f) Projeto de placas de circuito impresso baseados em microcontroladores AVR e o teste da mesma;
- g) Elaboração de rotinas básicas de *firmware*, usando ferramentas e ambientes de desenvolvimento para microcontroladores;
- h) Elaboração de algoritmos baseados em *Máquinas de Estados Finitos (Finite State Machines – FSMs)* para o tratamento de sinais provenientes dos sensores e da central eletrônica original do carro e geração do sinal de acionamento dos atuadores do *kit* GNV;
- i) Testes de integração *hardware/firmware* em bancada de laboratório;
- j) Desenvolvimento de sistema baseado em lógica nebulosa para tratamento de sinais de injeção de combustível original e dos sensores, e geração do sinal de acionamento dos bicos injetores de gás, a ser embarcado no sistema projetado;
- k) Testes de campo do sistema desenvolvido (*hardware*, *firmware* e lógica de funcionamento) no veículo convertido para GNV. Esses testes envolviam a leitura dos sensores do veículo e dos sinais de injeção da Unidade de Controle Eletrônico original e o acionamento dos bicos injetores de gás (consultar seção 7.2);

## 1.5 RESULTADOS OBTIDOS DO TRABALHO

Esta seção resume os resultados obtidos e documentados nesta dissertação:

- Foi criada uma metodologia de estudo do *kit* instalado, possibilitando a aquisição de dados relativos ao seu funcionamento em diferentes condições de funcionamento. A partir dessa metodologia, foram feitos diversos registros desses dados, o que serviu de base para o desenvolvimento do sistema.
- O *hardware* criado foi testado juntamente com a biblioteca de *firmware* e mostrou boa precisão na leitura dos sensores, emulação dos sensores e capacidade de acionamento dos bicos injetores de GNV.
- A lógica baseada em lógica nebulosa, no entanto, não se mostrou eficaz, provavelmente em razão da metodologia adotada (assunto abordado na seção 7.2.2).
- Criação de uma plataforma microcontrolada para o desenvolvimento de sistemas de controle GNV, a qual permite o desenvolvimento e teste de diferentes tipos de estratégias de controle.

- Por último, o sistema foi capaz de manter o funcionamento de motor após algumas tentativas de ajustes de parâmetros, ainda que de maneira instável e sem a utilização da lógica nebulosa, apenas com uma correção fixa no tempo de injeção de combustível original.

## 1.6 ESTRUTURA DA DISSERTAÇÃO

O texto da dissertação foi dividido em 6 grandes partes:

- *Introdução*: onde são apresentados a motivação do trabalho, os objetivos gerais e específicos, os aspectos metodológicos adotados e a abordagem utilizada para atacar o problema. O seu objetivo é proporcionar ao leitor uma visão global desta dissertação.
- *Fundamentação Teórica*: nessa seção, são apresentadas algumas matérias necessárias para o entendimento do texto que se segue. Nem todo o conhecimento necessário está presente nessa seção, já que se entende que o leitor tenha conhecimentos mínimos sobre o tema desta dissertação.
- *Metodologia usada para o projeto do controlador*: essa seção apresenta a metodologia utilizada no projeto, ou seja, quais procedimentos, técnicas, meios, ferramentas e etc. foram empregados para experimentação da estratégia concebida.
- *O projeto de Hardware do controlador*: a especificação, projeto e desenvolvimento do *hardware* estão descritos nesta seção.
- *Estratégia de funcionamento do controlador*: os métodos empregados para obtenção da estratégia de funcionamento, juntamente com a base de regras alcançada estão descritos nessa seção.
- *O projeto de Firmware do controlador*: a especificação, projeto e desenvolvimento do *firmware* estão descritos nesta seção.
- *Testes e resultados alcançados*: essa seção é dividida entre duas seções: *Testes de bancada* e *Testes de campo*, sendo que a primeira descreve os testes individuais de cada funcionalidade desenvolvida e a segunda descreve os testes feitos em campo, em um veículo convertido para GNV. São discutidos também os resultados alcançados em cada etapa
- *Conclusão e Trabalhos Futuros*: nessa seção apresenta-se uma reflexão sobre tudo o que foi discutido na dissertação e propostas para trabalhos futuros, com foco no planejamento das próximas etapas a serem realizadas para melhorar o desempenho do sistema desenvolvido.
- *Referências Bibliográficas*.
- *Anexos*: esse documento possui somente um anexo, que traz os protótipos de todas funções desenvolvidas nas bibliotecas de *firmware*.

## 2 FUNDAMENTAÇÃO TEÓRICA

O objetivo desta seção é fornecer ao leitor deste documento uma fundamentação teórica suficiente para o correto entendimento de todo o texto.

No decorrer dessa seção será feita uma revisão bibliográfica dos temas relacionados à esta dissertação e serão abordados os diferentes tópicos que foram usados no desenvolvimento deste trabalho:

- INJEÇÃO E IGNIÇÃO DE MOTORES DE CICLO OTTO;
- ASPECTOS TECNOLÓGICOS DA INJEÇÃO E IGNIÇÃO ELETRÔNICAS;
- GNV E CONSIDERAÇÕES EM RELAÇÃO AO SEU USO
- ASPECTOS BÁSICOS DE LÓGICA NEBULOSA (*FUZZY LOGIC*);

### 2.1 REVISÃO BIBLIOGRÁFICA

Nesta seção será feita uma revisão bibliográfica com o objetivo de reunir a produção científica relacionada principalmente a tópicos de sistemas inteligentes aplicados a sistemas de injeção eletrônica e à injeção eletrônica de GNV.

Existe uma grande quantidade de artigos relacionados ao uso de gás natural como combustível para motores, a maior parte motivada pela busca de uma alternativa mais econômica e/ou menos poluente para os combustíveis utilizados atualmente.

A maioria dos artigos estudados investiga temas relacionados à adoção do GNV (ou outros combustíveis alternativos) como um combustível alternativo mais limpo, mais abundante ou mais econômico. Neste sentido, em [12], Santarém, C.A. et al, fazem uma análise dos meios de transporte de GNV com base no período de *payback*. Em [13], Engerer, H. e Horn, M. sugerem o uso de GNV na Europa, analisando o caso da América Latina e os desafios enfrentados. Em [14], Janssen, A. et al sugerem 5 alternativas de indicadores para medição da eficácia de ações estratégicas na tentativa de implementação de um combustível alternativo, com o intuito de apoiar a penetração do GNV na Suíça. Em [16], Collantes, G. e Melaina, M. W. analisam a evolução conjunta de infra-estrutura de um combustível e do número de veículos movidos a esse combustível, o caso do GNV na Argentina é analisado como um caso de sucesso com o objetivo de criar-se lições para a implementação de hidrogênio nos USA.

Em [17], Flynn, P.C. analisa o caso mau sucedido do Canadá na adoção de GNV, apontando como principal fraqueza a falta de infra-estrutura, principalmente de abastecimento; novamente, os problemas encontrados são utilizados como lições para implementação de combustíveis alternativos no Canadá e nos USA. Em [19], Di Pascoli, S. et al avaliam o GNV como fonte limpa de energia, e tratam sobre a não difusão do Gás Natural como combustível, apontando não só a falta de infra-estrutura mas também o desconhecimento das vantagens do GNV como principais causas para a não difusão desse combustível, sugerindo por fim a implementação de uma campanha promocional para o uso de combustíveis mais verdes. Em [24], Thamsiroj, T. et al sugerem um roteiro para introdução de combustíveis gasosos no transporte, utilizando o caso de uso de gás natural renovável na Irlanda.

O GNV é também apontado extensivamente como alternativa ao uso de diesel, com motivação principalmente no caráter não-renovável do diesel e na alta emissão de poluentes resultante do seu uso, por diversos artigos. Em [8], Ramos, D.B. apresenta aspectos de um projeto de uma central eletrônica para uso de GNV em motores de ignição por compressão (diesel) utilizando lógica nebulosa; em [10], Oliveira, F.B. et al dissertam sobre o desenvolvimento de um *kit* de conversão para uso de GNV em motores diesel estacionários. Em [22], Carluciet, A.P. et al apresentam os resultados experimentais (e sua análise) de um motor a combustão movido à diesel e GNV simultaneamente.

Por outro lado, pesquisas envolvendo injeção eletrônica de combustível também são temas comuns de artigos e produções científicas. Além do já citado Ramos, D.B. (em [8]), em [7], Carvalho, T.C.O. trata sobre o desenvolvimento de uma central eletrônica aberta, incluindo uma plataforma com uma IHM (Interface Homem Máquina) para configuração dos parâmetros. Em [9], Veiga, M.R., apresenta o uma central eletrônica própria para injeção de GNV juntamente com álcool, com objetivo de aumentar-se a eficiência volumétrica e, conseqüentemente, a potência do motor movido à GNV. Em [18], Burger, P.R.B et al desenvolvem um trabalho muito parecido com Veiga, M.R., mas com o uso de centrais eletrônicas da Motec® para gerência do tempo de injeção de GNV e de álcool. Em [20], Lino, P. et al apresentam um modelo para predição da pressão na flauta de distribuição do combustível, permitindo, com isso, a criação de um sistema de controle de injeção do GNV.

Existe ainda um número significativo de pesquisas que investigam o uso de inteligência artificial em sistemas de gerenciamento de motores a combustão, a mais notável a pesquisa feita por Kalogirou, S.A. em [27], onde é feita uma

revisão sobre a utilização de inteligência artificial para modelamento e controle do processo de combustão. Outra pesquisa importante é a de Ramos, D.B. (em [8]), já citada como um caso de pesquisa sobre injeção de GNV em motores a diesel e como um caso de projeto de central eletrônica de combustível, que utilizou lógica nebulosa no projeto da central eletrônica. Em [11], Barghi, F. e Safavi A.A. falam sobre o desenvolvimento de um sistema com utilização de redes Neuro-fuzzy recorrentes para controle de injeção de combustível (CNG) em motores de ignição por centelha. Em [25], Huang, S.R. et al apresentam resultados experimentais de um sistema que se utiliza de lógica nebulosa para gerência de motores movidos a hidrogênio. Em [26], Kheir, N.A. et al descrevem a utilização de lógica nebulosa para controlar a regulagem do motor de forma a dar ênfase ou na economia de combustível ou no controle da emissão de poluentes em veículos híbridos.

Ainda relacionado ao tema abordado nesta dissertação estão artigos que apresentam dados comparativos relacionados ao uso de GNV em detrimento aos combustíveis convencionais. Em [15], Mello, P. et al apresentam resultados de ensaios sobre a potência máxima de veículos convertidos para uso de GNV com uso de kits de 3ª geração, apontando uma perda de potência na ordem de 20% e apontando os motivos principais como: (1) baixa eficiência térmica, (2) baixa eficiência volumétrica e (3) arrasto criado pelo misturador. Adicionalmente, o tema da redução da emissão de poluentes é analisado, principalmente hidro-carbonetos e monóxido de carbono (HC e CO). Em [21], Kalam, M.A. e Masjuki, H.H. apresentam resultados experimentais de um novo motor de injeção direta de GNV, criado a partir da modificação de um carro com injeção indireta, comprando o desempenho do motor utilizando injeção direta de GNV ao desempenho do mesmo motor movido à gasolina ou com injeção indireta de GNV. Em [23], Aslam, M.U et al fazem uma investigação sobre o uso de GNV em um veículo movido à gasolina, fazendo uma análise comparativa entre o GNV e a gasolina, com resultados apontando maior eficiência e menor emissão de CO, CO<sub>2</sub> e HC a favor do GNV, mas uma emissão maior de NO<sub>x</sub> em contrapartida.

A grande quantidade e diversidade de abordagens dos artigos relacionados ao uso de GNV, ou seja, do gás natural como combustível, reflete não só uma preocupação em se criar alternativas mais econômicas de combustível ou em se aproveitar o enorme potencial que esse combustível possui, mas também, e talvez principalmente, uma crescente preocupação ambiental e uma consciência da enorme contribuição dos automóveis nas alterações climáticas. Em adição, a complexidade de sistemas de injeção de combustível freqüentemente conduzem à utilização de técnicas de inteligência artificial como ferramenta.

## 2.2 INJEÇÃO E IGNIÇÃO DE MOTORES DE CICLO OTTO

O motor de ciclo Otto é composto por cilindros dentro dos quais se movimentam pistões. A combustão da mistura ar-combustível acontece dentro desses cilindros, aumentando a pressão interna, sendo a responsável pela movimentação do pistão dentro do cilindro, que se dá entre o **Ponto Morto Superior (PMS)** e o **Ponto Morto Inferior (PMI)**, o que promove a variação de seu volume interno. O PMI é a posição mais baixa que o pistão pode atingir dentro do cilindro (na qual o volume do cilindro é máximo), e o PMS é o oposto do PMI, sendo a posição mais alta que o pistão pode atingir dentro do cilindro (na qual o volume do cilindro é mínimo).

Cada cilindro é dotado de uma válvula de admissão, por onde entra a mistura ar-combustível, e uma válvula de exaustão (ou escape), por onde saem os gases resultantes da queima. Para iniciar a queima dentro da câmara produz-se uma centelha elétrica (faísca), através de uma vela de ignição.

O ciclo Otto consiste em 4 (quatro) etapas:

1. **Aspiração:** a válvula de admissão é aberta e a mistura ar-combustível é aspirada. A mistura ar-combustível é formada pelo ar admitido pela abertura da borboleta (que por sua vez é controlada pelo pedal de aceleração) misturado ao combustível dosado pelo sistema de injeção (eletrônica ou carburada). Inicia-se no PMS e termina no PMI.
2. **Compressão:** a mistura ar-combustível é comprimida dentro do cilindro com as duas válvulas fechadas. No fim dessa etapa, antes que se atinja o PMS, a centelha é produzida (processo de ignição). Inicia-se no PMI e termina no PMS.
3. **Expansão:** devido à alta pressão dentro do cilindro, causada pela combustão da mistura ar-combustível e pelo fato de as válvulas estarem fechadas, o pistão desloca-se, aumentando o volume ocupado pela mistura. Inicia-se no PMS e termina no PMI.
4. **Exaustão:** a válvula de escape se abre e o resultado da queima é expulso do cilindro pelo movimento de subida do pistão, iniciando um novo ciclo. Inicia-se no PMI e termina no PMS

A Figura 3 [28] ilustra essas 4 (quatro) etapas.

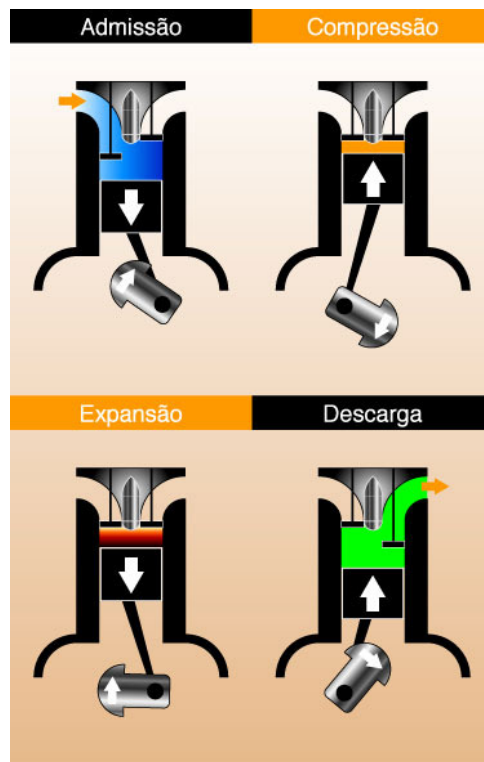


Figura 3: Etapas do ciclo OTTO [28]

No funcionamento dos motores de ciclo OTTO, existem duas importantes variáveis que devem ser controladas:

1. A quantidade de combustível dosada na mistura – a relação estequiométrica de ar-combustível deve ser obedecida para que a reação química de combustão produza somente  $\text{CO}_2$  e  $\text{H}_2\text{O}$  e que a quantidade máxima de energia seja liberada no processo. Quando a mistura de ar-combustível é ideal, todo carbono (C) é consumido, formando gás carbônico ( $\text{CO}_2$ ), todo hidrogênio (H) é consumido, formando vapor d'água ( $\text{H}_2\text{O}$ ) e todo nitrogênio ( $\text{N}_2$ ) presente na reação se mantém inalterado. Nessas condições, diz-se que a mistura ar-combustível é estequiométrica [29].
2. O instante de produção da centelha (ignição) – o processo de ignição se dá pela produção da centelha que inicia o processo de combustão dentro de cada cilindro. Devido ao fato de que a combustão da mistura ar-combustível leva um tempo para se completar, a centelha deve ser produzida pouco antes do PMS de tal maneira que essa combustão esteja completa quando o cilindro atingir o PMS. A distância do ponto em que a centelha foi produzida e o PMS é medida em graus e é chamada de **Avanço de Ignição** e seu valor ideal varia em função das condições de funcionamento do motor e da velocidade de chama do combustível utilizado. O **Avanço de Ignição** pode ser calculado pela Equação ( 1 ).

$$\text{Avanço de Ignição} = \frac{\text{Tempo entre ignição e PMS}}{\text{Período de rotação do Motor}} * 360^\circ \quad (1)$$

Inicialmente essas variáveis eram controladas por sistemas eletro-mecânicos, com o uso de carburadores para dosagem de combustível. Porém, com a crescente exigência de baixo consumo de combustível e menor emissão de poluentes, o sistema de injeção e ignição atual é eletrônico.

### 2.3 ASPECTOS TECNOLÓGICOS DA INJEÇÃO E IGNIÇÃO ELETRÔNICAS

A injeção eletrônica é um sistema não acionado pelo motor, comandado eletronicamente e que dosa a quantidade de combustível a ser injetado, controlando a mistura-ar-combustível em função das necessidades imediatas do motor. De modo semelhante, a ignição eletrônica permite que o motor trabalhe com o seu ponto de ignição sincronizado com as diversas condições de funcionamento do motor [29].

Existem 2 (dois) sistemas de injeção em motores de ciclo OTTO. O sistema *monoponto* e o sistema *multiponto*. O sistema *monoponto* (Figura 4) é caracterizado pela existência de um único bico injetor para todos os cilindros. Esse bico injetor fica localizado junto ao corpo de borboleta e é responsável pela injeção de combustível para todos os cilindros. Já no sistema *multiponto* (Figura 5), existe um bico injetor para cada um dos cilindros. Esses bicos ficam dispostos logo antes da válvula de admissão [31].

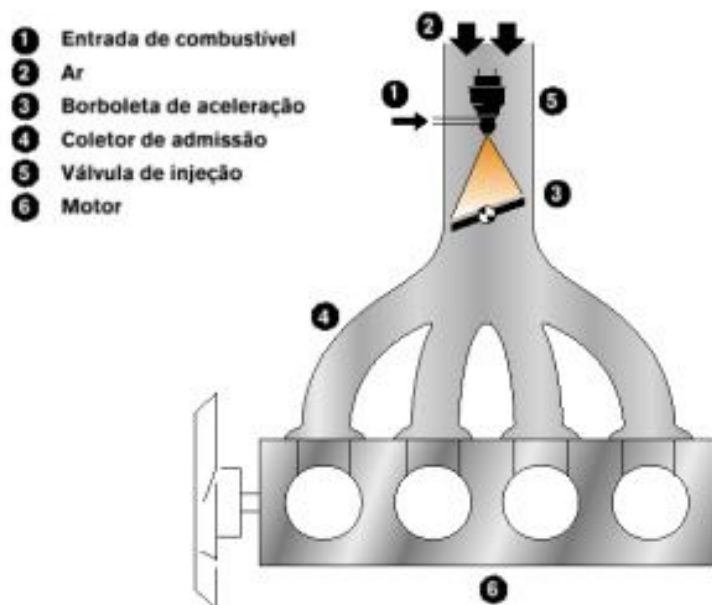


Figura 4: Sistema de injeção monoponto [31]

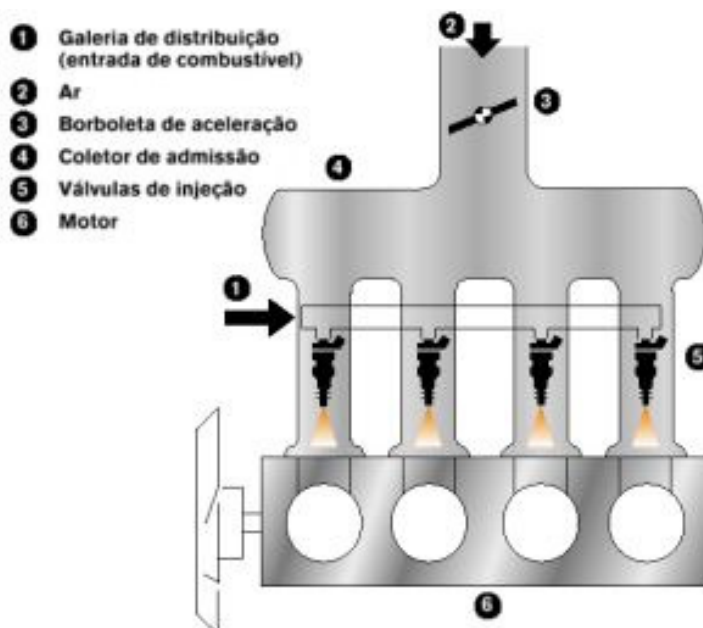


Figura 5: Sistema de injeção multiponto [31]

Em ambos os sistemas (*monoponto* ou *multiponto*), a quantidade de combustível injetada no cilindro é determinada pelo tempo de abertura da válvula de injeção (medida geralmente em milissegundos). Essa quantidade de combustível a ser injetada deve ser determinada de tal forma que a mistura ar-combustível seja estequiométrica.

De forma resumida, os sistemas de injeção e ignição eletrônica utilizam-se das leituras de alguns sensores dispostos no carro para realizar o cálculo do tempo necessário de abertura do bico eletro-injetor de combustível e do instante adequado à produção da centelha (avanço de ignição), respectivamente.

Os sistemas de injeção/ignição eletrônica são compostos de sensores, atuadores e uma UCE (Unidade de Comando Eletrônico). Em uma abordagem simplificada, a UCE é responsável por ler os sensores, fazer os cálculos de tempo de injeção e avanço de ignição e comandar os atuadores.

### 2.3.1 Os principais Sensores de um sistema de injeção/ignição eletrônica

Os sensores são responsáveis por disponibilizar à UCE as informações necessárias ao cálculo da quantidade de combustível que deve ser injetada e do avanço de ignição.

Seguem exemplos de sensores de um sistema de injeção/ignição eletrônica:

- a) **Sensor de temperatura da água do motor:** indica a temperatura da água de resfriamento do motor, e conseqüentemente a temperatura aproximada do bloco do motor.
- b) **Sensor de temperatura do ar aspirado:** indica a temperatura do ar a ser utilizado na mistura ar-combustível. É importante porque a temperatura está diretamente ligada à densidade do ar e, conseqüentemente, à massa de ar injetado.
- c) **Sensor de rotação do motor:** indica a frequência de rotação do motor. É uma variável bastante importante e é utilizada como entrada na determinação de tempo de injeção e no avanço de ignição.
- d) **Sensor de grau de abertura da borboleta:** indica o quanto a borboleta está aberta, ou seja, o quanto o motorista está pisando no acelerador. A borboleta (ligada ao acelerador) determina a quantidade de ar que entra na câmara de admissão. Sobre esse ar o(s) bico(s) injetor(es) jogar(á)(ão) o combustível para formar a mistura ar-combustível. É um dos principais sinais para determinar a quantidade de combustível a ser injetada.
- e) **Sensor de pressão do coletor:** também conhecido como sensor MAP (*Manifold Air Pressure*, ou Pressão Absoluta no Coletor), indica a pressão no coletor de ar. Essa informação é utilizada juntamente com a informação do sensor de temperatura do ar aspirado (b) para determinar a massa de ar admitida. Um ponto importante é que os sistemas de injeção precisam determinar a vazão mássica e não a vazão volumétrica de ar. Para executar isto existem diferentes métodos diretos ou indiretos. Os métodos diretos usam sensores de fluxo mássico baseados em sensores de fio/fita quente [30]. Os métodos indiretos usam a informação de um sensor de pressão e um sensor de temperatura do ar na câmara de admissão para determinar a massa de ar entrante.
- f) **Sonda lambda:** indica a quantidade de oxigênio nos gases de escape, informando à UCE se a mistura está muito rica em combustível ou muito pobre. Dessa forma a UCE pode ir corrigindo a quantidade de combustível injetada durante seu funcionamento. É importante observar que a relação mássica de ar-combustível ideal é determinada para cada combustível. No caso de gasolina a relação ideal da mistura está diretamente relacionada com a tentativa de maximizar o rendimento do motor (potência transmitida para o carro) e minimizar a emissão de poluentes, por exemplo: hidrocarbonetos ( $HxCy$ ), óxidos de nitrogênio ( $NOx$ ) e monóxido de carbono ( $CO$ ).

### 2.3.2 Principais atuadores controlados por um sistema de injeção/ignição eletrônica

Os atuadores são responsáveis por controlar o processo fisicamente a partir de sinais da UCE. Os principais atuadores utilizados em um sistema de injeção/ignição eletrônica são:

- a) **Bico eletro-injetor:** espirra combustível para realizar a mistura com o ar. O tempo em que fica aberto determina a quantidade de combustível da mistura.
- b) **Velas de ignição:** produzem a faísca que inicia o processo de combustão da mistura ar-combustível.

Esses atuadores são descritos com mais detalhes abaixo.

#### • O BICO ELETRO-INJETOR

O bico eletro-injetor é uma eletroválvula normalmente fechada (NF), comandada pela UCE que promove sua abertura energizando-a. O fechamento ocorre pela ação de uma mola interna quando a eletroválvula (bico eletro-injetor) é desenergizada [29]. As eletroválvulas são baseadas em um magneto que é atraído para o núcleo de um solenóide quando esse é percorrido por uma corrente elétrica.

A Figura 6 mostra uma vista em corte de um bico injetor, onde:

1. Filtro é um filtro de combustível para evitar seu entupimento;
2. Bobina é o solenóide que irá atrair o magneto quando energizado;
3. Armadura do Solenoide é o magneto atraído pelo solenóide quando esse é energizado;
4. Agulha é a peça mecânica que bloqueia a passagem de combustível, e que recolhe (juntamente com o magneto) quando o solenoide é energizado, permitindo a passagem do combustível;
5. Espigão é a abertura pela qual passa o combustível quando a agulha recolhe;



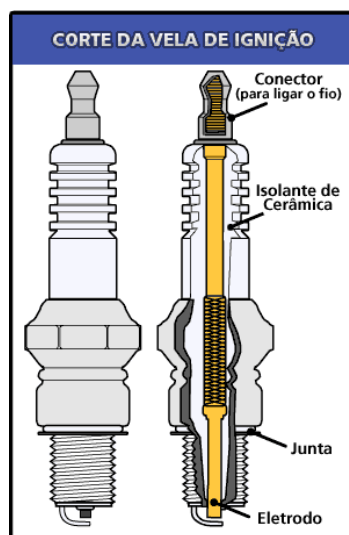
**Figura 6: Vista em corte de um bico injetor [32].**

Na prática, o solenóide é conectado à tensão da bateria de um lado e à UCE do outro, de forma que quando a UCE deseja comandar a abertura do bico injetor cria-se um caminho de corrente para o terra (GND), a bobina é energizada, a agulha recolhe e há injeção de combustível, e em condição de repouso, quando não há injeção de combustível, a UCE deixa seu terminal em alta impedância, nenhuma corrente flui pelo bico injetor e não há injeção de combustível. Uma forma comum de fazer esse tratamento é através do uso de MOSFETs (*Metal Oxide Semiconductor Field Effect Transistor*).

### • AS VELAS DE IGNIÇÃO

As velas de ignição são responsáveis por criar a centelha que inicia o processo de combustão. Essa centelha é criada a partir de uma altíssima diferença de tensão elétrica entre dois terminais separados por uma pequena distância entre si. A Figura 7 traz uma vista em corte de uma vela de ignição, onde:

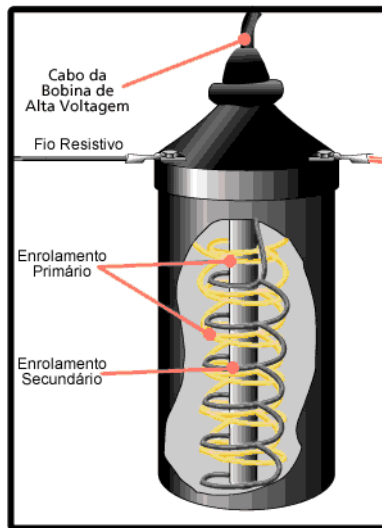
1. Conector, de onde vem a energia que gera a centelha;
2. Isolante de Cerâmica, que promove o isolamento térmico e elétrico;
3. Eletrodo, que conduz a energia elétrica até o local onde a centelha é produzida;
4. Junta, que conduz a energia que passa do eletrodo central ao eletrodo lateral de volta ao terra (GND).



**Figura 7: Vista em corte de uma vela de ignição [33].**

Para produção da tensão necessária à produção de centelha, é utilizada uma bobina - Figura 8. A bobina consiste de um enrolamento primário e um enrolamento secundário, e em sistemas de corrente alternada a tensão no primário é multiplicada no secundário segundo a relação de número de voltas em cada uma das bobinas, conforme a Equação (2).

$$V_{\text{secundário}} = V_{\text{primário}} * \frac{\text{Nro de Voltas}_{\text{secundário}}}{\text{Nro de Voltas}_{\text{primário}}} \quad (2)$$



**Figura 8: Vista em corte de uma bobina de ignição [33].**

Essa relação só é verdadeira em sistemas de corrente alternada pois é a variação de corrente no primário que provoca uma variação de campo magnético no núcleo que finalmente gera uma variação de corrente no secundário. Podemos dizer, portanto, que apenas uma variação de corrente no primário vai surtir efeito no secundário.

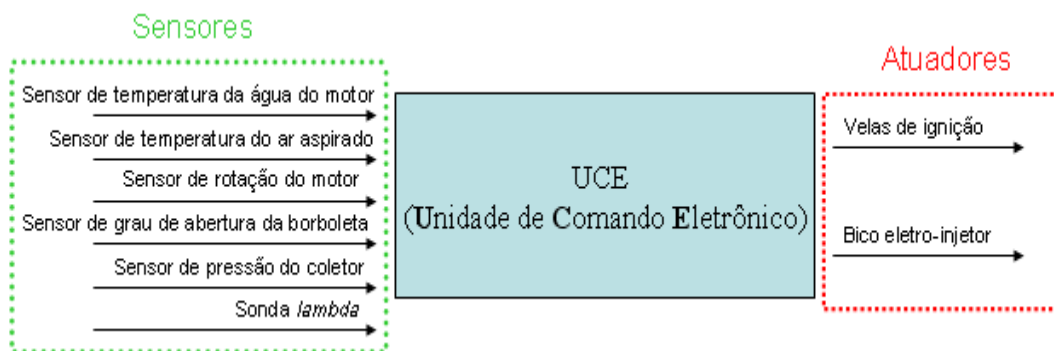
A produção de centelha acontece da seguinte maneira:

1. O primário tem um de seus terminais conectados à tensão da bateria e o outro à UCE, que inicialmente o deixa em alta impedância, impedindo a circulação de corrente;
2. Alguns instantes antes do momento de produção da centelha, a UCE conecta o terminal da bobina ao terra, criando um caminho para a corrente circular e carregando o primário;
3. Depois que uma corrente suficiente passa a circular pelo primário, a UCE interrompe o caminho de corrente abruptamente, de forma que essa variação cria a tensão necessária à produção da centelha no secundário.

Na prática, a tarefa de permitir e interromper o fluxo da corrente elétrica, tal qual no acionamento de bicos injetores, também é feita a com o uso de MOSFETs (*Metal Oxide Semiconductor Field Effect Transistor*).

### 2.3.3 A UCE (Unidade de Comando Eletrônico)

A UCE é responsável por controlar o funcionamento do motor, usando os sensores para obter a condição de funcionamento e os atuadores para modificar esse funcionamento. As duas saídas principais do controle da UCE são os sinais que controlam o tempo de injeção de combustível e o avanço da ignição. A Figura 9 ilustra um diagrama de blocos de um sistema de injeção/ignição eletrônica.



**Figura 9: Diagrama de blocos de um sistema de injeção/ignição eletrônica**

Como exemplos de fabricantes e modelos de UCE's podemos citar os seguintes: Bosch (Bosch Motronic, Bosch Le Jetronic, Bosch Monomotronic); Magneti Marelli (Magneti Marelli IAW, Magneti Marelli 1AVB); FIC (FIC EEC-IV, FIC EEC-V); Delphi (Delphi Multec).

### 2.3.4 Relação ar-combustível e os principais gases emitidos por motores a combustão

Como dito na seção 2.2, uma mistura é dita estequiométrica quando produz apenas  $H_2O$  e  $CO_2$ . A relação entre a massa de ar e a massa de combustível admitido é o que chamamos de relação-ar-combustível (RAC), e varia de combustível para combustível [29]. As principais relações estequiométricas podem ser encontradas na Tabela 1.

**Tabela 1: Relações estequiométricas dos principais combustíveis [34].**

Combustível	Proporção por Massa	Proporção por Volume	Porcentagem de combustível por massa
Gasolina	14,7 : 1	-	6,8%
Gás Natural	17,2 : 1	9,7 : 1	5,8%
Propano	15,5 : 1	23,9 : 1	6,45%
Etanol	9 : 1	-	11,1%
Metanol	6,4 : 1	-	15,6%
Hidrogênio	34 : 1	2,39 : 1	2,9%
Disel	14,6 : 1	0,094 : 1	6,8%

Quando há na mistura mais ar do que o ideal, a mistura é dita pobre (em combustível), no caso contrário, mais combustível que o ideal, a mistura é dita rica. A variação da RAC para uma relação não estequiométrica (tanto para mais rica quanto para mais pobre) causa a emissão de gases poluentes.

Segundo VON GLEHN, F. R. [29], os principais gases de emissão são exibidos na Tabela 2.

**Tabela 2: Principais gases poluentes**

$H_2O$	vapor de água
$CO_2$	dióxido de carbono
$N_2$	nitrogênio
$CO$	monóxido de carbono
$H_xC_y$	hidrocarbonetos
$NO_x$	óxido de nitrogênio, dióxido... etc.
$H_2$	hidrogênio
$CH_4$	metano
$SO_x$	óxido de enxofre, dióxido... etc.
$O_2$	oxigênio

Por ser o gás mais preocupante e agressivo, apenas o  $CO$  será abordado nessa dissertação, sendo um gás levemente inflamável, incolor, inodoro e muito perigoso devido à sua grande toxicidade.

O Monóxido de Carbono (CO) forma com a hemoglobina do sangue um composto mais estável do que ela e o oxigênio, podendo levar à morte por asfixia. A exposição a doses relativamente elevadas em pessoas saudáveis pode provocar problemas de visão, redução da capacidade de trabalho, redução da destreza manual, diminuição da capacidade de aprendizagem, dificuldade na resolução de tarefas complexas e até mesmo levar a morte. Concentrações abaixo de 400 ppm no ar causam dores de cabeça e acima deste valor são potencialmente mortais, tanto para plantas e animais quanto para alguns microrganismos [35].

A produção de Monóxido de Carbono é aumentada principalmente por dois fatores: uma RAC rica (muito combustível) e uma não homogeneidade da mistura. Em ambos os casos o que se dá é a falta da molécula de oxigênio na reação química, fazendo com que o produto da combustão seja o CO no lugar do CO<sub>2</sub>. Uma descrição mais abrangente de gases poluentes produtos da combustão de motores do ciclo Otto pode ser encontrada em VON GLEHN, F. R. [29].

## **2.4 GNV E CONSIDERAÇÕES EM RELAÇÃO AO SEU USO**

O Gás Natural é um dos resultados da decomposição de matéria sedimentada. É encontrado em rochas porosas no subsolo. Segundo trabalho publicado no Prêmio Petrobrás de Tecnologia em maio de 2007 [36], o Gás Natural é composto por gases inorgânicos e hidrocarbonetos saturados, com predominância do metano, e, em menor grau, o etano e o propano.

Devido às suas propriedades químicas, o Gás Natural é uma boa alternativa na substituição de combustíveis convencionais nos motores de ciclo OTTO com ignição por centelhamento. Quando usado em veículos, recebe o nome de GNV (Gás Natural Veicular).

### **2.4.1 Um breve histórico do GNV no Brasil [37] [38]**

A história do GNV no Brasil é abordada pois considera-se importante para compreender as tendências dessa alternativa de combustível nos dias atuais. Enumerados abaixo, estão importantes eventos na história do GNV no Brasil:

- 1854 – Começa a história do gás no Brasil, com a instalação das primeiras lâmpadas a gás no Rio de Janeiro. O gás ainda era produzido a partir do carvão mineral e mais tarde de materiais como hulha e nafta.
- 1873 – Instalação do mesmo sistema de lâmpadas a gás em São Paulo.
- 1936 – O GLP (Gás Liquefeito do Petróleo) começou a ser usado para cocção.
- Década de 1950 – O gás natural começa a ser usado no nordeste, com a produção inicialmente no estado da Bahia e utilizado praticamente todo pelas indústrias.
- 1959 – A Bahia chega a uma produção de 1 milhão de m<sup>3</sup>/dia, atingindo a marca de 3.3 milhões de m<sup>3</sup>/dia uma década depois [39].
- Anos 1980 – O gás natural passa a ser produzido na Bacia de Campos, no Rio de Janeiro; Construção do gasoduto Bolívia Brasil (Gasbol), consolidando a atual situação de oferta de gás. Devido ao fato do mercado interno de gás natural não ter se desenvolvido como se esperava, fato ocasionado por fatores como: a falta de um mercado maduro e a necessidade de recuperação do capital investido no gasoduto, o Brasil adotou medidas emergenciais e pouco eficazes;

Ainda nessa época, começam as primeiras discussões sobre o uso de GNV como combustível, com a criação de Comissões Governamentais para estudar-se o uso de GNV no lugar de óleo diesel tanto no transporte de cargas quanto no de passageiros. No entanto, o GNV não apresentou vantagens econômicas que justificassem o alto custo da conversão de frotas e a criação de Postos de Serviço.

- 1992 a 1994 – O GNV é liberado para taxistas e frotas de empresa e com isso cria-se primeiro Posto de Serviço, visando a venda do combustível para o público. Com o aumento da demanda, o mercado fica desabastecido de GNV.
- 1994 a 1996 – Dois fatores influenciaram a demanda por GNV em sentidos contrários, um contra e um a favor. Com o Plano Real veio a estabilização dos preços dos combustíveis, e conseqüentemente a impressão de que o GNV não trazia tanta economia como antes, mas em contrapartida foi dada aos taxistas uma isenção de impostos que gerou grande renovação da frota.
- 1997 – O GNV é liberado para uso em carros particulares.
- 24/02/2000 – O modelo de contrato adotado entre o Brasil e a Bolívia obriga o comprador (Brasil) a pagar um percentual sobre o gás contratado caso não consuma o valor estipulado em contrato, e é conhecido

como *take or pay*. Por esse motivo, o Brasil criou nesse ano, por meio do decreto 3.371 do Ministério de Minas e Energia [40], um programa que previa a construção imediata de termoeletricas movidas a Gás Natural – o Programa Prioritário de Termoeletricidade (PTT). Pode-se considerar que esse programa não tenha sido bem sucedido, já que a potência instalada não foi suficiente para evitar o chamado “apagão” em 2001, e incluem-se como fatores para esse fracasso: deficiências regulatórias e alto custo de produção frente à hidroeletricidade.

- Década 2000 – As reservas brasileiras aumentam consideravelmente com a descoberta de petróleo e gás nas camadas do pré-sal. Como informação, em 2000 as reservas provadas do país eram de 220.000 milhões de m<sup>3</sup> e passaram para 364.263 milhões de m<sup>3</sup> em 2008 segundo dados do Balanço Energético Nacional 2009, publicado pela Empresa de Pesquisa em Energia (EPE) do Ministério de Minas e Energia (MME) [41].

Seguindo o aumento das reservas, o Brasil passou a consumir mais gás natural, que alcançou uma participação de 9,29% em relação às outras fontes de energia. Mesmo assim, o Brasil ainda consome proporcionalmente muito menos gás natural que o mundo, 7% contra 21%, como fica evidenciado pela Figura 10.

Consumo de Energia Primária por tipo de Energético

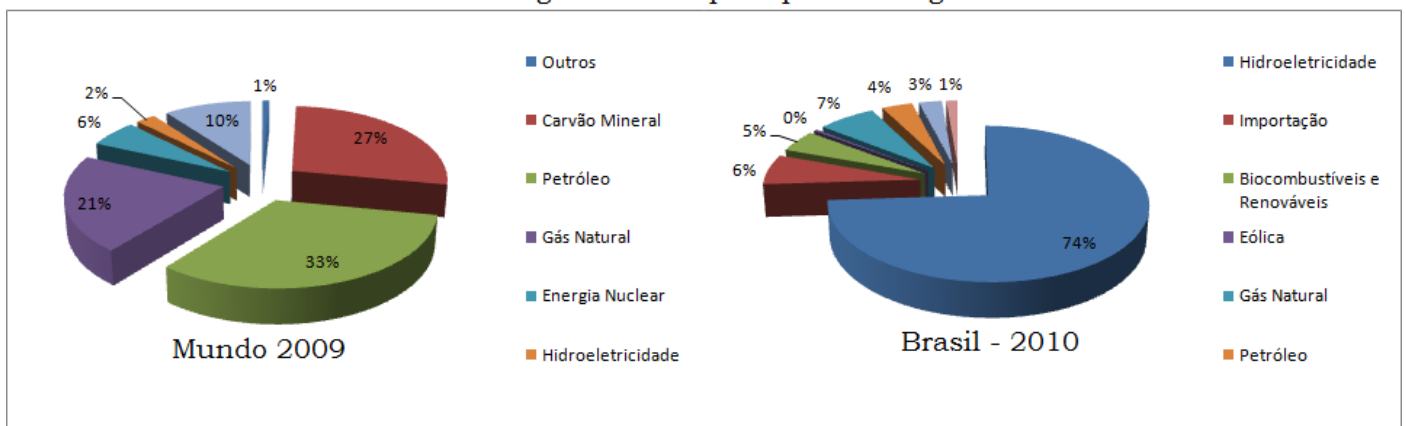


Figura 10: Consumo de Energia Primária por Tipo de Energético no mundo – [42] – e no Brasil – [43].

- Tendências futuras – Prevê-se que o Brasil chegue a uma produção nacional de gás natural de 250 milhões de m<sup>3</sup>/dia até 2030, e que a indústria permaneça como principal consumidora desse combustível. A participação do gás natural na geração de energia elétrica deve aumentar também, atingindo cerca de 15% no mesmo ano [44].

As tecnologias de conversão para GNV minimizam cada vez mais os impactos da substituição dos combustíveis originais, gasolina e álcool. O preço desses combustíveis vem apresentando, ao mesmo tempo, um crescimento proporcionalmente ao GNV, conforme Figura 11 extraída do Instituto Brasileiro de Petróleo, Gás e Biocombustíveis [1], que faz com que o retorno no investimento da conversão se dê em menos tempo.

Esses dois cenários somados tendem a reverter a atual tendência de queda das conversões anuais mostrada na Figura 1 da seção 1.1.

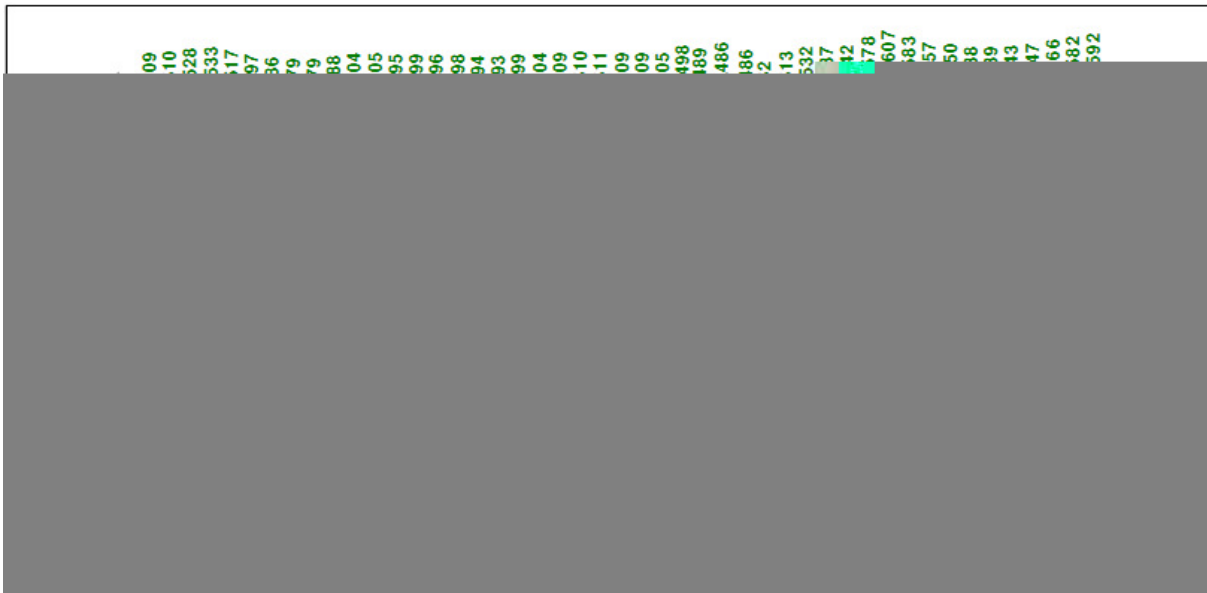


Figura 11: Gráfico dos preços dos combustíveis e do N° de conversões [1]

### 2.4.2 Modificações requeridas para o uso de GNV

Para converter carros movidos à gasolina ou álcool para o uso de GNV, algumas modificações devem ser feitas no carro.

A Figura 12 ilustra os componentes necessários à conversão [45]. Os componentes que aparecem nessa figura são:

1. **Redutor de pressão:** Reduz a pressão do gás do cilindro para a pressão normal de funcionamento.
2. **Eletroválvula:** Confere segurança extra, já que uma vez comutada para o combustível líquido, a válvula interrompe o fluxo de gás imediatamente.
3. **Bicos injetores de Gás:** Promovem a injeção de gás de forma seqüencial.
4. **Módulo de controle:** Realiza um controle do tempo de injeção preciso, lendo os sensores e agindo sobre os bicos injetores, o qual se pretende desenvolver neste trabalho de mestrado.
5. **Válvula de abastecimento de gás:** Válvula para abastecimento do veículo.
6. **Indicador de nível/chave comutadora:** Botão seletor de combustível e indicador de quantidade de gás no cilindro.
7. **Variador de avanço:** Promove a variação de avanço.



Figura 12: Modificações em um carro convertido para uso de GNV [45]

No que diz respeito ao controle dos sistemas de injeção e ignição, também existem modificações a serem feitas:

- **Sinal de acionamento do sistema de injeção:** dois fatores principais são diferentes quando se compara o GNV ao combustível usado pelo sistema de injeção eletrônica original. O primeiro diz respeito à relação estequiométrica do combustível. Enquanto a relação é de 13,4Kg de ar para cada 1Kg de gasolina e de 8Kg de ar para cada 1Kg de álcool, no caso do GNV são necessários 17Kg de ar para reagir com 1Kg de gás [46]. O segundo fator é relacionado com a vazão de gás do sistema de injeção, que varia de acordo com o *kit* instalado. Esses dois fatores fazem com que o tempo de acionamento do sistema de injeção do *kit* não seja exatamente igual ao da gasolina, e por isso tenha que ser recalculado.
- **Sinal de acionamento da vela de ignição:** o GNV tem uma velocidade de chama menor do que a gasolina e o álcool (0,45 m/s do GNV contra 0,8 m/s da gasolina e 0,6 m/s do álcool) [46]. Isso implica na necessidade de um maior avanço de ignição quando o motor estiver operando com GNV.

### 2.4.3 Kits disponíveis no mercado

Segundo um dossiê técnico elaborado pelo Serviço Brasileiro de Respostas Técnicas (BRT) em abril de 2007 [47], existem atualmente no mercado 5 gerações de *kits* de conversão:

- Primeira geração:** Utiliza acionamento pneumático para a liberação do fluxo de gás e regulagem mecânica e manual da vazão do gás, com chave comutadora de três estágios. Usado em veículos carburados.
- Segunda geração:** Utiliza-se de um mesclador para alimentação do gás. Possuem emuladores de bicos injetores e de sonda lambda e são usados em carros dotados de injeção eletrônica.
- Terceira geração:** Utiliza um motor de passo para promover a alimentação de GNV. O controle da vazão da mistura ar-GNV é feita eletronicamente em função da sonda lambda, rotação e carga do motor. São usados em carros com injeção eletrônica.
- Quarta geração:** A injeção de gás é feita por bicos de forma paralela no coletor de admissão, com redutor de pressão de dois estágios. Dessa forma, não existe a ocorrência de retorno de chama, e o desempenho é melhorado. Também usados em carros com injeção eletrônica.
- Quinta geração:** Os bicos injetores são acionados de forma sequencial, injetando gás no coletor de admissão, com redutor de pressão de dois estágios. Diminui a emissão de poluentes e a perda de potência, além de manter a eliminação da ocorrência do retorno da chama.

## 2.4.4 Vantagens do GNV [48]

Quando comparado à gasolina e ao álcool, o GNV apresenta os seguintes atrativos:

a) É uma fonte segura, pelos seguintes motivos:

- **Menor densidade que o ar atmosférico:** significa que o gás natural não irá se acumular em regiões inferiores, sendo suficiente a existência de sistemas de ventilação para permitir a sua dissipação.
- **Não toxicidade:** sua ingestão ou inalação acidental não provoca danos à saúde, pois as substâncias componentes do gás natural são inertes no corpo humano.
- **Alto limite de inflamabilidade inferior:** para que se atinjam as condições de auto-sustentação da combustão se faz necessária uma quantidade significativa de gás natural em relação à quantidade total de ar em um ambiente.
- **Faixa entre os limites de inflamabilidade inferior e superior é estreita:** o que significa dizer que além de ser difícil alcançar o limite inferior de inflamabilidade, caso isso ocorra a condição de diluição da mistura ar-gás natural que permite a auto-sustentação da é rapidamente perdida, pois logo se atinge o limite superior de inflamabilidade e o gás natural torna-se diluente do ar.

b) Causa menos impacto global, pois contribui menos para o aquecimento global: o uso de GNV tem importante papel na redução dos níveis de poluição atmosférica, uma vez que a sua combustão com excesso de ar tende a ser completa, liberando apenas dióxido de carbono (CO<sub>2</sub>) e água (H<sub>2</sub>O) [49].

c) Menor custo que a gasolina e o álcool: a relação R\$/Km chega a ser 50% melhor que a da gasolina (ver seção 1.1).

## 2.4.5 Desvantagens do GNV

Embora o GNV apresente algumas desvantagens em relação à gasolina, tais problemas podem ser minimizados ou até eliminados, como será descrito a seguir.

### 2.4.5.1 PROBLEMAS ENCONTRADOS NA CONVERSÃO PARA GNV

Já existem no mercado alguns automóveis equipados de fábrica para o uso de GNV. Entretanto, a maioria dos automóveis que se utiliza desse combustível ainda é formada por carros convertidos, ou seja, carros originalmente movidos a gasolina, álcool ou ambos (bicombustíveis) e que foram adaptados para o uso de GNV através de *kits* de conversão. O problema é que, em grande parte dos casos, os carros convertidos apresentam alguns (ou todos) dos seguintes problemas:

- Queda de rendimento:** Em alguns *kits*, a conversão utilizada funciona de um modo similar aos antigos carburadores, onde a quantidade de combustível admitida depende diretamente da diferença de pressão causada no coletor de admissão do motor. Devido à necessidade de utilização de elementos restritores nesses *kits* de conversão, os chamados “misturadores”, existe uma menor vazão de mistura gás-ar e, portanto, uma considerável perda de potência.
- Alta emissão de poluentes:** Alguns kits apresentam uma deficiência na homogeneização da mistura ar-GNV, diminuindo a eficiência da queima da mistura, e conseqüentemente gerando alto consumo e alta emissão de poluentes.
- Alto consumo:** Esse fator se deve à deficiência na homogeneização citada no item b) e também à baixa velocidade de combustão do GNV se comparado à gasolina ou ao álcool. Essa baixa velocidade de combustão exige que o kit faça o que se chama de avanço de ignição, ou seja, exige que a faísca seja produzida antes do que seria caso o motor fosse movido à gasolina ou álcool. O problema existe porque muitos dos kits atuais não fazem esse avanço de ignição adequadamente.

### 2.4.5.2 SOLUÇÕES PARA OS PROBLEMAS DE CONVERSÃO PARA GNV

Para melhorar o desempenho, o consumo e a emissão de poluentes, alguns *kits* já apresentam elementos inteligentes, que determinam a quantidade de gás a ser utilizada usando sensores da injeção original (ver seção 2.4.3). Isso permite que o *kit* tenha um maior controle sobre o processo, podendo calcular com maior precisão a quantidade de GNV da mistura e o melhor momento de produzir a centelha de ignição.

Resumidamente, os problemas apontados nessa seção podem ser minimizados, ou até resolvidos, trabalhando nos seguintes aspectos:



- Controle mais preciso da quantidade de GNV a ser utilizado na mistura, usando para isso os sinais dos sensores da UCE da injeção eletrônica original.
- Avanço de ignição mais preciso, usando também os sinais da UCE da injeção eletrônica original.

Em aspectos gerais, o uso de GNV se mostra compensador, sendo mais barato e menos poluente. Os problemas apontados nessa seção podem ser resolvidos em grande parte com as soluções apresentadas também nesta seção.

## 2.5 ASPECTOS BÁSICOS DE LÓGICA NEBULOSA (*FUZZY LOGIC*)

“A lógica sempre foi estudada em Inteligência Artificial como um formalismo para apresentação do conhecimento. A representação obtida é, no entanto, inflexível demais para lidar com conceitos do mundo real sem exigir um nível elevado de simplificação. Na lógica clássica lidamos com um mundo de clareza, rigor e precisão no qual conceitos reais não se encaixam com facilidade.

A habilidade humana para manipular conceitos imprecisos é vista como a principal diferença entre inteligência humana e artificial. Consideremos um paradoxo conhecido desde a grécia antiga: um homem com um fio de cabelo a mais do que um homem calvo é também calvo? Aparentemente, o bom senso no diz que sim. No entanto, esta afirmação em lógica clássica nos força a concluir que todos são calvos. Afinal, se um homem com nenhum fio de cabelo é calvo por definição, e aquele com 1 fio de cabelo a mais continua calvo, então são calvos os homens com 2 fios de cabelo, 3 fios de cabelo, ou qualquer número finito de fios de cabelo. Obviamente, o conceito de calvície não se descreve corretamente com um número máximo de fios de cabelo.

A lógica nebulosa (“*fuzzy logic*”) pretende formalizar a habilidade humana de chegar a decisões razoáveis baseadas em dados imprecisos e qualitativos. Com ela procura-se tratar a imprecisão e o caráter vago dos processos mentais humanos. Um artigo de Zadeh (1965), que propõe uma extensão da teoria clássica de conjuntos, costuma ser citado como ponto de partida das pesquisas sobre lógica nebulosa, ainda que estas relacionem-se com as lógicas multivaloradas, cuja origem é bem mais antiga.” [50]

Na teoria clássica de conjuntos só há duas possibilidades: um elemento pertence ou não a um determinado conjunto, e os limites que determinam essa pertinência são bem definidos. Quando falamos de conjuntos difusos, teoria introduzida por Lotfi Zadeh em 1965, afirma-se que a transição entre “pertencer a um conjunto” e “não pertencer a um conjunto” é gradual [51].

Em um conjunto nebuloso, um elemento possui um grau de pertinência em relação a um conjunto que varia desde a completa exclusão (grau de pertinência = 0) até a pertinência máxima (grau de pertinência = 1). Ou seja, quanto maior o grau de pertinência de um elemento em relação a um conjunto, mais esse elemento pertence a esse conjunto, ou mais compatível é esse elemento com esse grupo [52]. O conceito de conjunto nebuloso é uma generalização do conceito de conjunto da lógica clássica, portanto ele coincidirá com a lógica clássica se determinarmos que os graus de pertinência assumirão apenas dois valores: 0 ou 1.

Um sistema nebuloso (*fuzzy*) é baseado em regras, e funciona da seguinte maneira: as regras são aplicadas considerando os fatos (entradas do sistema ou antecedentes), a aplicação das regras passa por um mecanismo de inferência de onde saem as decisões (saídas do sistema ou consequentes). Essas regras tomam a forma de sentenças lógicas com variáveis lingüísticas (seção 2.5.3), por exemplo:

- Se a temperatura do motor está alta, aumente um pouco o tempo de injeção;
- Se a rotação está baixa e a velocidade está alta, diminua muito o tempo de injeção.

### 2.5.1 Conjunto Nebuloso

Um conjunto nebuloso é caracterizado por sua função de pertinência (vide Equação ( 3 )):

$$\mu: X \rightarrow [0,1] \quad (3)$$

onde  $\mu(x)$  representa o grau de pertinência do elemento  $x$  ao conjunto  $A$  e representa o quanto a categoria definida pelo conjunto  $A$  representa esse elemento. O conjunto  $X$  é todo o universo de valores que  $x$  pode assumir.

Como exemplo, suponha que  $A$  seja um conjunto que defina o conceito “baixa rotação” para os valores de rotação no universo  $X = [0,7000]$  RPM. Em lógica clássica, essa definição tomaria uma forma parecida com a Equação ( 4 ).

$$A = [0,2000]RPM \quad (4)$$

Com isso, valores abaixo de 2000 RPM são considerados como baixa rotação e valores acima disso não. Isso cria distorções do tipo: o valor de 2000 RPM é considerado baixa rotação, mas 2001 RPM não. Um conjunto nebuloso que defina baixa rotação seria definido por uma função de pertinência que poderia assumir uma forma parecida com a Equação ( 5 ).

$$\mu(x) = \begin{cases} 1 & \text{se } x \leq 1000 \\ \frac{3000 - x}{2000} & \text{se } 1000 < x \leq 3000 \\ 0 & \text{se } x > 3000 \end{cases} \quad (5)$$

De acordo com essa definição, valores de rotação abaixo de 1000 RPM são considerados baixa rotação, valores acima de 3000 RPM não são considerados baixa rotação, e valores entre 1000 RPM e 3000 RPM são considerados como baixa rotação em algum grau intermediário entre 0 e 1. Por exemplo, 2000 RPM será considerado 0,5 em “baixa rotação”.

A diferença entre os dois conjuntos pode ser visualizada na Figura 13.

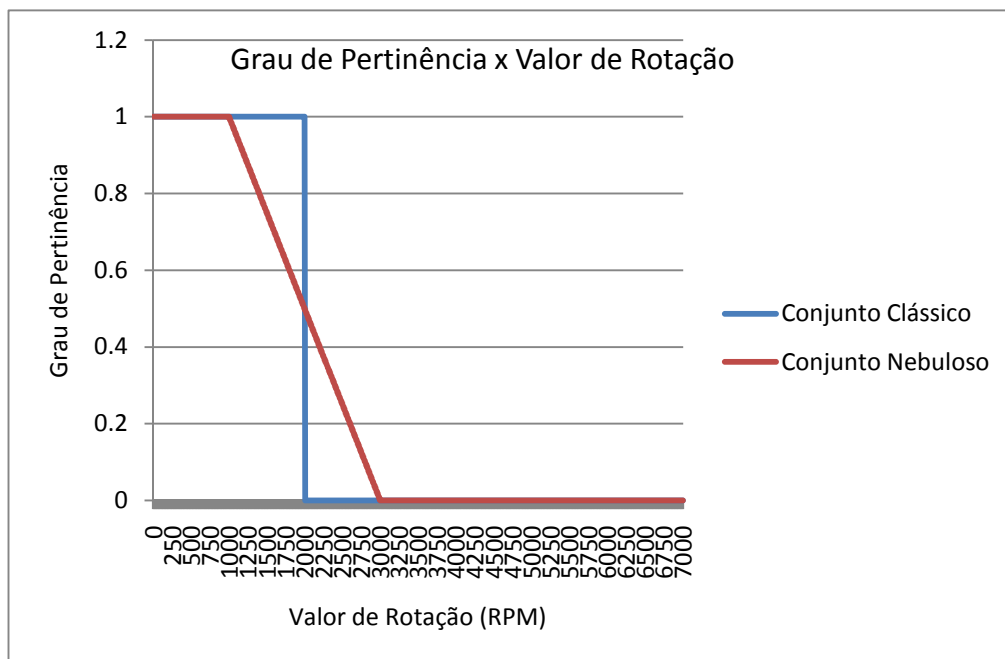


Figura 13: Diferença entre um conjunto clássico e um conjunto nebuloso

## 2.5.2 Funções de Pertinência

Existem diversas maneiras de definir uma função de pertinência, veremos aqui algumas das mais comuns.

### 2.5.2.1 Funções Direitas

As funções direitas possuem dois parâmetros:

1. A – ponto a partir do qual a função de pertinência começa a subir de 0 para 1 e antes do qual a função de pertinência será sempre igual a 0;
2. B – ponto a partir do qual a função de pertinência será sempre igual a 1;

A função direita tem a forma da Equação ( 6 ).

$$F_{direita}(x: A, B) = \max\left(\min\left(\frac{x - A}{B - A}, 1\right), 0\right) \quad (6)$$

Um exemplo de função de pertinência direita pode ser encontrado na Figura 14:

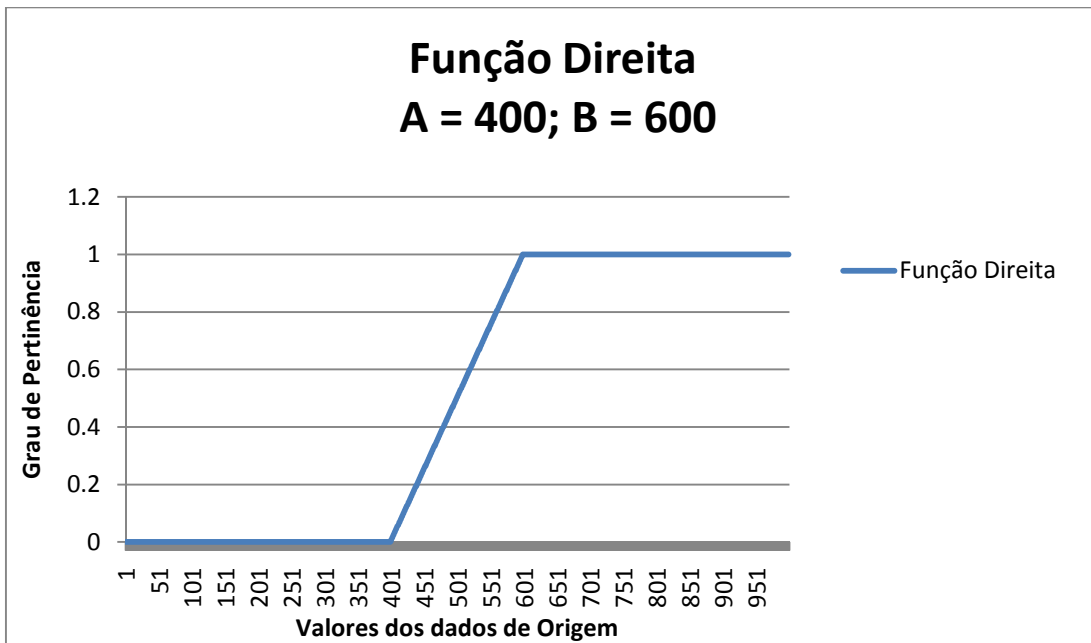


Figura 14: Exemplo de função de pertinência direita

#### 2.5.2.2 Funções Esquerdas

As funções esquerdas possuem dois parâmetros:

3. A – ponto a partir do qual a função de pertinência começa a cair de 1 para 0 e antes do qual a função de pertinência será sempre igual a 1;
4. B – ponto a partir do qual a função de pertinência será sempre igual a 0;

A função esquerda tem a forma da Equação ( 7 ).

$$F_{esquerda}(x: A, B) = \max\left(\min\left(\frac{B - x}{B - A}, 1\right), 0\right) \quad (7)$$

Um exemplo de função de pertinência esquerda pode ser encontrado na Figura 15:

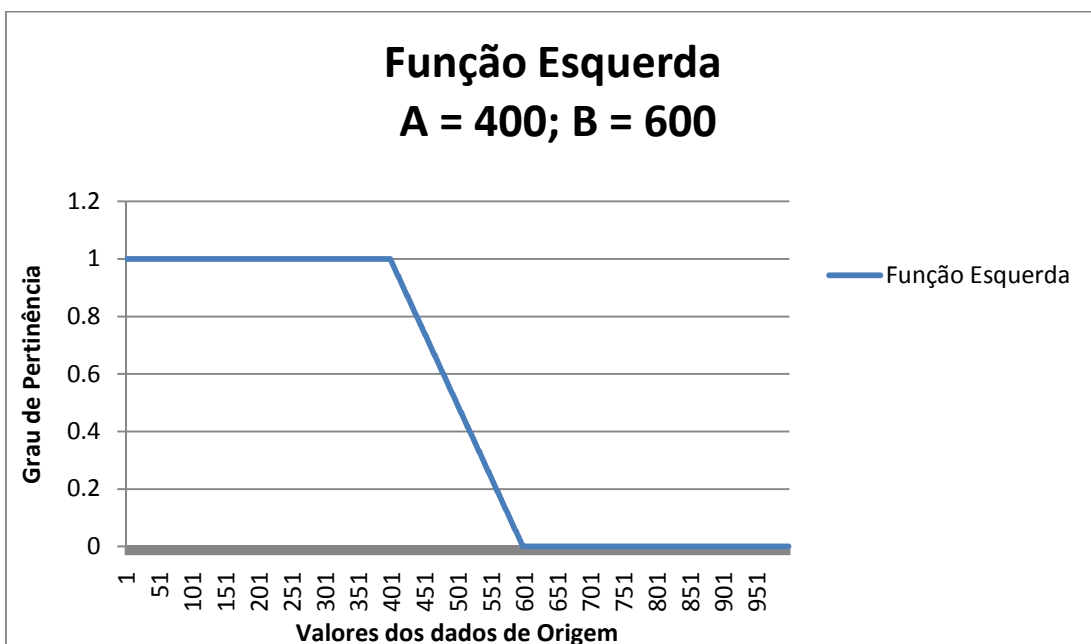


Figura 15: Exemplo de função de pertinência esquerda

### 2.5.2.3 Funções Triangulares

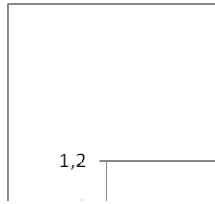
As funções triangulares possuem três

5. A – ponto a partir do qual a pertinência será sempre igual
6. B – ponto a partir do qual a fi
7. C – ponto a partir do qual a fi

A função triangular tem a forma da E

$$F_{triangular}$$

Um exemplo de função de pertinência



**Figura 1**

### 2.5.2.4 Funções Trapezoidais

As funções trapezoidais possuem qua

1. A – ponto a partir do qual a pertinência será sempre igual
2. B – ponto a partir do qual a fi
3. C – ponto a partir do qual a fi
4. D – ponto a partir do qual a fi

A função trapezoidal tem a forma da

$$F_{trapezoidal}$$

Um exemplo de função de pertinência



**Figura 17**

#### 2.5.2.5 Funções Gaussianas

As funções gaussianas possuem dois ]

1.  $\sigma$  – parâmetro que define a la
2.  $c$  – ponto central da função g;

A função gaussiana tem a forma da E

Exemplos de funções de pertinênci  
ilustrados diferentes valores de  $\sigma$ .

**Figura 18**

#### 2.5.2.6 Funções Sino Generalizadas

As funções sino generalizadas possue

3.  $b$  – parâmetro que define o pc
4.  $c$  – parâmetro que define a lai

A função sino generalizada tem a forma

$$F_{sinc}$$

Um exemplo de função de pertinência ilustrados diferentes valores de  $c$ .

**Figura 19**

### 2.5.3 Variáveis lingüísticas

Uma variável lingüística é definida com base na lógica clássica. Assim, uma variável RPM ou 3000 RPM, por exemplo.

Um valor lingüístico é nada menos que uma variável poderá assumir diversos valores pode ser considerada 0,3 “média rotação” e valores lingüísticos que uma determinada

Rotação = {“baixa rotação”, “média rotação”}

Velocidade = {“velocidade muito baixa”, “velocidade muito alta”};

O uso de variáveis lingüísticas permite

- Se a temperatura do motor estiver alta
- Se a rotação está baixa e velocidade

### 2.5.4 Base de Conhecimento

A base de conhecimento do sistema projetista do sistema nebuloso se utiliza de “agentes do sistema”, que são agentes que dominam o sistema (por exemplo), podendo ser ou não o projetista

A base de

## 2.5.5 Sistemas difusos, fuzzificação e defuzzificação.

A aplicação da lógica nebulosa (*fuzzy*), requer basicamente a aplicação de 4 passos:

- *Fuzzificação*: Transformação dos dados de entrada (números) em valores nebulosos (*fuzzy*) através das funções de pertinência.
- Aplicação de Regras: Ao contrário do que acontece na lógica clássica quando uma regra é aplicável ou não, em lógica nebulosa uma regra pode ser parcialmente aplicada.
- Combinação de Consequentes: Quando várias regras dizem respeito a uma mesma saída do sistema, é preciso combinar seus resultados quando mais de uma é parcial ou totalmente aplicável.
- *Defuzzificação*: O processo inverso da *fuzzificação*. O resultado da etapa acima de “Combinação de Consequentes” são valores de saída nebulosos (*fuzzy*), que precisam ser transformados nas grandezas de saída do sistema (números).

### 2.5.5.1 Fuzzificação

O processo de *fuzzificação* transforma os valores de entrada do sistema, que são representados por números, em valores de pertinência a conjuntos nebulosos (variáveis lingüísticas).

O processo de *fuzzificação* se dá por meio do cálculo da função de pertinência de cada conjunto nebuloso a partir do valor da variável de entrada. Nesse sentido, a escolha da função de pertinência deve ser feita considerando-se não só sua adequação ao sistema, mas também seu custo computacional, e, portanto, deve ser considerada a possibilidade de se usar tabelas de correspondências (as *look-up tables*) no lugar de calcular-se a função de pertinência, a fim de acelerar o processo de *fuzzificação*.

### 2.5.5.2 Aplicação de Regras

Como já foi dito, em um sistema nebuloso uma regra é aplicável em um nível que vai de 0 (totalmente não aplicável) a 1 (totalmente aplicável). Esse nível se chama **grau de ativação** da regra, e advém do fato de que os antecedentes de uma regra serão parcialmente verdadeiros também.

Por exemplo, a regra:

- Se a rotação está baixa e velocidade está alta, diminua pouco o tempo de injeção.

Pode ser escrita da seguinte maneira:

- Se rotação pertence ao conjunto “baixa rotação” e velocidade pertence ao conjunto “alta velocidade”, então faça a variação do tempo de injeção pertencer ao conjunto “variação pouco negativa”.

Está claro que se rotação possuir grau 1 de pertinência em relação ao conjunto “baixa rotação” e velocidade possuir grau 1 de pertinência em relação ao conjunto “alta velocidade”, então a variação de tempo de injeção tem que pertencer ao conjunto “variação pouco negativa” com grau de pertinência 1. Mas o que deve acontecer se, por exemplo, rotação possuir 0,4 de pertinência em relação ao conjunto “baixa rotação” e velocidade possuir 0,7 de pertinência em relação ao conjunto “alta velocidade”? Para esses casos, é definida uma forma de tratar as operações “e”, “ou”, “complemento”, “ou exclusivo” e todas outras que forem utilizadas no sistema. Deve-se considerar que para os limites 0 e 1 o resultado da operação deve coincidir com a lógica clássica.

Exemplos de como pode ser tratada a operação “e”:

- Multiplicando os operadores. Por exemplo:  $0,4 \text{ e } 0,7 = 0,4 * 0,7 = 0,28$ ;
- Operação de mínimo. Por exemplo:  $0,4 \text{ e } 0,7 = \text{mínimo}(0,4;0,7) = 0,4$ ;

Exemplos de como pode ser tratada a operação “ou”:

- Fazendo uma soma limitada. Por exemplo:  $0,4 \text{ ou } 0,7 = \text{mínimo}(0,4 + 0,7; 1) = 1$ ;
- Operação de máximo. Por exemplo:  $0,4 \text{ ou } 0,7 = \text{máximo}(0,4;0,7) = 0,7$ ;

Exemplo de como pode ser tratada a operação “complemento”:

- Subtraindo de 1 o valor da variável. Por exemplo: complemento de 0,4 =  $1 - 0,4 = 0,6$ ;

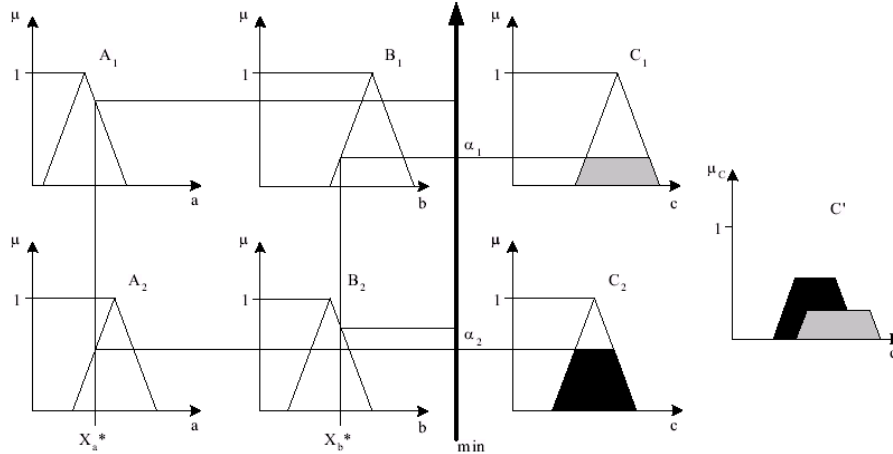
Supondo que optemos por utilizar uma operação de mínimo para a operação “e”, nossa regra do exemplo teria um grau de ativação de 0,4 (que é o resultado da combinação dos antecedentes 0,4 e 0,7).

### 2.5.5.3 Combinação de Conseqüentes

Quando várias regras dizem respeito a uma mesma variável de saída, é preciso combinar o resultado de todas essas regras levando-se em consideração os graus de ativação de cada regra. Os métodos de combinação mais utilizados são: o método de Mamdani [52][53], método de Tsukamoto [54] e o método de Takagi-Sugeno [55]. O método Mamdani é o método de implicação mais importante conhecido na literatura [56]

A Figura 20 mostra um exemplo do modelo de raciocínio tipo Mamdani para um sistema de duas regras com dois antecedentes A e B cada uma, considerando as regras abaixo:

- Se (a é A1) e (b é B1) então c é C1
- Se (a é A2) e (b é B2) então c é C2.



**Figura 20: Exemplo de combinação de consequentes segundo Mamdani.**

Repare que o resultado dessa Combinação de Conseqüentes é uma nova função de pertinência, representada pelo contorno da figura geométrica C' (soma dos trapézios preto e cinza) na Figura 20.

### 2.5.5.4 Defuzzificação

O processo de *Defuzzificação* transforma os valores nebulosos frutos da Combinação de Conseqüentes (C' na seção 2.5.5.3) em valores aplicáveis no sistema de controle. Existem diversas maneiras de fazer a *Defuzzificação*:

Defuzzificação Centro da Área (C-o-A), conforme a Equação ( 12 );

$$u = \frac{\sum_{i=1}^N u_i \mu_{out}(u_i)}{\sum_{i=1}^N \mu_{out}(u_i)} \quad (12)$$

Defuzzificação Centro do Máximo (C-o-M), conforme a Equação ( 13 );

$$u = \frac{\sum_{i=1}^N u_i \cdot \sum_{k=1}^n u_{o,k}(u_i)}{\sum_{i=1}^N \sum_{k=1}^n \mu_{o,k}(u_i)} \quad (13)$$

Defuzzificação Média do Máximo (M-o-M), conforme a Equação ( 14 );

$$u = \sum_{m=1}^M \frac{u_m}{M} \quad (14)$$

Os métodos listados acima não serão abordados com profundidade pois neste trabalho será utilizado o método proposto por Takagi-Sugeno [55] (seção 2.5.6).

### 2.5.6 Método Sugeno

Em um sistema de inferência tipo Sugeno as entradas são definidas como conjuntos nebulosos, mas as saídas são definidas em termos das entradas, geralmente como uma combinação linear. A principal vantagem deste sistema é a sua simplicidade de implementação, o que a torna uma ótima alternativa para sistemas embarcados.



Um sistema de inferência do tipo Sug

- Se rotação pertence ao conjunção faça a variação do tempo de i

O processo de cálculo da variável de (representados na Figura 21):

- Os valores de entrada são av função de pertinência (seção (seção 2.5.5.1);
- Com base nos graus de pertin Na figura Figura 21, esses vai
- O valor da variável de saída (  $F_x$  ) - Na figura Figura 21, es
- Finalmente, obtemos a saída relativas a essas saídas ( $Y_x$ ), ( 15 ).



Figura 21: Exemp

## 2.6 CONCLUSÕES DO CAPI

Nessa seção foram resumidos os tópicos do documento. No contexto de temas correlacionados, sensores, dispositivos e atuadores usados, os principais conceitos usados foram explicitados no contexto deste trabalho.

# 3 METODOLOGIA USADA PARA O PROJETO DO CONTROLADOR

Este capítulo descreve com detalhes todos os procedimentos seguidos para criação do sistema criado, objeto desta dissertação. Inicialmente, na seção 3.1, apresenta-se uma enumeração dos passos seguidos para obtenção dos resultados apresentados. Na seção 3.2 é criada a especificação funcional do produto, ou seja, cria-se o conceito do produto e detalham-se suas funcionalidades. Por último, na seção 3.3 é apresentado o estudo realizado de um *kit* de GNV comercial que serviu como base para o desenvolvimento da estratégia de funcionamento.

## 3.1 O PROJETO PASSO A PASSO

Todas as etapas seguidas na execução do projeto descrito por esta dissertação podem ser sumarizadas nos seguintes passos:

1. Estudo bibliográfico de sistemas de injeção eletrônica de gasolina e seus aspectos tecnológicos, considerações práticas em relação ao GNV e lógica nebulosa (seção 2).
2. Estudo elétrico prático sobre o sistema de injeção eletrônica, com foco na parte de acionamento – seção 2.3.2;
3. Criação de uma especificação funcional, com o intuito de se definir claramente o que se deseja desenvolver (seção 3.2)
4. Levantamento de dados de campo de um *kit* GNV comercial, a fim de guiar o desenvolvimento do *hardware* e a criação de uma estratégia de funcionamento baseada em lógica nebulosa (seção 3.3);
5. Criação de uma especificação de *hardware* com base nos estudos realizados acima (seções 4.2.1 e 4.3.1);
6. Desenvolvimento de duas PCIs (**Placas de Circuito Impresso**) seguindo as especificações do item acima (seção 4);
7. Criação de uma estratégia de funcionamento baseada em lógica nebulosa (seção 5);
8. Criação de uma biblioteca de *firmware* que servirá como base para embarcar-se o algoritmo no microcontrolador (seção 6.3);
9. Criação do *firmware* responsável pela execução da lógica proposta (seção 6.5);
10. Testes de bancada e de campo do conjunto *hardware* e *firmware* desenvolvidos, para avaliação de desempenho (capítulo 7);

## 3.2 DESCRIÇÃO FUNCIONAL DO PRODUTO

A descrição funcional de um produto é o primeiro passo no desenvolvimento desse. A concepção do produto se dá nessa etapa, quando se define quais interfaces o produto terá, quais funções deverá desempenhar e como será o seu funcionamento.

A descrição funcional será feita durante essa seção e deverá servir de base para as especificações de *hardware* e de *firmware* e para a criação da estratégia de funcionamento.

### 3.2.1 Breve descrição do produto

O projeto desenvolvido (tema dessa dissertação) tem a função de permitir a experimentação de diferentes estratégias de cálculo do tempo de injeção de GNV, além de apresentar uma primeira versão de um *hardware* capaz de realizar o acionamento dos bicos injetores de gás, realizar a leitura do tempo de injeção original, a leitura de alguns sensores e a emulação de outros. O produto deve possuir os seguintes diferenciais:

- Baixo custo: por se basear em um sistema de fabricação nacional e por utilizar um *hardware* simples, fruto de uma estratégia simplificada de funcionamento;
- Bom desempenho: apesar de implementar uma estratégia simplificada de funcionamento, trata-se de um sistema a ser utilizado em *kits* de 5ª geração (ver seção 2.4.3), que possibilita um controle bastante apurado da quantidade de gás injetado.

O produto é compatível com veículos 12 V, e suporta os transientes eletromagnéticos típicos de ambientes automotivos;

### 3.2.2 Funções

O sistema desenvolvido deve ser capaz de gerenciar o sistema de GNV quando em funcionamento nesse tipo de combustível, acionando os bicos injetores a partir da leitura de alguns sensores e atuando inclusive em alguns aspectos do sistema de injeção original, a fim de promover o correto funcionamento do veículo (sem perda de potência ou qualquer dano ao veículo).

Para isso, as seguintes funções devem ser previstas:

- *Leitura dos sinais:*
  - Sensor MAP, pelos motivos:
    - Utilizar o valor do sensor como entrada do sistema para determinar a quantidade correta de GNV a ser injetado;
    - Servir como base para o sinal emulado desse sensor.
  - Sonda Lambda, pelo motivo:
    - Utilizar o valor do sensor como entrada do sistema para determinar a quantidade correta de GNV a ser injetado;
  - Tempo de Injeção Original, pelo motivo:
    - Utilizar esse tempo como base para o tempo de injeção de GNV. Esse tempo será corrigido a partir dos outros sinais.
  - Rotação, pelo motivo:
    - Utilizar o valor do sensor como entrada do sistema para determinar a quantidade correta de GNV a ser injetado;
- *Emular o sinal do sensor MAP para modificar o avanço de ignição:*
  - Emular esse sensor é importante porque com esse artifício pode-se “ludibriar” a UCE original do carro, fazendo-a considerar um valor de pressão no coletor diferente do valor real no momento do cálculo do avanço de ignição, o que permite a quem emula esse sinal manipular indiretamente esse parâmetro.
  - Essa função pode também ser desempenhada por um módulo independente, o chamado “Variador de Avanço”. No caso de ser utilizado um desses “Variadores de Avanço”, a função de emulação do sensor MAP não deve ser utilizada.
- *Emular o sinal do sensor de Sonda Lambda:*
  - Durante o funcionamento normal do veículo, o sinal de Sonda Lambda informa se a UCE original está produzindo uma mistura rica ou pobre, a UCE original então reage corrigindo essa mistura, modificando parâmetros de cálculo interno.

Quando o veículo está funcionando a GNV, a UCE original não é mais responsável por determinar a quantidade de combustível, embora não “saiba” disso. Como resultado, a tentativa frustrada de correção da mistura faz com que a UCE crie parâmetros de injeção inadequados e pode culminar na indicação de falha de injeção.

Para evitar que isso aconteça, o sinal enviado à UCE original deve ser emulado de forma a indicar um funcionamento normal do motor, o que evitar que a UCE original faça correções inadequadas.
  - Hoje existem diversos produtos no mercado que realizam essa função (os chamados Emuladores de Sonda Lambda ou Simuladores de Sonda Lambda), e por esse motivo essa funcionalidade é apenas prevista, mas não implementada.
- *Cálculo do tempo de injeção de GNV a partir dos sinais lidos;*
- *Acionamento dos bicos injetores de GNV no tempo adequado;*
- *“Congelamento” dos bicos injetores de gasolina/álcool quando o veículo estiver funcionando a GNV;*

- *Acionamento de cargas externa a partir de saídas genéricas, o que confere flexibilidade na implementação de novas funcionalidades ou em testes de engenharia;*
- *Capacidade de armazenar configurações e mantê-las mesmo quando desenergizado;*
- *Conexão com mundo externo para depuração, configuração e monitoramento de funcionamento via interface serial.*

### 3.3 ESTUDO DE UM KIT GNV COMERCIAL

Antes de iniciar o desenvolvimento do sistema próprio, optou-se por se fazer um estudo sobre o funcionamento de um *kit* de GNV comercial, com o intuito de fazer uma espécie de “reconhecimento do terreno”. Essa etapa é extremamente importante, pois possibilita que alguns problemas e desafios sejam antecipados. Outro fator importante desse estudo é que a partir dele é possível calcular alguns parâmetros dos sinais a serem lidos ou produzidos pelo sistema e que servirão para guiar o projeto de *hardware*.

Esta seção descreve como foi feito esse estudo.

#### 3.3.1 Objetivos

O estudo de um *kit* GNV comercial tem como objetivos:

- Antecipar problemas e desafios no projeto do sistema próprio;
- Servir como base para o desenvolvimento do sistema de controle nebuloso, atuando no papel de especialista do sistema (seção 5);
- Possibilitar o levantamento de parâmetros para o desenvolvimento do *hardware*. Esses parâmetros não podem ser encontrados na literatura por seu caráter técnico e específico.

#### 3.3.2 Sistema de GNV utilizado

O sistema de GNV utilizado como base para alcançar os objetivos descritos na seção 3.3.1, foi instalado em um veículo modelo *Peugeot 206*, motor 1.4, ano 2005.

Em uma primeira abordagem, considera-se que o *kit* de GNV instalado seja uma “caixa preta” com entradas e saídas. Com o apoio do técnico que realizou a instalação do *kit* no veículo, foram identificados os sinais de entrada e saída do *kit*. Os sinais estão dispostos em um conector de 24 vias, 3 linhas de 8 sinais, e são apresentados na Tabela 3 com as informações: *localização* física do sinal no conector (linha A, B ou C e sinal de 1 a 8), *nome do sinal* e uma breve *descrição* do sinal.

**Tabela 3: Sinais do *kit* GNV comercial**

Localização	Nome do Sinal	Descrição
A1	Sinal de Gás	<p>Esse sinal é responsável por alimentar e indicar a todo o sistema de gás que o veículo deve funcionar a GNV. Para isso, desempenha as seguintes funções:</p> <ul style="list-style-type: none"> <li>• Alimenta os bicos injetores de GNV;</li> <li>• Alimenta o redutor de pressão;</li> <li>• Alimenta a válvula solenóide que libera a passagem de GNV;</li> <li>• Indica ao variador de avanço para iniciar o avanço de ignição;</li> </ul> <p>Esse sinal é criado internamente no <i>kit</i> comercial a partir do chaveamento da alimentação fornecida pela bateria.</p> <p>Sua leitura é interpretada de forma digital: ligado ou desligado.</p>

A2, A3, A4 e A5	Bico Gás 1, 2, 3 e 4	<p>Enquanto um dos terminais dos bicos injetores de gás se conecta ao “Sinal de Gás”, o outro terminal se conecta ao sinal “Bico Gás”, que cria um caminho para a corrente que deve passar pelo solenóide do bico quando se deseja injetar gás (ver seção 2.3.2 para mais informações sobre o funcionamento de bicos injetores).</p> <p>Sua leitura é interpretada de forma digital: ligado ou desligado.</p>
A6	Linha para diagnóstico	<p>Essa linha é utilizada para diagnóstico e configuração dos parâmetros (como os mapas, por exemplo) do <i>kit</i> comercial.</p> <p>Sua leitura é interpretada de forma digital: ligado ou desligado.</p>
A7	Pressão da Flauta	<p>Traz a informação de pressão de gás na flauta que distribui o GNV para os bicos de gás.</p> <p>Sua leitura é interpretada de forma analógica.</p>
A8	Sinal de Rotação	<p>Traz a informação do sinal de rotação. No caso do <i>kit</i> instalado, utilizou-se o sinal de ignição para leitura de rotação.</p> <p>Sua leitura é interpretada de forma digital: com a frequência do sinal sendo a informação útil.</p>
B1	GND	Sinal negativo de alimentação.
B2, B3, B4 e B5	Bico Original – UCE 1, 2, 3 e	<p>Para a instalação do <i>kit</i>, as conexões que ligam a UCE aos bicos de injeção originais são interrompidas e as extremidades que se conectam à UCE são ligadas ao <i>kit</i> por meio dos sinais “Bico Original – UCE” correspondentes (as outras extremidades se conectam a “Bico Original – Bico”).</p> <p>Sua leitura é interpretada de forma digital: ligado ou desligado.</p>
B6	Sonda Lambda	<p>Traz a informação do sensor de Sonda Lambda.</p> <p>Sua leitura é interpretada de forma analógica.</p>
B7	MAP	<p>Traz a informação do sensor MAP.</p> <p>Sua leitura é interpretada de forma analógica.</p>
B8	Pós Chave	Sinal pós-chave da alimentação do veículo. Informa se o veículo está ou não em funcionamento.
C1	Positivo da Bateria	Sinal positivo de alimentação.
C2, C3, C4 e C5	Bico Original – Bico 1, 2, 3 e 4	<p>Para a instalação do <i>kit</i>, as conexões que ligam a UCE aos bicos de injeção originais são interrompidas e as extremidades que se conectam aos bicos são ligadas ao <i>kit</i> por meio dos sinais “Bico Original – Bico” correspondentes (as outras extremidades se conectam a “Bico Original – UCE”).</p> <p>Sua leitura é interpretada de forma digital: ligado ou desligado.</p>

C6	V+ Sensores e Diagnóstico	Sinal de alimentação positivo dos sensores e da linha de diagnóstico. Essa alimentação é fornecida pelo <i>kit</i> comercial aos sensores.  Sua leitura é interpretada de forma digital: ligado ou desligado.
C7	Temperatura do Redutor de Pressão	Traz a informação de temperatura do redutor de pressão.  Sua leitura é interpretada de forma analógica.
C8	V- Sensores e Diagnóstico	Sinal de alimentação negativo dos sensores e da linha de diagnóstico. Essa alimentação é fornecida pelo <i>kit</i> comercial aos sensores.

### 3.3.3 Método de Medição

Para alcançar os objetivos descritos na seção 3.3.1, foi necessário adquirir dados de sinais selecionados do *kit* comercial enquanto o mesmo encontrava-se em funcionamento normal (motor funcionando).

A aquisição foi feita com um circuito de tratamento dos sinais e um arranjo de dispositivos de aquisição da National Instruments com conexão USB para um PC e gerenciadas por uma interface feita em LabVIEW. Dessa forma, para realizar a medição dos sinais, duas etapas são necessárias:

- Tratamento analógico dos sinais de entrada para adequação de níveis elétricos;
- Aquisição digital dos sinais;

Na seção 3.3.3.1 são enumerados os sinais adquiridos e as motivações para cada um desses sinais, na seção 3.3.3.2 o dispositivo de aquisição dos sinais da National Instruments é apresentado, na seção 3.3.3.3 é descrita a etapa de tratamento analógico dos sinais e na seção 3.3.3.4 é descrita a etapa de aquisição digital desses sinais.

#### 3.3.3.1 Sinais Medidos

Por limitações técnicas dos dispositivos de aquisição de sinais (seção 3.3.3.2) e pela quantidade disponível desses dispositivos (apenas 4), foi necessário que fosse escolhido um pequeno grupo de sinais do *kit* comercial a serem lidos. Abaixo se encontra uma listagem desses sinais e a justificativa por sua escolha:

- *Sinal de Gás (A1)*: esse sinal é importante na medida em que permite que se saiba o momento em que o sistema GNV passa a estar alimentado.
- *Bico Gás 1 (A2)*: carrega a informação do tempo de injeção de GNV, ou seja, é a saída do sistema. A partir desse sinal e das entradas do sistema pode-se estudar a estratégia de funcionamento do *kit* comercial.
- *Pressão da Flauta (A7)*: a vazão de gás guarda estreita relação com a pressão da flauta, e a partir desse sensor imagina-se que o *kit* comercial corrija o tempo de injeção de gás para se alcançar a massa correta de GNV a ser injetada.
- *Rotação (A8)*: a rotação é um dos parâmetros de funcionamento mais importantes do motor, e serve de entrada para o sistema de controle de injeção.
- *Bico Original – UCE 1 (B2)*: a partir desse sinal pode-se calcular o tempo de injeção de combustível original que a UCE origina está implementando (embora quando o veículo está funcionando a GNV nenhum combustível original seja injetado). O sistema a ser projetado irá utilizar esse tempo como base para a determinação do tempo de injeção de GNV.
- *Sonda Lambda (B6)*: esse sensor traz a informação de quantidade de oxigênio nos gases de escape (seção 2.3.1) e é utilizado para realimentação do sistema de controle do tempo de injeção tanto para a UCE original quanto para o *kit* comercial (dependendo do combustível em uso no momento).
- *MAP (B7)*: esse sensor indica a pressão no coletor de ar e traz como principal informação a carga a qual o motor está submetido, ou seja, o quanto de ar será admitido no cilindro. Serve de entrada para o sistema de controle de injeção.

- *Bico Original – Bico 1 (C2)*: a partir desse sinal pode-se descobrir quando o bico injetor de combustível original foi congelado, quando isso acontece, esse sinal passa a não ser mais igual ao Bico Original – UCE 1.
- *Temperatura do Redutor de Pressão (C7)*: a temperatura do redutor de pressão determina quando o sistema GNV está pronto para funcionar e é provavelmente utilizado para corrigir o tempo de injeção de GNV, já que a temperatura se relaciona com densidade volumétrica de combustível.

### 3.3.3.2 Dispositivo de Aquisição dos Sinais

O dispositivo de aquisição de sinais utilizado foi o DAQ-6009 da National Instruments. Esse dispositivo foi escolhido por sua disponibilidade na UnB (Universidade de Brasília). Suas características são:

- 8 entradas analógicas (14-bit, 48 kS/s);
- Impedância de entrada de 144 k $\Omega$  para as entradas analógicas;
- Faixa de tensão de entrada de 0V a 10V;
- 2 saídas analógicas (12-bit, 150 S/s);
- 12 entradas e saídas digitais;
- Contador de 32-bit;
- Alimentado pela USB para mobilidade.

É importante ressaltar que a taxa de aquisição das entradas analógicas de 48 kS/s é dividida entre todas as entradas que estejam sendo utilizadas.

### 3.3.3.3 Tratamento Analógico dos Sinais

Antes de desenvolver o sistema de aquisição de dados, foi preciso conhecer o comportamento e a forma de cada um dos sinais listados na seção 3.3.3.1. Para isto foi utilizado um osciloscópio e cada um dos sinais foi medido. Os resultados desta medição se apresentam nas figuras: Figura 22, Figura 23, Figura 24, Figura 25, Figura 26, Figura 27, Figura 28, Figura 29, Figura 30, Figura 31, Figura 32, Figura 33 e Figura 34. As faixas de tensão de cada sinal medido são apresentadas na Tabela 4.

**Tabela 4: Faixas de tensão dos sinais medidos**

<b>Nome do Sinal</b>	<b>Limites positivo e negativo de tensão</b>
<i>Sinal de Gás</i>	0V – Tensão do Alternador (cerca de 14V)
<i>Bico Gás</i>	0V – Sinal de Gás (cerca de 14V), com picos de até 38V
<i>Pressão da Flauta</i>	0V – ~2V
<i>Rotação</i>	0V – 30V com picos acima de 70V. O sinal de ignição é utilizado para medida de rotação.
<i>Bico Original – UCE</i>	0V – Tensão do Alternador (cerca de 14V), com picos de até 60V quando estiver rodando no combustível original.
<i>Sonda Lambda</i>	0V – 1V
<i>MAP</i>	0V – 2V. Porém, sabe-se que esse sinal pode atingir até 5V [29].
<i>Bico Original – Bico</i>	0V – Tensão do Alternador (cerca de 14V), com picos de até 60V quando estiver rodando no combustível original.

Temperatura do Redutor de Pressão

Até aproximadamente 1,7V

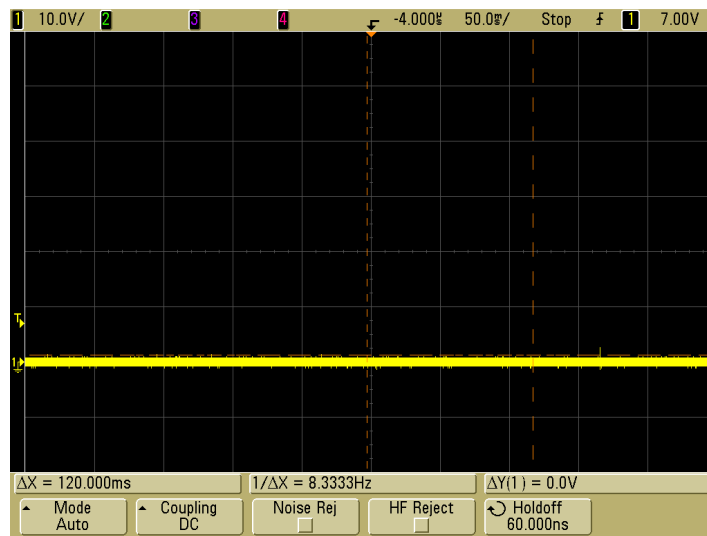


Figura 22: Sinal de Gás quando veículo está rodando no combustível original

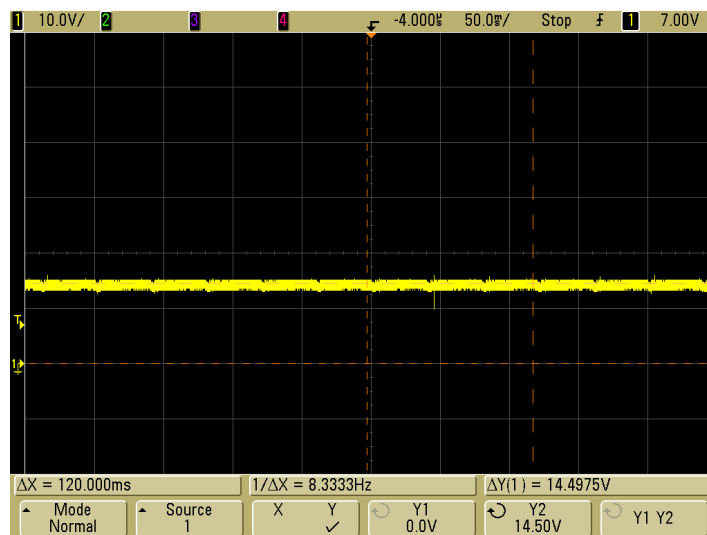


Figura 23: Sinal de Gás quando veículo está rodando no GNV.

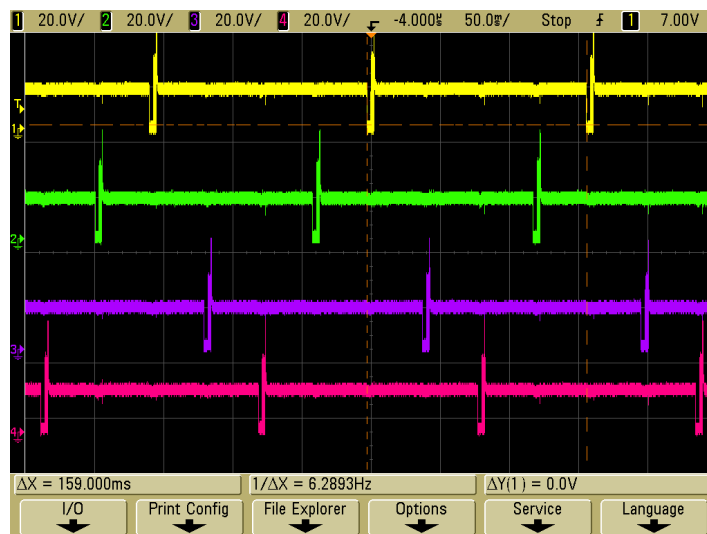


Figura 24: Sinal dos quatro bicos injetores (Bico Gás 1, 2, 3 e 4).



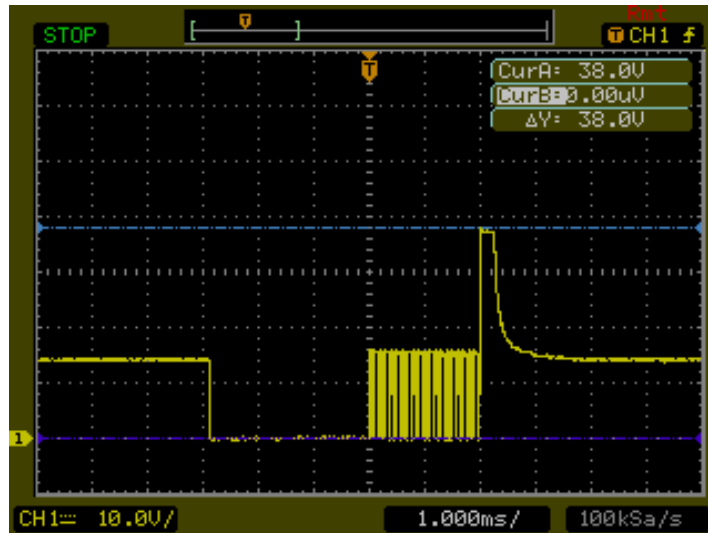


Figura 25: Sinal de um dos bicos injetores - visão aproximada

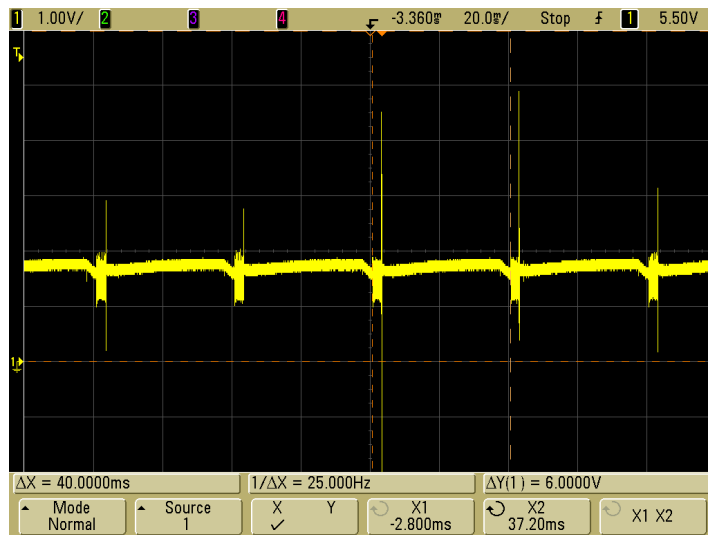


Figura 26: Sinal de Pressão na Flauta.

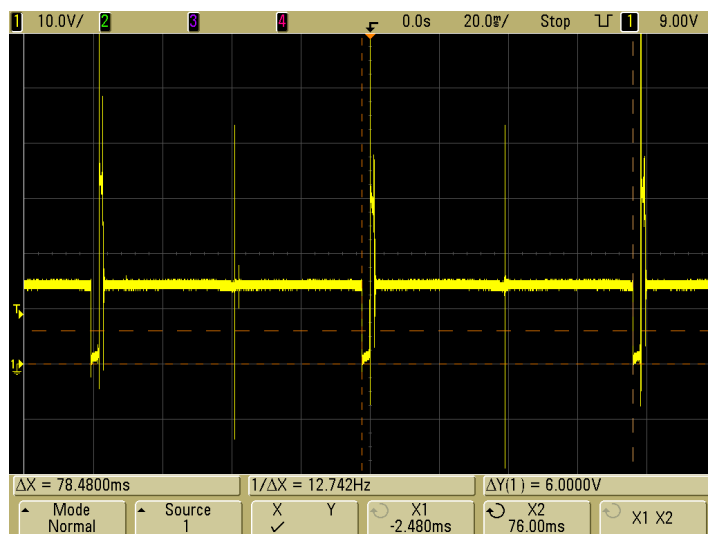
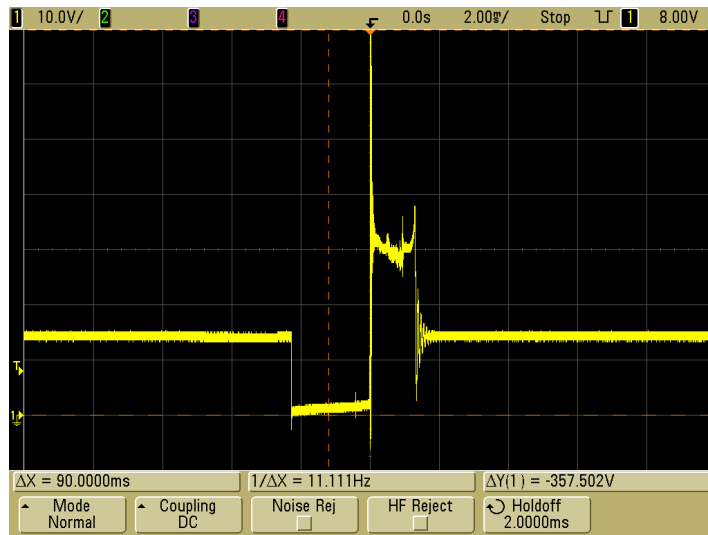
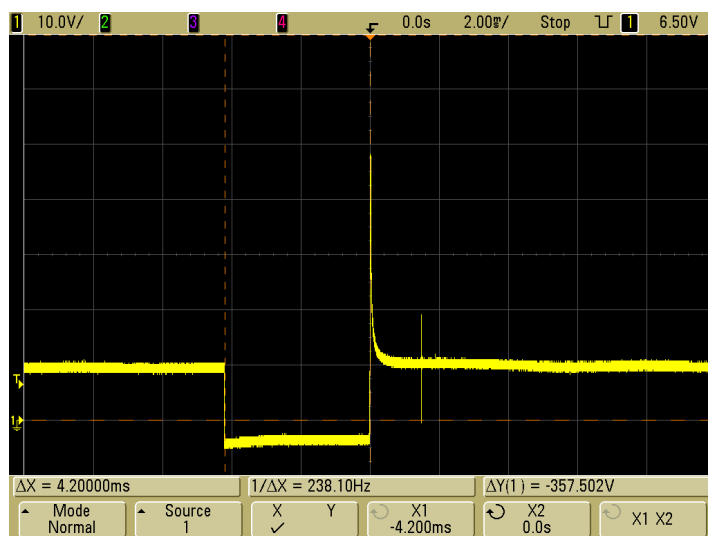


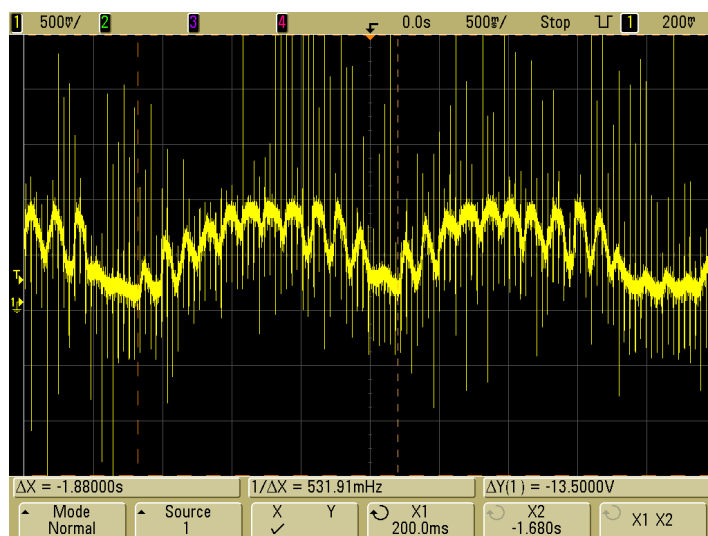
Figura 27: Sinal de Rotação - visão macro



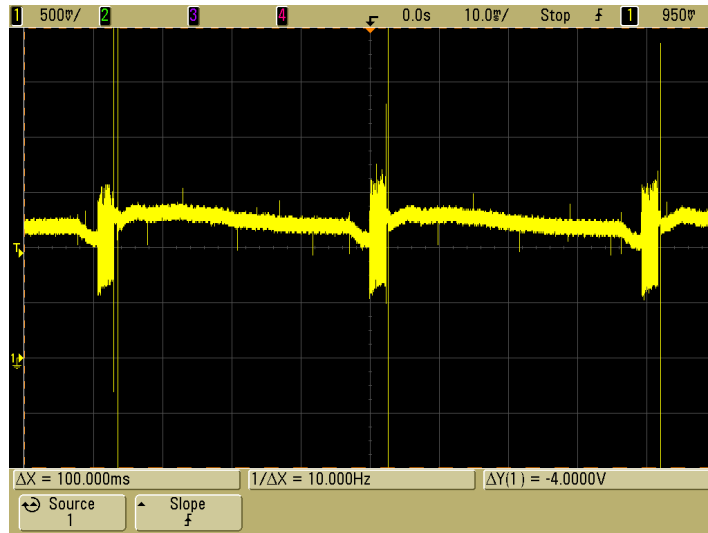
**Figura 28: Sinal de Rotação - visão aproximada**



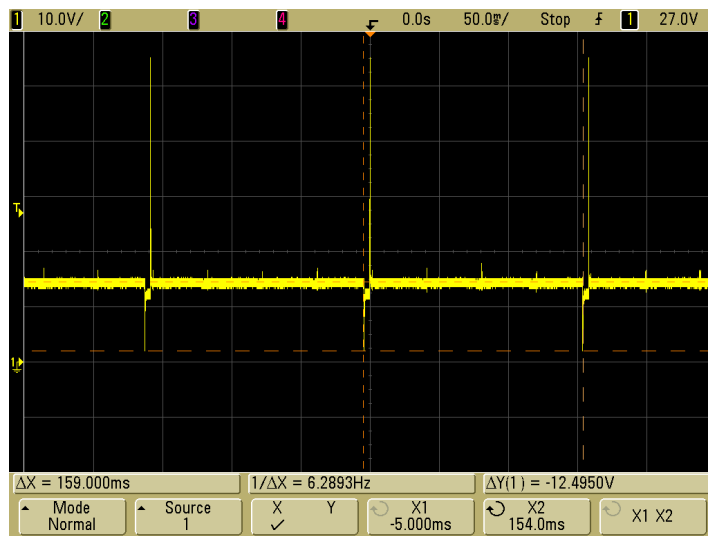
**Figura 29: Bico Original - UCE**



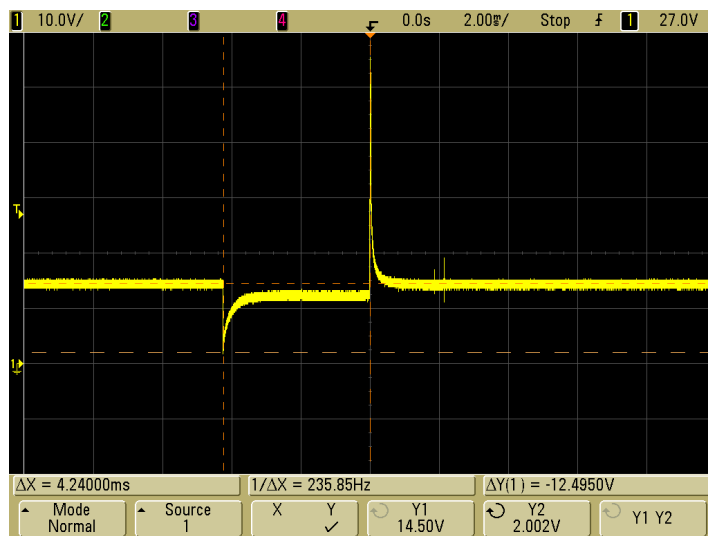
**Figura 30: Sinal Sonda Lambda**



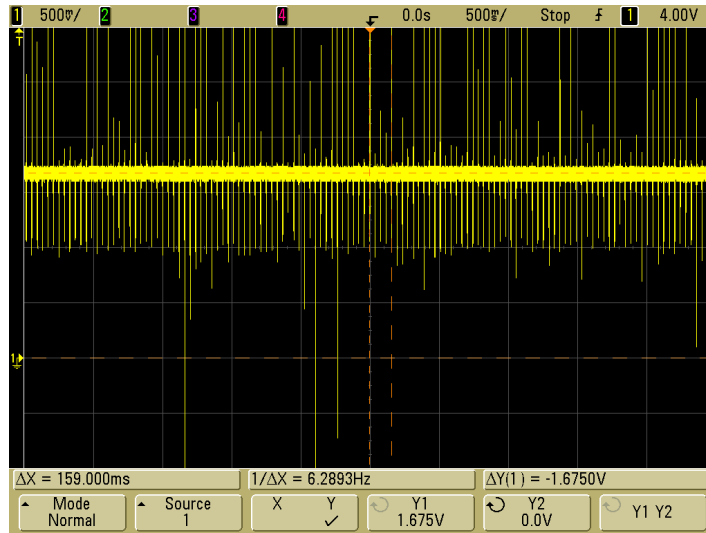
**Figura 31: Sinal do Sensor MAP**



**Figura 32: Sinal Bico Original - Bico, rodando no GNV. Visão macro**



**Figura 33: Sinal Bico Original - Bico, rodando no GNV. Visão aproximada**



**Figura 34: Sinal de Temperatura do Redutor de Pressão**

Os sinais foram então divididos em três grupos:

1. Sinais que necessitam de grande amplificação (até 2V): Pressão na Flauta, Sonda Lambda, Temperatura do Redutor de Pressão;
2. Sinais que não necessitam de amplificação nem de atenuação (2V - 10V): MAP;
3. Sinais que necessitam de atenuação (maiores que 10V): Bico Gás 1, Bico Original – UCE 1, Rotação, Bico Original – Bico 1, Sinal de Gás;

Para que os sinais lidos não sofressem influência do circuito de leitura, optou-se por utilizar impedâncias de entrada muito altas (na ordem de MΩ). No entanto, essa decisão trouxe dois inconvenientes:

1. Ruído – quanto maior a impedância na qual um sinal é medido, maior o ruído ao qual o sinal está sujeito. Esse inconveniente acabou se resolvendo sozinho, pois o próprio sistema de medição possui sua banda passante, o que acaba por filtrar o ruído.
2. Atenuação do sinal lido – a impedância do dispositivo de medição é de 144 kΩ, o que faz com que a impedância de entrada dezenas de vezes maior faça o sinal medido pelo dispositivo ser bastante atenuado. Para resolver esse problema foi utilizado um amplificador operacional.

Os circuitos resultantes estão representados nas figuras: Figura 35, Figura 36 e Figura 37.

Para o circuito amplificador, a relação entre a tensão de saída ( $V_{out}$ ) e a tensão de entrada ( $V_{in}$ ) é representada pela Equação ( 16 ).

$$V_{out} * \frac{R2}{R2 + R3} = V_{in} \therefore V_{out} = V_{in} \left( 1 + \frac{R3}{R2} \right) = V_{in} \left( 1 + \frac{16,8M}{8,2M} \right) \cong 3,05 V_{in} \quad (16)$$

Para o circuito amplificador, a relação entre a tensão de saída ( $V_{out}$ ) e a tensão de entrada ( $V_{in}$ ) é representada pela Equação ( 17 ).

$$V_{out} = V_{in} \quad (17)$$

Para o circuito amplificador, a relação entre a tensão de saída ( $V_{out}$ ) e a tensão de entrada ( $V_{in}$ ) é representada pela Equação ( 18 ).

$$V_{out} = V_{in} * \frac{R2}{R1 + R2} = V_{in} * \frac{2M}{2M + 5,6M} \cong 0,26 V_{in} \quad (18)$$

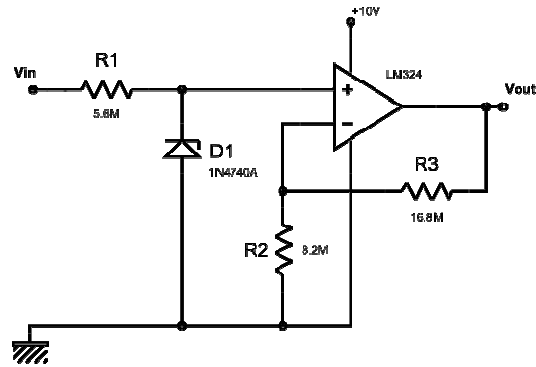


Figura 35: Circuito para sinais que necessitam de amplificação

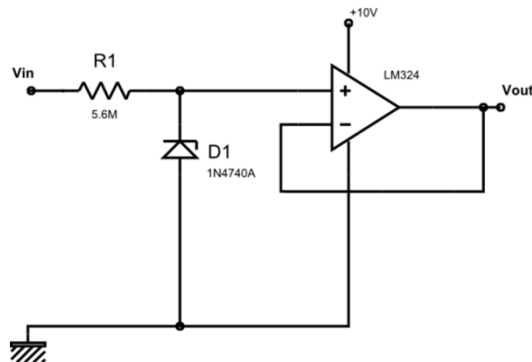


Figura 36: Circuito para sinais que não necessitam nem de amplificação nem de atenuação

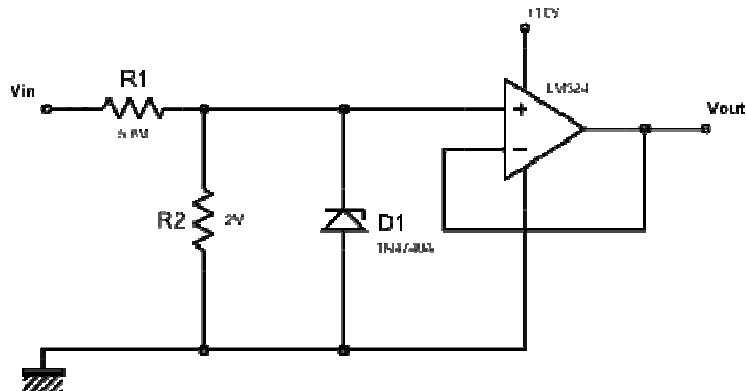


Figura 37: Circuito para sinais que necessitam de atenuação

#### 3.3.3.4 Aquisição digital dos Sinais

A aquisição digital dos sinais foi feita dividindo-se os sinais entre os 4 dispositivos de aquisição DAQ-6009 (ver seção 3.3.3.2) da seguinte maneira:

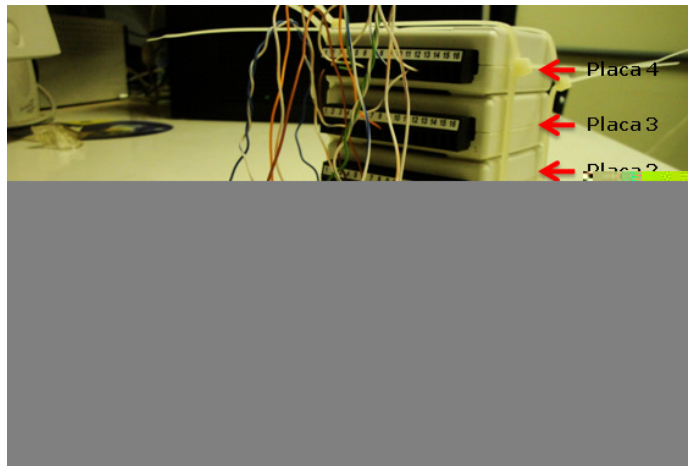
1º DAQ - Bico Gás 1, com taxa de 25 kHz;

2º DAQ - Bico Original – UCE 1, com taxa de 25 kHz;

3º DAQ - Rotação, com taxa de 25 kHz;

4º DAQ - Pressão da Flauta, Sonda Lambda, MAP, Bico Original – Bico 1, Temperatura do Redutor de Pressão, Sinal Gás, cada um com taxa de 5 kHz;

A Figura 38 traz um foto do sistema utilizado para aquisição, compreendendo os circuitos de tratamento analógico dos sinais (ver seção 3.3.3.3) e os dispositivos de aquisição digital dos sinais (ver seção 3.3.3.2).



**Figura 38: sistema de aquisição dos sinais**

As informações obtidas a partir da medição foram armazenadas nos arquivos mostrados na Tabela 5. As situações criadas na captura dos registros foram escolhidas na tentativa de se representar um maior número de condições de funcionamento do motor.

**Tabela 5: Arquivos com resultados de medição**

Nome do Arquivo	Descrição
<i>conmutacao_gasolina_Gas_1</i>	Desde o momento em que o carro foi ligado até que ocorresse a comutação de gasolina para GNV.
<i>aceleracao_desengatado_1</i>	Acelerações com o carro desengatado funcionando a GNV.
<i>aceleracao_0_100_trocaMarcha_3</i>	Carro acelerando de 0 a 100 km/h com trocas de marcha.
<i>aceleracao40_80_2</i>	Carro acelerando de 40 a 80 km/h em 3ª marcha.
<i>desaceleracao100_0_freando_3</i>	Carro diminuindo de 100 a 0 km/h com o carro desengatado.
<i>desaceleracao80_40_3Marcha_3</i>	Carro diminuindo de 80 a 40 km/h com o carro em 3ª marcha.
<i>velocidadeConstante_60_plano_2</i>	Carro em velocidade constante de 60 km/h em terreno plano.
<i>velocidadeConstante_60_subida_2</i>	Carro em velocidade constante de 60 km/h na subida.
<i>velocidadeConstante_60_descida_2</i>	Carro em velocidade constante de 60 km/h na descida.
<i>velocidadeConstante_Gas_Gasolina_2</i>	Comutação de GNV para gasolina em velocidade constante.
<i>velocidadeConstante_Gasolina_Gas_2</i>	Comutação de gasolina para GNV em velocidade constante.
<i>acelaracoes_Gas_desengatado_2</i>	Acelerações com o carro desengatado funcionando a GNV.
<i>deligar_2</i>	Desligamento do veículo enquanto está funcionando a GNV.

### 3.3.4 Cálculo dos parâmetros dos dados lidos

Nessa seção são descritos os parâmetros necessários para o projeto de *hardware* e de *firmware* a serem extraídos a partir dos arquivos obtidos no campo (seção 3.3.3.4), o método de cálculo desses parâmetros e os resultados alcançados. Foi dedicada uma seção para cada sinal, pois cada um possui diferentes parâmetros de interesse.

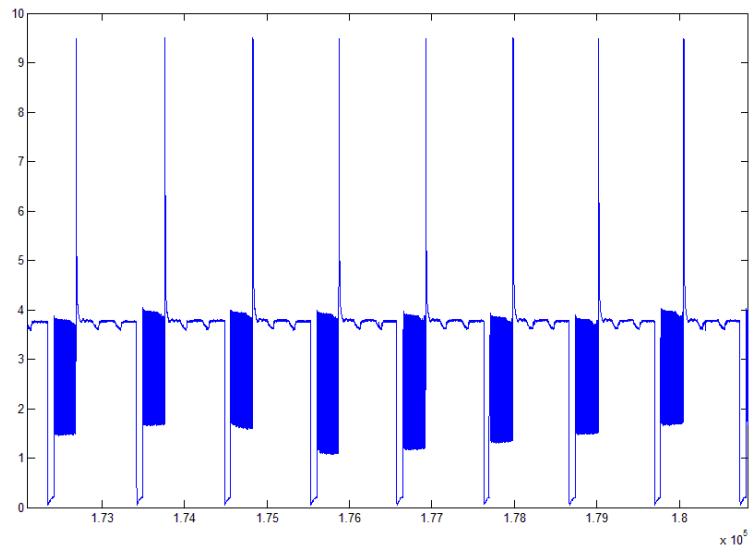
O arquivo *aceleracao\_0\_100\_trocaMarcha\_3* é utilizado na maioria dos casos para obtenção dos parâmetros pois considera-se que seja o registro com maior variedade de situações de funcionamento.

### 3.3.4.1 Sinal de Gás

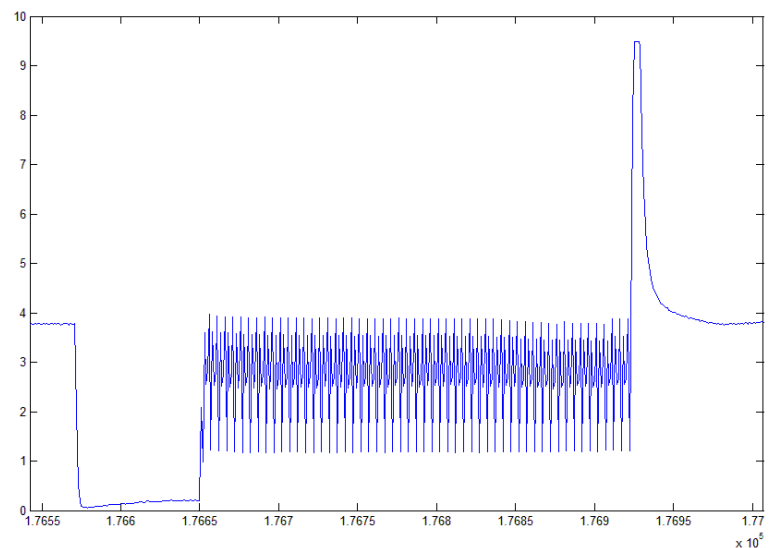
Esse sinal é produzido pelo *kit* de GNV a partir da comutação do sinal de alimentação do veículo (vindo do alternador), e, portanto, sua medição serviu apenas para informar quando o veículo está funcionando a GNV ou não. Dessa forma, nenhum parâmetro é necessário nesse caso.

### 3.3.4.2 Bico Gás 1

Esse sinal indica quando o bico de gás está ou não acionado, injetando GNV. Uma parte da forma de onda desse sinal obtida no arquivo *aceleracao\_0\_100\_trocaMarcha\_3* é mostrada na Figura 39 e Figura 40. Cada pulso negativo nessas figuras representa um ciclo de injeção composto de dois momentos: *peak* (parte constante) e *hold* (parte chaveada), ver adiante.



**Figura 39: Forma de onda do sinal Bico Gás 1 - vários ciclos de injeção**



**Figura 40: Forma de onda do sinal Bico Gás 1 - um único ciclo de injeção**

A primeira informação importante obtida desse sinal é a de que o ciclo de injeção é formado por dois momentos: um *constante* e outro com *chaveamento*. Isso acontece porque o bico de GNV utilizado no *kit* comercial é um bico de baixa impedância, o que possibilita sua rápida abertura, mas que ao mesmo tempo faz com que circule uma alta corrente quando submetido a um acionamento constante. Por esse motivo, o *kit* comercial utiliza o que na literatura se conhece como estratégia de funcionamento *peak and hold*, onde:

- *Peak*: acionamento constante, utilizado principalmente para rápida abertura dos bicos injetores;

- *Hold*: acionamento chaveado que limita a corrente passando pelo solenóide, utilizado para manter o bico injetor aberto. Deve-se levar em consideração que a força necessária no solenóide para manter o bico injetor aberto é menor que a força necessária para abrir esse mesmo bico injetor.

A Figura 40 mostra cada um desses momentos, mas a taxa de amostragem não foi suficiente para indicar o ciclo de trabalho e a frequência durante o período *hold*. Por esse motivo, foi utilizado um osciloscópio para realizar a aquisição desse sinal, o que é mostrado nas figuras: Figura 41, Figura 42 e Figura 43. Por problemas técnicos a Figura 43 não mostra a legenda, mas o valor foi anotado e é igual a 66,0us.

A Figura 41 mostra uma duração de 2,96ms de *peak*, e outros testes mostraram que esse tempo sempre se mantém, mesmo quando o tempo total de acionamento se modifica. Ou seja, para modificar o tempo total de acionamento, o *kit* comercial altera o tempo do *hold* e mantém o tempo do *peak*.

Os parâmetros obtidos desse sinal são apresentados nas Equações ( 19 ), ( 20 ) e ( 21 ).

$$Duração_{peak} = 2,96ms \quad (19)$$

$$f_{hold} = \frac{1}{100us} = 10 \text{ kHz} \quad (20)$$

$$Ciclo \ de \ Trabalho_{hold} = \frac{100us - 66us}{100us} = 34\% \quad (21)$$

Repare que o ciclo de trabalho foi feito considerando o tempo em baixa (100us – 66us) do sinal, porque é durante esse período que o caminho de corrente é criado. Assim, apenas 34% da corrente que flui durante o *peak* fluirá durante o *hold*. Para verificar essa afirmação, um resistor de potência (5 W) de 120 mΩ nominais foi colocado em série com o bico injetor de gás e as tensões em seus dois terminais foram medidas durante um ciclo de injeção, e logo após subtraídas, o resultado é apresentado na Figura 44.

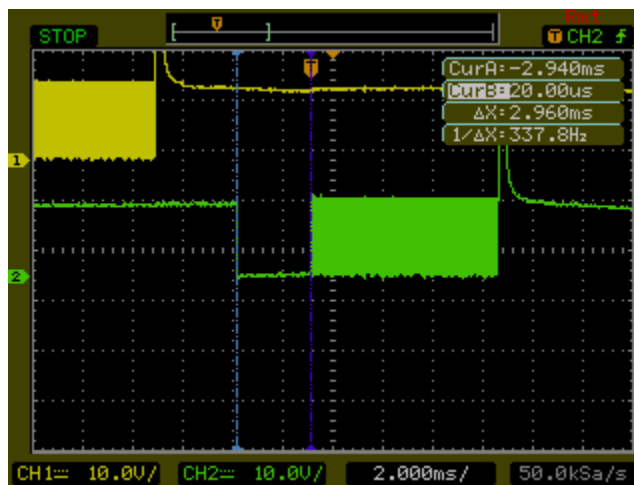


Figura 41: Tempo de *hold*



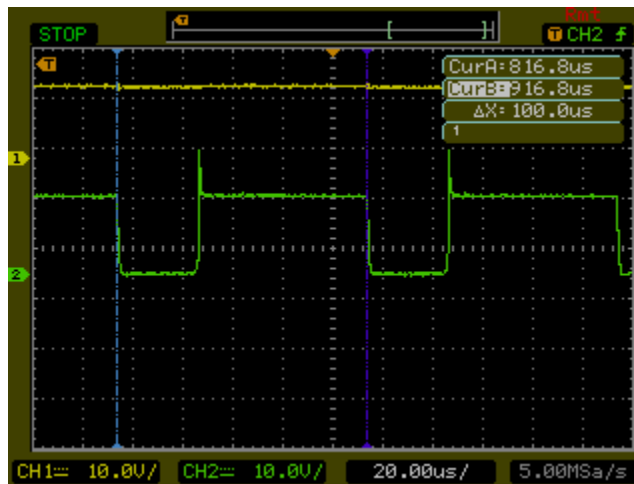


Figura 42: Período de chaveamento durante o hold

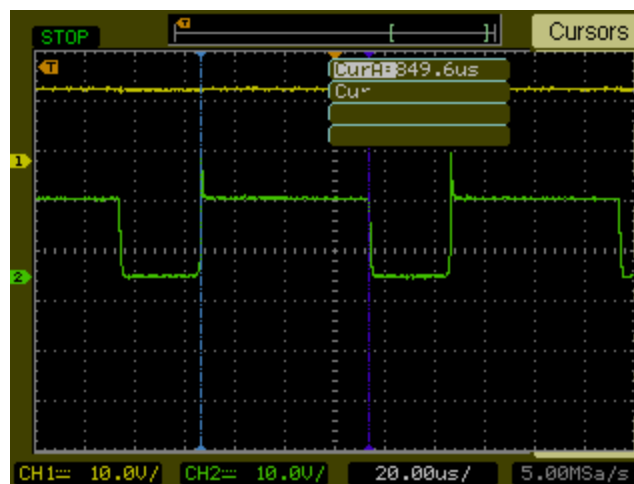


Figura 43: Tempo em alta durante o hold



Figura 44: Perfil de corrente no bico injetor de GNV (em rosa) – repare que a corrente durante o hold estabiliza em aproximadamente 34% da corrente máxima durante o peak.

Outro parâmetro importante obtido no arquivo *aceleracao\_0\_100\_trocaMarcha\_3* é que o tempo total de injeção foi de no máximo 20ms ao longo de todo percurso, fato representado na Equação ( 22 ).

$$\text{Tempo Total de Injeção}_{\text{máx}} = 20\text{ms} \quad ( 22 )$$

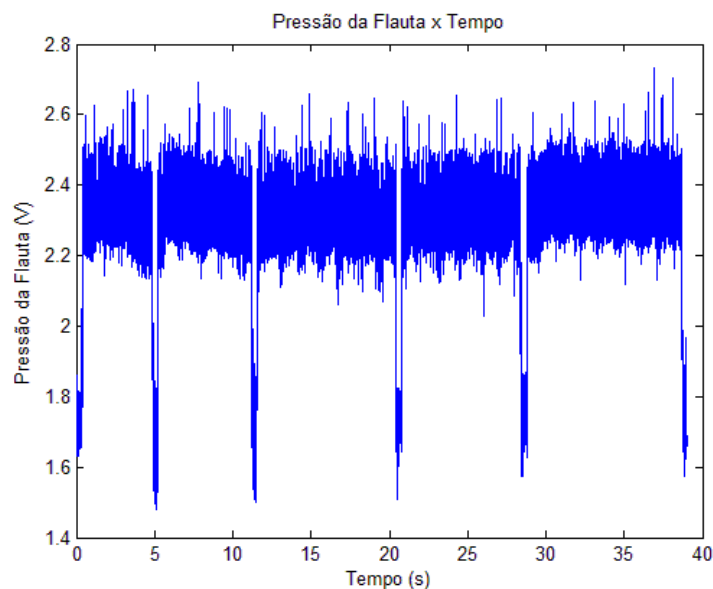
### 3.3.4.3 Pressão da Flauta

Esse parâmetro é lido pelo *kit* comercial para determinar a correção a ser aplicada ao tempo de injeção de GNV em função das variações na pressão da flauta.

Para determinar a correção, imagina-se que no instante em que se determina o tempo de injeção seja necessário se ter um valor médio da pressão na flauta. Para se calcular esse valor médio, um parâmetro importante é qual a **taxa de variação** dessa grandeza, ou seja, em que **faixa de frequência** a grandeza está contida, além da faixa de tensão que essa grandeza ocupa. Da seção 3.3.3.3 sabemos que a faixa de tensão ocupada pelo sinal do sensor de Pressão da Flauta é o apresentado na Equação ( 23 ).

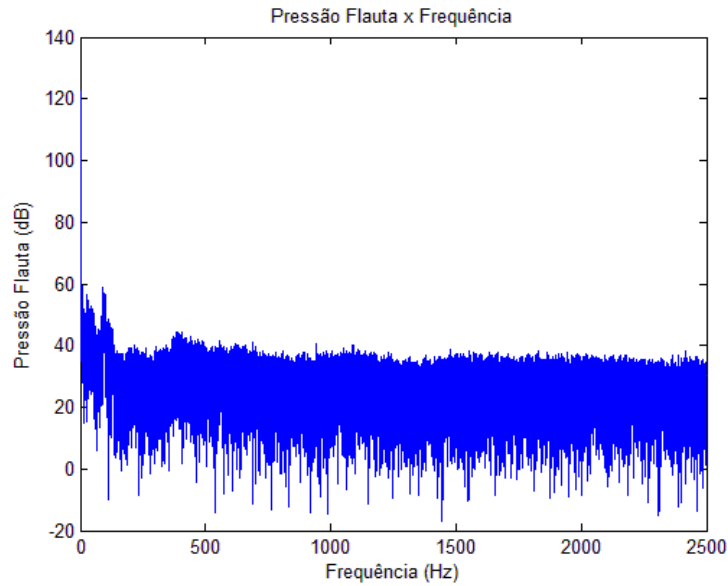
$$\text{Faixa de Tensão}_{\text{Pressão da Flauta}} = [0..3]V \quad ( 23 )$$

Para calcular a faixa de frequência do sinal, primeiramente obtivemos sua forma de onda a partir do arquivo *aceleracao\_0\_100\_trocaMarcha\_3* – vide Figura 45. As quedas no sinal de pressão da flauta acontecem nos instantes de troca de marcha, onde não há injeção de GNV pois não há aceleração. Como a ausência de fluxo de GNV resulta provavelmente em um aumento da pressão de GNV na flauta, isso indica que o sinal de Pressão da Flauta é inversamente proporcional à pressão na flauta.



**Figura 45: Pressão da Flauta**

A partir desses valores, foi calculada a FFT (*fast Fourier transform*) do sinal para se obter seu espectro. O resultado é mostrado na Figura 46.

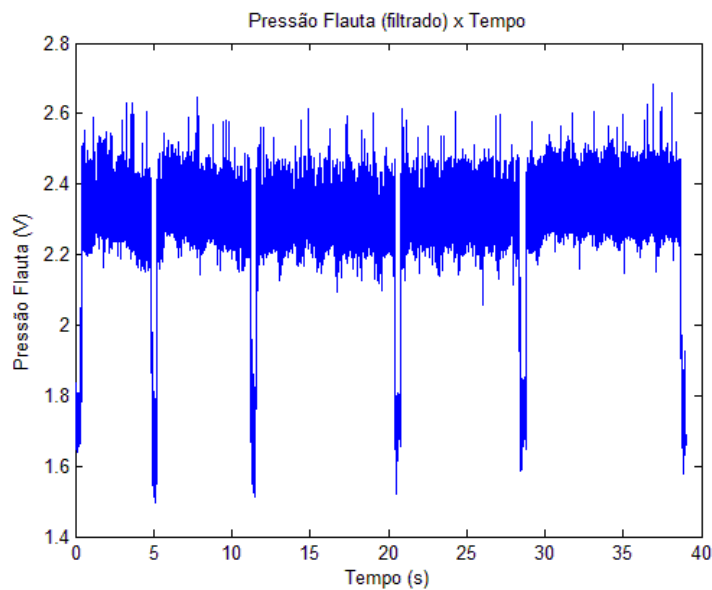


**Figura 46: FFT da Pressão da Flauta**

Foi então calculada uma frequência que abrigasse 85% do espectro, o valor é apresentado na Equação ( 24 ).

$$F_{cortePressão\ da\ Flauta} = 1617\ Hz \quad (24)$$

Para garantir que um filtro RC nessa frequência não descaracterizasse o sinal, eliminando informações importantes, foi feita uma filtragem digital equivalente a uma filtragem RC analógica com frequência de corte igual a  $F_{cortePressão\ da\ Flauta}$ . O resultado é apresentado na Figura 47, e mostra como essa frequência de corte não descaracteriza o sinal.



**Figura 47: Sinal de Pressão da Flauta filtrado**

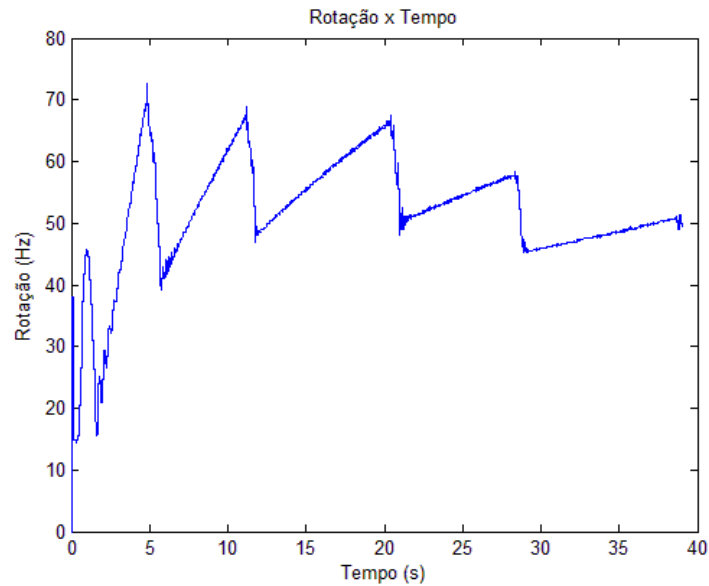
#### 3.3.4.4 Rotação

Esse sinal carrega a informação de rotação do veículo, e é utilizado como entrada para o sistema de controle baseado em lógica nebulosa (*fuzzy*). Os parâmetros importantes desse sinal são sua faixa de frequência – que é de fato a informação útil que esse sinal carrega – e sua amplitude máxima – necessária para o projeto de *hardware*.

Da seção 3.3.3.3 sabemos que a faixa de tensão ocupada pelo sinal do sensor de rotação é o apresentado na Equação ( 25 ).

$$Faixa\ de\ Tensão_{Rotação} = [0..30]V, \text{ com picos acima de } 70V \quad (25)$$

O sinal de rotação foi obtido a partir do arquivo *aceleracao\_0\_100\_trocaMarcha\_3* e processado para que fosse obtido a frequência do sinal, que está ilustrada na Figura 48. As quedas no sinal de rotação acontecem nos instantes de troca de marcha.



**Figura 48: Sinal de Rotação**

Como a rotação foi obtida a partir do sinal de ignição, é de se esperar que a frequência desse sinal seja igual à frequência do motor, ou seja, a rotação do motor nesse exemplo chegou a aproximadamente 70 Hz ou  $70 \cdot 60 = 4.200$  RPM. Considerando uma rotação máxima de 10.000 RPM, podemos considerar uma frequência máxima do sinal de acordo com a Equação ( 26 ).

$$F_{m\acute{a}xima_{Rota\c{c}ao}} = 10.000 \text{ RPM} = \frac{10.000}{60} \text{ Hz} \cong 166,67 \text{ Hz} \quad ( 26 )$$

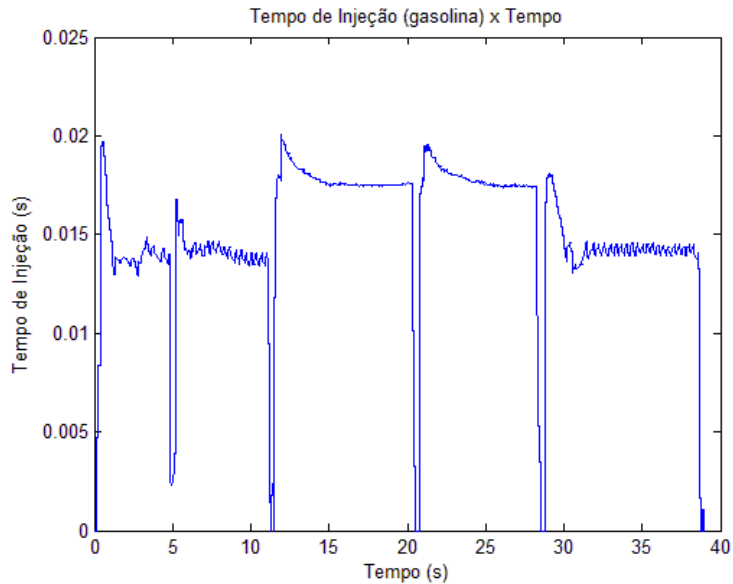
### 3.3.4.5 Bico Original – UCE 1

Esse é talvez o sinal mais importante a ser lido, pois carrega a informação que será utilizada como base para composição do tempo de injeção de GNV: o tempo de injeção de gasolina. As informações mais importantes desse sinal são sua amplitude e os valores máximo e mínimo de tempo de injeção.

Da seção 3.3.3.3 sabemos que a faixa de tensão ocupada pelo sinal do sensor de rotação é o apresentado na Equação ( 27 ).

$$Faixa \ de \ Tens\tilde{a}o_{Bico \ Original-UCE \ 1} = [0..V_{alternador}]V, \text{ com picos de at\acute{e} } 60V \quad ( 27 )$$

O sinal Bico Original – UCE 1 foi obtido a partir do arquivo *aceleracao\_0\_100\_trocaMarcha\_3* e processado para que fosse obtido a duração do sinal, que está ilustrada na Figura 49. As quedas no tempo de injeção acontecem nos instantes de troca de marcha.



**Figura 49: Tempo de Injeção de gasolina**

A partir desse sinal foram obtidos os parâmetros apresentados nas Equações ( 28 ) e ( 29 ).

$$\text{Tempo de Injeção}_{\text{gasolina}_{\text{máximo}}} = 0,0201 \text{ s} = 20,1 \text{ ms} \quad ( 28 )$$

$$\text{Tempo de Injeção}_{\text{gasolina}_{\text{mínimo}}} = 0,0010 \text{ s} = 1,0 \text{ ms} \quad ( 29 )$$

#### 3.3.4.6 Sonda Lambda

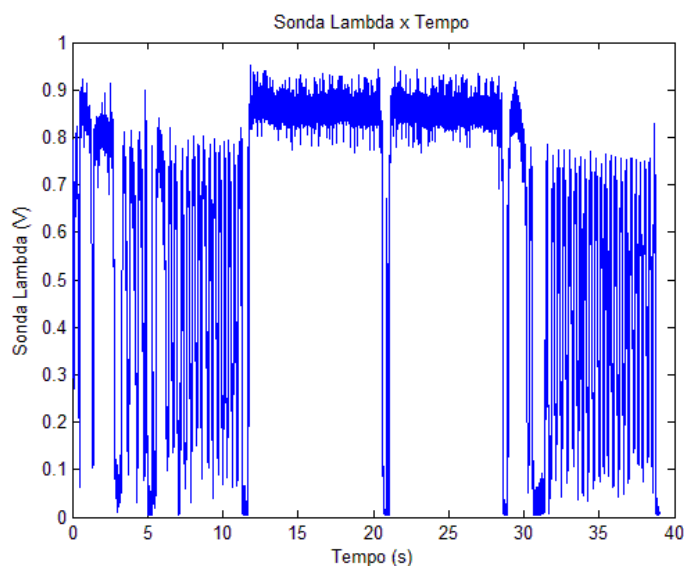
Esse parâmetro informa ao *kit* comercial se a estratégia de injeção está resultando em uma mistura muito rica ou muito pobre (ver seção 2.3.1), o que permite que essa estratégia seja corrigida.

O primeiro fator de interesse é a faixa de tensão que essa grandeza ocupa. Da seção 3.3.3.3 sabemos que a faixa de tensão ocupada pelo sinal do sensor de sonda lambda é o apresentado na Equação ( 30 ).

$$\text{Faixa de Tensão}_{\text{sonda Lambda}} = [0..1]V \quad ( 30 )$$

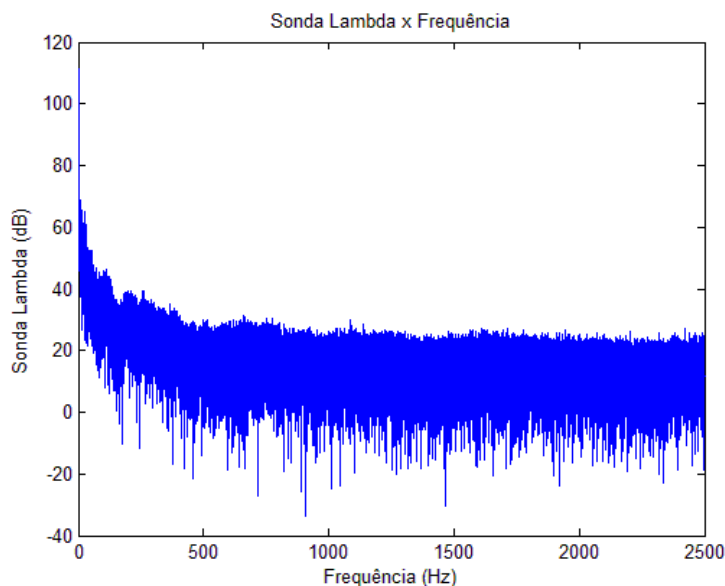
Para que uma filtragem desse sinal não acabe por descaracterizá-lo, um importante parâmetro desse sinal é sua **taxa de variação**, ou seja, em que **faixa de frequência** está contido.

Para calcular a faixa de frequência do sinal, primeiramente obtivemos sua forma de onda a partir do arquivo *aceleracao\_0\_100\_trocaMarcha\_3* – vide Figura 50. Os dois momentos em que o sinal ficou em nível alto aconteceram porque, por algum motivo, a mistura ficou excessivamente rica em combustível.



**Figura 50: Sensor de Sonda Lambda**

A partir desses valores, foi calculada a FFT (*fast Fourier transform*) do sinal para se obter seu espectro. O resultado é mostrado na Figura 51.

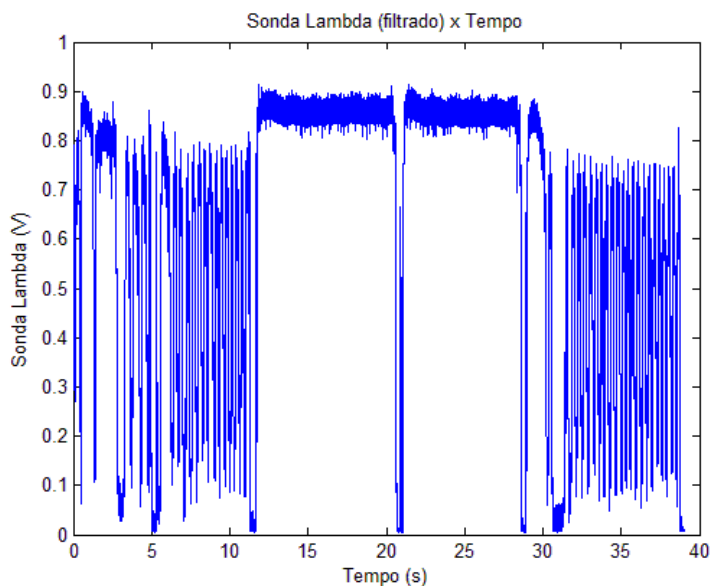


**Figura 51: FFT do sensor de Sonda Lambda**

Foi então calculada uma frequência que abrigasse 85% do espectro, o que está representado na Equação ( 31 ).

$$F_{corte_{Sonda\ Lambda}} = 639\ Hz \quad (31)$$

Para garantir que um filtro RC nessa frequência não descaracterizasse o sinal, eliminando informações importantes, foi feita uma filtragem digital equivalente a uma filtragem RC analógica com frequência de corte igual a  $F_{corte_{Sonda\ Lambda}}$ . O resultado é apresentado na Figura 52, e mostra como essa frequência de corte não descaracteriza o sinal.



**Figura 52: Sensor de Sonda Lambda filtrado**

### 3.3.4.7 MAP

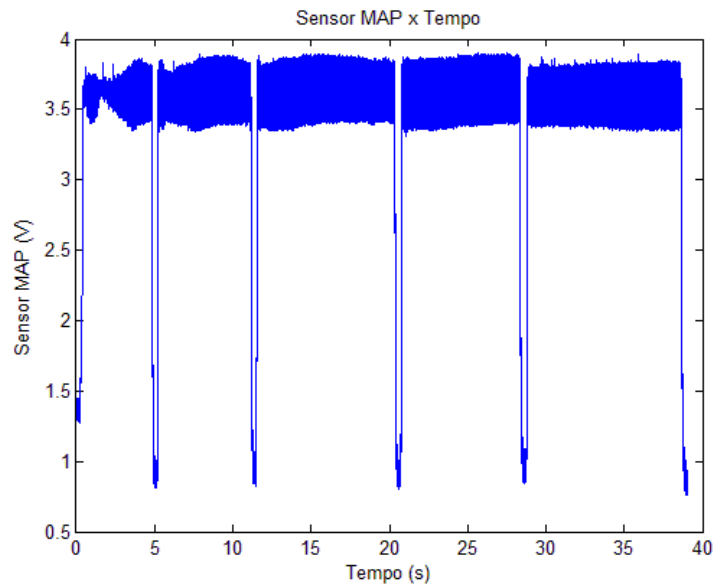
Esse parâmetro informa ao *kit* comercial a qual carga está submetida o motor, ou seja, qual a massa de ar sendo admitida a cada ciclo (ver seção 2.3.1).

O primeiro fator de interesse é a faixa de tensão que essa grandeza ocupa. Da seção 3.3.3.3 sabemos que a faixa de tensão ocupada pelo sinal do sensor de sonda lambda é o apresentado na Equação ( 32 ).

$$Faixa\ de\ Tensão_{Sensor\ MAP} = [0..5]V \quad (32)$$

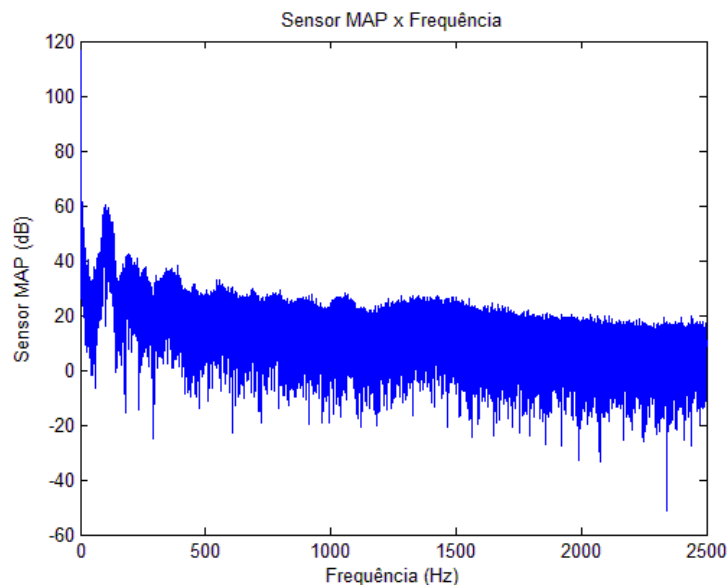
Para que uma filtragem desse sinal não acabe por descaracterizá-lo, um importante parâmetro desse sinal é sua **taxa de variação**, ou seja, em que **faixa de frequência** está contido.

Para calcular a faixa de frequência do sinal, primeiramente obtivemos sua forma de onda a partir do arquivo “aceleracao\_0\_100\_trocaMarcha\_3” – vide Figura 53. Os momentos de queda do sinal do sensor MAP acontecem durante as trocas de marcha.



**Figura 53: Sensor MAP**

A partir desses valores, foi calculada a FFT (*fast Fourier transform*) do sinal para se obter seu espectro. O resultado é mostrado na Figura 54.

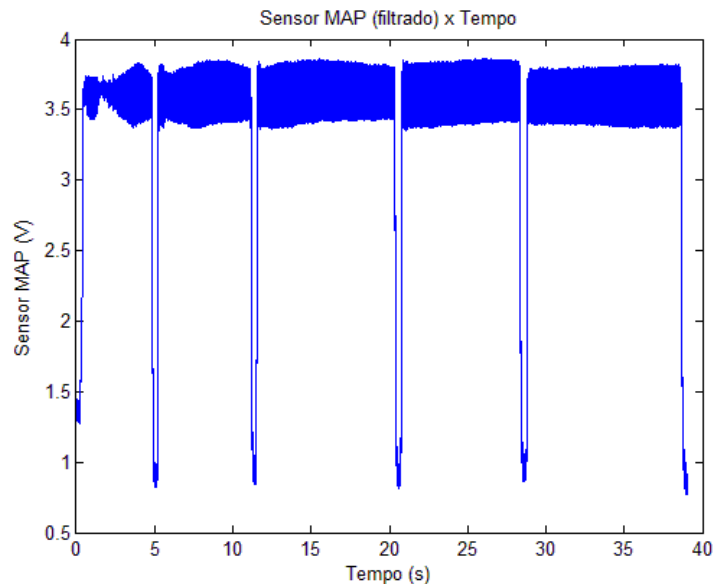


**Figura 54: FFT do Sensor MAP**

Foi então calculada uma frequência que abrigasse 85% do espectro, valor representado na Equação ( 33 ).

$$F_{corte_{Sensor\ MAP}} = 447\ Hz \quad (33)$$

Para garantir que um filtro RC nessa frequência não descaracterizasse o sinal, eliminando informações importantes, foi feita uma filtragem digital equivalente a uma filtragem RC analógica com frequência de corte igual a  $F_{corte_{Sensor\ MAP}}$ . O resultado é apresentado na Figura 55, e mostra como essa frequência de corte não descaracteriza o sinal.

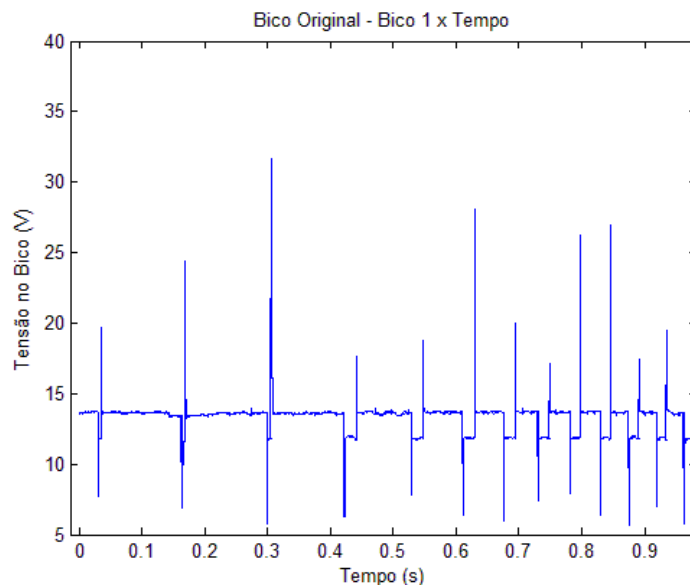


**Figura 55: Sensor de Sonda Lambda filtrado**

### 3.3.4.8 Bico Original – Bico 1

Quando o *kit* comercial congela os bicos injetores de gasolina, o que ele de fato faz é inserir uma resistência em série com o bico (entre os sinais Bico Original – Bico 1 e Bico Original – UCE 1) para limitar a corrente que passa pela solenóide, fazendo assim com que não haja força suficiente para o acionamento. Esse sinal irá coincidir com o sinal Bico Original – UCE 1 quando o veículo estiver funcionando no combustível original, e quando o veículo estiver funcionando no GNV esse sinal será um divisor resistivo entre o a resistência que “congela” o bico injetor e a resistência do bico injetor. Esse sinal foi capturado apenas para que a informação de “congelamento” dos bicos fosse obtida, e não deve ser lido no sistema que será fruto deste trabalho, portanto não há necessidade de se levantar parâmetros desse sinal.

Apenas para caráter ilustrativo, a Figura 56 traz a forma de onda desse sinal quando o bico injetor original está congelado (arquivo *aceleracao\_0\_100\_trocaMarcha\_3*).



**Figura 56: Bico Original - Bico 1**

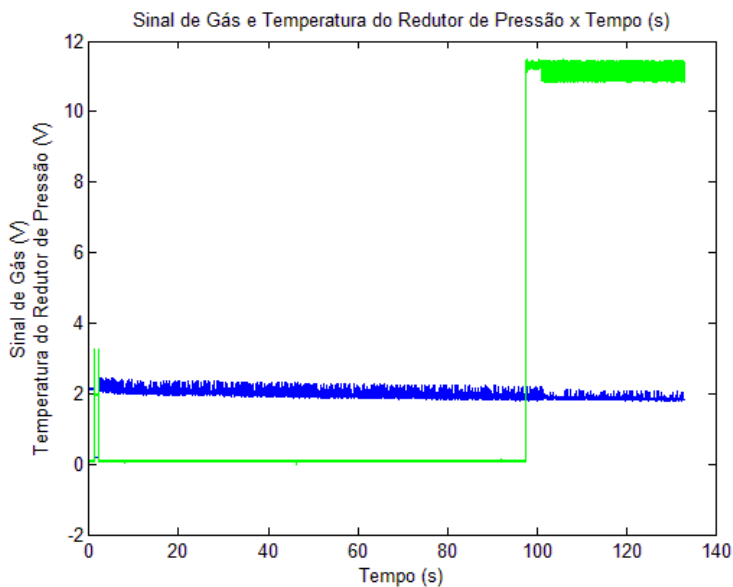
### 3.3.4.9 Temperatura do Redutor de Pressão

A temperatura do redutor de pressão é utilizada para determinar em qual momento o sistema de GNV pode passar a funcionar. Como a medida é feita a partir de um termistor NTC (*Negative Temperature Coefficient*), imagina-se que o *kit* comercial faça uma transformação dessa resistência para tensão, que foi o valor lido e armazenado nos arquivos.

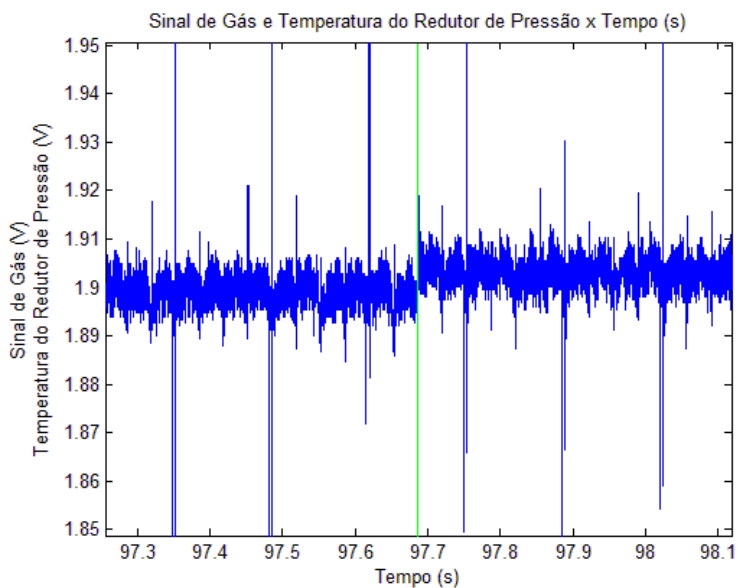


Para determinar qual a tensão presente nesse sinal quando o Sinal de Gás passa a alimentar o sistema GNV, a forma de onda de ambos foi obtida a partir do arquivo *comutacao\_gasolina\_Gas\_1* e está representada nas figuras: Figura 57 e Figura 58. Pode-se observar então que a comutação ocorre ao redor de 97s depois de o veículo ser ligado e que o valor no sinal de temperatura do redutor de pressão é igual representado na Equação ( 34 ).

$$Temperatura\ do\ Redutor\ de\ Press\tilde{a}o_{comut\tilde{a}o\ para\ GNV} \cong 1,9V \tag{34}$$



**Figura 57: Sinal de Gás e Temperatura do Redutor de Pressão**



**Figura 58: Sinal de Gás e Temperatura do Redutor de Pressão – Ampliação**

# 4 O PROJETO DO *HARDWARE* DO CONTROLADOR

O *hardware* de um produto é a porção física do seu circuito eletrônico, a parte que pode ser tocada. É o *hardware* que provê os meios necessários para a realização da tarefa à qual o produto se propõe, e é em cima do *hardware* que o *firmware* (ou *software*) é concebido (quando há alguma unidade de processamento). O *hardware* é responsável por integrar todas as partes do circuito, como memórias, unidades de processamento, interfaces digitais, interfaces analógicas e sensores, entre outros.

Em um projeto de *hardware*, muitos cuidados devem ser tomados para garantir o seu correto funcionamento nas diversas condições de uso para o qual tenha sido especificado o seu funcionamento. Entre esses cuidados podemos citar cuidados com as interfaces elétricas, com os parâmetros máximos dos componentes não sendo excedidos (tensão, corrente, potência), cuidados com a integridade dos sinais que trafegam pelas trilhas, cuidados com impedâncias controladas de trilhas, com emissão e interferência eletromagnética e etc.

Esta seção descreve o *hardware* que foi desenvolvido como fruto deste trabalho e as estratégias e especificações que levaram a ele.

## 4.1 Separação dos Circuitos Digitais e Analógicos dos Circuitos de Potência

Em uma **PCI (Placa de Circuito Impresso)** as conexões elétricas feitas pelas trilhas e/ou planos de cobre tem uma impedância idealmente nula. Na prática, entretanto, sinais de alta frequência (dezenas de megahertz para cima) ou de alta corrente (dezenas de ampéres para cima) podem causar uma significativa queda de tensão (na ordem de centenas de milivolts) ao longo das trilhas - isso é especialmente verdadeiro quando o sinal é de alta frequência e de alta corrente.

Por esse motivo, é aconselhável que se separem os planos de terra das partes do circuito com características diferentes:

- Sinais analógicos;
- Sinais de áudio;
- Sinais digitais;
- Sinais de potência;
- Etc.

Além disso, é interessante se ter diferentes PCIs para diferentes funções quando o projeto está na fase de desenvolvimento, pois dessa forma quando um circuito for modificado somente sua PCI precisa ser refeita, não havendo necessidade de compra e montagem de novos componentes da outra parte do circuito.

Pelos motivos expostos acima, decidiu-se que seriam feitas duas PCIs: uma para os circuitos digitais e analógicos (seção 4.2) e outra para a parte de potência (seção 4.3).

### 4.1.1 Disposição do SINAL\_GAS

Esse sinal é responsável por alimentar e indicar a todo o sistema de gás que o veículo deve funcionar a GNV. Para isso, desempenha as seguintes funções:

- Alimenta os bicos injetores de GNV;
- Alimenta o redutor de pressão;
- Alimenta a válvula solenóide que libera a passagem de GNV;
- Indica ao variador de avanço para iniciar o avanço de ignição;

Esse sinal é obtido a partir do sinal da bateria, e por alimentar todo o sistema GNV deve possuir capacidade de fornecer uma grande corrente, e, portanto, deverá ficar na placa de potência a ser desenvolvida (ver seção 4.3.3)

## 4.2 PCI – Circuitos digitais e analógicos

Nessa seção, cada parte da PCI desenvolvida que irá abrigar os circuitos digitais e analógicos é explicada com detalhes. Essa PCI é responsável por realizar a leitura dos sensores e do sinal de acionamento do bico injetor original, processar essa informação por uma lógica nebulosa e gerar um pulso digital com a largura adequada de acionamento

do bico injetor de GNV. Esse pulso será utilizado pela PCI responsável pela parte de potência (ver seção 4.3) para realizar o acionamento dos bicos injetores no tempo adequado.

#### 4.2.1 Especificações da PCI – Circuitos digitais e analógicos

Para pautar a criação de qualquer *hardware*, é adequado que se faça uma especificação técnica do *hardware* que se deseja, especificação essa que irá guiar todas as tomadas de decisão relativas ao desenvolvimento desse *hardware*. Esta seção traz o conjunto de especificações técnicas necessárias ao correto funcionamento do *hardware* e ao cumprimento das funções ao qual ele se propõe.

As especificações são:

- Tensão de entrada de 9 a 16V: o que irá garantir o correto funcionamento do circuito mesmo quando a bateria está bastante descarregada ou quando o alternador está com defeito;
- Resistente aos surtos automotivos da norma ISO 7637;
- Dotado da capacidade de inserir uma impedância em série com os bicos injetores originais quando esses forem congelados – isso irá limitar a corrente nesses bicos e evitar a injeção do combustível original do veículo. A potência desse resistor deve ser de suficiente, para garantir uma tensão rms (*root mean square*) sobre eles de até 14V;
- Capaz de realizar a leitura do sinal dos bicos injetores mesmo quando congelados, considerando que o transiente desse sinal possa atingir valores acima de 60V;
- Capaz de realizar a leitura do sinal do sensor MAP de até 5V @ 447Hz;
- Capaz de emular o sinal do sensor MAP, indo até 5V @ 447Hz;
- Capaz de realizar a leitura do sinal do sensor de Sonda Lambda de até 1V @ 639Hz;
- Capaz de emular o sinal do sensor de Sonda Lambda, indo até 1V @ 639Hz;
- Capaz de realizar a leitura do sinal de rotação com boa imunidade a ruído para que não ocorram leituras falsas;
- Possuir saídas genéricas capazes de fornecer até 5A na tensão da bateria;
- Possuir uma memória não volátil de 8KB para armazenagem de configurações e mapas;
- Possuir conector para gravação do *firmware* in-circuit;
- Possuir conector para *debug* via comunicação serial;
- CPU com as seguintes características:
  - Facilidade de aquisição;
  - Ferramentas de desenvolvimento de baixo custo;
  - Número suficiente de entradas e saídas digitais;
  - Interface de comunicação com a memória não volátil;
  - Pelo menos 2 conversores A/D (sensores MAP e de Sonda Lambda);
  - Pelo menos uma UART (*debug* e configuração);
  - Pelo menos 3 timers, existindo pelo menos 3 canais de PWM (emulação dos sensores MAP e de Sonda Lambda, geração do sinal de *hold* de acionamento dos bicos e medição de temporização internamente);
  - *Clock* de pelo menos 16MHz;
  - Memória flash de pelo menos 32KB;
  - Memória RAM de pelo menos 2KB;
- Área de prototipagem para testes de engenharia.

## 4.2.2 Proteções de Entrada

O ambiente elétrico automotivo pode ser bastante severo, com bastante ruído na linha de alimentação, e a norma que descreve os pulsos que podem ocorrer nesse ambiente é a ISO7637-2. Nessa norma, o pulso mais severo é o *Load Dump* (Figura 59), por ser um pulso de alta tensão e duração.

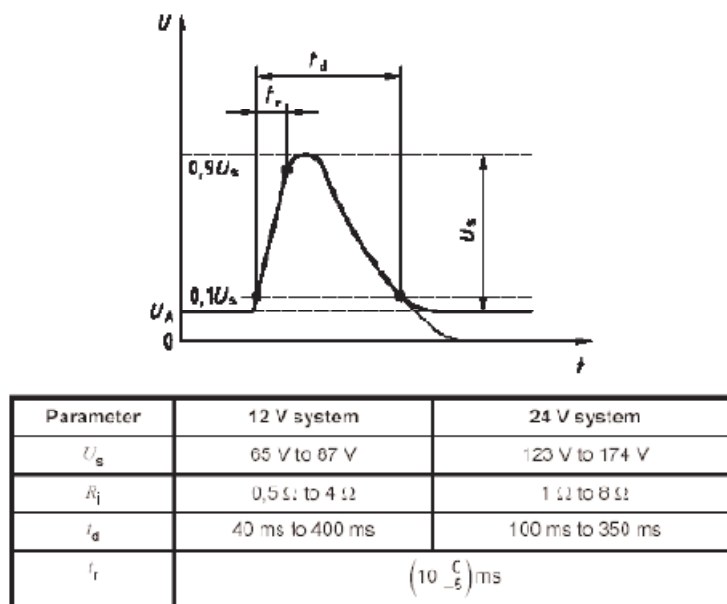


Figura 59: Load Dump

Para proteger o circuito contra esse pulso, foi utilizado um supressor de tensão conhecido como tranzorb da *LittelFuse*, o SLD24-018. Esse componente é especificado para suportar o pulso *Load Dump* e limita a tensão em aproximadamente 32V. Para mais informações, consultar o seu *datasheet*.

Além disso, foram inseridos no projeto os seguintes itens de proteção:

- Diodo D13 – protege o circuito contra inversão de polaridade;
- Fusível PPTC F1 – desconecta o circuito da alimentação em caso de sobre-corrente;
- Resistor R35 – previsto para limitar a corrente sobre o tranzorb durante o surto, mas montado como 0R a princípio (ou seja, sem funcionalidade).
- Varistor VR1 – previsto caso o tranzorb não se mostre eficaz o suficiente, é um supressor de tensão com características diferentes do tranzorb (mais lento e capaz de suportar maiores correntes).

O resultado pode ser visto na Figura 60.

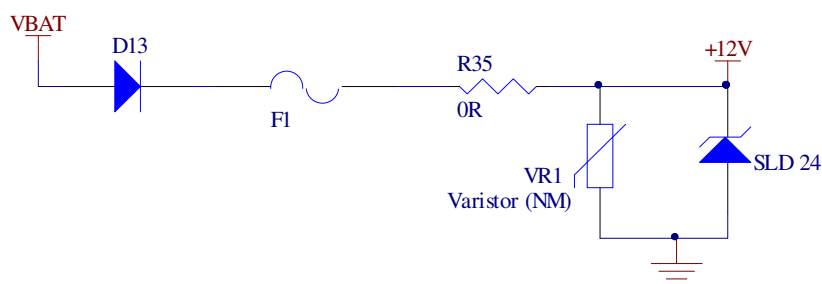


Figura 60: Esquemático do circuito de proteção de entrada.

## 4.2.3 Fonte de Alimentação

A fonte de alimentação é responsável por fornecer uma alimentação estável para todo o circuito alimentado por 5 V, para isso, seu projeto deve prever a capacidade de fornecer a soma de todas as correntes exigidas. Abaixo se encontram os principais circuitos aos quais a tensão de 5V se conecta e a corrente máxima de cada um deles:

Descrição	Part Number	Corrente Máxima
Microcontrolador	ATMega16-16PU	15 mA
Memória EEPROM	24LC64	3 mA
Inversor Lógico	74HC14	50 mA
CI Leitor de RPM	LM1815	6 mA
Pull-ups	Resistores diversos	5 mA
	<b>Total de Corrente</b>	79 mA

A fonte de 5 V também é ligada a dois conectores diferentes:

- Conector para gravação *in-circuit* do *firmware* – não se sabe exatamente qual corrente pode ser drenada nesse caso, pois isso depende da gravadora. Uma corrente de 100 mA deve ser mais que o suficiente.
- Conector para Entradas e Saídas digitais – nesse conector foram inseridas as entradas e saídas do microcontrolador não utilizadas em outras partes do circuito e a alimentação (5V + GND). Esse conector deve servir a propósitos de engenharia na fase de desenvolvimento, e portanto sua corrente deve ser arbitrada. Nesse caso, optou-se por uma corrente máxima de 200 mA.

A soma total das correntes máximas dos circuitos que se ligam ao 5 V com as correntes nos conectores é calculada na Equação ( 35 ).

$$I_{5V_{máxima}} = 79mA + 100mA + 200mA = 379mA \quad (35)$$

Considerando uma margem de 30% de segurança, a corrente especificada para a fonte de 5 V é calculada na Equação ( 36 ).

$$I_{5V} = 379mA * (1 + 0,3) = 492,7mA \quad (36)$$

Fazendo uma rápida busca paramétrica, encontrou-se a fonte utilizada para essa aplicação: **LM2576HVT-ADJ**. As principais características desse CI (Circuito Integrado) são:

- Corrente de saída de 3 A garantida;
- Ampla faixa de tensão de entrada: até 60V;
- Requer apenas 4 componentes externos;
- Oscilador interno com frequência fixa de 52 kHz;
- Capacidade de realizar desligamento com sinal em nível TTL, baixo consumo em modo *standby*;
- Alta eficiência;
- Usa indutores padrão facilmente encontrados;
- Desligamento por sobre-temperatura e proteção contra sobre-corrente;

Uma característica muito interessante desse CI, que também influenciou em sua escolha, é a facilidade em se selecionar os componentes externos necessário ao seu funcionamento. O próprio *datasheet* do componente traz um passo a passo dessa seleção, com os seguintes tópicos:

- *Programming Output Voltage*: ensina como selecionar os resistores de *feedback* para configurar a tensão de saída correta;
- *Inductor Selection (L1)*: ajuda na seleção do indutor da fonte;
- *Output Capacitor Selection (COUT)*: ajuda na seleção do capacitor de saída da fonte;
- *Catch Diode Selection (DI)*: ajuda na seleção do diodo da fonte;
- *Input Capacitor (CIN)*: ajuda na seleção do capacitor de entrada da fonte;

O *datasheet* traz ainda alguns “*Application Hints*” (dicas de aplicação), que são importantes para o projeto da fonte.

De posse desse procedimento presente no *datasheet* do componente, e ainda considerando os parâmetros abaixo, chegou-se ao projeto da Figura 61.

- Tensão de saída: 5 V;
- Tensão de entrada máxima: 16 V;
- Corrente de saída: 500 mA;

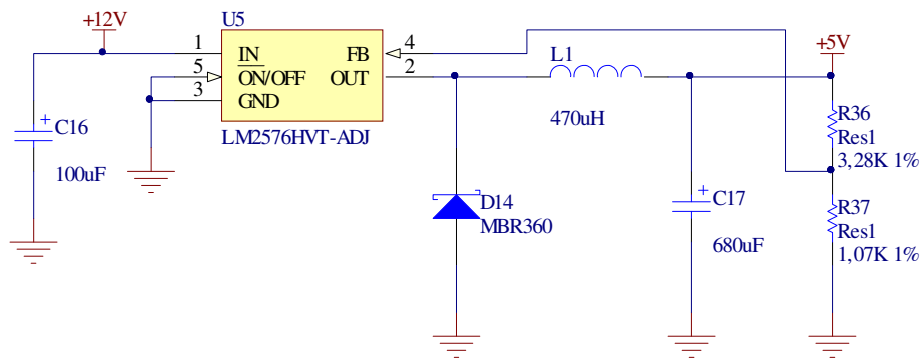


Figura 61: Esquemático da Fonte.

#### 4.2.4 CPU

O primeiro ponto a se abordar nesse tópico é a seleção do microcontrolador. Abaixo, detalhamos melhor cada uma das especificações de *hardware* descritas na seção 4.2.1 que dizem respeito ao microcontrolador:

- *Facilidade de aquisição*: muitas vezes, a aquisição de componentes eletrônicos no Brasil pode ser bem cara e demorada, pois esses componentes devem ser importados com alto custo de frete e imposto. Por esse motivo, estabeleceu-se que o microcontrolador a ser utilizado no projeto do *hardware* deveria ser de fácil aquisição no Brasil;
- *Ferramentas de desenvolvimento de baixo custo*: uma ferramenta de programação e debug de um microcontrolador pode possuir um alto custo dependendo do microcontrolador utilizado, e isso deve ser levado em consideração na seleção do microcontrolador a ser utilizado;
- *Número suficiente de entradas e saídas digitais*: o microcontrolador deve disponibilizar pelo menos:
  - *5 sinais de entradas*:
    - *1 Entrada Digital*: para leitura do sinal de rotação – é preferível que a entrada ao qual se conecta esse sinal possua a capacidade de causar uma interrupção de *software* para facilitar a leitura de frequência;
    - *4 Entradas Digitais*: para leitura dos sinais de injeção – também preferivelmente conectados a entradas capazes de causar uma interrupção de *software*;
  - *12 sinais de Saídas*:
    - *8 Saídas Digitais*: para acionamento dos bicos injetores (2 sinais para cada bico);
    - *4 Saídas Digitais*: para acionamento das saídas genéricas;
- *Interface de comunicação com a memória não volátil*: a interface de comunicação com a memória não volátil escolhida foi a I2C pelos motivos listados a seguir.
  - Existem memórias de fácil acesso no mercado que usam esse protocolo;
  - São fáceis de serem escritas e lidas, não exigindo níveis de tensão muito específicos.
- *Pelo menos 3 conversores A/D*: para leitura dos sinais de sensor de sonda lambda, sensor MAP e sensor de pressão da flauta;
- *Pelo menos uma UART*: para configuração dinâmica e debug do código;

- *Pelo menos 3 timers, existindo pelo menos 3 canais de PWM*: os *timers* são necessários para o funcionamento do *firmware* e serão explicados com mais detalhes na seção 6.4. Já os canais de PWM têm o propósito de:
  - Emular os sensores MAP e de Sonda Lambda, que são criados passando os sinais de PWM por filtros analógicos passa-baixas;
  - Gerar o sinal de *hold* da estratégia *Peak and Hold* (ver seção 3.3.4.2) de acionamento dos bicos injetores de gás.
- *Clock de pelo menos 15MHz*: esse é o *clock* mínimo estimado para que o microcontrolador consiga cumprir suas funções.
- *Memória Flash de pelo menos 32KB e Memória RAM de pelo menos 2KB*: essas são as memórias mínimas estimadas para comportar e executar o código necessário para que o microcontrolador consiga cumprir suas funções;

Com base nos requisitos descritos acima, o microcontrolador selecionado foi o **ATMega32-16PU** da Atmel, que possui as seguintes características:

- *Facilidade de aquisição*: pode ser comprado em sites de compra, as unidades utilizadas no desenvolvimento foram compradas no mercado livre – [www.mercadolivre.com.br](http://www.mercadolivre.com.br);
- *Ferramentas de desenvolvimento de baixo custo*: a Atmel possui uma ferramenta para programação e debug do **ATMega32** com baixo custo. Trata-se do **AVRDragon**, que sai ao preço de ~US\$ 100,00 no Brasil pela CN Tecnologia;
- *Número suficiente de entradas e saídas digitais*: o microcontrolador possui 40 pinos, sendo que para esse projeto os seguintes pinos estarão ocupados e, portanto, indisponíveis para serem utilizados como entradas e saídas digitais:
  - 4 pinos de alimentação – VCC (digital), AVCC (analógica), e 2 GNDs;
  - 1 pino de referência analógica – AREF;
  - 2 pinos para conexão do cristal oscilador;
  - 1 pino de reset;
  - 4 pinos de gravação – TCK, TDO, TMS e TDI (o pino de reset também é utilizado para gravação);
  - 2 pinos para comunicação I2C – SDA e SCL;
  - 3 pinos para leitura dos sinais analógicos: MAP, Sonda Lambda e Pressão de Flauta;
  - 2 pinos para comunicação RS232 de debug – TX e RX;
  - 4 pinos para geração de PWM, sendo que 2 foram utilizados para gerar os sinais emulados dos sensores MAP e de sonda lambda e 1 para gerar o sinal de *hold* da estratégia *Peak and Hold*;

Dessa forma, sobram 18 pinos para entradas e saídas digitais (=40-4-1-2-1-4-2-3-2-3). Esses pinos foram divididos da seguinte forma:

- 5 sinais de entradas
  - 4 entradas de leitura de tempo de injeção;
  - 1 entrada de leitura de rotação;
- 8 sinais de Saídas – 2 para cada pino de injeção (ver seção 4.2.11);
- 4 sinais de saída genéricos – a serem utilizados durante a fase de engenharia caso necessário;
- 1 sinal de GÁS – que avisa ao sistema de GNV (tanto circuitos internos da PCI quanto dispositivos externos) que o veículo está funcionando a gás;
- *3 pinos com capacidade de causar interrupção de software*: INT0, INT1 e INT2. O INT0 sendo utilizado para fazer a medição de rotação;
- *Uma interface de comunicação I2C*: utilizada para comunicação com a memória externa EEPROM;
- *Um conversor A/D com 8 canais*: utilizado para leitura dos sinais de sensor MAP, sensor de Sonda Lambda e sensor de Pressão da Flauta;

- *Uma USART*: que possui todas as funcionalidades de uma UART, mas que também pode ser utilizada para comunicação síncrona. Utilizada para servir de console;
- *Três timers*:
  - 1 *timer* de 16 bits com 1 canal de PWM;
  - 1 *timer* de 8 bits com 1 canal de PWM;
  - 1 *timer* de 8 bits com 2 canais de PWM.

Totalizando 4 canais de PWM;

- *Clock máximo de 16MHz*;
- *Memória Flash de 32KB*;
- *Memória RAM de 2KB*;
- Outras características não pertinentes à necessidade desse projeto, como:
  - *Real Time Counter* com oscilador independente;
  - SPI;
  - Comparador analógico;

A Figura 62 ilustra o esquema elétrico de conexões da CPU. Basicamente, o circuito pode ser dividido da seguinte maneira:

- *CPU\_OUT\_INJx* e *CPU\_OUT\_INJx\_2*: sinais de saída de injeção. Esses sinais são conectados ao circuito de geração do sinal de acionamento, responsável por fazer a abertura do bico de injeção de gás (seção 4.2.11);
- *TCK*, *TDO*, *TMS*, *TDI*: sinais ligados utilizados para gravação e *debug* da CPU;
- *RXD* e *TXD*: ligados ao conector “JP3” para serem utilizados para comunicação RS232. Esse conector ainda conta com pinos de alimentação para suprir um eventual circuito externo;
- *CPU\_INT0*: para leitura do sinal de rotação. Esse pino é capaz de causar interrupção por *software*;
- *CPU\_PWM0*: canal A do *Timer 1*, utilizado para emular o sinal do sensor MAP;
- *CPU\_PWM1*: canal B do *Timer 1*, utilizado para emular o sinal do sensor de sonda lambda;
- *CPU\_PWM2*: canal de PWM do *Timer 2*, utilizado para gerar o sinal de acionamento em *hold* juntamente com os sinais *CPU\_OUT\_INJx* e *CPU\_OUT\_INJx\_2* (seção 4.2.11);
- *NET\_RESET*: pino utilizado para *reset* do microcontrolador;
- *XTAL1* e *XTAL2*: pinos para conexão com o cristal oscilador externo de 16MHz;
- *VCC*, *AVCC* e *GND*s: pinos de alimentação do microcontrolador;
- *CPU\_IN\_INJx*: sinais de leitura do sinal de injeção da UCE original;
- *NET\_CPU\_SCL* e *NET\_CPU\_SDA*: sinais de comunicação I2C;
- *NET\_CPU\_OUT\_SGx*: sinais de acionamento do circuito de saída genérica;
- *NET\_CPU\_ADC0*: sinal de leitura do sensor MAP;
- *NET\_CPU\_ADC1*: sinal de leitura do sensor de sonda lambda;
- *NET\_CPU\_ADC2*: sinal de leitura do sensor de Pressão da Flauta;



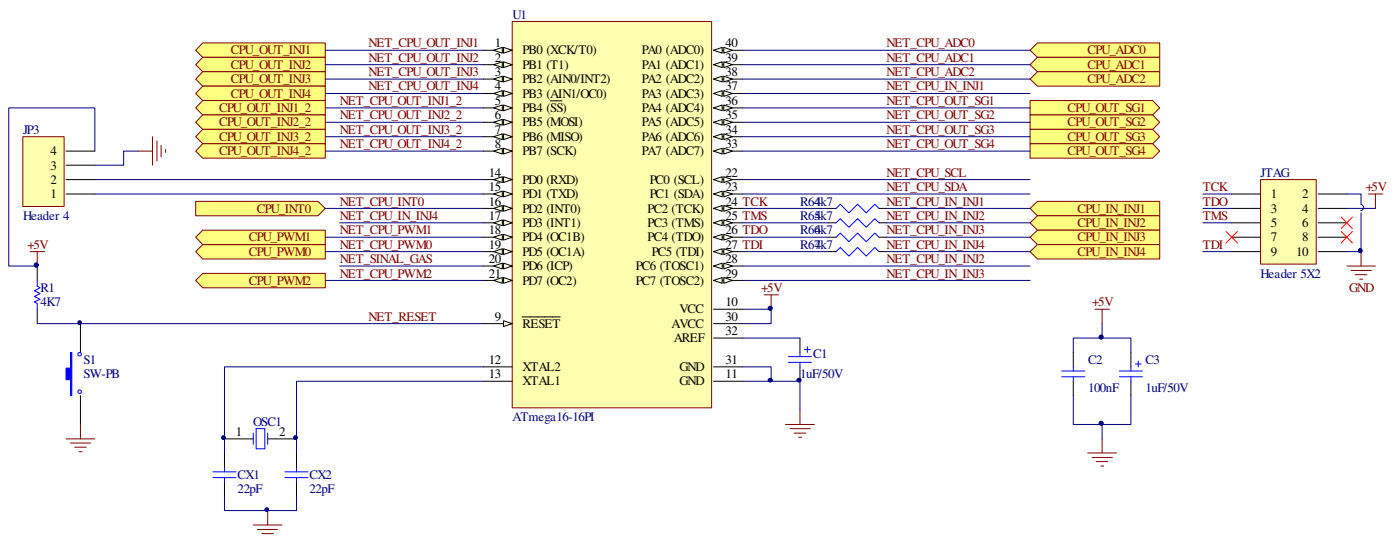


Figura 62: Esquema da CPU

#### 4.2.5 Leitura e Emulação do Sensor MAP

Os sensores MAP possuem geralmente uma saída de 0 a 5V[29], adicionalmente, testes mostram que as componentes de frequência principais desse sinal estão abaixo de 447 Hz (ver seção 3.3.4.7). Esses parâmetros foram os que alimentaram o projeto da Figura 63.

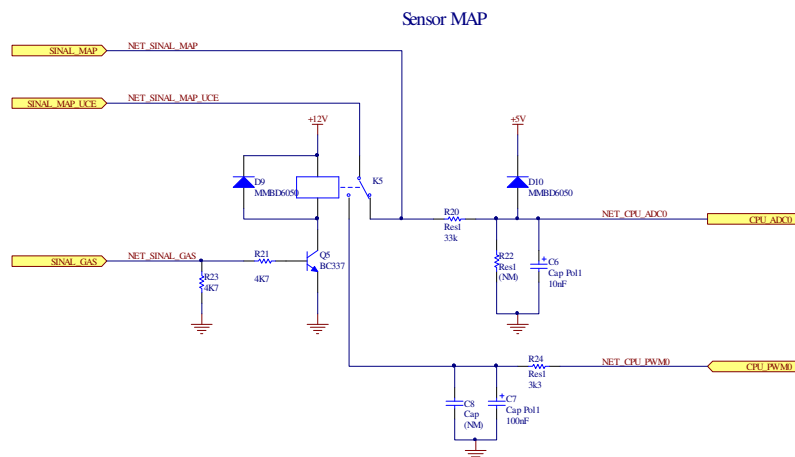


Figura 63: Circuito de leitura, emulação e comutação do Sensor MAP

Os sinais mostrados nesse circuito são:

- *SINAL\_MAP*: sinal provindo do sensor MAP, direto do conector para o mundo externo;
- *SINAL\_MAP\_UCE*: sinal que irá se conectar a entrada de sensor MAP da UCE, direto para o conector para o mundo externo;
- *SINAL\_GAS*: sinal que indica que o sistema de gás está em funcionamento e que, portanto, determina a comutação ou não do relé e define qual será o sinal que se conectará ao mundo externo: o sinal original do sensor MAP ou o sinal emulado do sensor MAP. Ver seção 4.1.1;
- *CPU\_ADC0*: sinal que se conecta à entrada analógica do microcontrolador;
- *CPU\_PWM0*: sinal que se conecta à saída PWM do microcontrolador, que irá criar o sinal emulado do sensor MAP.

O fio que conecta o sensor MAP à UCE deve ser interrompido (cortado) de forma que uma das extremidades será conectada a *SINAL\_MAP* e a outra a *SINAL\_MAP\_UCE*. Caso seja utilizado um módulo “Variador de Avanço” externo, o fio que liga o sensor MAP à UCE não deve ser interrompido e sim derivado para se conectar a *SINAL\_MAP* (para permitir a leitura desse sensor) e *SINAL\_MAP\_UCE* deve ficar desconectado.

Esse circuito pode ser dividido em três partes, são elas:

- *Circuito de leitura:*

- O circuito de leitura é bastante simples, formado por um filtro RC (resistor R20 + capacitor C6) e um diodo de “clamp” (D10), responsável por garantir que o sinal não ultrapasse o limite do microcontrolador.

A frequência de corte de um filtro RC é calculada pela Equação ( 37 ).

$$F_{corte} = \frac{1}{2\pi RC} \quad (37)$$

Considerando uma frequência de corte de 447 Hz e os valores comerciais de resistor de capacitor, definiu-se: R = 33000 Ω e C = 10 nF. Com esses parâmetros, a frequência de corte é calculada pela Equação ( 38 ).

$$F_{corte} = \frac{1}{2\pi * 33000 * 10 * 10^{-9}} = 482,29 \text{ Hz} \quad (38)$$

Outra preocupação diz respeito à potência do resistor, que por ser um componente SMD suporta apenas 0,1W, e deve-se garantir que essa potência não seja excedida. A potência de um resistor pode ser calculada pela Equação ( 39 ).

$$P = \frac{V^2}{R} \quad (39)$$

Isolando V na Equação ( 39 ), obtemos a Equação ( 40 ).

$$V = \sqrt{P * R} \quad (40)$$

Considerando os parâmetros: P = 0,1 W e R = 33000 Ω, obtemos a tensão da Equação ( 41 ).

$$V = \sqrt{0,1 * 33000} = 57,45 \text{ V} \quad (41)$$

Isso significa que a tensão máxima entre os terminais do resistor deve ser de 57,45 V, e como a tensão em um dos lados está limitada em 5,7 V (5 V da tensão de clamp + 0,7 V da queda do diodo MMBD6050) a tensão máxima de entrada pode ser 57,45 V + 5,7 V = 63,15 V. Essa tensão é muito acima do que se espera de uma tensão de um sensor MAP, o que faz o projeto robusto o suficiente desse ponto de vista.

O último parâmetro que deve ser levado em consideração é a corrente máxima suportada pelo diodo MMBD6050: 200 mA. Considerando que toda a corrente que passa por esse diodo passa também pelo resistor (já que a tensão drenada pelo microcontrolador é desprezível), a relação entre a tensão de entrada e a corrente passando pelo resistor (e conseqüentemente pelo diodo) pode ser expressa pela Equação ( 42 ).

$$I = \frac{V - V_{clamp}}{R} \quad (42)$$

Isolando V na Equação ( 42 ), obtemos a Equação ( 43 ).

$$V = I * R + V_{clamp} \quad (43)$$

Considerando os parâmetros: V<sub>clamp</sub> = 5,7 V, R = 30000 Ω e I = 200 mA, obtemos a Equação ( 44 ).

$$V = 200 * 10^{-3} * 33000 + 5,7 = 6605,7 \text{ V} \quad (44)$$

Ou seja, a tensão de entrada necessária para exceder a corrente máxima suportada pelo diodo MMBD6050 é de 6605,7 V (desnecessário dizer que esse é um valor irreal para um sinal de sensor MAP).

Por fim, o resistor não montado R22 é previsto para casos em que o Sensor MAP tenha sinais acima de 5V e que, portanto, precisem ser atenuados antes de serem amostrados pelo microcontrolador sem serem limitados pelo diodo.

- *Circuito de emulação:*

- O circuito de emulação é ainda mais simples, composto apenas por um filtro RC (resistor R24 + capacitor C7). A frequência desse filtro considerando os parâmetros ( $R = 3300 \Omega$  e  $C = 100 \text{ nF}$ ) é mostrada na Equação ( 45 ) e é idêntica ao filtro de entrada, como era de se esperar.

$$F_{corte} = \frac{1}{2\pi * 3300 * 100 * 10^{-9}} = 482,29 \text{ Hz} \quad (45)$$

A frequência é a mesma porque o sinal emulado deve possuir características semelhantes ao sinal original.

Outro importante parâmetro nesse filtro é a frequência gerada pelo PWM, que deve ser bastante atenuada para que esteja presente o mínimo possível no sinal de saída. A atenuação do filtro RC em decibéis em função da frequência do sinal de entrada é expressa pela Equação ( 46 ):

$$H_{dB} = 10 * \log\left(\frac{1}{2\pi * f * R * C + 1}\right) \quad (46)$$

A frequência do PWM será de 31250 Hz (ver seção 6.4). Para essa frequência, a atenuação no filtro RC é calculada pela Equação ( 47 ):

$$H_{dB} = 10 * \log\left(\frac{1}{2\pi * 31250 * 3300 * 100 * 10^{-9} + 1}\right) = -18.18 \text{ dB} \quad (47)$$

Essa atenuação equivale a 65,80 vezes, o que significa que o sinal do PWM que possui 5V de amplitude será atenuado para cerca de 79 mV.

A potência sobre o resistor R24 não deve ser preocupação, pois a corrente drenada do sinal gerado (lido pela ECU) é em geral muito baixa.

Por último, o capacitor C8 é previsto para casos em que seja identificada a necessidade de filtrar ruídos de alta frequência (provindos de outra fonte que não o microcontrolador) que por algum motivo acoplem ao sinal emulado.

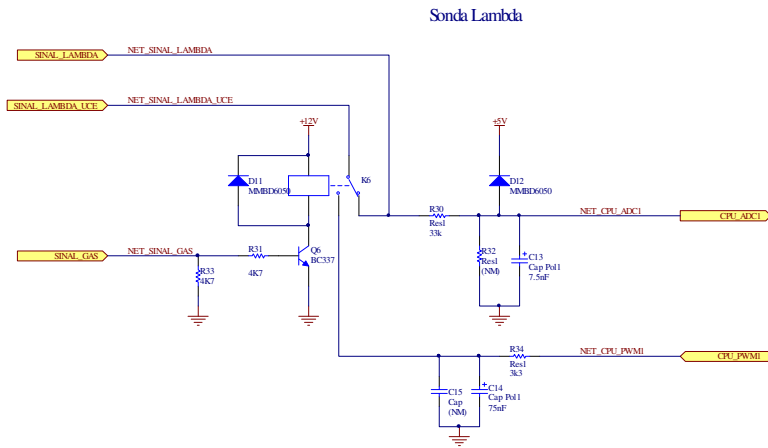
- *Circuito de Comutação*

- *O circuito de comutação* é responsável por selecionar qual sinal será enviado à UCE: o original ou o emulado. Essa comutação é feita a partir de um relé (K5), que é acionado a partir de um TBJ (Transistor Bipolar de Junção) – o Q5 – padrão no mercado eletrônico, o BC337. O resistor R21 limita a corrente sobre a base do transistor e o resistor R23 estabelece que quando há uma alta impedância no sinal SINAL\_GAS o TBJ está em modo de corte (não conduzindo).

No circuito de comutação projetado, o padrão é que o sinal que vai para a UCE seja o sinal original do sensor, isso porque é esse sinal que está no terminal NF (Normalmente Fechado) do relé. Quando se deseja emular o sinal do Sensor MAP, portanto, é necessário que o sinal seja produzido pelo microcontrolador a partir de um PWM para ser filtrado no circuito RC e por último que o relé seja comutado, colocando-se 5 V (nível lógico 1) em SINAL\_GAS e portanto fazendo com que o transistor Q5 entre em modo de saturação (conduzindo) e faça fluir corrente pela bobina do relé, fazendo-o comutar.

#### 4.2.6 Leitura e Emulação do sensor de Sonda Lambda

Os sensores de Sonda Lambda possuem geralmente uma saída de 0 a 1 V[29]. Além disso, testes mostram que as componentes de frequência principais desse sinal estão abaixo de 639 Hz (ver seção 3.3.4.6). Esses parâmetros foram os que alimentaram o projeto da Figura 64.



**Figura 64: Circuito de leitura, emulação e comutação do sensor de Sonda Lambda**

Os sinais mostrados nesse circuito são:

- *SINAL\_LAMBDA*: sinal provindo do sensor de Sonda Lambda, direto do conector para o mundo externo;
- *SINAL\_LAMBDA\_UCE*: sinal que irá se conectar a entrada de sensor de Sonda Lambda da UCE, direto para o conector para o mundo externo;
- *SINAL\_GAS*: sinal que indica que o sistema de gás está em funcionamento e que, portanto, determina a comutação ou não do relé e define qual será o sinal que se conectará ao mundo externo: o sinal original do sensor de Sonda Lambda ou o sinal emulado do sensor de Sonda Lambda. Ver seção 4.1.1;
- *CPU\_ADC1*: sinal que se conecta à entrada analógica do microcontrolador;
- *CPU\_PWM1*: sinal que se conecta à saída PWM do microcontrolador, que irá criar o sinal emulado do Sensor MAP.

Como o circuito de emulação não será utilizado, o fio que liga o sensor de sonda lambda à UCE deve ser derivado para se conectar a *SINAL\_LAMBDA* (para permitir a leitura desse sensor) e *SINAL\_LAMBDA\_UCE* deve ficar desconectado. Caso a emulação do sensor de sonda lambda seja implementada, o fio que conecta o sensor de sonda lambda à UCE deve ser interrompido (cortado) de forma que a extremidade que se liga ao sensor seja conectada a *SINAL\_LAMBDA* e a extremidade que se liga à UCE seja conectada a *SINAL\_LAMBDA\_UCE*.

Esse circuito pode ser dividido em três partes, são elas:

- *Circuito de leitura*:
  - O circuito de leitura é bastante simples, formado por um filtro RC (resistor R30 + capacitor C13) e um diodo de “clamp” (D12), responsável por garantir que o sinal não ultrapasse o limite do microcontrolador.

A frequência de corte de um filtro RC é calculada pela Equação ( 48 ).

$$F_{corte} = \frac{1}{2\pi RC} \quad (48)$$

Considerando uma frequência de corte de 639 Hz e os valores comerciais de resistor de capacitor, definiu-se:  $R = 33000 \Omega$  e  $C = 7,5 \text{ nF}$ . Com esses parâmetros, a frequência de corte é calculada pela Equação ( 49 ).

$$F_{corte} = \frac{1}{2\pi * 33000 * 7,5 * 10^{-9}} = 643,05 \text{ Hz} \quad (49)$$

Outra preocupação diz respeito à potência do resistor, que por ser um componente SMD suporta apenas 0,1W, e deve-se garantir que essa potência não seja excedida. A potência de um resistor pode ser calculada pela Equação ( 50 ).

$$P = \frac{V^2}{R} \quad (50)$$

Isolando V na Equação ( 50 ), obtemos a Equação ( 51 ).

$$V = \sqrt{P * R} \quad (51)$$

Considerando os parâmetros: P = 0,1 W e R = 33000 Ω, obtemos a tensão da Equação ( 52 ).

$$V = \sqrt{0,1 * 33000} = 57,45 V \quad (52)$$

Isso significa que a tensão máxima entre os terminais do resistor deve ser de 57,45 V, e como a tensão em um dos lados está limitada em 5,7 V (5 V da tensão de *clamp* + 0,7 V da queda do diodo MMBD6050) a tensão máxima de entrada pode ser 57,45 V + 5,7 V = 63,15 V. Essa tensão é muito acima do que se espera de uma tensão de um sensor de sonda lambda, o que faz o projeto robusto o suficiente desse ponto de vista.

O último parâmetro que deve ser levado em consideração é a corrente máxima suportada pelo diodo MMBD6050: 200 mA. Considerando que toda a corrente que passa por esse diodo passa também pelo resistor (já que a tensão drenada pelo microcontrolador é desprezível), a relação entre a tensão de entrada e a corrente passando pelo resistor (e conseqüentemente pelo diodo) pode ser expressa pela Equação ( 53 ).

$$I = \frac{V - V_{clamp}}{R} \quad (53)$$

Isolando V na Equação ( 53 ), obtemos a Equação ( 54 ).

$$V = I * R + V_{clamp} \quad (54)$$

Considerando os parâmetros:  $V_{clamp} = 5,7 V$ ,  $R = 33000 \Omega$  e  $I = 200 mA$ , obtemos a tensão da Equação ( 55 ).

$$V = 200 * 10^{-3} * 33000 + 5,7 = 6605,7 V \quad (55)$$

Ou seja, a tensão de entrada necessária para exceder a corrente máxima suportada pelo diodo MMBD6050 é de 6605,7 V! Desnecessário dizer que esse é um valor irreal para um sinal de sensor de Sonda Lambda.

Por fim, o resistor não montado R32 é previsto para casos em que o sensor de Sonda Lambda tenha sinais acima de 5V e que, portanto, devam ser atenuados antes de serem amostrados pelo microcontrolador.

- *Circuito de emulação:*

- O circuito de emulação é ainda mais simples, composto apenas por um filtro RC (resistor R34 + capacitor C14). A frequência desse filtro considerando os parâmetros ( $R = 3300 \Omega$  e  $C = 75 nF$ ) é mostrada na Equação ( 56 ) e é idêntica ao filtro de entrada, como era de se esperar.

$$F_{corte} = \frac{1}{2\pi * 3300 * 75 * 10^{-9}} = 643,05 Hz \quad (56)$$

A frequência é a mesma porque o sinal emulado deve possuir características semelhantes ao sinal original.

Outro importante parâmetro nesse filtro é a frequência gerada pelo PWM, que deve ser bastante atenuada para que esteja presente o mínimo possível no sinal de saída. A atenuação do filtro RC em decibéis em função da frequência do sinal de entrada é expressa pela Equação ( 57 ).

$$H_{dB} = 10 * \log\left(\frac{1}{2\pi * f * R * C + 1}\right) \quad (57)$$

A frequência do PWM será de 31250 Hz (ver seção 6.4). Para essa frequência, a atenuação no filtro RC é calculada pela Equação ( 58 ).

$$H_{dB} = 10 * \log\left(\frac{1}{2\pi * 31250 * 3300 * 75 * 10^{-9} + 1}\right) = -16.95 \text{ dB} \quad (58)$$

Essa atenuação equivale a 49,60 vezes, o que significa que o sinal do PWM que possui 5V de amplitude será atenuado para cerca de 101 mV.

A potência sobre o resistor R34 não deve ser preocupação, pois a corrente drenada do sinal gerado (lido pela ECU) é em geral muito baixa.

Por último, o capacitor C15 é previsto para casos em que seja identificada a necessidade de filtrar ruídos de alta frequência (provindos de outra fonte que não o microcontrolador) que por algum motivo acoplem ao sinal emulado.

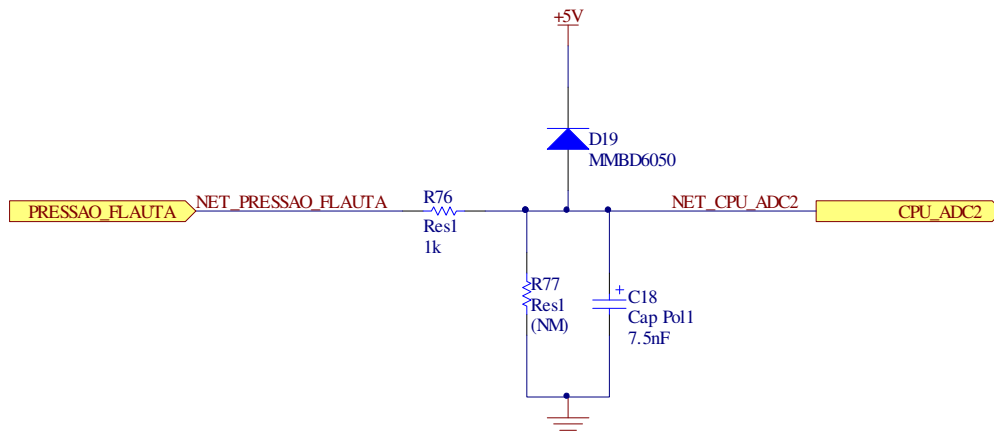
- *Circuito de Comutação*

- O circuito de comutação é responsável por selecionar qual sinal será enviado à UCE: o original ou o emulado. Essa comutação é feita a partir de um relé (K6), que é acionado a partir de um TBJ (Transistor Bipolar de Junção) – o Q6 – padrão no mercado eletrônico, o BC337. O resistor R31 limita a corrente sobre a base do transistor e o resistor R33 estabelece que quando há uma alta impedância no sinal SINAL\_GAS o TBJ está em modo de corte (não conduzindo).

No circuito de comutação projetado, o padrão é que o sinal que vai para a UCE seja o sinal original do sensor, isso porque é esse sinal que está no terminal NF (Normalmente Fechado) do relé. Quando se deseja emular o sinal do sensor de Sonda Lambda, portanto, é necessário que o sinal seja produzido pelo microcontrolador a partir de um PWM para ser filtrado no circuito RC e por último que o relé seja comutado, colocando-se 5 V (nível lógico 1) em SINAL\_GAS e portanto fazendo com que o transistor Q3 entre em modo de saturação (conduzindo) e faça fluir corrente pela bobina do relé, fazendo-o comutar.

#### 4.2.7 Leitura do sensor de pressão de flauta

O sensor de pressão de flauta estudado apresentou uma saída de 0 a ~3 V (ver seção 3.3.3.1). Além disso, testes mostram que as componentes de frequência principais desse sinal estão abaixo de 1617 Hz (ver seção 3.3.4.3). Esses parâmetros foram os que alimentaram o projeto da Figura 65.



**Figura 65: Circuito de leitura do sensor de Pressão da Flauta**

Os sinais mostrados nesse circuito são:

- *PRESSAO\_FLAUTA*: sinal provindo do sensor de Pressão da Flauta;
- *CPU\_ADC2*: sinal que se conecta à entrada analógica do microcontrolador;

O sinal do sensor de pressão da flauta deve se conectar ao sinal *PRESSAO\_FLAUTA* (para permitir a leitura desse sensor).

Diferentemente dos circuitos de leitura do sensor MAP (seção 4.2.5) e do sensor de sonda lambda (seção 4.2.6), esse circuito não prevê emulação do sinal, possuindo somente o circuito de leitura:

- *Circuito de leitura:*

- O circuito de leitura é bastante simples, formado por um filtro RC (resistor R76 + capacitor C18) e um diodo de “*clamp*” (D19), responsável por garantir que o sinal não ultrapasse o limite do microcontrolador.

A frequência de corte de um filtro RC é calculada pela Equação ( 59 ).

$$F_{corte} = \frac{1}{2\pi RC} \quad (59)$$

Considerando uma frequência de corte de 1617 Hz e os valores comerciais de resistor de capacitor, definiu-se:  $R = 1000 \Omega$  e  $C = 7,5 \text{ nF}$ . Com esses parâmetros, a frequência de corte é calculada pela Equação ( 60 ).

$$F_{corte} = \frac{1}{2\pi * 1000 * 7,5 * 10^{-9}} = 21220,66 \text{ Hz} \quad (60)$$

Outra preocupação diz respeito à potência do resistor, que por ser um componente SMD suporta apenas 0,1W, e deve-se garantir que essa potência não seja excedida. A potência de um resistor pode ser calculada pela Equação ( 61 ).

$$P = \frac{V^2}{R} \quad (61)$$

Isolando V na Equação ( 61 ), obtemos a Equação ( 62 ).

$$V = \sqrt{P * R} \quad (62)$$

Considerando os parâmetros:  $P = 0,1 \text{ W}$  e  $R = 1000 \Omega$ , obtemos a Equação ( 63 )

$$V = \sqrt{0,1 * 1000} = 10 \text{ V} \quad (63)$$

Isso significa que a tensão máxima entre os terminais do resistor deve ser de 10 V, e como a tensão em um dos lados está limitada em 5,7 V (5 V da tensão de *clamp* + 0,7 V da queda do diodo MMBD6050) a tensão máxima de entrada pode ser  $10 \text{ V} + 5,7 \text{ V} = 15,7 \text{ V}$ . Essa tensão é muito acima do que se espera de uma tensão de um sensor de Pressão de Flauta, o que faz o projeto robusto o suficiente desse ponto de vista.

O último parâmetro que deve ser levado em consideração é a corrente máxima suportada pelo diodo MMBD6050: 200 mA. Considerando que toda a corrente que passa por esse diodo passa também pelo resistor (já que a tensão drenada pelo microcontrolador é desprezível), a relação entre a tensão de entrada e a corrente passando pelo resistor (e conseqüentemente pelo diodo) pode ser expressa pela Equação ( 64 ).

$$I = \frac{V - V_{clamp}}{R} \quad (64)$$

Isolando V na Equação ( 64 ), obtemos a Equação ( 65 ).

$$V = I * R + V_{clamp} \quad (65)$$

Considerando os parâmetros:  $V_{clamp} = 5,7 \text{ V}$ ,  $R = 1000 \Omega$  e  $I = 200 \text{ mA}$ , obtemos a Equação ( 66 ).

$$V = 200 * 10^{-3} * 1000 + 5,7 = 205,7 \text{ V} \quad (66)$$

Ou seja, a tensão de entrada necessária para exceder a corrente máxima suportada pelo diodo MMBD6050 é de 205,7 V! Desnecessário dizer que esse é um valor irreal para um sinal de sensor de Sonda Lambda.

Por fim, o resistor não montado R77 é previsto para casos em que o sensor de Pressão de Flauta tenha sinais acima de 5V e que, portanto, devam ser atenuados antes de serem amostrados pelo microcontrolador.

#### 4.2.8 Leitura e Comutação dos Bicos Injetores originais

O sinal de injeção eletrônica deve ser utilizado pelo microcontrolador para medir-se o tempo de injeção implementado pela UCE original do veículo. O circuito projetado deve ser capaz das seguintes funcionalidades:

- “Congelamento” do bico injetor, de forma que não ocorra mais injeção do combustível original (gasolina/álcool);
- Leitura do sinal de injeção com a finalidade de se calcular o tempo de injeção. Essa funcionalidade deve estar ativa mesmo quando o bico injetor estiver “congelado”.

O projeto criado encontra-se na Figura 66.

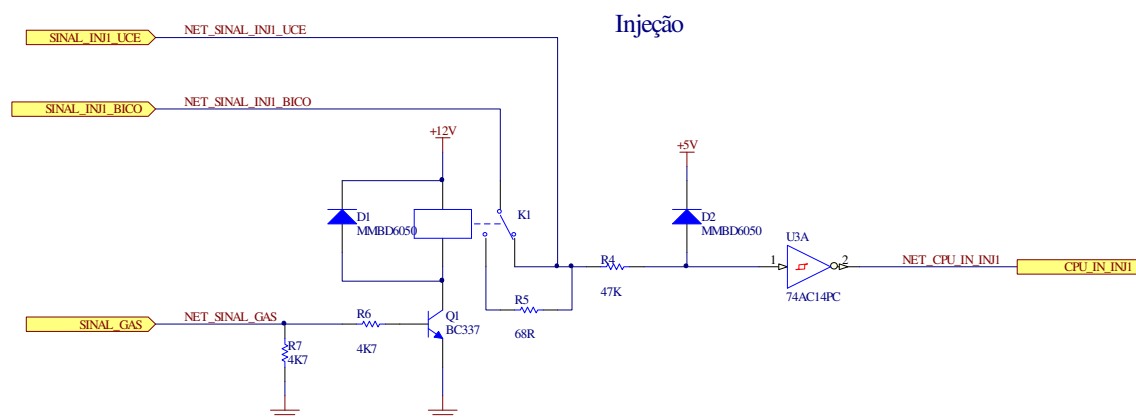


Figura 66: Circuito de leitura de sinal de injeção e congelamento dos bicos injetores.

Os sinais mostrados nesse circuito são:

- *SINAL\_INJI\_UCE*: sinal de acionamento dos bicos injetores providos da UCE;
- *SINAL\_INJI\_BICO*: terminal do bico injetor que se conecta à UCE;
- *SINAL\_GAS*: sinal que indica que o sistema de gás está em funcionamento e que, portanto, determina a comutação do relé, congelando o bico injetor. Ver seção 4.1.1;
- *CPU\_IN\_INJI*: sinal digital tratado, que carrega a informação de injeção provinda do sinal *SINAL\_INJI\_UCE*;

Repare que o fio que conecta o bico injetor à UCE deve ser interrompido (cortado) de forma que uma das extremidades será conectada a *SINAL\_INJI\_BICO* e a outra a *SINAL\_INJI\_UCE*.

Esse circuito pode ser dividido em três partes, são elas:

- *Circuito de leitura*:
  - Esse circuito é formado por três componentes: um resistor (R4), um diodo (D2) e um inversor lógico (U3A). O resistor e o diodo são responsáveis por limitar a tensão na entrada do inversor lógico em 5,7 V (5 V + 0,7 V da queda do diodo).

O inversor lógico (74AC14PC) suporta uma tensão de até 6.5 V em suas entradas (6 V + 1,5 V), e portanto não deve ter problemas com o circuito de limitação de tensão descrito acima. Adicionalmente, esse inversor lógico possui uma característica de *Schmitt Trigger*, o que faz com que tenha uma histerese entre os limites positivo e negativo de transição de entrada. Essa característica evita que ruídos de pequena amplitude próximos aos limites lógicos de entrada sejam refletidos na saída.

Outra preocupação diz respeito à potência do resistor, que por ser um componente SMD suporta apenas 0,1W, e deve-se garantir que essa potência não seja excedida. A potência de um resistor pode ser calculada com a Equação ( 67 ).



$$P = \frac{V^2}{R} \quad (67)$$

Isolando V na Equação ( 67 ), obtemos a Equação ( 68 ).

$$V = \sqrt{P * R} \quad (68)$$

Considerando os parâmetros: P = 0,1 W e R = 47000 Ω, obtemos a Equação ( 69 ).

$$V = \sqrt{0,1 * 47000} = 68,56 V \quad (69)$$

Isso significa que a tensão máxima entre os terminais do resistor deve ser de 68,56 V, e como a tensão em um dos lados está limitada em 5,7 V (5 V da tensão de *clamp* + 0,7 V da queda do diodo MMBD6050) a tensão máxima de entrada pode ser 68,56 V + 5,7 V = 74,26 V. Embora essa tensão seja bem próxima de valores observados em campo, isso não deve ser preocupação pois valores altos de tensão aparecem em intervalos bem curtos de tempo.

O último parâmetro que deve ser levado em consideração é a corrente máxima suportada pelo diodo MMBD6050: 200 mA. Considerando que toda a corrente que passa por esse diodo passa também pelo resistor (já que a tensão drenada pelo microcontrolador é desprezível), a relação entre a tensão de entrada e a corrente passando pelo resistor (e conseqüentemente pelo diodo) pode ser expressa pela Equação ( 70 ).

$$I = \frac{V - V_{clamp}}{R} \quad (70)$$

Isolando V na Equação ( 70 ), obtemos a Equação ( 71 ).

$$V = I * R + V_{clamp} \quad (71)$$

Considerando os parâmetros: V<sub>clamp</sub> = 5,7 V, R = 47000 Ω e I = 200 mA, obtemos a Equação ( 72 ).

$$V = 200 * 10^{-3} * 47000 + 5,7 = 9.405,7 V \quad (72)$$

Ou seja, a tensão de entrada necessária para exceder a corrente máxima suportada pelo diodo MMBD6050 é de 9.405,7 V! Desnecessário dizer que esse é um valor irreal para um sinal de sensor de injeção

- *Circuito de congelamento dos bicos:*

- Os bicos injetores de baixa impedância possuem uma resistência em torno de 2,5 Ω e os bicos injetores de alta impedância possuem uma resistência em torno de 17 Ω. Essas resistências definem a corrente que irá fluir por esse bico, acionando a eletro válvula e permitindo a passagem de combustível. O circuito de congelamento dos bicos consiste em apenas um único resistor de 68 Ω colocado em série com o bico injetor, o que limita a corrente que flui por ele e impede o seu acionamento.
- A potência suportada pelo resistor é calculada a partir de sua resistência e da tensão elétrica entre seus dois terminais, conforme a Equação ( 73 ).

$$P_r = \frac{V_{bicos}^2}{R} \quad (73)$$

A tensão V<sub>bicos</sub> é a tensão de alimentação do bico injetor, e deve atingir um valor máximo de 14 V para um veículo movido a 12 V, somando a isso a informação da resistência de 68 Ω do resistor, obtemos a Equação ( 74 ).

$$P_r = \frac{14^2}{68} = 2,88W \quad (74)$$

Portanto, definiu-se uma potência de 3W para o resistor.

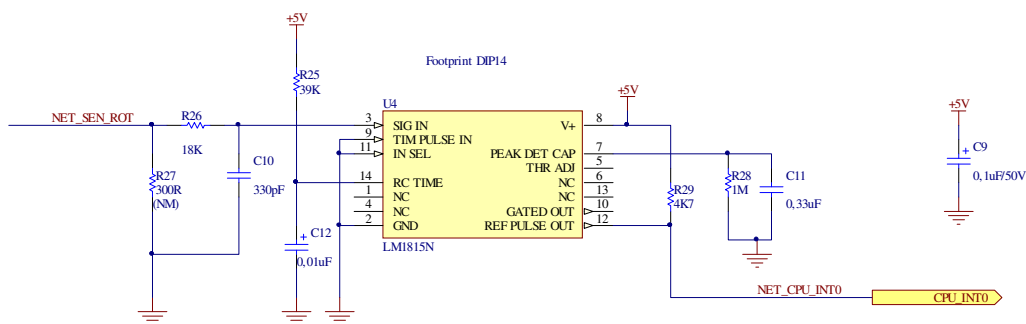
- **Circuito de Comutação**

- O *circuito de comutação* é responsável por determinar o “congelamento” ou não do bico injetor. Essa comutação é feita a partir de um relé (K1), que é acionado a partir de um TBJ (Transistor Bipolar de Junção) – o Q1 – padrão no mercado eletrônico, o BC337. O resistor R6 limita a corrente sobre a base do transistor e o resistor R7 estabelece que quando há uma alta impedância no sinal SINAL\_GAS o TBJ está em modo de corte (não conduzindo).

No circuito de comutação projetado, o padrão é que o bico injetor não esteja “congelado”, isso porque o sinal que se conecta ao NF (Normalmente Fechado) do relé é o sinal de acionamento da UCE sem a adição de um resistor em série. Quando se deseja congelar o bico injetor, portanto, é necessário que o relé seja comutado, colocando-se 5 V (nível lógico 1) em SINAL\_GAS e portanto fazendo com que o transistor Q1 entre em modo de saturação (conduzindo) e faça fluir corrente pela bobina do relé, fazendo-o comutar, inserindo uma resistência de 68  $\Omega$  em série com o bico injetor.

## 4.2.9 Sensor de Rotação

Para a função de leitura do sensor de rotação foi utilizado um CI (Circuito Integrado) projetado com essa finalidade, trata-se do LM1815N da *National Semiconductor*. Basicamente, esse CI produz um pulso com largura determinada toda vez que o sinal de entrada cruza o 0V em direção à tensão negativa. O seu circuito é ilustrado na Figura 67.



**Figura 67: Circuito de leitura do Sensor de Rotação**

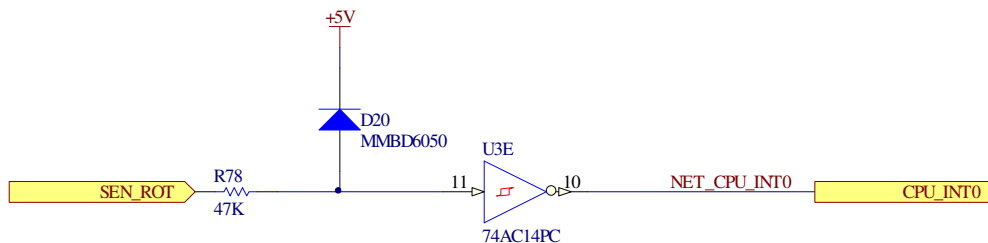
O circuito foi projetado com base no *datasheet* do componente e possui as seguintes características:

- Entrada de sinal de até  $\pm 54V$ .
- Histerese de 80% do valor de pico do sinal. Por exemplo, para um sinal de entrada de  $\pm 10V$ , um pulso só será detectado se após o pulso anterior o sinal ultrapassar o valor de 8V antes de cruzar o 0V em direção à tensão negativa novamente.
- Um pulso de aproximadamente 260 us é gerado toda vez que o sinal de entrada cruzar o 0V em direção à tensão negativa.
- Frequência do sinal máxima de aproximadamente 1,9 kHz.

O circuito do Sensor de Rotação é adequado para leitura de sensores magnéticos, mas não se mostrou eficaz na leitura de rotação a partir do sinal de ignição (que é o sinal utilizado no caso do *kit* instalado – ver seção 3.3.2). Por isso, utilizou-se um circuito de rotação alternativo (seção 4.2.10).

## 4.2.10 Sensor de Rotação Alternativo

Esse circuito foi projetado após a constatação de que o circuito descrito na seção 4.2.9 não era eficaz na leitura do sinal de ignição (utilizado para medida de rotação), e baseia-se no circuito de leitura dos bicos injetores (seção 4.2.8). O circuito é apresentado na Figura 68.



**Figura 68: Circuito alternativo do Sensor de Rotação**

O circuito Funciona da seguinte maneira:

Esse circuito é formado por três componentes: um resistor (R78), um diodo (D20) e um inversor lógico (U3E). O resistor e o diodo são responsáveis por limitar a tensão na entrada do inversor lógico em 5,7 V (5 V + 0,7 V da queda do diodo).

O inversor lógico (74AC14PC) suporta uma tensão de até 6.5 V em suas entradas (6 V + 1,5 V), e portanto não deve ter problemas com o circuito de limitação de tensão descrito acima. Adicionalmente, esse inversor lógico possui uma característica de *Schmitt Trigger*, o que faz com que tenha uma histerese entre os limites positivo e negativo de transição de entrada. Essa característica evita que ruídos de pequena amplitude próximos aos limites lógicos de entrada sejam refletidos na saída.

Outra preocupação diz respeito à potência do resistor, que por ser um componente SMD suporta apenas 0,1W, e deve-se garantir que essa potência não seja excedida. A potência de um resistor pode ser calculada pela Equação ( 75 ).

$$P = \frac{V^2}{R} \quad (75)$$

Isolando V na Equação ( 75 ), obtemos a Equação ( 76 ).

$$V = \sqrt{P * R} \quad (76)$$

Considerando os parâmetros: P = 0,1 W e R = 47000 Ω, obtemos a Equação ( 77 ).

$$V = \sqrt{0,1 * 47000} = 68,56 V \quad (77)$$

Isso significa que a tensão máxima entre os terminais do resistor deve ser de 68,56 V, e como a tensão em um dos lados está limitada em 5,7 V (5 V da tensão de *clamp* + 0,7 V da queda do diodo MMBD6050) a tensão máxima de entrada pode ser 68,56 V + 5,7 V = 74,26 V. Embora essa tensão seja bem próxima de valores observados em campo, isso não deve ser preocupação pois valores altos de tensão aparecem em intervalos bem curtos de tempo.

O último parâmetro que deve ser levado em consideração é a corrente máxima suportada pelo diodo MMBD6050: 200 mA. Considerando que toda a corrente que passa por esse diodo passa também pelo resistor (já que a tensão drenada pelo microcontrolador é desprezível), a relação entre a tensão de entrada e a corrente passando pelo resistor (e conseqüentemente pelo diodo) pode ser expressa pela Equação ( 78 ).

$$I = \frac{V - V_{clamp}}{R} \quad (78)$$

Isolando V na Equação ( 78 ), obtemos a Equação ( 79 ).

$$V = I * R + V_{clamp} \quad (79)$$

Considerando os parâmetros: V<sub>clamp</sub> = 5,7 V, R = 47000 Ω e I = 200 mA, obtemos a Equação ( 80 ).

$$V = 200 * 10^{-3} * 47000 + 5,7 = 9.405,7 V \quad (80)$$

Ou seja, a tensão de entrada necessária para exceder a corrente máxima suportada pelo diodo MMBD6050 é de 9.405,7 V (desnecessário dizer que esse é um valor irreal para um sinal de sensor de rotação).

#### 4.2.11 Sinal de acionamento dos Bicos

O circuito que gera o sinal de acionamento dos bicos é responsável por transformar os sinais digitais de saída do microcontrolador (5V) em sinais aptos a controlar o circuito de acionamento dos bicos injetores de gás (seção 4.3.2).

Os MOSFETs utilizados no acionamento dos bicos injetores de gás são do modelo IRF640NPBF da *International Rectifier*. Segundo o *datasheet* desse componente, o  $V_{GS(th)}$  (tensão entre porta e fonte na qual o MOSFET passa a estar acionado, comumente referenciada como  $V_T$  na literatura acadêmica) máximo do componente é de 4V. Considerando que a tensão de saída mínima do pino do microcontrolador ATmega32-16PU é de 4.2V e que a fonte do MOSFET se conecta ao GND, é esperado que se pergunte: porque a necessidade de um circuito de transformação dos sinais? Não seria possível acionar os MOSFETs diretamente a partir do microcontrolador?

O elemento chave para responder a essas perguntas reside no fato de que o MOSFET necessita não só que sua tensão  $V_{GS}$  (tensão entre porta e fonte) esteja acima do  $V_{GS(th)}$  para estar acionado, mas também que sua tensão  $V_{GD}$  (tensão entre porta e dreno) esteja acima de  $V_{GS(th)}$ . Esse fato é geralmente esquecido porque quando acionado a tensão no dreno e na fonte são muito semelhantes, o que nesse caso não é exatamente verdadeiro pela alta corrente que flui por esses componentes. Para calcular a tensão no dreno durante o estado acionado, deve-se multiplicar a corrente que flui por esse componente por sua resistência de acionamento ( $R_{DS(on)}$ ). Considerando os parâmetros das Equações ( 81 ) e ( 82 ) a tensão do dreno ( $V_{D máx}$ ) é calculada pela Equação ( 83 ).

$$\text{Corrente de Acionamento dos Bicos} = I_B = \frac{12 V}{2,15 \Omega} = 5,58A \quad (81)$$

$$R_{DS(on) máx} = 150m\Omega \quad (82)$$

$$V_{D máx} = 5,58 * 0,15 = 0,84V \quad (83)$$

Com isso, uma tensão de acionamento de 4,2 V do microcontrolador pode não ser suficiente para manter a tensão  $V_{GD}$  (que atinge um valor mínimo de  $4,2 - 0,84 = 3,36 V$ ) maior que o  $V_{GS(th)}$  de 4V.

Outro fator importante que justifica a necessidade do circuito é o fato de que quanto maior é a tensão  $V_{GS}$ , menor o valor de  $R_{DS(on)}$ , o que implica em uma maior corrente de acionamento (menor resistência em série) e conseqüentemente menor tempo de abertura dos bicos injetores.

Por último, para fazer a estratégia de *Peak and Hold* seriam necessários 4 canais de PWM, no entanto o microcontrolador ATmega32-16PU só dispõe de 4 canais mas 2 deles estão sendo utilizados para emular os sinais de MAP e Sonda Lambda. Para contornar esse fato, foi criado um esquema elétrico utilizando circuitos lógicos para ser possível utilizar apenas uma saída de PWM para gerar o sinal de todos os bicos injetores.

O circuito é apresentado na Figura 69.

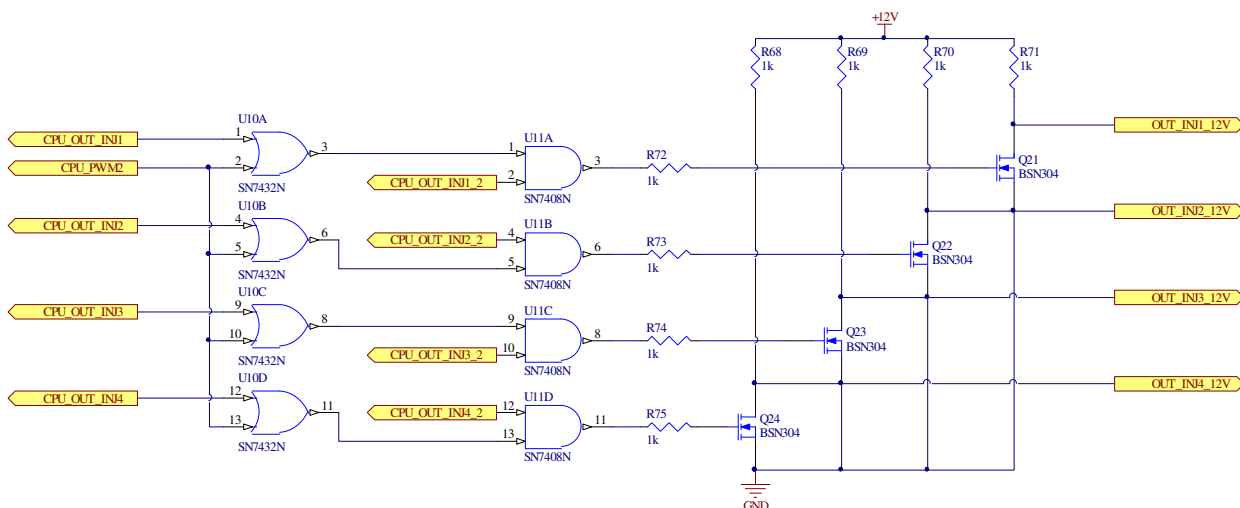


Figura 69: Circuito de criação do sinal de controle do circuito de acionamento dos bicos injetores de gás.

O circuito funciona da seguinte forma:

- O sinal conectado à porta de cada MOSFET BSN304 pode ser expressado da seguinte forma (na notação de álgebra booleana):
  - $(CPU\_PWM2 + CPU\_OUT\_INJx) * CPU\_OUT\_INJx\_2$ ;

O que resulta na seguinte tabela verdade:

CPU_PWM2	CPU_OUT_INJx	CPU_OUT_INJx_2	$(CPU\_PWM2 + CPU\_OUT\_INJx) * CPU\_OUT\_INJx\_2$
PWM 7,81 kHz @ 66%	0	0	0
PWM 7,81 kHz @ 66%	0	1	PWM 7,81 kHz @ 66%
PWM 7,81 kHz @ 66%	1	0	0
PWM 7,81 kHz @ 66%	1	1	1

O MOSFET BSN304 é utilizado para elevar a tensão de controle do circuito de acionamento dos bicos para 12V, mas sua utilização também inverte logicamente o sinal:

- Quando apresenta-se um valor lógico 1 na porta do MOSFET BSN304 (que possui um  $V_{GS(th)}$  de apenas 2V) ele passa a conduzir, apresentando um valor de 0V em  $OUT\_INJx\_12V$ , e portanto **não** comandando o acionamento do bico injetor de gás – estado *Idle*.
- Quando apresenta-se um valor lógico 0 na porta do MOSFET BSN304, ele não conduz, apresentando um valor de “+12V” (tensão da bateria) em  $OUT\_INJx\_12V$ , e portanto comandando o **acionamento constante** do bico injetor de gás – estado *Peak*.
- Quando apresenta-se um PWM na porta do MOSFET BSN304, ele inverte o sinal e o eleva para 12V. Desta forma, um PWM de 7,81 kHz a 66% com tensão máxima de 5V é transformado em um PWM de 7,81 kHz a 33% com tensão máxima de “+12V” (tensão da bateria), e portanto comandando o **acionamento chaveado** do bico injetor de gás – estado *Hold*.

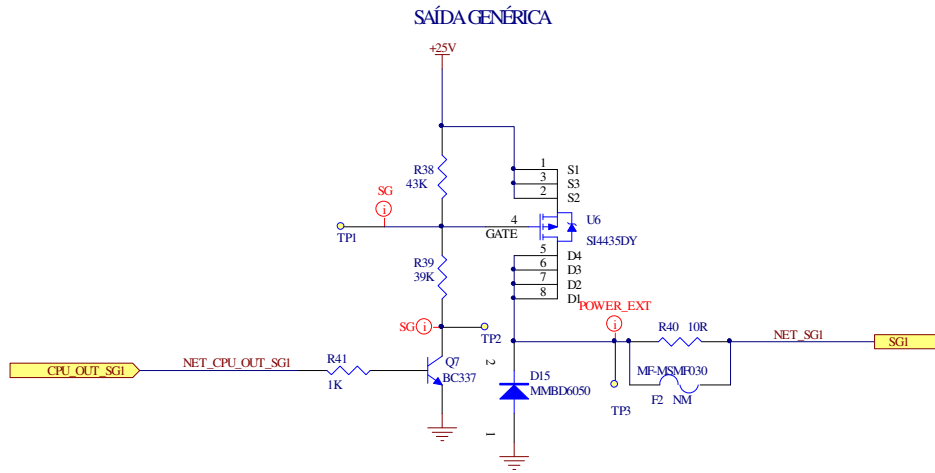
A tabela abaixo resume o acionamento dos bicos injetores de gás a partir dos sinais  $CPU\_OUT\_INJx$  e  $CPU\_OUT\_INJx\_2$ :

CPU_OUT_INJx	CPU_OUT_INJx_2	Acionamento do Bico
0	0	Acionamento em <i>Peak</i>
0	1	Acionamento em <i>Hold</i>
1	0	Acionamento em <i>Peak</i>
1	1	Não acionamento – <i>Idle</i>

#### 4.2.12 Saídas Genéricas

O circuito de saída genérica foi projetado para conferir flexibilidade ao produto, possibilitando-o acionar cargas externas.

O circuito é dotado de 4 saídas genéricas, que possuem uma alimentação específica que deve funcionar de 9V a 32V. Cada saída genérica é capaz de chavear essa alimentação através de um sinal TTL vindo do microcontrolador. O circuito é apresentado na Figura 70.



**Figura 70: Saída genérica**

O circuito funciona da seguinte maneira:

- Quando o sinal *CPU\_OUT\_SG1* estiver em nível lógico ‘0’, não haverá circulação de corrente pelos resistores R38 e R39 e pelo transistor Q7, e portanto a tensão na porta do MOSFET U6 será igual à tensão em sua fonte.
- Quando o sinal *CPU\_OUT\_SG1* estiver em nível lógico ‘1’, haverá uma circulação de corrente pelos resistores R38 e R39 e pelo transistor Q7, e portanto haverá uma queda de tensão no resistor R38, fazendo com que a tensão na porta do MOSFET U6 seja igual ao da Equação ( 84 )

$$V_{G_{u6}} = \frac{R39}{R38 + R39} * V_{in} \quad (84)$$

Onde  $V_{G_{u6}}$  é a tensão na porta do MOSFET U6 e  $V_{in}$  é a tensão que alimenta o circuito das saídas genéricas e é representada no esquema elétrico como “+25V”.

Dessa forma, podemos calcular a tensão entre a porta e a fonte do MOSFET U6 com a Equação ( 85 ).

$$V_{GS_{U6}} = \frac{R39}{R38 + 39} * V_{in} - V_{in} = V_{in} \left( \frac{R39}{R38 + R39} - 1 \right) \quad (85)$$

Onde  $V_{GS_{U6}}$  é a tensão entre a porta e a fonte do MOSFET U6.

Para o correto funcionamento do circuito, é necessário que em 9V (limite inferior da tensão de funcionamento do circuito) a tensão  $V_{GS_{U6}}$  seja menor que a tensão  $V_{GS(th)}$  mínima do MOSFET para que ocorra o acionamento e que em 32V (limite superior da tensão de funcionamento do circuito) a tensão  $V_{GS_{U6}}$  seja maior que a tensão  $V_{GS}$  mínima suportada pelo MOSFET, conforme as Equações ( 86 ) e ( 87 ).

$$V_{GS_{U6}} @ (V_{in} = 9V) < V_{GS(TH)_{min}} \quad (86)$$

$$V_{GS_{U6}} @ (V_{in} = 32V) > V_{GS_{min}} \quad (87)$$

Consultando o *datasheet* do componente, chegamos aos seguintes valores das Equações ( 88 ) e ( 89 ).

$$V_{GS(TH)_{min}} = -3V \quad (88)$$

$$V_{GS_{min}} = -20V \quad (89)$$

Definindo  $\alpha$  conforme a Equação ( 90 ).

$$\alpha := \frac{R39}{R38 + R39} \quad (90)$$

E aplicando as restrições de tensão das Equações ( 86 ), ( 87 ), ( 88 ) e ( 89 ), chegamos às Inequações ( 91 ) e ( 92 ).

$$9(\alpha - 1) < -3 \therefore \alpha < -\frac{3}{9} + 1 \therefore \alpha < 0,667 \quad (91)$$

$$32(\alpha - 1) > -20 \therefore \alpha > -\frac{20}{32} + 1 \therefore \alpha > 0,375 \quad (92)$$

Essas inequações podem ser reunidas na Inequação ( 93 ).

$$0,375 < \alpha < 0,667 \quad (93)$$

Dessa forma, definiu-se  $R38 = 43k\Omega$  e  $R39 = 39k\Omega$  de tal forma que  $\alpha$  assumiu o valor da Equação ( 94 ).

$$\alpha = \frac{R39}{R38 + R39} = \frac{43k}{39k + 43k} = 0,524 \quad (94)$$

Ou seja, um valor não só dentro da faixa limite calculada, mas bem próximo do valor central dessa faixa (0,521).

### 4.3 PCI – Circuitos de potência

Nessa seção, o projeto da PCI responsável pela parte de potência, ou seja, pelo acionamento dos bicos injetores e geração do *SINAL\_GAS* (seção 4.1.1), é explicada com detalhes.

Essa PCI é responsável por receber da PCI de circuitos digitais e analógicos (ver seção 4.2) o sinal digital que carrega a informação da largura de pulso de acionamento dos bicos e por realizar o acionamento dos bicos, criando um caminho de corrente por esses bicos (ver seção 2.3.2).

#### 4.3.1 Especificações do *hardware*.

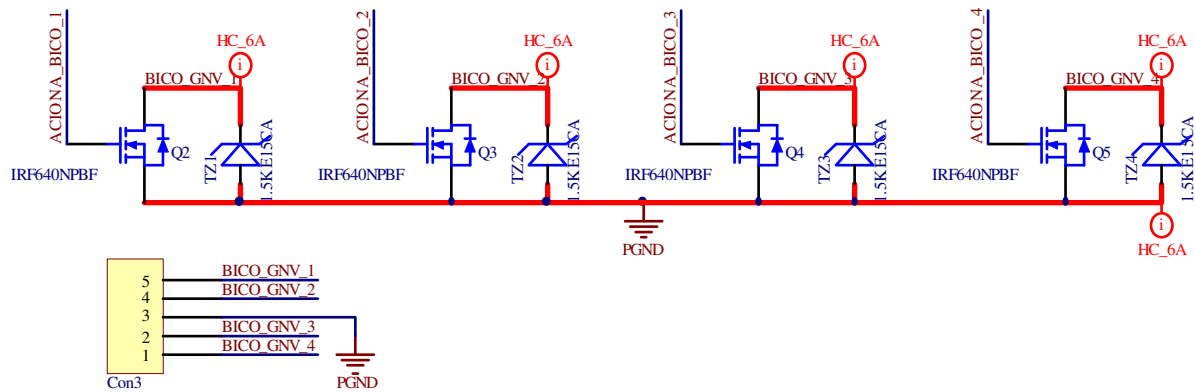
Essa seção traz as especificações técnicas da PCI – Circuitos de potência.

- Tensão dos sinais digitais de acionamento entre 9V e 16V – o sinal de acionamento do bico tem seu nível definido pela tensão da bateria (seção 4.2.11), que deve alcançar valores nessa faixa;
- Corrente de acionamento do bico de 6A de *peak* e 2A de *hold* – considerando que a impedância do bico injetor de gás foi medida como sendo 2  $\Omega$ , que a tensão entre seus terminais é de 12V quando acionado e que a corrente de *hold* é de aproximadamente 34% a corrente de *peak*;
- Impedância de acionamento de no máximo 200m $\Omega$  – dessa forma, a corrente sobre o bico injetor não é prejudicada por uma impedância em série com esse bico.
- Corrente do *SINAL\_GÁS* de até 20A (6A de até um bico em *peak*, 6A de até 2 bicos em *hold* e uma estimativa de até 8A para o resto do circuito);

#### 4.3.2 Circuito de Acionamento dos Bicos

O circuito de acionamento dos bicos foi projetado para possibilitar o acionamento dos bicos de GNV a partir de sinais na tensão da bateria.

O circuito é capaz de acionar 4 bicos a partir de 4 sinais na tensão da bateria. Esse acionamento é feito utilizando-se 1 FET-N e 1 tranzorb para cada acionamento de bico. O circuito é apresentado na Figura 71.



**Figura 71: Circuito de Acionamento dos Bicos**

O circuito funciona da seguinte maneira:

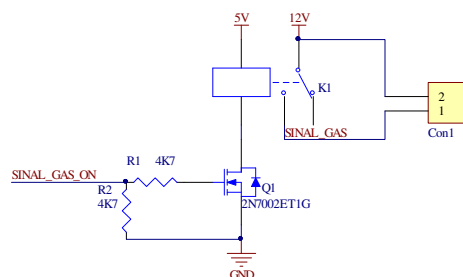
- Quando o sinal  $ACIONA\_BICO\_x$  correspondente assume o valor lógico “1” (ou seja,  $\sim 12V$ ), o FET-N aciona, criando um caminho de corrente entre o sinal  $BICO\_GNV\_x$  e o terra. O componente selecionado para essa tarefa é crítico pois é por ele que passa toda a corrente de acionamento dos bicos de GNV (ver seção 2.3.2), e após pesquisa chegou-se no IRF640NPBF da International Rectifier®. Esse componente possui uma tensão de acionamento ( $V_{gs(th)}$ ) de 4V (quando o especificado foi de pelo menos 9V) e apresenta uma impedância entre o dreno e a fonte de 150mΩ quando acionado por uma tensão de 10V (quando o especificado foi de menos de 200mΩ).
- Os tranzorbs servem para limitar a tensão entre o dreno e a fonte dos FETs-N, pois como está sendo feito o chaveamento de uma carga indutiva são esperados picos de tensão quando ocorrer o desacionamento dessas cargas (ver seção 3.3.3.3 – sinais dos bicos de original e de GNV) – a tensão deve ser limitada em 200V ( $V_{DSS}$  máxima do FET-N selecionado). O componente selecionado foi o 1.5KE15CA da Littelfuse®, que limita a tensão entre seus terminais em 21,2V. Considerando a corrente de acionamento dos bicos de 6A, a potência sobre esse componente durante sua atuação é calculada pela Equação ( 95 ).

$$P = V * I = 21,2 * 6 = 127,2W \quad (95)$$

Segundo o datasheet do componente, ele é capaz de suportar até 1.500W.

### 4.3.3 Circuito de geração do SINAL\_GAS

O SINAL\_GAS é responsável por alimentar o sistema GNV quando o veículo está utilizando esse combustível. Esse sinal é também utilizado para avisar alguns outros circuitos (como o Variador de Avanço) que o combustível GNV está sendo utilizado (ver seção 4.1.1). Esse circuito é ilustrado na Figura 72.



**Figura 72: Circuito de geração do SINAL\_GAS**

Esse circuito tem a mesma função dos circuitos descritos nas seções 4.2.5, 4.2.6 e 4.2.8, mas difere desses pela utilização de um FET-N no lugar de um transistor NPN. Essa escolha foi fruto de uma revisão na estratégia de acionamento, e resultou em um circuito mais simples, mais barato e com menor espaço ocupado na PCI. O circuito funciona da seguinte maneira:

- O componente selecionado para acionamento do relé foi o 2N7002ET1G da ON Semiconductors. Esse componente tem uma tensão de acionamento ( $V_{gs(th)}$ ) de 2,5V (suficientemente pequena considerando a tensão de saída do microcontrolador ATmega32-16PU de no mínimo 4,2V). O resistor R1 limita a corrente sobre a porta do FET-N durante a transição digital do sinal SINAL\_GAS\_ON e o resistor R2 mantém o relé desacionado quando existe uma alta impedância em SINAL\_GAS\_ON.



# 5 EST

Essa seção traz com detalhes a estratégia que foi introduzida simplificada na seção 2.5.

## 5.1 Descrição da estratégia

A UCE (Unidade de Comando Eletrônico) é responsável por determinar o tempo de injeção de combustível. Portanto, o tempo de injeção de combustível é determinado pela UCE.

Se aproveitando do fato descrito, a determinação do tempo de injeção de misturas ar-combustível quando é utilizada gasolina/álcool e de GNV possuem vazão e tempo de injeção diferentes.

Adicionalmente, existem diferenças que exigem correções diferentes para as diferentes misturas. A dissertação de mestrado é realizada essa correção (ver seção 2.5), evitando estudos químicos e dinâmicos.

A ilustração da estratégia de funcionamento é mostrada aqui apenas para facilitar o entendimento. O sistema desenvolvido congela os bicos de injeção de gasolina/álcool.



Figura 73: Estratégia de funcionamento

## 5.2 Procedimentos seguidos

Essa seção detalha os procedimentos seguidos.

Foram determinadas:

- Correção Fixas ( $k$ ) – seção 5.2.5.1
- Base de Regras – seção 5.2.5.2

A base de regras de um sistema de controle é baseada em seu conhecimento e experiência do sistema desenvolvido, seria muito difícil desenvolver um sistema de controle por utilizar-se um kit comercial de GNV. Foi desenvolvido um sistema de controle de GNV comercial e feito um estudo sobre o funcionamento do sistema.

Para todos os casos, o critério de avaliação para determinar o sucesso de cada tentativa de correção é descrito na seção 5.2.1. Os procedimentos utilizados na determinação das funções de pertinência de cada grandeza são descritos na 5.2.3. Para a determinação dos predecessores a serem utilizados na base de regra, foi utilizado o procedimento descrito na seção 5.2.4.

### 5.2.1 Parâmetro de Avaliação (*erro*)

Para verificar a eficácia de cada regra testada, foi criado um cálculo da diferença entre o resultado gerado pela regra e o valor lido do *kit* comercial. Ou seja, uma parâmetro que define o grau de semelhança entre o valor de saída produzido como fruto da(s) regra(s) nebulosa(s) e o valor de saída do *kit* comercial. Esse parâmetro foi definido pela Equação ( 96 ).

$$erro = \sum_{n=0}^N |TI_{GNV} - TI_{Gasolina} * \max(0, (k + FuzzyOut))| \quad (96)$$

$N$  é o tamanho do *log*

$TI_{GNV}$  é o tempo de injeção determinado pelo *kit* comercial e registrado no *log*

$TI_{Gasolina}$  é o tempo de injeção de gasolina determinado pela UCE original do carro e registrado no *log*

$k$  é a constante de correção fixa entre o tempo de injeção de gasolina e de GNV (ver seção 5.2.2)

$FuzzyOut$  é a saída do sistema nebuloso (ver seção 2.5.6)

*erro* pode ser descrito como a soma dos módulos das diferenças entre o valor de saída do *kit* comercial e o valor de saída gerado a partir da(s) regra(s) nebulosa(s). A função *max* retorna o maior valor entre seus operandos e é utilizada para evitar que alguma correção resulte em um multiplicador negativo do tempo e injeção de gasolina, já que na prática seria impossível implementar um tempo de injeção negativo.

O *log* mais utilizado nos testes foi o “aceleracao\_0\_100\_trocaMarcha\_3”, pois acredita-se que esse *log* represente a maior variedade de condições de funcionamento do veículo, e as condições mais importantes.

Como referência, o *erro* quando se aplica uma correção fixa de 100% (ou seja, quando o tempo de injeção de GNV é uma cópia exata do tempo de injeção de gasolina) é de 480.

### 5.2.2 Determinação da Correção Fixa ( $k$ )

Define-se como  $k$ , a correção fixa a ser aplicada sobre o tempo de injeção original do veículo (gasolina/álcool), a fim de gerar o tempo de injeção de GNV.

Para determinação de  $k$ , foram testados os seguintes valores de correção fixa:

- De 70% a 130%, variando-se de 0,1 em 0,1%

O valor de  $k$  que resultou em um menor *erro* foi de **99,7%**, com um *erro* de 476.

Pode-se verificar que o tempo de injeção de GNV promovido pelo *kit* comercial é em média bem parecido com o tempo de injeção de gasolina/álcool para o caso específico do veículo testado (Peugeot 206 1.4)

### 5.2.3 Determinação dos Conjuntos Nebulosos e suas Funções de Pertinência

Para determinação das funções de pertinência, primeiramente foram analisados os *logs* que acredita-se representem melhor as diferentes condições de funcionamento do motor:

- aceleracao\_0\_100\_trocaMarcha\_3;
- conmutacao\_gasolina\_Gas\_1;
- aceleracao\_desengatado\_1;
- desaceleracao80\_40\_3Marcha\_3;

Para detalhes sobre a obtenção dos dados e as condições de funcionamento do veículo durante sua obtenção, consulte a seção 3.3.

A partir da análise desses *logs*, foi elaborado o quadro resumo da Tabela 6.

**Tabela 6: Resumo da faixa de valores de cada sinal**

Arquivo	Pressão Flauta (V)		Sensor MAP (V)		Rotacao (Hz)		Temp (V)		Sonda Lambda (V)		Centralina Bico 1 (s)	
	Mín	Máx	Mín	Máx	Mín	Máx	Mín	Máx	Mín	Máx	Mín	Máx
aceleracao_0_100_trocaMarcha_3	1,48	2,46	0,50	3,80	0,00	73,00	1,51	1,57	0,00	0,85	0,00	0,02
conmutacao_gasolina_Gas_1	0,98	2,62	1,20	1,50	14,00	22,00	2,13	2,95	0,00	0,85	0,00	0,01
aceleracao_desengatado_1	1,64	2,62	0,50	3,80	14,00	90,00	2,13	2,20	0,00	0,92	0,00	0,03
desaceleracao80_40_3Marcha_3	1,51	1,97	0,65	1,05	30,00	70,00	1,64	1,77	0,00	0,05	0,00	0,00
Resumo	1,48	2,62	0,50	3,80	14,00	90,00	1,51	2,95	0,00	0,92	0,00	0,03

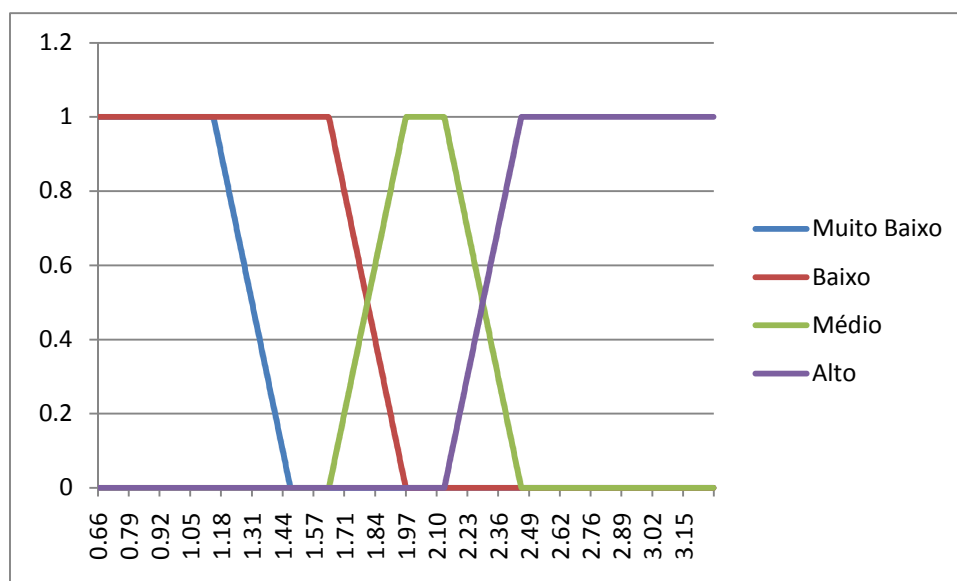
Valores que correspondem a situações especiais (não comuns)

Com base nos valores mínimo e máximo da Tabela 6, na análise de cada *log* individualmente e na minha experiência e base teórica individual, foram definidas os conjuntos nebuloso e suas regras de pertinência da seguinte forma:

- Pressão da Flauta:

**Tabela 7: Funções de Pertinência da Pressão da Flauta**

Valores	Formato	Limites			
		A	B	C	D
Muito Baixo	Left Func	1,15	1,48	-	-
Baixo	Left Func	1,64	1,97	-	-
Médio	Trapezoidal	1,64	1,97	2,13	2,46
Alto	Right Func	2,13	2,46	-	-

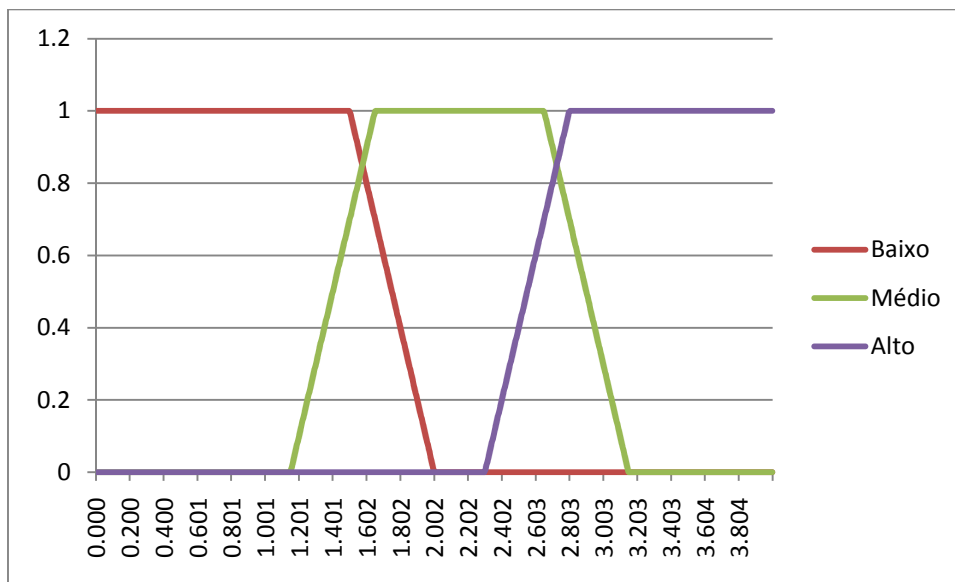


**Figura 74: Funções de Pertinência da Pressão da Flauta**

- Sensor MAP:

**Tabela 8: Funções de Pertinência do Sensor MAP**

Valores	Formato	Limites			
		A	B	C	D
Baixo	Left Func	1,5	2	-	-
Médio	Trapezoidal	1,15	1,65	2,65	3,15
Alto	Right Func	2,3	2,8	-	-

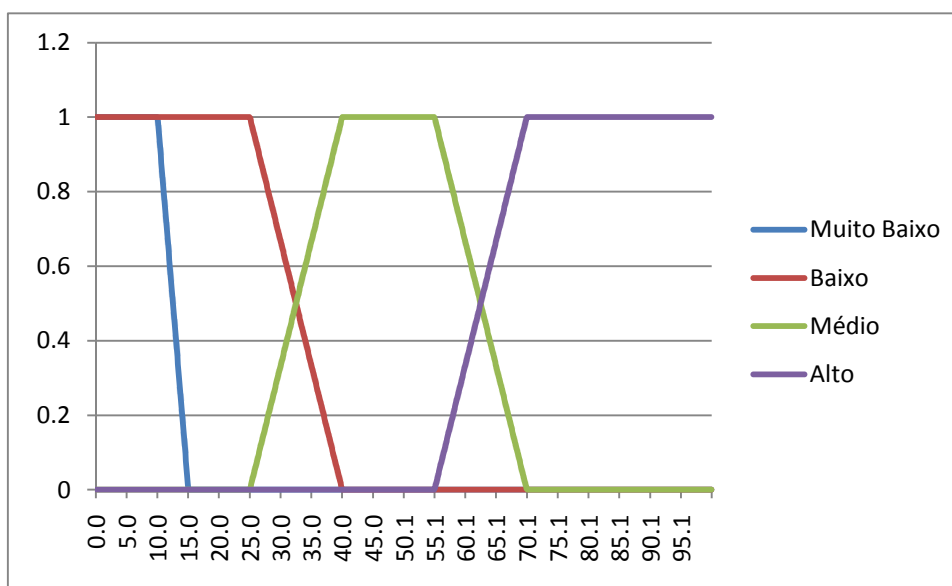


**Figura 75: Funções de Pertinência do Sensor MAP**

- Rotação:

**Tabela 9: Funções de Pertinência do Sensor de Rotação**

Valores	Formato	Limites			
		A	B	C	D
Muito Baixo	Left Func	10	15	-	-
Baixo	Left Func	25	40	-	-
Médio	Trapezoidal	25	40	55	70
Alto	Right Func	55	70	-	-

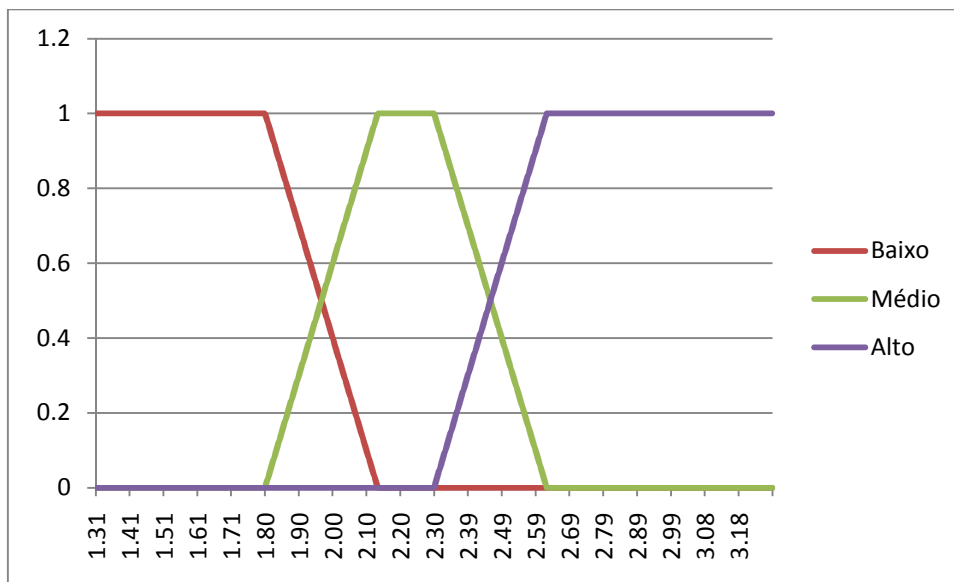


**Figura 76: Funções de Pertinência do Sensor de Rotação**

- Sensor de Temperatura:

**Tabela 10: Funções de Pertinência do Sensor de Temperatura**

Valores	Formato	Limites			
		A	B	C	D
Baixo	Left Func	1,80	2,13	-	-
Médio	Trapezoidal	1,80	2,13	2,30	2,62
Alto	Right Func	2,30	2,62	-	-

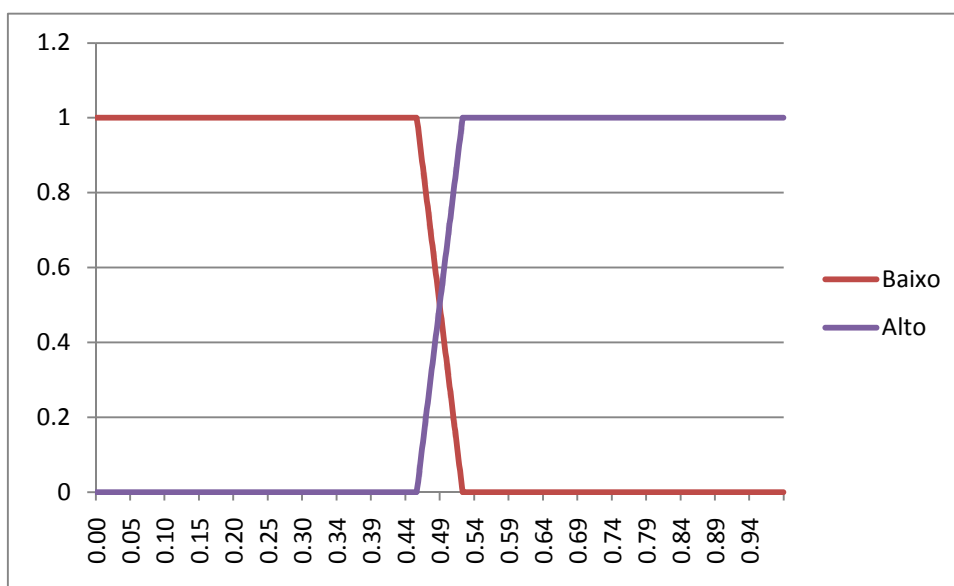


**Figura 77: Funções de Pertinência do Sensor de Temperatura**

- Sensor de Sonda Lambda:

**Tabela 11: Funções de Pertinência do Sensor de Sonda Lambda**

Valores	Formato	Limites			
		A	B	C	D
Baixo	Trapezoidal	0,46	0,52	-	-
Alto	Trapezoidal	0,52	0,46	-	-



**Figura 78: Funções de Pertinência do Sensor de Sonda Lambda**

- Centralina Bico 1 (Tempo de Injeção Original):

**Tabela 12: Funções de Pertinência do Sensor de Pressão da Flauta**

Valores	Formato	Limites			
		A	B	C	D
Baixo	Trapezoidal	0,003	0,009	-	-
Médio	Trapezoidal	0,003	0,009	0,012	0,018
Alto	Trapezoidal	0,018	0,012	-	-

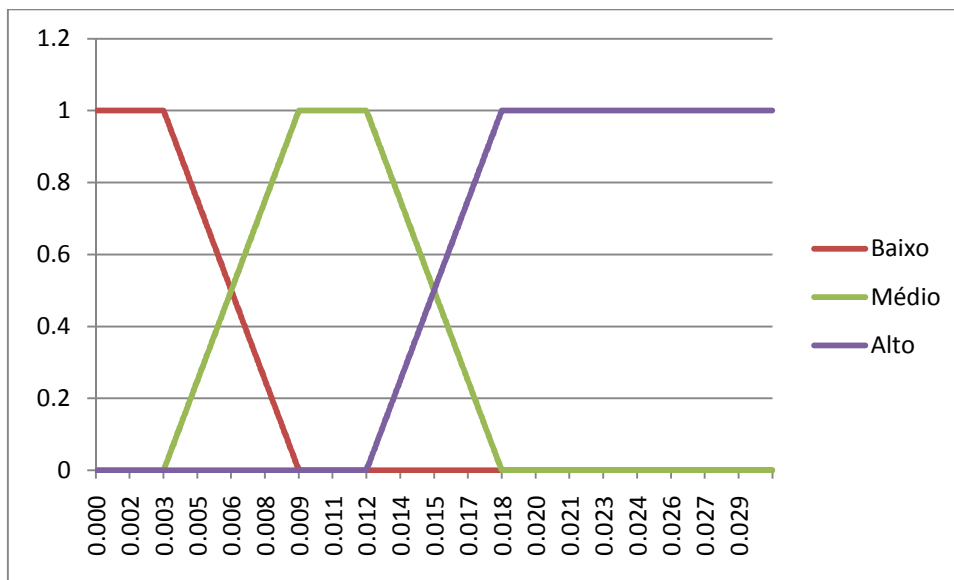


Figura 79: Funções de Pertinência do Sensor de Pressão da Flauta

## 5.2.4 Determinação dos Predecessores

Após a determinação das funções de pertinência e antes de determinar a base de regras, foi necessário determinar quais conjuntos de predecessores deveriam compor as regras. Para isso, foram feitos diversos testes para se desenvolver uma intuição de quais predecessores deveria ser combinados.

O mais importante dessa etapa é compreender que a determinação dos predecessores foi feita de forma empírica. Foram feitos inúmeros testes, variando-se as estratégias e parâmetros intuitivamente e verificando qual caminho produzia melhores resultados.

Alguns dos testes feitos foram:

- Teste de todas as possibilidades de regras com 1 predecessor, sendo esses predecessores relativos às grandezas:
  - Pressão da Flauta, Sensor MAP, Rotação, Sensor de Temperatura, Sensor de Sonda Lambda e Sensor de Pressão da Flauta;

Testando-se as condições:

- “Se (predecessor)” e “Se não (predecessor)”

Para todas as funções de pertinência de cada grandeza.

As saídas das regras representavam uma variação porcentual no tempo de injeção para mais ou para menos. Para cada regra, foram testadas:

- Saída entre -20% e 20%, com variação de 1% em 1%.

- Teste de todas as possibilidades de regras com 2 predecessores, sendo esses predecessores relativos às grandezas:

- Pressão da Flauta, Sensor MAP, Rotação, Sensor de Temperatura, Sensor de Sonda Lambda e Sensor de Pressão da Flauta;

Testando-se as condições:

- “Se (predecessor 1) e (predecessor 2)”;
- “Se não (predecessor 1) e (predecessor 2)”;
- “Se (predecessor 1) e não (predecessor 2)”;
- “Se não (predecessor 1) e não (predecessor 2).”

Para todas as funções de pertinência de cada grandeza.

As saídas das regras representavam uma variação porcentual no tempo de injeção para mais ou para menos. Para cada regra, foram testadas:

- Saída entre -20% e 20%, com variação de 1% em 1%.

Como resultado dos testes, verificou-se que os conjuntos nebulosos seguintes grupos de grandezas produziam resultados expressivos (baixo *erro*) e geralmente apareciam associados:

- Rotação, Pressão Flauta e  $TI_{Gasolina}$ ;
- Sonda Lambda, Pressão Flauta e  $TI_{Gasolina}$ ;

Os predecessores dessas grandezas foram então escolhidos para compor a base de regras, associados conforme descrito acima.

## 5.2.5 Criação da Base de Regras

Com os predecessores escolhidos, a decisão a se tomar passou a ser quais predecessores deveriam ser associados. Embora na análise feita na seção 5.2.4 algumas associações surgissem com mais frequência e mais força (como a regra “Se não Pressão da Flauta é Média e TI é Médio então correção é -6%”, com um *erro* resultante de 434), optou-se por fazer todas as associações possíveis dentre os grupos de grandezas já definidos:

- Rotação, Pressão Flauta e  $TI_{Gasolina}$ ;
- Sonda Lambda, Pressão Flauta e  $TI_{Gasolina}$ ;

O intuito foi conseguir o melhor resultado possível para esse grupo de grandezas, abrangendo todas as situações de funcionamento do veículo no que diz respeito a elas.

Por fim, decidiu-se por utilizar apenas uma condição:

- “Se (predecessor 1) e (predecessor 2) e (predecessor 3) então (saída)”.

Como foram implementadas todas as combinações possíveis, acredita-se que a falta de uma regra com um formato diferente seria suprida por uma ou mais regras com o formato definido acima. Por exemplo, a regra:

- “Se rotação é baixa e pressão de flauta **não** é alta e  $TI_{Gasolina}$  é baixo”

Teria sua falta suprida pelas regras:

- “Se rotação é baixa e pressão de flauta é baixa e  $TI_{Gasolina}$  é baixo”
- “Se rotação é baixa e pressão de flauta é média e  $TI_{Gasolina}$  é baixo”

Já que essas regras abrangem os casos em que “pressão da flauta não é alta”.

A determinação dos valores de saída de cada regra foi feita novamente por meio de um algoritmo computacional que testasse várias possibilidades. O algoritmo funciona da seguinte forma:

1. A saída de todas as regras é iniciada com 0% de correção.
2. Faça enquanto houver uma melhor de pelo menos 0,2% no *erro*
  - 2.1. Para todas as funções de pertinência de Rotação
    - 2.1.1. Para todas as funções de pertinência de Pressão de Flauta
      - 2.1.1.1. Para todas as funções de pertinência de  $TI_{Gasolina}$ 
        - 2.1.1.1.1. Calcule o grau de pertinência da regra: “Se (função de pertinência de Rotação) e (função de pertinência de Pressão da Flauta) e (função de pertinência de  $TI_{Gasolina}$ )”
        - 2.1.1.1.2. Varie a saída da regra de -5% a +5%, com passos de 1%
        - 2.1.1.1.3. Calcule o *erro* resultante para cada saída
        - 2.1.1.1.4. Faça com que a saída da regra seja igual àquela que resulte no menor *erro*
  - 2.2. Para todas as funções de pertinência de Sonda Lambda
    - 2.2.1. Para todas as funções de pertinência de Pressão de Flauta
      - 2.2.1.1. Para todas as funções de pertinência de  $TI_{Gasolina}$ 
        - 2.2.1.1.1. Calcule o grau de pertinência da regra: “Se (função de pertinência de Sonda Lambda) e (função de pertinência de Pressão da Flauta) e (função de pertinência de  $TI_{Gasolina}$ )”

- 2.2.1.1.2. Varie a saída da regra de -5% a +5%, com passos de 1%
- 2.2.1.1.3. Calcule o *erro* resultante para cada saída
- 2.2.1.1.4. Faça com que a saída da regra seja igual àquela que resulte no menor *erro*

Ou seja, a cada iteração as saídas de todas as regras eram revistas e podiam sofrer uma variação de até 5% para mais ou para menos. Com isso, todas as regras foram sofrendo pequenas variações em suas saídas a cada iteração, fazendo com que a base de regra ficasse cada vez mais consistente. Esse algoritmo garante ainda que a cada iteração o *erro* resultante da base de regras seja sempre menor ou igual ao *erro* da iteração anterior, já que a saída de cada regra só sofre alteração quando essa alteração resulta em um melhor desempenho (menor *erro*).

O resultado desse algoritmo está representado nas Tabela 13 e Tabela 14.

Para facilitar a análise do resultado, as Tabela 15, Tabela 16, Tabela 17 e Tabela 18 foram criadas. Nessas tabelas, é apresentado o valor médio das saídas das regras nas quais está presente a função de pertinência. Algumas observações:

- 16 das 19 regras com saída  $\geq 20\%$  tem o predecessor "TI é Baixo". Esse fato está em acordo com a alta média das saídas das regras que contém esse predecessor (60%). Ou seja, quando o  $TI_{Gasolina}$  é baixo, a correção deve ser maior;
- 10 das 14 regras com saída  $\leq -20\%$  tem o predecessor "Pressão da Flauta é Baixa". Esse fato está em acordo com a baixa média das saídas das regras que contém esse predecessor (-54%). Ou seja, quando a Pressão da Flauta está baixa, a correção deve ser menor. É bastante provável que o sensor de pressão da flauta seja inversamente proporcional e que valores de saída baixos do sensor representem valores altos de pressão da flauta, exigindo uma diminuição do tempo de injeção de GNV;
- Destacam-se as regras "Pressão da Flauta é Alta" e "Sonda Lambda é Alta", com altas médias: 27% e 33%, respectivamente;
- Destaca-se as regras "TI é Alto", com baixa média: -36%;



**Tabela 13: Base de Regras para Rotação, Pressão da Flauta e TI<sub>Gasolina</sub>**

Rotação	Pressão da Flauta	TI	Output (%)
Muito Baixa	Muito Baixa	Baixo	0%
Baixa	Muito Baixa	Baixo	0%
Média	Muito Baixa	Baixo	106%
Alta	Muito Baixa	Baixo	100%
Muito Baixa	Baixa	Baixo	23%
Baixa	Baixa	Baixo	-19%
Média	Baixa	Baixo	36%
Alta	Baixa	Baixo	6%
Muito Baixa	Média	Baixo	-6%
Baixa	Média	Baixo	23%
Média	Média	Baixo	155%
Alta	Média	Baixo	155%
Muito Baixa	Alta	Baixo	55%
Baixa	Alta	Baixo	34%
Média	Alta	Baixo	155%
Alta	Alta	Baixo	155%
Muito Baixa	Muito Baixa	Médio	0%
Baixa	Muito Baixa	Médio	0%
Média	Muito Baixa	Médio	0%
Alta	Muito Baixa	Médio	0%
Muito Baixa	Baixa	Médio	-119%
Baixa	Baixa	Médio	-148%
Média	Baixa	Médio	-155%
Alta	Baixa	Médio	-45%
Muito Baixa	Média	Médio	45%
Baixa	Média	Médio	6%
Média	Média	Médio	-2%
Alta	Média	Médio	2%
Muito Baixa	Alta	Médio	59%
Baixa	Alta	Médio	-8%
Média	Alta	Médio	-5%
Alta	Alta	Médio	2%
Muito Baixa	Muito Baixa	Alto	0%
Baixa	Muito Baixa	Alto	0%
Média	Muito Baixa	Alto	0%
Alta	Muito Baixa	Alto	0%
Muito Baixa	Baixa	Alto	0%
Baixa	Baixa	Alto	0%
Média	Baixa	Alto	-155%
Alta	Baixa	Alto	-155%
Muito Baixa	Média	Alto	-155%
Baixa	Média	Alto	-52%
Média	Média	Alto	2%
Alta	Média	Alto	5%
Muito Baixa	Alta	Alto	-155%
Baixa	Alta	Alto	-14%
Média	Alta	Alto	2%
Alta	Alta	Alto	0%

**Tabela 14: Base de Regras para Sonda Lambda, Pressão da Flauta e  $TI_{Gasolina}$**

Sonda Lambda	Pressão da Flauta	TI	Output (%)
Baixa	Muito Baixa	Baixo	117%
Alta	Muito Baixa	Baixo	0%
Baixa	Baixa	Baixo	-23%
Alta	Baixa	Baixo	19%
Baixa	Média	Baixo	-16%
Alta	Média	Baixo	155%
Baixa	Alta	Baixo	55%
Alta	Alta	Baixo	155%
Baixa	Muito Baixa	Médio	0%
Alta	Muito Baixa	Médio	0%
Baixa	Baixa	Médio	-155%
Alta	Baixa	Médio	69%
Baixa	Média	Médio	6%
Alta	Média	Médio	0%
Baixa	Alta	Médio	9%
Alta	Alta	Médio	0%
Baixa	Muito Baixa	Alto	0%
Alta	Muito Baixa	Alto	0%
Baixa	Baixa	Alto	-155%
Alta	Baixa	Alto	0%
Baixa	Média	Alto	-25%
Alta	Média	Alto	0%
Baixa	Alta	Alto	-17%
Alta	Alta	Alto	0%

**Tabela 15: Resumo das saídas dos Conjuntos Nebulosos de Rotação**

Rotação	Média (%)
Muito Baixa	-21%
Baixa	-15%
Média	12%
Alta	19%

**Tabela 16: Resumo das saídas dos Conjuntos Nebulosos de Sonda Lambda**

Sonda Lambda	Média (%)
Baixa	-17%
Alta	33%

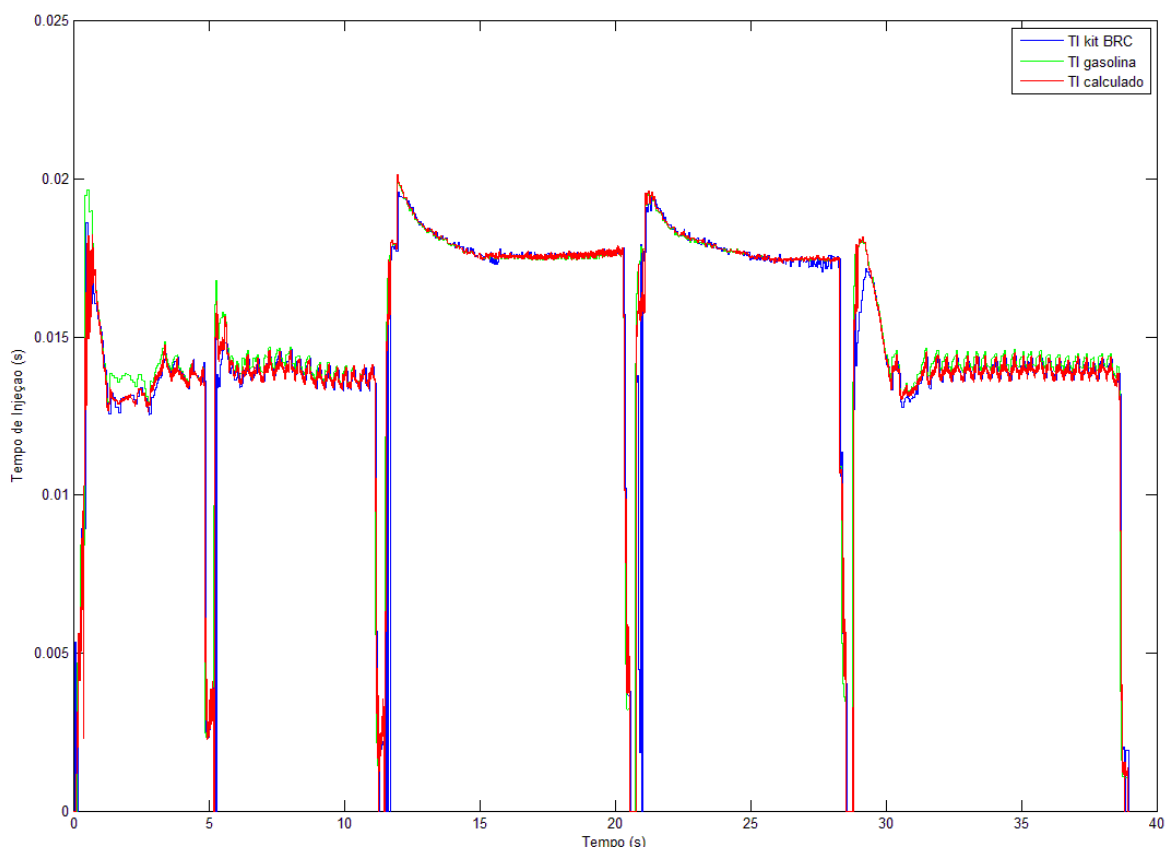
**Tabela 17: Resumo das saídas dos Conjuntos Nebulosos de Pressão da Flauta**

Pressão da Flauta	Média (%)
Muito Baixa	18%
Baixa	-54%
Média	17%
Alta	27%

**Tabela 18: Resumo das saídas dos Conjuntos Nebulosos de TI<sub>Gasolina</sub>**

TI	Média (%)
Baixo	60%
Médio	-18%
Alto	-36%

O *erro* resultante dessa base de regras foi de 334, uma melhora de ~30%. Embora não pareça muito significativo, esse resultado pode ser melhor analisado a partir do gráfico apresentado na Figura 80. Nele, pode-se ver claramente que o TI calculado pela base de regras criada (em vermelho) acompanha na maior parte dos casos o cálculo feito pelo *kit* comercial (em azul), em contraste com as divergências apresentadas em algumas situações entre o TI de injeção de gasolina e o Tempo de Injeção de GNV.



**Figura 80: Comparação entre os tempos de injeção de gasolina, de gás do *kit* comercial e o calculado pelo sistema de inferência nebuloso**

### 5.3 Conclusões

A métrica criada para o cálculo da eficiência da base de regras nebulosa (seção 5.2.1) se baseia na diferença entre o sinal gerado por essa base de regras e o sinal do *kit* comercial instalado no veículo, em uma tentativa de fazer com que a base de regras torne o funcionamento do sistema desenvolvido o mais parecido possível com o funcionamento do *kit* comercial. Essa métrica deve ser validada pela avaliação prática da eficiência da base de regras.

Outro ponto a se considerar é que segundo essa mesma métrica, a melhoria promovida pela base de regras não foi muito significativa, apesar das inúmeras tentativas em se definir bases de regras diferentes. No entanto, a análise gráfica do resultado (Figura 80) cria uma expectativa positiva para o teste dessa estratégia.

## 6 O PROJETO DO *FIRMWARE* DO CONTROLADOR

O *firmware* de um produto eletrônico é o programa inserido diretamente no algum componente do *hardware*. Embora não exista uma divisão formal entre *firmware* e *software*, o primeiro normalmente é associado a componentes eletrônicos que normalmente não sofrem atualizações depois de gravados no componente e realizam tarefas muito específicas, incluindo acesso ao *hardware*.

Nessa seção será apresentado o *firmware* do produto desenvolvido, partindo dos conceitos que guiaram o desenvolvimento desse *firmware* (seção 6.1), passando pelo modo em que o código foi estruturado (seção 6.2), pela estratégia de escalonamento das tarefas (seção 6.4), pela aplicação (seção 6.5) e finalizando com uma seção sobre as funções criadas (seção 6.6).

### 6.1 Conceitos que Guiaram o Desenvolvimento

No desenvolvimento de um *firmware* existem preocupações que devem ser levadas em consideração desde o primeiro momento:

- Além de desempenhar a função à qual se propõe, o *firmware* deve ser de fácil manutenção para correção de *bugs*, evolução de estratégias de funcionamento e inclusão de novas funcionalidades;
- O código deve ser estruturado de maneira a evitar que trechos de código sejam replicados em diferentes partes, isso tem dois motivos principais:
  - Uma vez que seja escrito um trecho que desempenhe uma determinada função e que esse trecho tenha sido testado e validado, ele pode ser utilizado com segurança por todo o restante do código. Além disso, quando o trecho precisar ser corrigido ou otimizado, será necessário que essa modificação seja aplicada em apenas um lugar do código, evitando que algum trecho seja esquecido e conseqüentemente não corrigido ou otimizado;
  - Trechos replicados de código aumentam o mesmo, consumindo mais memória de programa.
- O código deve ser estruturado de uma maneira que uma eventual troca de microcontrolador possa ser feita com o máximo de reutilização de código possível. Isso significa que os trechos de código em que haja acesso aos registradores devem ser isolados para que apenas esses devam ser reescritos em caso de troca do microcontrolador (ver seção 6.2).

### 6.2 Estruturação do Código

A estruturação de código foi feita de uma maneira a se obter diferentes níveis de abstração para a execução de cada tarefa, a Figura 81 ilustra como foi feita essa estruturação:

- **libATMega32** – O código nessa camada faz a interface do restante do código com os registradores do microcontrolador, incluindo o acesso ao *hardware* – como as interfaces de comunicação UART e SPI, o acesso às IO's e etc.

No caso de uma troca de microcontrolador, essa é a única camada que deve ser completamente reescrita. Para que isso seja possível, a interface deve ser mantida (**MCUDrivers(h)**).

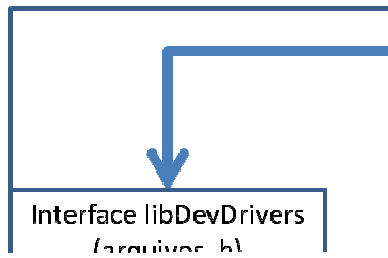
Essa camada é chamada pelas camadas **APPs** e **DEVDrivers** e se utiliza de algumas funcionalidades da camada **Common** (ver mais adiante);

- **libCommon** – Nessa camada estão algumas funcionalidades que podem ser utilizadas por vários trechos no código, mas que não se encaixam em nenhuma das outras camadas. Funções matemáticas e de manipulação de dados e de string são exemplos de funções dessas camadas.

Essa camada pode ser utilizada por todas as outras camadas (**APPs**, **DEVDrivers** e **MCUDrivers**) e não deve acessar nenhuma outra camada;

- **libDevDrivers** – Essa camada possibilita à camada de aplicação (**APPs**) o acesso a funcionalidades de dispositivos conectados ao *hardware*, se utilizando para isso da camada **MCUDrivers** para acesso de baixo nível ao *hardware*. Os dispositivos são o console (debug) – **CONSOLE.c** e **CONSOLE.h** – memória EEPROM (memória externa) – **EEPROM.c** e **EEPROM.h** – e os bicos injetores – **BICOS.c** e **BICOS.h**.
- **GNJet** – Nessa camada se encontra a lógica de funcionamento do produto em alto nível. As funções utilizadas por essa camada se utilizam da abstração criada pelas camadas **DEVDrivers**, **MCUDrivers** e

**Common.**



### 6.3 Bibliotecas desenvolvidas

As bibliotecas libDevDrivers, libCcservirem como base para o desenvolvimento da implementação das funcionalidades do projeto e ao *hardware*.

No Anexo I são apresentados os detalhes da organização desse documento.

Todas as funções implementadas pelo projeto funcionam quando a aplicação as invoca.

As seções a seguir trazem detalhes da

#### 6.3.1 libDevDrivers

A biblioteca libDevDrivers implementa as seguintes funcionalidades:

- Bicos de Gás – Essa biblioteca implementa a injeção de combustível originária do projeto implementadas por essa biblioteca.
  - Após inicializada, a biblioteca realiza a leitura dos bicos para ser corrigida. A Figura 82 realiza a leitura dos bicos e a correção de uma correção esta

## Fi

### A máquina de estado

- A máquina de estado para o combustível
- Assim que o estado “Ler” original e o estado de injeção de -100% a 300%
- Se a injeção para o estado injetores de g  
Nesse caso, “Idle”.
- Se a injeção estado “Ler” combustível  
Nesse caso, e o estado “Idle”
- Contadores i quando a inje

### As tarefas de leitura

- Pode-se realizar o envio via serial dos seguintes formatos:
  - String, inteiro sem sinal de 8 bits, inteiro sem sinal de 16 bits, inteiro com sinal de 16 bits, hexadecimal de 8 bits e hexadecimal de 16 bits.
- É possível habilitar ou desabilitar o envio de dados via console de cada bloco do *firmware* por vez. Para isso, todo o envio via console deve ser identificado com o bloco que deseja enviar o dado. Essa funcionalidade pode ser utilizada para habilitar ou desabilitar o envio de *log* de ações de blocos de *firmware*.
- A biblioteca prevê ainda um funcionamento não “tempo real” do console, onde os bytes só são efetivamente enviados e recebidos quando o usuário da biblioteca (no nosso caso, a aplicação) invoca a função criada para esse fim. Isso dá a possibilidade de impedir que o envio de grandes volumes de dados via Console bloqueie a execução do código.
- EEPROM – Essa biblioteca foi criada para permitir à aplicação escrever e ler dados da memória externa EEPROM de maneira transparente. As funcionalidades implementadas por essa biblioteca são:
  - Escrita de um byte em um endereço específico da memória;
  - Escrita de vários bytes a partir de um endereço específico da memória;
  - Leitura de um byte em um endereço específico da memória;
  - Leitura de vários bytes a partir de um endereço específico da memória;
  - Leitura do status do funcionamento biblioteca (ocupado ou disponível);
  - Para evitar que os ciclos de leitura e escrita na EEPROM travem o funcionamento do código, essas tarefas são feitas a partir da execução de uma máquina de estados que deve ser invocada de tempos em tempos – quanto maior a frequência de execução dessa máquina de estados, mais rapidamente as tarefas executadas por ela serão finalizadas.

A descrição de tal máquina de estados não é pertinente por se tratar de uma aplicação bem específica e não pertinente ao assunto abordado por esta dissertação;

### 6.3.2 libCommon

A biblioteca libCommon implementa funcionalidades gerais importantes para a execução de diversas tarefas. Por seu caráter abrangente de utilização, essas funcionalidades não puderam ser implementadas em nenhuma outra biblioteca. As funcionalidades implementadas por essa biblioteca foram:

- Permitir o acesso a um bit específico de uma palavra de maneira intuitiva e transparente;
- Permitir setar ou resetar de 1 a 8 bits de uma palavra de maneira intuitiva e transparente;
- Converter números de 8 ou 16 bits do formato hexadecimal para o formato decimal;
- Repetir qualquer operação até 16 vezes;
- Comparar duas strings para verificar se são iguais;
- Verificar o tamanho de uma string;

### 6.3.3 libFuzzy

A libFuzzy foi criada para permitir a criação e manipulação de conjuntos e regras nebulosas (*fuzzy*) de forma intuitiva. As funcionalidades implementadas por essa biblioteca foram:

- Criação de até 5 conjuntos nebulosos (*fuzzy*) por grandeza manipulada em lógica nebulosa (*fuzzy*);
- Para cada conjunto nebuloso, configuração de formato entre:
  - Função Direita, Função Esquerda e Função Trapezoidal (ver seção 2.5.2)
- Cálculo do grau de pertinência de cada conjunto nebuloso a partir do valor passado;

### 6.3.4 libATMega32

A libATMega32 foi criada com dois objetivos: permitir o acesso aos periféricos do microcontrolador ATMega32 de maneira transparente para a aplicação e permitir que a troca de microcontrolador seja transparente à aplicação e às demais bibliotecas, sendo necessário reescrever o código apenas dessa biblioteca.

Os periféricos abrangidos por essa biblioteca e as respectivas funcionalidades implementadas estão listadas adiante. Muitas das funcionalidades foram implementadas tanto por macros quanto por funções, a fim de possibilitar a rápida execução das mesmas quando necessário.

- Conversor Analógico Digital (ADC):
  - Habilitar o ADC;
  - Desabilitar o ADC;
  - Iniciar uma conversão;
  - Verificar se a conversão já foi finalizada;
  - Selecionar o canal a ser utilizado na conversão;
  - Selecionar o modo entre:
    - Conversão única, conversão contínua, conversão temporizada (com possibilidades de diferentes fontes de temporização);
  - Selecionar a precisão do ADC entre 8 e 10 bits;
  - Selecionar a referência para conversão entre:
    - Referência externa, referência interna (AVCC) e referência da fonte LDO interna (2,56V);
  - Selecionar a divisão do clock interno para geração do clock utilizado na conversão entre:
    - 2, 4, 8, 16, 32, 64 e 128;
  - Realizar a leitura da conversão AD em 8 ou 10 bits;
- Interrupção Externa:
  - Habilitar uma interrupção externa específica;
  - Desabilitar uma interrupção externa específica;
  - Selecionar a fonte de disparo da interrupção entre:
    - Borda de subida, borda de descida, ambas bordas de transição, nível lógico alto e nível lógico baixo;
  - Adicionar uma função de callback para a interrupção (pode haver mais de uma);
- Comunicação I2C:
  - Gerar sinal de start;
  - Gerar sinal de stop;
  - Enviar um byte;
  - Ler um byte;
  - Configurar a geração ou não de ACK;
  - A comunicação I2C é feita por uma máquina de estados que deve ser chamadas de tempos em tempos para que as ações sejam efetivas. Isso foi feito para evitar que o envio ou recebimento de um byte que por ventura esteja demorando muito tempo bloqueie o funcionamento do código;
- Entradas e Saídas (IO):
  - Para todos os casos, uma porta se refere a um conjunto de pinos;
  - Habilitar ou desabilitar o uso de pull-ups;
  - Configurar o pull-up de um pino;
  - Configurar o pull-up de uma porta;



- Selecionar a direção de um pino (entrada ou saída);
- Selecionar a direção de uma porta (entrada ou saída);
- Ler a direção de um pino (entrada ou saída);
- Ler a direção de uma porta (entrada ou saída);
- Configurar a saída de um pino (0 ou 1);
- Configurar a saída de uma porta (0 ou 1);
- Ler o estado de entrada de um pino (0 ou 1);
- Ler o estado de entrada de uma porta (0 ou 1);
- Timer 0:
  - Selecionar a fonte de clock entre:
    - Desabilitada, interna, interna dividida por 8, 64, 256 ou 1024, externa na borda de subida e externa na borda de descida;
  - Selecionar se timer irá ou não ter um pino de saída (para gerar um sinal de clock ou PWM);
  - Selecionar o nível em que irá acontecer o evento de *compare* (ver mais detalhes no *datasheet* do componente);
  - Selecionar o valor máximo do *timer* antes de reiniciar a contagem;
  - Acrescentar uma função de *callback* ao evento de *overflow*;
  - Acrescentar uma função de *callback* ao evento de *compare*;
  - Ler o valor do timer;
- Timer 1:
  - Selecionar a fonte de clock entre:
    - Desabilitada, interna, interna dividida por 8, 64, 256 ou 1024, externa na borda de subida e externa na borda de descida;
  - Selecionar se timer irá ou não ter um pino de saída (para gerar um sinal de clock ou PWM);
  - Selecionar o nível em que irá acontecer o evento de *compare* dos canais A e B (ver mais detalhes no *datasheet* do componente);
  - Selecionar o valor máximo do *timer* antes de reiniciar a contagem;
  - Acrescentar uma função de *callback* ao evento de *overflow*;
  - Acrescentar uma função de *callback* ao evento de *compare* do canal A;
  - Acrescentar uma função de *callback* ao evento de *compare* do canal B;
  - Ler o valor do timer;
- Timer 2:
  - Selecionar a fonte de clock entre:
    - Desabilitada, interna, interna dividida por 8, 32, 64, 128, 256 ou 1024;
  - Selecionar se timer irá ou não ter um pino de saída (para gerar um sinal de clock ou PWM);
  - Selecionar o nível em que irá acontecer o evento de *compare* (ver mais detalhes no *datasheet* do componente);
  - Selecionar o valor máximo do *timer* antes de reiniciar a contagem;
  - Acrescentar uma função de *callback* ao evento de *overflow*;
  - Acrescentar uma função de *callback* ao evento de *compare*;
  - Ler o valor do timer;
- UART:

- Inicializar a UART, selecionando:
  - Taxa de comunicação (*Baud Rate*);
  - Número de bits utilizados na transmissão de uma palavra: 5, 6, 7, 8 ou 9;
  - Paridade da comunicação: desabilitada, par ou ímpar;
  - Número de stopbits: 1 ou 2;
  - Modo síncrono ou assíncrono;
  - Polaridade do clock para aquisição do dados (borda subida ou descida) – só para o caso da comunicação síncrona;
  - Habilitação ou não da interrupção por recebimento de byte, e a função de *callback* em caso de habilitação;
  - Habilitação ou não da interrupção por envio de byte, e a função de *callback* em caso de habilitação;
- Habilitar e desabilitar a UART;
- Enviar um byte;
- Receber um byte;
- Limpar o buffer de recebimento da UART;

## 6.4 Escalonamento de Tarefas

Durante o funcionamento do *firmware*, várias tarefas estão em curso simultaneamente:

- Leitura do Sensor MAP;
- Leitura do sensor de Sonda Lambda;
- Leitura do Sensor de Rotação;
- Leitura do Sinal de Injeção Original;
- Simulação do MAP;
- Comutação de MAP;
- Simulação da Sonda Lambda;
- Comutação de Sonda Lambda;
- Comutação do Simulador de Bico;
- Sinais de Injeção Gás;
- Acesso à EEPROM;
- Acesso à comunicação Serial;

Uma vez que não é possível executar vários trechos de código simultaneamente, pois existe apenas um núcleo de processamento, o tempo de processamento deve ser dividido para a execução de cada uma das tarefas descritas acima. Para isso, foram utilizados dois *timers* presentes no microcontrolador:

- *Timer 1* – *Timer* de 16 bits com 2 saídas de *compare* (2 canais de PWM) e *prescaler* configurável;
- *Timer 2* – *Timer* de 8 bits com 1 saída de *compare* (1 canal de PWM) e *prescaler* configurável;

Todos os *timers* são capazes de gerar interrupção tanto em eventos de *compare* quanto em eventos de *overflow*.

A Tabela 19 sintetiza como foi feito o escalonamento das tarefas, as colunas dessa tabela são:

- Rotinas de Acesso ao *Hardware* – as rotinas que devem ser escalonadas;
- Temporização – o período de tempo entre execuções de uma tarefa seguido da frequência do PWM que gera o sinal, quando aplicável. Esses tempos foram definidos da seguinte maneira:
  - 32 us para leitura do Sensor de Rotação e 31.250 Hz para simulação dos sensores MAP e Sonda Lambda;

O Timer 1 é utilizado tanto para invocar a função de leitura do sensor de rotação em seu evento de *overflow* quanto para gerar o sinal de PWM para simulação dos sensores de Sonda Lambda e MAP. Por esse motivo, era necessário que timer contasse até no mínimo 255, o que resulta em uma resolução de 8 bits para simulação dos sensores.

Utilizando a frequência interna de 16 MHz, o timer iria sofrer *overflow* a uma frequência de  $16.000.000/256 = 62.500$  Hz – frequência do PWM - ou em períodos de  $1/62.500 = 16\mu s$  – período de chamada da função de leitura do sensor de rotação. O valor de 16us foi utilizado inicialmente como período dessa tarefa, no entanto, testes demonstraram que o processamento não era suficiente para executar essa tarefa a 16us, e esse valor teve que ser revisto.

Configurou-se então o timer para contar até 511, elevando a resolução do PWM para 9 bits, fazendo os canais de PWM operar a uma frequência de 31250 e fazendo a rotina de leitura do sensor de rotação ser chamada a cada 32 us.

- 96 us para leitura do sinal de injeção original e acionamento dos bicos de gás:

Com o intuito de otimizar o uso dos *timers*, optou-se por utilizar-se também o Timer 1 para essa tarefa, mas nesse caso a rotina promove uma divisão de tempo internamente, executando o seu algoritmo apenas a cada 3 eventos de *overflow* do Timer 1. Isso foi feito porque essa rotina demanda mais processamento e não seria possível executá-la a cada 32us. Adicionalmente, considerando que o tempo mínimo de injeção gasolina/álcool encontrado foi de 1ms (ver seção 3.3.4.5), um período de 96us (32us x 3) garante uma precisão melhor que 10% para leitura do sinal, o que imaginou-se ser suficiente.

- 7.812,5 Hz para acionamento do bico de gás no estado *Hold*:

Na seção 3.3.4.1, observou-se que a frequência de acionamento do bico injetor no estado *Hold* por parte do *kit* comercial instalado é de 10kHz e que o *duty cycle* é de 34%. No entanto, o único *timer* presente no microcontrolador que permite configuração de *duty cycle* e de frequência do PWM é o Timer 1, que como já vimos foi utilizado para as rotinas de leitura do sensor de rotação, geração de PWM para simulação dos sensores MAP e de Sonda Lambda, leitura do sinal de injeção original e acionamento dos bicos de gás. Os outros *timers* presentes no microcontrolador permitem a configuração de apenas um dos parâmetros: *duty cycle* ou frequência do PWM, pois configura-se ou o valor máximo de contagem do *timer*, que determina a frequência de *overflow*, ou o valor de *compare*, que determina o *duty cycle*. Como o importante do estado *hold* é manter a corrente de atuação mais baixa do que no estado *peak* e o *duty cycle* de 34% é essencial para isso, a frequência teve que ser selecionada a partir de *prescalers* permitidos pelo microcontrolador.

O timer selecionado foi o Timer 0. Como o valor máximo de contagem não pode ser definido, ele fica sendo igual a 255 (pois o *timer* é de 8 bits), o que naturalmente causa uma divisão da frequência interna por 256, criando uma frequência de *overflow* de  $16.000.000/256 = 62.500$  Hz. Para diminuir essa frequência, existem os *prescalers* internos, que no caso do Timer 0 podem ser: 8, 64, 256 ou 1024. O valor que resulta em uma frequência mais próxima à do *kit* comercial instalado é o *prescaler* de 8, com uma frequência resultante de  $62.500/8 = 7.812,5$  Hz.

- Tempo Real? – indica se a tarefa é ou não de tempo real, ou seja, se a tarefa precisa ser executada em um momento específico ou se o seu adiamento por um curto período de tempo não incorre em prejuízo ao funcionamento do produto;
- Arquivo de Acesso – indica em qual arquivo está a tarefa.
- Timer a ser utilizado – traz o *timer* adotado para a temporização da tarefa. O *timer* escolhido vem sempre acompanhado de algumas letras:
  - Ov – indica que a tarefa deve ser executada no *overflow* do *timer*;
  - OC – indica que o sinal é gerado por um PWM a partir de um evento de *compare* do *timer*. Quando é acompanhado de uma letra, essa letra indica qual canal do *compare* é utilizado para gerar o PWM.

Tabela 19: Escalonamento das tarefas

Rotinas de Acesso ao Hardware:	Granularidade	Tempo Real?	Arquivo de Acesso:	Timer a ser utilizado
<b>Entradas</b>				
Leitura do Sensor MAP	Fundo Rápido	Não	MAP.c	-
Leitura do sensor de Sonda Lambda	Fundo Rápido	Não	SenLambda.c	-
Leitura do Sensor de Rotação	32 us	Sim	SenRot.c	Timer 1 Ov
Leitura do sinal de Injeção original	96 us	Sim	Inj.c	Timer 1 Ov (x3)
<b>Saídas</b>				
Simulação do Sensor MAP	Fundo Rápido; 31.250 Hz	Não	MAP.c	- ; Timer 1 OCA
Comutação de Sensor MAP	Pontual	Não	MAP.c	-
Simulação do sensor de Sonda Lambda	Fundo Rápido; 31.250 Hz	Não	SenLambda.c	- ; Timer 1 OCA
Comutação do sensor de Sonda Lambda	Pontual	Não	SenLambda.c	-
Comutação do Simulador de Bico	Pontual	Não	Inj.c	-
Sinais de Injeção Gás	96 us; 7.812,5 Hz	Sim	Inj.c	Timer 1 Ov (x3); Timer 2 OC
<b>Interfaces</b>				
Acesso à EEPROM	Pontual	Não	EEPROM.c	-
Serial	Pontual	Não	CONSOLE.c	-
- Quando dois timers estão atribuídos a um sinal, o primeiro é usado para atualizar seu valor e o segundo para gerar o seu PWM				
- Para o caso específico do sinal de injeção, a saída pode ser um valor lógico (repouso ou Peak) ou um PWM (Hold)				
- O único timer em que é possível determinar seu valor máximo (TOP) e o valor de comparação (Compare) é o Timer 1				
- Quando há um fator multiplicativo na coluna "Timer a ser utilizado", a rotina será executada uma vez a cada "fator de multiplicação" vezes que ocorrer o evento do timer				

## 6.5 A Aplicação

A aplicação é a parte do código que trata da parte central do funcionamento do produto, utilizando, coordenando e configurando as funcionalidades disponíveis a partir das bibliotecas criadas.

A função da aplicação deste produto é estabelecer a correção a ser aplicada ao tempo de injeção de combustível original para determinação do tempo de injeção de GNV a partir da leitura dos sensores e da aplicação de uma lógica nebulosa a partir dessa leitura.

A Figura 83 traz um fluxograma que ilustra como foi elaborada a aplicação. A rotina é repetida tão rápido quanto possível, utilizando todo o tempo ocioso de processamento do microcontrolador.

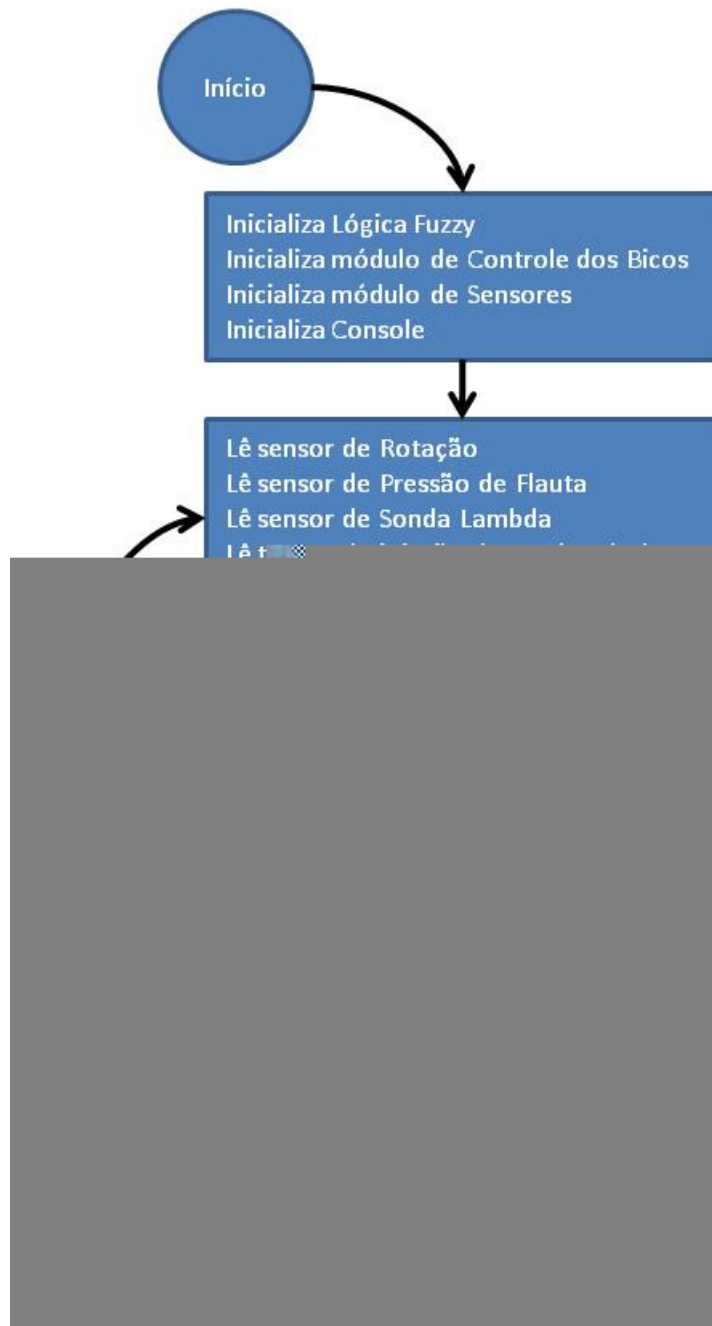


Figura 83: Fluxograma da Aplicação

## 6.6 Funções Criadas

No Anexo I são apresentados os protótipos de todas as funções e macros criadas que integram o *firmware* do produto. Somente os protótipos são apresentados, para que dessa maneira o leitor possa compreender como foi implementada na prática a estruturação apresentada na seção 6.2, sem que para isso o texto desta dissertação fique excessivamente extenso. Para maiores informações, consulte o código anexo a esta dissertação.

# 7 TESTES E RESULTADOS ALCANÇADOS

Nesta seção são apresentados os testes de bancada e de campo, que avaliaram o desempenho do sistema desenvolvido. Os testes de bancada são descritos na seção 7.1 e os testes de campo são descritos na seção 7.2.

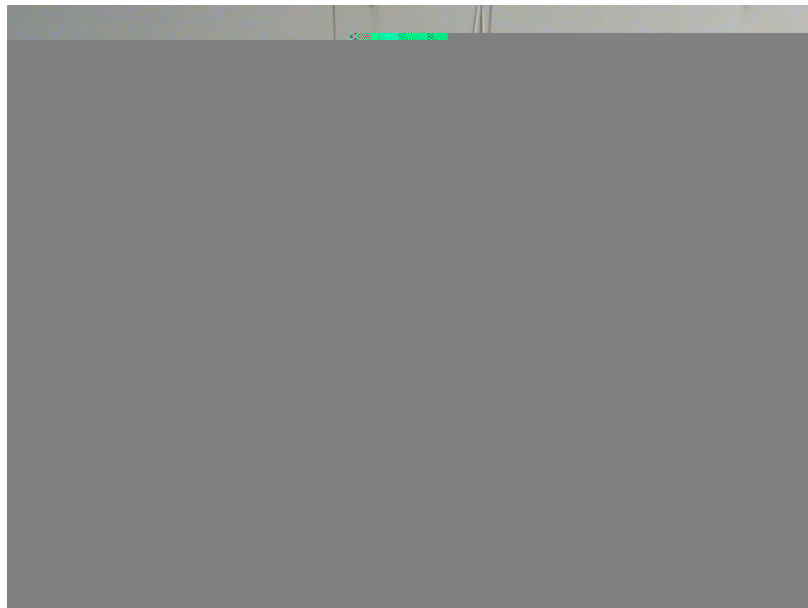
## 7.1 Testes de Bancada

Nessa seção serão apresentados os testes de bancada realizados no conjunto *hardware + firmware* e os resultados desses testes. Os testes têm como objetivo garantir que o conjunto *hardware + firmware* esteja com um projeto adequado, cumprindo suas funções, além de trazer a tona problemas que não tenham sido antecipados.

Para realização do teste foram utilizados:

- 1 Multímetro para medição de tensões;
- 1 Osciloscópio para visualizações dos sinais;
- 1 Fonte regulada para gerar diferentes tensões DC;
- 1 Bateria de no-break de 7Ah para alimentação do sistema com fornecimento de corrente adequado;
- 1 Circuito criado para gerar um sinal de injeção eletrônica emulado;
- 1 Circuito criado para converter o sinal TTL do microcontrolador para RS232 do computador;
- 1 Gravadora AVR Dragon para programação do microcontrolador;

Uma foto da bancada de testes pode ser encontrada na Figura 84.



**Figura 84:** Foto do ambiente de testes de bancada.

### 7.1.1 Fonte de Alimentação

Os testes realizados sobre a fonte de alimentação têm o objetivo de garantir que a fonte é capaz de cumprir suas funções, atendendo às especificações de faixa de entrada de tensão, de tensão de saída e de corrente de saída. Para tanto, foram realizados os seguintes testes:

- Regulação de linha, que verifica a tensão de saída em toda a faixa de operação de tensão de entrada da fonte;
- Regulação de carga, que verifica a tensão de saída em toda a faixa de operação de corrente de saída da fonte;

#### Regulação de Linha

O teste de regulação de linha consiste em variar a tensão de entrada e verificar o funcionamento da fonte de alimentação a partir da medição de sua tensão de saída. O resultado obtido está representado na Tabela 20. Os testes foram feitos com uma corrente de 510 mA na saída da fonte de alimentação.

**Tabela 20: Regulação de linha da fonte de alimentação**

Tensão de Entrada (V)	Tensão de Saída (V)	Regulação de Linha
6,970	4,836	-3,28%
8,050	4,960	-0,80%
9,080	4,965	-0,70%
10,060	4,969	-0,62%
11,000	4,973	-0,54%
12,000	4,977	-0,46%
13,050	4,981	-0,38%
14,040	4,984	-0,32%
14,980	4,986	-0,28%
16,000	4,989	-0,22%
16,970	4,992	-0,16%
18,030	4,994	-0,12%
18,980	4,996	-0,08%
20,000	4,998	-0,04%
20,990	5,000	0,00%
22,030	5,002	0,04%
23,040	5,003	0,06%
24,050	5,005	0,10%
25,040	5,007	0,14%
26,040	5,008	0,16%
27,020	5,009	0,18%
27,930	5,011	0,22%
29,090	5,012	0,24%
30,030	5,012	0,24%
31,020	5,013	0,26%
31,950	5,014	0,28%

### **Regulação de Carga**

O teste de regulação de linha consiste em variar a corrente de saída e verificar o funcionamento da fonte de alimentação a partir da medição de sua tensão de saída. O resultado obtido está representado na Tabela 21. Os testes foram feitos com uma tensão de 12 V na entrada da fonte de alimentação.

O cálculo da regulação de carga é feito a partir da Equação ( 97 ).

$$Regulação\ de\ Carga = \frac{Tensão\ de\ Saída - 5}{5} \quad (97)$$

**Tabela 21: Regulação de carga da fonte de alimentação**

Corrente de Saída (mA)	Tensão de Saída (V)	Regulação de Carga
0,000	5,067	1,34%
123,830	5,014	0,28%
238,310	5,020	0,40%
343,680	5,025	0,51%
505,800	5,035	0,69%
633,100	5,042	0,83%
758,000	5,049	0,98%
882,100	5,056	1,12%
1002,200	5,062	1,25%
1121,700	5,069	1,39%
1242,800	5,076	1,52%
1361,600	5,084	1,67%
1479,400	5,090	1,80%
1596,800	5,095	1,90%
1715,900	5,104	2,08%
1831,800	5,111	2,22%
1935,600	5,115	2,31%
2305,600	4,915	-1,70%
2751,600	4,910	-1,81%
3176,400	4,896	-2,08%

**Análise dos Resultados**

Em ambos os testes, Regulação de Linha e Regulação de Carga, os resultados encontrados foram extremamente satisfatórios, com um valor de desvio máximo de 164 mV, ou 3,28%, em relação ao valor de referência, 5 V. Assim sendo, a fonte de alimentação provou funcionar de 7 a 32 V de tensão de entrada, com saída de corrente até 3,17 A.

**7.1.2 Leitura dos sensores de Pressão da Flauta, Sensor MAP e Sonda Lambda**

Para testar a leitura dos sensores de Pressão da Flauta, Sensor MAP e Sonda Lambda, a entrada desses sensores foi conectada a uma fonte regulável com tensão conhecida (medida por um multímetro) e o valor lido pelos sensores foi enviado para o Console. O valor esperado para cada leitura em função da tensão de entrada (resultado da conversão AD) é calculado pela Equação ( 98 )

$$Valor\ Lido = \text{Mínimo} \left( 255; \left( 256 * \left( \frac{V_{in}}{5V} \right) \right) \right) \quad (98)$$

Onde a divisão o resultado da operação  $256 * \left( \frac{V_{in}}{5V} \right)$  é sempre arredondado para baixo.

A Tabela 22 traz o resultado do teste, juntamente com os valores de leitura esperados.



**Tabela 22: Resultados do teste de leitura dos sensores de Pressão da Flauta, Sensor MAP e Sonda Lambda**

Tensão de Entrada (V)	Valor lido do AD			Valor Esperado
	Pressão da Flauta	MAP	Sonda Lambda	
0,494	25	25	25	25
1,004	51	51	51	51
1,495	76	76	76	76
2,001	102	102	102	102
2,507	128	128	128	128
3,003	153	153	153	153
3,497	178	178	179	179
4,003	204	204	204	204
4,5	230	230	230	230
4,902	250	250	250	250
5,004	255	255	255	255

### Análise dos Resultados

No teste, o único valor lido diferente do esperado foi do sensor de Sonda Lambda na tensão de 3,497 V, no entanto esse não é um mal resultado considerando que uma tensão de 3,496 V corresponderia ao valor lido de 179, ou seja, um pequeno erro de leitura do multímetro ou uma pequena variação na calibração interna do AD podem justificar esse pequeno erro. Por esses motivos, o resultado foi considerado satisfatório.

### 7.1.3 Emulação dos sensores MAP e de Sonda Lambda

Nesse teste, o valor a ser utilizado na emulação dos sensores MAP e de Sonda Lambda foram enviado via Console e a saída foi lida por um multímetro.

Considerando que o timer responsável pela geração do PWM utilizado para emular os sensores MAP e de Sonda Lambda foi configurado para contar até 511 (ver seção 6.4), e que o pino de saída assume o valor de 0 V somente após atingir o valor enviado via console, o valor de saída esperado pode ser calculado pela Equação ( 99 ).

$$Tensão\ de\ Saída = \frac{Valor\ Enviado + 1}{512} * 5V \quad (99)$$

O resultado do teste para o emulador de Sensor MAP é exibido na Tabela 23 e o resultado do emulador de Sonda Lambda é exibido na Tabela 24, juntamente com os valores esperados, o erro entre o valor de saída e o valor esperado para cada caso, o erro máximo e mínimo.

**Tabela 23: Resultado do Teste de Emulação do Sensor MAP**

Valor Escrito	Tensão de Saída (V)	Valor Esperado (V)	Erro (V)
0	0,017	0,010	0,007
50	0,499	0,498	0,001
100	0,982	0,986	-0,004
150	1,465	1,475	-0,010
200	1,947	1,963	-0,016
250	2,43	2,451	-0,021
300	2,913	2,939	-0,026
350	3,396	3,428	-0,032
400	3,878	3,916	-0,038
450	4,361	4,404	-0,043
500	4,844	4,893	-0,049
510	4,941	4,990	-0,049
511	4,95	5,000	-0,050
	Erro máximo		0,007
	Erro mínimo		-0,050

**Tabela 24: Resultado do Teste de Emulação do Sensor de Sonda Lambda**

Valor Escrito	Tensão de Saída (V)	Valor Esperado (V)	Erro (V)
0	0,016	0,010	0,006
50	0,499	0,498	0,001
100	0,981	0,986	-0,005
150	1,464	1,475	-0,011
200	1,946	1,963	-0,017
250	2,429	2,451	-0,022
300	2,912	2,939	-0,027
350	3,395	3,428	-0,033
400	3,877	3,916	-0,039
450	4,36	4,404	-0,044
500	4,843	4,893	-0,050
510	4,94	4,990	-0,050
511	4,949	5,000	-0,051
	Erro máximo		0,006
	Erro mínimo		-0,051

### **Análise dos Resultados**

A variação máxima de tensão entre o valores esperado e o valor emulado foi de -50mV no caso do sensor MAP e de -51mV no caso de sensor de Sonda Lambda. Testes de campo irão indicar melhor se esses valores podem ser considerados satisfatórios ou se é preciso melhorar essa precisão.

#### **7.1.4 Leitura dos Bicos Injetores originais**

Para o teste do circuito de leitura do tempo de injeção dos bicos injetores originais, foi criado um simulador de sinal de bicos injetores. O simulador gera um sinal de 12 V com pulsos de 0 V com tamanho e frequência configuráveis – esse simulador foi criado com base no sinal obtido em campo conforme a seção 3.3.3.1. Para esse teste, foi variado o tamanho do pulso, com frequência fixa em ~24 Hz. O tamanho do pulso gerado foi monitorado com auxílio de um osciloscópio.

Como a taxa de amostragem do tempo de injeção é de 96us (seção 6.4), o valor esperado pode ser calculado segundo a Equação ( 100 ).

$$\text{Valor Lido} = \frac{TI}{96us} \quad (100)$$

A Tabela 25 traz o resultado dos testes juntamente com os valores esperados para cada caso.

**Tabela 25: Resultado dos testes de leitura do tempo de injeção**

Tempo de Injeção (ms)	Valor Lido	Valor Esperado
1,013	10	11
2,011	21	21
3,017	31	31
4,018	42	42
5,028	52	52
6,018	62	63
7,028	73	73
8,037	84	84
9,030	94	94
10,029	104	104

### Análise dos Resultados

Os valores obtidos no teste foram bastante satisfatórios, com apenas um caso com divergência entre o valor lido e o valor esperado, o que é justificável já que a leitura pode variar em  $\pm 96us$  (período de leitura do sinal).

#### 7.1.5 Sensor de Rotação

Para o teste do Sensor de Rotação, foi utilizado o mesmo circuito de simulador de bicos criado para o teste de Leitura dos Bicos de Injeção Originais (seção 7.1.4), mas dessa vez foi alterado o valor da frequência do pulso, mantendo-se o pulso em  $\sim 3$  ms. A frequência do sinal gerado foi medida com o auxílio de um osciloscópio.

A Tabela 26 traz o resultado dos testes, com a frequência do sinal de entrada em Hz e RPM, o valor lido em Hz e o erro entre o valor lido e o valor do sinal de entrada em Hz.

**Tabela 26: Resultado dos testes do Sensor de Rotação**

Freq (Hz)	Frequência (RPM)	Valor Lido (Hz)	Erro (Hz)
10	593	9	-1
16	984	16	0
33	1972	32	-1
48	2908	47	-1
62	3745	62	0
77	4606	76	-1
91	5441	90	-1
100	5994	99	-1
111	6659	111	0
125	7489	125	0
143	8556	142	-1
166	9978	167	1

## Análise dos Resultados

Os valores obtidos no teste foram bastante satisfatórios, com erros de no máximo 1 Hz para mais ou para menos, o que não deve influenciar significativamente o funcionamento do sistema.

### 7.1.6 Acionamento do Bico de Gás

Para o teste do circuito de acionamento do bico de gás, o sinal do simulador de bicos foi colocado na entrada de injeção original e o *firmware* foi modificado para copiar o tempo de injeção de entrada (do simulador). Então, um bico injetor de gás foi conectado ao circuito de acionamento de bicos de gás, e foram monitorados sua tensão e corrente.

O perfil de tensão do circuito de acionamento do bico é mostrado na Figura 85, já o perfil da corrente de acionamento é ilustrado na Figura 86.

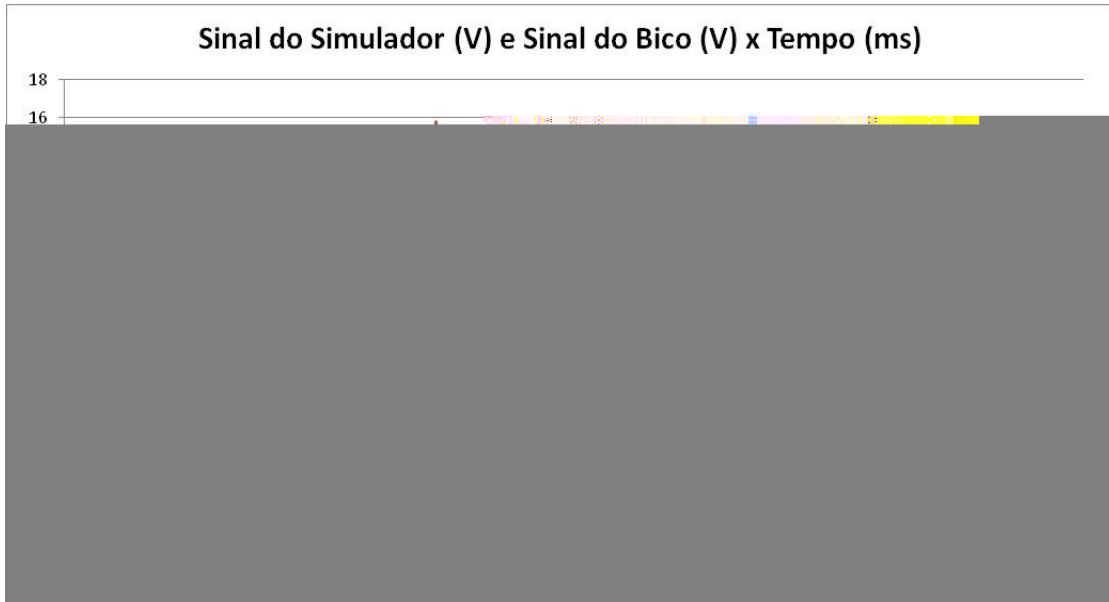


Figura 85: Teste do Acionamento do Bico de Gás

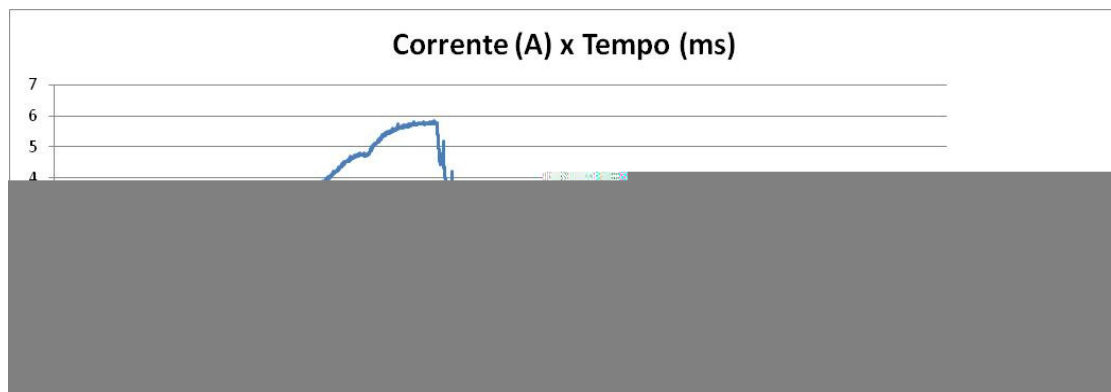


Figura 86: Teste da Corrente de Acionamento do Bico de Gás

## Análise dos Resultados

O resultado da Figura 85 traz duas importantes informações: a primeira diz respeito à capacidade do circuito de acionamento de criar um caminho de corrente para o GND, que foi atendida satisfatoriamente considerando que quando o bico está acionado a tensão fica abaixo de  $\sim 700\text{mV}$ , a segunda diz respeito a atuação dos tranzorbs de proteção, que limitaram a tensão sobre o circuito de acionamento em  $\sim 15\text{V}$ , protegendo-o do surto causado pelo acionamento de uma carga indutiva (o bico de acionamento).

O resultado da Figura 86 é mais ilustrativo, e sua semelhança com o resultado da Figura 44 da seção 3.3.4.2 demonstra que o circuito está funcionando como esperado: durante o período de *Peak* a corrente sobe até seu máximo de  $\sim 6\text{A}$ , e durante o período de *Hold* a corrente diminui para  $\sim 33\%$  da corrente máxima.

## 7.2 Testes de Campo

Esta seção descreve os testes de campo e seus resultados. Nos testes de campo, o sistema desenvolvido é testado no ambiente no qual deve funcionar, são os testes de campo que dizem se o sistema está pronto ou não para se tornar um produto.

São testados intrinsecamente nos testes de campo o *hardware*, o *firmware* e a estratégia de funcionamento.

### 7.2.1 Metodologia de Testes

Os métodos utilizados para realizar os testes de campo são descritos nesta seção.

Com a observação do funcionamento do *kit* comercial instalado, foi possível constatar que a função do *kit* de injeção de GNV vai além de calcular o tempo de injeção de GNV a partir do tempo de injeção de combustível original, mas passa também por alimentar o sistema de GNV, alimentar os sensores, trocar informações com a chave comutadora, determinar o momento adequado de comutação a partir da temperatura (seção 3.3.4.9) e gerenciar a estratégia *Peak and Hold* – por algum motivo, o *kit* comercial instalado inicia seu funcionamento com a estratégia de funcionamento convencional e após alguns minutos passa a atuar na estratégia *Peak and Hold*.

Por esse motivo, optou-se por permitir ao *kit* comercial instalado comandar o acionamento dos bicos injetores até que o sistema estivesse funcionando a gás (já na temperatura certa), na estratégia *Peak and Hold*. Após isso acontecer, o sistema desenvolvido assumia somente o controle dos bicos de gás.

Para assumir o controle dos bicos de gás, o circuito projetado para congelar os bicos injetores de gasolina (seção 4.2.8) foi utilizado para anular a atuação do *kit* comercial instalado sobre os bicos de gás, inserindo uma impedância em série com o bico injetor de gás no caminho criado por esse *kit*, limitando a corrente que flui por esse bico e, portanto, impedindo-o de ser acionado. Ao mesmo tempo, o sistema desenvolvido passa a atuar nos bicos injetores de gás, assumindo o controle da injeção. Na Figura 87 pode-se encontrar o circuito utilizado.

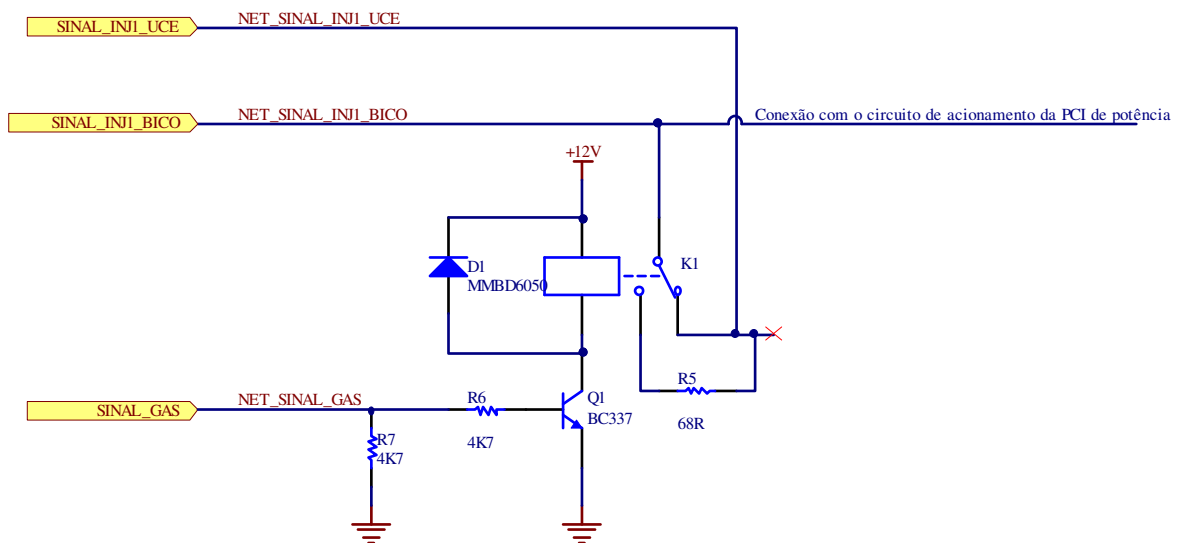


Figura 87: Circuito para congelamento do acionamento dos bicos por parte do kit comercial

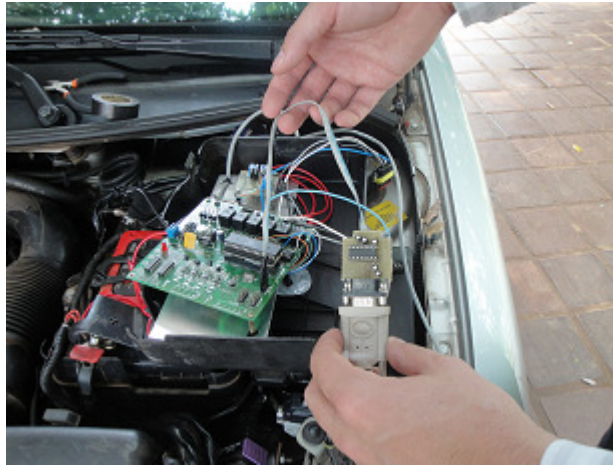
Para leitura dos sinais dos sensores e da injeção original foi utilizado um conector DB25, conforme a foto da Figura 88, para interromper o sinal de acionamento do bico de gás por parte do *kit* comercial, foi utilizado um par de conector selado de 4 vias, conforme a foto da Figura 89, onde um conector se liga aos bicos de gás e o outro ao *kit* comercial, e por último, para comunicação com o console para debug do sistema criado foi utilizada uma placa desenvolvida com essa finalidade (com um CI MAX232), conforme foto da Figura 90. As fotos: Figura 91, Figura 92 e Figura 93 mostram uma visão geral do sistema inserido no veículo, e, por último, a Figura 94 ilustra o sistema sendo comandado pelo computador. A PCI dos circuitos analógicos e digitais é a PCI verde à esquerda na Figura 92 e a PCI dos circuitos de potência é a PCI marrom à direita da mesma figura.



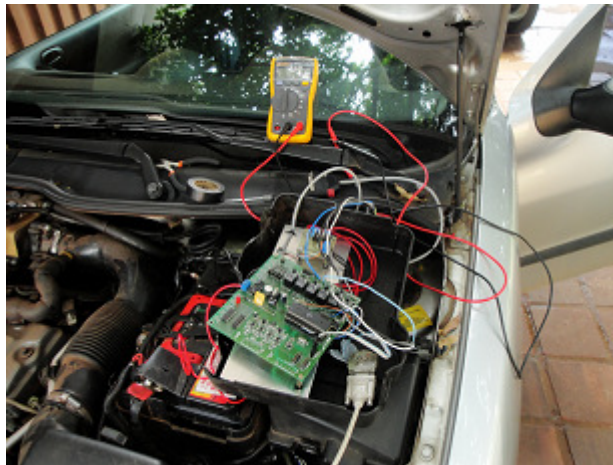
**Figura 88: Foto com o detalhe da conexão para leitura dos sinais dos sensores e da injeção de combustível original.**



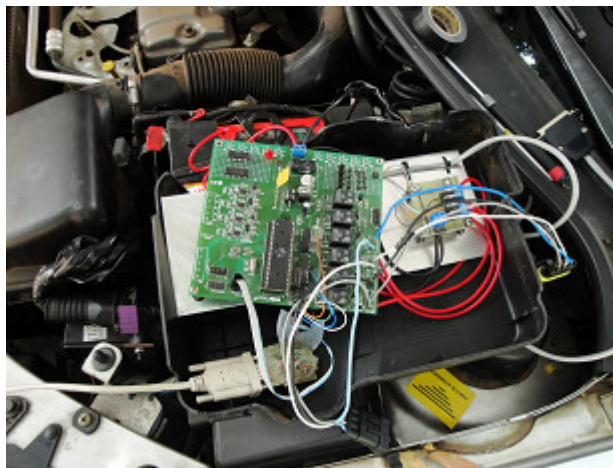
**Figura 89: Conexão que permite ao sistema em teste interromper o sinal de acionamento dos bicos de gás.**



**Figura 90: Placa para leitura do console**



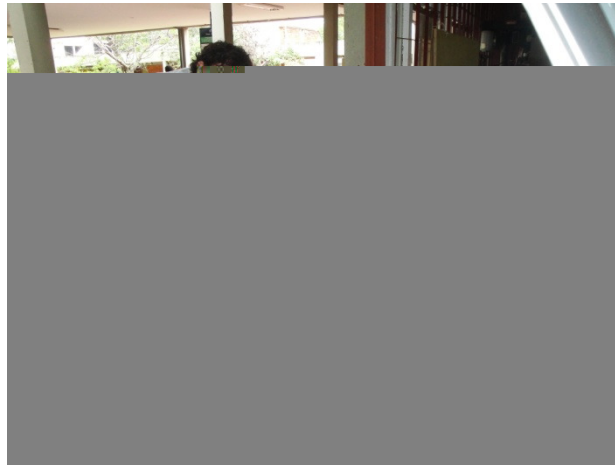
**Figura 91: Visão geral do sistema testado inserido no veículo de teste, 1ª foto.**



**Figura 92: Visão geral do sistema testado inserido no veículo de teste, 2ª foto.**



**Figura 93: Visão geral do sistema testado inserido no veículo de teste, 3ª foto.**



**Figura 94: Sistema sendo comandado a partir do computador.**

Foram testadas duas estratégias:

- Cópia do tempo de injeção com aplicação de uma correção fixa;
- Sistema de acionamento com utilização da estratégia utilizando a lógica nebulosa desenvolvida, descrita na seção 5.

## 7.2.2 Resultados

A análise dos resultados desses testes possui um caráter qualitativo, passando pela percepção da pessoa que realiza o teste sobre o bom funcionamento ou não do motor, pelo som produzido por esse. O parâmetro mais definido utilizado foi a capacidade do sistema em manter o motor funcionando sem deixar o carro morrer.

Em ambos os casos, quando o sistema desenvolvido assumia o controle dos bicos injetores durante o funcionamento em marcha lenta, o carro morria. Porém, quando a comutação entre os dois sistemas (*kit* comercial para o sistema desenvolvido) era feita com o motor levemente acelerado (rotação mais alta), a estratégia de cópia do tempo de injeção com correção fixa foi capaz de manter o motor funcionando, embora na maioria das vezes o motor morria quando o veículo voltava a funcionar em marcha lenta (em uma das vezes o motor se manteve funcionando por vários minutos). Foram testadas várias correções diferentes, de -50% a +100% do tempo de injeção de gasolina, e o que os testes indicaram é que a correção mais adequada variava de um dia de teste para outro, talvez por causa da pressão de gás no cilindro (já que o reservatório foi diminuindo a quantidade de gás), ou por causa da temperatura do motor (ou mesmo por causa da temperatura ambiente).

A estratégia utilizando lógica nebulosa criada mostrou ter um funcionamento menos regular do que a correção fixa, fazendo o motor morrer rapidamente após o sistema desenvolvido assumir a injeção de gás. Após uma análise mais minuciosa da base de regras, alguns problemas a serem corrigidos foram constatados:

- A correção pelo sensor de Sonda Lambda não faz sentido da maneira como foi utilizada, já que na média as regras diminuem o tempo de injeção de gás quando a mistura está pobre (sonda lambda baixa) e aumentam o tempo de injeção de gás quando a mistura está rica (sonda lambda alta) – ver seção 5.2.5. Essa correção é resultado do método matemático utilizado, que associou o baixo tempo de injeção de gás com



um sinal baixo de sonda lambda, e o alto tempo de injeção de gás com um alto sinal de sonda lambda, ignorando o fato que o tempo de injeção determinava o valor do sensor de Sonda Lambda, e não o contrário;

- A ausência do sinal do Sensor MAP pode ser considerada uma grande fraqueza do sistema, já que esse sensor carrega informações importantíssima sobre o regime de funcionamento do motor;
- A utilização do arquivo “aceleracao\_0\_100\_trocaMarcha\_3” para criação da base de regras não se mostrou adequada para manter o funcionamento do motor em marcha lenta, já que representa uma situação muito diferente de funcionamento do motor;
- O fato do motor possuir várias estratégias de diferentes para diferentes condições de funcionamento foi negligenciada, resultando na tentativa de criação de uma base de regras que abrangesse todas as estratégias de funcionamento, e portanto bem mais difícil de ser atingida;
- O método de avaliação da base de regras (com criação de um parâmetro de erro entre o sinal gerado pela base de regras e o gerado pelo *kit* comercial) não se mostrou adequado para criação de uma base de regras eficiente, sendo o teste de campo muito mais pertinente para avaliar essa eficácia.

Resumidamente, a tentativa de tratar o funcionamento do motor como uma “caixa preta”, ignorando os seus aspectos já conhecidos e buscando apenas se aproximar o máximo do funcionamento do *kit* comercial instalado, não gerou bons resultados. Uma nova estratégia que leve em consideração aspectos já conhecidos do funcionamento do motor e a observação de testes de campo (e não apenas de simulações matemáticas) parece ser uma estratégia mais adequada. Por último, parece ser mais adequado criar a base de regras testando o motor em cada situação individualmente, isso é, determinando qual a correção mais adequada para cada situação de cada variável de entrada do sistema de inferência nebuloso (rotação baixa, média e alta, velocidade alta, média e alta, tempo de injeção baixo, médio e alto e etc.).

## 8 CONCLUSÕES E TRABALHOS FUTUROS

Este documento apresentou o desenvolvimento de uma plataforma, constituída de duas PCI's e uma base de código embarcado, capaz de realizar a injeção de GNV e na qual se pode criar e testar diferentes estratégias de cálculo de tempo de injeção de GNV. Adicionalmente, foi realizado um estudo que teve por objetivo empregar o uso de lógica nebulosa no controle de injeção de GNV em veículos originalmente movidos à gasolina e/ou álcool, com o intuito de minimizar a necessidade de processamentos complexos e de armazenamento de mapas, resultando em um *hardware* de menor custo e competitivo.

O *hardware* e *firmware* da plataforma desenvolvida, necessária à validação da estratégia de cálculo criada, foram especificados, projetados, desenvolvidos e testados com sucesso. As funcionalidades implementadas se mostraram eficientes e precisas. O teste de campo serviu para confirmar que o sistema desenvolvido é capaz de realizar o controle do sistema de injeção de GNV, já que foi capaz de manter o funcionamento do motor em marcha lenta, ainda que de maneira instável.

Foram testadas diversas estratégias de cálculo do tempo de injeção baseadas em lógica nebulosa em simulações de computador, e após inúmeras tentativas uma estratégia foi definida. Os testes de campo, no entanto, indicaram que essa estratégia precisa ser ajustada a partir de um estudo mais profundo do desafio de se controlar a injeção de GNV. Algumas observações sobre a estratégia desenvolvida podem ser utilizadas em sua revisão: a utilização de uma métrica mais adequada de medição da eficiência da estratégia na simulação por computador e a utilização de conhecimentos prévios sobre o funcionamento do motor, entre outros. No entanto, o principal fator a ser considerado está associado à metodologia empregada, sugerindo-se a criação da base de regras com base na observação do comportamento do sistema em campo, no lugar da avaliação computacional, como foi feito neste trabalho.

Para trabalhos futuros, dois caminhos podem ser seguidos: utilização da plataforma desenvolvida para testes de novas estratégias de cálculo do tempo de injeção ou melhoria no projeto da plataforma desenvolvida:

Utilização da plataforma desenvolvida para testes de cálculo do tempo de injeção:

- *Criação da base de regras com base em observação de campo*: Aproveitando-se da experiência desenvolvida nesta dissertação, sugere-se a criação de uma nova base de regras com base na experimentação em campo. Uma outra abordagem é a utilização de um dinamômetro de rolo para a reprodução em laboratório das diversas condições de funcionamento do motor.
- *Criação de uma estratégia de funcionamento baseada em conhecimentos do sistema*: Diferentemente do que foi proposto nesta dissertação, uma nova abordagem seria a definição da correção sobre o tempo de injeção com base em conhecimentos sobre o funcionamento do motor e de seus sensores.
- **Melhoria no projeto da plataforma desenvolvida:**
  - *Substituição do microcontrolador*: embora o microcontrolador utilizado tenha atendido satisfatoriamente os objetivos deste projeto de mestrado, existem atualmente opções baseadas em núcleo ARM® mais baratas e eficientes, com ferramentas de desenvolvimento de baixo custo. Isso possibilitaria a criação de estratégias de funcionamento mais complexas, com o emprego de operações matemáticas mais elaboradas.
  - *Integração das placas de circuito digital e analógico e de potência*: como estratégia de desenvolvimento, a separação dos circuitos digitais e analógicos dos circuitos de potência trouxe várias vantagens (descritas na seção 4.1), porém essa separação não é adequada caso o sistema desenvolvido se torne um produto comercial.
  - *Relayout da(s) placa(s) desenvolvidas*: as placas desenvolvidas sofreram vários retrabalhos para incorporar novos circuitos não previstos inicialmente. Adicionalmente, os conectores empregados foram selecionados com o objetivo de facilitar testes rápidos, mas não são adequados para um produto comercial.

## 9 REFERÊNCIAS BIBLIOGRÁFICAS

[1] Raio X do Setor – IBP (acesso em 08/05/2011).

<http://www.ibp.org.br/services/DocumentManagement/FileDownload.EZTSvc.asp?DocumentID={BB336FA2-3E16-4CEE-A440-CC25EE5D66D3}&ServiceInstUID={008234CC-5DAA-4DB6-8D68-111B3FD798D3}>

[2] Brasil Econômico – edição de 06 de Abril de 2011 (acesso em 19/05/2011).

[http://www.brasileconomico.com.br/epaper/contents/BE\\_2011-04-06.pdf](http://www.brasileconomico.com.br/epaper/contents/BE_2011-04-06.pdf)

[3] Exército ocupa campos de petróleo e gás na Bolívia – Folha Online – 01 de Maio de 2006 (acesso em 20/05/2011).

<http://www1.folha.uol.com.br/folha/mundo/ult94u95485.shtml>

[4] Síntese de preços – Brasil – Abril de 2011 – ANP (acesso em 08/05/2011).

[http://www.anp.gov.br/preco/prc/Resumo\\_Mensal\\_Index.asp](http://www.anp.gov.br/preco/prc/Resumo_Mensal_Index.asp)

[5] Simulador de Economia de GNV – SCGAS (acesso em 08/05/2011).

<http://www.scgas.com.br/index.php/site/veicular/conteudos/simulador/idse/255>

[6] Site da *National Instruments* (acesso em 20/05/2011).

<http://www.ni.com/>

[7] Carvalho, T.C.O. Desenvolvimento de uma Central Eletrônica aberta para Controle de Motores de Combustão Interna. Belo Horizonte, MG, UFMG, 2005.

[8] Ramos, D.B., CONTROLADOR NEBULOSO PARA MOTOR DE IGNIÇÃO POR COMPRESSÃO OPERANDO COM GÁS NATURAL E ÓLEO DIESEL. Santa Maria, RS, UFSM, 2006.

[9] Veiga, M.R., DESENVOLVIMENTO DE UM GERENCIADOR ELETRÔNICO PARA MOTORES TRICOMBUSTÍVEL. São Paulo, SP, Poli/USP, 2010.

[10] Oliveira, F. B. ; Vandresen, M. ; Pereira, M. Desenvolvimento de Kit de Conversão De Motores Diesel Estacionários de Geradores Para Operar Com Mistura Diesel - Gás Natural. In: 9º Congresso Iberoamericano de Ingeniería Mecánica. Las Palmas de Gran Canaria : FEIBIM, 2009. v. 01. p. 173-180. IF-SC, 2009.

[11] Barghi, F., Safavi A.A. An Intelligent Control Policy for Fuel Injection Control of SI Engines (Case Study: CNG Engine). In: 15th International Conference on Intelligent Engineering Systems: INES, 2011. June 23-25, 2011.

[12] Santarém, C.A., Baioco, J., Bone, R.B., Filho, V.F. CUSTOS E BENEFÍCIOS ECONÔMICOS DE TECNOLOGIAS DE TRANSPORTE DE GÁS NATURAL NO BRASIL. In: 4º PDPETRO, Campinas, SP: ABPG 2007. 7.2.0119-1- 1

[13] Engerer, H., Horn, M. Natural gas vehicles: An option for Europe. In: Energy Policy, 2010. Ed. 38. p. 1017-1029.

[14] Janssen, A., Lienin, S., Gassmann, F., Wokaun, A. Model aided policy development for the market penetration of natural gas vehicles in Switzerland. In: Transportation Research, 2006. Part A 40. p. 316-333

[15] Mello, P., Pelliza, G., Cataluña, R., da Silva, R. Evaluation of the maximum horsepower of vehicles converted for use with natural gas fuel. In: Fuel, 2006. Ed. 85. p. 2180-2186

[16] Collantes, G., Melaina, M. W. The co-evolution of alternative fuel infrastructure and vehicles: A study of the experience of Argentina with compressed natural gas. In: Energy Policy, 2011. Ed. 39. p. 664-675

[17] Flynn, P.C. Commercializing an alternate vehicle fuel: lessons learned from natural gas for vehicles. Energy Policy, 2002. Ed. 30. p. 613-619

[18] Burger, P.R.B., Baeta, J.G.C., Valle, R.M. Desempenho de um Motor Multicombustível Operando Simultaneamente com GNV e Álcool Etílico em Diferentes Proporções

[19] Di Pascoli, S., Femia, A., Luzzati, T. Natural gas, cars and the environment. A (relatively) ‘clean’ and cheap fuel looking for users. In: Economical Ecologics, 2001. Ed. 38, p. 179-189

[20] Lino, P., Maione, B., Amorese, C., Modelling and predictive control of a new injection system for compressed natural gas engines. In: Control Engineering Practice, 2008. Ed. 16, p. 1216-1230

- [21] Kalam, M.A., Masjuki, H.H. An experimental investigation of high performance natural gas engine with direct injection. In: *Energy*, 2011. Ed. 36, p. 3563-3571
- [22] Carlucci, A.P., de Risi, A., Lagorfia, D., Naccarato, F. An experimental investigation of high performance natural gas engine with direct injection. In: *Energy*, 2008. Ed. 33, p. 256-263.
- [23] Aslam, M.U., Masjuki, H.H., Kalam, M.A., Abdesselam, H., Mahlia, T.M.I., Amalina, M.A. An experimental investigation of CNG as an alternative fuel for a retrofitted gasoline vehicle. In: *Fuel*, 2006. Ed. 85, p. 717-724.
- [24] Thamsiroj, T., Smyth, H., Murphy, J.D. A roadmap for the introduction of gaseous transport fuel: A case study for renewable natural gas in Ireland. In: *Renewable and Sustainable Energy Reviews*, 2011. Ed. 15, p. 4642-4651.
- [25] Huang, S.R., Lin, C.Y., Wu, C.Y., Wu, C.C., Chu, C.Y., Chen, H.T., Wang, S.H. Fuzzy control of a bio-hydrogen internal combustion engine generating system. In: *international journal of hydrogen energy*, 2011. Ed. 36, p. 8774-8778.
- [26] Kheir, N.A., Salman, M.A., Schouten, N.J. Emissions and fuel economy trade-off for hybrid vehicles using fuzzy logic. In: *Mathematics and Computers in Simulation*, 2004. Ed. 66, p. 155-172
- [27] Kalogirou, S.A., Artificial intelligence for the modeling and control of combustion processes: a review. *Progress in Energy and Combustion Science*, 2003. Ed. 29, p. 515-566.
- [28] MOTOR DE CICLO OTTO. (acesso em 08 de maio de 2011).

[http://www.castrol.com/liveassets/bp\\_internet/castrol/castrol\\_brazil/STAGING/local\\_assets/images/ask\\_the\\_expert/curso\\_desempenho001\\_br\\_375x588.jpg](http://www.castrol.com/liveassets/bp_internet/castrol/castrol_brazil/STAGING/local_assets/images/ask_the_expert/curso_desempenho001_br_375x588.jpg)

- [29] VON GLEHN, F. R., Curso Injeção Eletrônica - Conceitos Teóricos e Aspectos Práticos, Goiânia - GO, Ciclo Engenharia, junho de 2001, 4ª edição - volume 1 de 5.
- [30] Doebelin, E. O. *Measurement Systems, Application and Design*. MacGraw-Hill, 4ª edição, 1990.
- [31] Envenenado – Como Funciona – Injeção Eletrônica. (acesso em 10 de maio de 2011).

<http://www.envenenado.com.br/howwork/injecao/injecao.html>

- [32] Injeção Eletrônica e Mecânica - PARTE 13. (acesso em 18 de junho de 2011).

[http://www.fazerfacil.com.br/carros/injecao\\_13.htm](http://www.fazerfacil.com.br/carros/injecao_13.htm)

- [33] Como funciona o sistema de ignição de um automóvel (acesso em 18/06/2011).

<http://carros.hsw.uol.com.br/sistema-de-ignicao-de-um-automovel.htm>

- [34] Stoichiometry – Wikipédia. (acesso em 10 de maio de 2011).

<http://en.wikipedia.org/wiki/Stoichiometry>

- [35] Monóxido de Carbono – Wikipédia. (acesso em 10 de maio de 2011).

[http://pt.wikipedia.org/wiki/Mon%C3%B3xido\\_de\\_carbono](http://pt.wikipedia.org/wiki/Mon%C3%B3xido_de_carbono)

- [36] Prêmio Petrobrás de Tecnologia - Utilização de Gás Natural em Motores e Cenário do Uso no Brasil – (acesso em 14 de maio de 2011).

[http://www.petrobras.com.br/minisite/premiotecnologia/pdf/TecnologiaGas\\_GasNatural\\_Motores.pdf](http://www.petrobras.com.br/minisite/premiotecnologia/pdf/TecnologiaGas_GasNatural_Motores.pdf)

- [37] GasNet – O Site do Gás Natural – GNV. GNV – Gás Natural Veicular, Entendendo o GNV (acesso em 10 de maio de 2011).

[http://www.gasnet.com.br/novo\\_gnv/entendendo\\_gnv.asp](http://www.gasnet.com.br/novo_gnv/entendendo_gnv.asp)

- [38] Catedral do Gás – Histórico do Gás Natural no Brasil (acesso em 12 de maio de 2011).

<http://catedradogas.iee.usp.br/gasnatural/historicobrasil.htm>

- [39] Melo Filho, M. A história do gás natural: do Rio de Janeiro para o Brasil. Rio de Janeiro, 2005.

- [40] DECRETO Nº 3.371, DE 24 DE FEVEREIRO DE 2000 – Aneel (acesso em 14 de maio de 2011).

<http://www.aneel.gov.br/cedoc/dec20003371.pdf>

- [41] Empresa de Pesquisa Energética. Balanço Energético Nacional 2009. Rio de Janeiro, EPE: 2010.

- [42] *International Energy Agency, Key World Energy STATISTC*, 2011. França, IEA, 2011.

- [43] Empresa de Pesquisa Energética, Balanço Energético Nacional, 2011. Rio de Janeiro, EPE: 2011.

- [44] Empresa de Pesquisa Energética. Plano Nacional de Energia 2030. Rio de Janeiro, EPE: 2007.

- [45] Kit Sequent my07  
<http://www.gaspoint.com.br/gnv/kits/sequent-my07/kit-sequent-my07.html>
- [46] Globo Gás Brasil :: Revista Especializa em gás - ::Combustível original e GNV: Perda de potência tem solução técnica:: (acesso em 17 de agosto de 2011).  
<http://www.globogasbrasil.com.br/artigos/combustivel-original-e-gnv-perda-de-potencia-tem-solucao-tecnica/>
- [47] Serviço Brasileiro de Respostas Técnicas (BRT) – Dossiê Técnico Gás Natural Veicular (GNV) (acesso em 17 de agosto de 2011)  
<http://www.sbrt.ibict.br/dossie-tecnico/downloadsDT/NjU=>
- [48] GasNet – O Site do Gás Natural – GNV. Gás Natural - Vantagens (acesso em 14 de maio de 2011).  
[http://www.gasnet.com.br/novo\\_gasnatural/vantagens.asp](http://www.gasnet.com.br/novo_gasnatural/vantagens.asp)
- [49] Pelliza, G., Análise de Veículos Convertidos para o uso de Gás Natural, Porto Alegre – RS, 3 de fevereiro de 2003, Seção 6.3, pág. 42.
- [50] ROMARIZ, A., Apostila de Sistemas Inteligentes, pág. 40
- [51] Zadeh, L.A. (1965). “Fuzzy Sets.” In: Information and Control 8, University of California, USA, pág. 338 à 353.
- [52] Mamdani, EH., Assilian, S. (1975). “An Experiment in Linguistics Synthesis With a Fuzzy Logic Controller.” In: International Journal Man-Machine Studies, Vol 7-(1), pág. 1 à 13
- [53] Mamdani, EH. (1977). “Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis.” In: IEEE Transaction on Computers, Vol C-26, pág. 1182 à 1191.
- [54] Tsukamoto, Y. (1979). “An Approach to Fuzzy Reasoning Method.” In: Advances in Fuzzy Set Theory and Applications, Holanda, pág. 137 à 149.
- [55] Takagi, T., Sugeno, M. (1985). “Fuzzy Identification of Systems and its Applications to Modeling and Control.” In: IEEE Trans. Sys. Man. Cybern., Vol 15, No1, pág. 116 à 132.
- [56] Shaw, I.S., Godoy, M. (1999). “Controle e Modelagem Fuzzy.” In: Ed. Edgar Blücher LTDA, FAPESP, São Paulo.

# Anexo I      Protótipos das Funções e Macros

## Desenvolvidas

Esse anexo traz os protótipos das funções desenvolvidas. O código não é apresentado para que o texto desta dissertação não fique excessivamente extenso.

Para todos os casos, o arquivo que contém a palavra MACRO em seu nome traz versões em macro das funções criadas. Macros são trechos de código referenciados por um protótipo, que quando são chamados são inseridos no código no lugar da chamada. Isso significa que quando uma macro é utilizada no código, o seu conteúdo é inserido no lugar da sua chamada na fase de pré-compilação. Essa estratégia aumenta a utilização da memória de programa, mas muitas vezes é necessária porque sua utilização diminui o tempo de processamento, já que evita o salvamento e recuperação de contexto que uma função exige.

## 1      MCUDrivers

A biblioteca MCUDrivers traz todas as funções que dizem respeito aos periféricos do microcontrolador, e será específica para cada microcontrolador. As os protótipos das funções da biblioteca MCUDrivers devem ser mantidos em caso de criação de uma nova biblioteca para um novo microcontrolador para que dessa forma, a aplicação não sofra alterações.

Essa camada foi dividida nos seguintes arquivos:

- ADC.c, ADCMACROS.h e ADC.h – arquivos com as funções de manipulação do conversor analógico/digital;
- I2C.c, I2CMACROS.h e I2C.h – arquivos com as funções de utilização da interface de comunicação I2C;
- IO.c, IOMACROS.h e IO.h – arquivos com as funções de manipulação das I/Os (*Inputs* e *Outputs*);
- TIMER0.c, TIMER0MACROS.h e TIMER0.h – arquivos com as funções de manipulação do Timer 0 do microcontrolador;
- TIMER1.c, TIMER1MACROS.h e TIMER1.h – arquivos com as funções de manipulação do Timer 1 do microcontrolador;
- TIMER2.c, TIMER2MACROS.h e TIMER2.h – arquivos com as funções de manipulação do Timer 2 do microcontrolador;
- UART.c, UARTMACROS.h e UART.h – arquivos com as funções de utilização da interface de comunicação serial (UART);

Os arquivos .c contém o código das funções e é específico do microcontrolador utilizado (ATMega32-16PU), enquanto o arquivo .h contém os protótipos dessas funções.

### 1.1      ADC.h

As funções presentes nesse arquivo são:

- // Habilita o ADC  
void ADCEn(void);
- // Desabilita o ADC  
void ADCDis(void);
- // Inicia uma nova conversão  
void ADCStartConversion(void);
- // Indica se existe uma conversão em progresso  
uint8\_t ADCCConversionInProgress(void);
- // Seleciona o canal de entrada

- ```
void ADCSelectCh(ADC_CH adCH);
```
- // Seleciona o modo de operação  
void ADCSelectMode(ADC\_MODE adcMode);
  - // Seleciona a precisão do canal  
void ADCSelectPrecision(ADC\_PRECISION adcPrecision);
  - // Seleciona a referência para conversão  
void ADCSelectReference(ADC\_REF adcVref);
  - // Seleciona o clock do ADC  
void ADCSelectClock(ADC\_CLOCK adcClock);
  - // Le o resultado da conversão em 8 bits  
uint8\_t ADCRead8Bits(void);
  - // Le o resultado da conversão em 10 bits  
uint16\_t ADCRead10Bits(void);

Os tipos de dados criados para a execução dessas funções são:

- typedef enum {ADC\_SINGLE\_CONVERSION=0, ADC\_FREE\_RUNNING, ADC\_T0\_CO, ADC\_T0\_OV, ADC\_T1\_CO\_B, ADC\_T1\_OV} **ADC\_MODE**;
- typedef enum {ADC\_8\_BITS=0, ADC\_10\_BITS} **ADC\_PRECISION**;
- typedef enum {ADC\_AREF=0, ADC\_AVCC, ADC\_INT\_2V56} **ADC\_REF**;
- typedef enum {ADC\_0=0, ADC\_1, ADC\_2, ADC\_3, ADC\_4, ADC\_5, ADC\_6, ADC\_7} **ADC\_CH**;
- typedef enum {ADC\_INT\_DIV\_2=0, ADC\_INT\_DIV\_4, ADC\_INT\_DIV\_8, ADC\_INT\_DIV\_16, ADC\_INT\_DIV\_32, ADC\_INT\_DIV\_64, ADC\_INT\_DIV\_128} **ADC\_CLOCK**;
- typedef enum {ADC\_INT\_DIS=0, ADC\_INT\_EN} **ADC\_INT**;

## 1.2 I2C.h

As funções presentes nesse arquivo são:

- // Inicializa o módulo I2C  
void I2CInit(void);
- // Gera uma condição de Start  
void I2CGenStart(I2C\_STATUS \*I2CStatus\_p);
- // Gera uma condição de Stop  
void I2CGenStop(I2C\_STATUS \*I2CStatus\_p);
- // Envia um byte  
void I2CWriteByte(uint8\_t byte, I2C\_STATUS \*I2CStatus\_p);
- // Recebe um byte  
uint8\_t I2CReadByte(void);
- // Configura o módulo I2C para gerar ou não um sinal de ACK  
void I2CSetRxAck(I2C\_ACKNOWLEDGE ack, I2C\_STATUS \*I2CStatus\_p);
- // Essa função deve ser executada continuamente para que as ações sejam efetivadas  
void I2CTask(void);

Os tipos de dados criados para a execução dessas funções são:

- // Indica uma condição de geração ou não de ACK

```
typedef enum {I2C_ACK=0,I2C_NACK} I2C_ACKNOWLEDGE;
```

- // Armazena o estado do módulo I2C

```
typedef union  
{  
    uint8_t I2CFlags;  
    struct  
    {  
        uint8_t I2CStartOk : 1;  
        uint8_t I2CStopOk : 1;  
        uint8_t I2CBusy : 1;  
        uint8_t I2CTxOk : 1;  
        uint8_t I2CTXAck : 1;  
        uint8_t I2CRxOk : 1;  
        uint8_t I2CRxAck : 1;  
        uint8_t rsvd : 1;};  
    } I2C_STATUS;
```

### 1.3 IO.h

As funções presentes nesse arquivo são:

- // Habilita ou desabilita o uso de Pull Ups.  
void IOPullUpUse(IO\_PU\_STATUS PUUseStatus);
- // Estabelece o estado do PULL UP de um determinado pino de uma porta.  
// Retorna 1 em caso de sucesso e 0 caso não seja possível realizar a operação.  
uint8\_t IOSetPinPU(IO\_PU\_STATUS IOPU,IO\_PORT IOPort, uint8\_t IOPin);
- // Estabelece o estado do PULL UP de todos os pinos de uma porta. Cada bit de mask corresponde a um pino da porta.  
// Cada bit do valor retornado é igual a 1 em caso de sucesso e 0 caso não seja possível realizar a operação.  
uint8\_t IOSetPortPU(IO\_PORT IOPort, uint8\_t mask);
- // Estabelece a direção de um determinado pino de uma porta.  
void IOSetPinDirection(IO\_DIRECTION IODir,IO\_PORT IOPort, uint8\_t IOPin);
- // Estabelece a direção de todos os pinos de uma porta. Cada bit de mask corresponde a um pino da porta.  
void IOSetPortDirection(IO\_PORT IOPort, uint8\_t mask);
- // Lê a direção de um determinado pino de uma porta.  
IO\_DIRECTION IOGetPinDirection(IO\_PORT IOPort, uint8\_t IOPin);
- // Lê a direção de todos os pinos de uma porta. Cada bit do valor retornado corresponde a um pino da porta.  
uint8\_t IOGetPortDirection(IO\_PORT IOPort);
- // Estabelece a saída de um determinado pino de uma porta.  
void IOSetPinOutput(IO\_STATUS IOOut,IO\_PORT IOPort, uint8\_t IOPin);
- // Estabelece a saída de todos os pinos de uma porta. Cada bit de mask corresponde a um pino da porta.  
void IOSetPortOutput(IO\_PORT IOPort, uint8\_t mask);
- // Lê o estado de um determinado pino de uma porta.



```
IO_STATUS IOGetPinStatus(IO_PORT IOPort, uint8_t IOPin);
```

- // Lê o estado de todos os pinos de uma porta. Cada bit do valor retornado corresponde a um pino da porta.  
uint8\_t IOGetPortStatus(IO\_PORT IOPort);

Os tipos de dados criados para a execução dessas funções são:

- // Indica o estatus dos Pull Ups  
typedef enum {IO\_PU\_DIS=0,IO\_PU\_EN} IO\_PU\_STATUS;
- // Indica a direção da IO  
typedef enum {IO\_DIR\_IN=0,IO\_DIR\_OUT} IO\_DIRECTION;
- // Define as portas do microcontrolador  
typedef enum {IO\_PORTA=0,IO\_PORTB,IO\_PORTC,IO\_PORTD} IO\_PORT;
- // Indica os estados possíveis de uma IO  
typedef enum {IO\_LOW=0,IO\_HIGH=1} IO\_STATUS;

## 1.4 TIMER0.h

As funções presentes nesse arquivo são:

- // Inicializa o TIMER  
void T0Init(T0\_MODE t0Mode, T0\_PIN\_OUT t0PinOut, T0\_CLOCK t0Clock,T0\_OV\_INT t0OvInt,void (\*ovfCallBack\_p)(void), T0\_OC\_INT t0OcInt,void (\*compCallBack\_p)(void));
- // Seta a fonte do clock do timer  
void T0SetClock(T0\_CLOCK t0Clock);
- // Seta a opção de pinagem de saída do timer  
void T0SetPinOut(T0\_PIN\_OUT t0PinOut);
- // Seta o valor de comparação do Timer  
// Essa função só deve ser usada quando nos modos T0\_PWM\_PHASE\_CORRECT e T0\_PWM  
void T0SetCompare(uint8\_t t0Compare);
- // Seta o valor de TOP do Timer  
// Essa função só deve ser usada quando no modo T0\_CLEAR\_ON\_COMPARE  
void T0SetTOP(uint8\_t t0TOP);
- // Le o valor do timer  
uint8\_t T0ReadTimer(void);

Os tipos de dados criados para a execução dessas funções são:

- // Define os modos de operação do TIMER0  
typedef enum {T0\_NORMAL=0, T0\_CLEAR\_ON\_COMPARE, T0\_PWM\_PHASE\_CORRECT, T0\_PWM} T0\_MODE;
- // Define a configuração de saída do compare do TIMER0  
typedef enum {T0\_NO\_PIN\_OUT=0, T0\_TOGGLE, T0\_CLEAR, T0\_SET, T0\_PWM\_NORM, T0\_PWM\_INV} T0\_PIN\_OUT;
- // Define as fontes de clock do TIEMR0  
typedef enum {T0\_DIS=0, T0\_INT, T0\_INT\_DIV\_8, T0\_INT\_DIV\_64, T0\_INT\_DIV\_256, T0\_INT\_DIV\_1024, T0\_EXT\_FALLING\_EDGE, T0\_EXT\_RISING\_EDGE} T0\_CLOCK;
- // Define se a interrupção por overflow está ou não habilitada  
typedef enum {T0\_OV\_INT\_DIS=0,T0\_OV\_INT\_EN} T0\_OV\_INT;

- // Define se a interrupção por compare está ou não habilitada  
typedef enum {T0\_OC\_INT\_DIS=0,T0\_OC\_INT\_EN} T0\_OC\_INT;

## 1.5 TIMER1.h

As funções presentes nesse arquivo são:

- // Inicializa o TIMER  
// A opção T1\_TOGGLE não deve ser usada para a porta B.  
void T1Init(T1\_MODE t1Mode, T1\_PIN\_OUT t1APinOut, T1\_PIN\_OUT t1BPinOut, T1\_CLOCK t1Clock, T1\_OV\_INT t1OvInt, void (\*ovfCallBack\_p)(void), T1\_OC\_A\_INT t1OcAInt, void (\*compACallBack\_p)(void), T1\_OC\_B\_INT t1OcBInt, void (\*compBCallBack\_p)(void));
- // Seta a fonte do clock do timer  
void T1SetClock(T1\_CLOCK t1Clock);
- // Seta a opção de pinagem de saída do timer  
void T1SetPinOut(T1\_PIN\_OUT t1APinOut, T1\_PIN\_OUT t1BPinOut);
- // Seta o valor de comparação do Timer  
void T1SetCompare(TI\_OC\_OPT T1OCOpt, uint16\_t t1Compare);
- // Seta o valor de TOP do Timer  
void T1SetTOP(uint16\_t t1TOP);
- // Le o valor do timer  
uint16\_t T1ReadTimer(void);

Os tipos de dados criados para a execução dessas funções são:

- // Define os modos de operação do TIMER1  
typedef enum {T1\_NORMAL=0, T1\_CLEAR\_ON\_COMPARE, T1\_PWM\_PHASE\_CORRECT, T1\_PWM} T1\_MODE;
- // Define a configuração de saída do compare do TIMER1  
typedef enum {T1\_NO\_PIN\_OUT=0, T1\_TOGGLE, T1\_CLEAR, T1\_SET, T1\_PWM\_NORM, T1\_PWM\_INV} T1\_PIN\_OUT;
- // Define as fontes de clock do TIEMR1  
typedef enum {T1\_DIS=0, T1\_INT, T1\_INT\_DIV\_8, T1\_INT\_DIV\_64, T1\_INT\_DIV\_256, T1\_INT\_DIV\_1024, T1\_EXT\_FALLING\_EDGE, T1\_EXT\_RISING\_EDGE} T1\_CLOCK;
- // Define se a interrupção por overflow está ou não habilitada  
typedef enum {T1\_OV\_INT\_DIS=0,T1\_OV\_INT\_EN} T1\_OV\_INT;
- // Define se a interrupção por compare do canal A está ou não habilitada  
typedef enum {T1\_OC\_A\_INT\_DIS=0,T1\_OC\_A\_INT\_EN} T1\_OC\_A\_INT;
- // Define se a interrupção por compare do canal B está ou não habilitada  
typedef enum {T1\_OC\_B\_INT\_DIS=0,T1\_OC\_B\_INT\_EN} T1\_OC\_B\_INT;
- // Define qual dos canais de compare deve ser tratado  
typedef enum {T1\_OC\_A=0, T1\_OC\_B} TI\_OC\_OPT;

## 1.6 TIMER2.h

As funções presentes nesse arquivo são:

- // Inicializa o TIMER

```
void T2Init(T2_MODE t2Mode, T2_PIN_OUT t2PinOut, T2_CLOCK t2Clock, T2_OV_INT t2OvInt, void (*ovfCallBack_p)(void), T2_OC_INT t2OcInt, void (*compCallBack_p)(void));
```

- // Seta a fonte do clock do timer  
void T2SetClock(T2\_CLOCK t2Clock);
- // Seta a opção de pinagem de saída do timer  
void T2SetPinOut(T2\_PIN\_OUT t2PinOut);
- // Seta o valor de comparação do Timer  
// Essa função só deve ser usada quando nos modos T2\_PWM\_PHASE\_CORRECT e T2\_PWM  
void T2SetCompare(uint8\_t t2Compare);
- // Seta o valor de TOP do Timer  
// Essa função só deve ser usada quando no modo T2\_CLEAR\_ON\_COMPARE  
void T2SetTOP(uint8\_t t2TOP);
- // Le o valor do timer  
uint8\_t T2ReadTimer(void);

Os tipos de dados criados para a execução dessas funções são:

- // Define os modos de operação do TIMER2  
typedef enum {T2\_NORMAL=0, T2\_CLEAR\_ON\_COMPARE, T2\_PWM\_PHASE\_CORRECT, T2\_PWM} T2\_MODE;
- // Define a configuração de saída do compare do TIMER2  
typedef enum {T2\_NO\_PIN\_OUT=0, T2\_TOGGLE, T2\_CLEAR, T2\_SET, T2\_PWM\_NORM, T2\_PWM\_INV} T2\_PIN\_OUT;
- // Define as fontes de clock do TIEMR2  
typedef enum {T2\_DIS=0, T2\_INT, T2\_INT\_DIV\_8, T2\_INT\_DIV\_32, T2\_INT\_DIV\_64, T2\_INT\_DIV\_128, T2\_INT\_DIV\_256, T2\_INT\_DIV\_1024} T2\_CLOCK;
- // Define se a interrupção por overflow está ou não habilitada  
typedef enum {T2\_OV\_INT\_DIS=0, T2\_OV\_INT\_EN} T2\_OV\_INT;
- // Define se a interrupção por compare está ou não habilitada  
typedef enum {T2\_OC\_INT\_DIS=0, T2\_OC\_INT\_EN} T2\_OC\_INT;

## 1.7 UART.h

As funções presentes nesse arquivo são:

- // Inicializa a USART, e a habilita  
void UARTInit(uint32\_t baudRate, UART\_DATABITS dataBits, UART\_PARITY parity, UART\_STOPBITS stopBits, UART\_SYNC\_MODE syncAsyncMode, UART\_POLARITY polarity, UART\_RX\_INT rxIntEn, void (\*rxCallBack\_p)(void), UART\_TX\_INT txIntEn, void (\*txCallBack\_p)(void));
- // Habilita a USART  
void UARTEn(void);
- // Desabilita a USART  
void UARTDis(void);
- // Envia um byte via USART  
void UARTSendByte(uint8\_t byte);
- // Recebe um byte via USART e o escreve no ponteiro passado.

```
// Se houver um byte a ser recebido e ele for recebido normalmente, retorna 0, caso contrário retorna um código de erro.
```

```
UART_RX_ERRORS UARTReceiveByte(uint8_t *byte);
```

- // Limpa o buffer de recepção da USART.

```
void UARTRxFlush(void);
```

Os tipos de dados criados para a execução dessas funções são:

- // Define a quantidade de bits em cada palavra transmitida

```
typedef enum {DATABITS_5=0, DATABITS_6, DATABITS_7, DATABITS_8, DATABITS_9=7}  
UART_DATABITS;
```

- // Define os modos síncrono e assíncrono

```
typedef enum {ASYNC=0,SYNC} UART_SYNC_MODE;
```

- // Define os tipos de paridade

```
typedef enum {PARITY_DIS=0,PARITY_EVEN=2,PARITY_ODD=3} UART_PARITY;
```

- // Define a quantidade de stop bits

```
typedef enum {STOPBITS_1=0,STOPBITS_2} UART_STOPBITS;
```

- // Define a borda na qual o dado é amostrado

```
typedef enum {POL_FALLING=0,POL_RISING} UART_POLARITY;
```

- // Define os erros

```
typedef enum {ERR_NO_BYTE=1, ERR_FRAME, ERR_DATA_OVERRUN, ERR_PARITY}  
UART_RX_ERRORS;
```

- // Define a habilitação ou não de interrupção por envio de dados

```
typedef enum {DIS_TX_INT=0,EN_TX_INT} UART_TX_INT;
```

- // Define a habilitação ou não de interrupção por recebimento de dados

```
typedef enum {DIS_RX_INT=0,EN_RX_INT} UART_RX_INT;
```

## 2 Common

A biblioteca Common contém todas as funções, macros e definições que são comuns a todo o código, ou seja, que podem ser necessárias em qualquer outra camada. A criação dessa camada visa a não repetição de tarefas comuns de serem utilizadas em qualquer lugar do código, e tem também o objetivo de concentrar definições que serão comuns a todo o código.

Essa camada foi dividida nos seguintes arquivos:

- STRING.c e STRING.h – arquivos com as funções de manipulação de strings;
- CLOCK.h – arquivo com a definição de frequência de CLOCK;
- COMMONOP.h – arquivo com macros com operações comuns;

O arquivo STRING.c contém o código das funções de manipulação de strings, enquanto o arquivo STRING.h contém os protótipos dessas funções. O arquivo CLOCK.h contém apenas a definição da frequência interna do processador que será utilizada no restante do código para cálculo de temporizações e taxas de transmissão. O arquivo COMMONOP.h traz algumas macros que são frequentemente necessárias, como manipulação de bits, conversão de hexadecimal para decimal e repetição de operações.

### 2.1 STRING.h

As funções presentes nesse arquivo são:

- // Retorna o tamanho da String

```
uint8_t STRLen(char *string);
```

- // Compara os primeiros "size" caracteres de duas strings, até o limite de seu tamanho ('\0') e retorna 1 se forem iguais e 0 caso contrário.

```
uint8_t STRCmp(char *string1, char *string2, uint8_t size); // Habilita o ADC
```

## 2.2 CLOCK.h

Esse arquivo traz a seguinte definição:

- #define F\_CPU 20000000 // relógio com cristal externo de 20MHz;

O código deve ser feito utilizando essa definição de uma maneira tal que uma alteração da frequência de operação seja automaticamente refletida em todo o código, não alterando as taxas de transmissão e as temporizações por exemplo.

## 2.3 COMMONOP.h

As macros presentes nesse arquivo são:

- // Define uma máscara para o b-ésimo bit

```
#define BIT(b) (1<<b)
```

- // Seta o bit b1 do byte B

```
#define SET1b(B,b1) B |= BIT(b1)
```

- // Reseta o bit b1 do byte B

```
#define RESET1b(B,b1) B &= ~BIT(b1)
```

- // Seta os bits b1 e b2 do byte B

```
#define SET2b(B,b1,b2) B |= (BIT(b1) | BIT(b2))
```

- // Reseta os bits b1 e b2 do byte B

```
#define RESET2b(B,b1,b2) B &= ~(BIT(b1) | BIT(b2))
```

- // Seta os bits b1, b2 e b3 do byte B

```
#define SET3b(B,b1,b2,b3) B |= (BIT(b1) | BIT(b2) | BIT(b3))
```

- // Reseta os bits b1, b2 e b3 do byte B

```
#define RESET3b(B,b1,b2,b3) B &= ~(BIT(b1) | BIT(b2) | BIT(b3))
```

- // Seta os bits b1, b2, b3 e b4 do byte B

```
#define SET4b(B,b1,b2,b3,b4) B |= (BIT(b1) | BIT(b2) | BIT(b3) | BIT(b4))
```

- // Reseta os bits b1, b2, b3 e b4 do byte B

```
#define RESET4b(B,b1,b2,b3,b4) B &= ~(BIT(b1) | BIT(b2) | BIT(b3) | BIT(b4))
```

- // Seta os bits b1, b2, b3, b4 e b5 do byte B

```
#define SET5b(B,b1,b2,b3,b4,b5) B |= (BIT(b1) | BIT(b2) | BIT(b3) | BIT(b4) | BIT(b5))
```

- // Reseta os bits b1, b2, b3, b4 e b5 do byte B

```
#define RESET5b(B,b1,b2,b3,b4,b5) B &= ~(BIT(b1) | BIT(b2) | BIT(b3) | BIT(b4) | BIT(b5))
```

- // Seta os bits b1, b2, b3, b4, b5 e b6 do byte B  
#define SET6b(B,b1,b2,b3,b4,b5,b6) B |= (BIT(b1) | BIT(b2) | BIT(b3) | BIT(b4) | BIT(b5) | BIT(b6))
- // Reseta os bits b1, b2, b3, b4, b5 e b6 do byte B  
#define RESET6b(B,b1,b2,b3,b4,b5,b6) B &= ~(BIT(b1) | BIT(b2) | BIT(b3) | BIT(b4) | BIT(b5) | BIT(b6))
- // Seta os bits b1, b2, b3, b4, b5, b6 e b7 do byte B  
#define SET7b(B,b1,b2,b3,b4,b5,b6,b7) B |= (BIT(b1) | BIT(b2) | BIT(b3) | BIT(b4) | BIT(b5) | BIT(b6) | BIT(b7))
- // Reseta os bits b1, b2, b3, b4, b5, b6 e b7 do byte B  
#define RESET7b(B,b1,b2,b3,b4,b5,b6,b7) B &= ~(BIT(b1) | BIT(b2) | BIT(b3) | BIT(b4) | BIT(b5) | BIT(b6) | BIT(b7))
- // Seta os bits b1, b2, b3, b4, b5, b6, b7 e b8 do byte B  
#define SET8b(B,b1,b2,b3,b4,b5,b6,b7,b8) B |= (BIT(b1) | BIT(b2) | BIT(b3) | BIT(b4) | BIT(b5) | BIT(b6) | BIT(b7) | BIT(b8))
- // Reseta os bits b1, b2, b3, b4, b5, b6, b7 e b8 do byte B  
#define RESET8b(B,b1,b2,b3,b4,b5,b6,b7,b8) B &= ~(BIT(b1) | BIT(b2) | BIT(b3) | BIT(b4) | BIT(b5) | BIT(b6) | BIT(b7) | BIT(b8))
- // Converte o hexadecimal de 8 bits definido pelos caracteres 'H1' e 'H2' em um decimal e o atribui a 'D'  
#define HEX2DEC\_8(D,H1,H2) if(H1>=65) D = (H1-55)<<4; else D = (H1-48)<<4;\n if(H2>=65) D += (H2-55); else D += (H2-48);
- // Converte o hexadecimal de 16 bits definido pelos caracteres 'H1', 'H2', 'H3' e 'H4' em um decimal e o atribui a 'D'  
#define HEX2DEC\_16(D,H1,H2,H3,H4) if(H1>=65) D = (H1-55)<<12; else D = (H1-48)<<12;\n if(H2>=65) D = D+((H2-55)<<8); else D = D+((H2-48)<<8);\n if(H3>=65) D = D+((H3-55)<<4); else D = D+((H3-48)<<4);\n if(H4>=65) D += (H4-55); else D += (H4-48);
- // Repete a operação OP 2 vezes  
#define REPEAT\_OP\_2(OP) OP; OP
- // Repete a operação OP 3 vezes  
#define REPEAT\_OP\_3(OP) REPEAT\_OP\_2(OP); OP
- // Repete a operação OP 4 vezes  
#define REPEAT\_OP\_4(OP) REPEAT\_OP\_3(OP); OP
- // Repete a operação OP 5 vezes  
#define REPEAT\_OP\_5(OP) REPEAT\_OP\_4(OP); OP
- // Repete a operação OP 6 vezes  
#define REPEAT\_OP\_6(OP) REPEAT\_OP\_5(OP); OP
- // Repete a operação OP 7 vezes

- ```
#define REPEAT_OP_7(OP) REPEAT_OP_6(OP); OP
```
- // Repete a operação OP 8 vezes

```
#define REPEAT_OP_8(OP) REPEAT_OP_7(OP); OP
```

  - // Repete a operação OP 9 vezes

```
#define REPEAT_OP_9(OP) REPEAT_OP_8(OP); OP
```

  - // Repete a operação OP 10 vezes

```
#define REPEAT_OP_10(OP) REPEAT_OP_9(OP); OP
```

  - // Repete a operação OP 11 vezes

```
#define REPEAT_OP_11(OP) REPEAT_OP_10(OP); OP
```

  - // Repete a operação OP 12 vezes

```
#define REPEAT_OP_12(OP) REPEAT_OP_11(OP); OP
```

  - // Repete a operação OP 13 vezes

```
#define REPEAT_OP_13(OP) REPEAT_OP_12(OP); OP
```

  - // Repete a operação OP 14 vezes

```
#define REPEAT_OP_14(OP) REPEAT_OP_13(OP); OP
```

  - // Repete a operação OP 15 vezes

```
#define REPEAT_OP_15(OP) REPEAT_OP_14(OP); OP
```

  - // Repete a operação OP 16 vezes

```
#define REPEAT_OP_16(OP) REPEAT_OP_15(OP); OP
```

## 3 DEVDivers

A biblioteca DEVDivers contém as funções e macros de acesso aos dispositivos, virtuais ou não. Por exemplo, pode-se criar um relógio virtual que indique a hora atual para o sistema e que se utilize de acesso aos *timers* do microcontrolador para isso. Esse relógio seria um dispositivo virtual, uma vez que não haveria de fato um CI externo responsável por essa função.

Essa camada se utiliza da camada MCUDivers para acessar os periféricos e as *I/Os* do MCU. Sua função é tornar transparente à aplicação a manipulação de dispositivos externos, criando funções que terão uma utilização em alto nível como enviar strings pelo console, acionar bicos ou escrever dados na EEPROM.

Essa camada foi dividida nos seguintes arquivos:

- BICOS.c e BICOS.h – arquivos com as funções de interface com os bicos injetores (tanto originais quanto de gasolina);
- CONSOLE.c e CONSOLE.h – arquivos com funções de acesso ao console, a ser utilizado para configuração e depuração do código;
- EEPROM.c e EEPROM.h – arquivos com funções de acesso à EEPROM;

Os arquivos BICOS.c, CONSOLE.c e EEPROM.c contém os códigos das funções desenvolvidas, enquanto os arquivos BICOS.h, CONSOLE.h e EEPROM.h contém os protótipos dessas funções e as definições de tipos referentes a esses arquivos (que definem as interfaces com a camada DEVDivers).

### 3.1 BICOS.h

As funções presentes nesse arquivo são:

- // Inicializa o driver dos Bicos
- void BInit(void);

- // Estabelece o Tempo de Injeção do bico "BICO"
- void BSetITCorr(uint8\_t ITCorr\_);
- // Comuta os bicos para o combustível original ou para o combustível alternativo
- void BCommute(B\_STATE bState);

Os tipos de dados criados para a execução dessas funções são:

- // Define qual combustível está em uso: o original ou o alternativo (GNV)  
typedef enum {B\_ORIGINAL = 0,B\_ALTERNATIVE} **B\_STATE**;

## 3.2 **CONSOLE.h**

As funções presentes nesse arquivo são:

- // Inicia a serial para ser usada pelo console  
void CONSOLEInit(CONS\_ECHO\_STATE consEchoState,uint32\_t baudRate);
- // Inicia a serial para ser usada pelo console  
void CONSOLESetEcho(CONS\_ECHO\_STATE consEchoState);
- // Habilita ou desabilita um bloco a enviar dados pelo console  
void CONSOLESetBlock(CONS\_BLOCKS block, uint8\_t enable);
- // Envia uma string de até 128 caracteres.  
void CONSOLESendString(char \*str, CONS\_BLOCKS block);
- // Envia um inteiro sem sinal  
void CONSOLESendUint\_8(uint8\_t var, CONS\_BLOCKS block);
- // Envia um inteiro sem sinal de 16 bits  
void CONSOLESendUint\_16 (uint16\_t var, CONS\_BLOCKS block);
- // Envia um inteiro com sinal de 16 bits  
void CONSOLESendInt\_16(int16\_t var, CONS\_BLOCKS block);
- // Envia um inteiro de 8 bits em hexadecimal  
void CONSOLESendHex\_8(uint8\_t var, CONS\_BLOCKS block);
- // Envia um inteiro de 16 bits em hexadecimal  
void CONSOLESendHex\_16(uint16\_t var, CONS\_BLOCKS block);
- // Envia um byte pela serial  
void CONSOLESendByte\_8(uint8\_t var, CONS\_BLOCKS block);
- // Recebe uma string de até byteCount caracteres.  
void CONSOLEGetString(char \*string, uint8\_t byteCount);
- // Recebe um inteiro  
uint8\_t CONSOLEGetInt\_8(void);
- // Recebe um inteiro de 16 bits  
uint16\_t CONSOLEGetInt\_16(void);
- // Recebe um inteiro de 8 bits em hexadecimal  
uint8\_t CONSOLEGetHex\_8 (void);
- // Recebe um inteiro de 16 bits em hexadecimal  
uint16\_t CONSOLEGetHex\_16(void);



- // Essa função deve ser executada para que as ações seja efetivadas  
void CONSOLETask(void);

Os tipos de dados criados para a execução dessas funções são:

- // Enumera os blocos do console, que podem ser habilitados ou desabilitados individualmente  
typedef enum {CONS\_APP = 0, CONS\_IO, CONS\_I2C, CONS\_EEPROM} CONS\_BLOCKS;
- // Define se o console deve ou não ecoar os dados recebidos  
typedef enum {CONS\_ECHO\_OFF = 0,CONS\_ECHO\_ON} CONS\_ECHO\_STATE;

### 3.3 EEPROM.h

As funções presentes nesse arquivo são:

- // Inicializa a EEPROM  
void EEInit(uint16\_t EEBaud);
- // Desabilita a EEPROM  
void EEDis(void);
- // Escreve um byte na EEPROM  
void EEWrByte(uint16\_t address, uint8\_t byte);
- // Escreve um buffer na EEPROM. O buffer não deve ser deve ser modificado até ser completamente // escrito na EEPROM. Isso acontece quando o status da EEPROM for IDLE  
void EEWrBuffer(uint16\_t address, uint8\_t \*byte, uint8\_t size);
- // Lê um byte da EEPROM e o escreve na endereço passado. O Dado só pode ser considerado // válido quando o status da EEPROM for IDLE  
void EERdByte(uint16\_t address, uint8\_t \*byte);
- // Lê um byte da EEPROM e o escreve na endereço passado. O Dado só pode ser considerado // válido quando o status da EEPROM for IDLE  
void EERdBuffer(uint16\_t address, uint8\_t \*byte, uint8\_t size);
- // Verifica o status do módulo EEPROM  
EE\_STATUS EEModuleStatus(void);
- // Essa função deve ser executada para que as ações seja efetivadas  
void EETask(void);

Os tipos de dados criados para a execução dessas funções são:

- // Enumera os possíveis estados da memória EEPROM: em espera ou ocupada  
typedef enum {EE\_IDLE=0,EE\_BUSY} EE\_STATUS;