

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Módulo de controle de motor a combustão interna
ciclo Otto monocilindro, baseado na arquitetura de
dsPIC (Microchip)

André Luis Lui

**São Carlos
2016**

André Luis Lui

**Módulo de controle de motor a combustão interna ciclo
Otto monocilindro, baseado na arquitetura de dsPIC
(Microchip)**

Trabalho de Conclusão de Curso
apresentado à Escola de Engenharia de
São Carlos da Universidade de São
Paulo.

Curso de Engenharia da Computação

Orientador: Prof. Dr. Luís Carlos
Passarini

VERSÃO CORRIGIDA

São Carlos

2016

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

L952m Lui, André Luis
 Módulo de controle de motor a combustão interna
 ciclo Otto monocilindro, baseado na arquitetura de
 dsPIC (Microchip) / André Luis Lui; orientador Luiz
 Carlos Passarini. São Carlos, 2016.

 Monografia (Graduação em Engenharia de Computação)
 -- Escola de Engenharia de São Carlos da Universidade
 de São Paulo, 2016.

 1. Microcontrolador. 2. Eletrônica embarcada. 3.
 Ciclo Otto. 4. Combustão interna. 5. Módulo de controle
 de motor. 6. Hardware. I. Título.

FOLHA DE APROVAÇÃO

Nome: André Luis Lui

Título: “Módulo de controle de motor a combustão interna ciclo Otto monocilindro, baseado na arquitetura de dsPIC”

Trabalho de Conclusão de Curso defendido em 30 / 11 / 2016 .

Comissão Julgadora:

Resultado:

Prof. Associado Luiz Carlos Passarini
(Orientador) - SMM/EESC/USP

Aprovado

Prof. Associado Evandro Luis Linhari Rodrigues
SEL/EESC/USP

Aprovado.

Prof. Dr. Álvaro Costa Neto
SMM/EESC/USP

Aprovado

Coordenador do Curso Interunidades Engenharia de Computação pela EESC:

Prof. Dr. Maximilian Luppe

Ao meu pai, meu grande incentivador no mundo automobilístico.

Agradecimentos

Agradeço à Universidade de São Paulo (USP), professores e funcionários com os quais tive contato na graduação e ajudaram no meu desenvolvimento profissional.

Agradeço ao professor orientador desse projeto, Luís Carlos Passarini, pelo apoio e incentivo desde o início dos trabalhos.

Agradeço à equipe EESC – USP Formula SAE, pela experiência profissional e os amigos adquiridos.

Agradeço aos amigos, Thiago Pinheiro, José Ernesto, Carlos Martinez, Danilo Porto, Daniel Finimundi, Diogo Julião, Lucas Tavares Zanuzzo, Leandro Machado e Luciano Falqueto, por transformarem esses anos na USP os melhores que já vivi em minha vida.

Agradeço à minha namorada, Mariana Castilho Martins, pela ajuda com a revisão do texto deste trabalho e, principalmente, pelos momentos juntos passados.

Agradeço ao meu irmão, Gabriel Lui, pelo apoio de sempre.

Agradeço ao meu pai, Luis Carlos Lui, por ser um grande incentivador do meu envolvimento na área automobilística, sendo no projeto EESC-USP Formula SAE e nesse trabalho de conclusão de curso. Agradeço também por ser meu melhor amigo.

Agradeço a minha mãe, Gláucia Regina Paulino dos Santos Lui, por sempre me incentivar a seguir o caminho dos estudos e por me incentivar a seguir a carreira que escolhi. Agradeço também pelas palavras meigas, e duras, ditas nos momentos certos e, principalmente, por ser a minha melhor amiga.

Resumo

LUI, A. L. **Módulo de controle de motor a combustão interna ciclo Otto monocilindro, baseado na arquitetura de dsPIC.** 2016. Trabalho de Conclusão de Curso (Graduação) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2016.

Em 1951 foi lançado no mercado o primeiro sistema de injeção eletrônica, que proporcionava redução do consumo de combustível e aumentava a potência do veículo, em busca de maior eficiência energética. Atualmente, as ECU's são módulos de controle complexos, capazes de gerenciar os componentes do motor, a fim de dar ao veículo o melhor equilíbrio entre emissões de poluentes e desempenho. Além disso, as pesquisas da área automobilística também estão direcionadas para a busca pela eficiência energética, segurança e conforto. Com isso, o mercado nacional de veículos tem sua demanda por eletrônica embarcada crescente, sendo uma de suas implementações nos módulos de controle de motor. O presente trabalho visou o desenvolvimento de um projeto de um módulo eletrônico de controle de um motor monocilindro que consiste em um microcontrolador que analisa sinais dos sensores e gera os pulsos para a válvula injetora de combustível e para a bobina de ignição, além de um software para visualização e alteração dos parâmetros do motor através de uma interface USB. Essa implementação foi dividida em três grandes partes: hardware, firmware e software. Além disso, foram realizados testes visando verificação de que estes três grandes componentes desempenhavam corretamente suas funções. A partir destes testes, foram obtidos resultados em que os sinais gerados apresentaram um erro máximo de 6% no posicionamento da onda em relação a posição do virabrequim e um erro de 2,4% na largura do pulso gerado.

Palavras-chave: Microcontrolador, Eletrônica embarcada, Ciclo Otto, Combustão interna, Módulo de controle de motor, *Hardware*.

Abstract

LUI, A. L. **Engine control unit for an internal combustion, Otto cycle, single cylinder engine, based on dsPIC architecture.** 2016. Trabalho de Conclusão de Curso (Graduação) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2016.

In 1951 it was launched the world's first electronic fuel injection system, which provided reduced fuel consumption and increased vehicle power, looking for greater energy efficiency. Currently, the ECUs are complex control modules able to manage the operation of various engine components, in order to give a better balance between emissions and performance in a vehicle. In addition, the automotive field of research is also directed to the search for energy efficiency, safety and comfort. Taking that into account, the domestic vehicle market has an increasing demand for electronics use in vehicles, and one of its implementations is in engine control modules. This study aimed to develop an engine control module designed for a single cylinder engine consisting of a microcontroller that analyzes signals from sensors and generates pulses for the fuel injector and ignition coil, and a software that allows visualizing and altering the engine's parameters with an USB interface. The implementation of this project was divided into three major parts: hardware, firmware and software. Moreover, certification tests were applied to these three major components to ensure its functions were correctly played. Regarding the test results, the signal generated showed a maximum error of 6% in the positioning of the wave relative to the crankshaft position and a 2.4% error in the width of the generated pulse.

Keywords: Microcontroller, Automotive electronics, Otto cycle, Internal combustion, Engine control unit, hardware.

Lista de figuras

Figura 1 - Tempo de admissão do motor.	26
Figura 2 - Tempo de compressão do motor.	26
Figura 3 - Tempo de exaustão ou explosão do motor	27
Figura 4 - Tempo de exaustão do motor.	28
Figura 5 - Princípio de funcionamento de um sensor indutivo.	29
Figura 6 - Correspondência entre o sinal do sensor indutivo e uma roda fônica.	30
Figura 7 - Esquema do funcionamento do efeito Hall.....	31
Figura 8 - Esquema de funcionamento de sensores MAP analógicos.....	32
Figura 9 - Esquema de funcionamento de sensores MAP digitais.....	32
Figura 10 - Circuito de utilização do termistor PTC.	33
Figura 11 - Comparação entre os sistemas SPI (à esquerda) e o sistema MPI (à direita).	36
Figura 12 - Esquema do injetor de combustível em corte.	37
Figura 13 - Esquema da bobina de ignição em corte.	38
Figura 14 - Encapsulamento do DSPic 30f4013.....	40
Figura 15 - Memória E2PROM (25LC1024) e suas conexões com o microcontrolador.	41
Figura 16 - Circuito MAX232 e suas conexões com o microcontrolador.....	42
Figura 17 - Placa expansora do circuito MCP2200 e suas conexões com o microcontrolador.	42
Figura 18 - Circuito MCP2551 e suas conexões com o microcontrolador.	43
Figura 19 - Circuito do sensor de rotação a ser simulado.	44
Figura 20 - Entrada (azul) e saída (vermelho) do circuito de rotação do motor, para 1,5V de tensão de pico da entrada.....	44
Figura 21 - Entrada (azul) e saída (vermelho) do circuito de rotação do motor, para 5V de tensão de pico da entrada.....	45
Figura 22 - Circuito do sensor de rotação implementado.	45
Figura 23 - Circuito referente a um filtro passa baixa.	46

Figura 24 - Circuito do filtro passa baixa a ser simulado.	47
Figura 25 - Diagrama de bode, com ganho (linha contínua) e fase (linha pontilhada) do filtro passa baixa.	47
Figura 26 - Curva da tensão de saída do circuito dos sensores de temperatura, pela resistência do sensor.	48
Figura 27 - Circuito dos sensores de temperatura implementados.	48
Figura 28 - Circuito do sensor TPS.	49
Figura 29 - PLM, da Motec.	49
Figura 30 - Circuito para aferição da tensão da bateria implementado.	50
Figura 31 - Circuito para aferição da temperatura do ECU implementado.	51
Figura 32 - Injetor de combustível utilizado nesse projeto.	51
Figura 33 - Circuito do acionamento do injetor de combustível a ser simulado.	52
Figura 34 - Resultado da simulação do circuito de acionamento do injetor de combustível. Em vermelho, a entrada do circuito. Em azul, a saída.	52
Figura 35 - Circuito implementado do acionamento da válvula injetora de combustível.	53
Figura 36 - Bobinas de ignição utilizadas nesse projeto.	54
Figura 37 - Circuito do acionamento da bobina de ignição a ser simulado.	54
Figura 38 - Resultado da simulação do circuito de acionamento da bobina de ignição. Em vermelho, a entrada do circuito. Em azul, a saída.	55
Figura 39 - Circuito implementado do acionamento da bobina de ignição.	55
Figura 40 - Conexões implementadas do microcontrolador.	56
Figura 41 - Aba "Dados" do software.	67
Figura 42 - Aba "Debug" do software.	68
Figura 43 - Diagrama de estado para o cálculo do período do recebimento de mensagens na porta serial.	69
Figura 44 - Diagrama de estados do programa principal.	70
Figura 45 - Código do software desenvolvido em LabVIEW.	72
Figura 46 - Pulso da injeção de combustível com tempo de injeção de 10ms, fase de injeção de 50° e sem diferença entre o PMS e a falha da roda fônica.	74

Figura 47 - Figura 45 - Pulso da injeção de combustível com tempo de injeção de 10ms, fase de injeção de 150° e sem diferença entre o PMS e a falha da roda fônica.	74
Figura 48 - Pulso da injeção de combustível com tempo de injeção de 30ms, fase de injeção de 100° e sem diferença entre o PMS e a falha da roda fônica.	75
Figura 49 - Pulso da injeção de combustível com tempo de injeção de 5ms, fase de injeção de 100° e sem diferença entre o PMS e a falha da roda fônica.	75
Figura 50 - Pulso da injeção de combustível com tempo de injeção de 10ms, fase de injeção de 150° e 220° de diferença entre o PMS e a falha da roda fônica.	76
Figura 51 - Pulso da bobina de ignição com dwell time de 10ms, avanço de 30° e sem diferença entre o PMS e a falha da roda fônica.	79
Figura 52 - Pulso da bobina de ignição com dwell time de 10ms, avanço de -30° e sem diferença entre o PMS e a falha da roda fônica.	79
Figura 53 - Pulso da bobina de ignição com dwell time de 10ms, avanço de -150° e sem diferença entre o PMS e a falha da roda fônica.	80
Figura 54 - Pulso da bobina de ignição com dwell time de 20ms, avanço de -150° e sem diferença entre o PMS e a falha da roda fônica.	80
Figura 55 - Pulso da bobina de ignição com dwell time de 10ms, avanço de 30° e 35° de diferença entre o PMS e a falha da roda fônica.	81
Figura 56 - Tensão aplicada no injetor de combustível.	85
Figura 57 - Tensão aplicada na bobina de ignição.	85

Lista de tabelas

Tabela 1 - Situação da mistura e o valor da relação ar/combustível correspondente..	34
Tabela 2 - Medições relativas ao parâmetro de tempo de injeção.....	77
Tabela 3 - Medições relativas ao parâmetro da fase da injeção.....	77
Tabela 4 - Medições relativas ao parâmetro da diferença entre o PMS e a falha da roda fônica na injeção.....	78
Tabela 5 - Medições relativas ao parâmetro de tempo de carga da bobina (Dwell Time) na ignição.	81
Tabela 6 - Medições relativas ao parâmetro de avanço na ignição.....	82
Tabela 7 - Medições relativas ao parâmetro da diferença entre o PMS e a falha da roda fônica na ignição.....	83
Tabela 8 - Maiores erros referente à injeção.....	83
Tabela 9 - Maiores erros referente à ignição.....	84

Lista de siglas

- 3G - Terceira geração da telefonia móvel
- 4G - Quarta geração da telefonia móvel
- ABS - *Anti-lock Breaking System*: Sistema de freio antitravamento das rodas
- ASCII - *American Standard Code for Information Interchange*: Código Padrão Americano para o Intercâmbio de Informação.
- BSFC - *Brake Specific Fuel Consumption*: Consumo específico de combustível
- CAN - *Controller Area Network*
- CKP - *Crankshaft position sensor*: Sensor do virabrequim
- CMP - *Camshaft position sensor*: Sensor do comando de válvulas
- Conversor A/D - Conversor de sinal analógico para digital
- Denatran - Departamento Nacional de Trânsito
- DIP - *Dual In-line Package*: Empacotamento duplo em linha
- DSP - *Digital Signal Processor*: Processador digital de sinais
- ECT - *Engine Coolant Temperature*: Temperatura dos liquido de refrigeração
- ECU - *Engine Control Unit*: Unidade de controle do motor
- E2PROM - *Electrically-Erasable Programmable Read-Only Memory*: Memória programável de apenas leitura apagável eletronicamente.
- EGR - *Exhaust Gas Recirculation*: Recirculação dos gases da exaustão
- GND - Ground: Nível de tensão de referencia
- GPS - *Global Positioning System*: Sistema de posicionamento global
- IAT - *Intake Air Temperature*: Temperatura do ar da admissão
- MPI - *Multi Point Injection*: Sistema de injeção multiponto
- NTC - *Negative Temperature Coefficient*: Coeficiente de variação de resistência com a temperatura é negativo
- PLM - *Professional Lambda Meter*: Medidor do fator lambda profissional
- PMI - Ponto morto inferior
- PMS - Ponto morto superior

- PTC - *Positive Temperature Coefficient*: Coeficiente de variação de resistência com a temperatura é positivo
- SPI - *Single Point Injection*: Sistema de injeção simples
- SUV - *Sport Utility Vehicle*: Veículo utilitário esportivo
- TPS - *Throttle Position Sensor* Sensor da posição da borboleta de aceleração
- USB - *Universal Serial Bus*
- WOT - *Wide open throttle*: Posição da borboleta de aceleração de plena carga

Sumário

1	INTRODUÇÃO	23
1.1	Apresentação	23
1.2	Motivação e Objetivos	23
1.3	Estrutura deste documento	24
2	EMBASAMENTO TEÓRICO	25
2.1	Motor de combustão interna	25
2.2	Ciclo Teórico do Motor 4 Tempos	25
2.3	Sensores	28
2.3.1	Indutivos	28
2.3.2	Efeito Hall	30
2.3.3	Pressão do ar no coletor	31
2.3.4	Temperatura	32
2.3.5	Posição da borboleta	33
2.3.6	Sensor de excesso de oxigênio	34
2.4	Atuadores	35
2.4.1	Válvulas injetoras de combustível	35
2.4.2	Bobina de ignição	37
3	IMPLEMENTAÇÃO	39
3.1	Hardware	39
3.1.1	Microcontrolador	39
3.1.2	Circuito de entrada	43
3.1.3	Circuito de saída	51
3.1.4	Manufatura da placa de circuito impresso	55
3.2	Firmware	56
3.2.1	Interrupções	57
3.3	Software	66

3.3.1	Funcionamento.....	67
3.3.2	Desenvolvimento	68
4	TESTES, RESULTADOS E DISCUSSÕES.....	73
4.1	Sinal de controle do injetor de combustível.....	73
4.2	Sinal de controle da bobina de ignição	78
4.3	Análise dos erros.....	83
4.4	Tensão nos componentes de saída	84
4.5	Comunicação USB	85
5	CONCLUSÕES.....	87
5.1	Trabalhos futuros.....	88
6	REFERÊNCIAS BIBLIOGRÁFICAS	89
	APÊNDICE A – Projeto do hardware.	91
	APÊNDICE B – Projeto do firmware.	92
	APÊNDICE C – Projeto do software.	93

1 INTRODUÇÃO

1.1 Apresentação

Com a evolução do carro movido a combustíveis fósseis a preocupação ambiental com os gases que eram emitidos aumentou na mesma proporção. Com essa preocupação a Bosch lançou em 1951 o primeiro sistema de injeção eletrônica que proporcionava uma redução de 20% do consumo de combustível ao mesmo tempo que aumentava de 23 para 28 cavalos de potência o veículo Gutbrod Superior 600. Anos depois, a mesma empresa colocou o primeiro sistema de injeção eletrônica numa linha de montagem, equipando o modelo americano do veículo Volkswagen 1900, fazendo com o que o modelo atendesse às normas californianas, consideradas as mais difíceis da época (BOSH, 2016).

No Brasil, o primeiro sistema de injeção eletrônica surgiu em 1988 para que modelos nacionais atendessem às exigências locais. Além disso, em 1993, a Bosch apresentou o sistema de injeção eletrônica para veículos a álcool do mercado brasileiro, tornando a frota brasileira movida a álcool mais confiável e menos poluente (BOSH, 2016).

Atualmente, as ECU's (*Engine Control Units*) são módulos de controle complexos, capazes de gerenciar o funcionamento das válvulas injetoras de combustível, das bobinas de ignição, da válvula de vazão de ar da marcha lenta, da EGR (*Exhaust Gas Recirculation*), do controle de tração e de estabilidade, a fim de dar ao veículo o melhor desempenho para o propósito para que foi criado. Para o caso de um veículo como o "Ex", protótipo desenvolvido pela equipe EESC-USP Formula SAE no ano de 2012, a ECU visou fornecer o menor BSFC (*Brake Specific Fuel Consumption*), ou seja, o menor consumo específico. Já para o protótipo da equipe EESCuderia Mileage, o mesmo módulo de controle tem como objetivo trazer ao veículo a máxima eficiência energética possível.

1.2 Motivação e Objetivos

O mercado nacional de veículos aumenta cada vez mais sua demanda por eletrônica embarcada, sendo nas centrais de entretenimento (rádios, GPS (*Global Positioning System*), navegadores para internet, módulos 4G/3G, entre outras) ou nos módulos de controle de motor, onde a busca pela eficiência energética, segurança e conforto é o alvo das pesquisas relacionadas nessa área.

Segundo o Denatran (Departamento Nacional de Trânsito) a frota nacional de veículos em agosto de 2016 era de pouco mais de 93 milhões de veículos, havendo previsão para que esse número cresça nos próximos anos (DENATRAN, 2016). Com isso, a procura por veículos que contenham as tecnologias citadas anteriormente é crescente, sendo necessário qualificar profissionais que consigam atender essa demanda de mercado.

Além disso, a complexidade que o desenvolvimento de uma ECU envolve pode ser extrapolada para outros projetos que utilizem eletrônica embarcada, uma vez que o trabalho aqui apresentado contempla desenvolvimento de *firmware* em microcontrolador, de um *hardware* de processamento de dados e de um *software* para interação com o *hardware* de maneira amigável e eficaz.

1.3 Estrutura deste documento

Este documento foi dividido nos seguintes capítulos: esta introdução, que lista as motivações que inspiraram e os objetivos do trabalho produzido, seguida do capítulo de embasamento teórico que tem como objetivo familiarizar o leitor com alguns conceitos fundamentais para a compreensão deste projeto. Posteriormente é relatado no capítulo de implementação como esta foi feita a nível de *hardware*, *firmware* e *software*. Após, o capítulo de Testes, Resultados e Discussões detalha os testes feitos no projeto, cujos resultados são analisados e discutidos. Então, no capítulo de conclusão, é retomado o que foi desenvolvido no projeto.

2 EMBASAMENTO TEÓRICO

Nesse capítulo foram apresentados os conceitos teóricos para o entendimento do projeto desenvolvido.

2.1 Motor de combustão interna

O modo de propulsão mais comum encontrados nos veículos atualmente ainda é o motor de combustão interna. Nele, um combustível é queimado, fazendo com que sua energia química seja transformada em energia cinética, proporcionando o movimento do veículo. Existem dois grupos principais de motores a combustão interna, os motores ICE, de ignição por centelha, e os motores ICO, de ignição por compressão.

No Brasil, os motores ICE são utilizados na maioria dos veículos, entre eles os automóveis de passeio e as motocicletas. Nesse modelo, há uma vela de ignição que, em um momento específico do funcionamento do motor, gera uma faísca elétrica que inicia a queima da mistura de ar e combustível dentro da câmara de combustão. Já os motores ICO são encontrados em sua maioria em caminhões e em alguns SUV's (*Sport Utility Vehicles*), e funcionam com a compressão do ar, que é aquecido e com isso, inflama o combustível que foi injetado dentro dos cilindros (PASSARINI, 2002).

2.2 Ciclo Teórico do Motor 4 Tempos

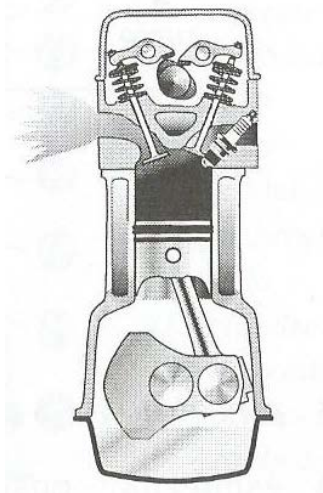
O ciclo de funcionamento de um motor de combustão interna, de ignição por centelha, naturalmente aspirado e em plena carga, ou WOT (*Wide Open Throttle*), foi implementado pelo engenheiro alemão Nikolaus Otto, que cedeu seu nome ao seu experimento, o Ciclo Otto (PULKRABEK, 1997).

O detalhamento de cada fase do ciclo teórico apresentada acima, se encontra a seguir (PASSARINI, 2002).

- Tempo de Admissão

Inicialmente na posição PMS (ponto morto superior), o pistão começa a descer em direção ao PMI (ponto morto inferior) gerando uma sucção dentro da câmara de combustão. Ao mesmo tempo, a válvula de admissão se abre, possibilitando que a mistura de ar e combustível seja aspirada. Ao fim do movimento do pistão, este se encontra na posição PMI e a válvula de admissão estará fechada. A figura 1 ilustra o tempo de admissão.

Figura 1 - Tempo de admissão do motor.

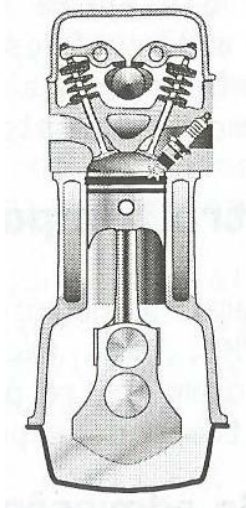


Fonte: PASSARINI, 2002, p. 6.

- Tempo de Compressão

Com a válvula de admissão fechada, o pistão começa seu movimento, saindo do PMS com destino ao PMS. Desse modo, toda a mistura aspirada no tempo de admissão é comprimida na região onde se encontra a vela. Pouco antes do pistão estar no PMS, ocorre o disparo da centelha, encerrando o tempo de compressão. A figura 2 ilustra o tempo de compressão.

Figura 2 - Tempo de compressão do motor.

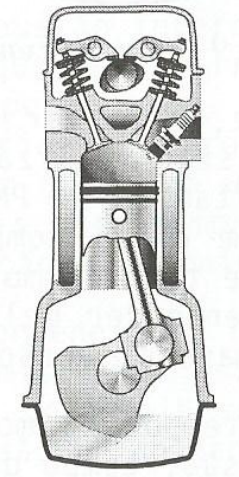


Fonte: PASSARINI, 2002, p. 6.

- Tempo de expansão ou explosão

Com a explosão da mistura, gases são gerados o que aumenta a pressão dentro da câmara de combustão. Desse modo, esses gases não têm por onde escapar, uma vez que as válvulas de admissão e de exaustão estão fechadas, a pressão gerada força o pistão a se deslocar para o PMI. Com o movimento do pistão, a pressão interna da câmara vai diminuindo. A figura 3 ilustra o tempo de expansão ou explosão.

Figura 3 - Tempo de exaustão ou explosão do motor

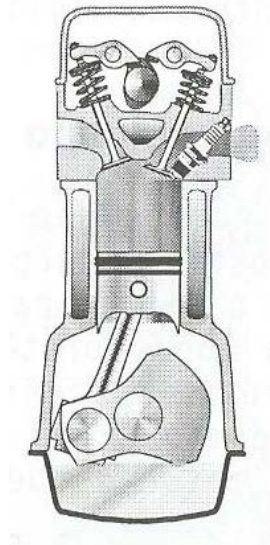


Fonte: PASSARINI, 2002, p. 7.

- Tempo de exaustão

Após a explosão, os gases gerados já não são mais úteis para a combustão e devem ser descartados. Assim, uma vez no PMI, o pistão começa a se deslocar em direção ao PMS. Ao mesmo tempo, a válvula de exaustão se abre, possibilitando que os gases gerados na combustão anterior sejam eliminados. Com o pistão se aproximando do PMS, a válvula de exaustão começa a fechar ao mesmo tempo que a de admissão começa a abrir. Dessa maneira, encerra o tempo de exaustão e o ciclo se repete com o tempo de admissão.

Figura 4 - Tempo de exaustão do motor.



Fonte: PASSARINI, 2002, p. 8.

2.3 Sensores

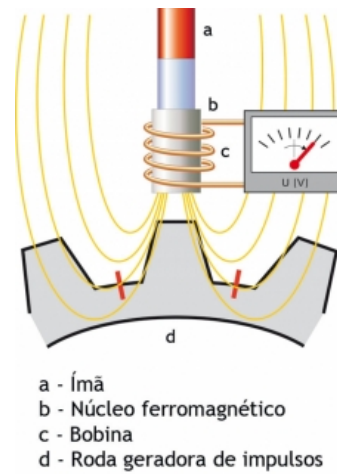
Os sensores têm um papel muito importante no gerenciamento eletrônico do motor. São eles os responsáveis por captar as características físicas que estão presentes em um determinado instante no motor, traduzindo-as em sinais elétricos. O funcionamento dos sensores indutivos, de efeito hall, de pressão do coletor de admissão, de temperatura, da posição da borboleta e o sensor lambda serão apresentados abaixo.

2.3.1 Indutivos

Os sensores indutivos estão presentes nos automóveis com a finalidade de detectar o movimento das partes mecânicas do motor, como a velocidade do virabrequim, posição do virabrequim e posição do comando de válvulas. Os sensores que obtêm os sinais que servirão de base para o cálculo dessas grandezas são o sensor CKP (*crankshaft position*) e o sensor CMP (*camshaft position*).

Esses sensores são compostos por um ímã permanente acoplado a uma bobina, com núcleo de material ferromagnético, e são instalados próximos a rodas dentadas, denominadas rodas fônicas, como é ilustrado na figura 5.

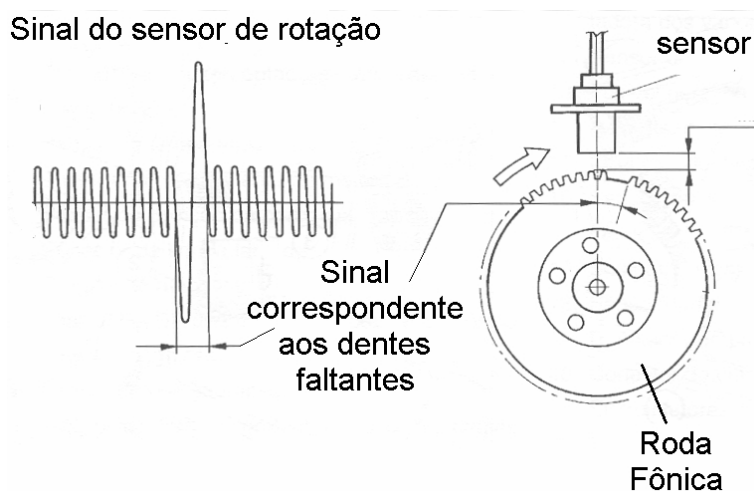
Figura 5 - Princípio de funcionamento de um sensor indutivo.



Fonte: CARROS INFOCO. 2016. Disponível em <http://www.carrosinfoco.com.br/carros/2012/07/injecao-eletronica-sensor-de-rotacao-vital-para-o-motor>

O funcionamento dos sensores indutivos é baseado na variação do campo magnético à frente do sensor. A movimentação de uma roda fônica faz o campo magnético, antes constante por conta do ímã permanente, variar. Para que seja detectado o início de uma nova volta, as rodas fônicas possuem referências, como a falta de um, ou mais, dentes. Dessa maneira, é induzida uma corrente na bobina do sensor, proporcional à variação do campo magnético, conforme apresentado na figura 6 (BRAGA, 2013).

Figura 6 - Correspondência entre o sinal do sensor indutivo e uma roda fônica.



Fonte: CARROS INFOCO. 2016. Disponível em

<http://www.carrosinfoco.com.br/carros/2012/07/injecao-eletronica-sensor-de-rotacao-vital-para-o-motor>

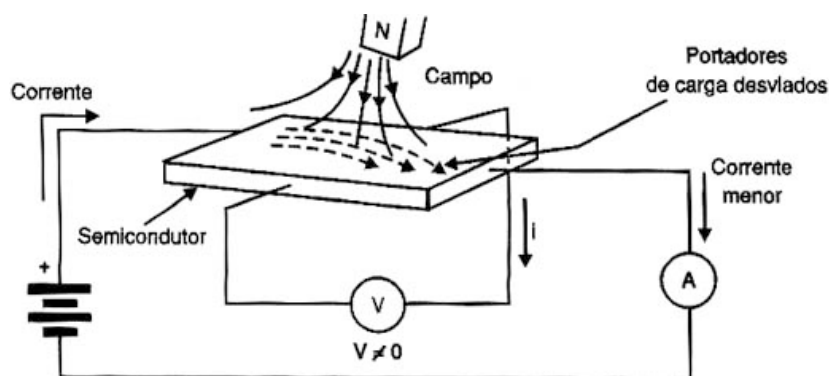
Devido ao seu princípio de funcionamento, os sensores indutivos são mais comumente utilizados onde há muita variação de campo magnético, uma vez que devido a essa variação, o sinal gerado possui valores de tensão altos, mantendo uma boa razão entre o sinal e o ruído.

2.3.2 Efeito Hall

O efeito Hall, descoberto por Edwin H. Hall em 1879, consiste na variação no caminho dos elétrons em uma placa de metal, quando submetidos a um campo magnético. O campo aplicado faz com que os elétrons mudem sua trajetória, acumulando nas extremidades da placa, assim, é possível detectar essa variação em forma de tensão, como é apresentado na figura 7 (INSTITUTO NEWTON C. BRAGA, 2016a).

Sensores de efeito Hall, assim como os sensores indutivos, são utilizados para a detecção de movimento das partes mecânicas do motor. Diferentemente dos sensores indutivos, esses sensores utilizam o efeito Hall para medir a presença de um campo magnético a sua frente.

Figura 7 - Esquema do funcionamento do efeito Hall.



Fonte: INSTITUTO NEWTON C. BRAGA. 2016. Disponível em <http://www.newtonbraga.com.br/index.php/como-funciona/6640-como-funcionam-os-sensores-de-efeito-hall-art1050>

Desse modo, sensores de efeito Hall apresentam grande aplicação nos casos em que há pouca variação de campo magnético, como é o caso do *Anti-lock Breaking System* (ABS). Além disso, por conta de seu processo de fabricação, esses sensores são mais caros que os indutivos.

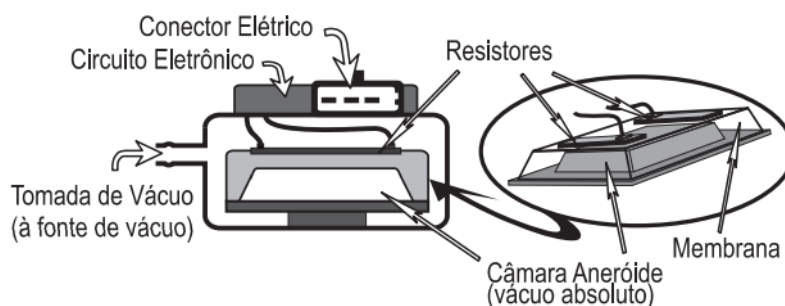
2.3.3 Pressão do ar no coletor

Para o controle mais preciso das funcionalidades do motor, é necessário medir quanto de ar está sendo aspirado pelo motor. Para isso, um dos métodos é a utilização de sensores que medem a pressão do ar no coletor de admissão, que junto com a medida da temperatura do ar fornecem a densidade para o cálculo da massa de ar aspirado pelo motor.

Esses sensores podem ser encontrados de duas maneiras: analógicos e digitais.

O funcionamento dos sensores analógicos se baseia, geralmente, em capsulas piezo-resistivas, em que um diafragma cerâmico divide duas capsulas, uma com vácuo, e outra com a pressão que se quer determinar. Com a deformação desse diafragma, o valor de sua resistência se altera, possibilitando que circuito auxiliares forneçam uma tensão proporcional à pressão absoluta que se quer medir. Pode haver sensores que não possuem a câmara de vácuo, nesse caso o valor apresentado na saída do sensor é referente à pressão atmosférica. O funcionamento desse modelo de sensor é ilustrado na figura 8 (MTE – THOMSON, 2016a).

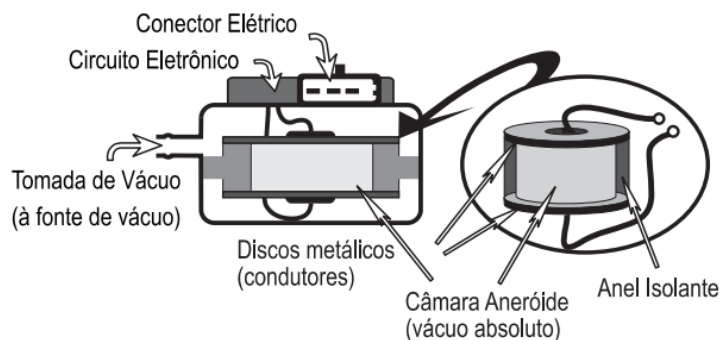
Figura 8 - Esquema de funcionamento de sensores MAP analógicos.



Fonte: MTE – THOMSON. 2016. Disponível em <http://www.mte-thomson.com.br/site/wp-content/uploads/2014/10/SENSOR-MAP.pdf>

Por outro lado, os sensores digitais, consistem em duas placas de alumínio paralelas separadas por um anel isolante, possuindo vácuo em seu interior. Essa configuração apresenta características elétricas semelhantes à de um capacitor. A pressão que se quer analisar se encontra ao redor desse cilindro, como pode ser visto na figura 9, podendo alterar as placas conforme seu valor. Com isso, circuitos elétricos auxiliares emitem um sinal pulsado cuja frequência depende do tempo de carga desse capacitor (MTE – THOMSON, 2016a).

Figura 9 - Esquema de funcionamento de sensores MAP digitais.



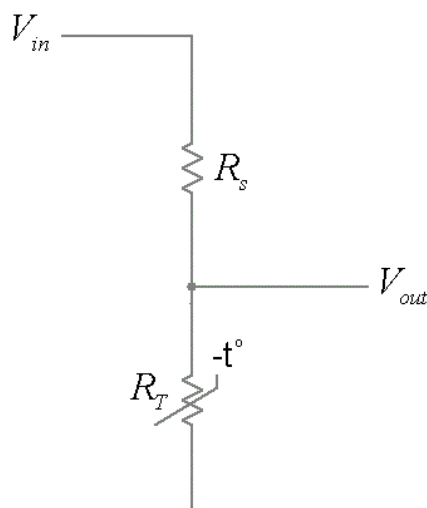
Fonte: MTE – THOMSON. 2016. Disponível em <http://www.mte-thomson.com.br/site/wp-content/uploads/2014/10/SENSOR-MAP.pdf>

2.3.4 Temperatura

A medição da temperatura nos veículos é baseada em termistores, nos quais ocorre a variação de sua resistência com a variação da temperatura.

Termistores podem ser apresentados como NTC (*Negative Temperature Coefficient*), nos quais a resistência diminui com o aumento de temperatura, ou como PTC (*Positive Temperature Coefficient*), nos quais a resistência aumenta com o aumento de temperatura. Em ambos os casos, sensores podem ser incorporados a circuitos de divisões de tensão, como apresentado na figura 10, a fim de se obter uma curva de tensão pela temperatura (THOMAZINI, D; ALBUQUERQUE, P. U. B, 2007).

Figura 10 - Circuito de utilização do termistor PTC.



Fonte: PeterVis. 2016. Disponível em

<http://www.petervis.com/electronics%20guides/calculators/thermistor/thermistor.html>

Na figura 10, “ R_t ” representa um termistor PTC, “ V_{in} ” a tensão de alimentação, “ R_s ” o resistor complementar do divisor de tensão e “ V_{out} ” a tensão de saída do circuito. Dessa forma, com o aumento de temperatura, a resistência de “ R_t ” aumenta, gerando uma tensão “ V_{out} ” proporcional à temperatura medida.

2.3.5 Posição da borboleta

Para encontrar o valor referente à posição da borboleta de aceleração é utilizado um potenciômetro acoplado ao eixo desta. Sendo o potenciômetro um mecanismo de divisão de tensão, é possível obter uma curva da posição da borboleta pela tensão, de modo linear. Desse modo, o valor de tensão na saída do sensor será alterado na mesma proporção em que se abre a borboleta de aceleração (*Throttle Position Sensor*) (MTE – THOMSON, 2016b).

2.3.6 Sensor de excesso de oxigênio

O sensor Lambda possui a função de indicar ao módulo de controle do motor a qualidade (pobre, estequiométrica ou rica) com que a queima da mistura de ar e combustível está ocorrendo. Para isso o sensor mais comum possui duas placas de óxido de zircônio, podendo ser composto por placas de óxido de titânio, uma que entra em contato com os gases resultantes da queima da mistura e outra que entra em contato com o ar atmosférico, que servirá como referência. Com a interação dos íons de oxigênio com o óxido de zircônio, este acaba por acumular cargas negativas, uma vez que o íon referente ao oxigênio é o O⁻, o mesmo ocorre com a placa externa. Como a disponibilidade de oxigênio entre as placas do óxido são diferentes, surge uma diferença de cargas que resulta em uma diferença de potencial que servirá de base para o sinal de saída do sensor THOMAZINI, D; ALBUQUERQUE, P. U. B, 2007).

O fator lambda é a razão entre a relação de ar e combustível da mistura que foi recém queimada, com a relação da mistura estequiométrica. Ou seja, o fator lambda respeita a relação:

$$\lambda = \frac{\left(\frac{Ar}{Combustível}\right)_{Corrente}}{\left(\frac{Ar}{Combustível}\right)_{Estequiométrico}}$$

Os valores para a relação mássica Ar / Combustível estequiométrica possui valores apresentados na tabela 1.

Tabela 1 - Situação da mistura e o valor da relação ar/combustível correspondente.

Relação Ar/Combustível	Situação da mistura
< 14,7 : 1	Rica
14,7 : 1	Estequiométrica
> 14,7 : 1	Pobre

Fonte: MTE – THOMSON. 2016. Disponível em <http://www.cursosonline.mte.com.br/licao/aula-14-sensor-lambda/>

No mercado, é possível encontrar dois tipos de sensores lambda baseados em óxido de zircônio, no que diz respeito à curva do sinal de saída, o sensor *narrow-band* e o sensor *wide-band*. O primeiro possui a faixa de transição entre as regiões de mistura rica e pobre muito estreita, fato que faz com que o ECU interprete o sinal como uma chave liga-desliga entre os modos de enriquecimento e empobrecimento da mistura. Já

os sensores *wide-band*, possuem essa faixa mais larga, proporcionando para as ECU's um controle melhor.

2.4 Atuadores

Os atuadores são os componentes que irão ser controlados pela ECU para interagir com o motor. Após a obtenção dos dados pelos sensores e o processamento desses pelo ECU, sinais eletrônicos são enviados aos atuadores a fim de alterar um comportamento do motor.

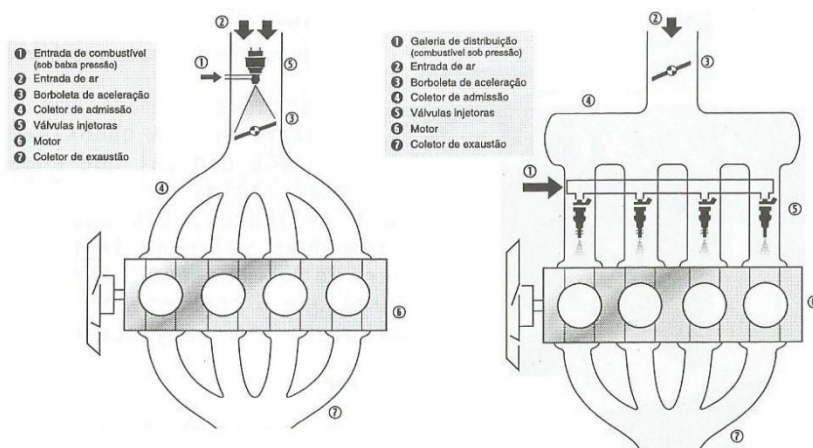
Para esse projeto os atuadores relacionados foram as válvulas injetoras de combustível e as bobinas de ignição.

2.4.1 Válvulas injetoras de combustível

São as válvulas injetoras que fornecem combustível finamente pulverizado para a queima na câmara de combustão. Quando esse fornecimento ocorre dentro da própria câmara, o sistema de injeção é direto. Porém quando o fornecimento do combustível ocorre ainda no coletor de admissão, o sistema de injeção é indireto.

Entre os sistemas de injeção indiretos podemos destacar o sistema multiponto (ou *Multi Point Injection* - MPI) aquele em que as válvulas injetoras estão dispostas adiante de cada cilindro no duto de admissão. Por outro lado, quando um sistema de injeção conta com apenas uma válvula injetora de combustível, localizada antes da união dos dutos de admissão, esse recebe o nome de monoponto (ou *Single Point Injection* - SPI). Por conta da melhor qualidade das emissões de gases poluentes em situações de transiente, o primeiro sistema é o que predomina nos carros atualmente. A figura 11 ilustra a diferença entre o MPI e o SPI (PASSARINI, 2002).

Figura 11 - Comparação entre os sistemas SPI (à esquerda) e o sistema MPI (à direita).

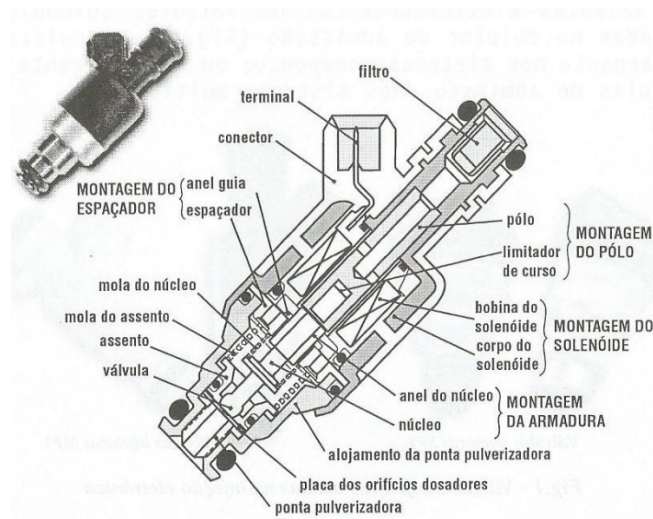


Fonte: PASSARINI, 2002, p. 24 e 27. Adaptado pelo autor.

Outra classificação que se dá ao sistema de injeção de combustíveis é referente a quantas injeções serão realizadas por ciclo do motor. Caso ocorra apenas uma injeção, antes do tempo de admissão, é caracterizado um sistema de injeção sequencial. Por outro lado, caso a quantidade total de combustível seja dividida em injeções antes do tempo de admissão e antes do tempo de explosão, é caracterizado um sistema semi-sequencial, ou ainda sistema simultâneo (SDF). Maiores detalhes podem ser encontrados no livro Motores automotivos: evolução, manutenção e tendências, de Fábio Melo de Castro.

A maioria das válvulas injetoras de combustível têm seu funcionamento baseado num sistema de solenoides, que consiste em uma bobina ligada aos terminais elétricos do injetor. Sendo assim, quando há corrente passando pelos terminais, a bobina gera uma força contrária e maior que a força da mola abrindo um espaço entre a armadura e seu assento que permite a saída do combustível pelo(s) orifício(s) dosador(es). Por outro lado, quando não há corrente passando pelos terminais da válvula injetora, o sistema fecha, com a mola e a pressão do combustível forçando a armadura contra seu assento, não permitindo a saída do combustível. A figura 12 ilustra os componentes mais importantes do injetor de combustível (PASSARINI, 2002).

Figura 12 - Esquema do injetor de combustível em corte.



Fonte: PASSARINI, 2002, p. 8.

A geração do pulso de controle da válvula injetora de combustível depende do tempo e da fase da injeção. O parâmetro referente ao tempo de injeção é responsável pela quantidade de combustível que o motor irá utilizar no próximo ciclo. Já a fase da injeção determina quando a injeção de combustível termina em relação à posição do virabrequim, uma vez que esta auxiliará na vaporização do combustível antes de entrar na câmara de combustão.

2.4.2 Bobina de ignição

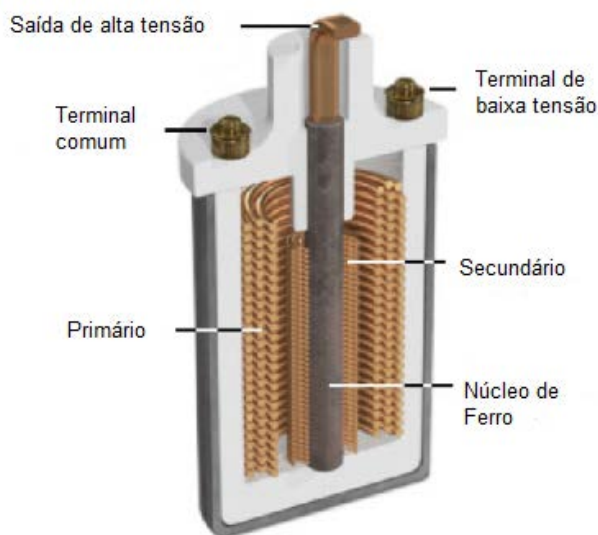
Após a mistura de ar e combustível ser comprimida durante o ciclo de compressão, a explosão da mistura não ocorre espontaneamente. Para isso, é necessária uma faísca que provém da vela de ignição.

Porém, sendo o ar atmosférico é um isolante elétrico, é necessário vencer a rigidez dielétrica do ar. Essa barreira é vencida quando o campo elétrico entre dois terminais é superior a 10 KV/cm e as partículas do ar entre eles estão ionizadas tornando o ar um condutor elétrico.

Para que, na vela, os valores de tensão sejam grandes o suficiente para vencer a rigidez dielétrica do ar, a bobina de ignição precisa agir como um transformador, elevando a tensão da bateria de 12 V para mais de 12 000 V na vela, dependendo do modelo de motor. A bobina de ignição possui dois enrolamentos, sendo que o primário possui poucas voltas e sua corrente é chaveada pelo circuito controlador da ignição no ECU. Já o secundário, possui um número bem maior de voltas e a descarga, que resulta

na faísca na vela, ocorre com o chaveamento do primeiro enrolamento, após o carregamento da bobina. A proporção entre as voltas dos fios do primeiro e do segundo enrolamento correspondem ao fator multiplicativo da tensão da bateria para a tensão na vela. A figura 13 ilustra uma bobina de ignição em corte (INSTITUTO NEWTON C. BRAGA, 2016b).

Figura 13 - Esquema da bobina de ignição em corte.



Fonte: INSTITUTO NEWTON C. BRAGA. 2016. Disponível

<http://www.newtoncbraga.com.br/index.php/como-funciona/6713-como-funciona-o-sistema-de-ignicao-art1342>. Adaptado pelo autor.

De modo semelhante ao sistema de injeção de combustíveis, a faísca (ou centelha) pode ocorrer em dois tempos de um ciclo do motor. Caso a centelha ocorra no final do tempo de compressão e no final do tempo de exaustão, esta recebe o nome de “centelha perdida”, ou *wasted spark*. Esse fenômeno não produz nenhum efeito sobre o funcionamento do motor, uma vez que não há mistura de ar e combustível aptos a serem queimados na câmara de combustão.

Como parâmetro de controle do acionamento da bobina, o pulso gerado pelo ECU possui o tempo de carga (*dwell time*) e o valor do avanço da centelha. O primeiro destes determina o tempo em que uma corrente precisará passar pelo enrolamento primário para carregar a bobina. Caso esse tempo seja maior que o necessário, poderá comprometer o componente que está sendo controlado. Já o valor do avanço determina quando a faísca surgirá na vela. Esse momento é determinado pelo fim da carga do enrolamento primário e é caracterizado como o valor, em graus, que falta para que o pistão atinja o PMS.

3 IMPLEMENTAÇÃO

A implementação desse projeto foi dividida em três grandes partes: *hardware*, *firmware* e *software*. O *hardware* desenvolvido consiste em uma placa de circuito impresso que suporta o microcontrolador e os circuitos que auxiliam a entrada e saída dos sinais elétricos. O *firmware* consiste nas rotinas programadas no microcontrolador que analisam os sinais de entrada e, a partir deles, geram os sinais de saída. Já o *software*, consiste em um programa desenvolvido na plataforma LabVIEW que permite a alteração dos parâmetros de controle sem que haja a necessidade de se reprogramar o código do microcontrolador.

Essa implementação teve como base um motor monocilindro, com 4 tempos, um sistema semi-sequencial para injeção de combustível e um sistema de ignição com centelha perdida. Além disso, o sistema de injeção de combustível possui tempo e fase de injeção fixos, assim como no sistema de ignição, o tempo de carga da bobina e o avanço.

As três partes serão detalhadas a seguir.

3.1 Hardware

O desenvolvimento do *hardware* do projeto é baseado na escolha do microcontrolador, suas interfaces, circuitos dos sinais de entrada, circuitos dos sinais de saída e a confecção da placa de circuito impresso que suportará tais circuitos. Os circuitos apresentados foram desenhados na ferramenta Altium Designer 13.2 e simulados na ferramenta LTSpice IV.

3.1.1 Microcontrolador

Para o desenvolvimento do projeto, foi utilizado o DSP (*Digital Signal Processor*) da Microchip, modelo DSPic 30f4013. Tal microcontrolador foi utilizado pelo fato de possuir a frequência de trabalho de 20 MHz, possibilitando cerca de 100 mil instruções por volta do motor em uma rotação de 10.000 rpm. Além disso, o microcontrolador possui o encapsulamento DIP (*Dual In-line Package*), apresentado na figura 14, e possuir todos os recursos necessários ao projeto (MICROCHIP, 2016a).

Figura 14 - Encapsulamento do DSPic 30f4013.



Fonte: MICROCHIP. 2016. Disponível em <http://www.microchip.com/wwwproducts/en/dsPIC30F4013>.

Tal encapsulamento facilita a elaboração da placa de circuito impresso, uma vez que os pinos de contatos são de fácil soldagem, podendo os componentes serem soldados manualmente. Além de possibilitar testes em uma matriz de contato.

3.1.1.1 Periféricos do microcontrolador

Os periféricos do microcontrolador utilizados nesse projeto são listados abaixo.

- Entradas e saídas digitais

São 30 pinos, dos 40 que o microcontrolador possui, que podem assumir a função de obter valores digitais, '0' ou '1'. Além disso, dois desses 30 pinos, podem gerar interrupções no código, fazendo com que o processador execute funções específicas para o sinal obtido (MICROCHIP, 2016a).

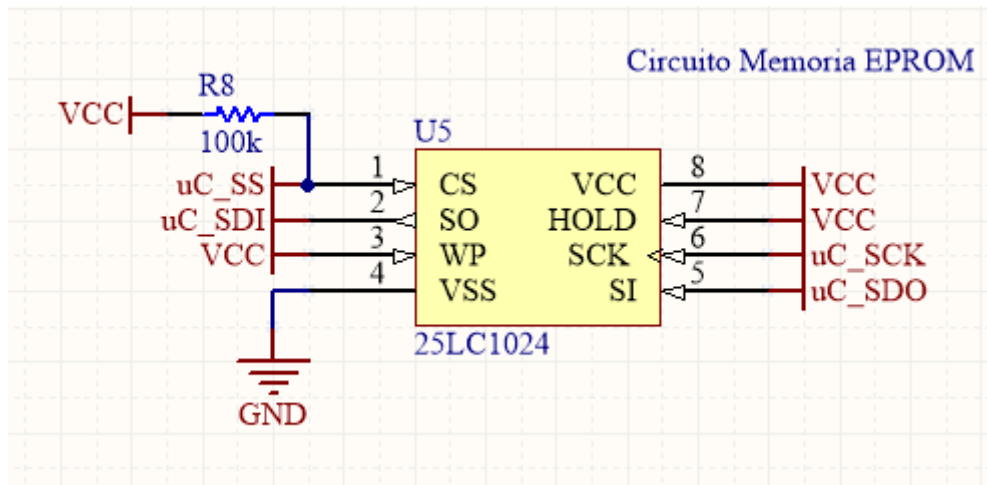
- Entradas analógicas

O microcontrolador escolhido possui um conversor A/D de 12 bits, 13 canais e uma frequência de amostragem de 200 ksps. Com ele, todos os sinais analógicos podem ser obtidos com uma precisão de, aproximadamente, 1,22 mV/bit.

- *Serial Peripheral Interface*

O barramento *Serial Peripheral Interface*, nesse projeto, possui a função de se comunicar com o módulo 25LC1024, uma memória de 1024 Kbits, que foi designada para armazenar os dados dos parâmetros de controle do motor (MICROCHIP, 2016b). Possibilitando que a ECU seja desligada da alimentação e seus dados não sejam perdidos. As conexões desses componentes são apresentadas na figura 15.

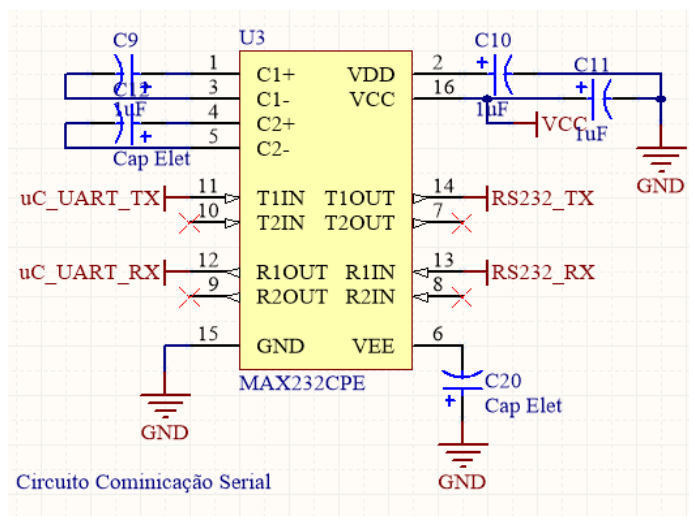
Figura 15 - Memória E2PROM (25LC1024) e suas conexões com o microcontrolador.



- Interface serial

Com dois canais, a interface serial foi utilizada para comunicação externa com o ECU. Um deles foi acoplado a um conversor de nível, que realiza a compatibilidade do sinal emitido pelo microcontrolador e o sinal do protocolo serial, que possui níveis de tensão de +12 V e -12V. As conexões desse conversor de nível se encontram na figura 16.

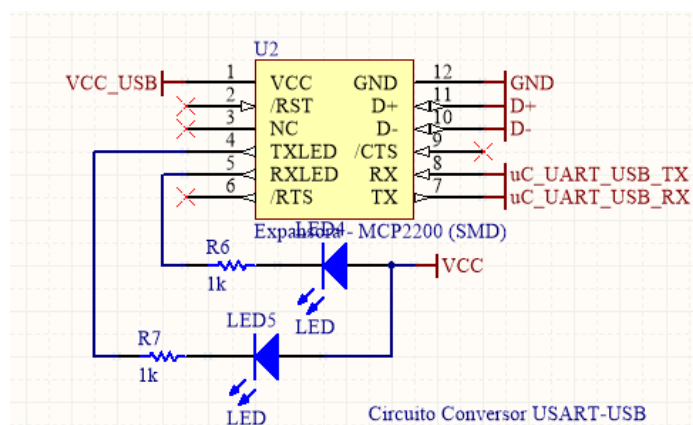
Figura 16 - Circuito MAX232 e suas conexões com o microcontrolador.



- Interface USB

Para obter uma conexão simplificada com o computador, foi utilizado o MCP2200, da Microchip, que é uma ponte entre o barramento serial e o USB (MICROCHIP, 2016c). Desse modo, um dos barramentos seriais disponíveis no microcontrolador foi utilizado como porta USB. As conexões da placa expansora que contém o MCP2200 e a placa do ECU estão apresentadas na figura 17.

Figura 17 - Placa expansora do circuito MCP2200 e suas conexões com o microcontrolador.

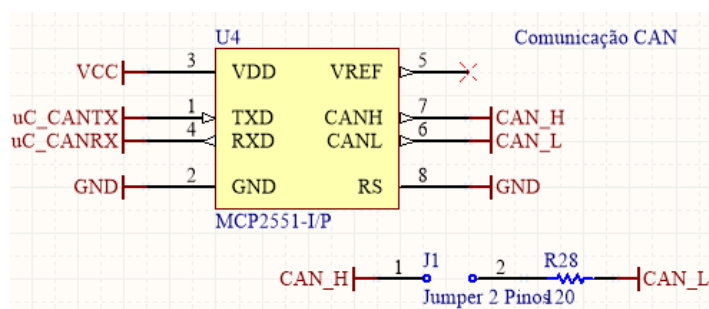


- Controller Area Network (CAN)

O barramento CAN é amplamente utilizado na indústria automobilística. Desse modo, a utilização desse barramento para comunicação com a ECU foi implementada. O microcontrolador possui os pinos CANTx e CANRx, que enviam e recebem dados

CAN, respectivamente. Porém, esses sinais enviados não estão dispostos em par diferencial, como o barramento CAN é descrito. Para seu funcionamento, é utilizado o MCP 2551, da Microchip, fazendo com que os sinais sejam colocados nos níveis de tensão corretos para seu funcionamento em par trançado (MICROCHIP, 2016d). As conexões referentes ao MCP 2551 estão apresentadas na figura 18.

Figura 18 - Circuito MCP2551 e suas conexões com o microcontrolador.



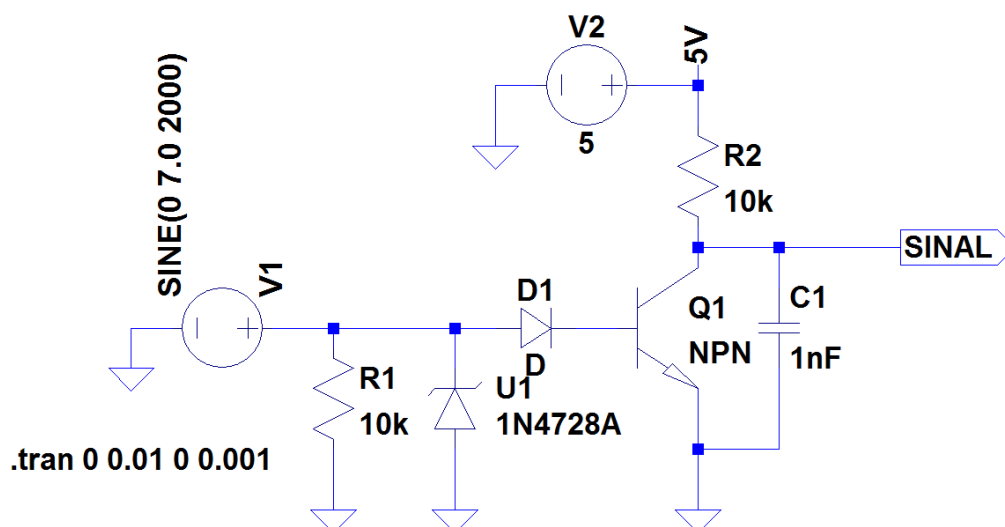
3.1.2 Circuito de entrada

3.1.2.1 Circuito do sensor da rotação

Para medir a rotação do motor, foi utilizado um sensor indutivo que fornece uma onda cujo valor de tensão é proporcional a variação do fluxo magnético. Para o projeto deste circuito, o sinal do sensor indutivo foi submetido a um resistor de *pull-down*, um diodo zener de 4,7 V e um diodo retificador. Com isso, quando o sensor estiver desconectado o nível de referência (GND) estará presente e o limite superior de tensão será de 4,7 V. Com isso, este sinal é conectado à base de um transistor bipolar NPN, que conduz uma corrente no resistor de 10K Ohms, gerando a tensão adequada para a leitura do microcontrolador.

O funcionamento do circuito descrito acima foi verificado com simulações que apresentam um sinal de entrada (fonte V1) senoidal com frequência de 2000 Hz e tensão de pico de 5 V e de 1,5 V. O circuito simulado é apresentado na figura 19.

Figura 19 - Circuito do sensor de rotação a ser simulado.



Como resultado da simulação, foram comparadas as curvas de entrada e saída do circuito. Obtendo os gráficos da figura 20, para tensão de pico de 1,5V na entrada do circuito, e os gráficos da figura 21, para tensão de pico de 5V na entrada do circuito.

Figura 20 - Entrada (azul) e saída (vermelho) do circuito de rotação do motor, para 1,5V de tensão de pico da entrada.

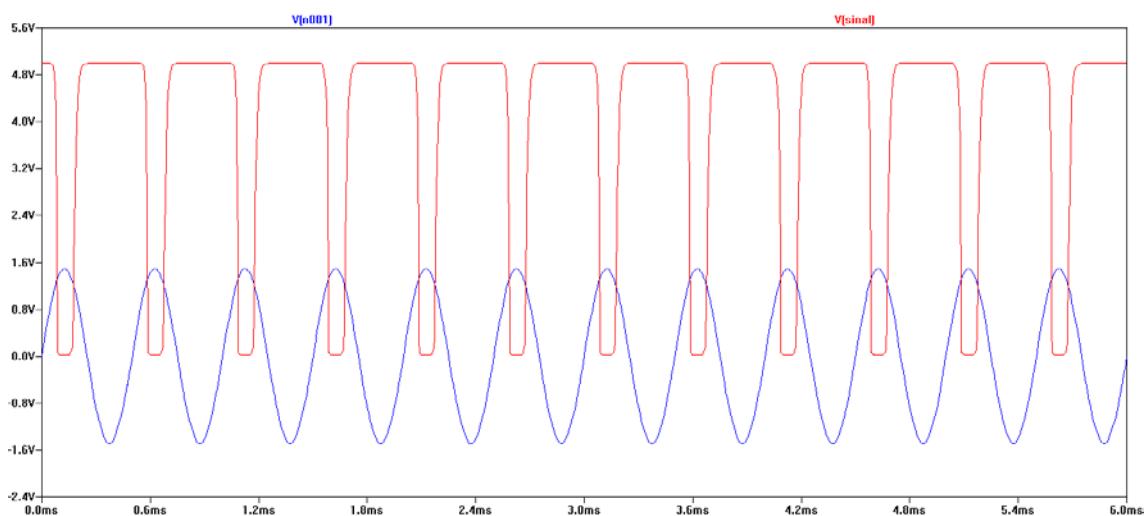
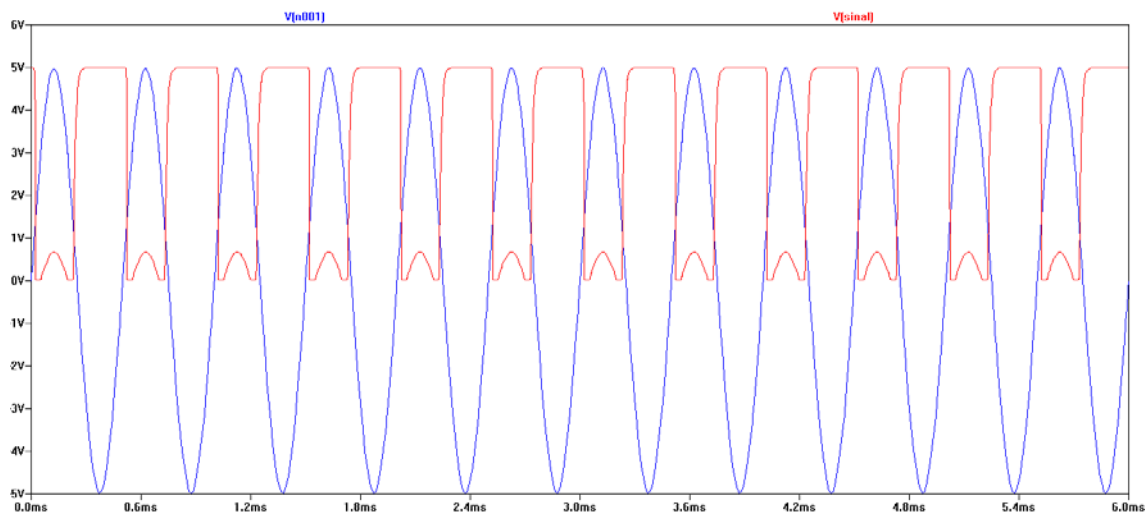
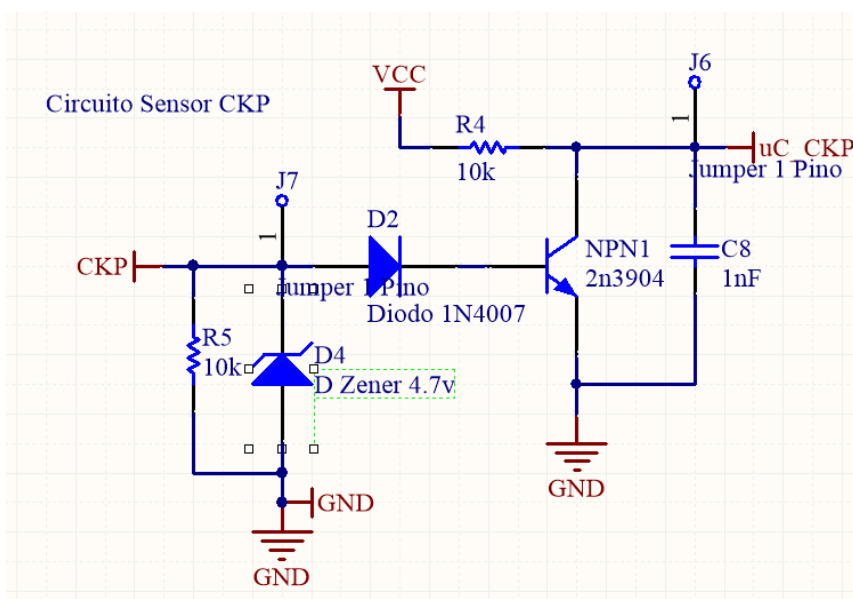


Figura 21 - Entrada (azul) e saída (vermelho) do circuito de rotação do motor, para 5V de tensão de pico da entrada.



Para ambos os casos, o sinal de saída é uma onda quadrada, com a tensão variando de 0 a 5V, de modo que este sinal seja apto a ser lido pelo microcontrolador, sem causar danos ao seu funcionamento. A partir disso, o esquemático utilizado no ECU se encontra na figura 22.

Figura 22 - Circuito do sensor de rotação implementado.



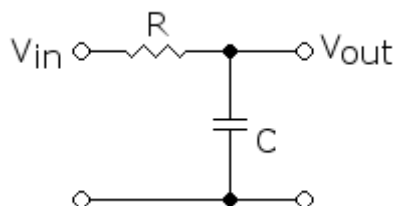
3.1.2.2 Circuito dos sensores analógicos

Os sensores analógicos têm como função monitorar o estado corrente do motor, possibilitando que a ECU realize possíveis alterações no pulso controlador da válvula injetora de combustível e da bobina de ignição. Como esse trabalho visa a

implementação de um sistema de injeção de combustível com tempo e fase de injeção fixos, assim como o tempo de carga da bobina e avanço, os sensores analógicos serão utilizados apenas em futuras atualizações do projeto.

Para a obtenção dos dados analógicos pelo microcontrolador, foi aplicado um filtro passa baixa nos circuitos, conforme apresentado na figura 23.

Figura 23 - Circuito referente a um filtro passa baixa.



A frequência de corte desse circuito foi baseada no funcionamento da bobina, uma vez que por esta passam as maiores correntes do ECU, o que a faz ser uma grande geradora de ruído nos outros circuitos. Assim, foi escolhida a rotação de 300 RPM para ser a frequência de corte do filtro, uma vez que essa é próxima a de rotação de um sistema sequencial de ignição. Como esse projeto visa implementar um sistema de ignição semi-sequencial, o filtro projetado cobre a implementação proposta.

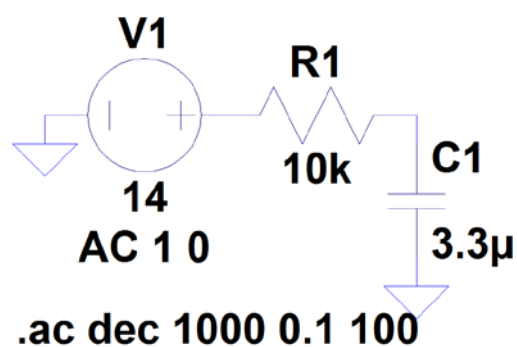
Desse modo, foi utilizada a equação a seguir para o cálculo da frequência de corte do filtro passa baixa.

$$F_c = \frac{1}{2 * \pi * R * C}$$

Fixando o valor de R para 10k Ω , que é um valor de resistência comercial, encontra-se o valor de C de 3,18 μF . Como esse valor não é encontrado no comércio, foi escolhido o valor de 3,3 μF . Assim, com os valores de R e C obtidos, a frequência de corte do filtro foi novamente calculada com a equação 1, obtendo o valor de 289 rpm.

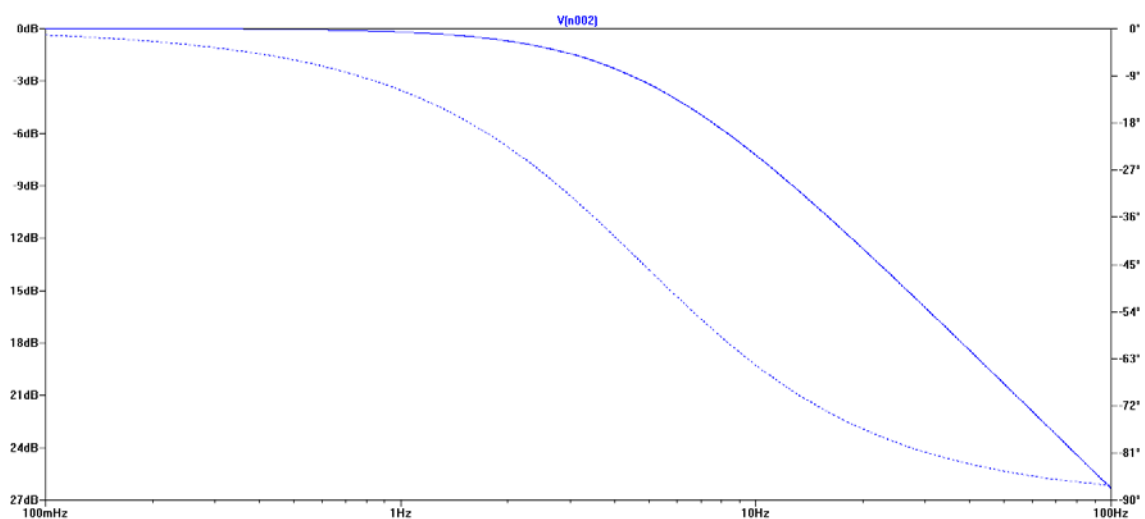
Com os valores obtidos acima, foi realizada uma simulação do filtro a fim de determinar seu comportamento. O circuito simulado se encontra na figura 24.

Figura 24 - Circuito do filtro passa baixa a ser simulado.



O sinal de entrada da simulação é de 1V de amplitude, variando a frequência de 0,1 Hz (6 rpm) até 100 Hz (6000 rpm). Como resultado da simulação foi obtido o diagrama de bode apresentado na figura 25.

Figura 25 - Diagrama de bode, com ganho (linha contínua) e fase (linha pontilhada) do filtro passa baixa.

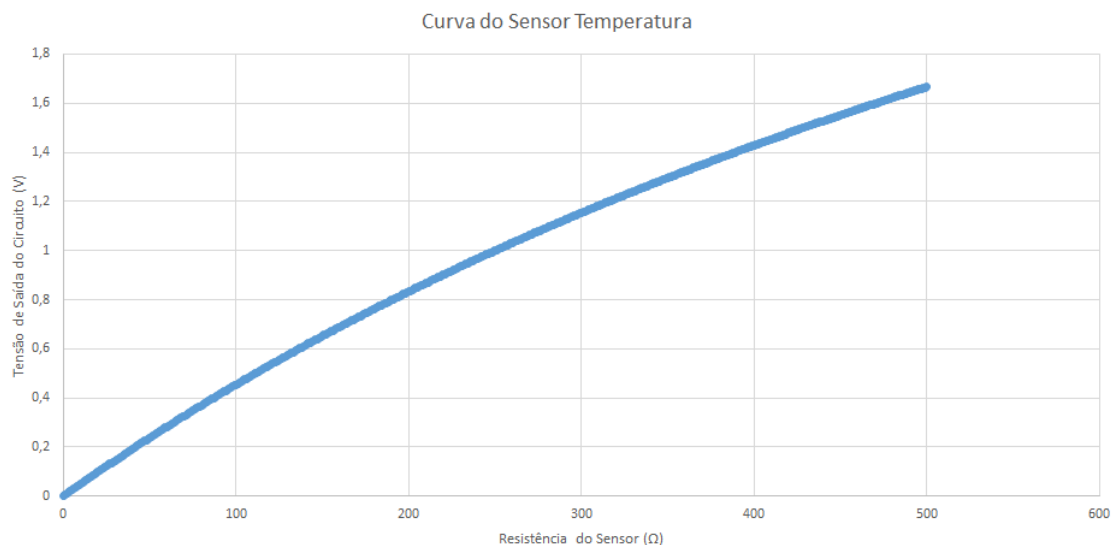


Com o resultado apresentado na figura 25, tem-se que a partir de 5 Hz (300 rpm), o sinal ruidoso começa a ter uma queda de 3 dB.

3.1.2.2.1 Circuito dos sensores de temperatura

Os sensores de temperatura encontrados nos automóveis atualmente são baseados em termistores. Com isso, para se obter uma tensão a ser lida pelo conversor A/D, foi utilizado o circuito apresentado na figura 7, obtendo a curva de resistência do sensor pela tensão de saída do circuito, apresentado na figura 26.

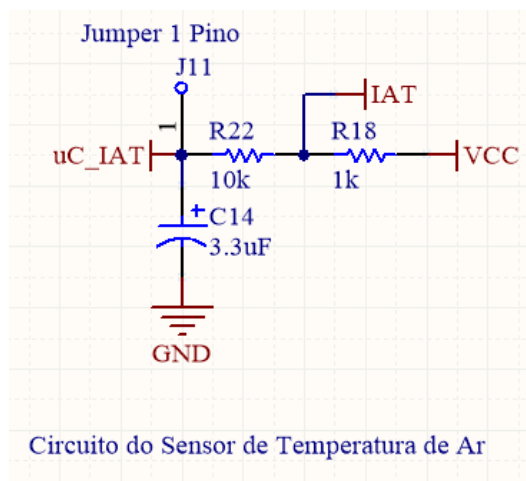
Figura 26 - Curva da tensão de saída do circuito dos sensores de temperatura, pela resistência do sensor.



Com a curva apresentada na figura 26, pode-se realizar o mapeamento do valor obtido no conversor A/D com a temperatura real obtida pelo sensor.

A demonstração desse circuito foi realizada conforme apresentado na figura 27.

Figura 27 - Circuito dos sensores de temperatura implementados.

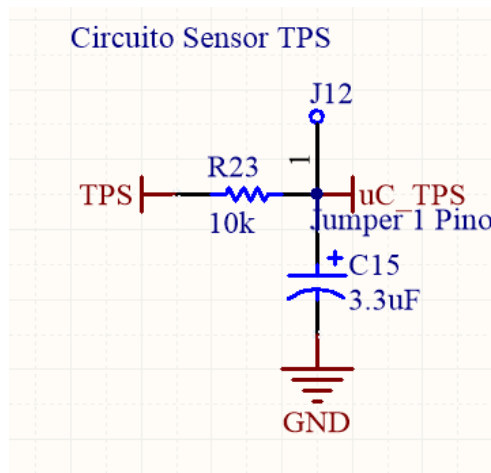


3.1.2.2.2 Circuito dos sensores de pressão, do sensor Lambda e da posição da borboleta de aceleração.

Os sensores de pressão e o TPS (*Throttle Position Sensor*) fornecem uma tensão proporcional à grandeza física que estão medindo, dentro da tensão de alimentação. Desse modo, para se obter tal valor, basta que o sinal do sensor esteja

colocado na entrada do conversor A/D. A implementação desse circuito se encontra na imagem 28.

Figura 28 - Circuito do sensor TPS.



Devido à complexidade do circuito para a obtenção do valor do lambda, foi esperado que este sinal fosse gerado por um circuito auxiliar que forneça um valor de tensão proporcional ao valor lambda medido. Para isso, o circuito projetado tem como entrada um sinal semelhante ao gerado pelo PLM (*Professional Lambda Meter*), da Motec, que pode ser visto na figura 29.

Figura 29 - PLM, da Motec.



Fonte: MOTEC. 2016. Disponível em <http://www.motec.com/plm/plmoverview/>

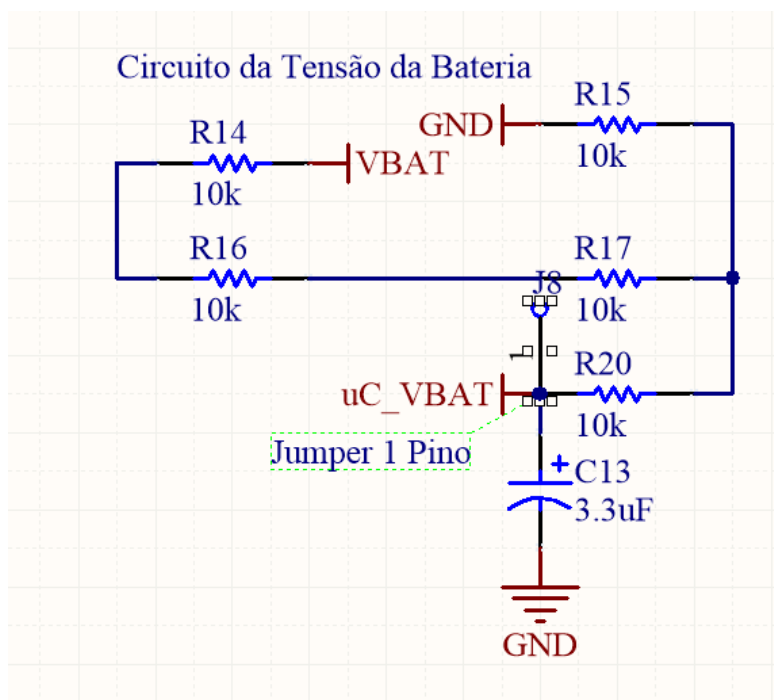
O PLM pode ser programado para gerar um sinal analógico, de 0V a 5V, proporcional ao fator lambda lido, desse modo o circuito projetado para esse sensor é semelhante ao apresentado na figura 28.

3.1.2.2.3 Circuito da medição da tensão da bateria.

Para a aferição da tensão da bateria, foi realizado um circuito de divisão de tensão, em que quatro resistores de $10\text{K } \Omega$ foram colocados em série. A tensão de saída do circuito divide a tensão da bateria em um quarto, possibilitando que tensões de até 20V da bateria sejam aferidas de maneira confiável.

O circuito implementado é apresentado na figura 30.

Figura 30 - Circuito para aferição da tensão da bateria implementado.

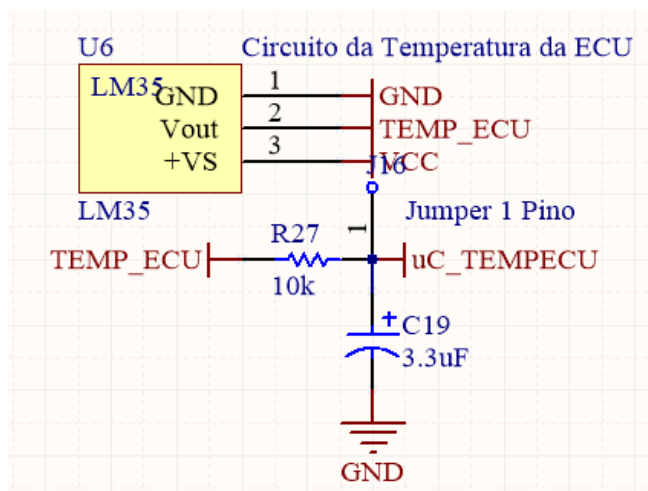


3.1.2.2.4 Circuito do sensor da temperatura da ECU

A fim de aferir a temperatura próximo a placa do ECU, foi utilizado o sensor de temperatura LM35. Este sensor tem a finalidade de monitorar a temperatura e detectar se esta está muito alta, o que pode ser indicio de mau funcionamento de algum circuito, ou sobrecarga de algum componente.

Desse modo, foi implementado o circuito apresentado na figura 31.

Figura 31 - Circuito para aferição da temperatura do ECU implementado.



3.1.3 Circuito de saída

3.1.3.1 Circuito da válvula injetora de combustível

O circuito do injetor de combustível consiste em acionar o injetor com uma corrente suficientemente grande para seu acionamento, seguindo a estratégia semi-sequencial de injeção de combustível, sabendo que a corrente máxima que as portas digitais do microcontrolador só conseguem fornecer 25mA.

O injetor utilizado neste projeto possui resistência de 15 Ohms, como o apresentado na figura 32.

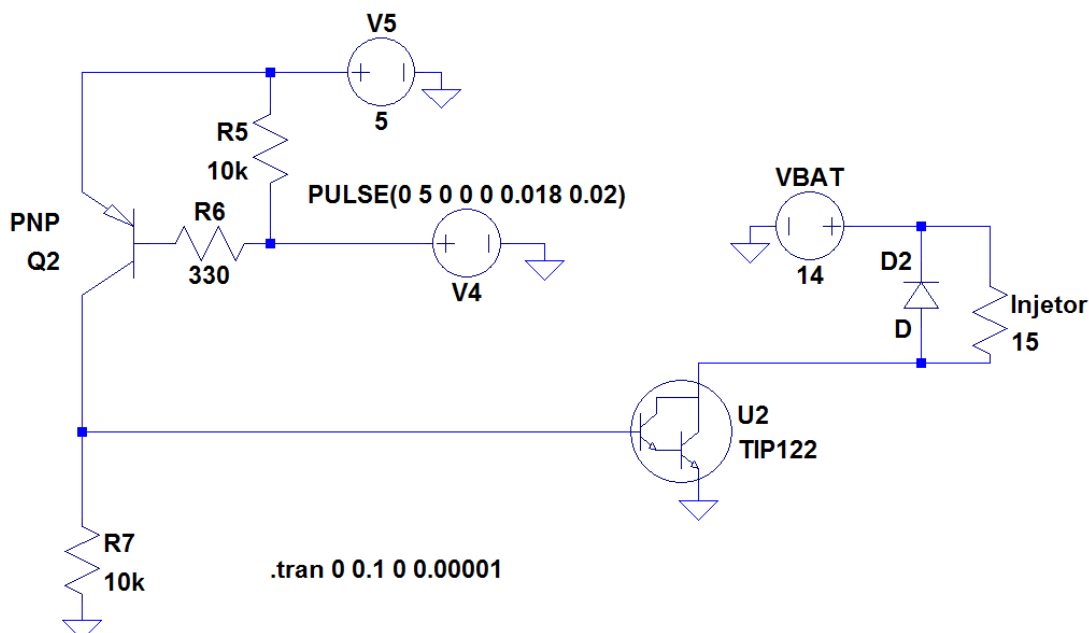
Figura 32 - Injetor de combustível utilizado nesse projeto.



Para o acionamento desse dispositivo, foi escolhido o componente TIP122, cuja corrente de funcionamento é de 5 A. Para que haja um isolamento entre o circuito de alta corrente e o do microcontrolador, o transistor bipolar PNP 2n3906 foi utilizado como

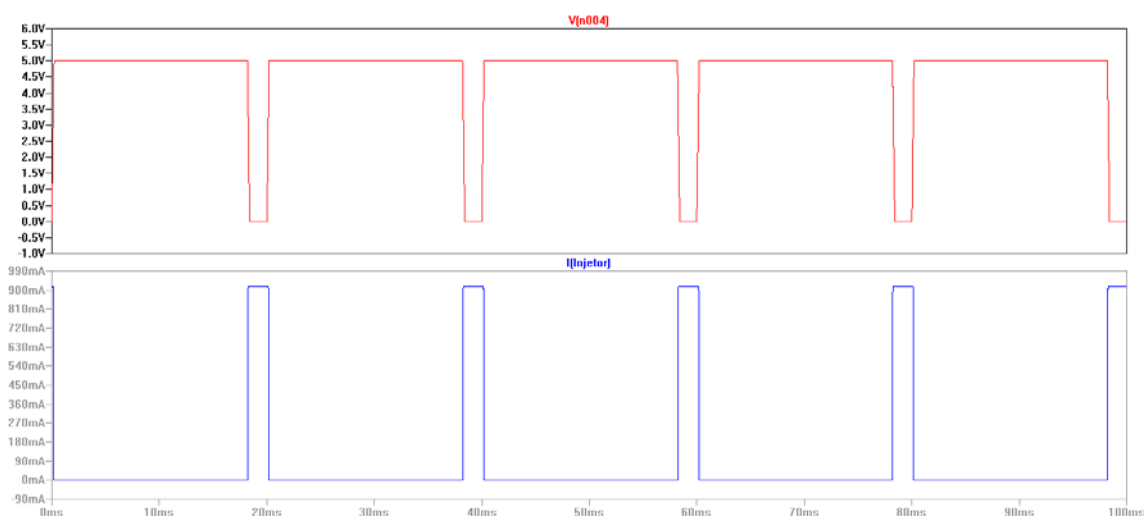
interface. Com isso, foi obtido o circuito para simulação apresentado na figura 33, em que o injetor de combustível foi representado por uma resistência de 15 Ω .

Figura 33 - Circuito do acionamento do injetor de combustível a ser simulado.



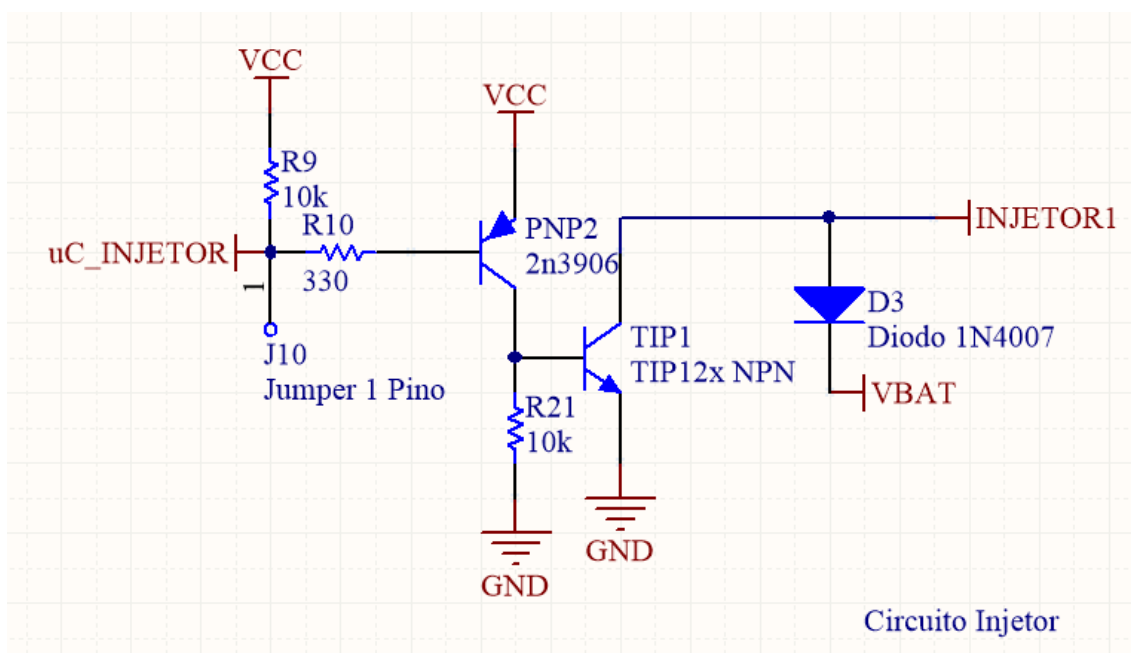
Para a simulação, foi gerada uma onda quadrada, cuja tensão varia de 0 a 5V, período corresponde a 0,02 segundos e o tempo do sinal em 5 V é de 0,018 segundos por período. Assim, as curvas da tensão de entrada e da corrente no injetor de combustível pelo tempo podem ser encontradas na figura 34.

Figura 34 - Resultado da simulação do circuito de acionamento do injetor de combustível. Em vermelho, a entrada do circuito. Em azul, a saída.



Na figura 35, nota-se que quando a tensão de entrada (V(n004)) está em 0V, a corrente que passa pelo injetor é próxima de 1A. Por outro lado, quando a tensão de entrada está em 5V, a corrente de saída está em 0A. Desse modo, com a escolha dos componentes e o circuito simulado, foi implementado o circuito que está ilustrado na figura 35.

Figura 35 - Circuito implementado do acionamento da válvula injetora de combustível.



3.1.3.2 Circuito da bobina de ignição

O circuito da bobina de ignição consiste no chaveamento de corrente para o enrolamento primário da bobina, seguindo uma estratégia de ignição com centelha perdida. Sabendo que a resistência desse componente é muito pequena (imensurável por multímetros convencionais), o componente eletrônico que realizará esse chaveamento deve ser muito robusto. Assim, para desempenhar esse papel, foi escolhido o transistor mosfet de potência, IRFB3004, cuja máxima corrente de dreno é de 195 A.

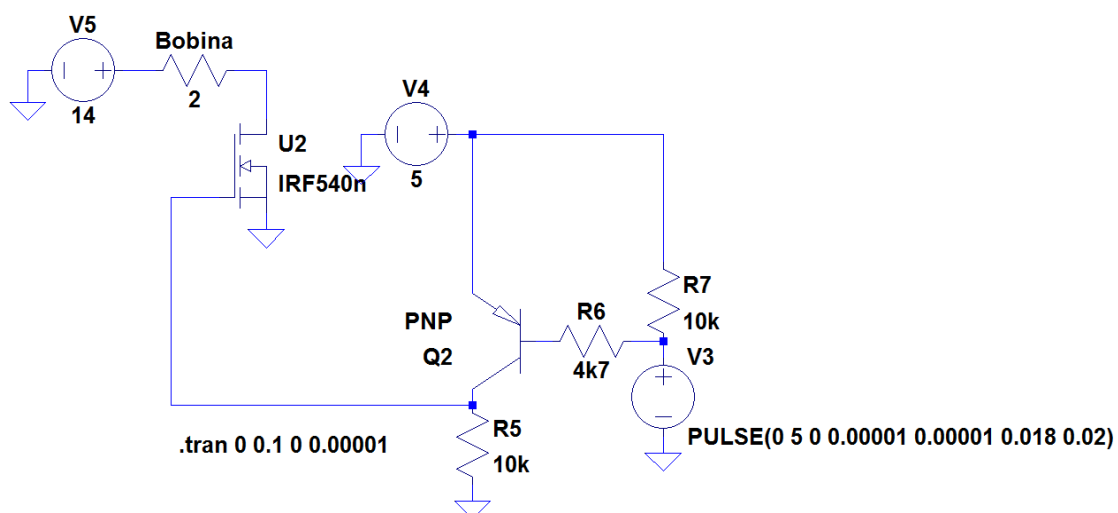
As bobinas utilizadas no projeto são das motos Honda CBR600RR, que são apresentadas na figura 36.

Figura 36 - Bobinas de ignição utilizadas nesse projeto.



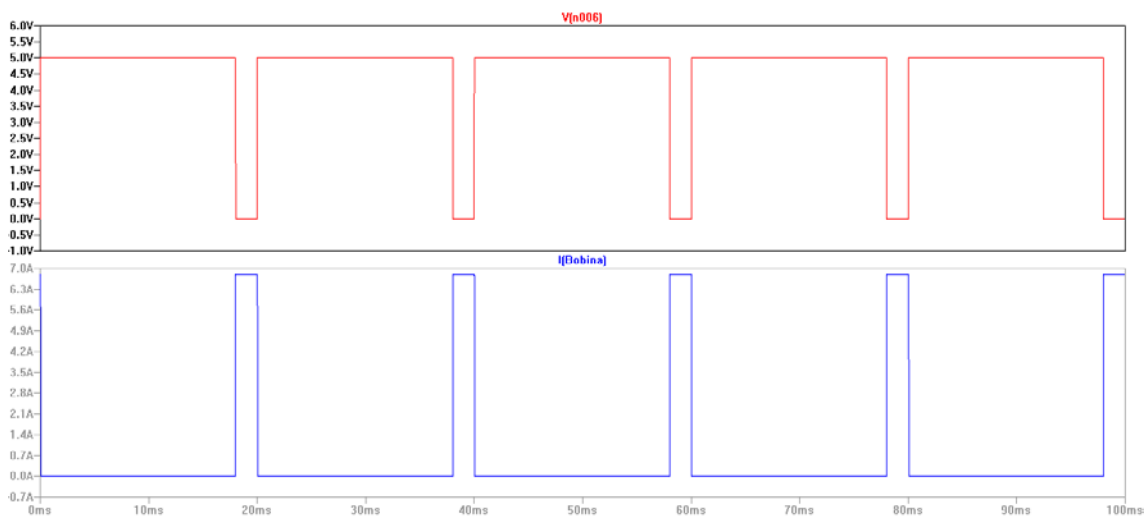
Para assegurar o funcionamento correto do circuito, este foi simulado a partir do esquemático apresentado na figura 37.

Figura 37 - Circuito do acionamento da bobina de ignição a ser simulado.



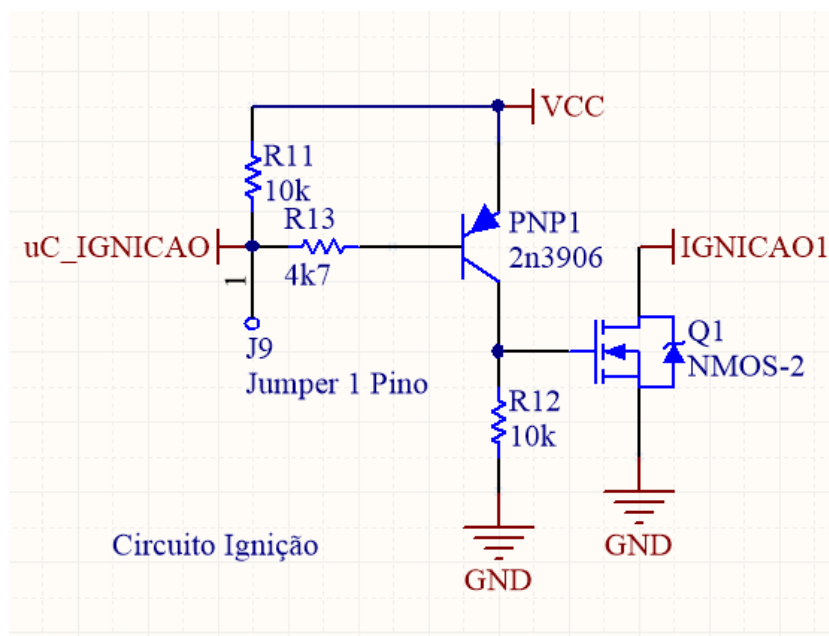
Nessa simulação, a resistência da bobina de ignição foi considerada 2 Ohms e o pulso gerado pelo microcontrolador para o acionamento do circuito possui período de 0,02 segundos, sendo que 0,018 segundos desse período, o pulso está com tensão de 5V. Como resultado da simulação, foram obtidas as curvas de tensão de entrada e corrente na bobina pelo tempo, como pode ser visto na figura 38.

Figura 38 - Resultado da simulação do circuito de acionamento da bobina de ignição. Em vermelho, a entrada do circuito. Em azul, a saída.



Desse modo, com a escolha dos componentes e o circuito simulado, foi implementado o circuito que está ilustrado na figura 39.

Figura 39 - Circuito implementado do acionamento da bobina de ignição.



3.1.4 Manufatura da placa de circuito impresso

A manufatura da placa da ECU foi realizada em uma fresadora, na qual as trilhas do circuito foram criadas por meio de uma fresa. Com todas as trilhas feitas, foi realizado um banho de estanho na placa de cobre, o que facilitou a soldagem dos componentes de maneira manual.

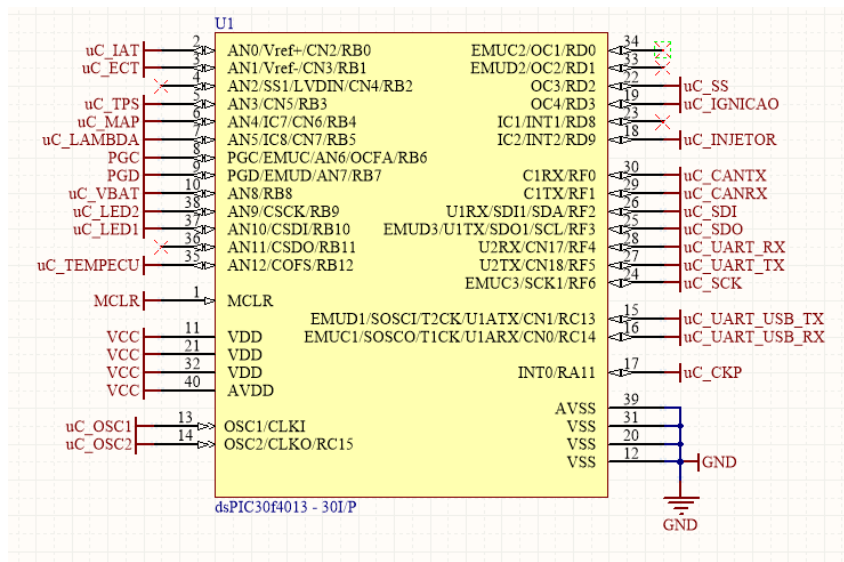
A alimentação da grande maioria dos componentes da placa se dá com uma tensão de 5V. Dessa maneira, os planos da placa são ligados a 5V e a terra, facilitando o roteamento desses pinos no projeto.

A fim de otimizar a etapa de depuração do projeto, a placa produzida possui 2 leds cuja função é indicar o funcionamento da placa como um todo, ou de alguma função específica no firmware. Além disso, foram colocados pinos nos sinais de entrada, nos sinais de saída dos circuitos e nos nos planos de terra e 5V, a fim de facilitar a utilização de instrumentos como osciloscópio e multímetros.

Visando a implementação em um veículo, os conectores utilizados são selados, impedindo a entrada de poeira e água por eles.

O microcontrolador e suas conexões são apresentados na figura 40.

Figura 40 - Conexões implementadas do microcontrolador.



3.2 Firmware

O firmware é o conjunto de instruções que vai embarcado em um equipamento eletrônico. Para uma ECU, essas instruções realizam cálculos de tempo para a geração dos pulsos de ignição de injeção e cálculo de compensações referente aos dados dos sensores que nele estão conectados, como o caso da tensão da bateria e temperaturas. Essas compensações não são implementadas neste projeto por se tratar de uma ECU que gera sinais de ignição com avanço fixo e sinais de injeção com tempo e fase fixos.

O código desenvolvido para esse projeto baseou-se no desenvolvimento por prototipação evolucionária, em que um protótipo inicial é produzido e refinado até que

todos os requisitos estejam cumpridos. A implementação foi realizada no ambiente de desenvolvimento MPLab X 3.26, da Microchip.

3.2.1 Interrupções

Interrupção é uma alteração no caminho normal do código, que ocorre quando um evento ocorre no microcontrolador. Entre esses eventos, os que possuem relação com este projeto são a mudança de estado de um sinal na porta digital, o passar de um determinado período de tempo, o recebimento de dados na interface serial e o término da conversão analógico para digital.

3.2.1.1 Interrupção de *Timer*

A interrupção de *timer* ocorre periodicamente em um determinado período de tempo previamente programado. O microcontrolador utilizado neste projeto consta com 5 timers, sendo que o *Timer 1*, *Timer 2* e *Timer 3* foram utilizados no projeto, com o código de suas interrupções apresentados abaixo.

3.2.1.1.1 *Timer 1*

A interrupção de *Timer 1* foi utilizada com a finalidade de depuração, fazendo com que um led azul mudasse seu estado em um período de 100ms. As rotinas utilizadas nessa função foram implementadas no código 1.

```

1 void      __attribute__((__interrupt__,      __auto_psv__))
2  _T1Interrupt(void)
3  {
4      IFS0bits.T1IF = 0;
5      LED_AZ = !LED_AZ;
6  }

```

Código 1 - Interrupção de Timer1.

3.2.1.1.2 *Timer 2*

A interrupção de *Timer 2* foi utilizada para incrementar variáveis contadoras de tempo. No caso, o tempo entre os dentes da roda fônica do virabrequim e o tempo de volta do motor. Para isso, a interrupção foi programada para ocorrer em um período de 10 μ s. Para uma roda fônica de 21 dentes, o valor escolhido abrange rotações de 92 a

6 milhões de rotações por minuto. Desse modo, a faixa de rotação de um motor real que pode chegar a 15.000 rotações por minuto, está dentro da margem teórica aplicada para essa interrupção. O código que corresponde a essas tarefas, se encontra no código 2.

```

1 void      __attribute__((__interrupt__,      __auto_psv__))
  _T2Interrupt(void)
2 {
3     IFS0bits.T2IF = 0;
4
5     u16_ctrl_tempoentredentes ++;
6     u16_ctrl_tempovolta ++;
7 }
8

```

Código 2 - Interrupção de Timer2.

3.2.1.1.3 *Timer 3*

A fim de determinar se o motor está em funcionamento ou não, foi implementado um contador que tem sua variável de cálculo de tempo zerada na interrupção externa 1, responsável pelo sinal do sensor de rotação. Assim, caso não seja detectado o sinal da rotação do motor num período de 100 ms, ocorre a interrupção de *timer 3*, que indica o não funcionamento do motor. Além disso, com a finalidade de depuração, quando o motor se encontra desligado, um led branco da placa muda seu estado. As instruções responsáveis por essas tarefas se encontram no código 3.

```

1 void      __attribute__((__interrupt__,      __auto_psv__))
  _T3Interrupt(void)
2 {
3     IFS0bits.T3IF = 0;
4     LED_BR = !LED_BR;
5     u08_ctrl_motorligado = 0;
6 }

```

Código 3 - Interrupção de Timer3.

3.2.1.1.4 Interrupção externa

A interrupção externa ocorre quando há uma mudança de nível na porta digital INT0. Para esse projeto, a porta INT0 foi utilizada para detectar o pulso gerado pelo circuito do sensor de rotação, sendo acionada pela detecção da borda de descida do sinal.

A implementação desse projeto visa que a roda fônica do virabrequim tenha um referencial para a detecção de uma volta, podendo essa referência ser um dente faltante. Desse modo, tendo o tempo entre os dentes dessa roda, quando esse dobra é detectado a falha. Com a detecção de uma volta, esse tempo é armazenado como sendo o tempo da volta que acabou de ocorrer (tempo da volta anterior), e o contador do tempo da volta é zerado a fim de se contar a volta que acabou de começar.

Independentemente da falha ser encontrada, o tempo entre os dentes da roda fônica é armazenado da mesma maneira nesse momento.

Como apresentado no item dedicado à interrupção de *Timer 3*, nesse momento a variável responsável pelo cálculo do *Timer 3* é zerada e a variável que indica o funcionamento do motor recebe o valor 1, indicando que o motor está em funcionamento.

As instruções que representam as tarefas indicadas acima são apresentadas no código 4.

```

1 void __attribute__((__interrupt__, __auto_psv__))
  _INT0Interrupt(void)
2 {
3     IFS0bits.INT0IF = 0;
4
5     if(u16_ctrl_tempoentredentes >
      (2*u16_ctrl_tempoanterior_cfalha) )
6     {
7         u16_ctrl_tempoanterior_volta = u16_ctrl_tempovolta;
8         u16_ctrl_tempovolta = 0;
9     }
10
11     u16_ctrl_tempoanterior_cfalha = u16_ctrl_tempoentredentes;
12     u16_ctrl_tempoentredentes = 0;
13
14     //Verificação se o motor está ligado
15     u08_ctrl_motorligado = 1;
16     //TMR3 = 7839;
17     TMR3 = 0;
18 }

```

Código 4 - Interrupção externa.

3.2.1.1.5 Interrupção de recebimento de dados na porta serial

A interrupção referente ao recebimento de dados pela porta serial tem como finalidade executar o primeiro tratamento dos dados recebidos. Para esse projeto, essa

interrupção é utilizada para receber os comandos que o *software* desenvolvido em LabView enviará, via interface USB, para a ECU.

Os comandos recebidos são bytes baseados na representação do código ASCII. Sendo assim, o caracter 'D' solicita que a ECU envie os dados referentes aos seus parâmetros para o *software*, o caracter 'E' solicita que os dados referentes aos parâmetros sejam gravados na memória E2PROM, já o caracter 'U' inicia o modo de atualização dos dados dos parâmetros, que faz com que a ECU espere por mais 8 bytes de configuração.

Com o comando 'D', a ECU envia ao *software* os seguintes 22 bytes: tempo de injeção, fase da injeção (*byte* mais e menos significativo), tempo de carga da bobina de ignição, avanço da centelha (*byte* mais e menos significativo), diferença entre a falha da roda fônica e o PMS (*byte* mais e menos significativo), valor referente ao sensor IAT (*Intake Air Temperature*) (*byte* mais e menos significativo) , valor referente ao ECT (*Engine Coolant Temperature*) (*byte* mais e menos significativo), valor referente ao TPS (*byte* mais e menos significativo), valor referente ao MAP (*byte* mais e menos significativo), valor referente ao valor de lambda (*byte* mais e menos significativo), valor referente à tensão da bateria (*byte* mais e menos significativo) e o valor referente à temperatura do ECU (*byte* mais e menos significativo).

Com o comando 'U', a ECU espera receber os seguintes 8 bytes: tempo de injeção, fase da injeção (*byte* mais e menos significativo), tempo de carga da bobina, avanço da centelha (*byte* mais e menos significativo) e o valor referente a diferença entre a falha da roda fônica e o PMS (*byte* mais e menos significativo).

Para ambos os comandos 'U' e 'E', a ECU envia a mensagem "OK!" para a confirmação para confirmação do recebimento desses comandos.

O comando utilizado para a implementação da interrupção por recebimentos de mensagens na porta serial se encontra no código 5.

```

1 void      __attribute__((__interrupt__,      __auto_psv__))
  _U1RXInterrupt(void)
2 {
3     IFS0bits.U1RXIF = 0;
4
5     if(! u08_flg_modorecebe){
6         switch (U1RXREG){
7             case 'D':
8                 EnviaDadosUSB();
9                 break;
10            case 'U':
11                u08_flg_modorecebe = 1;

```

```

12         u08_cnt_modorecebe = 0;
13         break;
14     case 'E':
15         SalvarDadosE2PROM();
16         WriteUART1_U08('O');
17         WriteUART1_U08('K');
18         WriteUART1_U08('!');
19         break;
20     default:
21         break;
22     }
23 }
24 else
25 {
26     switch (u08_cnt_modorecebe){
27         case 0:
28             u08_prmt_tempoinjecao = U1RXREG;
29             u08_cnt_modorecebe++;
30             break;
31         case 1:
32             u16_prmt_faseinjecao = U1RXREG;
33             u16_prmt_faseinjecao = u16_prmt_faseinjecao<<8;
34             u08_cnt_modorecebe++;
35             break;
36         case 2:
37             u16_prmt_faseinjecao = u16_prmt_faseinjecao |
U1RXREG;
38             u08_cnt_modorecebe++;
39             break;
40         case 3:
41             u08_prmt_dwellignicao = U1RXREG;
42             u08_cnt_modorecebe++;
43             break;
44         case 4:
45             s16_prmt_avancoignicao = U1RXREG;
46             s16_prmt_avancoignicao =
47 s16_prmt_avancoignicao<<8;
48             u08_cnt_modorecebe++;
49             break;
50         case 5:
51             s16_prmt_avancoignicao = s16_prmt_avancoignicao |
U1RXREG;
52             u08_cnt_modorecebe++;
53             break;
54         case 6:
55             u16_rmt_refPMS = U1RXREG;
56             u16_rmt_refPMS = u16_rmt_refPMS<<8;
57             u08_cnt_modorecebe++;
58             break;
59         case 7:
60             u16_rmt_refPMS = u16_rmt_refPMS | U1RXREG;
61             u08_cnt_modorecebe++;
62             break;
63     default:

```

```

62         break;
63     }
64
65     if(u08_cnt_modorecebe == 8)
66     {
67         u08_flg_modorecebe = 0;
68         WriteUART1_U08('O');
69         WriteUART1_U08('K');
70         WriteUART1_U08('!');
71     }
72 }
73 }

```

Código 5 - Interrupção por recebimento de mensagens na porta serial.

3.2.1.2 Funções para injeção e ignição.

Para que as ondas responsáveis pelo injetor de combustível e pela bobina sejam geradas, as funções “void injetor_tempo(unsigned char tempo, unsigned int fase)” e “void ignicao_avanco(unsigned char dwell, int avanco)” são chamadas na função principal (*main*).

A função “injetor_tempo”, é responsável por gerar o pulso de injeção a partir dos parâmetros de duração do pulso (“tempo”) e a fase da injeção, que corresponde ao momento que ocorrerá o término da injeção de combustíveis (“fase”). Já a função “ignicao_avanco”, gera o pulso a partir do tempo de carga da bobina (“dwell”) e o avanço da centelha (“avanco”), que determina quando a carga da bobina terminará, acarretando na centelha na vela de ignição.

Para a função referente ao pulso da ignição, o valor do avanço remete-se a um valor em graus em que a centelha deve ocorrer antes do PMS. Desse modo, primeiramente é realizada uma conversão a fim de transformar o valor do avanço na posição que a centelha deve ocorrer após o PMS. Para o caso de um valor de avanço de 10° antes do PMS, a função converte esse valor para a posição 350° depois do PMS, por exemplo.

Em ambos os casos, os parâmetros “fase” e “avanco” são ajustados a fim de considerarem o valor da diferença da falha da roda fônica e o PMS, desse modo, o referencial para início e término de uma volta do motor continua sendo a falha na roda fônica.

Após os ajustes e correções citados anteriormente, a função a partir do tempo da volta anterior calcula qual o momento, em períodos de tempo, que o pulso deverá iniciar e terminar o acionamento do componente a ser controlado. Com isso, duas

opções são possíveis: o tempo para o início é inferior ao tempo para o término do acionamento e o tempo para o término é inferior ao tempo para o início do acionamento do componente. Para este último caso, a parte ativa do pulso ocorre em uma volta do motor e termina em outra.

Para cada caso citado anteriormente, ocorre uma verificação em relação ao contador de tempo da volta corrente e os tempos de início e fim do acionamento desses componentes. As instruções referentes à função geradora do pulso de injeção se encontram no código 6. Já os comandos referentes à função geradora do pulso de ignição se encontram no código 7.

```

1 void injetor_tempo(unsigned char tempo, unsigned int fase){
2
3     float          aux_liga = 0,
4                   aux_desliga = 0;
5     unsigned int   aux_tempo = 0,
6                   aux_fase = 0;
7
8     //Calculo da fse com a nova referencia
9     if (fase + u16_rmt_refPMS > 360)
10        aux_fase = fase + u16_rmt_refPMS - 360;
11     else
12        aux_fase = fase + u16_rmt_refPMS;
13
14     //aux_desliga                                     =
15     ((float)u16_ctrl_tempoanterior_volta/360)*((float)fase);
16     aux_desliga                                     =
17     ((float)u16_ctrl_tempoanterior_volta/360)*((float)aux_fase);
18     aux_liga = u16_ctrl_tempoanterior_volta - ( ((unsigned
19     int)tempo)*100 - aux_desliga);
20     aux_tempo = tempo*100;
21
22     // Pulso se mantém dentro de uma volta
23     // 0 ----- Liga ----- Desliga ----- 360/0
24     if(aux_tempo <= aux_desliga){
25         //Posicao do "Liga"
26         aux_liga = aux_desliga - ( ((unsigned int)tempo)*100 );
27         if( (u16_ctrl_temповolta >= aux_liga) &&
28            (u16_ctrl_temповolta < aux_desliga) )
29             //PORTDbits.RD1 = 0;
30             INJ_1 = INJ_ON;
31         else
32             //PORTDbits.RD1 = 1;
33             INJ_1 = INJ_OFF;
34     }
35     //Pulso começa em uma volta e acaba em outra
36     // 0 ----- Liga ----- 360/0 ----- Desliga -----
37     else{

```

```

34     //Posicao do "Liga"
35     aux_liga = u16_ctrl_tempoanterior_volta - ( ((unsigned
int)tempo)*100 - aux_desliga);
36     if(      (u16_ctrl_tempovolta      >=      aux_liga)      ||
(u16_ctrl_tempovolta < aux_desliga) )
37         //PORTDbits.RD1 = 0;
38         INJ_1 = INJ_ON;
39     else
40         //PORTDbits.RD1 = 1;
41         INJ_1 = INJ_OFF;
42     }
43 }

```

Código 6 - Função geradora do pulso da injeção de combustível.

```

1 void ignicao_avanco(unsigned char dwell, int avanco){
2
3     float          aux_liga = 0,
4                   aux_desliga = 0;
5     unsigned int   aux_tempo = 0,
6                   aux_avanco = 0;
7
8     if(avanco >= 0)
9         aux_avanco = 360 - avanco;
10    else
11        aux_avanco = abs(avanco);
12
13    //Calculo da fse com a nova referencia
14    if (aux_avanco + u16_rmt_refPMS > 360)
15        aux_avanco = aux_avanco +u16_rmt_refPMS - 360;
16    else
17        aux_avanco = aux_avanco + u16_rmt_refPMS;
18
19    aux_desliga =
20    ((float)u16_ctrl_tempoanterior_volta/360)*((float)aux_avanco);
21    aux_liga = u16_ctrl_tempoanterior_volta - ( ((unsigned
int)dwell)*100 - aux_desliga);
22    aux_tempo = dwell*100;
23
24    // Pulso se mantém dentro de uma volta
25    // 0 ----- Liga ----- Desliga ----- 360/0
26    if(aux_tempo <= aux_desliga){
27        //Posicao do "Liga"
28        aux_liga = aux_desliga - ( ((int)dwell)*100 );
29        if(      (u16_ctrl_tempovolta      >=      aux_liga)      &&
(u16_ctrl_tempovolta < aux_desliga) )
30            IGN_1 = IGN_OFF;
31        else
32            IGN_1 = IGN_ON;
33    }
34    //Pulso começa em uma volta e acaba em outra
35    // 0 ----- Liga ----- 360/0 ----- Desliga -----
else{

```

```

36     //Posicao do "Liga"
37     aux_liga = u16_ctrl_tempoanterior_volta - ( ((unsigned
int)dwel1)*100 - aux_desliga);
38     if(      (u16_ctrl_tempovolta      >=      aux_liga)      ||
(u16_ctrl_tempovolta < aux_desliga) )
39         IGN_1 = IGN_OFF;
40     else
41         IGN_1 = IGN_ON;
42     }
43 }

```

Código 7 - Função geradora do pulso da bobina de ignição.

3.2.1.3 Função *main* (Principal)

A principal atribuição da função *main* é verificar a variável indicadora de que o motor está em funcionamento, chamando as funções de acionamento do injetor de combustível e da bobina de ignição. Além dessa atribuição, é na função principal que os dados armazenados na memória E2PROM são obtidos e a configuração do modo de funcionamento dos periféricos do microcontrolador é definida, como as portas digitais, os contadores de tempo, a porta serial, a porta da *Serial Peripheral Interface*, a interrupção externa e o conversor A/D,

Para o caso de o motor estar funcionando, além do acionamento dos dispositivos de saída, a comunicação com a serial é desabilitada, a fim de privilegiar o funcionamento do motor em relação aos periféricos de comunicação. Já para o caso do motor não estar funcionando, essa comunicação volta a ser estabelecida, os sinais dos pulsos do injetor de combustível e da bobina de ignição são desligados e as variáveis que armazenam os tempos necessários para a geração desses pulsos são zeradas.

Os comandos relacionados à função principal se encontram no código 8.

```

1  int main(int argc, char** argv) {
2
3     //Ports I/O
4     InitPortas();
5
6     //Timer1
7     InitTimer();
8
9     //UART 1
10    InitSerial();
11
12    //SPI
13    InitSPI();
14
15    //Interrupcao Externa 1

```

```

16  InitExt()
17
18  //ADC
19  InitADC();
20
21  LerDadosE2PROM();
22
23  while(1){
24
25      if(u08_ctrl_motorligado){
26
27          HabilitaIntSerial(0);
28          injetor_tempo(u08_prmt_tempoinjecao,
29 u16_prmt_faseinjecao);
30
31          ignicao_avanco(u08_prmt_dwellignicao,
32 s16_prmt_avancoignicao);
33
34      }
35      else{
36          //Habilita serial quando o motor esta rodando
37          HabilitaIntSerial(1);
38
39          INJ_1 = INJ_OFF;
40          IGN_1 = IGN_OFF;
41          u16_ctrl_tempoentredentes = 0;
42          u16_ctrl_tempoanterior_cfalha = 0;
43          u16_ctrl_temповolta = 0;
44          u16_ctrl_tempoanterior_volta = 0;
45          u08_ctrl_motorligado = 0;
46      }
47
48  }
49  return (EXIT_SUCCESS);
50 }

```

Código 8 - Função principal.

3.3 Software

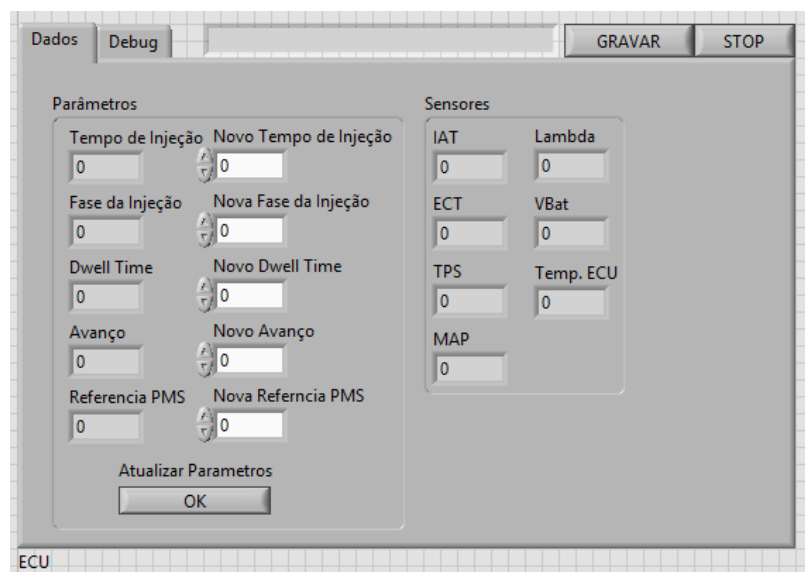
O software desenvolvido tem como função ser uma interface amigável entre o ECU e o usuário, possibilitando que este veja e altere os parâmetros do motor sem a necessidade de alterar o *firmware* apresentado anteriormente. Essa solução foi desenvolvida na plataforma LabVIEW, uma vez que esta possibilita a uma implementação mais rápida integrada com uma interface gráfica.

O código desenvolvido para esse projeto baseou-se no desenvolvimento por prototipação evolucionária, em que um protótipo inicial é produzido e refinado até que todos os requisitos estejam cumpridos.

3.3.1 Funcionamento

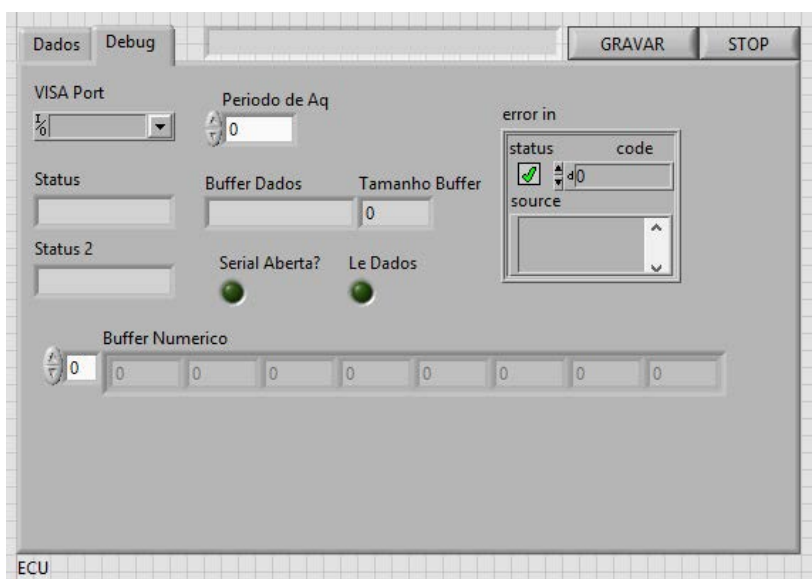
O programa se divide em duas abas. Na primeira, "Dados", os valores dos parâmetros e sensores são apresentados e há campos para que novos valores dos parâmetros sejam inseridos ao pressionar o botão "ATUALIZAR", como pode ser visto na figura 41.

Figura 41 - Aba "Dados" do software.



Já na outra aba, "Debug", são apresentados dados de controle do programa, como o nome da porta serial, o período de aquisição dos dados, buffers dos dados recebidos na serial, indicadores do estado atual das máquinas de estados implementadas, mensagem de erro do barramento serial e indicadores da leitura de dados, como é apresentado na figura 42.

Figura 42 - Aba "Debug" do software.



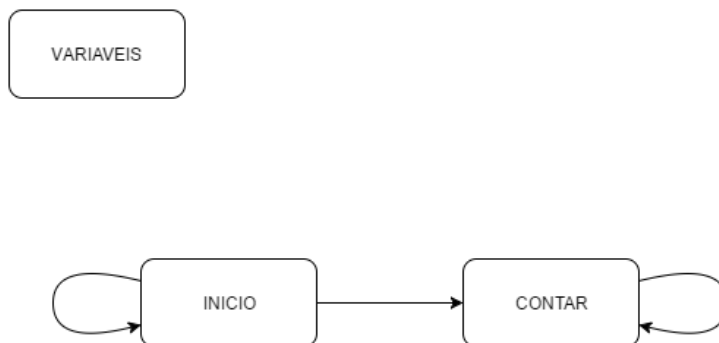
Além disso, há um campo onde são apresentadas mensagens sobre o funcionamento do programa, um botão “GRAVAR” para o armazenamento dos dados do ECU na memória E2PROM e um botão “STOP” que finaliza o funcionamento do programa de maneira segura.

3.3.2 Desenvolvimento

A desenvolvimento desse *software* utilizou como base máquinas de estado, em que cada estado foi implementado com pequenas tarefas muito bem definidas. Foram implementadas duas máquinas, sendo que uma é responsável pela contagem do período entre o envio do comando para o recebimento de mensagens, e a outra é responsável pelo funcionamento do restante do programa.

O funcionamento da primeira máquina se encontra na figura 43.

Figura 43 - Diagrama de estado para o cálculo do período do recebimento de mensagens na porta serial.



- **VARIABLES**

Estado em que os controladores e os indicadores do projeto são organizados.

- **INICIO**

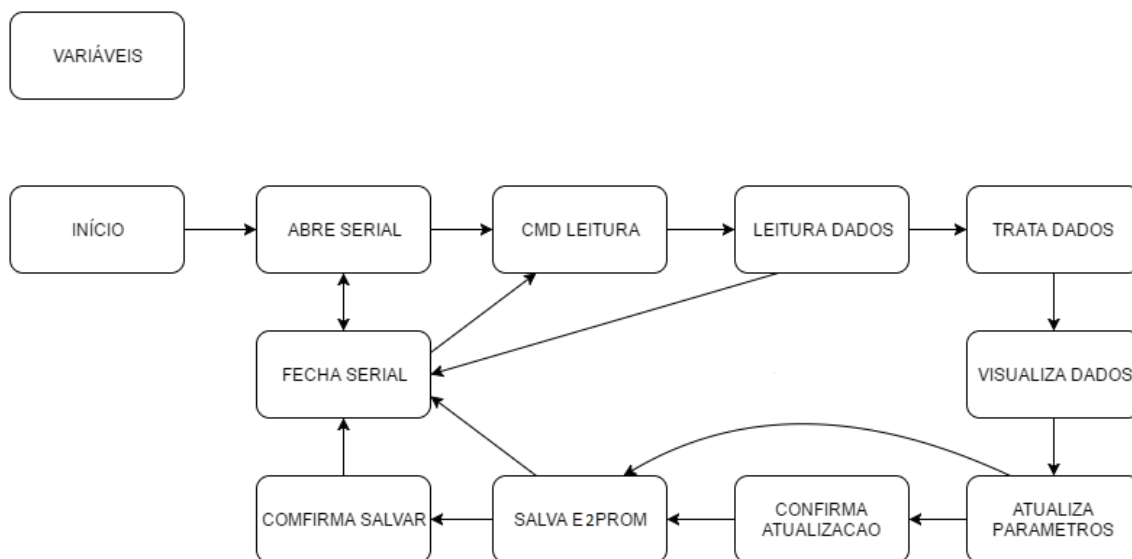
O período de aquisição dos dados é atualizado e caso o botão “STOP” não esteja pressionado, é invocado o estado “CONTAR”, caso contrário, fica nesse estado.

- **CONTAR**

Ocorre uma pausa com o período definido no estado “INICIO”, e a variável “Le Dados” é definida como verdadeira. A funcionalidade desse estado se repete até o botão “STOP” ser pressionado.

O funcionamento da segunda máquina se encontra na figura 44.

Figura 44 - Diagrama de estados do programa principal.



- **VARIAVEIS**

Estado em que os controladores e os indicadores do projeto são organizados.

- **INICIO**

As condições iniciais das variáveis utilizadas no projeto são definidas. O estado “ABRE SERIAL” é chamado.

- **ABRE SERIAL**

O nome da porta serial é verificado. Caso este for válido, a porta serial é aberta e o estado “CMD LEITURA” é chamado. Caso contrário, o estado “Fecha SERIAL” é chamado e um aviso é enviado ao usuário.

- **CMD LEITURA**

Caso a variável “Le Dados” for verdadeira, o comando ‘U’ é enviado ao ECU e o estado “LEITURA DADOS” é chamado. Caso contrário, o estado “FECHA SERIAL” é chamado.

- **LEITURA DADOS**

São lidos os 22 bytes recebidos do ECU e armazenados em um buffer do tipo String. Caso não for recebido nenhum dado, um aviso é enviado ao usuário. O estado “TRATA DADOS” é chamado.

- **TRATA DADOS**

O buffer com os dados recebidos é convertido do tipo String para um vetor com valores numéricos de 8 bits. O estado “VISUALIZA DADOS” é chamado.

- VISUALIZA DADOS

O vetor de bytes com os dados recebidos na serial tem seus valores divididos em seus respectivos indicadores. O estado “ATUALIZA PARAMETROS” é chamado.

- ATUALIZA PARAMETROS

Caso o botão “ATUALIZAR” for pressionado, o comando ‘U’ é enviado, junto com os 8 bytes esperados. Além disso, a cor do botão “GRAVAR” fica verde, indicando que o ECU possui parâmetros armazenados em memória volátil e o estado “CONFIRMA ATUALIZACAO” é chamado. Caso o botão “ATUALIZAR” não for pressionado, o estado “SALVA E2PROM” é chamado.

- CONFIRMA ATUALIZACAO

É esperado o recebimento dos caracteres “OK!” de confirmação. Caso estes venham, a mensagem “OK” é apresentada ao usuário. Caso contrário, a mensagem “FALHOU” é apresentada ao usuário. Além disso, o estado “SALVA E2PROM” é chamado.

- SALVA E2PROM

Caso o botão “SALVAR” seja pressionado, o comando ‘E’ é enviado ao ECU e o estado “CONFIRMA SALVAR” é chamado. Caso contrário, o estado “FECHA SERIAL” é chamado.

- CONFIRMA SALVAR

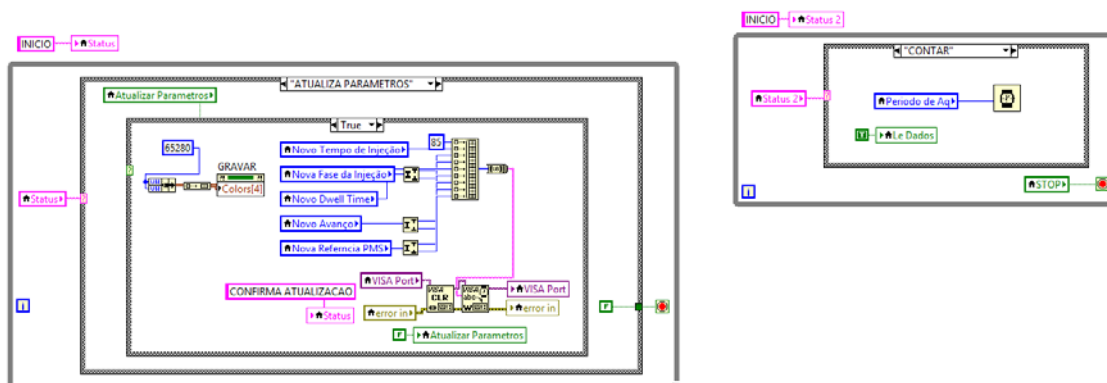
É esperado o recebimento dos caracteres “OK!” de confirmação. Caso estes venham, a mensagem “OK” é apresentada ao usuário. Caso contrário, a mensagem “FALHOU” é apresentada ao usuário. Além disso, o estado “FECHA SERIAL” é chamado.

- FECHA SERIAL

É verificado se o botão “STOP” foi pressionado ou se houve erro nas funções com a porta serial. Caso algo ocorreu, a serial é fechada e a execução do programa se encerra. Caso contrário, é verificado se a porta serial foi aberta anteriormente. Caso a porta serial esteja aberta, o estado “CMD LEITURA” é chamado, caso contrário o estado “ABRE SERIAL” é chamado.

O código que representa o estado “ATUALIZA PARAMETROS” e “CONTAR” das duas máquinas de estado se encontra na figura 45.

Figura 45 - Código do *software* desenvolvido em LabVIEW.



4 TESTES, RESULTADOS E DISCUSSÕES

Os testes realizados visam certificar que o *hardware*, *firmware* e *software* estão desempenhando suas funções como foram projetadas. Para os sinais de controle da válvula injetora de combustível e para o controle da bobina de ignição foi verificado o sinal gerado com a alteração dos parâmetros tempo e fase de injeção, tempo de carga da bobina, avanço da bobina e a diferença do PMS para a falha da roda fônica. Já para comunicação entre *software* e *hardware*, foi verificado a influência desta com os sinais de saída.

Para obter um sinal referente à rotação do motor, foi utilizado um motor elétrico cuja rotação é constante em 1700 rpm. Em seu eixo, foi acoplado uma engrenagem de 21 dentes que teve um cortado. Com isso, foi utilizado um sensor indutivo automotivo para medir a variação do campo magnético gerado por essa engrenagem.

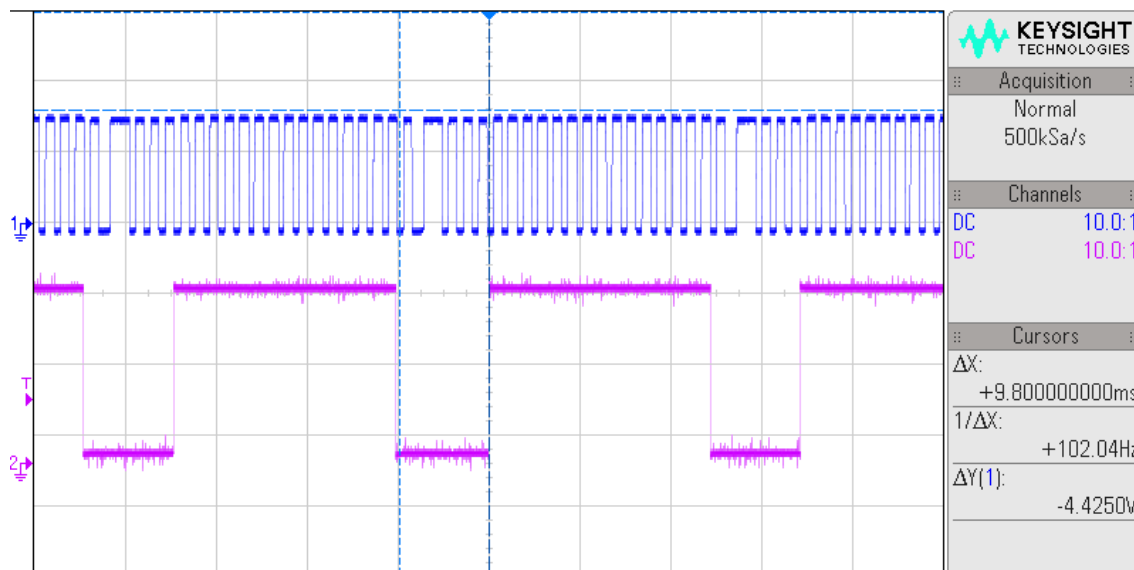
4.1 Sinal de controle do injetor de combustível

Para verificar que o sinal de controle do injetor de combustível pode ser gerado em todas as posições do motor, foram gerados pulsos que possuem nível baixo (tensão de 0V) fora do momento que ocorre a falha e no momento em que esta é detectada. Essas verificações foram realizadas alterando os parâmetros referentes à diferença do PMS e a falha da roda fônica, o tempo e fase da injeção de combustível.

Para todas as imagens provenientes do osciloscópio, a onda relativa ao canal 1 (azul) é o sinal proveniente do sensor de rotação na porta do microcontrolador. Já a onda do canal 2 (rosa) se refere ao pulso gerado para o injetor de combustível.

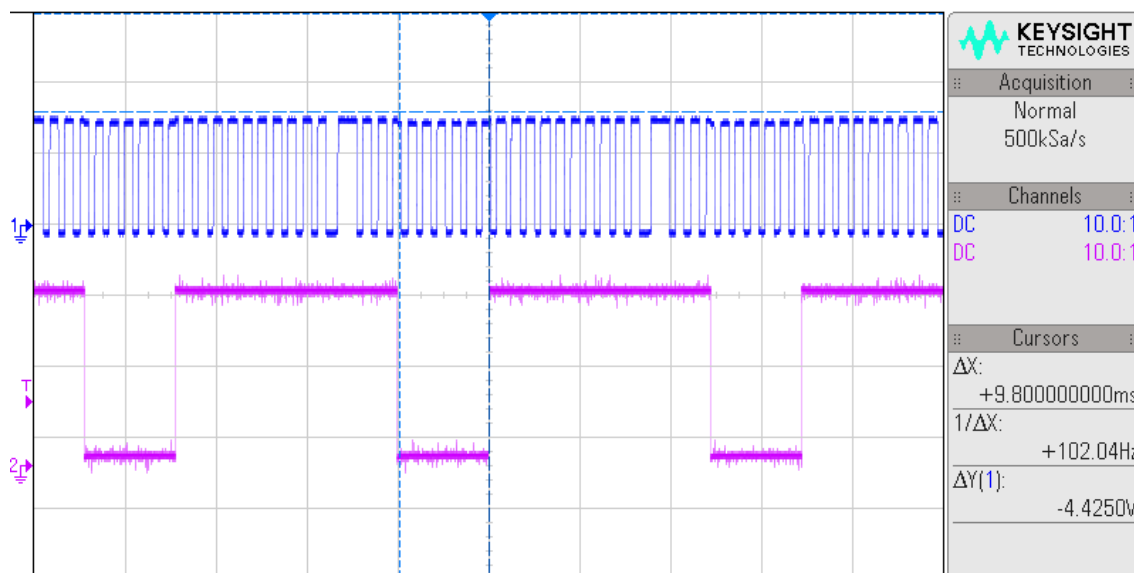
Para tempo de injeção de 10ms, fase de injeção de 50° e sem diferença entre o PMS e a falha na roda fônica, o nível baixo do pulso se encontra na região da falha. A imagem dessa onda pode ser vista na figura 46.

Figura 46 - Pulso da injeção de combustível com tempo de injeção de 10ms, fase de injeção de 50° e sem diferença entre o PMS e a falha da roda fônica.



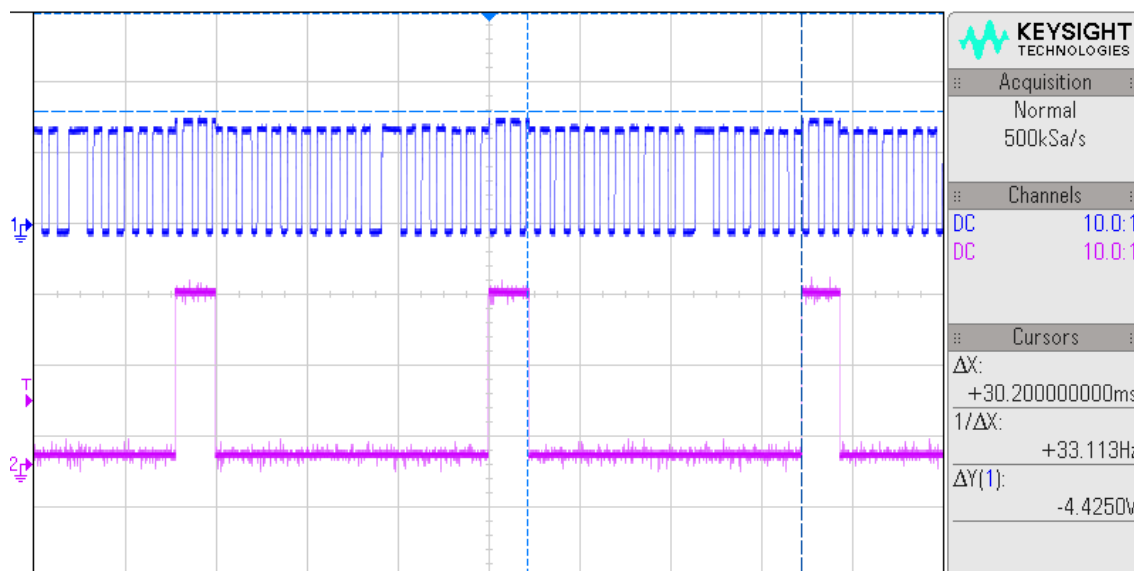
Para tempo de injeção de 10ms, fase de injeção de 150° e sem diferença entre o PMS e a falha na roda fônica, o nível baixo do pulso se encontra fora da região da falha. A imagem dessa onda pode ser vista na figura 47.

Figura 47 - Figura 45 - Pulso da injeção de combustível com tempo de injeção de 10ms, fase de injeção de 150° e sem diferença entre o PMS e a falha da roda fônica.



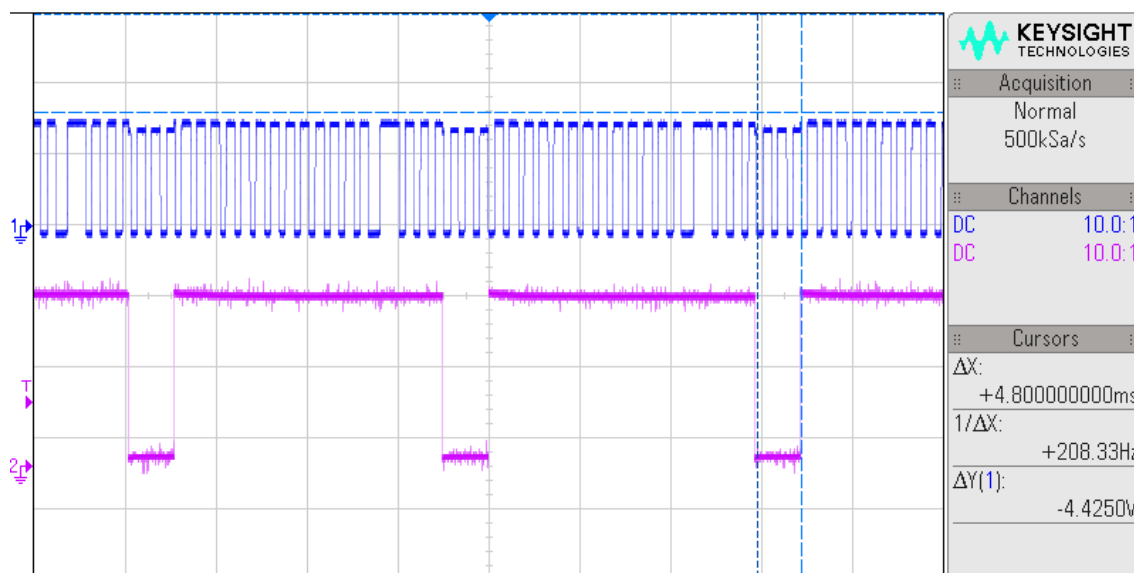
Para tempo de injeção de 30ms, fase de injeção de 100° e sem diferença entre o PMS e a falha na roda fônica, o nível baixo do pulso se encontra na região da falha. A imagem dessa onda pode ser vista na figura 48.

Figura 48 - Pulso da injeção de combustível com tempo de injeção de 30ms, fase de injeção de 100° e sem diferença entre o PMS e a falha da roda fônica.



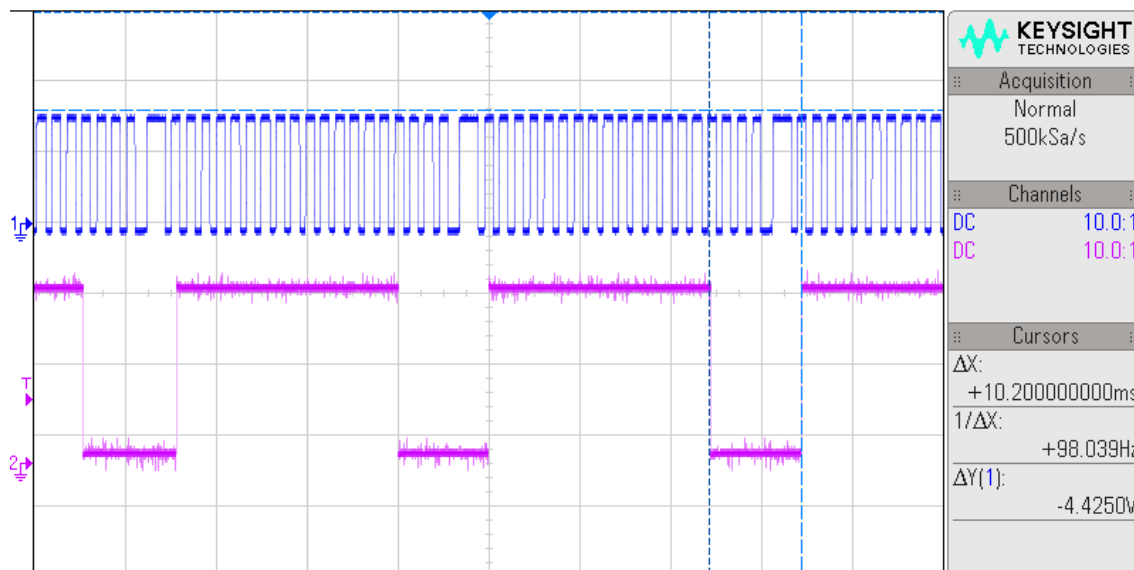
Para tempo de injeção de 5ms, fase de injeção de 100° e sem diferença entre o PMS e a falha na roda fônica, o nível baixo do pulso se encontra fora da região da falha. A imagem dessa onda pode ser vista na figura 49.

Figura 49 - Pulso da injeção de combustível com tempo de injeção de 5ms, fase de injeção de 100° e sem diferença entre o PMS e a falha da roda fônica.



Para tempo de injeção de 10ms, fase de injeção de 150° e 220° de diferença entre o PMS e a falha na roda fônica, o nível baixo do pulso se encontra fora da região da falha. A imagem dessa onda pode ser vista na figura 48.

Figura 50 - Pulso da injeção de combustível com tempo de injeção de 10ms, fase de injeção de 150° e 220° de diferença entre o PMS e a falha da roda fônica.



Com isso, foram realizadas medições de erro do sinal gerado em relação ao valor proposto. Para suprimir a oscilação que o motor elétrico possui, foram obtidas cinco amostras de cada análise e seu resultado foi considerado a média entre elas.

A fim de facilitar a demonstração gráfica dos dados, as tabelas seguintes terão o tempo de injeção será apresentado como sendo “Tempo Injeção”, a fase da injeção será apresentada como “Fase Injeção” e a diferença entre o PMS e a falha da roda fônica será apresentada como “Dif. PMS”.

Para o erro referente à variação do tempo de injeção, foram obtidos os dados da tabela 2.

Tabela 2 - Medições relativas ao parâmetro de tempo de injeção.

Entrada			Saída	
Tempo Injeção (ms)	Fase Injeção (°)	Dif. PMS (°)	Média do tempo de injeção (ms)	Erro (%)
1	10	0	0,96	4
5	10	0	5,12	2,4
10	10	0	10,06	0,6
15	10	0	15,06	0,4
20	10	0	20,16	0,8
25	10	0	25,16	0,64
30	10	0	30,12	0,4

Para o erro obtido com a variação da fase da injeção, foram obtidos os dados da tabela 3.

Tabela 3 - Medições relativas ao parâmetro da fase da injeção.

Entrada			Saída			
Tempo Injeção (ms)	Fase Injeção (°)	Dif. PMS (°)	Média do tempo da volta (ms)	Média do tempo da fase (ms)	Fase calculada (°)	Erro (%)
10	10	0	34,52	1,16	12,10	21
10	50	0	34,6	5,04	52,44	5
10	100	0	34,64	9,92	103,09	3
10	150	0	34,6	14,68	152,74	2
10	200	0	34,64	19,48	202,45	1
10	250	0	34,68	24,32	252,46	1
10	300	0	34,68	29,12	302,28	1
10	350	0	34,68	33,92	352,11	1

Para o erro obtido com a variação da diferença do PMS com a falha da roda fônica, foram obtidos os dados da tabela 4.

Tabela 4 - Medições relativas ao parâmetro da diferença entre o PMS e a falha da roda fônica na injeção.

Entrada			Saída			
Tempo Injeção (ms)	Fase Injeção (°)	Dif. PMS (°)	Média do tempo da volta (ms)	Média do tempo da fase (ms)	Fase calculada (°)	Erro (%)
10	50	10	34,6	5,88	51,18	2
10	50	50	34,64	9,92	53,09	6
10	50	100	34,6	14,64	52,32	5
10	50	150	34,64	19,44	52,03	4
10	50	200	34,6	24,28	52,62	5
10	50	250	34,6	29,08	52,57	5
10	50	300	34,64	33,92	52,52	5
10	50	350	34,64	38,76	52,82	6

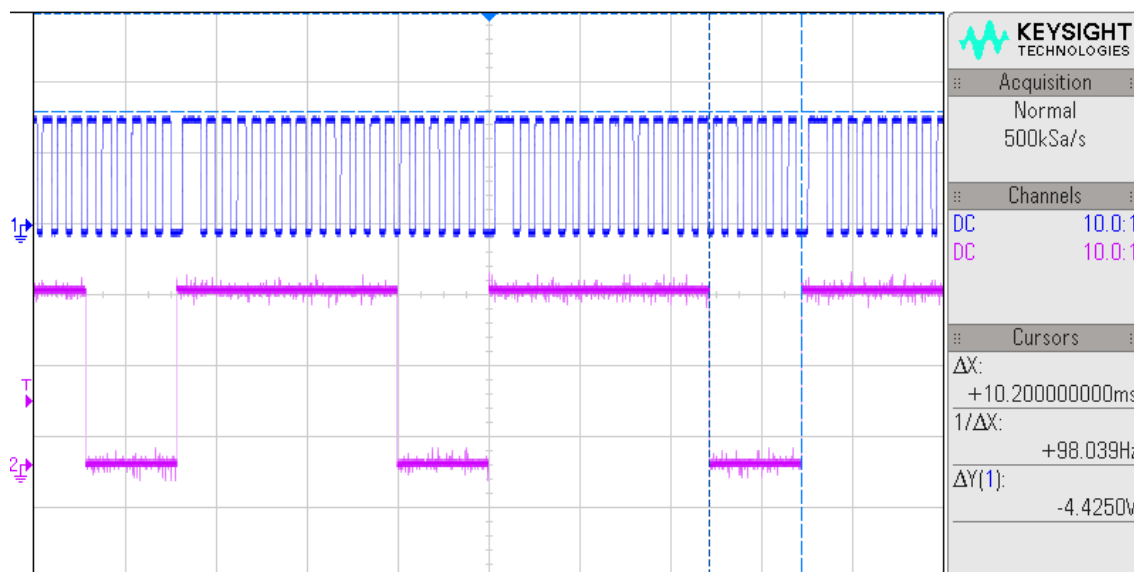
4.2 Sinal de controle da bobina de ignição

De modo semelhante aos dados que foram gerados com o sinal do injetor de combustível, foram gerados pulsos que possuem o nível baixo (tensão de 0V) fora do momento que ocorre a falha e no momento em que esta é detectada. Essas verificações foram realizadas alterando os parâmetros referentes à diferença do PMS e a falha da roda fônica, o tempo de carga da bobina "Dwell time" e o avanço da centelha.

Para todas as imagens provenientes do osciloscópio, a onda relativa ao canal 1 (azul) é o sinal proveniente do sensor de rotação na porta do microcontrolador. Já a onda do canal 2 (rosa) se refere ao pulso gerado para a para a bobina de ignição

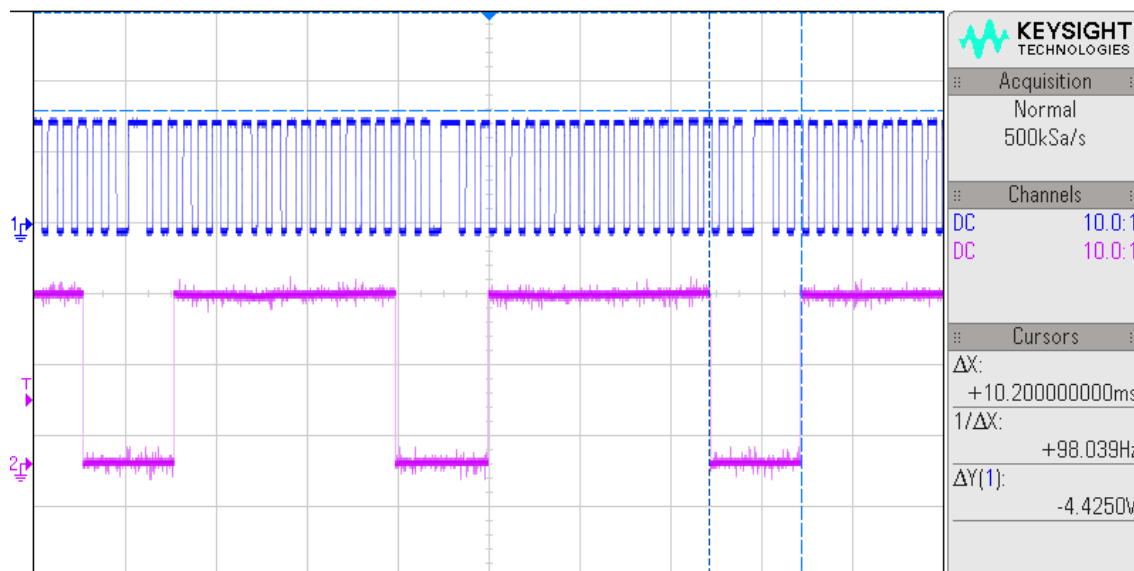
Para tempo de carga de bobina de 10ms, avanço de 30° e sem diferença entre o PMS e a falha na roda fônica, o nível baixo do pulso se encontra fora da região da falha. A imagem dessa onda pode ser vista na figura 51.

Figura 51 - Pulso da bobina de ignição com *dwell time* de 10ms, avanço de 30° e sem diferença entre o PMS e a falha da roda fônica.



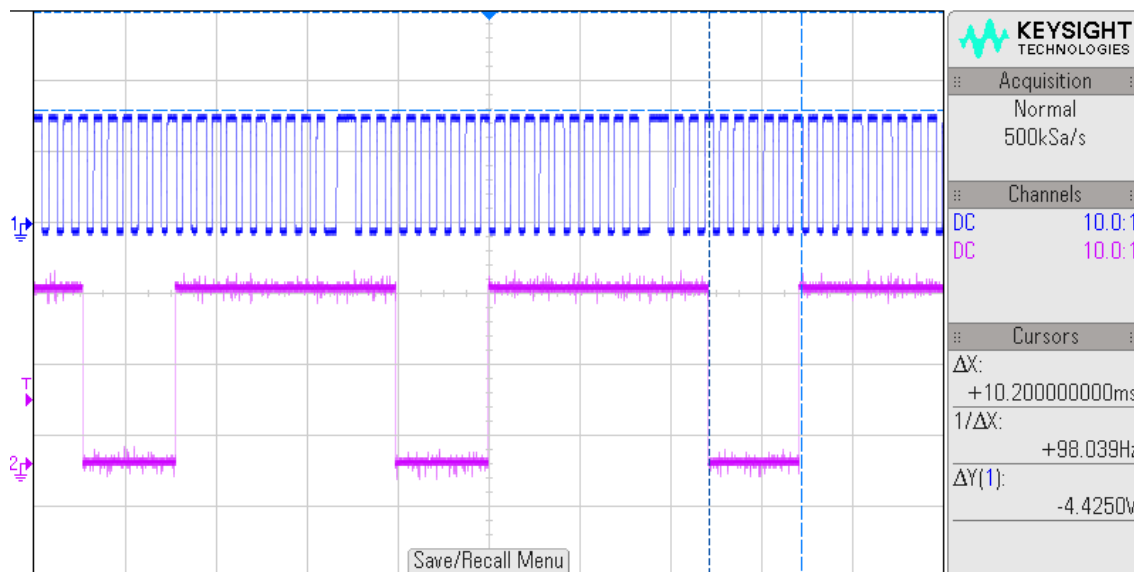
Para tempo de carga de bobina de 10ms, avanço de -30° e sem diferença entre o PMS e a falha na roda fônica, o nível baixo do pulso se encontra na região da falha. A imagem dessa onda pode ser vista na figura 52.

Figura 52 - Pulso da bobina de ignição com *dwell time* de 10ms, avanço de -30° e sem diferença entre o PMS e a falha da roda fônica.



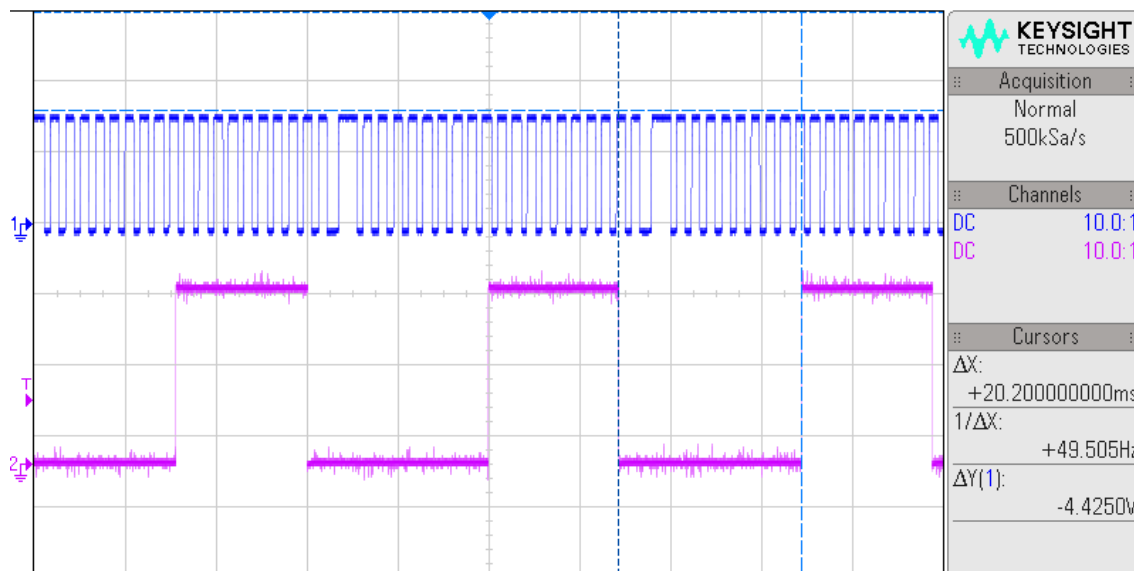
Para tempo de carga de bobina de 10ms, avanço de -150° e sem diferença entre o PMS e a falha na roda fônica, o nível baixo do pulso se encontra fora da região da falha. A imagem dessa onda pode ser vista na figura 53.

Figura 53 - Pulso da bobina de ignição com *dwell time* de 10ms, avanço de -150° e sem diferença entre o PMS e a falha da roda fônica.



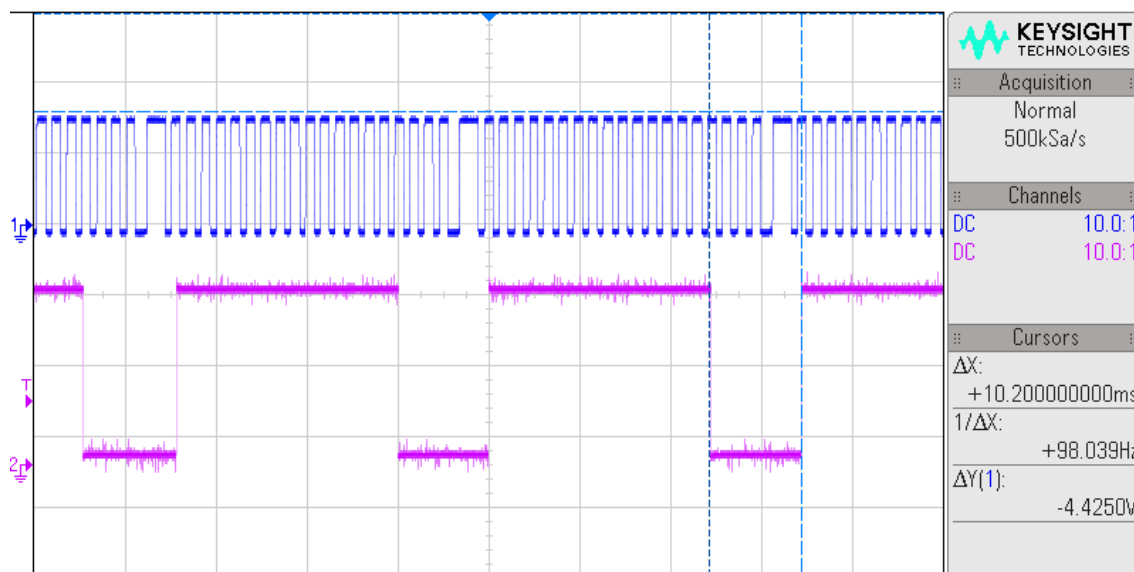
Para tempo de carga de bobina de 20ms, avanço de -150° e sem diferença entre o PMS e a falha na roda fônica, o nível baixo do pulso se empunha na região da falha. A imagem dessa onda pode ser vista na figura 54.

Figura 54 - Pulso da bobina de ignição com *dwell time* de 20ms, avanço de -150° e sem diferença entre o PMS e a falha da roda fônica.



Para tempo de carga de bobina de 10ms, avanço de 30° e com diferença de 35° entre o PMS e a falha na roda fônica, o nível baixo do pulso se empunha na região da falha. A imagem dessa onda pode ser vista na figura 55.

Figura 55 - Pulso da bobina de ignição com *dwell time* de 10ms, avanço de 30° e 35° de diferença entre o PMS e a falha da roda fônica.



Com isso, foram realizadas medições de erro do sinal gerado em relação ao valor proposto. Para suprimir a oscilação que o motor elétrico possui, foram obtidas cinco amostras de cada análise e seu resultado foi considerado a média entre elas.

Para o erro referente à variação do tempo de carga da bobina, foram obtidos os dados da tabela 5.

Tabela 5 - Medições relativas ao parâmetro de tempo de carga da bobina (*Dwell Time*) na ignição.

Entrada			Saída	
Dwell time (ms)	Avanço (°)	Dif. PMS (°)	Média do tempo de carga da bobina (ms)	Erro (%)
1	10	0	1,04	4
5	10	0	4,91	1,76
10	10	0	9,96	0,4
15	10	0	14,96	0,27
20	10	0	20,08	0,4
25	10	0	24,96	0,16
30	10	0	30,16	0,53

Para o erro obtido com a variação de avanço, foram obtidos os dados da tabela 6.

Tabela 6 - Medições relativas ao parâmetro de avanço na ignição.

Entrada			Saída			
Dwell time (ms)	Avanço (°)	Dif. PMS (°)	Média do tempo da volta (ms)	Média do tempo da fase (ms)	Avanço calculado (°)	Erro (%)
10	10	0	34,56	0,68	7,08	29
10	50	0	34,6	4,6	47,86	4
10	100	0	34,6	9,4	97,80	2
10	150	0	34,64	14,2	147,58	2
10	200	0	34,68	18,96	196,82	2
10	250	0	34,6	24,2	251,79	1
10	300	0	34,56	28,6	297,92	1
10	350	0	34,6	33,44	347,93	1

Para o erro obtido com a variação da diferença do PMS com a falha da roda fônica, foram obtidos os dados da tabela 7.

Tabela 7 - Medições relativas ao parâmetro da diferença entre o PMS e a falha da roda fônica na ignição.

Entrada			Saída			
Dwell time (ms)	Avanço (°)	Dif. PMS (°)	Média do tempo da volta (ms)	Média do tempo da fase (ms)	Avanço calculado (°)	Erro (%)
10	50	10	34,64	3,64	27,83	56
10	50	50	34,68	0,24	47,51	5
10	50	100	34,68	5,08	47,27	5
10	50	150	34,64	9,84	47,74	5
10	50	200	34,72	14,76	46,96	6
10	50	250	34,68	19,56	46,96	6
10	50	300	34,6	24,32	46,96	6
10	50	350	34,6	29,12	47,02	6

4.3 Análise dos erros

Com os dados obtidos com as tabelas 2, 3 e 4, referentes ao sinal do injetor de combustível, é possível notar que caso o sinal da injeção de combustível mude o estado logo após a falha ter sido detectada, o erro gerado no sinal é muito maior que o restante. Tal situação ocorreu para os casos apresentados na tabela 8.

Tabela 8 - Maiores erros referente à injeção.

Tempo Injeção (ms)	Fase Injeção (°)	Dif. PMS (°)	Erro (%)
1	10	0	4
10	10	0	21

Para o caso do erro de 4%, apresentado na variação do tempo de injeção, a média das outras medições é de 0,87%. Já para o caso do erro de 21%, apresentado na variação da fase da injeção, a média das outras medições foi de 2%.

O mesmo ocorre com os resultados obtidos nas tabelas 5, 6 e 7, referentes ao sinal da bobina de ignição, onde os maiores erros encontrados se encontram na tabela 9.

Tabela 9 - Maiores erros referente à ignição.

Dwell Time (ms)	Avanço (°)	Dif. PMS (°)	Erro (%)
1	10	0	4
10	50	10	56

Para o caso do erro de 4%, apresentado na variação do tempo de carga da bobina, a média das outras medições é de 0,28%. Para o caso do erro de 29%, apresentado na variação do avanço, a média das outras medições foi de 1,46%. Já para o caso de erro de 56%, apresentado na variação da diferença entre o PMS e a falha na roda fônica, a média das outras medidas foi de 5,6%.

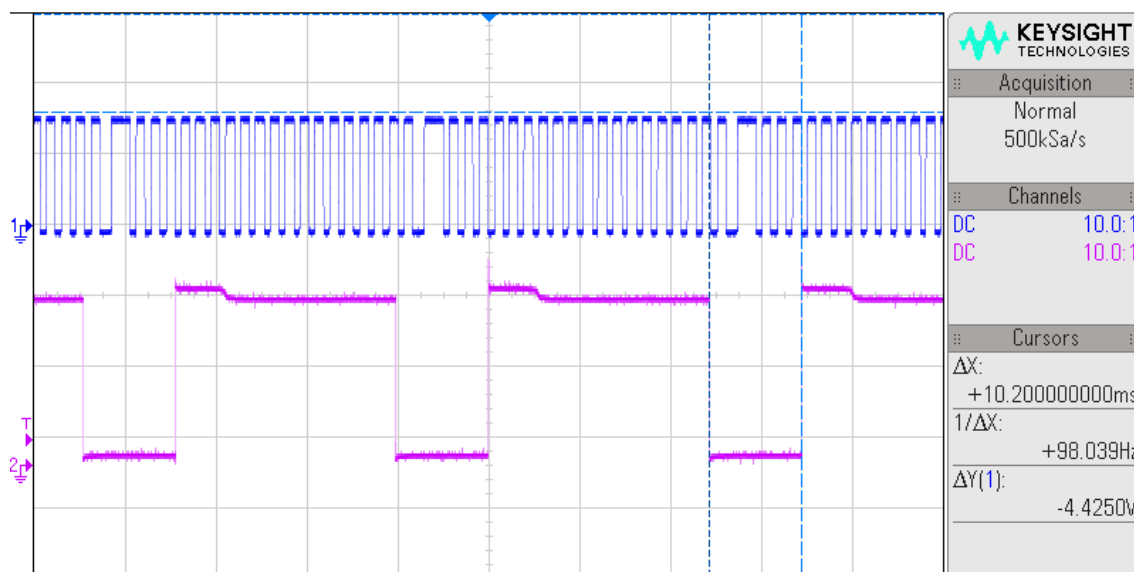
Para que esses erros sejam eliminados, a roda fônica deve ser instalada no motor com sua falha distante do PMS. Com isso, os erros obtidos na injeção de combustível podem ir de 4% para 0,4% em relação do tempo de injeção e de 21% para 1% em relação à fase da injeção. Já para a ignição, os erros podem ir de 4% para 0,16% em relação ao tempo de carga da bobina e de 56% para 6% em relação ao avanço.

4.4 Tensão nos componentes de saída

Após verificado que as ondas geradas pelo microcontrolador, responsáveis pelo acionamento do injetor de combustível e pela bobina de ignição, estão dentro das especificações do projeto, o ECU foi conectado a esses componentes. Ambos possuem dois terminais, um deles ligados à tensão da bateria e o outro foi conectado ao ECU.

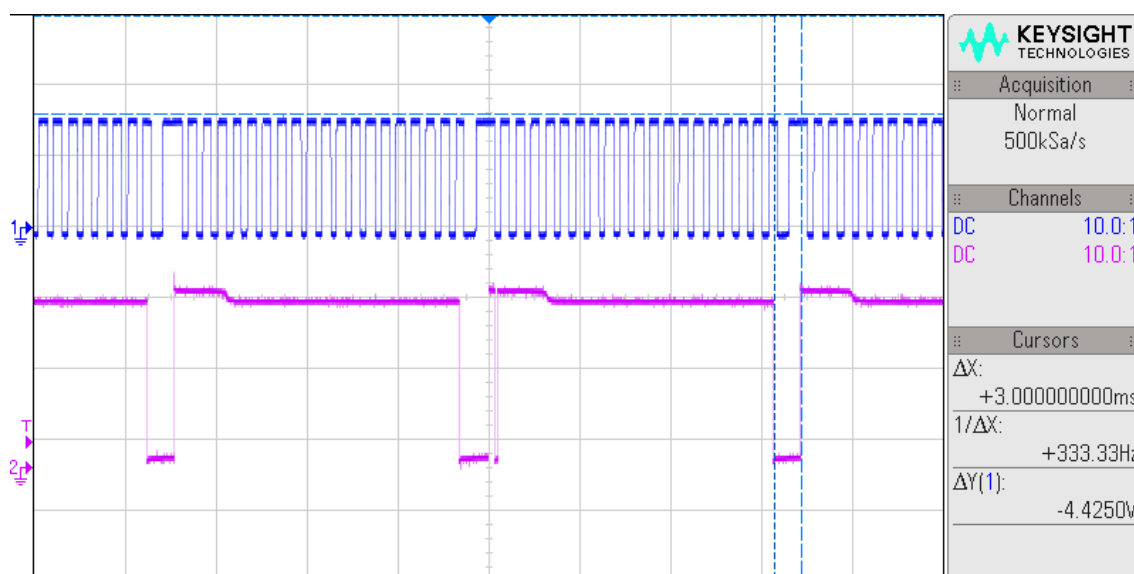
Assim, a tensão do injetor de combustível para tempo de injeção de 10ms, fase da injeção de 50° e sem diferenças do PMS e a falha da roda fônica, se encontra na figura 56.

Figura 56 - Tensão aplicada no injetor de combustível.



Já para o acionamento da bobina de ignição, foi utilizado tempo de carga da bobina de 3ms, avanço de 10° e sem diferença entre o PMS e a falha da roda fônica. A tensão aplicada nessas condições se encontra na figura 57.

Figura 57 - Tensão aplicada na bobina de ignição.



4.5 Comunicação USB

Os testes com a comunicação USB visaram estabelecer um modo de uso padrão do software com o hardware sendo solicitado com o motor desligado e ligado.

Considerando o motor desligado, não houve alteração, ou instabilidade, no sinal gerado para o controle do injetor de combustível ou no sinal gerado para o controle da bobina de ignição.

Já para o motor ligado, as rotinas de interrupção geraram muita instabilidade no sinal gerado para os componentes de saída, inviabilizando esse modo de implementação. Outra estratégia foi adotada, fazendo com que a ECU enviasse os dados periodicamente, sem um comando enviado pelo *software*, porém a instabilidade nos sinais de controle dos componentes de saída persistiu.

Com isso, o funcionamento da comunicação entre o *software* e a ECU foi limitado apenas para o momento em que o motor não estiver em funcionamento.

5 CONCLUSÕES

Atualmente a demanda pela utilização de eletrônica embarcada nos projetos envolvendo veículos é crescente. Isso advém do fato de que consumidores de todo o mundo estão buscando, cada dia mais, segurança e conforto ao se locomover. Além disso, novas normas ambientais forçam as empresas a criarem novas soluções para maior desempenho e eficiência energética em seus veículos.

O projeto aqui apresentado teve como objetivo a implementação de um módulo de controle de motor que é responsável pela potência gerada pelo motor e consumo de combustível do veículo.

O microcontrolador utilizado no presente trabalho, junto com o *firmware* desenvolvido apresentaram, durante seu desenvolvimento, algumas limitações que impossibilitaram a realização do monitoramento dos dados com o motor em funcionamento. Este fato não prejudica testes ou quaisquer funcionalidades, devido ao fato de que foram utilizados avanço fixo, tempo e fase de injeção fixos.

Ao implementar este módulo de controle em um motor de combustão interna, é necessário que se use uma roda fônica com a falha distante do PMS. A compensação deste parâmetro pode ser feita por meio de um software desenvolvido em LabVIEW, o que faz com que os maiores erros apresentados nesse projeto se reduzam de 56% para 6%.

Durante o desenvolvimento do projeto, foram criados um *hardware* de aquisição de dados que foi protegido contra ruídos, um *firmware* com um algoritmo eficiente, além de um *software* que atende às especificações referente à comunicação entre o usuário e o *hardware*, podendo ser produzido de maneira rápida. Todos esses grandes componentes do projeto contemplam todas as fases do desenvolvimento de componentes que envolvam eletrônica embarcada.

Por fim, conclui-se que o projeto apresentado nesse trabalho atenderia a aplicações como a de um motor estacionário para um gerador, entretanto, ficaria aquém das normas as quais os veículos são submetidos em que o erro para o fator lambda é de 1%.

5.1 Trabalhos futuros

Abaixo encontra-se uma lista de funcionalidades sugeridas para uma eventual continuação de implementação do projeto:

- Busca por um microcontrolador que permita a comunicação com o *software* sem comprometer os sinais de controle de injeção e ignição;
- Implementar a conversão dos dados obtidos com o conversor A/D e utilizá-los como base das compensações nos sinais de controle do injetor e ignição;
- Expandir o número de injetores de combustível e de bobinas de ignição controláveis pela ECU;
- Adicionar lógica de controle de injeção de combustível através do sensor lambda e lógica de controle de rotação quando o motor estiver em marcha lenta.

6 REFERÊNCIAS BIBLIOGRÁFICAS

BOSH. Bosh do Brasil. Disponível em <http://www.brasil.bosch.com.br/pt/br/br_main/our_company_1/history_1/history.html>. Acesso em 6 de agosto de 2016.

BRAGA, N. C. Eletrônica Automobilística. São Paulo: NCB, 2013.

CARROS INFOCO, Injeção Eletrônica: Sensor de Rotação, vital para o Motor. Disponível em < <http://www.carrosinfoco.com.br/carros/2012/07/injecao-eletronica-sensor-de-rotacao-vital-para-o-motor> > Acesso em 15/11/2016.

DENATRAN. Frota de Veículos – 2016. Disponível em <<http://www.denatran.gov.br/index.php/estatistica/261-frota-2016>>. Acesso em 7 de agosto de 2016.

INSTITUTO NEWTON C. BRAGA, Como funcionam os sensores de Efeito Hall (ART1050). Disponível em <<http://www.newtoncbraga.com.br/index.php/como-funciona/6640-como-funcionam-os-sensores-de-efeito-hall-art1050>>. Acesso em 20 de outubro de 2016.

INSTITUTO NEWTON C. BRAGA, Como funciona o Sistema de Ignição (ART1342). Disponível em < <http://www.newtoncbraga.com.br/index.php/como-funciona/6713-como-funciona-o-sistema-de-ignicao-art1342> >. Acesso em 1 de novembro de 2016.

MICROCHIP, dsPIC30F4013. Disponível em < <http://www.microchip.com/wwwproducts/en/dsPIC30F4013> >. Acesso em 3 de março de 2016.

MICROCHIP, 25LC1024. Disponível em < <http://www.microchip.com/wwwproducts/en/25LC1024> >. Acesso em 10 de março de 2016.

MICROCHIP, MCP2200. Disponível em < <http://www.microchip.com/wwwproducts/en/mcp2200> >. Acesso em 15 de março de 2016.

MICROCHIP, MCP2551. Disponível em < <http://www.microchip.com/wwwproducts/en/en010405> >. Acesso em 20 de março de 2016.

MOTEC, PLM. Disponível em < <http://www.motec.com/plm/plmoverview/> >. Acesso em 17/11/2016.

MTE – THOMSON, Informações técnicas Sensor de Pressão do Coletor MAP. Disponível em < <http://www.mte-thomson.com.br/site/wp-content/uploads/2014/10/SENSOR-MAP.pdf> >. Acesso em 25 de outubro de 2016.

MTE – THOMSON, Informações técnicas Sensor de Posição da Borboleta TPS. Disponível em < <http://www.mte-thomson.com.br/site/wp-content/uploads/2014/10/TPS.pdf> >. Acesso em 25 de outubro de 2016.

MTE – THOMSON, Sensor Lambda. Disponível em <<http://www.cursosonline.mte.com.br/licao/aula-14-sensor-lambda/>>. Acesso em 17/11/2016.

PASSARINI, L. C. Análise e projeto de válvulas injetoras. São Carlos: EESC – USP, 2002.

PULKRABEK, W. W. Engiennering Fundamentals of the Internal Combustion Engine. Platteville: Prattice Hall, 1997.

THOMAZINI, D; ALBUQUERQUE, P. U. B. Sensores Industriais - Fundamentos e Aplicações. São Paulo: Editora Érica Ltda., 2007.

APÊNDICE A – Projeto do *hardware*.

Os arquivos do projeto do *hardware* desenvolvido podem ser encontrados no repositório GitHub do autor:

https://github.com/andreluislui/Hardware_TCC

APÊNDICE B – Projeto do *firmware*.

Os arquivos fonte do projeto do *firmware* desenvolvido podem ser encontrados no repositório GitHub do autor:

<https://github.com/andreluislui/ECU>

APÊNDICE C – Projeto do *software*.

Os arquivos fonte do projeto do *software* desenvolvido podem ser encontrados no repositório GitHub do autor:

https://github.com/andreluislui/LabVIEW_TCC