

Faculdade de Tecnologia de Sorocaba
“José Crespo Gonzales”
Curso superior de Eletrônica automotiva

Sistema de aquisição de dados e calibração online aplicados à unidade de gerenciamento de um motor a combustão.

Lucas Wallace Martins da Silva
William Miranda Gusmão

Sorocaba
2019

RESUMO

A crescente busca do mercado automobilístico por melhor eficiência e emissões dos automóveis faz com que o motor a combustão seja cada vez mais bem gerenciado. Como os motores não são iguais entre si, as grandes indústrias automobilísticas realizam a calibração de parâmetros do módulo de gerenciamento do motor. Porém, estes procedimentos não possuem bibliografia acadêmica devido aos segredos que as montadoras desejam manter. Neste trabalho é desenvolvido um sistema de aquisição de dados e calibração em tempo real (online) de parâmetros, utilizando um módulo de gerenciamento do motor desenvolvido em trabalhos anteriores, e a plataforma QT Creator para criar a interface em um ambiente de computador, onde são coletados os principais dados do motor, podendo ser salvos em um planilha, e alterar parâmetros com ele em funcionamento, podendo observar a resposta do mesmo instantaneamente.

Palavras-chave: Calibração; Aquisição; ECU; Gerenciamento; Software.

LISTA DE FIGURAS

Figura 1. Principais componentes do motor a combustão interna.	12
Figura 2. Etapas de um motor a combustão interna ciclo Otto.	13
Figura 3. Sistema básico de injeção eletrônica.	14
Figura 4. Sistema da válvula eletro-injetora.	16
Figura 5. Válvula borboleta.	17
Figura 6. Bomba de combustível.	17
Figura 7. Unidade de controle do motor.	18
Figura 8. Roda fônica.	19
Figura 9. Sistema de sensores do motor.	20
Figura 10. Sensor de fase.	21
Figura 11. Sensor de pressão absoluta (MAP).	22
Figura 12. Sensor de temperatura.	23
Figura 13. Fenômeno de detonação.	24
Figura 14. Sensor de relutância variável.	26
Figura 15. Sensor efeito Hall.	27
Figura 16. Esquema de funcionamento do sensor de oxigênio.	28
Figura 17. Sistema de ignição.	29
Figura 18. Esquema de funcionamento bobina de ignição.	30
Figura 19. Tensão na vela de ignição.	31
Figura 20. Mapa de avanço de ignição.	32
Figura 21. Estrutura do microcontrolador.	33
Figura 22. Microcontrolador PIC.	34
Figura 23. Estrutura de memória do microcontrolador.	35
Figura 24. Entradas e saídas.	35
Figura 25. Conversor A/D.	36
Figura 26. Programação em linguagem C.	37
Figura 27. Interface QT Creator.	41
Figura 28. Comunicação Síncrona.	42
Figura 29. Comunicação SPI.	43
Figura 30. SPI com múltiplos escravos.	43
Figura 31. Barramento I2C.	44
Figura 32. Resistores de pull-up no barramento I2C.	45
Figura 33. Frame do I2C.	45
Figura 34. Comunicação Assíncrona.	46
Figura 35. Transmissão UART.	47
Figura 36. Ligação para comunicação UART.	47
Figura 37. Frame de transmissão UART.	48
Figura 38. Unidade de Gerenciamento utilizada.	49
Figura 39. Interface desenvolvida.	52
Figura 40. Tela de Aquisição de dados da interface.	53
Figura 41. Tela de Calibração Online da interface.	55
Figura 42. Arduino Uno utilizado.	56
Figura 43. Onda quadrada simulando roda fônica.	57
Figura 44. Lógica para gerar sinal de fase e rotação.	58
Figura 45. Osciloscópio.	59

Figura 46. Multímetro.....	59
Figura 47. Teste e validação da comunicação.....	60
Figura 48. Aquisição utilizando gerador de rotação e fase.....	61
Figura 49. Aquisição de dados do motor do VW POLO 1.6 FLEX.....	62
Figura 50. Interface antes da calibração.....	63
Figura 51. Interface depois da calibração.....	63

LISTA DE TABELAS

Tabela 1. Dados de aquisição salvos.	54
---	----

LISTA DE SIGLAS

ECU	<i>Eletronic Control Unit</i> (Unidade de controle eletrônica)
CAN	<i>Controller Area Network</i> (Rede de Área Controlada)
I2C	<i>Inter-Integrated Circuit</i> (Circuito Inter-Integrado)
ECM	<i>Engine Control Module</i> (Módulo de controle do motor)
MOSI	<i>Master Data Output, Slave Data Input</i> (Saída de dados mestre, Entrada de dados escravo)
MISO	<i>Master Data Input, Slave Data Output</i> (Entrada de dados mestre, Saída de dados escravo)
SxCLK	<i>Serial Clock</i>
SS	<i>Slave Select</i> (Seleção de Escravo)
SDA	<i>Serial Data</i>
SCL	<i>Serial Clock</i>
I2C	<i>Inter-Integrated Circuit</i>
SPI	<i>Serial Peripheral Interface</i> (Interface periférica serial)
ACK	<i>Acknowledgement Bit</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i> (Receptor/Transmissor Universal Assíncrono)
MAP	<i>Manifold Absolute Pressure</i> (Medidor de pressão absoluta no coletor de admissão)
USB	<i>Universal Serial Bus</i> (Porta Universal)
PMI	Ponto Morto Inferior
PMS	Ponto Morto Superior
PIC	<i>Peripheral Interface Controller</i>
PWM	<i>Pulse Width Modulation</i> (Modulação por largura de pulso)

Sumário

1. Introdução	8
1.1 Justificativa.....	9
1.2 Questão problema.....	9
1.3 Objetivo	9
1.4 Metodologia.....	10
2. Referencial Teórico.....	11
2.1 Motor a Combustão Interna	12
2.1.1 História do motor a Combustão Interna	12
2.1.2 Princípio de funcionamento de um motor 4 tempos	13
2.2 Injeção eletrônica	14
2.2.1 Válvulas eletro-injetoras	15
2.2.2 Válvula borboleta.....	16
2.2.3 Bomba de combustível.....	17
2.2.4 ECM (Engine Control Module)	18
2.2.5 Roda fônica	19
2.3 Sensores.....	20
2.3.1 Sensor de Fase	20
2.3.3 Sensor de Temperatura	22
2.3.4 Sensor de Detonação.....	23
2.3.5 Sensor de Rotação.....	25
2.3.6 Sensor de oxigênio.....	27
2.4 Sistema de ignição.....	29
2.4.1 Bobina de ignição	30
2.4.2 Cabos e velas de ignição	31
2.4.3 Avanço de ignição	32
2.5 Microcontroladores.....	33
2.5.1 Microcontrolador PIC.....	33
2.5.2 Memória.....	34
2.5.3 Entradas / Saídas	35
2.5.4 Conversor Analógico / Digital.....	36
2.6 Linguagem de Programação	36
2.6.1 Linguagem C	37
2.6.2 Linguagem C++	40
2.6.3 Programação orientada a objetos	40
2.7 Protocolos de Comunicação	41
2.7.1 Comunicação Síncrona	42
2.7.2 Comunicação Assíncrona	46
3. Materiais e Métodos	49
3.1 Unidade de Gerenciamento de Desenvolvimento.....	49

3.1.1 Bloco de gerenciamento	50
3.1.2 Bloco de admissão	50
3.1.3 Bloco de sincronismo	50
3.1.4 Bloco de comunicação	51
3.1.5 Protocolo de comunicação entre os microcontroladores	51
3.2 Software de Aquisição e Calibração.....	52
3.2.1 Aquisição de dados	52
3.2.2 Calibração Online de Parâmetros	55
3.3 Gerador de sinais de rotação e fase	56
3.3.1 Hardware	56
3.3.2 Software.....	57
3.4 Instrumentos de Medição.....	58
3.4.1 Osciloscópio	58
3.4.2 Multímetro.....	59
4. Análises e Resultados	60
5. Conclusão	64

1. Introdução

A indústria automotiva brasileira trabalha, de forma majoritária, reproduzindo ou ajustando projetos que são desenvolvidos no exterior. Mais especificamente, no tocante à injeção eletrônica Flex, o mercado nacional atua somente na etapa de calibração do produto. Todavia, com o advento do motor Flex, a partir de março de 2003, o país passou a ter quase 90% de sua frota de automóveis composta por veículos que carregam essa configuração. Dessa maneira, é interessante para a economia e para o crescimento nacional, incentivar, não somente a produção desses motores, mas também o desenvolvimento do sistema eletrônico que o controla, contemplando desde o hardware, software, realização de testes com simulação, bem como testes em campo e calibração (RUIZ, 2019).

A crescente busca, das montadoras de veículos automobilísticos, por melhorias relacionadas à eficiência, consumo e emissões de poluentes faz com que seja obrigatório um gerenciamento cada vez mais sofisticado dos motores de combustão interna a partir de uma central eletrônica, conhecida como ECU (*Electronic Control Unit*).

As reduções em consumo de combustível e emissões são duas frentes de pesquisas ativas na área de motores. As técnicas de controle aplicadas ao motor de combustão interna são de substancial importância para atender às restritas normas globais na limitação dos níveis de poluentes emitidos pelos automóveis. Dessa maneira, é de grande valia a atenção para as características dos componentes eletrônicos, mecânicos e teorias de controle que estão envolvidos no projeto que abrange o controle desse sistema (RUIZ, 2019).

Atualmente as unidades de controle eletrônico (ECU) aplicadas ao gerenciamento do motor são responsáveis por ler os dados dos sensores, tais como o sensor de rotação, sensor de temperatura do líquido de arrefecimento, sensor de temperatura do ar, sensor lambda, sensor MAP (Manifold Absolute Pressure), entre outros. Com o gerenciamento destes dados, é possível controlar o funcionamento de atuadores, tais como as válvulas eletro-injetoras de combustível, bobinas de ignição e válvula borboleta.

Na prática, cada modelo de motor é diferente, o que faz com que seja necessário calibrar os parâmetros para gerenciamento da ECU, visando obter os melhores resultados no motor, tanto em emissões, quanto em desempenho.

1.1 Justificativa

Devido à complexidade das estratégias de gerenciamento e diferenças de um motor para outro, muitos parâmetros precisam ser bem calibrados. Em um ambiente acadêmico isso se torna um desafio, devido à falta de recursos, disponibilidade de materiais acadêmicos, entre outros.

Este projeto possibilitará a calibração de maneira prática da ECM (*Engine Control Module*) desenvolvida pelos alunos da Faculdade de Tecnologia de Sorocaba em parceria com a Faculdade de Tecnologia de Santo André e a Universidade de São Paulo (JUNIOR; JATO; HIROKI, 2016; NOGUEIRA; ROQUE; SANTOS, 2017; RUIZ, 2018).

Está calibração busca eliminar a necessidade de desligar o motor e gravar um novo firmware nos microcontroladores do módulo quando for preciso alterar algum parâmetro. Facilitando a visualização e o entendimento das mudanças geradas pela alteração do parâmetro de interesse.

1.2 Questão problema

Dentro do contexto citado acima, envolvendo a busca por ajustes nos parâmetros existentes para funcionamento do motor a combustão ciclo Otto, testaremos a seguinte hipótese: “É possível realizar a aquisição de dados e a calibração dos parâmetros de um motor à combustão em tempo real, mantendo o mesmo em funcionamento, através de um software aplicativo em um ambiente acadêmico?”.

1.3 Objetivo

O objetivo do presente projeto consiste em desenvolver um sistema de aquisição de dados e calibração online, e implementá-lo em uma ECU de arquitetura descentralizada, desenvolvida anteriormente (JUNIOR; JATO; HIROKI, 2016; NOGUEIRA; ROQUE; SANTOS, 2017; RUIZ, 2018), onde será possível visualizar os principais dados do motor e alterar os parâmetros em tempo real, com o motor em funcionamento, através de um software a ser desenvolvido no ambiente QT Creator, a fim de acompanhar as respostas de funcionamento do motor em função das alterações nos parâmetros calibráveis.

1.4 Metodologia

Para atingir o objetivo deste projeto foi utilizado metodologia de pesquisa descritiva, na qual seu objetivo é descrever as características de uma população, um fenômeno ou experiência para o estudo realizado. A pesquisa está classificada como quantitativa de caráter exploratório, visando desenvolver um software no computador utilizando a C++, que é uma linguagem de programação baseada na linguagem C, e alterar o firmware presente no módulo eletrônico (ECU).

Toda a programação foi inserida no QT Creator e as duas ferramentas estão interligadas, ou seja, a interface do software foi desenvolvida no QT Creator, enquanto a funcionalidade, para calibração dos parâmetros, está codificada em programação C++.

O software desenvolvido possui uma comunicação serial entre ele e a ECU. Para análise e aquisições de dados, será utilizado osciloscópio e analisador lógico. A pesquisa bibliográfica complementar utilizada foi encontrada em revistas técnicas, sites, dissertações e livros.

2. Referencial Teórico

O objetivo do referencial teórico é fornecer um esquema completo o suficiente para desenvolver a pesquisa, e isso exige proposições teóricas. É essencial que se desenvolva uma teoria antes que se faça a coleta de dados para qualquer estudo de caso (YIN, 2001).

Este capítulo apresenta toda a base teórica necessária para o estudo, assim como o seu entendimento. Estão descritos o funcionamento do motor à combustão, injeção eletrônica descrição de sensores e atuadores, módulo de gerenciamento do motor, entre outros.

2.1 Motor a Combustão Interna

Os motores à combustão interna são máquinas que transformam a energia de uma reação química em energia mecânica, através de ciclos termodinâmicos que envolvem a compressão e expansão de gases, queima e exaustão (SCHULZ, 2009).

2.1.1 História do motor a Combustão Interna

O desenvolvimento dos motores à combustão interna iniciou-se em 1665, onde o combustível era o vapor. Em 1759, Henry Hood propôs a utilização de ar quente ao invés de vapor, ideia executada por George Caley em 1807. Outros motores a ar que operavam por combustão externa, onde o combustível era queimado fora dos cilindros foram desenvolvidos, destacando-se os motores de Robert Stirling elaborado no ano de 1816 e o modelo de John Ericson de 1826. Esses motores apresentavam um melhor rendimento por operarem com pressões superiores aos motores a vapor (TILLMANN, 2013).

Nicolaus Otto apresentou em 1876 seu protótipo de motor de quatro tempos que tinha peso e tamanho muito inferiores aos dos sistemas existentes, o que veio por iniciar a indústria do motor de combustão interna. A figura 1 ilustra um motor de combustão interna com ciclo Otto, e seus principais componentes (HEYWOOD, 1988).

Figura 1. Principais componentes do motor a combustão interna.

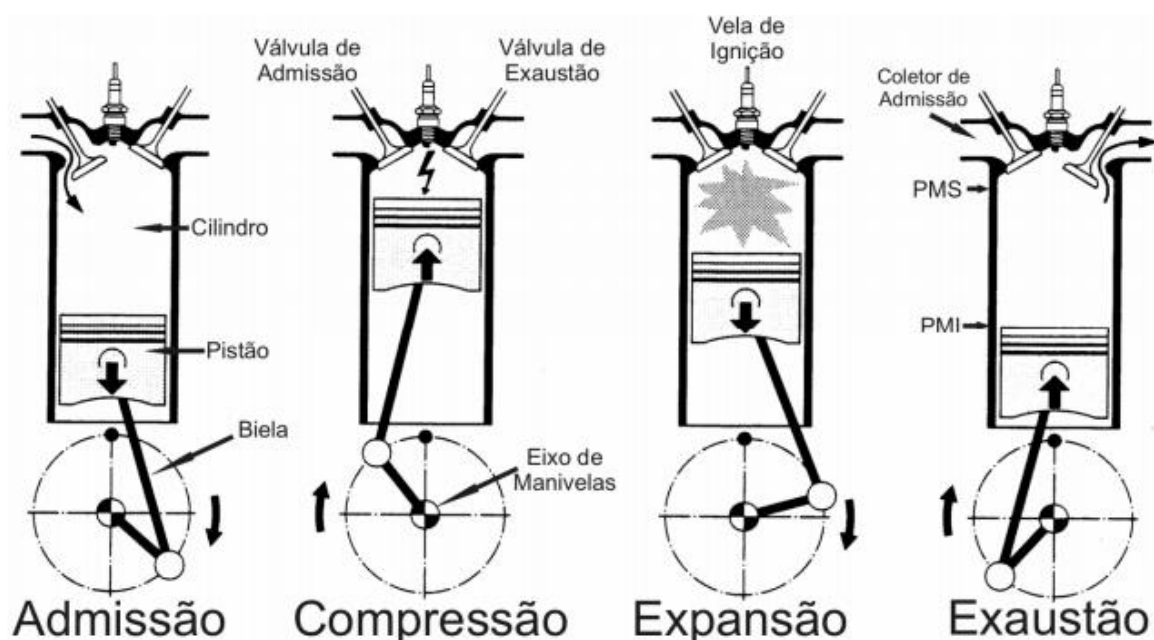


Fonte: DIAS (2015).

2.1.2 Princípio de funcionamento de um motor 4 tempos

Os motores de combustão interna ciclo Otto tem seu funcionamento baseado em quatro ciclos termodinâmicos, que ocorrem pelo deslocamento repetitivo do êmbolo (pistão) ao longo dos cilindros. As quatro etapas para o funcionamento são: admissão, compressão, expansão e exaustão (figura 2).

Figura 2. Etapas de um motor a combustão interna ciclo Otto.



Fonte: BRAGA (2007).

Conforme a figura 2, se considerarmos um único cilindro no sistema, o ciclo Otto, começa pela etapa de admissão. Nesta etapa a válvula de exaustão retorna à sua posição inicial, e a válvula de admissão é deslocada através do comando de válvulas, permitindo que a mistura carburante composta por ar e combustível seja aspirada para dentro do cilindro através da depressão gerada pelo deslocamento do êmbolo desde o Ponto Morto Superior (PMS) até o Ponto Morto Inferior (PMI) do cilindro (BASSHUYSEN, 2004).

Na etapa de compressão, ambas as válvulas estão na posição inicial, vedando a câmara de combustão. Com isso, o deslocamento do êmbolo do pistão desde o PMI até o PMS comprime a mistura carburante.

Na expansão, a vela de ignição produz uma centelha causando uma combustão na mistura carburante previamente comprimida. Com isso ocorre um aumento abrupto da pressão dentro da câmara de combustão, impulsionando o êmbolo do PMS para o PMI (DIAS, 2015).

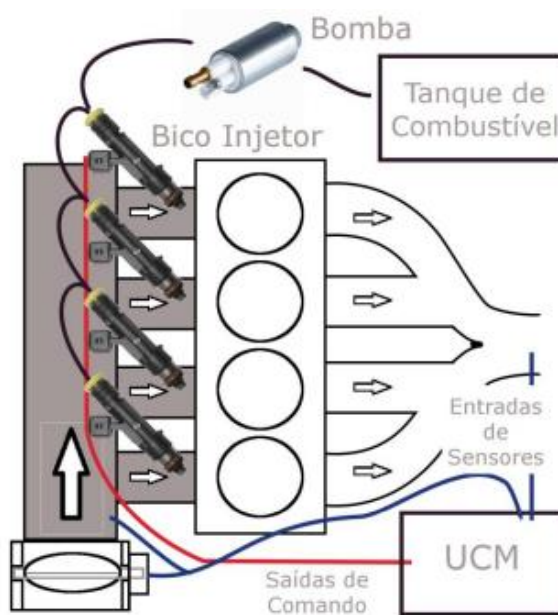
A etapa de exaustão é onde os gases, produtos da combustão, são expelidos da câmara de combustão pelo êmbolo, que se desloca do PMI até o PMS, com a válvula de exaustão deslocada de sua posição inicial.

2.2 Injeção eletrônica

Segundo Capelli (2010), a injeção eletrônica é composta por uma ou várias centrais para controlar todo o sistema do carro, partindo de performance e economia para segurança e conforto. Para o motor, o nome dado à essa central, designasse ECU (Eletronic Control Unit) ou ECM (Engine Control Module).

O sistema de injeção eletrônica tem como função injetar uma massa de combustível proporcional a massa de ar ingerida no coletor de admissão. Antigamente o carburador era responsável por essa função, porém, esse sistema emitia uma grande quantidade de poluentes por não conseguir fazer uma mistura estequiométrica, logo foram desenvolvidos os sistemas de injeção eletrônica indireta monoponto, alguns anos depois o multiponto e atualmente a grande tendência do mercado automotivo é a injeção direta multiponto. A figura 3 ilustra um sistema básico de injeção de combustível em um motor de quatro cilindros.

Figura 3. Sistema básico de injeção eletrônica.



Fonte: CHAVES (2016).

Este sistema de injeção de combustível geralmente é formado pelos seguintes componentes: válvulas eletro-injetoras, popularmente conhecidas como bicos injetores, bomba de combustível, ECM e os sensores.

2.2.1 Válvulas eletro-injetoras

A Válvula Eletro-Injetora ou Injetor controla a injeção de combustível de forma precisa, sendo responsável pela formação da mistura, injetando o combustível atomizado, na forma de finas gotículas, na corrente de ar que se dirige aos cilindros que é destinado ao coletor de admissão (Figura 4).

A injeção de combustível tem diferentes aplicações em diferentes sistemas (MTE-THOMSON, 2019).

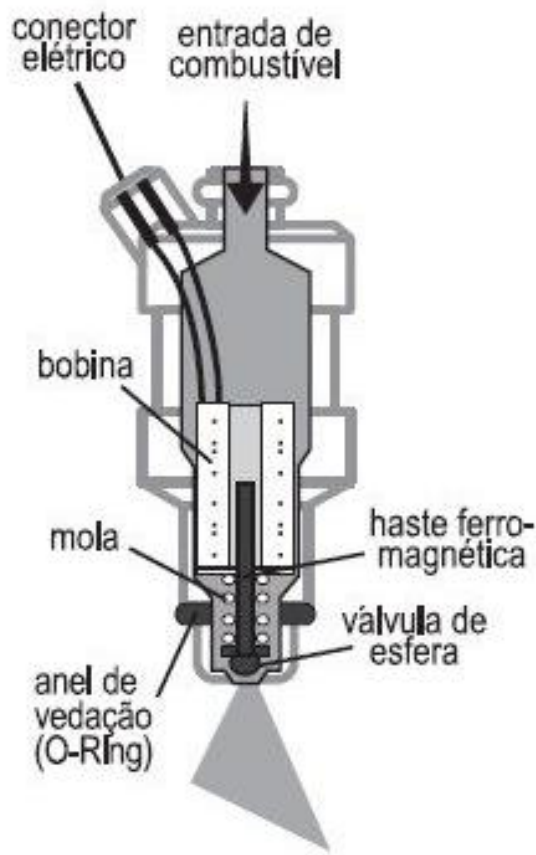
Segue exemplos:

- Nos sistemas monoponto (single point): O injetor é acionado em sincronismo com os ciclos de ignição; ou seja, para cada ciclo de ignição existe um ciclo de injeção.

Já no caso de sistemas multiponto, os injetores podem ser controlados de diversas maneiras:

- Injeção simultânea: Os injetores são acionados simultaneamente, a cada ciclo ignição.
- Injeção semi-sequencial ou “banco a banco”: Os injetores são acionados de forma alternada, por grupos: uma metade dos injetores é acionada num ciclo de ignição, a outra metade, no seguinte.
- Injeção sequencial: os injetores são acionados individualmente a cada ciclo de ignição, seguindo a ordem de ignição.

Figura 4. Sistema da válvula eletro-injetora.



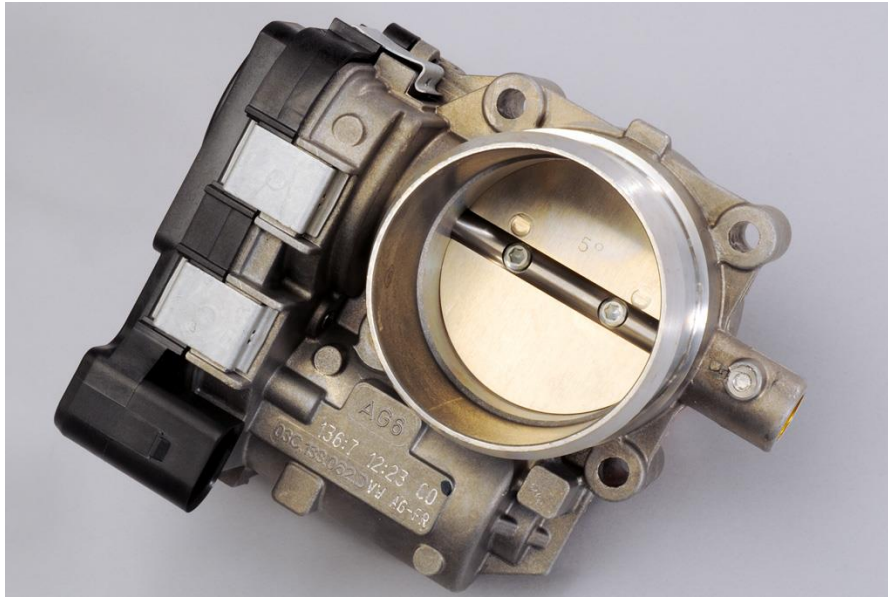
Fonte: MTE-THOMSON (2019).

2.2.2 Válvula borboleta

Antes do conceito de controle eletrônico da válvula borboleta, a massa de ar era administrada exclusivamente de acordo com a demanda do condutor, de maneira que o prato da válvula era diretamente conectado ao pedal acelerador por meio de um cabo de aço. Ao se pisar no acelerador, o cabo de aço acionava o prato da válvula, promovendo sua abertura e possibilitando a passagem de ar. Todavia, com essa configuração, muitas variáveis e condições internas e externas eram ignoradas. Sendo assim, a eficiência da combustão era prejudicada e, de modo geral, o desempenho do motor também (AHMED; ABBAS, 2012).

Para suprir as necessidades apresentadas no sistema anterior, foi desenvolvido um modulo de controle e incorporado a borboleta, tornando possível determinar um angulo exato que melhor contribua com a demanda imediata do motor. É possível observar um corpo de borboleta na figura 5.

Figura 5. Válvula borboleta.



Fonte: MIX AUTO (2017).

2.2.3 Bomba de combustível

A bomba de combustível é um componente que tem a função de manter a linha de combustível pressurizada, para que os bicos injetores possam realizar a liberação de combustível. A figura 6 ilustra uma bomba de combustível usada no carro VW GOL 1.6.

Figura 6. Bomba de combustível.



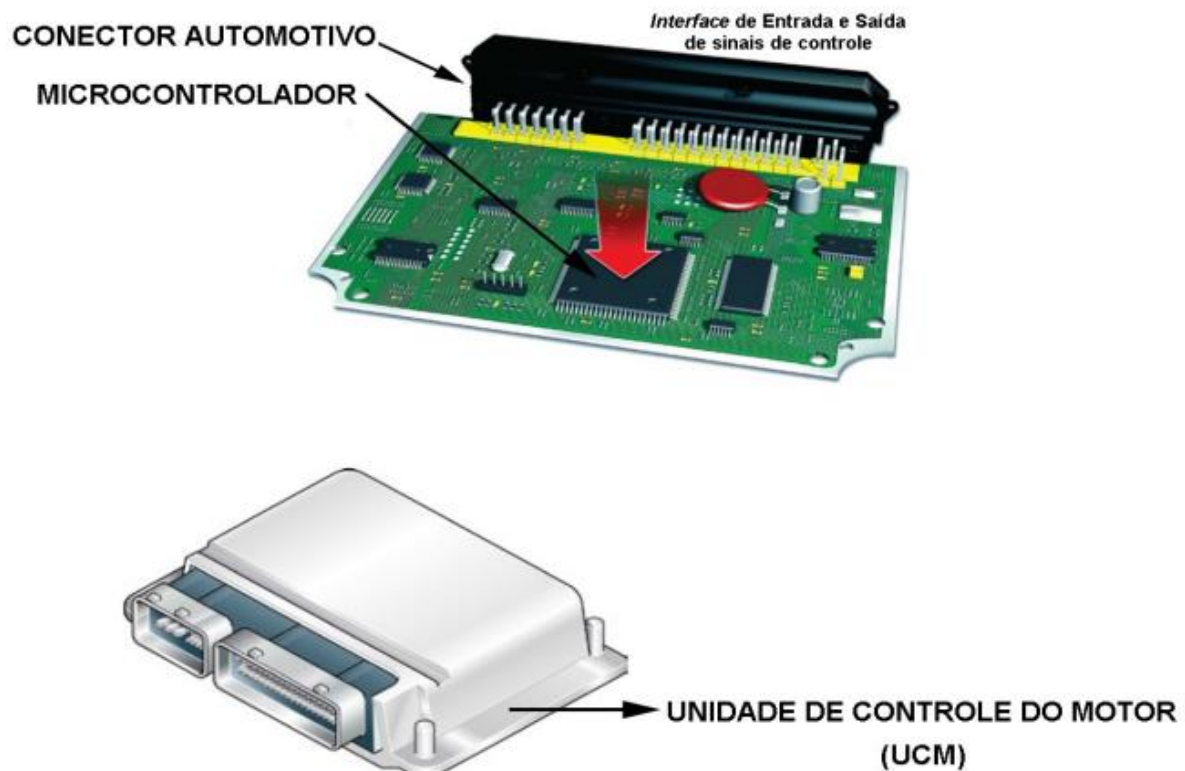
Fonte: GUSTAVO (2017).

2.2.4 ECM (Engine Control Module)

O Módulo de Controle do Motor é uma placa eletrônica que funciona como o cérebro do motor, realizando todo o controle de atuadores a partir de dados coletados dos sensores, de maneira que ele funciona da melhor forma possível, tanto em desempenho como em economia e em emissões (DIAS, 2015).

A ECM necessita utilizar estratégias para controlar o motor do veículo em todos os regimes de carga, ou seja, dependendo da solicitação de carga feita pelo motorista, o módulo utiliza sensores e atuadores para melhor atender essa necessidade (DIAS, 2015).

As estratégias utilizadas são inseridas via software, em um ou mais microcontroladores da placa eletrônica do módulo de gerenciamento. A figura 7 ilustra uma unidade de controle de motor da Volkswagen.



Fonte: VOLKSWAGEN (2009).

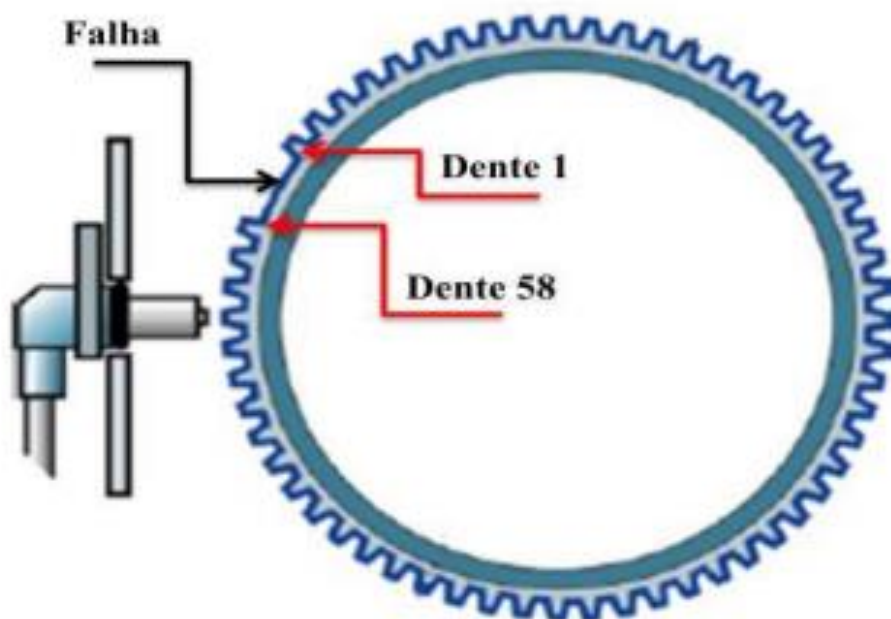
Pode – se considerar as seguintes funções à unidade de controle do motor:

- Controle da injeção de combustível;
- Controle da ignição;
- Controle da válvula borboleta;
- Controle do sistema de arrefecimento do motor;
- Controle de marcha lenta.

2.2.5 Roda fônica

A roda fônica é localizada acoplada ao eixo de manivelas, girando na mesma rotação do motor. Ela é feita de um material ferroso para que possa ter interação com o campo magnético do sensor de rotação. Ela possui 58 dentes e 2 “falhas” onde seria a localização de mais 2 dentes, como ilustra a figura 8. As falhas servem para que o módulo possa identificar o primeiro dente, de forma a sincronizar a injeção e ignição com a rotação do motor. A distância entre os dentes é de 6° (ALBALADEJO, 2013).

Figura 8. Roda fônica.

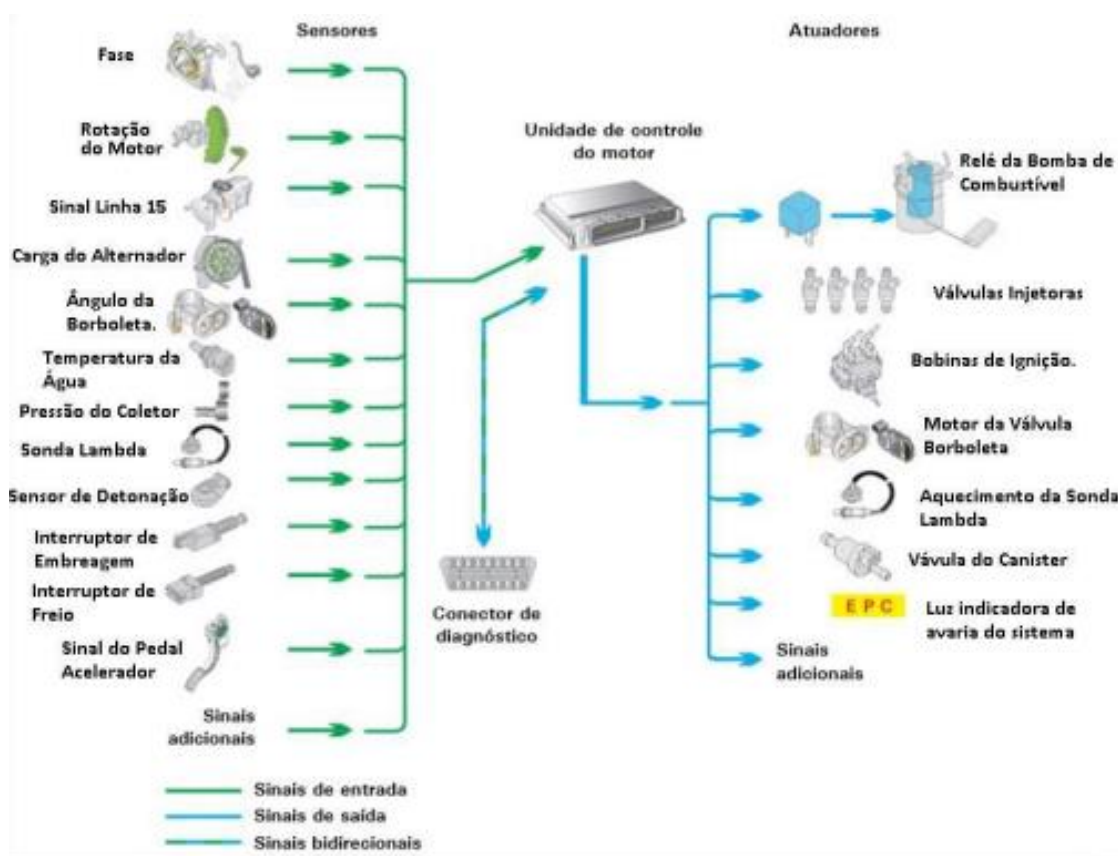


Fonte: ALBALADEJO (2013).

2.3 Sensores

Os sensores têm como função a conversão de fenômenos físicos e químicos em sinais elétricos, permitindo que a unidade controladora possa realizar a leitura desses fenômenos, controlando os atuadores de acordo com a comparação desses sinais com valores previamente estabelecidos e a estratégia de gerenciamento implementada. A figura 9 ilustra um sistema de sensores conectados à unidade de controle do motor (REBOLO, 2018).

Figura 9. Sistema de sensores do motor.



Fonte: VOLKSWAGEN (2009).

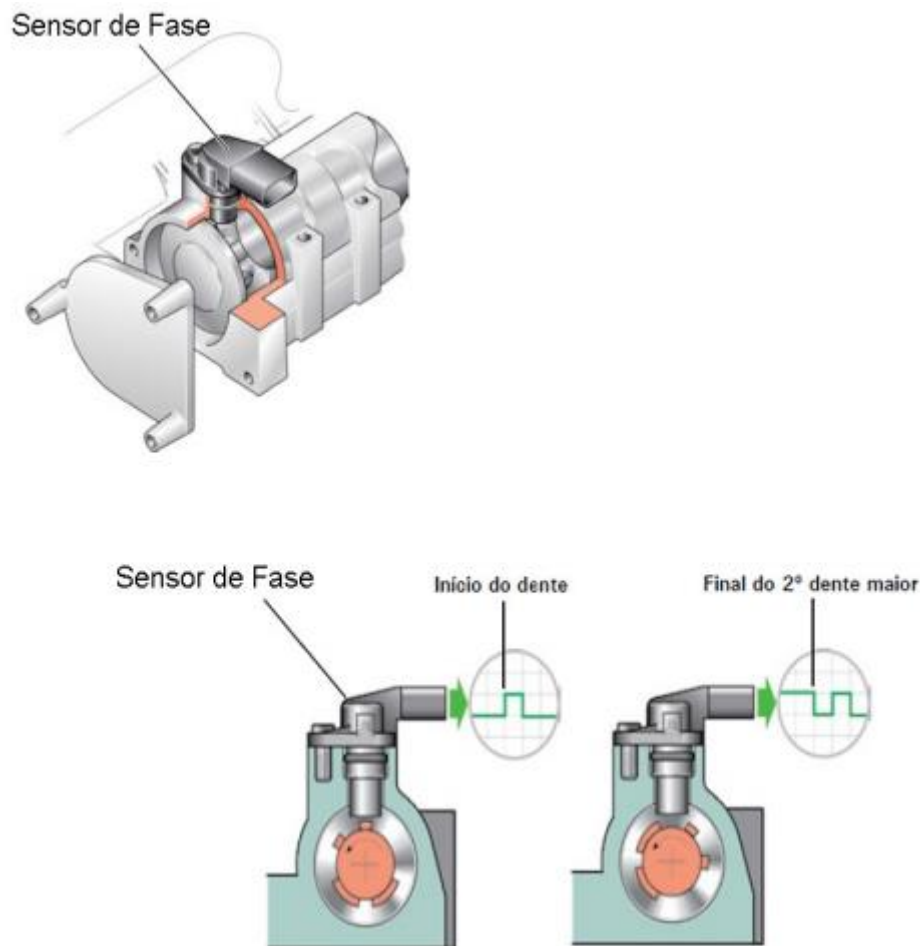
2.3.1 Sensor de Fase

Conhecido como sensor de rotação, é um componente importante no sistema de injeção eletrônica pois mede a rotação do eixo onde está instalado e informa a unidade de comando (ECU) (MTE-THOMSON, 2019).

Através do sinal pulsado enviado pelo sensor de rotação, a unidade de comando calcula a velocidade de rotação do eixo onde está instalado o sensor. Além disso é usado para

identificar em qual cilindro está ocorrendo a combustão. Ele faz a leitura através do efeito Hall, identificando a passagem dos dentes da roda fônica do comando de válvulas. A figura 10 ilustra onde é posicionado e como é o sinal gerado pelo sensor de fase.

Figura 10. Sensor de fase.



Fonte: VOLKSWAGEN (2009)

2.3.2 Sensor de Pressão Absoluta

O sensor de Pressão Absoluta (MAP) é um “manômetro” que transforma a pressão medida no coletor de admissão em sinais elétricos informando a ECU. (MTE-THOMSON, 2019).

O sensor MAP tem por função medir a pressão (ou depressão) do coletor de admissão e também medir a pressão atmosférica, informando a ECU para realizar os cálculos de admissão. A figura 11 ilustra como é um sensor MAP.

Figura 11. Sensor de pressão absoluta (MAP).



Fonte: MTE-THOMSON (2019)

2.3.3 Sensor de Temperatura

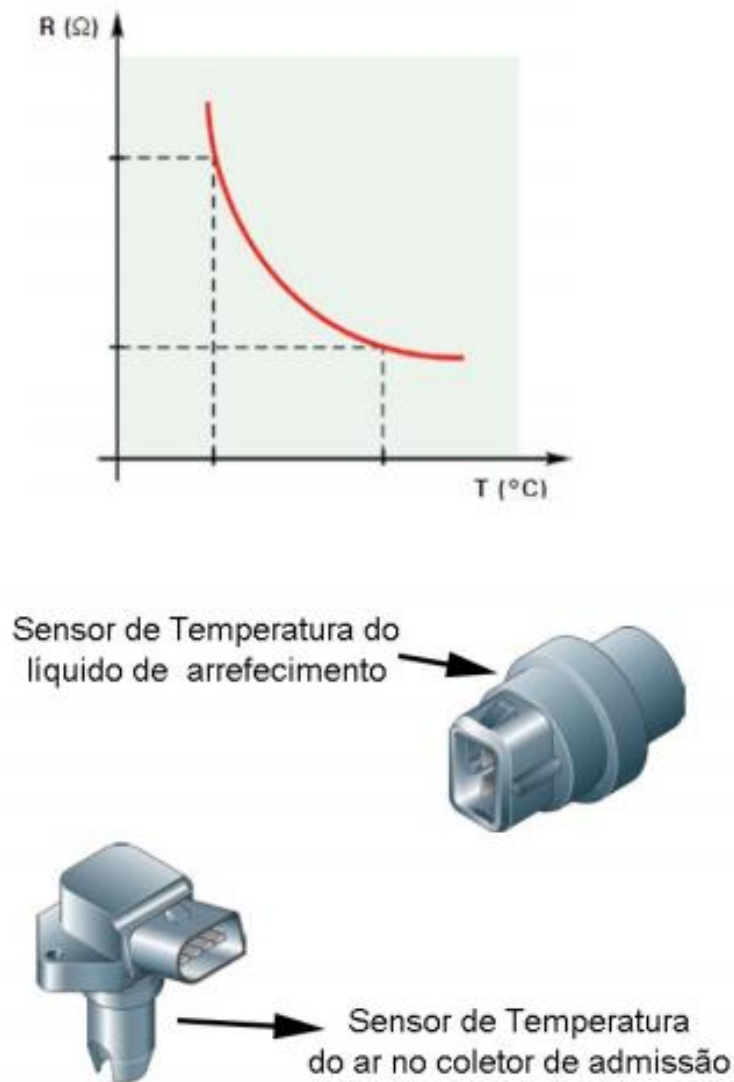
Os motores ciclo Otto geralmente realizam leituras nas temperaturas do líquido de arrefecimento e do ar dentro do coletor de admissão.

O sensor NTC (Negative Temperature Coefficient) varia a sua resistência em função da temperatura. Ele emite um sinal elétrico inversamente proporcional à temperatura medida (DIAS, 2015).

O sensor de temperatura do ar de admissão, juntamente com o sensor MAP, são de fundamental importância para o cálculo da massa de ar.

O sensor de temperatura do líquido de arrefecimento é fundamental para que o módulo de gerenciamento do motor realize estratégias de correção do tempo de injeção e ignição, que devem variar conforme a temperatura do motor. A figura 12 ilustra o sensor de temperatura e sua curva padrão.

Figura 12. Sensor de temperatura.



Fonte: VOLKSWAGEN (2009).

2.3.4 Sensor de Detonação

O sensor de detonação, também conhecido como sensor “knock”, tem como princípio um piezoelétrico, que converte energia mecânica em sinais elétricos.

Quando ocorre a ignição da mistura ar/combustível, gera uma chama com uma velocidade de expansão que depende de fatores como temperatura e propriedades geométrica (HIROKI, 2016).

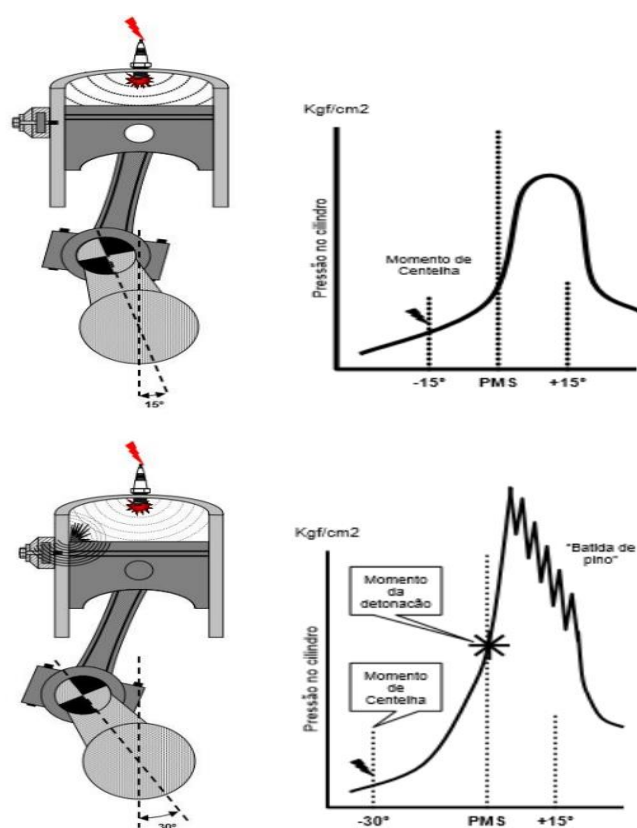
A velocidade de queima varia conforme a temperatura e intensidade de compressão.

Para obter o ponto exato de pressão máxima na câmara de combustão, é necessário adiantar o ponto de ignição, aproveitando a melhor potência e torque no motor (HIROKI, 2016).

Se a mistura alcançar condições críticas de pressão ou temperatura, ela se inflama espontaneamente, ocasionando numa colisão com a frente de chama gerada pela vela de ignição. Com isso há um aumento elevado da pressão, causando o fenômeno chamado de “detonação”, que é transmitido para o motor em forma de picos de vibração.

Esta vibração é detectada pelo sensor “knock”, como ilustra a figura 13.

Figura 13. Fenômeno de detonação.



Fonte: XAVIER (2015).

Segundo Hiroki (2016), detonação é um processo de combustão incorreta que ocorre dentro da câmara e as principais causas para que ocorra são:

- Temperatura elevada;
- Componentes mecânicos gastos;
- Gasolina adulterada;

- Ponto de ignição avançado demais;
- Mistura pobre;
- Vela de ignição sem as especificações corretas.

2.3.5 Sensor de Rotação

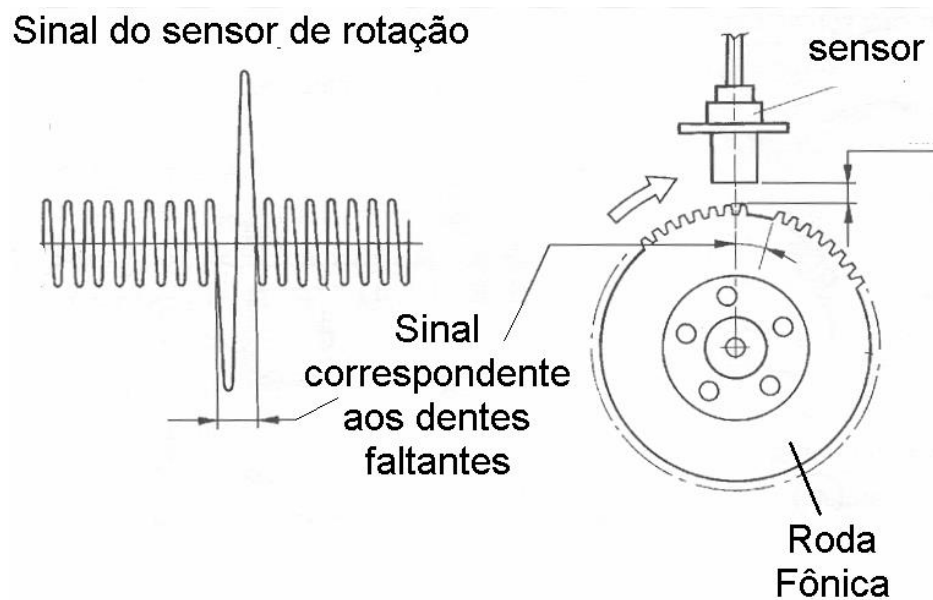
O sensor de rotação tem a função de fornecer à ECM a rotação do motor, para que possam ser calculados os avanços de ignição e abertura de ar. Dois tipos de sensores de rotação são mais utilizados: o sensor indutivo e o sensor Hall (RUIZ, 2019).

Tais sensores detectam a rotação por meio da Roda Fônica descrita anteriormente.

2.3.5.1 Sensor de relutância variável

O sensor de relutância variável é normalmente chamado de sensor indutivo. Ele é um sensor do tipo ativo, portanto não precisa de alimentação. Seu mecanismo consiste em um ímã envolvido por uma bobina com dois terminais. O sensor gera uma tensão diretamente proporcional ao campo magnético que flui pelo circuito. Conforme a roda fônica se movimenta, o campo magnético varia com o aproximar e afastar dos dentes. Quando o dente está totalmente alinhado ao sensor, a tensão é nula. A medida em que o dente se afasta, a tensão se eleva e passa a inverter o sentido da corrente, portanto o sinal de saída do sensor do tipo indutivo é uma onda senoidal que tem sua amplitude e frequência variáveis de acordo com a rotação do motor. A figura 14 ilustra o funcionamento de um sensor de relutância variável (BOSCH, 2005).

Figura 14. Sensor de relutância variável.

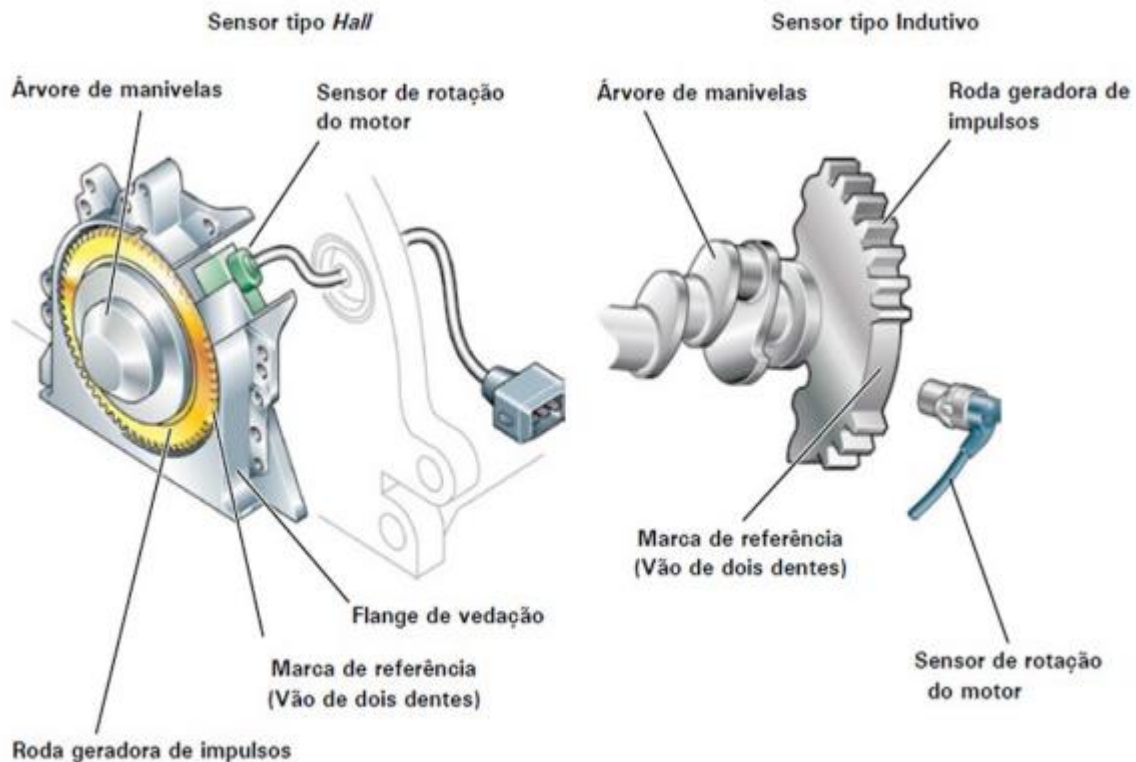


Fonte: SATO (2017).

2.3.5.2 Sensor efeito Hall

O sensor de efeito Hall é constituído por um ímã permanente em um material semicondutor com propriedades eletrônicas específicas. O material é alimentado com 5V ou 12V dependendo do sistema de gerenciamento. A alimentação proporciona uma corrente elétrica que flui pelo dispositivo semicondutor. De maneira semelhante à configuração anterior, a roda geradora de pulsos tem a finalidade de aproximar ou afastar o campo magnético gerado pelo ímã permanente. À medida que o campo magnético se aproxima do material semicondutor, a força magnética promove um desvio dos elétrons que estavam atravessando o material, ocasionando uma diferença de potencial nos terminais do sensor, quando o campo magnético se afasta, a diferença de potencial se esvai. A figura 15 ilustra o sensor de efeito hall, assim como sua posição no motor (VOLKSWAGEN, 2009).

Figura 15. Sensor efeito Hall.



Fonte: VOLKSVAGEN (2009).

2.3.6 Sensor de oxigênio

Conhecido como sonda lambda, o sensor de oxigênio fica localizado no escapamento do carro onde há a passagem dos gases provenientes da combustão e assim tem por característica medir a quantidade de oxigênio presente nos gases liberados pelo motor. A partir dos dados coletados o sensor informa a ECU se a mistura ar/combustível está rica (muito combustível) ou pobre (muito ar) como mostra a figura 16.

Quando a mistura está rica a tensão gerada em Volts pelo sensor é alta (900 milivolts), nesse instante o computador para de injetar combustível e a mistura fica pobre. O sensor então informa a ECU com uma tensão baixa (50 milivolts) e neste instante o computador volta a injetar mais combustível. Esse movimento de transição entre rico e pobre lembra a letra grega Lambda (λ) (MTE-THOMSON, 2019).

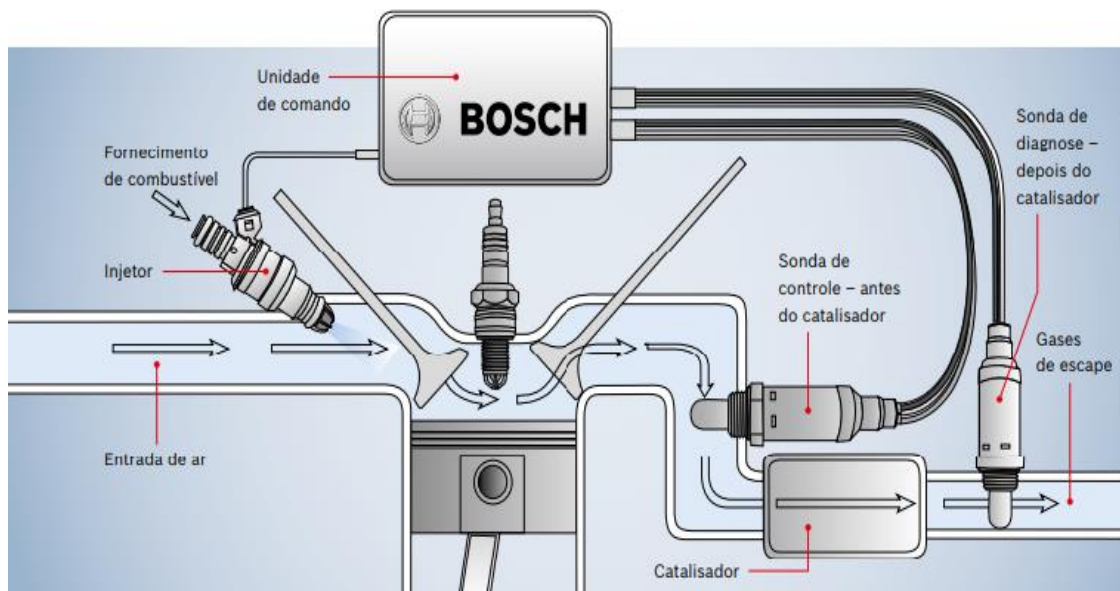
O sensor é formado por um eletrólito cerâmico de óxido de zircônio, estabilizado por óxido de ítrio, coberto por platina em suas superfícies interior e exterior. A superfície interior

está em contato direto com o ar atmosférico e com os gases de escape, enquanto a camada exterior é conectada à massa do veículo. (RUIZ, 2019).

Em temperaturas elevadas, mais especificamente acima de 300° C, o eletrólito cerâmico torna-se condutor e gera uma carga galvânica que caracteriza o teor de oxigênio dos gases de escape. Há ainda uma camada cerâmica porosa para revestir o eletrodo externo da sonda a fim de protegê-lo das agressões causadas pelos gases de escapamento (BRUNETTI, 2012).

O sensor de oxigênio tem uma melhor eficiência de trabalho acima 300° C, por esse motivo é localizado mais próximo possível do escape do motor e antes do catalisador. A fim de corrigir os desvios do sensor enquanto o motor está em fase de aquecimento são utilizados sensores de oxigênio aquecidos eletricamente para se atingir a temperatura de trabalho rapidamente.

Figura 16. Esquema de funcionamento do sensor de oxigênio.



Fonte: BOSCH (2016).

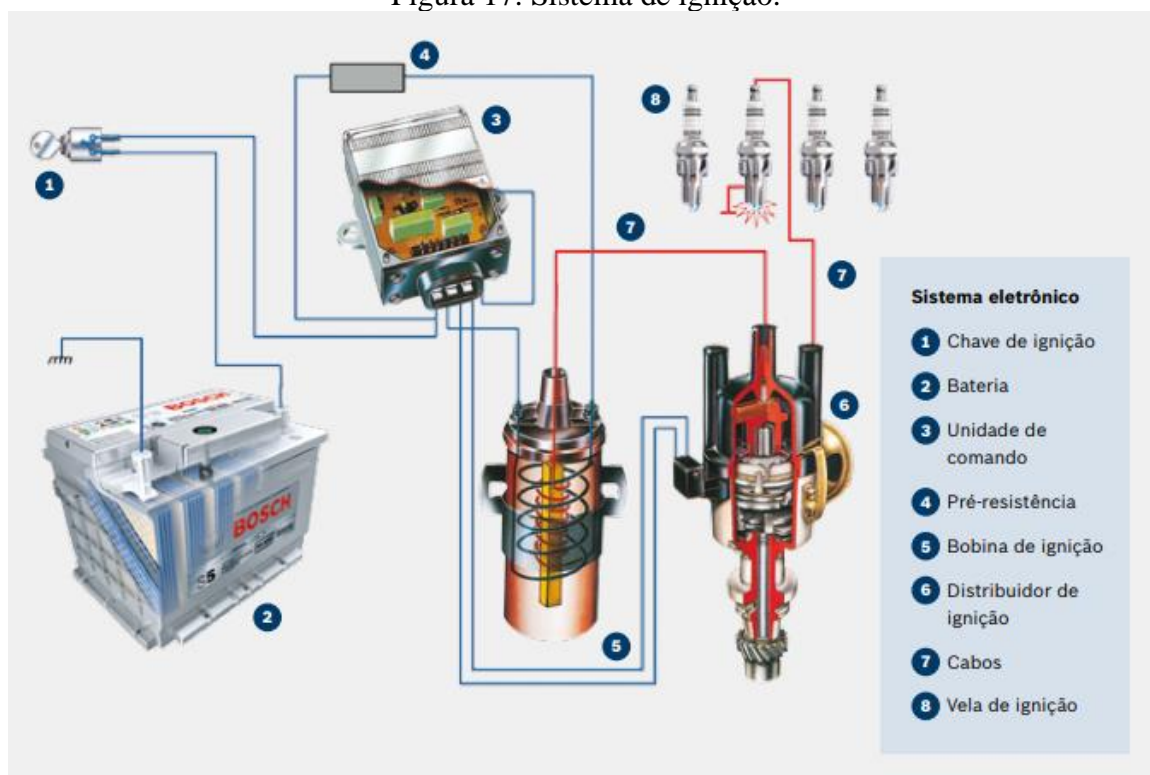
2.4 Sistema de ignição

Segundo Ruiz (2019), o sistema de ignição (figura 17) tem por objetivo de realizar uma faísca elétrica no momento de compressão da mistura ar/combustível e assim realizar a explosão e ter uma boa combustão. O conceito de ignição externa se baseia em uma centelha elétrica para iniciar a combustão. Já na auto-ignição a mistura se aquece à medida em que a compressão se estabelece tendo como resultado a sua ignição quando atinge determinada temperatura, ou quando o combustível é injetado no ar, onde as condições limites permitem a evaporação e a ignição.

No subsistema de ignição, a unidade tem por objetivo:

- O controle do ponto de ignição, que é ajustado conforme os mapas inseridos em sua memória;
- Controle de detonação seletivo por cilindro;
- Correção do ponto de ignição e correção da marcha lenta através da alteração do ponto de ignição.

Figura 17. Sistema de ignição.



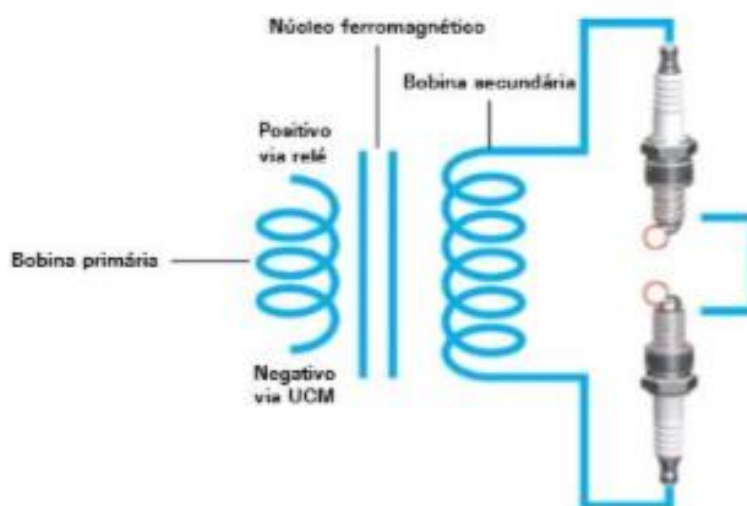
Fonte: BOSCH (2013).

2.4.1 Bobina de ignição

A bobina funciona como um transformador que aumenta uma tensão contínua pulsante, permitindo transformar uma corrente contínua em pulsos de alta tensão que serão conduzidos às velas de ignição através dos cabos de velas. Atualmente, as bobinas são comandadas pela central de gerenciamento do motor (ECM). Antigamente, as bobinas eram acionadas por um contato elétrico instalado dentro do distribuidor (platinado). (MARTINS, 2005).

O funcionamento da bobina de ignição baseia-se na lei da indução, consistindo de dois enrolamentos de cobre acoplados magneticamente (enrolamento primário e secundário) como mostra a figura 18. A energia armazenada no enrolamento primário, por um sinal de baixa tensão, é transformado em um sinal de alta tensão na saída do secundário. Nos sistemas controlados eletronicamente, a energização e posterior corte do primário é realizado por um transistor, que atuando como uma chave, fecha o circuito de alimentação do primário até que se tenha energia suficiente para suprir a demanda. O período em que o primário é energizado (denominado de ângulo de permanência) é controlado, juntamente com o avanço, pelo sistema de controle da ignição. Geralmente a tensão fornecida pelo secundário é da ordem de 30 kV, garantindo assim que a tensão de ruptura da vela de ignição seja alcançada e se tenha uma combustão eficiente. (Milhor, 2002, p. 35).

Figura 18. Esquema de funcionamento bobina de ignição.



Fonte: VOLKSWAGEN (2009).

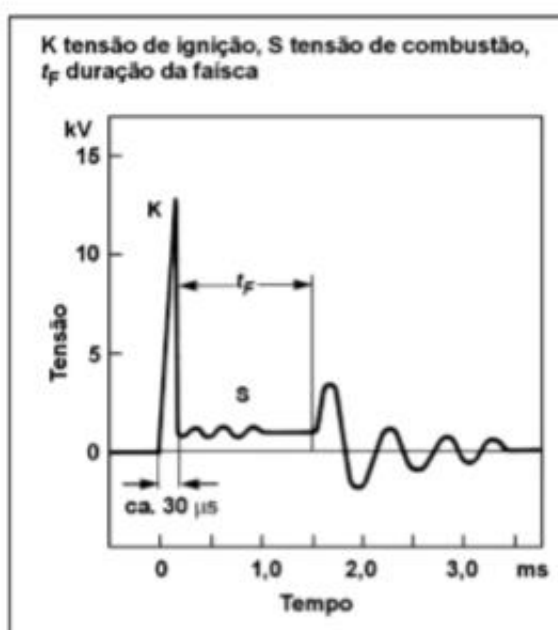
2.4.2 Cabos e velas de ignição

De acordo com Denton (2000), são cabos especialmente resistentes à alta tensão e às altas temperaturas existentes no compartimento do motor. Conduzem a alta tensão do distribuidor às velas. Existem alguns cabos de vela que são resistivos, essa característica visa eliminar interferências eletromagnéticas produzidas pela alta tensão (faísca). Essas interferências podem prejudicar o funcionamento dos componentes eletrônicos do veículo, tais como: rádio, unidade de comando da injeção eletrônica.

A vela proporciona a centelha a qual tem o seu papel fundamental na explosão da combustão, através da alta tensão gerada pela bobina e que é transmitida pelo cabo de vela é possível se ter uma faísca elétrica na câmara de combustão.

No momento da ignição, a tensão nos eletrodos da vela de ignição sobe abruptamente de zero até a tensão disruptiva (tensão de ignição). Lançada a faísca, a tensão na vela de ignição cai até a tensão de combustão. Durante o período de duração da faísca a mistura ar/combustível tem a oportunidade de se inflamar. Após a ruptura da centelha a tensão se estabelece periodicamente como mostra a figura 19 (BOSCH, 1998).

Figura 19. Tensão na vela de ignição.



Fonte: BOSCH (1998).

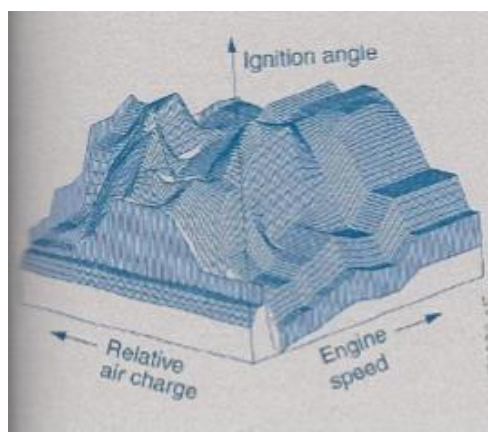
2.4.3 Avanço de ignição

Segundo Ruiz (2019), quando a ignição foi iniciada pela centelha, é necessário um intervalo de tempo em milissegundos até que a mistura ar/combustível seja completamente queimada pela combustão. Esse período permanece sensivelmente constante, contanto que a composição da mistura permaneça inalterada. O momento do ponto de ignição deve ser alterado de modo que a combustão e o pico de pressão ocorram instantes depois do cilindro atingir o PMS. O momento de centelha se dar por conta da posição do pistão que está acoplado na árvore de manivela através da biela, o valor de avanço ignição é determinado em graus (considerando o movimento de rotação do eixo virabrequim).

Conforme o motor tenha um aumento de rotação, o curso do pistão tem sua velocidade alterada proporcionalmente. Assim o ponto de ignição precisa ser avançado em relação ao PMS já que quanto maior for a velocidade do motor maior também deve ser o grau de antecipação da faísca para uma combustão completa.

O efeito na curva de combustão também está relacionado a carga do cilindro, assim com uma baixa carga no cilindro o ângulo de ignição deve ser atrasado (BOSCH, 2015). Sendo assim, a unidade de controle do motor armazena em sua memória de programa um mapa tridimensional de avanço de ponto de ignição para cada condição de Rotação (RPM) e composição da Mistura (geralmente obtida a partir da informação da carga a que o motor está submetido) se obtém um valor de ângulo (chamado de avanço da ignição). Esse mapa correlaciona a rotação do motor, a carga e o ângulo de ignição, conforme a figura 20.

Figura 20. Mapa de avanço de ignição.

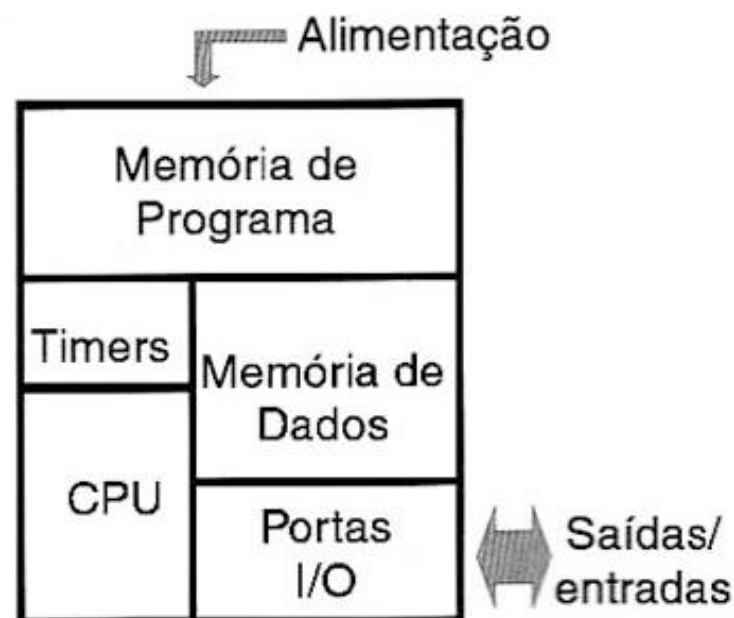


Fonte: BOSCH (2015).

2.5 Microcontroladores

Os Microcontroladores são como microprocessadores que podem ser programados para funções específicas. Em geral, eles são usados para controlar circuitos e, por isso, são comumente encontrados dentro de outros dispositivos, sendo conhecidos como "controladores embutidos". A estrutura interna de um microcontrolador apresenta um processador, bem como circuitos de memória e periféricos de entrada e saída. A figura 21 ilustra como é a estrutura de um microcontrolador (FORESTI, 2013).

Figura 21. Estrutura do microcontrolador.



Fonte: (<http://www.newtoncbraga.com.br/index.php/eletronica/52-artigos-diversos/13263-obasico-sobre-os-microcontroladores-parte-1-mic139>)

2.5.1 Microcontrolador PIC

A empresa Microchip Technology oferece uma série de circuitos integrados que chama de "Controladores de Interface Periférica", ou microcontroladores PIC. Eles combinam um microprocessador, memória e uma interface em um único chip, oferecendo aos desenvolvedores profissionais, técnicos e amadores um baixo custo, plataforma de fácil utilização para o desenvolvimento de sistemas eletrônicos controlados por software. Microcontroladores PIC vêm em uma variedade de recursos, a partir de unidades de 8 bits simples aos mais sofisticados chips de 32 bits. A figura 22 mostra um microcontrolador PIC

de 8 bits da família 18F (<http://ptcomputador.com/Ferragens/computer-drives-storage/46821.html>).

Figura 22. Microcontrolador PIC.

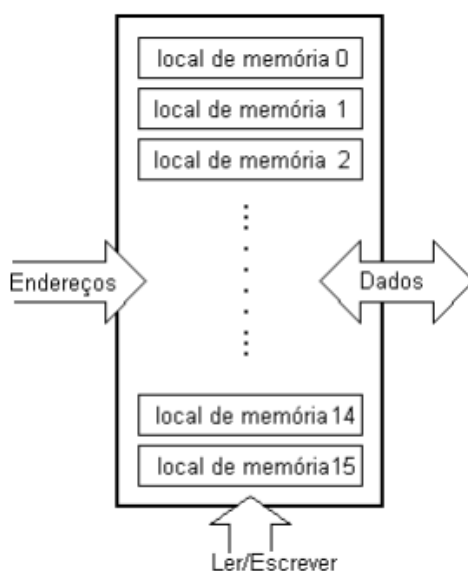


Fonte: MICROCONTROLLERS LAB (2017).

2.5.2 Memória

A memória é a parte do microcontrolador cuja função é guardar dados. A figura 23 mostra um exemplo de um modelo simplificado de uma unidade de memória. Para uma entrada específica, obtém-se a saída correspondente. A linha Ler/Escriver (R/W) determina quando estamos lendo ou escrevendo na memória (ASSIS, 2004).

Figura 23. Estrutura de memória do microcontrolador.

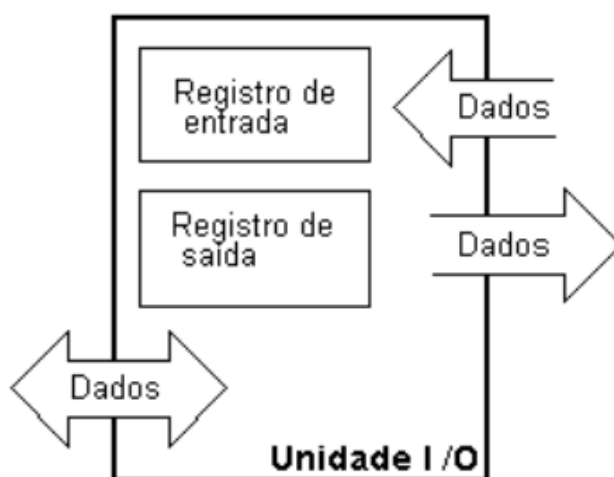


Fonte: ASSIS (2004).

2.5.3 Entradas / Saídas

Para que o microcontrolador possa ter contato com o mundo exterior, existem barramentos ligados às linhas de saída. Eles são chamados de “port’s”. Existem port’s de entrada, de saída, e de entrada/saída. No software, é necessário definir qual port irá trabalhar, e definir se ele enviará ou receberá dados. A figura 24 ilustra um exemplo simplificado de uma unidade de entrada/saída (I/O) (ASSIS, 2004).

Figura 24. Entradas e saídas.

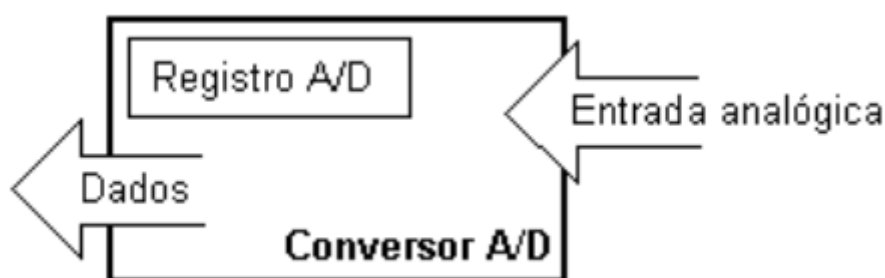


Fonte: ASSIS (2004).

2.5.4 Conversor Analógico / Digital

O microcontrolador pode apenas processar sinais digitais (zero e um), então, para que ele possa trabalhar com sinais analógicos, como de sensores por exemplo, é preciso convertê-los em valores binários. Esta tarefa é executada através de um bloco para conversão analógica-digital ou um conversor A/D. A figura 25 ilustra o funcionamento de forma simplificada do Conversor Analógico Digital (ASSIS, 2004).

Figura 25. Conversor A/D.



Fonte: ASSIS (2004).

2.6 Linguagem de Programação

Linguagem de programação se trata de um método padronizado, com um conjunto de regras sintáticas e semânticas para definir um programa e então expressar instruções para um computador.

A linguagem também permite que o indivíduo que estiver programando especifique como um dado vai ser armazenado ou transmitido, sobre quais irá atuar, entre outras muitas ações sob várias circunstâncias.

Uma das principais metas das linguagens de programação é permitir que os programadores tenham maior produtividade, permitindo expressar suas intenções mais facilmente do que quando comparado a código de máquina.

Dessa maneira, elas são projetadas para adotar uma sintaxe de nível mais alto, podendo ser mais facilmente entendida por humanos (GIRALDELI, 2011).

2.6.1 Linguagem C

A linguagem C foi criada por Dennis Ritchie nos laboratórios da Bell Telephone no ano de 1972.

Ela é considerada de propósito geral, ou seja, é uma linguagem capaz de ser usada para praticamente qualquer tipo de projeto, podendo facilmente ser usada em qualquer plataforma (Casavella)

Um programa em C é constituído de:

- Um cabeçalho contendo as diretivas do compilador onde se definem o valor de constantes simbólicas, declaração de variáveis, inclusão de bibliotecas e declaração de rotinas;
- Um bloco de instruções principal e outros blocos de rotinas (<https://monografias.brasilecola.uol.com.br/computacao/fundamentos-linguagem-c.htm>).

A figura 26 ilustra um exemplo de programação em linguagem C usando o software MPLAB X da Microchip e o compilador XC8.

Figura 26. Programação em linguagem C.

```
8
9  #include <xc.h>
10 #include <stdio.h>
11
12 #define LED PORTDbits.RD0
13
14 int x = 0;
15 char y = 1;
16
17 void main (void)
18 {
19     TRISD = 0;
20     while(1)
21     {
22         if(x == 0 && y == 0)
23         {
24             LED = 1;
25         }
26         else if (x == 0 && y == 1)
27         {
28             LED = 0;
29         }
30     }
31 }
```

Fonte: Autores.

2.6.1.1 Diretivas de compilação

São chamados de diretivas de compilação os comandos que são processados durante a compilação do programa.

Eles informam os compiladores do C quais são as constantes simbólicas usadas no programa e quais bibliotecas devem ser anexadas ao programa executável.

A diretiva “#include” informa ao compilador para incluir outros arquivos na compilação do programa.

A diretiva “#define” diz ao compilador quais serão as constantes simbólicas usadas no programa (GONÇALVES, 2003).

2.6.1.2 Declaração de variáveis

Segundo GONÇALVES (2003), as variáveis são espaços de memória reservados que guardam valores durante a execução de um programa.

Todas as variáveis devem ser declaradas antes de serem usadas, consistindo no nome de um tipo, seguido do nome da variável, seguido de ponto e vírgula, como no exemplo a seguir:

Ex: tipo_da_variável nome_da_variável

```
int contador;
```

O tipo de uma variável informa a quantidade de memória, em bytes, que a variável ocupará.

Os tipos de variáveis são:

char: Guarda um caractere;

int: Guarda um número inteiro;

float: Guarda um número real com certa precisão;

double: Guarda um número real com precisão maior que float;

void: Tipo vazio.

Temos algumas variações destes tipos, que são:

unsigned char: caractere sem sinal;
 long int: número inteiro, com domínio estendido;
 unsigned int: número inteiro positivo;
 unsigned long int: número inteiro positivo com domínio estendido.
 short int: inteiro com domínio reduzido.
 unsigned short int: inteiro positivo com domínio reduzido

2.6.1.3 Estruturas de controle

A linguagem C permite uma ampla variedade de estruturas de controle de fluxo de processamento. Duas estruturas das estruturas básicas (decisão e repetição) são muito utilizadas, e fáceis de entender (<https://monografias.brasilecola.uol.com.br/computacao/fundamentos-linguagem-c.htm>).

Estrutura de Decisão: Direciona o fluxo lógico para dois blocos distintos de instruções conforme uma condição de controle.

Pseudo-linguagem	Linguagem C
se <i>condição</i>	if (<i>condição</i>){
então <i>bloco 1</i>	<i>bloco 1</i> ;
senão <i>bloco 2</i>	} else{
fim se	<i>bloco 2</i> ;
	}

Estrutura de Repetição: Executa repetidamente um bloco de instruções até que uma condição de controle seja satisfeita.

Pseudo-linguagem	Linguagem C
faça	do{
<i>bloco</i>	<i>bloco</i> ;
até <i>condição</i>	} while (<i>condição</i>);

2.6.2 Linguagem C++

Segundo Pacievitch, o C++ é uma linguagem baseada na linguagem C, considerada de médio/alto nível. Seu desenvolvimento começou na década de 80, por Bjarne Stroustrup. O objetivo era melhorar uma versão do núcleo Unix. Para desenvolver a linguagem, foram acrescentados elementos de outras linguagens, na tentativa de criar uma linguagem com elementos.

Em 1985 apareceu a primeira versão oficial do C++.

Algumas características do C++:

- Possibilidade em programação de alto e baixo nível;
- É padronizado pela ISO;
- Compatibilidade com a linguagem C;
- Possui paradigmas de programação funcional, genérica, orientada a objetos e imperativa;
- Pode ser utilizada sem a necessidade de um ambiente de desenvolvimento sofisticado (<https://www.portalgsti.com.br/cplusplus/sobre/>).

2.6.3 Programação orientada a objetos

A programação orientada a objetos é um paradigma de programação que se baseia fundamentalmente no conceito de um conjunto componentes que interagem entre si para resolver um problema, denominados Objetos.

Os objetos são os “blocos de construção” de software na orientação a objetos, possuindo características próprias sendo capaz de executar determinadas ações.

Eles representam entidades no mundo real, como pessoas, correntes, aviões, contas, etc, assim como também podem representar gráficos, círculos, quadrados, cones, etc.

A comunicação entre os objetos é realizada à partir da troca de mensagens, que é um sinal enviado entre eles, requisitando um serviço usando uma operação programada no objeto chamado (REIS, 2015).

2.6.3.1 Software QT Creator

QT Creator é um software de desenvolvimento que possui uma versão gratuita, onde é possível criar aplicativos para plataformas de desktop e móveis.

Ele possui programação orientada a objetos sendo utilizado a linguagem de programação C++ (www.qt.io).

A figura 27 ilustra a interface gráfica de desenvolvimento do QT Creator.

Figura 27. Interface QT Creator.



Fonte: Autor.

2.7 Protocolos de Comunicação

Conforme os equipamentos eletrônicos vem agregando cada vez mais funções, torna-se mais comum a utilização de diversos circuitos integrados que trocam dados entre eles. Para ocorrer esta troca de dados, é necessário configurar um tipo de comunicação entre esses dispositivos.

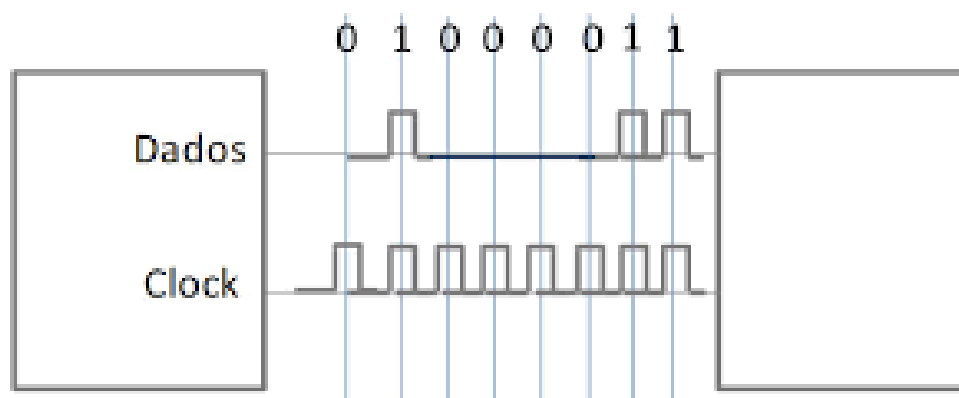
Diversas tecnologias de comunicação entre dispositivos foram desenvolvidas, podendo ser separadas em duas categorias: a comunicação síncrona e a comunicação assíncrona (SCHUNCKE, 2013).

2.7.1 Comunicação Síncrona

Na comunicação síncrona, ambos os dispositivos (emissor e receptor) devem estar em sincronia antes da comunicação iniciar, e permanecer em sincronia durante a transmissão de dados.

Para que ocorra a sincronização, ocorre o envio de uma configuração de bits de sincronização no início da mensagem a ser transmitida. A figura 28 ilustra de maneira simplificada uma comunicação síncrona (SCHUNCKE, 2013).

Figura 28. Comunicação Síncrona.



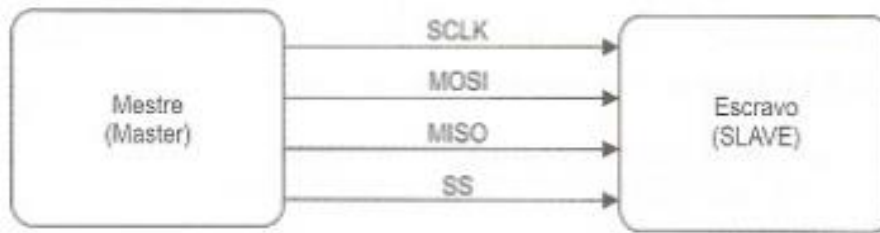
Fonte: SIMÕES (2019).

2.7.1.1 Comunicação SPI (Serial Peripheral Interface)

De acordo com Junior (2012), a tecnologia SPI (*Serial Peripheral Interface*) foi desenvolvida pela Motorola, e é consistida em um barramento full-duplex e síncrono, composto pelos sinais MISO (*Master data Input, Slave data Output*) e MOSI (*Master data Output, Slave data Input*), e sinais de controle SxCLK (*Serial Clock*) e SS (*Slave Select*).

A figura 29 ilustra a interface de comunicação com protocolo SPI.

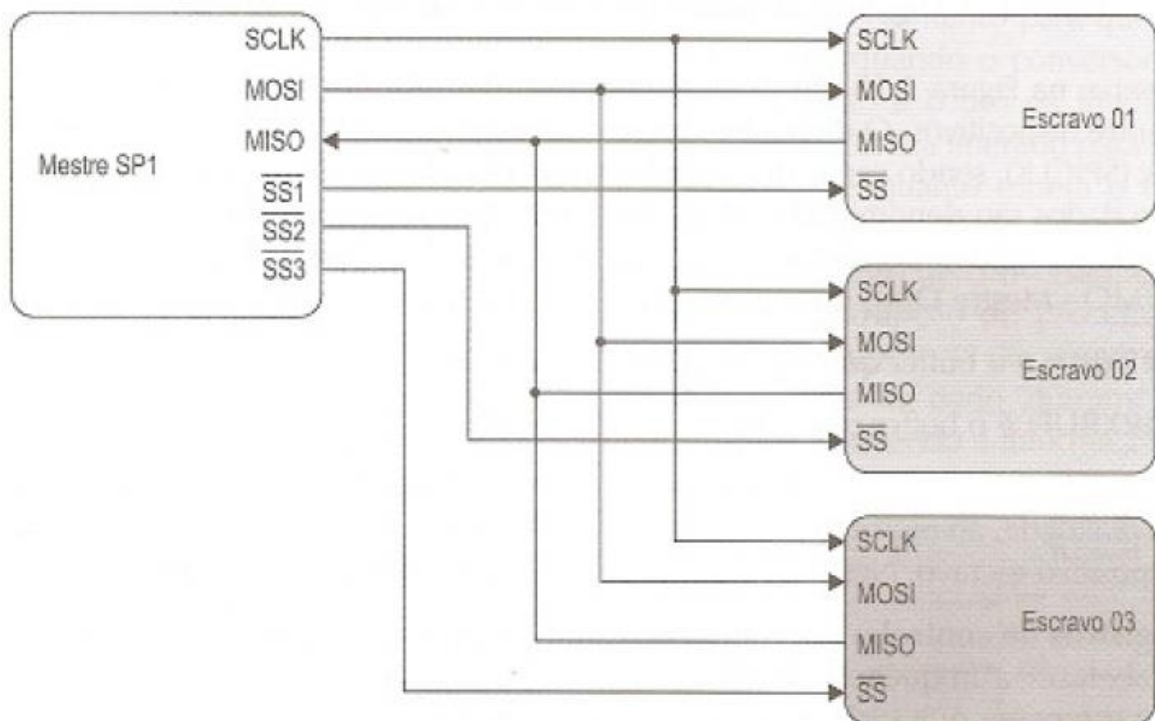
Figura 29. Comunicação SPI.



Fonte: JUNIOR (2012).

Com este protocolo de comunicação, é possível a comunicação entre um mestre e vários escravos. O mestre seleciona o escravo que fará com ele a transmissão de dados através do *Slave Select*, como mostra a figura 30.

Figura 30. SPI com múltiplos escravos.



Fonte: JUNIOR (2012).

O protocolo SPI é mais indicado para aplicações que transferem um fluxo contínuo de dados, já que não possui um mecanismo para confirmar os dados recebidos. A taxa de transmissão pode chegar a até 1 Mbps (JUNIOR, 2012).

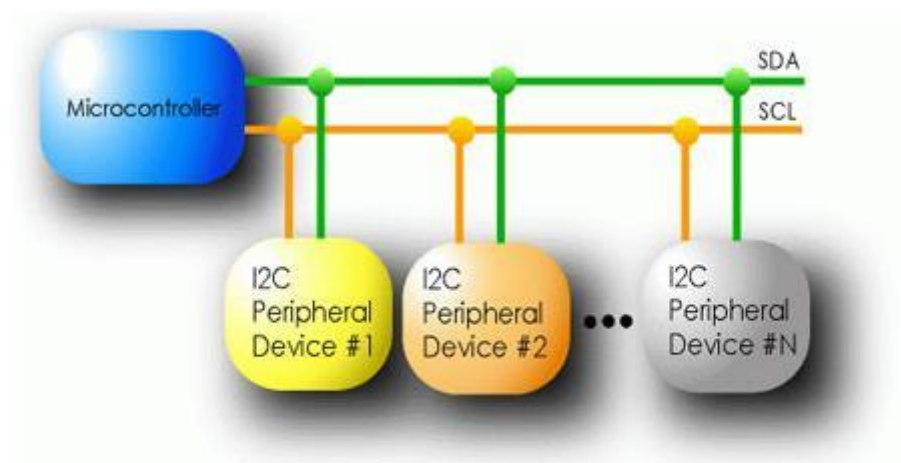
2.7.1.2 Protocolo I²C (Inter-Integrated Circuit)

O barramento I2C foi batizado pela empresa Philips, e possibilita a utilização de uma grande quantidade de componentes, os quais podem realizar várias funções, possibilitando uma troca de dados eficaz entre eles (HENRIQUE, 2012).

Este protocolo permite utilizar, em um mesmo sistema, componentes de tecnologias diferentes sem que haja incompatibilidade e nem conflitos na comunicação.

A transmissão de dados entre os dispositivos é realizada através de 2 fios: DAS (Serial Data) e SCL (Serial Clock). A figura 31 ilustra um barramento I2C com vários dispositivos ligados a ele.

Figura 31. Barramento I2C.



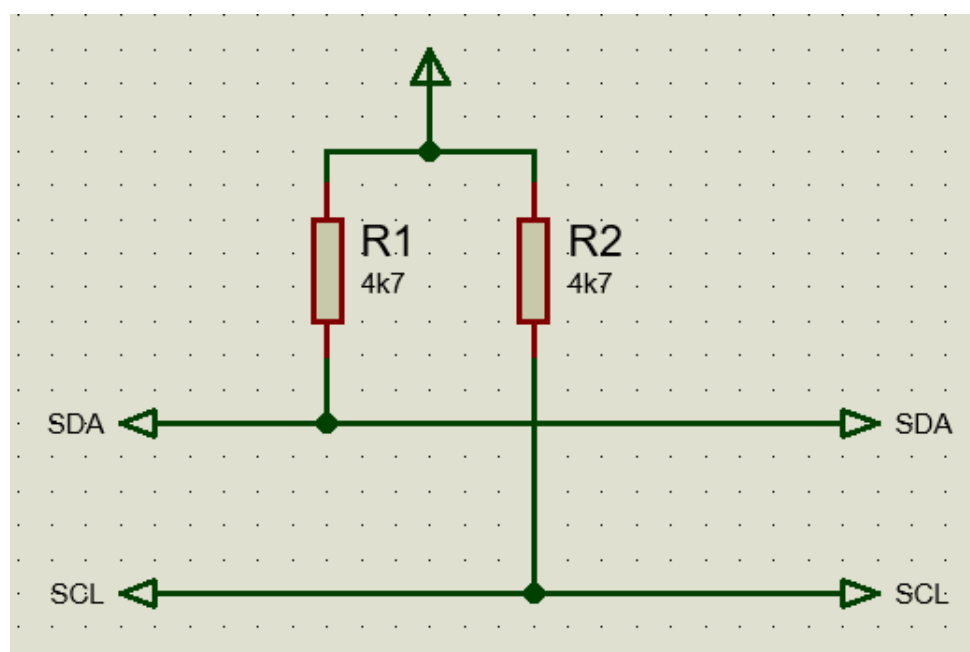
Fonte: HENRIQUE (2012).

Os dispositivos ligados ao I2C possuem um endereço fixo, e pode – se configurá-los para receber ou transmitir dados, sendo classificados como mestres ou escravos.

O barramento é do tipo multi – mestre, ou seja, mais de um dispositivo de controle pode ser conectado a ele.

As linhas SDA e SCL são bidirecionais e devem ser ligados ao positivo da alimentação através de uma fonte de corrente ou de um resistor pull-up, para garantir que as duas permaneçam em nível alto quando não houver dados no barramento. A figura 32 ilustra a ligação dos resistores ao barramento (HENRIQUE, 2012).

Figura 32. Resistores de pull-up no barramento I2C.



Fonte: Autor.

Segundo Henrique (2012), os dados inseridos na linha DAS devem ter comprimento de 1 byte, e o número de bytes que podem ser transferidos em cada operação de transferência é limitado, sendo enviado primeiramente o bit mais significativo.

Após a leitura / escrita de um byte no barramento, o dispositivo receptor gera um bit de reconhecimento (acknowledge), permitindo a fluência da transferência de dados.

No formato de transmissão mais utilizado, o transmissor envia os sinais para um escravo, conforme ilustra a figura 33.

Figura 33. Frame do I2C.



Fonte: Henrique (2012).

1. O mestre inicializa a transmissão através do bit de start (S).
2. O mestre envia o endereço físico do escravo ligado ao barramento I2C. O bit menos significativo desse endereço informa se vai ler ou escrever na memória.
3. Após a recepção do endereço, o escravo envia o acknowledge na linha.
4. O mestre envia os dados para o escravo, que responde cada byte com o bit ACK.
5. O mestre finaliza a comunicação com a condição de stop.

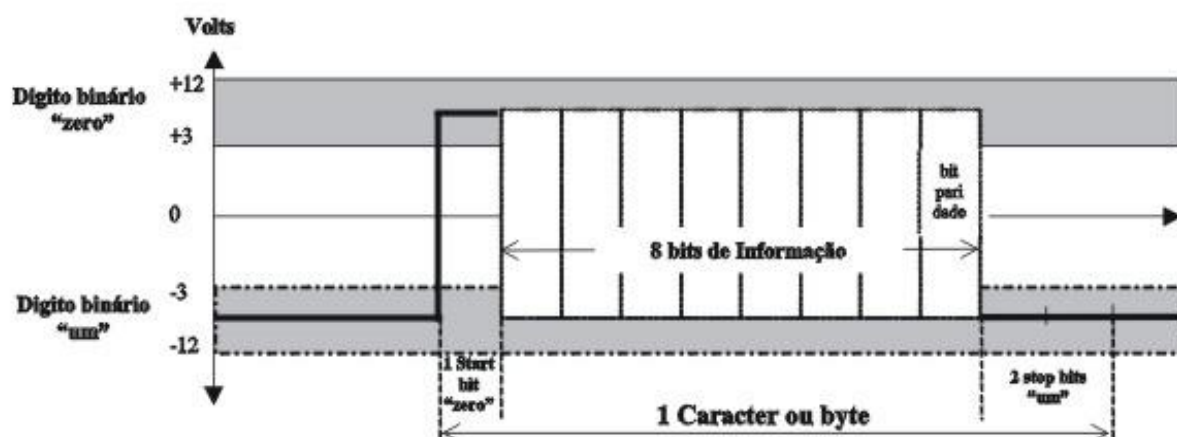
2.7.2 Comunicação Assíncrona

Na comunicação assíncrona, um bit é inserido no início e no fim da transmissão de um byte, e assim permite que o receptor entenda o que realmente foi transmitido.

Com isso, os dados podem ser transmitidos em qualquer ordem, e o receptor irá interpretar esses dados e colocá-los no lugar certo.

Porém, há uma desvantagem, que é a má utilização do canal, pois os caracteres são transmitidos irregularmente, além de os bits de controle adicionados no início e no fim de cada caractere, causando uma baixa eficiência na transmissão de dados. A figura X ilustra um envio de um caractere através de uma comunicação assíncrona (SCHUNCKE, 2013).

Figura 34. Comunicação Assíncrona.

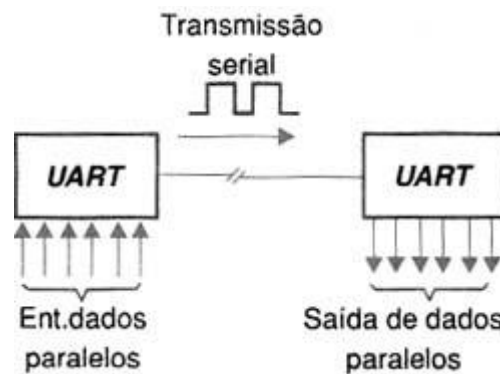


Fonte: Machado (2002).

2.7.2.1 Comunicação UART

UART é o acrônimo de *Universal Asynchronous Receiver/Transmitter* ou Receptor/Transmissor Universal Assíncrono. Ele serve para possibilitar a transmissão e recepção de dados originalmente disponíveis na forma paralela, como ilustrado na figura 35 (<http://home.roboticlab.eu/pt/examples/communication/uart>).

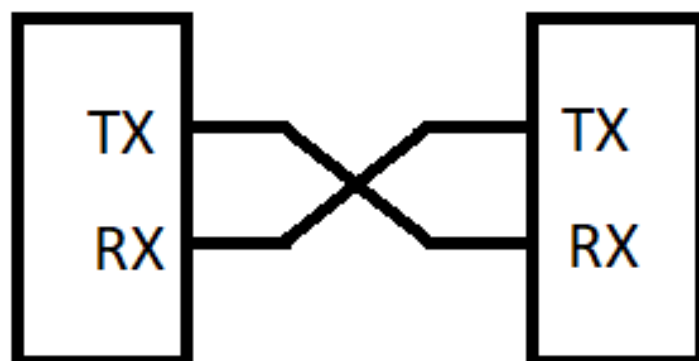
Figura 35. Transmissão UART.



Fonte: (<http://newtoncbraga.com.br/index.php/telecom-artigos/1709->)

Neste tipo de comunicação, há duas linhas ligadas, uma que transmite informação (TX), e outra que recebe informação (RX). As informações sempre se movem em uma direção. A figura 36 ilustra como deve ser feita a ligação entre dois dispositivos via UART.

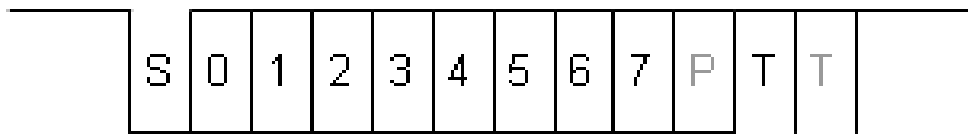
Figura 36. Ligação para comunicação UART.



Fonte: Autor.

A transmissão de dados na UART é feita em quadros (*frame*), onde é possível transmitir dados de 5 a 9 bits, sendo 8 bits o mais comum. Também há bits adicionais que acompanham os bits de dados. O primeiro deles é o bit de início (*start bit*), que é sempre 0, e o bit (ou bits) de parada (*stop bit*), que vem depois dos bits de dados e é sempre antes do bit de paridade. O bit de paridade indica se o número de bits de dados é par ou ímpar. A figura 37 ilustra um *frame* de transmissão via UART (<http://home.roboticlab.eu/pt/examples/communication/uart>).

Figura 37. Frame de transmissão UART.



Fonte: <http://home.roboticlab.eu/pt/examples/communication/uart>.

Outro parâmetro importante é a taxa de transmissão (*baudrate*), que determina o número de bits transmitidos por segundo. A transferência de dados pode ser usada para qualquer taxa de transmissão, sendo os valores mais usados: 9600 bps, 38.400 bps, 57.600 bps, 115.200 bps (<http://home.roboticlab.eu/pt/examples/communication/uart>).

3. Materiais e Métodos

Para que fosse realizado a calibração e aquisição de dados, foi utilizado um hardware da FATEC Santo André como plataforma de desenvolvimento, onde foram feitas alterações nos firmwares dos microcontroladores presentes na placa.

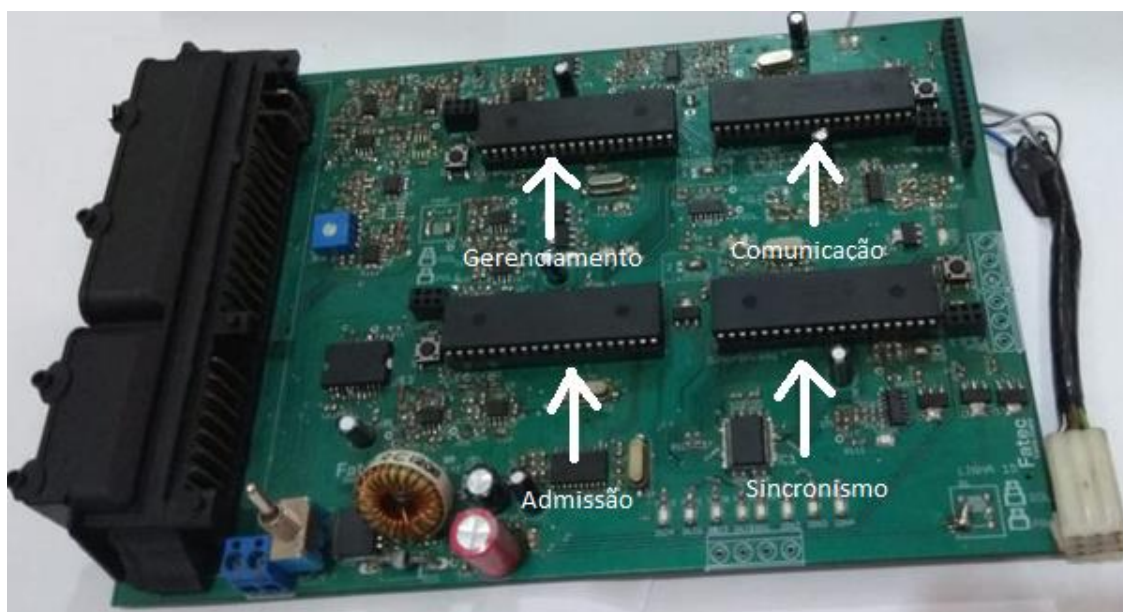
A interface de aquisição e calibração foi desenvolvido pelos autores utilizando o software gratuito QT Creator, programado em C++.

Antes de realizarmos os testes no carro VW POLO 1.6 da FATEC Sorocaba, foram feitos testes em bancada utilizando um gerador de sinais de rotação e fase desenvolvido pelos autores.

3.1 Unidade de Gerenciamento de Desenvolvimento

A unidade de gerenciamento conta com um hardware que foi disponibilizado pela FATEC de Santo André, essa plataforma disponibiliza 04 microcontroladores *Peripheral Interface Controller* (PIC) da família 18F46k22 contendo uma arquitetura de processamento descentralizada em que cada PIC tem sua função pré-estabelecida por versões do firmware que é grava no mesmo e todos trabalham de forma sincronizada para garantir a funcionalidade e comunicação entres eles. A figura 38 apresenta a unidade desenvolvida.

Figura 38. Unidade de Gerenciamento utilizada.



Fonte: Autores.

Como dito anteriormente cada PIC é responsável por um determinado bloco, sendo eles o gerenciamento, admissão, sincronismo e comunicação. A seguir será explicado os blocos mencionados.

3.1.1 Bloco de gerenciamento

O microcontrolador responsável pelo gerenciamento é tratado como o mestre do sistema, é ele que contém informações dos mapas de avanço de injeção e de ignição, contendo os cálculos referente a tempo de injeção e carregamento da bobina. O gerenciamento é responsável também por realizar a leitura de diversos sensores tais como o de temperatura de água, temperatura de ar, fase e MAP, com o objetivo de executar os cálculos correspondentes aos mesmos.

3.1.2 Bloco de admissão

Esse bloco é responsável por fazer o controle da válvula borboleta, para isso ele recebe como parâmetros os valores do sensor de posição do pedal do acelerador (TPS) com isso o sistema atua controlando a válvula borboleta através de um sinal pulsante modulado (PWM). Ressaltando que o valor recebido do sensor de posição do pedal do acelerador é proveniente do PIC comunicação.

3.1.3 Bloco de sincronismo

O microcontrolador responsável pelo sincronismo recebe os parâmetros calculados do gerenciamento através do PIC comunicação, a fim de atuar de forma adequada nas válvulas eletro-injetoras e bobinas de ignição, com o intuito de sincronizar os sinais de rotação, injeção, fase e ignição. Além disso, o gerenciamento também recebe os valores de temperatura calculado pelo gerenciamento para que possa ser acionado os eletro-ventiladores para realizar o arrefecimento do motor.

3.1.4 Bloco de comunicação

O PIC responsável pela comunicação tem por objetivo estabelecer o fornecimento de dados para os demais PIC's, em quesito de comunicação I2C esse microcontrolador é tratado como escravo (*slave*) já que ele é responsável por sempre enviar dados e não executar na leitura de sensores.

Ademais o PIC comunicação é responsável pelo envio de dados para o barramento CAN e para um protocolo de comunicação USB, em que permite se conectar a um computador para que sejam coletados os dados dos sensores pela interface de aquisição e calibração desenvolvida pelos autores.

3.1.5 Protocolo de comunicação entre os microcontroladores

O sistema de comunicação utilizado entre as placas é o I2C, na qual são necessários dois fios de comunicação e ligação em barramento.

Nesta aplicação estão sendo utilizados 2 PICs como Mestres (Gerenciamento e Sincronismo) e 2 PICs como Escravos (Admissão e Comunicação).

O protocolo funciona da seguinte maneira: O Gerenciamento, após realizar todos cálculos, envia todos os dados para o PIC de Comunicação e realiza uma leitura para verificar se houve calibração de parâmetros. Também envia o Set Point do ângulo válvula borboleta eletrônica para o PIC de Admissão, e realiza uma leitura, que contém o ângulo atual da borboleta.

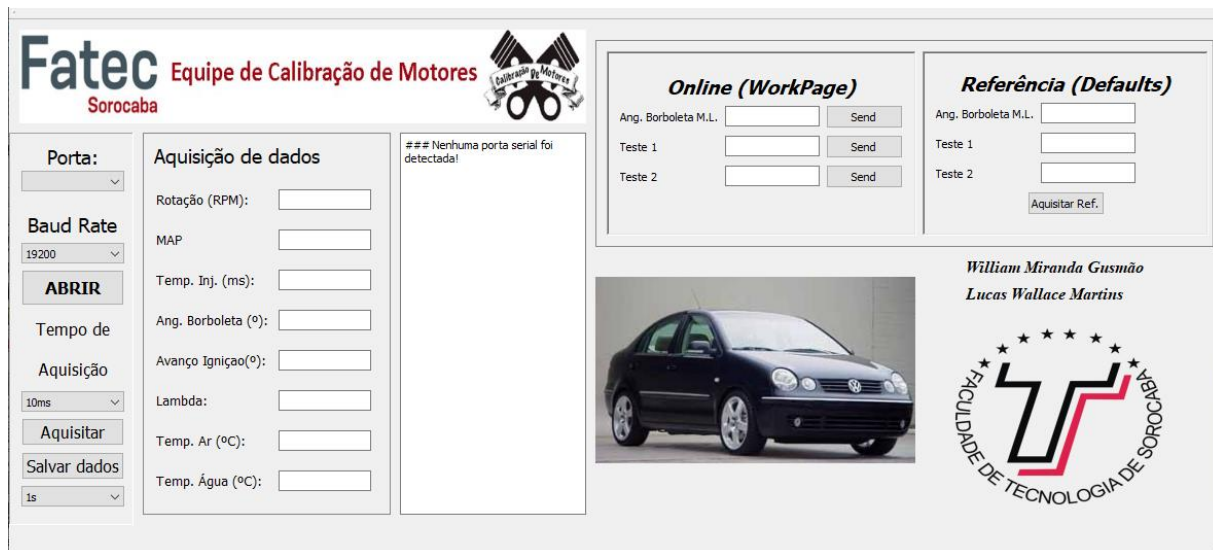
O PIC Sincronismo apenas realiza uma leitura do PIC Comunicação a fim de obter todos os dados necessários para controlar o ponto de ignição e todo o sistema de injeção de combustível.

Para que não houvesse conflito na comunicação havendo dois mestres, foi feita uma sincronização utilizando o sinal da roda fônica: Após o motor completar uma volta completa, o Gerenciamento irá realizar a comunicação com os Escravos, porém, na próxima volta ele ficará apenas realizando os cálculos enquanto o Sincronismo realiza a comunicação.

3.2 Software de Aquisição e Calibração

Para que fosse possível realizar a aquisição dos dados do motor, e calibração de parâmetros dele, foi necessário desenvolver uma interface visual, que pudesse comunicar-se via serial com a ECU. A figura 39 mostra a interface do software desenvolvido.

Figura 39. Interface desenvolvida.



Fonte: Autores.

O software, desenvolvido através da plataforma *Open Source* QT Creator, foi programado de forma que ele possa se conectar à uma entrada serial, que neste projeto é através de uma interface UART, com o microcontrolador de comunicação da ECU utilizando um *baud rate* de 19200 kbps, que é suficientemente rápido para a nossa aplicação.

3.2.1 Aquisição de dados

Estabelecido a comunicação, pode-se configurar o intervalo de tempo entre as aquisições dos seguintes dados: Rotação, Sensor MAP, Tempo de Injeção, Ângulo de borboleta, Avanço de ignição, Sensor Lambda, Temperatura do Ar de admissão e temperatura do líquido de arrefecimento.

Após pressionar o botão “Aquisitar”, a interface enviará com um intervalo de tempo definido acima o caractere “R” seguido do caractere “2” via serial. Com isso, o PIC de Comunicação identificará pelo comando que está sendo requisitado os dados citados acima, e,

então, enviará esses dados para o software. A figura 40 mostra a tela de aquisição de dados da ferramenta desenvolvida.

Figura 40. Tela de Aquisição de dados da interface.

The interface is divided into two main sections. The left section contains configuration options: 'Porta:' with a dropdown menu showing 'COM1', 'Baud Rate' with a dropdown menu showing '19200', an 'ABRIR' button, 'Tempo de Aquisição' with a dropdown menu showing '10ms', an 'Aquisitar' button, a 'Salvar dados' button, and a dropdown menu showing '1s'. The right section is titled 'Aquisição de dados' and lists several parameters with corresponding input fields: 'Rotação (RPM):', 'MAP', 'Temp. Inj. (ms):', 'Ang. Borboleta (°):', 'Avanço Ignição(°):', 'Lambda:', 'Temp. Ar (°C):', and 'Temp. Água (°C):'.

Fonte: Autores.

Para que fosse possível realizar uma análise mais aprofundada do motor, foi inserida a opção de salvar os dados que foram adquiridos em uma planilha de Excel, podendo posteriormente “plotar” esses dados em um gráfico, ou até mesmo modelar funções através de softwares matemáticos. É possível definir o tempo de aquisição que será salvo na planilha. Feito isso, basta pressionar o botão “Salvar Dados”.

Passado o tempo de aquisição configurado para salvar, já será gerado automaticamente uma planilha em Excel na pasta fonte do Software aplicativo. A tabela 1 mostra uma planilha salva após ser realizado o teste no carro didático da FATEC Sorocaba VW POLO 1.6 FLEX com intervalos de aquisição aproximadamente 100ms e o tempo total de aquisição salva foi de 5s.

Obs.: O tempo de injeção está em us(microssegundos);

Devido à falta de alguns componentes que precisavam ser importados e outros problemas, não foi possível realizar a leitura correta da sonda lambda e da temperatura do ar de admissão.

Tabela 1. Dados de aquisição salvos.

Tempo	Rotação	Ângulo	Temperatura		Lambda	Tempo Injeção
			Ar	Água		
53	1649	4.9	76	80	196	5923
164	1642	4.9	76	80	193	5703
276	1636	4.9	76	80	196	5923
372	1640	4.9	76	80	189	5703
500	1640	4.9	76	80	189	5703
596	1640	4.7	76	80	194	5703
708	1646	4.9	76	80	192	5703
819	1653	4.9	76	80	194	5703
931	1653	4.9	76	80	197	5703
1027	1664	5.1	76	80	192	5703
1140	1648	5.1	76	80	197	5703
1252	1657	4.8	76	80	193	5703
1364	1645	4.9	76	80	196	5703
1476	1648	5.1	76	81	192	5703
1587	1639	4.9	76	80	195	5703
1700	1639	4.9	76	80	195	5703
1795	1633	4.9	76	80	197	5703
1908	1638	5.1	76	81	190	5484
2020	1639	4.9	76	81	196	5703
2132	1636	4.9	77	80	192	5923
2245	1628	4.9	77	80	198	5686
2356	1633	5.1	76	80	195	5686
2452	1635	4.9	76	80	197	5703
2579	1635	4.9	76	80	197	5703
2675	1638	5.1	76	80	197	5703
2787	1644	5.1	76	80	195	5703
2900	1622	4.9	76	84	197	5703
3011	1622	4.9	76	84	197	5703
3107	1366	2.7	76	84	195	4387
3219	1227	2.9	76	82	193	4826
3330	1089	2.9	76	82	200	5045
3442	1107	2.9	77	81	201	5265
3555	1110	2.9	76	81	202	5249
3667	1110	2.9	76	81	202	5249
3778	1140	2.9	76	81	201	5265
3875	1158	2.9	76	81	200	5265
3987	1166	3	76	82	200	5265
4100	1182	3	76	82	201	5265
4211	1192	3	76	82	202	5045
4321	1192	3	76	82	202	5045
4435	1192	3	76	82	202	5045
4530	1177	2.8	76	82	199	5030
4643	1179	3	76	82	200	5045

4754	1184	2.9	76	82	201	5045
4866	1183	3	76	82	202	5045
4978	1189	2.9	76	82	201	5045

Fonte: Autores

3.2.2 Calibração Online de Parâmetros

A calibração online consiste em alterar um parâmetro importante do motor com ele já em funcionamento, e observar a resposta do mesmo em função dessa calibração.

Foi reservado um espaço na interface para a calibração do *Set Point* de ângulo de borboleta do motor em Marcha Lenta. Também é possível observar outros 2 espaços (Teste 1 e Teste 2) que foram colocados apenas para facilitar a inserção de outras calibrações futuramente.

O protocolo para a calibração foi desenvolvido da seguinte maneira: O usuário deverá inserir no espaço reservado para calibração o novo *Set Point* do ângulo borboleta em marcha lenta, e pressionar o botão “*Send*”. O software aplicativo primeiramente irá enviar para a ECU o caractere “E” seguido do caractere “1”, e então enviará o valor inserido pelo usuário, que será tratado logo em seguida pelos microcontroladores do módulo.

A figura 41 mostra um exemplo de calibração online do grau de abertura da borboleta eletrônica feita pelo software.

Figura 41. Tela de Calibração Online da interface.

The image shows a software window titled "Online (WorkPage)". Inside, there are three rows of input fields and buttons. The first row is labeled "Ang. Borboleta M.L." and has a text box with the value "5.0" and a "Send" button. The second row is labeled "Teste 1" and has an empty text box and a "Send" button. The third row is labeled "Teste 2" and has an empty text box and a "Send" button.

Fonte: Autores.

3.3 Gerador de sinais de rotação e fase

Para o desenvolvimento do projeto, foi necessário uma série de testes e correções. Porém, realizar os primeiros testes e correções diretamente no carro era muito difícil, pois nem sempre o carro estava disponível para uso, a bateria começava a se descarregar após alguns testes, e caso houvesse algum problema que impedisse o carro de funcionar, toda a lógica de comunicação também não funcionava.

Uma maneira de facilitar este processo foi desenvolver um gerador de sinais de rotação e fase, que foi suficiente para o desenvolvimento da comunicação entre os dispositivos.

3.3.1 Hardware

O hardware utilizado foi a plataforma de desenvolvimento Arduino Uno devido à sua facilidade de programar para acelerar ao máximo o desenvolvimento dispositivo auxiliar, e por já estar inserido na plataforma um conversor UART-USB para que fosse possível controlar a rotação através de um computador, sem a necessidade de adicionar mais componentes ao hardware. A figura 42 mostra a placa utilizada.

Figura 42. Arduino Uno utilizado.



Fonte: Autores.

3.3.2 Software

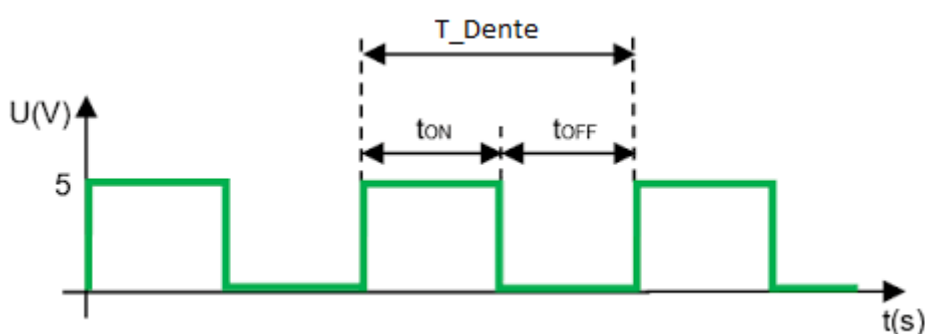
O software foi feito de maneira que o Gerador de sinais disponibilizasse no pino 13 do Arduino um sinal parecido com o sinal da roda fônica (onda quadrada) de um carro, simulando um valor de rotação em RPM configurado via monitor serial.

A lógica começa definindo a rotação através do monitor serial, onde deve-se obrigatoriamente digitar 4 caracteres, por exemplo: “0800” para uma rotação de 800 RPM.

A partir disso, o software realiza os seguintes cálculos:

- Transforma a rotação por minuto em rotação por segundo: $RPS = RPM/60$;
- Como rotações por segundo é considerado uma frequência (Hz), o software calcula o inverso desse valor para obter o tempo por volta em segundos: $t_Volta = 1/RPS$;
- Como uma volta completa da roda fônica possui 60 dentes (nesta parte não se deve considerar a existência da falha), divide-se o t_Volta pela quantidade de dentes:
 $t_Dente = t_Volta/60$;
- Dividindo o tempo t_Dente por 2, é obtido o tempo em que a saída de rotação deverá ficar em nível baixo e nível alto, como mostra a figura 43:
- $ton = toff = t_Dente/2$;

Figura 43. Onda quadrada simulando roda fônica.



Fonte: Autores.

Após isso, a saída é colocada em nível alto, e é mantida assim pelo tempo “ton”, e então colocada em nível baixo e mantida pelo tempo “toff”, realizando esses comandos dentro de um loop.

Utilizando um contador e uma estrutura de condição (IF), após 58 repetições do loop acima, o software simula a “falha” da roda fônica, mantendo a saída em nível baixo pelo tempo $ton+toff$ 2 vezes, e então o contador é zerado e o loop continua.

Para simular o sinal de fase, foi necessário apenas inverter o sinal do pino 12 o Arduino a cada “volta”.

A figura 44 mostra como foi feito no software a lógica que controla os sinais de rotação e fase.

Figura 44. Lógica para gerar sinal de fase e rotação.

```
cont ++;                // Incrementa 1 na variável "cont"
if(cont == 59)          // Se cont for igual a 60
    digitalWrite(fase, !fase); // Inverte o pino que simula a fase
if(cont < 59)           // Se cont for menor que 59
    digitalWrite(saidaRotacao, HIGH); // Coloca o pino de rotacao em nivel Alto
delayMicroseconds(tOn); // Mantém durante o tempo tOn
digitalWrite(saidaRotacao, LOW); // Coloca o pino de rotacao em nivel Baixo
delayMicroseconds(tOff); // Mantem durante o tempo tOff
if(cont == 60)          // Se cont for igual a 60
    cont = 0;           // O valor de "cont" agora é zero
```

Fonte: Autores.

3.4 Instrumentos de Medição

Para a obtenção dos sinais eletrônicos e valores dos sensores e atuadores foram necessário a utilização de algumas ferramentas de medição tais como osciloscópio de 04 canais e multímetro.

3.4.1 Osciloscópio

O osciloscópio é uma ferramenta que nos possibilitou analisar e compreender os sinais eletrônicos de cada sensor através da curva de resposta dele é mostrado na tela do aparelho, sinais como o de fase, rotação, injeção de combustível, tempo de ignição entre outros. É importante ressaltar que o osciloscópio realiza a leitura dos dados em relação ao tempo, tendo uma visão contínua do processo. Para conseguir obter essas informações o osciloscópio necessita de cabos chamados de pontas de prova em que é anexado ao terminal do sensor ou dispositivo desejado para realizar a obtenção dos sinais, a ferramenta utilizada conta com 04 canais o que nos permite analisar 04 curvas de respostas de maneira simultânea. A figura 45 representa a ferramenta citada acima.

Figura 45. Osciloscópio.



Fonte: Autores.

3.4.2 Multímetro

As principais utilizações do multímetro são as medições de tensão (contínua e alternada), corrente e resistência, disponibilizadas em diversas grandezas de escalas. O multímetro conta também com a escala “continuidade” que nos possibilita realizar testes em chicotes de sensores, trilhas de cobre em placas de circuito impresso etc. Assim como o osciloscópio, o multímetro também necessita de pontas de prova para poder ser conectado ao terminal desejado. Na figura 46 podemos observar a ferramenta citada.

Figura 46. Multímetro.



Fonte: Autores.

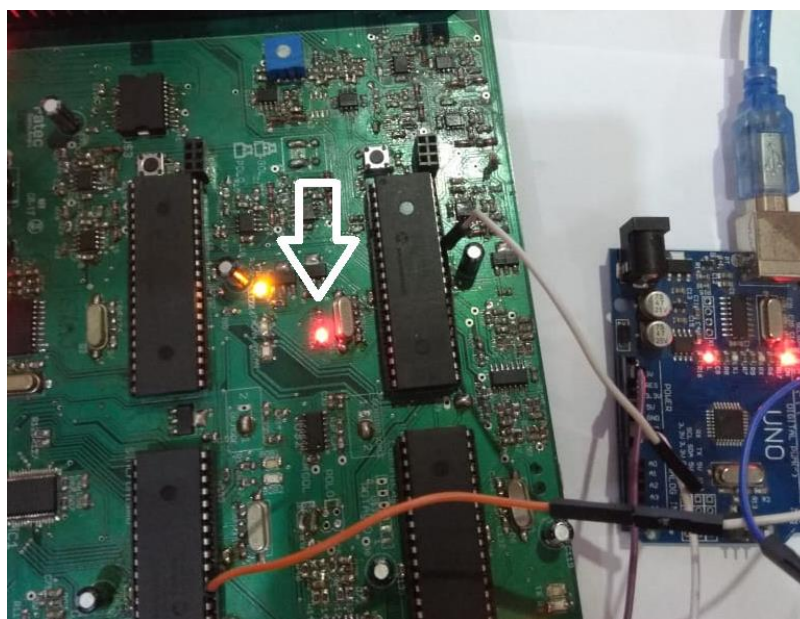
4. Análises e Resultados

Após testes realizados em bancada com o Gerador de Sinais de Rotação e Fase, foi possível observar que a comunicação interna através do protocolo I2C na ECU funcionou de maneira satisfatória, possibilitando a implementação do Sistema de Aquisição de Dados e Calibração Online.

Para realizar a validação da comunicação, o PIC Gerenciamento fez a Leitura de um *array* de dados do PIC de Comunicação. Caso os dados lidos fossem os mesmos que estavam declarados no firmware, ele deveria acender um Led presente na placa.

Como a comunicação I2C funcionou com os sinais gerados pelo Gerador de Sinais de Rotação e Fase, o Led mostrado pela figura 47 acendeu.

Figura 47. Teste e validação da comunicação.



Fonte: Autores.

Com o sistema de comunicação entre os microcontroladores funcionando adequadamente, já era possível realizar os testes do sistema de aquisição de dados. Neste sistema, o PIC Gerenciamento envia para o PIC Comunicação os seguintes dados: grau de avanço de ignição, tempo de injeção, estado do controle de marcha lenta, grau de avanço de injeção, tempo de carregamento da bobina, temperatura do líquido de arrefecimento, Set point do ângulo da borboleta, rotação do motor, pressão absoluta do ar de admissão, ângulo atual da válvula borboleta, temperatura do ar de admissão, valor lambda e tensão da bateria.

Com estes dados já presentes no microcontrolador que faz a comunicação, utilizando o protocolo I2C já explicado anteriormente, e juntamente com o software aplicativo, foram feitos testes em bancada.

Configurando uma rotação de 800 RPM no Gerador de Sinais de Rotação e Fase, o valor de rotação aquisitado variava entre 764 e 769 rpm. Houve uma pequena diferença, que pode ser explicada pelo fato de o tempo ton e toff serem muito pequenos, sendo facilmente influenciados pelo tempo das funções utilizadas pelo Arduino, diminuindo a precisão. A grande quantidade de comandos no PIC Gerenciamento que realiza a leitura de rotação também pode ser um fator que favoreça esta diferença.

A figura 48 mostra uma aquisição pelo software aplicativo com o Arduino gerando 800 rpm.

Figura 48. Aquisição utilizando gerador de rotação e fase.

The screenshot displays a software interface for data acquisition. On the left, a panel titled 'Aquisição de dados' contains several input fields with their current values: Rotação (RPM) is 769, MAP is 1, Temp. Inj. (ms) is 0, Ang. Borboleta (°) is 0, Avanço Ignição(°) is empty, Lambda is 99, Temp. Ar (°C) is 140, and Temp. Água (°C) is 113. In the center, a text area shows serial port status messages: '### Porta serial pronta para ser utilizada.', '### Porta serial aberta com sucesso!', and '### Tempo de aquisição: 100ms.'. On the right, a panel labeled 'COM3' shows a baud rate of '0800' and two lines of real-time data: 'Rotacao: 700' and 'Rotacao: 800'. At the bottom right, there are checkboxes for 'Auto-rolagem' (checked) and 'Show timestamp' (unchecked).

Fonte: Autores.

Após validar o funcionamento do sistema de aquisição de dados, foram feitos testes no carro didático da FATEC Sorocaba VW POLO 1.6 FLEX, onde o combustível dele estava com 100% de etanol.

Com o carro funcionando foi possível aquisitar com êxito os seguintes dados: Rotação do motor, pressão absoluta do ar de admissão, ângulo de borboleta, temperatura da água, tempo de injeção e grau de avanço de ignição. O software aplicativo foi desenvolvido de maneira a mostrar também a temperatura do ar e o valor lambda. Porém, após testes e conversas com as

últimas pessoas a trabalharem com a ECU de desenvolvimento, descobrimos que os mapas de temperatura do ar nunca foram calibrados, e como dito anteriormente, estavam faltando alguns componentes que possibilitariam a leitura do sensor lambda.

A figura 49 mostra a tela de aquisição feita com o carro em funcionamento.

Figura 49. Aquisição de dados do motor do VW POLO 1.6 FLEX.

Aquisição de dados	
Rotação (RPM):	1110
MAP	26
Temp. Inj. (ms):	5.2
Ang. Borboleta (°):	2.9
Avanço Ignição(°):	6
Lambda:	255
Temp. Ar (°C):	76
Temp. Água (°C):	81

Fonte: Autores.

Com a aquisição funcionando, o teste de calibração foi feito diretamente no carro didático, pois só assim seria possível ver uma resposta a alteração de um parâmetro, e verificar se a calibração online funcionou corretamente.

Com o protocolo de comunicação explicado anteriormente (3.2.2), aquisição de dados a cada 100ms ligada, e o carro funcionando em marcha lenta, foi alterado o *Set Point* de abertura da borboleta eletrônica, que estava programada para 2,5° de abertura, para 5° de abertura.


Instantaneamente após ser enviado o novo valor de abertura da válvula borboleta, o ângulo aquisitado pelo software também foi alterado, e a rotação subiu. Também foi possível observar que o tempo de injeção teve um aumento com esta calibração.

Após ser enviado o valor de 2,5°, o ângulo de borboleta retornou ao primeiro valor, ajustando o valor de rotação e tempo de injeção próximo ao que estava anteriormente.

As figuras 50 e 51 mostram a interface do software aplicativo antes e depois de calibrar um valor.

Figura 50. Interface antes da calibração.

Fatec
Sorocaba

Equipe de Calibração de Motores


Porta:
COM4

Baud Rate
19200

FECHAR

Tempo de Aquisição
100ms

PARAR

Salvar dados

1s

Aquisição de dados
Rotação (RPM): 1107
MAP: 25
Temp. Inj. (ms): 5.1
Ang. Borboleta (°): 2.7
Avanço Ignição(°): 5
Lambda: 255
Temp. Ar (°C): 75
Temp. Água (°C): 84


```


### Porta serial pronta para ser
utilizada.
### Porta serial aberta com
sucesso!
### Tempo de aquisição: 100ms.

```

Online (WorkPage)
Ang. Borboleta M.L.: Send
Teste 1: Send
Teste 2: Send

Referência (Defaults)
Ang. Borboleta M.L.:
Teste 1:
Teste 2:
Aquisitar Ref.




William Miranda Gusmão
Lucas Wallace Martins


Fonte: Autores

Figura 51. Interface depois da calibração.

Fatec
Sorocaba

Equipe de Calibração de Motores


Porta:
COM4

Baud Rate
19200

FECHAR

Tempo de Aquisição
100ms

PARAR

Salvar dados

1s

Aquisição de dados
Rotação (RPM): 1654
MAP: 25
Temp. Inj. (ms): 5.9
Ang. Borboleta (°): 5.1
Avanço Ignição(°): 6
Lambda: 255
Temp. Ar (°C): 75
Temp. Água (°C): 84


```


### Porta serial pronta para ser
utilizada.
### Porta serial aberta com
sucesso!
### Tempo de aquisição: 100ms.
### A aquisição de dados foi
Interrompida!

```

Online (WorkPage)
Ang. Borboleta M.L.: 5.0 Send
Teste 1: Send
Teste 2: Send

Referência (Defaults)
Ang. Borboleta M.L.:
Teste 1:
Teste 2:
Aquisitar Ref.



William Miranda Gusmão
Lucas Wallace Martins


Fonte: Autores

5. Conclusão

Neste projeto foi desenvolvido um sistema de aquisição de dados e calibração online de parâmetros, onde é possível calibrar o módulo de gerenciamento do motor e verificar sua resposta às alterações em tempo real, facilitando o processo de gerenciamento de um motor a combustão. Estes procedimentos são utilizados em grandes indústrias automobilísticas, porém, devido aos segredos que as montadoras desejam manter, não possuem bibliografia de fácil acesso.

O sistema desenvolvido funcionou de maneira satisfatória, onde foi possível adquirir quase todos os dados planejados e salvá-los em uma planilha, com exceção da temperatura do ar e valor lambda devido à falta de alguns componentes e alguns outros fatores descritos anteriormente. Entretanto, eles não foram um grande peso no desenvolvimento do projeto, e podem ser implementados futuramente.

Também foi possível realizar uma calibração em tempo real (online), alterando o *Set Point* do ângulo da válvula borboleta eletrônica com o carro em marcha lenta, e observar instantaneamente uma resposta do motor a esta mudança, aumentando a rotação e o tempo de injeção.

Concluimos que é possível a aquisição de dados e a calibração online para gerenciamento do módulo eletrônico de um motor a combustão interna em ambiente acadêmico, utilizando uma ECU projetada para desenvolvimento e estudos.

Em estudos futuros, pode-se implementar as aquisições da sonda lambda e temperatura do ar e acrescentar mais parâmetros a serem calibrados. Também é possível utilizar a ferramenta desenvolvida para gerar funções e modelos matemáticos em softwares como MATLAB, e com base nos dados adquiridos, calibrar o gerenciamento do motor para um melhor desempenho.

REFERÊNCIAS

- ALCÂNTARA DIAS, B. M. **Unidade microcontroladora para gerenciamento eletrônico de um motor de combustão interna ciclo Otto**. São Paulo. 2015. 269p. (Mestrado) Escola Politécnica, Universidade de São Paulo, São Paulo, 2015.
- BASSHUYSEN, R. V. **Internal Combustion Engine Handbook**. 1.ed. SAE International. 2004, 868p.
- SCHULZ, D. **MOTORES DE COMBUSTÃO INTERNA**. 2009. Disponível em: <https://www.if.ufrgs.br/~dschulz/web/combustao_interna.htm>. Acesso em: 10 abril 2019.
- HEYWOOD, J. B. **Internal Combustion Engine Fundamentals**. McGraw-Hill, 1988.
- CAPELLI, Alexandre. **Eletrônica automotiva: Injeção Eletrônica, Arquitetura do Motor e Sistemas Embarcados**. São Paulo: Érica, 2010.
- BRAGA, G. T. **Uma Contribuição ao Controle de Motores de Combustão Interna de Ignição por Centelha na Condição de Marcha Lenta**. 2007. Dissertação de Mestrado – Universidade Feral de Minas Gerais, Belo Horizonte – MG, 2007.
- CHAVES, M. H. **UNIDADE DE CONTROLE DE MOTORES DE COMBUSTÃO INTERNA BASEADA EM MICROCONTROLADOR E FPGA**. 2016. Dissertação de Mestrado – Instituto de Energia e Ambiente da Universidade de São Paulo, São Paulo – SP, 2016.
- GUEDES, M.A.C. **Estudo de compensadores para a válvula borboleta eletrônica automotiva**. 2014. 155 f. Dissertação de Mestrado – Escola Politécnica da Universidade de São Paulo, São Paulo, 2014.
- RUIZ, R. P. M. **Gerenciamento eletrônico de um motor flex por meio de unidade microcontrolada com arquitetura descentralizada**. 197p. Dissertação de Mestrado – Escola Politécnica da Universidade de São Paulo, São Paulo, 2019.
- YIN, R. K. **Estudo de caso: planejamento e métodos**. 2 ed. Porto Alegre: Bookman, 2001.
- MIX AUTO. **O que é, como funciona e onde comprar Corpo de Borboleta**. 2017. Disponível em : <<http://blog.mixauto.com.br/o-que-e-como-funciona-e-onde-comprar-corpo-de-borboleta/>>. Acesso em: 18 março, 2019.
- ALBALADEJO, F. S. **Desenvolvimento de uma unidade de gerenciamento eletrônico para motores de combustão interna do ciclo Otto**. 2013. Dissertação de Mestrado - Escola Politécnica da Universidade de São Paulo, 2013.

STEBEL, S. L.; SATO, G. Y. **EL 71D – Introdução à Engenharia Mecatrônica**. 2017. Disponível em: <http://paginapessoal.utfpr.edu.br/sato/material-didatico/engenharia-mecatronica/material-da-disciplina-de-introducao-a-engenharia-mecatronica/20170327_sensores.pdf>. Acesso em: 1 maio 2019.

Qual é o significado de Microcontroladores PIC. Disponível em: <<http://ptcomputador.com/Ferragens/computer-drives-storage/46821.html>>. Acesso em: 17 maio 2019.

MICROCONTROLLERS LAB. **PIC18F46K22 microcontroller introduction and features**. 2017. Disponível em: <<https://microcontrollerslab.com/pic18f46k22-microcontroller-introduction/>>. Acesso em: 7 jun. 2019.

ASSIS, P. D. K. B. **Microcontrolador**. 2004. 92f. Trabalho de Conclusão de Curso – Universidade Presidente Antônio Carlos, Barbacena, 2004.

GIRALDELI, F. Algoritmos e Linguagem de Programação em C. v1.4.1. Campus Serra, Espírito Santo, 2011.

CASAVELLA, E. **O que é Linguagem C?**. Disponível em: <<http://linguagemc.com.br/o-que-e-linguagem-c/>>. Acesso em: 17 maio 2019.

FUNDAMENTOS DA LINGUAGEM C. Disponível em: <<https://monografias.brasilecola.uol.com.br/computacao/fundamentos-linguagem-c.htm>>. Acesso em: 18 maio 2019.

GONÇALVES, C. **Fundamentos da Linguagem C**. 2003. Disponível em: <<http://deinf.ufma.br/~cav/c-basico.ppt>>. Acesso em: 18 maio 2019.

PACIEVITCH, Y. **C++**. Disponível em: <<https://www.infoescola.com/informatica/cpp/>>. Acesso em: 24 maio 2019.

O que é C++?. Disponível em: <<https://www.portalgsti.com.br/cplusplus/sobre/>>. Acesso em: 3 jun. 2019.

REIS, F. **Conceitos de Programação Orientada a Objetos**. 2015. Disponível em: <<http://www.bosontreinamentos.com.br/analise-de-sistemas/conceitos-de-programacao-orientada-a-objetos/>>. Acesso em: 3 jun. 2019.

What is QT. Disponível em: <www.qt.io>. Acesso em: 4 jun. 2019.

SIMÕES, H. M. C. **CARTÕES RFID E O MÓDULO MFRC522**. Disponível em: <<http://blog.moduloeletronica.com.br/cartoes-rfid-e-o-modulo-mfrc522/>>. Acesso em: 6 jun. 2019.

JUNIOR, H. T. **Estudo dos protocolos de comunicação das arquiteturas eletroeletrônicas automotivas, com foco nas suas características e respectivas aplicações, visando o direcionamento para o uso adequado e customizado em cada categoria de veículo.** 2012. 97f. Monografia de Pós-Graduação – Escola de Engenharia Mauá do Centro Universitário do Instituto Mauá de Tecnologia, São Caetano do Sul - SP, 2012.

HENRIQUE, T. **Comunicação I2C.** 2012. Disponível em: <<http://microcontrolandos.blogspot.com/2012/12/comunicacao-i2c.html>>. Acesso em: 7 jun. 2019.

MACHADO, M.; MACEDO, P. **Comunicação Local Assíncrona RS-232.** 2002. Disponível em: <<http://www4.pucsp.br/~victor/disciplinas2002/redes2002/t2a1-03.ppt>>. Acesso em: 7 jun. 2019.

Como funcionam as UARTs (TEL006). Disponível em: <<http://newtoncbraga.com.br/index.php/telecom-artigos/1709->>. Acesso em: 7 jun. 2019.
UART. 2018. Disponível em: <<http://home.roboticlab.eu/pt/examples/communication/uart>>. Acesso em: 8 jun. 2019.

O básico sobre os Microcontroladores – parte 1 (MIC139). Disponível em: <<http://www.newtoncbraga.com.br/index.php/electronica/52-artigos-diversos/13263-obasico-sobre-os-microcontroladores-parte-1-mic139>>. Acesso em: 17 maio 2019.

FORESTI, H. B. **Microcontroladores.** 2013. Disponível em: <<http://www.roboliv.re/conteudo/microcontroladores>>. Acesso em: 10 maio 2019.

Milhor, E. C. **Sistema de desenvolvimento para controle eletrônico dos motores de combustão interna ciclo Otto.** 2002.

BOSCH. **Linha de Ignição Bosch.** 2013. Disponível em : <<http://vdm.com.br/v1/wp-content/uploads/2013/11/BOSCH-IGNICAO.pdf>>. Acesso 09 junho, 2019.

DENTON, Tom. **Automobile Electrical and Electronic Systems.** 2.ed. Reino Unido: Arnold, 2000.

MILHOR, Carlos Eduardo. **Sistema de Desenvolvimento para Controle Eletrônico dos Motores de Combustão Interna Ciclo Otto.** Dissertação apresentada à escola de Engenharia de São Carlos da Universidade de São Paulo. Orientador: Prof. Dr. Luis Carlos Passarini. São Carlos, 2002. 88p.

MARTINS, Jorge. **Motores de Combustão Interna.** 1.ed. Portugal: Publindústria Edições Técnicas, 2005. 437p.

BOSCH. **Gasoline Engine Management: systems and components**. Friedrichshafen: Springer Vieweg, 2015, 354p.

BOSCH. **Manual de Tecnologia Automotiva**. 25. Ed. São Paulo: Edgard Blucher, 2005. 1232p.

BOSCH (1998). GmbH. **Automotive eletric/eletronic systems**. 2.ed. Warrendale, P.A.

AHMED, S. A.; ABBAS, K. **Design of Eletronic Throttle Valve Position Control System using Nonlinear PID Controller**. International Journal of Computer Application, Baghdad, Iraque, v. 59, n. 15, p. 975-8887, dez. 2012.

VOLKSWAGEN (Brasil). **Sistema de Gerenciamento de Motores Aspirados**. São Bernardo do Campo: Academia Volkswagen, 2009. 96p. Apostila técnica para treinamento do produto.

REBOLO, V. L.; TURINI, K. R.; TAKEUCHI, M. H. **Desenvolvimento de sistema de comunicação dedicado em uma unidade de gerenciamento automotivo**. 2018. Trabalho de Conclusão de Curso – FATEC Sorocaba, Sorocaba – SP, 2018.

HIROKI, G. B.; JUNIOR, E. I. A.; JATO, F. **DESENVOLVIMENTO DE UMA UNIDADE DE GERENCIAMENTO ELETRÔNICO PARA MOTOR VOLKSWAGEN 1.6L**. 2016. Trabalho de Conclusão de Curso – FATEC Santo André, Santo André – SP, 2016.

MTE-THOMSON. **Sensor de Rotação, Fase e Velocidade**. 2019. Disponível em: <<https://www.mte-thomson.com.br/produtosmte/sensor-de-rotacao/>>. Acesso em: 25 abril 2019.

BRUNETTI, Franco. **Motores de Combustão Interna**. 3. ed. São Paulo: Blucher, 2012. 553 p.

SCHUNCKE, Alex. **“Comunicação de dados Síncrona x Assíncrona”**. Disponível em: Acesso em 1 maio 2019.

MTE-THOMSON. **Aula 13 – VÁLVULA INJETORA**. 2019. Disponível em: <<https://cursosonline.mte-thomson.com.br/unit/aula-13-valvula-injetora/>>. Acesso em: 8 jun. 2019.

GUSTAVO. **3 Problemas Que Você Pode Ter Com a Bomba de Combustível [+ Nova Fraude dos Postos]**. 2017. Disponível em: <<https://doutormultas.com.br/bomba-combustivel/>>. Acesso em: 4 abril 2019.

XAVIER, F. **Sensores de detonação (KS ou Knock Sensor)**. 2015. Disponível em :
<<https://pt.slideshare.net/sergiofavarol/sensores-de-detonaao>>. Acesso em: 7 abril 2019.