

**MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
SEÇÃO DE ENGENHARIA ELÉTRICA**

**1º Ten DANTERNEI LUCAS DO NASCIMENTO JUNIOR
1º Ten VERÔNICA LUIZA LADEIRA DE SOUZA**

**IMPLEMENTAÇÃO DA ELETRÔNICA EMBARCADA NA VIATURA
BAJA DO EXÉRCITO BRASILEIRO**

RIO DE JANEIRO

2014

INSTITUTO MILITAR DE ENGENHARIA

1º Ten DANTERNEI LUCAS DO NASCIMENTO JUNIOR

1º Ten VERÔNICA LUIZA LADEIRA DE SOUZA

**IMPLEMENTAÇÃO DA ELETRÔNICA EMBARCADA NA VIATURA
BAJA DO EXÉRCITO BRASILEIRO**

Projeto de Fim de Curso (PROFIC) apresentado ao Curso de Graduação em Engenharia Elétrica, como requisito para obtenção da graduação do título de Engenheiro Eletrônico do Instituto Militar de Engenharia.

Orientador: Amarildo Teodoro da Costa – D.Sc., UFF

Rio de Janeiro

2014

c2014

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80 – Praia Vermelha
Rio de Janeiro - RJ CEP: 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade dos autores e do orientador.

NASCIMENTO JR., Danternei Lucas e DE SOUZA, Verônica Luiza Ladeira

Projeto de Fim de Curso (PROFIC) apresentado ao Curso de Graduação em Engenharia Elétrica, como requisito para obtenção da graduação do título de Engenheiro Eletrônico do Instituto Militar de Engenharia.

Projeto de Fim de Curso – Instituto Militar de Engenharia, 2014

1. Eletrônica Embarcada; 2. Viatura Baja; 3. Microcontrolador; 4. Barramento CAN

INSTITUTO MILITAR DE ENGENHARIA

1º Ten DANTERNEI LUCAS DO NASCIMENTO JUNIOR

1º Ten VERÔNICA LUIZA LADEIRA DE SOUZA

**IMPLEMENTAÇÃO DA ELETRÔNICA EMBARCADA NA VIATURA
BAJA DO EXÉRCITO BRASILEIRO**

Projeto de Fim de Curso (PROFIC) apresentado ao Curso de Graduação em Engenharia Elétrica, como requisito para obtenção da graduação do título de Engenheiro Eletrônico do Instituto Militar de Engenharia.

Orientador: Amarildo Teodoro da Costa – D.Sc., UFF

Aprovada em 05 de junho de 2014 pela seguinte Banca Examinadora:

Prof. Amarildo Teodoro da Costa – D.Sc., UFF

TC Mauro Cezar Rebello Cordeiro – D.Sc., PUC

Maj Ricardo Teixeira da Costa Neto – D.Sc., PUC

Rio de Janeiro

2014

SUMÁRIO

LISTA DE ILUSTRAÇÕES	7
1 INTRODUÇÃO	10
1.1 Objetivos	10
1.2 Contextualização	10
1.2.1 Projeto Baja SAE.....	10
1.2.2 Competição Baja SAE.....	10
1.2.3 Equipe Víbora.....	11
1.3 Justificativa.....	11
1.4 Metodologia	12
1.5 Estrutura da Monografia.....	13
2 REVISÃO BIBLIOGRÁFICA	14
2.1 Placa de Desenvolvimento	14
2.2 Conversão Analógico-Digitais	16
2.3 Interrupções	18
2.4 Parâmetros	18
2.4.1 Velocidade Instantânea.....	19
2.4.2 Temperatura do Motor.....	20
2.4.3 Nível do Combustível.....	22
2.5 Barramento CAN.....	23
2.6 O Microcontrolador 18F2580.....	26
2.7 O Transceiver MCP2551	28
3 PROJETOS	30
3.1 Hardware	30

3.2 Software	31
4 CRONOGRAMA	32
5 CONCLUSÃO	33
6 REFERÊNCIAS BIBLIOGRÁFICAS.....	34
7 APÊNDICES	35
7.1 Programa de Implementação na Placa de Desenvolvimento	35
7.2 Programa de Implementação no Barramento CAN.....	37
7.2.1 Display_LCD.c.....	37
7.2.2 Sensor_temperatura.c	41
7.2.3 Sensor_combustível.c.....	43

LISTA DE ILUSTRAÇÕES

FIG 2.1.1 - Pinagem do microcontrolador dsPIC33EP512MU810	14
FIG 2.1.2 - Bootloader para programação do dsPIC	15
FIG 2.1.3 - Característica Principais da Placa de Desenvolvimento	16
FIG 2.2.1 - Etapas da conversão analógico-digital	17
FIG 2.4.1.1 - Sensor <i>reed switch</i>	19
FIG 2.4.1.2 - Resistência de <i>pull-up</i> no sensor	20
FIG 2.4.2.1 - Sensor de temperatura LM35DZ.....	21
FIG 2.4.3.1 - Localização do sensor	22
FIG 2.4.3.2 - Sensor de boia de combustível	22
FIG 2.5.1 - Terminação de barramento padrão.....	24
FIG 2.5.2 - Velocidade de transmissão do barramento CAN	24
FIG 2.5.3 - Nível do sinal no barramento CAN.....	25
FIG 2.5.4 - Exemplo da passagem e aceitação dos dados entre os nós	26
FIG 2.6.1 - O PIC 18F2580	27
FIG 2.6.2 - O PIC 18F2580 com o controlador CAN embutido.....	27
FIG 2.6.3 - Esquemático do PIC 18F2580.....	28
FIG 2.7.1 - O <i>transceiver</i> fazendo a interface entre o barramento e o controlador CAN.....	29
FIG 2.7.2 - Esquemático do MCP 2551	29
FIG 3.1.1 - Diagrama de blocos do projeto utilizando barramento CAN.....	30
FIG 3.1.2 - Foto da implementação do barramento CAN.....	31

RESUMO

A Competição Anual SAE Brasil de Mini Baja reúne faculdades de todo o Brasil, com o objetivo de promover o desenvolvimento acadêmico de alunos de engenharia junto ao setor automotivo. Este trabalho apresenta um estudo da eletrônica embarcada com o intuito de implementá-la na viatura Baja do Exército Brasileiro, a qual pretende participar da competição SAE Brasil. Além do estudo da eletrônica embarcada, a qual envolve o trabalho com microcontroladores e uma Placa de Desenvolvimento, realizou-se o estudo e a implementação do barramento CAN, uma ferramenta bastante utilizada nos veículos automotivos atuais. Tal implementação seria única dentro do conjunto de viaturas Baja existentes no Brasil. Dessa forma, o trabalho tem por finalidade, iniciar o estudo da eletrônica embarcada na viatura Baja do Exército Brasileiro de tal forma que ela possa ser implementada, de maneira que atinja tanto os requisitos eletrônicos, como também os mecânicos.

ABSTRACT

The annual SAE Mini Baja Brazil Competition meets colleges throughout Brazil, with the objective to promote the academic development of engineering students with the automotive sector. This paper presents a study of automotive electronics in order to implement it in the car Baja Brazilian Army, which intends to participate in the SAE Brazil competition. Besides the study of automotive electronics, which involves working with microcontrollers and a board of Development, held the study and implementation of the CAN bus, a tool widely used in current automotive vehicles. Such an implementation would be unique within the set of existing Baja cars in Brazil. Thus, the study aims to initiate the study of electronics in the vehicle Baja Brazilian Army so that it can be implemented in a manner that achieves both electronic requirements, as well as the mechanics.

1 INTRODUÇÃO

1.1. Objetivos

Este trabalho tem como objetivos o estudo e a implementação de uma Eletrônica Embarcada na viatura Baja do Exército Brasileiro, a qual é desenvolvida pela Equipe Víbora, formada por diversos alunos do Instituto Militar de Engenharia.

O trabalho tem por finalidade o estudo do conceito geral de Eletrônica Embarcada, estudar a viabilidade de implementação deste trabalho na viatura Baja, estudar os conceitos de programação aplicada a microcontroladores, estudar e implementar os conceitos de barramento CAN e a implementação da Eletrônica Embarcada na viatura Baja.

1.2. Contextualização

1.2.1. Projeto Baja SAE

O projeto Baja SAE é um desafio lançado aos estudantes de engenharia que oferece a chance de aplicar na prática os conhecimentos adquiridos em sala de aula, visando incrementar sua preparação para o mercado de trabalho. Ao participar do projeto Baja SAE, o aluno se envolve com um caso real de desenvolvimento de projeto, desde sua concepção, projeto detalhado e construção. No Brasil o projeto recebe o nome de Projeto Baja SAE BRASIL.[1]

1.2.2. Competição Baja SAE

O projeto Baja SAE foi criado na Universidade da Carolina do Sul, Estados Unidos, sob a direção do Dr. John F. Stevens, sendo que a primeira competição ocorreu em 1976. O ano de 1991 marcou o início das atividades da SAE BRASIL, que em 1994, lançava o Projeto Baja SAE BRASIL. No ano seguinte, em 1995, era realizada a primeira competição nacional, na pista Guido Caloi, bairro do Ibirapuera, cidade de São Paulo. No ano seguinte a competição foi

transferida para o Autódromo de Interlagos, onde ficaria até o ano de 2002. A partir de 2003 a competição passou a ser realizada em Piracicaba, interior de São Paulo, no ECPA – Esporte Clube Piracicabano de Automobilismo. Desde 1997 a SAE BRASIL também apoia a realização de eventos regionais do Baja SAE BRASIL, através de suas Seções Regionais. Desde então dezenas de eventos foram realizados em vários estados do país como Rio Grande do Sul, São Paulo, Minas Gerais e Bahia. [1]

Os alunos que participam do Projeto Baja SAE devem formar equipes que representarão a Instituição de Ensino Superior ao qual estão ligados. Estas equipes são desafiadas anualmente a participar da Competição Baja SAE, evento que reúne os estudantes e promove a avaliação comparativa dos projetos. No Brasil a competição nacional recebe o nome de Competição Baja SAE BRASIL. Competições regionais são nomeadas como Etapa Sul, Sudeste e Nordeste.

1.2.3. Equipe Víbora

A equipe “Víbora”, formada por um grupo de alunos do Instituto Militar de Engenharia - IME, todos alunos da graduação de Engenharia Mecânica, possui como principal meta a participação no desafio BAJA SAE BRASIL, uma competição anual baseada na concepção e fabricação de um veículo *off-road* por parte dos estudantes de graduação. Com essa finalidade, a iniciativa busca ambientar o aluno à rotina completa de desenvolvimento de um projeto, com foco no desenvolvimento conceitual e nos processos de engenharia contidos na concepção desse modelo.

1.3. Justificativa

A Eletrônica Embarcada é uma área que está atualmente em rápida expansão, especialmente por ter o benefício de ser resistente a vibrações mecânicas, a condições climáticas adversas, a sobrecargas e interferências eletromagnéticas. É um sistema versátil, pois pode ser instalado em carros, navios, aviões, satélites, entre outros. Possui vasta utilização na comercialização veicular, podendo ser empregado para atuar em sistemas de freio ABS, câmbios

automatizados, injeção eletrônica, *air bags*, sistemas de áudio, GPS e uma infinidade de outras aplicações.

É indiscutivelmente uma aposta do mercado automotivo, devido a seu forte apelo comercial ao prover aos passageiros melhores condições de segurança, conforto a bordo, mobilidade e entretenimento. Como consequência disso, ocorre hoje rápido avanço da tecnologia nessa área da engenharia eletrônica.

A implementação desse tipo de tecnologia também é de fundamental importância para o veículo Baja do IME. Por meio dela, é possível a aferição de alguns relevantes parâmetros, que esboçam de maneira precisa o desempenho do carro. É extremamente benéfico que os dados obtidos sejam de conhecimento não apenas do piloto, mas também de toda a equipe, a qual pode traçar a melhor estratégia na competição e fazer ajustes para melhorar o desempenho do carro.

Em adição aos aspectos já levantados, pode-se citar como estímulo o fator interdisciplinar inerente aos projetos da competição Baja SAE. Nela, alunos de Engenharia Mecânica, Eletrônica, Elétrica e da Computação atuam conjuntamente para desenvolver o seu Baja em todas essas áreas. Essa é uma experiência muito valiosa para o aluno de graduação, já que proporciona uma grande troca de conhecimentos.

1.4. Metodologia

Este trabalho consiste no estudo e na implementação da Eletrônica Embarcada na viatura Baja do Exército Brasileiro com posterior aplicação do protocolo CAN Bus. Possui como base a busca de informações em referências sobre a Eletrônica Embarcada e o protocolo de comunicação em foco, além dos *datasheets* de todos os componentes eletrônicos são utilizados. Além disso, a pesquisa é feita utilizando material bibliográfico encontrado em diversos locais, tais como artigos e revistas especializadas.

Inicialmente, foi feita uma análise global do projeto em reuniões com alunos e professores envolvidos na concepção da viatura Baja, para que pudesse ser feito um esboço do trabalho em questão. A partir das informações levantadas, são escolhidos alguns parâmetros de interesse para que o estudo e a implementação da programação aplicada a microcontroladores seja realizada.

Em seguida é visto o que são sensores e como eles funcionam em conjunto com a eletrônica embarcada para controlar o sistema de interesse. Dessa forma, foram adquiridos: uma placa de desenvolvimento, sensores de temperatura, sensores de boia de combustível, sensores para distâncias (*reed switch*), PICs 18F2580 e *transceivers* MCP 2551. Com o desenvolvimento da programação e da aquisição dos materiais necessários, é possível montar o *hardware* do projeto bem como a realização dos testes para que assim o projeto pudesse tomar forma.

A partir de então, começou-se um estudo do funcionamento do protocolo de comunicação CAN-BUS, como ele envia e recebe suas mensagens e o *hardware* necessário para implementá-lo. Fez-se assim a aquisição de todos os componentes necessários para realizar a aplicação.

Para fazer o controle do sistema embarcado faz-se uso de uma Placa de Desenvolvimento, e de PICs com controladores CAN integrados, além de *Transceivers*. Neste projeto, a escolha é pelo trabalho com o Microcontrolador PIC 18F2580 devido a maior facilidade de estabelecer uma aplicação utilizando o protocolo CAN-BUS e com o *Transceiver* MCP2551.

Por fim, o *hardware* e o *firmware* global do projeto, finalizando assim o Projeto de Fim de Curso.

1.5. Estrutura da Monografia

O trabalho está estruturado em 5 capítulos. O capítulo 1 apresenta uma introdução sobre o Projeto de Fim de Curso, contextualizando e dando as justificativas para o projeto. O Capítulo 2 apresenta uma revisão dos conceitos teóricos utilizados no trabalho, além de uma síntese dos novos assuntos estudados. No terceiro capítulo, o projeto é detalhado em toda sua parte, tal como *hardware*, *software* e os resultados do projeto. No quarto, o cronograma do projeto é apresentado. No quinto e último capítulo, antes da conclusão, são analisados alguns desafios do projeto.

2 REVISÃO BIBLIOGRÁFICA

2.1. Placa de Desenvolvimento

A Placa de Desenvolvimento utilizada (mikromedia for dsPIC33EP®) é um sistema que fornece uma alta qualidade como plataforma de desenvolvimento. Ela é compacta, com muitos periféricos embutidos que permitem o desenvolvimento de dispositivos com conteúdo multimídia, possui suporte para compiladores de dsPIC30/33 e PIC24, tais como: mikroC, mikroBasic e mikroPascal. A parte central do sistema é um micro controlador dsPIC33EP512MU810 de 16 bits.

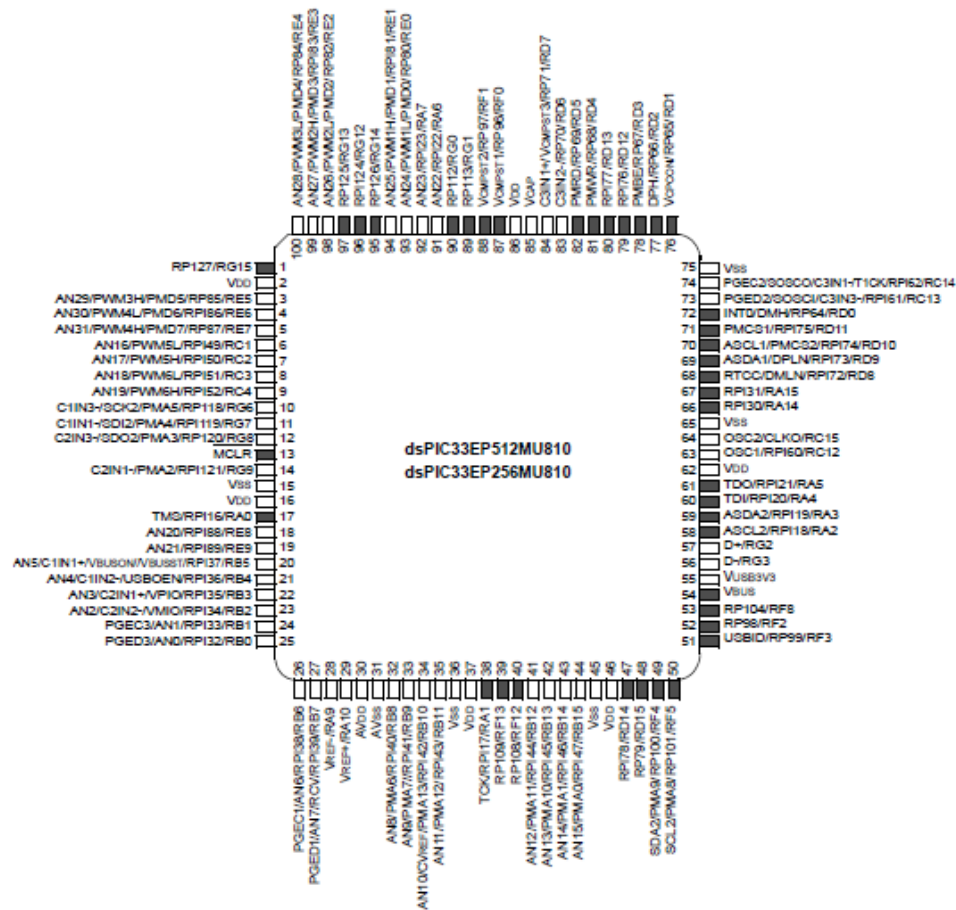


FIG 2.1.1 - Pinagem do microcontrolador dsPIC33EP512MU810

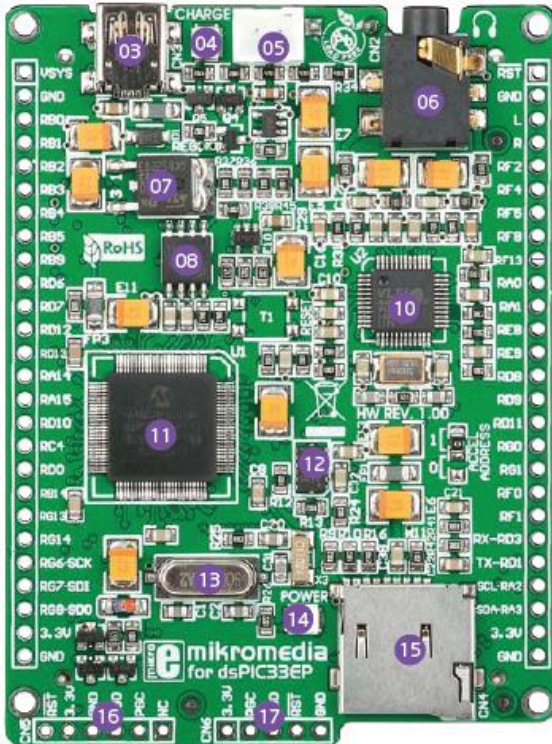
A Placa possui módulos de recursos integrados, como aparelho de som Codec MP3, tela de toque TFT 320x240, acelerômetro, conector USB, conector de áudio e espaço para cartão de memória. É possível salvar imagens, sons e outros arquivos de mídia no cartão de memória micros. Além disso, possui memória Flash de 8 Mbits, dentre outros recursos.

Ela vem pré-programada com bootloader USB HID, mas pode também ser programada com os programadores externos, tal como mikroProg™ ou ICD2 / 3. A placa é compacta e fina, e se encaixa perfeitamente na palma da mão, o que a torna uma plataforma conveniente para dispositivos móveis e para a utilização como Display do piloto da viatura Baja.



FIG 2.1.2 - Bootloader para programação do dsPIC

- 1) Pads de conexão
- 2) Display TFT de 320x240
- 3) Conector MINI-B USB
- 4) LED de indicação de CARGA
- 5) Conector de bateria
- 6) Conector de fone de ouvido
- 7) Regulador da fonte de alimentação
- 8) Memória Flash Serial



- 1) Botão RESET
- 2) Estéreo mp3 codificador / decodificador
- 3) Micro controlador dsPIC33EP512MU810
- 4) Acelerômetro
- 5) Cristal Oscilador
- 6) LED indicador de energia
- 7) Slot para cartão micro SD
- 8) Conector ICD2/3
- 9) Conector mikroProg

FIG 2.1.3 – Característica Principais da Placa de Desenvolvimento

2.2. Conversão Analógico-digital

O Conversor Analógico-digital é um dispositivo eletrônico que gera uma representação digital a partir de uma grandeza analógica. São muito úteis na interface entre dispositivos analógicos e digitais e tem variadas aplicações, como digitalização de áudio e vídeo, leitura de grandezas analógicas por meio de sensores, etc.

A ADC é realizada nas seguintes etapas:

- Amostragem: Fase em que o sinal analógico será retido em um valor fixo por um período de tempo;
- Quantização: Os pulsos do sinal serão arredondados para níveis de quantização;
- Codificação: Cada nível receberá um código binário.

O processo de Conversão Analógico-digital e suas etapas são mostrados na FIG 2.2.1.

No presente projeto, os parâmetros da temperatura e do nível de combustível são captados por sensores e sua leitura gera sinais analógicos. O sinal recebido, depois de digitalizado, é processado. Para o processamento desses sinais serão utilizados circuitos digitais e, por essa razão, a conversão A/D é de fundamental importância.

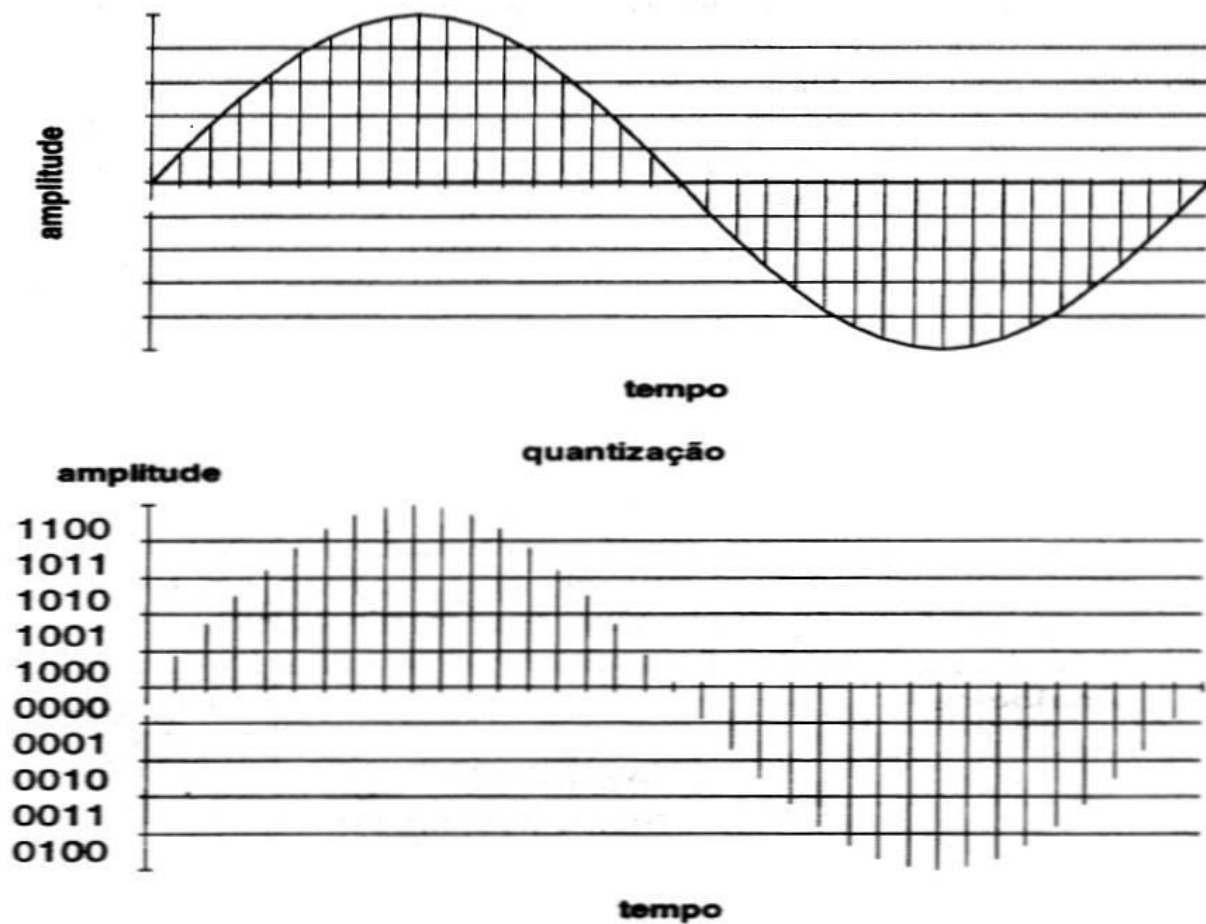


FIG 2.2.1 - Etapas da conversão analógico-digital

2.3. Interrupções

As interrupções são necessárias pois nem sempre é possível que o sistema reaja rapidamente a eventos externos. A maioria dos microcontroladores possuem pinos de interrupção (IRQ - *Interrupt Request*). Quando um destes pinos é ativado o microcontrolador para a sequência de execução apontada pelo PC e salta para uma rotina de tratamento da interrupção. Para a maioria dos microcontroladores é possível desabilitar interrupções. A maioria dos microcontroladores dispõe de interrupções não mascaráveis (não podem ser desabilitadas).

2.4. Parâmetros

Sensores são equipamentos pelos quais o microcontrolador é capaz de monitorar determinado processo. São transdutores, pois realizam a conversão de grandezas físicas em grandezas elétricas. Existem diversos tipos de sensores, destinados a mensurar os mais diversos tipos de parâmetros. Alguns deles são os sensores de posição, velocidade, presença e carga.

Os sensores de posição reportam a posição física de um objeto com respeito a um ponto de referência. Podem se utilizar de informação linear ou angular. Dentre estes, existem alguns subtipos, como os *encoders* e os potenciômetros, que podem ser de uma volta ou multivoltas. No presente projeto, utilizaremos sensor de posição do tipo potenciômetro para o cálculo do nível de combustível.

Já os sensores de proximidade informam ao microcontrolador se uma parte móvel está ou não em uma determinada posição. Podem ser óticos, de efeito Hall, etc. Será utilizado para a aferição da velocidade o sensor *reed switch*, que consiste na produção de tensão decorrente da presença de campo magnético. Já para a temperatura, será utilizado o sensor LM35DZ. [2]

Foram selecionados para serem aferidos três parâmetros, considerados os mais importantes tanto para o piloto como para o restante da equipe do projeto. São eles: velocidade instantânea, temperatura do óleo e nível de combustível.

2.4.1. Velocidade Instantânea

A informação da velocidade do veículo é fundamental para o motorista, especialmente se este tem por objetivo uma competição, já que reflete com precisão a performance obtida pelo Baja. Por meio desse parâmetro, é possível traçar uma melhor estratégia para se percorrer a pista, sabendo-se a máxima velocidade que pode ser imposta em cada ponto do percurso.

No presente projeto, utilizamos o sensor *reed switch* para a aferição da velocidade instantânea.

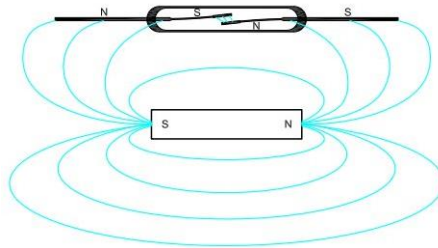


FIG 2.4.1.1 – Sensor reed switch

O sensor *reed switch* consiste em um par de contatos metálicos que se encontram abertos sob condições normais. Ao estarem sob o efeito de um campo magnético, os contatos se unem e, com isso, é fechado esse ramo do circuito. O pulso gerado nesse processo pode ser lido com a utilização de um microcontrolador.

No Baja, pode-se instalar um ímã na roda do carro e o sensor em seu eixo. Dessa forma, a cada giro completo, um pulso é gerado e contabilizado pelo microcontrolador presente na placa de desenvolvimento utilizada.

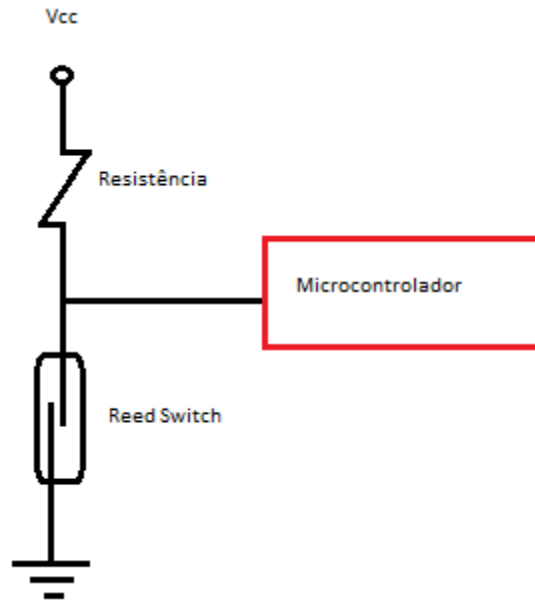


FIG 2.4.1.2 – Resistência de pull-up no sensor

O sensor é montado com uma resistência de *pull-up* (FIG 2.4.1.2). Cada vez que circuito se encontrar sob o efeito do campo magnético do ímã, o pino ligado a ele receberá o valor zero de voltagem e, por meio de uma interrupção que será chamada, esse pulso será contabilizado. O número de pulsos contabilizados no intervalo de tempo de um segundo multiplicado pelo comprimento da roda da viatura nos fornece, com precisão suficiente, a velocidade instantânea em metros por segundo (m/s).

2.4.2. Temperatura do Motor

A temperatura é um parâmetro muito importante para que o motor forneça toda a sua potência. Temperaturas elevadas podem causar a pré-ignição e detonação gerando falhas no funcionamento do motor e diminuição de sua vida útil. Valores abaixo do ideal também causam falhas no motor, como seu funcionamento irregular, aumento do consumo de combustível e desgaste excessivo das peças.

2.4.3. Nível de Combustível

A necessidade de se ter a informação da quantidade de combustível é bem clara. Com ela, pode-se definir o melhor momento para um *pit-stop* e não correr o risco de não completar o enduro, uma das provas da competição Baja SAE que tem duração de 4 horas.

No presente projeto, é utilizada uma boia de combustível. Essa boia funciona como um potenciômetro, que varia a resistência de acordo com o nível de líquido no tanque. Assim como o termopar, a saída nos condutores da boia é um sinal analógico de tensão e, por isso, também será processado por meio de um conversor analógico-digital.

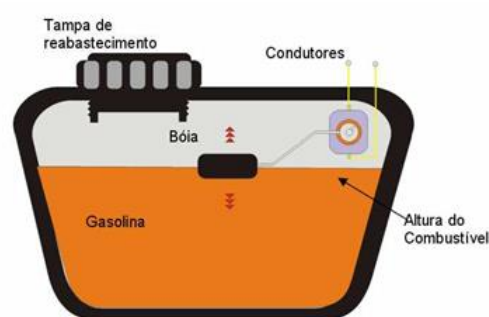


FIG 2.4.3.1 – Localização do sensor

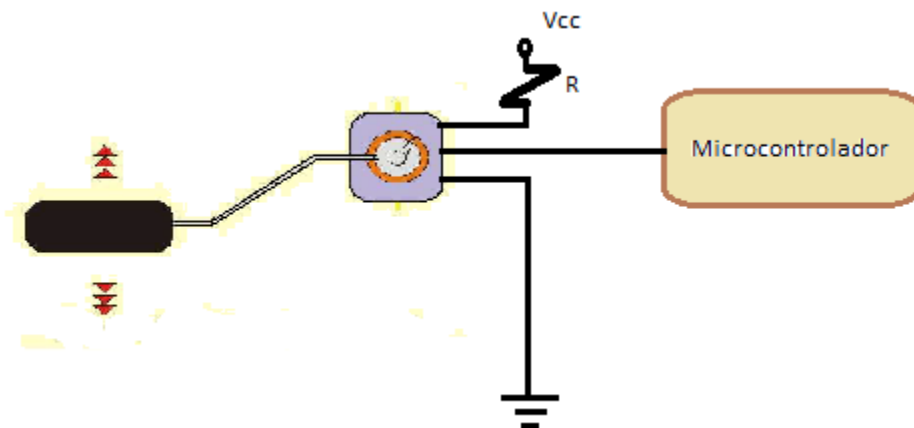


FIG 2.4.3.2 – Sensor de boia de combustível

2.5. Barramento CAN

O CAN (*Controller Area Network*) é um protocolo muito robusto que tem detecção de erros com sinalização e auto verificação. A camada física do barramento CAN pode ser constituída de algumas maneiras diferentes. Neste trabalho é adotado o barramento padrão (FIG 2.5.1), no qual consiste em um par de fios trançados ligados por duas resistências de 120 ohms (terminadores), representando um sistema embutido de comunicação serial de dois fios. Aos dois fios do barramento, denominados de CAN_H e CAN_L, são pendurados os nós do circuito, os quais são detalhados mais a frente. Os dados CAN defeituosos e os quadros remotos são retransmitidas automaticamente, semelhante à Ethernet.

Esse tipo de barramento tem como vantagens apresentar pouca complexidade com fios, ser de fácil manutenção, novos nós podem ser adicionados e removidos facilmente, nos quais a falha de um deles não danifica nem interrompe o sistema e, além disso, possui pouca interferência eletromagnética devido ao par de fios trançados, o qual possui como característica de funcionamento a tensão diferencial. A interferência eletromagnética, quando ocorre, atua nos dois fios, mas o efeito é eliminado tendo em vista que as tensões de efeito em cada fio se anula na subtração do cálculo da tensão diferencial. Apesar disso, apresenta algumas desvantagens como queda do sistema caso haja alguma falha no barramento principal (CAN_H ou CAN_L) e apresenta baixa taxa de dados em algumas aplicações.

O CAN, desenvolvido por Robert Bosch em 1980, é um dos barramentos mais utilizados na tecnologia veicular e é baseado no mecanismo de CSMA/CR (*Carrier Sense Multiple Access/Collision Resolution*) para evitar colisões de quadros durante as transmissões de dados no barramento. Cada nó pode monitorar o barramento e quando há detecção de que ele está ocioso, inicia-se a transmissão de dados. Quando outros nós, tentam enviar dados pelo barramento ao mesmo tempo, a arbitragem acontece e o nó com a maior prioridade (identificador de mensagem menor) ganha a arbitragem e envia seus próprios dados. Esse barramento tem um simples mecanismo de detecção de erros e recuperação. Os nós verificam a integridade das mensagens, olhando para os campos CRC. Se for detectado um erro, os outros nós do barramento são informados sinalizando mensagens de erro. Além disso, utiliza a tecnologia de multi-mestre, na qual todos os nós podem enviar e receber informações pelo barramento, diferentemente da

tecnologia mestre-escravo, na qual um dos nós (mestre) controla o tráfego de informações para os outros nós (escravos). [3] [4]

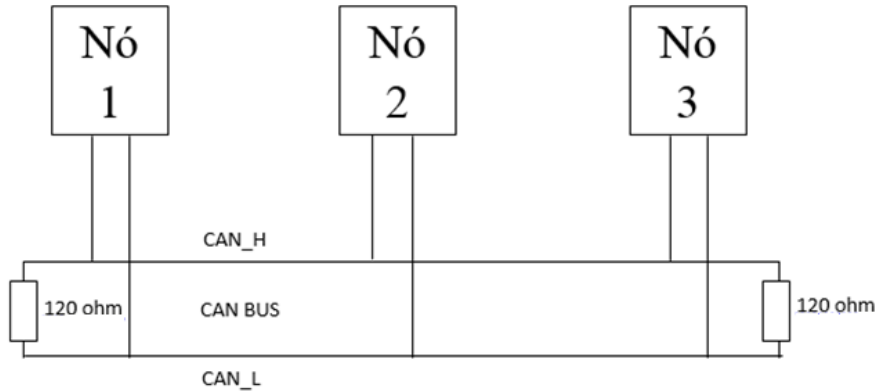


FIG 2.5.1 – Terminação de barramento padrão

A Taxa de transferência de dados depende da distância. Por exemplo, 1 Mbit/s, pode ser obtida em comprimentos de rede de 40m enquanto 250 Kbit/s pode ser conseguida em comprimentos de rede de 250 m. Quanto maior for a distância, menor a taxa máxima que pode ser alcançada (FIG 2.5.2).

Velocidade de transmissão do barramento CAN

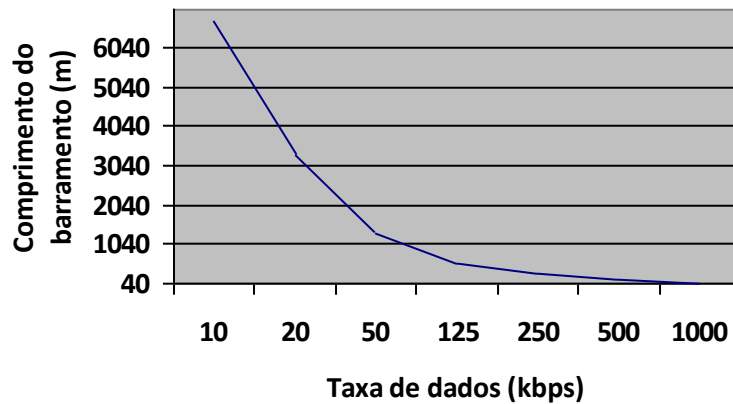


FIG 2.5.2 – Velocidade de transmissão do barramento CAN

O dado no barramento CAN é diferencial e ele especifica dois estados lógicos: dominantes e recessivos. A FIG 2.5.3 mostra o estado de sinais no barramento.

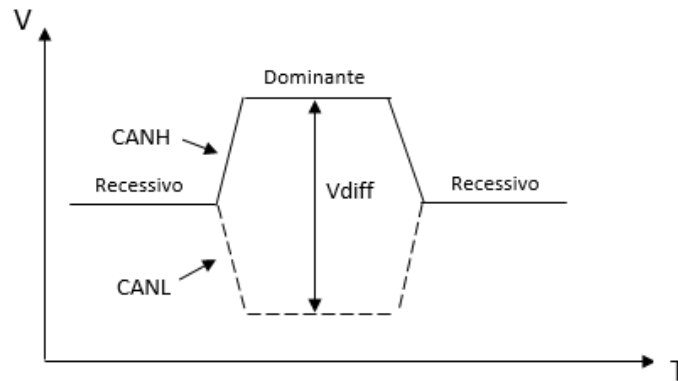


FIG 2.5.3 – Nível do sinal no barramento CAN

O estado recessivo é a lógica "1" e neste estado a tensão diferencial no barramento ($V_{diff} = CAN_H - CAN_L$) é idealmente 0 V (idealmente $CAN_H = CAN_L = 2,5$ V). Na prática, a tensão de saída diferencial recessiva é inferior a 0,05 V.

O estado dominante é a lógica "0" e neste estado a tensão diferencial no barramento ($V_{diff} = CAN_H - CAN_L$) é idealmente 2 V (idealmente $CAN_H = 3,5$ V e $CAN_L = 1,5$ V). Na prática, a tensão de saída diferencial dominante está entre 1,5 V e 3,0 V.

Quando vários nós do barramento tentam transmitir ao mesmo tempo, uma lógica de arbitragem é usada para conceder acesso ao barramento. Quando há arbitragem no barramento, um bit de estado dominante sempre prevalece sobre um bit de estado recessivo.

A maioria dos controladores CAN também podem ter sua construção baseada em “*Filter Masks*”. Estas máscaras são utilizadas para determinar que os bits do identificador da mensagem devem ser comparados com os valores dos filtros de aceitação. Por exemplo, definindo todos os valores de máscara de filtro para 1s irá certificar-se de que todos os bits dos identificadores de mensagens serão comparados com todos os pedaços de filtros de aceitação.

A Figura 2.5.4 mostra um exemplo onde o Nó SBOIA e Nó STEMP transmitem quadros de dados e o Nó DISPLAY aceita este quadro de dados.

A teoria deste trabalho, em sua maior parte, foi baseada no livro: “*Controller Area Network Projects*” de Dogan Ibrahim. Dessa forma, o leitor deste trabalho pode se aprofundar mais no assunto aqui apresentado utilizando a teoria presente neste livro.

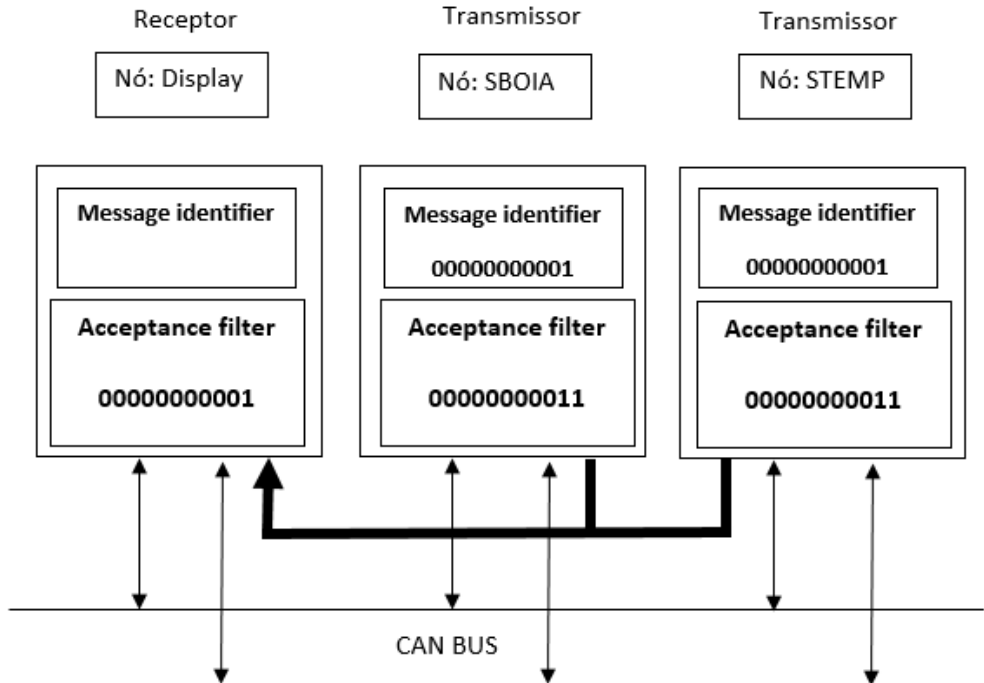


FIG 2.5.4 – Exemplo da passagem e aceitação dos dados entre os nós.

2.6. O Microcontrolador PIC 18F2580

Os microcontroladores são chips contendo um processador, memória e periféricos de entrada e saída, para aplicações em que o custo, energia e o espaço são razoavelmente limitados. Há duas principais diferenças entre os microcontroladores e os microprocessadores: a principal delas é que o primeiro é utilizado em funções específicas, dedicadas, enquanto que o outro é para propósitos mais gerais (como o uso em computadores). Por essa razão, não precisa ser muito rápido nem tampouco ter um conjunto de instruções extenso e poderoso. Além disso, não tem grande capacidade de processamento e seu custo é geralmente mais baixo.

Outra diferença é que os microcontroladores, além dos componentes lógicos e aritméticos usuais de um microprocessador de uso geral, integra outros elementos como memória de leitura e escrita para armazenamento de dados, e dispositivos periféricos como conversores analógico-digitais e interfaces de entrada e saída de dados.

São aplicados nas mais diferentes áreas, como em automação e controle, aparelhos residenciais, telefones celulares e eletrônica embarcada em geral. Devido a essa vasta aplicação, deve-se selecionar, dentre os inúmeros tipos existentes, o microcontrolador mais adequado ao objetivo que se tem definido.



FIG 2.6.1 – O PIC 18F2580

Existem diversos modelos de microcontroladores no mercado e para a aplicação deste trabalho foi escolhido o microcontrolador PIC18F2580 (FIG 2.6.1), pois este apresenta diversas funcionalidades que facilitaram o desenvolvimento do projeto, dentre elas possuir o controlador CAN embutido no chip (FIG 2.6.2).

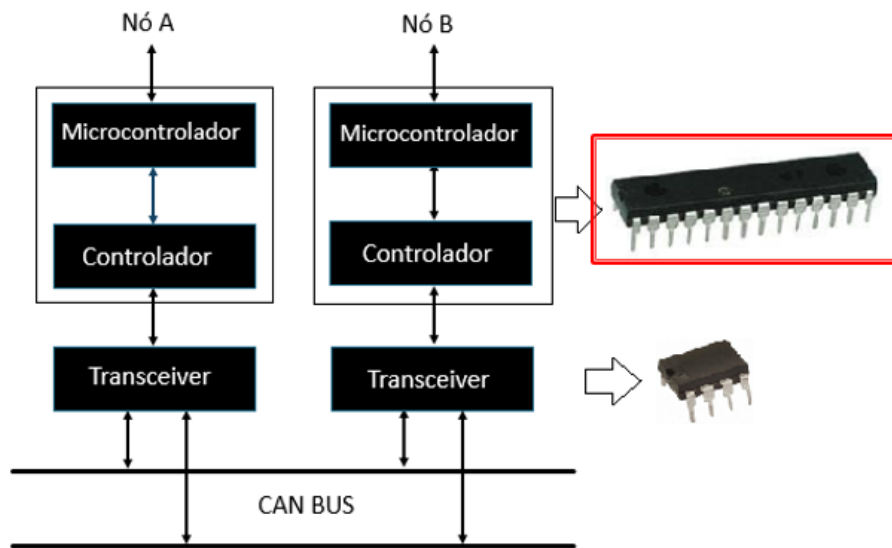


FIG 2.6.2 – O PIC 18F2580 com o controlador CAN embutido

O PIC 18F2580 é um microcontrolador de 28 pinos, que opera sobre uma tensão de 5V. Seu principal diferencial é o módulo de comunicação CAN e suas duas portas dedicadas para esta comunicação. Ele oferece as vantagens de todos os microcontroladores da família PIC18 - ou seja, alto desempenho computacional a um preço econômico e com a adição de alta

resistência. Além disso, apresenta melhorias para o projeto que tornam estes microcontroladores uma escolha lógica de alto desempenho e potenciais aplicações.

A FIG 2.6.3 representa um esquemático deste microcontrolador onde está exibida toda a sua pinagem, nas quais, em destaque, foram as utilizadas no trabalho.

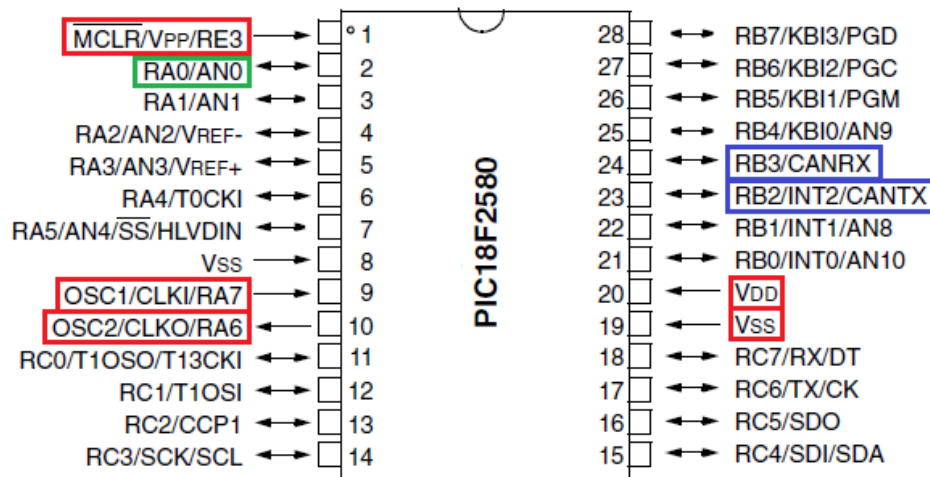


FIG 2.6.3 – Esquemático do PIC 18F2580

2.7. O Transceiver MCP 2551

Como os PICs trabalham na faixa de 0 a 5 V (Padrão TTL) e o CAN opera na faixa de 1,5 V a 3,5 V, é necessário um componente para converter os níveis lógicos e possibilitar o acoplamento dos circuitos. O componente que assume essa função é nomeado de *transceiver*. No trabalho realizado foi utilizado o *transceiver* MCP 2551 (FIG 2.7.1). [5]

O MCP2551 é um dispositivo CAN de altas velocidades, tolerante a falhas e que serve como a interface entre um controlador de protocolo CAN e o barramento físico. O MCP2551 fornece capacidade de transmissão e recepção diferencial para o controlador de protocolo CAN, e é totalmente compatível com a norma ISO-11898, incluindo 24V de requisitos. Ele opera em velocidades de até 1 Mb/s. Tipicamente, cada nó num sistema CAN deve ter um dispositivo para converter os sinais digitais gerados por um Controlador CAN para sinais adequados para transmissão dos cabos do barramento (diferencial de saída). Ele também fornece um tampão

entre o controlador CAN e os picos de alta tensão que podem ser gerados no barramento CAN por fontes externas (EMI, ESD, transientes elétricos, etc.).

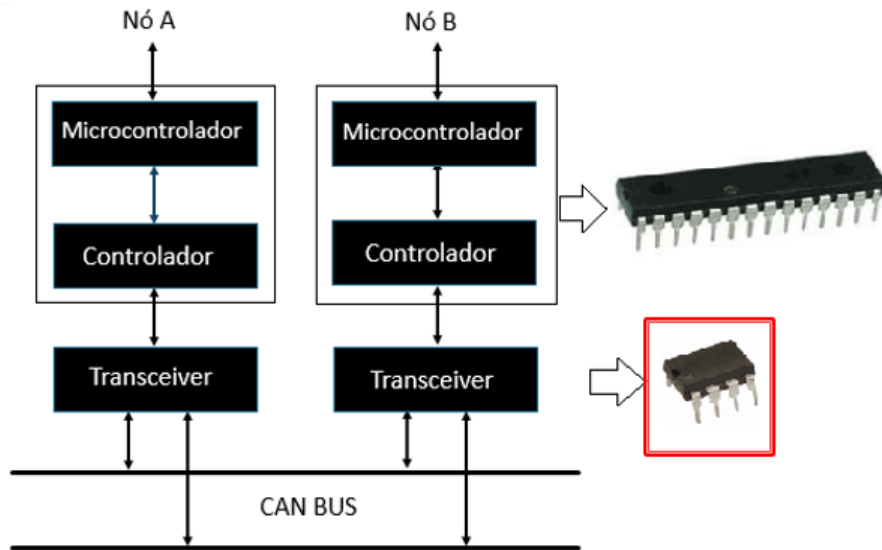


FIG 2.7.1 – O *transceiver* fazendo a interface entre o barramento e o controlador CAN

Além de regular as tensões entre os circuitos, o *transceiver* CAN também é responsável por estabilizar a forma de onda dos sinais. Essa função é controlada por um resistor em seu terminal RS (pino 8) como exibido na FIG 2.7.2, na qual representa um esquemático deste *transceiver* onde está exibida toda a sua pinagem, nas quais, em destaque, foram as utilizadas no trabalho.

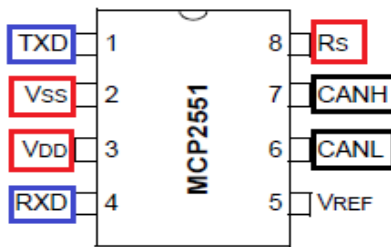


FIG 2.7.2 – Esquemático do MCP 2551

3 PROJETO

3.1. Hardware

O circuito final do projeto utiliza, além do microcontrolador dsPIC33EP512MU810, display TFT e outros recursos presentes na Placa de Desenvolvimento anteriormente mencionada, três PICs 18F2580 com três *transceivers* MCP 2551 para o barramento CAN, um display LCD, dois sensores de temperatura LM35, dois sensores de boia de combustível, e um reed switch.

O circuito do projeto foi dividido em duas partes. A primeira consiste em três PICs 18F2580 que recebem e processam o sinal do sensor de temperatura com o LCD. A segunda consiste na Placa de Desenvolvimento ligada aos sensores LM35, boia de combustível e reed switch.

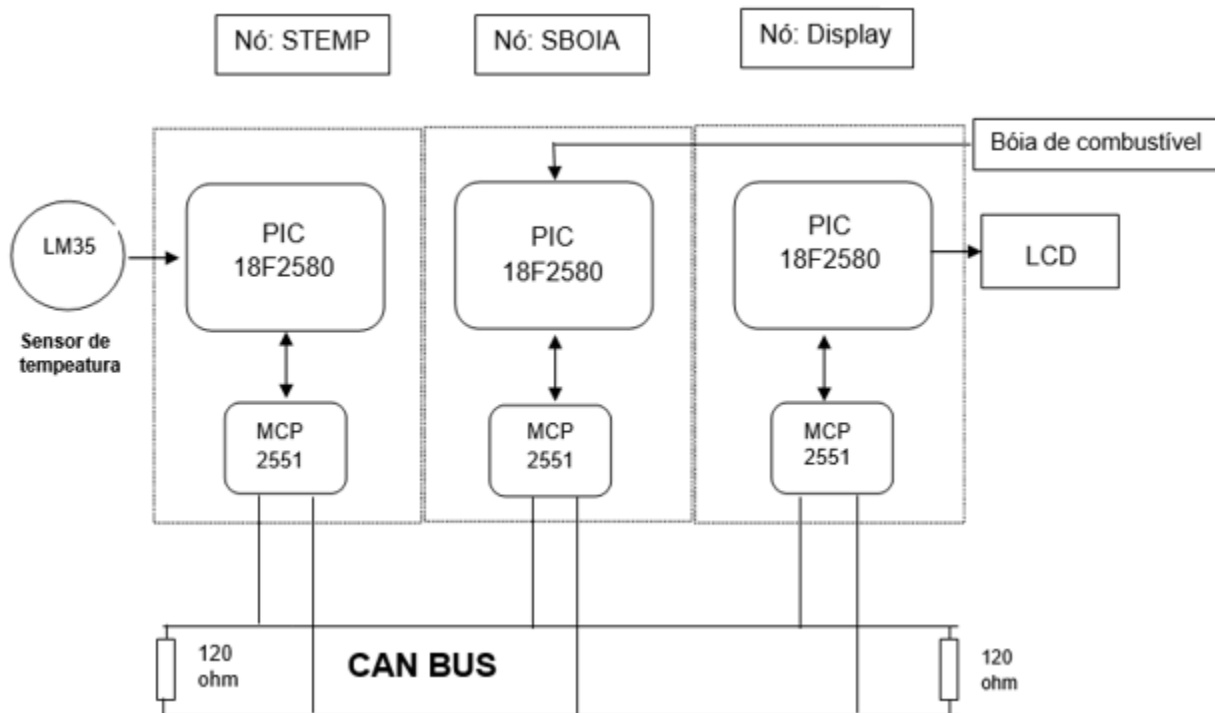


FIG 3.1.1 – Diagrama de blocos do projeto utilizando barramento CAN

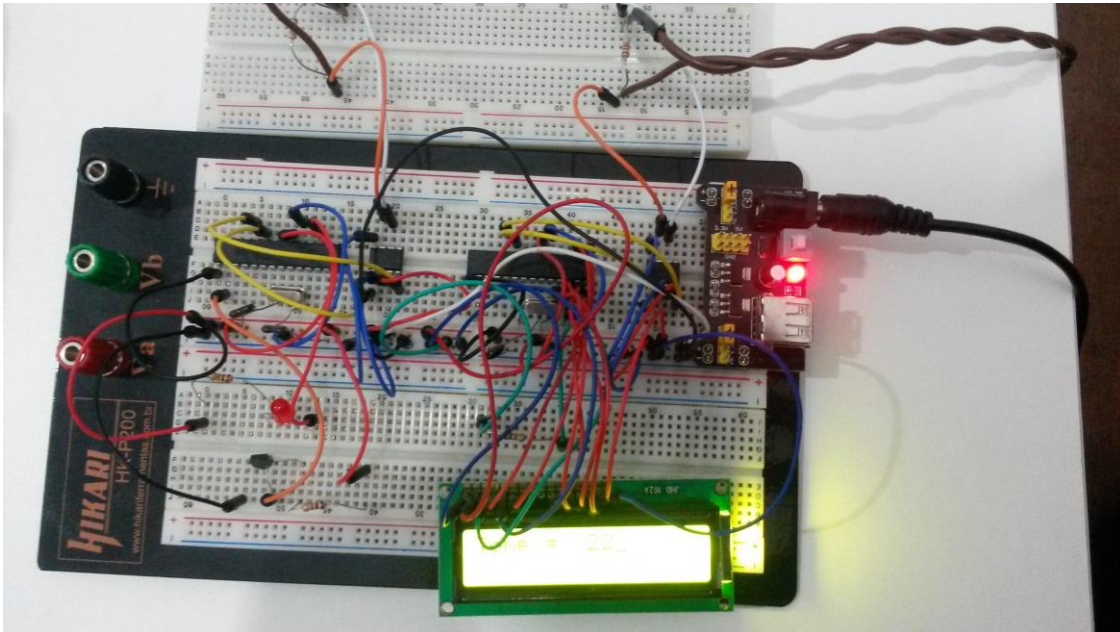


FIG 3.1.2 – Foto de parte implementação do barramento CAN

3.2. Software

Para o desenvolvimento das atividades do projeto, são utilizados dois softwares, ambos da Mikroelektronika®, para a implementação dos códigos utilizados para programar tanto a Placa de Desenvolvimento quanto os PICs utilizados no barramento CAN. Esses softwares são projetados para fornecer ao programador a solução mais fácil possível para o desenvolvimento de aplicações para sistemas embarcados, sem comprometer o desempenho ou o controle.

Na programação aplicada à Placa de Desenvolvimento foi utilizado o “mikroC PRO for dsPIC30/33 e PIC24”, uma poderosa ferramenta de desenvolvimento rico em recursos para microcontroladores dsPIC30/33 e PIC24.

Quanto à programação aplicada ao PIC 18F2580, utilizado para implementação da rede CAN, é utilizado o software “mikroC PRO for PIC”, uma ferramenta de desenvolvimento rico em recursos para microcontroladores PIC da Microchip®.

4 CRONOGRAMA

Conforme estabelecido, este trabalho cumpriu as metas relativas a cada etapa proposta pelo professor orientador do Projeto, de acordo com o seguinte esquema:

- Tarefa 1: Reunião com alunos e professores (Equipe Víbora) envolvidos na concepção da viatura Baja do exército Brasileiro;
- Tarefa 2: Levantamento das informações relevantes para o projeto;
- Tarefa 3: Escolha dos parâmetros a serem estudados;
- Tarefa 4: Estudo sobre a Eletrônica Embarcada;
- Tarefa 5: Aquisição dos materiais;
- Tarefa 6: Programação aplicada a microcontroladores;
- Tarefa 7: Testes e reformulação do Projeto;
- Tarefa 8: Estudo do protocolo CAN Bus;
- Tarefa 9: Implementação do protocolo CAN Bus no projeto;
- Tarefa 10: Testes e melhorias;
- Tarefa 11: Documentação e publicação dos resultados obtidos;
- Tarefa 12: Conclusão.

Tarefa\Mês	2013					2014					
	08	09	10	11	12	01	02	03	04	05	06
1	X	X									
2		X	X								
3				X							
4			X	X	X	X	X				
5					X	X			X	X	
6						X	X	X	X	X	
7								X			
8									X	X	
9										X	
10										X	
11										X	
12										X	

5 CONCLUSÃO

Apesar dos desafios encontrados e superados ao longo do projeto, os autores acreditam que ainda há muito mais desafios a serem ultrapassados na continuidade desse projeto. Alguns problemas surgiram na realização desse trabalho, mas nada que pudesse interferir na confecção dos mesmos, pois os alunos, o professor orientador e outros professores ajudaram a resolver esses obstáculos. Dessa forma, acredita-se que o projeto foi realizado dentro do esperado.

O primeiro grande problema enfrentado foi a dificuldade de obtenção de diversos dos materiais, que tiveram que ser comprados com recursos dos próprios alunos. Além disso, o alto custo dos ambientes de programação (MikroC for PIC e MikroC for dsPIC da Mikroelektronika®) foi um entrave adicional à implementação do trabalho.

Outra dificuldade foi a escolha de sensores que, do ponto de vista da eletrônica, atendessem aos objetivos do projeto e, simultaneamente, fossem compatíveis com aspectos de ordem prática que o Baja impunha ao projeto. Um exemplo disso foi a boia de combustível, já que não foi encontrado nenhum modelo comercial que coubesse no pequeno tanque de combustível da viatura.

Outro problema foi a programação destes componentes. Os programadores mais econômicos não são capazes de programar PICs da família 18F. Além disso, dificuldades foram encontradas na interpretação do *datasheet* da placa de desenvolvimento, a qual não deixava clara certas informações de importância para a execução do projeto, como a pinagem de comunicação CAN utilizada na Placa.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] O que é o Baja? (Projeto Baja SAE, a Competição Baja SAE e Histórico). Disponível em: <http://www.saebrasil.org.br/eventos/programas_estudantis/baja2014/oquee.aspx>.
- [2] T. V. Gonçalves, I. B. Queiroz, J. M. Ramos, L. G. Martins e A. M. Nogueira, “Projeto Mini Baja Como Estudo de Caso Para Instrumentação Eletrônica” em Anais do XXXIV COBENGE. Passo Fundo: Ed. Universidade de Passo Fundo, Setembro de 2006.
- [3] D. IBRAHIM, “*Controller Area Network Projects*”, Londres, Reino Unido: Elektor International, 2011.
- [4] B. N. Dantas, J. C. De Matos e V. L. Horta, “*Estudo de Dispositivos Utilizados Para Automação Residencial – Domótica*”, Rio de Janeiro – RJ: Instituto Militar de Engenharia, 2013. Iniciação a pesquisa.
- [5] G. Ângelo L. e V. Leal V., “*Estudo da Eletrônica Embarcada em Módulos de Controle em Veículos Automotivos*”, Rio de Janeiro – RJ: Instituto Militar de Engenharia, 2013. Iniciação a pesquisa.
- [6] Revista CESVI –Julho/Agosto2012 –Ano 15 nº 80.
- [7] Revista Automotive Business –Novembro 2010 –Ano 2 –Número 6.
- [8] SENAI São Paulo – Serviço Nacional de Aprendizagem Industrial. Disponível em: <<http://www.sp.senai.br> > Acessado em 17/10/2013.

7 APÊNDICES

7.1 Programa de implementação na Placa de Desenvolvimento

O seguinte código foi utilizado para implementação da comunicação dos sensores com a Placa de Desenvolvimento:

```
const code char far MainPicture[60586];
unsigned short TFT_DataPort at LATE;           // DATA port
unsigned short TFT_DataPort_Direction at TRISE; // DATA port

sbit TFT_RST at LATC1_bit;
sbit TFT_RS at LATB15_bit;
sbit TFT_CS at LATF12_bit;
sbit TFT_RD at LATD5_bit;
sbit TFT_WR at LATD4_bit;

sbit TFT_RS_Direction at TRISB15_bit;
sbit TFT_RST_Direction at TRISC1_bit;
sbit TFT_CS_Direction at TRISF12_bit;
sbit TFT_RD_Direction at TRISD5_bit;
sbit TFT_WR_Direction at TRISD4_bit;

unsigned int adcRes1;
unsigned int adcRes2;
int cont_vel;
int cont_rpm;
char txt1[6];
char txt2[6];
char out1[16];

//void INTx_IO_Init(void);

void get_pin(void)
{
    Delay_ms(4);
    if(PORTD.F1==0) {
        cont_vel++;
        IntToStr(cont_vel, out1);
        TFT_Write_Text(out1, 120, 40);
        Delay_ms(4); // fazer debounce
    }
}
```

```

    }
}

void Display_temp(void) {
    adcRes1 = ADC1_Get_Sample(1);
    WordToStr(adcRes1, txt1);
    TFT_Write_Text(txt1, 120, 80);
    TFT_Write_Text("Celsius", 200, 80);
}

void Display_comb(void) {
    adcRes2 = ADC2_Get_Sample(2);
    WordToStr(adcRes2, txt2);
    TFT_Write_Text(txt2, 120, 100);
    TFT_Write_Text("Litros", 200, 100);
}

void main() {
    cont_vel=0;
    cont_rpm=10;
    adcRes1=0;
    adcRes2=0;
    // Setting output frequency to 140MHz
    PLLFBD = 70;          // PLL multiplier M=70
    CLKDIV = 0x0000;     // PLL prescaler N1=2, PLL postscaler N2=2
    Delay_ms(150);

    // INTx_IO_Init();      /* Call function to initialize the External Interrupts */

    PORTB = 0x0000;
    TRISB.F1 = 1;        // set pin as input - needed for ADC to work
    TRISB.F2 = 1;
    ADC1_Init();
    ADC2_Init();

    //PORTD = 0x0000;
    TRISD.F1 = 1;        // setar pino como input

    TFT_Set_Default_Mode();
    TFT_Init_ILI9340_8bit(320, 240);

    while(1) {
        TFT_Fill_Screen(CL_WHITE);
        TFT_Set_Font(TFT_defaultFont, CL_RED, FO_HORIZONTAL);
        TFT_Write_Text("EQUIPE VIBORA", 10, 0);
    }
}

```

```

TFT_Set_Font(TFT_defaultFont, CL_BLACK, FO_HORIZONTAL);
TFT_Write_Text("VELOCIMETRO:", 10, 40);
get_pin();

TFT_Write_Text("m/s", 200, 40);

TFT_Write_Text("RPM:", 10, 60);
TFT_Write_Text("rpm", 200, 60);
TFT_Write_Text("TEMPERATURA:", 10, 80);
Display_temp();

TFT_Write_Text("COMBUSTIVEL:", 10, 100);
Display_comb();
Delay_ms(800);
    }
}

```

7.2 Programa de implementação no barramento CAN

Os códigos que descrevem a comunicação do sensor de temperatura com o LCD via barramento CAN estão apresentados nos subitens a seguir.

7.2.1 Display_LCD.c

```
// Conexões módulo LCD
```

```

sbit LCD_RS at RC4_bit;
sbit LCD_EN at RC5_bit;
sbit LCD_D4 at RC0_bit;
sbit LCD_D5 at RC1_bit;
sbit LCD_D6 at RC2_bit;
sbit LCD_D7 at RC3_bit;

```

```
// Pinagem LCD
```

```
sbit LCD_RS_Direction at TRISC4_bit;
```

```

sbit LCD_EN_Direction at TRISC5_bit;
sbit LCD_D4_Direction at TRISC0_bit;
sbit LCD_D5_Direction at TRISC1_bit;
sbit LCD_D6_Direction at TRISC2_bit;
sbit LCD_D7_Direction at TRISC3_bit;

void main()
{
    unsigned char temperature, rdata[8];
    unsigned short init_flag, len, rd_flag, read_flag;
    char SJW, BRP, Phase_Seg1, Phase_Seg2, Prop_Seg, txt[4];
    long id, mask, STEMP_ID, SBOIA_ID, LCD_ID;

    TRISC = 0;    // PORT C saída (LCD)
    TRISB = 0x08; // RB2 saída, RB3 entrada

    SBOIA_ID = 1;
    LCD_ID = 3;
    STEMP_ID = 5;

    // Parâmetros do barramento CAN

    SJW = 3;
    BRP = 8;
    Phase_Seg1 = 3;
    Phase_Seg2 = 3;
    Prop_Seg = 1;

    init_flag = _CAN_CONFIG_SAMPLE_THRICE &
                _CAN_CONFIG_PHSEG2_PRG_ON &
                _CAN_CONFIG_STD_MSG    &

```

```

        _CAN_CONFIG_DBL_BUFFER_ON &
        _CAN_CONFIG_VALID_STD_MSG &
        _CAN_CONFIG_LINE_FILTER_OFF;

read_flag = 0;

// Inicializa o módulo CAN

CANInitialize(SJW, BRP, Phase_Seg1, Phase_Seg2, Prop_seg, init_flag);

// Modo de Configuração CAN

CANSetOperationMode (_CAN_MODE_CONFIG, 0xFF);

mask = -1;

// Configura para 1 os bits da máscara1

CANSetMask(_CAN_MASK_B1, mask, _CAN_CONFIG_STD_MSG);

// Configura os bits da máscara2

mask = 0;
mask |= 4;
mask ^= -1;

CANSetMask(_CAN_MASK_B2, mask, _CAN_CONFIG_STD_MSG);

// Configura o identificador do filtro B2_F3 para 1

CANSetFilter(_CAN_FILTER_B2_F3, SBOIA_ID, _CAN_CONFIG_STD_MSG);

```

```

// Configura o módulo CAN para o modo NORMAL

CANSetOperationMode(_CAN_MODE_NORMAL, 0xFF);

// Configuração do LCD

Lcd_Init();
Lcd_Cmd(_LCD_CLEAR);          // Apaga LCD
Lcd_Out(1,1,"CAN BUS PFC");   // Texto de inicialização do LCD
Delay_ms(2000);               // Espera por 2 segundos

// Loop de programa. Lê a temperatura do nó que possui o sensor de temperatura e
coloca o valor no LCD

for(;;)    // Loop infinito
{
    rd_flag = CANRead(&id, rdata, &len, &read_flag);
    // Pega a temperatura do sensor
    if (rd_flag != 0 && id == STEMP_ID)
    {
        Lcd_Cmd(_LCD_CLEAR);
        Lcd_Out(1,1,"Temp = ");
        temperature = rdata[0];
        BytetoStr(temperature,txt); // Converte para string
        Lcd_Out(1,8,txt);          // Exibe no LCD
    }
    if (rd_flag != 0 && id == SBOIA_ID)
    {
        Lcd_Out(2,1,"Combust = ");
        temperature = rdata[0];
    }
}

```



```

        BytetoStr(temperature,txt); // Converte para string
        Lcd_Out(2,11,txt);        // Exibe no LCD
    }
}
}

```

7.2.2 Sensor_temperatura.c

```

void main()
{
    unsigned char sdata[8];
    unsigned short init_flag, send_flag;
    char SJW, BRP, Phase_Seg1, Phase_Seg2, Prop_Seg;
    unsigned int temp;
    long STEMP_ID, LCD_ID, SBOIA_ID;
    float temperature;
    long id, mask;

    TRISA = 0xFF; // Porta A entrada
    TRISB = 0x08; // RB2 saída, RB3 entrada
    TRISC = 0;   // Pinagem de saída

    SBOIA_ID = 1;
    LCD_ID = 3;
    STEMP_ID = 5;

    // Configuração Conversor AD

    ADCON1 = 0x80;

    // Parâmetros de tempo do barramento CAN

    SJW = 3;
    BRP = 8;
    Phase_Seg1 = 3;
    Phase_Seg2 = 3;
    Prop_Seg = 1;

    init_flag = _CAN_CONFIG_SAMPLE_THRICE &
                _CAN_CONFIG_PHSEG2_PRG_ON &
                _CAN_CONFIG_STD_MSG &
                _CAN_CONFIG_DBL_BUFFER_ON &

```

```

        _CAN_CONFIG_VALID_STD_MSG &
        _CAN_CONFIG_LINE_FILTER_OFF;

send_flag = _CAN_TX_PRIORITY_0    &
            _CAN_TX_STD_FRAME      &
            _CAN_TX_NO_RTR_FRAME;

// Inicialização do módulo CAN

CANInitialize(SJW, BRP, Phase_Seg1, Phase_Seg2, Prop_seg, init_flag);

// Modo de configuração do CAN

CANSetOperationMode(_CAN_MODE_CONFIG, 0xFF);

mask = -1;

// Configura todos os bits da máscara1 para 1

CANSetMask(_CAN_MASK_B1, mask, _CAN_CONFIG_STD_MSG);

// Configura todos os bits da máscara2 para 1

CANSetMask(_CAN_MASK_B2, mask, _CAN_CONFIG_STD_MSG);

// Configura identificador do filtro B2_F3 para 1

CANSetFilter(_CAN_FILTER_B2_F3, STEMP_ID, _CAN_CONFIG_STD_MSG);

// Configura módulo CAN para modo NORMAL

CANSetOperationMode(_CAN_MODE_NORMAL, 0xFF);

// Loop de programa

PORTC = 0x00; // LED na Porta C

for(;;) // Loop infinito
{
    temp = Adc_Read(0); // Lê a temperatura
    temperature = temp * 0.488; // em graus Celsius
    sdata[0] = (unsigned char)temperature; // Envia a temperatura para o nó do
Display
    CANWrite(STEMP_ID, sdata, 1, send_flag);

    if( temperature > 27 )

```

```

    {
        RC6_bit = 0;        // Acende o LED se a temperatura for maior que 27 graus
    }
    else ( RC6_bit = 1 ) ;    // Apaga o LED caso contrário

    Delay_Ms(1000);        // Espera 1 segundo

    }                        // Término do loop
}                            // Termina o programa

```

7.2.3 Sensor_combustível.c

```

void main()
{
    unsigned char sdata[8];
    unsigned short init_flag, send_flag;
    char SJW, BRP, Phase_Seg1, Phase_Seg2, Prop_Seg;
    unsigned int nivel;
    long STEMP_ID, LCD_ID, SBOIA_ID;
    float combustivel;
    long id, mask;

    TRISA = 0xFF; // PORT A entrada
    TRISB = 0x08; // RB2 saída, RB3 entrada
    TRISC = 0;    // Pinagem de saída

    STEMP_ID = 5;
    SBOIA_ID = 1;
    LCD_ID = 3;

    // Configuração Conversor AD

    ADCON1 = 0x80;

    // Parâmetros de tempo do barramento CAN

    SJW = 3;
    BRP = 8;
    Phase_Seg1 = 3;
    Phase_Seg2 = 3;
    Prop_Seg = 1;

    init_flag = _CAN_CONFIG_SAMPLE_THRICE &
                _CAN_CONFIG_PHSEG2_PRG_ON &

```

```

        _CAN_CONFIG_STD_MSG      &
        _CAN_CONFIG_DBL_BUFFER_ON &
        _CAN_CONFIG_VALID_STD_MSG &
        _CAN_CONFIG_LINE_FILTER_OFF;

send_flag = _CAN_TX_PRIORITY_0      &
            _CAN_TX_STD_FRAME      &
            _CAN_TX_NO_RTR_FRAME;

// Inicialização do módulo CAN

CANInitialize(SJW, BRP, Phase_Seg1, Phase_Seg2, Prop_seg, init_flag);

// Modo de configuração do CAN

CANSetOperationMode(_CAN_MODE_CONFIG, 0xFF);

mask = -1;

// Configura todos os bits da máscara1 para 1

CANSetMask(_CAN_MASK_B1, mask, _CAN_CONFIG_STD_MSG);

// Configura todos os bits da máscara2 para 1

CANSetMask(_CAN_MASK_B2, mask, _CAN_CONFIG_STD_MSG);

// Configura identificador do filtro B2_F3 para 1

CANSetFilter(_CAN_FILTER_B2_F3, SBOIA_ID, _CAN_CONFIG_STD_MSG);

// Configura módulo CAN para modo NORMAL

CANSetOperationMode(_CAN_MODE_NORMAL, 0xFF);

// Loop de programa.

PORTC = 0x00; // LED na Porta C

for(;;) // Loop infinito
{
    nivel = Adc_Read(0); // Lê o nível de combustível
    combustivel = nivel * 0.5; // Conversão da quantidade de combustível
    devido ao recipiente
    sdata[0] = (unsigned char)combustivel; // Envia a quantidade de combustível
    para o nó do Display
}

```

```
CANWrite(SBOIA_ID, sdata, 1, send_flag);

if( combustivel < 40 )
{
    RC6_bit = 1;          // Acende o LED se o nivel de combustivel for menor
que 40 u.v.
}
else ( RC6_bit = 0 ) ;    // Apaga o LED caso contrário

Delay_Ms(1000);          // Espera 1 segundo

}                          // Término do loop
}                          // Termina o programa
```