



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Volante Electrónico para o FST

Projecto e Construção da Electrónica para o Volante do Carro da
Formula Student

VÍTOR MANUEL DE ALMEIDA

Dissertação para obtenção do Grau de Mestre em
ENGENHARIA ELECTRÓNICA

Júri

Presidente: Prof. João Costa Freire
Orientador: Prof. Moisés Piedade
Co-Orientador: Prof. Francisco Alegria
Vogal: Prof. António Serralheiro

Outubro 2009

Agradecimentos

Quero em primeiro lugar agradecer ao Prof. Moisés Piedade por todo o seu apoio, dedicação e empenho durante este trabalho, pois foi imprescindível à sua realização.

Agradeço a todos os meus amigos e colegas da LEE sem exceção, em especial aos do ano 2003 e aos do ano 2004, pelo apoio em todos os momentos e pelas amizades criadas.

Agradeço ao Prof. Francisco Alegria e ao grupo FST, um obrigado também para o Sr. Pina, pela sua presença e disponibilidade.

Um agradecimento muito especial à minha mãe, pelo seu enorme esforço a todos os níveis para que eu pudesse alcançar esta meta, devo-lhe toda esta oportunidade e ainda um apoio constante em tudo. Agradeço também ao meu irmão pela sua ajuda e aconselhamento a todos os níveis.

Agradeço a todos os meus amigos pelo apoio, estímulo e animação que me transmitem pois sabem o quanto é para mim muitíssimo importante poder contar com todos vocês, seja a malta das bikes, do TT, do ténis, do piscinol, das médias, dos ajuntamentos, das nights e de tudo o resto...

Resumo

Esta dissertação surge pela vontade de desenvolver o carro de corrida da Formula Student do Instituto Superior Técnico tornando-o cada vez mais competitivo e evoluído. O volante de um carro de competição hoje em dia é muito mais que uma simples peça para controlar a direcção do carro. É no volante que se encontram informações como a rotação do motor, a mudança engrenada, a velocidade e a temperatura do motor. Estes dados são necessários ao piloto para otimizar a condução e também para se inteirar do estado do carro.

Neste trabalho projectou-se e construiu-se o volante para o carro da FST. O volante comunica com a restante electrónica presente no carro através do barramento CAN. É através deste barramento que recebe todos os dados do carro que são apresentados ao piloto.

O sistema é composto por um módulo de processamento e comunicações, denominado Módulo CAN PIC FST, e por um painel para representar as rotações do motor, a temperatura do motor, a velocidade, a mudança engrenada e indicar temperatura excessiva do motor, erro de centralina e posição neutra da caixa de velocidades. No painel existe um botão rotativo para seleccionar o modo de operação do volante. Os dez modos de operação possíveis distinguem-se pelos dados que apresentam no painel, permitindo assim ao piloto escolher aquele que representa os dados que pretende. Os modos *shutdown*, *reset*, teste e de demonstração não dão informações sobre o carro, por isso não se destinam a usar quando o carro está em prova, são no entanto úteis para verificar o estado do volante.

Palavras-Chave: Volante Electrónico, CAN, Formula Student, dsPIC

Abstract

This thesis emerges from the will to develop the race car of Formula Student from Instituto Superior Técnico making it more competitive and evolved. The steering wheel of a competition car nowadays is much more than a simple part to control the car's direction. It has information like engine rotations, shifting gear, speed and engine temperature. Those data are necessary for the pilot to optimize his driving and also to monitor the car's status.

During this work the steering wheel for the FST car was projected and built. The wheel communicates with the remaining electronics in the car through the CAN BUS, and receives all the data presented to the pilot.

The system consists in one processing and communications module, designated Módulo CAN PIC FST, and a panel named Painel Volante that represents the engine's rotations and temperature, speed, and warnings such as engine overheat, ECU error and neutral gearbox. On the panel there is a rotation switch to select the wheel's operation mode. Each of the ten available operation modes represents different data according to the pilot's choice. The shutdown, reset, test and demo modes doesn't provide information about the car. Not destined to be used during the race they are very helpful to verify the steering wheel status.

Keywords: Electronic steering wheel, CAN, Formula Student, dsPIC

Índice

Agradecimentos	I
Resumo	II
Abstract	III
Índice	IV
Lista de Figuras	V
Lista de Tabelas	VI
Abreviaturas	VII
Capítulo 1 - Introdução	1
1.1 Objectivos e Motivação	1
1.2 Estado da Arte	2
1.3 Análise aos Volantes de Fórmula 1	3
1.3.1 Volante Renault R23	3
1.3.2 Volante BMW Sauber	4
1.4 Organização da Dissertação	5
Capítulo 2 - Especificações e Arquitectura do Sistema	7
2.1 Especificações	7
2.1.1 Informações no volante	7
2.1.2 Especificações para integração de sistemas electrónicos no FST03	7
2.1.3 Barramento CAN	8
2.1.4 Barramento SPI	8
2.2 Arquitectura do sistema	9
Capítulo 3 - Módulo CAN PIC FST	11
3.1 Arquitectura do sistema	11
3.2 Esquema Eléctrico	12
3.2.1 Regulador DC-DC	12
3.2.2 DsPIC	13
3.2.3 Conectores	14
3.3 PCB do Módulo CAN PIC FST	16
3.4 Consumo do Módulo CAN PIC FST	17
3.5 Estimativa de custos	17
Capítulo 4 - Painel Volante	19
4.1 Arquitectura do sistema	19
4.2 Esquema Eléctrico	21
4.2.1 Drivers MAX7221	21
4.2.2 Botão selecção e conectores	22
4.3 Descrição Funcional	23
4.4 PCB do Painel Volante	24
4.5 Consumo do Painel Volante	25
4.6 Estimativa de custos	26
4.7 Placa USB CAN	27

Capítulo 5 - Software	29
5.1 Corpo Principal do Programa.....	29
5.2 Modos de Funcionamento.....	30
5.3 Versão 0	32
5.3.1 Aquisição Neutro e Erro Centralina	33
5.3.2 Aquisição da RPM	33
5.3.3 Aquisição da Temperatura do Motor	33
Capítulo 6 - Medidas e testes	37
6.1 Consumo	37
6.2 Testes.....	38
Capítulo 7 - Conclusões e Trabalho Futuro	39
Bibliografia	41
Anexo 1 – Esquemas eléctricos e PCBs	43
Anexo 2 – Software	53

Lista de Figuras

Figura 1 - Foto FST03 em Palmela [3].	2
Figura 2 - Volante Renault F1 R23 [10].	4
Figura 3 - Volante BMW Sauber [11].	4
Figura 4 - Esquema ligação SPI.	8
Figura 5 - Arquitectura do sistema.	9
Figura 6 - Arquitectura do Módulo CAN PIC FST.	12
Figura 7 - Montagem do regulador DC-DC.	13
Figura 8 - Esquema eléctrico do dsPIC.	14
Figura 9 - Esquema eléctrico dos conectores.	15
Figura 10 - Foto Módulo CAN PIC FST.	16
Figura 11 - Arquitectura do Painel Volante.	20
Figura 12 - Montagem do MAX7221.	21
Figura 13 - Esquema eléctrico do botão e conector do Painel Volante.	22
Figura 14 - Painel Volante.	23
Figura 15 - Foto Painel Volante.	25
Figura 16 - Arquitectura Placa USB-CAN.	27
Figura 17 - Fluxograma geral do <i>software</i> .	29
Figura 18 - Fluxograma dos ciclos funcionamentos 1,2 e 3.	32
Figura 19- Característica do termistor.	34
Figura 20 - Circuito condicionamento do termistor.	34
Figura 21 - Tensão em função da Temperatura no termistor.	35
Figura 22 - Esquema eléctrico Módulo CAN PIC FST.	43
Figura 23 - PCB do Módulo CAN PIC FST, <i>Top Layer</i> .	44
Figura 24 - PCB do Módulo CAN PIC FST, <i>Bottom Layer</i> .	44
Figura 25 - Componentes do Módulo CAN PIC FST, <i>Top Layer</i> .	45
Figura 26 - Componentes do Módulo CAN PIC FST, <i>Bottom Layer</i> .	45
Figura 27 - Esquema eléctrico Painel Volante.	47
Figura 28 - PCB do Painel Volante, <i>Top Layer</i> .	48
Figura 29 - PCB do Painel Volante, <i>Bottom Layer</i> .	48
Figura 30 - Componentes do Painel Volante, <i>Top Layer</i> .	50
Figura 31 - Foto do Painel Volante.	50
Figura 32 - Foto do sistema, Modulo CAN PIC FST + Painel Volante.	51
Figura 33 - Foto do Painel Volante no modo teste.	51

Lista de Tabelas

Tabela 1 - Legenda do Volante BMW Sauber.....	5
Tabela 2 - Estimativa dos custos de desenvolvimento do Módulo CAN PIC FST.....	18
Tabela 3 - Estimativa de custo do Módulo CAN PIC FST.....	18
Tabela 4 - Descrição Painel Volante.....	23
Tabela 5 - Escalonamento das RPM na Barra 12 <i>LEDs</i>	24
Tabela 6 - Intervalo RPM e temperatura máxima.....	24
Tabela 7 - Estimativa da potência máxima do Painel Volante.....	25
Tabela 8 - Estimativa dos custos de desenvolvimento do Painel Volante.....	26
Tabela 9 - Estimativa de custo do Painel Volante.....	26
Tabela 10 - Modos Funcionamento do Painel Volante.....	30
Tabela 11 - Consumo do sistema embebido.....	37
Tabela 12 - Legibilidade Painel Volante na presença de muita luz exterior.....	38
Tabela 13 - Lista componentes do Módulo CAN PIC FST.....	46
Tabela 14 - Lista componentes Painel Volante.....	49

Abreviaturas

ADC Analog-to-Digital Converter

BCD Binary Decimal Code

CAN Controller Area Network

ECU Engine control unit

FST Formula Student do Instituto Superior Técnico

I2C Inter-Integrated Circuit

LED Light Emitting Diode

MIPS Milhões de Instruções Por Segundo

PCB Printed Circuit Board

RPM Rotações por Minuto

SPI Serial Peripheral Interface

Capítulo 1

Introdução

Os sistemas electrónicos possuem um papel cada vez maior e mais importante no mundo automóvel, e em especial na competição automóvel onde o desenvolvimento tecnológico é muito importante para obter vantagem competitiva. Os pilotos actualmente nas grandes competições automóveis, como a Fórmula 1, têm um volante que é muito mais do que a forma de direccionar o carro. É no volante que estão as informações necessárias para optimizarem a condução e se inteirarem do estado da mecânica do carro. Na Formula Student [1] que é uma competição automóvel de universidades criada em 1998 no Reino Unido, existem provas de avaliação do desempenho do carro e também da tecnologia implementada no carro.

1.1 Objectivos e Motivação

O Objectivo principal deste trabalho é construir um volante electrónico para o carro da Formula Student do Instituto Superior Técnico (FST) [2] semelhante ao que se usa nos Fórmula 1.

A equipa da FST surgiu em 2001 sendo o primeiro carro, o FST01, apresentado em competição em 2002. Em 2005 surgiu em competição o FST02 e em 2006 o carro em uso actualmente o FST03, Figura 1. A equipa é composta por alunos do IST de vários cursos. Os carros são monolugares tipo Fórmula 1 projectados e construídos pela equipa com chassis tubular numa liga de aço e carroçaria em fibra de carbono e fibra de vidro. Utilizam motor de moto (Honda CBR600) e são capazes de desempenhos bastante interessantes como 4 segundos dos 0 aos 100 km/h e velocidade máxima da ordem dos 160 km/h.

A equipa da FST tem neste momento vários projectos de electrónica a decorrer para evoluir o carro, e serão esses projectos que irão fornecer os dados para o volante apresentar. A comunicação entre os sistemas electrónicos será feita pelo barramento Controller Area Network (CAN).

O volante deverá ser um sistema constituído por duas partes, um módulo de processamento e comunicação (Módulo CAN PIC FST) e um painel (Painel Volante) para representar as informações. O Módulo CAN PIC FST é uma placa projectada e construída com o objectivo de servir de base a todos os trabalhos de electrónica no carro. Este módulo será capaz de gerar a partir da bateria do carro a tensão indicada para o seu processador de sinal e todos os periféricos adoptados. O Painel Volante será a interface com o piloto, deverá fornecer todas as informações importantes ao piloto de forma intuitiva e de rápida consulta.

O volante deverá apresentar o valor rotação do motor, temperatura de motor, mudança engrenada e velocidade. Deverá ainda indicar a posição Neutra da caixa de velocidades, avisar em caso de temperatura de motor excessiva e também de erro de centralina.



Figura 1 - Foto FST03 em Palmela [3].

1.2 Estado da Arte

O volante nos automóveis nasceu da necessidade de controlar a direcção do carro. À medida que a indústria automóvel foi evoluindo apareceram novas necessidades como a criação de sistemas electrónicos para aquisição de dados como a velocidade, rotação do motor, consumo de combustível e temperatura de motor. Estes dados são necessários para o desenvolvimento de sistemas de controlo do carro, por exemplo o Electronic Stability Program (ESP) e o Antilock Braking System (ABS) mas ao ter estes dados em formato electrónico o painel de instrumentos tornou-se também um componente electrónico do automóvel em vez dos anteriores painéis de instrumentos com indicadores mecânicos [4]. O excesso de cablagem proveniente da interligação dos sistemas electrónicos foi solucionado com a utilização de um barramento único [5]. Os painéis de instrumentos têm sido constantemente evoluídos, passando a contar com cada vez mais informação sobre o carro e apresentada ao utilizador de melhor forma [6]. Pela posição que ocupa e por ser a peça com que o condutor está sempre em contacto, o volante começou a desempenhar também a função de painel de instrumentos [7]. Além de se tornar o substituto do painel de instrumentos, a electrónica também está presente nos novos volantes na sua função principal de direccionar o carro, especialmente na recente tecnologia Steer by Wire (SBW) [8]. Na competição automóvel e em especial na Fórmula 1, são onde se encontram os volantes tecnicamente mais evoluídos.

1.3 Análise aos Volantes de Fórmula 1

A pesquisa sobre os volantes de Fórmula 1 existentes foi efectuada sobretudo para determinar as informações mais úteis e a forma como são apresentadas ao piloto.

Na década de oitenta o volante de um Fórmula 1 era de forma redonda e com uma placa de metal no centro para uni-lo à coluna de direcção. A partir de 1990 os volantes de Fórmula 1 começaram a ter controlos da caixa de velocidades e botões para o rádio de comunicações com a equipa. Os carros de Fórmula 1 foram evoluindo a todos os níveis e ao surgirem as caixas de velocidades semi-automáticas, as embraiagens hidráulicas automáticas, os controlos de tracção, a possibilidade de modificar o valor limite de rotação do motor, ajustar diferenciais e inclinação das asas entre outras, o volante tornou-se uma das peças mais importantes do carro ao permitir monitorizar todos esses sistemas.

O volante de um Fórmula 1 pesa em média 1300g, e representa um investimento na ordem das dezenas de milhares de Euros. Para reduzir o peso e aumentar a rigidez, são feitos de fibra de carbono e titânio com aplicações em borracha para ter melhor aderência às mãos do piloto. Um dos problemas que afecta o volante é as vibrações a que é sujeito [9]. Mesmo assim, a opinião é unânime, todo o investimento é válido. Afinal, o papel deste dispositivo é decisivo nas corridas. Os actuais volantes permitem ao piloto estar a alterar afinações no carro pelas indicações que está a receber via rádio dos seus engenheiros enquanto roda a 300 km/h!

As funções mais utilizadas estão situadas de maneira que o polegar do piloto possa accioná-las rapidamente, e são usadas cores vivas para ajudar na rápida identificação das funções.

Através de uma análise geral verifica-se que todas as equipas utilizam volantes com muitas semelhanças, de seguida analisa-se os volantes da Renault e da BMW Sauber.

1.3.1 Volante Renault R23

Vejamos o volante da Renault na Figura 2. O maior destaque vai para a informação da rotação do volante de inércia do motor (8), pois é a principal informação para a condução do piloto. É representada numa barra de *LEDs*. Neste documento a rotação do volante de inércia do motor passará a ser designada apenas por rotação do motor. A velocidade, temperatura do motor e outras informações sobre o estado mecânico do carro são dadas ao piloto num *display LCD* (9), este LCD é bastante simples de forma a ser consultado rapidamente. Um botão rotativo (14) permite alternar entre as informações disponíveis para o LCD. O piloto utiliza auriculares e pode comunicar com os seus engenheiros accionado o rádio (5). O limitador de velocidade (17) para usar na via das *boxes*, o ajuste do diferencial (6) e controlo tracção (15). É também no volante que está o comando da caixa de velocidades (16) através de patilhas colocadas na parte de trás do volante, a da direita incrementa a da esquerda reduz. A embraiagem é também accionada por patilhas no volante (7).

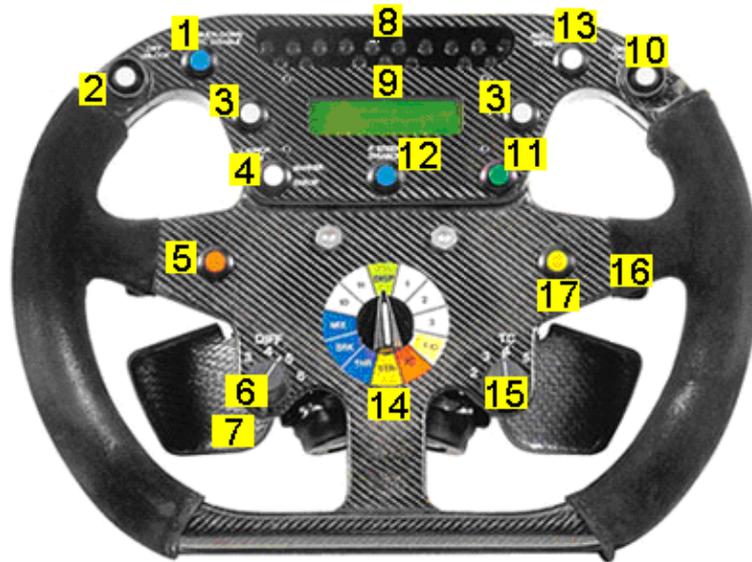


Figura 2 - Volante Renault F1 R23 [10].

1.3.2 Volante BMW Sauber

O volante usado pela equipa BMW Sauber em 2006, Figura 3, tem em comparação com o volante da Renault uma ergonomia mais trabalhada. No entanto mantém a informação da rotação do motor no topo através de uma barra de LEDs e um display LCD para passar as restantes informações consoante a selecção do piloto pelos botões (23). Consultando a Tabela 1, verifica-se que as restantes funções são semelhantes ao volante da Renault, bem como aos das restantes equipas.



Figura 3 - Volante BMW Sauber [11].

Tabela 1 - Legenda do Volante BMW Sauber.

1	Limitador Velocidade para via das boxes	12	Sensor Lambda
2	Diferencial +	13	Diagnóstico
3	Impulso do motor	14	Informação sobre ângulo da asa
4	Caixa velocidades +	15	Embraiagem
5	Controlo de tracção +	16	Seleccção de diferencial
6	Ajuste do impulso do motor	17	Controlo de tracção -
7	Patilha da embraiagem	18	Caixa velocidades -
8	Controlo de tracção	19	Corte motor
9	Informação de voltas pela equipa	20	Diferencial -
10	Burn out	21	Neutro da caixa de velocidades
11	Botão multifuncional	22	Muda página do display

1.4 Organização da Dissertação

Este relatório encontra-se dividido em sete capítulos, sendo no presente capítulo descrita a introdução ao problema e as soluções actualmente existentes.

No capítulo 2 definem-se as especificações e a arquitectura do sistema embebido a desenvolver.

No capítulo 3 e 4 apresenta-se o desenvolvimento do Módulo CAN PIC FST e do Painel Volante desde a sua concepção à produção e apresentando estimativas de custos.

No capítulo 5 descreve-se o *software* desenvolvido para o sistema embebido explicitando as funções desempenhadas e as duas versões desenvolvidas.

No capítulo 6 apresenta-se os testes realizados ao sistema para caracterizar o seu desempenho e as medidas efectuadas.

Por fim no capítulo 7 apresentam-se as conclusões do trabalho realizado e as propostas de trabalho futuro.

Capítulo 2

Especificações e Arquitectura do Sistema

2.1 Especificações

A definição das especificações para o sistema foi a primeira fase do trabalho.

2.1.1 Informações no volante

Partindo do que existe na Fórmula 1 e em reunião com o grupo da FST definiu-se o que era possível de implementar tendo em conta as diferenças que existem entre os carros de Fórmula 1 e o carro da FST em uso, o FST03.

As especificações adoptadas para o volante são as seguintes:

- Indicação da **Rotação do Motor** numa barra de *LEDs*. *LED* para indicar o intervalo de rotação do motor ideal para o piloto realizar a troca de mudança.
- LEDs* avisadores de **Temperatura de Motor Excessiva**, **Erro de Centralina** e da posição **Neutra da caixa de velocidades**.
- Indicação da **Mudança Engrenada**, **Velocidade** e **Temperatura de Motor** através de *displays* de 7 segmentos.
- Para que o painel seja simples, intuitivo e de rápida consulta utilizar um **botão de selecção da informação a mostrar nos *displays***.

Devido às características do carro foi decidido não implementar várias funções que constam nos volantes utilizados actualmente na Formula 1. Os controlos dos diferenciais, limitadores de velocidade e afinadores de asa são funções que não são possíveis de implementar pois não existe mecânica de suporte no carro da FST. O comando da caixa de velocidades é com patilhas no volante mas com um controlo já existente, ficando assim fora deste sistema.

2.1.2 Especificações para integração de sistemas electrónicos no FST03

Foi definido que no carro existirá um barramento CAN. Este barramento permitirá a comunicação entre todos os projectos de electrónica em desenvolvimento para o carro. Será através do barramento CAN que o volante receberá todos os dados provenientes dos sensores que existem no carro. Decidiu-se também em reunião que todos os projectos da equipa de electrónica do FST utilizariam um módulo base igual. Este módulo, denominado de Módulo CAN PIC FST, será comum a

todos os projectos permitindo assim uma base de trabalho igual a todos os elementos da equipa. Permite também uma partilha de *hardware* importante para a fase de desenvolvimento. Este módulo terá que ter capacidades para ser usado em todos os projectos de electrónica da FST. Definiu-se que seria baseado num processador digital de sinal, teria que possuir o protocolo CAN e estar apto a funcionar a partir da tensão da bateria do carro (12 V). Ficou decidido que o módulo seria totalmente desenvolvido nesta dissertação.

2.1.3 Barramento CAN

O barramento CAN [5] é o barramento adoptado para ligação de todos os sistemas electrónicos no carro. Este protocolo foi desenvolvido pela BOSCH nos anos 80 e já vai na 2ª versão (CAN 2.0B). As principais características do protocolo CAN são, de uma forma geral, a possibilidade de estabelecer comunicação entre dispositivos apenas através de dois fios, com velocidades de transmissão até 1 Mbit/s, podendo enviar um máximo de 8 *bytes* de dados em cada mensagem. Permite também que haja detenção de erros, dando a possibilidade do dispositivo causador dos erros se identificar, desligando-se, não interferindo mais no barramento.

Como electricamente a transmissão é realizada por sinais diferenciais, a comunicação tem uma elevada imunidade ao ruído.

2.1.4 Barramento SPI

O *Serial Peripheral Interface* (SPI) é um barramento *full-duplex* desenvolvido para comunicação entre circuitos integrados. Na sua configuração comum, ver Figura 4, este barramento é constituído por três linhas base, o SCK é o sinal de relógio gerado pelo mestre e apenas está activo durante a transmissão de dados, o SDI é o porto de entrada de dados e o SDO o porto de saída de dados. Por cada dispositivo escravo que se adiciona é necessário adicionar uma linha para poder fazer a selecção de dispositivos (*chip select*).

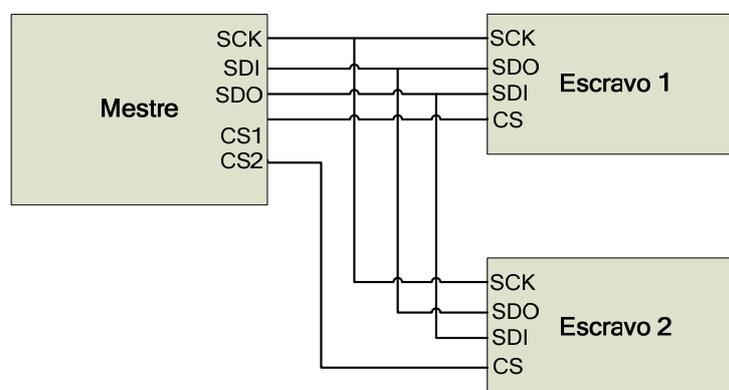


Figura 4 - Esquema ligação SPI.

2.2 Arquitectura do sistema

A Arquitectura do sistema embebido desenvolvido como mostra a Figura 5 tem dois blocos, o Painel Volante e o Módulo CAN PIC FST. A opção de dividir o sistema em duas partes foi tomada essencialmente pela decisão de se construir um módulo base comum a todos os projectos de electrónica no carro, como referido na secção 2.1.2, o Módulo CAN PIC FST. Ao dividir o sistema em Módulo CAN PIC FST e Painel Volante consegue-se também um sistema mais versátil podendo-se alterar uma das partes sem a necessidade de refazer tudo.

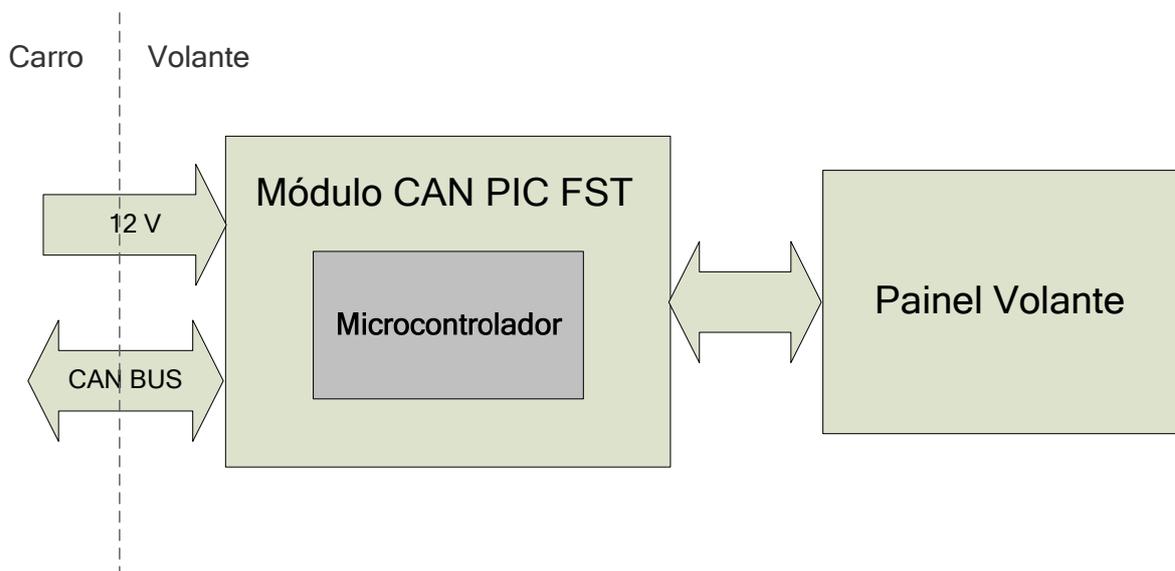


Figura 5 - Arquitectura do sistema.

Capítulo 3

Módulo CAN PIC FST

3.1 Arquitectura do sistema

O Módulo CAN_PIC _FST foi desenvolvido para servir de suporte a todos os trabalhos de electrónica desenvolvidos no carro, por isso as decisões na sua construção foram sempre de forma a torná-lo o mais genérico, poderoso e versátil possível.

O módulo tem por base um microcontrolador de 16-bits, o dsPIC30f6012A [13]. Esta escolha foi realizada em conjunto com todos os elementos da FST. Optou-se por este microcontrolador especializado em processamento digital de sinal, essencialmente devido a projectos que tratam sinais do carro, como é o caso da centralina. Este microcontrolador pode ir até 30 Milhões de Instruções Por Segundo (MIPS) possui um Analog-to-Digital Converter (ADC) de 12 *bits* com 16 canais e contém os principais protocolos de comunicação como o Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C) e dois barramentos de CAN. Estes protocolos poderão ser usados nas mais diversas aplicações tal como todas as outras capacidades deste processador pois é essa a principal função do Módulo CAN PIC FST, é como que uma “*Motherboard*” para o dsPIC. A pequena diferença de preço entre este microcontrolador especializado em processamento digital de sinal e um microcontrolador convencional equivalente, justifica também esta escolha.

No carro a tensão que existe disponível para alimentação são os 12 V da bateria. Como o microcontrolador adoptado funciona para tensões de alimentação entre 2,5 V e 5,5 V é necessário um regulador de tensão.

A opção de colocar a alimentação e os dois barramentos de CAN no mesmo conector (PCAN), deve-se ao facto de assim ser apenas necessário este conector para interligar todos os sistemas electrónicos presentes no carro. Para cada barramento de CAN é necessário um transceptor para fazer a interface do sinal digital do controlador CAN do dsPIC30f6012A para o sinal diferencial usado no barramento. Foi utilizado o MCP2551 [18].

A arquitectura do módulo está representada na Figura 6.

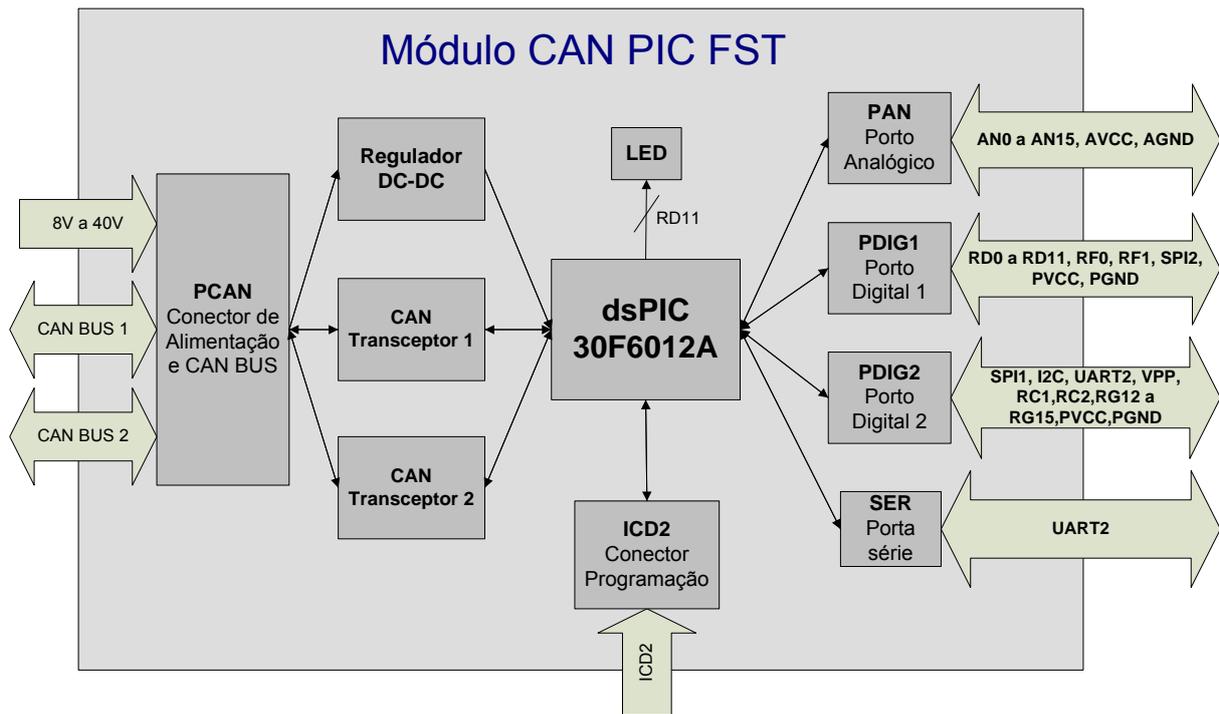


Figura 6 - Arquitectura do Módulo CAN PIC FST.

3.2 Esquema Eléctrico

Depois de definida a arquitectura do Módulo CAN PIC FST realizou-se o esquema eléctrico no *software* Altium Protel DXP 2004. O esquema eléctrico vai ser analisado dividindo-o em três partes. O regulador DC-DC, o dsPIC e os conectores.

O esquema eléctrico completo do Módulo CAN PIC FST encontra-se no Anexo 1.

3.2.1 Regulador DC-DC

O Módulo CAN PIC FST destina-se a ser usado no carro da FST, a tensão que existe disponível no carro são os 12,6 V da bateria. O microcontrolador adoptado funciona para tensões de alimentação entre 2,5 V e 5,5 V, logo é necessário um regulador de tensão. O valor de tensão pretendido para o funcionamento do módulo é 5 V, pois é um valor adequado ao microcontrolador adoptado e é também a tensão usada na maioria dos circuitos e sensores do carro. Foi utilizado um regulador comutado redutor de tensão, o LM2673-5.0 da National Semiconductor [14], com a montagem apresentada na Figura 7. Ao inserir este regulador o módulo pode ser alimentado com tensões entre 8 V e 40 V, que para esse intervalo de tensões de entrada o regulador apresenta à saída a tensão de 5 V com corrente máxima de 3 A. É então possível alimentar periféricos a 5 V pelo Módulo CAN PIC FST desde que a soma da corrente de todos os periféricos e do próprio módulo não

ultrapasse os 3 A. Ao inserir o regulador DC-DC consegue-se não só baixar a tensão para os 5 V como a estabilização da tensão, evitado assim problemas provenientes da variação da tensão da bateria do carro que ocorrem por exemplo no momento da ignição.

A montagem do regulador utilizada possui, segundo o fabricante do controlador, um rendimento de 88% para uma tensão de entrada de 12 V e uma corrente saída de 100 mA até 3 A. O rendimento deste regulador comutado é elevado quando comparado com os reguladores lineares do tipo queda de tensão reduzida (low dropout, LDO).

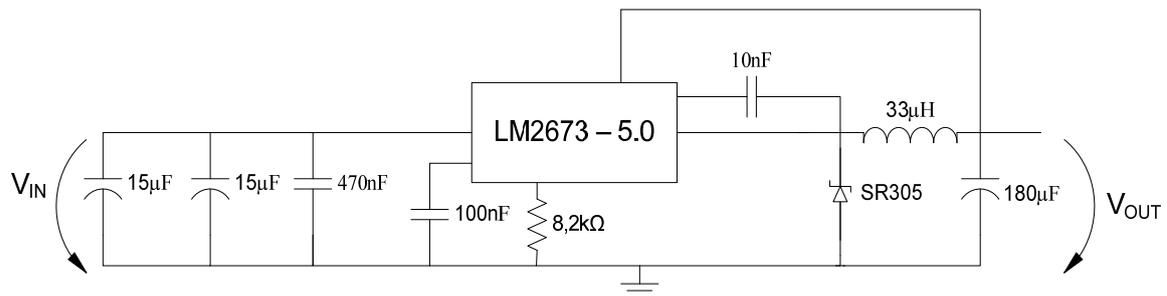


Figura 7 - Montagem do regulador DC-DC.

3.2.2 DsPIC

O dsPIC é o componente central do módulo e como se pode verificar pela Figura 8 tem os seus 64 pinos conectados. Colocou-se um LED no registo RD11 para o programador usar como entender, por exemplo verificar se o programa continua a correr correctamente. O microcontrolador possui como referido anteriormente dois controladores de CAN. Colocou-se para cada controlador um Transceptor de CAN MCP2551 da Microchip [18] para fazer a interface entre a linha de transmissão e recepção que é diferencial e os sinais digitais do controlador de CAN do dsPIC. Este Transceptor além de interface entre o controlador de CAN e o barramento CAN, funciona também como *buffer* protegendo o controlador de CAN. Na escolha deste transceptor teve-se em conta a máxima tensão de alimentação que suporta 7 V, funcionando portanto aos 5 V disponíveis no módulo.

Foi efectuada a montagem com um cristal de 32.768 kHz para o relógio de tempo real nos pinos 47 e 48 do dsPIC segundo as especificações das suas características [13]. O relógio de tempo real foi incluído no módulo pois é útil para várias aplicações, como por exemplo associar a hora e data a uma mensagem CAN. Para o oscilador do dsPIC foi efectuada a montagem com um cristal de 16 MHz nos pinos 38 e 39, esta montagem permite escolher um dos vários modos de operação do oscilador do dsPIC descritos na folha de características [13].

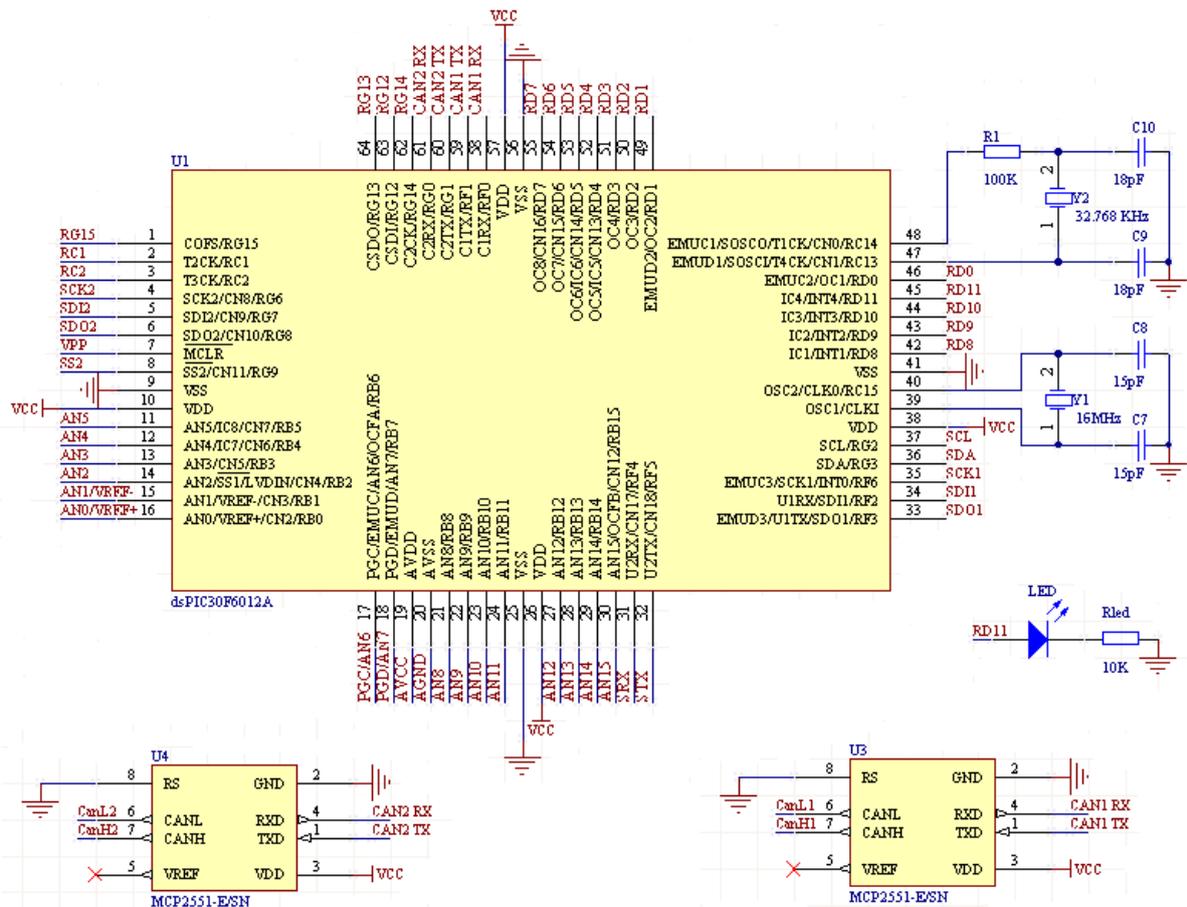


Figura 8 - Esquema eléctrico do dsPIC.

3.2.3 Conectores

Como se pode ver na Figura 9 os 5 V da saída da montagem do regulador, PVCC, são disponibilizados nos conectores PDIG1 e PDIG2 para que se possa alimentar directamente pelo módulo os periféricos com 5V. O dsPIC e todos os outros componentes electrónicos da parte digital do módulo são alimentados a 5V, *net* denominada VCC. A *net* VCC é a mesma da PVCC mas separada logo na saída do regulador. O mesmo acontece no caso da parte analógica do circuito, sendo esta *net* denominada de AVCC. Esta técnica de separação é usada tanto para os 5 V como para a massa, daí a existência de três *nets* de massa. A massa do circuito de potência (PM), a massa do circuito digital (GND) e a massa do circuito analógico (AGND). A necessidade de criar estes três conjuntos de alimentações é para reduzir as interferências na parte analógica geradas pelos componentes digitais e propagadas pela alimentação. Além da separação analógica/digital também interessou separar a montagem do regulador da parte do circuito digital pois os valores de correntes em jogo são de grandezas diferentes. Ao separar estas alimentações logo à saída do regulador permitiu usar pistas com 1mm de largura para suportar os 3 A máximos na montagem do regulador DC-DC e nas fichas PDIG1 e PDIG2. No restante circuito eléctrico as pistas tem 0,254mm de largura que é suficiente para a corrente que as percorre, facilitando assim o *routing* e permitindo reduzir a área do módulo.

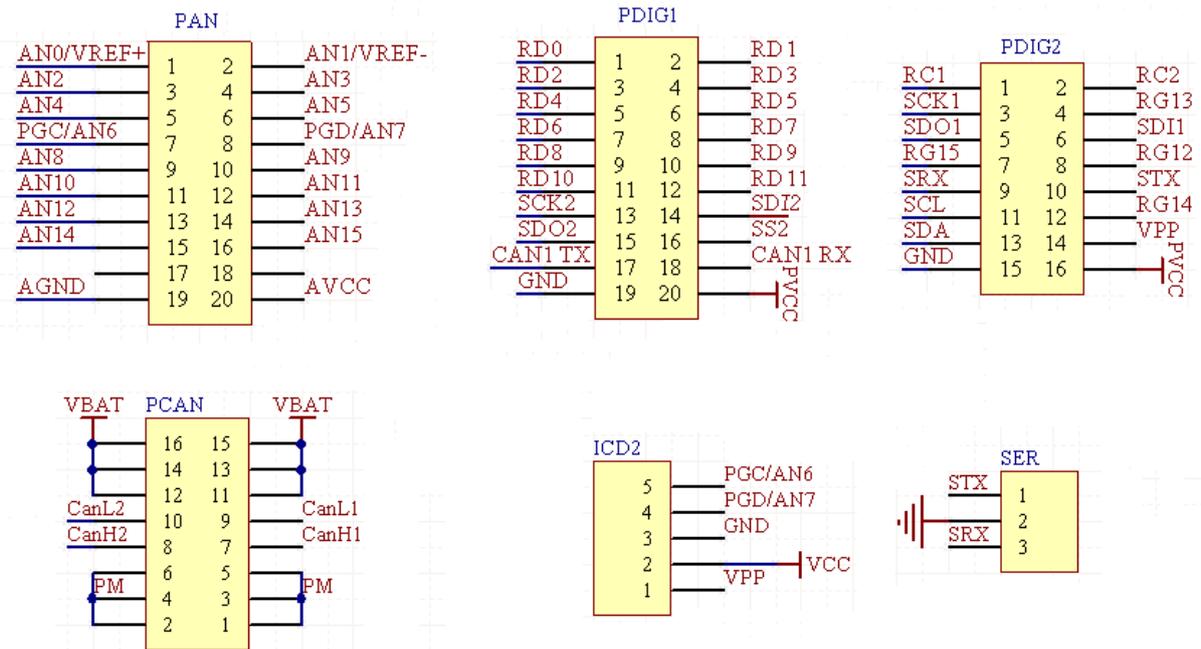


Figura 9 - Esquema eléctrico dos conectores.

Os seis conectores do Modulo CAN PIC FST, Figura 9, foram projectados desta forma pelas seguintes razões:

Alimentação de energia (Power) e CAN BUS (PCAN) – Conector para alimentação do módulo e com dois barramentos de CAN, agrupou-se desta forma para que este seja o conector de ligação á cablagem do carro.

Porta Série (SER) – Conector que possui apenas a UART2 do dsPIC, este protocolo simples de comunicação pode ser bastante útil para algumas aplicações, por exemplo ligação do módulo a um PC.

In-Circuit Debugger 2 (ICD2) – Conector para programação do dsPIC utilizando o MPLAB ICD 2, principal ferramenta de desenvolvimento para microcontroladores da Microchip.

Porto Digital 1 (PDIG1) – Conector com registos digitais do dsPIC e a interface SPI2. O PVCC permite alimentar periféricos a 5 V.

Porto Digital 2 (PDIG2) – Conector com registos digitais do dsPIC e as interfaces SPI1, I2Ce UART2. O PVCC permite alimentar periféricos a 5 V.

Porto Analógico (PAN) – Conector para o 12-bit Analog-to-Digital Converter (ADC) do dsPIC.

3.3 PCB do Módulo CAN PIC FST

Utilizando o *software* Altium Protel DXP foi projectado o PCB. As preocupações ao nível da concepção da PCB foram essencialmente reduzir a área ao mínimo e separar tanto quanto possível a parte analógica da parte digital. Tendo em vista a redução de área a maioria dos componentes escolhidos possuem encapsulamentos Surface-mount Device (SMD). Dado o elevado número de conexões projectou-se a PCB em duas camadas. Na PCB existem três planos de massa denominados GND, PGND e AGND e pistas com diferentes larguras pelas razões apresentadas na secção 3.2.3.



Figura 10 - Foto Módulo CAN PIC FST.

A disposição dos componentes foi efectuada de forma a facilitar o *routing*, mas também separando a parte analógica da parte digital e da parte do regulador DC-DC. Existiu a preocupação de colocar todos os conectores na parte exterior do PCB para facilitar as ligações.

A forma rectangular foi escolhida também com vista a facilitar a colocação do módulo nos vários sistemas a que se destina, uma vez que estes sistemas estarão distribuídos pelo carro em caixas de alumínio. As dimensões exteriores do PCB do módulo são 8,8 cm X 5 cm. Como se pode ver pela Figura 10 existe a cada canto um furo de 3 mm de diâmetro para aparafusar a PCB às caixas de alumínio.

Foram produzidos três protótipos do módulo no laboratório de circuitos impressos do IST-Taguspark para detectar erros e comprovar o seu funcionamento. Estes protótipos foram construídos utilizando a técnica do furo metalizado. Esta técnica é usada para ligar as duas camadas de cobre do substrato PCB através do depósito de cobre nas vias [15]. As pistas são depois desenhadas pela fresadora CNC [16]. Os equipamentos utilizados no laboratório são do fabricante LPKF.

As máscaras de PCB, a lista de componentes e a sua disposição encontram-se no Anexo 1.

3.4 Consumo do Módulo CAN PIC FST

A estimativa do consumo foi efectuada de forma a melhor caracterizar o módulo.

A potência máxima dissipada por um sistema embebido exclusivamente alimentado pelo módulo ocorre quando o regulador DC-DC atinge o máximo da corrente de saída.

Dado que para uma tensão de entrada do Módulo CAN PIC FST de 12 V, como é o caso no carro, o rendimento do regulador presente no módulo é de 88% e a tensão de saída 5 V com corrente máxima de 3 A. A potência máxima fornecida pelo regulador é então os 15 W, dados pela expressão (1).

$$P = V \times I \quad (1)$$

A potência fornecida pelo regulador é então a potência útil no ponto de vista do rendimento, como o rendimento é de 88% pela expressão (2) obtém-se então a máxima potência dissipada pelo Módulo CAN PIC FST, que é 17 W.

$$\eta = \frac{P_{util}}{P_{Consumida}} \times 100 \quad (2)$$

Como o módulo está preparado para alimentar os seus próprios periféricos, interessa também determinar qual a potência máxima disponível para alimentar os periféricos. Para determinar a potência disponível para alimentar periféricos no Módulo CAN PIC FST, basta então fazer a diferença entre a potência máxima disponível no regulador e a dissipada pelo circuito. A potência máxima dissipada pelo circuito presente no módulo além do regulador é de 2 W. Este valor é obtido pela expressão (1) nas condições de máximo consumo do dsPIC e dos transceptores de CAN dadas pelos fabricantes, que são 250 mA e 150 mA respectivamente.

A potência máxima disponível para periféricos no Módulo CAN PIC FST é então 13 W.

3.5 Estimativa de custos

O módulo destina-se a ser usado em muitos projectos, como tal optou-se por mandar produzir as PCBs a uma fábrica. Esta decisão justifica-se porque o processo de produção no laboratório é lento e tem elevados custos, devido ao desgaste da ferramenta da fresadora e ao preço dos reagentes em especial no caso da grafite. A Tabela 2 tem uma estimativa dos custos de desenvolvimento do módulo partindo do pressuposto que o projectista que a desenvolve tem uma remuneração de 1000 €/mês.

Tabela 2 - Estimativa dos custos de desenvolvimento do Módulo CAN PIC FST.

Processo	Dias	Custo
Projecto da arquitectura do módulo	2	90,90 €
Seleccção de componentes	2	90,90 €
Execução do esquema eléctrico	1	45,45 €
Execução das máscaras de PCB	5	227,25 €
Fabrico do protótipo	6	272,70 €
Teste e correcções	2	90,90 €
Total	18	818,10 €

Os valores presentes na Tabela 2 são custos de engenharia não recorrente, interessa agora realizar uma estimativa do custo de produção de cada módulo, pois será esse o valor a ter em conta cada vez que se pondera a utilização do Módulo CAN PIC FST num projecto, Tabela 3. A estimativa de custo de cada módulo é baseada numa encomenda de 50 unidades. Os preços apresentados foram obtidos nos fornecedores *Eurocircuits* e *Farnell*.

Tabela 3 - Estimativa de custo do Módulo CAN PIC FST.

Componete	Qtd	Preço	Custo
PCB	1	3,58 €	3,58 €
Microcontrolador	1	9,43 €	9,43 €
Transciver CAN	2	0,98 €	1,96 €
Regulador DC-DC	1	4,51 €	4,51 €
Diodo Schottky	1	0,38 €	0,38 €
Bobine	1	1,46 €	1,46 €
Condensadores	19	0,15 €	2,85 €
Resistências	8	0,05 €	0,40 €
Conectores	6	1,12 €	6,72 €
Cristais	2	0,80 €	1,60 €
LED	1	0,16 €	0,16 €
Total			33,05 €

A análise de custo demonstra que o custo com a concepção, desenvolvimento e testes são muito elevados quando comparados com o custo de produção. O custo de produção apurado de 33,05 € é suficientemente baixo para este módulo poder ser utilizado em projectos futuros.

Capítulo 4

Painel Volante

4.1 Arquitectura do sistema

O Painel Volante é a interface entre o piloto e o sistema. É no painel que se encontram disponíveis as informações para o piloto e o botão para seleccionar o que pretende ver nos *displays*.

Tendo em vista as especificações definidas na secção 2.1.1, optou-se por uma barra de 12 *LEDs* para representação da rotação do motor. Dos quais quatro verdes, quatro amarelos e quatro vermelhos para distinguir mais facilmente o intervalo de rotação em que o motor se encontra. Para assinalar o intervalo de rotação ideal para a troca de mudança colocaram-se dois *LEDs* azuis.

Para controlar os *LEDs* e os *displays* de 7 segmentos usou-se um *driver* da MAXIM o MAX7221 [17]. Ao utilizar este *driver* consegue-se controlar todos os *LEDs* e os *displays* por uma comunicação série, o SPI, evitando assim ter que usar um porto do microprocessador para cada *LED*, o que seria bastante “desagradável”.

São utilizados *LEDs* e *displays* de alto brilho em todos os casos uma vez que o volante será utilizado em condições de muita luz exterior como é o caso de dias de sol, mas também poderá ser usado em dias de pouca luz pelo que muito brilho nos *LEDs* e *displays* pode incomodar o piloto. Este problema levou à introdução de reguladores de intensidade luminosa que permitem ajustar a corrente fornecida pelos *drivers* aos *LEDs* e *displays*. Além destes reguladores por *hardware* que podem ser ajustados a qualquer momento, existe também a possibilidade configurar os *drivers* [17] por *software* ao nível do brilho. Por esta razão foram utilizados dois *drivers*, pois a possibilidade de um diferente ajuste de brilho em cada *driver*, permite destacar a informação mais importante, que neste caso é a indicação da mudança engrenada e do intervalo de rotação ideal para troca de mudança.

O Painel Volante foi desenhado para ser um periférico do Módulo CAN PIC FST, e está ligado ao módulo pelo porto digital 2 (PDIG2). Neste porto do Módulo CAN PIC FST estão disponíveis os 5 V para alimentação do Painel Volante, o SPI para controlar os *drivers* e os registos digitais usados para ler o botão de selecção.

A arquitectura do Painel Volante é apresentada na Figura 11.

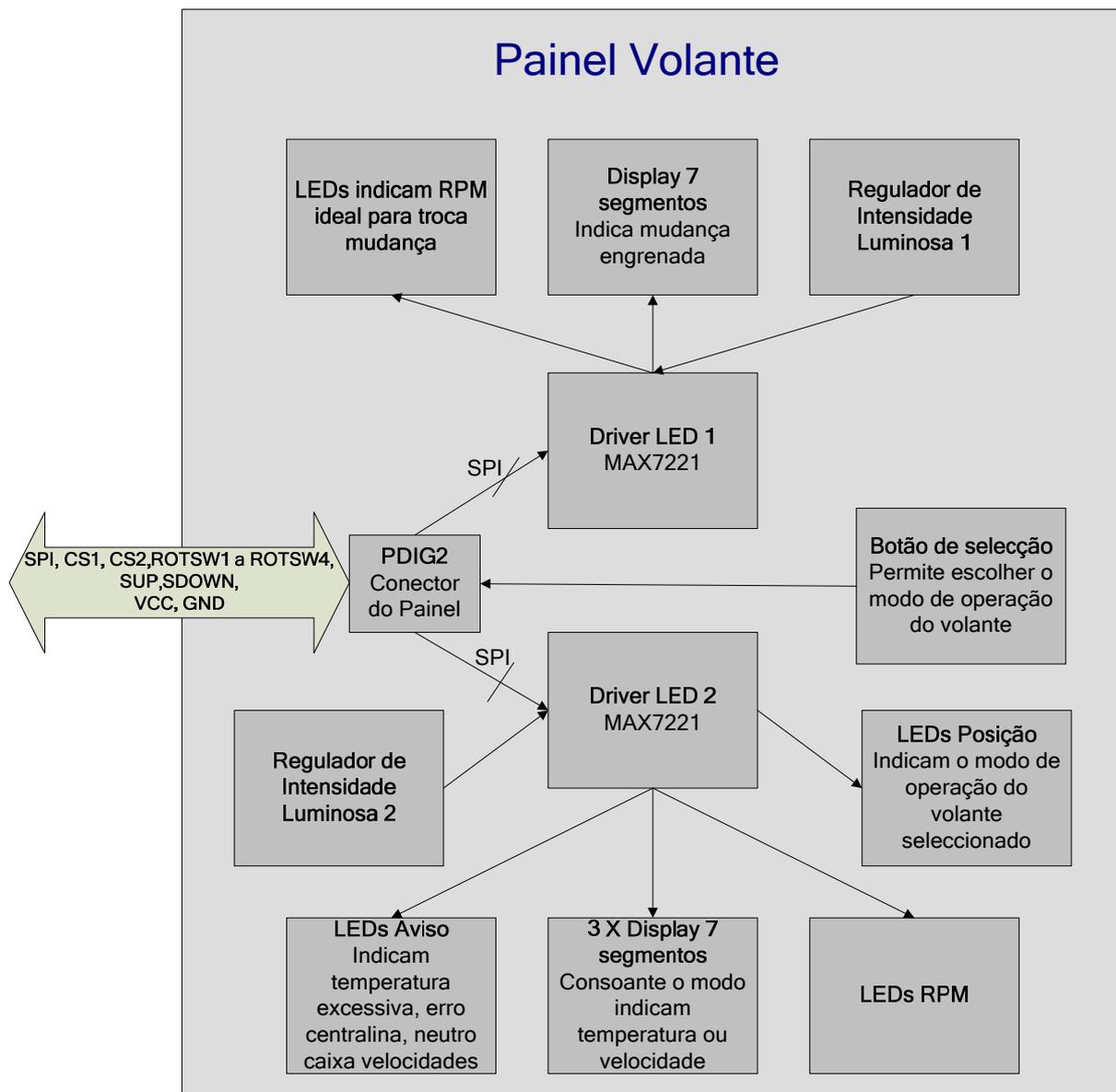


Figura 11 - Arquitectura do Painel Volante.

4.2 Esquema Eléctrico

Depois de definida a arquitectura do Painel Volante realizou-se o esquema eléctrico no *software* Altium Protel DXP 2004.

O esquema eléctrico completo do Painel Volante encontra-se no Anexo 1.

4.2.1 Drivers MAX7221

O esquema eléctrico do Painel Volante tem por base os dois drivers de *LEDs* MAX7221 [17]. Foi escolhido este *driver* por ser indicado para trabalhar a tensões de alimentação entre 4 V e 5,5 V, o que está perfeitamente de acordo com os 5 V que temos disponíveis. O MAX7221 tem capacidade para fazer a interface entre um microprocessador e 64 *LEDs* individuais ou 8 displays de 7 segmentos de cátodo comum. Permite também fazer uma utilização mista, em que uns dígitos usam o Binary Decimal Code (BCD) e outros ficam se codificação para usar com *LEDs* individuais, o que é útil nesta esta aplicação. A comunicação com o microprocessador é feita por SPI, ficando os dados guardados na memória RAM interna de 8X8. Pela montagem do MAX7221 presente na Figura 12 verifica-se que SEG A a SEG DP são as fontes de corrente e são accionadas em conjunto com os DIG0 a DIG 7 consoante os dados presentes na memória interna de forma a representar em cada display o valor correcto. No caso de serem usados 8 displays a frequência de refrescamento mínima é de 500 Hz.

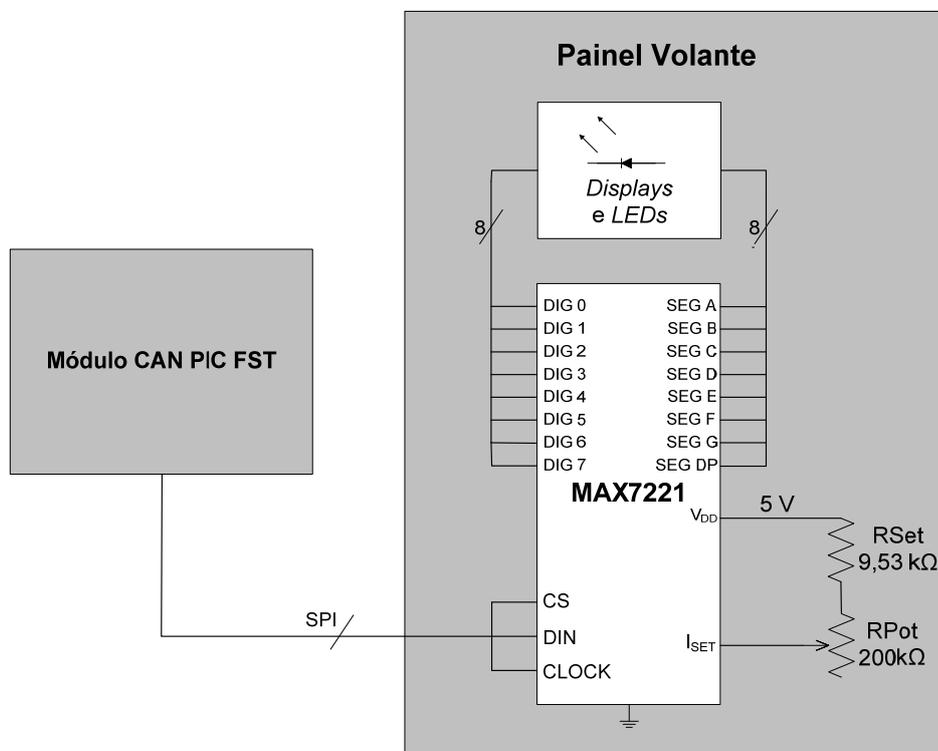


Figura 12 - Montagem do MAX7221.

A montagem realizada no Painel Volante está representada na Figura 12. A corrente no porto I_{SET} define a corrente fornecida pelos segmentos (SEG A a SEG DP). A corrente máxima aconselhada pelo fabricante nos segmentos é 40 mA, o que corresponde a uma resistência de 9,53 k Ω no porto I_{SET} . Além da resistência de valor fixo para garantir o limite máximo de corrente, optou-se por colocar um potenciômetro para se poder ajustar a corrente fornecida pelos segmentos e assim conseguir ajustar o brilho também por *hardware*. A escolha dos *displays* foi realizada em conjunto com a dos *drivers*. Os principais critérios de escolha foram a intensidade luminosa, o tamanho, o preço e a arquitectura de cátodo comum para ser compatível com os *drivers*.

4.2.2 Botão selecção e conectores

Para seleccionar a diferentes informações que o Painel Volante pode disponibilizar, optou-se por utilizar um botão rotativo. O botão rotativo escolhido possui 10 posições o que é mais que suficiente para a aplicação. Na Figura 13 encontra-se o esquema eléctrico do botão e do conector do painel. O conector do painel foi desenhado de forma a ser totalmente compatível com o PDIG2 do Módulo CAN PIC FST, ver Figura 9. No esquema eléctrico do botão apenas interessa destacar que este codifica a posição em código binário, nos pinos ROTSW1 a ROTSW4. São colocadas resistências de *Pull-Down* para garantir o nível lógico 0 quando os pinos são colocados em alta impedância pelo botão. As resistências de *Pull-Down* vão provocar um consumo de corrente quando é colocado o pino no nível lógico 1, como foram usadas resistências de 10 k Ω essa corrente será 0,5 mA que é um valor baixo quando comparado com as correntes consumidas pelos restantes componentes do circuito.

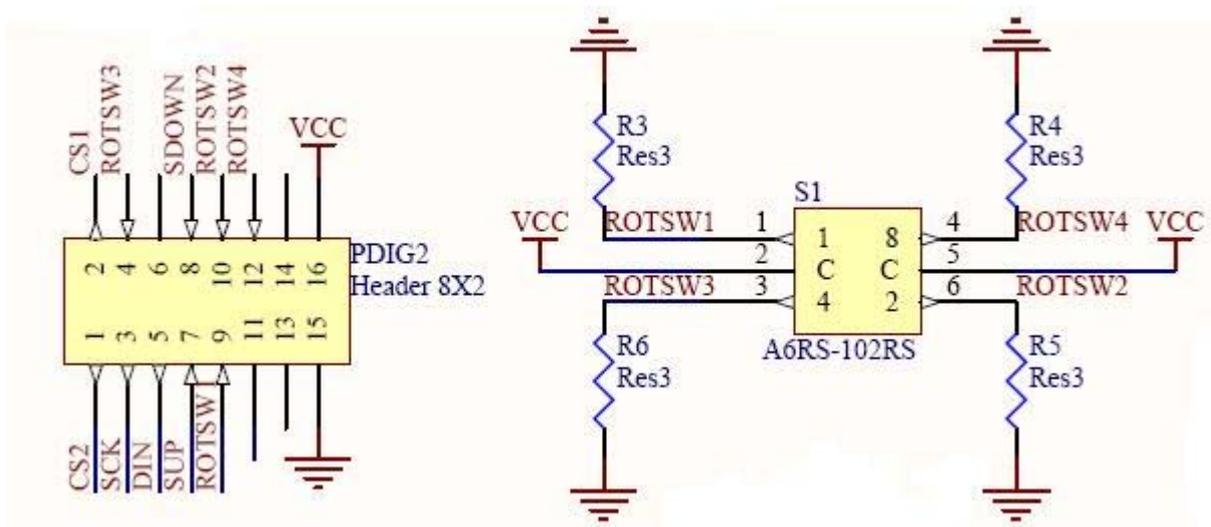


Figura 13 - Esquema eléctrico do botão e conector do Painel Volante.

4.3 Descrição Funcional

Na Figura 14 está ilustrado o Painel Volante e na Tabela 4 a descrição das principais funções.

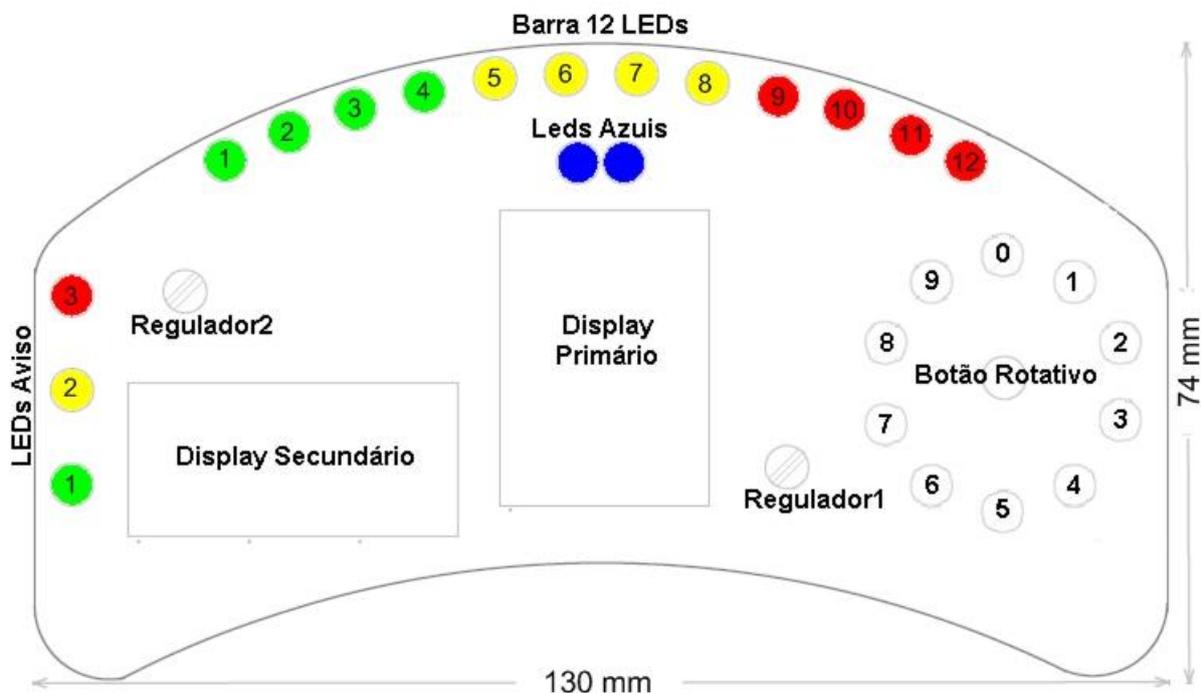


Figura 14 - Painel Volante.

Tabela 4 - Descrição Painel Volante.

	Indicação	Principais considerações
Barra 12 Leds	Rotação do motor	Colocado no topo e com esquema de cores para o piloto se aperceber rápida e claramente do regime do motor
LEDs Azuis	Intervalo rotação do motor ideal para passagem de caixa	Colocado em posição central e com LEDs de cor azul para captar atenção
LEDs Aviso	1-Neutro caixa velocidades 2-Erro centralina 3-Temperatura motor excessiva	Utilização de cores distintas para fácil distinção de cada aviso
Display Primário	Mudança Engrenada	Posição central e <i>display</i> de grandes dimensões
Display Secundário	Temperatura do motor ou velocidade	Três dígitos são suficientes para representar a velocidade em km/h e a temperatura em °C

Além das funções descritas na Tabela 4, o Painel Volante possui também o botão rotativo que permite escolher entre a visualização da temperatura do motor e a velocidade, respectivamente posição 1 e posição 2. As restantes oito posições do botão são utilizadas para demonstrações e testes ao volante, como descrito na Tabela 10. O Regulador1 serve para ajustar o brilho do *Display* Primário e dos *LEDs* Azuis, enquanto que o Regulador2 ajusta o brilho da Barra 12 *LEDs*, os *LEDs* de Aviso e o *Display* Secundário.

Em reunião com o grupo FST, incluindo os pilotos, decidiu-se os valores a atribuir a cada um dos *LEDs* da rotação, o intervalo de rotação considerado ideal para a passagem de caixa e o valor a partir do qual a temperatura do motor é excessiva, ver Tabela 5 e Tabela 6. Estes valores são definidos por *Software*, não existindo qualquer limitação ao nível do *hardware*, sendo portanto possível alterá-los por programação se desejado.

Tabela 5 - Escalonamento das RPM na Barra 12 *LEDs*.

<i>LED</i>	1	2	3	4	5	6	7	8	9	10	11	12
RPM	500	2000	8000	8500	9000	9500	10000	10500	11000	11500	12000	12500

Tabela 6 - Intervalo RPM e temperatura máxima.

<i>LEDs</i> Azuis	<i>LED</i> Aviso 3
11000 a 12000 RPM	90 °C

A definição dos valores da Tabela 5 e Tabela 6 têm essencialmente a ver com as características mecânicas do carro. O *LED1* da rotação liga a partir das 500 rpm para que sempre que o motor esteja em funcionamento o *LED1* esteja ligado, já que o ralenti do motor do carro ronda as 1000 a 1500 rpm. O *LED2* liga a partir das 2000 rpm para indicar que o carro está a uma rotação superior ao ralenti normal. Os *LEDs* 3 a 12 ligam a partir de rotações entre as 8000 e 12500 rpm pois é neste intervalo de rotações que o motor desenvolve a maior potência. Estes dados foram fornecidos pela equipa do FST.

4.4 PCB do Painel Volante

Utilizando o *software* Altium Protel DXP foi projectada a PCB partido das medidas e forma exterior convenientes para ser implementado no interior do volante convencional actualmente usado no FST, que é o OMP Formula Quadro [12]. A PCB foi projectada em duas camadas e os componentes dispostos de forma a respeitar as considerações da secção 4.3.

Foram produzidos dois Painel Volante no laboratório de circuitos impressos do IST-Taguspark pelas mesmas técnicas utilizadas aquando da produção dos protótipos do Módulo CAN PIC FST, secção 3.3.

Construiu-se uma caixa em fibra de carbono, em colaboração com o grupo de mecânica do FST, para fixar o painel no centro do volante OMP Formula Quadro. A Figura 15 mostra o Painel Volante colocado na caixa e com todos os LEDs ligados.

No Anexo 1 encontram-se as máscaras de PCB, a disposição dos componentes e fotos do Painel Volante.



Figura 15 - Foto Painel Volante.

4.5 Consumo do Painel Volante

A estimativa do consumo máximo do Painel Volante foi realizada para garantir que este pode ser alimentado exclusivamente pelo Módulo CAN PIC FST.

Para estimar o consumo máximo do Painel Volante, considera-se a situação de máxima corrente nos *drivers* de LEDs e a posição 7 do botão de selecção já que esta posição coloca três resistências de *Pull-Down* a dissipar corrente, ver Figura 13. Para os 5V de tensão de entrada a corrente máxima, indica pelo fabricante, no driver Max7221 é 330 mA. Existe ainda um consumo de 0,52 mA associado ao pino I_{SET} de cada driver correspondentes à situação em que o potenciómetro está a zero, ver Figura 12.

Na Tabela 7, encontra-se a estimativa da potência máxima dissipada no Painel Volante dada pela expressão (1). Como se comprova pelos valores obtidos os consumos nas resistências de *Pull-Down* e no I_{SET} são desprezáveis quando comparados com o consumo dos *drives* que alimentam todos os *LEDs* do painel.

Tabela 7 - Estimativa da potência máxima do Painel Volante.

	Corrente máxima (mA), para $V_{IN} = 5V$	Potência (W)
Drivers	$2 \times 330 + 2 \times 0,52 = 661,4$	3,31
Botão Selecção	$3 \times 0,5 = 1,5$	0,01
Total	662,9	3,32

A potência máxima do Painel Volante é 3,32 W, o que indica que pode ser alimentado pelo Módulo CAN PIC FST, dado que o módulo pode fornecer até 13 W para os periféricos, como determinado na secção 3.4.

4.6 Estimativa de custos

Foi realizada uma estimativa de custos do Painel Volante, pois apesar de não se destinar a produzir vários exemplares, o custo é sempre um dado importante de um projecto.

Na estimativa do custo de engenharia não recorrente, ou seja os custos de desenvolvimento admite-se que o projectista que a desenvolve tem uma remuneração de 1000 €/mês. A estimativa realizada encontra-se na Tabela 8.

Tabela 8 - Estimativa dos custos de desenvolvimento do Painel Volante.

Processo	Dias	Custo
Projecto da arquitectura do painel	1	45,45 €
Seleção de componentes	2	90,90 €
Execução do esquema eléctrico	1	45,45 €
Execução das máscaras de PCB	3	136,35 €
Fabrico do protótipo	5	227,25 €
Teste e correcções	1	45,45 €
Total	13	590,85 €

Apesar de não estar prevista a utilização do Painel Volante noutros projectos, ao contrário do que acontece com o Módulo CAN PIC FST, realizou-se também uma estimativa do custo de produção por unidade. A estimativa apresentada na Tabela 9 é baseada em valores dos fornecedores Eurocircuits e Farnell para uma encomenda de 50 unidades.

Tabela 9 - Estimativa de custo do Painel Volante.

Componete	Qtd	Preço	Custo
PCB	1	5,54 €	5,54 €
Driver LEDs	2	5,61 €	11,22 €
Botão Rotativo	1	3,46 €	3,46 €
LED Alto Brilho	27	0,26 €	7,02 €
Display Principal	1	1,06 €	1,06 €
Display Secundário	3	0,60 €	1,80 €
Resistências	6	0,05 €	0,30 €
Conector	1	1,08 €	1,08 €
Potenciómetro	2	0,73 €	1,46 €
Total			32,94 €

A análise de custos efectuados demonstra que o custo com a concepção, desenvolvimento e testes são bem mais elevados que o custo de produção, como é comum neste tipo de projectos.

4.7 Placa USB CAN

Como o barramento CAN do carro da FST ainda se encontrava em desenvolvimento numa outra dissertação e também para poder testar em laboratório a comunicação CAN implementada no sistema foi utilizada a Placa USB CAN. Esta placa foi desenvolvida no âmbito de outra dissertação.

A Placa USB CAN faz a interface entre a comunicação USB de um computador e o barramento CAN. Usando esta placa e a ferramenta de *software* desenvolvida na linguagem *C Sharp* consegue-se monitorizar um barramento CAN através do computador.

A Placa USB CAN tem a arquitectura representada na Figura 16. O microcontrolador usado na placa é o PIC18F4550. Este microcontrolador possui interface USB mas não tem integrado o controlador de CAN, daí a necessidade de utilizar o controlador de CAN MCP2515 [20], que possui interface SPI tal como o PIC18F4550. O Transceptor MCP2551 é utilizado pelas mesmas razões que no Módulo CAN PIC FST descritas na secção 3.2.2.

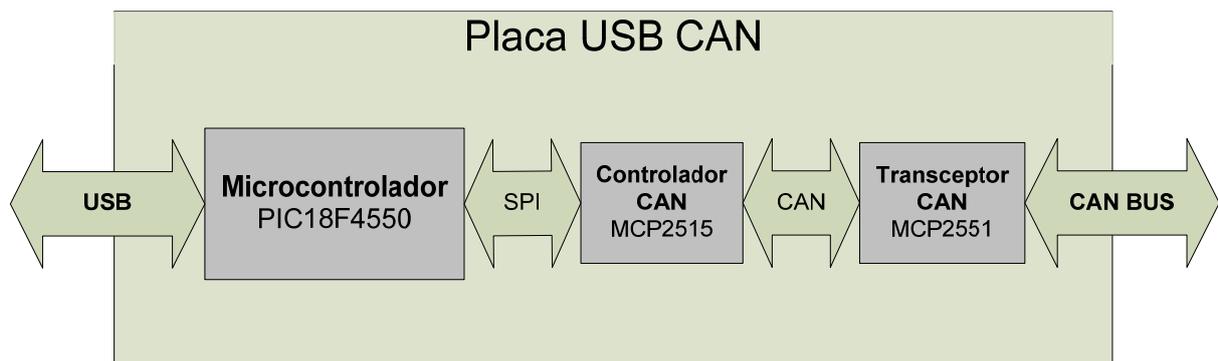


Figura 16 - Arquitectura Placa USB CAN.

Capítulo 5

Software

Para que todo o sistema construído desempenhe as funções para que foi projectado é necessário desenvolver o software adequado para o microcontrolador do sistema. O microcontrolador do sistema é o dsPIC30f6012A do Módulo CAN PIC FST. O Painel Volante é um periférico do Módulo CAN PIC FST e não possui nenhum microcontrolador para programação.

O *software* foi realizado na ferramenta *MPLAB IDE v8.20A* com o compilador *MPLAB C30 Toolsuite* [21]. Esta ferramenta de desenvolvimento pertence ao *Kit MPLAB ICD2*.

No anexo 2 encontra-se as principais partes do código realizado.

5.1 Corpo Principal do Programa

Na Figura 17 encontra-se um fluxograma explicativo do funcionamento geral do *software* implementado.

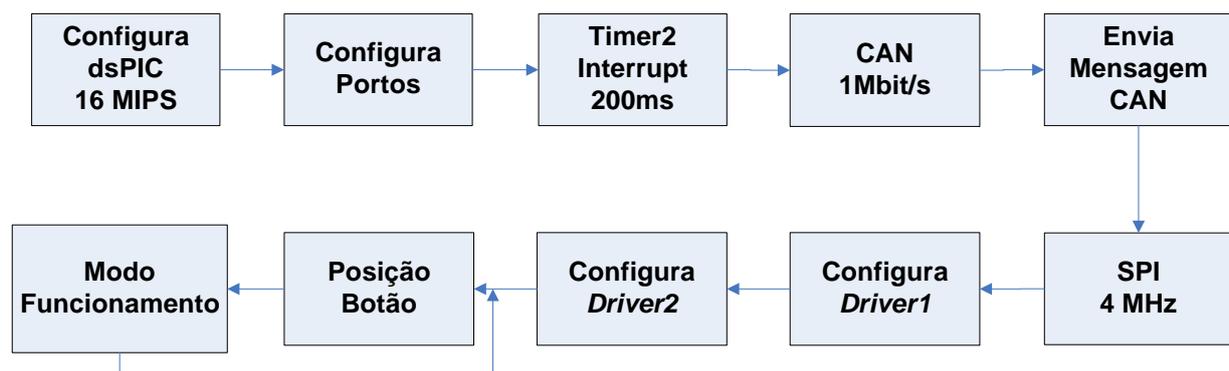


Figura 17 - Fluxograma geral do *software*.

O dsPIC é configurado para funcionar a 16 MIPS, configuram-se o porto correspondente ao *LED* do Módulo CAN PIC FST (RD11) como saída, configuram-se também como de saída os portos utilizados para fazer o *chip select* (RC1 e RC2) na comunicação SPI. São configurados como entradas os portos que fazem a leitura do botão de selecção (RF4,4 e RG13,15). Para verificar se o programa desenvolvido para o dsPIC continua a correr ao longo do tempo sem problemas, implementou-se uma interrupção associada ao timer2 que ocorre a cada 0,2 segundos fazendo piscar o *LED* do Módulo CAN PIC FST.

O controlador de CAN é configurado para funcionar a um *bit rate* de 1 Mbit/s, pois é o *bit rate* definido para barramento CAN do carro. As mensagens de CAN possuem um identificador que permite configurar os filtros de recepção de mensagens para que o controlador de CAN detecte quais as mensagens no barramento que têm os dados dos sensores que interessam ao volante, desta forma o controlador de CAN descarta automaticamente as mensagens que não contém informações úteis ao volante, gerando apenas interrupções quando existem novos dados referentes aos sensores

que interessam ao volante. Depois de inicializado o controlador de CAN, é enviada uma mensagem para assinalar o funcionamento do volante.

O controlador de SPI é configurado e inicializado para funcionar a uma frequência de 4 MHz. É pelo SPI que se comandam os *drivers* de LEDs do Painel Volante. Para realizar a configuração do *driver1* começa-se por fazer um *reset*, de seguida escolhe-se que dígitos ficam com codificação BCD, define-se a intensidade luminosa que neste caso foi definida como máxima pois já temos o ajuste por *hardware*. Define-se o número de dígitos em uso para que o *driver* faça o refrescamento apenas desses dígitos conseguindo assim uma maior frequência de refrescamento o que leva a uma maior intensidade luminosa. Por fim coloca-se o *driver* em modo de funcionamento normal. O processo repete-se para o *driver2*.

5.2 Modos de Funcionamento

Depois de inicializado todo o sistema, este passa a um ciclo infinito em que lê a posição do botão de selecção e executa o respectivo modo de funcionamento. Este ciclo de funcionamento possui os dez modos de funcionamento do Painel Volante, um para cada posição do botão como representado na Tabela 10.

Tabela 10 - Modos Funcionamento do Painel Volante.

Posição do Botão	Designação	Descrição
0	<i>Reset</i>	Reinicia os <i>drivers</i>
1	Race1	Apresenta Rotação do Motor, Mudança Engrenada, Temperatura do Motor, Erro Centralina, Neutro e Temperatura Excessiva consoante os dados que chegam pelo CAN
2	Race2	Apresenta Rotação do Motor, Mudança Engrenada, Velocidade, Erro Centralina, Neutro e Temperatura Excessiva consoante os dados que chegam pelo CAN
3	Demo	Simula Mudança Engrenada, Rotações e Temperatura do Motor
4	Normal1	<i>Drivers</i> funcionarem no modo normal
5	Shutdown	Desliga os <i>drivers</i> de LEDs apagando todo o Painel Volante
6	Normal2	<i>Drivers</i> funcionarem no modo normal
7	DemoFST	Passa as iniciais FST no <i>display</i> primário
8	Teste Off	Desliga o modo teste dos <i>drivers</i>
9	Teste	Liga todos os LEDs do Painel Volante

Verifica-se que os modos designados como Race1 e Race2 são os dois modos mais importantes do volante, dado que é nestes modos que o Painel Volante está a desempenhar a sua função de informar o piloto dos dados do carro que chegam periodicamente pelo CAN. A diferença entre estes dois modos é apenas no display secundário, no modo Race1 é a Temperatura do Motor em °C enquanto que no modo Race2 é apresentada a estimativa da velocidade do carro em km/h. A estimativa de velocidade apresentada é calculada fazendo a média da velocidade das duas rodas da frente, já que as rodas traseiras são motrizes e por exemplo no arranque têm tendência a patinar. É feito a média das duas rodas porque em curva por exemplo a roda interior percorre uma distância mais curta que a exterior no mesmo espaço de tempo.

O modo Demo é um modo de demonstração em que se representa no Painel Volante três contadores simulando, a rotação do motor entre 0 e 13500 rpm, a temperatura do motor entre 0 e 140 °C e a mudança engrenada entre 1ª e 6ª.

O modo Reset utiliza-se para reiniciar os drivers de LEDs do Painel Volante. Começa-se por apagar todas as configurações e dados anteriores do driver, de seguida escolhe-se que dígitos ficam com codificação BCD, define-se a intensidade luminosa que neste caso foi definida como máxima pois já temos o ajuste por hardware. Por fim define-se o número de dígitos em uso no driver e coloca-se em funcionamento normal. Este modo é útil por exemplo para uma situação em que se desligue o Painel Volante do Módulo CAN PIC FST e volte a ligar, pois nesta situação é necessário reconfigurar os drivers.

O modo Normal1 e Normal2 são modos em que apenas se envia o comando para os drivers funcionarem no modo normal.

O modo Shutdown foi implementado para colocar o painel na situação de menor consumo, desligando os drivers dos LEDs.

O modo Teste liga todos os LEDs, permitindo verificar se existe um problema com algum. Neste caso estamos na situação de maior consumo do painel.

Na Figura 18 encontra-se um fluxograma explicativo dos três modos mais utilizados que são o Race1, o Race2 e o Demo.

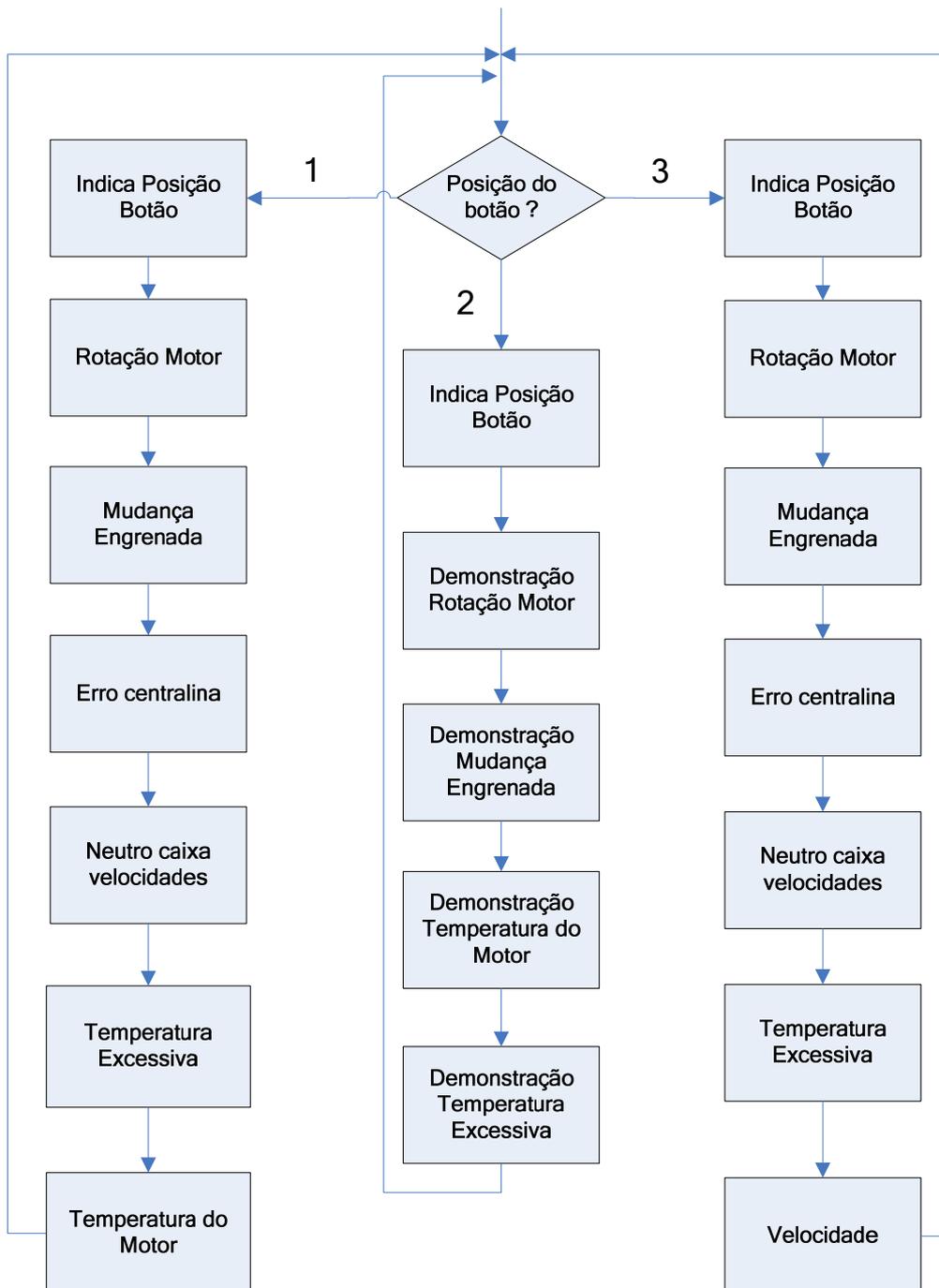


Figura 18 - Fluxograma dos ciclos funcionamentos 1,2 e 3.

5.3 Versão 0

Na fase inicial foi realizada uma versão do *software*, denominada versão 0, que obtém os dados da posição neutra da caixa de velocidades, o erro de centralina, a rotação e a temperatura do motor directamente dos sensores presentes no carro. Na versão final todos os dados são obtidos através do barramento CAN.

A versão 0 foi útil para testar o volante enquanto não existia o barramento CAN, esta versão permite também caso se pretenda que seja o volante a ler os sensores do carro e colocar esses dados no CAN. Esta solução não foi utilizada pois implicaria colocar toda a cablagem dos sensores no volante, o que não é uma boa solução já que o volante deve ter o mínimo de ligações possíveis.

5.3.1 Aquisição Neutro e Erro Centralina

A aquisição do sinal de neutro da caixa de velocidades e do erro de centralina com os sensores originais do carro é bastante simples. Colocou-se uma resistência de *pull-up* para os 5 V em cada um dos sinais, já que ambos os sensores o que fazem é simplesmente colocar a linha à massa no caso afirmativo de aquilo que indicam. Basta então ler cada uma das linhas com um porto digital do dsPIC para obter a indicação da posição neutra da caixa de velocidades e o do erro de centralina.

5.3.2 Aquisição da RPM

A versão 0 obtém a rotação do motor através de um sinal quadrado que está disponível na centralina com frequência f proporcional ao número de rotações por minuto (RPM) do motor, dado pela expressão (3).

$$f = \frac{RPM}{30} \quad (3)$$

Para determinar a frequência do sinal quadrado usou-se o porto *Input Capture* do dsPIC (RD10), que gera uma interrupção a cada mudança de nível lógico permitindo o cálculo do tempo entre cada mudança de nível (período), obtendo assim a frequência. Foi necessário colocar um divisor resistivo para baixar os 12 V do sinal da centralina para os 5 V do porto do dsPIC.

5.3.3 Aquisição da Temperatura do Motor

O valor da temperatura do motor é dado por um termístor cerâmico do tipo *Negative Temperature Coefficient* (NTC) que se encontra em contacto com o líquido de refrigeração do motor. A resistência do termístor varia exponencialmente com a temperatura, e é dada pela expressão (4).

$$R(T) = Ae^{BT} \quad (4)$$

Como a característica do termistor não está disponibilizada pelo fabricante, foi necessário determinar a sua característica. Para tal usou-se um forno e um multímetro em modo ohmímetro para registar três valores de resistência a temperaturas diferentes. Utilizando a ferramenta *MatLab* determinou-se então o $A = 36650$ e o $B = -0,03438$. A característica do termistor obtida encontra-se na Figura 19.

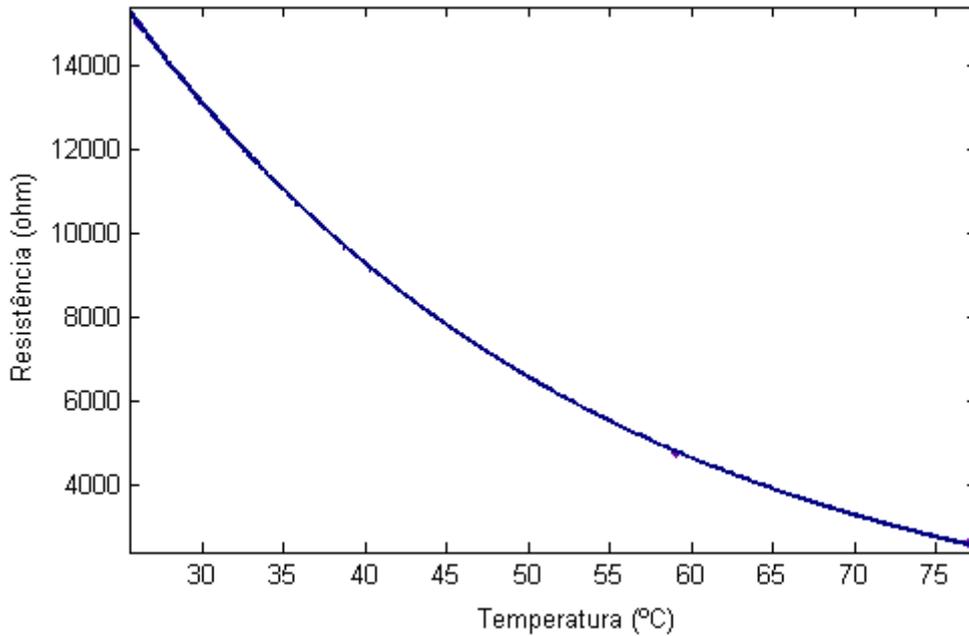


Figura 19- Característica do termistor.

Para podermos medir a variação de tensão com a temperatura utiliza-se o circuito de condicionamento para a termistor representado na Figura 20, que é um divisor resistivo.

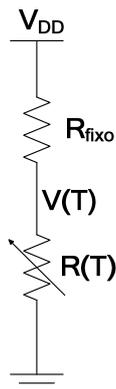


Figura 20 - Circuito condicionamento do termistor.

Pela expressão do divisor resistivo e com a expressão (4), obtemos a expressão (5) que nos dá a tensão em função da temperatura $V(T)$.

$$V(T) = V_{DD} \frac{Ae^{BT}}{Ae^{BT} + R_{fixo}} \quad (5)$$

Manipulando algebricamente a expressão (5), obtemos a expressão (6) que nos dá a temperatura em função da tensão.

$$T(v) = \frac{1}{B} \left[\ln\left(\frac{R_{fixo}}{A}\right) + \ln\left(\frac{V}{V_{DD} - V}\right) \right] \quad (6)$$

A resistência R_{fixo} influencia a gama de valores para os quais a montagem é mais sensível, foi usado uma $R_{fixo} = 3,6 \text{ k}\Omega$, com $V_{DD} = 5 \text{ V}$. Nestas condições a curva da tensão em função da temperatura encontra-se representada na Figura 21.

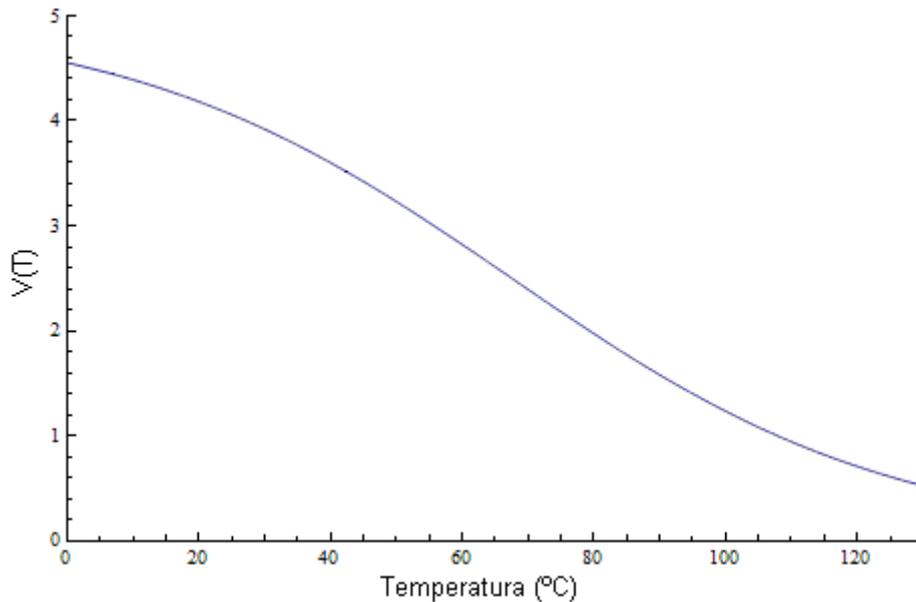


Figura 21 - Tensão em função da Temperatura no termistor.

O valor de $V(T)$ é lido usando o ADC do dsPIC, através da expressão (6) conseguimos então determinar o valor da temperatura correspondente.

Capítulo 6

Medidas e testes

Depois de construído o sistema embebido foram realizadas medidas e testes para verificar o seu funcionamento.

6.1 Consumo

Mediu-se para os vários modos em que o sistema pode operar a corrente de entrada do sistema com a tensão de entrada igual a 12 V, ver Tabela 11. São apresentadas as correntes de entrada para os vários modos seleccionados pelo botão e também a situação em que o periférico não está ligado ao Módulo CAN PIC FST, esta situação dá-nos o consumo do Módulo CAN PIC FST a correr o programa desenvolvido. Através da corrente e tensão de entrada, pela expressão (1) determina-se a potência dissipada pelo sistema.

Tabela 11 - Consumo do sistema embebido.

Modo	Corrente (mA)	Potência (W)
Sem periférico	50	0,60
Reset	60	0,72
Race1 e 2	120 a 210	1,44 a 2,52
Demo	130 a 200	1,80 a 2,40
Normal1 e 2	60	0,72
Shutdown	50	0,60
DemoFST	60 a 90	0,72 a 1,08
Teste	250	3,00

Ao analisar a Tabela 11 verificamos que o Módulo CAN PIC FST tem um consumo de 0,6 W a correr o programa desenvolvido. Como o rendimento do regulador DC-DC nestas condições é 88%, como referido anteriormente, pela expressão (2) obtemos a potência dissipada no circuito do módulo exceptuando o regulador, que é 0,53 W. Como o circuito referido está alimentado a 5 V, esta potência corresponde a uma corrente de 106 mA. Neste circuito os únicos elementos dissipadores de potência são os transceptores de CAN que nesta situação não estão a comunicar, tendo portanto um consumo muito baixo e o dsPIC que está a funcionar a 16 MIPS. O fabricante indica que o dsPIC tem um consumo típico de 74 mA a funcionar a 10 MIPS e de 140 mA a funcionar a 20 MIPS, como nesta situação se encontra a funcionar a 16 MIPS os 106 mA estão de acordo com o indicado pelo fabricante, validando as medidas obtidas.

6.2 Testes

A primeira fase de testes foi efectuada ainda com a versão 0 do *software*. Os testes realizados demonstraram que o sistema adquire os valores esperados e representando-os. Depois de implementado o barramento CAN, este foi testado usando a placa USB CAN e um computador tendo-se verificado a correcta recepção e representação dos dados recebidos.

O sistema foi também testado no exterior para poder avaliar o seu desempenho em condições reais de um dia de sol. Com os reguladores na posição de maior intensidade luminosa obtêm-se os resultados apresentados na Tabela 12, onde se verifica que a solução dos LEDs é uma boa solução, no entanto os display de 7 segmentos apesar de terem sido escolhidos os que apresentava a maior intensidade luminosa acabam por não ter a legibilidade desejada.

Tabela 12 - Legibilidade Painel Volante na presença de muita luz exterior.

	Barra 12 LEDs	LEDs Aviso	LEDs Azuis	Display Primário	Display Secundário
Legibilidade	Boa	Boa	Boa	Suficiente	Fraca

Realizou-se também um teste durante a noite para avaliar na condição de ausência de luz exterior, neste caso é necessário baixar a intensidade luminosa utilizando os reguladores dado que na posição de maior intensidade o Painel Volante apresenta uma luminosidade que provoca o encandeamento. Regulado a intensidade consegue-se obter então uma boa legibilidade em todos os casos sem provocar o encandeamento.

Ao longo dos testes verificou-se que a força necessária para mudar a posição do botão de selecção é muito pouca e que as posições estão demasiado próximas entre si. Este facto é uma desvantagem porque durante a corrida o piloto usa luvas.

Visitando o endereço de internet <http://www.youtube.com/watch?v=XdgrURN7sgg> é possível visualizar um vídeo do funcionamento do sistema embebido desenvolvido.

Capítulo 7

Conclusões e Trabalho Futuro

O desenvolvimento deste trabalho iniciou-se com a definição das especificações para o sistema embebido. Neste ponto as principais limitações foram a definição de que funções das utilizadas no mundo da competição podia ser aplicadas no carro da FST tendo em conta as suas limitações mecânicas.

Tendo em vista a concretização das especificações elaborou-se a arquitectura do sistema, os esquemas eléctricos e as máscaras de PCB. Ao longo desta fase adquiriu-se conhecimentos em especial no uso da ferramenta *Altium Protel DXP* e também no processo de análise e selecção de componentes. Existiu também a preocupação com a área e a compatibilidade geométrica com os sistemas adjacentes. Realizaram-se estimativas de consumo para verificar os limites máximos de potência em vários pontos de sistema embebido.

A fase seguinte foi a produção dos protótipos. Esta fase do trabalho levou à utilização e compreensão das ferramentas do laboratório de circuitos impressos do IST-Taguspark, em especial a fresadora CNC e a técnica de produção por furo metalizado.

Foi realizada a estimativa de custo da fase de desenvolvimento e da produção dos sistemas projectados, e a conclusão a retirar é que o custo de desenvolvimento é muito superior ao custo de produção.

O *software* necessário para o sistema foi realizado numa primeira fase para adquirir os dados dos sensores directamente do carro e posteriormente implementado o protocolo CAN como definido nas especificações.

Os testes realizados ao sistema comprovaram o seu funcionamento, mas permitiram também identificar os pontos fracos. Conclui-se que o botão de selecção utilizado não é o mais adequado para esta aplicação, pois tem as diferentes posições demasiado próximas sendo necessário aplicar pouca força para este comutar entre as posições. Esta característica torna o botão difícil de manusear com as luvas que o piloto utiliza. No futuro deverá substituir-se o botão por outro para solucionar este problema.

A utilização dos *LEDs* de alto brilho revelou-se uma boa solução. No entanto, os *displays* de 7 segmentos, especialmente no caso do *display* secundário, não garantem uma boa legibilidade em condições de muita luz exterior. No futuro opção de substituir os *displays* de 7 segmentos por um *display* LCD adequado permitirá aumentar a legibilidade e também as capacidades de representar informação. Nas condições de ausência de luz exterior, como é o caso da utilização durante a noite, o problema do encandeamento é solucionado pelos reguladores de intensidade luminosa presentes no painel. O sistema desenvolvido permite a regulação da intensidade luminosa como já foi referido, mas de uma forma manual, ou seja o utilizador têm que ajustar o regulador para a intensidade desejada. Será uma boa evolução inserir um sensor de luz para medir a luz exterior a que o sistema está sujeito e fazer o ajuste automático da intensidade luminosa do painel, conseguindo assim um Painel Volante que se adapta à luz exterior.

Bibliografia

- [1] Formula Student, <http://www.formulastudent.com>, Acesso em Outubro de 2009.
- [2] Projecto FST, <http://www.dem.ist.utl.pt/~fst/big/index.php>, Acesso em Outubro de 2009.
- [3] Olhares, http://olhares.aeiou.pt/fst-03_no_12o_aniversario_do_kip_foto3111804.html, Acesso em Outubro de 2009.
- [4] Dr Peter Thoma, "Future Needs for Automotive Electronics", *IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pp. 532-539, October 10-12, 1994.
- [5] BOSCH, "CAN Specification version 2.0", Robert Bosch GmbH (1991).
- [6] Mr. Anthony Slack, "Prospective Developments in Automotive Instrumentation", *Electronics Manufacturing Technology Symposium*, pp. 10-15, 27-29 Abril, 1998.
- [7] Farina, J.; Rodriguez-Andina, J.J.; Doval, "Design and Evaluation of a New Kind of Remote Control Switches for Automotive Applications", *IEEE International Symposium on Industrial Electronics*, Vol. 1, pp. 347-351, 4-7 May, 2004.
- [8] Tong-Jin Park, Chang-soo Han, Sang-Ho Lee, "Development of the electronic control unit for the rack-actuating steer-by-wire using the hardware-in-the-loop simulation system", *Mechatronics*, Vol. 15, pp.899-918, Outubro, 2005.
- [9] L. Cocco, P.Gazzola, S.Rapuano, L. Rossi, "Vibrations Measurements of Formula One Car Electronic Devices", IMEKO, Florence, Italy, 2008.
- [10] SuperLicença, <http://www.superlicenca.com.br/0,0,314,9551,00.html>, Acesso em Outubro de 2009.
- [11] F1technical, <http://f1technical.net/articles/30>, Acesso em Outubro de 2009.
- [12] MotorSPOT, <http://www.motorspotstore.com/formulaquadro.html>, Acesso em Outubro de 2009.
- [13] MICROCHIP, "dsPIC30F6011A/6012A/6013A/6012A Data Sheet, High-Performance, 16-Bit, Digital Signal Controllers", 2008.

- [14] NACIONAL SEMICONDUCTOR, "LM2673 SIMPLE SWITCHER 3A Step-Down Voltage Regulator with Adjustable Current Limit", Abril 2005.
- [15] LPKFMinicontac, http://www.lpkfusa.com/RapidPCB/ThroughHolePlating/minicontac_rs.htm
Acesso em Outubro de 2009.
- [16] LPKFProtoMat, <http://www.lpkfusa.com/protomat/s62.htm>, Acesso em Outubro de 2009.
- [17] MAXIM, "MAX7219/MAX7221 serial interfaced, 8-digit LED Display Drivers", Julho 2003.
- [18] MICROCHIP, "MCP2551 High-Speed CAN Transceptor", 2003.
- [19] KINGBRIGH, "SC-52-11SRWA Super Brigh Red", Novembro 2006.
- [20] MICROCHIP, "MCP2515 Stand-Alone CAN Controller With SPI Interface", 2007.
- [21] MPLABIDE, http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469, Acesso em Outubro de 2009.

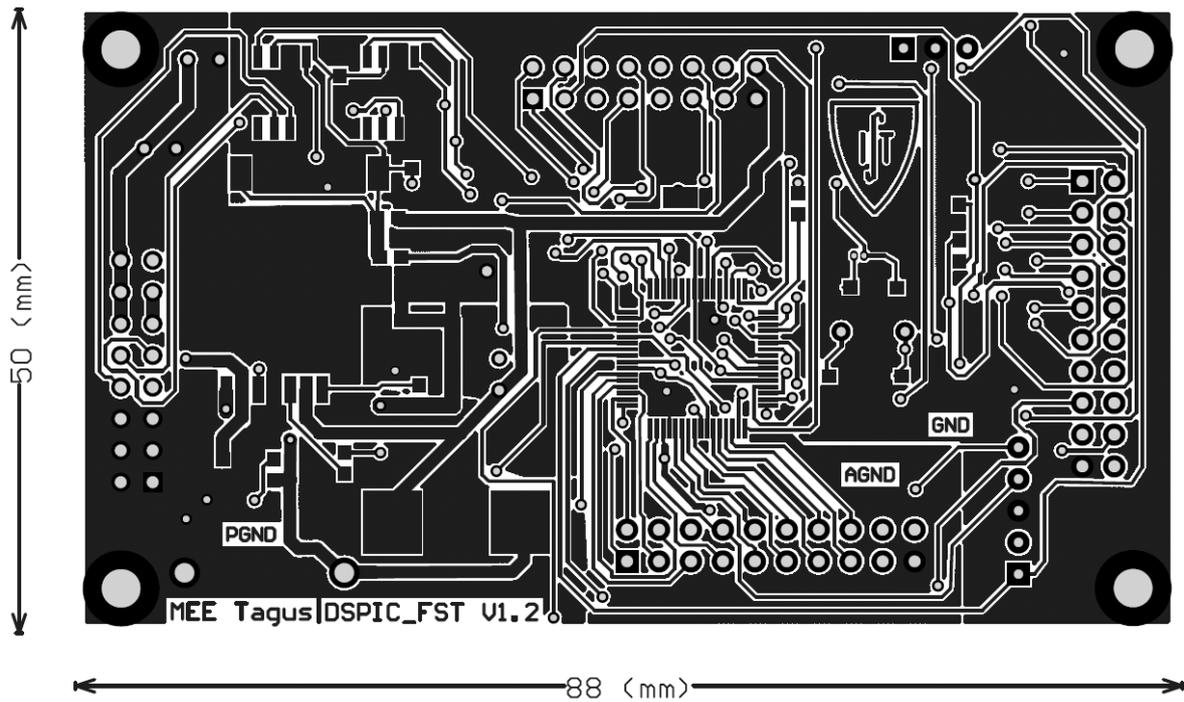


Figura 23 - PCB do Módulo CAN PIC FST, *Top Layer*.

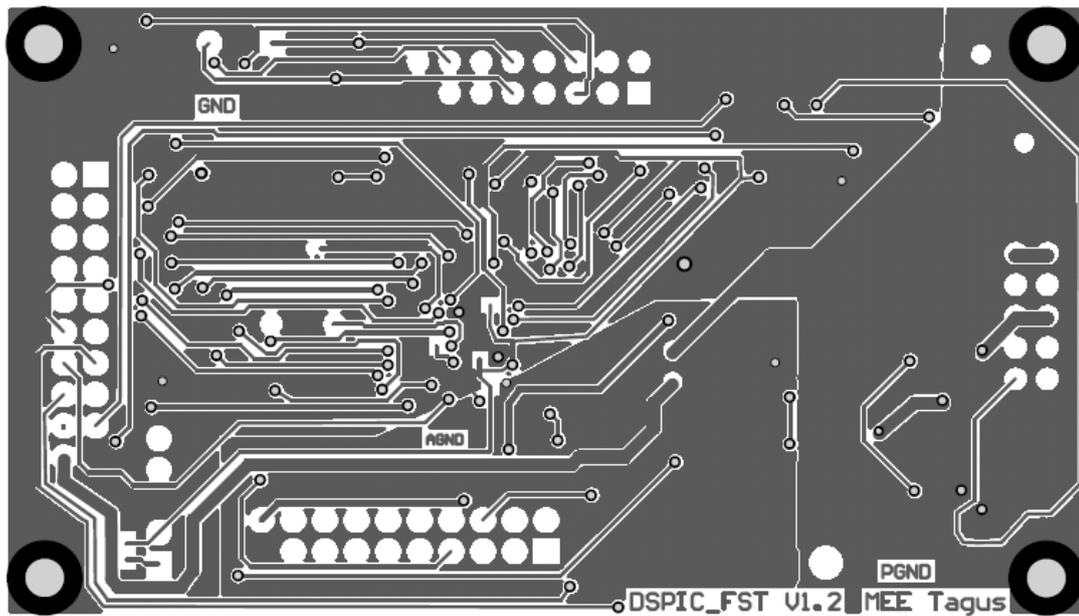


Figura 24 - PCB do Módulo CAN PIC FST, *Bottom Layer*.

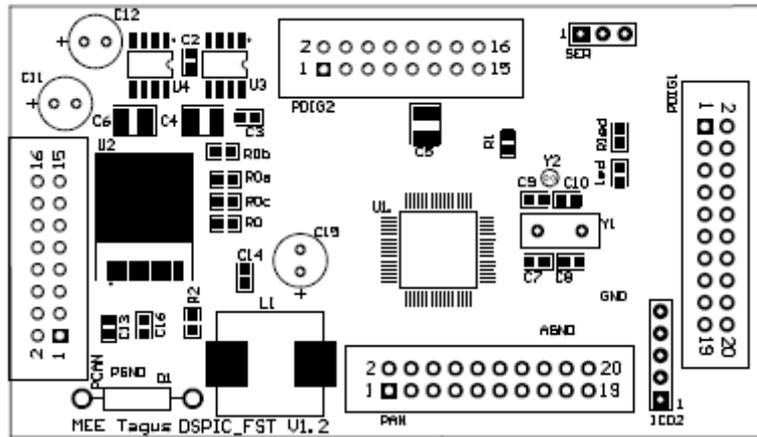


Figura 25 - Componentes do Módulo CAN PIC FST, *Top Layer*.

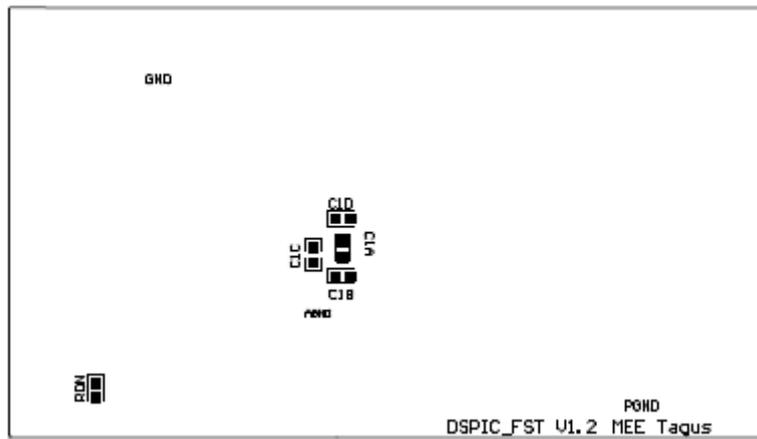
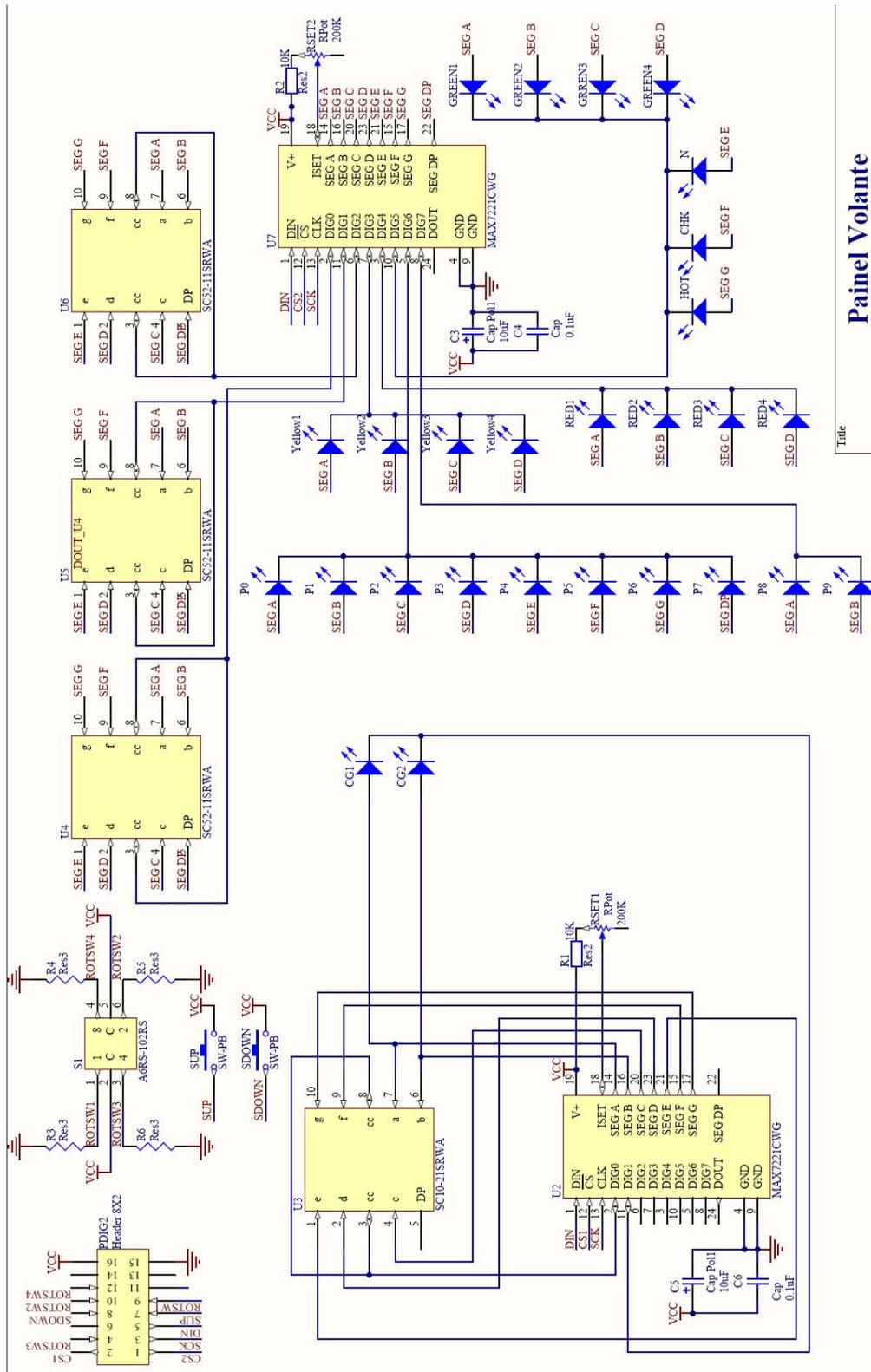


Figura 26 - Componentes do Módulo CAN PIC FST, *Bottom Layer*.

Tabela 13 - Lista componentes do Módulo CAN PIC FST.

Designação	Comentário	Descrição
C1A	Cap 100nF, 0603	Capacitor 100nF SMD 0603
C1B	Cap 100nF, 0603	Capacitor 100nF SMD 0603
C1C	Cap 100nF, 0603	Capacitor 100nF SMD 0603
C1D	Cap 100nF, 0603	Capacitor 100nF SMD 0603
C2	Cap 100nF, 0603	Capacitor 100nF SMD 0603
C3	Cap 100nF, 0603	Capacitor 100nF SMD 0603
C4	Cap 10uF, 1210	Capacitor 10uF SMD 1210
C5	Cap 10uF, 1210	Capacitor 10uF SMD 1210
C6	Cap 10uF, 1210	Capacitor 10uF SMD 1210
C7	Cap 15pF, 0603	Capacitor 15pF SMD 0603
C8	Cap 15pF, 0603	Capacitor 15pF SMD 0603
C9	Cap 18pF, 0603	Capacitor 18pF SMD 0603
C10	Cap 18pF, 0603	Capacitor 18pF SMD 0603
C11	Cap Pol 15uF	Polarized Capacitor 15uF
C12	Cap Pol 15uF	Polarized Capacitor 15uF
C13	Cap 470nF, 0603	Capacitor 470nF SMD 0603
C14	Cap 100nF, 0603	Capacitor 100nF SMD 0603
C15	Cap Pol 180uF	Polarized Capacitor 180uF
C16	Cap 10nF, 0603	Capacitor 10nF SMD 0603
D1	SB350	Schottky Diode
ICD2	Socket 5-Pin	Programming Port
L1	Epcos B82477-G433-M	Inductor
LED	LED, 0603	LED SMD 0603
PAN	IDC Connector 20-Pin	Analog Port
PCAN	IDC Connector 16-Pin	Power and CANBUS
PDIG1	IDC Connector 20-Pin	Digital Port1
PDIG2	IDC Connector 16-Pin	Digital Port2
R0	Res 0k, 0603	Resistor 0K SMD 0603
R0a	Res 0k, 0603	Resistor 0K SMD 0603
R0b	Res 0k, 0603	Resistor 0K SMD 0603
R0c	Res 0k, 0603	Resistor 0K SMD 0603
R1	Res 100K, 0603	Resistor 100K SMD 0603
R2	Res 8K, 0603	Resistor 8K SMD 0603
Rled	Res 10K, 0603	Resistor 10K SMD 0603
RON	Res 10K, 0603	Resistor 10K SMD 0603
SER	Socket 3-Pin	Serial Port
U1	dsPIC30F6012A	dsPIC30F6012A-30I/PT
U2	LM2673-5.0	DC Voltage Regulator
U3	MCP2551-E/SN	High Speed CAN Transceiver, 8-Pin Plastic SOIC, Extended Grade
U4	MCP2551-E/SN	High Speed CAN Transceiver, 8-Pin Plastic SOIC, Extended Grade
Y1	16 MHz	Crystal Oscillator 16 MHz
Y2	32.768 KHz	Crystal Oscillator 32.768 KHz



Panel Volante

Figura 27 - Esquema eléctrico Panel Volante.

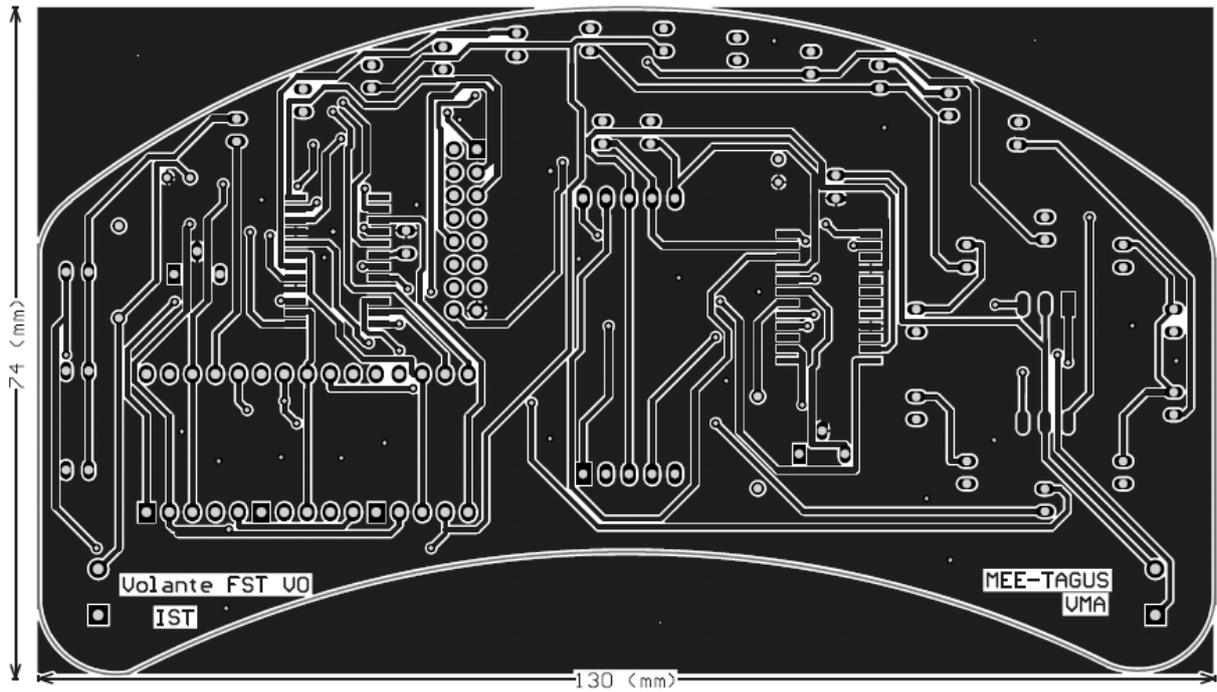


Figura 28 - PCB do Painel Volante, *Top Layer*.

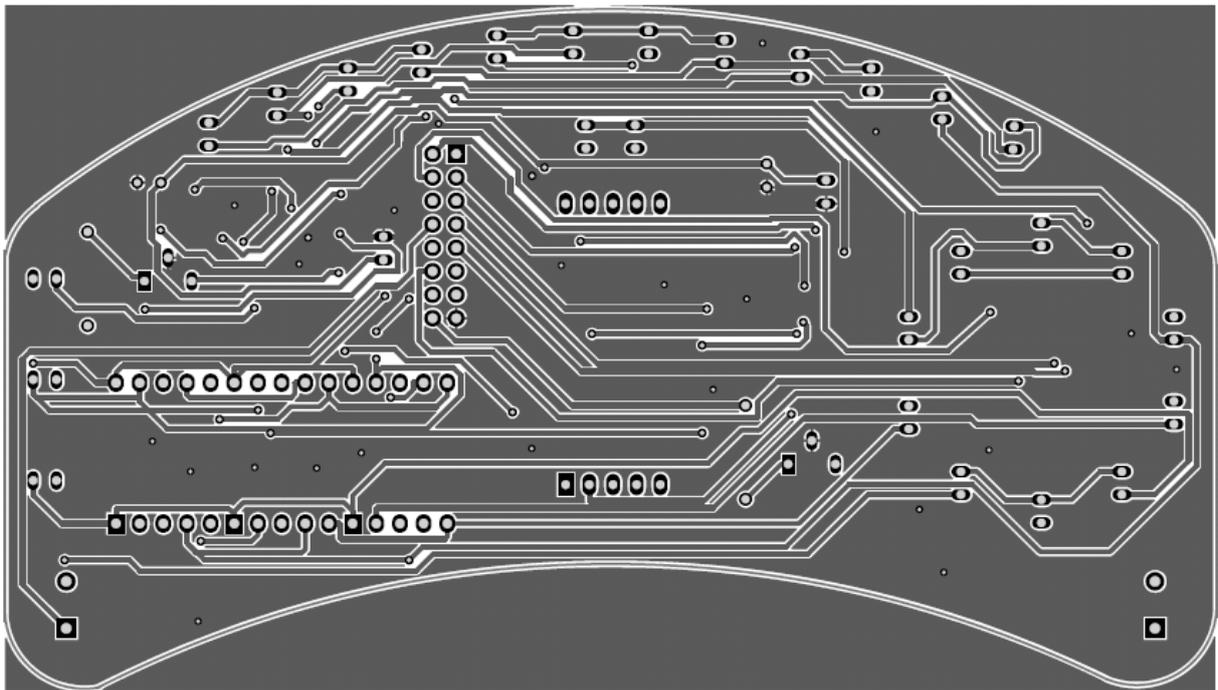


Figura 29 - PCB do Painel Volante, *Bottom Layer*.

Tabela 14 - Lista componentes Painel Volante

Designator	Comment	Description
C3	Cap Pol 10uF	Polarized Capacitor 10uF
C4	Cap 100nF	Capacitor 100nF
C5	Cap Pol 10uF	Polarized Capacitor 10uF
C6	Cap 100nF	Capacitor 100nF
CG1	LED Blue 5mm	LED High Brightness Blue 5mm
CG2	LED Blue 5mm	LED High Brightness Blue 5mm
CHK	LED Yellow 5mm	LED High Brightness Yellow 5mm
GREEN1	LED Green 5mm	LED High Brightness Green 5mm
GREEN2	LED Green 5mm	LED High Brightness Green 5mm
GREEN4	LED Green 5mm	LED High Brightness Green 5mm
GRREN3	LED Green 5mm	LED High Brightness Green 5mm
HOT	LED Red 5mm	LED High Brightness Red 5mm
N	LED Green 5mm	LED High Brightness Green 5mm
P0	LED Blue 5mm	LED High Brightness Blue 5mm
P1	LED Green 5mm	LED High Brightness Green 5mm
P2	LED Green 5mm	LED High Brightness Green 5mm
P3	LED Green 5mm	LED High Brightness Green 5mm
P4	LED Red 5mm	LED High Brightness Red 5mm
P5	LED Red 5mm	LED High Brightness Red 5mm
P6	LED Red 5mm	LED High Brightness Red 5mm
P7	LED Yellow 5mm	LED High Brightness Yellow 5mm
P8	LED Yellow 5mm	LED High Brightness Yellow 5mm
P9	LED Yellow 5mm	LED High Brightness Yellow 5mm
PDIG2	IDC Connector 16-Pin	Digital Port2
R1	Res 9.53k	Resistor 9.53k
R2	Res 9.53k	Resistor 9.53k
R3	Res 10k, 0603	Resistor 10k SMD 0603
R4	Res 10k, 0603	Resistor 10k SMD 0603
R5	Res 10k, 0603	Resistor 10k SMD 0603
R6	Res 10k, 0603	Resistor 10k SMD 0603
RED1	LED Red 5mm	LED High Brightness Red 5mm
RED2	LED Red 5mm	LED High Brightness Red 5mm
RED3	LED Red 5mm	LED High Brightness Red 5mm
RED4	LED Red 5mm	LED High Brightness Red 5mm
RSET1	Potenciometer 200k	BOURNS TRIMMER, 200K
RSET2	Potenciometer 200k	BOURNS TRIMMER, 200K
S1	A6RS-102RS	OMRON Rotary DIP Switch
SDOWN	SW-PB	Switch
SUP	SW-PB	Switch
U2	MAX7221CWG	Maxim Serially Interfaced, 8-Digit LED Display Driver
U3	SC10-21SRWA	Kingbrigh Led Display Red 25.4mm
U4	SC52-11SRWA	Kingbrigh Led Display Red 13.2mm
U5	SC52-11SRWA	Kingbrigh Led Display Red 13.2mm
U6	SC52-11SRWA	Kingbrigh Led Display Red 13.2mm
U7	MAX7221CWG	Maxim Serially Interfaced, 8-Digit LED Display Driver
Yellow1	LED Yellow 5mm	LED High Brightness Yellow 5mm
Yellow2	LED Yellow 5mm	LED High Brightness Yellow 5mm
Yellow3	LED Yellow 5mm	LED High Brightness Yellow 5mm
Yellow4	LED Yellow 5mm	LED High Brightness Yellow 5mm

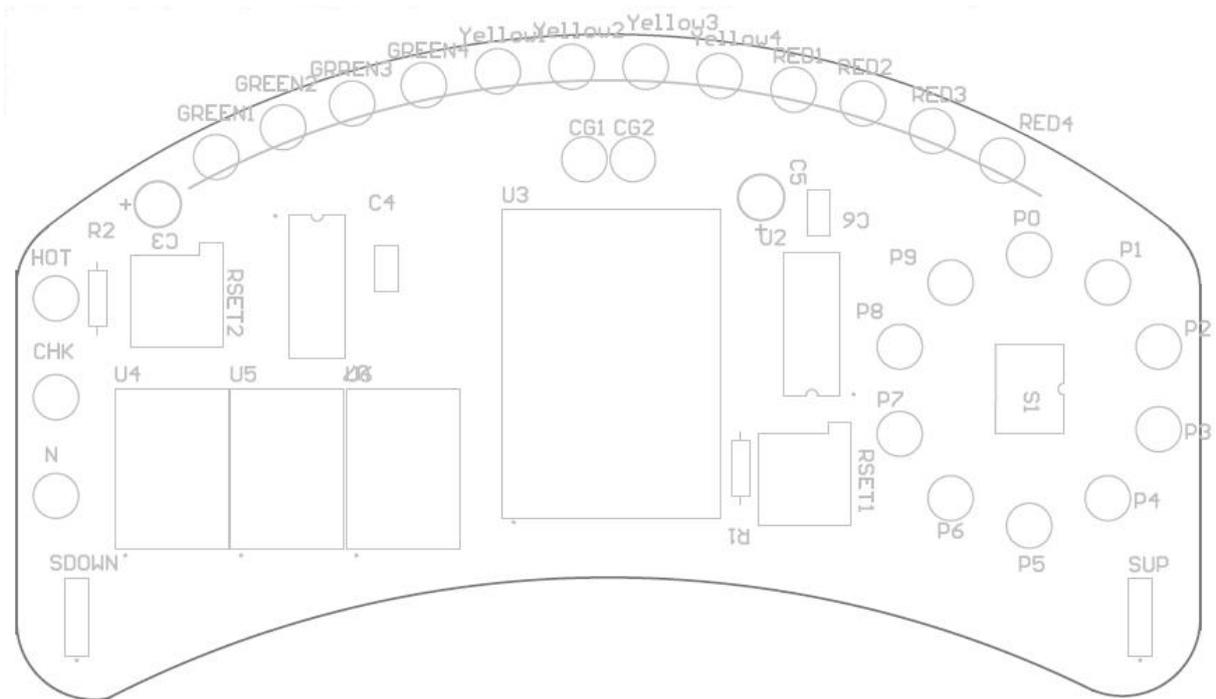


Figura 30 - Componentes do Painel Volante, Top Layer.

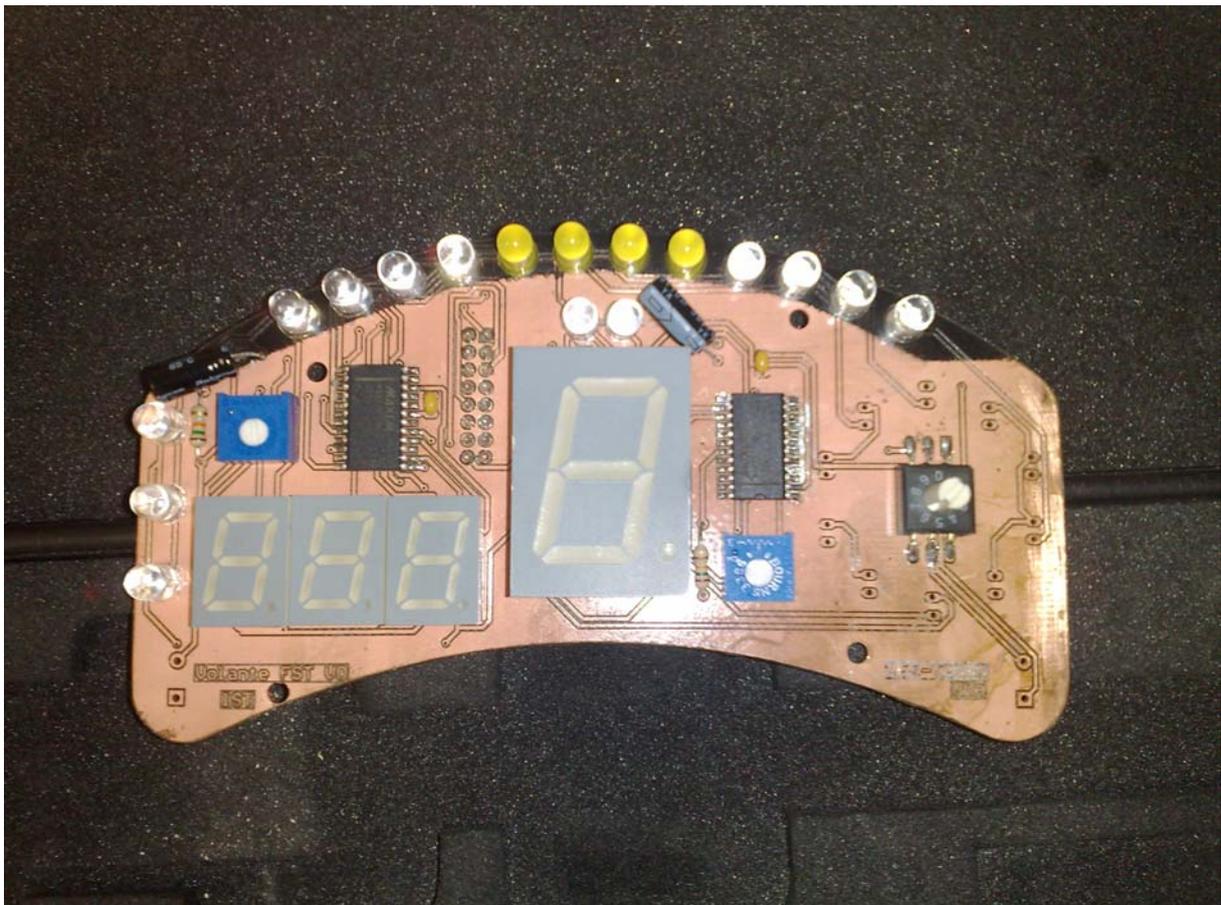


Figura 31 - Foto do Painel Volante.

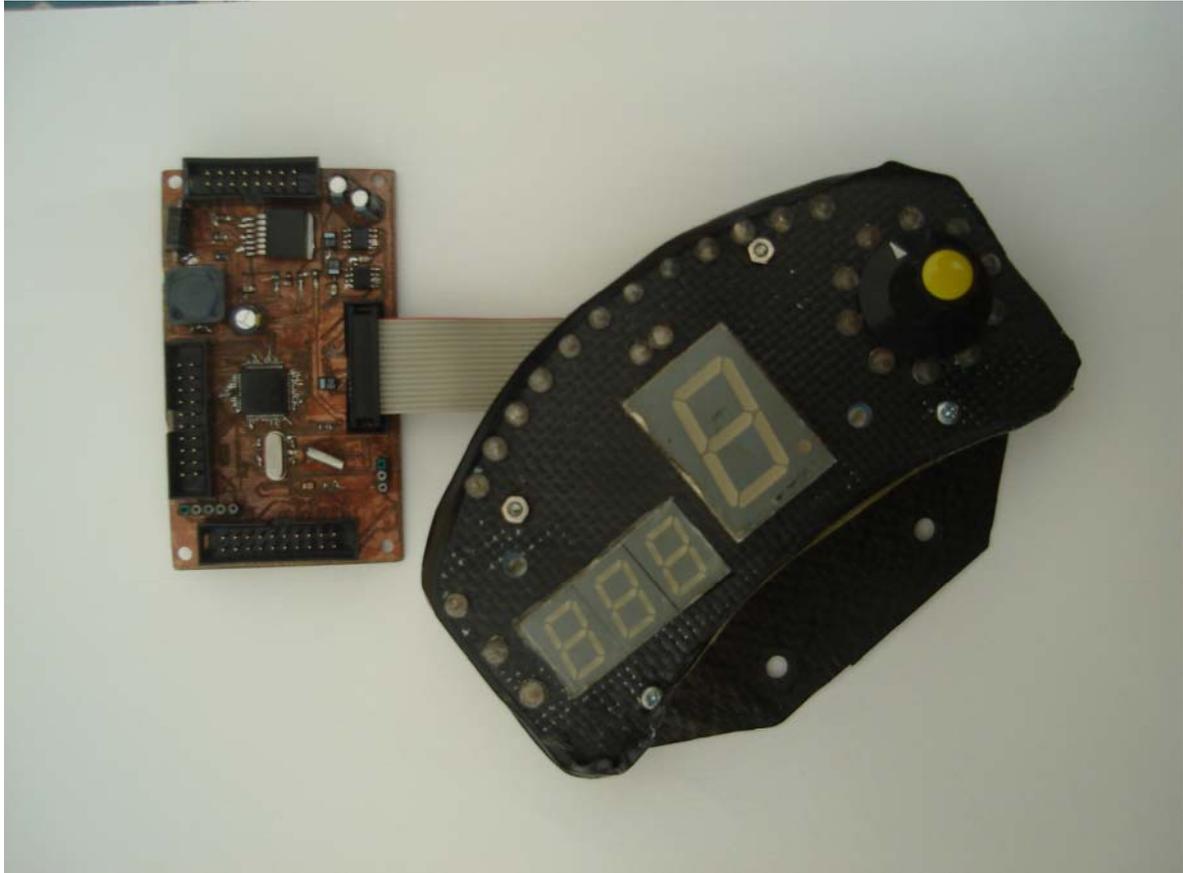


Figura 32 - Foto do sistema, Modulo CAN PIC FST + Painel Volante.



Figura 33 - Foto do Painel Volante no modo teste.

Anexo 2 – Software

```
int main (void)
{

//configurar Portos
_TRISD11=0;          /*Setting RD11 as output bit, led da placa CAN_PIC_FST*/
_RC1=1;              //garantir que ã faz enable, cs de um MAX7221
_RC2=1;              //garantir que ã faz enable, cs do outro MAX7221
_TRISC1=0;           /*Setting RC1 as output bit, enable de um MAX7221*/
_TRISC2=0;           /*Setting RC2 as output bit, enable do outro MAX7221*/
_TRISF4=1;           //Setting RF4 as input bit, peso1 do botão
_TRISF5=1;           //Setting RF5 as input bit, peso8 do botão
_TRISG13=1;          //Setting RG13 as input bit, peso4 do botão
_TRISG15=1;          //Setting RG15 as input bit, peso2 do botão

_TRISD0=0;

    timer2_init(); //iniciar timer 2

    init_interrupt( ); //iniciar interrupções para o can
    init_can1( ); //init_can2( ); //inicia can
    config_recebe_can1( ); //config_recebe_can2( ); //prepara para receber
    config_envia_can1( 231, 4, 0, 0, 0, 0 ); //envia mensagem pelo CAN SID 231 para
    assinalar inicio do volante

//configurar SPI
OpenSPI1(FRAME_ENABLE_OFF&FRAME_SYNC_OUTPUT&ENABLE_SDO_PIN&SPI
I_MODE16_ON&SPI_SMP_ON&SPI_CKE_OFF&SLAVE_ENABLE_OFF&CLK_POL_A
CTIVE_LOW&MASTER_ENABLE_ON&SEC_PRESCAL_1_1&PRI_PRESCAL_4_1,
SPI_ENABLE&SPI_IDLE_STOP&SPI_RX_OVFLOW_CLR);

//configurar driver_1 Max7221
write_MAXIM(Test_mode_normal,1); //
Reset_MAXIM(1); //fazer reset para apagar todos os
leds
write_MAXIM(Decode_mode_1,1); //Configurar os digitos que se querem em decode
mode ou não
write_MAXIM(Intensity_1,1); //definir intesidade
write_MAXIM(Scan_Limit_1,1); //definir quantos digitos percorre
write_MAXIM(Normal_mode,1); //colocar no modo de funcionamento normal
```

```

//configurar driver_2 Max7221
write_MAXIM(Test_mode_normal,2); //
Reset_MAXIM(2); //fazer reset para apagar todos os
leds
write_MAXIM(Decode_mode_2,2); //Configurar os digitos que se querem em decode
mode ou não
write_MAXIM(Intensity_2,2); //definir intesidade
write_MAXIM(Scan_Limit_2,2); //definir quantos digitos percorre
write_MAXIM(Normal_mode,2); //colocar no modo de funcionamento
normal

```

```
while(1){
```

```
Posicao_Rotswt=(_RF5*8)+(_RG13*4)+(_RG14*2)+_RF4; //lê a posição do botão
```

```
switch(Posicao_Rotswt){ //decide qual a função seleccionada pelo botão
```

```

    case 0: //faz reset aos drivers
        Actualiza_digito(0x40,dig6_1,Mask_Posicoes0_7,1);
        Actualiza_digito(0x0,dig7_1,Mask_Posicoes8_9,1); //assinalar posição botão

```

```

        //configurar driver_1 Max7221
        write_MAXIM(Test_mode_normal,1); //
        Reset_MAXIM(1); //fazer reset para apagar todos os leds
        write_MAXIM(Decode_mode_1,1); //Configurar os digitos que se
        querem em decode mode ou não
        write_MAXIM(Intensity_1,1); //definir intesidade
        write_MAXIM(Scan_Limit_1,1); //definir quantos digitos percorre
        write_MAXIM(Normal_mode,1); //colocar no modo de
        funcionamento normal

```

```

        //configurar driver_2 Max7221
        write_MAXIM(Test_mode_normal,2); //
        Reset_MAXIM(2); //fazer reset para apagar todos os leds
        write_MAXIM(Decode_mode_2,2); //Configurar os digitos que se
        querem em decode mode ou não
        write_MAXIM(Intensity_2,2); //definir intesidade
        write_MAXIM(Scan_Limit_2,2); //definir quantos digitos percorre
        write_MAXIM(Normal_mode,2); //colocar no modo de
        funcionamento normal

```

```
break;
```

```

    case 1: //Mostra informações RPMs e Temperatura
        Actualiza_digito(0x20,dig6_1,Mask_Posicoes0_7,1);

```

```

    Actualiza_digito(0x0,dig7_1,Mask_Posicoes8_9,1);//assinalar posição botão

    Trata_RPM(RPM);
    Gear(GEAR);
    Verifica_CHK();
    Verifica_N();
    Verifica_Hot(Temp);

    count++;

    if(count==5000){
    count=0;
        Tempaux=Temp;
        if(Tempaux<30) Tempaux=1000;
        Numero_Displays(Tempaux);
    }

    break;

    case 2:                //mostra info RPM e SPEED
    Actualiza_digito(0x10,dig6_1,Mask_Posicoes0_7,1);
    Actualiza_digito(0x0,dig7_1,Mask_Posicoes8_9,1);//assinalar posição botão

    Trata_RPM(RPM);
    Gear(GEAR);
    Verifica_CHK();
    Verifica_N();
    Verifica_Hot(Temp);

    count1++;
    if(count1==1000){
    count1=0;
    Numero_Displays(SPEED);
    }

    break;

    case 3:                //Demosntração RPMs e Temperatura
    Actualiza_digito(0x08,dig6_1,Mask_Posicoes0_7,1);
    Actualiza_digito(0x0,dig7_1,Mask_Posicoes8_9,1);//assinalar posição botão

    RPMdem++;
    if(RPMdem>13500) RPMdem=0;
    Trata_RPM(RPMdem);

    Geardem++;

```

```

if((Geardem>>12)>6) Geardem=0;
Gear(Geardem>>12);

Tempdem++;
if((Tempdem>>8)>140) Tempdem=0;
Numero_Displays(Tempdem>>8);
Verifica_Hot(Tempdem>>8);

//Verifica_CHK();
//Verifica_N();

break;

case 4: //coloca drivers no mode funcionamento normal
Atualiza_digito(0x04,dig6_1,Mask_Posicoes0_7,1);
Atualiza_digito(0x0,dig7_1,Mask_Posicoes8_9,1); //assinalar posição botão

write_MAXIM(Normal_mode,1);
write_MAXIM(Normal_mode,2);

break;

case 5: //coloca drivers desligadas
Atualiza_digito(0x02,dig6_1,Mask_Posicoes0_7,1);
Atualiza_digito(0x0,dig7_1,Mask_Posicoes8_9,1); //assinalar posição botão

write_MAXIM(Shutdown_mode,1);
write_MAXIM(Shutdown_mode,2);

break;

case 6: //coloca drivers no mode funcionamento normal
Atualiza_digito(0x01,dig6_1,Mask_Posicoes0_7,1);
Atualiza_digito(0x0,dig7_1,Mask_Posicoes8_9,1); //assinalar posição botão

write_MAXIM(Normal_mode,1);
write_MAXIM(Normal_mode,2);

break;

case 7: //Faz passar FST no display central e envia mensagem de CAN
Reset_MAXIM(1);
Reset_MAXIM(2);

Atualiza_digito(0x80,dig6_1,Mask_Posicoes0_7,1);
Atualiza_digito(0x0,dig7_1,Mask_Posicoes8_9,1); //assinalar posição botão

```

```
//config_envia_can1( 231, 4, 7, 0, 0, 0 ); //envia mensagen pelo CAN SID 231
para assinalar posição7
```

```
__delay32(delay_fst);
Atualiza_digito(0x47,dig0_2,Mask_Display_no_decode,2);
__delay32(delay_fst);
Atualiza_digito(0x00,dig0_2,Mask_Display_no_decode,2);
__delay32(delay_fst);
Atualiza_digito(0x5b,dig0_2,Mask_Display_no_decode,2);
__delay32(delay_fst);
Atualiza_digito(0x00,dig0_2,Mask_Display_no_decode,2);
__delay32(delay_fst);
Atualiza_digito(0x0f,dig0_2,Mask_Display_no_decode,2);
__delay32(delay_fst);
Atualiza_digito(0x00,dig0_2,Mask_Display_no_decode,2);
__delay32(delay_fst);
```

```
break;
```

```
case 8: //desliga test_mode
Atualiza_digito(0x40,dig7_1,Mask_Posicoes8_9,1);
Atualiza_digito(0x0,dig6_1,Mask_Posicoes0_7,1); //assinalar posição botão
```

```
write_MAXIM(Test_mode_normal,1);
write_MAXIM(Test_mode_normal,2);
```

```
break;
```

```
case 9: //Mete em test_mode
Atualiza_digito(0x20,dig7_1,Mask_Posicoes8_9,1);
Atualiza_digito(0x0,dig6_1,Mask_Posicoes0_7,1); //assinalar posição botão
```

```
write_MAXIM(Test_mode_test,1);
write_MAXIM(Test_mode_test,2);
```

```
break;
```

```
    }
}
return 0;
```

```
}
```

```

/*-----
* Função: void Numero_Displays( unsigned int numero)
*
*           Função que representa um inteiro entre 0 e 999 nos displays em decode mode
*           Caso numero maior que 999 mete ---
*           O parametro numero tem que ser um inteiro
*           Faz também o aviso de RPM para troca mudança
*           Atenção ver se está em decode ou não! tabela no datacheet do maxim7221
*-----*/
void Numero_Displays( unsigned int numero){

    unsigned int Dados_dig0_1;
    unsigned int Dados_dig1_1;
    unsigned int Dados_dig2_1;
    unsigned int aux;

    if(numero>999){ // Caso numero maior que 999 mete ---
        Dados_dig0_1=10;
        Dados_dig1_1=10;
        Dados_dig2_1=10;

        }else{
            aux=floor(numero/10);
            Dados_dig0_1=floor(aux/10); //determina 1ºdigito
            if(Dados_dig0_1==0)Dados_dig0_1=0xf; //se dig + a direita é 0 apaga-o
            Dados_dig1_1=(aux%10); //determina 2ºdigito
            if((Dados_dig1_1==0)&&(Dados_dig0_1==0xf)) Dados_dig1_1=0xf; //se dig + a
direita esta apagado e do meio é 0 apaga-o tb
            Dados_dig2_1=(numero%10); //determina 3ºdigito
        }

    Actualiza_digito(Dados_dig0_1,dig0_1,Mask_Display_decode,1); //mete dados no Driver
max7221
    Actualiza_digito(Dados_dig1_1,dig1_1,Mask_Display_decode,1);
    Actualiza_digito(Dados_dig2_1,dig2_1,Mask_Display_decode,1);
}

/*-----
* Função: write_MAXIM(unsigned int data_SPI, int driver)
*
*           Para escrever nos drivers MAXIM7221
*           data_SPI são os dados a enviar
*           driver é para escolher o driver: 2 para Gear e RPM_ideal, 1 para RPMs 3 displays ...
*-----*/
void write_MAXIM(unsigned int data_SPI, int driver){

    if(driver==1){

```

```
SPI1STATbits.SPIROV = 0;      // Reset overflow bit
_RC1=0;                       //fazer chipselect
WriteSPI1(data_SPI);         //função write do SPI(tem alterações à original,forum microchip)
_RC1=1;                       //fazer chipselect
}
```

```
if(driver==2){
SPI1STATbits.SPIROV = 0;      // Reset overflow bit
_RC2=0;                       //fazer chipselect
WriteSPI1(data_SPI);         //função write do SPI
_RC2=1;                       //fazer chipselect
}
}
```