



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Sistemas Electrónicos para o ProjectoFST

Rui Alexandre de Sousa Andrade

Dissertação para obtenção do Grau de Mestre em **Engenharia
Electrotécnica e de Computadores**

Júri

Presidente: Prof. Doutor Marcelino Bicho dos Santos

Orientador: Prof. Doutor Moisés Simões Piedade

Co-orientador: Prof. Doutor Leonel Augusto Pires Seabra de Sousa

Vogal: Prof. Doutor Francisco André Corrêa Alegria

Outubro 2009

Resumo

A presente dissertação tem por objectivo descrever a arquitectura dos sistemas electrónicos que estão a ser desenvolvidos para o Projecto Formula Student Técnico (ProjectoFST). Apresenta-se também o desenvolvimento de alguns sensores desenvolvidos para utilização em veículos motorizados de competição e do sistema de aquisição de dados projectado para adquirir os dados gerados por esses mesmos sensores. As decisões de projecto tomadas têm em especial atenção os custos e a possibilidade de implementação e teste dos sistemas no veículo em tempo útil, sendo ainda dada prioridade a sensores cujos dados sejam realmente úteis para avaliação do desempenho do veículo.

Os Sistemas Electrónicos do ProjectoFST incluem o sistema de aquisição de dados baseado em três módulos, colocados um em cada zona do veículo, que adquirem os sinais dos sensores vizinhos e os comunicam num barramento CAN para que possam ser armazenados por um outro sistema para futura análise. Os módulos da frente e de trás adquirem dados de sensores de deslocamento dos amortecedores, temperatura à superfície dos pneus em dois locais distintos e velocidade da roda enquanto o módulo central adquire sinais quantidade de acelerador, ângulo de viragem do volante, revoluções por minuto do motor e temperatura do líquido de refrigeração do motor.

O sistema completo foi implementado no protótipo FST-03 e testado em pista e os dados adquiridos são reveladores do seu bom funcionamento e da mais-valia que é já para o ProjectoFST, a equipa de *Formula Student* do IST, que vê assim, 8 anos depois da sua criação, finalmente implementado num dos seus veículos um sistema de aquisição de dados.

Abstract

The purpose of the current thesis is to describe the architecture of the electronic systems being developed for Proyecto Formula Student Técnico (ProjectoFST). It introduces the operation of a group of sensors developed to be implemented in motorsports vehicles and of a sensor network, connected by a bus, designed to acquire the signals produced by those sensors. Project decisions and choices have in consideration cost and feasibility issues, meaning that paths were taken so that the systems could be simple enough to be implemented and tested in a running vehicle in a short period of time. Sensors were also given priorities so more important sensors were developed first, even if they were simpler.

ProjectoFST's electronic system include the data acquisition system based in three developed data acquisition modules, one in each area of the car, that acquire the data from the surrounding sensors and communicate that data to a CAN bus so that another system, subject of a different thesis, can store it in a flash memory for future analysis. The front and rear data acquisition modules collect data from dumper displacement sensors, tire surface temperature at different latitudes and wheel speed, while the middle module acquires data from the throttle position sensor, steering angle sensor, engine revolutions and engine cooler temperature.

The system was implemented in the ProyectoFST third car, the FST-03, and the gathered data proves its correct operation and the great added value that it already is to the ProyectoFST team, IST's Formula Student team, that eight years after being created has the first data acquisition system implemented in one of its cars.

Índice

Resumo	III
Abstract	IV
Índice.....	V
Lista de Figuras	XI
Lista de Tabelas.....	XIII
Lista de Acrónimos.....	XV
1. Introdução.....	1
1.1. Objectivo	1
1.2. Motivação.....	1
1.3. O Projecto FST e a Formula Student.....	1
1.4. Sistemas de aquisição de dados no desporto motorizado	2
1.4.1. Sistemas desenvolvidos por equipas de <i>Formula Student</i>	2
1.4.2. Sistemas comerciais.....	4
1.4.2.1. MoTeC.....	4
1.4.2.2. McLaren Electronics.....	5
1.4.2.3. Bosch Motorsport.....	5
1.5. Organização da Dissertação	5
2. Arquitectura do Sistema Electrónico FST.....	7
2.1. Introdução	7
2.2. Objectivos do sistema.....	8
2.3. Especificação dos sensores e actuadores.....	9
2.3.1. Sensores de deslocamento linear	9
2.3.2. Sensor de velocidade	9
2.3.3. Sensor de temperatura sem contacto por radiação térmica.....	9
2.3.4. Acelerómetros	9
2.3.5. Sensor de pressão hidráulica.....	9
2.3.6. Sensor de ponto morto	9
2.3.7. Sensor de rotação do motor.....	9
2.3.8. Sensor de temperatura do líquido de refrigeração.....	10
2.3.9. Amperímetro	10

2.3.10. Sistema de monitorização de temperatura e pressão dentro dos pneus	10
2.3.11. Sensor de acelerador	10
2.4. Barramentos eléctricos de comunicações	10
2.4.1. Barramento SMBus	10
2.4.1.1. Camada física do SMBus	10
2.4.1.2. Camada de protocolo do SMBus	11
2.4.2. Barramento CAN	12
2.4.2.1. Trama de dados	12
2.4.2.2. Trama de dados estendida	13
2.4.2.3. Pedido de transmissão remoto	13
2.4.2.4. Trama de erro	13
2.4.3. Barramento SPI	13
2.4.4. Barramento USB	14
2.4.4.1. Camada física	14
2.4.4.2. Camada de Protocolo	15
2.5. Protocolo de comunicação entre módulos	15
2.5.1. Introdução	15
2.5.2. Eficiência do barramento CAN	16
2.5.3. Mensagens	16
2.5.4. Taxa de ocupação do barramento	16
2.6. Módulos de aquisição e processamento digital	18
2.6.1. Introdução	18
2.6.2. Componentes do Módulo Processador DSPICFST	18
2.6.2.1. Micro-controlador	18
2.6.2.2. Alimentação	19
2.6.3. Programação	19
2.6.4. Arquitectura do programa de interface com os sensores	19
2.6.4.1. Aquisição de sinais analógicos	20
2.6.4.2. Cálculo da frequência de uma onda quadrada	21
2.6.4.3. Comunicação através de SMBus	21
2.6.4.4. Comunicação através do barramento CAN	21

2.6.5. Consumo de energia	24
2.7. Sistema de interface USB-CAN.....	24
2.7.1. Introdução	24
2.7.2. Objectivo	24
2.7.3. Módulo de interface USB-CAN	24
2.7.3.1. Comunicação SPI com o controlador MCP2515	25
2.7.3.2. Comunicação USB com um PC.....	26
2.7.4. Programa de interface USB-CAN no PC	27
2.7.5. Aplicações do sistema	28
3. Sistemas sensoriais	29
3.1. Sensor de deslocamento linear	29
3.1.1. Introdução	29
3.1.2. Teoria de funcionamento	29
3.1.3. Circuito de excitação do primário	30
3.1.3.1. Ponte de Wien simples.....	31
3.1.3.2. Ponte de Wien com estabilização de amplitude	31
3.1.4. Condicionamento de sinal.....	33
3.1.5. Consumo de energia	35
3.1.6. Característica real e cálculo teórico do deslocamento	35
3.1.6.1. Aproximação polinomial de 3ª ordem da característica	36
3.1.6.2. Aproximação por interpolação linear dos pontos reais	36
3.1.7. Implementação do algoritmo no módulo de aquisição	38
3.2. Sensor de temperatura à superfície dos pneus	38
3.2.1. Introdução	38
3.2.2. Escolha do sensor	38
3.2.3. Configuração do dispositivo.....	39
3.2.4. Posicionamento do sensor.....	40
3.2.5. Leitura do valor de temperatura	40
3.2.6. Barramento SMBus de um módulo de aquisição.....	40
3.2.7. Algoritmo de leitura.....	41
3.2.8. Consumo de energia	41

3.3. Sensor de velocidade	41
3.3.1. Introdução	41
3.3.2. Sensor de efeito de Hall.....	42
3.3.3. Cálculo da velocidade.....	42
3.3.4. Implementação de um filtro de mediana.....	42
3.3.5. Consumo de energia	43
3.4. Sensor de temperatura do líquido de refrigeração	43
3.4.1. Introdução	43
3.4.2. Sensor.....	43
3.4.3. Circuito de condicionamento.....	43
3.4.4. Característica da termistância cerâmica.....	44
3.4.5. Implementação no módulo de aquisição	44
3.4.6. Consumo de energia	45
3.5. Sensor de acelerador.....	45
3.5.1. Introdução	45
3.5.2. O sensor	45
3.6. Sensor sem fios de temperatura e pressão dentro dos pneus	46
3.6.1. Introdução	46
3.6.2. Circuito emissor.....	46
3.6.2.1. Sensor de pressão e temperatura MLXY8020	47
3.6.2.2. Micro-controlador rfPIC12F675F	48
3.6.3. Circuito receptor	48
3.6.4. Modelação ASK.....	49
3.6.5. Protocolo de comunicação.....	49
3.6.6. Algoritmo de medição de pressão e temperatura no circuito emissor	50
3.6.7. Algoritmo de aquisição do sinal gerado pelo circuito receptor.....	50
3.6.8. Ritmo de transmissão de dados.....	50
4. Implementação dos sistemas no FST-03.....	51
4.1. Introdução	51
4.2. Arquitectura do Sistema Electrónico.....	51
4.3. Instalação mecânica dos diversos sensores no FST-03	51

4.3.1. Sensores de velocidade das rodas	51
4.3.2. Sensores de deslocamento da suspensão	52
4.3.3. Sensor de ângulo de viragem do volante.	52
4.3.4. Sensores de temperatura da superfície dos pneus.....	53
4.4. Placas de circuito impresso.....	54
4.4.1. Circuito de excitação e condicionamento de sinal dos LVDT	54
4.4.2. Suporte ao barramento SMBus e sensores <i>MLX9016</i>	55
4.4.3. Condicionamento de suporte aos sensores de origem	55
4.5. Interconexão de módulos.....	55
4.6. Caixas dos módulos de aquisição de sinais	55
4.7. Ligação dos módulos de aquisição de sinais aos sensores	57
4.8. Conclusões.....	57
5. Resultados Experimentais.....	59
5.1. Medição da temperatura à superfície dos pneus	59
5.2. Medição da velocidade	59
5.3. Medição da rotação do motor e do sensor de acelerador	60
5.4. Conclusões.....	60
6. Conclusões e perspectivas de trabalho futuro.....	61
6.1. Arquitectura do sistema	61
6.1.1. Conclusões	61
6.1.2. Trabalho futuro	61
6.2. Sensores de velocidade das rodas.....	62
6.2.1. Conclusões	62
6.2.2. Melhorias do sistema actual.....	62
6.3. Sensores de temperatura da superfície dos pneus.....	62
6.4. Módulo de Interface USB-CAN.....	63
6.5. Sensores de deslocamento linear	63
6.5.1. Conclusões	63
6.5.2. Melhorias do sistema actual.....	63
6.5.3. Nova abordagem ao sensor de deslocamento	63
6.6. Sensor de pressão e temperatura no interior do pneu	64

Referências.....	65
Anexo A – Esquemáticos das PCI utilizadas.....	67

Lista de Figuras

Figura 1 – Arquitectura do sistema vencedor do prémio “ <i>Bosch Best use of Electronics</i> ” .	3
Figura 2 – PCI MiniCAN da Universidade de Zwickau.	3
Figura 3 – Volante do protótipo de 2009 da Universidade de Zwickau.	4
Figura 4 – Sistema completo de aquisição de dados MoTeC.	4
Figura 5 – Módulo de armazenamento HSL-500 da <i>McLaren Electronics</i> .	5
Figura 6 – Arquitectura geral do sistema.	8
Figura 7 – Configuração do barramento SMBus utilizada.	11
Figura 8 – Legenda dos diagramas de envio e recepção de palavras.	11
Figura 9 – Diagrama de escrita de palavra com verificação de erro.	11
Figura 10 – Diagrama de leitura de palavra com verificação de erro.	12
Figura 11 – Trama de dados.	13
Figura 12 – Diagrama de ligações num barramento SPI de escravos independentes.	14
Figura 13 – Sequência de bits codificada em NRZI com <i>bit stuffing</i> .	15
Figura 14 – Módulo de aquisição e processamento de sinais.	18
Figura 15 – Diagrama da arquitectura do micro-controlador <i>dsPIC30F6012A</i> .	19
Figura 16 – Fluxograma exemplo da rotina de interrupção do temporizador de aquisição.	20
Figura 17 – diagrama das partes constituintes de um bit.	22
Figura 18 – Diagrama do sistema de Interface USB-CAN.	24
Figura 19 – PCI do sistema de Interface USB-CAN.	25
Figura 20 – Interface visual do programa de comunicação USB-CAN.	27
Figura 21 – Formulário de adição de mensagens CAN à lista de envio.	28
Figura 22- constituição de um LVDT convencional.	29
Figura 23 – Constituição do LVDT desenvolvido.	30
Figura 24 – Ponte de Wien.	31
Figura 25 – Circuito ponte de Wien com estabilização de amplitude.	32
Figura 26 – Característica da amplitude das oscilações com o valor do potenciómetro.	33
Figura 27 – Circuito de condicionamento de sinal para um LVDT.	33
Figura 28 – Característica de um conjunto LVDT e placa de condicionamento.	35
Figura 29 – Aproximação polinomial de 3ª ordem à característica real do LVDT.	36
Figura 30 – Aproximação por Interpolação Linear à característica real do LVDT.	37
Figura 31 – Roda com um ângulo de <i>camber</i> negativo.	38
Figura 32 – Diagrama de posicionamento dos sensores de temperatura.	40
Figura 33 – Esquema do barramento SMBus nas rodas dianteiras e traseiras.	40
Figura 34 – Fluxograma da função responsável pela leitura dos sensores de temperatura.	41
Figura 35 – Esquema para medição da velocidade com um sensor de efeito de Hall.	42
Figura 36 – Circuito de condicionamento do termistor.	44
Figura 37 – Característica do termistor.	44

Figura 38 – Característica $V_o(T)$ do circuito da Figura 36.....	45
Figura 39 – Variação do consumo do sensor com a Temperatura.....	45
Figura 40 – Diagrama do sensor de acelerador.	46
Figura 41 – Diagrama da PCI do sensor e emissor de RF.	46
Figura 42 – PCI de suporte ao sensor e circuito emissor de RF.....	47
Figura 43 – Diagrama do micro-controlador rfPIC12F675F.....	48
Figura 44 – PCI do circuito receptor.	49
Figura 45 – Exemplo de modelação ASK.	49
Figura 46 – Trama de dados da comunicação RF.....	49
Figura 47 – Montagem do sensor de efeito de Hall no FST-03.....	51
Figura 48 - LVDT montado no amortecedor do FST-03.	52
Figura 49 – LVDT montado na caixa de direcção do FST-03.	53
Figura 50 – Apoio dianteiro dos sensores de temperatura da superfície dos pneus.	53
Figura 51 – Apoio traseiro e PCI dos sensores de temperatura da superfície dos pneus.....	54
Figura 52 – PCI de excitação e condicionamento de sinal de um LVDT.....	54
Figura 53 – PCI de suporte aos sensores de temperatura da superfície dos pneus.	55
Figura 54 – Dimensões da caixa dos módulos de interface com sensores.	56
Figura 55 – Conector SureSeal 120-8552-202.....	56
Figura 56 – Projecto da caixa dos módulos de interface com sensores.....	56
Figura 57 – Conector Mini-SureSeal 120-8552-102.	57
Figura 58 – Conector SureSeal 120-1806-000.....	57
Figura 59 – Comparação entre velocidade e temperatura na parte exterior (azul) e interior (vermelho) da superfície do pneu.....	59
Figura 60 – Velocidade das rodas dianteiras direita (preto) e esquerda (vermelho).....	60
Figura 61 – Comparação de gráficos da rotação e do acelerador.	60
Figura 62 – Arquitectura proposta para o sistema de excitação e condicionamento de sinal de um LVDT.	64
Figura 63 – Esquemático da PCI de excitação e condicionamento de sinal de um LVDT.	67
Figura 64 – Esquemático da PCI de suporte aos sensores de temperatura da superfície dos pneus.....	68
Figura 65 – Esquemático da PCI de suporte aos sensores fornecidos pelo fabricante do motor.	68
Figura 66 – Esquemático da PCI de suporte ao sensor de pressão e emissor de radiofrequência.....	69
Figura 67 – Esquemático da PCI de suporte ao receptor de rádio frequência.	70

Lista de Tabelas

Tabela 1 – descrição dos condutores num barramento USB.....	14
Tabela 2 – Eficiência do barramento CAN em função do número de bytes de dado.....	16
Tabela 3 – Frequência de amostragem de cada sensor.....	17
Tabela 4 – Funções disponíveis na biblioteca <i>i2c.h</i>	21
Tabela 5 - Descrição das componentes configuráveis de um bit.	22
Tabela 6 – Tabela de verdade do sistema máscara e filtro.	23
Tabela 7 - Tipos de mensagem no protocolo USB.	26
Tabela 8 – Característica e parâmetros para cálculo da interpolação linear.	37
Tabela 9 – Lista de comandos SMBus reconhecidos pelo MLX90614.....	39
Tabela 10 – Sequência de comandos SMBus para programação de um dispositivo.	39
Tabela 11 – Tabela dos estados do sensor de pressão e temperatura.....	47

Lista de Acrónimos

ADC	<i>Analog-to-Digital Converter</i>
ASK	<i>Amplitude Shift Keying</i>
CA	Corrente Alternada
CAN	<i>Controller Area Network</i>
CC	Corrente Contínua
CRC	<i>Cyclic Redundancy Check</i>
dsPIC	<i>digital signal Peripheral Interface Controller</i>
ECT	<i>Engine Coolant Temperature</i>
ECU	<i>Engine Control Unit</i>
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
FSK	<i>Frequency Shift Keying</i>
FST	<i>Formula Student Técnico</i>
IAT	<i>Intake Air Temperature</i>
I ² C	<i>Inter-Integrated Circuit</i>
IS	Interface com Sensores
LVDT	<i>Linear Variable Differential Transformer</i>
MAP	<i>Manifold Air Pressure</i>
PCI	Placa de Circuito Impresso
SAE	<i>Society of Automotive Engineering</i>
SPI	<i>Serial Peripheral Interface</i>
SMBus	<i>System Management Bus</i>
RAM	<i>Random Access Memory</i>
RF	<i>Radio Frequency</i>
RPM	Rotações Por Minute
USB	<i>Universal Serial Bus</i>

1. Introdução

Neste capítulo pretende-se estabelecer a motivação e os objectivos do trabalho realizado, bem como dar uma visão panorâmica do que é a Formula Student e de como têm evoluído os sistemas electrónicos nesta competição.

1.1. Objectivo

Este trabalho tem por objectivos o desenvolvimento de uma arquitectura de suporte a uma rede sensores e actuadores, o desenvolvimento de alguns desses sensores e actuadores e a implementação do sistema completo num protótipo de Formula Student do ProjectoFST, o FST-03.

1.2. Motivação

Este trabalho surge como continuação e aprofundamento do trabalho realizado pela equipa de electrónica do ProjectoFST nos últimos anos e tem por motivação a criação de uma rede de sensores e actuadores integrada capaz de recolher, em tempo real, dados relevantes para o diagnóstico e monitorização da dinâmica de um carro de competição.

A monitorização dinâmica permite aumentar o conhecimento da equipa sobre os próprios automóveis, possibilitando assim que veículos futuros vejam o seu desempenho melhorado, assim como também permite uma melhor avaliação dos acontecimentos em tempo real, facilitando o processo de afinação do veículo e melhorando o resultado final dessa mesma afinação. A monitorização dinâmica do veículo em tempo real vem ainda possibilitar a actuação de sistemas de ajuda à condução, como sejam o controlo de tracção ou um sistema de mudanças semi-automáticas.

Por outro lado, o diagnóstico em tempo real permite a detecção rápida de problemas e a sua fonte, resultando, em algumas situações, na minimização das consequências desse problema.

Como escola de engenharia de topo que o IST é, deve promover a realização deste tipo de sistemas por parte dos estudantes que pretendem participar nas competições como a FSAE. Para além da significativa verba financeira poupada pela equipa, a aplicação prática dos conhecimentos teóricos adquiridos nas aulas em desenvolvimento de sistemas deste tipo pelos alunos proporciona-os uma primeira abordagem àquilo que é realmente engenharia.

1.3. O Projecto FST e a Formula Student

A Formula Student é, desde 1998, a versão Europeia da Formula Society of Automotive Engineering (FSAE). Esta última nasceu nos EUA em 1981 como uma competição de engenharia destinada a estudantes universitários. Segundo o site oficial da FSAE¹ “O conceito da Formula SAE é que uma empresa de manufactura fictícia contrata uma empresa de projecto de engenharia para

¹ <http://students.sae.org/competitions/formulaseries/about.htm>

desenvolver um carro de competição do tipo Formula². O protótipo será avaliado pelo seu potencial como um produto comercial. (...) Cada equipa de estudantes projecta, constrói e testa o seu protótipo baseada numa série de regras cujo objectivo é tanto garantir o funcionamento do protótipo como promover a solução inteligente de problemas”.

O ProjectoFST nasceu como a equipa de Formula Student do Instituto Superior Técnico em 2001 e conta à data com três protótipos construídos.

Como nos primeiros anos a equipa só contava com alunos de engenharia mecânica, os primeiros dois protótipos são muito pobres a nível eléctrico, estando reduzidos ao mínimo essencial para garantir o correcto funcionamento do veículo, mas com o nascimento da equipa de electrónica em 2006, o terceiro protótipo conta já com um sistema eléctrico bem pensado, feito à medida das necessidades e com um grau de fiabilidade muito elevado, contando ainda com um sistema eléctrico de mudanças sequenciais sem embraiagem.

Paralelamente ao sistema eléctrico de suporte do terceiro protótipo, a equipa de electrónica começou a trabalhar em projectos como redes de sensores, aquisição de dados, telemetria e sistema de ganhos de injeção, sendo este documento relativo à segunda versão do projecto de redes de sensores, e estando as primeiras versões dos outros projectos a ser finalizadas também agora.

1.4. Sistemas de aquisição de dados no desporto motorizado

No que diz respeito às restantes equipas que participam regularmente na *Formula Student*, é comum o uso de centralinas e sistemas de aquisição de dados comerciais, havendo ainda equipas que desenvolvem as suas próprias versões da centralina *open-source MegaSquirt* [1].

1.4.1. Sistemas desenvolvidos por equipas de *Formula Student*

Entre as poucas equipas que fogem à regra e constroem os seus próprios sistemas de aquisição de dados encontra-se a universidade *West Saxon University of Applied Sciences of Zwickau*, na Alemanha. Esta equipa apresentou no seu protótipo de 2008 o sistema de aquisição de dados da Figura 1, vencedor do prémio Bosch para a melhor electrónica. Este sistema é baseado na pequena PCI da Figura 2 que consiste num módulo *Controller Area Network* (CAN) de interface com os sensores.

² Monolugar de rodas descobertas.

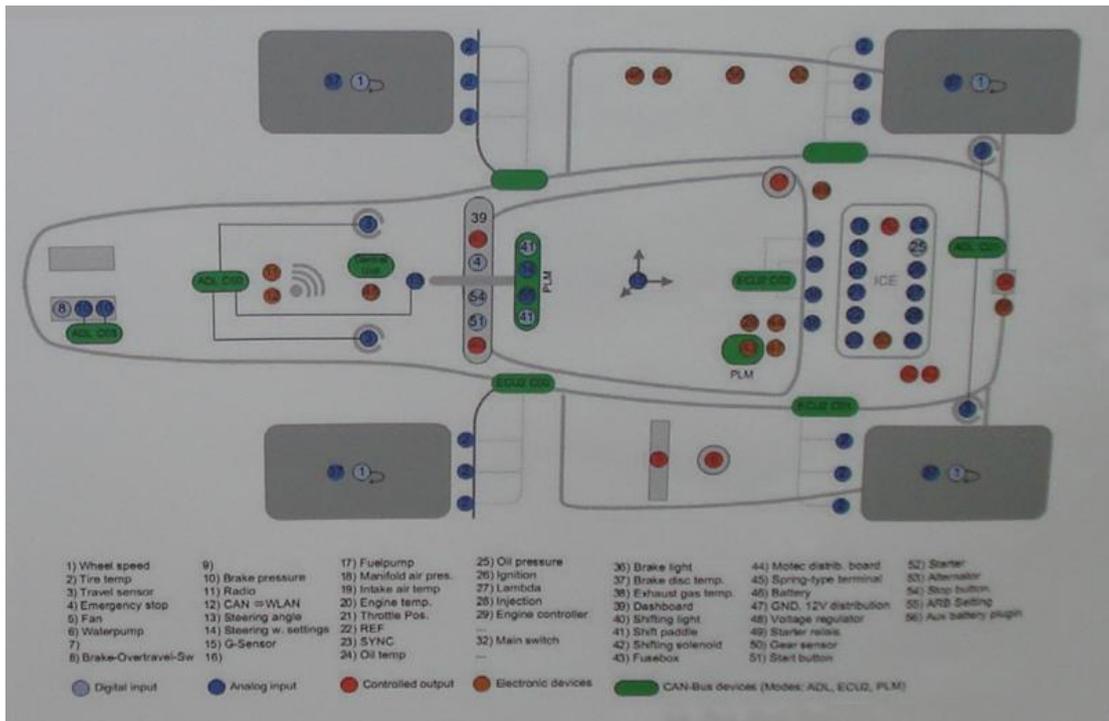


Figura 1 – Arquitetura do sistema vencedor do prêmio “Bosch Best use of Electronics”.

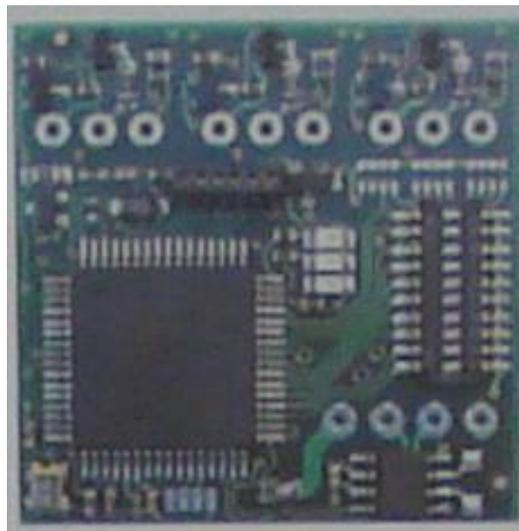


Figura 2 – PCI MiniCAN da Universidade de Zwickau.

A mesma equipa apresentou ainda o volante da Figura 3, um volante totalmente em fibra de carbono com um módulo de informações ao piloto e controlo do veículo. Este apresenta a informação da relação de caixa engrenada e das rotações por minuto do motor, tendo ainda outro mostrador cujas informações podem ser escolhidas através de um selector rotativo. Apresenta ainda dois potenciômetros para ajuste do controlo de tracção e das rotações máximas do motor, bem como botões para colocação do carro em neutro, telecomunicações do piloto com a box e aquilo que parecem ser dois botões de ajuste da barra de torção do veículo.



Figura 3 – Volante do protótipo de 2009 da Universidade de Zwickau.

1.4.2. Sistemas comerciais

1.4.2.1. MoTeC

No que diz respeito a soluções comerciais, os sistemas mais utilizados no desporto motorizado são os da empresa australiana MoTeC [2]. O sistema topo de gama desta empresa, baseado no módulo de aquisição e armazenamento de dados *Advanced Central Logger* (ACL), é extremamente completo e rápido, está preparado para lidar com qualquer tipo de sensores e a programação é feita em ambiente gráfico intuitivo.



Figura 4 – Sistema completo de aquisição de dados MoTeC.

Como se pode ver pelo diagrama da Figura 4, o sistema pode conter até 8 *Versatile Input Module* (VIM) onde se ligam os sensores e que com a devida programação passam os dados dos sensores para o barramento CAN, sendo depois armazenados pelo ACL a uma taxa de até 5000

amostras por segundo numa memória de 1 GB. Os dados armazenados podem posteriormente ser descarregados para um PC via Ethernet, mas podem também ser enviados em tempo real para a box com recurso a um sistema GSM ou com um sistema de rádio, ambos também da MoTeC. Para além deste sistema topo de gama, a MoTeC disponibiliza também uma variada gama de sistemas menos desenvolvidos, vendendo também sensores de todos os tipos e mostradores de informações ao piloto. Esta empresa é também ainda a líder de mercado em centralinas programáveis na Formula Student e nas equipas privadas de competições como a GP2 e IRC.

1.4.2.2. McLaren Electronics

No que toca às grandes competições automóveis como a Formula1 ou o *World Rally Championship* (WRC) já os fornecedores de sistemas electrónicos são outros. É nestas competições que surgem a Magneti-Marelli [3] e a McLaren Electronics [4]. No caso da Magneti-Marelli revelou-se extremamente difícil encontrar informações sobre os seus sistemas, mas no caso da McLaren Electronics já foi possível obter mais informações.

A McLaren Electronics, para além de sensores de todos os tipos, comercializa um módulo de aquisição HSL-500 da Figura 5 com 6 ligações CAN, 4 GB de armazenamento e uma capacidade 400000 amostras por segundo e é baseado num microprocessador PowerPC. A empresa comercializa ainda centralinas totalmente programáveis, mostradores de informações ao piloto, sistemas completos de telemetria e receptores de GPS.



Figura 5 – Módulo de armazenamento HSL-500 da McLaren Electronics.

1.4.2.3. Bosch Motorsport

Também a Bosch, na sua divisão de *Motorsport* [5], disponibiliza sistemas de sensores e aquisição de dados de variados tipos. No seu sistema topo de gama, o *MSI 55* alimenta e condiciona o sinal dos sensores, disponibilizando então os dados por um barramento FireWire a um sistema de armazenamento como o *CardMemory C 55*, que armazena os dados num cartão de memória *flash* até 1 GB.

1.5. Organização da Dissertação

A dissertação está organizada de forma a serem primeiramente definidas as questões de nível superior, como a arquitectura global de todo o sistema electrónico, e daí descendo para o desenvolvimento dos sensores, a sua implementação no protótipo, os resultados experimentais e finalmente as conclusões e propostas de trabalho futuro.

2. Arquitectura do Sistema Electrónico FST

Neste capítulo estabelece-se a arquitectura do sistema como um todo, dando uma visão panorâmica e definindo-se quais os sensores e actuadores a desenvolvidos.

2.1. Introdução

O sistema consiste numa rede de módulos de aquisição de dados responsáveis por amostrar os sinais dos sensores e disponibiliza-los, através de um barramento, a uma unidade de armazenamento e transmissão por *ZigBee* [6] e a um mostrador de informação ao piloto no volante. Foi para tal escolhido o barramento CAN por ser altamente robusto em ambientes como o do protótipo, ter uma taxa de transmissão de dados máxima de 1 Mb/s e estar amplamente difundido entre os fabricantes de micro-controladores.

O sistema idealizado é constituído por módulos de aquisição de sinais, denominados módulos de Interface com Sensores (IS), responsáveis por amostrar os sinais dos sensores e disponibilizá-los num dos barramentos CAN para que sejam recebidos por módulos com diversas funções. Entre eles estão um mostrador de informações no volante, um módulo de processamento (Dyno) que calcula potência, binário, perdas aerodinâmicas e perdas de rolamento do veículo, um módulo (TX/RX) de armazenamento e transmissão de dados por rádio. O sistema dispõe ainda de um módulo de controlo do motor (ECU) que utiliza dados de sensores de pressão e temperatura do ar, posição do acelerador e rotações por minuto do motor para calcular a quantidade de combustível que deve ser injectada e o instante correcto para a injeção e ignição do combustível.

Todos estes sistemas são implementados em módulos de processamento semelhantes. Estes módulos, descritos em 2.6, podem ser utilizados em todos os sistema e possuem capacidade para lidar com dois barramentos CAN. Estão projectados para poderem ser directamente acoplados a outras PCI, economizando espaço. No que toca aos sistemas IS, o mesmo veículo poderá ter quantos forem necessários, sendo que para suporte dos sensores base, tendo em conta o número de conectores necessários e o tamanho das caixas, decidiu-se que nesta primeira fase seriam três, um em cada zona do carro e com as funções descritas em seguida.

- **IS 1 - Dianteiro**
 - Sensores de deslocamento dos amortecedores dianteiros;
 - Sensores de velocidade das rodas dianteiras;
 - Sensores de temperatura da superfície dos pneus dianteiros;
- **IS 2 - Traseiro**
 - Sensores de deslocamento dos amortecedores traseiros;
 - Sensores de velocidade das rodas traseiras;
 - Sensores de temperatura da superfície dos pneus traseiros;
- **IS 3 - Intermédio**
 - Sensor de neutro da caixa de velocidades;
 - Sensor de rotação do motor (RPM);

- Indicação de erro da *Engine Control Unit* (ECU);
- Sensor de temperatura do líquido de refrigeração do motor (ECT);
- Sensor de ângulo da direcção;
- Sensores de pressão e temperatura no interior dos pneus;
- Sensor de acelerador (TPS);
- Sensor de força de travagem dianteira;
- Sensor de força de travagem traseira;
- Temperatura na admissão de ar (IAT);
- Pressão na admissão de ar (MAP);

Na Figura 6 pode observar-se a arquitectura do sistema desenvolvido.

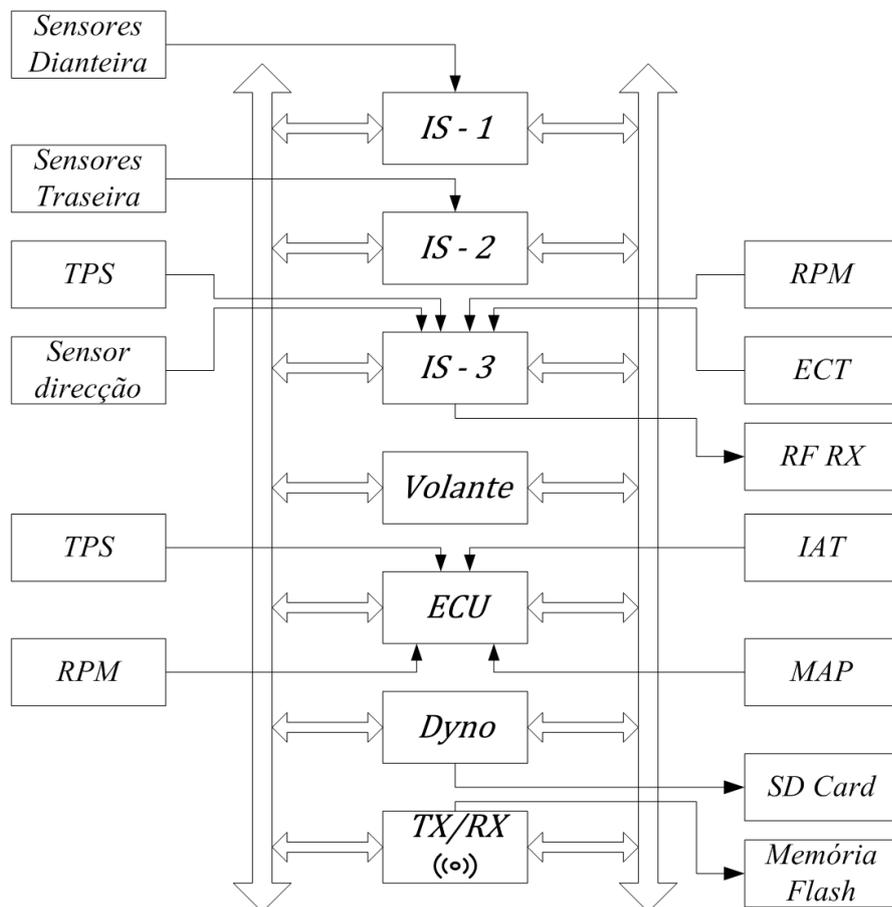


Figura 6 – Arquitectura geral do sistema.

2.2. Objectivos do sistema

O sistema tem por objectivo disponibilizar, através de dois barramentos CAN, os dados adquiridos dos sensores ao módulo de armazenamento e transmissão por ZigBee, para que sejam armazenados para posterior análise ou para serem analisados de imediato na *box*, bem como para que alguns desses dados possam ser disponibilizados ao piloto em tempo real no mostrador do volante. Pretende-se que o sistema total seja o mais barato e versátil possível.

2.3. Especificação dos sensores e actuadores

2.3.1. Sensores de deslocamento linear

Os sensores de deslocamento são usados tanto na medição do deslocamento da suspensão, com um sensor montado em paralelo com o amortecedor, como na medição do ângulo da direcção, com o sensor fixado à caixa de direcção e solidário com o tirante da direcção. O sensor deverá ser capaz de medir deslocamentos lineares com uma excursão máxima de 10 cm a uma frequência máxima de 10 Hz .

2.3.2. Sensor de velocidade

Os sensores de velocidade são utilizados para medição da velocidade das rodas. Devem ser capazes de medir velocidades até 200 km/h e funcionar com uma tensão de alimentação de 5 V .

2.3.3. Sensor de temperatura sem contacto por radiação térmica

Os sensores de temperatura sem contacto são utilizados para medir a temperatura da superfície dos pneus a uma frequência mínima de 5 Hz a uma distância de segurança mínima de 2 cm .

Estes sensores podem ainda ser úteis para medição de temperatura dos discos de travão, mas nesse caso a frequência de amostragem deverá ser pelo menos de 20 Hz .

2.3.4. Acelerómetros

Os acelerómetros são úteis para o cálculo das forças e vibrações a que determinados componentes do veículo estão sujeitos. Pretende-se ter uma leitura da aceleração nos três eixos dimensionais a uma frequência de cerca de 100 Hz mas a frequência de saída pode ser inferior a 10 Hz , a tensão de alimentação deverá ser de 5 V e a aceleração a medir não superará os $3g$.

2.3.5. Sensor de pressão hidráulica

O sensor de pressão hidráulica, quando é colocado no circuito hidráulico de travagem permite saber a pressão na linha de travagem e conseqüentemente força de travagem que está a ser aplicada pelo piloto. Se for combinado com um sensor de deslocamento no pedal permite ainda diagnosticar problemas no circuito de travagem. O sensor deverá ser capaz de medir pressões máximas de 60 bar com uma frequência de amostragem de 10 Hz .

2.3.6. Sensor de ponto morto

O sensor de ponto morto ou de neutro na caixa de velocidades vem de origem com o motor/caixa de velocidades e consiste num interruptor na caixa de velocidades. O interruptor encontra-se fechado à massa quando a caixa está em neutro e aberto em qualquer outra situação.

2.3.7. Sensor de rotação do motor

O sensor de rotação do motor, que vem de origem com o motor, consiste num sensor de rotação baseado num sensor magnético de efeito de Hall localizado na árvore de cames, é lido directamente pela ECU que a partir dele gera uma onda quadrada de frequência f proporcional ao número de RPM que o motor está a realizar, dada por (1).

$$f = \frac{RPM}{30} \quad (1)$$

2.3.8. Sensor de temperatura do líquido de refrigeração

Este sensor faz parte do equipamento inicial do motor do FST-03 (mota Honda CBR600F4i) e consiste numa termistância cerâmica do tipo *Negative Temperature Coefficient* (NTC) com um dos terminais ligado à massa. Para efeitos de diagnóstico, deverá ser amostrado a 1 Hz .

2.3.9. Amperímetro

Um amperímetro colocado em série com a bateria permite a monitorização da corrente nesta, possibilitando a detecção de problemas com o alternador bem como dando à equipa uma ideia da corrente total necessária para o funcionamento do carro. Deverá medir uma corrente máxima de 200 A a uma frequência de amostragem de 10 Hz .

2.3.10. Sistema de monitorização de temperatura e pressão dentro dos pneus

Este sistema consiste em sensores de temperatura e de pressão que se encontram dentro dos pneus alimentados por pilhas e que comunicam por rádio frequência com um receptor. A comunicação deverá funcionar a $1,5\text{ m}$ de distância, a pressão a medir no interior do pneu varia entre $2,5\text{ Bar}$ e $4,5\text{ Bar}$ e a temperatura interna não excederá os $100\text{ }^\circ\text{C}$.

2.3.11. Sensor de acelerador

O sensor de acelerador faz parte da cablagem original do sistema (derivado da mota Honda CBR600F4i) e consiste num potenciómetro solidário com a válvula borboleta da admissão de ar do motor. Este sensor é alimentado com 5 V , apresentando assim na saída um valor de tensão entre 0 V e 5 V proporcional ao ângulo de rotação da borboleta de admissão.

2.4. Barramentos eléctricos de comunicações

No decorrer do desenvolvimento dos diversos sistemas surgiu a necessidade de utilização de alguns barramentos eléctricos de comunicação.

2.4.1. Barramento SMBus

O *System Management Bus* (SMBus) [7] é um barramento de comunicação multi-mestre de dois condutores baseado no barramento *Inter-Integrated Circuit* (I^2C) podendo trabalhar numa gama entre os 10 kb/s e os 100 kb/s .

2.4.1.1. Camada física do SMBus

O barramento é composto por dois condutores, um de sinal de relógio (SCL) e outro de dados (SDA). Na configuração da Figura 7, utilizada neste trabalho, o sistema SMBus tem um mestre e vários escravos, sendo o mestre que controla o sinal de relógio e que partilha o controlo da linha de dados com os escravos.

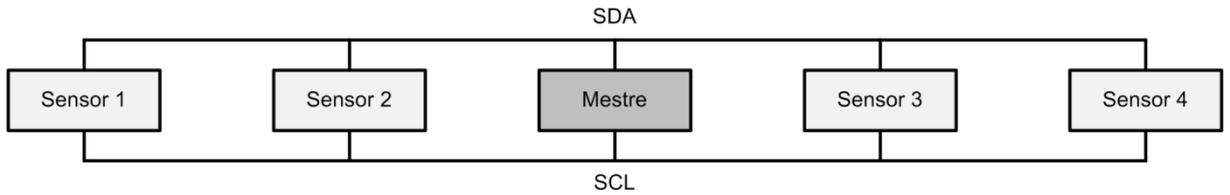


Figura 7 – Configuração do barramento SMBus utilizada.

2.4.1.2. Camada de protocolo do SMBus

O protocolo é bastante complexo, contendo vários formatos de mensagens das quais se passam a explicar os que foram usados no decorrer do trabalho, mais especificamente nos sensores de temperatura da superfície dos pneus.

- S Condição de início
- Sr Condição de início repetida
- Rd Leitura
- Wr Escrita
- A Confirmação
- PEC Pacote de verificação de erros
- Do mestre para o destinatário
- Do destinatário para o mestre
- P Condição de final

Figura 8 – Legenda dos diagramas de envio e recepção de palavras.

2.4.1.2.1. Escrita de palavra com verificação de erro

A escrita de palavra com verificação de erro, como se pode ver pela Figura 9, começa com a condição de início e com o mestre a enviar 7 bits de identificação do destinatário seguidos de um bit que indica que a próxima palavra será um comando do mestre. O mestre envia de seguida os 8 bits de comando, seguidos dos 16 bits de dados e do pacote de verificação de erro, terminando com a condição de finalização. A cada 8 bits enviados pelo mestre, este liberta o controlo do barramento para que o destinatário possa confirmar que os dados foram recebidos.

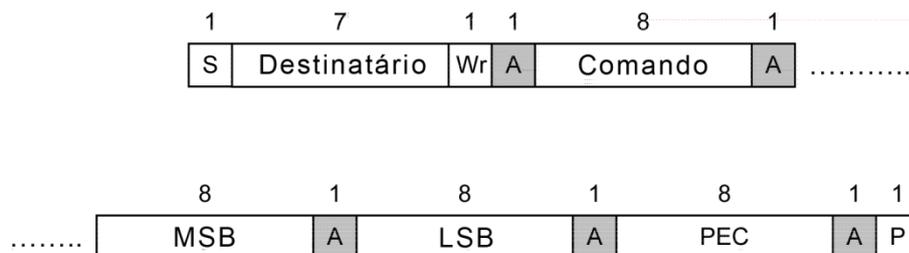


Figura 9 – Diagrama de escrita de palavra com verificação de erro.

2.4.1.2.2. Leitura de palavra com verificação de erro

A leitura de palavra com verificação de erro, como se pode ver pela Figura 10, começa com a condição de início e com o mestre a enviar 7 bits de identificação do destinatário seguidos de um bit que indica que a próxima palavra será um comando do mestre. O mestre envia de seguida os 8 bits de comando e volta a enviar os 7 bits de identificação do destinatário mas enviando desta vez um bit a indicar que deverá ser o destinatário a escrever no barramento em seguida. O mestre liberta então o controlo do barramento para que o destinatário possa enviar os 16 bits de dados e os 8 bits do pacote de verificação de erro. A cada 8 bits enviados pelo mestre, este liberta o controlo do barramento para que o destinatário possa confirmar que os dados foram recebidos e vice-versa.

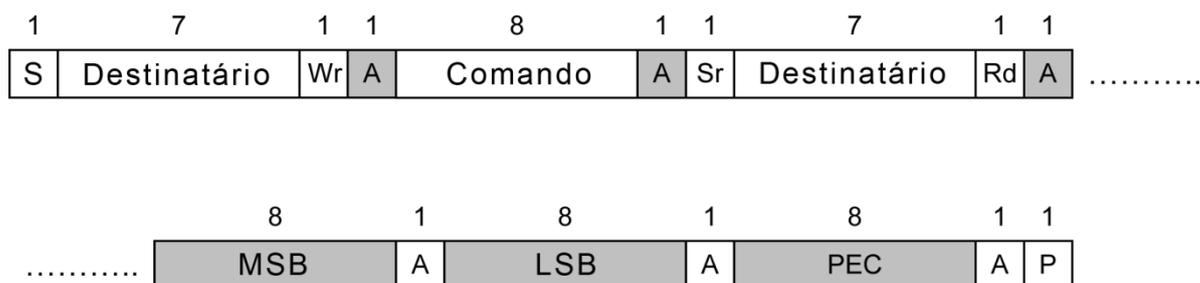


Figura 10 – Diagrama de leitura de palavra com verificação de erro.

2.4.2. Barramento CAN

O barramento *Controller Area Network* (CAN) [8] foi desenvolvido especificamente para aplicação em automóveis, sendo extremamente robusto em ambientes de elevado ruído e permitindo comunicação até 1 Mbit/s com um comprimento máximo de 40 m . Pelas características previamente enunciadas, e por ser um barramento de difusão de informação (*broadcasting*) multi-mestre amplamente difundido nos fabricantes de micro-controladores, foi escolhido como forma de comunicação entre os módulos que compõem o sistema electrónico do veículo.

As mensagens num barramento CAN podem ser de quatro formatos distintos.

2.4.2.1. Trama de dados

Uma trama de dados (Figura 11) é gerada por um nó que pretende difundir dados no barramento, e tal como todos os outros tipos de mensagem, começa com um bit de *Start-of-Frame* (Início de trama). O *SOF* é seguido pelo campo de arbitrariedade, que consiste em 11 bits de identificação da mensagem e no bit de *Remote Transmission Request* (Pedido de transmissão remoto), que diferencia a Trama de dados do Pedido de transmissão remoto (ver 2.4.2.3). Segue-se o campo de controlo composto pelo bit de *IDE* (extensão de identificador), um bit reservado e quatro bits que indicam quantos bytes terá o campo de dados. Finalmente são enviados os bytes de dados e mensagem termina com os 16 bits do campo de verificação de erros, ou *CRC* (*Cyclic Redundancy Check*), seguidos de um período de espera por assentimentos de recepção sem erros e da condição de final de trama.

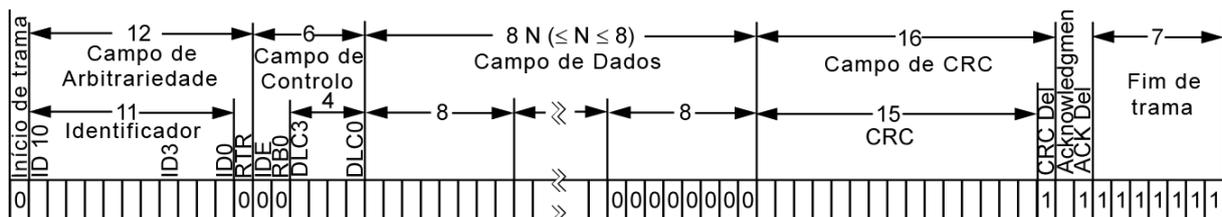


Figura 11 – Trama de dados.

2.4.2.2. Trama de dados estendida

A trama de dados estendida difere da trama de dados normal porque o bit de *IDE* vem activado e é sucedido de mais *18 bits* de identificação, sendo o restante da mensagem semelhante. Isto permite que sejam reconhecidas no sistema *536870912* mensagens diferentes, em vez das apenas *2048* da trama de dados normal.

2.4.2.3. Pedido de transmissão remoto

O pedido de transmissão remoto é uma trama semelhante à de dados mas em que o bit *RTR* vem activado, não sendo enviados quaisquer dados. Este tipo de mensagem serve para sinalização de eventos ou para que um nó do barramento requirite dados aos restantes.

2.4.2.4. Trama de erro

A trama de erro é gerada por qualquer nó do barramento que detecte um erro e é constituída por dois campos, um campo de sinalização do erro e um campo delimitador. Existem dois tipos de campo de sinalização, que dependem do estado de erro do nó que detecta o erro.

Se o nó que detecta o erro está no estado “activo ao erro”, então este interrompe a transmissão da mensagem actual gera uma trama de erro com um campo de sinalização correspondente à sinalização activa. Este campo, composto por *6 bits* dominantes consecutivos, viola propositadamente a regra de *bit stuffing*, o que é detectado por todos os restantes nós do barramento, causando que todos enviem por sua vez tramas de erro. Depois de todos os nós enviarem as suas tramas de erro, o barramento volta ao normal e o nó interrompido reenvia a mensagem interrompida.

No caso do nó que detecta o erro estar no estado “passivo ao erro”, este transmite uma trama de erro com o campo de sinalização correspondente à sinalização passiva, que consiste em 6 bits recessivos consecutivos. Isto significa que não haverá transmissão da trama de erro a não ser que o nó que detecta o erro seja também o nó que detém o barramento nesse instante.

2.4.3. Barramento SPI

O barramento *Serial Peripheral Interface* (SPI) [9] é um barramento *full-duplex* desenvolvido para comunicação entre circuitos integrados. Na sua configuração mais usual, apresentada na Figura 12, este barramento é constituído por 3 condutores base e um extra por cada dispositivo escravo que se acrescente. De seguida apresenta-se a função de cada condutor do barramento.

- SCK (*Serial Clock*) – Sinal de relógio gerado pelo dispositivo mestre para todos os dispositivos escravos;
- SDI (*Serial Data Input*) – Entrada do dispositivo;
- SDO (*Serial Data Output*) – Saída do dispositivo;
- SS (*Slave Select*) – Sinal de selecção de dispositivo escravo.

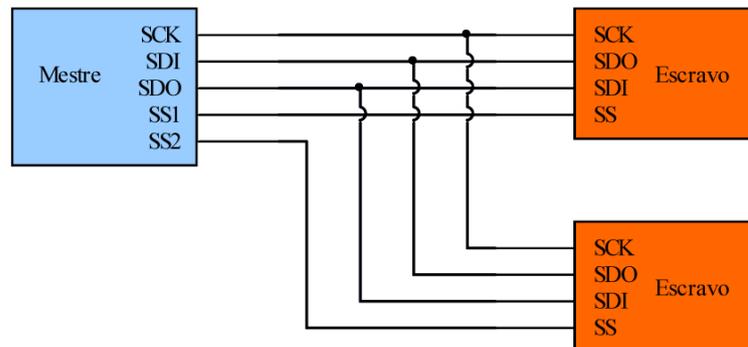


Figura 12 – Diagrama de ligações num barramento SPI de escravos independentes.

2.4.4. Barramento USB

O barramento *Universal Serial Bus* (USB) [10] foi desenvolvido em meados dos anos 90 por um consórcio de empresas de electrónica e está amplamente difundido na indústria dos computadores pessoais e seus periféricos. Uma rede USB consiste num sistema anfitrião, normalmente um PC que gere o barramento, os periféricos, e um ou mais *hubs* que providenciam a ligação entre periféricos e PC.

2.4.4.1. Camada física

A camada física do barramento USB é feita por quatro condutores (ver Tabela 1) que consistem em alimentação e dados. A transmissão de bits é feita através dos condutores $D+$ e $D-$, que estão sempre em níveis de tensão opostos. A codificação dos bits é feita através do método NRZI (*Non Return to Zero Inverted*) com *bit stuffing*. O método NRZI consiste em fazer variar a tensão da linha quando se pretende transmitir um zero ou mantê-la quando se pretende transmitir um um, e a implementação de *bit stuffing* significa que a cada seis 1 consecutivos é transmitido um 0, que apesar de não conter informação útil para a mensagem, permite ao receptor detectar que o barramento não ficou bloqueado num nível de tensão (ver Figura 13).

Tabela 1 – descrição dos condutores num barramento USB.

Pino	Nome	Cor	Funcionalidade
1	V_{DD}	Vermelho	Alimentação, +5 V
2	$D-$	Branco	Dados polarização negativa
3	$D+$	Verde	Dados polarização positiva
4	GND	Preto	Referência de tensão

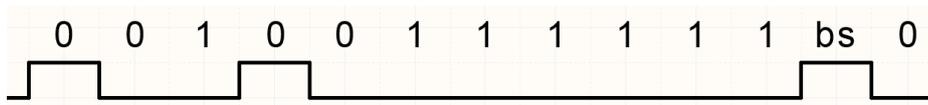


Figura 13 – Sequência de bits codificada em NRZI com *bit stuffing*.

2.4.4.2. Camada de Protocolo

Num barramento USB, as mensagens são transmitidas por pacotes, sendo cada pacote constituído por diversos campos que dependem do tipo de mensagem a transmitir. Os campos possíveis são o identificador (PID), endereço (ADDR) e (ENDP), os dados (DATA), o número de trama e o campo de verificação de erro (CRC).

2.4.4.2.1. Pacote de *Token*

Os pacotes de *token* são transmitidos pelo anfitrião para sinalizar a transmissão de dados e é constituído por identificador, endereço, *endpoint* e verificação de erro.

2.4.4.2.2. Pacote de Dados

Os pacotes de dados são enviados pelo anfitrião ou pelos dispositivos quando pretendem transmitir dados e são constituídas por PID, DATA e CRC.

2.4.4.2.3. Formato das Transacções

As transacções podem ser Controlo, *Bulk*, Interrupção ou isócronas. Passa-se a explicar o funcionamento das transacções do tipo *bulk* dado serem as utilizadas neste projecto.

As transacções do tipo *bulk* começam com o envio de um pacote *token* por parte do anfitrião indicando o sentido da transmissão de dados, seguido dos dados e acabando com um assentimento por parte do receptor da mensagem, caso esta tenha sido recebida sem erros.

2.5. Protocolo de comunicação entre módulos

2.5.1. Introdução

Como já foi referido anteriormente, os módulos comunicam entre si através do barramento CAN. Uma vez que o sistema é composto por vários módulos e este barramento não garante sequencialidade das mensagens, torna-se necessário introduzir um sistema de etiquetas temporais que acompanhará todas as mensagens para que estas possam ser ordenadas correctamente. Para que o sistema tenha uma precisão de centésimos de segundo, decidiu-se que essa etiqueta temporal deveria corresponder ao número de centésimos de segundo passados desde o início da contagem até ao momento da amostragem do sinal. Uma vez que os micro-controladores têm uma arquitectura de 16 bits, decidiu-se ainda que esta seria também a dimensão da etiqueta temporal de forma a simplificar o processo, o que significa que a etiqueta pode contar até 10 minutos e 55 segundos antes de ocorrer um *overflow*. Na realidade esta situação nunca se dá porque o módulo de transmissão e

envio de dados por rádio envia, com intervalos inferiores, mensagens de sincronização de barramento que reinicializam a contagem.

2.5.2. Eficiência do barramento CAN

O tamanho de uma trama de dados CAN é de 44 bits mais 8 bits por cada byte de dados que se pretende enviar. A eficiência de utilização η do barramento cresce com o número de bits de dados n_{dados} que se envia em cada trama por (2) com os resultados da Tabela 2.

$$\eta = \frac{n_{dados}}{n_{total}} \quad (2)$$

Tabela 2 – Eficiência do barramento CAN em função do número de bytes de dado.

Bytes de dados	Total de bits	Eficiência
0	44	0,00%
1	52	15,38%
2	60	26,67%
3	68	35,29%
4	76	42,11%
5	84	47,62%
6	92	52,17%
7	100	56,00%
8	108	59,26%

2.5.3. Mensagens

Uma vez que os dados dos sensores que se pretendem enviar são maioritariamente variáveis de 16 bits, houve necessidade de decidir se as tramas deveriam ser de 8 bytes e transportar dados de 3 sensores mais a etiqueta temporal ou se cada trama deveria ter apenas 4 bytes, transportando apenas dados de um sensor e a etiqueta temporal. Para que no futuro não seja necessário alterar o protocolo caso se pretendam fazer alterações na constituição do sistema e para simplificar os programas de armazenamento e envio de dados por rádio (*ZigBee*, na implementação presente), decidiu-se que devia existir uma mensagem para cada sensor, obtendo-se assim mensagens de 4 bytes, 2 de dados e 2 de etiqueta temporal. Estipularam-se então segundo a Tabela 3 as mensagens do protocolo definido.

2.5.4. Taxa de ocupação do barramento

De maneira a haver uma percepção da possibilidade de acrescentar mais sensores ou de aumentar as frequências de amostragem dos já existentes, foi realizada uma análise da taxa de ocupação do barramento. Tendo em conta que cada trama de dados utilizada tem uma dimensão $D = 76$ bits e que a frequência de amostragem f_i de cada sensor, é possível determinar por (3) a taxa de transmissão de dados T .

$$T = D \cdot \sum_i^n f_i \quad (3)$$

A aplicação de (3) ao protocolo definido na Tabela 3 resulta numa taxa de transmissão de dados $T = 25,69 \text{ kb/s}$, que num barramento capaz de 1 Mb/s perfaz uma taxa de ocupação de $2,57\%$.

Tabela 3 – Frequência de amostragem de cada sensor.

ID	Descrição ³		Frequência [Hz]
0	Sincronização		-
32	Velocidade DD		10
33	Velocidade DE		10
34	Velocidade TD		10
35	Velocidade TE		10
36	LVDT DD		20
37	LVDT DE		20
38	LVDT TD		20
39	LVDT TE		20
41	RPM		10
42	TPS		20
43	ECT		1
44	Neutro		-
45	Ângulo direcção		20
51	IR DD _i		5
52	IR DD _e		5
53	IR DE _i		5
54	IR DE _e		5
55	IR TD _i		5
56	IR TD _e		5
57	IR TE _i		5
58	IR TE _e		5
61	Pressão DD	Temperatura DD	1
62	Pressão DE	Temperatura DE	1
63	Pressão TD	Temperatura TD	1
64	Pressão TE	Temperatura TE	1
71	Aceleração X		20
72	Aceleração Y		20
73	Aceleração Z		20
74	Orientação X		20
75	Orientação Y		20
76	Orientação Z		20
81	Humidade Ar		1
82	Temperatura Ar		1
83	Pressão Atmosférica		1

Tanto a mensagem do sensor de neutro como a de sincronização não têm uma frequência definida pois não são periódicas, isto é, são mensagens enviadas quando ocorre um determinado evento que não tem um período bem definido.

³ XY_Z – X (Dianteira; Traseira); Y (Direita; Esquerda); Z (interior; exterior)

2.6. Módulos de aquisição e processamento digital

2.6.1. Introdução

Os módulos de aquisição e processamento da Figura 14 são responsáveis por adquirir os sinais dos sensores e efectuar o processamento necessário para que estes sejam inteligíveis. De maneira a reduzir os custos de produção e o tempo de desenvolvimento, estes módulos devem ser todos iguais e capazes de efectuar qualquer tarefa de processamento de dados que seja necessária por qualquer um dos sistemas. Devem ainda ter a capacidade de comunicar no barramento CAN de forma a poderem enviar informação útil ao módulo de armazenamento. O projecto deste módulo deve ainda permitir que estes sejam acoplados a outros módulos de maneira a adicionar novas funcionalidades de forma prática e compacta. O módulo DSPICFST foi inicialmente desenvolvido pelo Prof. Moisés Piedade tendo especial cuidado com os aspectos de compatibilidade electromagnética e a convivência de sinais analógicos e digitais na mesma placa. Posteriormente, este módulo foi alvo de três revisões sucessivas realizadas pelo aluno Vítor Almeida do Mestrado em Engenharia Electrónica do IST para o ProjectoFST. Está em desenvolvimento uma nova versão do módulo processador, MSPFST, que é baseado no micro-controlador MSP 430 da Texas Instruments [11].



Figura 14 – Módulo de aquisição e processamento de sinais.

2.6.2. Componentes do Módulo Processador DSPICFST

2.6.2.1. Micro-controlador

Para esta tarefa foi escolhido o micro-controlador *dsPIC30F6012A* [12] da *Microchip* (ver Figura 15). Este micro-controlador da família *dsPIC* (*digital signal peripheral interface controller*) foi escolhido por já possuir *hardware* específico para os barramentos *CAN*, *SPI*, *I²C* e *RS232*, 16 portas para conversão analógico-digital e cálculo de frequências de ondas quadradas, bem como um ritmo de processamento que pode atingir as 30 MIPS (milhões de instruções por segundo).

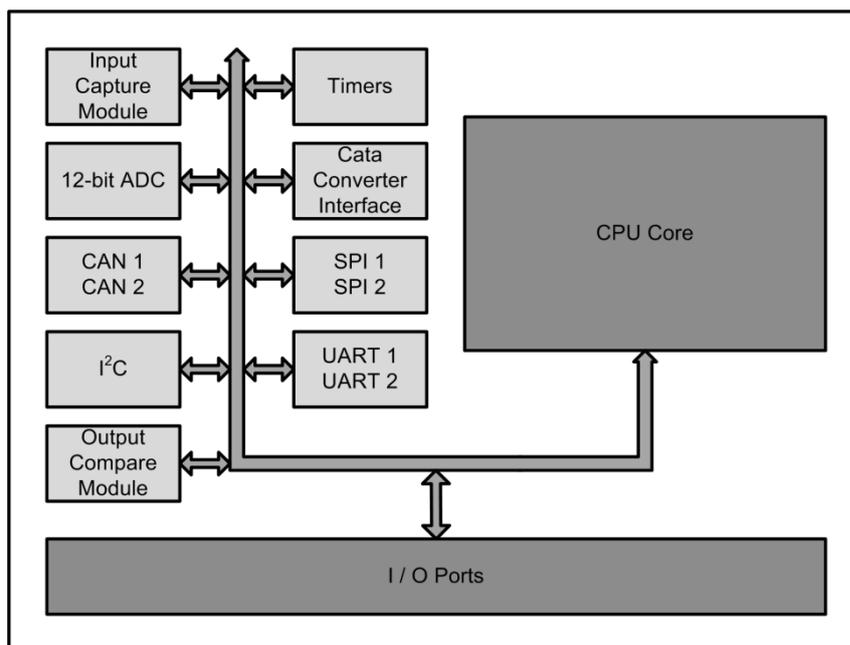


Figura 15 – Diagrama da arquitectura do micro-controlador *dsPIC30F6012A*.

2.6.2.2. Alimentação

Uma vez que o micro-controlador tem uma tensão de alimentação de $5V$ e a única fonte de alimentação disponível no veículo é a bateria de $12V$, foi necessário introduzir um circuito regulador de tensão para alimentação do *dsPIC*. Uma vez que cada uma destas placas é também responsável pela alimentação de módulos de expansão que estejam ligados a esta ou de sensores que esteja a adquirir, este regulador deve possuir uma boa capacidade de corrente para que não existam limitações quanto aos sistemas que com elas se possam realizar. Para tal foi escolhido o regulador de tensão LM2673-5.0 [13], um circuito com tensão de entrada V_i que pode variar entre $8V < V_i < 40V$ e que disponibiliza uma tensão à saída de $V_o = 5V$, com uma capacidade de corrente máxima de $3A$ e uma eficiência $\eta = 88\%$.

2.6.3. Programação

A programação destes micro-controladores pode ser feita de várias formas, mas devido à quantidade de informação disponível, à qualidade das ferramentas e capacidade de integração entre si, decidiu-se utilizar as ferramentas da Microchip. Estas são o compilador C30, um compilador de linguagem C específico para processadores desta família, o ambiente integrado de desenvolvimento MPLAB IDE e o programador e depurador de código MPLAB ICD2.

2.6.4. Arquitectura do programa de interface com os sensores

O programa do micro-controlador é responsável pela inicialização dos vários módulos necessários, pela aquisição dos sinais dos sensores, o seu processamento e pelo envio dos dados pelos barramentos CAN.

O programa desenvolvido consiste numa função principal *main* que é responsável pela inicialização de algumas variáveis e dos módulos, incluindo o temporizador responsável pelo tempo de amostragem, entrando depois em *loop* infinito. Quando o temporizador, configurado na *main*, chega ao valor de tempo pretendido para amostragem (período base T_b) este chama uma rotina de

interrupção `_T1Interrupt` na qual é feita a aquisição e envio por CAN dos dados adquiridos dos sensores. De forma a não sobrecarregar o barramento com um grande número de mensagens num curto espaço de tempo deixando-o depois inactivo por um grande espaço de tempo, decidiu-se que a leitura dos sensores não seria simultânea, ou seja, apenas um sensor é adquirido a cada chamada de `_T1Interrupt` havendo uma variável de estado $cont_p$ que é incrementada até que todos os sensores tenham sido lidos e esta atinja um determinado valor limite N . Desta forma é possível uma leitura de cada sensor com um período mínimo de $T_{min} = T_b \times N$, sendo ainda possível especificar, através de uma outra variável P para cada sensor, quantos T_{min} devem passar entre cada aquisição desse sensor, resultando o período de cada sensor num múltiplo do período mínimo T_{min} .

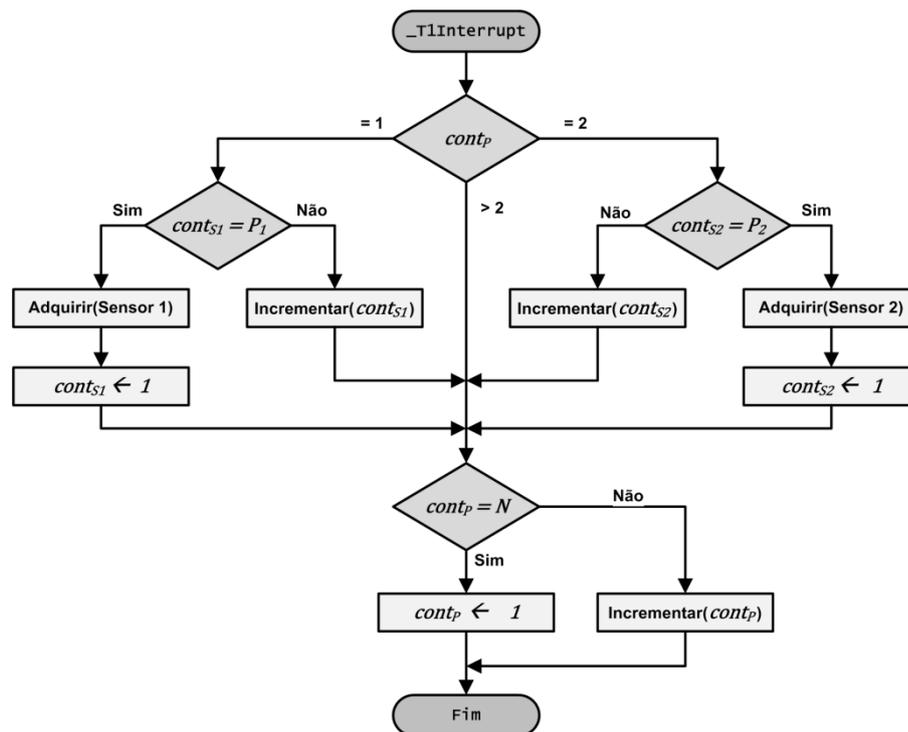


Figura 16 – Fluxograma exemplo da rotina de interrupção do temporizador de aquisição.

Pode então calcular-se a frequência de cada sensor i a partir de (4).

$$f_i = \frac{1}{T_b \times N \times P_i} \quad (4)$$

Como o número máximo de sensores é 6 e a máxima frequência de amostragem pretendida é de 10 Hz decidiu-se utilizar $N = 10$ com $T_b = 10$ ms.

2.6.4.1. Aquisição de sinais analógicos

A aquisição de sinais analógicos faz-se recorrendo ao módulo de conversão analógica-digital do micro-controlador. Este é baseado num conversor de aproximações sucessivas de 12 bits que pode ser utilizado em 16 das portas do micro-controlador e tem uma frequência de amostragem máxima de 200 ksp/s. Uma vez que todos os sensores são desenhados para saídas proporcionais a 5 V, esta é também a tensão de referência do conversor. Para realizar uma amostragem basta então chamar a função `adc12_start(int ch)` em que `ch` é a porta que se pretende adquirir. Esta

função está definida nos ficheiros *ADC12bits.c* e *ADC12bits.h* e para ser utilizada deve primeiro ser chamada uma vez a função de configuração *adc12_init* definida nos mesmos ficheiros.

2.6.4.2. Cálculo da frequência de uma onda quadrada

Para calcular a frequência de ondas quadradas é utilizado o módulo de *Input Capture*. Trata-se de um módulo que pode ser configurado para automaticamente armazenar num *buffer* dedicado o valor de um temporizador quando é detectado um flanco da onda quadrada na entrada. Pode então ser calculada a frequência dessa onda computando a diferença entre duas entradas desse *buffer* e sabendo a unidade de tempo do temporizador T_{min} .

O módulo pode ser configurado para armazenar o valor do temporizador e chamar uma rotina de interrupção a cada flanco ascendente do sinal. Nesta rotina calcula-se a diferença entre dois valores consecutivos do temporizador B e armazena-se num *buffer* circular para posterior aplicação de filtros se necessário. Tendo em conta o número de *overflows* $N_{overflows}$ ocorridos no temporizador entre os dois flancos ascendentes é possível calcular a frequência f do sinal adquirido a partir de (5).

$$f = \frac{1}{B_i - B_{i-1} + N_{overflows} \ll 16 T_{min}} \quad (5)$$

2.6.4.3. Comunicação através de SMBus

A comunicação SMBus é feita com recurso ao módulo *I²C* do micro-controlador configurado para funcionamento em modo SMBus.

Uma vez que o fabricante dos dispositivos disponibiliza bibliotecas de funções para controlar o módulo de *I²C* nas operações mais comuns da Tabela 4, estas serão utilizadas na criação de funções mais complexas responsáveis pela comunicação.

Tabela 4 – Funções disponíveis na biblioteca *i2c.h*.

Função	Descrição
<code>void AckI2C(void)</code>	Gera uma condição de assentimento no barramento
<code>void CloseI2C(void)</code>	Desliga o módulo de <i>I²C</i>
<code>void IdleI2C(void)</code>	Espera até que o barramento esteja em modo <i>Idle</i>
<code>unsigned char</code>	Leitura um byte do barramento
<code>void MasterWriteI2C(unsigned char data_out)</code>	Escrita de um byte no barramento
<code>void NotAckI2C(void)</code>	Gera uma condição de não assentimento no barramento
<code>void RestartI2C(void)</code>	Gera uma condição de reinicialização do barramento

2.6.4.4. Comunicação através do barramento CAN

Como já foi referido anteriormente, o micro-controlador escolhido é detentor de um módulo de processamento CAN capaz de tratar de toda a implementação da camada física do barramento.

Antes de ser utilizado, o módulo de CAN deve ser devidamente configurado. Começa-se por colocar o módulo em modo de configuração, de seguida programa-se o módulo de forma a obter um *bitrate* de *1 Mb/s* e configura-se o sistema de filtros e máscaras para que sejam recebidas apenas as

mensagens pretendidas. Depois de completa a configuração do módulo de CAN, este deve ser novamente colocado no modo de funcionamento normal e está pronto para comunicar.

2.6.4.4.1. Configuração do ritmo de transmissão

O ritmo de transmissão é calculado com base em vários parâmetros configuráveis. Tudo se inicia com a definição da base de tempo T_Q pois os restantes parâmetros múltiplos desta. Os parâmetros que devem então ser configurados estão representados na Figura 17 e descritos na Tabela 5.

Tabela 5 - Descrição das componentes configuráveis de um bit.

Va	Nome	Descrição
SJ	Sincro	Tempo reservado a sincronia.
P	Propa	Tempo reservado à propagação do sinal por todo o
S	1º	Tempo do primeiro segmento do bit
S	2º	Tempo do segundo segmento do bit

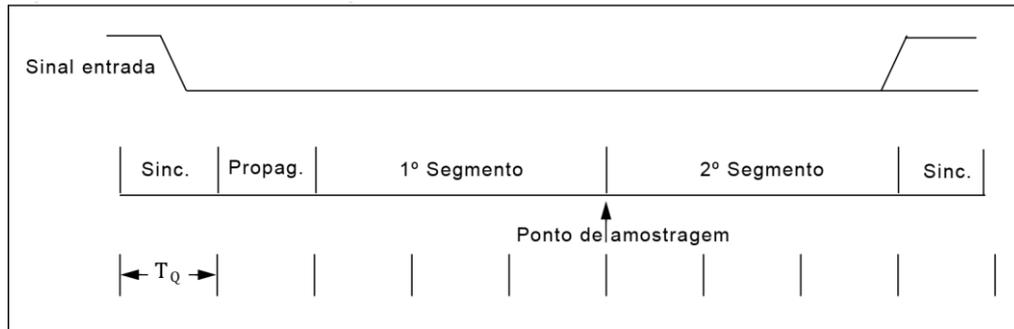


Figura 17 – diagrama das partes constituintes de um bit.

Como se pode ver pela Figura 17, o tempo total do bit T_b é o somatório de todos os tempos e a amostragem do bit é feita no instante entre os dois segmentos, ou seja o tempo de amostragem pode ser escrito como uma percentagem do tempo total de bit pela expressão (7).

Uma vez que é utilizada uma frequência de relógio $FCY = 5 \text{ MHz}$ no micro-controlador, e que o fabricante fornece a expressão $T_Q = \frac{1}{2 \text{ FCY}} = 100 \text{ ns}$, se utilizarmos $SJW = 1$, $PRSEG = 1$, $SEG1PH = 5$ e $SEG2PH = 3$ obtém-se o tempo de bit $T_b = 10 T_Q$, que por (8) nos permite calcular o *bitrate*.

$$T_b = SJW + PRSEG + SEG1PH + SEG2PH T_Q \quad (6)$$

$$T_s = \frac{SJW + PRSEG + SEG1PH}{SJW + PRSEG + SEG1PH + SEG2PH} \quad (7)$$

$$NBR = \frac{1}{T_b} = 1 \text{ Mb/s} \quad (8)$$

2.6.4.4.2. Configuração das máscaras e dos filtros de mensagem

É ainda necessário configurar o sistema de máscaras e filtros para que sejam recebidas apenas as mensagens pretendidas. Este sistema é útil nos casos em que a capacidade de

processamento seja crítica pois permite descartar imediatamente mensagens indesejadas por hardware. Como não é esse o caso, os filtros e máscaras são programados para receberem todas as mensagens, sendo estas posteriormente tratadas por software.

Quando é recebida uma mensagem, a aplicação da máscara ao identificador resulta em quais os bits que devem de ser tidos em conta na filtragem, devendo esses bits ser iguais aos correspondentes do filtro para que a mensagem seja dada como aceite. Em suma, a aceitação de mensagens pelo sistema de máscaras e filtros resume-se à Tabela 6.

Tabela 6 – Tabela de verdade do sistema máscara e filtro.

Máscara	Filtro	Identificador	Aceite/Rejeitado
0	X	X	Aceite
1	0	0	Aceite
1	0	1	Rejeitado
1	1	0	Rejeitado
1	1	1	Aceite

Como se pretende que todas as mensagens sejam aceites, a máscara foi toda programada 0.

Para configuração do módulo CAN foi desenvolvida a função `can1_init` que está definida nos ficheiros `can.c` e `can.h` e automaticamente configura o módulo de *hardware* CAN para as configurações previamente especificadas.

2.6.4.4.3. Envio de mensagens CAN

O envio de mensagens CAN é feito através dos *buffers* de saída. Os parâmetros da mensagem são escritos em *buffers* e é accionado um sinal que dá a ordem de envio da mensagem. O envio da mensagem é então tratado pelo módulo de hardware que está atento ao barramento e quando detecta a sua disponibilidade dá início ao envio da mensagem.

O módulo de hardware contém 3 *buffers* de saída, sendo eles idênticos. No caso de um primeiro estar já a ser utilizado, deve ser utilizado um dos outros até que todos estejam cheios.

Para efectuar o envio de mensagens no barramento CAN foi desenvolvida a função `can1_tx_std` dos ficheiros `can.c` e `can.h`. Esta função, que recebe como parâmetros o identificador da mensagem, o número de *bytes* de dados a enviar e um vector que contém esses *bytes*, configura automaticamente os registos de um *buffer* de saída que esteja disponível ou retorna um erro que indica que os *buffers* de saída já estão a ser utilizados.

2.6.4.4.4. Recepção de mensagens CAN

Como já foi referido anteriormente, a recepção de mensagens CAN é efectuada pelo módulo de *hardware*. Este módulo contém três *buffers* de recepção, sendo um deles responsável pela recepção de todas as mensagens. Essas mensagens são posteriormente enviadas para o sistema de máscaras e filtros que aceita, ou não, a nova mensagem. O sistema pode ser programado de forma que quando o primeiro *buffer* está cheio as mensagens são enviadas para o segundo.

Quando uma mensagem CAN é recebida e aceite, o módulo de *hardware* CAN activa uma *flag* que acciona a rotina de interrupção `_C1Interrupt` que está também definida nos ficheiros *can.c* e *can.h*.

2.6.5. Consumo de energia

O módulo DSPICFST, quando programado para funcionar como Interface de Sensores (IS-1 e IS-2 de 2.1) consome uma corrente média de $I_m = 24,2mA$ a uma tensão de alimentação de $V_{BAT} = 12V$, o que perfaz um consumo de $P = I_m \times V_{BAT} = 290,4mW$.

2.7. Sistema de interface USB-CAN

2.7.1. Introdução

Para o desenvolvimento deste projecto com o barramento CAN surge a necessidade de um dispositivo que permita analisar o que se passa no barramento e possibilite uma interacção directa com este. Uma vez que as ferramentas comerciais que permitem efectuar estas funções, como o USB-8473 da *National Instruments*, têm um custo elevado (acima dos 300 € a versão mais barata), decidiu-se desenvolver um sistema completo com funcionalidades semelhante.

2.7.2. Objectivo

O objectivo deste sistema é efectuar a ponte entre o barramento CAN e um PC através do barramento USB, sem alimentação externa de modo a permitir comunicação bidireccional com total flexibilidade. Isto permitirá analisar os dados que os módulos de aquisição de dados transmitem para o barramento CAN, simular a existência de módulos de aquisição que na realidade não estão presentes no sistema e controlar o funcionamento dos diversos módulos a partir do PC.

2.7.3. Módulo de interface USB-CAN

O sistema desenvolvido é baseado no “Módulo de Interface e Controlo de Testes” de [14], cuja arquitectura está exposta na Figura 18. Apesar da placa de circuito impresso (PCI) da Figura 19 ter sido aproveitada, todo o programa do micro-controlador foi novamente desenvolvido pois era específico para a aplicação que estava a ser utilizado e neste projecto pretende-se desenvolver um sistema geral independente da aplicação, responsável apenas por fazer a ponte entre os barramentos USB e CAN.

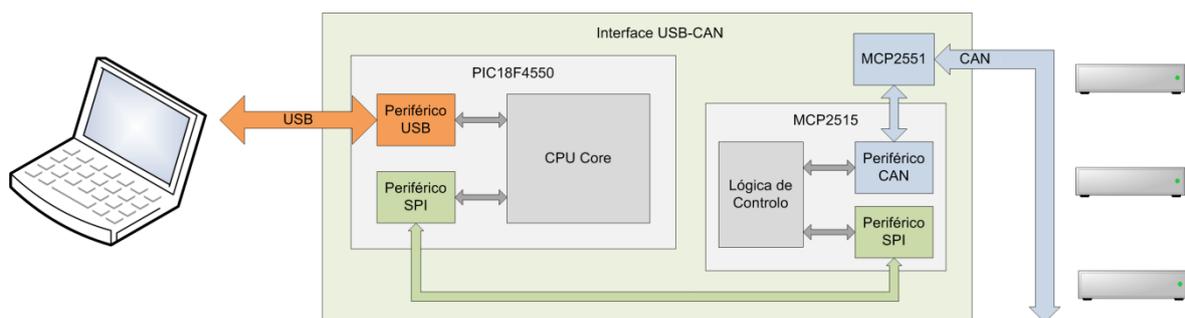


Figura 18 – Diagrama do sistema de Interface USB-CAN.

O sistema é baseado no micro-controlador *PIC18F4550* [15]. Este micro-controlador possui um módulo interno de USB capaz de trabalhar a *12 Mbps* e para o qual o fabricante *Microchip* já disponibiliza o programa e uma biblioteca DLL de funções para comunicação com este a partir de um PC. Para efectuar a ligação ao barramento CAN, recorre-se ao controlador CAN *MCP2515* [16] pois possui uma interface SPI que, quando ligada ao periférico SPI do micro-controlador, permite efectuar a transformação desejada de USB para CAN e vice-versa.

É ainda necessário recorrer a um transceptor de CAN, neste caso o *MCP2551* [17], para transformar os níveis de tensão do controlador *MCP2515* para os níveis correctos definidos na especificação do barramento CAN.

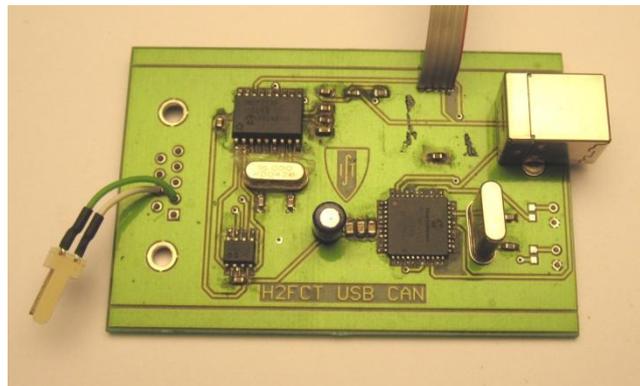


Figura 19 – PCI do sistema de Interface USB-CAN.

2.7.3.1. Comunicação SPI com o controlador MCP2515

O controlador de CAN *MCP2515* disponibiliza uma série de comandos SPI que permitem efectuar a sua configuração e transmitir e receber mensagens do barramento CAN. Estes comandos funcionam à base de escritas e leituras na memória *Random Access Memory* (RAM) do controlador e foi escrita uma função para a realização de cada um num *PIC18F4550*. Estas funções encontram-se nos ficheiros *mcp2515_v2.c* e *mcp2515.h* e de seguida explica-se as funcionalidades de cada uma.

```
BYTE CAN2515ReadByte(BYTE reg)
```

- Leitura do registo *reg*.

```
void CAN2515WriteByte(BYTE reg, BYTE data)
```

- Escrita de *data* no registo *reg*.

```
void CAN2515BitModify(BYTE reg, BYTE mask, BYTE data)
```

- Modificação dos bits do registo *reg* seleccionados em *mask* pelos de *data*.

```
void CAN2515Reset(void)
```

- Reinicialização do controlador *MCP2515*.

void CAN2515Request2Send(BYTE buffers)
 - Pedido de transmissão da mensagem CAN armazenada nos buffers.

void CAN2515LoadArray(BYTE start, BYTE length, BYTE *vect)
 - Escrita dos *length* registos seguintes ao registo *start* para o vector *vect*.

BYTE CAN2515ReadStatus(void)
 - Leitura do estado do controlador.

void CAN2515ReadArray(BYTE start, BYTE length, BYTE *vect)
 - Leitura dos *length* registos seguintes ao registo *start* para o vector *vect*.

BYTE CAN2515RXStatus(void)
 - Leitura do estado de recepção do controlador.

void CAN2515RecvRXBN(BYTE * msg_recv)
 - Leitura da mensagem do *buffer* de recepção N para o vector *msg_recv*.

void CAN2515SendTXBN(BYTE *msg_send)
 - Escrita da mensagem do vector *msg_send* no buffer de transmissão N.

2.7.3.2. Comunicação USB com um PC

A comunicação com o PC é feita através de um protocolo desenvolvido para permitir que o módulo de interface comunique com o PC de forma robusta. Existem 8 tipos de mensagem que estão descritos na Tabela 7. O programa do micro-controlador verifica ciclicamente a chegada de mensagens enviadas pelo PC e se alguma mensagem for recebida, o programa lida com ela, senão verifica a chegada de mensagens CAN e envia-as para o PC.

Tabela 7 - Tipos de mensagem no protocolo USB.

Nome	Valor	Descrição
READ_VERSION	0x00	Mensagem do PC para o dispositivo a inquirir sobre a versão do <i>firmware</i> programada neste.
PC2CAN_MSG	0x01	Transmissão de uma mensagem para o barramento CAN por parte do PC.
CAN2PC_MSG	0x03	Envio de uma mensagem CAN do barramento para o PC.
READ_STATUS	0x05	Pedido de leitura do estado do controlador de CAN MCP2515 por parte do PC.
SEND_ERROR	0x06	Mensagem enviada do Interface USB-CAN para o PC quando ocorre um erro na transmissão de mensagens para o CAN
RESET	0x07	Envio de uma ordem para reinicialização do controlador de CAN por parte do PC.
SETUP_BITRATE	0x08	Mensagem enviada do PC para o dispositivo com os valores dos registos do controlador de CAN para reconfiguração do <i>bitrate</i> .
UNKNOWN	0xFF	Resposta de uma das partes na eventualidade de recepção de mensagens com identificador desconhecido (possível quando utilizadas versões diferentes).

2.7.4. Programa de interface USB-CAN no PC

O programa USB-CAN que corre no PC foi escrito em linguagem C# utilizando uma ferramenta grátis, o *Microsoft Visual C# 2008 Express*. Com recurso à biblioteca DLL de funções disponibilizadas pela *Microchip*, foi escrito um programa que não só identifica a ligação de um dispositivo de interface como é robusto o suficiente para se poder ligar e desligar este dispositivo durante funcionamento sem qualquer problema.

Ao inicializar, ou quando um dispositivo é ligado numa porta USB, o programa verifica se se trata de um dispositivo de Interface USB-CAN e, se sim, inicializa a comunicação. O utilizador dispõe da lista de mensagens da Figura 20, que podem ser criadas uma de cada vez no programa através do botão *Add* que inicializa o formulário da Figura 21 ou lidas directamente de um ficheiro de texto com extensão “.can”. Um *click* no botão *Send to CAN* descarrega de forma ordenada as mensagens dessa lista para um buffer de saída que é depois despejado por uma *thread* que verifica ciclicamente a entrada de novas mensagens a envia-as via USB para o dispositivo de Interface USB-CAN conforme o protocolo especificado.

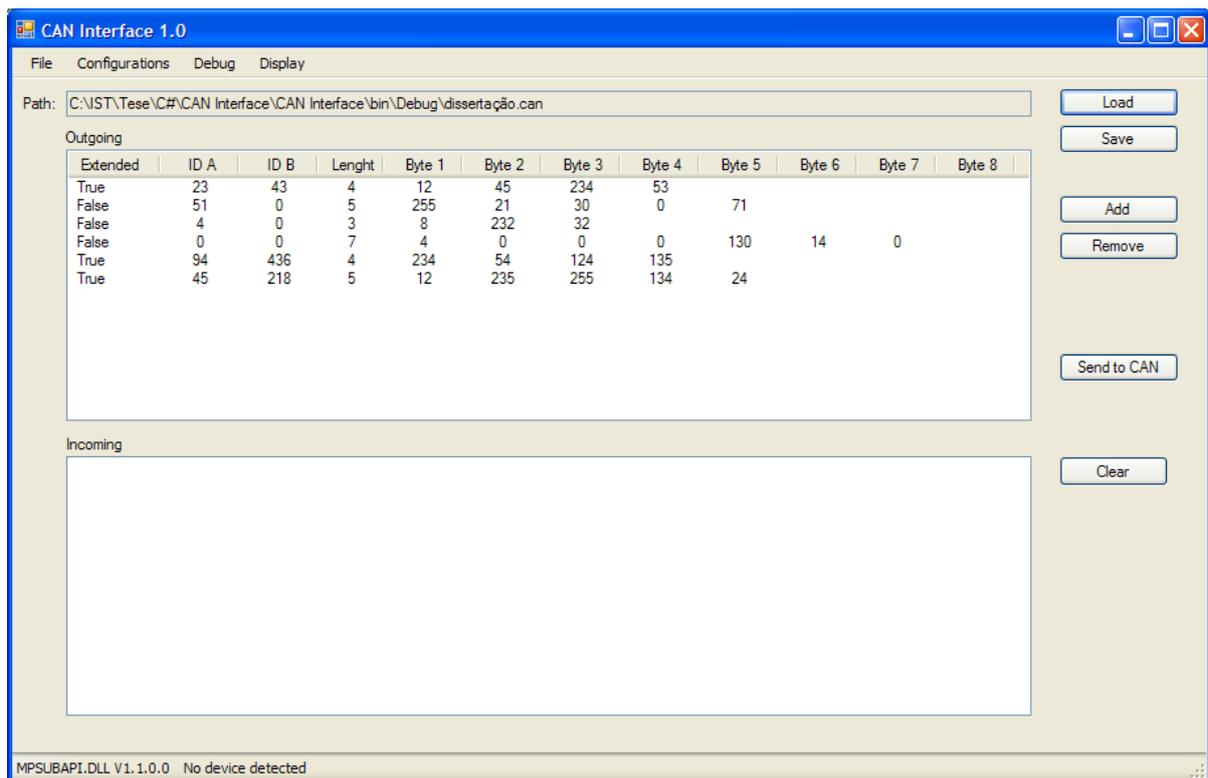


Figura 20 – Interface visual do programa de comunicação USB-CAN.

Figura 21 – Formulário de adição de mensagens CAN à lista de envio.

No que diz respeito à recepção de mensagens CAN via dispositivo de Interface USB-CAN, a mesma *thread* que trata do envio das mensagens para o USB está também constantemente a verificar o USB para a recepção de mensagens, e quando é recebida uma mensagem esta é prontamente enviada para a lista de mensagens recebidas.

2.7.5. Aplicações do sistema

A maneira como este sistema está desenvolvido permite que o PC funcione como apenas mais um nó no barramento CAN, podendo o programa do PC ser alterado para fazer virtualmente tudo o que seja necessário. Isto torna este sistema numa potente ferramenta de desenvolvimento de aplicações CAN bem como numa possível parte essencial de outros projectos.

Durante o desenvolvimento do restante trabalho, este sistema foi utilizado apenas como um analisador de barramento CAN, servindo para verificar se os módulos IS enviavam correctamente as mensagens devidas, mas foi desenvolvido com o propósito de auxiliar o desenvolvimento de futuros projectos que envolvam o barramento CAN no Projecto FST. As suas potencialidades permitem, por exemplo, simular em laboratório uma rede de módulos de aquisição de dados a enviar mensagens CAN sem que estes existam na realidade, sendo para isso apenas necessário alterar o programa do PC.

3. Sistemas sensoriais

3.1. Sensor de deslocamento linear

3.1.1. Introdução

A suspensão de um veículo é o sistema mecânico de maior importância para o seu comportamento dinâmico, sendo portanto essencial a obtenção de dados sobre o seu desempenho para melhor afinação e compreensão da dinâmica do veículo.

Para realizar uma tarefa deste género podem ser usados potenciômetros lineares ou *Variable Differential Transformers (LVDTs)*. Uma vez que os potenciômetros têm um ciclo de vida mais reduzido e os LVDTs mais fáceis de fabricar à medida desejada, foram estes os escolhidos para este trabalho.

3.1.2. Teoria de funcionamento

O LVDT é, como o nome indica, um dispositivo que se baseia no princípio de funcionamento do transformador para calcular um deslocamento.

Na sua configuração convencional, apresentada na Figura 22, um LVDT consiste em três enrolamentos, um primário central a dois secundários de dimensões L no interior dos quais se desloca um núcleo de dimensão $2L$ acoplado a uma haste, resultando num dispositivo de dimensões que variam entre $3L$ e $4L$ para medir um deslocamento entre 0 e L .

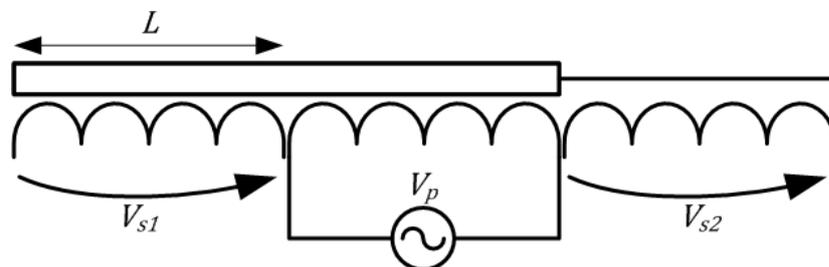


Figura 22- constituição de um LVDT convencional.

No entanto, uma vez que o destino destes dispositivos é um protótipo de competição, no qual o espaço escasseia e o peso é crucial, decidiu-se que um dispositivo de comprimento máximo de $4L$ para medir um deslocamento de L seria inaceitável.

Desenvolveu-se portanto uma configuração de LVDT alternativa, apresentada na Figura 23, que consiste num primário de tamanho $2L$ e dois secundários de tamanho L sobrepostos ao primário, no interior dos quais se desloca um núcleo de tamanho L acoplado a uma haste de tamanho L . Isto permite reduzir o tamanho dos LVDT's de $3L-4L$ para $2L-3L$, mantendo a funcionalidade intacta e reduzindo significativamente o peso uma vez que o núcleo, que representa a maior parte do peso total deste dispositivo, é reduzido para metade.

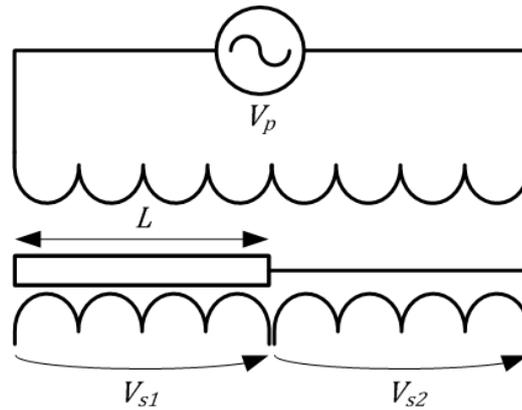


Figura 23 – Constituição do LVDT desenvolvido.

Admitindo que os enrolamentos secundários são exactamente iguais, na condição de não haver núcleo ou de este se encontrar na posição intermédia, ao aplicarmos uma tensão V_p no primário, as tensões V_{s1} e V_{s2} nos secundários são semelhantes. Como a permeabilidade magnética relativa do ferro é de aproximadamente $\mu_r = 700$, ou seja, setecentas vezes maior que a do ar, isto significa que ao deslocarmos o núcleo de um dos secundários para o outro, por exemplo do secundário 1 para o secundário 2, a ligação magnética entre o primário e o secundário 1 aumenta, enquanto a do primário com o secundário 2 diminui. Isto significa que a tensão V_{s1} aumenta enquanto V_{s2} diminui. A ligação magnética depende da porção de enrolamento secundário e primário que estão acoplados, isto é: da distância que o núcleo de ferro acopla os dois enrolamentos, uma vez que o resto dos enrolamentos apenas são acoplados por ar. Desta forma podemos concluir que a relação entre o deslocamento e as tensões nos secundários é dada por (9), sendo K uma constante relativa à sensibilidade do LVDT.

$$X(V_{s1}, V_{s2}) = K \frac{V_{s1} - V_{s2}}{V_p} \quad (9)$$

3.1.3. Circuito de excitação do primário

A excitação do enrolamento primário é feita através de um oscilador em Ponte de Wien [18] da Figura 24. Como o enrolamento primário apresenta uma baixa impedância, o amplificador operacional utilizado para implementar a Ponte de Wien foi o LM386 [19], originalmente desenvolvido para aplicações de áudio.

3.1.3.1. Ponte de Wien simples

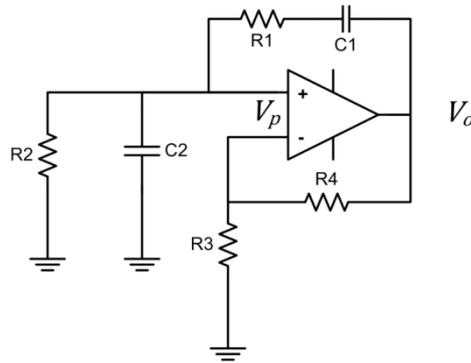


Figura 24 – Ponte de Wien.

O circuito pode ser analisado [18] como tendo um ganho de malha aberta $A(s)$ correspondente à montagem inversora e um ganho de retroacção $B(s)$ correspondente ao restante circuito, assim:

$$A(s) = \frac{V_o}{V_p} = 1 + \frac{R_4}{R_3} \quad (10)$$

$$B(s) = \frac{V_p}{V_o} = \frac{\frac{s}{R_1 C_2}}{s^2 + \frac{3}{R_1 C_2} s + \frac{1}{R_1 C_2 R_2 C_1}} \quad (11)$$

Fazendo $R_1 = R_2 = R$ e $C_1 = C_2 = C$ e tendo em conta a frequência (12), de (11) conclui-se (13)

$$\omega_0^2 = \frac{1}{R_1 C_2 R_2 C_1} \quad (12)$$

$$B(s) = \frac{\omega_0 s}{s^2 + 3\omega_0 s + \omega_0^2} \quad (13)$$

$$B(j\omega_0) = \frac{1}{3} \quad (14)$$

A partir do critério de Barkhausen (15), conclui-se que para existir oscilação $A(j\omega_0)$ deve ser igual a 3.

$$A(s)B(s) = 1 \quad (15)$$

3.1.3.2. Ponte de Wien com estabilização de amplitude

Para que o circuito descrito anteriormente entre em oscilação é necessário $A(s) > 3$ durante o início das oscilações. Para tal é necessário fazer algumas alterações ao circuito, resultando no circuito da Figura 25.

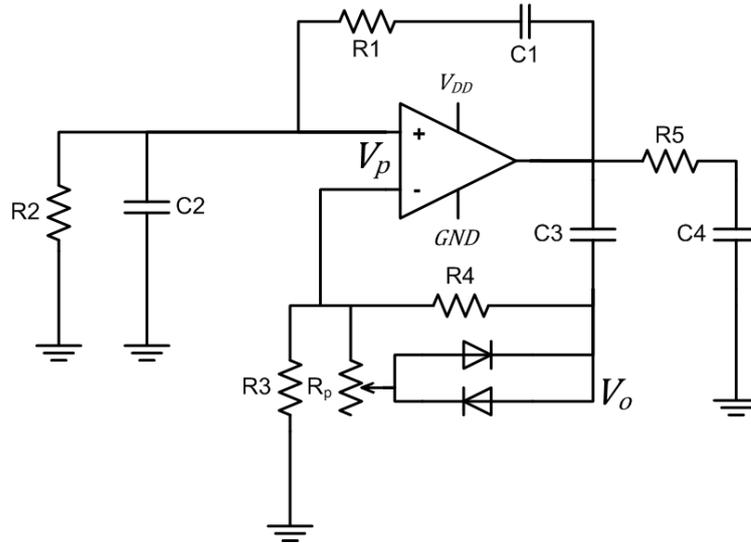


Figura 25 – Circuito ponte de Wien com estabilização de amplitude.

No caso do circuito da Figura 25, o ganho A é dado por (16). Quando não existem oscilações $R_D \rightarrow \infty$, sendo o ganho $A(0) = 1 + R_4/R_3 > 3$, ou seja, o circuito é instável e dá-se início à oscilação. A resistência R_D começa então a decrescer, reduzindo assim o ganho até que este seja $A(j\omega) = 3$ e a amplitude das oscilações estabilize.

$$A(j\omega) = 1 + \frac{(R_p + R_D) \parallel R_4}{R_3} \quad (16)$$

Para calcular a amplitude das oscilações estabilizadas, a resistência incremental dos díodos (17) foi manipulada de forma a obter-se a característica de V_D (18).

$$R_D = \frac{V_T}{I_D} = \frac{V_T}{I_S \left(e^{\frac{V_D}{nV_T}} - 1 \right)} \quad (17)$$

$$V_D = nV_T \ln \left(\frac{V_T}{I_S R_D} + 1 \right) \quad (18)$$

Reescrevendo a equação (16) de forma a isolar a resistência R_D obtém-se (19), que permite calcular o valor de resistência incremental dos díodos para que $A(j\omega) = 3$ e as oscilações estabilizem. Esse valor, quando utilizado em (18), permite o cálculo de V_D , que através de (20) permite determinar a amplitude das oscilações estabilizadas.

$$R_D = \frac{R_4 R_p - R_3 (A - 1) (R_4 + R_p)}{R_3 (A - 1) - R_4} \quad (19)$$

$$V_O = I_D R_p + V_D + R_3 \left(\frac{I_D R_p + V_D}{R_4} + I_D \right) \quad (20)$$

No caso de um díodo com $I_s = 10^{-12} mA$ e $n = 1$, a amplitude das oscilações muda com a resistência variável do potenciómetro segundo o gráfico da Figura 26 num circuito com $R_1 = R_2 = 10k\Omega$, $C_1 = C_2 = 1nF$, $R_3 = 150\Omega$ e $R_4 = 600\Omega$.

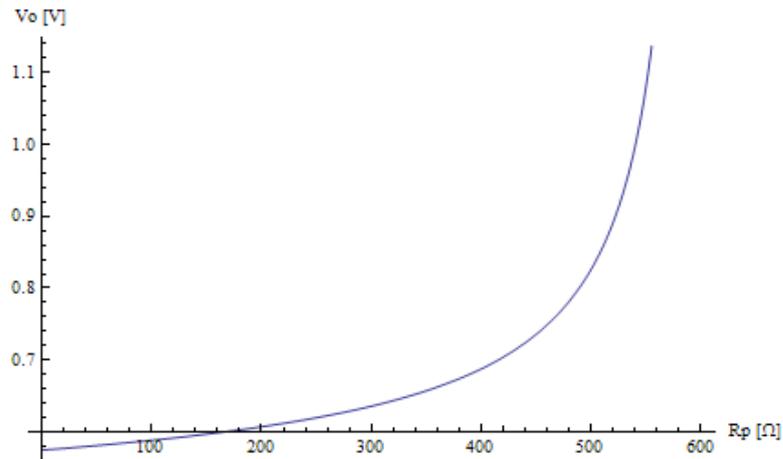


Figura 26 – Característica da amplitude das oscilações com o valor do potenciómetro.

3.1.4. Condicionamento de sinal

Como ficou definido em 2.3.1, pretende-se que a saída do sistema LVDT seja uma tensão CC entre 0 V e 5 V proporcional ao deslocamento do núcleo. Como foi referido em 3.1.2, o deslocamento do núcleo é proporcional à diferença das diferenças de tensão dos secundários, o que pode ser feito com uma montagem amplificadora substractora. No entanto, a desfasagem dos sinais dos secundários impõe algum tipo de tratamento prévio destes, e neste caso decidiu-se fazer uma conversão CA/CC antes de subtrair. Para tal foi implementado um rectificador de meia onda seguido de filtragem passa-baixo por um filtro Sallen&Key [20] de segunda ordem. No final, como a subtracção resulta em valores que variam entre -2,5 V e +2,5 V, foi somada uma tensão constante de 2,5 V.

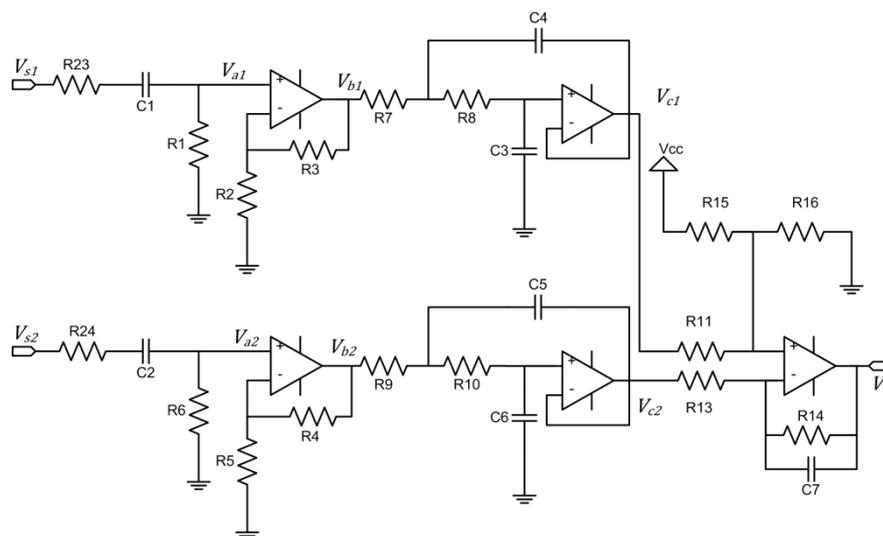


Figura 27 – Circuito de condicionamento de sinal para um LVDT.

A primeira parte do circuito é um filtro passa-alto com a função de transferência de (21) introduzido para filtrar qualquer componente CC ou componentes de ruído de menor frequência. Pretende-se ainda um ganho de 1/2 para proteger as entradas do amplificador operacional de tensões demasiado elevadas.

$$\frac{V_{a1}}{V_{s1}}(s) = K \frac{s}{s + \omega_{p1}} \quad (21)$$

$$K = \frac{R_1}{R_1 + R_{23}} \quad (22)$$

$$\omega_{p1} = \frac{1}{C_1(R_1 + R_{23})} \quad (23)$$

A partir de (22) sabe-se que para obter um ganho de $K = \frac{1}{2}$ devemos utilizar $R_1 = R_{23} = R$, e como não há necessidade de muita precisão na frequência de corte, foram escolhidos componentes *standard* $R = 1 \text{ k}\Omega$ e $C_1 = 18 \text{ nF}$, com os quais se obtém uma frequência de corte $f = \frac{\omega_{p1}}{2\pi} \approx 4,4 \text{ kHz}$.

A segunda secção do circuito consiste num amplificador não inversor, e uma vez que se trata de um amplificador *rail-to-rail* alimentado a 5 V, resulta num amplificador rectificador de meia onda com um ganho $G_r = 1 + \frac{R_4}{R_3}$.

Passa-se de seguida à filtragem passa-baixo de forma a obter-se a média do sinal e para tal é utilizada uma secção passa-baixo Sallen&Key de 2ª ordem com a função de transferência de (26).

$$\omega_c^2 = \frac{1}{C_3 C_4 R_7 R_8} \quad (24)$$

$$Q = \frac{\sqrt{C_3 C_4 R_7 R_8}}{C_3 (R_7 + R_8)} \quad (25)$$

$$\frac{V_{c1}}{V_{b1}}(s) = \frac{\omega_c^2}{s^2 + \frac{\omega_c}{Q}s + \omega_c^2} \quad (26)$$

Finalmente temos o amplificador somador e subtrator que pode ser analisado através do teorema da sobreposição de fontes. Como o circuito apresenta três fontes de tensão, a sinal de saída é composto também por três componentes.

$$V_o(CC) = V_{DD} \frac{R_{11} \parallel R_{16}}{R_{15} + R_{11} \parallel R_{16}} \left(1 + \frac{R_{14}}{R_{13}} \right) \quad (27)$$

$$V_o(V_{c1}) = V_{c1} \frac{R_{15} \parallel R_{16}}{R_{11} + R_{15} \parallel R_{16}} \left(1 + \frac{R_{14}}{R_{13}} \right) \quad (28)$$

$$V_o(V_{c2}) = -\frac{R_{14}}{R_{13}} \quad (29)$$

Tendo em conta estas características, escolheram-se os seguintes valores para os componentes, que resultam num ganho do amplificador rectificador $G_r = 6.56$, uma frequência de

corde de $f_c = 60 \text{ Hz}$ no filtro, um offset de $2,5 \text{ V}$ no somador e um ganho de 3 nas entradas do amplificador subtrator.

$$\begin{aligned}
 R_1 = R_6 = R_{23} = R_{24} &= 1 \text{ k}\Omega \\
 R_2 = R_5 &= 18 \text{ k}\Omega \\
 R_3 = R_4 = R_7 = R_8 = R_9 = R_{10} &= 100 \text{ k}\Omega \\
 R_{11} = R_{13} &= 10 \text{ k}\Omega \\
 R_{14} &= 30 \text{ k}\Omega \\
 R_{15} = R_{16} &= 60 \text{ k}\Omega \\
 C_1 = C_2 = C_3 = C_6 &= 18 \text{ nF} \\
 C_4 = C_5 &= 39 \text{ nF} \\
 C_7 &= 0,1 \text{ }\mu\text{F}
 \end{aligned}$$

3.1.5. Consumo de energia

Os circuitos de excitação do primário e condicionamento de sinal consomem uma corrente média de $I_m = 9,7 \text{ mA}$, que à tensão de alimentação utilizada de $V_{DD} = 5 \text{ V}$ perfaz um consumo de energia de $P = I_m \times V_{DD} = 48,5 \text{ mW}$.

3.1.6. Característica real e cálculo teórico do deslocamento

Devido às diferenças de construção entre os vários LVDT's e às tolerâncias dos componentes do circuito de excitação e condicionamento de sinal, foi decidido que seriam obtidas as características de todos os pares LVDT - Condicionamento como na Figura 28, uma vez que estes apresentam diferenças significativas entre si.

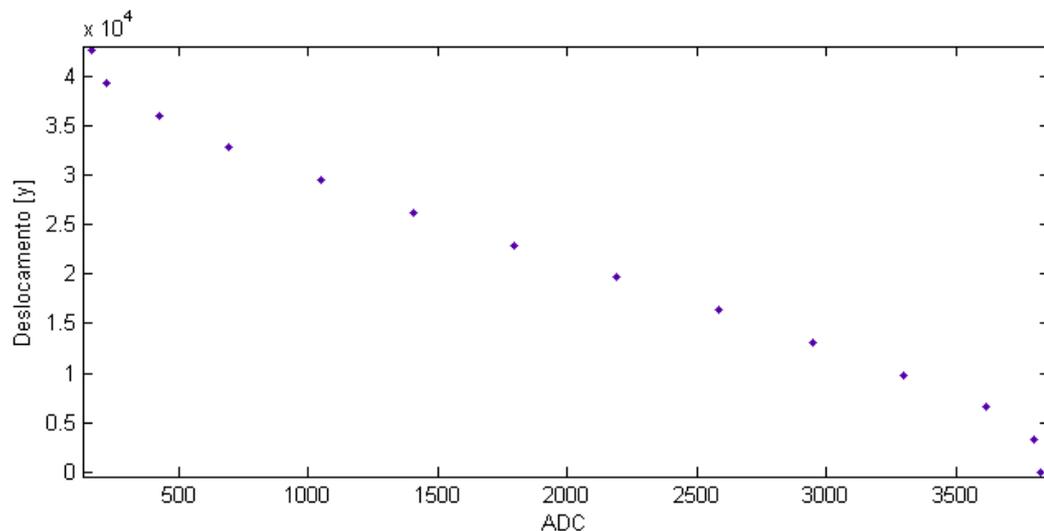


Figura 28 – Característica de um conjunto LVDT e placa de condicionamento.

Devido às não linearidades evidentes nos extremos da característica, de forma a reduzir o processamento necessário os LVDT's são normalmente utilizados apenas na zona linear, o que leva a que sejam ainda maiores. Como a restrição de dimensões é realmente prioritária, decidiu-se que seria obtida a característica real do LVDT e calculada uma aproximação. Para tal, desenvolveram-se dois métodos de cálculo que passam a ser descritos e comparados.

3.1.6.1. Aproximação polinomial de 3ª ordem da característica

De forma a obter o valor de deslocamento a partir do valor lido pelo Conversor Analógico-Digital (ADC), recorre-se à ferramenta *cftool* do *MATLAB* para obter uma aproximação polinomial cúbica, apresentada na Figura 29, da característica do sensor.

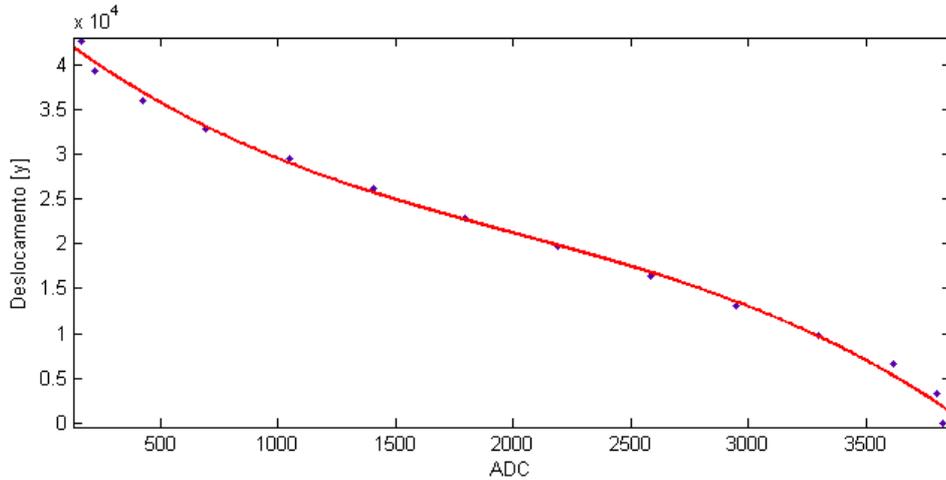


Figura 29 – Aproximação polinomial de 3ª ordem à característica real do LVDT.

Os valores dos parâmetros P , calculados pelo *cftool*, são então fornecidos ao sistema de envio de dados por *ZigBee* que efectua os cálculos da equação (30) à recepção dos dados, ficando portanto o sistema de aquisição livre de processamento.

$$y(x) = P_1x^3 + P_2x^2 + P_3x^1 + P_4 \quad (30)$$

Este método tem no entanto o inconveniente de não ser suficientemente preciso nas extremidades da característica, de ser necessário programar 4 parâmetros no sistema de envio de dados por cada sensor e de requerer uma quantidade de processamento considerável. Decidiu-se então implementar outro método para este efeito.

3.1.6.2. Aproximação por interpolação linear dos pontos reais

A partir dos pontos tirados das características, são calculados os declives m e intersecção das ordenadas b de cada segmento de recta que liga dois pontos adjacentes e guardados num vector, tendo já em conta que a saída deve ser um número de *16 bits*, sendo o máximo correspondente a *10 cm* e o mínimo *0 cm*.

A característica originalmente tirada consiste nas colunas 1 e 3. A partir do vector de deslocamento D calcula-se o vector y por meio de (31), sendo posteriormente a partir de y e x calculados m e b por (32) e (33).

$$y_i = D_i \frac{65536}{10} \quad (31)$$

$$m_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \quad (32)$$

$$b_i = y_i - m_i x_i \quad (33)$$

O algoritmo requer que sejam armazenados x_i , b_i e m_i na memória interna do microcontrolador, sendo que para calcular o valor de y de um determinado x o algoritmo começa por percorrer o vector x até encontrar $x_i < x$. Basta então aplicar (34) com o valor de i encontrado para se saber o valor de y .

$$y = m_{i-1}x + b_{i-1} \quad (34)$$

Tabela 8 – Característica e parâmetros para cálculo da interpolação linear.

D [cm]	y	x	m	b
0	0	4096	0,00	0,00
0	0	3823	-121,37	463998,93
0,5	3277	3796	-18,21	72385,29
1	6554	3616	-10,21	43457,48
1,5	9830	3295	-9,31	40505,33
2	13107	2943	-9,03	39675,07
2,5	16384	2580	-8,34	37897,13
3	19661	2187	-8,27	37753,45
3,5	22937	1791	-8,47	38102,65
4	26214	1404	-9,28	39247,73
4,5	29491	1051	-9,13	39084,67
5	32768	692	-12,27	41258,61
5,5	36044	425	-15,99	42837,78
6	39321	220	-59,58	52429,00
6,5	42598	165	0,00	42598,00
6,5	42598	0	-	-

Este algoritmo resulta numa característica aproximada semelhante à da Figura 30, que apesar das descontinuidades na derivada, apresenta um erro consideravelmente menor ao da aproximação polinomial de 3ª ordem.

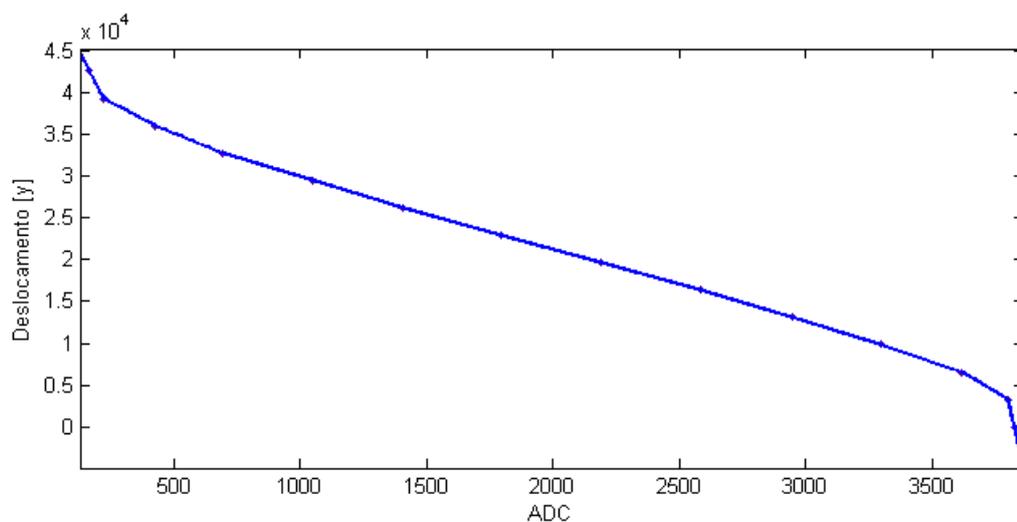


Figura 30 – Aproximação por Interpolação Linear à característica real do LVDT.

3.1.7. Implementação do algoritmo no módulo de aquisição

Uma vez que a implementação do algoritmo requer precisão de cálculo com números fracionários grandes, de maneira a não serem necessárias alterações ao código para diferentes valores dos parâmetros, decidiu-se utilizar variáveis do tipo vírgula flutuante até porque não existem problemas de capacidade de processamento.

Para se adquirir um valor de deslocamento de um LVDT, começa-se por ordenar uma conversão analógica-digital na porta correspondente, o valor da conversão é localizado no vector armazenado de x (unsigned int) e, com recurso as entradas correspondentes dos vectores m (float) e b (float), aplicado o algoritmo descrito na secção anterior.

3.2. Sensor de temperatura à superfície dos pneus

3.2.1. Introdução

Num veículo de competição, a temperatura da superfície de contacto do pneu com o chão é um factor de extrema importância para o desempenho deste. Enquanto o pneu não atinge a sua temperatura ideal de funcionamento a sua aderência ao chão é inferior à desejada, enquanto se essa temperatura for ultrapassada, a velocidade de degradação do pneu aumenta, degradando-se mais rapidamente do que é desejado e comprometendo a performance do veículo uma vez que à medida que o pneu se desgasta perde também aderência.

A temperatura da superfície de contacto do pneu com o chão varia, entre outras coisas, conforme a afinação da suspensão. Uma roda com *camber* negativo (ver Figura 31) apresentará uma temperatura mais elevada na parte mais interior do pneu do que na parte exterior.

Por tudo isto, os dados de temperatura da superfície de um pneu ao longo de um período de tempo são importantes para diagnóstico e afinação da suspensão, sendo também útil ter várias medições ao longo do eixo transversal do pneu.

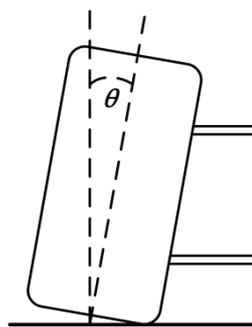


Figura 31 – Roda com um ângulo de *camber* negativo.

3.2.2. Escolha do sensor

Uma vez que se pretende medir a temperatura dos pneus com o carro em andamento, decidiu-se utilizar um sensor de temperatura sem contacto realizado por medição de radiação infra-vermelha.

Os termómetros de infra-vermelhos calculam a temperatura de um determinado material medindo a radiação do corpo negro emitida por esse material. A radiação emitida por um determinado

material depende linearmente da temperatura desse material e do coeficiente de emissão factor, do material. O factor de emissividade de um determinado material é uma relação da quantidade de energia radiada por esse material pela quantidade de energia radiada por um corpo negro à mesma temperatura. A densidade de energia (por unidade de área) radiada é proporcional à quarta potência de temperatura e depende do coeficiente de emissão da superfície relativamente ao corpo negro (radiador perfeito). O sensor é baseado nas leis de radiação de Stefan Boltzmann, Lei de Wien [21].

O sensor escolhido para esta tarefa foi o MLX90614 [22], da *Melexis*, por ser um sensor com processamento já incorporado e por permitir integração num barramento SMBus, o que facilita a integração de vários sensores na medição da temperatura ao longo do pneu. Dentro da família MLX90614 foi escolhido o MLX90614ESFAAA, que tem uma tensão de alimentação de 5 V, uma pilha termoelétrica e um campo de visão de 90°.

3.2.3. Configuração do dispositivo

Os dispositivos MLX90614 vêm calibrados de fábrica mas sem identificador de SMBus programado. Deve-se portanto programar um identificador diferente em cada dispositivo juntamente com o factor de emissividade correcto do pneu. Tanto o factor de emissividade como o identificador SMBus estão gravados na memória EEPROM interna do dispositivo e podem ser programados com recurso aos comandos SMBus da Tabela 9 reconhecidos pelo dispositivo.

Tabela 9 – Lista de comandos SMBus reconhecidos pelo MLX90614.

Comando	Descrição
000X XXXX	Acesso à posição de memória RAM 000XXXXX.
010X XXXX	Acesso à posição de memória EEPROM 000XXXXX.
1111 0000	Leitura de indicadores.
1111 1111	Ordenar modo de Stand-by.

De modo a programar o identificador SMBus e a emissividade devem ser enviados os comandos SMBus de escrita de palavra com verificação de erro (ver 2.4.1.2.1) da Tabela 10. De notar que todos os dispositivos respondem quando o destinatário é 0x00. Antes de se fazer uma escrita na *Electrically Erasable Programmable Read-Only Memory* (EEPROM) é necessário realizar uma limpeza da célula. O factor de emissividade $0 < \varepsilon < 1$ é armazenado em memória com uma resolução de 16 bits, sendo necessário fazer a conversão para Q_{15} sem sinal. De notar ainda que após feitas as alterações o dispositivo deve ser reinicializado para que tome efeito a alteração do identificador SMBus.

Tabela 10 – Sequência de comandos SMBus para programação de um dispositivo.

Ordem	Descrição	Destinatário	Comando	MSB	LSB
1	Limpeza da célula de memória	0x00	0x4E	0	
2	Programação do identificador SMBus	0x00	0x4E	Id. pretendido	
3	Limpeza da célula de memória	0x00	0x44	0	
4	Programação do factor de emissividade ε	0x00	0x44	65535	

3.2.4. Posicionamento do sensor

Visto se pretender ter a possibilidade de medir a temperatura do pneu em três posições diferentes, foi necessário calcular a distância máxima a que os sensores se podem encontrar do pneu para que o seu campo de visão abranja apenas superfície de pneu sem que haja sobreposição entre os campos de visão dos três sensores.

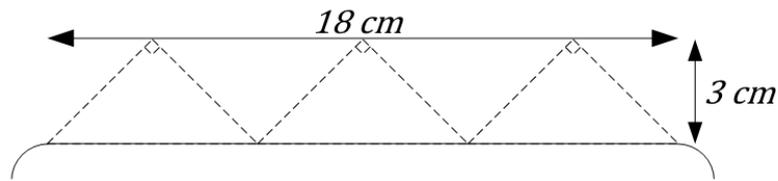


Figura 32 – Diagrama de posicionamento dos sensores de temperatura.

Dividindo a largura do pneu $L = 18 \text{ cm}$ em três secções iguais obtemos uma largura $l = 6 \text{ cm}$ para cada secção. Como os sensores escolhidos têm um campo de visão $FOV = 90^\circ$, significa que cada sensor deve estar a uma distância máxima $d_{max} = 3 \text{ cm}$ da superfície do pneu.

3.2.5. Leitura do valor de temperatura

O valor de temperatura actual do objecto da medição, bem como a temperatura ambiente, é armazenado na memória RAM do dispositivo e pode ser lido via SMBus através do comando de leitura de palavra com verificação de erro (ver 2.4.1.2.2) com o comando $0x06$.

Para se obter o valor de temperatura em graus centígrados é necessário recorrer a (35).

$$T [^\circ\text{C}] = 0,02 \times MSB \ll 8 + LSB + 273,15 \quad (35)$$

3.2.6. Barramento SMBus de um módulo de aquisição

Na prática, os módulo de interface com os sensores da frente e de trás são responsáveis pela leitura da temperatura dos dois pneus da frente e de trás, tendo cada pneu dois sensores de temperatura, um em cada extremo, como mostra a Figura 33.

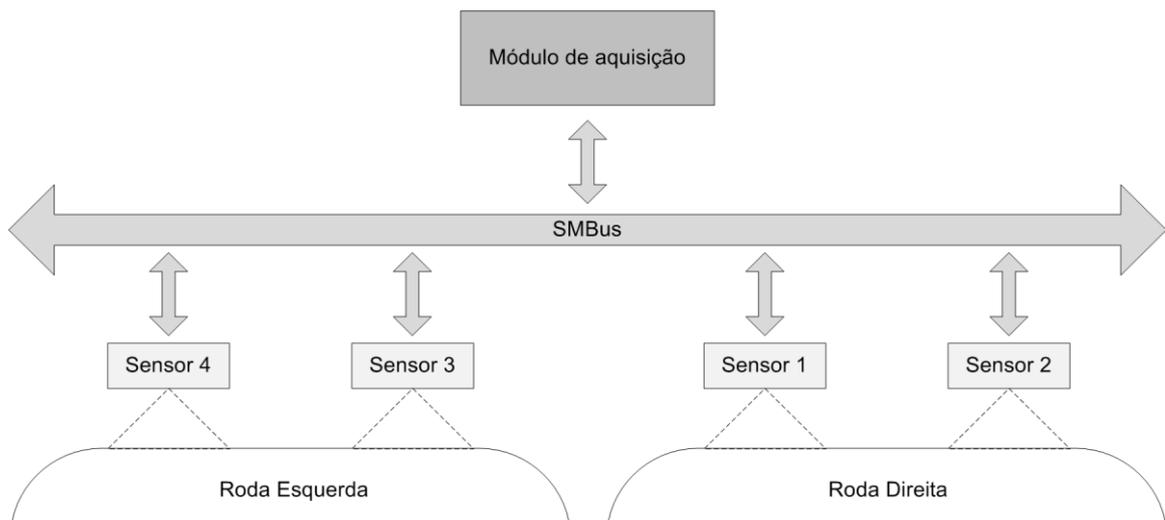


Figura 33 – Esquema do barramento SMBus nas rodas dianteiras e traseiras.

Para prevenir erros como a tentativa de leitura de sensores danificados ou desligados, foi implementado um algoritmo de procura que envia mensagens de leitura para todos os identificadores de sensor e verifica quais as que obtêm resposta, essa informação é depois armazenada num vector cuja primeira posição corresponde ao número de sensores detectados seguida dos identificadores de cada sensor disponível, o que no caso da figura de cima seria $SMB = \{4,1,2,3,4\}$.

3.2.7. Algoritmo de leitura

A leitura dos sensores de temperatura, como foi especificada em 2.3.3, deve ser feita a uma frequência de amostragem de 5 Hz. Para tal, utiliza-se o algoritmo da Figura 16, em que quatro dos sensores chamam a mesma função. Essa função é responsável por percorrer o vector SMB fazendo a leitura de um sensor de cada vez que é chamada. O algoritmo está representado na Figura 34.

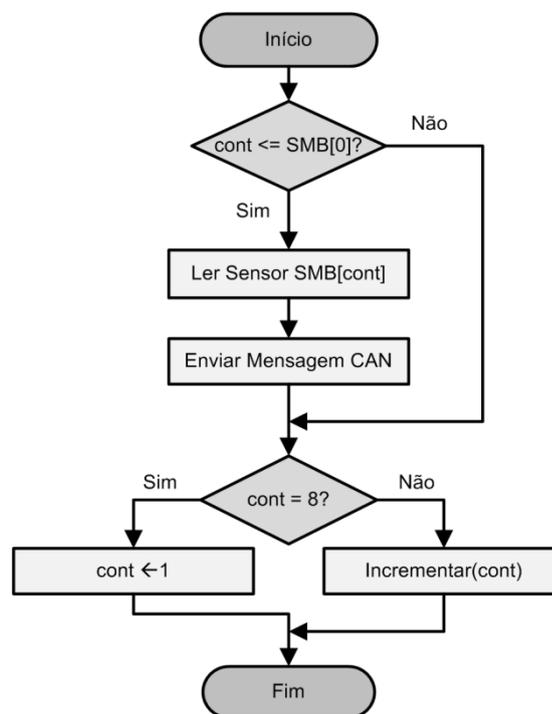


Figura 34 – Fluxograma da função responsável pela leitura dos sensores de temperatura.

3.2.8. Consumo de energia

Cada sensor MLX9016, com uma tensão de alimentação de $V_{DD} = 5V$ apresenta uma corrente média de $I_m = 1,22mA$, o que perfaz um consumo de energia de $P = I_m \times V_{DD} = 6,1mW$.

3.3. Sensor de velocidade

3.3.1. Introdução

Num veículo de competição é importante ter dados sobre a velocidade de rotação de todas as rodas e não apenas da velocidade linear do carro porque as rodas de um carro não giram sempre todas à mesma velocidade. No caso de um arranque rápido, por exemplo, as rodas motrizes tendem a patinar, girando mais rapidamente do que as rodas não motrizes. Noutra caso, quando um veículo efectua uma curva, a roda interior à curva percorre uma distância menor do que a roda exterior,

girando portanto a uma velocidade menor para percorrer menos distância no mesmo espaço de tempo. Quando uma travagem repentina faz o veículo derrapar, bloqueando ou não alguma das rodas também dificilmente a situação é exactamente igual em todas as rodas, e finalmente, quando um desequilíbrio no veículo provoca que alguma das rodas motorizadas fique suspensa o diferencial redirecciona a potência para a roda oposta, provocando novamente uma diferença de velocidade entre rodas.

Havendo portanto o interesse de estudar o comportamento do veículo em todas estas situações, há a necessidade de desenvolver um sistema que permita a obtenção da velocidade de todas as rodas.

3.3.2. Sensor de efeito de Hall

O dispositivo escolhido para esta tarefa foi o ATS675 [23] da *Allegro*. Trata-se de um sensor de efeito de Hall que pode ser alimentado a 5V e que na saída apresenta um valor lógico correspondente a ter, ou não, um material ferromagnético a uma distância inferior a 5 mm.

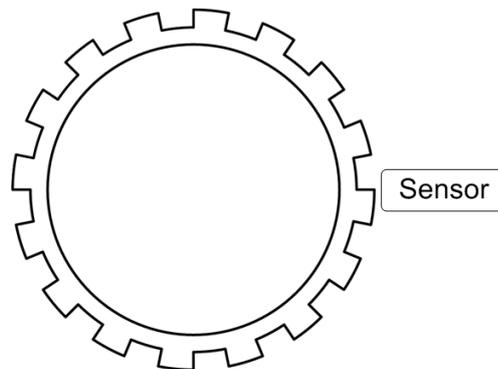


Figura 35 – Esquema para medição da velocidade com um sensor de efeito de Hall.

3.3.3. Cálculo da velocidade

O sinal de saída do dispositivo escolhido é uma onda quadrada, a *high* na presença de um dos dentes da roda e a *low* no vazio. Desta forma, sabendo o número N de dentes da roda e o raio r do pneu podemos calcular a velocidade v da roda a partir da frequência f do sinal, dada por (36).

$$v(f) = 2\pi r \frac{f}{N} \text{ m/s} \quad (36)$$

3.3.4. Implementação de um filtro de mediana

Na prática, o sensor magnético colocado nas rodas do veículo é accionado por uma roda semelhante à da Figura 35 com 17 dentes cuja função real no carro é servir de anilha em que um dos dentes é dobrado a 90° para trancar uma outra peça mecânica, isto na prática significa que do ponto de vista do sensor magnético trata-se de uma roda com de 17 dentes com um dente em falta. Significa então que o sinal gerado pelo sensor numa roda a girar a uma velocidade v tem um período T dado por (37) e $2T$ no dente em falta.

$$T(v) = \frac{2\pi r}{17v} [s] \quad (37)$$

Isto levanta o problema do valor de velocidade lido cair para metade de cada vez que o dente em falta passa pelo sensor. De maneira a contornar este problema, foi implementado um filtro de mediana. Este tipo de filtro retira erros instantâneos deste tipo sem comprometer a qualidade do sinal.

O filtro de mediana consiste em ordenar o *buffer* por ordem crescente e utilizar o valor da mediana.

3.3.5. Consumo de energia

O sensor escolhido, alimentado a $V_{DD} = 5V$ apresenta uma corrente de $I_m = 10mA$, o que perfaz um consumo de energia de $P = I_m \times V_{DD} = 50mW$

3.4. Sensor de temperatura do líquido de refrigeração

3.4.1. Introdução

Num veículo de competição, os dados de temperatura do líquido de refrigeração são necessários pois permitem avaliar o desempenho do sistema de refrigeração numa determinada pista. Uma vez que o motor de combustão interna tem uma gama de temperatura ideal de funcionamento restrita, os sistemas de refrigeração líquida dos veículos são projectados com um radiador à medida para que a temperatura estabilize nessa gama ideal de funcionamento. A monitorização da temperatura do líquido de refrigeração permite portanto a validação do projecto do sistema de refrigeração bem como o diagnóstico em tempo real de todo o sistema, permitindo que falhas potencialmente destrutivas para o motor sejam detectadas a tempo.

3.4.2. Sensor

O sensor utilizado para esta medida é uma termistência fornecida pelo fabricante juntamente com o motor. O sensor em questão é originalmente utilizado para fornecer uma medida da temperatura no mostrador de informações ao condutor da mota CBR600F4i. Uma vez que esse mostrador não é utilizado no carro, o sensor fica disponível.

Uma termistência cerâmica é um tipo de resistência cujo valor varia exponencialmente com a temperatura segundo (38).

$$R(T) = Ae^{BT} \quad (38)$$

3.4.3. Circuito de condicionamento

De forma a se obter uma tensão que possa ser computada para cálculo da temperatura a partir do termistor, é utilizado o circuito simples da Figura 36 baseado num divisor resistivo.

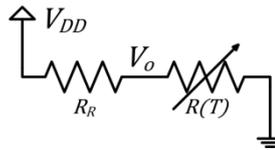


Figura 36 – Circuito de condicionamento do termistor.

A partir da fórmula da resistência do termistor (38) e da fórmula do divisor resistivo podemos concluir que a tensão de saída V_o varia com a temperatura pela equação (39), que depois de alguma manipulação algébrica permite obter a expressão da temperatura T em função da tensão de saída V_o por (40).

$$V_o(T) = V_{DD} \frac{Ae^{BT}}{Ae^{BT} + R_R} \quad (39)$$

$$T(V_o) = \frac{1}{B} \left[\ln\left(\frac{R_R}{A}\right) + \ln\left(\frac{V_o}{V_{DD} - V_o}\right) \right] \quad (40)$$

3.4.4. Característica da termistância cerâmica

Uma vez que a característica do sensor não é disponibilizada pelo fabricante, foi necessário emergê-lo em água aquecida e com um termómetro e um multímetro em modo de ohmímetro obter pontos da sua característica. Obtiveram-se então três valores de resistência a temperaturas diferentes e com o auxílio da ferramenta *cftool* do *MATLAB* foram obtidos os valores de $A = 36650$ e $B = -0,03438$.

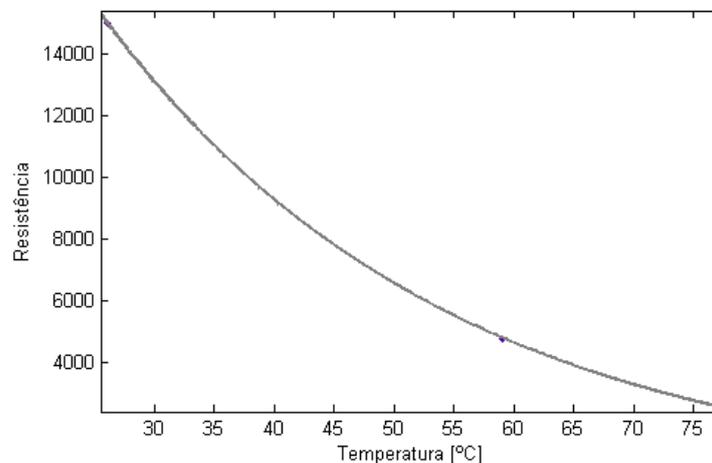


Figura 37 – Característica do termistor.

3.4.5. Implementação no módulo de aquisição

Para fazer a leitura do valor de temperatura o sinal V_o é lido directamente com recurso ao ADC do módulo de interface com sensores. A escolha da resistência de referência é importante pois, como se pode ver pela equação (40), esta provoca alterações na característica $V_o(T)$, alterando o regime de temperatura a que o sistema é mais sensível. Foi então escolhido $R_R = 3600 \Omega$ que com uma tensão $V_{DD} = 5 V$ produz a característica $V_o(T)$ da Figura 38.

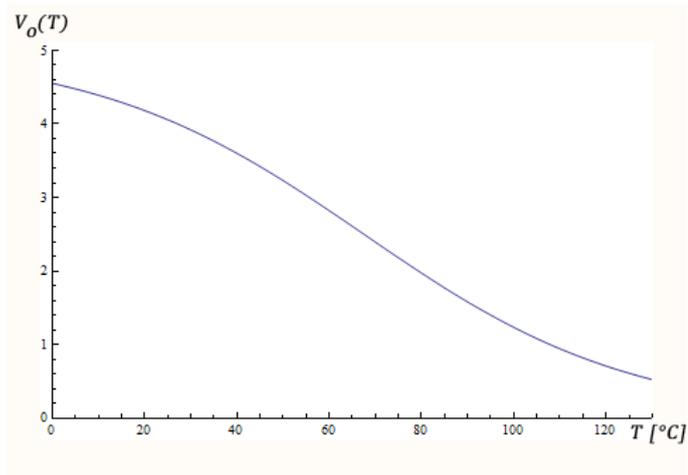


Figura 38 – Característica $V_o(T)$ do circuito da Figura 36.

3.4.6. Consumo de energia

Uma vez que o circuito apresenta uma tensão de alimentação constante e uma resistência que varia com a temperatura, podemos concluir que a potência será também variável com a temperatura.

Da expressão da potência $P = VI = \frac{V^2}{R} = \frac{V_{DD}^2}{R_R + Ae^{BT}}$ podemos obter o gráfico da Figura 39.

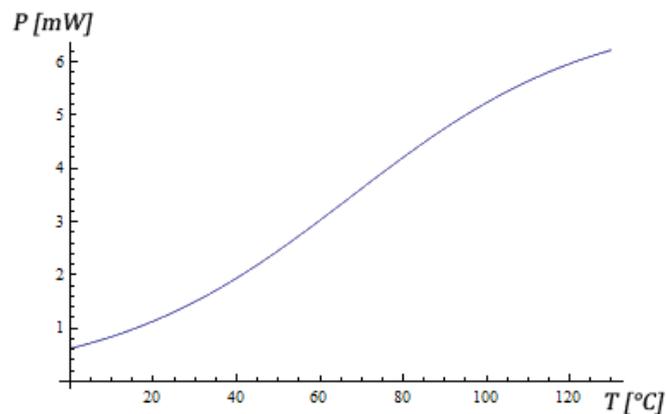


Figura 39 – Variação do consumo do sensor com a Temperatura.

3.5. Sensor de acelerador

3.5.1. Introdução

A obtenção de dados sobre a quantidade de acelerador é importante na avaliação do desempenho dos pilotos e na correção de técnicas de pilotagem, bem como na avaliação dos efeitos do acelerador no comportamento dinâmico do carro em pista.

3.5.2. O sensor

O sensor de acelerador utilizado é um potenciômetro fornecido pelo fabricante juntamente com o motor e a sua resistência varia entre 0 e R ohms. Este sensor é colocado de forma ao seu movimento ser solidário com o da válvula borboleta de admissão de ar, e como é alimentado directamente com 5 V, a tensão de saída V_o varia com o ângulo θ da válvula borboleta segundo (41).

O valor da tensão V_o é portanto obtido pelo conversor AD do micro-controlador da placa de interface com o sensor.

$$V_o = V_{DD} \frac{\theta}{\theta_{\max}} [V] \quad (41)$$

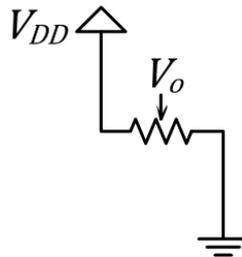


Figura 40 – Diagrama do sensor de acelerador.

3.6. Sensor sem fios de temperatura e pressão dentro dos pneus

3.6.1. Introdução

Quando a temperatura aumenta no interior de um pneu, pela lei dos gases perfeitos (42), uma vez que o volume é aproximadamente constante, a pressão aumenta também. Esse aumento de pressão altera a forma do pneu e conseqüentemente a superfície de contacto do pneu com o chão.

A informação sobre temperatura e pressão no interior dos pneus é portanto importante para perceber o seu funcionamento, bem como para avaliar o seu desempenho com uma determinada afinação da suspensão do veículo.

$$PV = nRT \quad (42)$$

3.6.2. Circuito emissor

O circuito emissor utilizado da Figura 41 foi desenvolvido por Pedro Lopes, colaborador do ProjectoFST, no âmbito da sua tese de mestrado [18] e foi escolhido por ser baseado num sensor de pressão e temperatura digital MPXY8020 da *Freescale* [24] com um micro-controlador rfPIC12F675F da *Microchip* [25].

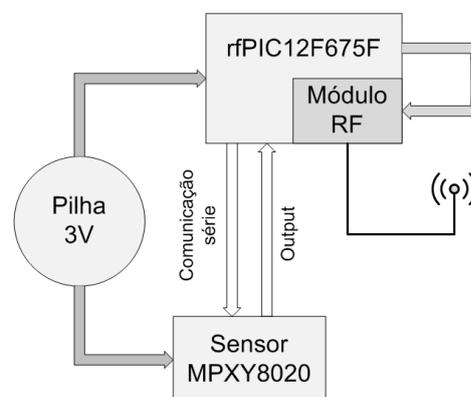


Figura 41 – Diagrama da PCI do sensor e emissor de RF.

A antena deste circuito, cuja PCI é apresentada na Figura 42, está já embebida na PCI que conta ainda com portos para depuração e programação e é alimentada por uma pilha de lítio CR2025 [26] de 3V.

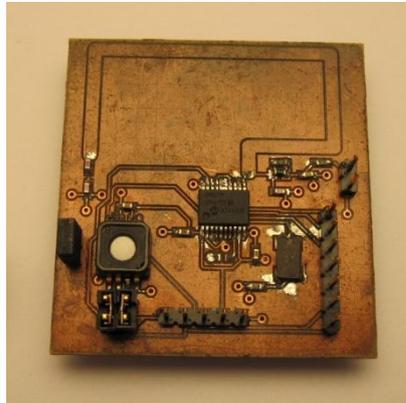


Figura 42 – PCI de suporte ao sensor e circuito emissor de RF.

3.6.2.1. Sensor de pressão e temperatura MLXY8020

O sensor MPXY8020 [24] é um circuito integrado com sensores de pressão e de temperatura embebidos e possui um porto de comunicação série que é utilizado receber instruções do micro-controlador. O sensor de pressão está calibrado para uma gama de funcionamento entre *2.5 bar* e *4.5 bar*.

Este sensor tem 4 estados de funcionamento controlados pelos pinos S0 e S1 descritos na Tabela 11. Quando o sensor se encontra num modo de medição, o valor de tensão do sensor a medir é comparado com um valor programado no sensor e o valor lógico resultante dessa comparação é posteriormente passado para o pino OUT quando o dispositivo passa para o modo de activação de saída. Isto significa que de modo a ter-se uma medida exacta, é necessário repetir o procedimento de medição várias vezes alterando o valor de referência do comparador.

Quanto à comunicação, trata-se de uma série convencional, ou seja, existe o sinal de dados DATA e o sinal de relógio CLK, ambos controlados pelo mestre que começa por colocado o *bit* mais significativo em DATA e posteriormente eleva CLK, repetindo este processo sequencialmente por ordem de significância dos *bits*.

Tabela 11 – Tabela dos estados do sensor de pressão e temperatura.

S1	S0	Estado
0	0	Em espera
0	1	Medição de pressão
1	0	Medição de temperatura
1	1	Activação de saída

A partir da gama de funcionamento do sensor e do valor resultante do algoritmo de medida, podemos concluir que o valor de pressão em *bar* pode ser calculado por (43).

$$P(x) = 2,5 + \frac{x}{128} \quad (43)$$

Quanto ao cálculo do valor de temperatura, a partir de vários pontos da característica do sensor providenciados pelo fabricante, pôde fazer-se uma aproximação polinomial cúbica com a ajuda da aplicação *cftool* do *MATLAB*, que resultou em (44) para o cálculo da temperatura em graus Célsius.

$$T(x) = 8,943 \times 10^{-6} x^3 - 0,005634 x^2 + 1,746 x - 103,6 \quad (44)$$

3.6.2.2. Micro-controlador rfPIC12F675F

O micro-controlador rfPIC12F675F [25] da *Microchip* [27], cujo diagrama se encontra na Figura 43, foi especialmente desenvolvido para aplicações de radiofrequência, possuindo já um módulo independente de transmissão de dados por *Ultra High Frequency* (UHF) com modulação *Amplitude-shift Keying* (ASK) e *Frequency-shift Keying* (FSK).

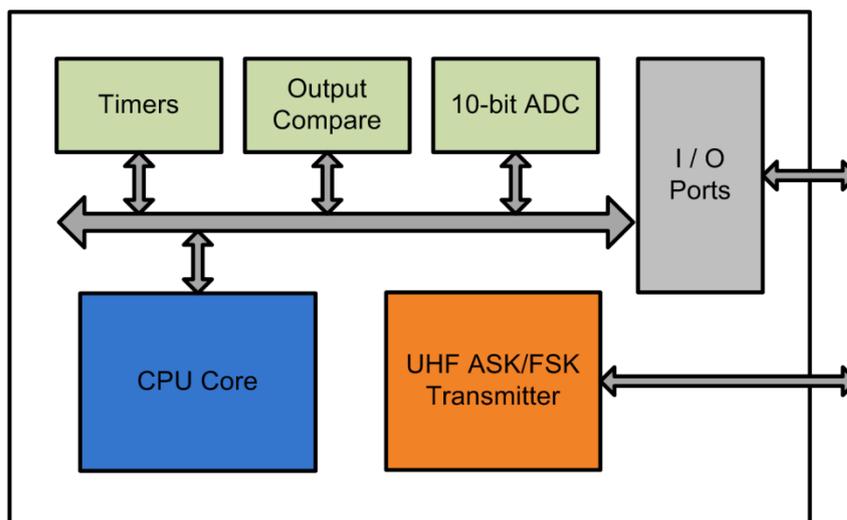


Figura 43 – Diagrama do micro-controlador rfPIC12F675F.

No âmbito deste projecto foi utilizada a modelação ASK a uma frequência de *433,92 MHz* pois a PCI disponível para desenvolvimento está projectada para este tipo de modelação.

3.6.3. Circuito receptor

O circuito receptor utilizado [18] é um circuito disponibilizado pelo fabricante *Microchip* próprio para o tipo de circuito emissor utilizado. É baseado no circuito integrado rfRXD0420 [28] receptor de UHF ASK / FSK / FM com o circuito externo aconselhado pelo fabricante em [29] para a frequência pretendida de *433,92 MHz*.

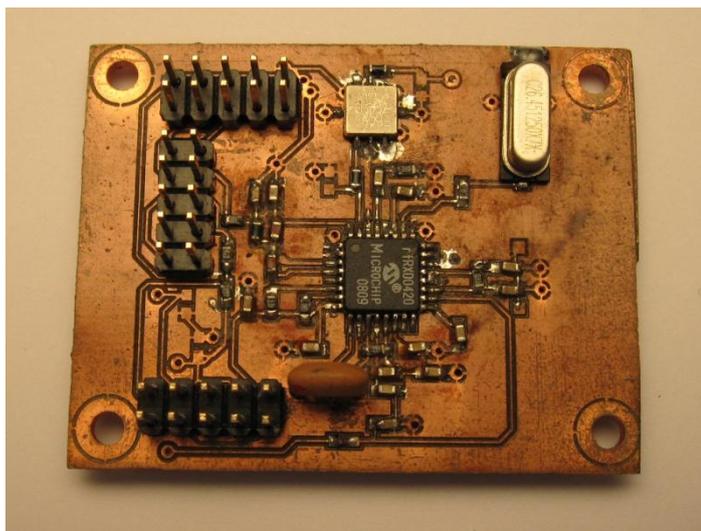


Figura 44 – PCI do circuito receptor.

3.6.4. Modelação ASK

A modulação digital de amplitude *Amplitude-shift Keying* de sinais de radiofrequência, utilizada neste projecto, consiste na modulação de um sinal de frequência constante pelo sinal digital que se pretenda transmitir, como se pode ver pelo exemplo da Figura 45.

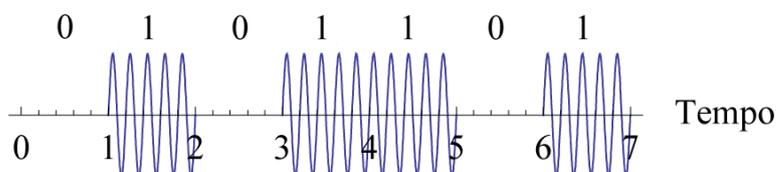


Figura 45 – Exemplo de modelação ASK.

3.6.5. Protocolo de comunicação

A comunicação é unidireccional e só existe um tipo de trama, a trama de dados. Esta consiste num preâmbulo de sincronização, um byte de identificação do emissor, um byte com o valor medido de pressão, um byte com o valor medido de temperatura e finalmente um byte de verificação de erro por CRC.



Figura 46 – Trama de dados da comunicação RF.

O preâmbulo é necessário pois o circuito receptor tem controlo automático de ganho. Quando não é detectado qualquer sinal na entrada, este ganho começa a crescer até que o ruído é amplificado ao ponto de ser detectado um 1 lógico inexistente. O ganho começa então a descer até que é novamente detectado um 0 lógico à entrada e todo o processo recomeça. O preâmbulo permite portanto que o ganho se auto-ajuste para o valor correcto, da mesma maneira que impede que o sinal

lógico de ruído aleatório à saída do circuito receptor não despoleta uma leitura por parte do micro-controlador que o monitoriza. Na improvável ocorrência do sinal de ruído apresentar uma combinação de bits semelhante à do preâmbulo e com os tempos correctos, o micro-controlador que faz a aquisição do sinal detectará erradamente uma trama de dados e adquirirá os bits que compõem a trama para somente no final a descartar por ter o byte de verificação de erro incoerente com o resto da trama. O preâmbulo utilizado é, em binário, 10101010 pois deve conter transições de estado lógico para ajuste do controlo automático de ganho.

3.6.6. Algoritmo de medição de pressão e temperatura no circuito emissor

Como o circuito emissor é alimentado por uma bateria de pequenas dimensões, com uma quantidade de energia limitada, é importante que este consuma o mínimo de energia possível. Dado que as variáveis medidas não variam com uma frequência elevada, decidiu-se que o micro-controlador deve fazer as medições e o envio dos dados à velocidade máxima permitida e posteriormente entrar em modo de baixo consumo até que se deva fazer nova medição.

O programa começa por levar a cabo uma pesquisa binária, alternando o modo de funcionamento do sensor MLXY8020 entre medição e activação de saída, enviando depois o novo valor de referências ao comparador do sensor. Depois de repetir este procedimento uma vez para cada bit do valor a medir, ou seja 8 vezes, para o sensor de pressão, o processo é repetido mas desta vez para medição da temperatura. Calcula-se então o valor de verificação de erro para a trama que se pretende enviar e procede-se ao envio da trama. Coloca-se então o sensor MLXY8020 em modo de espera, configura-se o temporizador para retirar o micro-controlador do modo de baixo consumo quando chegar o momento correcto e coloca-se o micro-controlador em modo de baixo consumo.

3.6.7. Algoritmo de aquisição do sinal gerado pelo circuito receptor

Uma vez que quando o emissor não está a transmitir o circuito receptor gera ruído (efeito descrito em 3.6.5) o receptor tem de ser capaz de distinguir entre ruído e sinal útil. Para tal é utilizada a funcionalidade de *Input Capture* do micro-controlador, que monitoriza continuamente o sinal e guarda num *buffer* os valores de tempo passados entre transições lógicas. A cada transição, o programa verifica se o tempo está dentro da gama aceitável, e se verificar 8 transições aceitáveis seguidas, começa a leitura de uma trama. O valor médio de tempo entre as 8 transições é calculado e utilizado para programar um temporizador que indica o instante de leitura de cada bit da trama para que eventuais desvios na frequência do emissor comprometam a recepção dos dados. No final é calculado o valor de verificação de erro e comparado com o adquirido, se estes coincidirem, a mensagem é dada como válida, senão é descartada.

3.6.8. Ritmo de transmissão de dados

O ritmo de transmissão de dados do circuito emissor é limitado pela capacidade de alternar de 1 para 0 do conjunto emissor-receptor como se verificou por um simples teste laboratorial onde se monitorizou num osciloscópio o sinal de saída do micro-controlador do emissor e o sinal à saída do receptor. Após a realização deste teste, foi possível maximizar o ritmo de transmissão de dados do sistema.

4. Implementação dos sistemas no FST-03

4.1. Introdução

Para validar o funcionamento do sistema e de modo a ter um sistema permanente no protótipo que permita a aquisição de dados, desenvolveu-se todo o material necessário para que o sistema desenvolvido fosse aplicado no protótipo de forma definitiva e fossem realizados testes.

4.2. Arquitectura do Sistema Electrónico

A arquitectura do sistema implementado no FST-03 é uma simplificação do sistema descrito no Capítulo 0 na medida em que foram apenas desenvolvidas algumas das suas partes constituintes. O sistema desenvolvido e implementado no FST-03, cujo funcionamento foi testado em pista, consiste em três módulos de IS e um módulo de armazenamento e transmissão de dados por ZigBee, estando este último à responsabilidade de David Copeto no âmbito da sua dissertação de mestrado.

Para estes testes estavam ainda apenas disponíveis os sensores de temperatura à superfície dos pneus, os sensores de velocidade de cada uma das rodas, os sensores de deslocamento do amortecedor, o sensor de ângulo de viragem, o sensor de acelerador, o sensor de rotações do motor e finalmente o sensor de temperatura do líquido de refrigeração do motor.

4.3. Instalação mecânica dos diversos sensores no FST-03

4.3.1. Sensores de velocidade das rodas

Os sensores de velocidade das rodas, cujo funcionamento foi descrito em 3.3, foram colados aos porta-cubo do veículo, a uma distância de 1 mm da anilha dentada com a ajuda de uma cunha e depois de terem sido introduzidos dentro de uma caixa que contém também uma placa perfurada com o condensador e resistência necessários para o funcionamento do sensor. O resultado final é apresentado na fotografia da Figura 47.



Figura 47 – Montagem do sensor de efeito de Hall no FST-03.

4.3.2. Sensores de deslocamento da suspensão

Para os sensores de deslocamento da suspensão foram utilizados os sensores de deslocamento linear descritos em 3.1. Estes sensores, que foram construídos tendo como objectivo minimizar o peso, têm como estrutura principal um tubo de fibra de carbono de 6 mm de diâmetro interior e 8 mm de diâmetro exterior, por fora do qual é feito um enrolamento de 120 mm que constitui o primário e sobre o qual são posteriormente enrolados os dois secundários de 60 mm . Numa das extremidades é colado um suporte e na outra entra o núcleo. O núcleo é um cilindro de ferro macio com 6 mm de diâmetro e 60 mm de comprimento colado a um varão de carbono com 60 mm , na extremidade do qual é colado outro suporte. O sistema completo é montado em dois suportes em forma de U, fixos nos parafusos dos amortecedores e assim solidários com o seu movimento e está apresentado na fotografia da Figura 48.



Figura 48 - LVDT montado no amortecedor do FST-03.

4.3.3. Sensor de ângulo de viragem do volante.

O sistema de direcção do veículo é composto por um volante, que está ligado a uma caixa de direcção por meio de um cardan. O movimento angular no volante é transformado em movimento linear na caixa de direcção e levado até às rodas por tirantes. É portanto possível determinar o ângulo do volante sabendo o deslocamento linear da caixa de direcção. Para tal foi construindo um LVDT semelhante ao de 4.3.2 mas com um comprimento total de 160 mm e capaz de medir deslocamentos

até 8 mm. Este LVDT foi montado de forma ao seu movimento ser solidário com o da caixa de direcção do veículo e a montagem final é apresentada na Figura 49.



Figura 49 – LVDT montado na caixa de direcção do FST-03.

4.3.4. Sensores de temperatura da superfície dos pneus

Uma vez que, como foi visto em 3.2, estes sensores devem estar sempre a uma distância máxima de 3 cm do pneu, foi necessário desenvolver um suporte dedicado para eles. Este apoio, apresentado na Figura 50 e na Figura 51, deve seguir o movimento de direcção da roda e manter uma distância constante ao pneu independentemente do estado da suspensão. Foi então feito um apoio que se suporta nos parafusos do porta-cubo e contorna a roda até a parte superior onde um tubo quadrado serve de apoio e protecção aos sensores.



Figura 50 – Apoio dianteiro dos sensores de temperatura da superfície dos pneus.



Figura 51 – Apoio traseiro e PCI dos sensores de temperatura da superfície dos pneus.

4.4. Placas de circuito impresso

4.4.1. Circuito de excitação e condicionamento de sinal dos LVDT

De maneira a implementar os circuitos descritos em 3.1 foi desenvolvida uma PCI apresentada na Figura 52. O objectivo desta placa é estar associada ao LVDT de maneira que, para o sistema de aquisição de dados, este possa ser tratado como um bloco com apenas três ligações, duas de alimentação e a de sinal. O esquemático desta PCI é apresentado no Anexo A e os ficheiros de CAD estão no CD que acompanha esta dissertação.

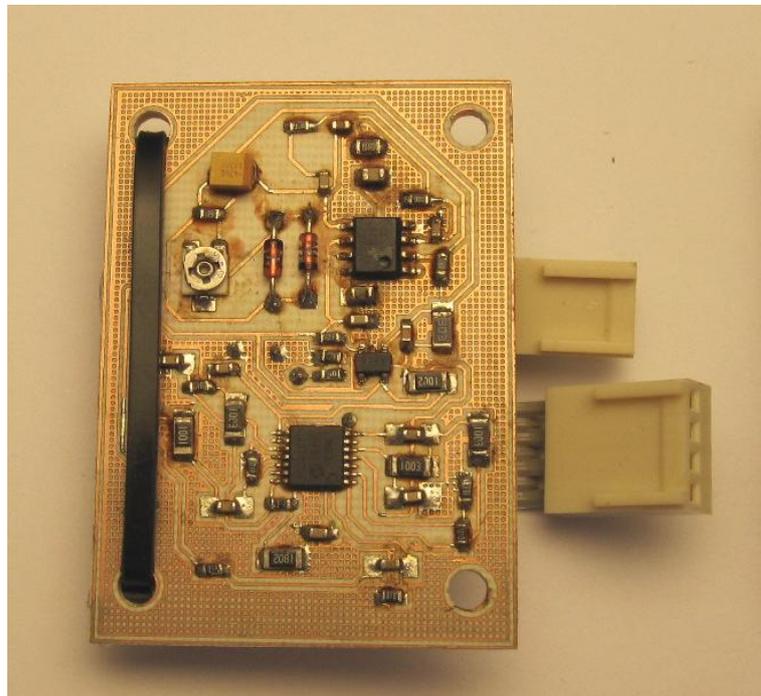


Figura 52 – PCI de excitação e condicionamento de sinal de um LVDT.

4.4.2. Suporte ao barramento SMBus e sensores *MLX9016*

De modo a suportar os sensores de temperatura da superfície do pneu foi desenvolvida uma PCI que além de um conector, tem lugar para 3 sensores *MLX9016* igualmente espaçados de *6 cm*, as resistências de *pull-up* do barramento SMBus e um condensador de desacoplamento para cada sensor. O esquemático desta PCI é apresentado no Anexo A, os ficheiros de CAD estão no CD que acompanha esta dissertação e o resultado final na Figura 53.

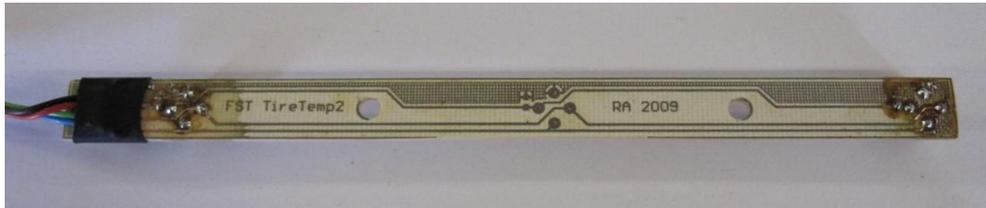


Figura 53 – PCI de suporte aos sensores de temperatura da superfície dos pneus.

4.4.3. Condicionamento de suporte aos sensores de origem

Com o objectivo de baixar a tensão dos sensores originais de *12 V* para *5 V* e para fazer o circuito da Figura 36 e do sensor de ponto-morto foi desenhada a placa de circuito impresso cujo esquemático é apresentado no Anexo A e os ficheiros de CAD estão no CD que acompanha esta dissertação.

4.5. Interconexão de módulos

A ligação entre os módulos de interface com sensores é feita através do barramento CAN cujo suporte físico são 2 condutores, CANH e CANL, aos quais se juntaram outros dois de alimentação. A alimentação do barramento é feita pela bateria através do relé de ignição, o que significa que o sistema só se liga quando o carro está alimentado.

Uma vez que o regulador de tensão das placas de interface é capaz de fornecer uma corrente de *3 A* e o sistema tem 3 placas, tanto os cabos como todos os conectores devem tolerar correntes máximas de *9 A*. Os conectores utilizados no suporte físico do barramento CAN são os SureSeal 120-1806-000 da Figura 58 pois são impermeáveis e suportam correntes de *15 A* nos seus pinos.

4.6. Caixas dos módulos de aquisição de sinais

As caixas utilizadas são fabricadas a partir de um perfil quadrado de alumínio cortado à medida pretendida no qual são colocadas duas tampas em que se montam conectores com flange. O projecto das caixas foi feito pelo Prof. Moisés Piedade e pelo aluno Miguel Guedes, colaborador do ProjectoFST. Foi desenvolvido um conjunto de ferramentas que permitem a construção rápida, precisa e económica de caixas de diferente comprimento a partir de um perfil de alumínio bruto. O projecto dos componentes da caixa é apresentado na Figura 54. Os conectores utilizados nestas caixas são os SureSeal 120-8552-202 [30] da Figura 55 porque permitem uma boa fixação, impermeabilização e apresentam grande robustez. Na Figura 56 apresenta-se o aspecto de uma destas caixas sem conectores montados.

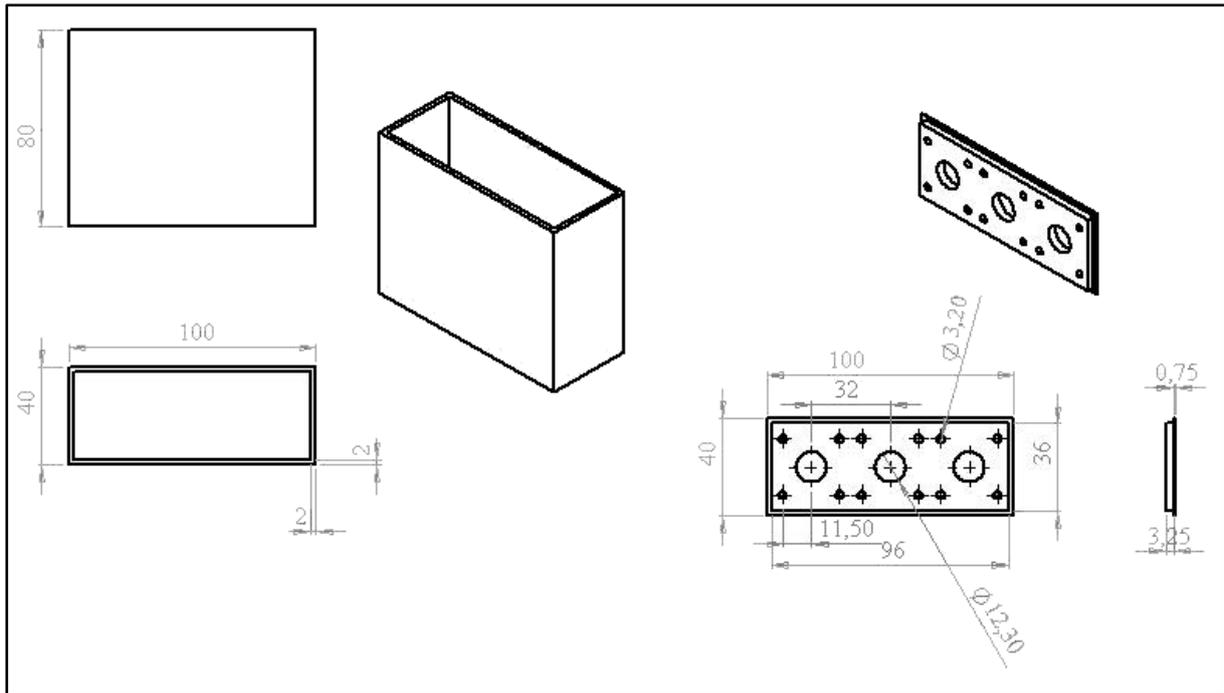


Figura 54 – Dimensões da caixa dos módulos de interface com sensores.

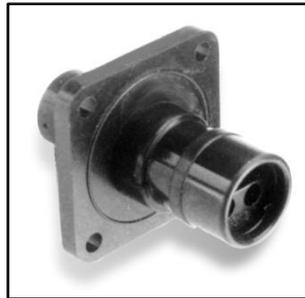


Figura 55 – Conector SureSeal 120-8552-202.

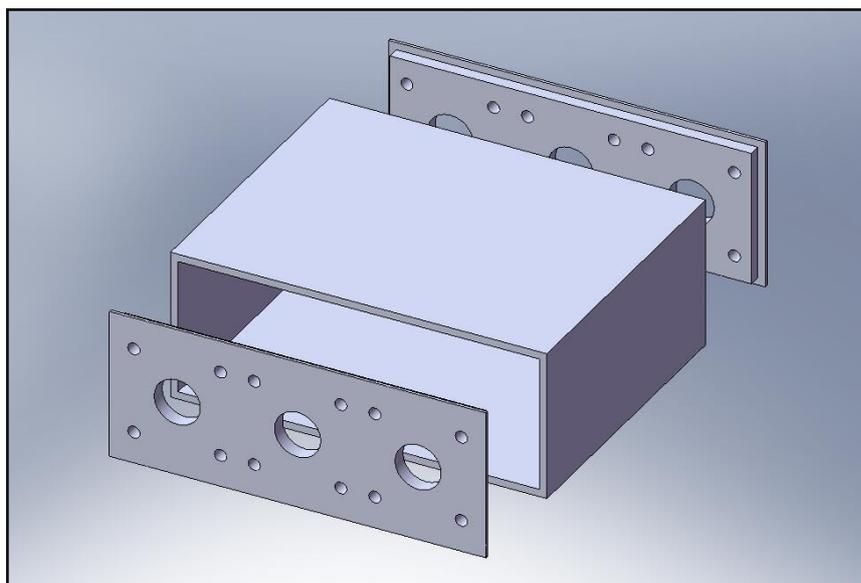


Figura 56 – Projecto da caixa dos módulos de interface com sensores.

4.7. Ligação dos módulos de aquisição de sinais aos sensores

A ligação dos módulos aos sensores é feita por cabos feitos à medida com as fichas Mini-SureSeal [30] 120-8552-102 da Figura 57 na extremidade do sensor e SureSeal 120-1806-000 da Figura 58 na extremidade da caixa.



Figura 57 – Conector Mini-SureSeal 120-8552-102.



Figura 58 – Conector SureSeal 120-1806-000.

4.8. Conclusões

A colocação das placas de excitação e condicionamento de sinal dentro das caixas construídas demonstrou-se uma tarefa complicada pois as caixas apresentam muito pouco espaço e as PCI não foram desenhadas tendo em conta a maneira como seriam posteriormente fixadas. Desta experiência concluiu-se que cada módulo de interface com sensores deveria ter uma PCI de suporte que se acoplasse directamente ao módulo de processamento e que continha todos os circuitos e ligações necessários a esse módulo. Notou-se também pela primeira vez que cada caixa está limitada a 5 sensores se for pretendido que cada sensor tenha a sua ligação por um conector independente à caixa, o que está longe de ser ideal pois com tão poucos sensores fica muita capacidade de processamento por utilizar e a introdução de mais módulos torna-se problemática devido ao espaço disponível no veículo.

5. Resultados Experimentais

De maneira a validar o sistema e também para se obterem os primeiros dados sobre o veículo, este foi levado ao Kartódromo Internacional de Palmela onde foram efectuados testes nos sistemas electrónicos que foram desenvolvidos no âmbito deste projecto. O sistema testado é composto por 3 módulos de aquisição de dados, um na dianteira, um ao centro e outro na traseira do veículo. Para estes testes estavam ainda apenas disponíveis os sensores de temperatura à superfície dos pneus, os sensores de velocidade de cada uma das rodas, os sensores de deslocamento do amortecedor, o sensor de ângulo de viragem, o sensor de acelerador, o sensor de rotações do motor e finalmente o sensor de temperatura do líquido de refrigeração do motor.

5.1. Medição da temperatura à superfície dos pneus

No que diz respeito à medição da temperatura à superfície dos pneus, os resultados foram os esperados no que diz respeito ao funcionamento dos sensores mas surpreendentes no que diz respeito à informação adquirida.

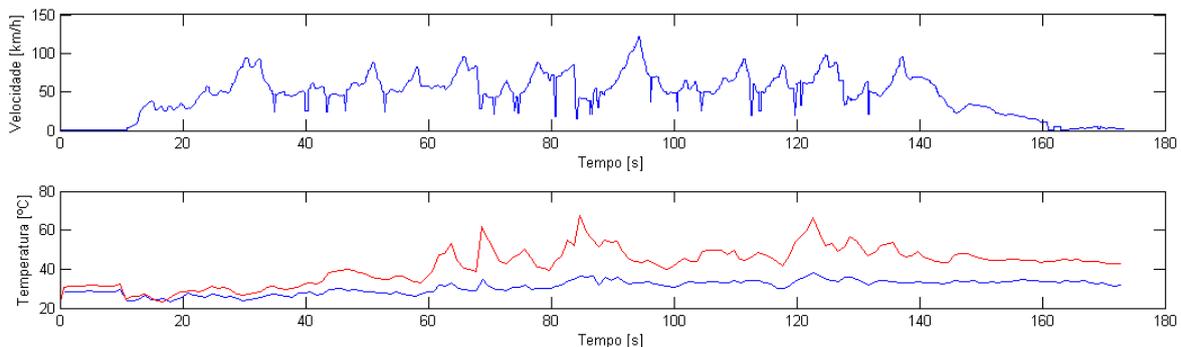


Figura 59 – Comparação entre velocidade e temperatura na parte exterior (azul) e interior (vermelho) da superfície do pneu.

Como se pode ver pela Figura 59, a diferença de temperaturas entre a parte interior e exterior do pneu, prevista em 3.2.1, existe e é de facto em algumas situações superior ao que se esperava, chegando o mesmo pneu a apresentar diferenças superiores a 30 °C. De notar ainda que na mesma parte do pneu foram detectadas variações de temperatura de 30 °C num intervalo de apenas 5 segundos.

5.2. Medição da velocidade

A medição de velocidade de todas as rodas permite que no futuro sejam implementados sistemas de controlo de arranque e mudanças semi-automáticas. Neste momento, permite avaliar o desempenho dos pilotos na prova de arranque e estimar a posição do carro na pista através de pontos de referência, como a recta da meta onde a velocidade é máxima.

A Figura 60 permite observar a diferença de velocidade entre as duas rodas dianteiras, quase nula nas rectas e com diferenças significativas nas curvas, especialmente as mais apertadas.

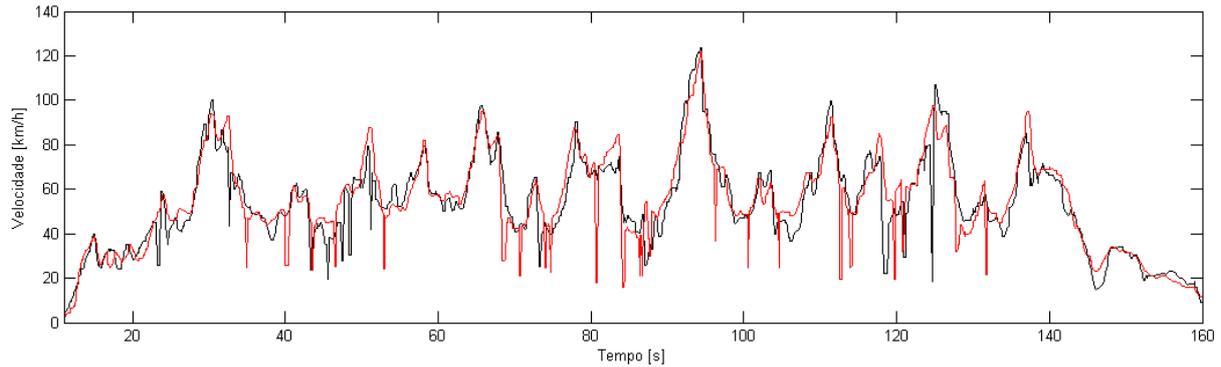


Figura 60 – Velocidade das rodas dianteiras direita (preto) e esquerda (vermelho).

5.3. Medição da rotação do motor e do sensor de acelerador

Apesar de se tratarem ambos de sensores simples, os dados adquiridos destes são essenciais para contextualizar os dados adquiridos dos restantes sensores. A Figura 61 compara os dados destes dois sensores e permite confirmar o bom funcionamento de ambos pois é notória a subida da rotação do motor quando o piloto pisa o acelerador.

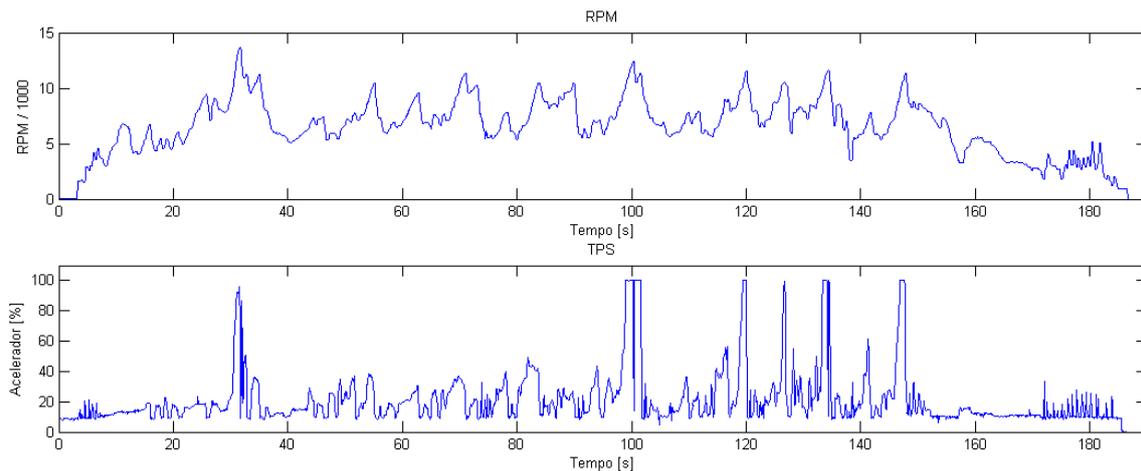


Figura 61 – Comparação de gráficos da rotação e do acelerador.

5.4. Conclusões

Com estes testes foi possível concluir o bom funcionamento do sistema e de todos os sensores com excepção dos LVDTs, cujos dados adquiridos não são aqui revelados pois um erro de cálculo no sistema de armazenamento e envio por *ZigBee* invalidou essa hipótese. No entanto os LVDTs foram extensivamente testados em laboratório.

É possível concluir que os sensores de temperatura da superfície dos pneus devem ser amostrados a uma frequência superior à utilizada, pois os pneus sofrem variações de temperatura superiores às esperadas em intervalos de tempo muito curtos.

No que diz respeito aos sensores de velocidade das rodas, os sensores funcionam correctamente mas o algoritmo deve ser melhorado para lidar com o bloqueio das rodas nas travagens pois é importante obter dados correctos sobre o funcionamento do veículo nessa situação.

6. Conclusões e perspectivas de trabalho futuro

6.1. Arquitectura do sistema

6.1.1. Conclusões

No que diz respeito à arquitectura do sistema, concluiu-se que se tomou a abordagem correcta para uma primeira implementação. No projecto desta arquitectura houve necessidade de obter o compromisso certo entre centralizar todo o sistema e atomizar o sistema ao máximo. No que diz respeito a centralizar, o problema está nas cablagens. Uma vez que para se ligar todos os sensores ao mesmo módulo é necessário que cada sensor esteja ligado por cabos a esse módulo, o comprimento de cabo necessário para implementação do sistema aumenta e aumenta também a complexidade pois esses cabos necessitam de uma ligação física ao módulo por meio de um conector, o que dado o número de sensores implica que a caixa teria de ser maior para albergar todos os conectores necessários ou que todos os cabos deveriam estar ligados a um único conector, o que não é prático.

A divisão em três módulos, um por cada zona do veículo, foi então a solução encontrada, pois continua a permitir a centralização local dos sensores, o que facilita bastante o desenvolvimento do sistema, mas já elimina o problema do número de conectores e comprimento dos cabos.

6.1.2. Trabalho futuro

No entanto, o futuro parece estar na semi-atomização do sistema. Apesar da centralização efectuada ser útil para o desenvolvimento do sistema, com um baixo número de PCI a terem de ser desenhadas e uma solução estandardizada de processamento para todos os sensores, acredita-se que uma vez que um determinado tipo de sensor já seja bem conhecido e esteja a funcionar correctamente num sistema centralizado como o efectuado, deve passar-se ao passo seguinte e torná-lo independente dos módulos centrais de processamento. Isto permitira reduzir o comprimento de cabo e o número de conectores necessários, mas mais importante ainda, libertava os módulos de aquisição de dados centrais para desenvolvimento de novos sensores. Se isto não for feito, a adição de novos sensores implicará a adição de novos módulos de interface, pois os actuais já estão esgotados em termos de espaço dentro das caixas e de conectores, e mesmo que não estivessem, efectuar alterações nos módulos já existentes implicaria que futuros desenvolvedores do projecto percebessem tudo o que foi feito até ao momento, o que não é de todo a solução ideal.

No que diz respeito aos sensores que foram desenvolvidos, pelo menos os sensores de velocidade e os sensores de temperatura da superfície dos pneus podem ser tornados independentes do restante sistema com pouco esforço e ganhos consideráveis.

6.2. Sensores de velocidade das rodas

6.2.1. Conclusões

Os sensores de velocidade das rodas funcionam como previsto e os seus dados foram essenciais para a contextualização dos dados dos restantes sensores, sendo possível a partir da velocidade do veículo e conhecendo bem a pista saber a posição do veículo. Apesar do bom funcionamento, o algoritmo deve ser melhorado para lidar o bloqueio das rodas que ocorre em travagens fortes. No instante em que as rodas bloqueiam a roda dentada pára de rodar e, há falta de novos dados, se ocorrer uma leitura o sistema envia o valor antigo. O valor de velocidade só é então actualizado quando a roda volta a rodar e provoca um novo flanco no sensor. O valor de velocidade lido, dado por $\frac{dx}{dt}$, está portanto errado pois dx corresponde apenas à distância percorrida pela roda a rodar, e dt corresponde ao total de tempo passado desde que a roda bloqueou. Desta forma não se obtém nem a velocidade linear real da roda, nem a velocidade de rotação real da roda, estando os dados adquiridos nestes instantes simplesmente incorrectos. Mesmo assim, o fenómeno provoca uma baixa repentina no valor de velocidade adquirido pelos sensores, o que apesar dos dados estarem incorrectos, permite saber que ocorreu um bloqueio de rodas nos instantes vizinhos.

6.2.2. Melhorias do sistema actual

No que diz respeito ao trabalho futuro a realizar neste sensor, para além das melhorias no algoritmo, é importante implementar o sistema introduzido em 6.1.2. No caso deste sensor, o sistema final poderia ser baseado num micro-controlador PIC18F2480 e um regulador de tensão de 5 V e um transceptor CAN MCP2551, o que resultaria numa PCI pequena e simples que poderia ser colocada directamente no local em que é colado o sensor de efeito de Hall, sendo o único cabo deste sistema a ligação ao barramento CAN.

6.3. Sensores de temperatura da superfície dos pneus

Dos testes feitos com estes sensores foi possível concluir que uma frequência de amostragem a 1 Hz, como foi utilizada, é insuficiente. Estes sensores devem trabalhar a uma frequência de amostragem de 10 Hz e é importante que seja feita a calibração do factor de emissividade, sendo para tal necessário efectuar uma experiência em que se aquece o pneu e se mede a sua temperatura com um termómetro de contacto, ajustando-se depois o valor do factor de emissividade do pneu para que o valor lido por este sensor corresponda ao medido com o termómetro de contacto.

Com já foi referido em 6.1.2, também este sensor pode se tornar independente dos módulos de aquisição de dados de forma simples e com pouco desenvolvimento adicional. Tal como em 6.2, também para este sensor um PIC18F2480 seria suficiente com um regulador de tensão de 5 V e um transceptor CAN, sendo neste caso possível acrescentar estes componentes à PCI já existente que serve de suporte ao barramento SMBus e aos dispositivos sensores, e sendo a única ligação a do barramento CAN.

6.4. Módulo de Interface USB-CAN

O módulo de Interface USB-CAN foi utilizado durante o desenvolvimento de todo este projecto como analisador de todas as mensagens transmitidas no barramento CAN e funcionou correctamente. Acontece no entanto que se forem enviadas várias mensagens CAN num curto espaço de tempo este módulo não as detecta todas. Suspeita-se que isto se deva ao facto de não estar a ser utilizada capacidade do controlador de CAN MCP2515 de gerar interrupções. Em vez disso, o programa do micro-controlador PIC18F4550 faz verificação periódica, o que é claramente mais lento e pode levar a que os *buffers* de recepção encham sem que o micro-controlador tenha tempo de os esvaziar, havendo portanto perda de mensagens.

Alterando a PCI do módulo de Interface USB-CAN para ligar a porta de interrupção do controlador de CAN MCP2515 a uma das portas do micro-controlador PIC18F4550 que têm capacidade de gerar interrupções, e criando um *buffer* interno no micro-controlador que armazene imediatamente as mensagens recebidas do controlador de CAN e liberte imediatamente o *buffer* de recepção, é possível superar este problema, sendo as mensagens recebidas enviadas por USB para o PC *a posteriori*, quando o programa do micro-controlador atingir a parte de lidar com a USB. Este trabalho não foi no entanto desenvolvido pois o sistema actual é suficiente para o trabalho desenvolvido e o aperfeiçoamento deste sistema não ser a prioridade deste projecto.

6.5. Sensores de deslocamento linear

6.5.1. Conclusões

No que diz respeito aos sensores de deslocamento linear LVDT, utilizados na medição do curso da suspensão e do ângulo de viragem, as alterações para encurtar as suas dimensões sem prejudicar o funcionamento foram um sucesso e até melhoraram a ligação magnética entre primário e secundários, aumentando a força electromotriz e reduzindo assim a influência de ruídos no resultado final. No que diz respeito às PCI de excitação e condicionamento de sinal, as alterações feitas às placas antigas reduziram consideravelmente os custos pois deixou de ser necessário o regulador de tensão de 5 V e a placa passou a ser de uma camada apenas, ficando simplificado o processo de fabrico e baixando consideravelmente o tempo de fabrico.

6.5.2. Melhorias do sistema actual

Para futuras melhorias deste sistema, também estes sensores podem tornar-se independentes do sistema de aquisição de dados. Tal como para os restantes sensores em que já discuti esta questão, também aqui é possível acrescentar à placa de excitação e condicionamento de sinal um micro-controlador simples como o PIC18F2480, um transceptor CAN e um regulador de tensão, tornando assim esta placa independente, com ligações apenas ao barramento CAN e ao LVDT.

6.5.3. Nova abordagem ao sensor de deslocamento

É ainda possível alterar completamente o sistema de excitação e condicionamento de sinal para que contenha menos componentes, passando o processamento a estar todo no micro-controlador.

No sistema proposto da Figura 62 o micro-controlador é, através de um temporizador, responsável pela criação de uma onda quadrada que passa por um filtro passa-baixo que a transforma em sinusoidal e um amplificador de potência para ser aplicada ao primário do LVDT.

As diferenças de tensão dos secundários e do primário são amostradas directamente e todo o restante processamento é feito de modo digital por um *Digital Signal Processor* (DSP).

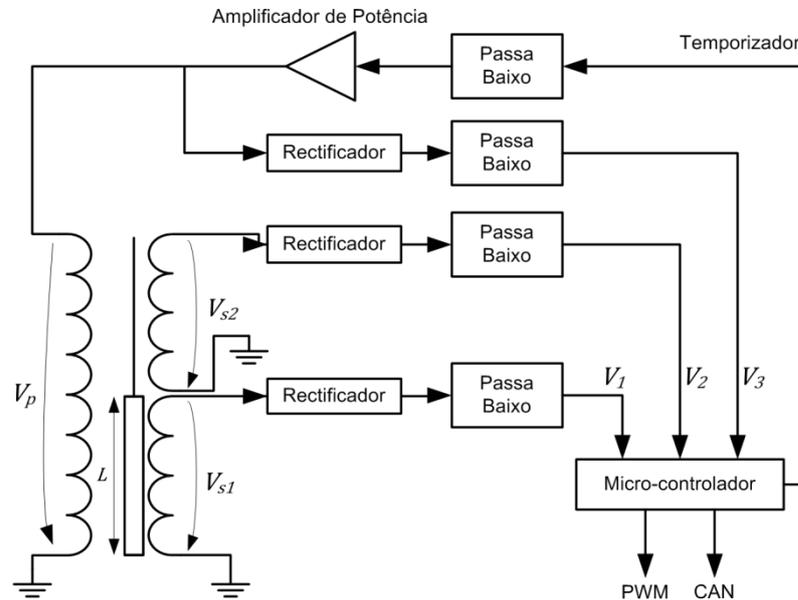


Figura 62 – Arquitectura proposta para o sistema de excitação e condicionamento de sinal de um LVDT.

6.6. Sensor de pressão e temperatura no interior do pneu

O sensor de pressão e temperatura no interior do pneu é um sistema útil mas com o qual não foi possível efectuar todos os testes que pretendiam. Seria interessante testar o funcionamento do sistema com vários emissores e um só receptor, não foi no entanto possível realizar esse teste pois o sensor foi descontinuado e não foi possível comprar mais sensores. Mesmo assim aumentou-se o ritmo de transmissão e o tempo em que o sistema está em modo de baixo consumo ao máximo para reduzir o consumo e assim estender o tempo de vida da bateria.

Uma vez que o sensor utilizado actualmente foi descontinuado, recomenda-se que no futuro se utilize o circuito integrado MPXY8300 [31] da Freescale. Este sistema integra num único circuito integrado um micro-controlador, um sensor de pressão, um sensor de temperatura, um acelerómetro e um módulo de rádio frequência para transmissão de dados. Utilizando este circuito o tamanho da PCI do sensor podia ser reduzida significativamente e simplificada.

Referências

- [1] <http://www.megasquirt.org>, Outubro 2009.
- [2] <http://www.motec.com/>, Outubro 2009.
- [3] <http://www.magnetimarelli.com/>, Outubro 2009.
- [4] <http://www.mclarenelectronics.com/>, Outubro 2009.
- [5] <http://www.bosch-motorsport.de/>, Outubro 2009.
- [6] <http://www.zigbee.org/>, Outubro 2009.
- [7] “System Management Bus (SMBus) Specification Version 2.0”, SMB Implementers Forum, Agosto 2000.
- [8] “CAN Specification Version 2.0”, Robert Bosch GmbH, 1991.
- [9] “Overview and Use of the PICmicro Serial Peripheral Interface”, Microchip Technology Inc., 2000.
- [10] “Universal Serial Bus Revision 3.0 Specification”, USB Implementers Forum, Novembro 2008.
- [11] <http://focus.ti.com/mcu/docs/mcuprodooverview.tsp?sectionId=95&tabId=140&familyId=342>, Outubro 2009.
- [12] “dsPIC30F6012A Data Sheet”, Microchip Technology Inc., 2008.
- [13] “LM2673 3A Step-Down Voltage Regulator with Adjustable Current Limit Data Sheet”, National Semiconductor, Fevereiro 2008.
- [14] C. Pinto, J. Santos, “Activador e Ensaizador de Pilhas de Hidrogénio em Ambiente de Produção”, IST, Outubro 2006.
- [15] “PIC18F4550 Data Sheet”, Microchip Technology Inc., 2007.
- [16] “MCP2515 Stand-Alone CAN Controller With SPI Interface”, Microchip Technology Inc., 2007.
- [17] “MCP2551 High Speed CAN Transceiver Data Sheet”, Microchip Technology Inc., 2007.
- [18] Pedro Lopes, “Sistema de Sensores para Carros de Competição Integrado na Formula Student”, IST, Lisboa, Setembro 2008.
- [19] “LM386 Low Voltage Audio Power Amplifier Data Sheet”, National Semiconductor, Agosto 2000.
- [20] “Analysis of the Sallen-Key Architecture Application Report”, Texas instruments, Julho 1999.
- [21] http://en.wikipedia.org/wiki/Wien-Planck_law, Outubro 2009.
- [22] “MLX90614 Family Data Sheet”, Melexis, 2009.
- [23] “Self-Calibrating TPOS Speed Sensor IC Optimized for Automotive Cam Sensing Applications Data Sheet”, Allegro Microsystems, Inc.

- [24] "MPXY8000 Technical Data", Freescale Semiconductor, 2006.
- [25] "rfPIC12F675F Data Sheet", Microchip Technology Inc., 2003.
- [26] http://en.wikipedia.org/wiki/List_of_battery_sizes, Outubro 2009.
- [27] <http://www.microchip.com>, Outubro 2009.
- [28] "rfRXD0420 Data Sheet", Microchip Technology Inc., 2003.
- [29] Steven Bible, "rfRXD0420 Receiver Module", Microchip Technology Inc., 2003.
- [30] "SureSeal Catalog", SureSeal Connections, 2008.
- [31] "Tire Pressure Monitoring Sensor Product Specification", Freescale Semiconductor Inc., 2009.

Anexo A – Esquemáticos das PCI utilizadas

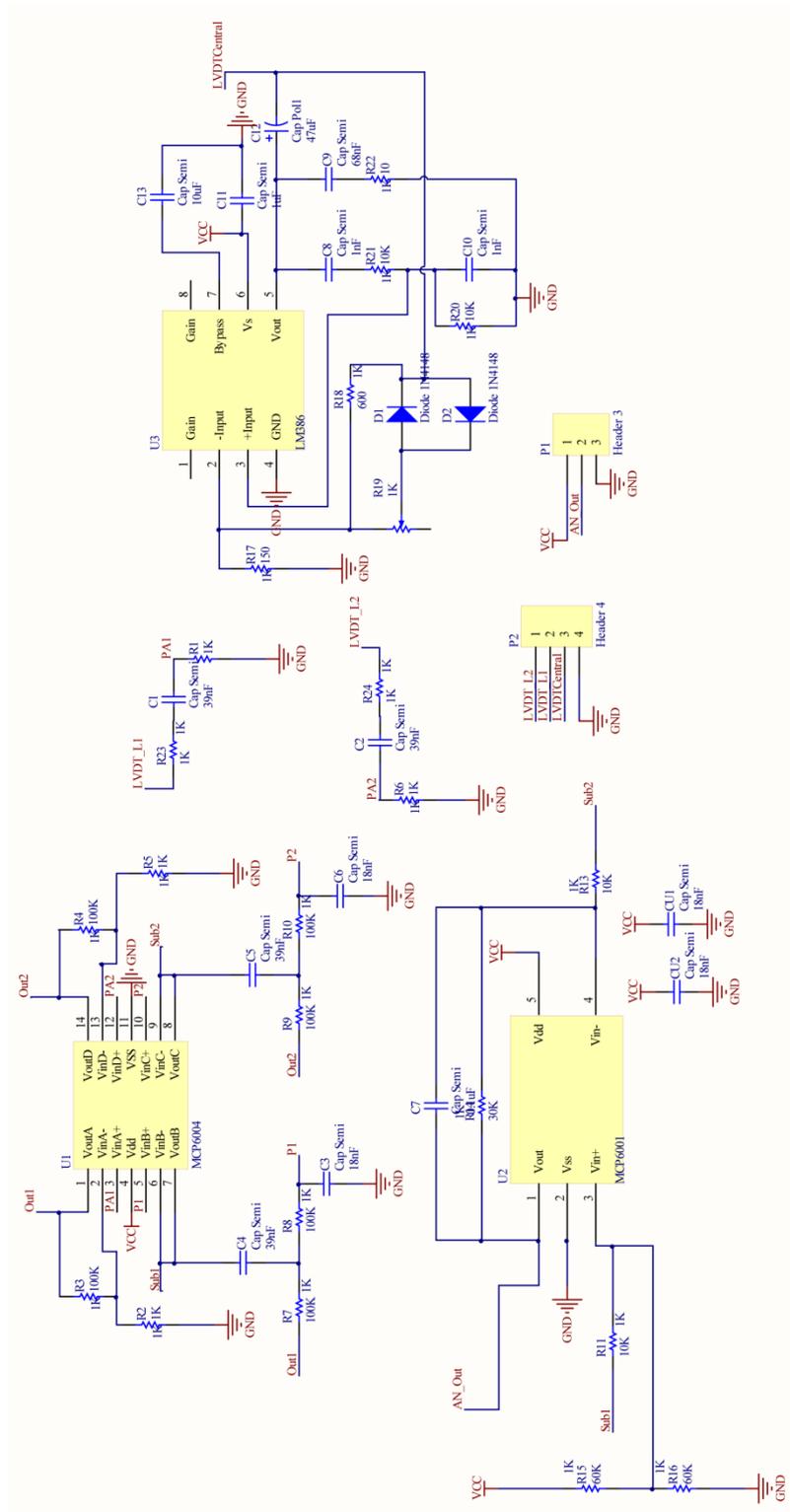


Figura 63 – Esquemático da PCI de excitação e condicionamento de sinal de um LVDT.

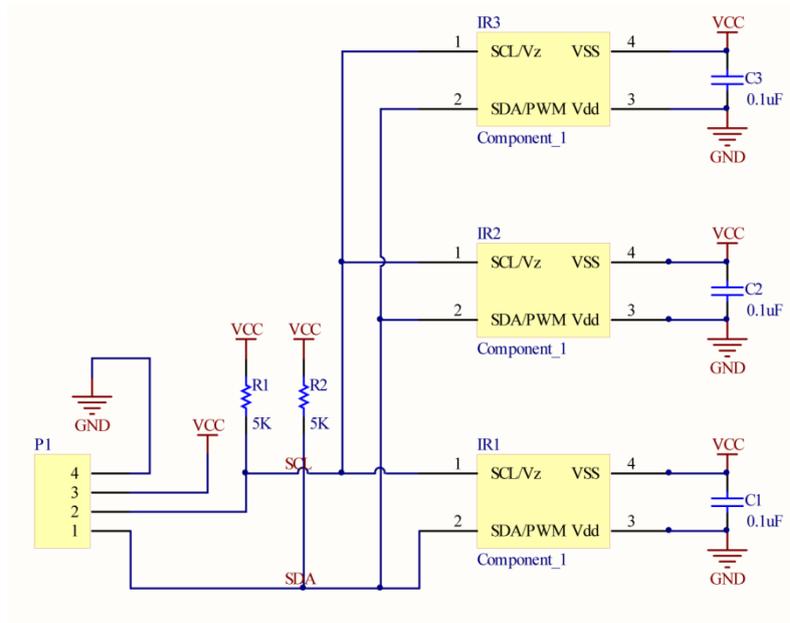


Figura 64 – Esquemático da PCI de suporte aos sensores de temperatura da superfície dos pneus.

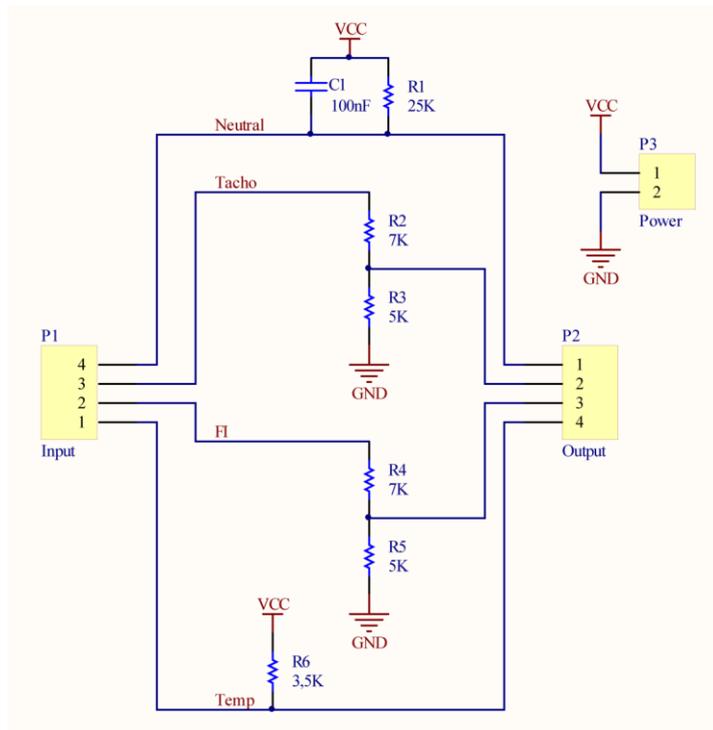


Figura 65 – Esquemático da PCI de suporte aos sensores fornecidos pelo fabricante do motor.

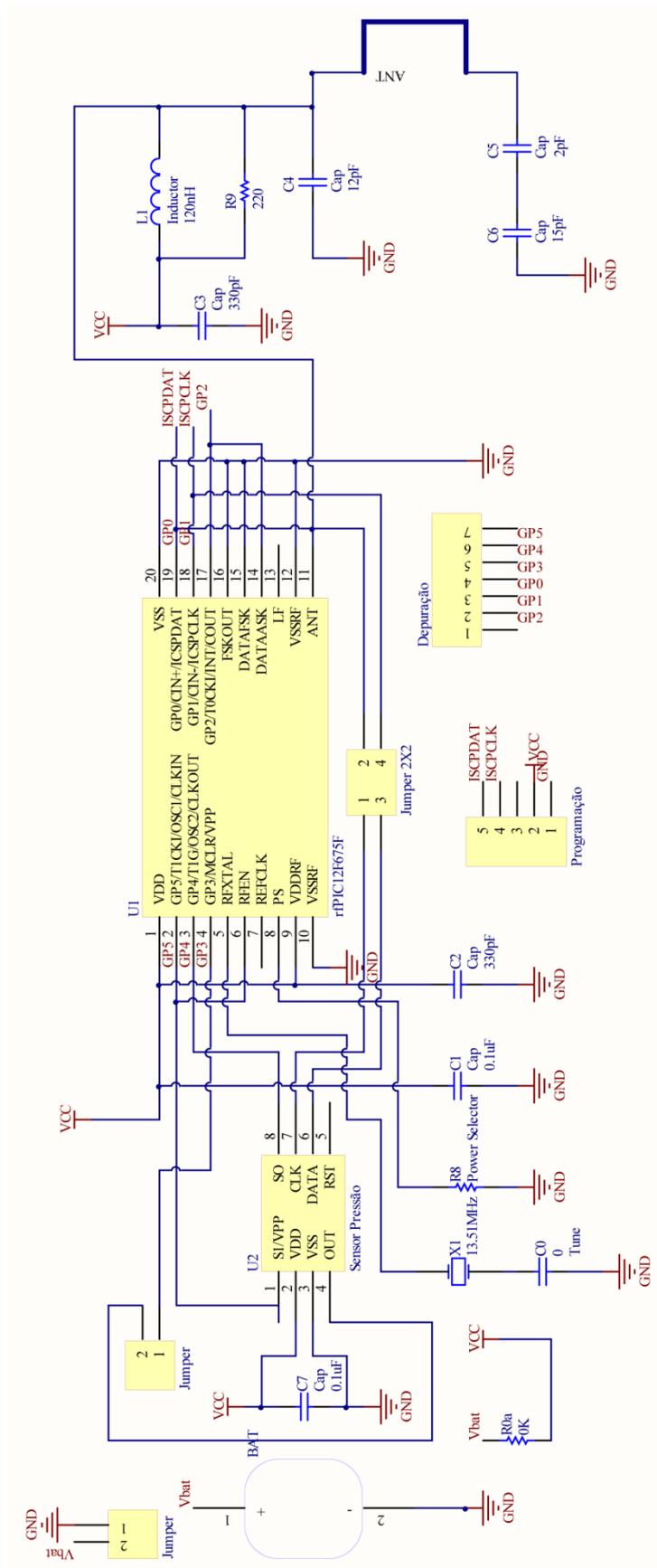


Figura 66 – Esquemático da PCI de suporte ao sensor de pressão e emissor de radiofrequência.

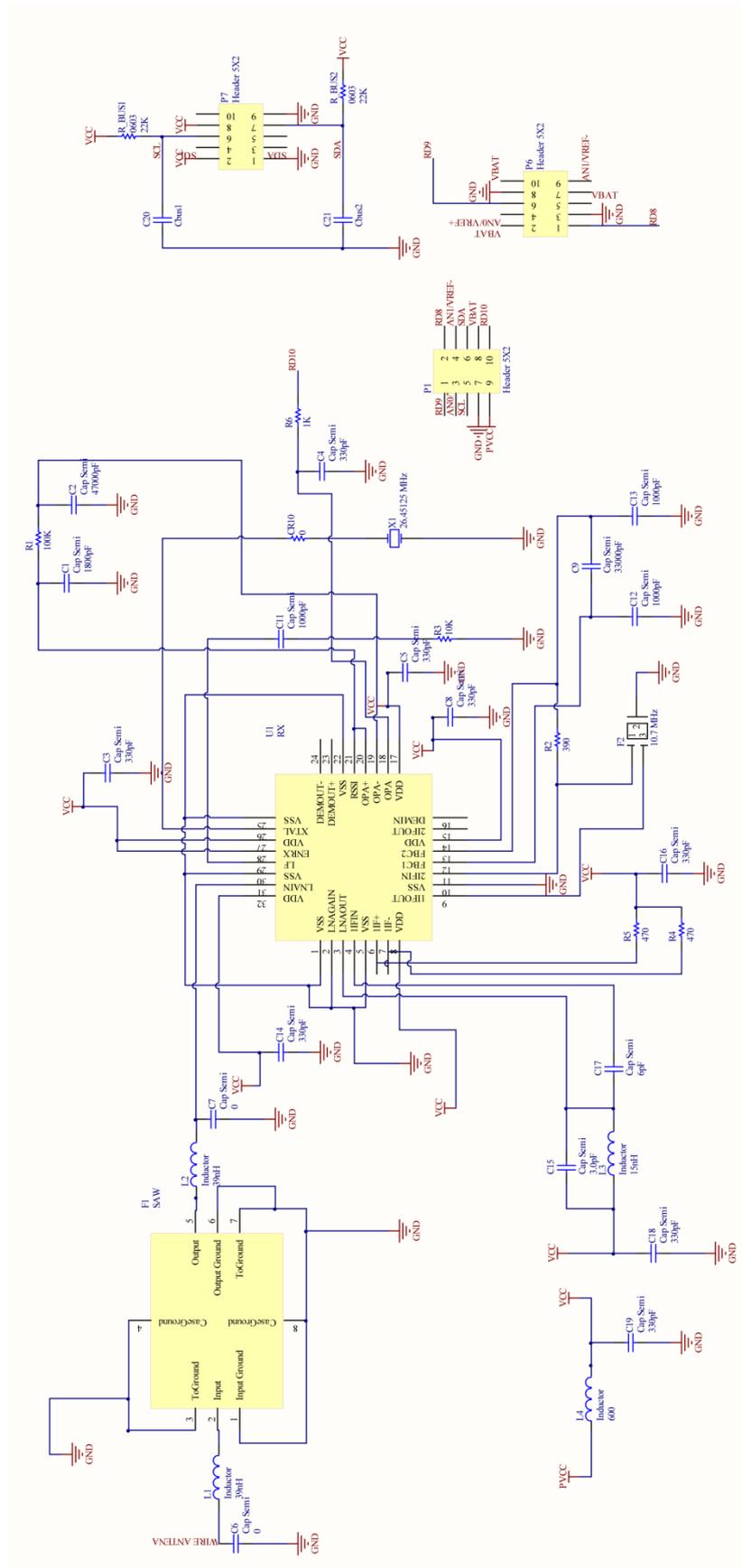


Figura 67 – Esquemático da PCI de suporte ao receptor de rádio frequência.