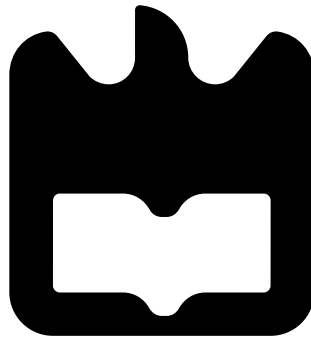




**Luís Emanuel Ferreira  
Cristóvão**

**Sistema de Telemetria para o concurso  
MotoStudent**







**Luís Emanuel Ferreira  
Cristóvão**

**Sistema de Telemetria para o concurso  
MotoStudent**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Automação Industrial, realizada sob a orientação de Pedro Nicolau Faria da Fonseca, Professor do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e co-orientação científica de Paulo Bacelar Reis Pedreiras, Professor do Departamento Electrónica, Telecomunicações e Informática da Universidade de Aveiro.



## **O júri**

Presidente

**Prof. Doutor Miguel Armando Riem de Oliveira**

Professor Auxiliar da Universidade de Aveiro

Vogais

**Prof. Doutor Frederico Miguel do Céu Marques dos Santos**

Professor Adjunto do Instituto Superior de Engenharia de Coimbra (arguente)

**Prof. Doutor Pedro Nicolau Faria da Fonseca**

Professor Auxiliar da Universidade de Aveiro (orientador)



## **Agradecimentos**

Em primeiro, à minha família por todo o apoio e ajuda durante todo o meu percurso académico, em especial à minha avó Joaquina.

Aos meus orientadores, Professor Pedro Nicolau Faria da Fonseca e Professor Paulo Bacelar Reis Pedreiras, por aceitarem este tema, pelos conselhos e ideias partilhadas ao longo de todo o projeto e pelo o seu tempo disponibilizado para o mesmo.

À equipa Motochanics por me apresentar este tema, pela aprendizagem adquirida e por ajudar a tornar este projeto real.

Aos meus amigos e colegas pelas ideias partilhadas e pelos bons e maus momentos.

Por último, mas não menos importante, a todas as empresas que acreditaram no projeto e ajudaram em várias partes do desenvolvimento do mesmo.





## Resumo

Nos dias que correm, na área da competição motorizada a recolha de dados é uma ferramenta bastante importante para o aumento da eficiência dos veículos em prova. A equipa MotochanicsUA formada por estudantes da Universidade de Aveiro, que conta com duas participações na competição internacional Motostudent, pretende melhorar a sua prestação em prova. A equipa com a experiência adquirida, ao longo do tempo concluiu que a falta de dados do desempenho do protótipo em pista é uma lacuna no desenvolvimento do mesmo. Esta equipa com baixo orçamento verificou que a aquisição de uma solução de mercado para a recolha de dados era demasiado cara, surgindo assim a ideia de desenvolver uma solução de telemetria adequada às suas necessidades.

Neste projeto é desenvolvida uma solução de recolha e transmissão de dados em tempo real entre a mota e a *box*. Inicialmente foi feito um estudo das tecnologias que permitem a aquisição das grandezas requeridas pela equipa, dos métodos de transmissão de dados com e sem fios e de algumas soluções disponíveis no mercado. Após o estudo realizado, é apresentada a arquitetura e uma proposta de implementação para a solução. Nesta sequência é apresentado todo o *hardware* desenvolvido para a solução, desde de um sistema que faz toda a aquisição e transmissão de dados até outro que faz a receção dos mesmos. Para o correto funcionamento do *hardware* é apresentado o desenvolvimento de *firmware*, com as capacidades de inicializar as componentes implementadas e de executar as tarefas necessárias, para que a solução cumpra com os requisitos propostos. Para verificar o correto funcionamento da solução, são apresentados vários testes às componentes individualmente e com a mota em pista.

Com este projeto é obtido um protótipo de um sistema de telemetria para a equipa Motochanics, com a capacidade de adquirir, guardar, transmitir e apresentar dados em tempo real.



## **Abstract**

Nowadays, data acquisition is a very important tool on motorized competition matters in order to improve vehicle performance on the race. MotochanicsUA team formatted by Aveiro University students count with two participations in the international competition Motostudent, and has objective to improve her performance in future events. With acquired experience along the time concluded that the lack of data about performance of their prototype on the track is a relevant gap in its development. With a low budget and expensive market offers to telemetry systems, the idea to develop one system with dedicated specs was born.

In this project was developed a solution to solve this gap. It allows the acquisition and record of all bike sensors data and send it to pit in real time. At beginning looking to team needs was made a study about possible technologies for data type acquisition technologies, transmission type (wireless or wired communications) and available market solutions. After study is presented for solution the architecture and application proposal. Next, all the hardware developed for solution is detailed, from data acquisition, record and transmission system on bike to another system that does the data reception on the pit. For correct hardware work, the firmware development is presented with abilities for initialize all components and perform necessary tasks in order to made solution accomplish with the necessary requirements. To check the correct working of the solution, several tests are carried out individually on the components implemented and with the moto on track.

With this project, a prototype of a telemetry system for the Motochanics team is obtained, with the ability to acquire, store, transmit and present data in real time.



# Conteúdo

<b>Lista de Figuras</b>	<b>v</b>
<b>Lista de Tabelas</b>	<b>ix</b>
<b>Lista de Siglas</b>	<b>xi</b>
<b>1 Motivação e objetivos</b>	<b>1</b>
1.1 A competição e os seus requisitos . . . . .	1
1.2 A equipa <i>Motochanics</i> . . . . .	3
1.3 Apresentação do problema . . . . .	3
1.4 História da telemetria . . . . .	5
1.4.1 Telemetria em competições motorizadas . . . . .	5
1.5 Estrutura do documento . . . . .	5
<b>2 Requisitos e estado da arte</b>	<b>7</b>
2.1 Levantamento de sensores e taxa de transmissão de dados . . . . .	7
2.2 Medição de grandezas físicas . . . . .	8
2.2.1 Temperatura . . . . .	9
2.2.2 Pressão . . . . .	10
2.2.3 Deslocamento linear . . . . .	11
2.3 Comunicações entre dispositivos . . . . .	12
2.3.1 Protocolos de comunicação com fios . . . . .	12
2.3.1.1 <i>Universal Asynchronous Receiver Transmitter</i> (UART) . . . . .	12
2.3.1.2 <i>Inter-Integrated Circuit</i> (I <sup>2</sup> C) . . . . .	13
2.3.1.3 <i>Serial Peripheral Interface</i> (SPI) . . . . .	14
2.3.1.4 <i>Controller Area Network</i> (CAN) . . . . .	15
2.3.2 Comunicação sem fios de longo alcance . . . . .	17
2.3.2.1 Serviços móveis associados a uma operadora . . . . .	17
2.3.2.2 Utilização de uma banda <i>ISM</i> . . . . .	20
2.3.2.3 Análise qualitativa do método de comunicação sem fios . . . . .	22
2.4 Soluções de mercado . . . . .	23
2.5 Requisitos para a solução . . . . .	25
<b>3 Arquitetura e proposta de implementação</b>	<b>27</b>
3.1 Constituição dos sistemas da solução . . . . .	27
3.1.1 Sistema de aquisição, gravação e transmissão de dados . . . . .	27
3.1.2 Sistema de receção de dados . . . . .	28
3.2 Escolha de sensores e dispositivos . . . . .	29
3.2.1 Aquisição de valores da ECU . . . . .	29
3.2.2 Módulos para sensores adicionais . . . . .	30

3.2.2.1	Temperatura dos pneus . . . . .	31
3.2.2.2	Pressão dos travões . . . . .	32
3.2.2.3	Posição das suspensões . . . . .	32
3.2.3	Módulo inercial . . . . .	33
3.2.4	Módulo GPS . . . . .	34
3.2.5	Armazenamento de dados em memória não volátil . . . . .	35
3.2.6	Porta de <i>debug</i> e transmissão de dados para ecrã . . . . .	35
3.2.7	Transmissão de dados sem fios . . . . .	36
3.3	Estudo da unidade de processamento a utilizar na solução . . . . .	37
3.4	Arquitetura da solução . . . . .	38
<b>4</b>	<b>Desenvolvimento de <i>hardware</i></b> . . . . .	<b>41</b>
4.1	<i>Hardware</i> comum . . . . .	41
4.1.1	Alimentação . . . . .	42
4.1.2	Portas para programação e <i>debugging</i> do microcontrolador . . . . .	45
4.1.2.1	Interface para ICSP . . . . .	45
4.1.2.2	UART1 . . . . .	46
4.1.3	Oscilador externo . . . . .	47
4.2	Módulo para medição de sensores adicionais . . . . .	48
4.2.1	Entradas analógicas . . . . .	48
4.2.2	Sensor de temperatura . . . . .	50
4.2.3	<i>Transceiver</i> CAN . . . . .	51
4.3	Sistema de aquisição, gravação e transmissão de dados . . . . .	51
4.3.1	<i>Transceivers</i> CAN . . . . .	51
4.3.2	Medição de acelerações . . . . .	52
4.3.3	Módulo GPS . . . . .	52
4.3.4	Transmissão de dados para ecrã . . . . .	53
4.3.5	Armazenamento de dados em memória não volátil . . . . .	53
4.3.6	Transmissão de dados RF . . . . .	54
4.4	Sistema para receção de dados . . . . .	55
4.5	Desenvolvimento de PCB's . . . . .	55
4.5.1	Módulo para aquisição de sensores adicionais . . . . .	56
4.5.2	Sistema de aquisição, gravação e transmissão de dados . . . . .	57
4.5.3	Sistema para receção de dados . . . . .	57
4.6	Custo de construção da solução . . . . .	58
<b>5</b>	<b>Desenvolvimento de <i>firmware</i></b> . . . . .	<b>61</b>
5.1	Ambiente de desenvolvimento . . . . .	61
5.2	Configurações globais da solução . . . . .	62
5.2.1	Configuração das fontes de relógio . . . . .	62
5.2.2	Estrutura de dados para grandezas . . . . .	63
5.3	Módulo para medição de sensores adicionais . . . . .	63
5.3.1	Função principal e inicialização das interfaces . . . . .	65
5.3.2	Configurações . . . . .	65
5.3.2.1	Aquisição de dados . . . . .	66
5.3.2.2	Transmissão de dados . . . . .	68
5.3.2.3	Temporizador . . . . .	69
5.3.3	Rotina de interrupção . . . . .	70
5.4	Sistema de aquisição, gravação e transmissão de dados . . . . .	71
5.4.1	Função principal e inicialização das interfaces . . . . .	71

5.4.2	Configurações . . . . .	71
5.4.2.1	Aquisição de dados . . . . .	72
5.4.2.2	Gravação de dados . . . . .	77
5.4.2.3	Transmissão de dados . . . . .	78
5.4.2.4	Temporizadores . . . . .	83
5.4.2.5	Interrupções . . . . .	85
5.4.3	Rotinas de interrupção . . . . .	86
5.4.3.1	Leitura da ECU . . . . .	87
5.4.3.2	Receção de dados dos módulos para sensores adicionais . . . . .	88
5.4.3.3	Receção de dados do módulo GPS . . . . .	88
5.4.3.4	Leitura do módulo inercial . . . . .	90
5.4.3.5	Interrupções associadas à transmissão de dados . . . . .	91
5.4.3.6	Cronometrar tempo e guardar dados no cartão microSD . . . . .	92
5.5	Sistema para a receção de dados . . . . .	93
5.5.1	Função <i>main</i> e inicialização das interfaces . . . . .	94
5.5.2	Configuração de <i>interfaces</i> . . . . .	95
5.5.2.1	UART1 . . . . .	95
5.5.2.2	SPI - <i>Transceiver</i> RFM95PW . . . . .	95
5.5.3	Rotina de interrupção . . . . .	98
<b>6</b>	<b>Testes e validações</b>	<b>99</b>
6.1	Ferramentas de teste utilizadas . . . . .	99
6.1.1	Instrumento de medida . . . . .	99
6.1.2	Interface gráfica desenvolvida . . . . .	100
6.2	Testes individuais às funções da solução . . . . .	101
6.2.1	Leitura dos sensores analógicos . . . . .	102
6.2.2	Leitura de temperaturas através do pirómetro . . . . .	103
6.2.3	Leitura da ECU . . . . .	104
6.2.4	Recepção de dados dos módulos de sensores adicionais . . . . .	106
6.2.5	Leitura do módulo inercial . . . . .	108
6.2.6	Leitura do módulo GPS . . . . .	110
6.2.7	Comunicação com cartão <i>microSD</i> . . . . .	111
6.2.8	Transmissão de dados para interface gráfica . . . . .	113
6.2.9	Transmissão de dados para o ecrã . . . . .	116
6.2.10	Peso e dimensões da solução . . . . .	117
6.3	Teste à solução desenvolvida em pista . . . . .	118
<b>7</b>	<b>Conclusões e trabalhos futuros</b>	<b>121</b>
7.1	Especificações do sistema . . . . .	121
7.2	Comparação com os produtos de mercado . . . . .	122
7.3	Trabalhos futuros . . . . .	123
	<b>Bibliografia</b>	<b>125</b>
	<b>Anexo A Módulo para sensores adicionais</b>	<b>131</b>
A.1	Esquemático . . . . .	131
A.2	<i>Layout</i> da PCB . . . . .	139
A.3	Custos de produção da PCB . . . . .	145
A.4	Desenhos técnicos caixa de protecção . . . . .	147

<b>Anexo B Sistema de aquisição, gravação e transmissão de dados</b>	<b>153</b>
B.1 Esquemático . . . . .	153
B.2 <i>Layout</i> da PCB . . . . .	161
B.3 Custo de produção da PCB . . . . .	167
B.4 Desenhos técnicos caixa de proteção . . . . .	169
<b>Anexo C Sistema para receção de dados</b>	<b>175</b>
C.1 Esquemático . . . . .	175
C.2 <i>Layout</i> da PCB . . . . .	181
C.3 Custo de produção da PCB . . . . .	187
C.4 Desenhos técnicos caixa de proteção . . . . .	189
<b>Anexo D Desenvolvimento de <i>firmware</i></b>	<b>195</b>
D.1 Sistema de aquisição, gravação e transmissão de dados . . . . .	195
D.1.1 <i>Array</i> dos pedido a realizar à ECU . . . . .	195
<b>Anexo E Resultados dos testes realizados</b>	<b>197</b>
E.1 Leitura de temperaturas através do pirómetro . . . . .	197
E.1.1 Dados do teste à comunicação com o pirómetro . . . . .	197
E.1.2 Dados do teste à taxa de aquisição ao pirómetro . . . . .	197
E.2 Leitura da ECU . . . . .	197
E.2.1 Dados do teste à taxa de aquisição da ECU . . . . .	197
E.3 Recepção de dados dos módulos de sensores adicionais . . . . .	197
E.3.1 Captura efetuada no barramento CAN2 . . . . .	197
E.3.2 Teste ao número máximo de pacotes transmitidas pelo barramento CAN2 . . . . .	197
E.4 Leitura do módulo inercial . . . . .	197
E.4.1 Dados do teste à taxa de aquisição do módulo inercial . . . . .	197
E.5 Leitura do módulo GPS . . . . .	198
E.5.1 Captura das mensagens transmitidas pelo módulo GPS . . . . .	198
E.6 Escrita no cartão micro SD . . . . .	198
E.6.1 Dados guardados no cartão durante 20 minutos . . . . .	198
E.7 Transmissão de dados por banda ISM . . . . .	198
E.7.1 Dados recebidos a uma distância de 1 km . . . . .	198
E.7.2 Dados recebidos a uma distância de 2.6 km . . . . .	198
E.7.3 Mensagens recebidas pela interface gráfica gráfica . . . . .	198
E.8 Transmissão de dados para ecrã . . . . .	198
E.8.1 Captura das mensagens transmitidas para o ecrã . . . . .	198
E.9 Teste em pista . . . . .	198
E.9.1 Dados recolhidos durante o teste em pista . . . . .	198
E.9.2 Filme do teste em pista . . . . .	198



# Lista de Figuras

1.1	Logótipos das duas classes da competição <i>MotoStudent</i> [1]. . . . .	1
1.2	Circuito de Aragon para a competição <i>MotoStudent</i> [2]. . . . .	2
1.3	Logótipo da equipa <i>Motochanics</i> . . . . .	3
1.4	Prototipo desenvolvido para a edição de 2018. . . . .	3
1.5	Diagrama geral da solução a desenvolver. . . . .	4
2.1	Ficha diagnóstico e adaptador OBD2. . . . .	8
2.2	Conector para diagnóstico presente na moto. . . . .	8
2.3	Comparação entre NTC e PTC [7]. . . . .	10
2.4	Pirómetro [8]. . . . .	11
2.5	Efeito piezoelétrico [9]. . . . .	11
2.6	Potenciómetro linear com o formato cilíndrico [10]. . . . .	11
2.7	Exemplo de ligação para comunicação UART. . . . .	12
2.8	Estrutura da mensagem do protocolo UART. . . . .	13
2.9	Exemplo de um barramento I <sup>2</sup> C [14]. . . . .	13
2.10	Exemplo de escrita através de I <sup>2</sup> C [15]. . . . .	14
2.11	Exemplo de leitura através de I <sup>2</sup> C [15]. . . . .	14
2.12	Barramento SPI padrão. . . . .	15
2.13	Exemplo de troca de dados num barramento SPI [17]. . . . .	15
2.14	Barramento CAN padrão. . . . .	15
2.15	Constituição do tempo para transmissão de um <i>bit</i> [20]. . . . .	17
2.16	Diferenças das ondas portadoras obtidas, para diferentes tipos de modulação. . . . .	22
2.17	Relação entre BER e SNR para ASK, FSK, PSK e OOK. . . . .	22
2.18	<i>AiM MXm Compact Dash Data Logger</i> [33]. . . . .	23
2.19	<i>Starlane Davinci-II S Dashboard</i> [34]. . . . .	24
3.1	Diagrama de blocos do sistema de aquisição, gravação e transmissão de dados. . . . .	28
3.2	Diagrama de blocos do sistema de receção e apresentação de dados. . . . .	28
3.3	Duas possibilidades de comunicação com um barramento <i>Controller Area Network</i> (CAN). . . . .	29
3.4	Módulos GPS em estudo, sendo (a) Quectel L80 e (b) ORG4472. . . . .	34
3.5	Diagrama de blocos de um cartão <i>Secure Digital</i> (SD). . . . .	35
3.6	Placa de desenvolvimento Max32 [61]. . . . .	37
3.7	Arquitetura geral do sistema a instalar na moto. . . . .	38
3.8	Diagrama de blocos do módulo para sensores adicionais. . . . .	38
3.9	Diagrama de blocos do sistema para aquisição de dados. . . . .	39
3.10	Diagrama de blocos do sistema para receção de dados. . . . .	39

4.1	Teste individual ao consumo de energia do <i>transceiver</i> rádio. . . . .	42
4.2	Reguladores de tensão para alimentação do sistema principal. . . . .	44
4.3	Reguladores de tensão para alimentação dos módulos e do sistema de recepção de dados. . . . .	44
4.4	Circuito de comutação para alimentação de 5V. . . . .	44
4.5	Circuito de programação, <i>debugging</i> e reinicialização do sistema. . . . .	46
4.6	Circuito para interface UART1. . . . .	46
4.7	Circuito para o oscilador do sistema. . . . .	48
4.8	Circuito para o regulador de tensão de 10V. . . . .	48
4.9	Circuito para filtro <i>Sallen-Key</i> . . . . .	49
4.10	Circuito de amplificação para acondicionamento de sinal. . . . .	50
4.11	Regulador de tensão de 2,5V. . . . .	50
4.12	Circuito de ligação do sensor MLX60914. . . . .	51
4.13	Circuito para o <i>transceiver</i> CAN. . . . .	51
4.14	Circuitos para <i>transceivers</i> CAN. . . . .	52
4.15	Circuito para módulo MM7150. . . . .	52
4.16	Circuito para módulo Quectel L80. . . . .	53
4.17	Circuito para comunicação com o ecrã. . . . .	53
4.18	Descrição dos pinos de um cartão microSD [77]. . . . .	54
4.19	Circuito para cartão microSD. . . . .	54
4.20	Circuito para o <i>transceiver</i> RFM95PW no sistema de aquisição. . . . .	55
4.21	Construção do módulo para sensores adicionais. . . . .	56
4.22	Construção do sistema para aquisição de dados. . . . .	57
4.23	PCB do sistema de recepção de dados. . . . .	58
5.1	Fluxograma da rotina para inicialização das <i>interfaces</i> . . . . .	65
5.2	Fluxograma da rotina de interrupção do <i>Timer2</i> . . . . .	70
5.3	Fluxograma da rotina para inicialização das interfaces. . . . .	71
5.4	Formato dos pedidos e resposta à ECU. . . . .	73
5.5	Fluxograma de inicialização do cartão <i>microSD</i> . . . . .	78
5.6	Máquina de estados implementada no módulo transmissor. . . . .	82
5.7	Rotina de interrupção associada ao controlador CAN1. . . . .	88
5.8	Rotina de interrupção associada ao controlador CAN2. . . . .	89
5.9	Rotina de interrupção na recepção de dados por UART5. . . . .	90
5.10	Rotina de interrupção associada ao módulo inercial. . . . .	91
5.11	Formato da mensagem enviada pelo módulo RF. . . . .	91
5.12	Rotina de interrupção para INT1. . . . .	92
5.13	Rotina de interrupção para <i>Timer2</i> . . . . .	92
5.14	Rotina de interrupção para <i>Timer3</i> . . . . .	94
5.15	Fluxograma da rotina para inicialização das interfaces. . . . .	94
5.16	Máquina de estados implementada no modulo RF. . . . .	96
5.17	Rotina de interrupção para INT1. . . . .	98
6.1	Osciloscópio <i>PicoScope 2207B</i> e o respetivo <i>software</i> . . . . .	100
6.2	Tabela de variáveis da interface gráfica. . . . .	101
6.3	Aspeto final da interface gráfica. . . . .	102
6.4	Ligação efetuada para o teste do acondicionamento de sinal. . . . .	102
6.5	Sinais obtidos durante o teste ao acondicionamento de sinal. . . . .	103
6.6	Ligação efetuada para o teste da comunicação com o pirómetro. . . . .	104
6.7	Comunicação I2C com o pirómetro para leitura das temperaturas. . . . .	104

6.8	Ligação efetuada para realização de testes ao barramento CAN1. . . . .	105
6.9	Exemplo de um conjunto de dados obtidos do barramento CAN1. . . . .	105
6.10	Ligação efetuada para realização de testes ao barramento CAN2 do sistema de aquisição. . . . .	107
6.11	Pacotes transmitidos pelo barramento dedicado aos sensores adicionais. . .	107
6.12	Ligação efetuada para o teste da comunicação com o módulo inercial. . . .	108
6.13	Sinais capturados durante a leitura do módulo inercial. . . . .	109
6.14	Ligação efetuada para realização de testes ao módulo GPS. . . . .	110
6.15	Sinais capturados durante o teste da comunicação com o módulo GPS. . .	110
6.16	Ligação efetuada para os testes à comunicação com o cartão microSD. . .	111
6.17	Exemplo de um pacote guardado no cartão SD. . . . .	112
6.18	Taxa de gravação de dados no cartão microSD durante 20 minutos. . . .	112
6.19	Ligação efetuada para os testes à comunicação com módulo rádio. . . .	113
6.20	Amostra do barramento SPI durante transmissão de um pacote para o módulo rádio. . . . .	113
6.21	Amostra de barramento SPI durante transmissão de 10 pacotes. . . . .	114
6.22	Localização dos sistemas durante o terceiro teste. . . . .	114
6.23	Resumo dos dados ao teste da distância de transmissão entre sistemas. . .	115
6.24	Taxa de receção de dados na interface gráfica durante 20 minutos. . . .	116
6.25	Ligação efetuada para os testes à comunicação com <i>ecrã</i> . . . . .	116
6.26	Pacote enviado para o ecrã capturado pelo osciloscópio. . . . .	117
6.27	Circuito do kartódromo internacional de Leiria. . . . .	119
6.28	Sistema de aquisição de dados instalado na mota. . . . .	119



# Lista de Tabelas

1.1	Sensores e taxas de leitura requeridas pela equipa. . . . .	4
2.1	Valores a serem lidos da ECU ou por novos sensores. . . . .	9
2.2	Estudo das grandezas/sensores a obter e a sua taxa de dados. . . . .	9
2.3	Constituição de uma mensagem <i>standard</i> . . . . .	16
2.4	Constituição de uma mensagem extensa. . . . .	16
2.5	Especificações do sistema GSM. . . . .	18
2.6	Especificações do sistema UMTS. . . . .	19
2.7	Especificações do sistema LTE. . . . .	19
2.8	Bandas ISM Sub-GHz de uso geral na Europa. . . . .	20
2.9	Comparação entre as diferentes faixas de frequência ISM. . . . .	20
2.10	Tabela de comparação para diferentes tipos de modulação. . . . .	23
2.11	Características <i>AiM MXm Compact Motorcycle Dash DataLogger</i> . . . . .	24
2.12	Estudo do custo para aquisição de uma solução <i>AiM</i> . . . . .	24
2.13	Características <i>Starlane Davinci-II S</i> . . . . .	25
2.14	Estudo do custo para aquisição de uma solução <i>Starlane</i> . . . . .	25
3.1	Vantagem e desvantagens da descentralização do sistema. . . . .	30
3.2	Características de dois pirômetros da marca <i>Melexis</i> . . . . .	31
3.3	Características de dois sensores de pressão. . . . .	32
3.4	Características de dois potenciômetros lineares. . . . .	33
3.5	Características de dois módulos inerciais. . . . .	34
3.6	Características de dois dispositivos GPS. . . . .	35
3.7	Características de duas interfaces <i>Universal Serial Bus (USB)-Universal Asynchronous Receiver-Transmitter (UART)</i> . . . . .	36
3.8	Características de três <i>transceivers</i> RF. . . . .	36
3.9	Levantamento do número de interfaces da unidade de processamento. . . . .	37
4.1	Estimativa do consumo de corrente. . . . .	43
4.2	Características de dois conversores CC-CC [65][66]. . . . .	43
4.3	Seleção de condensadores conforme fonte de relógio. . . . .	47
4.4	Custos totais da solução. . . . .	58
5.1	Comparação entre duas estruturas de dados para variáveis adquiridas. . . . .	64
5.2	Número de <i>bytes</i> por mensagem. . . . .	80
5.3	Configurações efetuadas ao módulo RFM95PW [89]. . . . .	82
5.4	Valores de PR2 perante os valores de frequência e <i>Prescale</i> . . . . .	84
5.5	Resultados do teste realizado ao número de pedidos à ECU. . . . .	87
5.6	Frequências de aquisição para as variáveis adquiridas da ECU. . . . .	87
5.7	Exemplo de uma mensagem do tipo GPRMC. . . . .	89

5.8	Configurações efetuadas ao módulo RFM95PW. . . . .	96
6.1	Especificações do osciloscópio <i>PicoScope 2207B</i> [94]. . . . .	100
6.2	Dados do teste à comunicação com o pirómetro. . . . .	104
6.3	Dados do teste à taxa de aquisição ao pirómetro. . . . .	105
6.4	Pacotes recolhidos no primeiro teste executado ao barramento CAN1. . .	105
6.5	Resumo dos pacotes capturados no barramento CAN1 durante 10s. . . . .	106
6.6	Pacotes recolhidos no primeiro teste executado ao barramento CAN2 . . .	107
6.7	Resumo da captura efetuada no barramento CAN2. . . . .	107
6.8	Resumo do teste ao número máximo de pacotes enviados pelo barramento CAN2. . . . .	108
6.9	Dados do teste à leitura do módulo inercial. . . . .	109
6.10	Dados do teste à taxa de aquisição ao módulo inercial. . . . .	109
6.11	Pacotes transmitidos pelo módulo GPS durante a captura de dados. . . .	110
6.12	Resumo dos dados ao teste à taxa de aquisição do módulo GPS. . . . .	111
6.13	Resumo dos dados guardados no cartão microSD. . . . .	113
6.14	Resumo dos dados obtidos no teste ao alcance da comunicação rádio. . . .	115
6.15	Resumo dos dados recebidos pela interface gráfica. . . . .	116
6.16	Resumo dos dados ao teste da taxa de transmissão para o ecrã. . . . .	117
6.17	Peso por componente do sistema de aquisição e transmissão de dados. . .	118
6.18	Diferenças entre os casos de estudo do sistema de aquisição. . . . .	118
6.19	Peso por componente do sistema de receção de dados. . . . .	118
6.20	Resumo dos testes efetuados em pista. . . . .	120
7.1	Características da solução desenvolvida. . . . .	122
7.2	Comparação entre os produtos de mercado e a solução desenvolvida. . . .	123
A.1	Custos de produção do módulo de sensores adicionais. . . . .	145
B.1	Custos de produção do sistema para aquisição, recolha e transmissão de dados. . . . .	167
C.1	Custos de produção do sistema para receção de dados. . . . .	187

# Lista de Siglas

**3GPP** *Third Generation Partnership Project.*

**ACK** *Acknowledge.*

**ADC** *Analog to Digital Converter.*

**AFA** *Adaptative Frequency Agility.*

**ANACOM** *Autoridade Nacional de Comunicações.*

**ASK** *Amplitude Shift Keying.*

**BER** *Bit Error Rate.*

**BPSK** *Binary Phase-Shift Keying.*

**CAD** *Computer Aided Design.*

**CAN** *Controller Area Network.*

**CC** *Corrente Contínua.*

**CI** *Circuito Integrado.*

**CRC** *Cyclic Redundancy Check.*

**DAC** *Digital to Analog Converter.*

**DLC** *Data Length Code.*

**ECU** *Engine Control Unit.*

**EEPROM** *Electrically Erasable Programmable Read Only Memory.*

**EOF** *End Of Frame.*

**ESD** *ElectroStatic Discharge.*

**FAT** *File Allocation Table.*

**FDMA** *Frequency Division Multiple Access.*

**FFF** *Fused Filament Fabrication.*

**FIFO** *First In First Out.*

**FSK** *Frequency Shift Keying.*

**GMSK** *Gaussian Minimum Shift Keying.*

**GPS** *Global Positioning System.*

**GSM** *Global System for Mobile Communication.*

**HID** *Human Interface Device.*

**I<sup>2</sup>C** *Inter-Integrated Circuit.*

**ICSP** *In Circuit Serial Programming.*

**IDE** *Identifier Extension Bit.*

**IDE** *Integrated Development Environment.*

**IFS** *Inter Frame Space.*

**ISM** *Industrial, Scientific and Medical.*

**LBT** *Listen Before Talk.*

**LSB** *Least Significant Byte.*

**LTE** *Long Term Evolution.*

**MEF** *Motor Engineering Foundation.*

**MSB** *Most Significant Byte.*

**NTC** *Negative Temperature Coefficient.*

**OBD** *On-Board Diagnostic.*

**OFDM** *Orthogonal Frequency Division Multiplexing.*

**OOK** *On-Off Keying.*

**PBCLK** *Peripheral Bus Clock.*

**PCB** *Printed Circuit Board.*

**PID** *Parameter Identification (in messageing protocol).*

**PSK** *Phase Shift Keying.*

**PTC** *Positive Temperature Coefficient.*

**PWM** *Pulse With Modulation.*

**QPSK** *Quadrature Phase Shift Keying.*

**RTD** *Resistance Temperature Detector.*



**RTR** *Remote Transmission Request.*

**SCK** *Serial Clock.*

**SCL** *Serial Clock Line.*

**SD** *Secure Digital.*

**SDA** *Serial Data Line.*

**SDI** *Slave Digital Input.*

**SDO** *Slave Digital Output.*

**SMBus** *System Management Bus.*

**SMT** *Surface Mount Technology.*

**SNR** *Signal-to-Noise Ratio.*

**SOF** *Start Of Frame.*

**SPI** *Serial Peripheral Interface.*

**SRR** *Substitute Remote Request.*

**SS** *Slave Select.*

**SYSCLK** *System Clock.*

**TDMA** *Time Division Multiple Access.*

**THT** *Through-Hole Technology.*

**UART** *Universal Asynchronous Receiver-Transmitter.*

**UMTS** *Universal Mobile Telecommunications Systems.*

**USB** *Universal Serial Bus.*

**VREG** *Virtual Registers.*

**WCDMA** *Wide-band Code Division Multiple Access.*



# Secção 1

## Motivação e objetivos

A Universidade de Aveiro é representada na competição *Motostudent* pela equipa *Motochanics* desde 2016. Com a experiência adquirida ao longo dos anos pela equipa, verificou-se que a falta de dados da prestação do prototipo desenvolvido é uma lacuna no desenvolvimento do mesmo. A equipa de baixo orçamento verificou que uma solução de mercado para a recolha de dados era demasiado cara, surgindo assim a ideia de desenvolver uma solução de telemetria adequada às suas necessidades.

Esta secção faz uma apresentação sucinta da competição internacional *Motostudent*, desde a organização até algumas das normas a ter em conta para a realização do trabalho relatado no presente documento. De seguida é feita uma breve apresentação da equipa que representa a Universidade de Aveiro, incluindo o histórico das suas participações em edições anteriores. É feita também, uma apresentação do problema bem como as vantagens da solução a desenvolver. Para finalizar esta secção, são apresentados alguns acontecimentos históricos relacionados com os avanços tecnológicos na área da telemetria ao longo dos anos.

### 1.1 A competição e os seus requisitos

Segundo a organização do evento, a competição *MotoStudent* é promovida pelas organizações *Motor Engineering Foundation* (MEF) e *TechnoPark MotorLand* [1]. A referida competição é uma prova entre equipas representativas de universidades de vários países do mundo. A competição é constituída por duas classes, “Petrol” (combustão interna) e “Electric” (propulsão 100% elétrica). Na figura 1.1 são apresentados os logótipos das classes integrantes na competição.



Figura 1.1: Logótipos das duas classes da competição *MotoStudent*[1].

Esta competição realiza-se bianualmente em Alcaniz (Teruel) Espanha, no circuito MotorLand Aragon que se estende por 5344 metros representado na figura 1.2. Este evento oferece aos estudantes envolvidos, a oportunidade de aplicarem conhecimentos adquiridos em contexto escolar, num projeto real aplicado à indústria, cujo objetivo é desenvolver

e produzir um protótipo de competição. As equipas durante um período de 18 meses, têm a possibilidade de mostrar as suas capacidades de criação e inovação aplicadas num projeto. Os alunos participantes desenvolvem capacidades de trabalho em:

- Trabalho de equipa,
- Aptidão e habilidade no trabalho,
- Execução de projetos com pequenos orçamentos,
- Planeamento de trabalho e cumprimento datas.

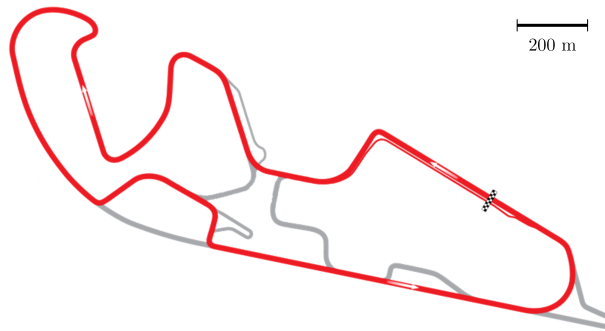


Figura 1.2: Circuito de Aragon para a competição MotoStudent [2].

Este evento torna-se uma mais valia, visto que o mesmo não se restringe unicamente à construção de um protótipo e participação numa corrida, mas sim na criação de um modelo de negócio, direcionado para a fabricação de motos. Posto isto, é essencial descrever todos os processos de desenvolvimento e fabricação, considerando sempre os custos envolvidos para implementação de um modelo a nível industrial.

A competição apresenta-se dividida em duas fases. Na primeira fase (MS1), os projetos são avaliados do ponto de vista industrial, avaliando aspetos tais como o custo de implementação, o design, a inovação, a industrialização, o projeto de engenharia e apresentações em público. A segunda fase (MS2) é constituída pelos testes estáticos, dinâmicos e pela a corrida final. A participação na corrida final só é autorizada quando a moto é aprovada nos teste estáticos e dinâmicos.

Para todas as edições, a organização da prova emite um regulamento que define as regras a serem cumpridas. Para a presente dissertação é necessário ter em consideração as normas aplicadas aos sistemas eletrónicos, impostas pela organização, para a edição de 2020. Deve ser tido em conta que:

- É proibido o uso de sistemas de controlo ou de assistência de condução, tais como controlo de tração, ABS, *anti-wheelie*<sup>1</sup>, etc,
- É permitida a apresentação de informações e alertas do sistema num painel de controlo para o piloto,
- É permitido o uso de sistemas de aquisição de dados relativos aos parâmetros do motor, dinâmica e comportamentos do piloto,
- É permitido o uso de todos os tipos de sensores, desde que estes não afetem nenhuma regra dos regulamentos técnicos.
- O *software* utilizado pode ser um produto comercial ou de auto criação,
- Todos os componentes e cablagem do sistema de aquisição de dados têm de estar devidamente fixos e colocados em áreas seguras,
- É proibido qualquer tipo de comunicação entre a moto e as boxes durante a corrida, sendo só possível obter valores com uma paragem nas boxes.

<sup>1</sup>Sistema que evita a perda de contacto da roda dianteira da moto com o solo.

## 1.2 A equipa *Motochanics*

A *Motochanics*, fundada em Março de 2015 é uma equipa sediada no Departamento de Engenharia Mecânica da Universidade de Aveiro. Este projeto nasceu do gosto pelo desporto motorizado num grupo de alunos que frequentavam o curso de engenharia mecânica. Na figura 1.3 é apresentado o logótipo da equipa.



Figura 1.3: Logótipo da equipa *Motochanics*.

A equipa já conta com duas participações no evento da classe “Petrol” (edições de 2016 e 2018). Na primeira edição ficou em 26º lugar de 35 equipas e na segunda obteve uma melhor classificação, alcançando o 16º lugar em 47 participantes. A equipa continua com a ambição de um dia se tornar uma referência universitária no motociclismo nacional e internacional. É importante referir que todo o trabalho descrito no presente documento vai ser testado no último modelo desenvolvido pela equipa, representado na figura 1.4.



Figura 1.4: Protótipo desenvolvido para a edição de 2018.

## 1.3 Apresentação do problema

Com a presente dissertação pretende-se desenvolver uma solução modular, que monitorize em tempo real e remotamente diversas grandezas físicas da mota durante a sua prestação em pista. A monitorização permite à equipa ajustar e otimizar o desempenho do protótipo em competição.

No início do projeto as informações recebidas por parte da equipa dividem-se em duas vertentes. A primeira indica as grandezas ou os valores que devem ser recolhidas da mota, e a segunda refere as especificações que o sistema deve ter. Na tabela 1.1 está apresentada a lista com os valores a serem lidos bem como a sua taxa de amostragem. As especificações técnicas requeridas pela equipa para a solução são:

- Alimentação do sistema 12 V de Corrente Contínua (CC),
- Guardar valores recolhidos em cartão *Secure Digital* (SD),
- Transmissão de dados para o ecrã da mota já desenvolvido, com uma taxa de atualização de 10 Hz,
- Transmissão de dados em tempo real perante escolha do utilizador,

Tabela 1.1: Sensores e taxas de leitura requeridas pela equipa.

Grandeza / variável	Unidade de variável	Frequência de leitura (Hz)
Seletor de mudança	unit.	1
Posição do acelerador	%	10
Pressão do travão dianteiro	%	20
Pressão do travão traseiro	%	20
Rotações do motor	RPM	10
Velocidade atual	km/h	10
Suspensão frontal	mm	20
Suspensão traseira	mm	20
Temperatura pneu dianteiro	°C	20
Temperatura pneu traseiro	°C	20
Aceleração em linha	<i>g</i>	10
Aceleração lateral	<i>g</i>	10
Aceleração vertical	<i>g</i>	10
<i>Row</i>	graus	10
<i>Pitch</i>	graus	10
<i>Yaw</i>	graus	10
Tempo de avanço da ignição para o cilindro	%	10
Pressão absoluta do coletor de admissão	inHg	10
Temperatura do ar de admissão	°C	2
Temperatura da refrigeração do motor	°C	1
Coordenadas GPS	°	10

- Compatibilidade com o motor RC250R da marca *KTM*,
- Solução com proteção contra poeiras e gotas de água,
- Sistema modular com possibilidade de adicionar novos sensores,
- Dimensões e peso reduzidos.

Tendo em conta os requisitos requeridos pela equipa, no final do projeto é pretendido obter uma solução composta por dois sistemas, sendo eles um transmissor e um recetor de dados, como mostra a figura 1.5. O sistema de transmissão será mais complexo, para além de efetuar o envio dos dados do veículo em movimento para as *boxes*, também terá de obter diversas grandezas físicas, a partir de diversos módulos a desenvolver. É de referir que para a presente dissertação é pretendido o desenvolvimento de hardware e software para aquisição, transmissão e receção de dados.

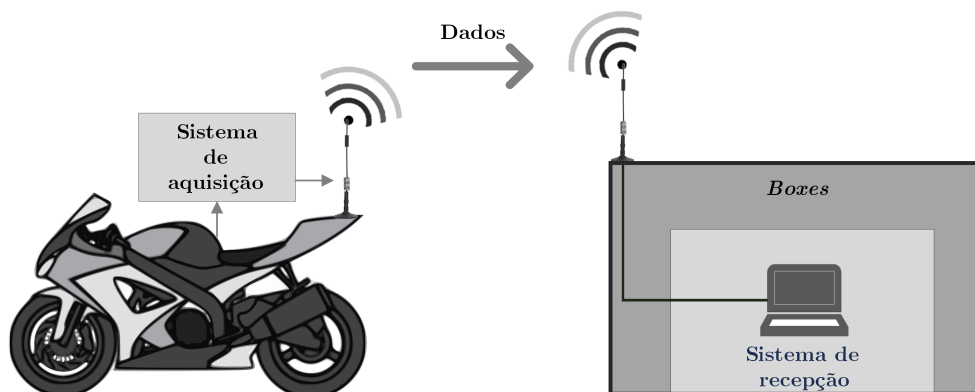


Figura 1.5: Diagrama geral da solução a desenvolver.

## 1.4 História da telemetria

A palavra telemetria, com origem grega *téle* (à distância) e *metron* (medida), faz referência a algo que efetua medições à distância de grandezas que não estão acessíveis. Este conceito começa por ser utilizado no auge da revolução industrial. Inicialmente a telemetria era efetuada através de comunicação por fios. Em 1845 é aplicado um dos primeiros circuitos de transmissão de dados desenvolvido na Rússia, efetuando uma ligação entre o *Winter Palace* e a sede do exército na cidade de São Petersburgo. No decorrer do tempo foram desenvolvidas aplicações de monitorização para diferentes sistemas, desde meteorologia (França no ano 1874), atividade sísmica (Rússia no ano 1906), cargas elétricas em redes (EUA no ano 1912) e bloqueios e níveis de água (Canal do Panamá, entre 1913 e 1914)[3].

Em meados do século XX, surge o sistema *Molchanov*. Este é o primeiro com comunicação sem fios desenvolvido em conjunto pela França e pela Rússia. Este sistema efetuava medições de temperatura e pressão, convertendo-as em código *Morse* transmitido sem fios[4]. Este conceito foi aplicado nas mais diversas áreas desde da agricultura até à medicina.

### 1.4.1 Telemetria em competições motorizadas

Atualmente, a telemetria desempenha um papel de grande importância na competição motorizada, sendo que os primeiros sistemas deste tipo de telemetria foram implementados na *Fórmula 1* nos anos 80. Ao longo dos anos têm vindo a ser monitorizados um maior número de sensores. O uso desta tecnologia ajuda a melhorar o desempenho das equipas em competição, pois possibilita monitorizar todo o veículo e manter uma comunicação com o piloto durante a prova para melhorar o rendimento, tanto do veículo como do próprio piloto. Na competição *MotoGP* a telemetria foi implementada mais tarde, trazendo para a competição as vantagens referidas anteriormente. Atualmente, cada vez mais, a telemetria desempenha um papel de grande importância nas diversas competições motorizadas, verificando-se um crescimento ao longo dos anos.

## 1.5 Estrutura do documento

Após a apresentação do problema, o documento está organizado em vários capítulos da seguinte forma:

- Secção 2: Apresentação de requisitos do projeto e o estado da arte,
- Secção 3: Estudo da arquitetura da solução e discussão da mesma,
- Secção 4: Documentação de todo o desenvolvimento de *hardware*,
- Secção 5: Documentação de *software* desenvolvido e das máquinas de estados da solução,
- Secção 6: Apresentação de testes realizados e respetivos resultados,
- Secção 7: Conclusões sobre o trabalho desenvolvido bem como possíveis melhorias do mesmo.





## Secção 2

# Requisitos e estado da arte

Esta secção está dividida em cinco partes. Inicialmente é feito um estudo que verifica quais são os valores que podem ser obtidos diretamente do veículo, assim como, quais as diferentes formas de obter valores desconhecidos. De seguida é apresentado uma estimativa da quantidade de dados a serem transmitidos pela a mota.

Tendo em conta os valores que têm de ser lidos através de nova instrumentação, é feito um estudo de diferentes tipos de sensores para aquisição de temperatura, de pressão ou de deslocamento.

Como a comunicação entre dispositivos/interfaces é fundamental para a realização deste projeto, são apresentados diferentes protocolos de comunicação com e sem fios. Para as comunicações com fio são abordados os protocolos *Universal Asynchronous Receiver-Transmitter* (UART), *Inter-Integrated Circuit* (I<sup>2</sup>C), *Serial Peripheral Interface* (SPI) e *Controller Area Network* (CAN), sendo o último largamente utilizado pela indústria automóvel. Para a comunicação sem fios de longo alcance é feita uma apresentação e comparação de dois métodos para transmissão de dados, sendo que um requer o uso de serviços prestados por uma operadora e o outro não.

No decorrer desta secção são apresentados dois produtos disponíveis no mercado, que contêm algumas das características requeridas para a solução, assim como o seu preço.

Para concluir são apresentados os requisitos e as especificações a ter em conta no presente documento.

### 2.1 Levantamento de sensores e taxa de transmissão de dados

No início do projeto a equipa disponibilizou as informações anteriormente referidas na secção 1.3, sobre as especificações a ter em conta no sistema. O motor escolhido, pela organização do evento, para a competição de 2020 é um *KTM* do modelo RC250R. Na tabela 1.1, é apresentada a lista de variáveis requerida pela equipa bem como a sua taxa de atualização. Tendo em conta que a motorização da mota é igual à utilizada na edição anterior da competição, foi obtida a informação de quais os valores que estão disponíveis na mota, através de um estudo da mesma.

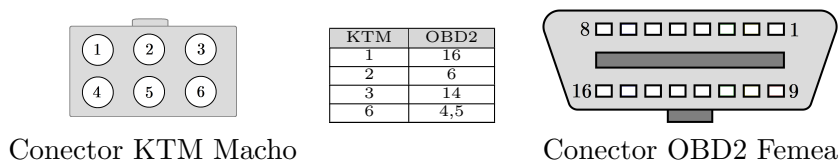
Inicialmente com o esquema elétrico da mota fornecido pela equipa, é verificada a existência de vários sensores requeridos para a solução. Do esquema também faz parte uma ficha de diagnóstico. Com recurso à referida ficha e a um adaptador *On-Board Diagnostic* (OBD) apresentados na figura 2.1, foi feito um diagnóstico dos valores armazenados na *Engine Control Unit* (ECU) da mota. A ligação entre a ficha diagnóstico



Ficha diagnostico KTM      Adaptador OBD2 [5]

Figura 2.1: Ficha diagnóstico e adaptador OBD2.

e o adaptador OBD está representada na figura 2.2. Com recurso ao *software TouchScan OBD* [6], foi feita uma sessão de gravação dos dados emitidos pela mota em funcionamento.



Conector KTM Macho

Conector OBD2 Femea

Figura 2.2: Conector para diagnóstico presente na mota.

Com os valores obtidos pela sessão anteriormente referida, é possível verificar quais os valores que podem ser lidos diretamente da ECU e quais requerem instalação de novos sensores. Na tabela 2.1 são apresentadas as conclusões retiradas do diagnóstico realizado à mota. Com esta tabela é possível resumir que grandezas terão de ser instrumentadas, sendo elas:

- Pressão hidráulica,
- Velocidade atual,
- Deslocamento de uma suspensão,
- Temperatura de um pneu,
- Aceleração,
- Inclinação,
- Localização.

De modo a ser possível fazer uma estimativa, dos dados a transmitir, foi atribuída uma gama de valores, frequência e o número de bits para cada variável, para obter um valor estimado da taxa de dados a enviar por segundo. De modo a calcular uma taxa de transmissão que garanta o envio de todas as variáveis é utilizado um coeficiente de segurança de 2. Multiplicado o valor estimado pelo coeficiente de segurança, é obtida a taxa de envio de 5.35 Kbits/s. Os valores atribuídos a cada variável e o estudo da quantidade de dados a serem enviados, estão apresentados na tabela 2.2. É preciso ter em conta que a quantidade de dados a ser enviada é um valor estimado e não real. Com os valores adquiridos e estimados é possível conhecer algumas das características que o sistema irá contemplar.

## 2.2 Medição de grandezas físicas

Para uma monitorização mais completa da mota, é necessário efetuar medições de diferentes grandezas físicas. Deste modo, esta secção apresenta diferentes soluções para medição de temperatura, pressão e compressão.

Tabela 2.1: Valores a serem lidos da ECU ou por novos sensores.

Grandeza / variável	Leitura ECU	Sensor adicional
Selector de mudança	X	
Posição do acelerador	X	
Pressão do travão dianteiro		X
Pressão do travão traseiro		X
Rotações do motor	X	
Velocidade atual	X	
Suspensão frontal		X
Suspensão traseira		X
Temperatura pneu dianteiro		X
Temperatura pneu traseiro		X
Aceração em linha		X
Aceração lateral		X
Aceração vertical		X
<i>Row</i>		X
<i>Pitch</i>		X
<i>Yaw</i>		X
Tempo de avanço de ignição para o cilindro	X	
Pressão absoluta do coletor de admissão	X	
Temperatura do ar de admissão	X	
Temperatura da refrigeração motor	X	
Coordenadas GPS		X

Tabela 2.2: Estudo das grandezas/sensores a obter e a sua taxa de dados.

Variável / Sensor	Unid.	Gama de valores	Freq. de envio (Hz)	N° Bytes	Taxa (Kbit/s)	
Seletor de mudança	unit.	[0,6]	1	1	0,008	
Posição de acelerador	%	[0,100]	10	1	0,080	
Posição do travão frontal	%	[0,100]	20	1	0,160	
Posição do travão traseiro	%	[0,100]	20	1	0,160	
Rotações do motor	RPM	[0,15000]	10	2	0,160	
Velocidade atual	km/h	[0,255]	10	1	0,080	
Suspensão frontal	mm	[-100,100]	20	1	0,160	
Suspensão traseira	mm	[-100,100]	20	1	0,160	
Acelerómetro em linha	g	[0,00,1,00]	10	1	0,080	
Acelerómetro lateral	g	[0,00,1,00]	10	1	0,080	
Acelerómetro vertical	g	[0,00,1,00]	10	1	0,080	
<i>Roll</i>	°/s	[0,1,90,0]	10	1	0,080	
<i>Pitch</i>	°/s	[0,1,90,0]	10	1	0,080	
<i>Yaw</i>	°/s	[0,1,90,0]	10	1	0,080	
Temperatura pneu frontal	°C	[0,150]	20	1	0,160	
Temperatura pneu traseiro	°C	[0,150]	20	1	0,160	
Avanço de ignição para cilindro 1	%	[0,255]	10	1	0,080	
Pressão de ar absoluta de admissão	kPa	[0,70]	10	1	0,080	
Temperatura do ar de admissão	°C	[0,255]	2	1	0,016	
Temperatura da refrigeração do motor	°C	[0,255]	1	1	0,008	
Coordenadas GPS	Latitude	°	[-90,90]	10	4	0,320
	Longitude	°	[-180,180]	10	5	0,400
<b>Total</b>					<b>2,672</b>	

### 2.2.1 Temperatura

Como indicado anteriormente na secção 2.1 será necessário obter temperaturas de diferentes objetos/ambientes. Deste modo foi efetuado um estudo das principais for-

mas de medição de temperatura com/sem contacto. De seguida, irá ser abordado o funcionamento dos sensores por resistência, termístor e pirómetro.

### Sensor de resistência

Os sensores de resistência são conhecidos por *Resistance Temperature Detector* (RTD), estes são baseados na dependência de um material condutor com a variação de temperatura. A sua resistividade está diretamente ligada ao tipo de material utilizado no sensor, sendo que o tipo de material é escolhido conforme a sua aplicação final. Os materiais mais utilizados na construção de RTDs são a platina (Pt) e o níquel (Ni), sobretudo pela linearidade que apresentam, sendo estes bastante aplicados em sensores de alto desempenho.

### Termístor

Um termístor é baseado num semiconductor. Este tem a capacidade de variar o valor resistivo consoante a temperatura a que está exposto, tal como os RTDs referidos anteriormente. Existem dois tipos de termístores, sendo eles, *Negative Temperature Coefficient* (NTC) e *Positive Temperature Coefficient* (PTC). O comportamento dos termístores é apresentado na figura 2.3, que mostra a relação entre o valor resistivo e a temperatura. Este tipo de sensor apresenta uma maior sensibilidade em certas zonas de temperatura. O NTC apresenta maior sensibilidade na zona das temperaturas baixas, enquanto o PTC mostra esse comportamento na zona das temperaturas altas.

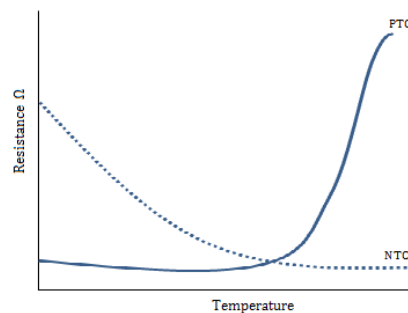


Figura 2.3: Comparação entre NTC e PTC [7].

### Pirómetro

Um pirómetro, conforme está apresentado um exemplo na figura 2.4, é um sensor capaz de medir temperaturas através da radiação térmica emitida por um objeto ou corpo. Quanto maior é a quantidade de radiação emitida por um objeto, mais elevada é a temperatura que este apresenta. Este tipo de sensores foram desenvolvidos para obter temperaturas de corpos com difícil acesso ou com existência de movimento. Comparando com outros métodos referenciados anteriormente, a sua principal vantagem é o facto de não requerer contacto físico com o objeto a medir.

#### 2.2.2 Pressão

Para efetuar medições de pressão ou força são utilizados sensores piezoelétricos. Este tipo de sensores são caracterizados por possuírem a capacidade, de produzir uma carga elétrica quando sujeitos a diferentes forças. A carga produzida por um material piezoelétrico está diretamente relacionada com a orientação e intensidade da força aplicada



Figura 2.4: Pirómetro [8].

no mesmo. Na figura 2.5 é possível verificar como as cargas estão dispostas sobre tração (a) e compressão (b). Na utilização deste tipo de sensores é obtido um valor em tensão

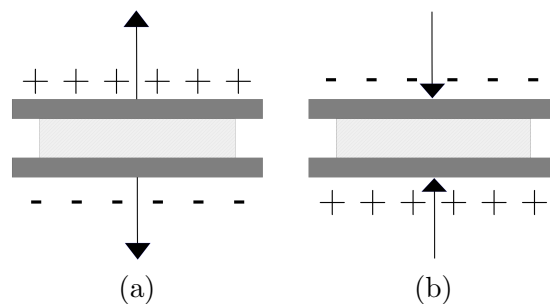


Figura 2.5: Efeito piezoelétrico [9].

proporcional à carga aplicada sobre a capacidade do sensor. O valor da capacidade do sensor esta relacionado com a construção e as características do material piezoelétrico escolhido para o mesmo.

### 2.2.3 Deslocamento linear

Para efetuar a medição de um deslocamento linear do curso de um cilindro, é medida a distância entre os dois pontos de apoio do objeto a medir. São enumeradas duas opções de conseguir medir esta grandeza. A primeira opção é através de sensores de ultrassons. No entanto, esta solução não apresenta uma boa resolução e também não permite medir distâncias curtas (distância a medir  $> 15$  cm). A segunda opção é efetuar a medição com recurso a um potenciômetro linear, como o representado na figura 2.6, em que a resistência do sensor varia linearmente com a posição em que o êmbolo do potenciômetro se encontra.



Figura 2.6: Potenciômetro linear com o formato cilíndrico [10].

## 2.3 Comunicações entre dispositivos

Como referido na secção 1.3 a utilização de sensores, de dispositivos de memória e a transmissão de dados ponto a ponto, são fundamentais para a resolução do problema. Deste modo foi feito um estudo de alguns protocolos de comunicação, sendo este dividido em dois grupos, com e sem fios.

Para a comunicação com fios em sistemas embutidos existem vários protocolos que diferem tanto a nível de *software* como também de *hardware*, deste modo serão abordados os mais utilizados. Para a comunicação sem fios, são abordadas diferentes formas de comunicação de longo alcance como por exemplo para transmissão de dados ponto a ponto.

### 2.3.1 Protocolos de comunicação com fios

Nesta secção serão abordados quatro protocolos de comunicação com fios maioritariamente utilizados em sistemas embutidos. Todos os protocolos abordados efetuam comunicações série, entre outras, podem variar as seguintes características:

- Número de dispositivos que se podem interligar no mesmo barramento de um determinado protocolo,
- Assíncronos ou síncronos,
- Implementação a nível de *hardware*,
- Implementação a nível de *software*,
- Velocidades de transmissão.

#### 2.3.1.1 *Universal Asynchronous Receiver Transmitter (UART)*

O protocolo UART é um tipo de comunicação série bidirecional que não requer sincronização entre dispositivos. O módulo de comunicação UART é constituído por três sub-módulos sendo eles:

- Gerador de *baudrate* - tem como função gerar um sinal de relógio local para controlar o os restantes sub-módulos,
- Recetor - converte a informação recebida no pino Rx em dados,
- Transmissor - converte os dados que pretendem ser enviados, de acordo com a configuração do módulo UART, e realiza a transmissão da mensagem através do pino Tx.

Na figura 2.7 em (a) é mostrado um diagrama geral de um módulo UART, como também é apresentado em (b) um exemplo da interligação de dois dispositivos a comunicarem através deste protocolo[11].

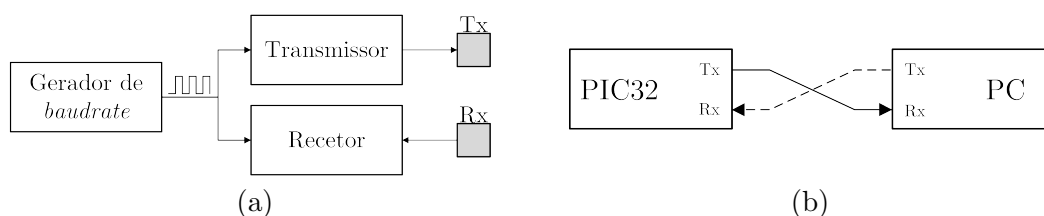


Figura 2.7: Exemplo de ligação para comunicação UART.

Neste protocolo a sincronização entre dispositivos é realizada a nível de *software*, com recurso do uso de bits de início e fim de mensagem. Através do valor de *baudrate* os

dispositivos interpretam cada bit recebido através do tempo associado a cada um, ou seja cada bit é representado pelo o período de  $(1/Baudrate)$  segundos. A estrutura de uma mensagem enviada por UART é composta por bits de início, representação dos dados a enviar em (5,6,7 ou 8)bits, um bit de paridade para verificação dos erros de transmissão e por fim os bits que sinalizam o fim da mensagem. Na figura 2.8 é apresentada a estrutura de uma mensagem enviada através do protocolo UART baseada em [12].

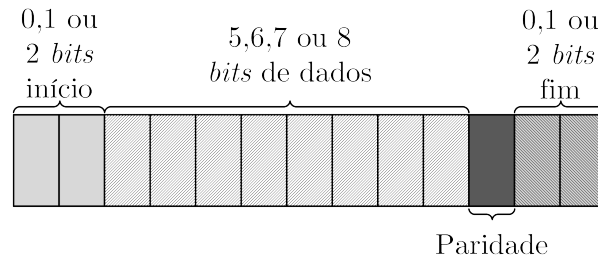


Figura 2.8: Estrutura da mensagem do protocolo UART.

### 2.3.1.2 *Inter-Integrated Circuit (I<sup>2</sup>C)*

O protocolo I<sup>2</sup>C foi introduzido pela *Philips* no início da década de 80, como um barramento normalizado para comunicação entre circuitos integrados em produtos eletrónicos de consumo em massa. Após a sua introdução, este protocolo foi utilizado em milhões de dispositivos tais como televisões, telefones, rádios para carros entre outros produtos eletrónicos [13].

Um barramento I<sup>2</sup>C é constituído por mestres e escravos, sendo que este protocolo permite a presença de um ou mais mestres. A nível de *hardware* este barramento é formado por duas linhas. A linha de relógio é denominada por *Serial Clock Line (SCL)* e a linha para transferência de dados é denominada por *Serial Data Line (SDA)*. O sinal de relógio é gerado pelo mestre sendo esta linha unidirecional, enquanto a linha de dados é bidirecional ou seja é utilizada para o mestre enviar dados ao escravo e vice-versa. Quando é requerida uma comunicação bidirecional é obrigatório o uso de resistências *pull-up*. Na figura 2.9 é apresentado um exemplo de um barramento I<sup>2</sup>C com um só mestre e  $n$  escravos em que os dados que circulam na linha SDA são bidirecionais [12].

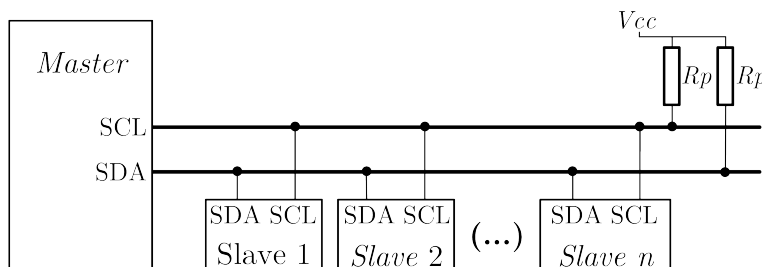


Figura 2.9: Exemplo de um barramento I<sup>2</sup>C [14].

Neste protocolo o endereçamento é feito ao nível de *software*. No início da transmissão de dados é enviado um *byte* formado pelo endereço do escravo de 7 bits e com o bit menos significativo a indicar se o mestre pretende escrever ou ler um registo. Todos os *bytes* que circulam na linha de dados, são reconhecidos pelo o recetor dos mesmos dados. Isto

é, por exemplo caso o mestre escreva um byte no escravo, por cada byte enviado pelo o mestre, o escravo emite um bit de reconhecimento. Mas caso o mestre deseje efetuar uma leitura de um valor, este tem de fazer o reconhecimento dos bytes recebidos quando o escravo envia a informação.

Na figura 2.10 é mostrado um exemplo da escrita de um byte no escravo, enquanto na figura 2.11 é apresentado um exemplo do mestre a fazer um pedido de leitura do valor guardado num endereço de memória do escravo.

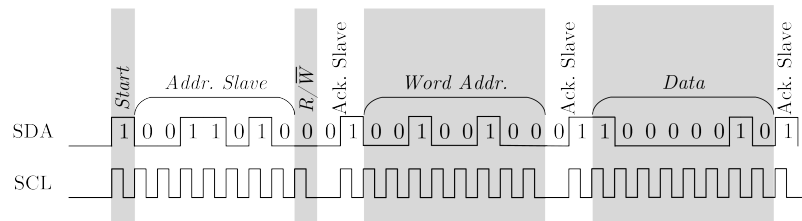


Figura 2.10: Exemplo de escrita através de I<sup>2</sup>C [15].

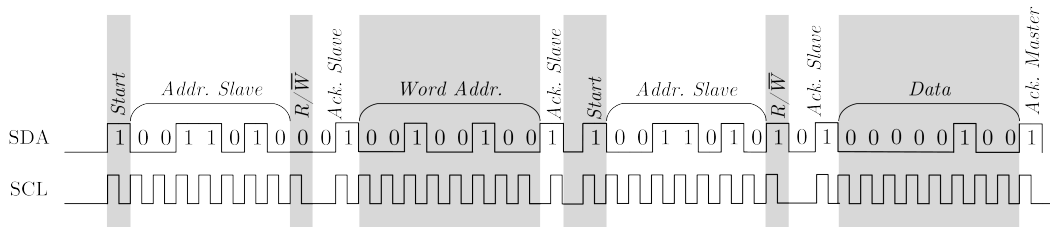


Figura 2.11: Exemplo de leitura através de I<sup>2</sup>C [15].

### 2.3.1.3 Serial Peripheral Interface (SPI)

O protocolo SPI é um dos mais utilizados para transferência de dados entre dispositivos e um micro-controlador. Os dispositivos podem ser *Electrically Erasable Programmable Read Only Memory* (EEPROM), *Digital to Analog Converter* (DAC), *Analog to Digital Converter* (ADC), sensores ou até um microcontrolador. Este protocolo é adequado para a comunicação entre circuitos integrados para baixas ou médias velocidades de transferência de dados entre periféricos.

Neste protocolo a transferência de dados ocorre entre um mestre (*Master*) e um ou mais escravos (*Slaves*). O mestre tem a função de fornecer um sinal de relógio *Serial Clock* (SCK) de modo a garantir o sincronismo durante a transmissão de dados. A transferência de dados ocorre unicamente quando é feita uma manipulação do sinal de relógio. O mestre através da linha *Slave Digital Output* (SDO) recebe os dados emitidos por um escravo, enquanto a utilização da linha *Slave Digital Input* (SDI) transmite os dados para o escravo. De modo a ser possível ter vários escravos num barramento SPI a comunicar um de cada vez, este protocolo utiliza um sinal digital controlado pelo o mestre, associado a cada escravo *Slave Select* (SS). A dimensão de um barramento SPI está limitada à quantidade de sinais digitais que o mestre consegue controlar. Quando é realizada uma transmissão de dados, o mestre aciona SS associado a um escravo, sendo que os restantes escravos ignoram os dados que são transmitidos. Na figura 2.12 está representado um diagrama que representa a implementação de um barramento de dados SPI com um mestre e  $n$  escravos.



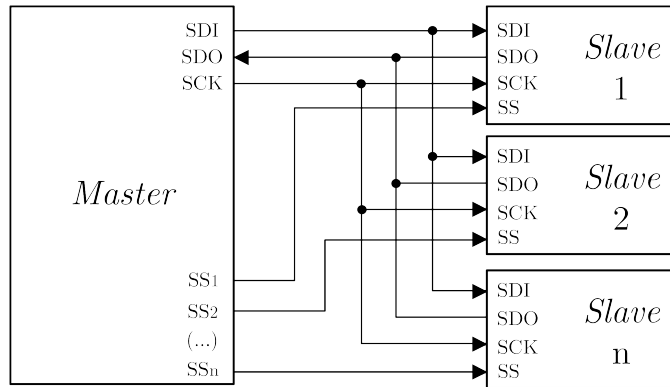


Figura 2.12: Barramento SPI padrão [16].

Se o mestre pretende enviar uma mensagem, este começa por ativar o SS do escravo com que pretende comunicar. De seguida são colocados os dados na linha SDI começando pelo o bit mais significativo. O escravo simultaneamente responde ao pedido, colocando os dados requeridos na linha SDO. Na figura 2.13 é mostrado um exemplo de leitura de dados a um escravo num barramento SPI.

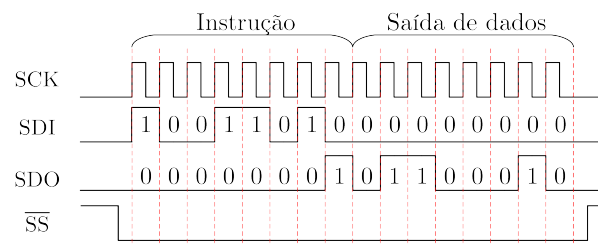


Figura 2.13: Exemplo de troca de dados num barramento SPI [17].

### 2.3.1.4 Controller Area Network (CAN)

O protocolo CAN foi desenvolvido pela BOSCH [18], sendo disponibilizado em meados dos anos 80. Atualmente é aplicado na indústria, nos navios, em veículos automóveis, entre outros. Na Figura 2.14 está representado um caso padrão de um barramento CAN.

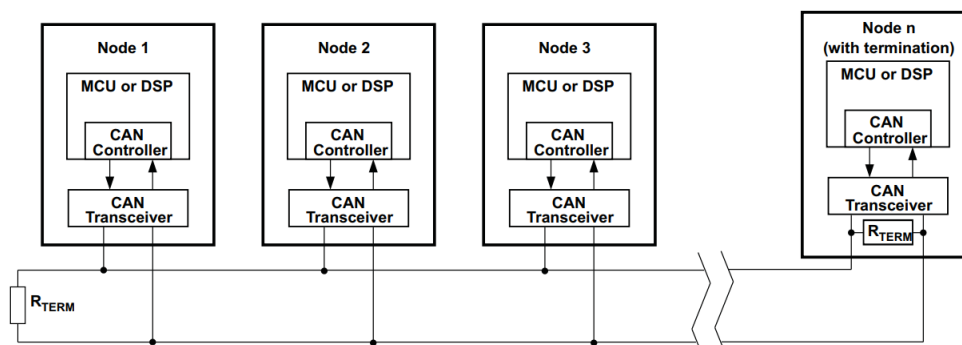


Figura 2.14: Barramento CAN padrão [19].

O CAN é um protocolo de comunicação série síncrono de elevada fiabilidade. O

barramento é constituído por dois ou mais nós, em que todos podem enviar ou receber mensagens. O identificador atribuído a cada mensagem define a prioridade da mesma. As mensagens com menor prioridade ficam em espera até que não exista nenhuma mensagem de prioridade superior. A utilização deste sistema tem como principais vantagens:

- Baixo custo de planeamento e instalação,
- Diagnóstico de erros,
- Redução da cablagem total da rede,
- Funcionamento em tempo real.

A norma ISO-11898:2003, estabelece o formato de mensagens *standard* para um barramento CAN, como representado na Tabela 2.3. Este formato permite a utilização de 2048 identificadores. Posteriormente foi alterada de modo a ser possível utilizar um maior número de identificadores (aproximadamente 537 milhões), surgindo assim o formato alargado, apresentado na Tabela 2.4.

Tabela 2.3: Constituição de uma mensagem *standard*.

S O F	11-bit Identifier	R T R	I D E	r0	DLC	0...8 Bytes Data	CRC	ACK	E O F	I F S
-------------	----------------------	-------------	-------------	----	-----	------------------	-----	-----	-------------	-------------

Tabela 2.4: Constituição de uma mensagem extensa.

S O F	11-bit Identifier	S R R	I D E	18-bit Identifier	R T R	r1	r0	DLC	0...8 Bytes Data	CRC	ACK	E O F	I F S
-------------	----------------------	-------------	-------------	----------------------	-------------	----	----	-----	------------------	-----	-----	-------------	-------------

Os parâmetros definidos nos dois tipos de mensagens possuem as seguintes funcionalidades:

- *Start Of Frame* (SOF) - Um só bit que marca o início de mensagem, sendo usado para sincronismo dos nós,
- Identificador - Estabelece a prioridade da mensagem. O que tiver menor valor neste campo tem maior prioridade,
- *Remote Transmission Request* (RTR) - Pedido de transmissão remota. Quando é feito um pedido de informação por outro nó, este tem prioridade sobre outras que são transmitidas periodicamente,
- *Substitute Remote Request* (SRR) - É um substituto do parâmetro (RTR), para o formato extenso,
- *Identifier Extension Bit* (IDE) - É dominante para formato de 11 bits. Mensagem com identificador de 11 bits ganha arbitragem a mensagem de 29 bits,
- r0 e r1 - Bits de reserva,
- *Data Length Code* (DLC) - Composto por 4 bits, que indica o número de *bytes* a serem transmitidos,

- *Data* - Dados a serem enviados,
- *Cyclic Redundancy Check (CRC)* - Verificação de redundância cíclica, que contém o número de bits transmitidos para ajuda de detecção de erros na transmissão,
- *Acknowledge (ACK)* - Verificação feita por todos os nós, ou seja, se algum não receber corretamente a mensagem, é feito um novo envio da mesma. Este campo é composto por 2 bits, sendo o primeiro usado para reconhecimento e o segundo como delimitador,
- *End Of Frame (EOF)* - Bit que indica o fim da mensagem,
- *Inter Frame Space (IFS)* - Contêm o tempo requerido pelo o controlador, ao mover corretamente a mensagem recebida para a área de *buffer*.

O tempo para a transmissão de cada bit está dividido em quatro segmentos de tempo, sendo eles o de sincronização, o de propagação, o da fase 1 e o da fase 2, como representado na figura 2.15. Entre a fase 1 e a fase 2 é realizada a recolha de uma ou de três amostras do *bit*, que determinam o estado do mesmo, sendo configurável o número de amostras recolhidas. A configuração da velocidade do módulo CAN é efetuada através de uma função disponível na biblioteca, que calcula o valor do *prescaler* recorrendo aos valores dos registos anteriormente referidos, ao valor da frequência do sistema (SYSCLK) e ao valor do *baudrate* do barramento CAN.

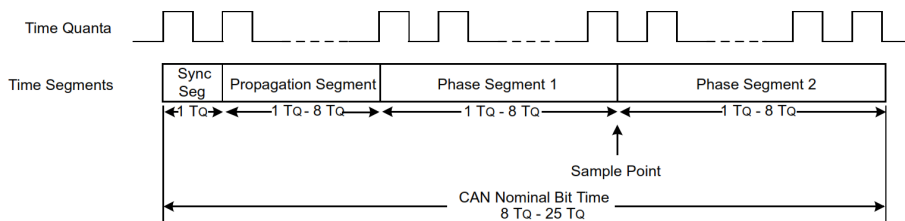


Figura 2.15: Constituição do tempo para transmissão de um *bit*[20].

### 2.3.2 Comunicação sem fios de longo alcance

A tecnologia de radiofrequência ou sem fios como é geralmente conhecida, é baseada em fenómenos de ondas eletromagnéticas trabalhando no intervalo do espectro entre os 3 Hz e 300 GHz. Hoje em dia o uso desta tecnologia torna-se indispensável, sendo aplicada em diversas áreas do quotidiano da humanidade. As pesquisas acerca deste tipo de tecnologia foram iniciadas teoricamente por James Maxwell em 1864 e posteriormente verificadas por Heinrich Hertz em 1887. Guglielmo Marconi em 1895 demonstrou que o uso desta tecnologia era bastante eficaz como meio de comunicação. Com o desenvolvimento da válvula termiônica nos finais do século XIX, a tecnologia de rádio tornou-se num meio de comunicação em massa [21].

De seguida são abordados dois métodos de transmissão, que diferem na necessidade de recorrer a um serviço prestado por uma operadora móvel.

#### 2.3.2.1 Serviços móveis associados a uma operadora

No decorrer do século XX, com o avanço do desenvolvimento dos componentes semicondutores e dos circuitos integrados, os dispositivos de comunicação ficaram muito mais

compactos e portáteis resultando na revolução das comunicações móveis [21]. Com o aumento do número comunicações e da distância entre dispositivos, diferentes entidades criaram infraestruturas para suportar uma maior área de cobertura e um maior número de utilizadores surgindo assim as primeiras operadoras móveis. De seguida são descritas três formas de comunicação sem fios através do recurso ao uso de infraestruturas .

### ***Groupe Spécial Mobile (GSM)***

Na Conferência Europeia de Administração de Correios e Telecomunicações em 1982, foi definida uma nova forma de comunicações móveis na Europa, devido à incompatibilidade entre diferentes sistemas analógicos, denominada como GSM [22].

A comunicação GSM utiliza para transmitir e recuperar informações duas técnicas de modulação em simultâneo, denominadas *Time Division Multiple Access* (TDMA) e *Frequency Division Multiple Access* (FDMA). Estas técnicas utilizam pacotes de dados em instantes e intervalos de frequência específicos. Deste modo, é possível que ocorram várias conversas na mesma frequência, usando diferentes intervalos de tempo. As frequências utilizadas para a transmissão e recepção são diferentes. As características desta tecnologia estão representadas na Tabela 2.5.

Tabela 2.5: Especificações do sistema GSM.[23]

Especificação	GSM
Método de acesso	TDMA/FDMA
Frequência (MHz)	
Equipamento móvel para base	890 até 915
Base para equipamento móvel	935 até 960
Largura de banda por canal (kHz)	200
Número de canais	124
Utilizadores por canal	8
Tipo de modulação	GMSK
Taxa de transmissão	Até 34 (Kbps)

A GSM utiliza uma largura de banda total de 25 MHz dividida em 125 ondas portadoras com uma largura de 200 kHz cada. A primeira portadora não é utilizada, de modo a evitar interferência com outros sistemas, resultando em 124 portadoras úteis. Cada canal tem uma taxa de transferência de dados de aproximadamente 271 Kbps com possibilidade de alocar 8 utilizadores. Ou seja, cada utilizador tem taxas de transferência máximas de sensivelmente 34 Kbps.

### ***Universal Mobile Telecommunications Systems (UMTS)***

Com o avanço exponencial da tecnologia, e com a necessidade da transmissão de uma maior quantidade de dados, foi redesenhado um novo sistema denominado *Universal Mobile Telecommunications Systems* (UMTS). A implementação desta tecnologia utiliza alguns componentes constituintes no sistema GSM, com algumas modificações, contudo a estrutura do funcionamento do UMTS foi completamente revista e refeita relativamente ao sistema GSM. Na Tabela 2.6 estão indicadas as principais características da comunicação UMTS.

Com a comunicação UMTS é utilizado um novo método de acesso denominado *Wide-band Code Division Multiple Access* (WCDMA). Os utilizadores não estão separados por tempo ou frequência multiplexada como no GSM, mas sim por um código atribuído a

Tabela 2.6: Especificações do sistema UMTS.[24]

Especificação	UMTS
Método de acesso	WCDMA
Frequência (MHz)	
Equipamento móvel para base	1920 até 1980
Base para equipamento móvel	2110 até 2170
Largura de banda por canal (MHz)	5
Número de canais	12
Tipo de modulação	QPSK
Taxa de transmissão	Até 384 (Kbps)

cada um. Deste modo é possível atingir a velocidade de recepção de dados até aos 384 Kbps e de envio de dados limitada entre os 64 e os 128 kbps. Na Europa este sistema trabalha com a recepção de dados na frequência entre os 2110 e 2170 MHz, dividida por 12 canais, cada um com uma largura de banda de 5MHz. Já para o envio de dados são utilizados os mesmos 12 canais com a mesma largura de banda, em que a diferença observada é a frequência do seu funcionamento entre os 1920 e 1980 MHz. O tipo de modulação adoptado deixou de ser o *Gaussian Minimum Shift Keying* (GMSK) presente no GSM para ser o *Quadrature Phase Shift Keying* (QPSK).

### ***Long Term Evolution (LTE)***

Um conjunto de organizações denominado por *Third Generation Partnership Project* (3GPP) redesenham uma nova rede, resultando por fim numa nova tecnologia, à qual foi atribuída o nome de *Long Term Evolution* (LTE). As principais características desta tecnologia estão apresentadas na Tabela 2.7.

Tabela 2.7: Especificações do sistema LTE.[24]

Especificação	LTE
Método de acesso	FDD/TDD
Largura de banda (MHz)	1.25 até 20
Largura de banda por canal (kHz)	180
Tipo de modulação	OFDM
Taxa de transmissão	Até 300 Mbps

Uma das grandes diferenças verificada nesta nova tecnologia, é a forma como são enviados os sinais na largura de banda disponível. Isto é, em vez de o sinal ser dividido sobre a onda portadora, por exemplo 5 MHz, a LTE transmite os dados através de muitas ondas portadoras com uma largura de banda de 180 KHz. É assim substituída uma única transmissão rápida, por um fluxo de dados que é dividido em muitos fluxos de dados mais lentos que são transmitidos simultaneamente. Este tipo de modulação é denominado por *Orthogonal Frequency Division Multiplexing* (OFDM).

Para conseguir enviar um número maior de portadoras é alargado o canal de transmissão, resultando assim no aumento da velocidade de transmissão de dados. A tecnologia LTE utiliza várias larguras de banda, que estão definidas entre 1.25MHz e 20MHz.

Em suma, esta tecnologia trabalha em várias bandas de frequência, variando de país para país.

### 2.3.2.2 Utilização de uma banda ISM

Para que o sistema de telemetria, não esteja dependente de um serviço prestado por uma operadora, é possível utilizar faixas de frequência sem licenciamento, denominadas por bandas *Industrial, Scientific and Medical* (ISM). Estas faixas de frequência são reservadas para fins industriais, uso científico ou médico. Na Tabela 2.8, de acordo com a legislação disponibilizada pela Autoridade Nacional de Comunicações (ANACOM), na revisão de Abril de 2018, são apresentadas diferentes limitações nas faixas de frequência para uso livre.

Tabela 2.8: Bandas ISM Sub-GHz de uso geral na Europa [25].

Faixa de frequências (MHz)	Potência (dBm / mW)	Espaçamento de canais (kHz)	Duty-Cycle
433.05 – 434.79	+10 / 10	Sem limite	< 10%
433.05 – 434.79	0 / 1	Sem limite	Sem limite
433.05 – 434.79	+10 / 10	< 25	Sem limite
868 – 868.6	+14 / 25	Sem limite	< 1%
868.7 – 869.2	+14 / 25	Sem limite	< 0.1%
869.3 – 869.4	+10 / 10	< 25	Sem limite
869.4 – 869.65	+27 / 501	< 25	< 10%
869.7 -870	+7 / 5	Sem limite	Sem limite
2400 – 2483.5	+7.85 / 6	Sem limite	Sem limite

É importante referir que a legislação publicada pela ANACOM vai ao encontro da legislação em vigor na Europa, logo o sistema de telemetria pode trabalhar em qualquer país da comunidade Europeia com as mesmas configurações.

A tabela 2.9 mostra as vantagens e desvantagens entre a utilização das bandas ISM de 433 MHz, 868 MHz e 2400 MHz. Com a utilização das faixas de frequência de maior valor, tem a possibilidade de enviar maior quantidade de dados, porem a distância entre emissor e recetor diminui. Com esta condição é necessário escolher uma faixa de frequência que consiga transmitir todos os dados para uma distância que cubra o perímetro da pista onde se realiza a competição.

Tabela 2.9: Comparação entre as diferentes faixas de frequência ISM [26].

Faixas de frequências	Prós	Contras
433 MHz	– Longo alcance.	– Faixa de frequência sobrelotada.
	– Melhor penetração em ambientes <i>indoor</i> .	– Baixas taxas de transferência de dados.
868 MHz	– Menor congestionamento de banda.	– Plano de faixa de frequência complexo.
	– Alcance razoável.	
2400 MHz	– Altas taxas de transferência de dados.	– Curto alcance.

### Tipos de modulação

Para efectuar o envio de dados através de uma banda ISM, é preciso modular uma onda sinusoidal na frequência da faixa escolhida, denominada como portadora. É possível inserir informação numa onda portadora através da sua modulação em amplitude, frequência ou fase. Estes tipos de modulação são *Amplitude Shift Keying* (ASK),

*Frequency Shift Keying (FSK), Phase Shift Keying (PSK) e On-Off Keying (OOK) [27].*

**ASK** É a modulação por amplitude de uma onda portadora. Esta onda tem sempre a mesma frequência, mas é modelada com duas amplitudes. Cada valor lógico binário está associado a uma e só uma amplitude. Este tipo de modulação requer pouca ocupação do espectro e é de fácil implementação, mas é muito vulnerável ao ruído e apresenta mau desempenho em ambientes com poluição espectral [28].

**FSK** É uma das técnicas de modular sinais digitais, através da manipulação da frequência da onda portadora. Este tipo de modulação consiste na existência de duas ondas com diferentes frequências que representam os estados 1 e 0. Este tipo de modulação tem maior imunidade à poluição presente no ambiente, mas é de difícil implementação e não é a mais eficiente na utilização do espectro.

**PSK** É quando a onda portadora é modulada por manipulação de fase. A modulação por BPSK consiste em codificar um código binário, em que os estados 0 e 1 são modulados pelas equações 2.1 e 2.2. Este tipo de modulação requer pouca largura de banda e tem uma boa imunidade ao ruído, contudo, requer sincronização em frequência e fase entre o transmissor e o receptor.

$$s_1(t) = A\cos 2\pi f_c t, \quad 0 \leq t \leq T, \quad \text{for } 1 \quad (2.1)$$

$$s_2(t) = -A\cos 2\pi f_c t, \quad 0 \leq t \leq T, \quad \text{for } 0 \quad (2.2)$$

**OOK** É uma versão modificada da modulação ASK. Enquanto na modulação ASK o valor lógico 0 é representado por uma onda de menor amplitude, na modulação OOK não é produzida nenhuma onda portadora. Neste tipo de modulação só é enviada uma onda portadora caso o valor lógico seja 1. Este tipo de transmissão é o mais eficiente dos três apresentados anteriormente, pois quando é transmitido o valor lógico 0 não é dissipada potência de emissão, sendo que por outro lado não é possível distinguir o envio de “0” da inexistência de comunicação [29].

### Comparação entre tipos de modulação

Para começar a comparar os tipos de modulação abordados anteriormente, na figura 2.16 são apresentados os formatos das ondas portadoras geradas por cada tipo em resposta a um sinal binário. As ondas geradas pelos tipos de modulação ASK, OOK, FSK e PSK em resposta ao sinal binário a enviar.

De modo a comparar o desempenho dos diferentes tipos de modulação de sinal foi tido em conta o erro de transmissão e as eficiências espectral e energética. Com a relação do *Bit Error Rate* (BER) e o *Signal-to-Noise Ratio* (SNR), é possível verificar o erro de transmissão associado a cada tipo. O BER é o número de bits com erro observados na saída de um recetor, dividido pelo número total de bits enviados num determinado período. O SNR é o valor que compara a potência do sinal emitido com a potência do ruído de fundo expresso em decibéis. Com a figura 2.17 é possível visualizar o gráfico que apresenta o erro de transmissão consoante cada tipo de modulação. É possível

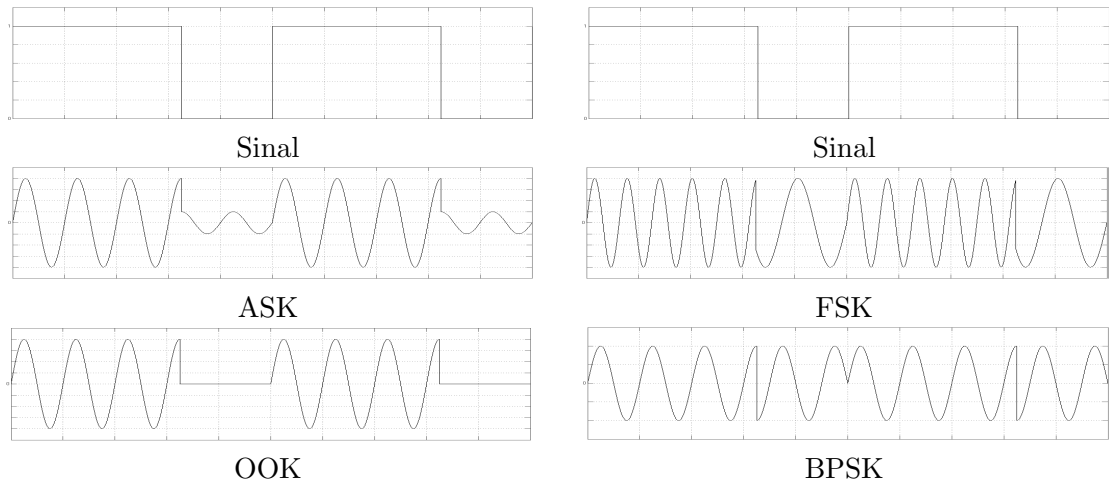


Figura 2.16: Diferenças das ondas portadoras obtidas, para diferentes tipos de modulação.

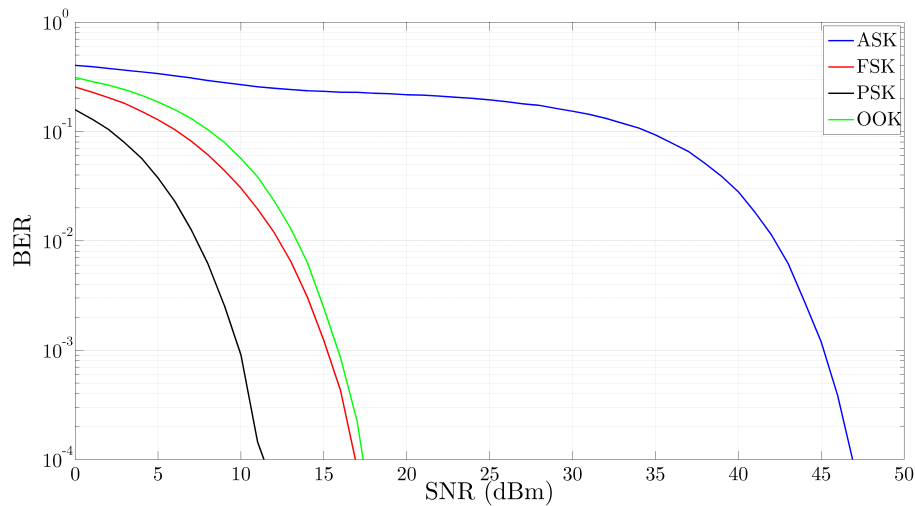


Figura 2.17: Relação entre BER e SNR para ASK, FSK, PSK e OOK [30].

observar que o tipo de modulação com maior imunidade ao ruído é por PSK, e com maior sensibilidade ao ruído é por ASK.

De forma resumida, a Tabela 2.10 compara as principais vantagens e desvantagens entre os diferentes tipos de modulação. Com as características apresentadas é verificado que o tipo de modulação com uma boa eficiência espectral [28] e maior imunidade ao ruído é a utilização do tipo PSK. Também se pode observar que o OOK possui uma boa eficiência espectral e energética, contudo o seu erro de transmissão é um pouco superior aos tipos PSK e FSK. Além do tipo FSK apresentar um bom resultado a nível de erro de transmissão, este método utiliza uma largura de banda superior aos métodos apresentados e tem baixas taxas de transferência.

### 2.3.2.3 Análise qualitativa do método de comunicação sem fios

De modo a escolher qual é o método utilizado para a transmissão de dados sem fios é feita uma análise entre a utilização de serviços prestados por operadoras e o uso de uma banda ISM.



Tabela 2.10: Tabela de comparação para diferentes tipos de modulação [28][29][31][32].

Tipo de modulação	Prós	Contras
ASK	– Requer pouca ocupação do espectro	– Vulnerável ao ruído
	– Fácil implementação	– Baixas taxas de transferência
	– Maior eficiência energética	
FSK	– Maior imunidade ao ruído	– Requer maior ocupação do espectro
		– Difícil implementação
		– Baixas taxas de transferência
PSK	– Requer pouca ocupação do espectro	– Requer sincronização em frequência em fase entre o transmissor e receptor
	– Maior imunidade ao ruído	
OOK	– Requer pouca ocupação do espectro	– Maior sensibilidade ao ruído do que os tipos PSK e FSK.
	– Maior eficiência energética na transmissão	

Alguns tipos de serviços prestados por operadoras seriam capazes de garantir as taxas de dados requeridas para a solução final. Para a utilização de um serviço é necessário estar dependente das infraestruturas de uma operadora móvel, da qual não é garantido a cobertura de serviço móvel em todo o percurso referido na secção 1.1. Com a utilização de uma banda ISM, é possível definir qual é a potência de transmissão do sinal, seleccionar o tipo de modulação dentro dos disponíveis, num *transceiver* rádio, e também seleccionar qual é a largura de banda a utilizar bem como a frequência da onda portadora de informação. Este método, tem como principal vantagem não ser associado a um serviço de uma operadora para utilização de uma banda ISM.

## 2.4 Soluções de mercado

Existem soluções de mercado desenvolvidas exclusivamente para o ramo da competição. No entanto, como este tipo de produtos são muito específicos, os preços encontrados para equipamentos deste tipo são elevados, tendo em consideração o baixo orçamento da equipa. De seguida são apresentadas duas possíveis soluções que se encontram à venda.

### AiM MXm Compact Dash Data Logger

Este equipamento é um produto fabricado pela a marca AiM[33], apresentado na figura 2.18. O equipamento é o produto que apresenta o preço mais baixo dentro de todas a gamas da marca AiM. Este produto só tem a possibilidade de enviar dados do equipamento para um computador quando a aquisição de dados está parada. De acordo



Figura 2.18: AiM MXm Compact Dash Data Logger[33].

com o fabricante as características do produto estão descritas na Tabela 2.11. É preciso ter em conta que para ser possível obter certas grandezas é necessário utilizar módulos adicionais, como por exemplo para leitura da compressão nas suspensões, temperatura dos pneus e pressão nos circuitos dos travões. Como tal foi feito um estudo do custo, apresentado na Tabela 2.12, para a aquisição deste produto com os respetivos módulos para a medição de todas as grandezas.

Tabela 2.11: Características *AiM MXm Compact Motorcycle Dash DataLogger*.

Caraterística	Descrição
Ecrã (resolução)	268*128
Conexões CAN	2
GPS	Integrado
Módulos externos	Expansão de Canal Expansão de Termopar Controlador Lambda SmartyCam HD
Entradas analógicas configuráveis	4 (até 1000Hz cada uma)
Plataforma inercial	3 eixos acelerómetro 3 eixos giroscópio 3 eixos magnetómetro
Comunicação sem fios	Sim
Memória interna	4 GB

Tabela 2.12: Estudo do custo para aquisição de uma solução *AiM*.

Descrição	Qtd.	Preço/Uni.	Preço
<i>Aim MXm Compact Motorcycle Dash Data Logger</i>	1	1 099 €	1 099 €
Medição da temperatura do pneu	2	112 €	224 €
Medição da compressão da suspensão	2	252 €	504 €
Medição da pressão no travão	2	200 €	400 €
		Total	2 227 €

## Starlane Davinci-II S Dashboard

É um produto fabricado pela a marca Starlane [34], apresentado na figura 2.19. Esta marca apresenta várias gamas de produtos, mas esta é a única gama capaz de implementar sensores para medição de travões, suspensões e temperatura dos pneus. Este equipamento não tem a opção para transmissão de dados em tempo real. De acordo



Figura 2.19: *Starlane Davinci-II S Dashboard* [34].

com o fabricante as características desta gama estão descritas na Tabela 2.13. Tal como o produto apresentado anteriormente, no caso de obter certas grandezas é obrigatório o uso de módulos adicionais. Como tal foi feito um estudo do custo, apresentado na Tabela 2.14, para a aquisição deste produto com os respetivos módulos para a medição de todas as grandezas.

Tabela 2.13: Características *Starlane Davinci-II S*.

Starlane Davinci-II S	
Ecrã (resolução)	800*400
Conexões CAN	1
GPS	Integrado
Módulos externos	WID-C (3 entradas analógicas + 1 CAN BUS)
Entradas analógicas	1 (posição de acelerador)
Plataforma inercial	3 eixos acelerómetro
	3 eixos giroscópio
	3 eixos magnetómetro
Comunicação sem fios	Sim (Bt 4.0 e Wifi)

Tabela 2.14: Estudo do custo para aquisição de uma solução *Starlane*.

Descrição	Qtd.	Preço/Uni.	Preço
<i>Starlane Davinci-II S</i>	1	1 045 €	1 045 €
<i>WID-C</i> Módulo de Expansão	2	230 €	460 €
Medição da temperatura do pneu	2	195 €	390 €
Medição da compressão da suspensão	2	230 €	460 €
Medição da pressão no travão	2	230 €	460 €
Total			2 815 €

## 2.5 Requisitos para a solução

Tendo em conta o estado de arte e as soluções de mercado foram definidos os requisitos para a solução, sendo estes compostos pelas características físicas da solução, pelos valores que têm de ser adquiridos e pelas as comunicações que a solução têm de suportar.

Os requisitos mínimos a ter em conta na constituição física da solução são:

- A solução é constituída por um sistema transmissor e um sistema recetor,
- Todos os equipamentos instalados na mota, são alimentados através da ficha diagnóstico disponível na mesma com um valor de tensão igual a 12 V CC,
- Solução com proteção contra poeiras e gotas de água,
- Sistema de aquisição de dados deve ser capaz de suportar implementações futuras,
- Peso total dos equipamentos instalados na mota tem de ter um peso inferior a 1.5 kg,
- O sistema transmissor tem de ser compacto.

A solução tem de ter a capacidade mínima de adquirir as grandezas referidas anteriormente na secção 1.3. Deste modo a solução tem de possuir as seguintes características:

- Um barramento CAN para adquirir dados através da ficha diagnóstico da mota e outro barramento para expansões futuras,

- Conseguir adquirir no mínimo 6 entradas analógicas, sendo duas para os sensores de pressão, duas para obter a posição das suspensões e as restantes para obter temperaturas dos pneus,
- Adquirir valores das acelerações a que o mota e o piloto estão sujeitos,
- Adquirir a localização da mota no globo terrestre.

Para os requisitos mínimos relacionados com as comunicações que a solução deve suportar são:

- Guardar todos os dados adquiridos pela solução num dispositivo de memória externo,
- Transmitir todos os dados adquiridos para um ecrã externo a uma taxa de atualização de 10 Hz,
- Transmitir os dados adquiridos em tempo real com uma taxa mínima de atualização de 5 Hz,
- Desabilitar a transmissão de dados em tempo real perante escolha do utilizador.

Esta secção permitiu verificar quais são as grandezas que têm de ser adquiridas da mota, verificar alguns dos métodos para obter diferentes tipos de valores, efetuar o estudo métodos de comunicação com e sem fios, verificar soluções de mercado com algumas das características que a solução tem de possuir e definir quais são os requisitos a ter em conta no desenvolvimento da mesma. É importante referir que esta secção apresenta os principais fundamentos teóricos para o desenvolvimento da solução desde da aquisição de dados até a transmissão dos mesmos.

Deste modo a próxima secção descreve a arquitetura e uma proposta de implementação da solução de modo a que a mesma cumpra com os requisitos mínimos apresentados anteriormente.

## Secção 3

# Arquitetura e proposta de implementação

Esta secção tem como objetivo descrever a arquitetura da solução desenvolvida. Inicialmente, tendo em conta os requisitos mínimos da solução especificados na secção 2.5 foi realizada uma distribuição dos blocos funcionais em dois sistemas. O primeiro sistema contém os blocos necessários para a aquisição, a gravação e a transmissão dos valores das grandezas medidas, enquanto o segundo sistema suporta a visualização em tempo real.

Por cada bloco funcional constituinte do sistema são apresentadas várias possibilidades de implementação, sendo feita uma comparação e a escolha de cada sensor ou dispositivo a implementar.

Com as especificações requeridas pelos sensores e dispositivos selecionados, é apresentada a unidade de processamento que cumpre com os requisitos da solução.

Por fim é apresentada a arquitetura para cada sistema. Nas arquiteturas apresentadas é feita uma distribuição dos dispositivos, pelas interfaces da unidade de processamento selecionada.

### 3.1 Constituição dos sistemas da solução

Nesta secção é apresentada a distribuição dos blocos funcionais da solução por dois sistemas como representado anteriormente na figura 1.5. Para cada sistema será apresentado um diagrama que distribui os blocos funcionais por partes. Para cada componente é feita uma breve descrição da sua função.

#### 3.1.1 Sistema de aquisição, gravação e transmissão de dados

Este sistema é constituído por quatro blocos funcionais, sendo eles a aquisição dados, o registo e leitura em memória não volátil, a parte computacional e a transmissão dos valores para o subsistema de receção de dados e para o ecrã da mota já existente.

A figura 3.1 apresenta o diagrama de blocos para o subsistema em questão. Para a aquisição de valores é preciso ter em conta os dados apresentados anteriormente na tabela 2.2. A ficha diagnóstico existente na cablagem da mota permite obter a alimentação para este subsistema e também possibilita a leitura de valores guardados na ECU. Como a mota de origem não possui os sensores para aquisição de todas as variáveis requeridas é necessário adicionar os seguintes sensores:

- 2 sensores para medir a temperatura nos pneus,

- 2 sensores para medir a pressão hidráulica nos circuitos dos travões,
- 2 potenciômetros lineares para medir a compressão nas suspensões,
- acelerómetro e giroscópio para medir as acelerações exercidas na mota em movimento,
- módulo *Global Positioning System* (GPS) para obter os valores da localização absoluta da mota.

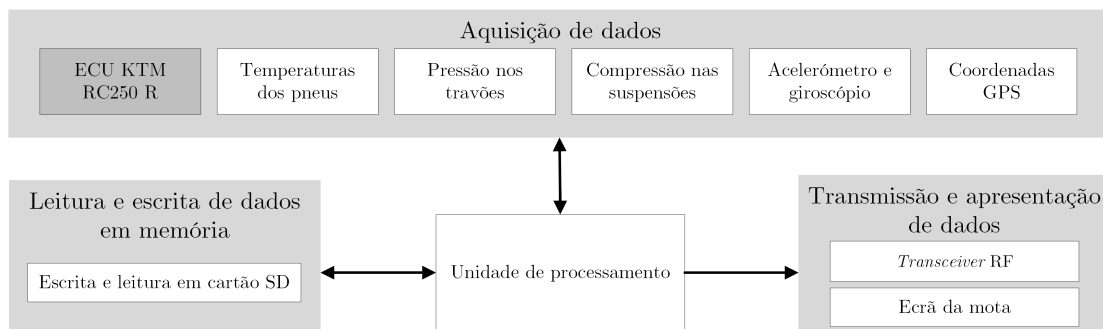


Figura 3.1: Diagrama de blocos do sistema de aquisição, gravação e transmissão de dados.

De modo a que o piloto tenha acesso a parte dos dados em andamento, é efetuada a transmissão de dados para um ecrã anteriormente desenvolvido pela equipa.

De modo a transmitir os dados recolhidos para a *box* em tempo real, quando a mota se encontra em testes ou treinos, é utilizado um módulo rádio que opera numa banda ISM.

Como redundância ao sistema de transmissão de dados e para garantir as frequências de aquisição requeridas, todos os valores obtidos são também guardados localmente em memória não volátil, tendo-se optado por um cartão SD. Para configurar o sistema, é efetuada a leitura dos parâmetros de funcionamento de um ficheiro armazenado no cartão SD. A configuração do sistema é um aspeto importante, uma vez que a mota durante a corrida não pode efetuar a transmissão de dados por rádio devido ao regulamento da competição.

Por fim, é utilizada uma unidade de processamento que faz a gestão dos processos associados às componentes deste sistema.

### 3.1.2 Sistema de receção de dados

Este sistema é constituído por três partes, sendo elas a receção de dados transmitidos da mota em pista, o envio de informação recebida para uma interface gráfica ou terminal série e a parte computacional.

Na figura 3.2 está representado o diagrama de blocos para o sistema em questão.

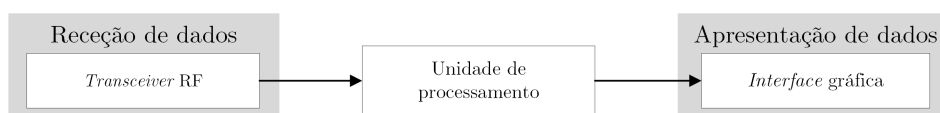


Figura 3.2: Diagrama de blocos do sistema de receção e apresentação de dados.

De modo a este sistema consiga receber dados remotamente em tempo real, utilizado-se um módulo rádio que opera com mesma banda ISM e modulação que o emissor,

localizado no sistema de aquisição, gravação e transmissão de dados. É importante referir que a solução desenvolvida não depende deste sistema para coletar dados, mas sim unicamente para a visualização em tempo real do comportamento da mota na pista numa interface gráfica. Com esta funcionalidade, a equipa localizada na *box* consegue efetuar uma análise e implementar melhorias na mota num menor intervalo de tempo.

É utilizado uma unidade de processamento para receber os dados do módulo rádio e enviar os mesmos para uma interface gráfica num computador.

## 3.2 Escolha de sensores e dispositivos

Nesta secção é feita a apresentação das escolhas realizadas para os sensores e componentes a serem utilizados no sistema. Os critérios de escolha dos componentes incluíram a satisfação dos requisitos, bem como o fator custo.

### 3.2.1 Aquisição de valores da ECU

Para a leitura de valores armazenados na ECU como referido na secção 2.1, utiliza-se a ficha diagnóstico disponibilizada na cablagem de origem da mota. Tendo em conta que a ficha contém os pinos CAN-H e CAN-L, é necessário adicionar uma interface CAN (controlador e *transceiver*) para a comunicação entre o sistema e a ECU.

Como nesta parte do documento ainda não foi definido qual a unidade de processamento a utilizar no sistema, existem duas possibilidades para trabalhar com um barramento CAN, apresentadas na figura 3.3. A possibilidade (a) é constituída por uma

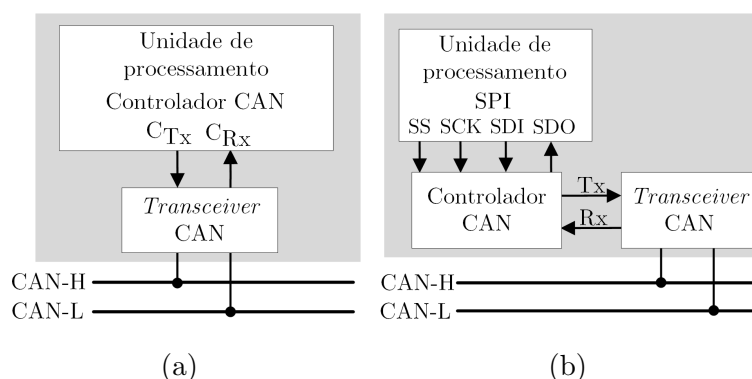


Figura 3.3: Duas possibilidades de comunicação com um barramento CAN.

unidade de processamento com um controlador CAN embutido e um *transceiver* CAN externo. Já a alternativa (b), passa por utilizar um controlador CAN externo, em que a comunicação com a unidade de processamento é feita através do protocolo SPI, e um *transceiver* CAN externo. Para ambos os casos apresentados verifica-se que é sempre necessário o uso de um *transceiver* CAN. Este dispositivo tem duas funções, a primeira função é converter os sinais entre o barramento e o controlador CAN para valores de tensão suportados, e a segunda é proteger o controlador CAN.

Um dos *transceivers* disponíveis no mercado é o Circuito Integrado (CI) MCP2561 produzido pela *Microchip*. Caso a unidade de processamento não possua um controlador CAN embutido, uma possível solução passa por utilizar o controlador externo tal como o CI MCP2515, também produzido pela *Microchip*.

Para obter os valores da ECU é utilizado um controlador CAN dedicado a comunicação com a mesma. Deste modo é possível garantir que todas as mensagens que

circulam no barramento não vão influenciar no funcionamento da mota. Isto é, como não é do conhecimento do autor todos os identificadores reconhecidos pela ECU, neste barramento só são transmitidas mensagens definidas pela norma SAE J1979 [35].

### 3.2.2 Módulos para sensores adicionais

Perante a lista de valores que a equipa forneceu no início deste desenvolvimento, é verificado que existe um conjunto de sensores que são repetidos. Isto é, por cada roda da mota são lidas três grandezas, sendo elas a temperatura do pneu, a pressão no travão e a posição do êmbolo da suspensão. Para instalar os sensores existem duas abordagens para o fazer. A primeira possibilidade passa por desenvolver um sistema centralizado, ou seja ter um sistema que garante o funcionamento de todos os requisitos. A segunda possibilidade tem como objetivo descentralizar a aquisição de dados, sendo assim o sistema formado por dois módulos dedicados a aquisição dos valores de cada roda e pelo sistema que efetua os restantes processos associados à aquisição, gravação e transmissão de dados. Ao aglomerar estes três sensores num módulo são verificadas algumas vantagens e desvantagens para o sistema de aquisição, tais como as apresentadas na Tabela 3.1.

Tabela 3.1: Vantagem e desvantagens da descentralização do sistema.

Implementação de sensores	Vantagens	Desvantagens
Sistema centralizado	Menor custo de produção	Cablagem mais complexa
	Desenvolvimento de um elemento	Vulnerável a interferência externos
	Menor número de componentes	Processamento centralizado
		Dimensões superiores
Sistema descentralizado	Cablagem mais simples	Custo de produção superior
	Instalação simples	Desenvolvimento de vários elementos
	Maior imunidade a interferência externos	
	Processamento descentralizado	
	Dimensões reduzidas	

Tendo em conta as diferenças entre as duas abordagens apresentadas, de modo a que a solução possua uma maior imunidade a interferências externas e se possa simplificar a instalação do sistema na mota, foi optado por implementar uma solução descentralizada. Logo a aquisição de dados será constituída por um sistema de aquisição, gravação e transmissão de dados que aglomera todas as variáveis adquiridas pelos módulos dedicados à leitura de sensores adicionais.

De modo a interligar os módulos associados a cada roda, é adicionado um segundo barramento CAN ao sistema de aquisição de dados. Com um segundo barramento são obtidas vantagens para o presente projeto, bem como para a equipa a longo prazo. Para o presente projeto esta abordagem simplifica a instalação do sistema na mota, por utilizar só uma interface do sistema de aquisição de dados. De futuro com a utilização de um segundo barramento CAN, caso a equipa deseje adquirir novas grandezas são obtidas as seguintes vantagens:

- Expansibilidade até um total de 29 módulos, pois este suporta até 30 nós [18],
- Não há necessidade de refazer toda a cablagem, pois esta é adaptável às necessidades,



- Não há necessidade de alterar *hardware* do sistema no módulo de aquisição de dados.

Para a implementação do segundo barramento CAN cada módulo tem de ter no mínimo uma interface CAN, como indicado na secção 3.2.1, logo a implementação de um segundo barramento também tem uma desvantagem, sendo ela o número total de componentes utilizados na construção do conjunto para o sistema de aquisição de dados. Tendo em conta os requisitos do presente projeto, de seguida são apresentadas as soluções para aquisição das grandezas a obter pelos dois módulos.

### 3.2.2.1 Temperatura dos pneus

Através da informação disponibilizada pela equipa *Repsol Honda*, a temperatura de um pneu durante uma prova de *MotoGp* pode atingir temperaturas de 120 °C [36]. Tendo em conta que os pneus são elementos móveis na moto, para obter a sua temperatura em andamento, perante os sensores apresentados anteriormente na secção 2.2.1, o sensor mais adequado é o pirómetro. Com a pesquisa realizada sobre este tipo de sensores disponíveis no mercado, foram encontradas duas soluções produzidos pela marca *Melexis* [37], sendo elas:

- MLX90614ESF-AAA,
- MLX90614ESF-BBA.

As características para cada sensor estão apresentadas na tabela 3.2. Perante os dois produtos apresentados é verificado que a diferença com maior importância para a solução é o número de zonas de medição por sensor. Isto é, o número de pontos numa superfície em que pode ser obtida uma temperatura. Neste caso, o preço dos sensores não é relevante, porque foi obtido o apoio do fabricante para o desenvolvimento desta solução. De modo a existir a possibilidade de medir a temperatura da superfície do pneu em duas zonas, a escolha recaiu sobre o sensor MLX60914ESF-BBA.

Tabela 3.2: Características de dois pirómetros da marca *Melexis* [38][39].

Parâmetro	MLX90614ESF-AAA			MLX90614ESF-BBA			Unid.	
	Min.	Típico	Máx.	Min.	Típico	Máx.		
Alimentação	4.5	5	5.5	2.6	3	3.6	V	
Corrente	-	1.3	2	-	1.3	2	mA	
Precisão (FE)	-	±0.5	-	-	±0.5	-	°C	
Número de zonas	-	1	-	-	2	-	Unit.	
Comunicação								
PWM	Resolução	-	10	-	-	10	-	bits
	Resolução de medida	-	0.14	-	-	0.14	-	°C
	Faixa de medição	-20	-	120	-20	-	120	°C
I <sup>2</sup> C	Resolução	-	17	-	-	17	-	bits
	Resolução de medida	-	0.02	-	-	0.02	-	°C
	Faixa de medição	-70	-	380	-70	-	380	°C
Preço	12,71 €			20,73 €				

Os valores obtidos por este sensor podem ser recolhidos através de um sinal *Pulse With Modulation* (PWM) ou por I<sup>2</sup>C. Tendo em conta que é obtida uma maior resolução da temperatura com a utilização do protocolo I<sup>2</sup>C, este é o método escolhido para a aquisição desta grandeza.

### 3.2.2.2 Pressão dos travões

O correto funcionamento dos travões na mota é fundamental para a segurança do piloto. Tendo em conta as características das bombas de travão, fornecidas pela a organização do evento [40], verifica-se que a sua pressão máxima de trabalho é 70 bar. O fabricante dos travões recomenda a utilização de óleo hidráulico do tipo DOT4, sendo que este contém agentes corrosivos [41]. Para escolher o sensor indicado para o sistema, foram tidas em conta as seguintes características:

- Intervalo de pressão mínima de trabalho [0-70]bar,
- Tensão mínima de alimentação 12 V,
- Preparado para trabalhar com substâncias corrosivas.

Com a pesquisa realizada foram encontrados dois sensores compatíveis com o sistema. Um da marca Schneider e outro produzido pela Cynergy3. Tendo em conta as características apresentadas na tabela 3.3 relativamente aos dois sensores, verifica-se que existem três diferenças entre eles. A primeira é a precisão de medida, a segunda é o tipo de sinal da saída analógica e a terceira é o preço de cada sensor. O sensor com menor preço, com maior precisão de medida e com uma saída analógica de [0-10] V é o produto XMLG100D71. O sensor IPS-G10035 tem como vantagem possuir uma saída analógica em corrente, que se torna mais imune a ruídos produzidos pelo o funcionamento da mota. Tendo em conta que os dois sensores cumprem com os requisitos do sistema, o sensor escolhido é o produto XMLG100D71, porque foi obtido o apoio da marca para o projeto.

Tabela 3.3: Características de dois sensores de pressão[42][43][44].

Parâmetro	Schneider XMLG100D71	Cynergy3 IPS-G10035	Unit.
Alimentação	[11.4...33]	[9-32]	V
Corrente	<20	-	mA
Precisão de medida (FE)	±0.3	±0.5	bar
Faixa de medição	[0...100]	[0...100]	bar
Saída analógica	[0...10] V	[4...20] mA	-
Preço por unidade	112.28	150.00	€

De modo a obter o valor digital que represente a pressão nos circuitos hidráulicos dos travões, é necessário recorrer a um ADC. Para converter o valor analógico gerado pelo o sensor de pressão, é necessário acondicionar o sinal para os níveis de tensão da entrada do ADC.

### 3.2.2.3 Posição das suspensões

O bom funcionamento das suspensões permite ao piloto sentir-se mais confortável na mota, podendo resultar num melhor desempenho em pista. Para a equipa, a leitura da posição do êmbolo das suspensões permite conhecer o seu regime de trabalho. A equipa com estes dados pode assim fazer uma afinação mais precisa das suspensões, de modo a aumentar o desempenho do veículo em pista.

Para procurar os sensores indicados para o sistema, foram verificados os cursos úteis das suspensões escolhidas pela equipa. As suspensões são produzidas, unicamente para o evento *Motostudent*, pela a marca *Andreani MHS*. A folha das características das suspensões, disponibilizada pela a organização, indica que a suspensão dianteira tem um

curso de 110 mm e a traseira tem um curso de 59 mm. No início da pesquisa deste tipo de sensores, foi verificado que os fabricantes apresentam produtos com o curso distribuído em intervalos de 25mm entre [25-200]mm. Deste modo foram selecionados três sensores com o comprimento de 75mm para a traseira, e três sensores com o comprimento de 125mm para a dianteira como apresentado na Tabela 3.4.

Tabela 3.4: Características de dois potenciômetros lineares[45][46][47].

Parâmetro	Miran		Fama		TE Connectivity		Unid.
	Série KPM 12	Série KPM 12	Série MLS130	Série MLS130	Série MLP	Série MLP	
Curso do sensor	75	125	75	125	75	125	mm
Precisão linear (FE)	±0,1	±0.05	±1		±0.5		%
Vibrações suportadas	-		até 2000		até 2000		Hz
Diâmetro externo	12,7		13		9,5		mm
Peso	-		94	108	33	43	g
Preço por unidade	33,26		165,96		721,52		€

Em termos de precisão linear ao fim de escala, os produtos disponibilizados pela a marca *Miran* são os que apresentam melhores valores. Como para este sistema, os sensores serão sujeitos a vibrações provenientes do funcionamento da mota e também pela a irregularidade do piso, os produtos das marcas Fama e TE são os únicos que apresentam dados relativamente à robustez e à imunidade a vibrações. Em termos de diâmetro e peso os sensores que apresentam melhores valores são os da série MLP, uma vez que, o fabricante da série KPM não apresenta qualquer dado sobre o peso associado a cada sensor. Comparando as três séries de sensores relativamente ao preços, verifica-se que os sensores da série KPM apresentam um preço mais baixo e que os da série MLP têm um custo mais elevado.

Tendo em conta o baixo orçamento da equipa e que não foi obtido nenhum apoio para estes sensores, a escolha recaiu nos produtos da marca *Miran*.

Como para a medição dos travões apresentada na secção 3.2.2.2, neste caso também será necessário recorrer ao uso de acondicionamento de sinal e de um ADC para obter o valor digital da posição do êmbolo das suspensões.

### 3.2.3 Módulo inercial

Na construção de uma mota é tido em conta a localização do centro de massa da mesma. O centro de massa tem uma ação importante sobre o comportamento dinâmico de uma mota. Com a obtenção dos valores inerciais, tais como os ângulos de Euler (*roll*, *pitch* e *yaw*), é possível averiguar três comportamentos da mota [48]:

- O (*roll*) indica a velocidade a que a mota se inclina lateralmente,
- O (*pitch*) indica que é imposta ao piloto uma maior dificuldade na mudança de direção quando este apresenta um valor elevado,
- O (*yaw*) indica a facilidade da transferência de massa da traseira para a frente da mota e vice-versa. Quando o valor deste ângulo é elevado indica que num caso de travagem o piloto pode ser projetado para a frente e no caso de aceleração a roda da frente pode perder o contacto com a pista.

A monitorização dos valores inerciais é uma mais valia para melhorar e tornar mais segura a condução da mota. Com a pesquisa realizada sobre módulos inerciais disponíveis no mercado, foram encontradas duas possíveis soluções:

- MM7150-AB1 produzido pela Microchip,

- OKY3255 produzido pela OkyStar .

A Tabela 3.5 apresenta as principais características dos módulos inerciais. Perante os dois produtos apresentados é verificado que as duas diferenças com maior importância para a solução são a resolução de cada ângulo e a comunicação disponibilizada por cada um. Dado que o módulo foi oferecido pelo fabricante, o módulo a utilizar é o MM7150-AB1. Este módulo será incluído no sistema de aquisição de dados, sendo que a comunicação entre este e a unidade de processamento do sistema é realizada através do protocolo I<sup>2</sup>C.

Tabela 3.5: Características de dois módulos inerciais [49][50].

Parâmetro	MM7150-AB1			OKY3255			Unid.
	Min.	Típico	Máx.	Min.	Típico	Máx.	
Alimentação	2.3	-	3.6	3	-	5	V
Corrente	-	13.65	-	-	15	-	mA
Resolução	-	0.061	-	-	0.1	-	°
Número de eixos	-	9	-	-	9	-	Unit.
Comunicação	I <sup>2</sup> C			SPI, UART			
Preço	25,43			44,62			€

### 3.2.4 Módulo GPS

Para efetuar uma melhor interpretação dos dados adquiridos a associação das grandezas obtidas a uma posição absoluta na pista é fundamental. É necessário que o sistema de aquisição de dados adquira a posição absoluta da mota no globo terrestre. Com a utilização de um módulo GPS é possível obter as coordenadas de um objeto no globo terrestre em ambientes *outdoor*. Através da pesquisa de dispositivos GPS de baixo custo e de dimensões reduzidas, os produtos que mais se evidenciam são:

- Módulo L80 produzido pela Quectel,
- ORG4472-PM04 fabricado pela OriginGPS.

Na figura 3.4 estão representados os dois produtos em estudo e de seguida são apresentadas as características dos mesmos na Tabela 3.6.

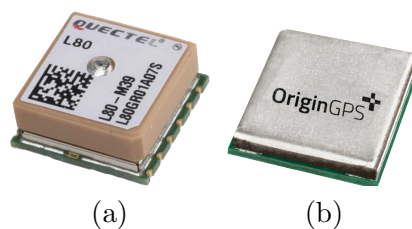


Figura 3.4: Módulos GPS em estudo, sendo (a) Quectel L80 e (b) ORG4472.

A diferença entre os dois produtos apresentados é que o ORG4472-PM04 requer desenvolvimento de eletrónica para ser possível fazer a integração do mesmo, enquanto que o Quectel L80 não precisa. Perante esta grande diferença, de modo a poupar tempo no desenvolvimento da solução e pelo preço ser mais reduzido, a escolha para módulo GPS é o dispositivo Quectel L80. Tendo em conta a escolha realizada é verificado que só pode ser utilizado o protocolo UART para comunicação com este módulo.

Tabela 3.6: Características de dois dispositivos GPS [51] [52].

Parâmetro	Quectel L80			ORG4472			Unid.
	Min.	Típico	Máx.	Min.	Típico	Máx.	
Alimentação	3.0	-	4.3	1.71	1.80	1.89	V
Corrente	-	20	25	-	37	47	mA
Precisão horizontal	-	<2.5	-	-	<2.5	-	m
Sensibilidade	-	-165	-	-	-163	-	dBm
Taxa atualização	-	10	-	-	5	-	Hz
Comunicação	UART			I <sup>2</sup> C, SPI e UART			
Preço	14,38			22,60			€

### 3.2.5 Armazenamento de dados em memória não volátil

Para o desenvolvimento da solução é tida em conta a necessidade de ter garantido que todos os dados recolhidos, mesmo que sejam transmitidos por rádio são guardados num dispositivo de memória. De modo a ter um acesso mais prático ao dispositivo de memória, devido a grande parte dos computadores terem um leitor de cartões SD, a escolha do dispositivo de memória para este sistema é um cartão SD. O diagrama de blocos da figura 3.5 mostra como é realizada a comunicação entre uma unidade de processamento (*host*) e um cartão SD.

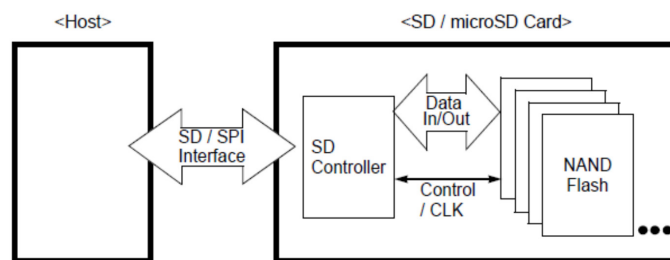


Figura 3.5: Diagrama de blocos de um cartão SD [53].

Um cartão SD, internamente possui uma interface para receber/enviar informação, da qual comunica com outro dispositivo através do protocolo SPI. Neste caso só é necessário utilizar uma interface SPI da unidade de processamento para efetuar a comunicação com o cartão SD e um conector para o mesmo no módulo principal do sistema.

### 3.2.6 Porta de *debug* e transmissão de dados para ecrã

Perante os requisitos da equipa para a transmissão de dados para o painel de instrumentos, foi verificado que teria que ser utilizado o protocolo USB para a comunicação entre o sistema de transmissão de dados e o ecrã. Também foi verificado pelo autor que seria uma mais valia para o desenvolvimento de código todos os sistemas e módulos possuírem uma porta para *debug*. Tendo em conta as necessidades referidas e também de modo a reduzir o número de componentes diferentes no sistema, a pesquisa para estes dois requisitos foi feita em conjunto. As principais características sobre os resultados obtidos da pesquisa para interface USB estão representados na Tabela 3.7.

Os dois produtos apresentados são produzidos pela FTDI Chip, em que as principais diferenças entre eles são o consumo de corrente, o tamanho dos *buffers* de dados e o preço. Como para todo o sistema são utilizadas várias interfaces deste tipo, foi dada maior importância ao consumo de corrente e ao preço. O componente para a interface USB-UART escolhido foi o FT232RQ.

Tabela 3.7: Características de duas interfaces USB-UART. [54][55].

Parâmetro	FT232RQ			FT232BL			Unid.	
	Min.	Típico	Máx.	Min.	Típico	Máx.		
Alimentação	4	-	5.25	4	-	5.25	V	
Corrente	-	15	-	-	25	-	mA	
Buffer	Tx	256			128			Byte
	Rx	128			384			
Preço unitário	4.04			5.16			€	

### 3.2.7 Transmissão de dados sem fios

Para a comunicação entre os sistemas é preciso ter em conta uma taxa de transmissão de dados na ordem dos kbit/s e uma distância razoável entre pontos. Sabendo que se optou pela comunicação através de uma banda ISM como indicado na secção 2.3.2.3, é preciso escolher qual a frequência base em que vai ser transmitida a informação.

Através da Tabela 2.9 é possível verificar as principais diferenças entre as frequências ISM disponibilizadas para utilização em território Europeu. A faixa dos 433 MHz tem como principais desvantagens estar sobrelotada e oferecer baixas taxas de transferência de dados, logo é excluída a opção de trabalhar nesta faixa de frequência. A banda ISM dos 2400 MHz apresenta uma alta taxa de transmissão de dados, no entanto esta tem um curto alcance, logo conclui-se que esta banda também não se adequa à solução. Assim, fica a opção de transmitir os dados entre sistemas através da faixa de frequência dos 868 MHz.

Sabendo que os *transceivers* a utilizar tem de trabalhar na banda dos 868 MHz e que a taxa de dados tem que ser superior a 5.35 kbit/s, como indicado anteriormente na secção 2.1, foi efetuada uma pesquisa sobre algumas das soluções de mercado disponíveis com as características indicadas. Assim foram encontrados três possíveis produtos, apresentados na Tabela 3.8.

Tabela 3.8: Características de três *transceivers* RF [56][57][58].

Parâmetro	RFM95PW-868-S2			RF-LORA-868-SO			SX868 RF			Unid.
	Min.	Típico	Máx.	Min.	Típico	Máx.	Min.	Típico	Máx.	
Alimentação	1.8	-	3.7	1.8	3.3	3.6	2.4	-	3.6	V
Corrente TX	-	650	700	-	125	-	-	55	-	mA
Corrente RX	-	13	15	-	10	-	-	40	-	mA
Potencia TX	-	-	27	-	-	20	-	-	13	dBm
Sensibilidade RX	-	-	-139	-	-	-130	-	-	-113	dBm
Tipo de modulação	FSK, GFSK, GMSK, LORA, MSK, OOK			FSK, GFSK, GMSK, LORA, MSK, OOK			GFSK			
Frequência	-	868	-	-	868	-	-	868	-	MHz
Bitrate FSK	1.2	-	300	1.2	-	300	-	-	-	Kbps
Bitrate OOK	1.2	-	32.768	1.2	-	32.768	-	-	-	Kbps
Bitrate GFSK	-	-	-	-	-	-	10	-	80	Kbps
Comunicação	SPI			SPI			SPI, USART			
Preço	17.33			15.30			29.37			€

Para a escolha do *transceiver* mais adequado à solução é tida em conta a potência de transmissão, a sensibilidade de receção de dados, os tipos de modulação e os bit rates de transmissão associados a cada um. Dos três produtos em estudo, o *transceiver* SX868 RF é o que apresenta piores prestações em todas as características. Relativamente aos outros dois *transceivers*, o produto RF-LORA-868-SO destaca-se por ter um consumo energético menor, mas o RFM95PW-868-S2 apresenta melhores características em termos de potência de transmissão e sensibilidade de receção. Tendo em conta que para

uma potência de transmissão superior, a distância entre pontos suportada tende a ser maior, a escolha recai sobre o produto RFM95PW-868-S2.

### 3.3 Estudo da unidade de processamento a utilizar na solução

Sendo já conhecido o número mínimo e o tipo de interfaces a utilizar, é possível efetuar a escolha da unidade de processamento. De modo a facilitar o trabalho de desenvolvimento, a unidade de processamento a escolher é igual para todos os sistemas da solução.

Na Tabela 3.9 são contabilizadas as interfaces necessárias para os sistemas ou módulos da solução. O valor indicado por cada coluna da linha "Total", corresponde ao maior número de cada tipo de interfaces a utilizar entre todos os sistemas ou módulos. Os requisitos mínimos para a unidade de processamento a utilizar são sete pinos digitais (entradas ou saídas), dois canais analógicos, uma interface I2C, uma interface SPI, três interfaces UART, três pinos de interrupção e por fim dois controladores CAN.

Tabela 3.9: Levantamento do número de interfaces da unidade de processamento.

Sistema/Módulos	Interface da unidade de processamento						
	IO's Digitais	Canais Analog.	I2C	SPI	UART	Interrupções	CAN
Aquisição	7	-	1	1	3	3	2
Sensores adicionais	-	2	1	-	1	-	1
Receção	-	-	1	-	-	-	-
Total	7	2	1	1	3	3	2

De modo a facilitar o desenvolvimento da solução em termos de testes de circuitos eletrónicos bem como para o início de desenvolvimento de código, é uma mais valia o uso de uma unidade de processamento que se encontre embutido numa placa de desenvolvimento disponível no mercado.

Para o desenvolvimento do sistema surgiram duas possibilidades de placas de desenvolvimento que cumprem com os requisitos da solução, sendo elas a placa Max32 produzida pela *Digilent* e a placa STM32MP157A-DK1 fabricada pela *StMicroelectronics*. A placa Max32 tem como unidade de processamento o microcontrolador PIC32MX795F512L e a placa STM32MP157A-DK1 utiliza um microprocessador STM32MP157A [59][60].

A nível de desenvolvimento de hardware é mais simples a implementação de um microcontrolador do que um microprocessador, pois o STM32MP157A não está preparado para interagir diretamente com as suas interfaces. Tendo em conta esta diferença e o conhecimento prévio do autor da implementação de um microcontrolador em sistemas embebidos, foi selecionada a placa de desenvolvimento Max32 representada na figura 3.6.

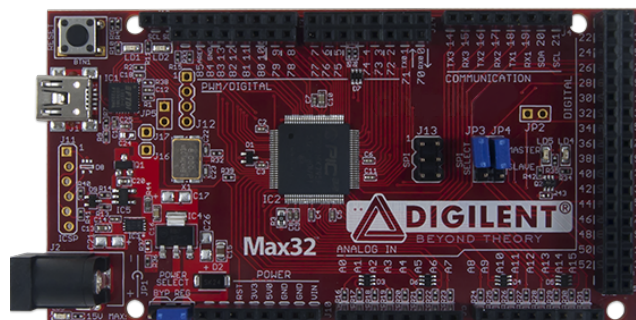


Figura 3.6: Placa de desenvolvimento Max32 [61].

### 3.4 Arquitetura da solução

Para concluir esta secção, serão apresentados e descritos os diagramas de blocos de cada parte constituinte da solução. Como referido, a solução é constituída por dois sistemas, o de transmissão e o de receção de dados.

A arquitetura do sistema de aquisição, gravação e transmissão de dados instalado na mota é formado pelo sistema principal e pelos dois módulos para aquisição de sensores adicionais, um na roda traseira e outro na dianteira. Tendo em conta que a ECU é fundamental para o funcionamento da mota e o ecrã é um requisito proposto pela equipa, a arquitetura geral do sistema a instalar na mota está representada na figura 3.7.

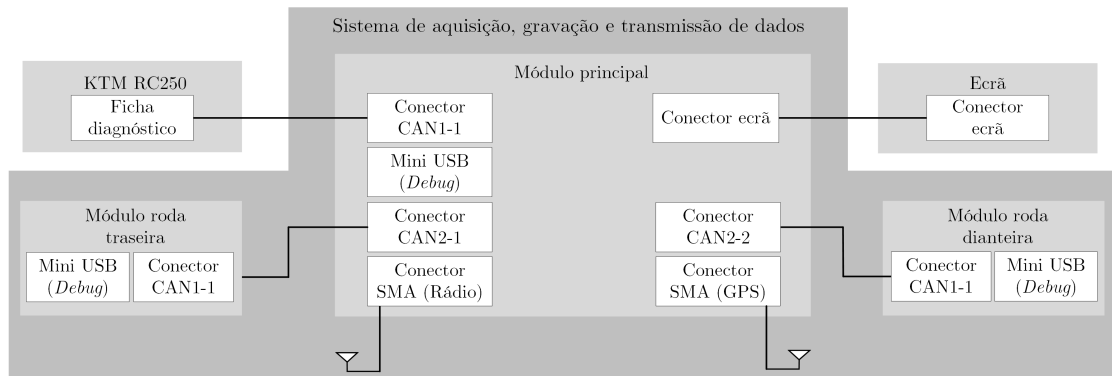


Figura 3.7: Arquitetura geral do sistema a instalar na mota.

O diagrama de blocos da figura 3.8 apresenta a arquitetura para os sub-módulos dos sensores adicionais, tais como leitura da pressão dos travões, posição dos êmbolos das suspensões e temperatura dos pneus.

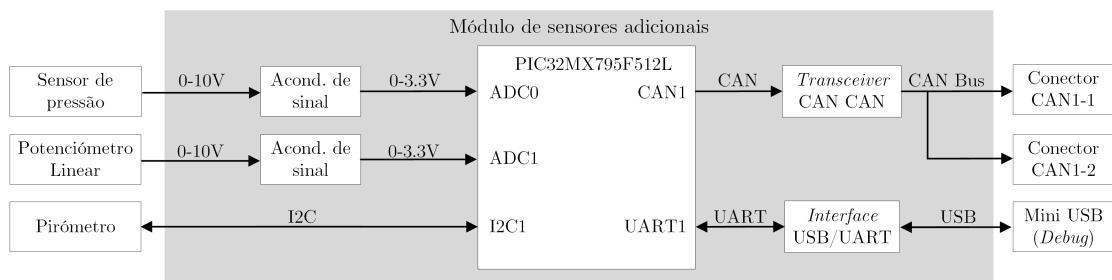


Figura 3.8: Diagrama de blocos do módulo para sensores adicionais.

Os sensores para a leitura dos travões e das suspensões utilizam dois canais do ADC do microcontrolador, os canais ADC0 e ADC1. Por cada canal utilizado é efetuado um acondicionamento de sinal, de modo que o sinal emitido pelo sensor possa ser lido pelo respetivo canal do ADC. O pirómetro utilizado para a leitura da temperatura dos pneus utiliza a interface I2C1 do microcontrolador. De modo a ser possível efetuar o *debug* durante o desenvolvimento do código deste módulo, foi adicionada uma interface UART/USB. O módulo para sensores adicionais tem embutido uma interface CAN, de modo a que os dados adquiridos sejam enviados para o sistema principal. Para a mesma interface CAN existem dois conectores para que a implementação de expansões futuras possam ser realizadas pela equipa.



O diagrama de blocos presente na figura 3.9 é referente à constituição do sistema principal de aquisição, gravação e transmissão de dados.

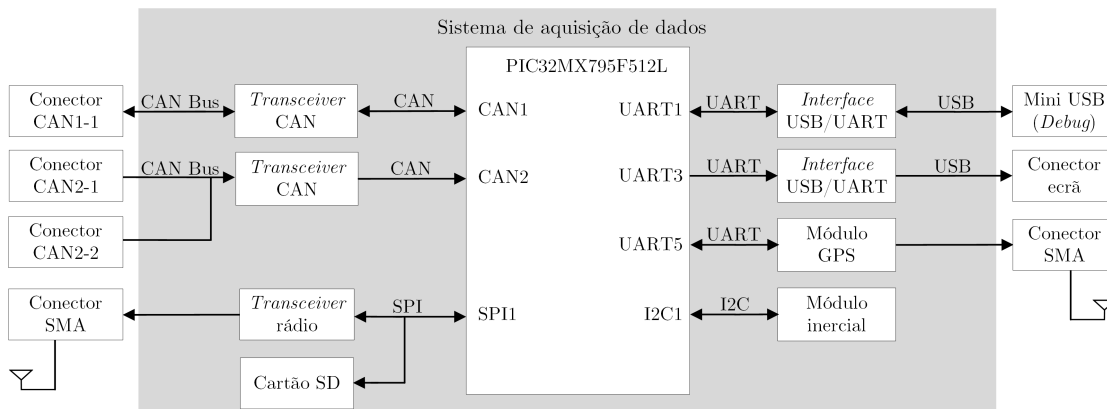


Figura 3.9: Diagrama de blocos do sistema para aquisição de dados.

Como referido anteriormente é utilizada uma interface CAN dedicada à leitura dos valores provenientes das ECU da mota. A interface CAN1 é constituída por um *transceiver* e pelo primeiro controlador que se encontra embutido no microcontrolador. Para este sistema receber os dados provenientes dos dois módulos associados aos sensores adicionais, é utilizada a interface CAN2 que se distingue pela utilização do segundo controlador CAN do microcontrolador associado ao sistema principal. De modo a obter as acelerações e ângulos de Euler é utilizado o módulo inercial, em que a comunicação entre o módulo MM7150 e o microcontrolador do sistema principal é realizada através da interface I2C1. Para finalizar a aquisição de dados é utilizado um módulo GPS que utiliza a interface UART5 do microcontrolador também no sistema principal.

A transmissão de dados na mota é efetuada para dois componentes, o painel de instrumentos e o *transceiver* rádio. Para o piloto obter a informação da mota em funcionamento, é utilizada a interface UART3 do microcontrolador de modo a enviar a informação disponível da mota para o painel de instrumentos. Para transmissão de dados por rádio é utilizada a interface SPI1 do microcontrolador para a comunicação com o *transceiver* rádio. De modo a gravar todos os dados recolhidos quando a mota se encontra em movimento e como redundância a possíveis falhas de comunicação rádio, é utilizado um cartão SD, sendo que este utiliza a mesma interface que o módulo rádio.

Este sistema também dispõe de uma porta para *debug*, de modo a facilitar o desenvolvimento de código durante a sua construção. A porta de *debug* implementada utiliza a interface UART1 do microcontrolador.

Por fim para completar toda a solução é utilizado o sistema para receção de dados, da qual o diagrama de blocos se encontra representado na figura 3.10.

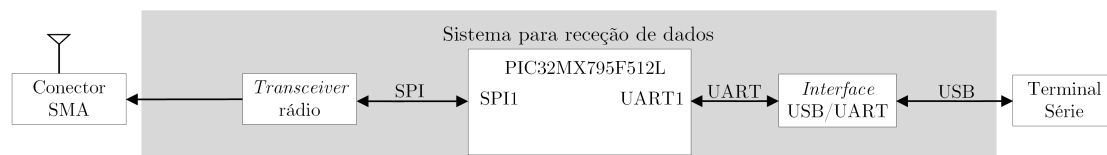


Figura 3.10: Diagrama de blocos do sistema para receção de dados.

Para efetuar a receção de dados é utilizado um *transceiver* rádio que comunica com o microcontrolador através da interface SPI1. Todos os dados recebidos por rádio são

enviados para um computador, através da interface UART1 do microcontrolador. A alimentação deste sistema é efetuada através da ligação efetuada pela porta USB utilizada para efetuar a transmissão de dados para uma interface gráfica.

Com esta secção foi possível verificar quais são os dispositivos selecionados para a solução de modo a que a mesma cumpra com requisitos mínimos. Também foi obtida uma proposta para a arquitetura da solução, sendo esta constituída por um sistema descentralizado que efetua toda a aquisição e transmissão de dados e por um sistema que realiza a recepção dos mesmos. É preciso ter em conta que o sistema de aquisição ao ser descentralizado é formado por dois módulos desenvolvidos para a aquisição de sensores adicionais.

Com a arquitetura definida nesta secção de seguida é apresentado o desenvolvimento efetuado a nível de *hardware* para as varias componentes da solução.

## Secção 4

# Desenvolvimento de *hardware*

Esta secção tem como objetivo descrever o desenvolvimento de *hardware* da solução. Inicialmente é apresentada a base de implementação comum entre todas as partes constituintes da solução. Esta base é constituída pela alimentação do microcontrolador e das componentes associadas a cada parte da solução, pela porta de programação *In Circuit Serial Programming* (ICSP), pela interface para *debugging* e pelo oscilador externo.

No dimensionamento do módulo para aquisição de sensores adicionais são descritos três tópicos. O primeiro refere-se ao acondicionamento de sinal associado às entradas analógicas, o segundo apresenta a implementação do pirómetro e o terceiro descreve a interface CAN para comunicação com o sistema principal de aquisição de dados.

Para o sistema principal de aquisição, gravação e transmissão de dados são apresentadas e descritas as componentes com as seguintes funções:

- Comunicação com a ECU e com os módulos para aquisição de sensores adicionais através de barramentos CAN,
- Leitura de valores de acelerações e ângulos de *Euler* adquiridas pelo módulo inercial,
- Aquisição dos valores das coordenadas GPS,
- Transmissão de dados para ecrã da mota,
- Comunicação com cartão microSD,
- Transmissão dos dados recolhidos por RF.

O sistema de receção de dados é composto por duas componentes, sendo elas, uma para a receção de dados por RF e a outra para comunicação de dados para um terminal série ou para uma interface gráfica.

No final da descrição de cada parte constituinte da solução é apresentada a *Printed Circuit Board* (PCB) associada a cada um.

Após a descrição de todas as componentes de *hardware* da solução, apresenta-se o custo associado à produção de toda a solução. Por fim realiza-se uma comparação entre o preço de montagem da solução com os preços de custo das soluções de mercado apresentadas na secção 2.4.

### 4.1 *Hardware* comum

Como referido anteriormente na secção 3.3, de modo a facilitar o desenvolvimento dos diferentes componentes da solução é utilizado o mesmo microcontrolador, sendo que para este desenvolvimento foi tido em conta o esquema da placa Max32 [62].

### 4.1.1 Alimentação

Para dimensionar a alimentação comum a todas as partes constituintes da solução, foi realizada uma estimativa do consumo de energia. A estimativa do consumo de corrente para toda a solução foi realizada sobre o sistema de aquisição, gravação e transmissão de dados, porque é o que possui um maior número de dispositivos integrados, sendo estes os que apresentam consumos mais elevados. Também foi tido em conta que o valor da tensão de alimentação para todo sistema de aquisição de dados é de 12 V CC, sendo esta obtida através da ficha diagnóstico disponível na cablagem da mota.

Como o fabricante do *transceiver* RFM95PW868 não indica os valores de consumo de energia durante a transmissão de dados, foi feito um pequeno teste de modo a verificar o consumo do mesmo. Na figura 4.1 estão apresentados os valores obtidos nesse teste. É preciso ter em conta que durante o teste o *transceiver* alternou o seu estado de funcionamento entre o modo transmissor e de *standby*. Deste modo é possível verificar que enquanto o módulo se encontra a transmitir dados consome uma corrente de aproximadamente 700 mA.

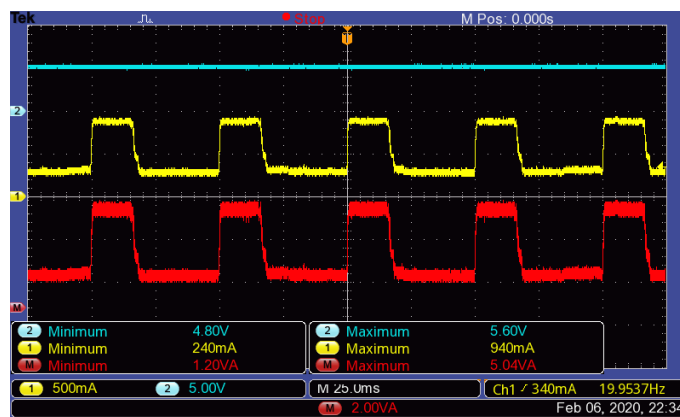


Figura 4.1: Teste individual ao consumo de energia do *transceiver* rádio.

Na Tabela 4.1 é apresentada a estimativa realizada ao consumo de energia para diferentes níveis de tensão no sistema de aquisição de dados, perante as características disponibilizadas pelos fabricantes dos dispositivos integrados no sistema. É preciso ter em consideração que no funcionamento normal do sistema as componentes não vão estar a funcionar todas ao mesmo tempo, logo os valores das correntes indicados aplicam-se à situação mais desfavorável. Para os dispositivos que são alimentados a 3.3V é preciso um regulador de tensão que forneça uma corrente de superior a 233.65 mA. Já para alimentar o regulador de tensão com saída de 5V é preciso garantir uma corrente mínima de 973.65 mA.

Com os dados referidos anteriormente foi verificado se os reguladores de tensão que são utilizados na placa Max32, conseguiriam suportar a corrente necessária para o funcionamento de todo o sistema. Os reguladores utilizados na placa Max32 são:

- MCP1725-330F produzido pela *Microchip*, regulador linear de tensão para 3.3V com a corrente máxima de saída de 500 mA e com o preço de 0.46€[63],
- NCP1117ST50T3G fabricado pela *ON Semiconductor*, regulador linear de tensão para 5V com a corrente máxima de saída de 800 mA e com o preço de 0.29€[64].

Através dos valores disponibilizados nos *datasheets* dos reguladores de tensão indicado anteriormente, verifica-se que o CI MCP1725-330F consegue suportar a corrente

Tabela 4.1: Estimativa do consumo de corrente.

Componente	3.3V	5V
PIC32 (80MHz)	120 mA	-
<i>Debug</i> - FT232RQ	-	15 mA
Ecrã - FT232RQ	-	15 mA
Módulo GPS	25 mA	-
Cartão SD (escrever)	75 mA	-
Módulo inercial	13.65 mA	-
<i>Transceiver</i> CAN1	-	5 mA
<i>Transceiver</i> CAN2	-	5 mA
Módulo RFM95PW (+27dBm)	-	700 mA
Conversor 5V-3.3V		233.65 mA
Total	233.65 mA	973.65 mA

necessária para os dispositivos alimentados a uma tensão de 3.3 V e que o regulador NCP117ST50T3G não tem capacidade de fornecer a corrente necessária ao sistema principal. Logo foi efetuada uma pesquisa de componentes que regulam uma tensão de 12 V para 5 V com capacidade de fornecer a corrente requerida pelo sistema de aquisição de dados.

De modo a dimensionar a alimentação do sistema com um coeficiente de segurança foi multiplicado o valor da corrente estimada por um fator de 1.2, sendo verificado que a corrente fornecida pelo CI tem de ser superior a 1.2 A aproximadamente. Após a pesquisa foram encontrados dois componentes que cumprem com os descritos anteriormente, sendo eles o OKI-78SR-5/1.5 produzido pela *Murata* e o R-78B5.0-1.5 fabricado pela *Recom*. As principais características dos componentes encontrados são apresentadas na tabela 4.2.

Tabela 4.2: Características de dois conversores CC-CC [65][66].

Parâmetro	OKI-78SR-5/1.5			R-78B5.0-1.5			Unid.
	Min.	Típico	Máx.	Min.	Típico	Máx.	
Alimentação	7	12	36	6.5	-	18	V
Corrente de saída	0	-	1.5	-	-	1.5	A
Eficiência	89	90.5	-	92	-	94	%
Precisão de saída ( $I_o=1.5A$ )	-	$\pm 3$	$\pm 4$	-	$\pm 2$	$\pm 3$	%
Preço	7.37			10.43			€

Tendo em conta que se pretende desenvolver uma solução de baixo custo e que a eficiência energética não é o mais importante para o presente documento, o conversor escolhido para o sistema de aquisição de dados é o OKI-785R-5/15.

De modo a dimensionar o circuito de alimentação foi tido em conta o esquema da placa Max32, sendo que o circuito de alimentação para o sistema de aquisição de dados é apresentado na figura 4.2. Para sinalização dos níveis de tensão disponíveis em cada sistema, foram adicionados os LED1, LED2 e LED3 que sinaliza os níveis de tensão de 12 V, 5 V e 3.3 V respetivamente. Os condensadores C11, C12, C13 e C14 foram inseridos como é recomendado pelos fabricantes. É de notar que existe um divisor resistivo, formado pelas as resistências R7 e R8, que tem como objetivo obter o valor de tensão VCMP. O valor de tensão referido é necessário para o circuito que será abordado mais à frente nesta secção, que tem como objetivo efetuar a comparação da alimentação do sistema entre a tensão proveniente da mota (12V) ou pelo conector MicroUSB (5V) utilizado para *debug*. Para reduzir interferências provenientes das correntes de retorno

que circulam no plano de GND, adicionaram-se as resistências R26 e R56 de modo a criar ramificações do mesmo.

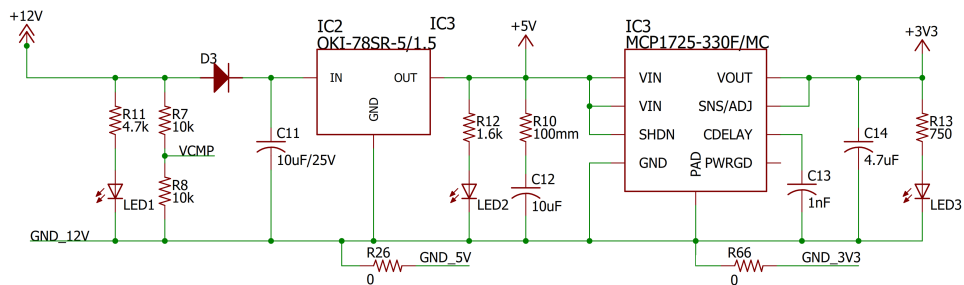


Figura 4.2: Reguladores de tensão para alimentação do sistema principal.

Para os módulos de sensores adicionais e para o sistema de recepção de dados foi utilizado outro circuito de alimentação. Como os dispositivos que são alimentados a 5 V requerem o consumo de corrente inferior a 800 mA, utilizou-se o circuito apresentado na figura 4.3. É de notar que a única diferença é o regulador de tensão de 12V para 5V, pois o CI NCP1117ST50T3G tem um custo bastante inferior em comparação com o regulador OKI-78SR-5/1.5.

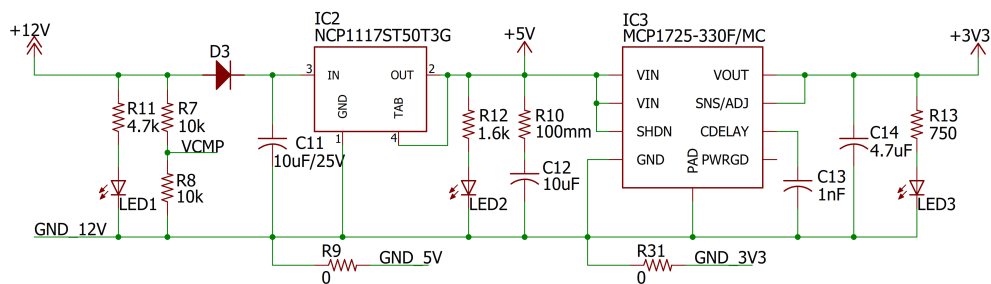


Figura 4.3: Reguladores de tensão para alimentação dos módulos e do sistema de recepção de dados.

A figura 4.4 apresenta o circuito de comutação da alimentação do sistema, que compara o valor a saída do regulador de 3.3V com o valor de tensão VCMP referido anteriormente. Este circuito tem como função garantir a alimentação de 5V no sistema, sendo esta proveniente da alimentação de 12V ou da ficha MicroUSB como indicado anteriormente. Isto permite que o microcontrolador não seja reiniciado sempre que está inserida uma fonte de tensão.

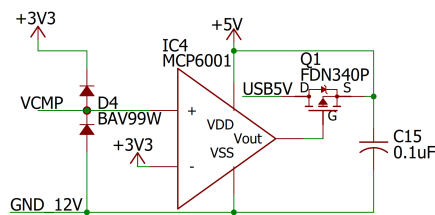


Figura 4.4: Circuito de comutação para alimentação de 5V.

### 4.1.2 Portas para programação e *debugging* do microcontrolador

Para ser possível fazer a atualização do código do microcontrolador existem duas possibilidades. A primeira possibilidade recorre à utilização de um programador externo, denominado de ICSP, tal como por exemplo o *chipKIT PGM* da *Digilent* ou *PICKit3* fornecido pela *Microchip*. Para ser possível programar através de uma ferramenta como a indicada anteriormente é preciso implementar um circuito de interface entre o microcontrolador e a ferramenta. A segunda possibilidade para atualizar o código do microcontrolador é através do uso de um *bootloader* que não necessita do uso de uma ferramenta de programação. O *bootloader*, na família PIC32, pode ser implementado através das seguintes interfaces [67]:

- Porta UART,
- Dispositivo USB baseado numa classe HID,
- USB *host* baseado numa classe MSD,
- Porta ethernet,
- Cartão SD.

Para todas as componentes foi implementada a interface UART como porta de recurso para programação das mesmas. Isto permite de futuro aos membros da equipa alterar o programa sem ser obrigatório o uso de uma ferramenta de programação.

De modo a facilitar o desenvolvimento de *firmware*, é feito o *debug* do código, que pode ser realizado através de uma ferramenta de programação ou por uma interface de comunicação. Ao fazer o *debug* com a ferramenta de programação é possível verificar linha a linha as ações tomadas pelo o código implementado. Para utilizar este método é necessário recorrer a um ambiente de desenvolvimento, sendo neste caso utilizado o *software* MPLab desenvolvido pela *Microchip*. Com a utilização da interface de comunicação para *debug*, é possível colocar o microcontrolador a enviar mensagens para um terminal, caso o código incluía instruções para esse fim.

#### 4.1.2.1 Interface para ICSP

O microcontrolador escolhido tem dois conjuntos de pinos para interligação com uma ferramenta de programação ICSP. Cada conjunto é formado por dois pinos sendo eles o pino PGECx e o pino PGEDx. Para o dimensionamento do circuito de programação, através de uma ferramenta ICSP, o fabricante do microcontrolador sugere as seguintes características [59]:

- Manter o comprimento das linhas o mais reduzido possível entre o conector ICSP e microcontrolador,
- Se existir a possibilidade do conector ICSP sofrer um evento de *ElectroStatic Discharge* (ESD), é recomendado a utilização de uma resistência em série que não exceda os 100  $\Omega$ ,
- Não é recomendado o uso resistências *pull-up*, díodos em série e condensadores para evitar interferências durante as comunicação entre o programador e o microcontrolador,

Tendo em conta as características apresentadas, o circuito implementado para a interface com a ferramenta ICSP está representado na figura 4.5.

Os pinos identificados como P32\_PGD e P32\_PGC no esquema estão ligados aos pinos PGED2 e PGEC2 do microcontrolador. Foi tomada esta decisão porque o outro conjunto de pinos está sobreposto com os canais analógicos AN0 e AN1. De modo a

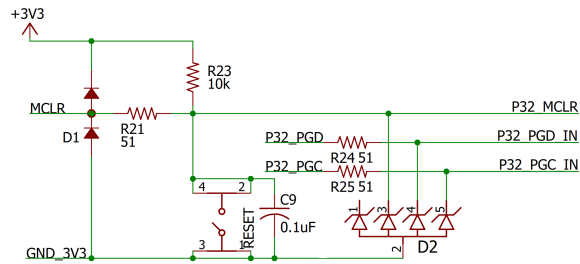


Figura 4.5: Circuito de programação, *debugging* e reinicialização do sistema.

proteger o microcontrolador de eventos de ESD foi adicionado o CI PESD5V0L4UG [68]. Como indicado anteriormente é recomendado pelo fabricante do microcontrolador o uso de resistências nos pinos de interface ICSP que não excedam o  $100\ \Omega$ , deste modo foram adicionadas as resistências R21, R24 e R25 com o valor de  $51\ \Omega$ . Para reinicializar o microcontrolador é utilizada a linha MCLR. A reinicialização pode ser efetuada pela ferramenta de programação ou então pelo utilizador através do botão BT1. O condensador C9 e a resistência R23 foram adicionadas como é recomendado pelo *datasheet* do microcontrolador. Por fim o díodo D1 tem como função proteger o pino MCLR do microcontrolador da descarga do condensador C9.

#### 4.1.2.2 UART1

O microcontrolador selecionado tem seis interfaces UART em que para cada uma delas são utilizados dois pinos UxRX e UxTX. Como referido na secção 3.2.6, o CI utilizado para fazer a interface entre a porta MicroUSB e UART do microcontrolador é o FT232RQ.

Tendo em conta as interfaces definidas na secção 3.4, é verificado que foi definido como porta de *debug* a interface UART1 do microcontrolador. É utilizada esta interface do microcontrolador porque os pinos associados à mesma, ainda se encontravam livres. Sendo este o primeiro CI implementado no sistema a utilizar o protocolo UART, foi selecionado o controlador UART1 do microcontrolador. Tendo em conta que a placa *chipKIT* também utiliza o mesmo CI como interface USB, o circuito da figura 4.6 foi baseado no esquema da placa de desenvolvimento e nas recomendações do *datasheet* do CI [62][54].

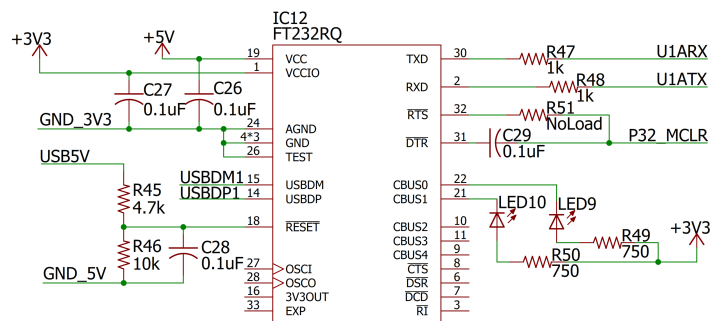


Figura 4.6: Circuito para interface UART1.

De acordo com o fabricante a alimentação do CI tem de estar dentro do intervalo de  $[4 ; 5.25]$  V, logo a tensão aplicada ao terminal VCC do mesmo é de 5V. Como a tensão de trabalho do microcontrolador é de 3.3 V os sinais digitais do CI têm de operar com



a mesma tensão, logo é aplicado ao pino VCCIO 3.3 V. De modo a evitar interferências nas alimentações descritas foram adicionados os condensadores C27 e C26. Para que o CI se encontre ativo quando está ligado um dispositivo externo à ficha MicroUSB, é utilizado o divisor de tensão formado por R45 e R46 de modo a que seja aplicada uma tensão de 3.3 V ao pino de RESET. O condensador C28 é utilizado para reduzir o nível de interferências no sinal aplicado ao pino de RESET. Os pinos USBDM1 e USBDP1 da ficha MicroUSB são ligados diretamente ao CI como recomendado pelo fabricante. Para a comunicação entre o CI e o microcontrolador é ligado o pino TXD ao pino U1ARX e o pino RXD ao U1ATX do microcontrolador. As resistências R47 e R48 foram adicionadas de modo a garantir que a corrente nunca seja superior a 3.3 mA. Como é utilizada esta interface do microcontrolador para o programar é necessário que este CI seja capaz de reinicializar o microcontrolador, deste modo é utilizado o pino DTR ligado ao pino P32\_MCLR. O condensador C29 é utilizado para evitar o *bouncing* na linha de reinicialização do microcontrolador. O LED9 e o LED10 tem como objetivo indicar a troca de dados entre o CI e o microcontrolador. Por fim as resistências R49 e R50 foram calculadas para a corrente de funcionamento dos LEDS escolhidos para a solução.

### 4.1.3 Oscilador externo

O microcontrolador tem a possibilidade de utilizar uma fonte de relógio externa ou interna, sendo que oscilador interno é pouco preciso e estável e também não é possível atingir a frequência máxima de funcionamento suportado pelo microcontrolador. Como todos os sistemas e módulos da solução vão utilizar comunicações, é necessário ter um sinal de relógio estável e preciso, o que levou ao uso de um oscilador externo como fonte de relógio para a solução.

Existem três tipos de osciladores externos indicados no *datasheet* do microcontrolador, sendo eles o oscilador de baixa potencia (LX), um cristal comum (XT) e um cristal de alta velocidade(HS). A utilização de osciladores de baixa velocidade requerem pouca potência, mas não permitem atingir a velocidade máxima de relógio que o microcontrolador suporta, enquanto os cristais de alta velocidade requerem mais potência devido ao aumento da frequência. Através da folha de cálculo disponibilizada pela *Microchip*, foi verificado qual era o tipo de oscilador que seria mais indicado para a solução tendo em conta que se pretendia uma frequência de relógio do sistema de 80 MHz [69].

Tendo em conta as recomendações da *Microchip*, o circuito associado ao oscilador está representado na figura 4.7. Como fonte de relógio foi escolhido um cristal de quartzo com a frequência de ressonância de 8MHz. O valor dos condensadores C24 e C25 foi atribuído consoante a Tabela 4.3.

Tabela 4.3: Seleção de condensadores conforme fonte de relógio [70].

Tipo de oscilador	Freq. Cristal	Condensadores
LX	32kHz	33pF
	200kHz	15pF
XT	200kHz	47-68pF
	1MHz	15pF
	4MHz	15pF
HS	4MHz	15pF
	8MHz	15-33pF
	20MHz	15-33pF

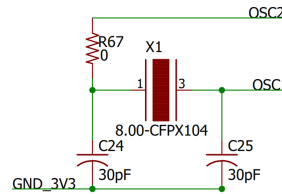


Figura 4.7: Circuito para o oscilador do sistema.

## 4.2 Módulo para medição de sensores adicionais

O objetivo da implementação do módulo para a aquisição de sensores adicionais como referido na secção 3.2.2, é agregar as grandezas associadas a cada roda, de modo a reduzir a quantidade e complexidade da cablagem associada ao sistema de aquisição de dados. Para adquirir os valores associados à pressão dos circuitos hidráulicos e à posição do êmbolo da suspensão, é necessário acondicionar o sinal de modo a que este possa ser lido pelo ADC embutido no microcontrolador. Para adquirir a temperatura dos pneus é utilizado um sensor de temperatura que comunica com o microcontrolador por I<sup>2</sup>C. Para enviar a informação para o sistema de aquisição de dados é adicionada uma interface CAN.

No decorrer desta secção são apresentadas as considerações tidas em conta de modo a garantir o funcionamento de todas as componentes referidas. No anexo A.1 encontra-se o esquema completo para este módulo.

### 4.2.1 Entradas analógicas

Através da análise do *datasheet* do sensor de pressão XMLG100D71, é verificado que a alimentação tem de estar dentro do intervalo de [11.4 ; 33] V e que sua saída varia entre [0 ; 10] V. Já para os potenciômetros lineares o valor máximo da sua saída analógica é igual ao seu valor de alimentação.

Para simplificar o *hardware* decidiu-se dimensionar o mesmo acondicionamento de sinal para os dois sensores, garantido que a sua saída analógica seja igual. Para tal é preciso alimentar o potenciômetro com uma tensão de 10V. Para garantir uma tensão de 10V estável, foi adicionado um regulador de tensão UA78L10ACD. Este regulador é capaz de suportar uma corrente de saída de 100 mA quando é alimentado até uma tensão máxima de entrada de 25V. O circuito associado ao regulador de 10V esta apresentado na figura 4.8.

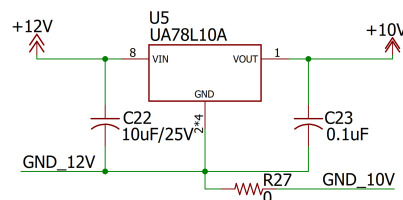


Figura 4.8: Circuito para o regulador de tensão de 10V.

Para interligar os sensores ao módulo, são utilizados conectores com proteção contra poeiras, contra infiltração de água e preparados para cabos blindados. A utilização de cabos blindados previne a interferência de sinais externos ao sinal analógico emitido pelo sensor. Para o dimensionamento do acondicionamento de sinal é requerido que o valor de tensão máximo colocado à entrada de cada canal do ADC do microcontrolador seja 3.3 V.

Também, como referido na secção 2.1, estas duas grandezas têm de ser adquiridas a uma frequência de 20 Hz, logo para que o sinal tenha maior imunidade contra interferências externas é adicionado um filtro passa-baixo. O teorema de Nyquist diz que a frequência de corte do filtro deve ser metade da frequência de amostragem. Logo para este caso foi definido que a frequência de corte dos filtros passa-baixo é de 10 Hz [71].

De modo a filtrar as perturbações do sinal analógico, foi aplicado um filtro passa-baixo com a arquitetura *Sallen-Key*. Este filtro tem a vantagem de obter um ganho positivo na sua saída, não sendo necessário o uso de uma tensão de alimentação negativa no amplificador operacional. De modo a simplificar o cálculo para os valores dos componentes passivos foi tido em consideração  $R=R2=R3=3.3\text{ k}\Omega$ ,  $C=C16=C17$  e  $f_c=10\text{ Hz}$ . Para obter o valor dos condensadores foi utilizada a equação 4.1 [72] [73].

$$f_c = \frac{1}{2\pi\sqrt{R2R3C16C17}} = \frac{1}{2\pi RC} \Leftrightarrow C = 4.82\text{ }\mu\text{F} \quad (4.1)$$

O circuito para a alimentação e o filtro passa-baixo associados ao sensor de pressão estão representados na figura 4.9. O circuito para o potenciómetro é igual ao do sensor de pressão. Na saída do filtro é obtida um valor  $V_o$  de  $[0 ; 10]\text{ V}$ , logo foi adicionado um amplificador operacional *rail to rail* alimentado a 10V. Como é necessário ter dois filtros passa-baixo, um para cada sensor analógico, foi utilizado o CI MC33202DG que é composto por dois amplificadores operacionais.

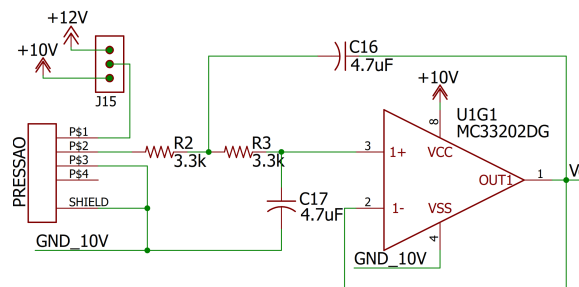


Figura 4.9: Circuito para filtro *Sallen-Key*.

Na figura 4.10 é apresentado o esquema associado ao andar de acondicionamento de sinal para a entrada do ADC. Para dimensionar o andar de acondicionamento de sinal, foi inicialmente verificado que o valor do ganho é de 0,33 como mostra a equação 4.2.

$$A_v = \frac{V_o}{V_i} = \frac{3.3}{10} \approx 0.33 \quad (4.2)$$

Tendo em conta que o valor obtido para o ganho é inferior a 1, é verificado que não é possível utilizar um amplificador não-inversor. O valor do ganho no amplificador inversor é obtido através da relação das resistências  $R4$  e  $R5$ . Para calcular o valor da resistências foi tido em conta que o valor de  $R4$  é igual a  $10\text{ k}\Omega$ , tendo sido obtido o valor de  $3.3\text{ k}\Omega$  para a resistência  $R5$  como mostra a equação 4.3.

$$|A_v| = \frac{R5}{R4} \Leftrightarrow R5 = |A_v| \times R4 = 10k \times 0,33 = 3.3\text{ k}\Omega \quad (4.3)$$

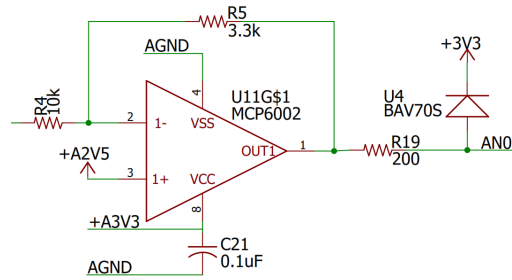


Figura 4.10: Circuito de amplificação para acondicionamento de sinal.

Ao ser utilizado um amplificador na configuração inversora, a saída do mesmo é negativa se a tensão à entrada é positiva. De modo a garantir que o valor analógico esteja entre  $[0 ; 3.3]$  V foi utilizada uma tensão de referência no terminal não inversor. Assumindo que o valor de tensão à entrada do amplificador é de 10V e que na sua saída é pretendido obter um valor de 0V, foi obtido o valor de 2,48V para a tensão de referência como mostra a equação 4.4.

$$V_{ref} = \frac{\left(\frac{R_5}{R_4}\right) V_i + V_o}{1 + \frac{R_5}{R_4}} \Leftrightarrow V_{ref} = \frac{\left(\frac{3,3k}{10k}\right) \times 10}{1 + \frac{3,3k}{10k}} = 2,48V \quad (4.4)$$

Para uma referência estável foi utilizado um regulador de tensão de 2.5V sendo ele o CI LM4120AIM5-2.5. Na figura 4.11 mostra o esquema implementado para o regulador de tensão.

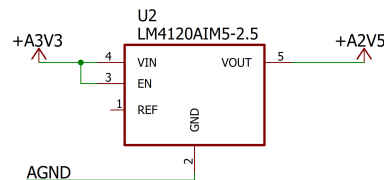


Figura 4.11: Regulador de tensão de 2,5V.

## 4.2.2 Sensor de temperatura

Como referido na secção 3.2.2.1, utiliza-se para medir a temperatura o sensor MLX60914, que comunica com o microcontrolador pelo protocolo I<sup>2</sup>C.

Os pinos da interface I<sup>2</sup>C1 estão ocupados pela interface USB-UART implementada para programação e *debug*, deste modo foi utilizada a interface I<sup>2</sup>C2 para comunicar com este sensor. Na implementação de um barramento I<sup>2</sup>C é obrigatória a presença de resistências *pull-up*, sendo recomendado que o valor das mesmas seja aproximado de 2.4kΩ [74].

Para interligar o sensor ao módulo é utilizado um conector com proteção contra poeiras, contra infiltração de água e habilitado ao uso de um cabo blindado. A utilização do cabo blindado minimiza as interferências externas nas linhas de comunicação.

O esquema aplicado para a interligação do sensor com o módulo é apresentado na figura 4.12.

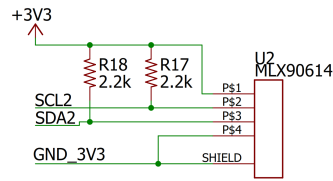


Figura 4.12: Circuito de ligação do sensor MLX60914.

### 4.2.3 *Transceiver* CAN

Ao implementar uma interface CAN é necessário escolher um dos dois controladores CAN do microcontrolador e utilizar um *transceiver* CAN, que neste caso é utilizado o CI MCP2561. São utilizados dois conectores que partilham a mesma interface CAN de modo a tornar a instalação do sistema na mota mais simples. Estes conectores também têm a função de receber a alimentação para o módulo. Para este módulo é utilizado o controlador CAN1 do microcontrolador, pois não existe nenhuma sobreposição com outro dispositivo neste módulo. Para implementar o *transceiver* anteriormente referido, foi tido em conta o esquema recomendado pelo fabricante . O circuito implementado está representado na figura 4.13 [75]. As resistências indicadas estão no esquemático de modo a que a PCB se encontre preparada para efetuar a terminação de cada barramento CAN. Isto é caso o nó não seja um extremo do barramento as resistências não são aplicadas.

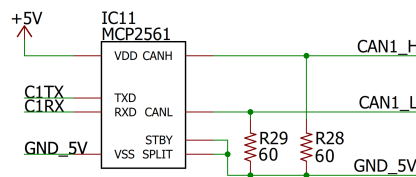


Figura 4.13: Circuito para o *transceiver* CAN.

## 4.3 Sistema de aquisição, gravação e transmissão de dados

De seguida serão apresentados os circuitos implementados para ligação de todas as partes constituintes do sistema de aquisição, gravação e transmissão de dados. No anexo B.1 encontra-se o esquema completo deste módulo.

### 4.3.1 *Transceivers* CAN

Este sistema está preparado para trabalhar com duas interfaces CAN, como indicado na secção 3.4. Para efetuar a leitura dos valores armazenados na ECU é utilizado o controlador CAN1 do microcontrolador, enquanto que para efetuar as leituras dos módulos dos sensores adicionais é utilizado o controlador CAN2. Só existe um conector ligado ao controlador CAN1, enquanto que o controlador CAN2 é partilhado por dois conectores, tal como foi aplicado no módulo de sensores adicionais. Na figura 4.14 estão apresentados os circuitos para os *tranceivers* associados aos controladores CAN do microcontrolador. A escolha das resistências para os circuitos foi realizada através das recomendações indicadas pelo fabricante [75]. As resistências indicadas estão no esquemático de modo a que a PCB se encontre preparada para efetuar a terminação de cada barramento CAN. Isto é caso o nó não seja um extremo do barramento as resistências não são aplicadas.

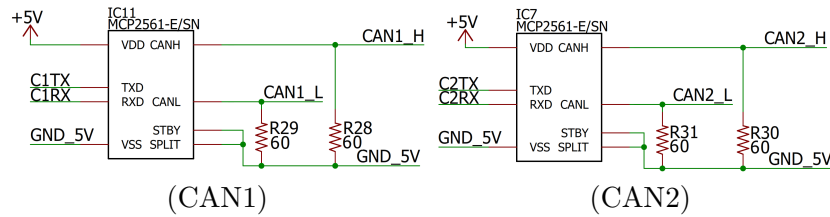


Figura 4.14: Circuitos para *transceivers* CAN.

### 4.3.2 Medição de acelerações

Como indicado na secção 3.2.3, o módulo escolhido para adquirir os ângulos de *Euler* e as acelerações é o MM7150, que comunica com o microcontrolador por I<sup>2</sup>C. Visto do microcontrolador este módulo utiliza a interface I<sup>2</sup>C1, duas saídas digitais para reiniciar ou ativar o módulo e uma entrada digital utilizada como interrupção.

O esquema aplicado para a interligação do módulo inercial com o microcontrolador é apresentado na figura 4.15. Foram adicionadas as resistências R18 e R38 porque o fabricante do módulo recomenda a utilização de resistências *pull-up* para os pinos SH\_WAKE e SH\_RESET. Como indicado na secção 4.2.2, também foram adicionadas as resistências de *pull-up* aos pinos de comunicação I<sup>2</sup>C. O pino de interrupção utilizado foi INT0 porque no decorrer do desenvolvimento este pino ainda se encontrava disponível [49].

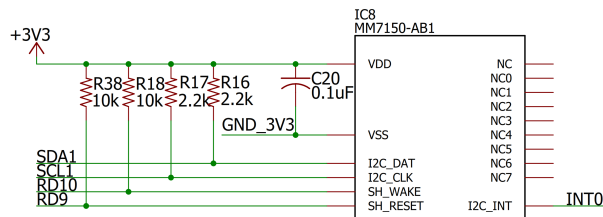


Figura 4.15: Circuito para módulo MM7150.

### 4.3.3 Módulo GPS

Como indicado na secção 3.2.4, para adquirir a localização da mota é utilizado o módulo GPS Quectel L80 que comunica com o microcontrolador através do protocolo UART. Este módulo visto do microcontrolador necessita de uma saída digital para o reiniciar, uma entrada digital para informação de antena detetada e uma interface UART para troca de informações.

O esquema associado à ligação do módulo GPS ao sistema de aquisição de dados está representado na figura 4.16. Através do *datasheet* do módulo é verificado que o mesmo só se encontra em funcionamento quando o pino RESET apresenta um valor lógico 1, sendo que o controlo do funcionamento deste módulo é realizado através do pino RA0 do microcontrolador configurado como saída. Deste modo é utilizada a resistência *pull-down* R14, para garantir que o módulo só se encontra em funcionamento quando a saída RA0 apresenta o valor lógico 1. Para realizar as comunicações entre os dispositivos é utilizada a interface UART5 do microcontrolador, a qual é ligada aos pinos RXD e TXD do módulo. O pino EX\_ANT do módulo está ligado diretamente a um conector SMA para possibilitar a ligação a uma antena externa. Quando o módulo deteta uma antena

externa ativa a sua saída AADET\_N, sendo esta ligada ao pino RD2 do microcontrolador configurado como entrada. Foi adicionada a resistência *pull-up* R15 ao sinal, como é recomendado pelo fabricante do módulo GPS.

De modo a evitar variações na alimentação do módulo foram adicionados dois condensadores de tântalo, sendo estes um pouco mais caros do que os condensadores cerâmicos mas são os que apresentam maior eficiência volumétrica, isto é maior capacidade num volume menor.

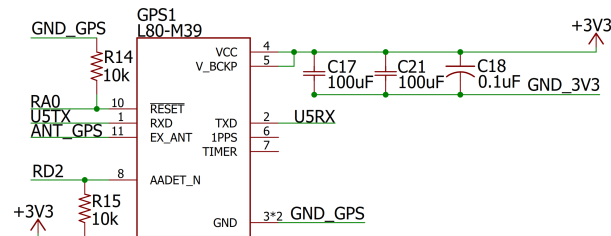


Figura 4.16: Circuito para módulo Quectel L80.

#### 4.3.4 Transmissão de dados para ecrã

A transmissão de dados para o ecrã é realizada com uma interface UART do microcontrolador através do CI FT232RQ referido na secção 3.2.6. Para esta funcionalidade do sistema é utilizada a interface UART2. O circuito implementado está representado na figura 4.17, sendo que as ligações e os valores dos componentes utilizados seguiram as mesmas indicações do circuito apresentado na secção 4.1.2.2. Neste caso o circuito só difere do referido anteriormente porque não é necessário que esta interface reinicialize o microcontrolador, sendo esta interface utilizada para a transmissão de dados e não para programação ou *debug*.

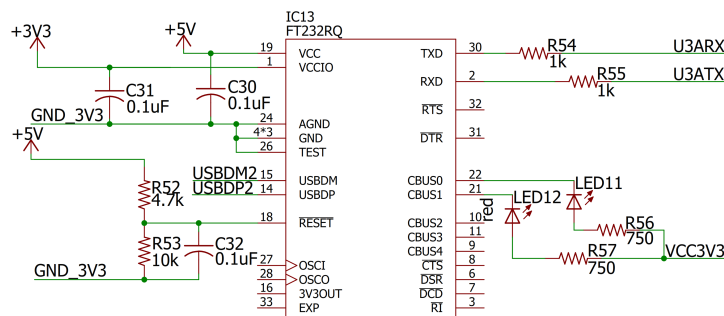


Figura 4.17: Circuito para comunicação com o ecrã.

Para efetuar a ligação ao ecrã é utilizado o mesmo tipo de conector que é usado para os *transceivers* CAN, pois só é necessário utilizar quatro condutores, sendo dois para comunicação e dois para alimentação do ecrã. Estes conectores possuem proteção contra água e poeiras.

#### 4.3.5 Armazenamento de dados em memória não volátil

Para o armazenamento de dados foi efetuada a escolha entre diferentes formatos de cartões SD. Como se pretende desenvolver uma solução de dimensões reduzidas foi optado pelo formato de um cartão microSD.

Como referido na secção 3.2.5, é utilizado um barramento SPI para efetuar a comunicação entre o dispositivo de memória e o microcontrolador. Para efetuar a comunicação entre os dois é utilizada a interface SPI2 do microcontrolador e uma saída digital do mesmo para fazer a seleção do dispositivo de memória.

Tendo em conta os pinos de um cartão microSD apresentados na figura 4.18, é possível verificar como são associados os pinos do cartão microSD aos pinos das interfaces do microcontrolador. Através do *datasheet* da *Kingston Technology* é possível verificar que a tensão de alimentação do cartão microSD tem de estar no intervalo de [2.7 ; 3.6] V [76].



Figura 4.18: Descrição dos pinos de um cartão microSD [77].

Para efetuar a ligação entre o módulo principal e o cartão microSD é utilizado um conector para tornar mais simples o acesso ao cartão por parte da equipa. Na figura 4.19 está apresentado o circuito para aceder à memória não volátil. A seleção do dispositivo de memória é realizado pelo pino RA3 do microcontrolador, quando este apresenta um valor lógico baixo. De modo a evitar *bouncing* do sinal de seleção do dispositivo, foi adicionada a resistência *pull-up* R19 ao pino RA3. Para ser possível verificar visualmente se o acesso ao cartão está a ser realizado, foi adicionado o LED4 da qual este só funciona quando o sinal de relógio do barramento apresenta um valor lógico alto e o dispositivo se encontra selecionado. Os sinais associados ao barramento SPI2 são ligados de acordo com o esquema de pinos do cartão microSD.

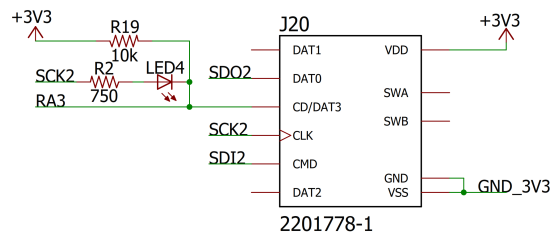


Figura 4.19: Circuito para cartão microSD.

#### 4.3.6 Transmissão de dados RF

Para efetuar a transmissão de dados sem fios é utilizado o *transceiver* RFM95PW-868, como referido na secção 3.2.7. De modo a operar com este dispositivo, a interface ao microcontrolador requer duas entradas digitais para interrupção, uma saída digital para reiniciar o dispositivo, uma ligação para um barramento SPI e uma saída digital para seleção do dispositivo durante a comunicação com o mesmo.



O esquema de implementação do *transceiver* RF no módulo principal do sistema de aquisição de dados está representado na figura 4.20. Para as comunicações entre o microcontrolador e o *transceiver* é utilizado o pino RA2 para selecionar dispositivo e o barramento SPI2, sendo este partilhado com o cartão microSD. O microcontrolador através do pino RC3 tem a capacidade de reinicializar o *transceiver*. O *transceiver* tem seis pinos digitais que podem ser configurados para diferentes eventos durante o funcionamento do mesmo. De modo a otimizar o funcionamento da solução foram associados dois pinos do módulo RF às interrupções externas INT1 e INT2 do microcontrolador. Para ligar uma antena foi adicionado um conector SMA ao pino ANT do *transceiver*. Para evitar variações na alimentação do *transceiver* foram adicionados três condensadores de tântalo, sendo eles C16, C22 e C23.

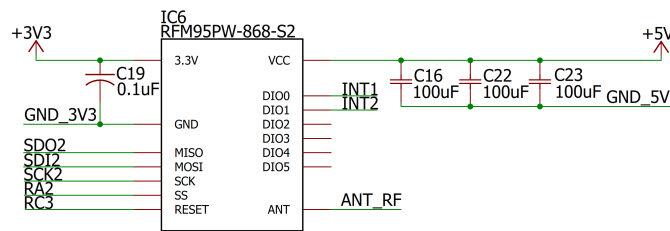


Figura 4.20: Circuito para o *transceiver* RFM95PW no sistema de aquisição.

## 4.4 Sistema para receção de dados

Todas as componentes do sistema para receção de dados foram abordadas anteriormente. De modo a facilitar o desenvolvimento das bibliotecas associadas ao módulo rádio é utilizado o mesmo esquema apresentado na secção 4.3.6. Para enviar as informações recebidas para um terminal série ou para uma aplicação é utilizada a interface de programação ou *debug* descrita na secção 4.1.2.2. No anexo C.1 encontra-se o esquema completo deste sistema.

## 4.5 Desenvolvimento de PCB's

Nos primeiros testes efetuados na mota foi verificado que durante o funcionamento da mesma, o sistema instalado era muito sensível a interferências e ocorriam perdas de comunicação entre o microcontrolador e os dispositivos. De modo a evitar esses problemas foram desenvolvidas PCB's para os módulos de sensores adicionais e para os sistemas de aquisição e receção de dados.

No dimensionamento das PCB's foi tido em conta a redução do tamanho e peso total da solução. Para o desenvolvimento de PCB's é preciso ter em consideração que existem duas tecnologias de montagem dos componentes na placa, *Through-Hole Technology* (THT) e *Surface Mount Technology* (SMT). Pelo THT, os *pads* associados aos componentes passam por todas as camadas existentes, condicionando a área para posicionamento dos componentes em várias camadas e o espaço para as pistas. Os componentes SMT além de utilizarem uma só camada para o seu posicionamento também têm dimensões mais reduzidas, tornando assim os sistemas mais compactos. Os componentes SMT radiam menos energia e também são menos vulneráveis a interferências externas [78].

Para o dimensionamento das PCB's foram tidas em consideração as seguintes condições:

- Minimizar o número de componentes THT,
- Utilização de duas camadas,
- Colocação de todos os componentes numa só face da placa,
- Minimizar o tamanho final da placa,
- Camada de cobre de 35  $\mu\text{m}$ .

O dimensionamento *Computer Aided Design* (CAD) da PCB foi efetuado com recurso ao *software* EAGLE. Tendo em conta a camada de cobre e as correntes que circulam em cada conexão, foi realizado o cálculo para a largura das pistas com recurso a uma calculadora online que utiliza as fórmulas da norma IPC-2221 [79].

De modo a proteger e facilitar a instalação de todo o sistema, foram desenvolvidas caixas para as PCB's. O dimensionamento CAD das caixas foi realizado no *software* SolidWorks2017, tendo em conta o tamanho total de cada PCB bem como a localização dos conectores. As caixas foram produzidas com recurso a uma impressora 3D que utiliza a tecnologia *Fused Filament Fabrication* (FFF).

#### 4.5.1 Módulo para aquisição de sensores adicionais

A PCB para o módulo de sensores adicionais tem dimensões externas de 65\*55 mm, sendo a versão final da PCB apresentada no anexo A.2. A PCB dimensionada tem um custo final de produção de 2,08€ por unidade sem componentes.

Na figura 4.21 é apresentada a PCB em três fases de construção. A primeira fase mostra a PCB recebida pelo fabricante sem componentes. A segunda fase apresenta a PCB com todos os componentes soldados, assim como a identificação de todos os conectores. Os conectores (a) e (b) são ligados ao sensor de pressão e ao potenciómetro linear, o conector (c) é para o sensor de temperatura, o (d) é a porta USB para programação e *debug* e por fim os conectores (e) e (f) são para a interligação do módulo ao sistema de aquisição de dados. A terceira e última fase representa o módulo concluído.

O custo associado à produção e montagem deste módulo é de 67,75€ por unidade, sendo a discriminação do preço por componente apresentada no anexo A.3.

Os desenhos técnicos associados à caixa de proteção estão representados no anexo A.4.

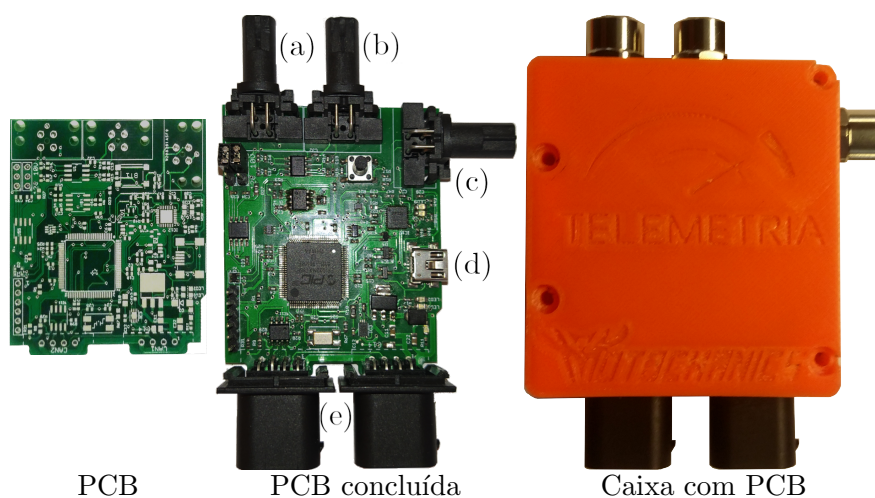


Figura 4.21: Construção do módulo para sensores adicionais.

### 4.5.2 Sistema de aquisição, gravação e transmissão de dados

A PCB para do sistema de aquisição, gravação e transmissão de dados tem dimensões externas de 90\*110 mm, sendo a sua disposição final apresentada no anexo B.2. A PCB dimensionada tem um custo de produção de 5,47€ por unidade sem componentes.

Na figura 4.22 são apresentadas as três fases de construção da PCB. A primeira fase mostra a PCB recebida pelo fabricante sem componentes. A segunda fase apresenta a PCB com todos os componentes soldados bem como a identificação de todos os conectores. As funções de cada conector são respetivamente:

- (a) - Alimentar e enviar informação para o ecrã,
- (b) e (c) - Alimentar e receber informação dos módulos para sensores adicionais,
- (d) - Obter alimentação para o sistema e realizar a comunicação com a ECU,
- (e) - Ranhura para cartão microSD,
- (f) - SMA para antena do transmissor RF,
- (g) - SMA para antena externa do módulo GPS,
- (h) - Porta USB para programação e *debug*.

A terceira e última fase representa o módulo concluído.

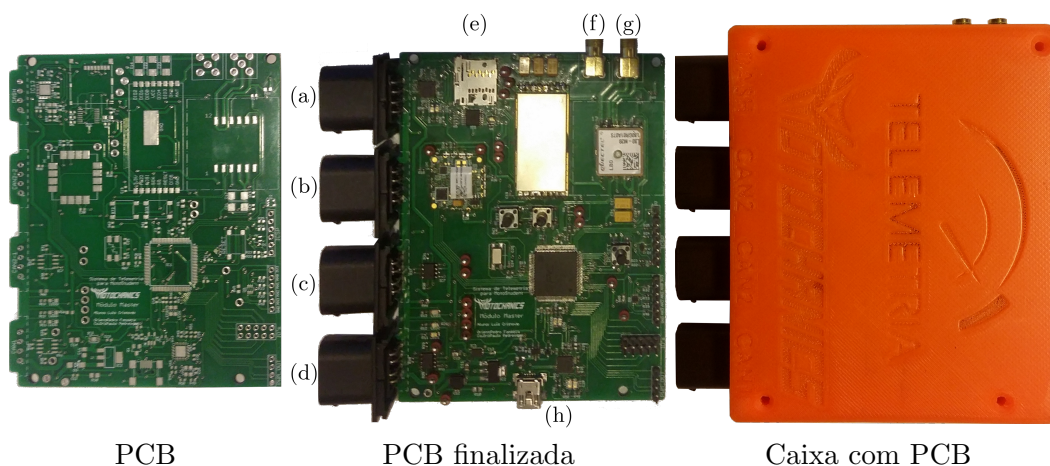


Figura 4.22: Construção do sistema para aquisição de dados.

O custo associado à produção e montagem desta PCB é de 129.91€ por unidade, sendo a discriminação do preço por componente apresentada no anexo B.3.

Os desenhos técnicos associados à caixa de proteção estão representados no anexo B.4.

### 4.5.3 Sistema para receção de dados

A PCB para o sistema de receção de dados tem as dimensões externas de 76\*65 mm, sendo a disposição final apresentada no anexo C.2. A PCB dimensionada tem um custo de produção de 2,42€ por unidade.

Na figura 4.23 são apresentadas as três fases da construção da PCB. A primeira fase mostra a PCB recebida pelo fabricante sem componentes. A segunda fase apresenta a PCB com todos os componentes soldados bem como a identificação de todos os conectores. O conector (a) é a porta USB para programação e *debug*, o conector SMA (b) é para a antena de receção de dados RF e o conector (c) é para uma alimentação externa. A terceira e última fase apresenta o módulo concluído.

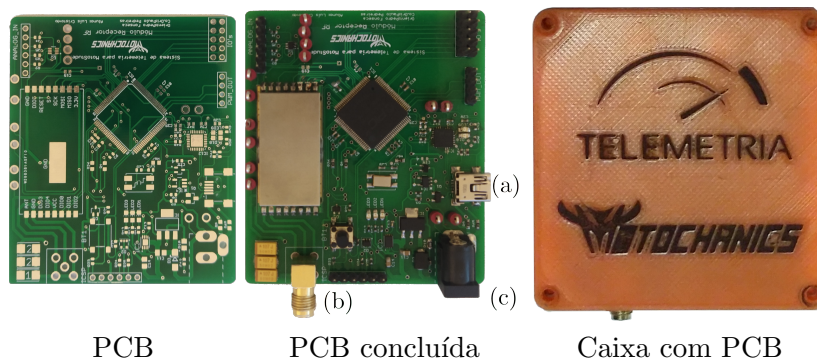


Figura 4.23: PCB do sistema de receção de dados.

O custo associado à produção deste sistema é de 41,86€ por unidade, cuja a descrição do preço por componente é apresentada no anexo C.3.

Os desenhos técnicos associados à caixa de proteção estão representados no anexo C.4.

## 4.6 Custo de construção da solução

Tendo em conta o que foi retratado nesta secção, verifica-se que o *hardware* desenvolvido tem a capacidade de cumprir com todos os requisitos apresentados na subsecção 2.5. É de referir que a solução foi desenvolvida de modo a reduzir o tamanho e peso final das componentes instaladas na mota.

Sendo um dos objetivos principais construir uma solução de baixo custo, realizou-se um estudo do custo associado a produção de uma solução completa. Neste estudo foram consideradas as PCB's anteriormente referidas, os sensores adicionais, os conectores externos, as antenas RF e GPS e o filamento PLA para a impressão das caixas de proteção. Na tabela 4.4 está apresentado o estudo dos custos associado à solução completa.

Tabela 4.4: Custos totais da solução.

Item	Descrição	Quant.	Preço /Unit.	Preço Total
PCB.MODULO.SENSORES	Módulo para sensores adicionais	2	67,750 €	135,500 €
PCB.SUB.EMISSOR	Sistema para aquisição, gravação e transmissão	1	129,910 €	129,910 €
PCB.SUB.RECETOR	Sistema de receção	1	41,860 €	41,860 €
XMLG100D71	Sensor de pressão; 0-100 bar; 0-10V	2	112,280 €	224,560 €
KPM 12-75	Potenciómetro linear 75mm	1	33,260 €	33,260 €
KPM 12-125	Potenciómetro linear 125mm	1	33,260 €	33,260 €
MLX90614ESF-AAA	Sensor de temperatura infravermelhos	2	12,710 €	25,420 €
T4113001041-000	Conector M12 Macho; 4 Pinos	6	4,030 €	24,180 €
2-967640-1	Conector AMP Femea; 4 Pinos	6	1,960 €	11,760 €
962885-1	Pino de cravar para ficha AMP	24	0,450 €	10,800 €
967067-1	Borracha de isolamento para pino ficha AMP	24	0,210 €	5,040 €
ANT-868-CW-HWR-SMA	Antena RF 868 MHz; 2,3dBi	2	12,980 €	25,960 €
GPS-ANT043	Antena GPS; 3dBi; 1,575GHz	1	5,630 €	5,630 €
05BQFIL028	Filamento PLA; 1,75mm; 1kg; Laranja	1	17,900 €	17,900 €
Total				725,04 €

Considerando o preço da produção desta solução ou da aquisição de uma solução de mercado, apresentadas na secção 2.4, verifica-se que a produção da mesma representa no pior dos casos cerca 32% do custo de aquisição de uma solução de mercado. É preciso ter em conta que não está contabilizado no custo da solução qualquer valor associado ao desenvolvimento da mesma. Isto é, ao ser adquirido um produto de mercado o valor

de compra já inclui o preço de desenvolvimento, da produção e margem de lucro para o fabricante.

Por outro lado, a solução desenvolvida possui uma característica que as soluções de mercado não apresentam, tal como a transmissão de dados em tempo real, sendo esta um fator diferenciador em relação aos produtos de mercado. Com os preços de custo inferiores a 32% do custo de aquisição de outra solução de mercado e com um fator diferenciador apresentado, esta solução poderá tornar-se num produto interessante para equipas de baixo orçamento, disponibilizando assim margem para suportar os custos associados à produção e ao desenvolvimento, obtendo ainda margem de lucro na venda.

A secção que se segue apresenta o desenvolvimento do *firmware* de modo a garantir o funcionamento do *hardware* retratado nesta secção tendo em consideração os requisitos mínimos da solução.



## Secção 5

# Desenvolvimento de *firmware*

Para a solução desenvolvida se tornar funcional é preciso que os microcontroladores sejam programados para uma ou mais tarefas. O desenvolvimento de código que permite o controlo de baixo nível do *hardware* dimensionado para um sistema é denominado de *firmware*. Esta secção tem por objetivo descrever todo o desenvolvimento de *firmware* para a solução.

Inicialmente são apresentadas as escolhas da plataforma utilizada para desenvolvimento de *firmware* e da linguagem de programação usada.

Para o desenvolvimento de *firmware* são apresentadas as características comuns entre eles, tais como, as configurações de relógio e a estrutura de dados preparada para todas as grandezas adquiridas.

A descrição do *firmware* desenvolvido para o módulo de medição de sensores adicionais do sistema de aquisição de dados começa por apresentar a descrição geral da função principal e da rotina de inicialização das interfaces utilizadas do microcontrolador. Para concluir a apresentação do *firmware* deste módulo é apresentada a rotina de interrupção implementada.

Também para o módulo principal do sistema de aquisição, gravação e transmissão de dados começa-se por apresentar a descrição geral da função principal e da rotina de inicialização das suas componentes. Por cada componente utilizada neste sistema são apresentadas as configurações efetuadas bem como a justificação das mesmas. De seguida é feita uma apresentação das interrupções utilizadas bem como a justificação das prioridades definidas, sendo que para cada rotina é descrito o seu fluxograma de funcionamento.

O *firmware* para o funcionamento do sistema de recepção de dados é apresentado no mesmo formato dos anteriores, sendo que é iniciado com a apresentação da função principal e da inicialização do sistema. De seguida são apresentadas as configurações das interfaces utilizadas do microcontrolador. Por fim é feita a descrição da rotina de interrupção implementada, sendo apresentado o seu fluxograma de funcionamento.

### 5.1 Ambiente de desenvolvimento

Para o desenvolvimento desta solução a *Microchip* disponibiliza o compilador *MPLAB XC32/32++*. Sendo este o único compilador disponibilizado pela *Microchip*, todos os programas para a solução foram desenvolvidos em C/C++ [80].

Como ambiente de desenvolvimento foi utilizado o *software* *MPLAB X Integrated Development Environment* (IDE) também disponibilizado pela *Microchip*. Este *software* disponibiliza várias ferramentas dedicadas ao desenvolvimento de programas para

os microcontroladores e microprocessadores da *Microchip*. As principais ferramentas disponibilizadas no *software* são:

- Editor de texto com ajudas de programação que também é utilizado como janela para *debug*,
- Gestão de projeto que fornece a integração entre o IDE e as ferramentas de linguagem,
- Ferramenta para *debug* que permite criar pontos de paragem no código e executar o mesmo passo a passo, com a possibilidade de consultar os valores das variáveis,
- Ambiente de simulação que permite simular programas em qualquer microcontrolador da *Microchip*,
- Possibilidade de adicionar ferramentas ao ambiente de desenvolvimento, como por exemplo o compilador *MPLAB XC32/32++*.

## 5.2 Configurações globais da solução

Para o desenvolvimento do *firmware* de toda a solução foram definidas configurações comuns entre os sistemas, tendo em conta a existência de vários microcontroladores com a mesma base de implementação, como referido na secção 4.1. As configurações comuns para toda a solução são associadas aos osciladores e à declaração da estrutura de dados preparada para guardar todas as grandezas adquiridas.

### 5.2.1 Configuração das fontes de relógio

Tendo em conta que as fontes de relógio das *interfaces* do microcontrolador dependem do oscilador principal é importante definir todos os microcontroladores da solução com a mesma configuração de relógio. Se os microcontroladores possuírem configurações diferentes entre eles, as suas *interfaces* tem de ser configuradas tendo em conta o valor de *System Clock* (SYSCLK) do microcontrolador.

De modo a garantir que as configurações de relógio são efetuadas antes do programa ser executado, estes registos são configurados com o comando ”#pragma config”. Como referido anteriormente na secção 4.1.3, o oscilador utilizado em todas as componentes da solução é um cristal de quartzo com uma frequência de ressonância de 8 MHz. Tendo em conta o cristal selecionado, as configurações relativas ao oscilador estão apresentadas no código fonte 5.1. As configurações foram obtidas com auxílio de uma folha de cálculo disponibilizada pela *Microchip* [69].

```
// DEVCFG1
#pragma config FNOSC = PRIPLL           // Oscillator Selection Bits (Primary Osc w/PLL (XT))
#pragma config FSOSCEN = OFF           // Secondary Oscillator Enable (Disabled)
#pragma config POSCMOD = XT            // Primary Oscillator Configuration (XT osc mode)
#pragma config OSCIOFNC = OFF         // CLKO Output Signal Active on the OSCO Pin
// (Disabled)
#pragma config FPBDIV = DIV_2         // Peripheral Clock Divisor (Sys_Clk/2)
#pragma config FWDTEN = OFF           // Watchdog Timer Enable (WDT Disabled)

// DEVCFG2
#pragma config FPLLIDIV = DIV_2       // PLL Input Divider (2x Divider)
#pragma config FPLLMUL = MUL_20      // PLL Multiplier (20x Multiplier)
#pragma config FPLL0DIV = DIV_1      // System PLL Out Clock Divider (PLL Divide by 1)
#pragma config UPLLIDIV = DIV_2      // USB PLL Input Divider (2x Divider)
#pragma config UPLEN = ON             // USB PLL Enable (Enabled)
```

Código fonte 5.1: Código para configuração dos osciladores.

No decorrer desta secção, a configuração das *interfaces* dos microcontroladores foram tidas em conta as configurações referidas anteriormente, da qual se obtêm as seguintes frequências de relógio:



- $\text{SYSCLK} = 80 \text{ MHz}$ ,
- *Peripheral Bus Clock* (PBCLK) = 40 MHz.

É preciso ter em consideração que o valor do SYSCLK foi configurado para a frequência máxima suportada pelos microcontroladores como indicado no *datasheet*. O relógio associado aos periféricos poderia ser configurado para a mesma frequência do sistema, mas de modo a evitar que os microcontroladores sobrequeçam o valor do PBCLK foi definido para metade do SYSCLK [59].

### 5.2.2 Estrutura de dados para grandezas

De modo a ser possível ter variáveis associadas às grandezas adquiridas organizadas de forma uniforme entre os sistemas da solução, foi definida uma estrutura comum de dados. A arquitetura da estrutura de dados influencia vários fatores da solução, sendo que os principais são:

- Velocidade do processamento de dados,
- Quantidade de dados a serem transmitidos,
- Velocidades de transmissão de dados,
- Distância entre os sistemas emissor e recetor.

Para efetuar a escolha da arquitetura da estrutura de dados foi feita uma comparação do tamanho e da organização da mesma, tendo em conta diferentes números de bits utilizados por cada variável. Na tabela 5.1 é apresentada a comparação entre duas arquiteturas possíveis.

No caso A foi definido que as variáveis seriam organizadas em múltiplos de 8 *bits*, enquanto que no caso B cada variável utilizaria um número de bits necessários para a representação da grandeza. Em termos de execução o caso A requer uma maior ocupação de memória de dados e uma maior quantidade de dados a serem transmitidos, tendo como vantagem o aumento da velocidade de processamento de dados. Implementando a estrutura do caso B verifica-se uma redução na ocupação de memória de dados, uma diminuição da quantidade de dados transmitidos, tendo como desvantagem uma velocidade de processamento de dados inferior.

Tendo em conta as diferenças entre os dois casos e que a transmissão de dados sem fios é uma limitação da solução, a escolha da estrutura de dados para a mesma é a representada pelo caso B, uma vez que a quantidade de dados a transmitir é menor.

## 5.3 Módulo para medição de sensores adicionais

No desenvolvimento de *firmware* para o módulo de sensores adicionais foi tido em conta a necessidade da utilização do ADC, de uma interface I<sup>2</sup>C, de um controlador CAN e de um temporizador do microcontrolador.

Este módulo está preparado para ler a temperatura dos pneus em dois pontos, a pressão de um travão, e a posição do êmbolo de uma suspensão. Os valores das grandezas são sempre do tipo, porque os microcontroladores não possuem co-processador aritmético não estando assim otimizados para funcionar com virgula flutuante. Para o caso do sensor de temperatura os valores podem variar no intervalo de [0 ; 200] °C, mas o valor enviado vai variar entre [0-2000] de modo a enviar uma casa decimal.

Tabela 5.1: Comparação entre duas estruturas de dados para variáveis adquiridas.

Variável		Intervalo	Unid.	Caso A N° bits	Caso B N° bits
Tempo	Minuto	[0-60]	unit.	8	6
	Segundo	[0-60]	unit.	8	6
	Milissegundos	[0-99]	unit.	8	7
Número de volta		[0-255]	unit.	8	8
Latitude	Graus	[0-90]	°	8	8
	Minutos	[0-60]	unit.	8	8
	Décimas de minutos	[0-9999]	unit.	16	14
	Norte/Sul	[1-2]	unit.	8	2
Longitude	Graus	[0-180]	°	8	8
	Minutos	[0-60]	unit.	8	8
	Décimas de minutos	[0-9999]	unit.	16	14
	Este/Oeste	[1-2]	unit.	8	2
Velocidade		[0-255]	km/h	8	8
Rotações do motor		[0-16384]	rpm	16	14
Mudança		[0-6]	unit.	8	3
Posição do acelerador		[0-100]	%	8	7
Pressão dos travões	Roda traseira	[0-1000]	bar	16	10
	Roda dianteira	[0-1000]	bar	16	10
Temperatura dos pneus	Traseiro	[0-2000]	°C	16	11
	Dianteiro	[0-2000]	°C	16	11
Suspensão	Traseira	[0-100]	mm	8	7
	Dianteira	[0-125]	mm	8	7
Tempo de ignição		[0-100]	%	8	7
Pressão de ar na admissão		[0-70]	kPa	8	7
Temperatura do ar da admissão		[0-255]	°C	8	8
Temperatura da refrigeração do motor		[0-255]	°C	8	8
Nível de combustível		[0-100]	%	8	7
Aceleração em linha	Sinal	[0-1]	unit.	8	1
	Valor	[0-2000]	g	16	11
Aceleração lateral	Sinal	[0-1]	unit.	8	1
	Valor	[0-2000]	g	16	11
Aceleração vertical	Sinal	[0-1]	unit.	8	1
	Valor	[0-2000]	g	16	11
<i>Roll</i>	Sinal	[0-1]	unit.	8	1
	Valor	[0-90]	°/s	8	7
<i>Pitch</i>	Sinal	[0-1]	unit.	8	1
	Valor	[0-180]	°/s	8	8
<i>Yaw</i>		[0-360]	°/s	16	9
Fim de estrutura		-	-	16	10
Total			bits	392	288
			bytes	49	36

Para o sensor de pressão o valor pode variar no intervalo [0 ; 100] bar, da qual é enviado um valor que varia entre [0-1000] de modo a enviar uma casa decimal. O valor da posição das suspensões varia entre os [0 ; 125] mm [81].

Tendo em conta que a solução foi desenvolvida para possuir dois módulos deste tipo, foi preciso definir qual seria o número de identificação associado a cada módulo. Isto é, os módulos vão partilhar o mesmo barramento CAN para enviar a informação para o sistema de aquisição de dados, logo cada módulo tem um identificador diferente para o sistema reconhecer de onde vêm os dados recebidos. Para definir o identificador de cada módulo não foi tida em consideração nenhuma razão em particular, sendo que os mesmos estão definidos da seguinte forma:

- 0x150 - Sensores associados à traseira,
- 0x151 - Sensores associados à dianteira.

Tendo em consideração o formato de mensagens *standard* apresentado na secção 2.3.1.4, a mensagem CAN enviada por estes módulos contém a informação das grandezas medidas da seguinte forma:

- SID - Identificador associado à posição do módulo na mota,
- DLC - Número de bytes contidos na mensagem,
- Data[0] - Temperatura 1 pneu (*Most Significant Byte* (MSB)),
- Data[1] - Temperatura 1 pneu (*Least Significant Byte* (LSB)),
- Data[2] - Temperatura 2 pneu (MSB),
- Data[3] - Temperatura 2 pneu (LSB),
- Data[4] - Pressão do travão (MSB),
- Data[5] - Pressão do travão (LSB),
- Data[6] - Posição da suspensão (MSB),
- Data[7] - Posição da suspensão (LSB).

No decorrer desta secção estão apresentadas as principais configurações deste módulo, bem como a rotina de interrupção associada ao funcionamento do mesmo.

### 5.3.1 Função principal e inicialização das interfaces

Quando o módulo é ligado começa por executar a função principal. Esta função tem dois objetivos, sendo eles invocar a função de inicialização das componentes utilizadas do microcontrolador e entrar num ciclo infinito. Dentro da função principal não é realizada mais nenhuma operação além das indicadas anteriormente, pois todas as ações associadas à aquisição de dados e à transmissão dos mesmos pelo barramento CAN são executadas a partir de uma rotina de interrupção.

Na figura 5.1 está apresentado o fluxograma associado à função que inicializa as componentes associadas ao microcontrolador.

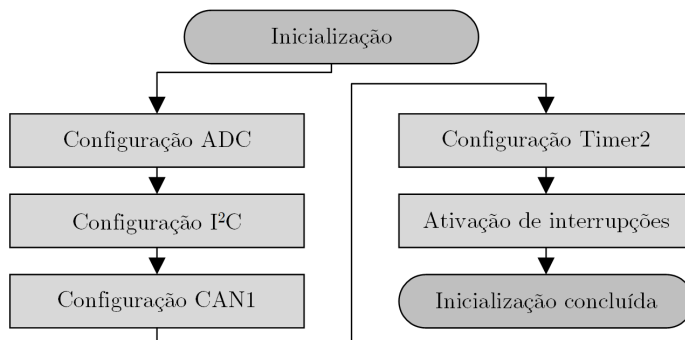


Figura 5.1: Fluxograma da rotina para inicialização das *interfaces*.

### 5.3.2 Configurações

Nesta secção são apresentadas as configurações efetuadas para os vários blocos funcionais associados a este módulo. Inicialmente são apresentadas as configurações associadas ao grupo responsável pela aquisição de dados, nomeadamente o módulo ADC e à interface I<sup>2</sup>C do microcontrolador. Para a transmissão de dados para o sistema de aquisição de dados, são apresentadas as configurações efetuadas ao controlador CAN1 do microcontrolador. Por fim é apresentada a configuração do temporizador utilizado para garantir as taxas de atualização requeridas para este módulo.

### 5.3.2.1 Aquisição de dados

No grupo da aquisição de dados do módulo para medição de sensores adicionais, estão associadas as interfaces utilizadas do microcontrolador para medir os sensores referidos na secção 3.4. Para medir a pressão de um travão e a posição do êmbolo de uma suspensão são utilizados dois canais do módulo ADC do microcontrolador. Para adquirir a temperatura do pneu é utilizada a interface I<sup>2</sup>C2 do microcontrolador.

De seguida serão descritas as configurações efetuadas para cada interface utilizada do microcontrolador neste módulo.

#### ADC

O módulo ADC do microcontrolador é um conversor de 10 bits com a capacidade de ler até 16 entradas analógicas. Neste caso, para o módulo de sensores adicionais são utilizados dois canais do ADC disponível no microcontrolador, sendo eles os canais AN0 e AN1 como é visível no anexo A.1.

Para configurar os dois canais analógicos é necessário configurar os registos associados aos pinos das entradas analógicas e ao ADC do microcontrolador. A configuração dos canais começa por definir os pinos associados aos canais AN0 e AN1 como entradas digitais através do registo TRISB. Para que o microcontrolador assuma estes dois pinos como entradas analógicas é necessário configurar o registo AD1PCFG associado a cada pino com o valor 0. Depois das entradas configuradas, passa-se à configuração dos parâmetros do ADC do microcontrolador através dos registos AD1CON1, AD1CON2 e AD1CON3.

O registo AD1CON1 permite configurar o formato do valor convertido para um inteiro de 16 bit (FORM), colocar o ADC no modo de auto conversão (SSRC), indicar se o *buffer* do ADC é substituído após uma nova conversão (CLRASAM) e habilitar o módulo ADC (ON).

O registo AD1CON2 permite definir o número de conversões realizadas para gerar uma interrupção (SMPI) e definir a origem dos valores de tensão de referência para o conversor (VCFG).

O último registo a ser configurado é AD1CON3 que permite definir a fonte de relógio para o conversor (ADRC), o tempo para adquirir cada amostra (ADCS) e o número de amostras por conversão (SAMC).

Considerando que a fonte de relógio do ADC é a frequência do PBCLK, é possível calcular o valor associado à duração da conversão de um canal analógico. Por recomendação do fabricante é preciso garantir que o tempo de aquisição ( $T_{AD}$ ) é superior a 83.33 ns, sendo que o valor para o registo ADCS obtido é igual a 1 como mostra a equação 5.1. Tendo em conta o valor obtido, verifica-se que o valor final para  $T_{AD}$  é de 100 ns, como mostra a equação 5.2 [82].

$$T_{AD} = \frac{2}{PBCLK} \cdot (ADCS + 1) \Leftrightarrow ADCS = \frac{83.33 * 10^{-9}}{\frac{2}{40 * 10^6}} - 1 = 0.666 \simeq 1 \quad (5.1)$$

$$T_{AD} = \frac{2}{PBCLK} \cdot (ADCS + 1) = \frac{2}{40 \cdot 10^6} \cdot (1 + 1) = 100 \text{ ns} \quad (5.2)$$

O tempo para obter o valor digital de um canal analógico depende do valor de ( $T_{AD}$ ), do número de ciclos de amostra (SAMC) e do número de conversões realizadas até ser despertada a interrupção. De modo a que o condensador de *sampling* carregue até um valor próximo do valor do sensor por cada conversão o sinal é adquirido durante

5 ( $T_{AD}$ ), configurado através do registo SAMC. Considerando o valor  $T_{AD}$  calculado anteriormente, que é obtido cada sinal após 5 ciclos de amostra, que a interrupção é gerada após a ADC realizar duas conversões e que o tempo de conversão é de  $(12 \cdot T_{AD})$ , o tempo que o ADC demora a obter as duas conversões é de  $2.2 \mu\text{s}$ , como mostra a equação 5.3.

$$T_{ADC} = ((T_{AD} \cdot SAMC) \cdot (SMPI + 1)) + (12 \cdot T_{AD}) = (100 \text{ ns} \cdot 5) \cdot (2) + (12 \cdot 100 \text{ ns}) = 2.2 \mu\text{s} \quad (5.3)$$

A configuração dos registos associados ao funcionamento do módulo ADC está apresentada no código fonte 5.2. O módulo depois de ser configurado fica habilitado.

```
//CONFIG AN0-AN1
TRISBbits.TRISB0 = 1; //Digital Input RB0
TRISBbits.TRISB1 = 1; //Digital Input RB1
AD1PCFGbits.PCFG0 = 0; //Active RB0 as analog input
AD1PCFGbits.PCFG1 = 0; //Active RB1 as analog input
AD1CON1bits.CLRASAM = 0; //Buffer is overwritten with new conversion
AD1CON1bits.FORM = 0; //Integer 16-bit
AD1CON1bits.SSRC = 7; //Auto Convert
AD1CON2bits.SMPI = 1; //Interrupt is gerated after convert 2 samples
AD1CON2bits.VCFG = 0; //Voltage reference (AVdd-AVss)V
AD1CON3bits.ADRC = 0; //ADC use PBCLK as clock
AD1CON3bits.SAMC = 5; //Number of samples for conversion
AD1CON3bits.ADCS = 1; //With this value TAD>83.33ns
AD1CON1bits.ON = 1;
```

Código fonte 5.2: Código para configuração da *interface* I<sup>2</sup>C.

## Pirómetro

No módulo para medição de sensores adicionais é utilizada a interface I<sup>2</sup>C2 como pode ser verificado no anexo A.1. Sendo o pirómetro MLX60914 o único sensor a ser ligado a esta interface, de modo a definir as configurações da mesma, verificou-se no *datasheet* que a velocidade de comunicação pode ser definida entre [10 ; 100] kHz, o endereçamento do dispositivo é composto por 7 *bits* e trabalha com especificações *System Management Bus* (SMBus). Para configurar a interface é necessário configurar dois registos do microcontrolador, sendo eles I2C2CON e I2C2BRG.

Tendo em conta as especificações do pirómetro, o registo I2C2CON permite configurar os níveis lógicos das entradas para que estes sejam compatíveis com as especificações de um barramento SMBus (SMEN), habilitar a receção de dados (RCEN) e ligar a interface (ON). De modo a tornar a aquisição de dados mais rápida a velocidade definida para esta interface é de 100 kHz. Para configurar a velocidade de comunicação para esta interface ( $F_{SCK}$ ) é preciso definir o valor do registo I2C2BRG. Perante o *datasheet* do microcontrolador o registo I2C2BRG é configurado com o valor de 197, como mostra a equação 5.4 [74].

$$I2C2BRG = \frac{PBCLK}{2 \cdot FSCK} - 1 - \frac{PBCLK \cdot T_{PGOB}}{2} = \frac{40 \text{ MHz}}{2 \cdot 100 \text{ kHz}} - 1 - \frac{40 \text{ MHz} \cdot 130 \text{ ns}}{2} = 197 \quad (5.4)$$

Para trabalhar com o pirómetro também é necessário definir os parâmetros associados ao modo de comunicação (PWMCTRL) e a configuração das partes analógicas e digitais do sensor (CONFIGREG). De modo a configurar o modo de comunicação é preciso desabilitar a saída do sinal em modo PWM (PWMEN) e definir o pino do sensor SDA como *open-drain* (SDAOD), porque na comunicação por I<sup>2</sup>C o pino de dados tem de estar sempre em *open-drain*.

Para a configuração do sensor associada à aquisição de sinal é necessário definir o número de pontos adquiridos pelo sensor, quais as temperaturas guardadas na memória

RAM do sensor e os filtros digitais *Finite Impulse Response* (FIR) e *Infinite Impulse Response* (IIR). Os filtros digitais FIR e IIR foram configurados conforme as recomendações do fabricante. O sensor também foi configurado para efetuar a leitura da temperatura em dois pontos do pneu (DUAL) e guardar na sua memória as temperaturas associadas a esses pontos (OBJ) [83].

A configuração dos registos associados à interface I<sup>2</sup>C e ao funcionamento do pirómetro está apresentada no código fonte 5.3.

```
//INTERFACE CONFIG - I2C2
I2C2CONbits.SMEN=1; //Enable SMBus input logic
I2C2CONbits.RCEN=1; //Enable receive mode
I2C2CONbits.ON=1; //Turn ON I2C2 module
I2C2BRG=197; //Set BaudRate I2C
//DEVICE CONFIG - MLX60914
PWMCTRL.bt=0; //Reset variable PWMCTRL
PWMCTRL.PWMEN=0; //Disable PWM mode
PWMCTRL.SDAOD=0; //Configure SDA pin as Open Drain
write16(MLX90614.PWMCTRL, PWMCTRL.bt); //Send variable PWMCTRL to device
ConfigReg.bt=0; //Reset variable ConfigReg
ConfigReg.IIR=4; //IIR filter (100%) a1=1,b1=0
ConfigReg.Obj=3; //Measure Tobj1 and Tobj2
ConfigReg.Dual=1; //Dual mode value=1
ConfigReg.FIR=4; //FIR filter = 128
write16(MLX90614.CONFIG, ConfigReg.bt); //Send variable ConfigReg to device
```

Código fonte 5.3: Código para configuração da *interface* I2C2 e do pirómetro.

### 5.3.2.2 Transmissão de dados

No módulo de sensores adicionais os dados adquiridos são transmitidos para o sistema de aquisição de dados através do controlador CAN1 do microcontrolador. Este barramento só é utilizado para transmitir os dados adquiridos para sistema de aquisição, logo o controlador CAN está configurado como um nó transmissor. Para a correta configuração de um controlador CAN no microcontrolador é necessário efetuar sete passos:

- 1°- Colocar o controlador CAN em modo de configuração,
- 2°- Efetuar as configurações relativas ao modo de funcionamento e da velocidade de comunicação,
- 3°- Associar a memória *first in first out* (FIFO) ao controlador,
- 4°- Definir os canais do controlador,
- 5°- Definir o filtro e a máscara para associar a cada canal utilizado,
- 6°- Configurar as interrupções associadas ao controlador,
- 7°- Colocar o controlador em modo normal.

De modo a facilitar o desenvolvimento de *firmware* para este controlador é utilizada a biblioteca disponibilizada pela *Microchip*. Tendo em conta que não é requerida uma velocidade específica para o barramento CAN dedicado à aquisição de sensores adicionais, de modo a facilitar a configuração dos vários controladores CAN de todo o sistema, a velocidade utilizada nos barramentos CAN é igual à velocidade do barramento CAN da ECU [84].

Para descobrir a velocidade do barramento CAN da ECU, foi utilizado um adaptador OBD, como o apresentado na secção 2.1. Ao efetuar a ligação com a ECU através do adaptador OBD, sendo desconhecidas as configurações do barramento CAN da ECU, o *software TouchScan* efetua várias tentativas de configuração da comunicação CAN. Quando é efetuada a comunicação com a ECU, através do *software* é possível ver que a velocidade de comunicação é de 500 kbit/s.

As configurações efetuadas para ajustar o relógio associado ao módulo CAN foram baseadas no exemplo disponibilizado pela *Microchip*. Após a velocidade do módulo estar configurada é preciso associar a memória aos canais do controlador CAN. Neste caso só

é utilizado o canal zero do controlador como transmissor, em que a memória associada a este canal está preparada para guardar até 8 mensagens CAN. Como não é realizado qualquer tipo de filtragem das mensagens recebidas por este controlador CAN, porque não está definido nenhum canal para a receção de dados, não é necessário definir nenhum filtro nem nenhuma máscara.

Também não são associadas interrupções a este controlador CAN, logo estas não são configuradas. A configuração deste módulo CAN é efetuada através do código fonte 5.4 [20].

```
// Step 1: Turn on CAN module and switch it to configuration mode
CANEnableModule(CAN1, TRUE);
CANSetOperatingMode(CAN1, CAN_CONFIGURATION);
while(CANGetOperatingMode(CAN1) != CAN_CONFIGURATION);

// Step 2: Configure the CAN Module Clock
canBitConfig.syncJumpWidth      = CAN_BIT_2TQ;
canBitConfig.propagationSegTq    = CAN_BIT_3TQ;
canBitConfig.phaseSeg1Tq        = CAN_BIT_3TQ;
canBitConfig.phaseSeg2Tq        = CAN_BIT_3TQ;
canBitConfig.sample3Time        = TRUE;
canBitConfig.phaseSeg2TimeSelect = TRUE;
CANSetSpeed(CAN1, &canBitConfig, SYSTEM_FREQ, CAN_BUS_SPEED);

// Step 3: Assign the buffer area to the CAN module.
CANAssignMemoryBuffer(CAN1, CAN1MessageFifoArea, (1 * 8 * 16));

// Step 4: Configure channel 0 for TX and size of 8 message buffers.
CANConfigureChannelForTx(CAN1, CAN_CHANNEL0, 8, CAN_TX_RTR_DISABLED, CAN_LOW_MEDIUM_PRIORITY);

// Step 6: Disable interrupt to CAN1.
INTEnable(INT_CAN1, INT_DISABLED);

// Step 7: Switch the CAN mode.
CANSetOperatingMode(CAN1, CAN_NORMAL_OPERATION);
while(CANGetOperatingMode(CAN1) != CAN_NORMAL_OPERATION);
```

Código fonte 5.4: Código para configuração do controlador CAN1.

### 5.3.2.3 Temporizador

Neste módulo é importante manter a aquisição e transmissão de dados à taxa de 20 Hz, sendo esta a frequência de aquisição para os sensores adquiridos pelo módulo. De modo a garantir uma frequência fixa, definiu-se um temporizador do microcontrolador com a função de despertar uma interrupção à frequência desejada. No módulo de sensores adicionais foi adicionado um temporizador de modo a que uma rotina seja executada à frequência de 20 Hz, sendo este o valor requerido para a aquisição das medidas efetuadas pelos sensores adicionais. O temporizador configurado neste módulo está associado ao (TMR2) do microcontrolador. Para o correto funcionamento do temporizador é necessário configurar o registo (T2CON), definir o valor do período (PR2) e configurar a interrupção associada ao temporizador.

Para definir a frequência ( $f_{Timer2}$ ) igual a 20 Hz é preciso calcular o valor para o registo PR2, de modo a que o mesmo não seja superior ao valor 65535, devido ao contador associado ao temporizador ser um valor composto por 16 *bits*. Para que este valor não seja ultrapassado é ajustado o valor do *Prescale* (TCKPS), de modo a que este seja o mais pequeno possível para reduzir o erro associado ao contador do temporizador. Tendo em conta a frequência anteriormente referida e considerando um *Prescale* de 1:32, o valor de PR2 é igual a 62500 como mostra a equação 5.5.

$$PR2 = \frac{PBCLK}{f_{Timer2} \cdot Prescaler} = \frac{40 \text{ MHz}}{20 \text{ Hz} \cdot 32} = 62500 \quad (5.5)$$

De modo a ser possível executar uma rotina quando o contador do temporizador atinge o valor de PR2, é necessário configurar a interrupção associada ao mesmo. A configuração da interrupção tem de definir a prioridade e sub-prioridade da mesma através

do registo (IPC2), habilitar a interrupção (IEC0) e limpar a *flag* associada ao temporizador 2 (IFS0). Sendo esta a única interrupção a ser executada no módulo de aquisição de sensores adicionais, a prioridade e a sub-prioridade são definidas com o valor 1. Neste caso não existiria qualquer diferença na execução do código caso os valores das prioridades fossem diferente dos valores referidos, pois neste módulo só é executada uma rotina de interrupção. A configuração efetuada para o temporizador e a sua interrupção está apresentado no código fonte 5.5.

```
//CONFIG TIMER2 ----- Prescaler 1:32; PR2 Preload = 62500;
T2CONbits.ON = 0; //Disable Timer2
T2CONbits.TCKPS2=1; //Configure Prescaler to 1:32 (bit 2)
T2CONbits.TCKPS1=0; //Configure Prescaler to 1:32 (bit 1)
T2CONbits.TCKPS0=1; //Configure Prescaler to 1:32 (bit 0)
PR2 = 62500; //Set period value
TMR2 = 0; //Reset counter value
IFS0bits.T2IF = 0; //Reset interrupt flag
IEC0bits.T2IE = 1; //Enable Timer2 interrupt
IPC2bits.T2IP = 1; //Set priority level 1
IPC2bits.T2IS = 1; //Set sub-priority level 1
T2CONbits.ON = 1; //Enable Timer2
```

Código fonte 5.5: Código para configuração do *Timer2*.

### 5.3.3 Rotina de interrupção

Este módulo tem a função de adquirir e transmitir os dados associados ao travão, à temperatura do pneu e à posição do êmbolo da suspensão para o sistema de aquisição. Para garantir que os dados são enviados à mesma frequência é utilizada a interrupção gerada pelo temporizador. A *flag* associada ao temporizador é ativada quando a contagem do mesmo atinge o valor do período definido, despertando assim rotina de interrupção.

A rotina de interrupção começa por efetuar a leitura das temperaturas guardadas na memória RAM do pirómetro associadas aos dois pontos de leitura do mesmo. De seguida são efetuadas as leituras, da entrada analógica AN0 associada ao sensor de pressão e da entrada AN1 associada à posição do êmbolo da suspensão. Depois dos valores serem adquiridos é construída a mensagem CAN com o formato indicado na secção 5.3. Os valores associados às temperaturas do pneu e à pressão dos travão são multiplicados por dez, de modo a enviar um número inteiro, por exemplo para uma pressão de 52.4 bar o valor enviado é igual a 524, para evitar operações que requerem o uso da virgula flutuante. Após a mensagem estar definida é enviada pela interface CAN1 do módulo. Por fim a *flag* associada à interrupção do temporizador é desabilitada. O fluxograma associado a interrupção do *Timer2* está apresentada na figura 5.2.

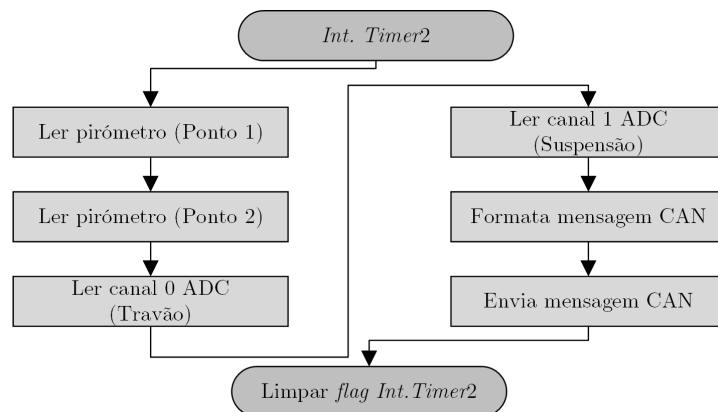


Figura 5.2: Fluxograma da rotina de interrupção do *Timer2*.



## 5.4 Sistema de aquisição, gravação e transmissão de dados

No desenvolvimento de *firmware* do sistema de aquisição, gravação e transmissão de dados foi tido em conta a necessidade da utilização de dois controladores CAN, de um barramento I<sup>2</sup>C, de duas interfaces UART, de um barramento SPI e de dois temporizadores.

Este sistema está preparado para efetuar pedidos à ECU da mota, receber os dados adquiridos pelos módulos de sensores adicionais, efetuar a leitura dos valores adquiridos pelo módulo inercial, adquirir a localização absoluta da mota no globo terrestre, transmitir dados para o ecrã da mota, guardar os valores num cartão microSD e transmiti-los por uma interface RF na banda ISM dos 868 MHz.

Esta secção começa por abordar a função principal e a sequência da configuração de todas as componentes do sistema de aquisição. De seguida são apresentadas as configurações efetuadas para cada componente utilizada, tendo em consideração os dispositivos selecionados. Por fim serão descritas as rotinas de interrupção e apresentados os respetivos diagramas de blocos associados a este sistema.

### 5.4.1 Função principal e inicialização das interfaces

O sistema ao ser ligado começa por executar a função principal. Esta função tem dois objetivos, sendo eles invocar a função de inicialização das interfaces utilizadas do microcontrolador e entrar num ciclo infinito. Dentro da função *main* não é realizada mais nenhuma operação além das indicadas anteriormente, pois todas as ações associadas ao sistema são executadas a partir de rotinas de interrupção.

Na rotina de inicialização são configuradas as interfaces e os dispositivos associados ao sistema, pela sequência apresentada no fluxograma da figura 5.3.

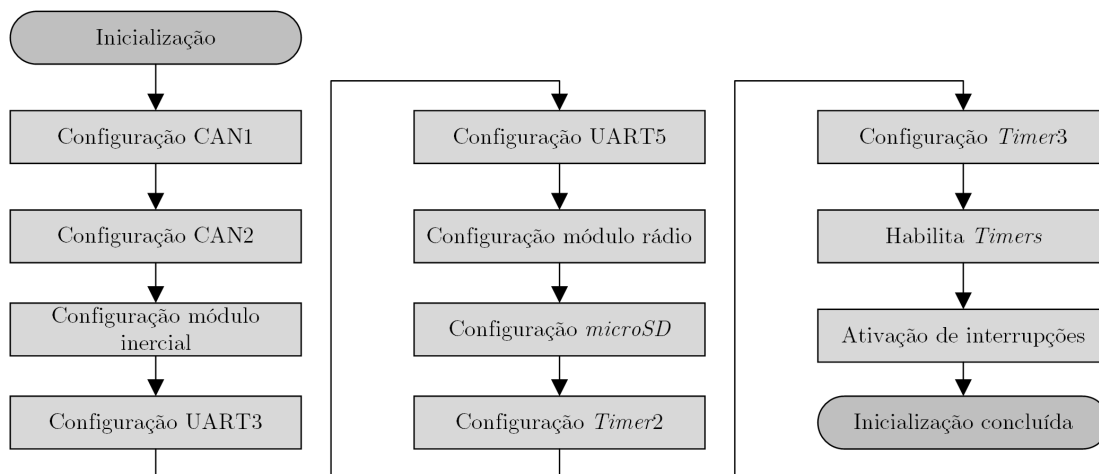


Figura 5.3: Fluxograma da rotina para inicialização das interfaces.

### 5.4.2 Configurações

Tendo em conta os requisitos que têm de ser cumpridos pelo sistema de aquisição, gravação e transmissão de dados, nesta secção serão apresentadas as configurações aplicadas às interfaces e aos dispositivos. As configurações foram divididas em cinco grupos, sendo eles a aquisição, a gravação e a transmissão dos dados recolhidos, os temporizado-

res utilizados e as interrupções do sistema. As interfaces do microcontrolador utilizadas bem como a sua respetiva função são:

- CAN1 para efetuar a comunicação com a ECU,
- CAN2 para receber os dados recolhidos pelos módulos de sensores adicionais,
- I<sup>2</sup>C1 comunicar com o módulo inercial,
- UART3 interface para transmissão de dados para o ecrã,
- UART5 para comunicar com módulo GPS,
- SPI2 estabelece a comunicação com cartão microSD e *transceiver* rádio,
- Temporizadores 2 e 3 permitem executar rotinas periodicamente.

#### 5.4.2.1 Aquisição de dados

No grupo da aquisição de dados do sistema estão inseridos os blocos funcionais associados à leitura da ECU, à receção de dados dos módulos de sensores adicionais, à leitura do módulo inercial e à receção dos dados emitidos pelo módulo GPS. Por cada bloco funcional é feita uma descrição da configuração da interface do microcontrolador bem como do dispositivo associado. Por cada configuração é apresentado o respetivo código de fonte.

#### Leitura da ECU

Algumas das grandezas requeridas pela equipa são obtidas da ECU da mota, sendo que para efetuar a comunicação com a mesma é utilizado o controlador CAN1 do microcontrolador. De modo a configurar o controlador CAN foi realizado um estudo de como é efetuado o diagnóstico da ECU através do protocolo OBD, para configurar os canais associados ao controlador, bem como os filtros e as máscaras adicionadas para filtrar as mensagens que circulam no barramento.

Para ser obtida informação da ECU é necessário efetuar um pedido à mesma para que seja devolvida a grandeza que se deseja obter. Isto é, para efetuar um pedido à ECU é enviada uma mensagem com o identificador 0x7DF. A ECU depois de processar o pedido envia uma mensagem para o barramento com a informação requerida, em que mensagem de resposta tem o identificador 0x7E8.

O pedido é composto pelo identificador, pelo número de *bytes* que a mensagem contém (DLC) e por 8 *bytes* de dados. Nos *bytes* de dados enviados na mensagem, o primeiro *byte* indica à ECU o número de *bytes* que contêm dados, o segundo *byte* especifica o modo de diagnóstico e o terceiro *byte* indica qual é o *Parameter Identification (in messageing protocol)* (PID) que pretende ser obtido. Já a resposta emitida pela ECU difere no identificador, como já foi referido, e na organização dos dados. O primeiro *byte* também indica o número de *bytes* que contêm dados, o segundo *byte* indica que é uma resposta do modo de diagnóstico, o terceiro *byte* indica qual é o PID consultado e os restantes *bytes* contêm os dados associados ao parâmetro consultado. Na figura 5.4 é apresentado um exemplo do formato de uma mensagem CAN para efetuar um pedido à ECU e a respetiva resposta enviada pela mesma [85].

As configurações efetuadas ao controlador CAN1 começam por associar uma variável à memória *First In First Out* (FIFO) do controlador, bem como por definir o número de mensagens guardadas por cada canal configurado. Neste caso a variável está preparada para alocar até oito mensagens por canal. De seguida são configurados os canais, sendo que o canal 0 será para a transmissão e o canal 1 para a receção de mensagens. O canal 0 é utilizado para enviar os pedidos à ECU, e o canal 1 tem a função de receber as mensagens enviadas pela ECU.

## Pedido enviado para a ECU

SID=0x7DF	DLC=0x08	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
		N° Bytes=0x02	Modo=0x01	PID	0x00	0x00	0x00	0x00	0x00

## Resposta enviada da ECU

SID=0x7E8	DLC=0x08	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
		N° Bytes=0x03	Modo=0x41	PID	Dados[0]	0xAA	0xAA	0xAA	0xAA

Figura 5.4: Formato dos pedidos e resposta à ECU.

Ao canal 1 é associado um filtro e uma máscara de modo a que o controlador CAN só processe mensagens com o identificador igual a 0x7E8. Para realizar esta configuração o filtro é definido com o valor 0x7E8 e a máscara com o valor 0xFFF. Ao controlador está associada uma rotina de interrupção, sendo esta abordada na secção 5.4.2.5. A configuração deste controlador é efetuada através do código fonte 5.6.

```
// Step 1: Turn on CAN module and switch it to configuration mode
CANEnableModule(CAN1,TRUE);
CANSetOperatingMode(CAN1, CAN_CONFIGURATION);
while(CANGetOperatingMode(CAN1) != CAN_CONFIGURATION);

// Step 2: Configure the CAN Module Clock
canBitConfig.syncJumpWidth = CAN_BIT_2TQ;
canBitConfig.propagationSegTq = CAN_BIT_3TQ;
canBitConfig.phaseSeg1Tq = CAN_BIT_3TQ;
canBitConfig.phaseSeg2Tq = CAN_BIT_3TQ;
canBitConfig.sample3Time = TRUE;
canBitConfig.phaseSeg2TimeSelect = TRUE;
CANSetSpeed(CAN1,&canBitConfig,SYSTEM_FREQ,CAN_BUS_SPEED);

// Step 3: Assign the buffer area to the CAN module.
CANAssignMemoryBuffer(CAN1,CAN1MessageFifoArea,(2 * 8 * 16));

// Step 4:Configure channel 0 for TX and size of 8 message buffers.
CANConfigureChannelForTx(CAN1, CAN_CHANNEL0, 8, CAN_TX_RTR_DISABLED, CAN_LOW_MEDIUM_PRIORITY);
CANConfigureChannelForRx(CAN1, CAN_CHANNEL1, 8, CAN_RX_DATA_ONLY);

// Step 5:Configure filters and mask
CANConfigureFilter(CAN1, CAN_FILTER0, 0x7E8, CAN_SID);
CANConfigureFilterMask(CAN1, CAN_FILTER_MASK0, 0xFFF, CAN_SID, CAN_FILTER_MASK_IDE_TYPE);
CANLinkFilterToChannel(CAN1, CAN_FILTER0, CAN_FILTER_MASK0, CAN_CHANNEL1);
CANEnableFilter(CAN1, CAN_FILTER0, TRUE);

// Step 6: Switch the CAN mode.
CANSetOperatingMode(CAN1, CAN_NORMAL_OPERATION);
while(CANGetOperatingMode(CAN1) != CAN_NORMAL_OPERATION);
```

Código fonte 5.6: Código para configuração do controlador CAN1.

## Módulos de sensores adicionais

No microcontrolador do sistema de aquisição de dados, o controlador CAN2 está configurado para efetuar a recepção dos dados recolhidos pelos módulos de sensores adicionais. As configurações para o funcionamento do controlador CAN2, no que diz respeito à fonte de relógio, são iguais às configurações do controlador CAN1 dos módulos de sensores adicionais, como referido na secção 5.3.2.2. Já as configurações relacionadas com os canais e as funções dos mesmos são diferentes. Neste caso também é aplicado um filtro e uma máscara ao canal dedicado à receção de mensagens, de modo a que este sistema só processe mensagens para o qual está preparado. O controlador CAN2 também está preparado para gerar uma interrupção quando recebe uma mensagem, de modo a que os dados recebidos sejam processados pelo microcontrolador.

As configurações efetuadas ao controlador CAN2 começam por associar uma variável à memória FIFO do controlador, bem como por definir o número de mensagens guardadas por cada canal configurado. Neste caso a variável está preparada para alocar até oito mensagens por canal. De seguida são configurados os canais, sendo o canal 0 para a

transmissão e o canal 1 para a recepção de mensagens. O canal 0 foi configurado para que em futuras extensões ao sistema o controlador já se encontre preparado para efetuar a transmissão de mensagens.

Ao canal 1 do controlador CAN2 é associado um filtro e uma máscara, de modo a que o microcontrolador só processe mensagens com o identificador contido no intervalo [0x150,0x15F]. Tendo em conta que os dois módulos de sensores adicionais utilizam os identificadores 0x150 e 0x151, o canal 1 já se encontra preparado para receber mais 14 identificadores em futuras melhorias. Para realizar esta configuração o filtro é definido com o valor 0x150 e a máscara com o valor 0xFF0. Ao controlador está associada uma rotina de interrupção, sendo esta abordada na secção 5.4.2.5. Por fim a configuração deste controlador é efetuada através do código fonte 5.7.

```
// Step 1: Turn on CAN module and switch it to configuration mode
CANEnableModule(CAN2,TRUE);
CANSetOperatingMode(CAN2, CAN_CONFIGURATION);
while (CANGetOperatingMode(CAN2) != CAN_CONFIGURATION);

// Step 2: Configure the CAN Module Clock
canBitConfig.syncJumpWidth      = CAN_BIT_2TQ;
canBitConfig.propagationSegTq    = CAN_BIT_3TQ;
canBitConfig.phaseSeg1Tq        = CAN_BIT_3TQ;
canBitConfig.phaseSeg2Tq        = CAN_BIT_3TQ;
canBitConfig.sample3Time        = TRUE;
canBitConfig.phaseSeg2TimeSelect = TRUE;
CANSetSpeed(CAN2,&canBitConfig,SYSTEM.FREQ,CAN_BUS_SPEED);

// Step 3: Assign the buffer area to the CAN module.
CANAssignMemoryBuffer(CAN2,CAN2MessageFifoArea,(2 * 8 * 16));

// Step 4: Configure channel 0 for TX and size of 8 message buffers.
CANConfigureChannelForTx(CAN2,CAN_CHANNEL0,8,CAN_TX_RTR_DISABLED,CAN_LOW_MEDIUM_PRIORITY);
CANConfigureChannelForRx(CAN2,CAN_CHANNEL1,8,CAN_RX_FULL_RECEIVE);

// Step 5: Configure filters and mask
CANConfigureFilter(CAN2,CAN_FILTER0,0x150,CAN_SID);
CANConfigureFilterMask(CAN2,CAN_FILTER_MASK0,0xFF0,CAN_SID,CAN_FILTER_MASK_ANY_TYPE);
CANLinkFilterToChannel(CAN2,CAN_FILTER0,CAN_FILTER_MASK0,CAN_CHANNEL1);
CANEnableFilter(CAN2,CAN_FILTER0,TRUE);

// Step 6: Switch the CAN mode.
CANSetOperatingMode(CAN2,CAN_NORMAL_OPERATION);
while (CANGetOperatingMode(CAN2) != CAN_NORMAL_OPERATION);
```

Código fonte 5.7: Código para configuração do controlador CAN2.

## Módulo inercial

No sistema de aquisição é utilizado um barramento I<sup>2</sup>C para efetuar a comunicação entre o microcontrolador e módulo inercial MM7150, sendo este barramento associado à interface I<sup>2</sup>C1 do microcontrolador. Este módulo também requer que o microcontrolador tenha configurado um pino como entrada digital associado a uma interrupção externa, sendo este ligado ao pino RD0. Como o módulo inercial é o único dispositivo ligado a este barramento I<sup>2</sup>C, de modo a efetuar as configurações da interface do microcontrolador, verificou-se no *datasheet* que a velocidade de comunicação pode ser definida entre [100 ; 400] kbit/s e que o endereçamento do dispositivo é composto por 7 *bits*. Para configurar o barramento é necessário configurar dois registos do microcontrolador, sendo eles o I2C1CON e o I2C1BRG [49].

Tendo em conta as especificações do módulo inercial, o registo I2C1CON habilita a interface (ON). De modo a tornar a aquisição de dados mais rápida, a velocidade definida para esta interface é de 400 kbit/s. Para configurar a velocidade de comunicação para esta interface ( $F_{SCK}$ ) é preciso definir o valor do registo I2C1BRG. Perante o *datasheet* do microcontrolador o registo I2C1BRG é configurado com o valor de 47, como mostra a equação 5.6.

$$I2C1BRG = \frac{PBCLK}{2 \cdot F_{SCK}} - 1 - \frac{PBCLK \cdot T_{PGOB}}{2} = \frac{40 \text{ MHz}}{2 \cdot 400 \text{ kHz}} - 1 - \frac{40 \text{ MHz} \cdot 130 \text{ ns}}{2} = 47 \quad (5.6)$$

Para comunicar com o módulo inercial, o código desenvolvido está baseado num exemplo disponibilizado pela *Microchip*, sendo este preparado para trabalhar com o microcontrolador PIC24FJ128GA010 de 16 *bits*. A configuração do módulo começa por inicializar os *Virtual Registers* (VREG) para os relatórios de dados. Após a inicialização dos registos VREG é obtida a tabela com os descritores *Human Interface Device* (HID) do módulo inercial. Com as tabelas obtidas é efetuado o envio das ordens para habilitar e reinicializar o módulo. De seguida são adquiridos os descritores dos relatórios emitidos pelo módulo inercial. Tendo em conta os descritores dos relatórios, obtêm-se os mesmos para os sensores válidos no módulo inercial. Tendo em conta que se pretende obter os valores do acelerómetro e do giroscópio, estes são habilitados e é configurada a sensibilidade e a taxa de atualização do sensor. Por fim a configuração da interface e do módulo inercial é efetuada através do código fonte 5.8 [86][87].

```
//INTERFACE CONFIG - I2C1
TRISDbits.TRISD0=1; //Set Interrupt INT0 as
input //Turn ON I2C1 module
I2C1CONbits.ON=1; //Set BaudRate I2C
I2C1BRG=47;

//Declaration of registers and initialize module MM7150
memset(&VREGS, 0x00, sizeof(VREGS)); //Initialize VREG registers
hid_i2c_descriptor_handler(GET_HID_DESC); //Get HID descriptor from
SSC7150
hid_i2c_cmd_process(ucBuf, POWER_ON, ARB.ID); //Issue HID Power ON
VREGS.SHC.reset = VREG_RESET_INIT; //Set SHC reset bit
hid_i2c_descriptor_handler(GET_RPT_DESC); //Get HID Report

//Go through the sensor list but only GetFeatureReport for valid ids
for (ucSensor_num = 0; ucSensor_num <= NUM_SENS; ucSensor_num++){
    if (SENSOR[ucSensor_num].id != 0xFF && SENSOR[ucSensor_num].id != 0){
        // Get feature report for all valid sensors and update VREGS register
        hid_i2c_cmd_process(ucBuf, HID_GET_RPT_FEAT, SENSOR[ucSensor_num].id);
    }
}
//Enable and configure the accelerometer and to the gyrometer.
HOST_SF_LIB_VREG_write(VREG_SHC, wRMWdata | VREG_SHC_ACC_EN); //Enable accelerometer
HOST_SF_LIB_VREG_write(VREG_ACSEN, 1); //Change acc sensitivity
HOST_SF_LIB_VREG_write(VREG_ACDXR, MAX_DATA_RATE); //Change acc data rate
HOST_SF_LIB_VREG_write(VREG_SHC, wRMWdata | VREG_SHC_GYRO_EN); //Enable the Gyrometer
HOST_SF_LIB_VREG_write(VREG_GYSEN, 0x01); //Change gyro sensitivity
HOST_SF_LIB_VREG_write(VREG_GYDR, MAX_DATA_RATE); //Change gyro data rate
```

Código fonte 5.8: Código para configuração da interface I<sup>2</sup>C1.

## Módulo GPS

De modo a obter a localização absoluta da mota no globo terrestre é utilizado o módulo GPS Quectel L80. O microcontrolador comunica com o módulo através da interface UART5, como referido na secção 3.4. O formato das mensagens para configuração do módulo e para a transmissão de informação para o microcontrolador são definidas pelo protocolo NMEA 0183 [88].

A distinção das mensagens é feita através do início da sequência de caracteres das mesmas, sendo que as mensagens podem apresentar as seguintes sequências:

- \$GP, define as mensagens que contêm os dados adquiridos pelo módulo,
- \$PMKT, define as mensagens para configuração do módulo.

Para transmitir os dados adquiridos, este módulo utiliza os sete formatos de mensagens, GPRMC, GPVTG, GPGGA, GPGSA, GPGSV, GPGLL e GPTXT. Dos formatos de mensagens suportados pelo módulo, só o GPRMC contém toda a informação requerida para a solução, que disponibiliza os seguintes dados:

- Data e horas no fuso horário UTC,
- Posição absoluta da mota (latitude e longitude),
- Velocidade no solo em nós,
- Curso sobre o solo em graus,

- Variação magnética em graus,
- Modo de posicionamento.

Quando o módulo é inicializado começa a trabalhar com as configurações padrão, sendo elas a velocidade de comunicação de 9600 bit/s, a taxa de atualização de 1 Hz e o envio de todos os formatos de mensagem suportados pelo módulo.

Tendo em conta os requisitos da solução de modo a atingir a taxa de atualização e a reduzir a quantidade de dados recebidos que não são utilizados, o módulo é configurado com os seguintes parâmetros:

- Taxa de atualização para 10 Hz,
- Transmissão de dados só no formato GPRMC.

No protocolo NMEA 0183 é indicado que a taxa de atualização pode ser alterada através da mensagem no formato PMTK300, sendo que esta define o período entre mensagens. De modo a associar os valores recolhidos pelo sistema a uma localização geográfica mais precisa, o módulo GPS é configurado para atualizar os dados a sua taxa de atualização máxima. Como referido na secção 3.2.4 o módulo GPS tem a capacidade de atualizar os dados a uma frequência máxima de 10 Hz, sendo este parâmetro configurado através da mensagem "\$PMTK300,100,0,0,0,0\*2C<CR><LF>" [88].

De modo a configurar o módulo para enviar só as mensagens do formato GPRMC é enviada a mensagem de configuração PMTK314, que define quais são as mensagens que o módulo deve emitir. A mensagem enviada para efetuar esta configuração é "\$PMTK314,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0\*29<CR><LF>".

Para configurar a interface UART5 do microcontrolador para a comunicação com o módulo GPS é necessário ajustar três registos. O registo (U5MODE) define o modo de funcionamento da interface, o registo (U5BRG) ajusta a velocidade de comunicação e o registo (U5STA) configura o sentido da transmissão de dados.

O modo de funcionamento da interface do microcontrolador é configurada para 8 *bits* de dados sem paridade (PDSEL), para um *bit* de paragem (STSEL) através do registo (U5MODE).

De modo a que o microcontrolador receba a informação num curto espaço de tempo para que o mesmo possa processar outras tarefas enquanto o módulo GPS não envia novos dados, o *baudrate* após a configuração do módulo GPS é ajustado para o valor máximo de 115 200 bit/s. Neste caso foram calculados dois valores para o registo (U5BRG), um para configurar o módulo GPS e outro para receber a informação do mesmo. Assim para a velocidade de 9600 bit/s é utilizado o valor do registo (U5BRG) igual a 265 e para a velocidade de 115 200 bit/s de 21, como mostram as equações 5.7 e 5.8

$$U5BRG_{9600} = \frac{PBCLK}{16 \cdot BaudRate} - 1 = \frac{40 \text{ MHz}}{16 \cdot 9600} - 1 \approx 260 \quad (5.7)$$

$$U5BRG_{115200} = \frac{PBCLK}{16 \cdot BaudRate} - 1 = \frac{40 \text{ MHz}}{16 \cdot 115200} - 1 \approx 21 \quad (5.8)$$

Para configurar o módulo GPS é necessário transmitir mensagens do microcontrolador para o módulo GPS. Deste modo é preciso habilitar a transmissão (UTXEN) e a receção (URXEN) de dados através do registo (U5STA).

A esta interface do microcontrolador está associada uma rotina de interrupção, sendo a mesma abordada na secção 5.4.2.5. Tendo em conta todas as especificações referidas anteriormente, a interface e o módulo GPS são configurados através do código fonte 5.9.

```

//Config UART5
U5MODEbits.PDSEL=0; //8-bit data and no parity
U5MODEbits.STSEL=0; //1 Stop bit
U5BRG=260; //Configure baudrate UART 9600 bps
U5STAbits.URXEN=1; //Enable Rx channel
U5STAbits.UTXEN=1; //Enable Tx channel
U5MODEbits.ON=1; //Enable UART5 module

//Config IO's to GPS Module
TRISDbits.TRISD2=1; //Active antenna detected as input
TRISAbits.TRISA0=0; //Reset module as output
reset_pin(0); //Power on module

//Config GPS Module
Send_PMTK314(0,1,0,0,0,0); //Configure output NMEA messages only RMC format
Send_PMTK300(100); //Configure output rate to 10Hz (100ms)
Send_PMTK251(115200); //Configure module baudrate to 115200 bps

//Config final interface baudrate
U5MODEbits.ON=0; //Enable UART5 module
U5BRG=21; //Configure baudrate UART 1152000 bps
U5MODEbits.ON=1; //Disable UART5 module

```

Código fonte 5.9: Código para configuração da interface UART5 e do módulo GPS.

### 5.4.2.2 Gravação de dados

No sistema de aquisição de dados, é utilizado um cartão microSD para guardar todos os dados adquiridos. Como referido anteriormente na secção 3.2.5, esta funcionalidade é bastante importante porque durante a corrida não é permitida a transmissão de dados, logo os dados adquiridos pelo sistema durante a prova seriam perdidos. A implementação do cartão também trás a vantagem de a equipa ter a possibilidade de configurar alguns parâmetros do sistema tais como, habilitar a transmissão de dados sem fios, seleccionar a taxa de transmissão de dados entre outros.

Tendo em consideração o *hardware* referido na secção 4.3.5, é possível verificar que o cartão microSD partilha o barramento SPI com o *transceiver* rádio. Este barramento está associado à interface SPI2 do microcontrolador. Para configurar a interface no microcontrolador é necessário configurar os registos (SPI2CON) e (SPI2BRG).

De modo a que a interface SPI2 do microcontrolador comunique com os dois dispositivos, é necessário configurar o modo de comunicação para 16 *bits* (MODE32 e MODE16), habilitar o modo *master* (MSTEN), indicar que a linha de dados é alterada quando o sinal de relógio muda do estado alto para o estado baixo (CKE), especificar qual é o nível lógico do relógio conforme o estado pretendido (CKP) e por fim habilitar a interface (ON). Também são configurados dois pinos como saídas digitais para fazer a seleção do dispositivo.

Para configurar o *baudrate* da interface do microcontrolador foram verificadas as velocidades de comunicação dos dois dispositivos, sendo que o *transceiver* rádio e o cartão, se for da classe 10, suportam velocidades de comunicação até aos 10 Mbit/s. Tendo em consideração o valor indicado, obtem-se o valor para o registo SPI2BRG igual a 2, que configura o barramento para uma velocidade de 8 Mbit/s, como indica a equação 5.9 [53][89].

$$F_{SCK} = \frac{F_{PBCLK}}{2 \cdot SPI2BRG + 1} = \frac{40 \text{ MHz}}{2 \cdot 2 + 1} = 8 \text{ Mbit/s} \quad (5.9)$$

A configuração da interface SPI2 é efetuada através do código fonte 5.10. Depois da interface do microcontrolador estar configurada, efetua-se a inicialização e a configuração do cartão *microSD*. Para realizar as operações entre o microcontrolador e o cartão microSD são utilizadas as bibliotecas *FATFS*, que permitem criar um sistema de ficheiros *File Allocation Table* (FAT) [90].

```

//Config SPI2
SPI2CONbits.ON = 0; // Disable module SPI2

```

```

SPI2BRG = 2;           // FSK = FPBCLK/(2*SPIxBRG+1); FSK = 8MHz@PBclk=0MHz
SPI2BUF = 0;          // Clear the receive buffer
/* SPI2CON settings */
SPI2CONbits.MODE32=0; // Disable 32 bits mode
SPI2CONbits.MODE16=1; // Enable 16 bits mode
SPI2CONbits.SSEN = 0; // Device disabled to slave mode
SPI2CONbits.MSTEN = 1; // Device enabled to master mode
SPI2CONbits.CKE = 1;  // Output data changes on transition from active to idle
SPI2CONbits.CKP = 0;  // Clock is idle when low, active when high
SPI2CONbits.ON = 1;   // Enable module SPI2
/* SPI2 Slave Select Config */
TRISAbits.TRISA3 = 0; //Slave Select Pin A3 microSD

```

Código fonte 5.10: Código para configuração da interface SPI2.

A inicialização do cartão começa por verificar a comunicação entre o dispositivo e o microcontrolador e por associar um número à unidade de memória. De seguida é associado ao cartão inicializado, um objeto que define o sistema de ficheiros utilizado FAT. De modo a que o cartão fique pronto a ser utilizado durante o funcionamento do sistema, abre-se a pasta na raiz do dispositivo de memória.

Com o cartão pronto a ser utilizado é feita a leitura dos parâmetros de configuração como por exemplo a taxa de envio de mensagens pelo transmissor RF. Após a inicialização do cartão são verificados quantos ficheiros existem com dados anteriormente adquiridos pelo sistema, de modo a não escrever em ficheiros já existentes. Para finalizar este processo é criado um novo ficheiro de texto com o nome "REC\_xx.bin", em que as letras "xx" representam o número de ficheiros anteriormente criados. O fluxograma associado à inicialização do cartão *microSD* está representado na figura 5.5.

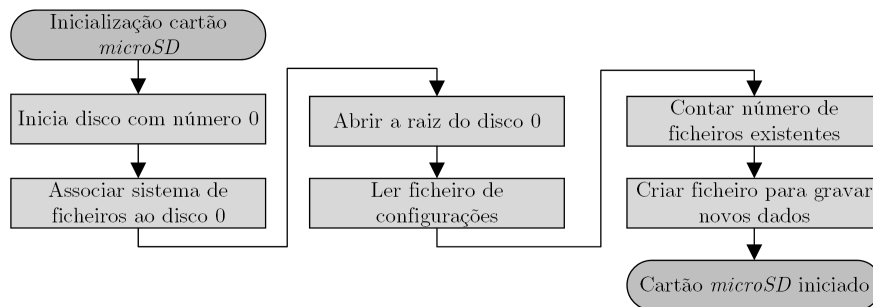


Figura 5.5: Fluxograma de inicialização do cartão *microSD*.

#### 5.4.2.3 Transmissão de dados

No grupo da transmissão de dados do sistema de aquisição, está inserido o bloco funcional que efetua a transmissão local dos dados recolhidos para o ecrã da mota e o bloco funcional associado à transmissão remota de dados por uma interface rádio operando numa banda ISM. Por cada bloco funcional é feita uma descrição da configuração da interface do microcontrolador bem como do dispositivo associado. Por cada configuração é apresentado o respetivo código de fonte.

#### Ecrã da mota

A equipa definiu como requisito para a solução a transmissão de dados adquiridos a uma taxa de atualização de 10 Hz para um ecrã já desenvolvido. O ecrã desenvolvido pela equipa está preparado para receber os dados através de um barramento USB, sendo que não está definido nenhum formato da mensagem a ser recebida pelo mesmo. Para minimizar o número de instruções necessárias para construir uma mensagem específica para o ecrã, é enviada uma *string* composta por 42 *bytes* que corresponde ao tamanho



da estrutura de dados definida anteriormente na secção 5.2.2 mais 6 *bytes* que indicam o início e o fim do pacote.

Como é referido na secção 4.3.4 é utilizado um conversor USB-UART para efetuar a comunicação entre o ecrã e o microcontrolador. Para enviar para o ecrã o microcontrolador utiliza a interface UART3. Para um correto funcionamento da interface UART3 do microcontrolador na comunicação com o ecrã, é preciso configurar os registos associados à mesma, do modo de funcionamento (U3MODE), da velocidade de comunicação (U3BRG) e definir o sentido da transmissão de dados (U3STA).

Com o registo U3MODE é configurado o modo de funcionamento da interface, onde é definido o formato dos dados (PDSEL), indicado o número de *bits* utilizados para sinalizar o fim da mensagem (STSEL) e habilitar a interface do microcontrolador (ON). Neste caso a comunicação é efetuada com 8 *bits* sem paridade e utiliza um *bit* de sinalização de fim da mensagem.

A velocidade de comunicação é definida pelo valor do registo (U3BRG) que é calculado através do valor do relógio associado aos periféricos PBCLK e do *baudrate* escolhido. De modo a que o microcontrolador envie os dados recolhidos num curto espaço de tempo para que o mesmo possa processar um maior número de tarefas associadas ao sistema, o *baudrate* é configurado para a velocidade de 115 200 bit/s. O valor obtido para o registo U3BRG é igual a 21, como mostra a equação 5.10 [91].

$$U3BRG_{115200} = \frac{PBCLK}{16 \cdot BaudRate} - 1 = \frac{40 \text{ MHz}}{16 \cdot 115200} - 1 = 20.7 \approx 21 \quad (5.10)$$

Como este sistema só vai transmitir os dados adquiridos para o ecrã, com o registo U3STA é configurada a interface de modo a habilitar a transmissão de dados (UTXEN) e desabilitar a receção dos mesmos (URXEN).

A configuração dos registos associados à interface UART3 está apresentada no código fonte 5.11.

```
//CONFIG UART3 - Dashboard
U3MODEbits.PDSEL=0;           //8-bit data and no parity
U3MODEbits.STSEL=0;          //1 Stop bit
U3BRG=520;                    //Configure baudrate UART
U3STAbits.URXEN=0;           //Disable Rx channel
U3STAbits.UTXEN=1;           //Enable Tx channel
U3MODEbits.ON=1;             //Enable UART3 module
```

Código fonte 5.11: Código para configuração da interface UART3.

## Módulo RF

Para a transmissão de dados por uma banda ISM como referido na secção 3.4 é utilizado o *transceiver* RFM95PW. Este dispositivo comunica com o microcontrolador através de um barramento SPI. O barramento SPI é utilizado também pelo cartão microSD, sendo controlado pela interface SPI2 do microcontrolador. As configurações efetuadas à interface SPI2 do microcontrolador estão apresentadas na secção 5.4.2.2. De modo a transmitir os dados recolhidos anteriormente pelo sistema, é necessário inicializar o *transceiver* rádio.

Para inicializar este *transceiver* é preciso primeiro definir alguns parâmetros de comunicação, tais como a frequência da onda portadora ( $F_{rf}$ ), o valor do *bitrate*, o valor do desvio de frequência ( $F_{dev}$ ) e a largura de banda utilizada ( $RF_{BW}$ ). De modo a definir quais são os valores para configurar o módulo RF, verificou-se na tabela 2.8, qual seria a banda mais adequada para a solução. Para utilizar a potência de transmissão mais elevada é utilizada a frequência na gama dos [869.4-869.65]MHz. A utilização desta banda requer o cumprimento dos seguintes requisitos [92]:

- Frequência da onda portadora no intervalo [869.4-869.65]MHz,
- Potência de transmissão  $\leq 500\text{mW}$  ou  $+27\text{dBm}$ ,
- Canais com largura de banda total de 25 kHz,
- Pode ser utilizada a totalidade da faixa como um canal para transmissão de dados de alto debito,
- *Duty-cycle*  $\leq 10\%$  ou utilização de todo período para dispositivos com *Adaptative Frequency Agility* (AFA) se for utilizado a técnica *Listen Before Talk* (LBT),
- O *duty-cycle*, LBT ou técnica equivalente não pode ser dependente do utilizador.

Considerando os requisitos indicados anteriormente foi feito um estudo, para que os dados transmitidos à frequência definida pela equipa não ultrapassem o *duty-cycle* de 10%. Tendo em conta que no máximo serão enviadas 20 mensagens por segundo e que cada mensagem é constituída por 58 bytes, como apresentado na tabela 5.2, é necessário enviar 9280 bit/s como mostra a equação 5.11.

Tabela 5.2: Número de bytes por mensagem.

Parâmetros de mensagem	Nº Bytes
Preâmbulo	15
Palavra de sincronização	3
Comprimento de mensagem	1
Endereço de recetor	1
Mensagem	36
CRC	2
Total	58

$$Tx_{BitRate} = N^{\circ} Mensagens \cdot (N^{\circ} Bytes \cdot 8) = 20 \cdot (58 \cdot 8) = 9280 \text{ bit/s} \quad (5.11)$$

Para garantir que não é ultrapassado o *duty-cycle*, o *bitrate* tem que ser superior a 92.8 kbit/s, logo é definido um *bitrate* de 100 kbit/s. Considerando a utilização da totalidade da faixa disponível como um canal de transmissão de dados, possibilita a utilização de uma largura de banda de 250 kHz. A largura de banda utilizada pelo canal de transmissão de dados ( $O_{BW}$ ) calculou-se através da equação 5.12, sendo esta uma simplificação da regra de *Carson*[93].

$$O_{BW} = F_{dev} \cdot 2 + BitRate \quad (5.12)$$

Tendo em conta os valores indicados através da inequação 5.13 é obtido o valor máximo de 75 kHz para a frequência de desvio. Considerando a utilização de uma frequência de desvio com o valor de 50 kHz e um *bitrate* de 100 kHz, obtém-se uma largura de banda utilizada de 200 kHz. Para definir a frequência da onda portadora foi tido em conta o valor da largura de banda e do início da faixa de frequência, sendo obtido o valor de 869.5 MHz como mostra a equação 5.14.

$$F_{dev} \geq \frac{O_{BW} - BitRate}{2} \geq \frac{250k - 100k}{2} \geq 75 \text{ kHz} \quad (5.13)$$

$$RF_{BW} = RF_{base} + \frac{RF_{BW}}{2} = 869.4 \text{ MHz} + \frac{200 \text{ kHz}}{2} = 869.5 \text{ MHz} \quad (5.14)$$

Para calcular os registos associados ao módulo RF, têm-se em conta os valores referidos anteriormente e os indicados no *datasheet* do módulo [89], sendo eles:

- Oscilador interno do módulo RF ( $F_{OSC}$ ), com um frequência igual 32 MHz,
- Passo de ajuste de frequências ( $F_{STEP}$ ) de 61 Hz,
- *Bitrate* para transmissão de dados igual a 100 kbit/s,
- Frequência de desvio ( $F_{dev}$ ) igual a 50 kHz,
- Frequência para onda portadora ( $F_{RF}$ ) igual a 869.5 MHz,

Para ajustar os registos no módulo (RegBitRateMsb e RegBitRateLsb) associados ao valor de *bitrate* da transmissão de dados, o valor calculado para a palavra ( $BitRate_{val}$ ) é igual a 320 como apresentado na equação 5.15.

$$BitRate_{val} = \frac{F_{OSC} - \frac{BitRate}{16}}{BitRate} = \frac{32 \text{ MHz} - \frac{100 \text{ kbit/s}}{16}}{100 \text{ kbit/s}} = 320 \quad (5.15)$$

De forma a configurar os registos (RegFdevMsb e RegFdevLsb) associados ao valor do desvio de frequência ( $F_{dev_{val}}$ ), obteve-se o valor igual a 820 como mostra a equação 5.16.

$$F_{dev_{val}} = \frac{F_{dev}}{F_{step}} = \frac{50 \text{ kHz}}{61 \text{ Hz}} = 820 \quad (5.16)$$

De modo a configurar o valor da onda portadora ( $F_{rf_{val}}$ ) para uma frequência de 869.5 MHz, definem-se os três registos da mesma (RegFrMsb, RegFrMid e RegFrLsb) para um valor igual 14252460, como apresentado na equação 5.17.

$$F_{rf_{val}} = \frac{F_{rf}}{F_{step}} = \frac{869.4 \text{ MHz}}{61 \text{ Hz}} = 14252460 \quad (5.17)$$

Tendo em consideração que neste sistema este módulo só trabalha como emissor, os registos associados ao transmissor são configurados de modo a que o módulo RF utilize a sua potência máxima para a transmissão de dados. A configuração da potência que o transmissor utiliza é realizada através dos registos RegPaConfig e RegPaRamp.

Para a manipulação dos pacotes de dados são configurados dois aspetos, o modo de transmissão e a definição associada ao formato e ao tamanho da mensagem que o módulo vai enviar. Em relação ao modo da transmissão de dados, este sistema vai efetuar o envio dos mesmos por pacotes. Sendo utilizada a transmissão de dados por pacotes, configura-se o formato dos mesmos para que sejam formados pelo número de *bytes* já referidos.

Este módulo tem a possibilidade de habilitar uma máquina de estados interna para processar a transmissão e recepção de dados. Neste caso a máquina de estados está configurada, para que o módulo efetue o envio de um pacote por cada vez que a mesma é iniciada. Também é definido que a máquina de estados só pode ser iniciada com o módulo no modo de *standby*. Na figura 5.6 está apresentada a máquina de estados implementada neste sistema para efetuar a transmissão de dados.

Para finalizar a configuração do módulo rádio é atribuído ao pino DIO0, o estado de quando é efetuado o envio de um pacote. Todas as configurações efetuadas aos registos internos do módulo estão apresentadas na tabela 5.3.

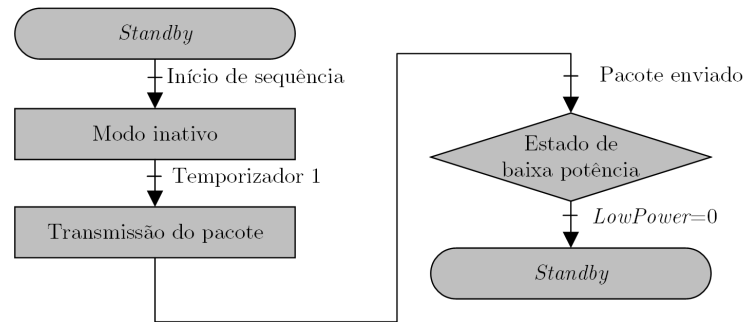


Figura 5.6: Máquina de estados implementada no módulo transmissor.

Tabela 5.3: Configurações efetuadas ao módulo RFM95PW [89].

Nome (Endereço)	Variável	Valor e descrição
Configurações gerais		
RegOpMode (0x01)	Modo de longo alcance	0 →Modo FSK/OOK
	Tipo de modulação	0 →FSK
	Modo de baixa frequência	0 →Frequência utilizada superior a 568MHz
	Modo	1 →Modo de <i>Standby</i>
RegBitrateMsb (0x02)	<i>Bitrate</i> (15:8)	320 →Para um <i>bitrate</i> 100 kbitseg
RegBitrateLsb (0x03)	<i>Bitrate</i> (7:0)	
RegFdevMsb (0x04)	Ajuste do desvio de frequência	820 →Desvio de frequência 50 kHz
RegFdevLsb (0x05)		
RegFrfMsb (0x06)	Ajuste da frequência da onda portadora	14252460 →Frequência da onda portadora 869,5Mhz
RegFrfMid (0x07)		
RegFrfLsb (0x08)		
Configurações do transmissor		
RegPaConfig (0x09)	Seleção de PA	1 →Potência de sinal ate +20 dBm
	Potência de saída	15 →Potência de saída para 17 dBm
RegPaRamp (0x0A)	Filtro de modelação	0 →Sem filtro
	Rampa de PA	9 →Rampa com 40 us
Configurações de manipulação de pacotes de dados		
RegPreambleMsb (0x25)	Tamanho do preâmbulo	15 →Preâmbulo da mensagem de 15 bytes
RegPreambleLsb (0x26)		
RegSyncConfig (0x27)	Reinício automático do modo Rx	1 →Ligado, sem esperar por o bloqueio PLL
	Polaridade do preâmbulo	0 →Palavra enviada 0xAA (padrão)
	Sincronização	1 →Ligada
	Condição do preenchimento FIFO	0 →Na deteção da palavra de sincronização

RegSyncValue1 (0x28)	Primeiro byte de sincronização	Byte com o valor 0xAA
RegSyncValue2 (0x29)	Segundo byte de sincronização	Byte com o valor 0x5E
RegSyncValue3 (0x2A)	Terceiro byte de sincronização	Byte com o valor 0xD4
RegPacketConfig1 (0x30)	Formato do pacote	1 →Comprimento de pacote variável
	Verificação CRC habilitada	1 →Ligada
	Apagar FIFO na falha de CRC	0 →Apaga FIFO na falha de CRC
	Filtragem por endereço	1 →Endereço igual ao do recetor
RegPacketConfig2 (0x31)	Modo de transmissão de dados	1 →Por pacotes de dados
Configuração da máquina de estados		
RegSeqConfig1 (0x36)	<i>IdleMode</i>	0 →Início a partir do modo de standby
	<i>FromStart</i>	2 →Entra no modo de transmissão
	<i>LowPowerSelection</i>	0 →Desliga a máquina de estados
Configuração dos pinos digitais do módulo		
RegDioMapping1 (0x40)	Utilização do pino DIO0	0 →Pacote enviado

#### 5.4.2.4 Temporizadores

No sistema de aquisição de dados são utilizados dois temporizadores para garantir as taxas requeridas para a solução. Os temporizadores garantem a taxa de transmissão de dados quando o *transceiver* rádio está operacional, o registo do tempo decorrido desde a inicialização do sistema para associar os dados adquiridos a um dado período de tempo e finalmente para garantir a taxa de gravação dos dados adquiridos no cartão microSD. De seguida serão descritas as configurações efetuadas a cada temporizador utilizado.

#### Temporizador 2 - Transmissão de dados por rádio

Este temporizador tem como função despertar uma interrupção, de modo a efetuar as transmissões de dados por rádio. Este sistema está preparado para enviar dados até uma frequência de 20 Hz. O valor para a maior frequência de envio foi definido tendo em conta a frequência máxima de aquisição de algumas grandezas e o mínimo com a taxa requerida pela a equipa é de 5 Hz.

O valor do temporizador é calculado através das configurações guardadas num ficheiro de configuração no cartão *microSD*. Caso não exista nenhum ficheiro de configuração no cartão *microSD*, a frequência padrão de envio de dados é de 10 Hz. Tendo em consideração que para configurar o temporizador do microcontrolador (TMR2) é preciso configurar os registos (T2CON) e (PR2) como referido também para o módulo de sensores adicionais na secção 5.3.2.3.

De modo a que o valor associado ao temporizador seja alterado perante as configurações efetuadas pela equipa, foi verificado qual é o valor mínimo para a frequência de transmissão, tendo em conta que o temporizador é configurado como um temporizador de 16 *bits*. A equação para o calculo do PR2, depende do valor da frequência e do *Prescale* do temporizador, como é indicado na equação 5.18. A tabela 5.4 mostra os valores obtidos do registo PR2 para diferentes valores de frequência e de *Prescale*. As células

que aparecem sombreadas a vermelho indicam que o valor de PR2 é superior a 65535, logo o temporizador não pode ser configurado com essas características. É verificado que não é possível ter um temporizador para uma frequência inferior a 3 Hz.

$$PR2 = \frac{PBCLK}{f_{Timer2} \cdot Prescaler} \quad (5.18)$$

Tabela 5.4: Valores de PR2 perante os valores de frequência e *Prescale*.

Frequência (Hz)	<i>Prescale</i> =16 TKPS=4	<i>Prescale</i> =32 TKPS=5	<i>Prescale</i> =64 TKPS=6	<i>Prescale</i> =256 TKPS=7
1	2500000	1250000	625000	156250
2	1250000	625000	312500	78125
3	833333	416667	208333	52083
4	625000	312500	156250	39063
5	500000	250000	125000	31250
6	416667	208333	104167	26042
7	357143	178571	89286	22321
8	312500	156250	78125	19531
9	277778	138889	69444	17361
10	250000	125000	62500	15625
11	227273	113636	56818	14205
12	208333	104167	52083	13021
13	192308	96154	48077	12019
14	178571	89286	44643	11161
15	166667	83333	41667	10417
16	156250	78125	39063	9766
17	147059	73529	36765	9191
18	138889	69444	34722	8681
19	131579	65789	32895	8224
20	125000	62500	31250	7813

Para facilitar o desenvolvimento do código, a frequência para a transmissão de dados só pode ser configurada pela equipa no intervalo [3 ; 20] Hz. Caso o valor não esteja definido dentro do intervalo é definido pelo valor padrão de 10 Hz.

Para a correta configuração do temporizador, tendo em consideração que o valor da frequência já foi obtido na inicialização do cartão *microSD* e que se encontra armazenado na variável (*freq\_tx*), os registos do mesmo são configurados através do código fonte 5.12.

```
//CONFIG TIMER2
T2CONbits.ON = 0;
TMR2 = 0;
if(freq_tx > 2 && freq_tx < 10){
    T2CONbits.TCKPS=7;
    PR2=(4000000)/(freq_tx*256);
} else if(freq_tx > 9 && freq_tx < 20){
    T2CONbits.TCKPS=6;
    PR2=(4000000)/(freq_tx*64);
} else if(freq_tx == 20){
    T2CONbits.TCKPS=5;
    PR2=(4000000)/(freq_tx*32);
} else{
    T2CONbits.TCKPS=6;
    PR2=62500;
}
//Disable Timer2
//Reset counter value
//Configure timer between 3-9 Hz
//Configure Prescaler to 1:256
//Set period time
//Configure timer between 10-19 Hz
//Configure Prescaler to 1:64
//Set period time
//Configure timer between 20 Hz
//Configure Prescaler to 1:32
//Set period time
//Configure timer to 10 Hz
//Configure Prescaler to 1:64
//Set period time
```

Código fonte 5.12: Código para configuração do *Timer2*.

### Temporizador 3 - Cronómetro

Para associar as variáveis a um tempo específico e ser possível calcular o tempo por volta em pista, utiliza-se o temporizador (TMR3) para criar um cronómetro interno no microcontrolador desde que o sistema de aquisição de dados é iniciado. As variáveis associadas ao cronómetro são incrementadas com uma frequência de 40 Hz, ou seja com uma resolução de tempo de 25 ms.

O temporizador é configurado da mesma forma que o anterior, sendo que neste caso os registos que contêm as configurações são (T3CON) e (PR3). Neste caso é possível utilizar um *Prescale* igual a 16 com o valor de período de 62500 como mostra a equação 5.19.

$$PR3 = \frac{PBCLK}{f_{Timer3} \cdot Prescaler} = \frac{40 \text{ MHz}}{40 \text{ Hz} \cdot 16} = 62500 \quad (5.19)$$

A configuração do *Timer3* é efetuada através do código fonte 5.13.

```
//CONFIG TIMER3 ----- Prescaler 1:8; PR3 Preload = 50000;
T3CONbits.ON = 0; //Disable Timer3
T3CONbits.TCKPS2=1; //Configure Prescaler to 1:16 (bit 2)
T3CONbits.TCKPS1=0; //Configure Prescaler to 1:16 (bit 1)
T3CONbits.TCKPS0=0; //Configure Prescaler to 1:16 (bit 0)
PR3 = 62500; //Set period value
TMR3 = 0; //Reset counter value
```

Código fonte 5.13: Código para configuração do *Timer3*.

#### 5.4.2.5 Interrupções

Para que o sistema de aquisição de dados execute as funções a partir de rotinas de interrupção, são configuradas as mesmas que estão associadas as interfaces do microcontrolador. Na configuração das interrupções é preciso ter consideração quais são as interfaces que estão associadas a uma rotina de interrupção bem como a sua prioridade. Tendo em conta o que foi referido na configuração das interfaces, é possível verificar qual é a lista das interfaces que precisam de ter uma rotina de interrupção. A lista das rotinas de interrupção utilizada bem como a função associada é:

- CAN1 na receção de dados da ECU,
- CAN2 na receção de dados dos módulos de sensores adicionais,
- UART5 na receção de dados do módulo GPS,
- INT0 indica ao microcontrolador que o módulo inercial tem novos dados para serem lidos,
- INT1 indica ao microcontrolador que o *transceiver* RF terminou o envio de um pacote,
- TMR2 utilizada para enviar um pacote pelo *transceiver* RF,
- TMR3 cronometrar o tempo e guardar os dados no cartão microSD.

Para definir as prioridades e sub-prioridades das interrupções foram tidos em conta os requisitos mínimos da solução indicados na secção 2.5. As interrupções são abordadas pela ordem de maior para a menor prioridade.

A interrupção com maior prioridade está associada à atualização do cronómetro e à criação de um pacote de dados para o cartão microSD. Tendo em conta que guardar os dados no cartão microSD é fundamental para evitar perda de dados, os dados guardadas no cartão são os valores atualizados das grandezas adquiridas pelo sistema quando o valor do cronómetro foi atualizado. Deste modo é possível afirmar que cada vez que o sistema atualiza o cronómetro é efetuada uma recolha de dados para guardar no cartão microSD.

Logo a interrupção associada ao temporizador 3 foi definida com uma prioridade e sub-prioridade com níveis 6 e 1 respetivamente.

Seguidamente a interrupção com maior prioridade é associada à aquisição da localização GPS da mota, de modo a associar as grandezas adquiridas a uma localização específica da pista. Tal como a referência temporal, a localização GPS também é uma referência bastante importante de modo a associar algumas grandezas à localização da mota, sendo assim a interrupção associada à receção de dados pela interface UART5 foi definida com uma prioridade e sub-prioridade com níveis 5 e 1 respetivamente.

As interrupções associadas unicamente à aquisição de dados estão definidas como média prioridade, tal como a interrupção associada ao controlador CAN1, ao controlador CAN2 e à interrupção externa INT0 gerada pelo módulo inercial. Deste modo estas interrupções foram definidas com a prioridade e sub-prioridade com níveis 4 e 1 respetivamente.

Com menor prioridade foram definidas as interrupções responsáveis pela transmissão de dados por rádio, sendo elas a interrupção do temporizador 2 e a interrupção externa INT1. Como o objetivo principal é efetuar a recolha de dados a transmissão de dados não é fundamental, deste modo estas interrupções estão configuradas com a prioridade e sub-prioridade com os níveis 2 e 1 respetivamente.

A configuração das interrupções é efetuada através do código fonte 5.14.

```
//-----INTERRUPTS-----//
//-----Config. Interrupts - CAN1-Rx event-----//
CANEnableChannelEvent(CAN1, CAN_CHANNEL1, CAN_RX_CHANNEL_ANY_EVENT, TRUE);
CANEnableModuleEvent(CAN1, CAN_RX_EVENT, TRUE);
INTSetVectorPriority(INT_CAN_1_VECTOR, INT_PRIORITY_LEVEL_4);
INTSetVectorSubPriority(INT_CAN_1_VECTOR, INT_SUB_PRIORITY_LEVEL_1);
INTEnable(INT_CAN1, INT_ENABLED);
//-----Config. Interrupts - CAN2-Rx event-----//
CANEnableChannelEvent(CAN2, CAN_CHANNEL1, CAN_RX_CHANNEL_NOT_EMPTY, TRUE);
CANEnableModuleEvent(CAN2, CAN_RX_EVENT, TRUE);
INTSetVectorPriority(INT_CAN_2_VECTOR, INT_PRIORITY_LEVEL_4);
INTSetVectorSubPriority(INT_CAN_2_VECTOR, INT_SUB_PRIORITY_LEVEL_1);
INTEnable(INT_CAN2, INT_ENABLED);
//-----Config. Interrupts - UART5-Rx event (Modulo GPS)-----//
IPC12bits.U5IP=5; //UART5 priority
IPC12bits.U5IS=1; //UART5 sub-priority
IFS2bits.U5RXIF=0; //Clear interrupt flag
IEC2bits.U5RXIE=1; //Active interrupt
//-----Config. Interrupts - INT0 data available (Modulo inercial)-----//
TRISDbits.TRISD0=1; //Set Interrupt INT0 as input
INTCONbits.INT0EP=0; //Falling edge event
IPC0bits.INT0IP=4; //INT0 priority
IPC0bits.INT0IS=1; //INT0 sub-priority
IFS0bits.INT0IF=0; //Clear interrupt flag
IEC0bits.INT0IE=1; //Active interrupt
//-----Config. Interrupts - INT1 packet sended (Modulo RF)-----//
TRISEbits.TRISE8=1; //Set Interrupt INT1 as input
INTCONbits.INT1EP=0; //Falling edge event
IPC1bits.INT1IP=2; //INT1 priority
IPC1bits.INT1IS=1; //INT1 sub-priority
IFS0bits.INT1IF=0; //Clear interrupt flag
IEC0bits.INT1IE=1; //Active interrupt
//-----Config. Interrupts - TIMER2-----//
IPC2bits.T2IP = 2; //TIMER2 priority
IPC2bits.T2IS = 1; //TIMER2 sub-priority
IFS0bits.T2IF = 0; //Clear interrupt flag
IEC0bits.T2IE = 1; //Active interrupt
//-----Config. Interrupts - TIMER3-----//
IPC3bits.T3IP = 6; //TIMER3 priority
IPC3bits.T3IS = 1; //TIMER3 sub-priority
IFS0bits.T3IF = 0; //Clear interrupt flag
IEC0bits.T3IE = 1; //Active interrupt
```

Código fonte 5.14: Código para configuração das interrupções.

### 5.4.3 Rotinas de interrupção

Como referido anteriormente no início desta secção, as funções deste sistema são executadas a partir de interrupções associadas às interfaces do microcontrolador. De modo a apresentar o funcionamento deste sistema, são expostas as rotinas de interrupção associadas a cada interface, bem como os seus fluxogramas de funcionamento.



Neste sistema são utilizadas as interrupções associadas aos dois controladores CAN, à interface UART5, as interrupções externas INT0 e INT1 e aos dois temporizadores utilizados.

#### 5.4.3.1 Leitura da ECU

Tendo em conta que os valores que são obtidos através da ECU requerem um pedido efetuado à mesma, foi efetuado um teste para verificar qual é o tempo desde que é realizado um pedido à ECU até à receção dos dados no microcontrolador. A tabela 5.5 mostra o resultado de dez testes com a duração de um segundo, em que são apresentados os números de pedidos realizados e recebidos e o tempo médio de resposta a um pedido. O teste realizado mostra que o processo de obter uma variável demora cerca de 10 ms. Sendo assim possível realizar cerca de 100 pedidos de informação ECU por segundo.

Tabela 5.5: Resultados do teste realizado ao número de pedidos à ECU.

Parâmetro	Amostra com duração de 1 s									
	1	2	3	4	5	6	7	8	9	10
Pedidos realizados	100	100	100	100	100	100	100	100	100	100
Pedidos recebidos	100	100	100	100	100	100	100	100	100	100
Tempo médio por pedido (ms)	10	10	10	10	10	10	10	10	10	10

Os valores obtidos através da ECU perante os requisitos propostos pela equipa estão apresentados na tabela 5.6. Tendo em conta o número de pedidos realizados à ECU por segundo, verifica-se que é possível obter todas as variáveis com as taxas de aquisição superiores às propostas pela equipa.

Tabela 5.6: Frequências de aquisição para as variáveis adquiridas da ECU.

PID (Hex.)	Variável	Aquisição Requerida (Hz)	Aquisição Implementada (Hz)
0x05	Temperatura da refrigeração do motor	1	4
0x0B	Pressão de ar na admissão	10	20
0x0D	Velocidade	5	10
0x0E	Avanço de ignição para cilindro 1	10	20
0x0F	Temperatura do ar na admissão	2	6
0x11	Posição do acelerador	10	20
0x21	RPM	10	20

Deste modo foi feita uma seleção dos valores que sofrem maior variação em menor espaço tempo, tal como o valor das RPM, como variáveis a uma taxa aquisição superior. Por exemplo os valores associados à temperatura do ar de admissão ou ao seletor de mudança, como não têm grande variação, são definidos a uma taxa de aquisição inferior. Para obter estas grandezas foi criado um *array* com 100 endereços de identificação associados aos pedidos a realizar à ECU. Os endereços estão organizados no *array* de forma uniforme, sendo este apresentado no anexo D.1.1.

Tendo em consideração que o controlador CAN1 está configurado para gerar uma interrupção após receção de mensagens pelo canal 1, depois da configuração de todas as interfaces antes do processo entrar no ciclo infinito, efetua-se o primeiro pedido de dados à ECU. Quando a ECU devolve os valores requeridos é chamada pela primeira vez a rotina de interrupção. Depois deste ponto, toda a aquisição de dados a partir da ECU é realizada através desta rotina.

Ao ser iniciada a rotina de interrupção associada ao controlador CAN1, é obtida a mensagem da memória FIFO. Com os valores recebidos já lidos é libertada a memória associada à receção de mensagens pelo canal 1. Após a mensagem ser obtida, esta é decodificada consoante os dados que são recebidos no terceiro *byte* de dados da mesma. De modo a receber um novo valor da ECU é feito um pedido à mesma, do endereço guardado no índice "i" do *array* construído para os pedidos da ECU. Para que na próxima vez que a interrupção for gerada seja realizado o pedido de um novo identificador, o índice associado à consulta do *array* é inicializado ou incrementado conforme o tamanho do *array* e o valor atual do índice. O fluxograma da rotina de interrupção do controlador CAN1 está apresentada na figura 5.7.

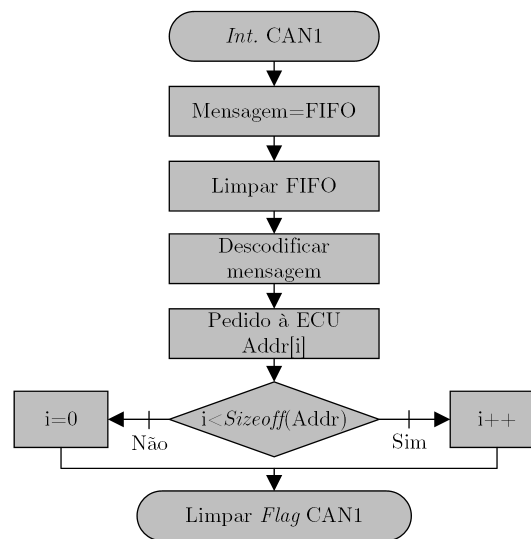


Figura 5.7: Rotina de interrupção associada ao controlador CAN1.

#### 5.4.3.2 Receção de dados dos módulos para sensores adicionais

De modo a receber os dados emitidos pelos módulos de sensores adicionais, é utilizado o controlador CAN2 do microcontrolador. Para este caso não é feita nenhuma transmissão ou pedido aos módulos. Estes estão preparados para enviar autonomamente mensagens a uma taxa de aquisição de 20 Hz. Quando é recebida uma mensagem por este controlador é executada a interrupção associada ao mesmo.

No início da interrupção é obtida a mensagem da memória FIFO associada ao canal 1 do controlador. Depois de ser adquirida a mensagem é lido o endereço da mesma, que identifica o módulo do qual foi enviada a mensagem. Sabendo a origem dos dados, os mesmos são guardados nas variáveis predefinidas para tal. Como a solução está inicialmente desenvolvida para suportar dois módulos de sensores adicionais, um na traseira (CAN ID=0x150) e outro na dianteira da mota (CAN ID=0x151), esta rotina de interrupção só está preparada para receber estes dois identificadores. O fluxograma associado a interrupção do controlador CAN2 está apresentado na figura 5.8.

#### 5.4.3.3 Receção de dados do módulo GPS

No momento em que é iniciada a receção de uma mensagem transmitida pelo módulo GPS, é despertada a interrupção associada à interface UART5. Como referido anteriormente, o módulo GPS está configurado para efetuar a transmissão de mensagens do tipo

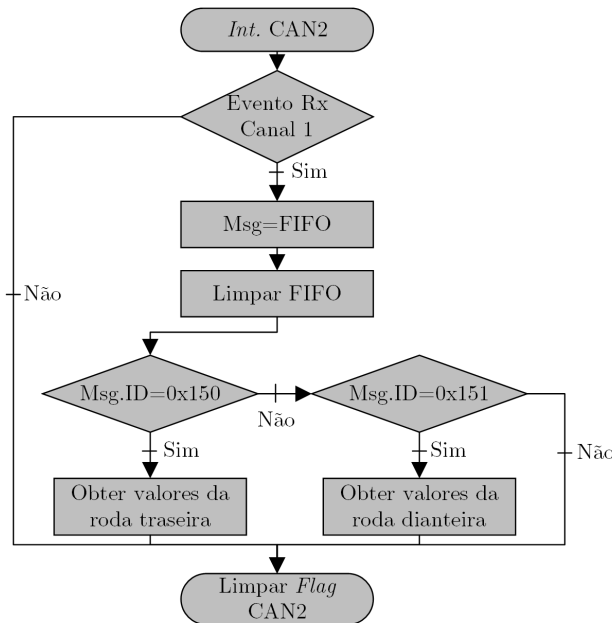


Figura 5.8: Rotina de interrupção associada ao controlador CAN2.

GPRMC, sendo este tipo definido pelo formato indicado na tabela 5.7. O formato deste tipo de mensagem tem um comprimento variável, sendo que os dados são separados por vírgulas e a mensagem contém sempre os mesmos caracteres no início e no fim, de modo a ser possível identificar mensagens com diferentes comprimentos.

Tabela 5.7: Exemplo de uma mensagem do tipo GPRMC.

<b>Exemplo:</b>	
\$GPRMC,013732.000,A,3150.7238,N,11711.7278,E,0.00,0.00,220413,,A*68<CR><LF>	
Parâmetro	Descrição
\$	Início de mensagem
GPRMC	Identificador de mensagem
Hora UTC	Hora no fuso horário UTC (hhmmss.sss)
Dados válidos	Indica validade de dados ('V'-inválido;'A'-válido)
Latitude	Latitude no formato graus e minutos (ddmm.mmmm)
Norte/Sul	Indicador da longitude (norte/sul)
Longitude	Longitude no formato graus e minutos (dddmm.mmmm)
Este/Oeste	Indicador da longitude (este/oeste)
Velocidade	Velocidade em nós (1 kn=1.852 km/h)
COG	Curso sobre o solo em graus
Data	Informação do dia, mês e ano (ddmmaa)
Varição magnética	Varição magnética em graus
Indicador Este/Oeste	Indicador da variação magnética este/oeste
Modo de posicionamento	Identifica o modo de posicionamento utilizado pelo módulo
*	Sinalização do fim dos campos com dados
Checksum	Valor do checksum associado aos dados recebidos
<CR><LF>	Fim de mensagem

O sistema ao ser inicializado declara duas variáveis globais, sendo elas um *array* de caracteres (*buffer*), outra é uma variável associada ao índice do *array*. Quando é iniciada a interrupção associada à recepção de dados pela interface UART5, é verificado se existem

dados para serem lidos e se a *flag* associada à interrupção está ativa. Se a condição não se verificar a *flag* é limpa e a rotina de interrupção é abandonada. Caso existam dados para serem lidos, o *array* de caracteres no índice (*i*) toma o valor do registo U5RXREG. Como não existe nenhuma mensagem emitida por este módulo com um carácter e também como são precisos dois caracteres para sinalizar o fim da mensagem, é efetuada uma verificação do valor atual do índice. Se o valor for superior ao valor 0 é verificado se o array já possui o fim de mensagem composto pela a sequência (<CR><LF>). Caso o valor do índice seja 0 ou se não tiver sido encontrado o fim de mensagem o valor do índice é incrementado, a *flag* associada à interrupção é reinicializada e a rotina é abandonada. Por fim caso seja detetado o fim da mensagem, esta é decodificada conforme o início da mesma, a variável associada ao índice e à *flag* da interrupção são reinicializadas e a interrupção é abandonada. O fluxograma associado a esta rotina está representado na figura 5.9.

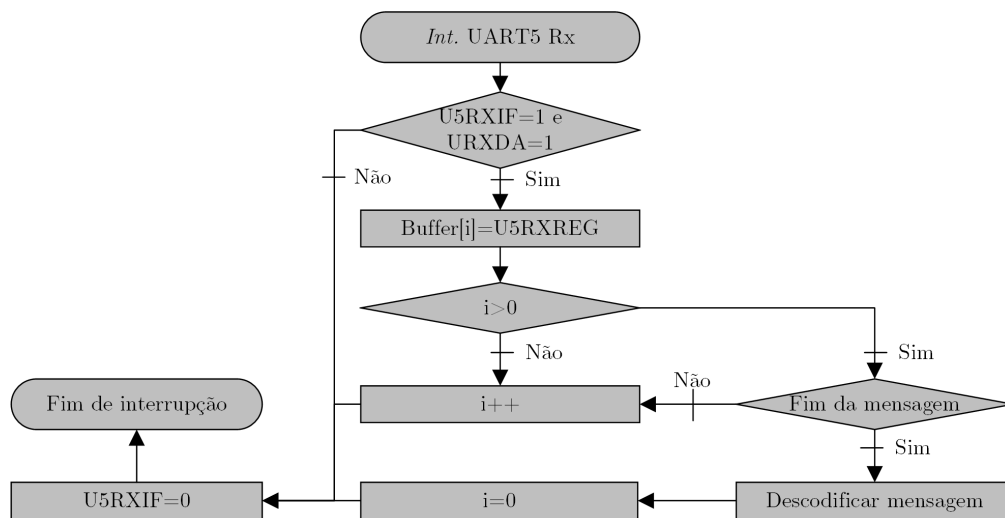


Figura 5.9: Rotina de interrupção na recepção de dados por UART5.

#### 5.4.3.4 Leitura do módulo inercial

Para o microcontrolador efetuar a leitura do módulo inercial, é utilizada a interrupção externa INT0 como referido na secção 5.4.2.1. O módulo inercial está preparado para ativar a sua saída digital, que está ligada ao pino RD0 do microcontrolador, quando este deteta um novo valor adquirido por um sensor. Ou seja se o sistema de aquisição de dados não se movimentar não é gerada nenhuma interrupção para efetuar uma nova leitura. Deste modo o microcontrolador só efetua uma nova leitura do módulo inercial quando é necessário, tornando assim o *firmware* mais eficiente.

Quando a interrupção é iniciada é efetuada uma leitura de cada sensor utilizado do módulo inercial, neste caso os valores associados ao acelerómetro e ao giroscópio. Para efetuar a leitura de cada sensor foram utilizadas as bibliotecas disponibilizadas pela *Microchip*. Após ser efetuada a leitura dos sensores, a *flag* associada à interrupção INT0 é reinicializada e a rotina é abandonada. O fluxograma associado a esta rotina está apresentado na figura 5.10.

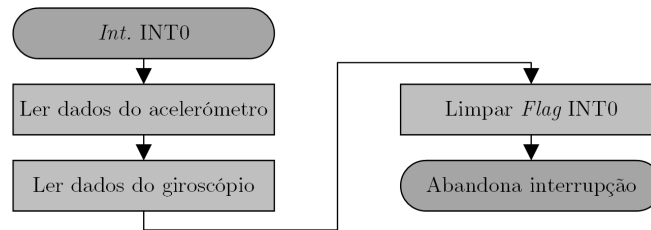


Figura 5.10: Rotina de interrupção associada ao módulo inercial.

#### 5.4.3.5 Interrupções associadas à transmissão de dados

A transmissão dos dados pelo módulo rádio está dividida em duas fases. A primeira fase está relacionada com a preparação da mensagem de dados para ser enviada. A segunda fase consiste em enviar a mensagem para o módulo rádio e dar a ordem de transmissão da mesma.

Na inicialização deste sistema é efetuada a primeira e única preparação da mensagem fora de interrupções. Quando o sistema se encontra no ciclo infinito esta tarefa é realizada pela interrupção INT1. Como referido anteriormente, a interrupção INT1 está configurada para ser despertada quando o módulo rádio gerar uma interrupção à sua saída associada ao pino DIO0, que indica ao microcontrolador que já foi enviada a última mensagem.

A mensagem a ser enviada é composta pelo número de bytes que a mesma ocupa, pelo endereço do recetor rádio e pelos dados propriamente ditos. Após a mensagem ser formada com os referidos parâmetros, o próprio módulo RF adiciona ainda o preâmbulo, a palavra de sincronização e o valor do CRC. O formato da mensagem rádio enviada está representada na figura 5.11.

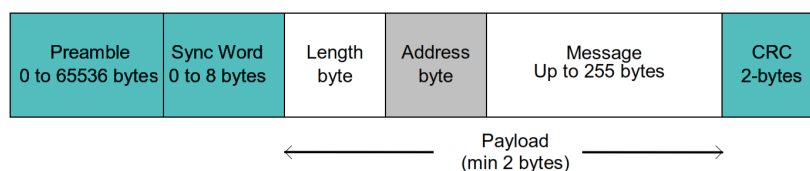


Figura 5.11: Formato da mensagem enviada pelo módulo RF.

Quando é iniciada a rotina de interrupção associada ao pino DIO0 do módulo rádio, contam-se o número de *bytes* que são utilizados para enviar toda a informação, de modo a formatar a mensagem que será enviada pelo módulo rádio. Com o número de bytes obtido é formatada a mensagem rádio. Com o processo de formatação dos dados concluído é asserrida uma variável global de controlo (*Rx\_ done*) com o valor 1, de seguida é desativada a interrupção INT1 e é ligado o LED5 para sinalizar que se encontra uma mensagem pronta para envio. Após todo o processo indicado, é limpa a *flag* associada à rotina da interrupção INT1. O fluxograma associado a esta rotina está representado na figura 5.12.

Tendo a mensagem pronta para ser enviada, só é necessário ser executada a ordem do envio da mesma. De modo e dar a ordem para efetuar o envio da mensagem é utilizada a interrupção associada ao temporizador (TMR2). Ao ser executada a rotina de interrupção é verificado se a mensagem já se encontra disponível para o envio através da variável global (*Rx\_ done*). Quando a mensagem não se encontra disponível para envio, é limpa a *flag* associada a interrupção e a mesma é abandonada. Já quando é detetado que a mensagem se encontra disponível é efetuado o envio da mesma para o

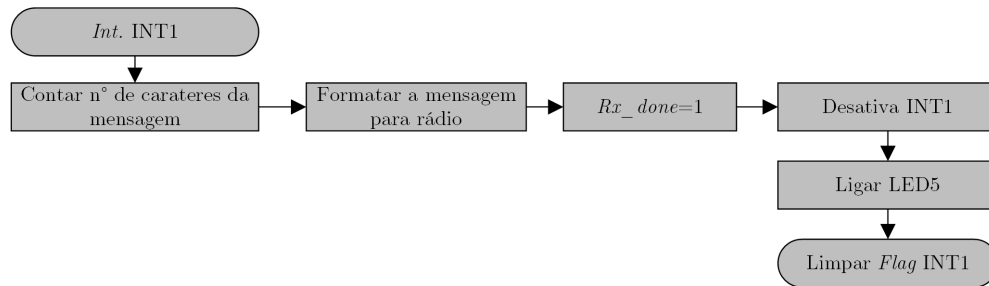


Figura 5.12: Rotina de interrupção para INT1.

módulo rádio. Quando o módulo recebe a mensagem ele adiciona o resto dos parâmetros necessários para completar a mesma e fica a aguardar a ordem para efetuar o envio. Assim que é dada a ordem para o envio da mensagem, a variável global de controlo (*Rx\_done*) é asserida com valor 0 e é ativada a interrupção INT1. Para finalizar a sinalização de uma mensagem pronta para envio é desligado o LED5 e a rotina de interrupção é abandonada. O fluxograma desta rotina está apresentado na figura 5.13.

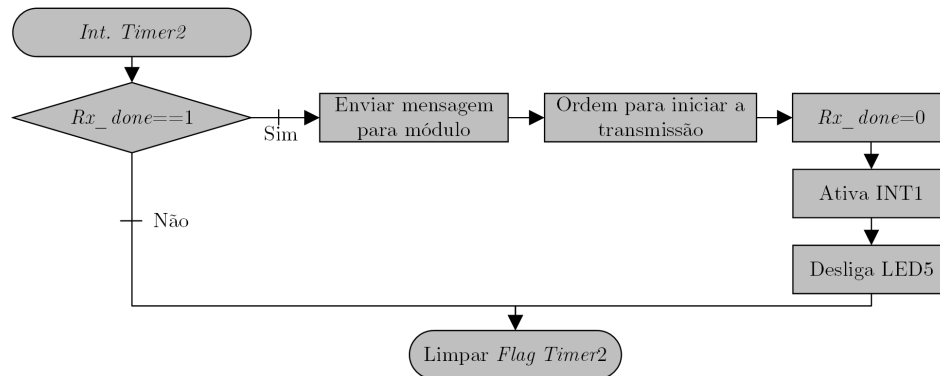


Figura 5.13: Rotina de interrupção para *Timer2*.

Por fim, assim que o módulo efetuar o envio da mensagem, volta a ser despertada a interrupção associada ao pino DIO0 e tudo é repetido. O processo é desencadeado até que o módulo seja desligado. Com esta forma garante-se que é enviada uma mensagem de cada vez, que não é enviado um maior número de mensagens do que as configuradas pelo utilizador e que no pior dos casos é enviada a informação com o tempo de atraso igual ou inferior ao valor do temporizador.

#### 5.4.3.6 Cronometrar tempo e guardar dados no cartão microSD

Todos os dados recolhidos são associados a um determinado instante de tempo. Para que o microcontrolador tenha uma referência de tempo é utilizado o temporizador (TMR3) que tem a função de atualizar um cronómetro interno no microcontrolador. Como referido anteriormente este temporizador está configurado para gerar uma interrupção à frequência de 40 Hz. Como os dados têm de ser guardados no cartão microSD a uma taxa de 20 Hz, são utilizadas três variáveis globais que garantem a recolha de dados para o cartão microSD.

Quando o sistema é ligado são declaradas várias variáveis globais. Uma delas é do tipo da estrutura referida na secção 5.2.2, outra é associada aos milissegundos decorridos e duas são para guardar os dados no ficheiro do cartão microSD. A estrutura de dados

possui três variáveis associadas ao tempo, mas como a atualização dos milissegundos é efetuada a cada 25 ms, a variável da estrutura só está preparada para trabalhar em múltiplos de 10 ms, logo é utilizada uma variável auxiliar para esta unidade de tempo.

Ao ser gerada a interrupção associada ao temporizador é feita a verificação do valor da contagem da variável auxiliar associada aos milissegundos decorridos. Caso este valor seja igual a 975, a variável é reinicializada e é feita a verificação do valor associado aos segundos guardado na estrutura de dados. Caso a verificação do valor dos segundos seja igual a 59, a mesma é reinicializada e o valor associado aos minutos é incrementado. Caso a verificação dos segundos seja inferior a 59, só é incrementada a variável associada aos segundos. Caso o valor dos milissegundos seja inferior a 975, a mesma é incrementada em 25 unidades. Deste modo é garantida a atualização de todas as variáveis associadas ao tempo.

Como os valores no cartão microSD são atualizadas a cada 50 ms e o temporizador está programado para gerar uma interrupção a cada 25 ms, é utilizada uma variável de controlo (*Sd\_aux*) para armazenar os dados a serem guardados no cartão à taxa de atualização pretendida. Caso a variável possua o seu valor igual a 0, esta é incrementada e a rotina de interrupção é abandonada. Quando a variável de controlo tem o valor 1 são armazenados os valores para serem guardados no cartão. Para reduzir o número de vezes que o sistema acede ao cartão microSD, foi criado um *array* para armazenar os dados até que o mesmo fique cheio. Isto permite que não seja gasto tanto tempo associado ao acesso do cartão. O *array* é composto por 4096 bytes, porque a biblioteca FATFS suporta blocos de dados até esse tamanho. Cada mensagem para o cartão microSD é composta por 42 bytes, sendo 36 bytes associados aos dados adquiridos e 6 caracteres para sinalizar o início e o fim da mensagem. Assim foi calculado o número de amostras que cabem dentro do *array* criado, sendo obtido o valor de 97 mensagens. Como apresentado na equação 5.20 ficam a faltar preencher 22 bytes para completar o *array* ao fim estarem armazenadas 97 mensagens. Deste modo quando o *array* tem armazenadas 97 mensagens, os últimos 22 bytes são preenchidos com o valor 0. É utilizada também uma variável que indica o número de mensagens armazenadas para o cartão microSD, de modo a que o sistema saiba quantas mensagens já estão armazenadas no *array*.

$$\text{Bytes não utilizados} = 4096 - (97 \times 42) = 22 \text{ Bytes} \quad (5.20)$$

Quando estão armazenadas as 97 mensagens, os dados anteriormente guardadas no *array* são adicionados ao final do ficheiro guardado no cartão. Para efetuar a escrita das mensagens no cartão é preciso abrir o ficheiro, guardar os dados e fechar o mesmo. Depois do ficheiro estar atualizado a variável associada às mensagens armazenadas no *array* é reinicializada e a interrupção é abandonada. Caso o número de mensagens seja inferior a 97, a mensagem é adicionada ao *array*, a variável associada ao número de mensagens armazenadas é incrementada e a interrupção é abandonada.

O fluxograma associado à rotina de interrupção do temporizador (TMR3) está apresentado na figura 5.14.

## 5.5 Sistema para a receção de dados

No desenvolvimento do *firmware* para o sistema de receção de dados foi tido em conta a necessidade de uma interface UART e de um barramento SPI.

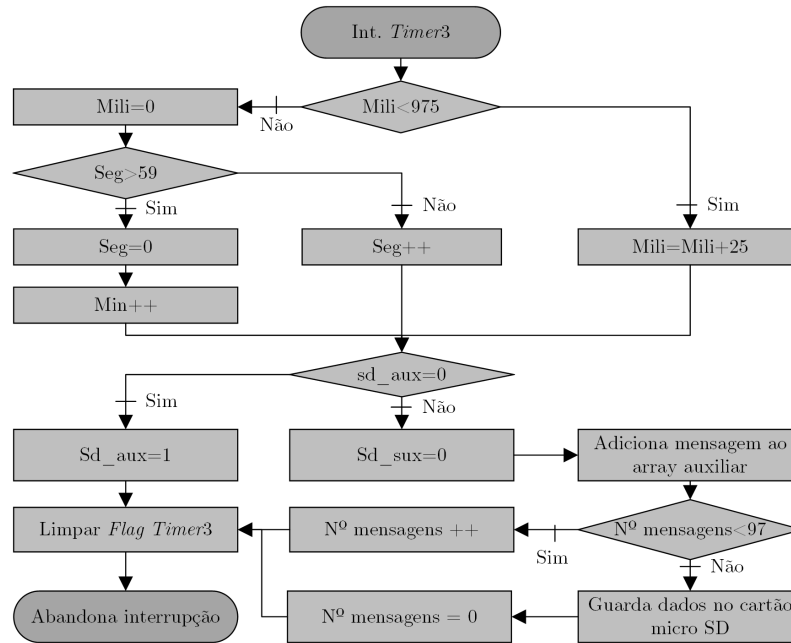


Figura 5.14: Rotina de interrupção para *Timer3*.

Este sistema está preparado para receber os dados transmitidos pela mota, numa banda ISM na gama dos 868 MHz, e enviar os mesmos para uma interface gráfica instalada num computador.

Esta secção começa por abordar a função *main* e a sequência da configuração da interface UART e do barramento SPI. De seguida são apresentadas as configurações efetuadas para cada componente utilizada, tendo em consideração os dispositivos utilizados. Por fim é apresentada a rotina de interrupção associada a este sistema.

### 5.5.1 Função *main* e inicialização das interfaces

O sistema quando é ligado começa por executar a função *main*. Esta função tem dois objetivos neste sistema, invocar a função de inicialização das interfaces utilizadas do microcontrolador e entrar num ciclo infinito. Dentro da função *main* não é realizada mais nenhuma operação além das indicadas anteriormente, em que todas as ações associadas ao sistema são executadas a partir da rotina de interrupção gerada pelo módulo rádio.

Na rotina de inicialização são configuradas as *interfaces* e os dispositivos associados ao sistema pela sequência apresentada no fluxograma da figura 5.15.

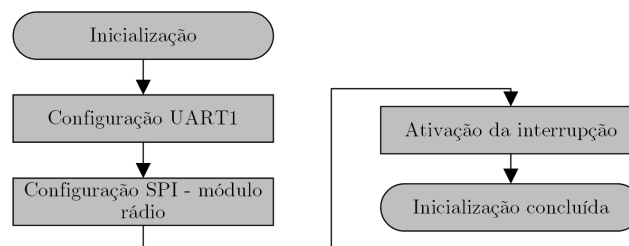


Figura 5.15: Fluxograma da rotina para inicialização das interfaces.



### 5.5.2 Configuração de *interfaces*

Tendo em conta os requisitos que têm de ser cumpridos pelo sistema de recepção de dados, nesta secção serão apresentadas as configurações aplicadas às interfaces e aos dispositivos deste sistema. As interfaces do microcontrolador e a sua respetiva função neste sistema são:

- UART1 envia os dados recebidos para uma interface gráfica,
- SPI2 estabelece a comunicação com o *transceiver* rádio.

#### 5.5.2.1 UART1

Os dados recebidos da mota, devem ser analisados em tempo real enquanto a mota está em testes. Deste modo é utilizada a interface UART1 para enviar os dados recebidos para um computador.

A quantidade de dados recebidos pode variar com o número de mensagens enviadas pelo transmissor. Isto é, como referido anteriormente, a equipa pode definir o número de mensagens enviadas pelo sistema de aquisição e transmissão de dados, logo este sistema está preparado para receber e transmitir para um computador até 20 mensagens por segundo. A mensagem enviada para um computador é formada pelo conjunto de 36 *bytes* recebidos pelo módulo rádio e por duas sequências de sinalização, a de início "INI" e a de fim "FIM" da mensagem. Tendo em conta que o comprimento da mensagem é de 42 *bytes* e que no máximo são recebidas 20 mensagens por segundo, é preciso ter uma taxa de transmissão de dados superior a 6720 bps.

Para configurar a interface UART1 de modo a efetuar a comunicação entre o microcontrolador e um computador, são configurados os registos (U1MODE), (U1BRG) e (U1STA) tal como foi apresentado anteriormente na secção 5.4.2.4.

Neste caso é preciso calcular o valor do registo U1BRG, pois o valor do *baudrate* é superior. Tendo em conta o valor do *baudrate* igual a 9600 bps, o valor obtido para o registo U1BRG é igual a 260 como mostra a equação 5.21.

$$U1BRG = \frac{PBCLK}{16 \cdot BaudRate} - 1 = \frac{40 \text{ MHz}}{16 \cdot 9600} - 1 \approx 260 \quad (5.21)$$

Tendo em conta todas as especificações referidas anteriormente, os registos associados a interface UART1 são configurados através do código fonte 5.15.

```
// Config UART1
U1MODEbits.PDSEL=0;           //8-bit data and no parity
U1MODEbits.STSEL=0;          //1 Stop bit
U1BRG=260;                    //Configure baudrate UART
U1STAbits.URXEN=0;           //Disable Rx channel
U1STAbits.UTXEN=1;           //Enable Tx channel
U1MODEbits.ON=1;             //Enable UART5 module
```

Código fonte 5.15: Código para configuração da interface UART1.

#### 5.5.2.2 SPI - *Transceiver* RFM95PW

Neste sistema é utilizada a interface SPI2 do microcontrolador para efetuar a receção dos dados pelo *transceiver* rádio RFM95PW. Como o *transceiver* é igual ao utilizado no sistema de aquisição, gravação e transmissão de dados, a interface SPI2 é configurada com os mesmos valores que estão referidos anteriormente na secção 5.4.2.2.

Para efetuar a receção de mensagens dos dados recolhidos na mota o módulo rádio tem que estar preparado para trabalhar com as mesmas configurações efetuadas no transmissor. As definições comuns aos dois sistemas estão relacionadas com as configurações

gerais e com as da manipulação dos pacotes de dados. Nas definições gerais são definidos os valores do modo de funcionamento, do *bitrate* das frequências da onda portadora e do desvio. Na manipulação dos pacotes de dados, as configurações efetuadas são iguais às do transmissor à exceção de que no recetor é também definido o nó do endereço do mesmo, com o valor 2.

Neste caso os registos do módulo associados ao transmissor não são configurados, assim só é necessário definir os ajustes do recetor. Os registos do recetor configuram o ganho do amplificador de baixo ruído (LNA) e o evento que inicia a receção de uma mensagem. Habilita também a deteção do preâmbulo e configura o valor da largura de banda de receção. Para definir o valor da largura de banda de receção para 200 kHz, o registo (RxBwExp) é igual a 1 como mostra a equação 5.22.

$$RxBwExp = \log_2 \left( \frac{\frac{Fosc}{RxBw}}{RxBwMant} \right) - 2 = \log_2 \left( \frac{\frac{32 \text{ MHz}}{200 \text{ kHz}}}{20} \right) - 2 = 1 \quad (5.22)$$

Como referido anteriormente, este módulo tem a possibilidade de configurar uma máquina de estados interna, para efetuar a receção de dados. Neste sistema a máquina de estados implementada no módulo rádio está definida para efetuar a receção de uma mensagem e ficar a aguardar a próxima. Depois de ser iniciada a sequência o emissor fica sempre no modo de receção de dados. O início da máquina de estados só é efetuado se o módulo se encontrar no modo de *standby*. Na figura 5.16 está apresentado o fluxograma associado à máquina de estados do módulo rádio, implementada neste sistema para efetuar a receção de dados.

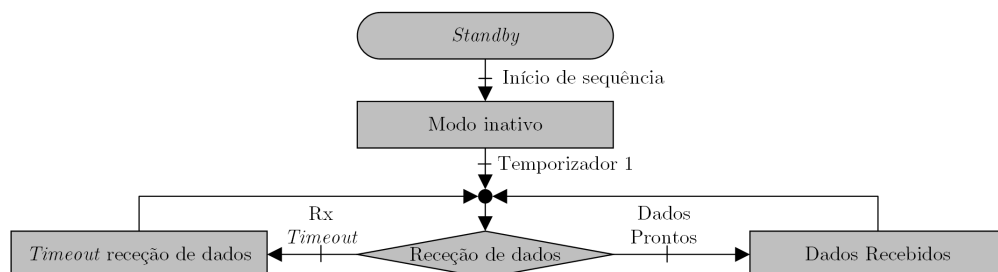


Figura 5.16: Máquina de estados implementada no módulo RF.

Para finalizar a configuração do módulo rádio é atribuído ao estado do pino DIO0 um sinal que indica quando um pacote está pronto para ser lido. Todas as configurações efetuadas aos registos internos do módulo estão apresentadas na tabela 5.8.

Tabela 5.8: Configurações efetuadas ao módulo RFM95PW.

Nome (Endereço)	Variável	Valor e descrição
Configurações gerais		
RegOpMode (0x01)	Modo de longo alcance	0 → Modo FSK/OOK
	Tipo de modulação	0 → FSK
	Modo de baixa frequência	0 → Frequência utilizada superior a 568MHz
	Modo	1 → Modo de <i>Standby</i>

RegBitrateMsb (0x02)	<i>Bitrate</i> (15:8)	320 →Para um <i>bitrate</i> 100 kbitseg
RegBitrateLsb (0x03)	<i>Bitrate</i> (7:0)	
RegFdevMsb (0x04)	Ajuste do desvio de frequência	820 →Desvio de frequência 50 kHz
RegFdevLsb (0x05)		
RegFrFMsb (0x06)	Ajuste da frequência da onda portadora	14252460 →Frequência da onda portadora 869,4Mhz
RegFrFMid (0x07)		
RegFrFLsb (0x08)		
Configurações do recetor		
RegLna (0x0C)	Ganho do amplificador de baixo ruído (LNA)	1 →Ganho mais elevado
	Ajuste de corrente LNA alta frequência	3 →Impulso ligado, corrente LNA 150 %
RegRxConfig (0x0D)	Controlo automático do ganho LNA	1 →Ganho é controlado automaticamente
	<i>Trigger</i> de recepção de pacote	6 →Quando é detetado o preâmbulo
RegRxBw (0x12)	RxBwMant	1 →RxBwMant=20 para (RxBw=200 kHz)
	RxBwExp	1 →RxBwExp=1 para (RxBw=200 kHz)
RegPreambleDetect (0x1F)	Deteção do preâmbulo do pacote	1 →Habilitado
	Tamanho do preâmbulo	2 →Preâmbulo constituído por 3 bytes
Configurações de manipulação de pacotes de dados		
RegPreambleMsb (0x25)	Tamanho do preâmbulo	15 →Preâmbulo da mensagem de 15 bytes
RegPreambleLsb (0x26)		
RegSyncConfig (0x27)	Reinício automático do modo Rx	1 →Ligado, sem esperar por o bloqueio PLL
	Sincronização	1 →Ligada
	Condição do preenchimento FIFO	0 →Na deteção da palavra de sincronização
RegSyncValue1 (0x28)	Primeiro byte de sincronização	Byte com o valor 0xAA
RegSyncValue2 (0x29)	Segundo byte de sincronização	Byte com o valor 0x5E
RegSyncValue3 (0x2A)	Terceiro byte de sincronização	Byte com o valor 0xD4
RegPacketConfig1 (0x30)	Formato do pacote	1 →Comprimento de pacote variável
	Verificação CRC habilitada	1 →Ligada
	Apagar FIFO na falha de CRC	0 →Apaga FIFO na falha de CRC
RegPacketConfig2 (0x31)	Filtragem por endereço	1 →Endereço igual ao do recetor
	Modo de transmissão de dados	1 →Por pacotes de dados
Configuração da máquina de estados		

RegSeqConfig1 (0x36)	<i>IdleMode</i>	0 → Início a partir do modo de standby
	<i>FromStart</i>	1 → Entra no modo de recepção
	<i>LowPowerSelection</i>	1 → Desliga a máquina de estados
RegSeqConfig2 (0x37)	<i>FromReceive</i>	1 → Pacote recebido quando interrupção PayloadReady
	<i>FromRxTimeout</i>	0 → Volta ao modo recepção
	<i>FromPacketReceived</i>	1 → Modo recepção
Configuração dos pinos digitais do módulo		
RegDioMapping1 (0x40)	Utilização do pino DIO0	0 → Mensagem pronta para ser lida

### 5.5.3 Rotina de interrupção

Neste sistema que efetua a recepção de dados, é utilizada uma só rotina de interrupção associada ao módulo de interrupções externas do microcontrolador, sendo a interrupção INT1 associada ao pino RB8 do microcontrolador.

Esta interrupção é gerada quando a entrada digital do pino RB8, altera o estado lógico de zero para um. Este sinal digital é emitido pelo *transceiver* RFM95PW. Como referido anteriormente o *transceiver* está configurado para quando este tiver uma mensagem pronta a ser obtida, acionar o seu pino digital DIO0. Ou seja, quando o *transceiver* recebe uma mensagem indica ao microcontrolador que existem dados a serem obtidos, logo é iniciada a rotina de interrupção INT1.

A rotina de interrupção começa por efetuar a leitura da mensagem armazenada na memória do *transceiver*. A função utilizada para efetuar a leitura da mensagem retorna um valor que indica se a mensagem que foi obtida é válida ou se ocorreu um evento de *timeout*. Caso a mensagem seja válida é criada uma *string* que contém os dados recebidos pelo *transceiver* e duas sequências de sinalização, sendo elas a de início "INI" e a de fim "FIM". Depois de ser obtida a *string* com a mensagem pronta a ser enviada para um computador, utiliza-se a interface UART1 para efetuar a transmissão de dados. O fluxograma associado a esta rotina está apresentado na figura 5.17.

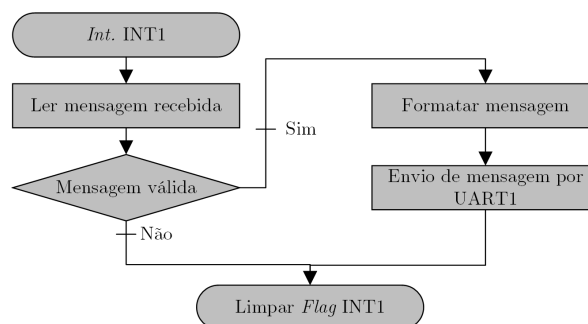


Figura 5.17: Rotina de interrupção para INT1.

A rotina ao ser abandonada, deixa o microcontrolador em estado de espera até ser gerada uma nova interrupção.

Após a apresentação das configurações e das rotinas de interrupção implementadas na solução, a próxima secção apresenta os testes realizados à mesma de modo a verificar o seu correto funcionamento.

## Secção 6

# Testes e validações

Esta secção está dividida em três partes e tem como objetivo apresentar os testes realizados à solução desenvolvida e os respetivos resultados. A primeira parte apresenta as ferramentas utilizadas para realizar os testes, a segunda parte mostra os testes realizados às componentes da solução individualmente e a terceira parte apresenta o teste realizado com a mota em pista.

Nas ferramentas utilizadas é apresentado o osciloscópio e a interface gráfica desenvolvida para realizar parte dos testes. Os testes individuais às componentes da solução são abordados pela seguinte ordem, primeiro aquisição, segundo a gravação e terceiro a transmissão de dados. Os testes individuais tem como objetivo verificar as características da solução em relação aos valores adquiridos e à sua taxa de atualização bem como testar todas as comunicações utilizadas. Após a apresentação dos testes individuais são apresentadas as características associadas ao peso e dimensão das partes constituintes da solução.

O último teste efetuado à solução é o que verifica o funcionamento do conjunto de todas as componentes desenvolvidas, sendo este realizado com a solução a funcionar num ambiente real.

### 6.1 Ferramentas de teste utilizadas

Para efetuar os testes aos diferentes componentes da solução foram utilizadas duas ferramentas que permitiram obter os resultados apresentados nesta secção. De modo a efetuar o teste aos diferentes tipos de comunicação e do funcionamento dos dispositivos utilizados na solução desenvolvida, foi utilizado um osciloscópio digital com a capacidade de efetuar descodificação dos sinais emitidos por cada tipo de comunicação. Já para efetuar o teste da transmissão dos dados entre o sistema instalado na mota para um recetor ligado a um computador foi desenvolvida uma interface gráfica para a solução.

#### 6.1.1 Instrumento de medida

Para realizar parte dos testes à solução foi utilizado um osciloscópio de pequenas dimensões mas com bastante versatilidade. A ferramenta utilizada é o produto *PicoScope 2207B* produzido pela *Pico Technology*. Este osciloscópio requer sempre o uso de um computador com o *software* desenvolvido pelo fabricante. Todas as funcionalidades associadas a esta ferramenta são acessíveis através do software. Na figura 6.1 está apresentado o osciloscópio *PicoScope 2207B* e o *software* requerido para o uso do mesmo.

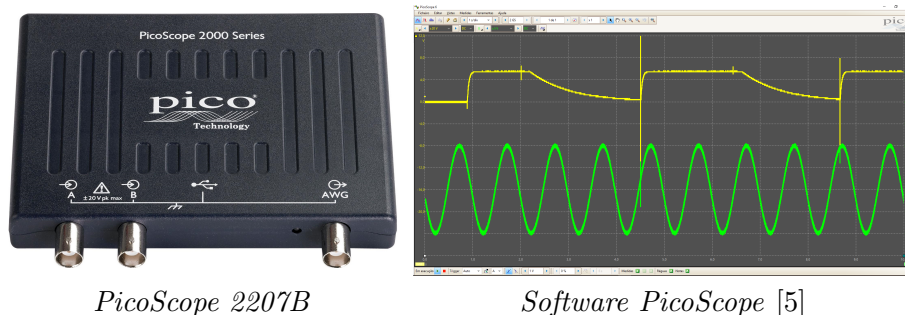


Figura 6.1: Osciloscópio *PicoScope 2207B* e o respetivo *software*.

O osciloscópio *PicoScope 2207B* é um equipamento compacto de baixo custo em que o mesmo é alimentado por uma porta USB. Este osciloscópio tem diversas ferramentas embutidas tais analisador de espectro, gerador de sinais, gerador de ondas arbitrárias e decodificador diversos protocolos. As principais características desta ferramenta são apresentadas na tabela 6.1.

Tabela 6.1: Especificações do osciloscópio *PicoScope 2207B* [94].

PicoScope 2207B	
Faixa	70MHz
Número de canais	2
Resolução do ADC	8 bits
Intervalos de entrada	$\pm 20$ mV, $\pm 50$ mV, $\pm 100$ mV, $\pm 200$ mV, $\pm 500$ mV, $\pm 1$ V, $\pm 2$ V, $\pm 5$ V, $\pm 10$ V, $\pm 20$ V
Acoplamento de sinal	CA/CC
Precisão CC	$\pm 3\%$ fim de escala
Taxa de amostras USB	9,6MS/s
Descodificação série	1-Wire, ARINC 429, CAN, CAN FD, DALI, DCC, DMX512, FlexRay, USB 1.1, I <sup>2</sup> C, I <sup>2</sup> S, LIN, Manchester, MODBUS, PS/2, SPI, UART/RS-232

### 6.1.2 Interface gráfica desenvolvida

No decorrer do desenvolvimento de toda a solução surgiu a necessidade de efetuar a visualização dos dados enviados pelo sistema de aquisição de dados. Nas primeiras comunicações realizadas entre os dois sistemas, os dados recebidos eram verificados por um terminal série instalado num computador, sendo que os dados recebidos vinham codificados tal como a estrutura de dados apresentada na secção 5.2.2. De modo a ser possível verificar as grandezas recolhidas foi desenvolvida uma interface que ao receber os dados do sistema recetor descodifica a mensagem transmitida.

O desenvolvimento da interface para descodificação dos dados recebidos pela solução foi desenvolvida no *software* de programação *QT Creator*, sendo que já era do conhecimento prévio do autor o desenvolvimento de aplicações neste *software* [95].

A interface gráfica começou por ser desenvolvida para efetuar uma comunicação série com o sistema recetor de dados da solução e realizar a gravação dos dados para um ficheiro de texto. Com a interface a receber e a guardar os dados, de seguida foram associados os valores recebidos a variáveis associadas a cada grandeza adquirida pela solução. Para à aplicação também efetuar a representação de valores graficamente foram tidas em conta algumas funções adicionais, tais como habilitar a vista da variável num gráfico,

alterar a cor do traço no mesmo e habilitar uma vista dinâmica da variável associada às coordenadas GPS. Após o desenvolvimento desta parte o aspeto da aplicação está representado na figura 6.2.















Ícone	Descrição
	Botão para criar novo perfil
	Botão para abrir perfil existente
	Botão para guardar perfil aberto
	Visualizar variável em escala de cor na pista
	Botão para alterar a cor da variável no gráfico
27 km/h	Indicador da grandeza adquirida pela solução
	Botão para habilitar visualização no gráfico
	Botão para iniciar a gravação de dados
	Botão para parar a gravação de dados
	Botão para efetuar ligação com a solução
	Indicador do estado da ligação com a solução
	Indicador da receção de dados
	Indicador se estão a ser guardados os dados

Figura 6.2: Tabela de variáveis da interface gráfica.

De modo a facilitar a análise de dados recolhidos durante o funcionamento da solução, foram implementados gráficos de visualização dos dados em tempo real, uma janela de modo a visualizar a localização da mota em pista, manómetros a indicar a velocidade e as RPM da mota, um gráfico a representar as acelerações e as inclinações da mota, uma tabela com os tempos por volta, barras para indicação da pressão nos travões, temperatura dos pneus, posição do embolo das suspensões e do acelerador. O aspeto final da interface gráfica desenvolvida está representada na figura 6.3.

A interface também possui algumas funcionalidades de análise de dados em modo *offline*, isto é tem a possibilidade de abrir os ficheiros guardados no computador ou então importar os dados escritos no cartão de memória. Para facilitar a análise dos dados por volta, a interface também possui uma lista de seleção das voltas registadas para representar graficamente os dados recolhidos por volta.

Esta ferramenta além de ter permitido recolher dados dos testes realizados à solução, também está preparada para que a equipa de futuro possa retirar vantagem do uso desta solução e implementar melhorias na mota.

## 6.2 Testes individuais às funções da solução

Nesta secção são apresentados os testes realizados individualmente a cada componente da solução. Os testes tem como objetivo verificar a qualidade dos dados adquiridos,



Figura 6.3: Aspeto final da interface gráfica.

as comunicações realizadas em toda a solução, testar algumas componentes para verificar se suportam maiores taxas de transmissão de dados de modo a que a solução esteja preparada para melhorias futuras.

Para cada componente testada é apresentado o método de como foram realizados os testes bem como os resultados de cada um. Ao longo desta secção são obtidas algumas das características técnicas da solução. É importante referir que existem componentes utilizados para os quais os fabricantes dos mesmos garantem certas características, logo não são realizados testes a erros e as especificações, tal como por exemplo a precisão da aquisição dos dados GPS.

### 6.2.1 Leitura dos sensores analógicos

No módulo de sensores adicionais como já referido são adquiridos duas entradas analógicas, uma para um sensor de pressão e outra para um potenciómetro linear. De modo a efetuar o teste ao acondicionamento de sinal foi utilizado um potenciómetro para simular uma entrada analógica de  $[0 ; 10]$  V e verificar a saída do circuito de acondicionamento de sinal entre os  $[0 ; 3.3]$  V. Neste teste utilizou-se o osciloscópio, em que o canal A mede o sinal colocado à entrada do ADC do microcontrolador e o canal B mede o sinal à saída do potenciómetro. Na figura 6.4 está apresentada a ligação efetuada para realizar o teste ao circuito de acondicionamento de sinal.

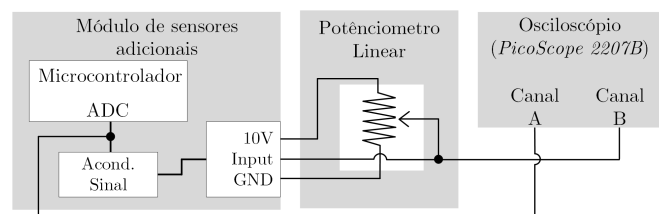


Figura 6.4: Ligação efetuada para o teste do acondicionamento de sinal.

O teste ao funcionamento do acondicionamento de sinal tem como objetivo verificar se o sinal colocado a entrada do ADC corresponde ao dimensionamento referido na secção



4.2.1. Pelo *hardware* desenvolvido é de esperar que o sinal à saída do acondicionamento tenha relação com o valor emitido pelo o sensor como representado na equação 6.1. De modo a verificar o correto funcionamento do acondicionamento de sinal foi utilizada uma ferramenta do osciloscópio, que permite representar um sinal resultante de um dos canais com uma expressão matemática. Assim é possível comparar o valor que está à saída do circuito de acondicionamento de sinal, com o valor esperado. Para verificar a resposta do acondicionamento de sinal foi efetuada uma captura do osciloscópio durante um período de 10 s.

Na figura 6.5 está apresentado a captura de sinais efetuada com o osciloscópio durante o teste.

$$V_o = V_{Ref} \cdot \left(1 + \frac{3.3k}{10k}\right) - V_i \cdot \left(\frac{3.3k}{10k}\right) = 2.48 \cdot (1.33) - V_i \cdot (0.33) \quad (6.1)$$

Através da comparação entre o sinal à saída do acondicionamento de sinal e o esperado,

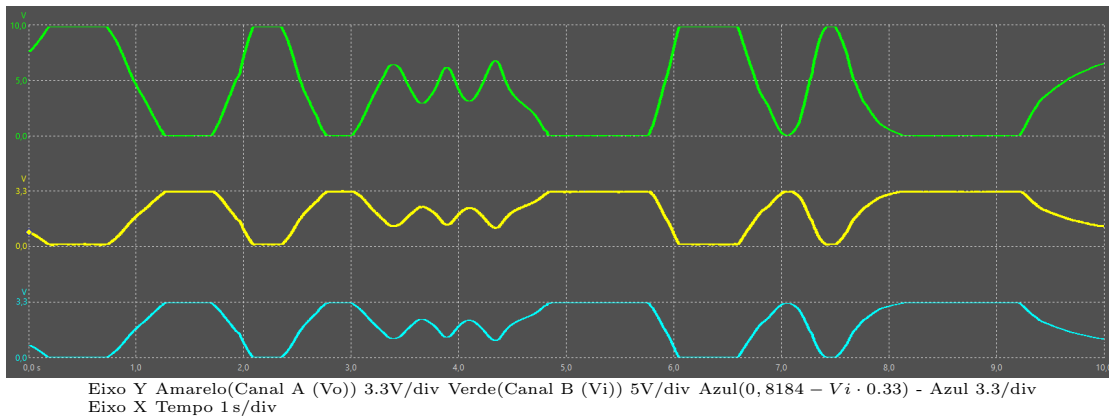


Figura 6.5: Sinais obtidos durante o teste ao acondicionamento de sinal.

verifica-se que a resposta do circuito é idêntica à esperada. Também é possível verificar que o acondicionamento de sinal trabalha para todos os valores emitidos pelo sensor no intervalo  $[0 ; 10]$  V. Com o teste realizado conclui-se que todo o acondicionamento de sinal funciona como o esperado.

## 6.2.2 Leitura de temperaturas através do pirómetro

As temperaturas dos pneus da mota nesta solução são adquiridas através de pirómetros tal como referido na secção 3.4. Tendo em conta que o fabricante já indica as especificações associadas à resolução e aos erros de medida, foram efetuados dois testes para este dispositivo que verificam a comunicação entre o pirómetro e o microcontrolador. Para efetuar os testes à comunicação com o pirómetro foi utilizado o osciloscópio, com o canal A e B associados aos canais SCL e SDA respetivamente do barramento I<sup>2</sup>C2 do microcontrolador, como mostra a figura 6.6.

O primeiro teste efetuado verifica o funcionamento da comunicação pelo protocolo I<sup>2</sup>C entre o microcontrolador e o pirómetro, sendo realizadas duas consultas da temperatura medida pelo o pirómetro. Neste teste é realizada a leitura das temperaturas guardadas nos endereços 0x07 e 0x08 do pirómetro, que correspondem aos dois pontos de leitura do sensor. Na figura 6.7 são representados os sinais capturados pelo osciloscópio durante o primeiro teste, sendo que os dados associados aos pacotes recolhidos estão apresentados na tabela 6.2. No anexo E.1.1 estão apresentados todos os pacotes

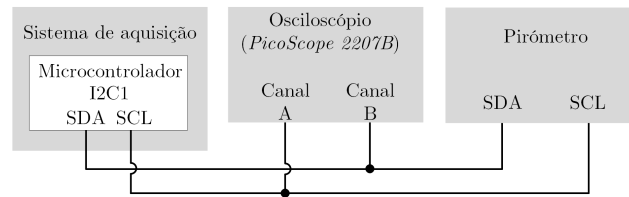


Figura 6.6: Ligação efetuada para o teste da comunicação com o pirómetro.

capturados pelo osciloscópio durante o primeiro teste realizado. É possível verificar que por cada consulta realizada ao sensor são transmitidos dois pacotes, sendo um de escrita e outro de leitura. O primeiro pacote indica ao sensor que pretende ser efetuada a leitura dos dados guardados no endereço 0x07 e o segundo pacote transmitido contém os dados obtidos pelo sensor. É verificado que cada consulta realizada demora cerca de 1.1 ms.

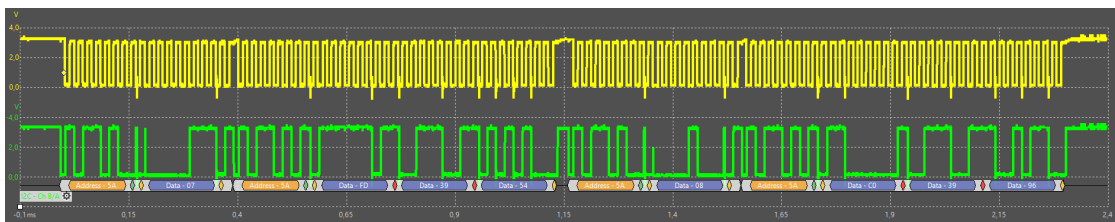


Figura 6.7: Comunicação I2C com o pirómetro para leitura das temperaturas.

Tabela 6.2: Dados do teste à comunicação com o pirómetro.

Pacote	Bits de endereço	Endereço	Leitura/ Escrita	Dados	Duração (ms)	Duração por leitura (ms)
1	7	5A	Escrita	7	0,398	1,142
2	7	5A	Leitura	FD 39 54	0,744	
3	7	5A	Escrita	8	0,398	1,142
4	7	5A	Leitura	C0 39 96	0,744	

O segundo teste realizado ao pirómetro tem como objetivo verificar a taxa de aquisição das temperaturas dos pneus. Durante este teste foram consultados os dois endereços do pirómetro que correspondem aos dois pontos de leitura do sensor. De modo a facilitar o tratamento de dados recolhidos no teste, este foi realizado durante o período de 10s. No decorrer do teste foram realizadas 20 leituras por segundo dos endereços 0x07 e 0x08, sendo que o tempo médio por cada leitura foi de 1.142 ms. O resumo dos dados recolhidos está apresentado na tabela 6.3, sendo que os mesmos estão apresentados no anexo E.1.2. Tendo em conta os resultados obtidos é verificado que a leitura da temperatura dos pneus é efetuada a uma taxa de aquisição superior à requerida pela equipa.

### 6.2.3 Leitura da ECU

Como referido na secção 5.4.2.1, a aquisição de variáveis a partir da ECU da mota é feita através de pedidos de informação realizados à mesma. A aquisição de dados é efetuada por três fases, sendo elas a do pedido, a do processamento do mesmo pela ECU e o envio da resposta ao pedido realizado. De modo a verificar a aquisição das grandezas associadas ao funcionamento do motor, realizaram-se dois testes ao barramento CAN dedicado à aquisição de dados a partir da ECU. Para a realização dos dois testes indicados

Tabela 6.3: Dados do teste à taxa de aquisição ao pirómetro.

Endereço consultado	Pacote de consulta		Pacote de leitura		Duração média (ms)	Taxa de atualização (hz)
	Enviados	Duração (ms)	Recebidos	Duração (ms)		
0x07	200	0,398	200	0,744	1,142	20
0x08	200	0,398	200	0,744	1,142	20

utilizou-se o osciloscópio indicado na secção 6.1, da qual os canais A e B do mesmo estão conectados respetivamente aos canais CAN-H e CAN-L do barramento CAN, como indicado na figura 6.8.

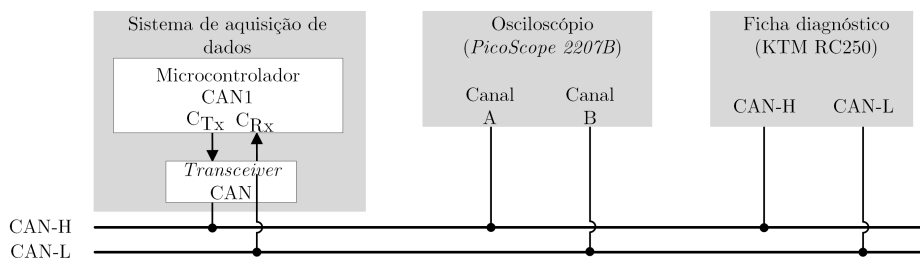


Figura 6.8: Ligação efetuada para realização de testes ao barramento CAN1.

O primeiro teste analisa a aquisição de uma grandeza da ECU, sendo realizada a troca de dois pacotes pelo barramento CAN. Neste caso foi analisado um pedido efetuado para obter a leitura da posição do acelerador. Na figura 6.9, são representados os pacotes adquiridos pelo osciloscópio durante o processo de leitura da grandeza referida anteriormente. Os dados dos dois pacotes que foram capturados no decorrer deste teste estão apresentados na tabela 6.4. Com os dados associados aos pacotes é possível verificar os tempos das diferentes fases do processo.

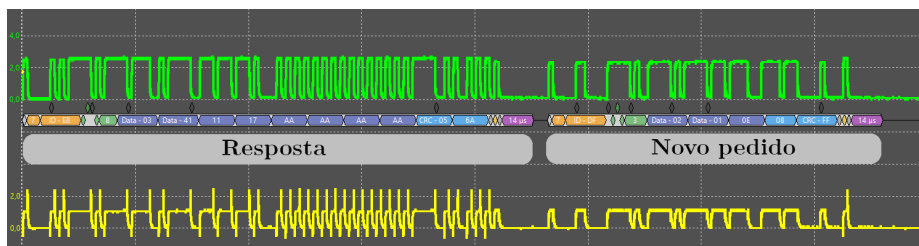


Figura 6.9: Exemplo de um conjunto de dados obtidos do barramento CAN1.

Tabela 6.4: Pacotes recolhidos no primeiro teste executado ao barramento CAN1.

Pacote	Início (ms)	Fim (ms)	Duração (ms)	ID	DLC	Dados	CRC	CRC (válido)	Pacote válido
1	0,234	0,382	0,148	7 DF	3	02 01 11	02 92	1	1
2	10,009	10,235	0,226	7 E8	8	03 41 11 16 AA AA AA AA	62 82	1	1

O tempo utilizado para enviar um pedido à ECU é de aproximadamente 0.15 ms, o processamento por parte da mesma demora cerca de 9.6 ms e o pacote que contém a informação requerida demora 0.23 ms para ser transmitido pelo barramento. No total este pedido demorou cerca de 10 ms.

De modo a verificar a existência de possíveis falhas na transmissão de pacotes, obter a taxa de aquisição e calcular um valor médio da aquisição para cada variável requerida à ECU, realizou-se uma segunda captura de pacotes no barramento CAN durante um maior período de tempo. Como os dados recolhidos pelo osciloscópio têm de ser tratados, de modo a obter a quantidade de dados necessária para obter informações válidas e de modo a facilitar o trabalho de análise, foi definido um período de captura de 10s. Dos dados capturados é apresentada a tabela 6.5 que resume a informação adquirida durante o período de tempo indicado anteriormente.

Tabela 6.5: Resumo dos pacotes capturados no barramento CAN1 durante 10s.

Parâmetro	Pacotes de pedidos		Pacotes com respostas		Duração média pedido mais resposta (ms)	Taxa de atualização (Hz)
	Enviados	Válidos	Recebidas	Válidas		
Temp. de refrigeração	40	40	40	40	9,991	4
Pressão na admissão	200	200	200	200	9,933	20
Velocidade	100	100	100	100	9,953	10
Avanço na ignição	200	200	200	200	9,970	20
Temp. do ar de admissão	60	60	60	60	9,944	6
Posição do acelerador	200	200	200	200	9,966	20
RPM	200	200	200	200	9,952	20

Tendo em conta os valores apresentados são verificados vários acontecimentos. É possível conferir que todos os pacotes enviados pelo barramento CAN são transmitidos sem qualquer tipo de falha, pois por cada pedido enviado pelo o sistema de aquisição de dados é obtida uma resposta por parte da ECU. Também se verifica que o tempo de processamento de um pedido é de aproximadamente 10 ms. Com o teste é verificada a taxa de atualização obtida para cada variável proveniente da ECU. Todos os dados recolhidos estão contidos em formato digital no anexo E.2.1.

Com os dois testes realizados é possível verificar o correto funcionamento do controlador CAN1 do sistema de aquisição de dados e que a ligação física entre o sistema e a ECU se encontra funcional sem perda de dados.

#### 6.2.4 Recepção de dados dos módulos de sensores adicionais

Os dados obtidos pelos os módulos de sensores adicionais, são enviados por um barramento dedicado para sensores que não se encontrem instalados na mota. Este barramento, como referido na secção 5.4.2.2, está associado ao controlador CAN2 do sistema de aquisição de dados. Neste caso não são realizados pedidos de informação aos módulos de sensores adicionais, sendo apenas realizada a recepção de mensagens emitidas por eles.

De modo a verificar o correto funcionamento da comunicação e a capacidade máxima de pacotes transmitidos por este barramento foram realizados três testes. Tal como os testes apresentados na secção 6.2.3, os canais do osciloscópio foram conectados com a mesma ordem ao barramento de sensores, como mostra a figura 6.10.

O primeiro teste verifica a transmissão dos pacotes pelos dois módulos de sensores adicionais para o sistema de aquisição de dados. Neste caso foram analisados dois pacotes, sendo enviado um por cada módulo existente no barramento CAN. Os sinais recolhidos dos canais associados ao barramento CAN durante a transmissão de dois pacotes estão apresentados na figura 6.11. Os dados recolhidos durante a captura indicada estão apresentados na tabela 6.6. É verificado que cada pacote demora cerca de 222  $\mu$ s

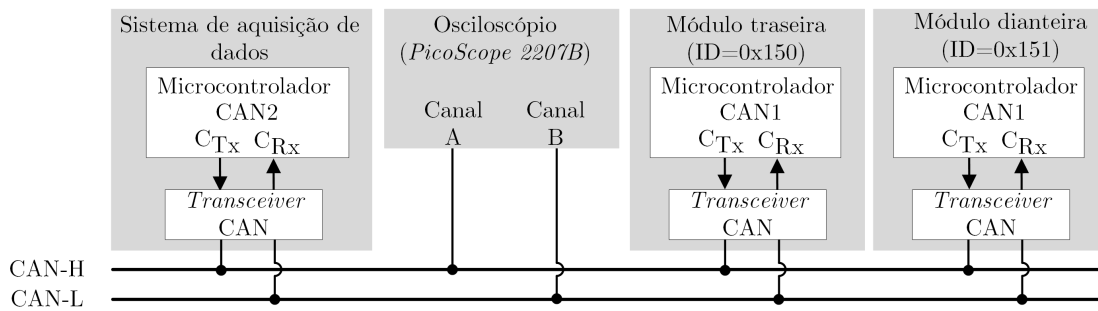


Figura 6.10: Ligação efetuada para realização de testes ao barramento CAN2 do sistema de aquisição.

e que é apresentado em primeiro o pacote com o identificador 0x150, porque este tem uma maior prioridade sobre 0x151, tendo em conta as características do funcionamento de um barramento CAN apresentadas anteriormente na secção 2.3.1.4.

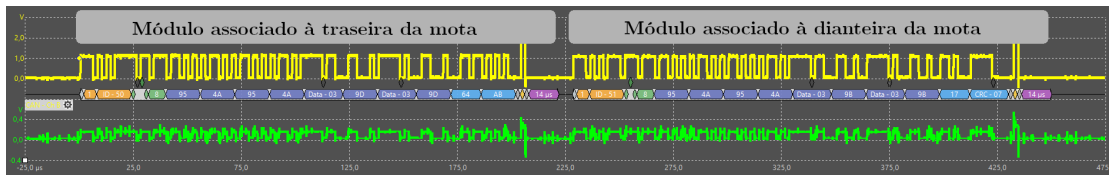


Figura 6.11: Pacotes transmitidos pelo barramento dedicado aos sensores adicionais.

Tabela 6.6: Pacotes recolhidos no primeiro teste executado ao barramento CAN2 .

Pacote	Início (ms)	Fim (ms)	Duração (ms)	ID	DLC	Dados	CRC	CRC (válido)	Pacote válido
1	0,000	0,222	0,222	1 50	8	95 4A 95 4A 03 9D 03 9D	64 AB	1	1
2	0,228	0,450	0,222	1 51	8	95 4A 95 4A 03 9B 03 9B	17 07	1	1

De modo a verificar a troca de pacotes entre os módulos e o sistema de aquisição, nas condições do funcionamento da aplicação final, foi feita uma captura durante 10s para ser adquirida a quantidade de dados necessária para obter informações válidas e de modo a facilitar o trabalho de análise. Os dados recolhidos durante este teste estão apresentados no anexo E.3.1. Na tabela 6.7 é apresentado o resumo dos dados adquiridos durante a captura referida anteriormente. É possível verificar que cada módulo envia 200 pacotes durante 10 segundos, logo o sistema de aquisição de dados tem uma taxa de atualização para estas variáveis de 20 Hz, tal como era pretendido. Também é verificado que o tempo de transmissão para cada pacote tem uma duração aproximada de 220  $\mu$ s.

Tabela 6.7: Resumo da captura efetuada no barramento CAN2.

Módulo	ID	Pacotes recebidos	Pacotes válidos	Pacotes por segundo	Duração média da transmissão ( $\mu$ s)
Sensores da traseira da mota	0x150	200	200	20	222,753
Sensores da dianteira da mota	0x151	200	200	20	223,237

Como este barramento também está desenvolvido de forma a suportar expansões futuras do número de módulos de sensores adicionais, foi efetuado um teste para verificar

a quantidade máxima de pacotes que é possível transmitir no barramento por segundo. Para executar este teste, os dois módulos do sistema efetuam o envio de pacotes com um determinado intervalo de tempo. Para definir o período entre cada pacote transmitido, foi tido em conta o resultado obtido para a duração da transmissão de um pacote no teste anterior, sendo este intervalo definido com o valor de 440  $\mu$ s. Os dois módulos para este teste só diferem do identificador enviado para o sistema de aquisição de dados. Este teste foi realizado durante o período de 2s, para facilitar a análise dos dados recolhidos.

O teste efetuado, verifica que este barramento com as configurações indicadas na secção 5.4.2.1, consegue suportar a transmissão de aproximadamente 4300 pacotes de dados por segundo. Os dados recolhidos durante este teste estão apresentados no anexo E.3.2. O tempo de transmissão de pacotes, em média mantém-se igual ao teste realizado em condições de funcionamento normais. Também é verificado que o pacote com o valor do identificador mais baixo é enviado através do barramento mais vezes devido a sua prioridade. A tabela 6.8 apresenta o resumo do teste realizado.

Tabela 6.8: Resumo do teste ao número máximo de pacotes enviados pelo barramento CAN2.

Módulo	ID	Mensagens recebidas	Mensagens válidas	Mensagens por segundo	Duração média da transmissão ( $\mu$ s)
Sensores da traseira da mota	0x150	4760	4760	2380	0,222
Sensores da dianteira da mota	0x151	4008	4008	2004	0,221
Total		8768	8768	4384	0,222

Tendo em conta os valores obtidos nos três testes realizados, é possível concluir que o barramento dimensionado para a transmissão de dados provenientes dos módulos de sensores adicionais suporta os requisitos pretendidos para a solução, como também se encontra preparado para receber cerca de 21 vezes mais a quantidade de pacotes transmitidos atualmente.

### 6.2.5 Leitura do módulo inercial

As acelerações e os ângulos de Euler a que a mota está sujeita são dados adquiridos através do módulo inercial integrado no sistema de aquisição de dados referido na secção 4.3.2. Tendo em conta que as especificações associadas à resolução e aos erros de medida são disponibilizadas pelo fabricante, foram efetuados dois testes à comunicação entre o módulo e o microcontrolador. Para realizar os testes foi utilizado o osciloscópio com os canais A e B associados aos pinos SCL e SDA da interface I<sup>2</sup>C2 do microcontrolador, como mostra a figura 6.12.

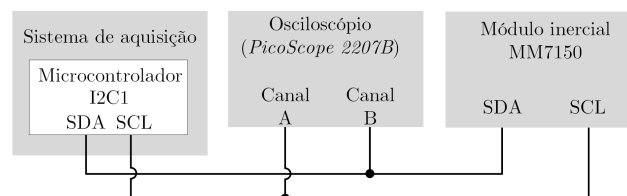


Figura 6.12: Ligação efetuada para o teste da comunicação com o módulo inercial.

O primeiro teste tem como objetivo analisar os dados transmitidos durante a leitura do acelerómetro e do giroscópio. Com este teste é possível verificar o endereço do módulo inercial, o comprimento do pacote de dados e a duração da transmissão dos mesmos por

cada sensor consultado. Neste caso foi analisado um pacote associado ao acelerómetro e outro ao giroscópio. Os dados recolhidos pelo osciloscópio durante a captura indicada estão apresentados na figura 6.13, sendo que os dados associados à captura efetuada estão representados na tabela 6.9. É verificado que o endereço do módulo inercial é igual ao valor 0x40 e que para efetuar a leitura de cada sensor só é realizada a transmissão de um só pacote. Cada sensor tem um comprimento de dados específico, sendo que o pacote associado à leitura do acelerómetro tem 12 *bytes* de dados enquanto o pacote do giroscópio é formado por 11 *bytes*.

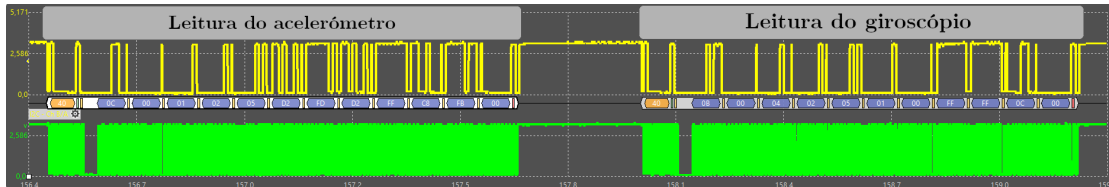


Figura 6.13: Sinais capturados durante a leitura do módulo inercial.

Tabela 6.9: Dados do teste à leitura do módulo inercial.

Pacote	Duração (ms)	Endereço	Endereço + R/W	R/W	Dados
1	1,50437	40	81	Read	0C 00 01 02 05 EB FF CB FF D1 FB 00
2	1,16902	40	81	Read	0B 00 04 02 05 02 00 FF FF 5F 01

O segundo teste tem como objetivo verificar a taxa de leitura de cada sensor associado ao módulo inercial. Este teste foi realizado durante um intervalo de tempo de 10 segundos, de modo a facilitar o tratamento dos dados recolhidos. No anexo E.4.1 estão apresentados todos os dados capturados durante o teste, sendo que na tabela 6.10 é apresentado um resumo dos mesmos. É verificado que o módulo emite dados a uma frequência variável tal como está referido na secção 5.4.3.4, o módulo emite uma interrupção quando são detetadas variações nos sensores, de modo a informar o microcontrolador que existem dados a serem consultados. Com este teste também se verifica que o sistema de aquisição de dados tem a capacidade de processar mais do que 10 amostras por segundo de cada sensor, sendo esta taxa definida como requisitos mínimos na secção 2.5.

Tabela 6.10: Dados do teste à taxa de aquisição ao módulo inercial.

Intervalo de tempo (s)	Nº amostras Acelerómetro	Nº amostras Giroscópio	Duração média Acelerómetro (ms)	Duração média Giroscópio (ms)
[0-1]	12	18	1,771	1,433
[1-2]	12	19	1,279	1,625
[2-3]	12	19	1,734	1,334
[3-4]	18	20	1,822	1,308
[4-5]	14	18	1,758	1,422
[5-6]	11	19	1,582	1,465
[6-7]	15	19	1,848	1,171
[7-8]	15	19	1,609	1,554
[8-9]	14	21	1,320	1,700
[9-10]	13	18	1,670	1,485

### 6.2.6 Leitura do módulo GPS

A localização da mota em pista é obtida através do módulo GPS integrado no sistema de aquisição de dados referido na secção 4.3.3. Tendo em conta as características do módulo disponibilizadas pelo fabricante é possível verificar qual é a precisão dos valores obtidos pelo mesmo, foram realizados dois testes à comunicação entre o módulo e o microcontrolador. Para obter os dados no decorrer dos testes foram utilizados os dois canais do osciloscópio. O canal A é associado ao pino Rx da interface UART5 do microcontrolador e o canal B é associado ao pino RE4 do microcontrolador, como mostra a figura 6.14. Nos testes realizados o pino RE4 tem como função verificar quando está a ser adquirida uma nova mensagem.

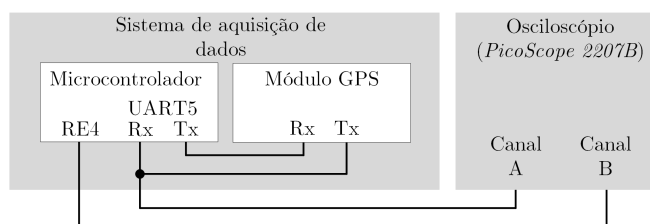


Figura 6.14: Ligação efetuada para realização de testes ao módulo GPS.

O primeiro teste efetuado à comunicação do módulo GPS com o microcontrolador tem como objetivo verificar as mensagens transmitidas pelo módulo GPS, bem como o tempo utilizado para a transmissão de cada uma. Para o primeiro teste foi efetuada a captura de dois pacotes que estão apresentados na tabela 6.11, sendo a referida captura apresentada na figura 6.15. Através do canal B do osciloscópio é verificado que o módulo envia um conjunto de dois pacotes de cada vez que o mesmo efetua uma transmissão, sendo que por cada transmissão de dados é enviado um pacote do tipo "GPRMC" e outro "GPTXT". A informação da localização da mota está contida no pacote "GPRMC" enquanto o pacote "GPTXT" é enviado automaticamente pelo módulo no final de cada conjunto de pacotes. Com o teste realizado é possível verificar que o módulo está a funcionar perante as configurações efetuadas na secção 5.4.2.1, sendo verificada a transmissão de pacotes do tipo "GPRMC" para o microcontrolador. Por fim, com este teste é possível verificar também que o módulo demora cerca de 18 ms a transmitir cada conjunto de pacotes.

Tabela 6.11: Pacotes transmitidos pelo módulo GPS durante a captura de dados.

Pacote	Duração (ms)	Dados recebidos
1	12,39	\$GPRMC,061642.400,A,3944.6004,N,00853.5662,W,7.40,299.97,080420,,A*7FCRLF
2	5,92	\$GPTXT,01,01,02,ANTSTATUS=OPEN*2BCRLF

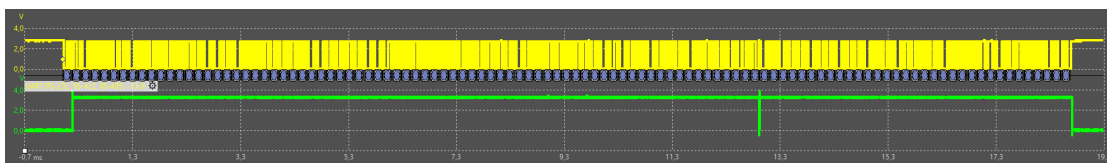


Figura 6.15: Sinais capturados durante o teste da comunicação com o módulo GPS.

O segundo teste realizado tem como objetivo verificar a taxa de atualização dos



pacotes transmitidos pelo módulo GPS. Este teste foi realizado durante um período de 20 segundos, de modo a garantir a recolha de dados suficientes para análise. No anexo E.5.1 estão apresentados todos os pacotes recolhidos durante o teste, sendo que a tabela 6.12 representa o resumo dos dados recolhidos durante o teste realizado.

Tabela 6.12: Resumo dos dados ao teste à taxa de aquisição do módulo GPS.

Tipo de pacote	Nº pacotes transmitidos	Nº bytes por pacote	Duração (ms)	Taxa de atualização (Hz)
\$GPRMC	200	74	12,385	10
\$GPTXT	200	37	5,921	10

No decorrer do teste foram transmitidos um total de 400 pacotes, sendo 200 do tipo "GPRMC" e os restantes do tipo "GPTXT". Também se verifica que cada pacote do tipo "GPRMC" é constituído por 74 *bytes*, que a transmissão do mesmo demora cerca de 12.4 ms e que são recebidos 10 pacotes por segundo. O pacote "GPTXT" é constituído por 37 *bytes*, que a transmissão do mesmo demora cerca de 5.9 ms e que também são recebidos 10 pacotes por segundo. Por fim é verificado que o requisito mínimo para a taxa de aquisição da localização da mota é cumprido.

### 6.2.7 Comunicação com cartão *microSD*

A solução desenvolvida tem como objetivo recolher os dados durante o funcionamento da mota, sendo que para garantir o registo de todos os valores adquiridos é utilizado o cartão *microSD*. De modo a garantir o correto funcionamento com o cartão foram realizados dois testes. O primeiro teste verifica a transmissão de um pacote de dados para o cartão *microSD*. O segundo teste tem como objetivo verificar a taxa de registo de valores no cartão.

Para efetuar o primeiro teste utilizou-se o osciloscópio para verificar o tempo que demora a transmitir um pacote de dados para o cartão. O canal A está associado a uma linha de dados do barramento SPI (SDO) e o canal B está associado ao pino que faz a seleção do dispositivo (SS), como mostra a figura 6.16. Como as linhas de dados são

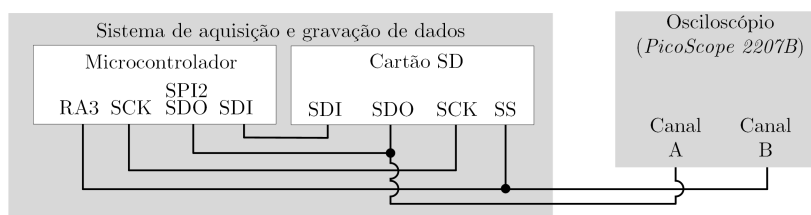


Figura 6.16: Ligação efetuada para os testes à comunicação com o cartão *microSD*.

partilhadas com o módulo rádio, utilizou-se o pino associado (SS) para efetuar o *trigger* no osciloscópio. Através da figura 6.17 é possível verificar que demora cerca de 14 ms a registar um pacote composto por 4096 *bytes* como referido anteriormente na secção 5.4.3.6. Através do *datasheet* do cartão *microSD* é possível verificar que mesmo suporta velocidades até 25 Mbit/s, mas como a velocidade de escrita está limitada pelo relógio do barramento de dados a velocidade máxima de escrita é de 8 Mbit/s [76].

Para poder calcular a velocidade mais precisa de escrita no cartão *microSD* era necessário utilizar um terceiro canal no osciloscópio associado ao sinal de relógio do barramento, de modo a contar o número correto de *bytes* transmitidos para o cartão.

Deste modo como temos o comprimento dos dados escritos e o tempo associado a todo o processo para efetuar a escrita de um pacote, foi calculada a velocidade aproximada do registo de dados. Neste caso o valor obtido é inferior à velocidade real, porque para efetuar um registo no cartão são precisas fazer outras operações, tais como abrir e fechar o ficheiro, logo são enviados um maior numero de bytes. A velocidade aproximada para o registo de um pacote de dados no cartão é de 2.34 Mbit/s, como mostra a equação 6.2.

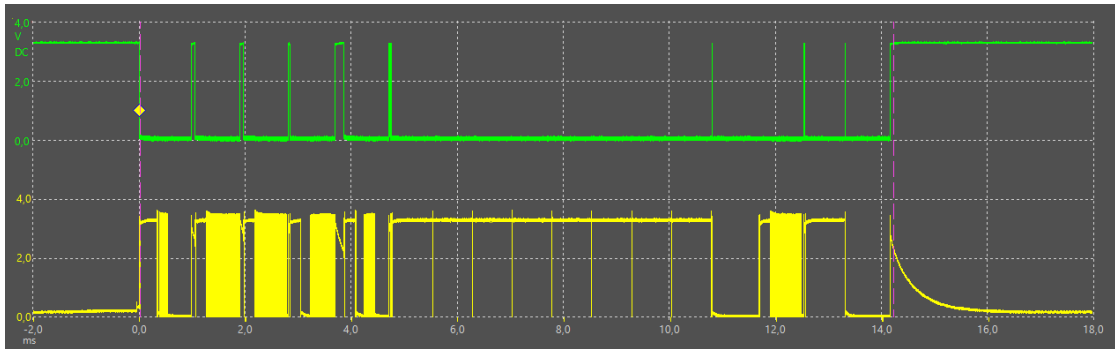


Figura 6.17: Exemplo de um pacote guardado no cartão SD.

$$\text{Velocidade de escrita} = \frac{N \text{ Bytes} \times 8 \text{ bit}}{\text{Tempo}} = \frac{4095 \text{ B} \times 8 \text{ bit}}{14 \text{ ms}} = 2.34 \text{ Mbit/s} \quad (6.2)$$

O segundo teste tem como objetivo verificar se são guardadas cerca de 20 amostras por segundo, sendo este um requisito para a solução como mostra a secção 2.5. Este teste foi realizado por duas tarefas. A primeira tarefa resume-se a ligar o sistema durante um período de 20 minutos. A segunda tarefa é analisar os dados guardados no cartão microSD, sendo utilizada a interface gráfica desenvolvida. Ao abrir o ficheiro através da interface, esta gera um ficheiro com as amostras guardadas e o tempo a que as mesmas estão associadas. Os dados guardados durante o período de recolha estão apresentados no anexo E.6.1. Através dos valores registados foi obtido o gráfico da figura 6.18, que indica o número de mensagens guardadas no cartão por segundo bem como o valor médio para a taxa amostras registadas. A tabela 6.13 apresenta o resumo dos dados recolhidos durante o teste efetuado.

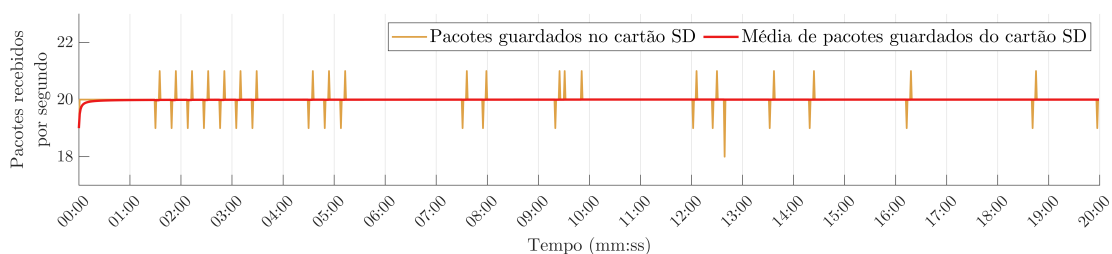


Figura 6.18: Taxa de gravação de dados no cartão microSD durante 20 minutos.

Com os testes realizados é possível verificar a velocidade aproximada da escrita de dados no cartão, que é cumprido o requisito mínimo da solução de registar os dados obtidos a uma taxa de 20 Hz e que é garantido que o sistema consegue recolher dados durante um período de 20 minutos.

Tabela 6.13: Resumo dos dados guardados no cartão microSD.

Nº de amostras	Duração do teste (s)	Amostras por segundo	Tamanho do ficheiro (kB)
24000	1200	20	1048,58

### 6.2.8 Transmissão de dados para interface gráfica

Para a transmissão de dados entre o sistema de aquisição de dados e a interface gráfica, foram efetuados quatro testes. Os testes realizados têm como objetivo verificar o funcionamento da transmissão de dados aplicada na solução. Neste caso são verificados os sistemas de transmissão e de receção de dados bem como a receção dos dados na interface gráfica. Para os dois primeiros testes foi utilizado o osciloscópio de modo a ser possível analisar as comunicações entre o microcontrolador e o módulo RFM95PW no sistema de aquisição de dados, tal como é apresentado na figura 6.19. Tal como os testes apresentados na secção 6.2.7, o canal A está associado a uma linha de dados do barramento SPI (SDO) e o canal B está associado ao pino que faz a seleção do dispositivo (SS). Como as linhas de dados são partilhadas com o cartão microSD, utilizou-se o pino associado (SS) para efetuar o *trigger* no osciloscópio.

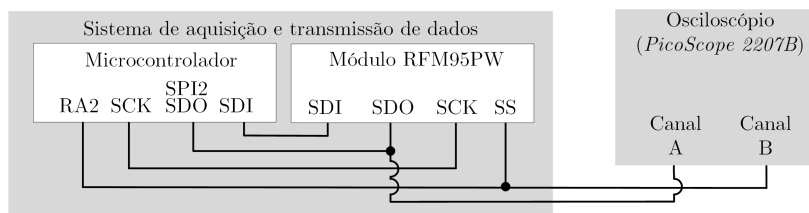


Figura 6.19: Ligação efetuada para os testes à comunicação com módulo rádio.

Com a recolha de dados efetuada no primeiro teste, apresentada na figura 6.20 é possível verificar que o microcontrolador demora cerca de 98  $\mu$ s a enviar e a dar ordem de transmissão de um pacote pelo módulo rádio. Como só existem disponíveis 2 canais no osciloscópio, não foi possível verificar o número de bytes enviados ou recebidos. Tendo em conta que a o pacote de dados é composto por 38 *bytes* e que o tempo de comunicação entre o microcontrolador e o módulo rádio é obtida uma velocidade aproximada do barramento de 3.88 Mbit/s, como mostra a equação 6.3.

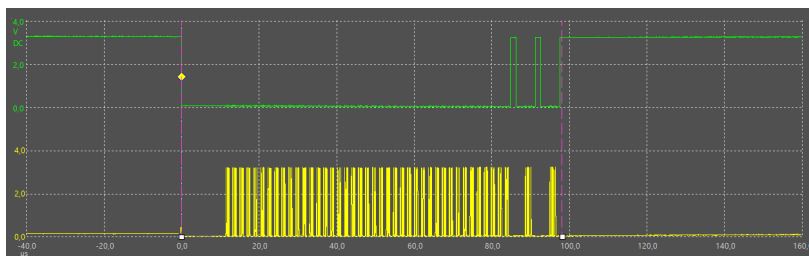


Figura 6.20: Amostra do barramento SPI durante transmissão de um pacote para o módulo rádio.

$$\text{Velocidade de transmissão} = \frac{N \text{ Bytes} \times 8 \text{ bit}}{\text{Tempo}} = \frac{38 \text{ B} \times 8 \text{ bit}}{98 \mu\text{s}} = 3.88 \text{ Mbit/s} \quad (6.3)$$

O segundo teste consiste em verificar quantos pacotes são enviados por segundo para o módulo rádio. Como o osciloscópio não consegue decodificar o barramento SPI sem o pino do SS, durante este teste foi só adquirida uma captura do osciloscópio durante o período de 1 segundo. A figura 6.21 apresenta a captura efetuada aos dois sinais associados aos canais do osciloscópio, da qual é possível verificar que são enviados 10 pacotes por segundo para o módulo rádio.

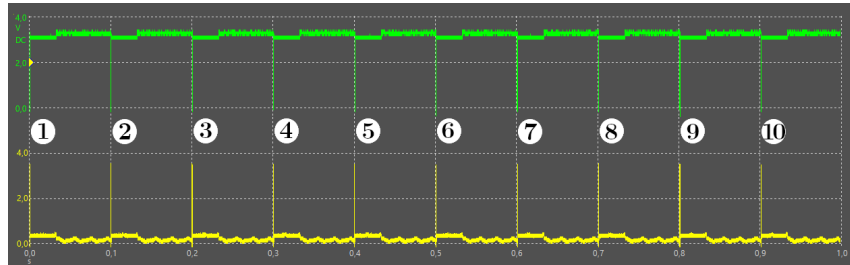


Figura 6.21: Amostra de barramento SPI durante transmissão de 10 pacotes.

O terceiro teste tem como objetivo verificar o funcionamento da transmissão de dados a distâncias superiores a 2 km. Foi estabelecida esta distância tendo em conta que o raio máximo obtido entre transmissor e o recetor na pista de Aragon, indicada na secção 1.1, é de 1.55 km. Este teste foi realizado num dia de chuva em que o transmissor enviava 15 pacotes por segundo e que o mesmo se encontrava a 1 km e a 2.6 km do recetor. Na figura 6.22 é apresentado um mapa do local onde foi efetuada a recolha de dados, bem como os locais onde se encontravam os dois sistemas. Para facilitar a recolha de dados durante este teste fixou-se o transmissor numa localização, sendo neste caso o recetor e interface gráfica os objetos móveis da solução.

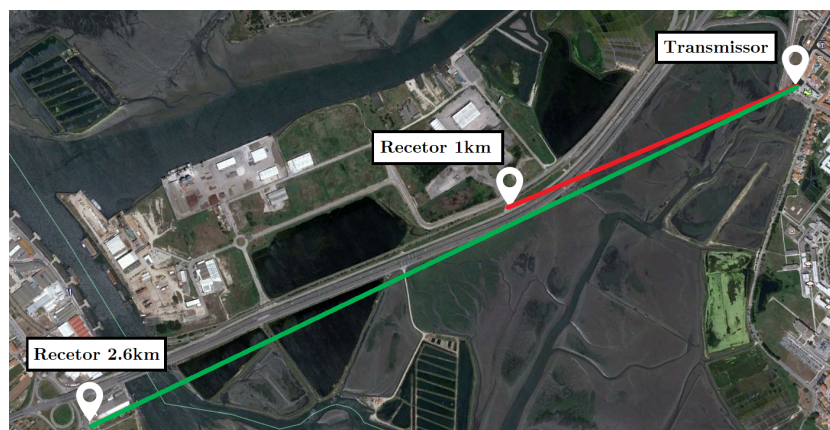


Figura 6.22: Localização dos sistemas durante o terceiro teste.

Para cada posição do recetor foram recolhidos pacotes durante 45 s, sendo que os dados recebidos pela interface gráfica estão presentes nos anexos E.7.1 e E.7.2. Na tabela 6.14 é apresentado o resumo dos dados capturados durante o teste às distâncias referidas anteriormente. É possível verificar que existe uma menor perda de dados quando a distância entre os dois sistemas é mais reduzida. Através dos dados recolhidos a 1 km entre os sistemas é verificado que foram recebidos 517 pacotes o que resulta numa média de 11.49 pacotes por segundo, enquanto à distância de 2.6 km foram recebidos 421 pacotes, o que indica uma média de 9.36 pacotes por segundo.

Tabela 6.14: Resumo dos dados obtidos no teste ao alcance da comunicação rádio.

Duração da recolha em segundos	Recolha a 1 km		Recolha a 2,6 km	
	Pacotes recolhidos	Média pacotes por segundo	Pacotes recolhidos	Média pacotes por segundo
45	517	11,49	421	9,36

Através dos dados recebidos pela interface gráfica também é apresentado o resumo dos mesmos nos gráficos apresentados na figura 6.23, da qual estes representam o número de pacotes recebidos pela interface gráfica por cada segundo.

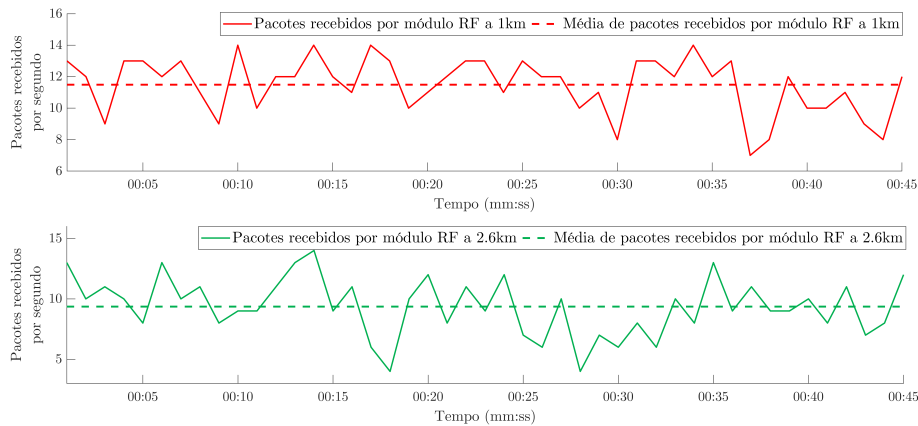


Figura 6.23: Resumo dos dados ao teste da distância de transmissão entre sistemas.

Com os dados obtidos é possível concluir que à distância de 1 km são recebidos corretamente cerca de 76% dos pacotes transmitidos enquanto que à distância 2.6 km são recebidos cerca de 62.4%. Com o teste realizado é possível verificar que a transmissão de dados por rádio não garante a receção de todos os dados adquiridos pelo o sistema de aquisição.

O quarto teste realizado à transmissão de dados para a interface gráfica tem como objetivo verificar se existem falhas na transmissão de dados durante um maior período de tempo, em que o transmissor se encontra a uma distância de 100 m do recetor. Este teste foi realizado ao mesmo tempo que se realizou o teste de guardar os dados no cartão microSD referido na secção 6.2.7. No decorrer do teste a interface gráfica ao receber cada pacote guardou os valores recebidos num ficheiro, de modo a ser possível verificar se os dados são exatamente iguais às amostras guardadas pelo o cartão microSD no sistema de aquisição de dados. No anexo E.7.3 estão apresentados os pacotes recebidos pela interface gráfica através do sistema recetor, bem como indica qual é o número do pacote guardado no cartão microSD.

A primeira análise efetuada sobre os dados recebidos pelo sistema recetor, tem como objetivo verificar quantos pacotes são recebidos por segundo, bem como a média do mesmo ao fim de 20 minutos. A segunda análise passa por verificar se a mensagem recebida está guardada no cartão microSD. Como os dados enviados por rádio são parte da informação guardada no cartão microSD pelo sistema transmissor, é possível verificar se a informação após ser transmitida e recebida pela interface gráfica coincide com os valores guardados antes dos mesmos serem enviados. Na tabela 6.15 está apresentado o resumo das análises referidas anteriormente. É possível verificar que foram recebidos 9400 pacotes durante os 20 minutos que resulta numa média de 7.83 pacotes por segundo. Através da comparação é possível verificar que existem erros na transmissão de dados

em cerca de 0.04% do pacotes recebidos.

Tabela 6.15: Resumo dos dados recebidos pela interface gráfica.

Duração do teste (s)	Nº de pacotes recebidos	Média de pacotes por segundo	Nº de pacotes válidos	% de erro na transmissão
1200	9400	7,83	9396	0,04%

Na figura 6.24 está apresentado o gráfico correspondente ao número de pacotes recebidos por segundo bem como a comparação com o valor médio. É possível verificar através da figura que o valor do número de mensagem recebidas por segundo varia entre 4 a 10 e o que o seu valor médio ronda 7 mensagens por segundo.

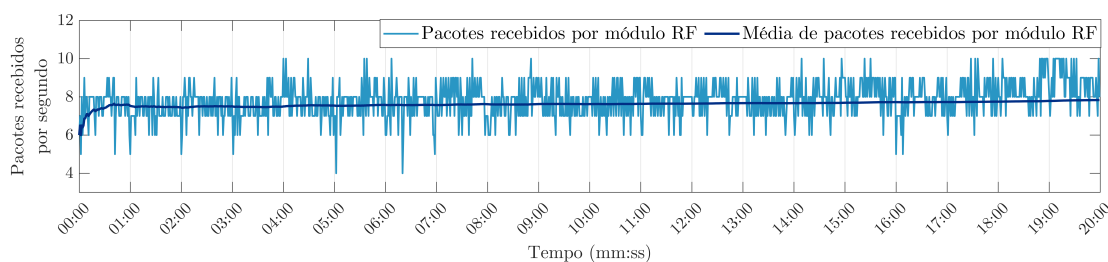


Figura 6.24: Taxa de receção de dados na interface gráfica durante 20 minutos.

Com os testes realizados é possível verificar que a comunicação entre os microcontroladores e os módulos rádio funciona corretamente, que a transmissão de dados funciona até uma distância de 2.6 km, que o número de pacotes válidos é menor a distâncias superiores e à distância de 100 m cerca de 99.96% dos pacotes recebidos não apresentam qualquer tipo de erro.

### 6.2.9 Transmissão de dados para o ecrã

Como referido na secção 2.5 um dos requisitos da solução é enviar os dados recolhidos pelo sistema de aquisição de dados para um ecrã desenvolvido pela a equipa *Mototechnics*. Foram efetuados dois testes para verificar que a transmissão de dados para o ecrã é efetuada corretamente. Para os testes realizados foi utilizado o canal A do osciloscópio associado ao pino transmissor da interface UART3 do microcontrolador, como está representado na figura 6.25.

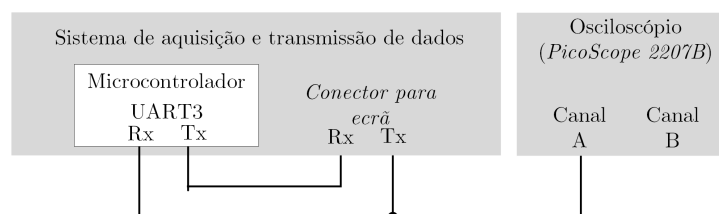


Figura 6.25: Ligação efetuada para os testes à comunicação com ecrã.

O primeiro teste tem como objetivo analisar a transmissão de um só pacote para o ecrã. Com os dados adquiridos durante o teste é possível verificar o comprimento de cada pacote bem como o tempo que o microcontrolador demora a enviar os dados para o ecrã. Com os dados recolhidos pelo osciloscópio é possível verificar que são enviados 42 bytes por cada pacote enviado, sendo que este pacote demorou cerca de 3.53 ms a ser

transmitido. Na figura 6.26 é apresentada a captura do osciloscópio durante o primeiro teste. Tendo em conta que são enviados 10 *bits* por cada *byte* enviado e que o pacote é constituído por 42 bytes, o *baudrate* obtido na transmissão de dados para o ecrã é de 118 980 bit/s como mostra a equação 6.4.



Figura 6.26: Pacote enviado para o ecrã capturado pelo osciloscópio.

$$Baudrate_{UART3} = \frac{N^{\circ} Bytes \times 10}{Tempo} = \frac{42 \times 10}{3.53 \text{ ms}} = 118\,980 \text{ bit/s} \quad (6.4)$$

O segundo teste tem como objetivo verificar a taxa de transmissão de dados para o ecrã. Este teste foi realizado durante um intervalo de tempo de 10 segundos, de modo a facilitar o tratamento dos dados recolhidos. No anexo E.8.1 estão apresentados todos os dados capturados durante o teste, sendo que na tabela 6.16 é apresentado um resumo dos mesmos. É verificado que o microcontrolador durante o teste transmitiu 100 pacotes de dados, que foi obtida uma taxa de atualização de 10 Hz e uma duração média da transmissão de dados igual a 3.44 ms por cada pacote transmitido.

Tabela 6.16: Resumo dos dados ao teste da taxa de transmissão para o ecrã.

Duração do teste	N <sup>o</sup> pacotes transmitidos	N <sup>o</sup> bytes por pacote	Duração média (ms)	Taxa de atualização (Hz)
10	100	42	3.44	10

Por fim com os dois testes apresentados verifica-se que o sistema de transmissão tem a capacidade de enviar 10 pacotes por segundo para o ecrã e que cumpre com os requisitos da solução apresentados na secção 2.5.

### 6.2.10 Peso e dimensões da solução

Com toda a solução desenvolvida e verificada para concluir os testes, são feitos testes de modo a obter tamanho e peso associado a cada componente da solução. O primeiro sistema a ser avaliado é o de aquisição, gravação e transmissão de dados que fica instalado na mota. Este sistema pode variar o seu peso e tamanho consoante o número de sensores adicionais. A tabela 6.17 mostra o peso e a dimensão de cada componente do sistema instalado na mota, tendo em conta os conectores utilizados.

Vendo esta solução vista como um produto de mercado, esta poderia ser aplicada em vários formatos. Foram definidos três formatos de modo a verificar a versatilidade da solução. O primeiro caso em estudo é constituído pelas componentes referidas na tabela 6.17. O segundo caso efetua a leituras dos valores da ECU, acelerações e posição GPS, sem analisar sensores adicionais. O terceiro caso é na instalação desta solução num veículo de 4 rodas, tal como um *kart*. Na tabela 6.18 é apresentada a diferença da quantidade de peças e do peso de implementação desta solução nos casos em estudo. O primeiro caso, sendo este o sistema desenvolvido é composto por 13 elementos e tem um peso total de 1.13 kg. O segundo caso é o mais simples dos três sendo construído

Tabela 6.17: Peso por componente do sistema de aquisição e transmissão de dados.

Componente	Peso (g)	Comprimento (mm)	Largura (mm)	Altura (mm)
Sistema transmissor de dados	162	121,2	112	29
Módulo de sensores adicionais	100	78,5	100,1	26,7
Antena rádio	55	150	28	28
Sensor de pressão	118	98	23	37
Pirómetro + Cabo	35			
Extensómetro	115	220	12	12
Cabos de ligação	60			

Tabela 6.18: Diferenças entre os casos de estudo do sistema de aquisição.

Componente	Peso (g)	Caso 1		Caso 2		Caso 3	
		Quant.	Peso (g)	Quant.	Peso (g)	Quant.	Peso (g)
Sistema transmissor de dados	162	1	162	1	162	1	162
Módulo de sensores adicionais	100	2	200	0	0	4	400
Antena rádio	55	1	55	1	55	1	55
Sensor de pressão	118	2	236	0	0	2	236
Pirómetro + Cabo	35	2	70	0	0	4	140
Extensómetro	115	2	230	0	0	4	460
Cabos de ligação	60	3	180	1	60	5	300
<b>Total</b>		13	1133	3	277	21	1753

por 3 componentes e a sua instalação aumenta cerca de 300 g ao veículo. O terceiro caso é o mais complexo de todos, sendo composto por 21 componentes e tem um peso total de 1.75 kg. Caso seja pretendido obter maior quantidade de dados o peso associado ao sistema vai aumentar tal como os custos associados, sendo assim possível adequar o sistema a vários casos.

O segundo sistema a ser avaliado é o que está responsável por receber os dados remotamente. Caso o utilizador necessite de recolher dados remotamente para qualquer um sistema instalado na mota, este é composto sempre pelo mesmo número de componentes. O peso associado a este sistema está apresentado na tabela 6.19.

Tabela 6.19: Peso por componente do sistema de receção de dados.

Componente	Peso (g)	Comprimento (mm)	Largura (mm)	Altura (mm)
Sistema recetor de dados	74	78.8	84.3	15.5
Antena rádio	55	150	28	28

Com os testes realizados verifica-se que o peso total da solução não ultrapassa o valor igual a 1.5 kg definido nos requisitos mínimos da solução na secção 2.5.

### 6.3 Teste à solução desenvolvida em pista

Nesta secção é apresentado o teste de toda a solução desenvolvida com a mota em pista. De modo a verificar o funcionamento de todo o sistema em ambiente real foi realizado um teste com a solução desenvolvida instalada na mota. O circuito onde é realizada a competição *Motostudent* é em Aragon, sendo este um circuito bastante longe e que obriga a um maior custo de deslocação e de logística, foi realizado o teste a toda a solução num kartódromo. O circuito utilizado para a realização do teste foi o kartódromo



internacional de Leiria, cujo o trajeto em teste está apresentado a vermelho na figura 6.27.



Figura 6.27: Circuito do kartódromo internacional de Leiria.

O teste tem como objetivo verificar a aquisição dos valores guardados no cartão microSD bem como a transmissão de dados em tempo real, sendo utilizada como ferramenta de verificação dos valores a interface gráfica desenvolvida apresentada na secção 6.1.2. Como esta solução está desenvolvida para o protótipo da moto atualmente em construção, não foram instalados todos os sensores requeridos para a solução na moto em teste. Neste caso foi instalado o sistema de aquisição de dados apresentado na figura 6.28.

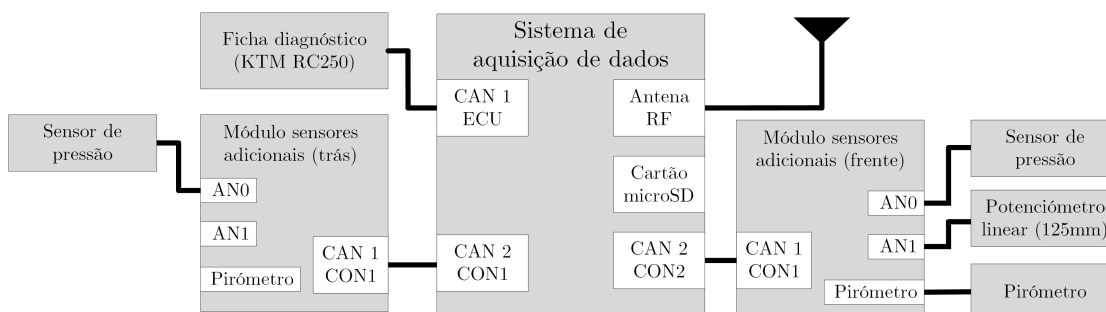


Figura 6.28: Sistema de aquisição de dados instalado na moto.

O teste efetuado à solução consistiu em realizar três sessões de treino, sendo que os valores recolhidos para o cartão SD estão apresentados no anexo E.9.1 e o resumo dos mesmos está apresentado na tabela 6.20. Nas três sessões realizadas existe sempre a volta número zero, da qual está associada a inicialização do sistema instalado na moto quando a mesma é ligada. O piloto só iniciava a sessão de treino após ser recebido um sinal de GPS válido. Na primeira sessão foram efetuadas três voltas completas ao circuito, sendo que o ficheiro adquirido tem um tamanho de 354 KB. Na segunda sessão de treino foram realizadas duas voltas ao circuito, tendo sido gerado um ficheiro com 179 KB. Durante a última sessão de treino efetuada foram efetuadas três voltas ao circuito, sendo que o ficheiro gerado ocupa cerca de 303 KB. O funcionamento do sistema de transmissão de dados em tempo real pode ser visualizada através do vídeo disponível no anexo E.9.2.

Após a apresentação de todos os testes realizados à solução, verifica-se que a mesma tem a capacidade de adquirir, processar, guardar, transmitir e receber dados da moto em pista. Com os testes realizados é possível verificar o funcionamento das componentes

desenvolvidas para a solução, como também confirmar se os requisitos mínimos para solução foram cumpridos.

Tabela 6.20: Resumo dos testes efetuados em pista.

Teste	Volta	Tempo (mm:ss.ss)	Número de amostras	Tamanho do ficheiro de dados
1	0	01:39.90	1998	354 KB
	1	01:15.55	1511	
	2	01:12.70	1454	
	3	03:02.95	3658	
2	0	01:23.95	1679	179 KB
	1	01:28.55	1770	
	2	00:46.00	920	
3	0	01:56.50	2330	303 KB
	1	01:25.20	1704	
	2	01:25.35	1705	
	3	01:22.30	1646	

A secção que se segue apresenta as conclusões retiradas após todo o desenvolvimento da solução de telemetria desenvolvida para a competição *Motostudent*.

## Secção 7

# Conclusões e trabalhos futuros

Após o desenvolvimento deste projeto pode afirmar-se que foi idealizado e criado um prototipo de mercado "chave na mão" para a recolha de dados num veículo de competição, sendo este capaz de adquirir, tratar, guardar, transmitir e apresentar os dados ao utilizador final. Esta secção tem como objetivo apresentar as conclusões do trabalho desenvolvido através da descrição das características da solução, comparando-a com os dois produtos de mercado apresentados na secção 2.4 e por fim apresentar algumas sugestões para trabalhos futuros.

### 7.1 Especificações do sistema

Com o desenvolvimento do sistema e os resultados obtidos nos testes realizados é possível definir as características do sistema, sendo elas as grandezas que o mesmo consegue adquirir, as comunicações suportadas, as dimensões e peso bem com o preço associado a produção do sistema. Na tabela 7.1 são apresentadas as características da solução desenvolvida.

Os custos associados à solução são divididos em duas partes, a primeira refere-se à produção da solução desenvolvida e a segunda está associada aos sensores, às antenas e aos conectores utilizados. Considerando a solução desenvolvida formada por um conjunto, de dois módulos para sensores adicionais, o filamento 3D para a proteção de *hardware*, um sistema transmissor e um sistema recetor, é verificado que este tem um custo de produção igual a 325.17 €. Os custos associados as todas as componentes que foram adquiridas sem qualquer tipo de desenvolvimento, tais como os dois sensores de pressão, os dois potenciómetros lineares, os dois pirómetros, as duas antenas RF e os conectores utilizados, tem um custo de adicional de 399.87 €. Logo a instalação da solução completa tem custo total de 725.04 €.

Os requisitos mínimos referidos na secção 2.5 estão divididos em três partes, a primeira está associada a constituição física da solução, a segunda é relacionada com as grandezas físicas que tinham de ser adquiridas e a terceira com as comunicações que a solução devia ter. Os requisitos da constituição da solução em geral foram cumpridos. No entanto a nível de proteção contra poeiras e gotas de água poderia ser melhorado e a solução poderia ser mais compacta caso tivesse sido optado por uma solução centralizada no sistema de aquisição de dados. Em relação aos requisitos das grandezas que tinham de ser adquiridas bem como a sua taxa de atualização, verifica-se que estes foram cumpridos. Os requisitos relacionados com as comunicações do sistema foram todos cumpridos. É de referir que também está desenvolvida uma interface gráfica que não estava incluída nos requisitos mínimos.

Tabela 7.1: Características da solução desenvolvida.

Caraterística	Descrição
Sistema de aquisição e transmissão de dados	
Alimentação do sistema	12V CC
Controlador CAN 1	Leitura da ECU
Controlador CAN 2	Suporta até 29 módulos adicionais
GPS	Integrado com taxa de atualização até 10Hz
Acelerações e giroscópio	3 eixos acelerómetro e 3 eixos giroscópio
Conector para dispositivo de memória externa	Cartão microSD(até 64GB)
Porta para comunicação com ecrã	Protocolo UART com <i>baudrate</i> de 115 200 bit/s
Transmissão de dados	Banda ISM 868 MHz
Dimensões	121.2x112x29 mm
Peso	162 g
Proteção	PLA
Módulo para sensores adicionais	
Alimentação do módulo	12V CC
Controlador CAN 1	Transmissão de dados (20 Hz)
Entradas analógicas	2 entradas de 0-10V (20 Hz)
Leitura do pirómetro	Protocolo I <sup>2</sup> C
Dimensões	78.5x100.1x29 mm
Peso	100 g
Proteção	PLA
Sistema de recepção de dados	
Alimentação do sistema	[5-12]V CC
Recepção de dados	Banda ISM 868 MHz
Comunicação com interface gráfica	USB
Dimensões	78.4x84.3x15.5 mm
Peso	100 g
Proteção	PLA

## 7.2 Comparação com os produtos de mercado

Com as especificações da solução desenvolvida e dos dois produtos de mercado apresentados na secção 2.4, é possível verificar as diferenças entre eles. Na tabela 7.2 estão apresentadas as características bem como o preço associado a cada solução de mercado e à desenvolvida.

A solução desenvolvida é a única das três que não apresenta o ecrã incluído no sistema, mas está preparada para transmitir dados adquiridos para um ecrã. Comparando as interfaces CAN disponibilizadas por cada solução, verifica-se que só o produto da Aim iguala o número de controladores CAN. Em relação ao GPS, todos têm o módulo de localização integrado, sendo o produto da Starlane e a solução desenvolvida, os únicos que tem uma taxa de atualização de 10 Hz. Em termos de entradas analógicas nenhuma das soluções de mercado suporta um número tão elevado de sensores analógicos, mas em comparação o produto da Aim suporta 4 entradas analógicas configuráveis até 1000Hz cada uma, enquanto que a solução desenvolvida está preparada para uma atualização de 20 Hz por cada entrada analógica. Comparando as plataformas inerciais, todas são equivalentes quanto ao número de eixos por sensor. Em termos de comunicações nenhuma das soluções de mercado tem a capacidade de efetuar a transmissão enquanto a mota está em funcionamento, sendo estas só utilizadas para configuração ou fazer o *download* dos dados guardados. Ou seja, a solução desenvolvida é a única das três que efetua trans-

missão de dados em tempo real até uma distância máxima de 2.6 km. Todas as soluções apresentadas tem uma interface gráfica para analisar os dados recolhidos. Comparando o preço de custo associado a cada produto é verificado que a solução desenvolvida está muito abaixo das restantes, sendo que neste caso só é indicado o preço de produção, ou seja não estão incluídos os custos associados ao desenvolvimento da solução, suporte técnico, logística, marketing ou custos administrativos.

Tabela 7.2: Comparação entre os produtos de mercado e a solução desenvolvida.

Características	Aim MXm	Starlane Davinci II R	Solução desenvolvida
Ecrã (resolução)	268*128	800*400	
Porta USB			Para dados ecrã
Ligações por CAN	2	1	2
GPS	Integrado	Integrado até (10Hz)	Integrado até (10Hz)
Módulos externos	Camara HD	WID-C	Para sensores analógicos
Entradas analógicas	4 (até 1000Hz cada uma)	1 Acelerador	Até 58 sensores (20 Hz)
Plataforma inercial	3 eixos acelerómetro	3 eixos acelerómetro	3 eixos acelerómetro
	3 eixos giroscópio	3 eixos giroscópio	3 eixos giroscópio
	3 eixos magnetómetro	3 eixos magnetómetro	3 eixos magnetómetro
Comunicação sem fios	<i>Bluetooth</i>	Wifi e Bluetooth	Rádio por banda ISM
Memória interna	4 GB	999 voltas	
Memória externa			Cartão microSD (até 16GB)
Distância da transmissão			Até 2,6 km
Sistema de receção dados			Sim
Interface gráfica	Sim	Sim	Sim
Preço	2 227 €	2 815 €	725,04 €

Tendo em conta as diferenças referidas é possível verificar que os objetivos foram cumpridos, pois foi construída uma solução mais económica e que se diferencia das soluções de mercado por efetuar a transmissão de dados em tempo real.

### 7.3 Trabalhos futuros

Após todo o desenvolvimento da solução é possível torna-la mais robusta e com maior número de funcionalidades. Para trabalhos futuros são feitas as seguintes sugestões:

- Implementação desta solução num veículo de 4 rodas, para testar a solução em termos da capacidade de processamento de uma maior quantidade de dados,
- Desenvolver um novo módulo de sensores adicionais capaz de suportar dois pirómetros e quatro entradas analógicas para facilitar a implementação do sistema num veículo de 4 rodas,
- Redesenhar o sistema de aquisição de dados com ecrã embutido de modo a reduzir a cablagem e tornar a solução mais compacta, tal como os produtos de mercado analisados,
- Adicionar funcionalidades ao sistema de recolha de dados tal como, recolha de trajetórias e possibilidade de adicionar pontos de cronometragem na pista,
- Tornar a transmissão de dados entre os dois sistemas bidirecional de modo a evitar perda de pacotes e tornar a transmissão de dados mais robusta,
- Melhorar a interface gráfica porque o desenvolvimento da mesma baseou-se na recolha e apresentação dos valores obtidos, sendo assim possível adicionar funcio-

nalidades de análise de dados para simplificar a compreensão dos valores apresentados ou até de criar automaticamente o ficheiro de configuração para o sistema de recolha de dados.

# Bibliografía

- [1] E. F. Moto and M. Technopark, “Motostudent,” <https://bit.ly/2ZPloyD>, Acedido: 12-02-2020.
- [2] R. Circuits, “Motorland Aragón,” <https://bit.ly/2ZSWvCb>, Acedido: 12-02-2020.
- [3] W. J. Mayo-Wells, “The origins of space telemetry,” *Technology and Culture*, vol. 4, no. 4, pp. 499–514, 1963.
- [4] J. Muehlner, L. MISSILES, and S. C. I. S. CA., *TRENDS IN MISSILE AND SPACE RADIO TELEMETRY*. Defense Technical Information Center, 1962.
- [5] Ebay, “OBD2 ELM327 V1.5,” <https://ebay.to/2MeTMLj>, Acedido: 30-09-2019.
- [6] OCTech, “TouchScan,” <https://bit.ly/3eyyWT4>, Acedido: 30-09-2019.
- [7] Ametherm, “PTC Thermistors vs. NTC Thermistors for Inrush Current,” <https://bit.ly/2ZPEXQV>, Acedido: 7-01-2020.
- [8] Melexis, “*Digital plug & play infrared thermometer in a TO-can*,” <https://bit.ly/2XgXcn0>, Acedido: 9-01-2020.
- [9] M. Projetado, “Efeito Piezoelétrico – Entenda como funciona o buzzer,” <https://bit.ly/2TROSSY>, Acedido: 9-01-2020.
- [10] Shop-Racing, “*Linear displacement sensors suspension displacement sensor 100 mm AIM*,” <https://bit.ly/2TTcz2R>, Acedido: 13-01-2020.
- [11] N. R. Laddha and A. Thakare, “A review on serial communication by uart,” *International journal of advanced research in computer science and software engineering*, vol. 3, no. 1, 2013.
- [12] S. Heath, “Embedded systems design. newnes,” 2003.
- [13] D. Paret and C. Fenger, *The I2C bus: from theory to practice*. Wiley New York, 1997.
- [14] Mankar, Jayant and Darode, Chaitali and Trivedi, Komal and Kanoje, Madhura and Shahare, Prachi, “Review of I2C protocol,” *Int J*, vol. 2, no. 1, 2014.
- [15] U. del País Vasco, “Unidad 15: Reloj I2C,” <https://bit.ly/3dncDjm>, Acedido: 15-01-2020.
- [16] N. Anand, G. Joseph, S. S. Oommen, and R. Dhanabal, “*Design and implementation of a high speed Serial Peripheral Interface*,” pp. 1–3, 2014.

- [17] Corelis, “SPI Tutorial,” <https://bit.ly/2Bgtp5s>, Acedido: 15-01-2020.
- [18] T. Instruments, “Introduction to the Controller Area Network (CAN),” <https://bit.ly/2XHQGol>, Acedido: 22-02-2019.
- [19] —, “Introduction to the Controller Area Network (CAN),” <https://bit.ly/36JcKD3>, Acedido: 24-02-2019.
- [20] *Section 34. Controller Area Network (CAN)*, Rev. c ed., Microchip Technology Inc, Maio 2012.
- [21] C. Coleman, *An introduction to radio frequency engineering*. Cambridge University Press, 2004.
- [22] M. W. O. Siegmund M. Redl, Matthias K. Weber, *GSM and personal communications handbook*. Artech House Publishers, 1998.
- [23] A. Mehrotra, *GSM system engineering*. Artech House, Inc., 1997.
- [24] M. Sauter, *From GSM to LTE: an introduction to mobile networks and mobile broadband*. John Wiley & Sons, 2010.
- [25] M. Loy, R. Karingattil, and L. Williams, “Ism-band and short range device regulatory compliance overview,” 2005.
- [26] K. Abhaykumar, “Overview of ism bands and software-defined radio experimentation,” *Wireless Personal Communications*, vol. 97, no. 3, pp. 3743–3756, 2017.
- [27] X. Fuqin, *Digital Modulation Techniques*. Artech House Publishers, 2000.
- [28] S. A. Abrantes, “Apontamentos de telecomunicações 2,” 2004.
- [29] M. Chernyshev, “Verification of primitive sub ghz rf replay attack techniques based on visual signal analysis,” 2013.
- [30] B. Nikesh, “Digital Modulation:ASK, PSK, FSK,” <https://bit.ly/3cfsleP>, Acedido: 24-02-2019.
- [31] R. Chhilar, J. Khurana, and S. Gandhi, “Modulation formats in optical communication system,” *IJCEM International Journal of Computational Engineering & Management*, vol. 13, pp. 110–115, 2011.
- [32] D. L. Ash, “A comparison between ook/ask and fsk modulation techniques for radio links,” *Technical Report*, 1992.
- [33] T. Aim, <https://bit.ly/3ckFVxw>, Acedido: 24-02-2019.
- [34] P. E. Starlane, <https://bit.ly/2XgG5lq>, Acedido: 22-12-2019.
- [35] S. Standard, “Sae j1979: E/e diagnostic test modes,” *Vehicle EE Systems Diagnostics Standards Committee. SAE International*, 2006.
- [36] Box REPSOL, “Choosing the right tyre for a race is often the key to victory.” <https://bit.ly/2XLEHpN>, Acedido: 25-01-2019.
- [37] I. E. Melexis, *MLX90614 family, Datasheet Single and Dual Zone Infra Red Thermometer in TO-39*, 2018.



- [38] M. Electronics, “MLX90614ESF-AAA-000-TU,” <https://bit.ly/3aya5ws>, Acedido: 11-12-2019.
- [39] —, “MLX90614ESF-BBA-000-SP,” <https://bit.ly/3cMjr9k>, Acedido: 11-12-2019.
- [40] *J. Juan Brake Systems*, “Rear racing master cylinder with remote reservoir die cast,” <https://bit.ly/2zyeNgX>, Acedido: 18-12-2019.
- [41] *Castrol*, “React Performance DOT 4,” <https://bit.ly/3bJJbmE>, Acedido: 18-12-2019.
- [42] *XMLG100D71-Datasheet*, Schneider Eletric, Dezembro 2019.
- [43] E. Shops, “Xm1g100d71,” <https://bit.ly/3eTWFOM>, Acedido: 18-12-2019.
- [44] Farnell, “IPS-G1003-5,” <https://bit.ly/2Ydz9q9>, Acedido: 18-12-2019.
- [45] MIRAN, “Linear Position Sensor,” <https://bit.ly/2xRyLTv>, Acedido: 19-12-2019.
- [46] F. Engineering, “MLS 130 Linear Displacement Sensor,” <https://bit.ly/2zuOk3H>, Acedido: 19-12-2019.
- [47] T. Connectivity, “MLP Linear Potenciometer,” <https://bit.ly/2Y863Is>, Acedido: 19-12-2019.
- [48] V. Cossalter, *Motorcycle Dynamics*, 2nd ed. LULU, 2006.
- [49] *MM7150 Motion Module*, Rev. b ed., Microchip Technology Inc., Janeiro 2016.
- [50] *OKY3255*, Rev. 1 ed., Okystar Technology, Setembro 2017.
- [51] Q. W. Solutions, *L80 Compact GPS Module with Patch Antenna*, Rev. 1.3 ed., 2017.
- [52] OriginGPS, *Spider ORG4472 Datasheet*, Rev.C05 ed., Junho 2018.
- [53] S. Talent, “Micro SD Data Sheet,” April 2012.
- [54] *FT232R USB UART IC Datasheet*, FTDI Chip, 4 2019, ver 2.15.
- [55] *FT232BL/BQ USB UART IC Datasheet*, FTDI Chip, 11 2011, ver 2.2.
- [56] H. RF, “RFM98PW/RFM95PW Enhanced Power Long Range Transceiver Module,” 2006.
- [57] RF Solutions, “RF LoRa LongRange Transceiver,” <https://bit.ly/3azpRXW>, Acedido: 24-01-2019.
- [58] Digi International, “DIGI XBEE SX 868,” <https://bit.ly/3cTkVyO>, Acedido: 24-01-2019.
- [59] *PIC32MX5XX/6XX/7XX Family Data Sheet*, Rev. k ed., Microchip Technology Inc, Setembro 2019.
- [60] *STM32MP157A/D*, Rev. 2 ed., STMicroelectronics, Agosto 2019.
- [61] Amazon, “Digilent chipKIT Max32 Microcontroller Board with Mega R3 Headers,” <https://amzn.to/2MjUzdV>, Acedido: 06-01-2020.

- [62] *chipKIT Max32*, Digilent Inc., 10 2014, ver E.0.
- [63] *MCP1725- Datasheet*, Rev. b ed., Microchip Technology Inc, Dezembro 2007.
- [64] *NCP1117, NCV1117*, Rev. 29 ed., ON Semiconductor, Julho 2018.
- [65] *OKI-78SR-E Series*, Rev. 1 ed., Murata Power Solutions Inc., 2017.
- [66] *R-78B-1.5*, Rev. 1 ed., Recom Inc., 2014.
- [67] G. Ramachandra, *PIC32 Bootloader*, rev. b ed., Microchip Technology Inc, Abril 2012.
- [68] *PESDxL4UF; PESDxL4UG; PESDxL4UW*, Rev. 4 ed., NXP Semiconductors, Fevereiro 2008.
- [69] Robert Smith, “*PIC32 OSC Configuration*,” <https://bit.ly/36PIYLz>, Acedido: 03-02-2019.
- [70] D. Matthews, *Basic PICmicro® Oscillator Design - AN849*, Microchip Technology Inc, 2002.
- [71] B. C. Baker, *AN699- Anti-Aliasing, Analog Filters for Data Acquisition Systems*. Microchip Technology Inc., 1999.
- [72] J. Karki, “Active low-pass filter design,” *Texas Instruments application report*, 2000.
- [73] *Analysis of the Sallen-Key Architecture*, Texas Instruments Inc., Setembro 2002.
- [74] *Section 24. Inter-Integrated Circuit (I2C)*, Rev. g ed., Microchip Technology Inc, Agosto 2016.
- [75] *MCP2561/2*, Rev. c ed., Microchip Technology Inc, Julho 2014.
- [76] *microSDHC memory card Flash Storage Media*, 4900180th ed., Kingston Technology.
- [77] Electronic BO, “SD and MicroSD Memory Card Pinout.” <https://bit.ly/3fbL6SZ>, Acedido: 24-02-2020.
- [78] J. Catsoulis, *Designing Embedded Hardware: Create New Computers and Devices*. ”O’Reilly Media, Inc.”, 2005.
- [79] “Digikey PCB Trace Width Calculator,” <https://bit.ly/2SvFDwM>, Acedido: 24-05-2019.
- [80] Microchip, “*MPLAB® XC Compilers*,” <https://bit.ly/2ucTXBe>, Acedido: 01-02-2020.
- [81] Microchip Technology, Inc., “*Understanding Floating-Point Values*,” <https://bit.ly/2SM3vfV>, Acedido: 19-02-2020.
- [82] *Section 17. 10-bit Analog-to-Digital Converter (ADC)*, Rev. e ed., Microchip Technology Inc, Agosto 2011.
- [83] *MLX90614 family*, Rev. 13 ed., Melexis, Setembro 2019.

- 
- [84] *Edwards AFB*, “*PIC32 CAN Examples*,” <https://bit.ly/2OOXzAz>, Acedido: 12-02-2020.
- [85] “Road vehicles - Diagnostic communication over Controller Area Network (Do-CAN),” 2016.
- [86] Microchip Technology Inc, “Explorer 16 (PIC24FJ) Host Sample Codes for MM7150-PICTAIL, v1.3.3,” <https://bit.ly/3cwT2Np>, Acedido: 16-12-2019.
- [87] *MM7150 Motion Module PICtail Plus Evaluation Board (AC243007) with Explorer 16 Development Board User’s Guide*, Rev. e ed., Microchip Technology Inc, Novembro 2018.
- [88] *L80 GPS Protocol Specification*, Rev. 1.2 ed., Quectel, Agosto 2014.
- [89] *RFM95W/96W/98W*, Ver. 2.0 ed., HopeRF Electronic, Julho 2019.
- [90] “*FatFs - Generic FAT Filesystem Module*,” <https://bit.ly/32HviBG>, Acedido: 23-02-2020.
- [91] *Section 21. UART*, Rev. h ed., Microchip Technology Inc, Fevereiro 2017.
- [92] Anacom, “Isenção de licença de estação,” 2018.
- [93] Silicon Labs, “*Calculation of occupied bandwidth for an FSK modulated signal*,” <https://bit.ly/2TkJu0D>, Acedido: 28-02-2020.
- [94] *Pico Technology*, “PicoScope® 2000 Series,” <https://bit.ly/3fo7rgi>, Acedido: 28-04-2020.
- [95] *The Qt Company*, “*Why Qt?*” <https://bit.ly/2WFkpO8>, Acedido: 28-04-2020.



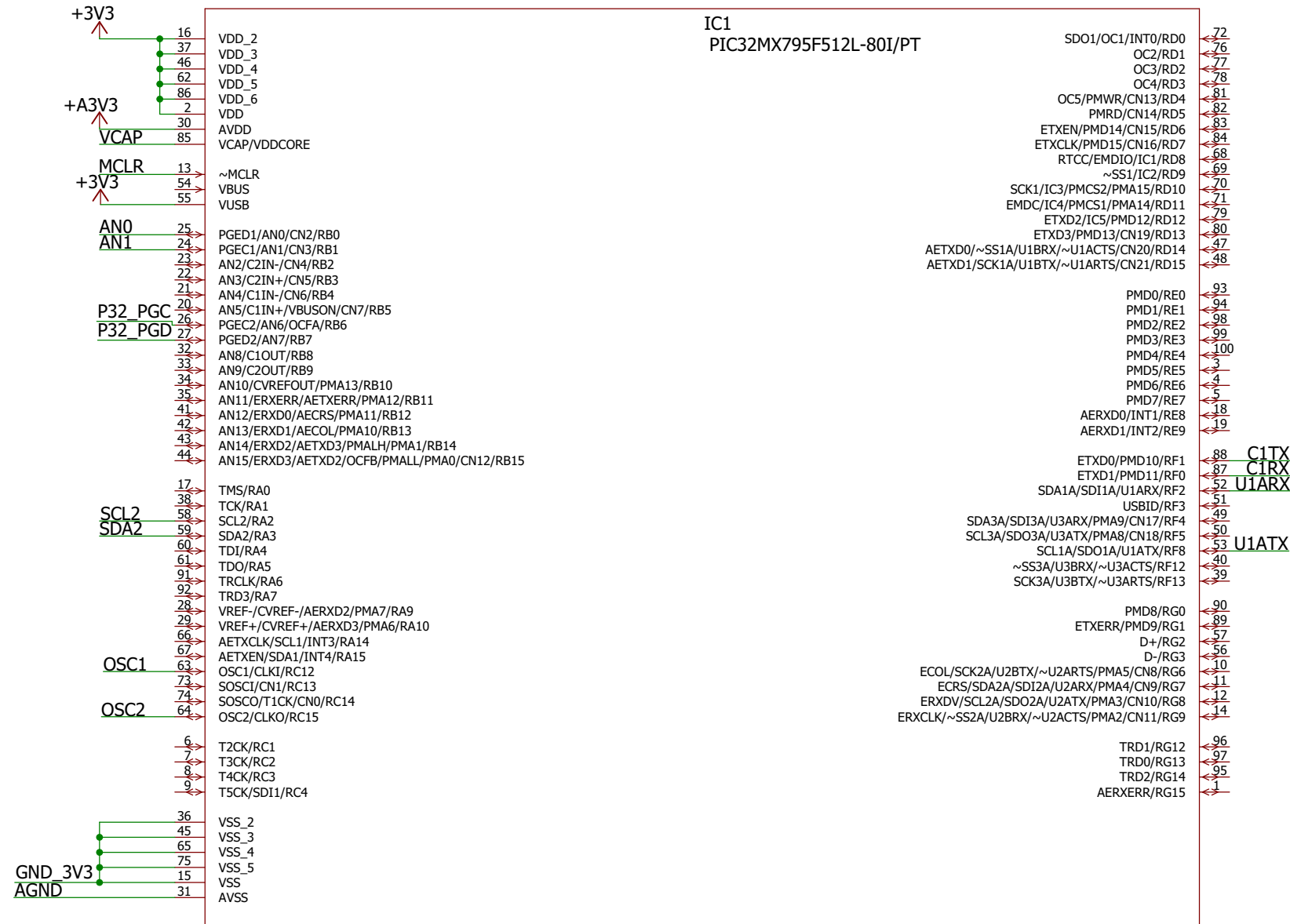
## Anexo A

# Módulo para sensores adicionais

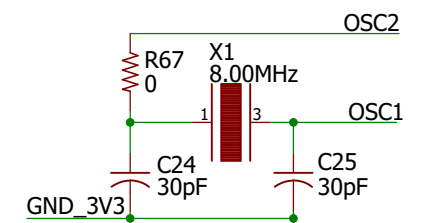
### A.1 Esquemático



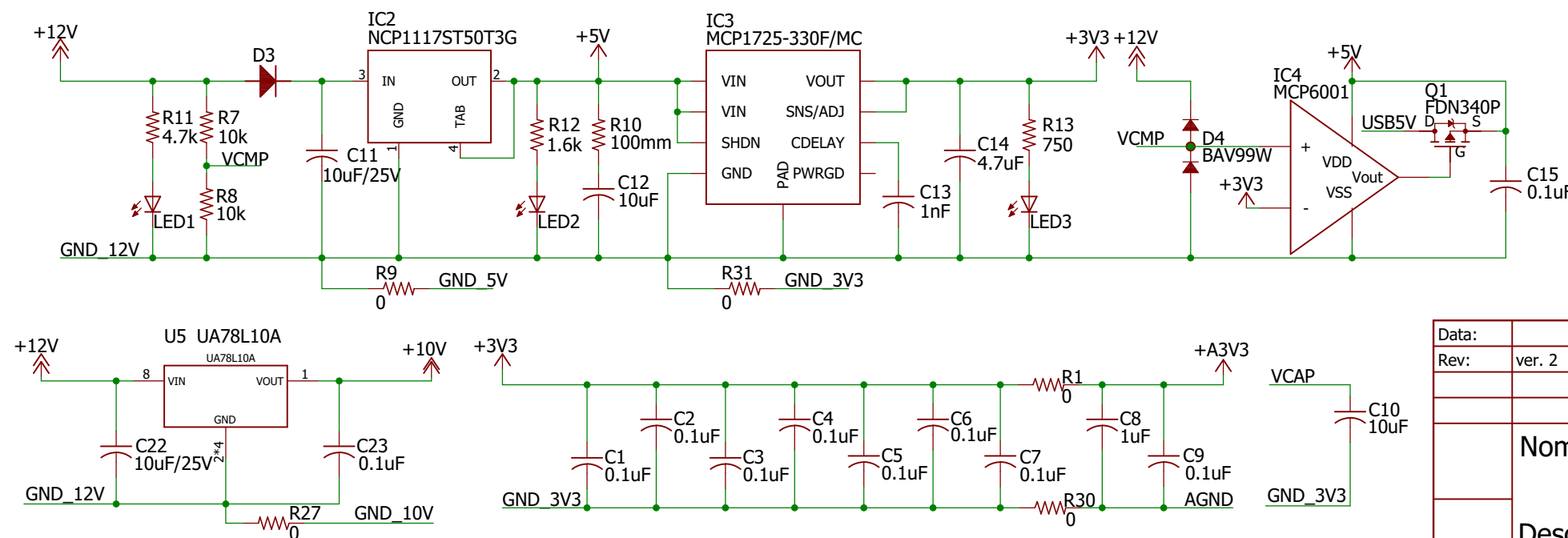
# Microcontrolador



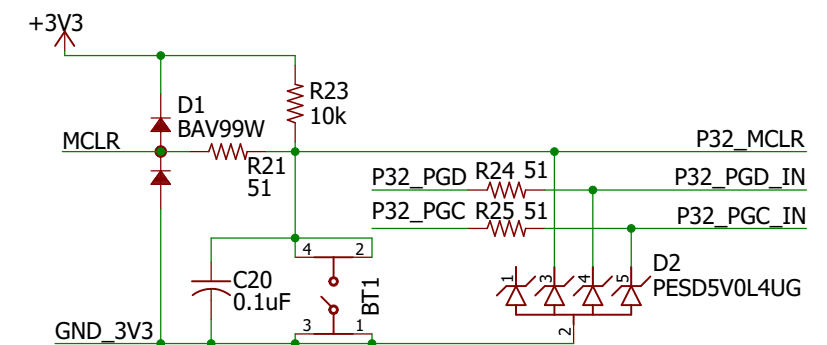
## Oscilador



## Alimentação do sistema



## ICSP



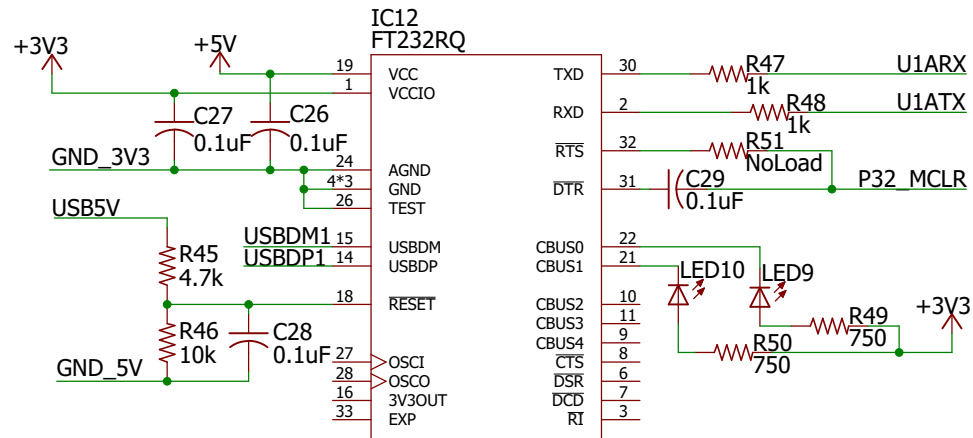
Data:	Projeto:	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev: ver. 2	Sistema de Telemetria para concurso MotoStudent	Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
	Nome da PCB: Módulo para sensores adicionais	Página: 1
	Descrição da folha: Microcontrolador e alimentação do sistema	Número de Páginas: 3

Luís Emanuel Ferreira Cristóvão

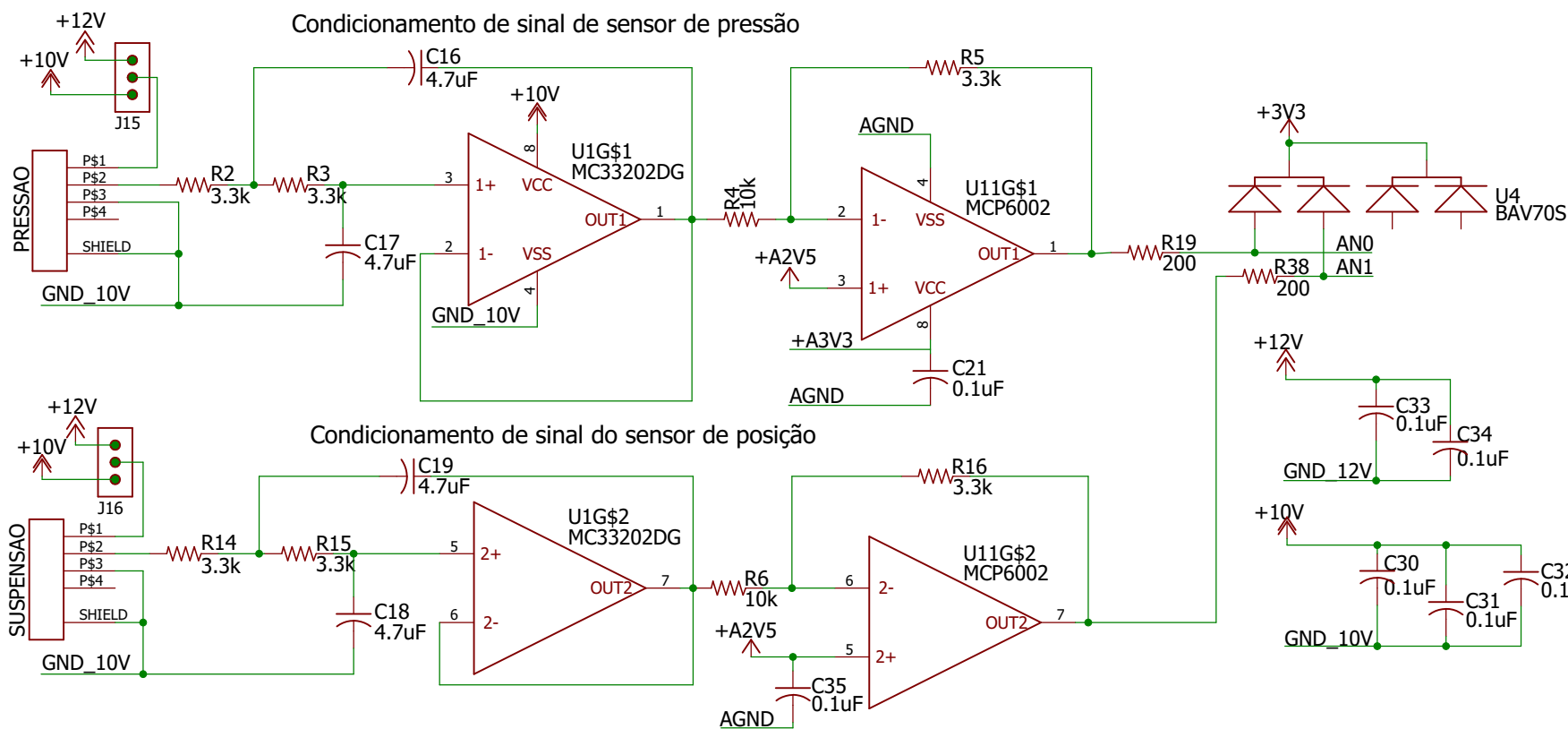
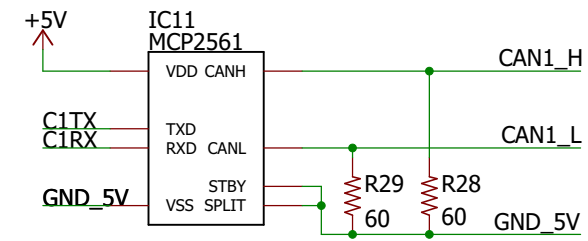
*Dissertação de Mestrado*



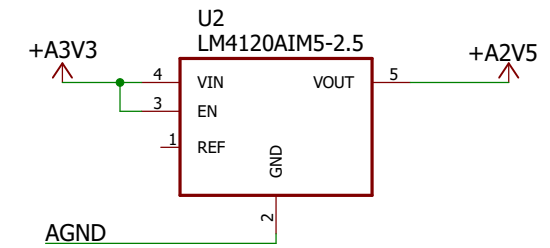
## USB MINI - INTERFACE UART1



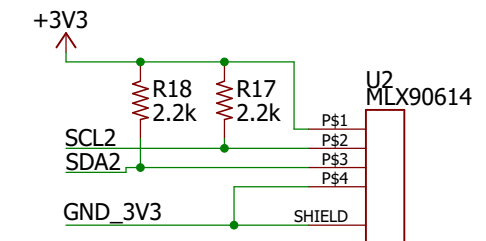
## Transceiver CAN1 - Sensores Adicionais



## Fonte de referência 2.5V



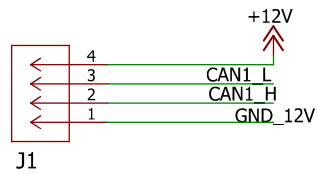
## I2C - Sensor IR



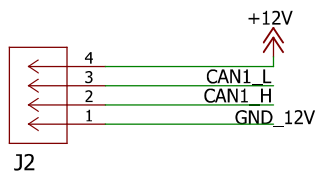
Data:		Projeto:	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 2	Sistema de Telemetria para concurso MotoStudent	Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
		Nome da PCB: Módulo para sensores adicionais	Página:  2
		Descrição da folha: Interfaces	Número de Páginas: 3



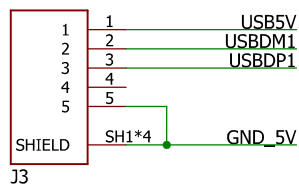
## Conector Sensores adicionais - IN



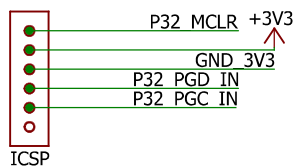
## Conector Sensores adicionais - OUT



## USB MICRO



## Porta de programação e debug

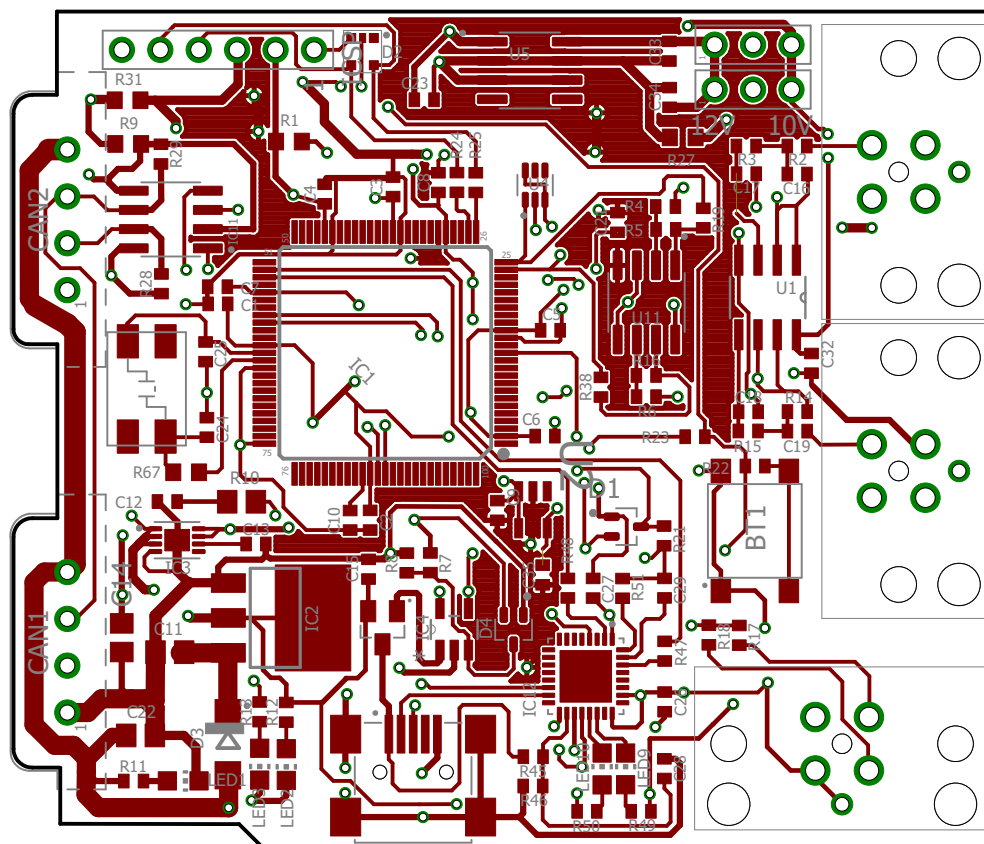


Data:		Projeto:	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 2	Sistema de Telemetria para concurso MotoStudent	Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
			Página:  3
	Nome da PCB: Módulo para sensores adicionais		Número de Páginas: 3
	Descrição da folha: IO'S e conectores		



## A.2 *Layout* da PCB

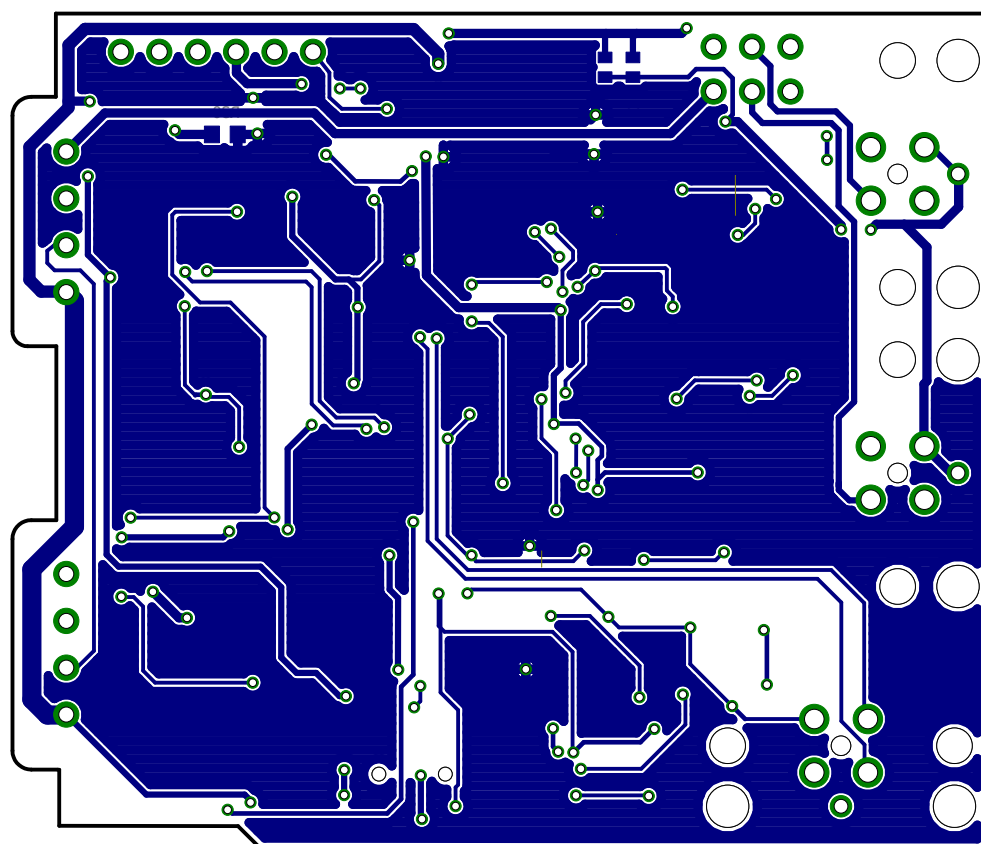




Data:	15/07/2019 14:10	Projeto: Sistema de Telemetria para concurso MotoStudent	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1		Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Escala:	2/1		
	Nome da PCB:Módulo para sensores adicionais		Página:
	Descrição da folha:Camada Superior		1
			Número de Páginas: 2







Data:	15/07/2019 14:10	Projeto:	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1	Sistema de Telemetria para concurso MotoStudent	Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Escala:	2/1		
		Nome da PCB: Módulo para sensores adicionais	Página:  2
		Descrição da folha: Camada Inferior	Número de Páginas: 2



### A.3 Custos de produção da PCB

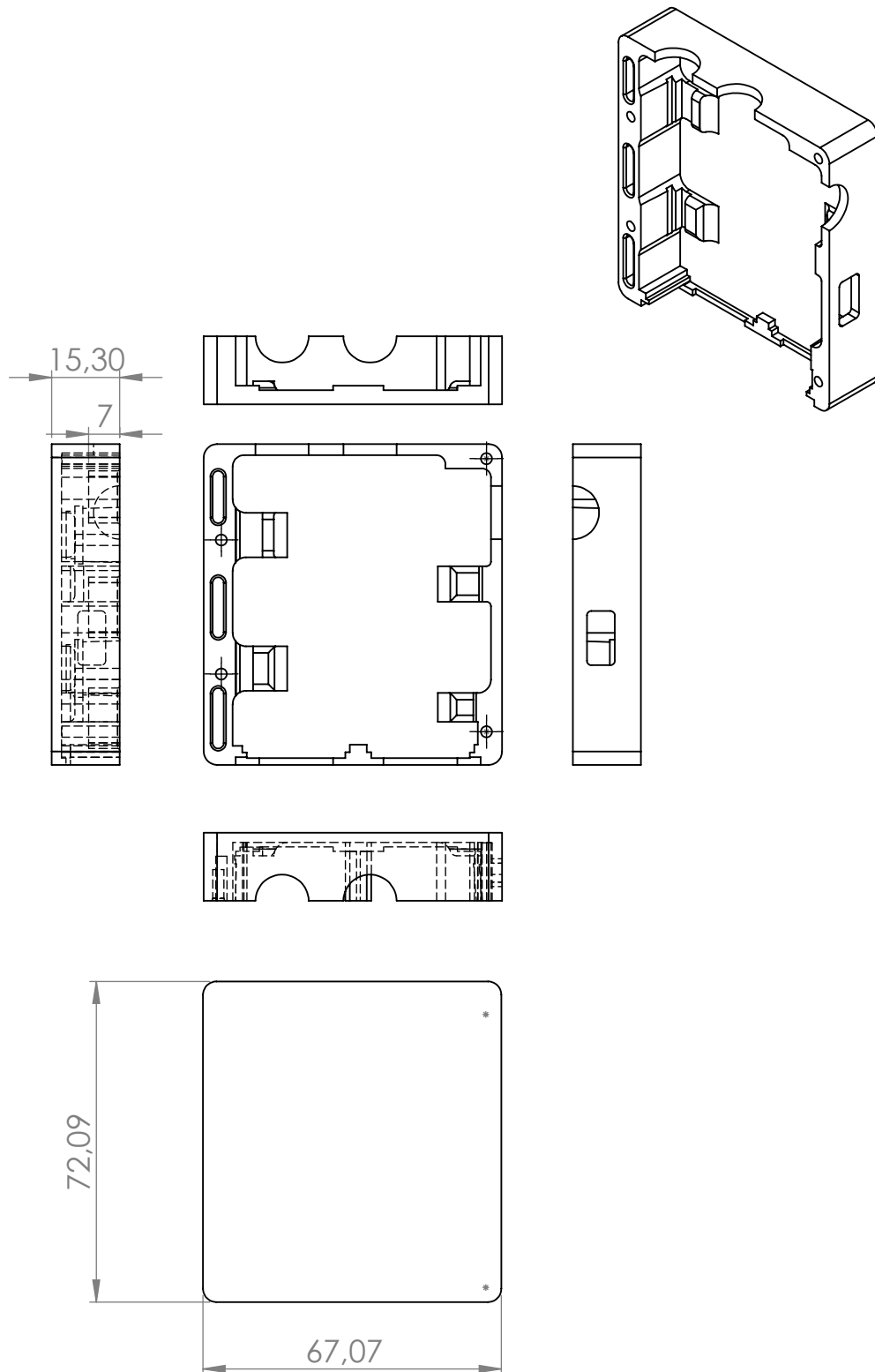
Tabela A.1: Custos de produção do módulo de sensores adicionais.

Item	Descrição	Quant.	Preço /Unit.	Preço Total
PIC32MX795F512L I/PT	Microcontrolador PIC32	1	8,380 €	8,380 €
GRM155R61A104KA01D	Condensador cerâmico; 100nF; SMD 0402	21	0,007 €	0,156 €
CL05A105KP5NNNC	Condensador cerâmico; 1uF; SMD 0402	1	0,020 €	0,020 €
GRM155R60J106ME44D	Condensador cerâmico; 10uF; SMD 0402	2	0,168 €	0,336 €
CL21A106KAYNNNG	Condensador cerâmico; 10uF; SMD 0805	2	0,187 €	0,374 €
VJ0402Y102KXACW1BC	Condensador cerâmico; 1nF; SMD 0402	1	0,009 €	0,009 €
CL21A475KOFNNNE	Condensador cerâmico; 4,7uF; SMD 0805	5	0,126 €	0,645 €
C0402C300J5GAC	Condensador cerâmico; 30pF; SMD 0402	2	0,027 €	0,053 €
SMD0603-0R	Resistência; 0Ω; SMD 0603; 0,1W; ±5%	2	0,009 €	0,018 €
SMD0402-2K2-1%	Resistência; 2,2kΩ; SMD 0402; 63mW; ±1%	6	0,009 €	0,054 €
SMD0402-10K-1%	Resistência; 10kΩ; SMD 0402; 63mW; ±1%	6	0,009 €	0,054 €
SMD0402-3K3-1%	Resistência; 3,3kΩ; SMD 0402; 63mW; ±1%	6	0,009 €	0,054 €
CS0805-0R1-1%-LTC	Resistência; 0,1kΩ; SMD 0805; 125mW; ±1%	1	0,097 €	0,097 €
SMD0402-4K7-1%	Resistência; 4,7kΩ; SMD 0402; 63mW; ±1%	2	0,009 €	0,018 €
SMD0402-1K6-1%	Resistência; 1,6kΩ; SMD 0402; 63mW; ±1%	1	0,009 €	0,009 €
SMD0402-750R-1%	Resistência; 750Ω; SMD 0402; 63mW; ±1%	3	0,009 €	0,027 €
SMD0402-200R-1%	Resistência; 200Ω; SMD 0402; 63mW; ±1%	3	0,009 €	0,027 €
SMD0402-51R-1%	Resistência; 51Ω; SMD 0402; 63mW; ±1%	3	0,009 €	0,027 €
SMD0402-60R-1%	Resistência; 60Ω; SMD 0402; 63mW; ±1%	2	0,009 €	0,018 €
SMD0402-1K-1%	Resistência; 1kΩ; SMD 0402; 63mW; ±1%	2	0,009 €	0,018 €
TACTM-69N-F	Botão 2 posições; 0,05A/12VDC; SMT	1	0,051 €	0,051 €
BAV99W	Díodo comutador; 100V; 150mA; SMD SOT232	2	0,022 €	0,045 €
PESD5V0L4UG	Díodo; 6,8V; 2,5A; SMD SOT353	1	0,121 €	0,121 €
SK24-DIO	Díodo retificador Schottky; 40V; 2A; SMD	1	0,068 €	0,068 €
BAV70S.115	Díodo comutador; 100V; 250mA; SMD	1	0,060 €	0,060 €
KP-2012LSURCK	LED vermelho; 1,75-2,5V; SMD 0805	5	0,096 €	0,482 €
FDN342P	Transístor P-MOSFET; 20V; 2A; 0,5W; SOT-3	1	0,145 €	0,145 €
NCP1117ST50T3G	Regulador de tensão; 5V; 1A; SOT223	1	0,288 €	0,288 €
MCP1725-3302E/MC	Regulador de tensão; 3,3V	1	0,450 €	0,450 €
MCP6001T-I/OT	Amplificador operacional; 1MHz; 1,8÷5,5V; 1 Canal; SOT23-5	1	0,203 €	0,203 €
MCP2561-E/SN	Transceiver CAN; Canais: 1; 1Mbps; SO8	1	0,810 €	0,810 €
FT232RQ	Interface USB-UART; Full Speed; 3,3÷5VDC; QFN32	1	4,000 €	4,000 €
MCP6002T-I/SN	Amplificador operacional; 1MHz; 1,8÷5,5V; Canais: 2; SO8	1	0,480 €	0,480 €
UA78L10ACD	Regulador de tensão; 10V; 0,1A; SO8	1	0,235 €	0,235 €
MC33202DR2G	Amplificador operacional; 2,2MHz; 2÷12V; 2 Canais; SO8	1	0,480 €	0,480 €
LM4120AIM5-2.5/NOP	Regulador de tensão 2,5V +-0,2%	1	1,690 €	1,690 €
1-967657-1	Conetor AMP 4 pinos THT	2	6,370 €	12,740 €
MX-67803-8020	Tomada; USB AB mini; horizontal; SMT	1	0,660 €	0,660 €
T4145435041-001	Conetor M12; 4 pinos THT; Shield	3	10,530 €	31,590 €
8.00M-CFPX104	Ressonador: de quartzo; 8MHz; ±20ppm; 18pF; SMD	1	1,021 €	1,021 €
PCB-55*65	PCB com transporte	1	2,070 €	2,070 €
Total				68,27 €



## A.4 Desenhos técnicos caixa de proteção

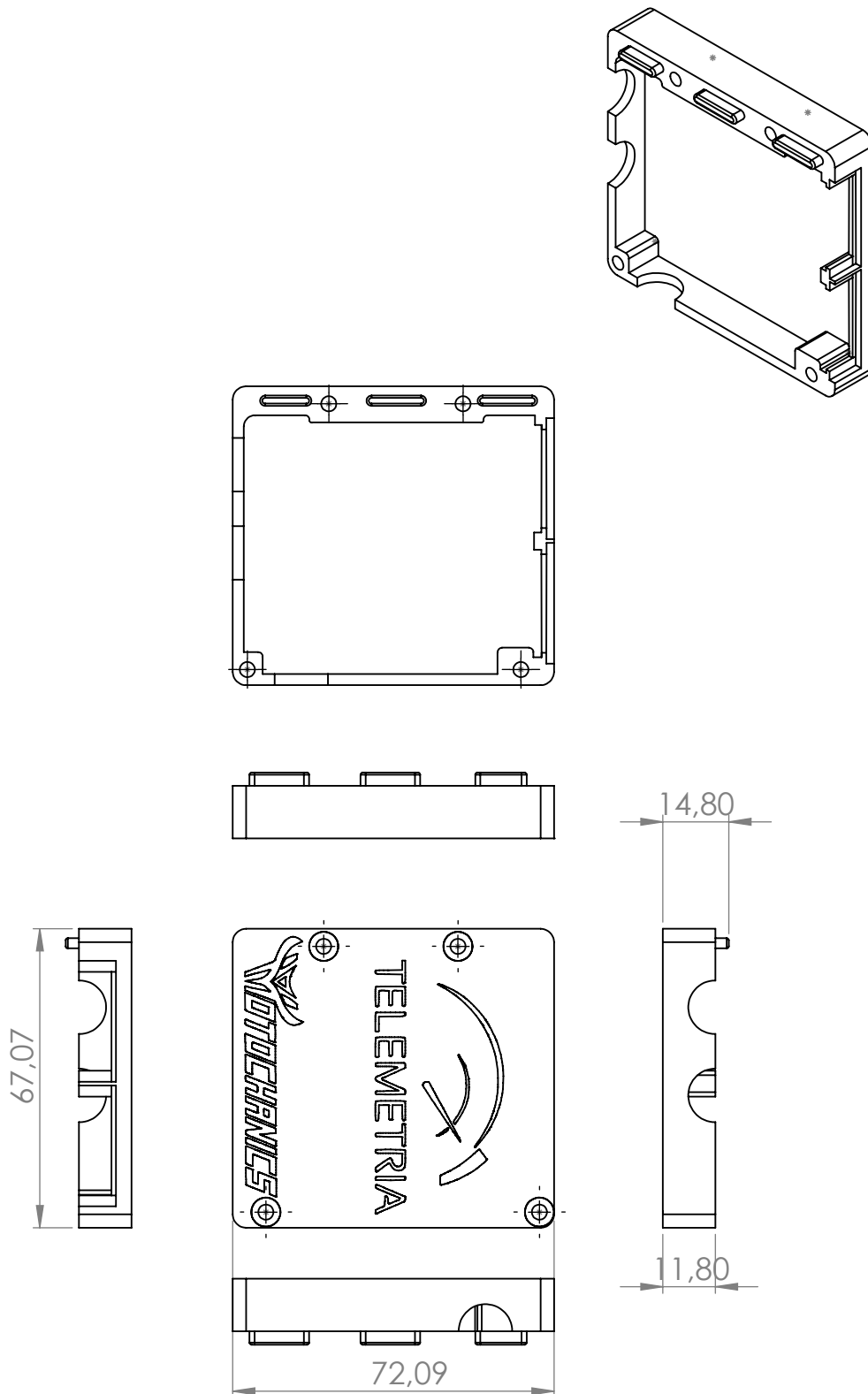




Data:	23/8/2019 17:26	Projeto: <b>Sistema de Telemetria para concurso MotoStudent</b>	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1		Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Escala:	2/3		
Material:	PLA		
	Nome: Caixa de proteção para módulo de sensores adicionais		Página:
	Descrição da folha: Base da caixa		1
			Número de Páginas: 2







Data:	23/8/2019 17:46	Projeto: <b>Sistema de Telemetria para concurso MotoStudent</b>	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1		Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Escala:	2/3		
Material:	PLA		
	Nome: Caixa de proteção para módulo de sensores adicionais		Página:
	Descrição da folha: Tapa da caixa		2
			Número de Páginas: 2



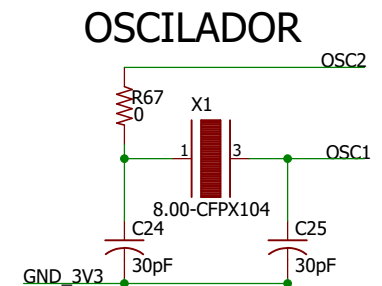
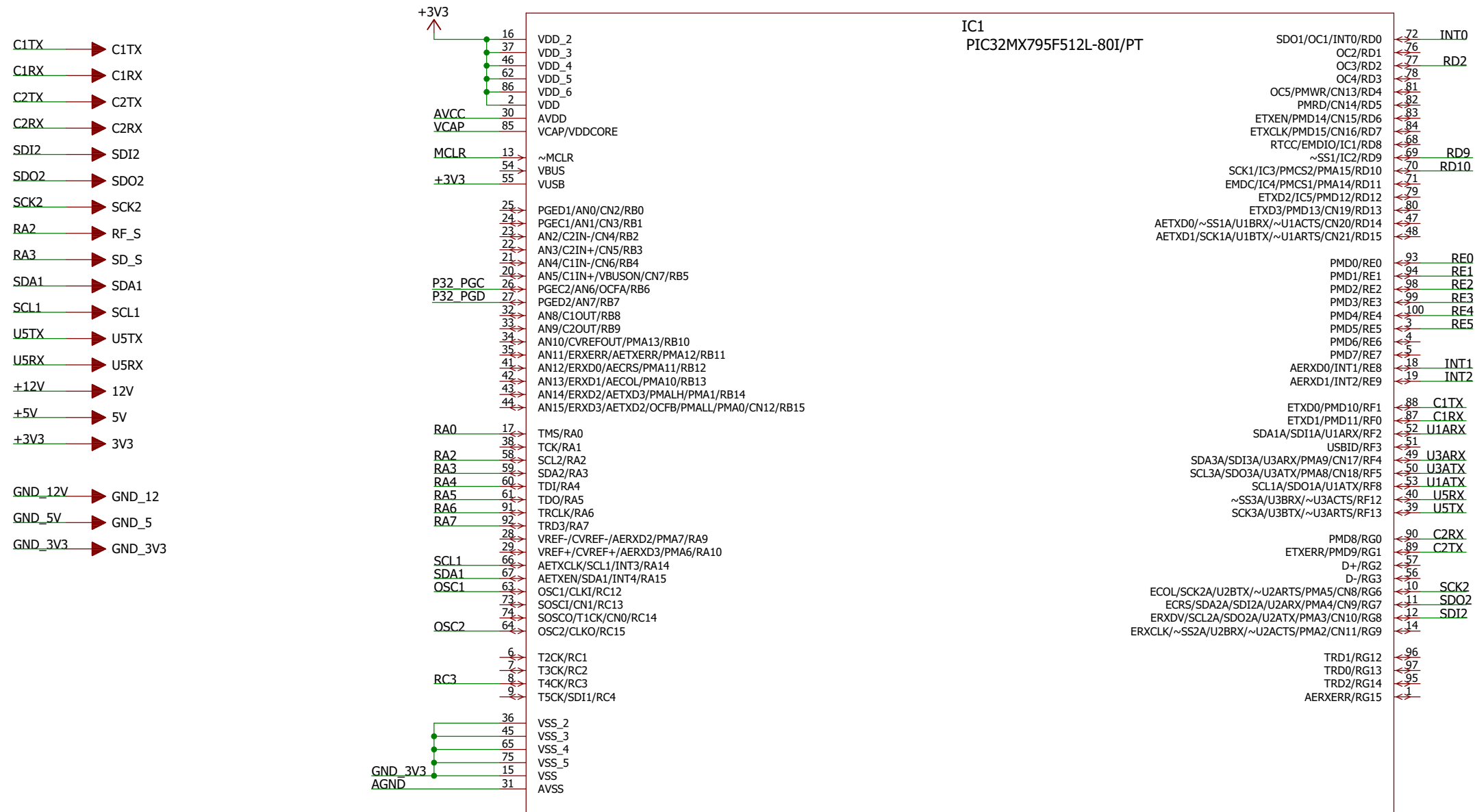
## Anexo B

# Sistema de aquisição, gravação e transmissão de dados

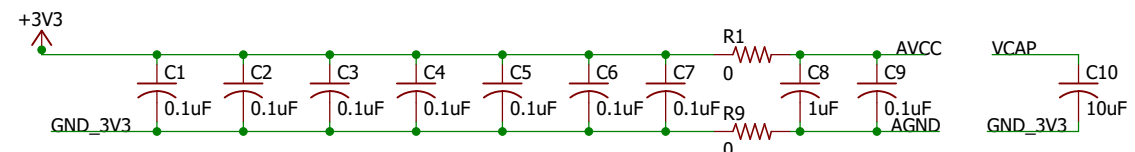
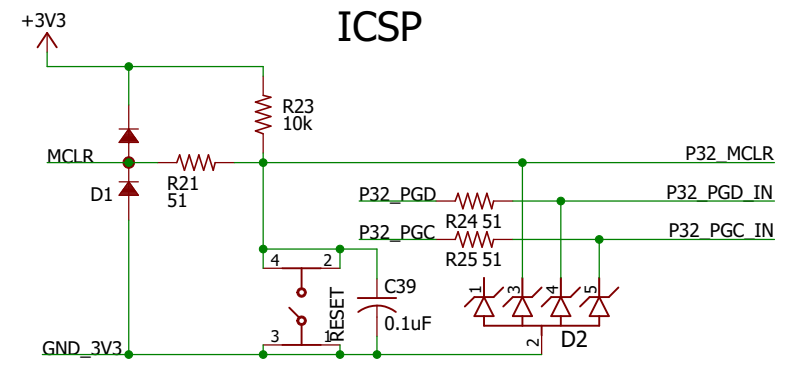
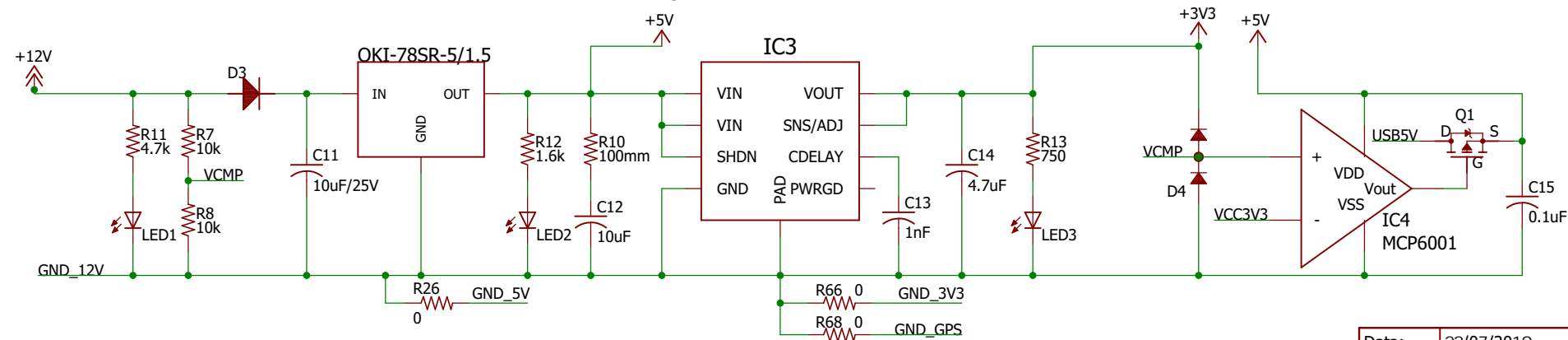
### B.1 Esquemático



# Microcontrolador



## ALIMENTAÇÃO DO SISTEMA

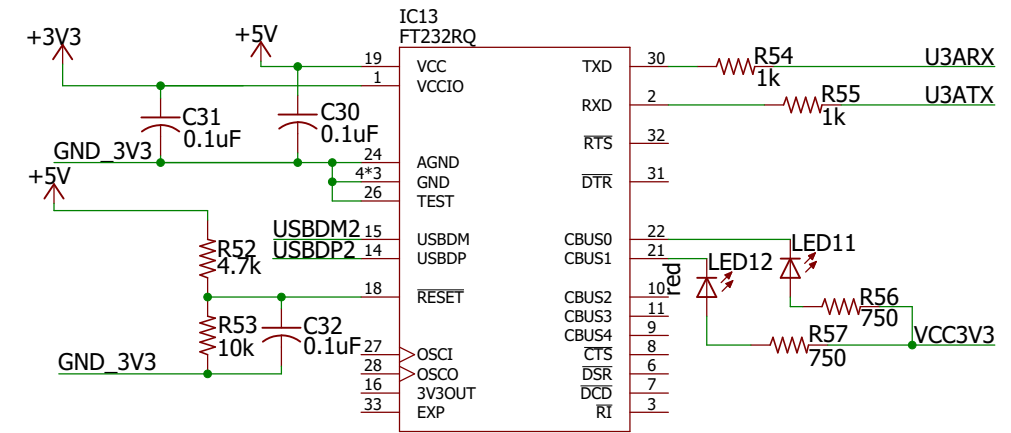


Data:	22/07/2019	Projeto:	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 2	Sistema de Telemetria para concurso MotoStudent	Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Nome da PCB: Sistema de aquisição, gravação e transmissão de dados		Página:  1	
Descrição da folha: Microcontrolador e alimentação do sistema		Número de Páginas: 3	

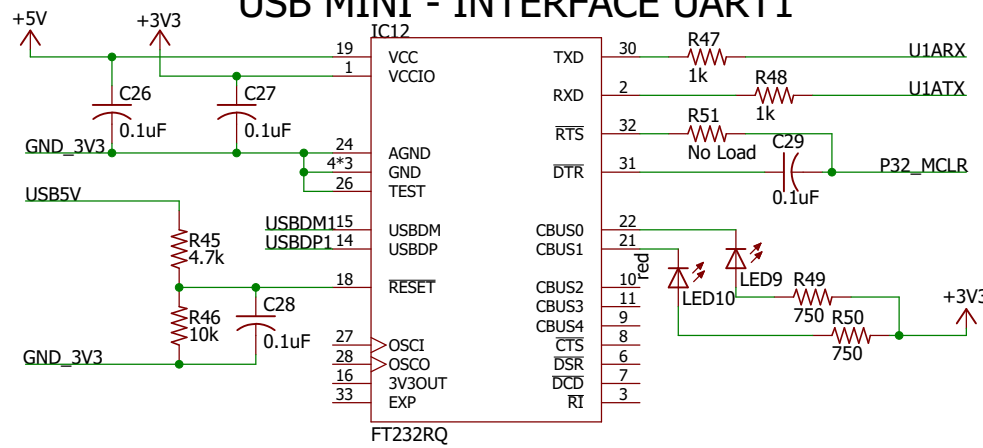
Luís Emanuel Ferreira Cristóvão

*Dissertação de Mestrado*

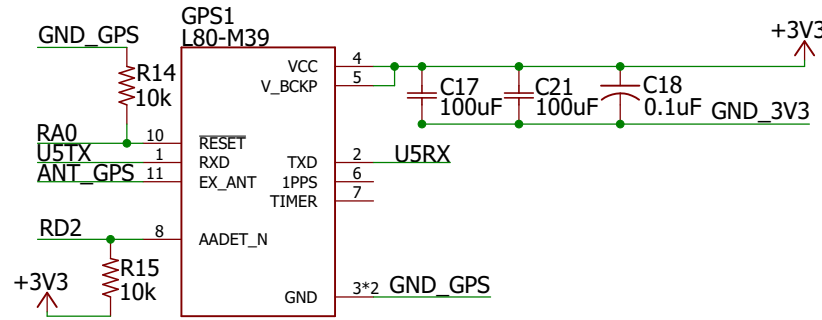
ECRÃ - INTERFACE UART2



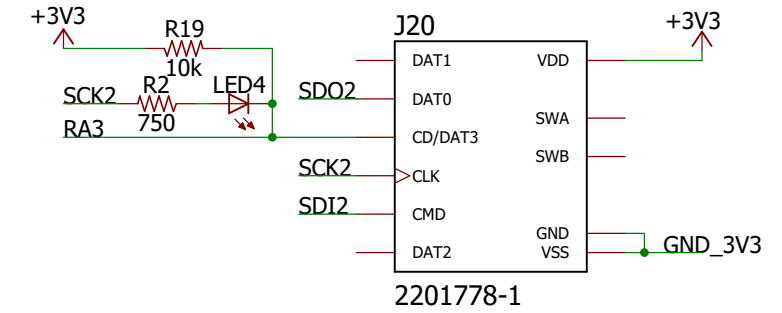
USB MINI - INTERFACE UART1



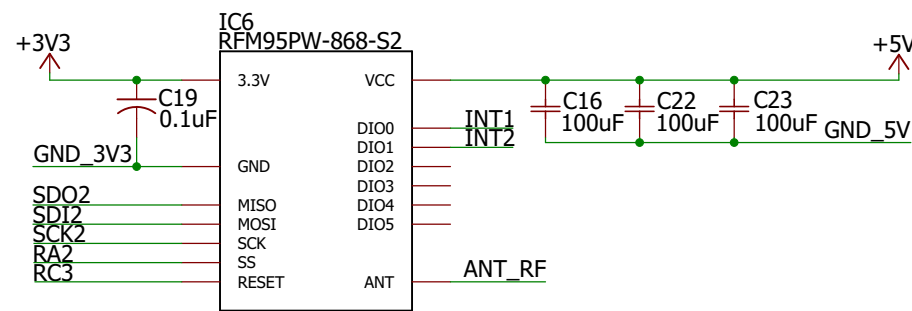
Módulo GPS - INTERFACE UART 5



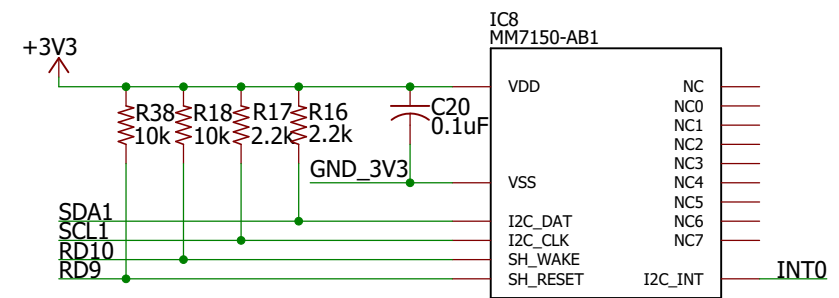
Leitura Cartão SD - INTERFACE SPI2



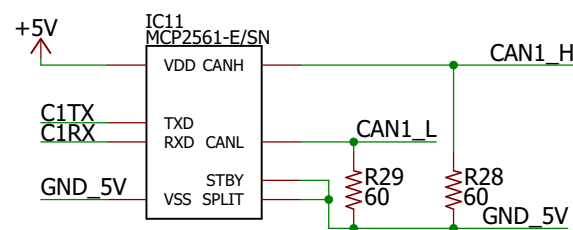
Módulo Rádio RFM95PW - INTERFACE SPI2



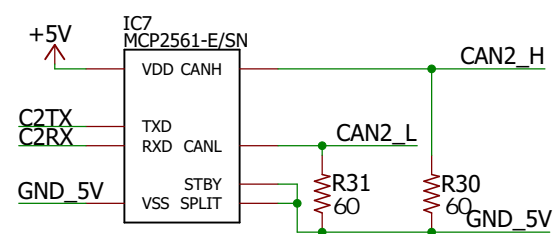
Módulo Inercial - INTERFACE I2C1



Transceiver CAN1 - ECU KTM



Transceiver CAN2 - Sensores Adicionais



Data:	22/07/2019	Projeto:	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 2		
		Nome da PCB: Sistema de aquisição, gravação e transmissão de dados	Página:  2

Sistema de Telemetria para  
concurso MotoStudent

Orientador: Prof. Pedro Fonseca  
Co-orientador: Prof. Paulo Pedreiras

---

Luís Emanuel Ferreira Cristóvão

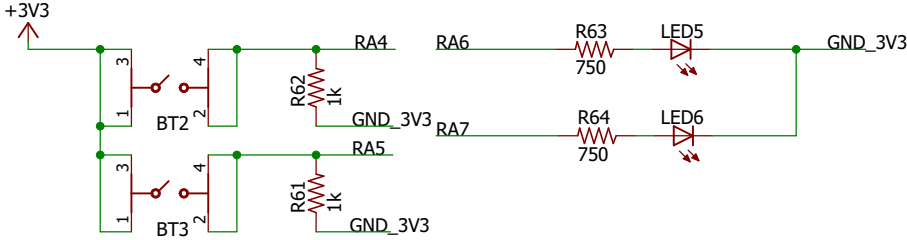
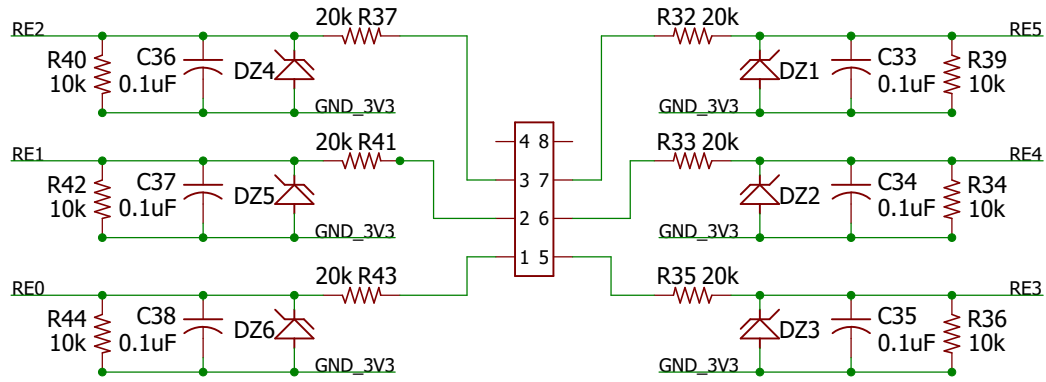
*Dissertação de Mestrado*

---

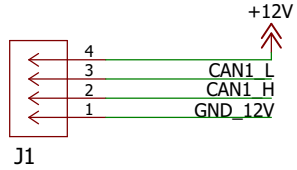


Botões e sinalização de controlo de transmissão e gravação de dados

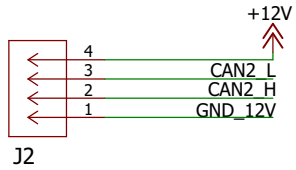
Entradas e Saidas Digitais Adicionais



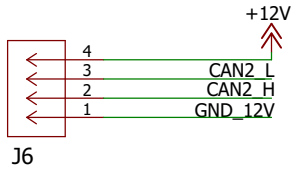
Conector mota e CAN1



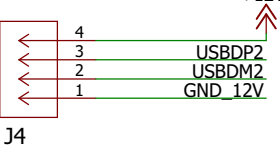
Conector sensores externos e CAN2-1



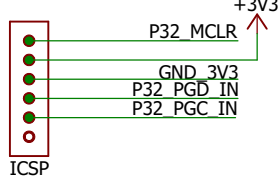
Conector sensores externos e CAN2-2



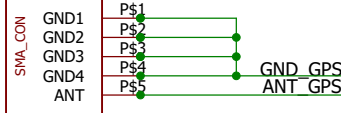
Conector Dashboard



Porta de programação e debug



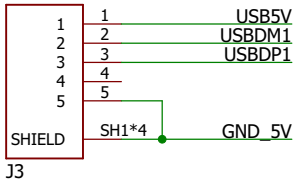
Conector Antena de GPS



Conector Antena do módulo rádio



USB MICRO



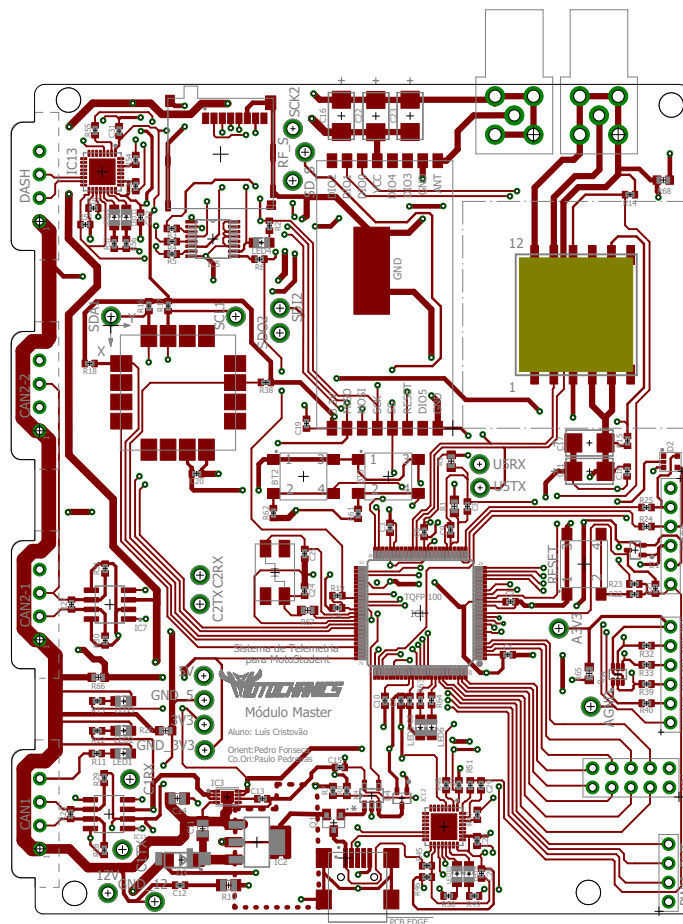
Data:	22/07/2019	Projeto: Sistema de Telemetria para concurso MotoStudent	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 2		Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Nome da PCB: Sistema de aquisição, gravação e transmissão de dados		Descrição da folha: IO's e Connectores	Página:  3
			Número de Páginas: 3

Luís Emanuel Ferreira Cristóvão

*Dissertação de Mestrado*

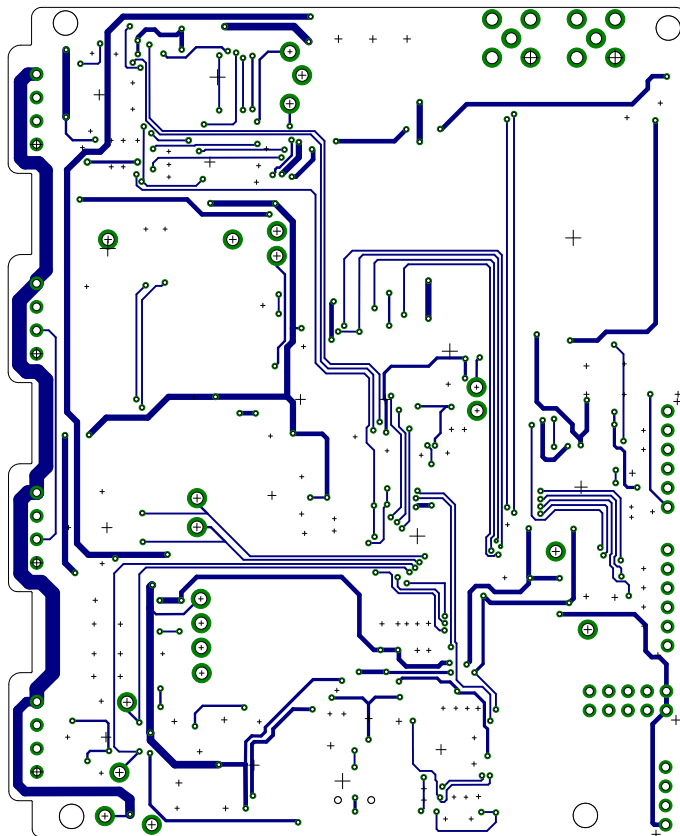
## **B.2 *Layout* da PCB**





Data:	22/07/2020 21:06	Projeto: Sistema de Telemetria para concurso MotoStudent	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1		Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Escala:	1/1		
	Nome da PCB: Sistema de aquisição, gravação e transmissão de dados		Página:  1
	Descrição da folha: Camada Superior		Número de Páginas: 2





Data:	22/07/2020 21:06	Projeto: <b>Sistema de Telemetria para concurso MotoStudent</b>	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1		Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Escala:	1/1		
	Nome da PCB: Sistema de aquisição, gravação e transmissão de dados		Página:  <b>2</b>
	Descrição da folha: Camada Inferior		Número de Páginas: 2





### B.3 Custo de produção da PCB

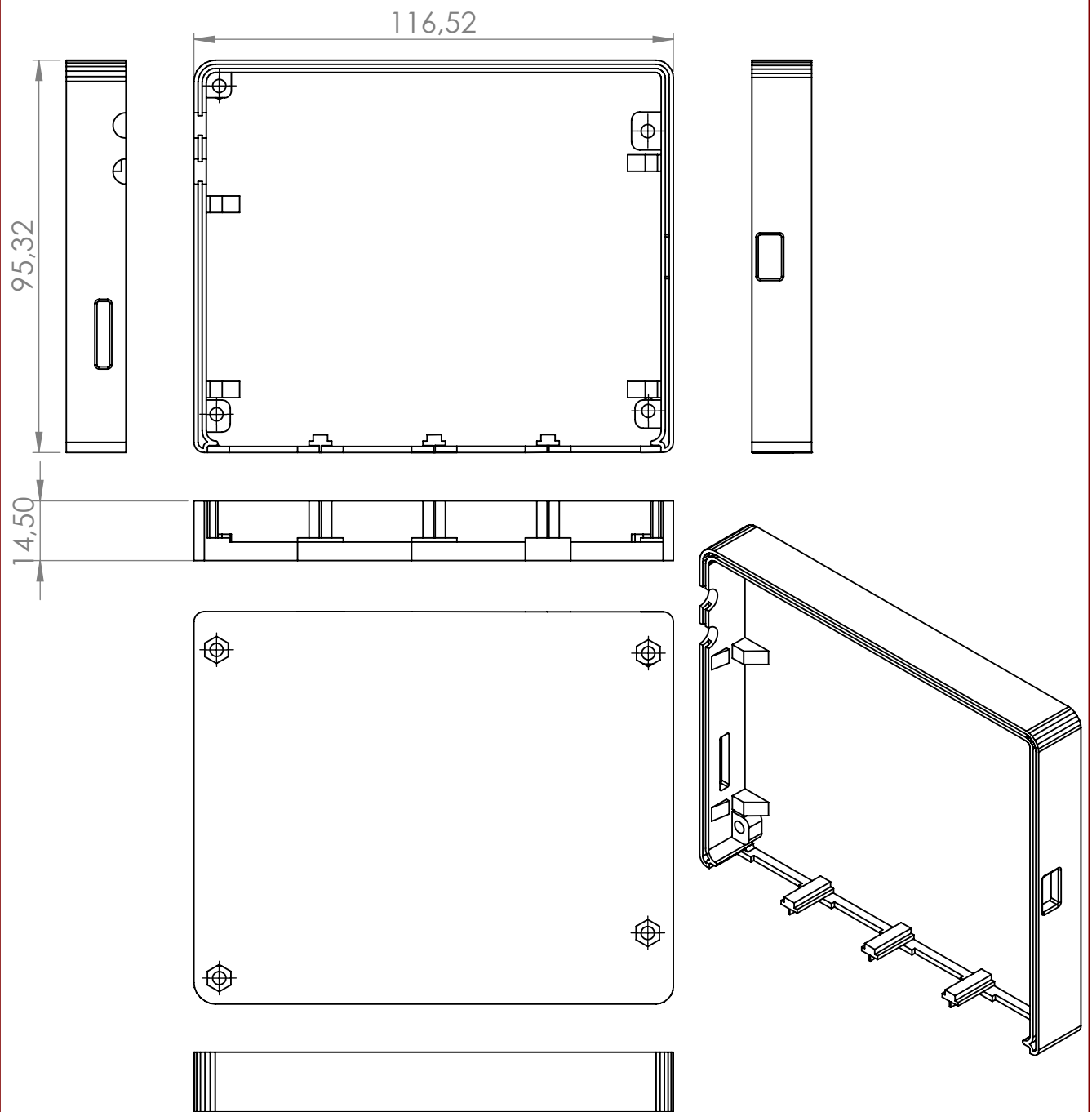
Tabela B.1: Custos de produção do sistema para aquisição, recolha e transmissão de dados.

Item	Descrição	Quant.	Preço /Unit.	Preço Total
PIC32MX795F512L 801/PT	Microcontrolador PIC32	1	8,380 €	8,38 €
GRM155R61A104KA01D	Condensador cerâmico; 100nF; SMD 0402	19	0,007 €	0,14 €
CL05A105KP5NNNC	Condensador cerâmico; 1uF; SMD 0402	1	0,020 €	0,02 €
GRM155R60J106ME44D	Condensador cerâmico; 10uF; SMD 0402	2	0,168 €	0,34 €
CL21A106KAYNNNG	Condensador cerâmico; 10uF; 25V; SMD 0805	1	0,187 €	0,19 €
VJ0402Y102KXACW1BC	Condensador cerâmico; 1nF; SMD 0402	1	0,009 €	0,01 €
CL21A475KOFNNNE	Condensador cerâmico; 4,7uF; SMD 0805	1	0,126 €	0,13 €
T491C107K006AT	Condensador de tantalio; 100uF; SMD 2312	5	0,280 €	1,40 €
SMD0603-0R	Resistência; 0Ω; SMD 0603; 0,1W; ±5%	16	0,009 €	0,15 €
SMD0402-750R-1%	Resistência; 750Ω; SMD 0402; 63mW; ±1%	3	0,009 €	0,03 €
SMD0402-2K2-1%	Resistência; 2,2kΩ; SMD 0402; 63mW; ±1%	2	0,009 €	0,02 €
SMD0402-10K-1%	Resistência; 10kΩ; SMD 0402; 63mW; ±1%	4	0,009 €	0,04 €
CS0805-0R1-1%-LTC	Resistência; 100mΩ; SMD 0805; 0,125W; ±1%	1	0,097 €	0,10 €
SMD0402-4K7-1%	Resistência; 4,7kΩ; SMD 0402; 63mW; ±1	2	0,009 €	0,02 €
SMD0402-1K6-1%	Resistência; 1,6kΩ; SMD 0402; 63mW; ±1	1	0,009 €	0,01 €
SMD0402-120R-1%	Resistência; 120Ω; SMD 0402; 63mW; ±1	2	0,009 €	0,02 €
SMD0402-51R-1%	Resistência; 51Ω; SMD 0402; 63mW; ±1	3	0,009 €	0,03 €
SMD0402-200R-1%	Resistência; 200Ω; SMD 0402; 63mW; ±1	5	0,009 €	0,04 €
SMD0402-60R-1%	Resistência; 60Ω; SMD 0402; 63mW; ±1	4	0,009 €	0,04 €
SMD0402-1K-1%	Resistência; 1kΩ; SMD 0402; 63mW; ±1	2	0,009 €	0,02 €
TACTM-69N-F	Botão 2 posições; 0,05A/12VDC; SMT	1	0,051 €	0,05 €
BAV99W	Díodo comutador; 100V; 150mA; SMD SOT232	1	0,022 €	0,02 €
PESD5V0L4UG	Díodo; 6,8V; 2,5A; SMD SOT353	1	0,121 €	0,12 €
SK24-DIO	Díodo retificador Schottky; 40V; 2A; SMD	1	0,068 €	0,07 €
BAV70S.115	Díodo comutador; 100V; 250mA; SMD	1	0,060 €	0,06 €
KP-2012LSURCK	LED vermelho; 1,75-2,5V; SMD 0805	10	0,096 €	0,96 €
FDN342P	Transístor P-MOSFET; 20V; 2A; 0,5W; SOT-3	1	0,145 €	0,14 €
5-1814400-1	Tomada SMA; fêmea; angulares 90°; 50Ω; THT	2	3,160 €	6,32 €
NCP1117ST50T3G	Regulador de tensão; 5V; 1A; SOT223	1	7,370 €	7,37 €
MCP1725-3302E/MC	Regulador de tensão; 3,3V	1	0,450 €	0,45 €
MCP6001T-I/OT	Amplificador operacional; 1MHz; 1,8÷5,5V; 1 Canal	1	0,203 €	0,20 €
74ABT125PW.112	Inversor lógico; Canais 4; SMD TSSOP-14	1	0,512 €	0,58 €
RFM95PW-868S2	Transceiver RF; FSK,GFSK,GMSK,LoRa,OOK; 868MHz	1	17,330 €	17,33 €
MCP2561-E/SN	Transceiver CAN; Canais: 1; 1Mbps; SO8	2	0,810 €	1,62 €
MM71150-AB1	MM7150 Motion Module	1	21,201 €	21,21 €
FT232RQ	Interface USB-UART; Full Speed; 3,3÷5VDC; QFN32	2	4,000 €	8,00 €
908-4085	Modulo GPS Quectel L80 Update 10Hz	1	14,380 €	14,38 €
1-967657-1	Conector AMP 4 pinos THT	4	6,370 €	25,48 €
MX-67803-8020	Tomada; USB AB mini; horizontal; SMT	1	0,660 €	0,66 €
2201778-1	Conetor micro SD-CARD	1	1,960 €	1,96 €
8.00M-CFPX104	Ressonador: de quartzo; 8MHz; ±20ppm; 18pF; SMD	1	1,021 €	1,02 €
C0402C300J5GAC	Condensador cerâmico; 30pF; SMD 0402	2	0,027 €	0,05 €
KEYS5005	Pino para teste	22	0,240 €	5,27 €
PCB-90*110	PCB com transporte	1	5,470 €	5,47 €
Total				129,91 €



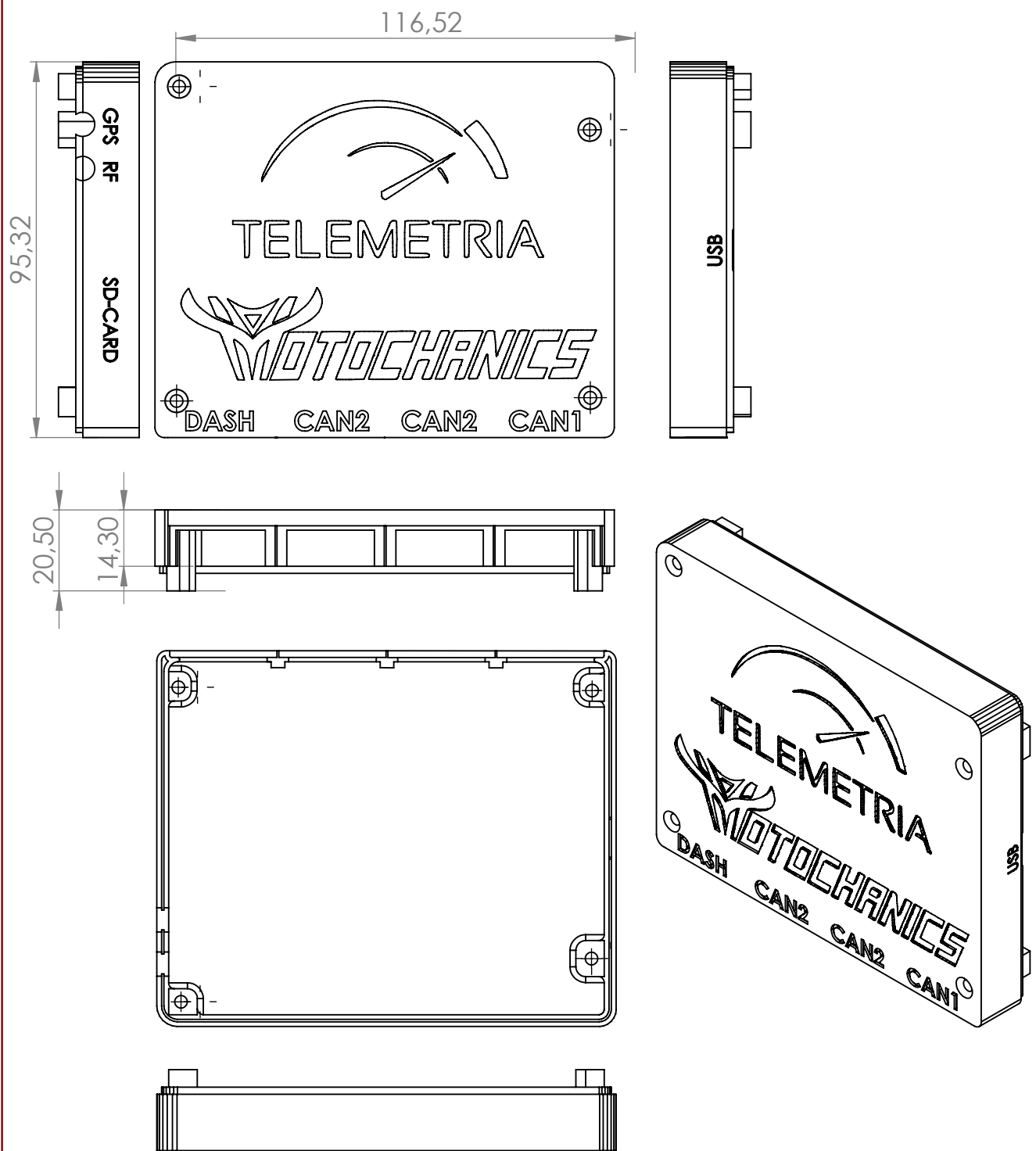
## **B.4 Desenhos técnicos caixa de proteção**





Data:	28/8/2019 11:47	Projeto: <b>Sistema de Telemetria para concurso MotoStudent</b>	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1		Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Escala:	2/3		
Material:	PLA		
	Nome: Caixa de proteção para sistema de aquisição, gravação e transmissão de dados		Página:  1
	Descrição da folha: Base da caixa		Número de Páginas: 2





Data:	28/8/2019 11:47	Projeto: Sistema de Telemetria para concurso MotoStudent	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1		Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Escala:	2/3		Página:
Material:	PLA		2
	Nome: Caixa de proteção para sistema de aquisição, gravação e transmissão de dados		Número de Páginas: 2
	Descrição da folha: Tapa da caixa		





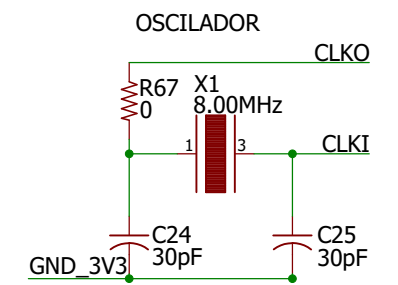
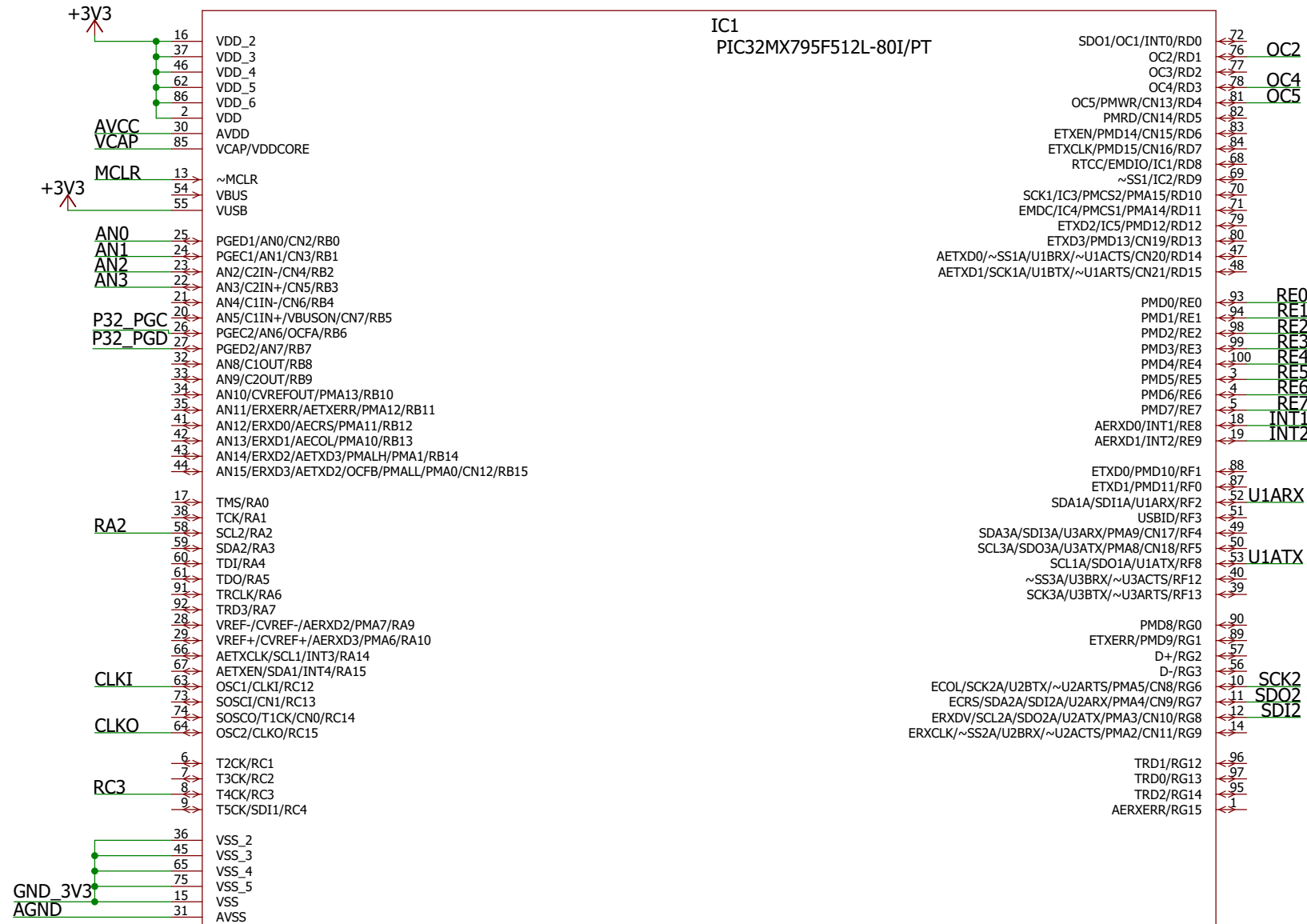
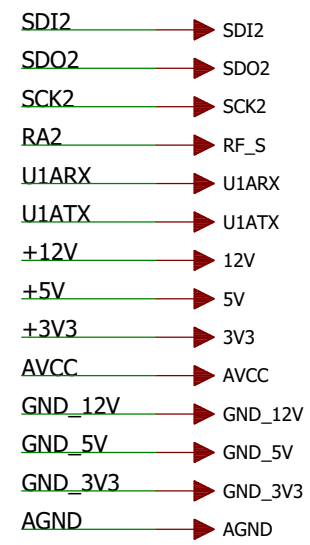
## Anexo C

# Sistema para recepção de dados

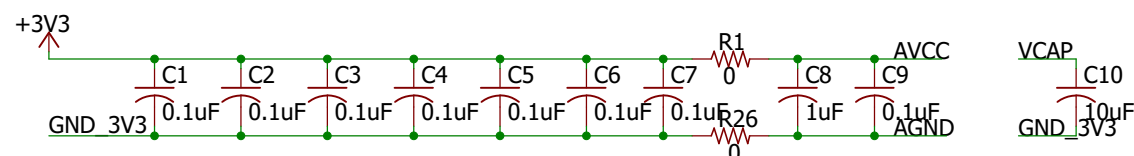
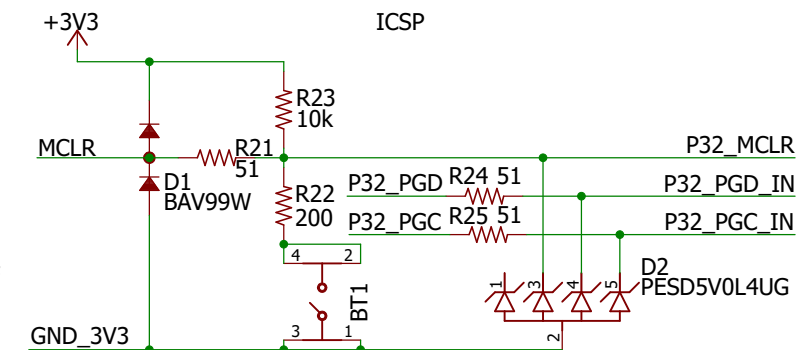
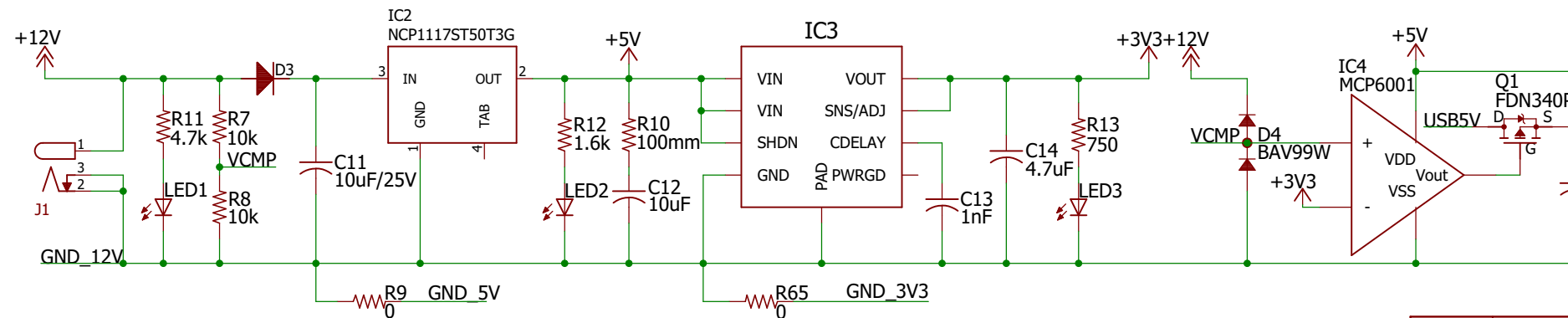
### C.1 Esquemático



Microcontrolador



ALIMENTAÇÃO DO SISTEMA

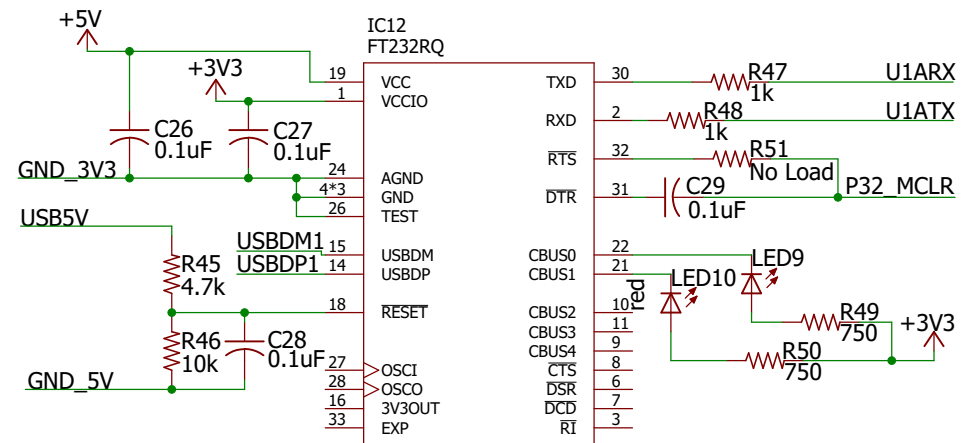


Data:	29/07/2020 21:54	Projeto:	Luis Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1	Sistema de Telemetria para concurso MotoStudent	Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Nome da PCB: Sistema de recepção de dados		Página:  1	
Descrição da folha: Microcontrolador e alimentação do sistema		Número de Páginas:  2	

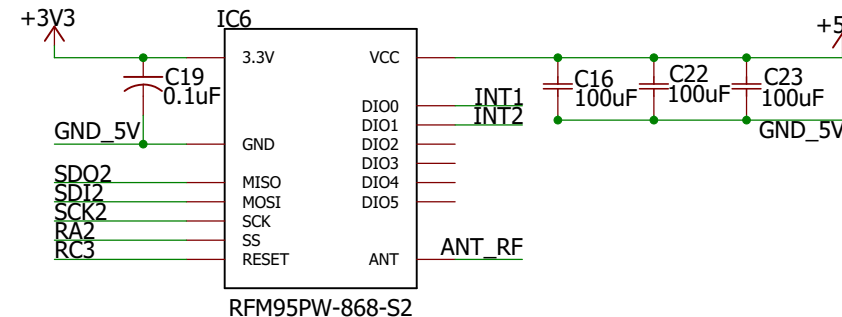
Luís Emanuel Ferreira Cristóvão

*Dissertação de Mestrado*

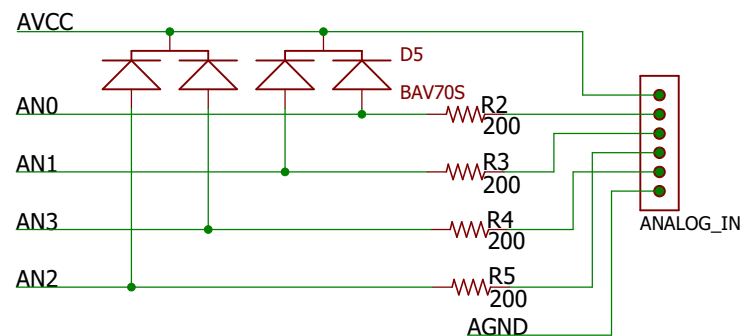
### USB MINI - INTERFACE UART1



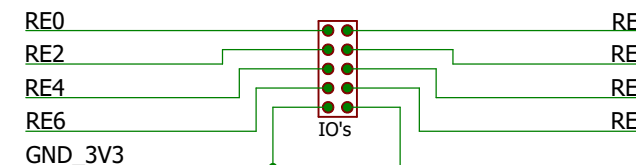
### Módulo Rádio RFM95PW - INTERFACE SPI2



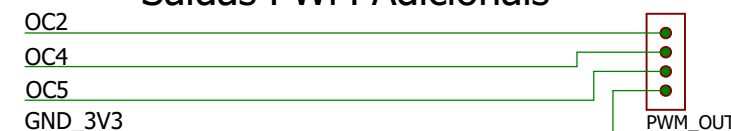
### Entradas Analógicas Adicionais



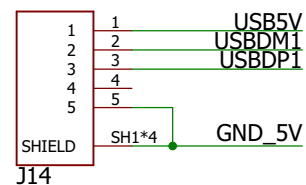
### Entradas e Sidas Digitais Adicionais



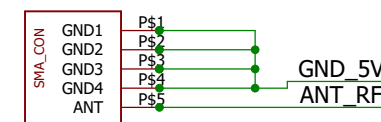
### Saidas PWM Adicionais



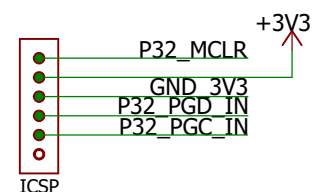
### USB MICRO



### Conector Antena do módulo rádio



### Porta de programação e debug



Data:	29/07/2020 21:54	Projeto:	Luis Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1	Sistema de Telemetria para concurso MotoStudent	Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Nome da PCB: Sistema de receção de dados			Página: 2
Descrição da folha: Interfaces, IO's e conectores			Número de Páginas: 2

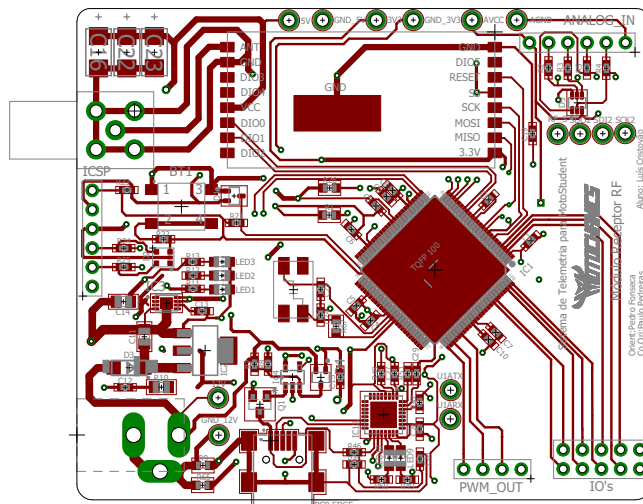
Luís Emanuel Ferreira Cristóvão

*Dissertação de Mestrado*

## **C.2 *Layout* da PCB**

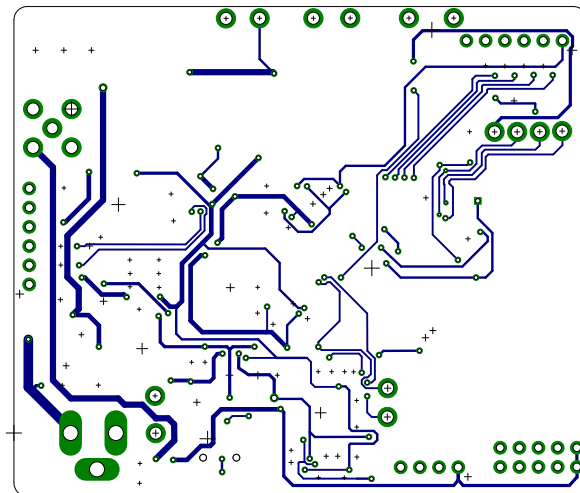






Data:	29/07/2020 22:06	Projeto:	Luís Emanuel Ferreira Cristovão Nº Mec:80886	
Rev:	ver. 1			Sistema de Telemetria para concurso MotoStudent
Escala:	1/1			
		Nome da PCB: Sistema de receção de dados	Página:	
		Descrição da folha: Camada Superior	1	
			Número de Páginas: 2	





Data:	29/07/2020 22:06	Projeto: <b>Sistema de Telemetria para concurso MotoStudent</b>	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1		Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Escala:	1/1		
		Nome da PCB: Sistema de receção de dados	Página:  <b>2</b>
		Descrição da folha: Camada Inferior	Número de Páginas: 2



### C.3 Custo de produção da PCB

Tabela C.1: Custos de produção do sistema para receção de dados.

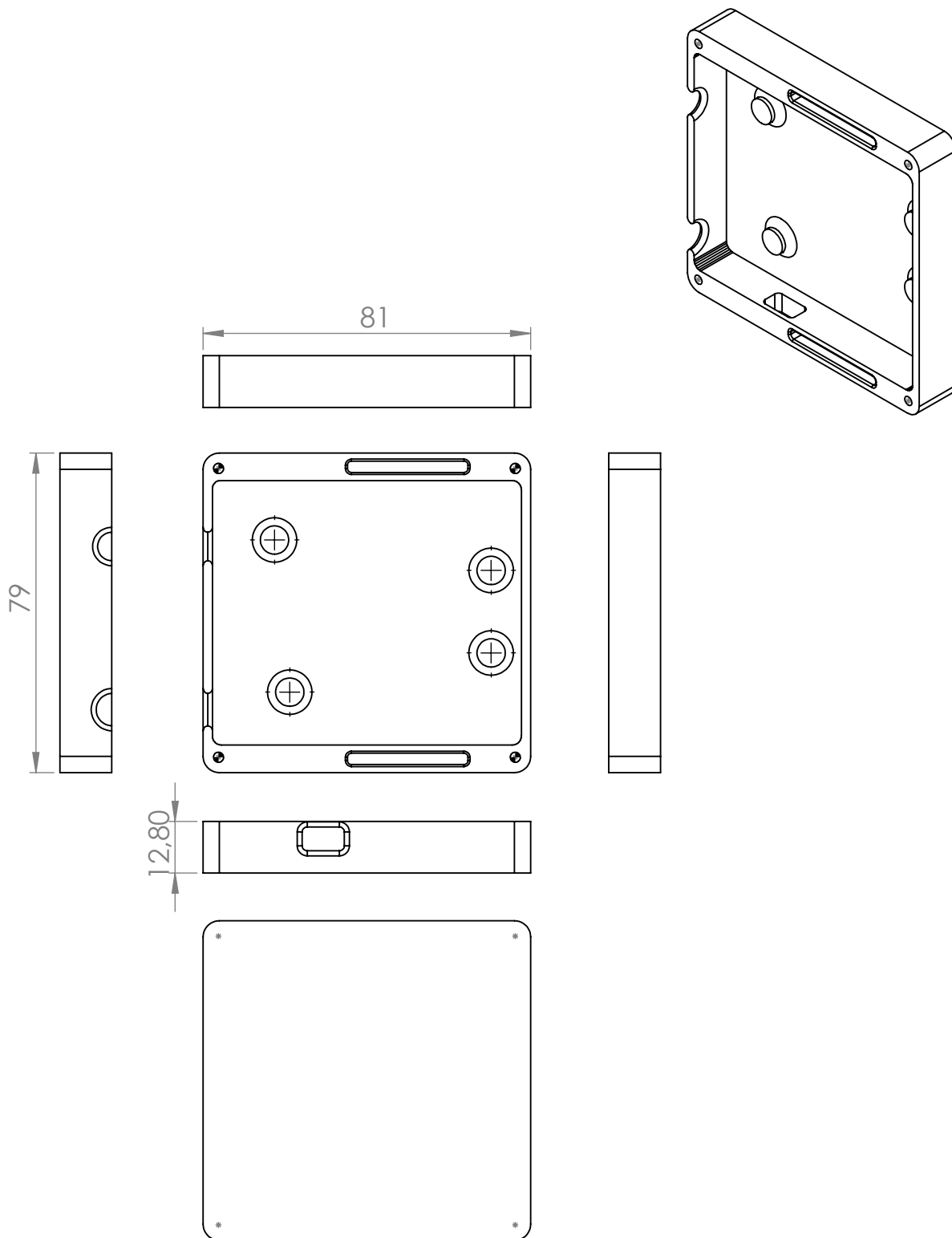
Item	Descrição	Quant.	Preço /Unit.	Preço Total
PIC32MX795F512L 80I/PT	Microcontrolador PIC32	1	8,380 €	8,380 €
GRM155R61A104KA01D	Condensador cerâmico; 100nF; SMD 0402	14	0,007 €	0,104 €
CL05A105KP5NNNC	Condensador cerâmico; 1uF; SMD 0402	1	0,020 €	0,020 €
GRM155R60J106ME44D	Condensador cerâmico; 10uF; SMD 0402	2	0,168 €	0,336 €
CL21A106KAYNNNG	Condensador cerâmico; 10uF; 25V; SMD 0805	1	0,187 €	0,187 €
VJ0402Y102KXACW1BC	Condensador cerâmico; 1nF; SMD 0402	1	0,009 €	0,009 €
CL21A475KOFNNNE	Condensador cerâmico; 4,7uF; SMD 0805	1	0,126 €	0,126 €
T491C107K006AT	Condensador de tantalio; 100uF; SMD 2312	5	0,280 €	1,402 €
SMD0603-0R	Resistência; 0Ω; SMD 0603; 0,1W; ±5%	16	0,009 €	0,147 €
SMD0402-200R-1%	Resistência; 200Ω; SMD 0402; 63mW; ±1%	5	0,009 €	0,045 €
SMD0402-10K-1%	Resistência; 10kΩ; SMD 0402; 63mW; ±1%	4	0,009 €	0,036 €
CS0805-0R1-1%-LTC	Resistência; 100mΩ; SMD 0805; 0,125W; ±1%	1	0,097 €	0,097 €
SMD0402-4K7-1%	Resistência; 4,7kΩ; SMD 0402; 63mW; ±1%	2	0,009 €	0,018 €
SMD0402-1K6-1%	Resistência; 1,6kΩ; SMD 0402; 63mW; ±1%	1	0,009 €	0,009 €
SMD0402-750R-1%	Resistência; 750Ω; SMD 0402; 63mW; ±1%	3	0,009 €	0,027 €
SMD0402-1K-1%	Resistência; 1kΩ; SMD 0402; 63mW; ±1%	2	0,009 €	0,018 €
SMD0402-51R-1%	Resistência; 51Ω; SMD 0402; 63mW; ±1%	3	0,009 €	0,027 €
TACTM-69N-F	Botão 2 posições; 0,05A/12VDC; SMT	1	0,051 €	0,051 €
BAV99W	Díodo comutador; 100V; 150mA; SMD SOT232	1	0,022 €	0,022 €
PESD5V0L4UG	Díodo; 6,8V; 2,5A; SMD SOT353	1	0,121 €	0,121 €
SK24-DIO	Díodo retificador Schottky; 40V; 2A; SMD	1	0,068 €	0,068 €
BAV70S.115	Díodo comutador; 100V; 250mA; SMD	1	0,060 €	0,060 €
KP-2012LSURCK	LED vermelho; 1,75-2,5V; SMD 0805	5	0,096 €	0,482 €
FDN342P	Transístor P-MOSFET; 20V; 2A; 0,5W; SOT-3	1	0,145 €	0,145 €
ADC-002-2	Tomada de alimentação DC; macho; 1A; THT	1	0,258 €	0,258 €
5-1814400-1	Tomada SMA; fêmea; angulares 90°; 50Ω; THT	1	2,310 €	2,310 €
MX-67803-8020	Tomada; USB AB mini; horizontal; SMT	1	0,660 €	0,288 €
MCP1725-3302E/MC	Regulador de tensão; 3,3V	1	0,450 €	0,450 €
MCP6001T-I/OT	Amplificador operacional; 1MHz; 1,8÷5,5V; 1 Canal	1	0,203 €	0,203 €
RFM95PW-868S2	Transceiver RF; FSK,GFSK,GMSK,LoRa,OOK; 868MHz	1	17,330 €	17,330 €
FT232RQ	Interface USB-UART; Full Speed; 3,3÷5VDC; QFN32	1	4,000 €	4,000 €
8.00M-CFPX104	Ressonador: de quartzo; 8MHz; ±20ppm; 18pF; SMD	1	1,021 €	1,021 €
C0402C300J5GAC	Condensador cerâmico; 30pF; SMD 0402	2	0,027 €	0,053 €
KEYS5005	Pino para teste	14	0,240 €	3,356 €
PCB-76*65	PCB com transporte	1	2,420 €	2,420 €
Total				41,86 €



## C.4 Desenhos técnicos caixa de proteção

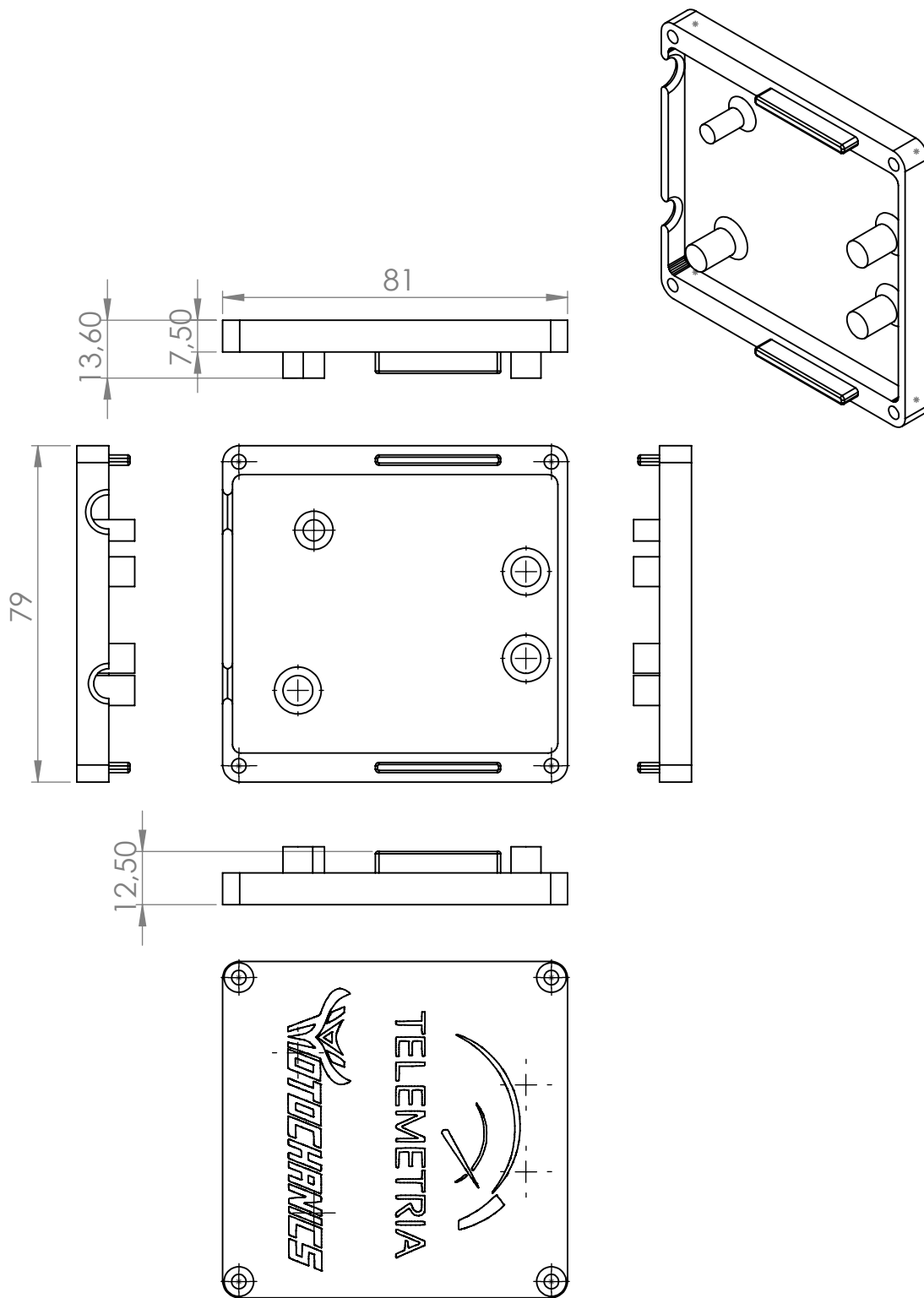






Data:	15/8/2019 14:48	Projeto: <b>Sistema de Telemetria para concurso MotoStudent</b>	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1		Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Escala:	2/3		
Material:	PLA		
	Nome: Caixa de proteção para sistema recepção de dados		Página:  1
	Descrição da folha: Base da caixa		Número de Páginas: 2





Data:	15/8/2019 14:48	Projeto: <b>Sistema de Telemetria para concurso MotoStudent</b>	Luís Emanuel Ferreira Cristovão Nº Mec:80886
Rev:	ver. 1		Orientador: Prof. Pedro Fonseca Co-orientador: Prof. Paulo Pedreiras
Escala:	2/3		
Material:	PLA		
	Nome: Caixa de proteção para sistema recepção de dados		Página:
	Descrição da folha: Tapa da caixa		2
			Número de Páginas: 2



## Anexo D

# Desenvolvimento de *firmware*

### D.1 Sistema de aquisição, gravação e transmissão de dados

#### D.1.1 *Array* dos pedido a realizar à ECU

Link do ficheiro (<https://bit.ly/2XUjKZU>).



## **Anexo E**

# **Resultados dos testes realizados**

### **E.1 Leitura de temperaturas através do pirómetro**

#### **E.1.1 Dados do teste à comunicação com o pirómetro**

Link do ficheiro (<https://bit.ly/3dWM18r>).

#### **E.1.2 Dados do teste à taxa de aquisição ao pirómetro**

Link do ficheiro (<https://bit.ly/3cDXrOe>).

### **E.2 Leitura da ECU**

#### **E.2.1 Dados do teste à taxa de aquisição da ECU**

Link do ficheiro (<https://bit.ly/2ykRZ44>).

### **E.3 Recepção de dados dos módulos de sensores adicionais**

#### **E.3.1 Captura efetuada no barramento CAN2**

Link do ficheiro (<https://bit.ly/2LEyLJt>).

#### **E.3.2 Teste ao número máximo de pacotes transmitidas pelo barramento CAN2**

Link do ficheiro (<https://bit.ly/36oy94B>).

### **E.4 Leitura do módulo inercial**

#### **E.4.1 Dados do teste à taxa de aquisição do módulo inercial**

Link do ficheiro (<https://bit.ly/3bNqmhu>).

## **E.5 Leitura do módulo GPS**

### **E.5.1 Captura das mensagens transmitidas pelo módulo GPS**

Link do ficheiro (<https://bit.ly/3kfCblJ>).

## **E.6 Escrita no cartão micro SD**

### **E.6.1 Dados guardados no cartão durante 20 minutos**

Link do ficheiro (<https://bit.ly/37kQdPu>).

## **E.7 Transmissão de dados por banda ISM**

### **E.7.1 Dados recebidos a uma distância de 1 km**

Link do ficheiro (<https://bit.ly/35dVHc8>).

### **E.7.2 Dados recebidos a uma distância de 2.6 km**

Link do ficheiro (<https://bit.ly/37q47j5>).

### **E.7.3 Mensagens recebidas pela interface gráfica gráfica**

Link do ficheiro (<https://bit.ly/3jgLcJV>).

## **E.8 Transmissão de dados para ecrã**

### **E.8.1 Captura das mensagens transmitidas para o ecrã**

Link do ficheiro (<https://bit.ly/3dG6UWI>).

## **E.9 Teste em pista**

### **E.9.1 Dados recolhidos durante o teste em pista**

Link do ficheiro (<https://bit.ly/3nuSufO>).

### **E.9.2 Filme do teste em pista**

Link do ficheiro (<https://bit.ly/32OGa24>).



