

José Carlos Meireles Monteiro Metrôlho

REDE CAN PARA COMANDO DE ACTUADORES  
EM ESTUFAS AGRÍCOLAS



UNIVERSIDADE DO MINHO

Escola de Engenharia

Departamento de Electrónica Industrial

1999

José Carlos Meireles Monteiro Metrôlho

REDE CAN PARA COMANDO DE ACTUADORES  
EM ESTUFAS AGRÍCOLAS

Dissertação submetida à Universidade  
do Minho, para a obtenção do grau de  
Mestre em Electrónica Industrial.

UNIVERSIDADE DO MINHO

1999

Dissertação realizada sob orientação científica do  
Professor Catedrático Carlos Alberto Monteiro  
Caridade e Couto, do Departamento de Electrónica  
Industrial da Universidade do Minho.

Esta dissertação e respectivo Mestrado em Electrónica Industrial decorreu, parcialmente, no âmbito do programa PRAXIS XXI promovido pela Junta Nacional de Investigação Científica e Tecnológica (JNICT) através da atribuição da bolsa de Mestrado com a referência BM/6751/95.

# Resumo

A crescente importância da utilização de estufas, devido à necessidade de obter com êxito, culturas em ambientes e/ou épocas, que à partida não seriam as mais favoráveis, exige que se desenvolvam sistemas para controlar algumas grandezas consideradas importantes tais como, a temperatura, humidade, radiação solar, etc.

No âmbito desta dissertação, pretende-se efectuar o controlo eficiente do ambiente de uma estufa a partir de dados previamente recolhidos e processados. Este trabalho completa outros já realizados, na Universidade de Trás-os-Montes e Alto Douro, nomeadamente a aquisição de dados provenientes de diversos tipos de sensores [1], bem como a implementação de algoritmos de controlo [2], tendo como objectivo ser uma solução para a etapa final, ou seja, a actuação.

O sistema desenvolvido permite a execução de acções de controlo após recepção de comandos, provenientes de uma estação base situada na estufa, de acordo com o resultado do processamento de algoritmos de controlo adequados à cultura que se pretenda controlar. Assim, permite o accionamento de diversos dispositivos actuadores, os quais podem ser de diferentes tipos e colocados em diferentes locais da estufa, condicionando o ambiente da mesma.

No sistema é utilizado o protocolo de comunicações *Controller Area Network* (CAN), uma vez que este suporta de forma eficiente e segura o controlo em tempo real e também porque o mesmo serve de base à rede de aquisição do sistema, no qual esta dissertação se enquadra. O protocolo série CAN permite a implementação de uma estrutura em rede com utilização de um número considerável de nós e elevadas taxas de transmissão, o que possibilita a realização do controlo rápido em diversos pontos distribuídos pela estufa, embora neste tipo de aplicação, devido à inércia do ambiente de uma estufa, a robustez, fiabilidade, custo e facilidade de utilização sejam os factores predominantes.

É importante referir, também, que devido ao posicionamento dos actuadores numa estufa e ao seu tipo de alimentação, é justificável a utilização de uma arquitectura “cableada” na rede de actuação sem necessidade de recorrer a outros meios de comunicação.

Assim, pretende-se desenvolver uma componente que sendo dependente de outras num sistema de controlo de estufas agrícolas, permita concluir esse controlo através da comunicação com os circuitos de accionamento dos actuadores de uma forma fiável, robusta, simples e económica.

**Palavras chave:** Estufas agrícolas, *Controller Area Network* (CAN), *Serial Linked I/O Device* (SLIO), actuadores, liga/desliga, motor de indução, IGBTs, accionamento.

# Abstract

The increasing importance of greenhouse production is due to the necessity to obtain cultures in environments and/or seasons, that from the first point of view would not be the most favourable. This problem demands the development of systems to control some important variables such as: temperature, humidity (of the air and soil), solar radiation, etc.

In the scope of this dissertation, it is intended to achieve efficient control of the greenhouse environment based on data previously collected and processed. This work completes the research developed, in the University of Trás-os-Montes e Alto Douro, namely the data acquisition from several type of sensors [1], as well as the implementation of control algorithms [2], with the objective of obtaining a solution for the final stage, the actuation.

The developed system allows the execution of control actions after the reception of commands coming from a base station located in the greenhouse. This is done according with the results obtained by processing adequate control algorithms in accordance to the crop that it is intended to pick. Thus, it allows the drive of several actuator devices, which can be of different types and located on several places in the greenhouse, conditioning its environment.

The Controller Area Network (CAN) communication's protocol it is used, in this system, because it supports real time control with efficient and insurance. It also serves as base to the system acquisition network, in which this dissertation fits. The CAN serial protocol, allows the implementation of a network structure using a considerable number of nodes and high baud rate, making possible the accomplishment of high speed control in several points distributed through the greenhouse. However, in this type of application, due to the greenhouse environment's inertia, the robustness, reliability, cost and easiness of use are the predominant factors.

It is also important to note that due to the actuators position in the greenhouse and to its type of feed, the use of a wired architecture in the actuation network is justifiable without necessity to appeal to a wireless solution.

Thus, it is intended to develop one component that being dependent of others in a agricultural greenhouses control system, allows to conclude this control through the communication with the actuator's drive circuits on a reliable, robust, simple and economic way.

**Keywords:** Greenhouses, *Controller Area Network (CAN)*, *Serial Linked I/O Device (SLIO)*, actuators, on/off, induction motor, IGBTs, drive.

# Agradecimentos

O autor deseja manifestar o seu mais sincero agradecimento a todas as instituições e pessoas que, com a sua valiosa colaboração, contribuíram para que a realização deste trabalho fosse possível.

Ao professor Catedrático Carlos Alberto Monteiro Caridade e Couto, agradeço a orientação científica, o incentivo, as sugestões e discussões sempre oportunas, a confiança e apoio constantes ao longo de todo o trabalho. Fico-lhe também muito grato pelas sugestões feitas durante a escrita da dissertação e revisão final.

Ao Professor Associado Doutor Júlio Manuel de Sousa Barreiros Martins agradeço o apoio e colaboração prestada sempre que foi necessário.

Aos Eng.<sup>os</sup> João Luíz Afonso, Graça Maria Henriques Minas e Pedro Mestre agradeço a colaboração prestada durante as várias fases deste trabalho e as sugestões que sempre deram quando solicitados.

Ao Dr. Paulo Moura Oliveira e aos Eng.<sup>os</sup> Joaquim José de Castro Ferreira e Victor Manuel de Jesus Filipe agradeço a colaboração prestada durante a fase de verificação ortográfica da presente dissertação.

Aos técnicos das oficinas da secção de engenharias da Universidade de Trás-os-Montes e Alto Douro, na pessoa do Eng.<sup>o</sup> Manuel Marques, agradeço o apoio dado na implementação do *hardware*.

Aos meus pais, agradeço o carinho e o apoio constantes, que muito contribuíram para a realização deste trabalho.

A todos aqueles que, de forma directa ou indirecta, contribuíram para a realização deste trabalho, o meu sincero obrigado.

# ÍNDICE

<b>1. Introdução</b>	<b>1</b>
1.1 Controlo computacional de estufas agrícolas	1
1.1.1 Componentes de um sistema computacional	2
1.1.2 Topologias de sistemas computacionais	4
1.2 Sistema de controlo utilizando microcontroladores	4
1.3 O trabalho proposto	5
1.4 Organização da dissertação	7
<b>2. Acções de actuação em estufas agrícolas</b>	<b>9</b>
2.1 Introdução	9
2.2 Estufas	9
2.2.1 Vantagens	9
2.2.2 Desvantagens	9
2.2.3 Tipos de estufas	10
2.3 Controlo da atmosfera numa estufa	12
2.3.1 Controlo de temperaturas elevadas	12
2.3.1.1 Ventilação	12
2.3.1.2 Refrigeração por humedificação – “ <i>Cooling Systems</i> ”	16
2.3.1.3 Ecrãs de sombreamento	18
2.3.1.4 Outras técnicas	19
2.3.2 Controlo de temperaturas baixas	19
2.3.2.1 Aquecedores unitários	20
2.3.2.2 Sistemas de água quente	20
2.3.2.3 Sistemas de aquecimento a vapor	21
2.3.2.4 Factores relevantes para selecção do sistema de aquecimento	21
2.3.3 Irrigação	22
2.3.3.1 Métodos de distribuição de água -rega	23
2.3.4 Humidade	26
2.3.4.1 Regulação da humidade do ar	27
2.3.5 Luminosidade	31
2.3.5.1 Iluminação	31
2.3.6 Controlo do teor de dióxido de carbono (CO <sub>2</sub> )	34
2.4 Resumo	34
<b>3. Protocolo de comunicações CAN</b>	<b>37</b>
3.1 Sistemas em rede vs sistemas centralizados	37
3.2 Introdução	37
3.2.1 Um pouco de história	37
3.2.2 Características do CAN	38
3.3 Método de endereçamento	40
3.4 Processo de arbitragem não destrutiva	41
3.5 Formato das <i>frames</i>	42
3.6 Transferência de mensagens	43
3.6.1 <i>Frame</i> de Dados	44
3.6.2 <i>Frame</i> Remota	44
3.6.3 <i>Frame</i> de Erro	45

3.6.4	<i>Frame</i> de Sobrecarga	46
3.6.5	Intervalo <i>Interframe</i>	47
3.7	Codificação	48
3.8	Ordem de transmissão de <i>bits</i> numa <i>frame CAN</i>	48
3.9	Prioridade	48
3.10	Taxas de transmissão	49
3.11	Tamanho de uma rede CAN	49
3.12	Arquitectura dos controladores CAN	49
3.13	Métodos de detecção e sinalização de erro	50
3.13.1	Detecção de erros	50
3.13.1.1	Erros ao nível da mensagem	50
3.13.1.1	Erros ao nível do <i>bit</i>	51
3.13.1	Sinalização de erros	51
3.14	O nível físico	51
3.14.1	Meio físico	52
3.14.2	Transmissão diferencial	52
3.14.3	Tolerância a falhas no barramento	53
3.15	Tempo de <i>bit</i> e sincronização	53
3.15.1	Representação e definição do tempo de <i>bit</i>	53
3.15.2	Sincronização	54
3.16	Configurações base de circuitos integrados	54
3.16.1	Configuração utilizando controladores CAN discretos	54
3.16.2	Configuração utilizando microcontrolador com controlador CAN	54
3.16.3	Configuração SLIO CAN	55
3.17	Porque escolher o CAN?	60
<b>4.</b>	<b>Descrição do sistema global</b>	<b>61</b>
4.1	Introdução	61
4.2	O sistema global	61
4.3	Rede de actuação	64
4.4	A arquitectura da rede de actuação proposta	64
<b>5.</b>	<b>Hardware do sistema</b>	<b>67</b>
5.1	Hardware de comunicações	67
5.1.1	Nó <i>Host</i>	67
5.1.1.1	O microcontrolador	68
5.1.1.2	<i>Real Time Clock</i> (RTC)	71
5.1.1.3	Alimentação das placas	72
5.1.1.4	Interface RS-232	73
5.1.1.5	Interface com o barramento CAN	74
5.1.1.6	Memória	75
5.1.1.7	Circuito para <i>backup</i> das memórias	76
5.1.2	Nó <i>Slave Motor</i>	77
5.1.3	Nós SLIO CAN	77
5.1.4	Circuito para aquisição de valores de temperatura	80
5.2	Hardware de actuação	81
5.2.1	Circuito de accionamento usando relé	81
5.2.2	Circuito de accionamento para actuadores DC	82
5.2.3	Circuito para accionamento de um motor de indução	83
5.2.3.1	Introdução	83

5.2.3.2 Implementação	89
<b>6. Software da rede de actuação</b>	<b>93</b>
6.1 Estrutura de comunicações da rede	94
6.1.1 As mensagens CAN	95
6.1.2 Mensagens entre o PC-supervisor e as estações com microcontrolador	98
6.1.2.1 Mensagem de configuração	98
6.1.2.2 Mensagem de alarme	102
6.1.2.3 Mensagem de <i>set-point</i> (s)	102
6.1.2.4 Mensagem de aquisição	103
6.1.2.5 Mensagem para relógio de tempo real (RTC)	105
6.1.2.6 Mensagem do PC-supervisor para actuação	106
6.1.3 Mensagens entre o nó <i>Host</i> e os nós SLIO CAN	107
6.1.3.1 Mensagem do nó <i>Host</i> para os SLIOs	107
6.1.3.2 Mensagem dos SLIOs para o <i>Host</i>	109
6.1.4 Calibração dos SLIOs	110
6.1.4.1 Mensagem de calibração	112
6.2 <i>Software</i> residente no <i>Host</i>	113
6.2.1 Iniciação	113
6.2.2 Atendimento a interrupções	114
6.2.2.1 Rotina de serviço à interrupção não mascarável (NMI)	114
6.2.2.2 Rotina de serviço à interrupção CAN	115
6.2.2.3 Rotina de serviço à interrupção do relógio de tempo real (RTC)	116
6.2.3 Selecção do modo de controlo (Local/Remoto)	118
6.2.4 Programa principal	118
6.3 <i>Software</i> residente no nó <i>Slave Motor</i>	119
6.3.1 Programa principal	119
6.3.2 Modulador PWM	120
6.3.2.1 Modulação de largura de impulso sinusoidal	120
6.3.2.2 Frequência do sinal PWM	121
6.3.2.3 Valor dos impulsos	121
6.3.2.4 A memória	122
6.3.2.5 Programa para implementar o modulador PWM	123
<b>7. Testes e resultados</b>	<b>125</b>
7.1 Rede CAN	125
7.2 Circuitos de accionamento	127
7.2.1 Solução utilizando relés	127
7.2.2 Accionamento do motor de indução monofásico	127
7.2.2.1 Ambiente de teste	127
7.2.2.2 Teste do <i>hardware</i> implementado	129
7.2.2.3 Teste do modulador PWM	131
<b>8. Conclusões e sugestões para trabalho futuro</b>	<b>135</b>
8.1 Conclusões	135
8.2 Sugestões para trabalho futuro	136

## **Bibliografia**

**137**

### **Anexos**

Anexo 1 - Placas de Circuito Impresso	A1
Anexo 2 - Utilização das saídas analógicas	A9
Anexo 3 - Mapeamento de memória	A11
Anexo 4 – Tipos de <i>frames</i> e registos do SLIO P82C150	A13
Anexo 5 - Aquisição analógico-digital utilizando os SLIOs	A17
Anexo 6 - Iniciação do microcontrolador e do controlador CAN	A21
Anexo 7 - Especificações do módulo PM20CSJ060	A33

### **Glossário**

# ÍNDICE DE FIGURAS

<b>Figura 1.1</b>	Sistema de controlo em malha fechada.	2
<b>Figura 1.2</b>	Diagrama de blocos do SCMF computacional.	6
<b>Figura 2.1</b>	Classificação de estufas quanto à estrutura e ao perfil externo.	10
<b>Figura 2.2</b>	Fotografia de estufa com sistema de comando eléctrico de aberturas zenitais.	13
<b>Figura 2.3</b>	Estufa com técnica de abertura lateral enrolável.	13
<b>Figura 2.4</b>	Trajecto do ar utilizando ventilação estática por aberturas zenitais.	14
<b>Figura 2.5</b>	Ventilação forçada.	15
<b>Figura 2.6</b>	Sistema de refrigeração e humificação.	16
<b>Figura 2.7</b>	Fotografia de estufa contendo ecrãs de sombreamento.	18
<b>Figura 2.8</b>	Sistema de aquecimento utilizando água quente.	20
<b>Figura 2.9</b>	Sistema utilizando água quente.	21
<b>Figura 2.10</b>	Fotografia do sistema de rega gota-a-gota.	25
<b>Figura 2.11</b>	Sistema completo de rega gota-a-gota.	26
<b>Figura 2.12</b>	Fotografia de nebulização por injector.	28
<b>Figura 2.13</b>	Sistema de nebulização – “ <i>Fog System</i> ”.	29
<b>Figura 2.14</b>	Fotografia de um desumidificador.	30
<b>Figura 2.15</b>	Sistema de desumidificação.	30
<b>Figura 2.16</b>	Fotografia de estufa usando lâmpadas fluorescentes.	32
<b>Figura 2.17</b>	Fotografia de lâmpadas vapor de sódio a alta pressão em estufas.	33
<b>Figura 2.18</b>	Instalação para distribuição uniforme de luminosidade.	33
<b>Figura 3.1</b>	Método de endereçamento.	40
<b>Figura 3.2</b>	Formato dos diferentes tipos de <i>frames</i> .	42
<b>Figura 3.3</b>	<i>Frame</i> de dados	44
<b>Figura 3.4</b>	<i>Frame</i> remota.	45
<b>Figura 3.5</b>	<i>Frame</i> de erro.	45
<b>Figura 3.6</b>	<i>Frame</i> de sobrecarga.	46
<b>Figura 3.7</b>	Intervalo <i>interframe</i> para estações que não sejam “passivas-ao-erro” ou que tenham sido receptoras da mensagem anterior.	47
<b>Figura 3.8</b>	Intervalo <i>interframe</i> para estações “passivas-ao-erro” que tenham transmitido a mensagem anterior.	47
<b>Figura 3.9</b>	Método de inserção de <i>bit</i> .	48
<b>Figura 3.10</b>	Ordem de transmissão dos <i>bits</i> .	48
<b>Figura 3.11</b>	Representação física de <i>bit</i> .	52
<b>Figura 3.12</b>	Divisão do tempo de <i>bit</i> .	53
<b>Figura 3.13</b>	Sistema CAN típico.	56
<b>Figura 3.14</b>	Sistema SLIO CAN.	56
<b>Figura 3.15</b>	Relação entre o identificador do SLIO (P82C150) e o identificador CAN <i>standard</i> .	56
<b>Figura 3.16</b>	SLIO e outros nós CAN partilhando o mesmo barramento.	57
<b>Figura 3.17</b>	Mensagem de calibração.	58
<b>Figura 3.18</b>	Sistemas actuais.	59
<b>Figura 3.19</b>	Sistemas futuros.	59
<b>Figura 4.1</b>	Configuração geral do sistema.	63
<b>Figura 4.2</b>	Controlo de um conjunto de estufas.	63
<b>Figura 4.3</b>	Arquitectura da rede de actuação proposta.	66
<b>Figura 4.4</b>	Diagrama de blocos dos vários constituintes do sistema.	66
<b>Figura 5.1</b>	Arquitectura da rede de actuação pretendida.	67
<b>Figura 5.2</b>	Diagrama de blocos do microcontrolador P8xC592.	69
<b>Figura 5.3</b>	Diagrama de blocos da placa <i>Host</i> .	71
<b>Figura 5.4</b>	Esquemático do circuito do nó <i>Host</i> .	71-A
<b>Figura 5.5</b>	Esquemático do circuito da fonte para 5 V DC.	72

<b>Figura 5.6</b>	Esquemático do circuito para monitorização da tensão de alimentação.	73
<b>Figura 5.7</b>	Esquemático do circuito que permite comunicação RS-232 com o <i>Host</i> .	73
<b>Figura 5.8</b>	Esquemático da componente do circuito responsável pela <i>interface</i> com o barramento CAN.	75
<b>Figura 5.9</b>	Memória externa do nó <i>Host</i> .	76-A
<b>Figura 5.10</b>	Diagrama de blocos do sistema SLIO CAN.	78
<b>Figura 5.11</b>	Campo de arbitragem das <i>frames</i> do SLIO CAN P82C150.	79
<b>Figura 5.12</b>	Esquemático do circuito da placa SLIO CAN.	80
<b>Figura 5.13</b>	Esquemático do circuito para aquisição de valores de temperatura.	81
<b>Figura 5.14</b>	Esquemático do circuito para accionamento liga/desliga utilizando relé.	81
<b>Figura 5.15</b>	Esquemático de uma aplicação utilizando o transistor BUK101-50GS.	82
<b>Figura 5.16</b>	Classificação de motores eléctricos.	83
<b>Figura 5.17</b>	Topologia dum sistema de accionamento AC.	84
<b>Figura 5.18</b>	Modulação PWM de uma onda sinusoidal usando uma portadora triangular.	85
<b>Figura 5.19</b>	Inversor monofásico em ponte completa.	85
<b>Figura 5.20</b>	Efeito do “tempo morto” ( $t_{\Delta}$ ).	86
<b>Figura 5.21</b>	Símbolo e característica (corrente - tensão) do IGBT: (a) símbolo de um IGBT canal-n; (b) característica de transferência.	87
<b>Figura 5.22</b>	Diagrama de blocos do sistema implementado.	89
<b>Figura 5.23</b>	Esquemático do circuito para accionamento do motor de indução monofásico.	90-A
<b>Figura 6.1</b>	Componentes de <i>software</i> do sistema.	93
<b>Figura 6.2</b>	Elementos que compõem a arquitectura da rede de actuação.	94
<b>Figura 6.3</b>	Relação (em termos de comunicação) entre os vários nós da rede de actuação.	95
<b>Figura 6.4</b>	<i>Frame</i> de dados CAN.	95
<b>Figura 6.5</b>	Formato dos dois <i>bytes</i> que constituem os campos de arbitragem e controlo de uma <i>frame</i> de dados.	96
<b>Figura 6.6</b>	Sentido(s) do fluxo de mensagens em função do tipo de mensagens na rede de actuação.	98
<b>Figura 6.7</b>	Formato de uma mensagem de configuração.	100
<b>Figura 6.8</b>	Formato de uma mensagem para pedido de configuração.	101
<b>Figura 6.9</b>	Formato dos campos de arbitragem e controlo numa mensagem de alarme.	102
<b>Figura 6.10</b>	Formato da <i>frame</i> de dados para envio de <i>set-points</i> por parte do nó PC-supervisor.	103
<b>Figura 6.11</b>	Formato da <i>frame</i> remota para pedido de <i>set-points</i> .	103
<b>Figura 6.12</b>	Formato da <i>frame</i> para realizar aquisição.	104
<b>Figura 6.13</b>	Formato da mensagem enviada pelo nó <i>Host</i> após aquisição (A/D) realizada pelos nós SLIO CAN.	104
<b>Figura 6.14</b>	Formato da mensagem enviada por um nó com microcontrolador (excepto o <i>Host</i> ) após aquisição A/D.	105
<b>Figura 6.15</b>	Formato da <i>frame</i> para configuração do RTC.	105
<b>Figura 6.16</b>	Formato da <i>frame</i> contendo os valores do RTC.	106
<b>Figura 6.17</b>	Formato das <i>frames</i> de dados para comunicação entre o PC-supervisor e os nós (com microcontrolador) da rede de actuação.	106
<b>Figura 6.18</b>	Mensagem para arrefecimento.	107
<b>Figura 6.19</b>	Formato de uma <i>frame</i> de dados do SLIO P82C150.	108
<b>Figura 6.20</b>	Mensagem para habilitar o pino E/S.	109
<b>Figura 6.21</b>	Mensagem para activar o pino de E/S.	109
<b>Figura 6.22</b>	Formato de uma mensagem do SLIO com o valor de uma conversão A/D.	110
<b>Figura 6.23</b>	Processo completo de calibração dos nós SLIO (P82C150) do sistema.	111
<b>Figura 6.24</b>	Procedimento geral para sistemas com barramento CAN com nós possuindo o SLIO P82C150.	112
<b>Figura 6.25</b>	Mensagem de calibração.	113
<b>Figura 6.26</b>	Fluxograma para a rotina de serviço à interrupção NMI.	115
<b>Figura 6.27</b>	Fluxograma da rotina de serviço a uma interrupção CAN.	116
<b>Figura 6.28</b>	Fluxograma para a rotina de serviço à interrupção RTC.	117
<b>Figura 6.29</b>	Formato da mensagem para execução em modo remoto.	118
<b>Figura 6.30</b>	Fluxograma do programa residente no <i>Host</i> .	119-A
<b>Figura 6.31</b>	Fluxograma do programa residente na placa <i>Slave Motor</i> .	119-B
<b>Figura 6.32</b>	Formato da <i>frame</i> para ajuste da frequência a sintetizar.	119

<b>Figura 6.33</b>	Comutação do sinal PWM.	121
<b>Figura 6.34</b>	Fluxograma do código para sintetizar uma onda sinusoidal com frequência $f$ .	
<b>Figura 7.1</b>	Fotografia da estufa utilizada durante os testes da rede CAN.	125
<b>Figura 7.2</b>	Amostras da actuação feita utilizando a rede SLIO CAN implementada no âmbito desta dissertação (temperatura e estado dos actuadores).	126
<b>Figura 7.3</b>	Infra-estrutura utilizada na fase de ensaios.	127
<b>Figura 7.4</b>	Aspecto gráfico do DADiSP 4.0.	128
<b>Figura 7.5</b>	Gerador de onda quadrada.	129
<b>Figura 7.6</b>	Forma de onda lida antes da colocação dos circuitos de “amortecimento”.	130
<b>Figura 7.7</b>	Adição de circuitos de “amortecimento”.	130
<b>Figura 7.8</b>	Forma de onda lida após colocação dos circuitos de “amortecimento”.	130
<b>Figura 7.9</b>	Modulação PWM ( $f=50$ Hz): a) forma de onda da tensão; b) espectro da tensão.	132
<b>Figura 7.10</b>	Modulação PWM ( $f=25$ Hz): a) forma de onda da tensão; b) espectro da tensão.	133

# ÍNDICE DE TABELAS

<b>Tabela 2.1</b>	Resumo das actuações e respectivo controlo normalmente utilizados em estufas agrícolas.	35
<b>Tabela 3.1</b>	Distância máxima aconselhada, entre dois nós extremos, para uma rede CAN e SLIO CAN.	59
<b>Tabela 5.1</b>	Seleccção de memória.	76
<b>Tabela 6.1</b>	Códigos de aceitação dos nós (com microcontrolador) que constituem a rede de actuação.	97
<b>Tabela 6.2</b>	Significado dos três <i>bits</i> menos significativos do identificador.	97
<b>Tabela 6.3</b>	Configuração existente por <i>default</i> no <i>buffer</i> de configuração do nó <i>Host</i> .	99
<b>Tabela 6.4</b>	Atribuição de nºtarefa/actuação por <i>default</i> .	99
<b>Tabela 6.5</b>	Tipos de actuação.	107

## ABREVIATURAS

<b>A/D</b>	Conversor analógico-digital
<b>ABS</b>	<i>Anti-lock Braking System</i>
<b>AC</b>	Componente Alternada
<b>ACK</b>	Reconhecimento; <i>Acknowledge</i>
<b>ACR</b>	<i>Acceptance Code Register</i>
<b>ACR</b>	<i>Analogue Configuration Register</i>
<b>ALE</b>	<i>Address Latch Enable</i>
<b>AMR</b>	<i>Acceptance Mask Register</i>
<b>BAC</b>	Bomba a Calor
<b>CAN</b>	<i>Controller Area Network</i>
<b>Cia</b>	<i>CAN in Automation</i>
<b>CMR</b>	<i>Common Mode Rejection</i>
<b>CPU</b>	<i>Central Processing Unit</i>
<b>CRC</b>	<i>Cyclic Redundancy Code</i>
<b>CSMA/DCR</b>	<i>Carrier Sense Multi-Access with Deterministic Collision Resolution</i>
<b>CTR</b>	<i>Current Transfer Ratio</i>
<b>DC</b>	Componente Contínua
<b>DLC</b>	<i>Data Length Code</i>
<b>DPTR</b>	Ponteiro de Dados; <i>Data Pointer</i>
<b>E/S</b>	Entrada/Saída
<b>EMI</b>	<i>Electromagnetic Interference</i>
<b>EOF</b>	Fim de frame; <i>End Of Frame</i>
<b>ETP</b>	Evapotranspiração Potencial
<b>GTO</b>	<i>Gate Turn-Off Thyristors</i>
<b>HID</b>	<i>Hight Intensity Discharge</i>
<b>HP</b>	<i>Horse Power</i>
<b>IDE</b>	<i>Identifier Extension</i>
<b>IGBT</b>	<i>Insulated Gate Bipolar Transistors</i>
<b>ISO</b>	<i>International Standard Organisation</i>
<b>MI</b>	Motor de Indução
<b>MOSFET</b>	<i>Metal-Oxide Semiconductor Field Effect Transistor</i>
<b>NRZ</b>	<i>Non-Return-to-Zero</i>
<b>OSI</b>	<i>Open Systems Interconnection</i>
<b>PC</b>	<i>Personal Computer</i>
<b>PCON</b>	<i>Power Control Register</i>
<b>PSEN</b>	<i>Program Strobe Enable</i>
<b>PWM</b>	<i>Pulse Width Modulation</i>
<b>PWMP</b>	<i>Prescaler Frequency Control Register</i>
<b>RAM</b>	<i>Random Access Memory</i>
<b>RD</b>	Linha de Leitura
<b>RF</b>	Rádio Frequência
<b>RFI</b>	<i>Radio Frequency Interference</i>
<b>RTC</b>	<i>Real Time Clock</i>
<b>RTR</b>	<i>Remote Transmission Request</i>
<b>SCMA</b>	Sistema de Controlo em Malha Aberta

<b>SCMF</b>	Sistema de Controlo em Malha Fechada
<b>SFR</b>	Registo de Funções Especiais; <i>Special Function Register</i>
<b>SLIO</b>	<i>Serial Linked Input Output</i>
<b>SOF</b>	Início de Frame; <i>Start of Frame</i>
<b>SP</b>	<i>Stack Pointer</i>
<b>SRR</b>	<i>Substitute Remote Request</i>
<b>TCP/IP</b>	<i>Transfer Control Protocol/Internet Protocol</i>
<b>UART</b>	<i>Universal Asynchronous Receiver-Transmitter</i>
<b>WR</b>	Linha de Escrita

# 1 INTRODUÇÃO

## 1.1 CONTROLO COMPUTACIONAL DE ESTUFAS AGRÍCOLAS

A agricultura tem utilizado, tradicionalmente, métodos empíricos que não rentabilizam a sua produção, assim como está muito dependente de condições meteorológicas.

Com o objectivo de incrementar a produção agrícola por metro quadrado, foram desenvolvidas as estufas agrícolas que, para além da razão referida, permitem cultivos de elevado valor acrescentado. Estas permitem também a elaboração de estudos de conceitos de causa-efeito, que permitem a construção de modelos para melhorar a produção e a qualidade de determinada colheita.

Aliado à característica das dimensões das estufas serem limitadas, o que permite melhor actuação no processo, com a construção de modelos específicos para a cultura em causa, é possível através de equipamentos como: sensores, actuadores e sistemas de aquisição e controlo baseados em microcontroladores ou sistemas computacionais, fazer a monitorização e o controlo do processo em tempo real.

A vantagem da utilização de sistemas de aquisição de dados e controlo, em estufas agrícolas, relaciona-se com a cada vez maior importância em rentabilizar os meios energéticos. A racionalização destes meios ganha particular relevo no caso de estufas com controlo remoto, localizadas em regiões sem instalações de sistemas de energia eléctrica, e que usam, por exemplo, energias fósseis para a sua manutenção. A redução de custos energéticos ajuda por si a um aumento na rentabilidade económica do processo.

Com o contínuo decréscimo do custo de computadores e microcontroladores, os sistemas computacionais têm vindo a ganhar cada vez maior aceitação em aplicações de controlo ambiental de sistemas agrícolas, quando se tratam de aplicações em que:

- São controladas muitas variáveis;
- Existe um número elevado de dispositivos a controlar;
- É necessária atenção frequente ou constante;
- A mão de obra é cara.

As tecnologias de controlo por computador utilizam sistemas computacionais e outro *hardware* para monitorizar condições físicas ambientais, tomam decisões sobre as acções necessárias para modificar esse ambiente, e accionam dispositivos que provocam alterações do mesmo. Por exemplo, um computador pode ser utilizado para monitorizar a temperatura, e ligar ou desligar sistemas de aquecimento, para manter uma temperatura constante.

Geralmente os sistemas de controlo baseados em meios computacionais são utilizados para:

- Melhorar a fiabilidade e eficácia do sistema estufa;
- Reduzir a carga de trabalho do produtor;
- Melhorar a concordância entre as necessidades da cultura e o ambiente da estufa;
- Providenciar informação para ajudar a identificar problemas;

- Melhorar a qualidade e crescimento da colheita;
- Reduzir consumos energéticos, por unidade de produção, e melhorar os proveitos económicos.

### 1.1.1 Componentes de um sistema computacional

Para permitir as características anteriormente enunciadas é necessário um sistema de controlo em malha fechada (SCMF) computacional, ilustrado na Figura 1.1, que execute as seguintes etapas:

- Monitorização das variáveis de estado (temperatura, humidade, radiação solar, concentração de CO<sub>2</sub>, etc.);
- Comparação das variáveis de estado com o estado desejado para as mesmas;
- Decisão sobre as acções que é necessário realizar para alterar o estado do sistema;
- Execução de acções (ventilação, aquecimento, sombreamento, humedificação, etc.).

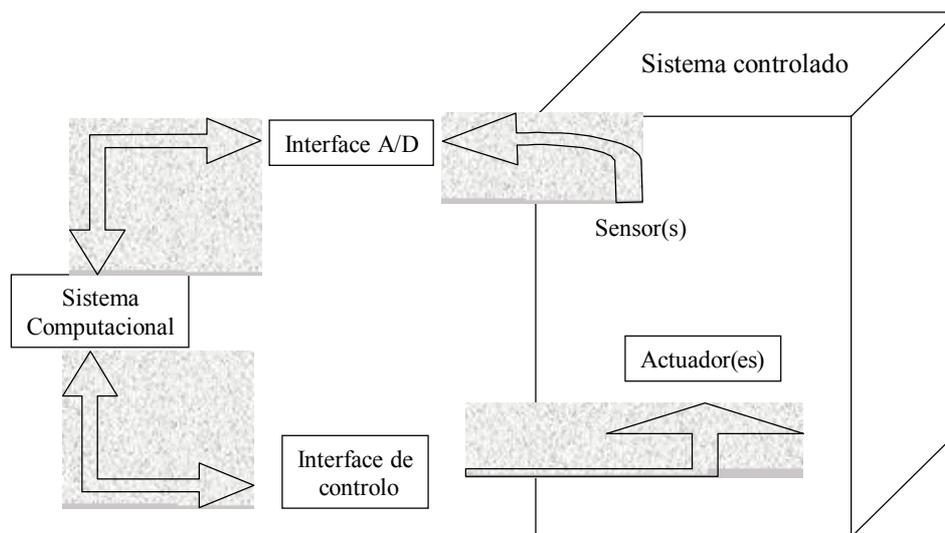


Figura 1.1 – Sistema de controlo em malha fechada.

A execução destas etapas exige uma combinação de *hardware* e *software* que deve ser implementada de acordo com a aplicação específica.

#### Componentes de *hardware*

- **Sensores.** Os sensores podem ser de diferentes tipos (contínuos ou discretos) capazes de medir diferentes grandezas (humidade relativa, temperatura, radiação solar, concentração de CO<sub>2</sub>, etc.). A sua escolha deve ser cuidada uma vez que da boa recolha de dados depende o desempenho geral do sistema.

- **Interface analógico-digital, A/D.** Uma vez que os computadores trabalham internamente com números (dígitos), os sinais eléctricos resultantes dos sensores devem ser convertidos para dados digitais. Isto é conseguido através de *hardware* específico designado por interface A/D.
- **Sistema computacional.** O *hardware* de conversão A/D é directamente ligado ao sistema computacional. Este sistema pode ser constituído por um computador pessoal, *Personal Computer* (PC), ou uma máquina dedicada apenas às tarefas de controlo geralmente baseada num microcontrolador. O tipo de máquina depende da aplicação, é dependente das características ambientais, complexidade do sistema controlado e da velocidade com que as conversões devem ser efectuadas. Muitas aplicações em ambientes agrícolas podem ser economicamente viáveis utilizando PC, o que é demonstrado pela utilização deste equipamento em sistemas de controlo, comercializados por diversos fabricantes.
- **Interface de controlo.** Através de *software*, são tomadas decisões para alterar o sistema controlado. As alterações são conseguidas recorrendo a dispositivos do sistema que afectem as variáveis monitorizadas. Estes dispositivos são designados por actuadores, respondendo a sinais provenientes da interface de controlo.

## Componentes de *software*

O *software* é utilizado nestes sistemas para implementar procedimentos adequados ao sistema que se pretenda controlar. Estes procedimentos são em geral bastante complexos, mas transparentes para o utilizador, porque este está geralmente mais preocupado com a facilidade de utilização e desempenho do sistema. O *software* deve proporcionar uma interface que permita a definição das características do sistema a ser controlado e a atribuição dos recursos de *hardware* de uma forma simples. A performance do *software* é medida pela capacidade do sistema computacional para manter o estado pretendido.

## Especificação dos parâmetros a monitorizar e a controlar

Num SCMF uma componente muito importante é a lógica utilizada para tomar decisões sobre a operação seguinte a realizar.

Sistemas ambientais como os biológicos em produção agrícola, são extremamente complexos por natureza. De facto, a sua complexidade é tal que descrever e prever o seu comportamento, em termos físicos e matemáticos, pode não ser possível [3]. Por forma a gerir esses sistemas, os mesmos devem ser simplificados e descritos em termos de um conjunto de valores mensuráveis que se saiba terem influência sobre a produção. Seguindo o exemplo das estufas, pode verificar-se que, a qualidade e quantidade dependem: da radiação solar, temperatura, humidade, concentração de CO<sub>2</sub>, idade das plantas, qualidade da água aplicada na irrigação, programas de nutrição e muitos outros factores. Uma vez que é caro, por vezes impraticável, ou impossível controlar todas estas variáveis, o conjunto de variáveis controladas é geralmente restrito a um conjunto menor, tal como: temperatura, humidade, radiação solar e concentração de CO<sub>2</sub>. Ou seja, o estado do sistema é definido pelo valor de um conjunto de variáveis mensuráveis.

As acções de actuação permitidas, por seu lado, dependem das variáveis de estado que se pretendam controlar em determinado sistema e da existência de actuadores para influenciar essas variáveis. Assim, para que seja realizado o controlo dos parâmetros mencionados, os mecanismos envolvidos consistem essencialmente na actuação recorrendo a:

- Sistemas de aquecimento e ventilação/arrefecimento;
- Sistemas de rega e nebulização;
- Sistemas para injeção de CO<sub>2</sub>;
- Painéis de sombreamento, etc.

## 1.1.2 Topologias de sistemas computacionais

### Sistema centralizado

A forma mais simples de configurar um sistema computacional consiste na utilização de um único computador central que inclui o necessário suporte de *hardware* e *software* para aquisição de dados e controlo. A Figura 1.1 ilustra um sistema deste tipo.

### Sistema distribuído

Um computador central pode ser ligado a outros dispositivos que executam tarefas específicas. Em sistemas de controlo mais avançados o computador central é utilizado para efectuar o *download* de instruções para controladores inteligentes ou outras unidades satélite.

A comunicação entre o computador central e as unidades satélite pode ser realizada através de:

- Comunicação por ligação série ou paralela;
- Ligação por telefone utilizando comunicação série;
- Onda portadora utilizando modulação pela linha de potência;
- Barramento de comunicações.

A ligação por telefone e por ligações série podem ser implementadas utilizando ligação com ou sem fios.

## 1.2 SISTEMAS DE CONTROLO UTILIZANDO MICRO-CONTROLADORES

O grande desenvolvimento das técnicas e tecnologias da integração no fabrico de circuitos integrados, capazes de conter num único *chip* um microcontrolador e vários periféricos, deu origem ao aparecimento de microcontroladores dedicados, capazes de atingir elevadas performances a baixo custo contribuindo para o aumento da sua popularidade. Esta atingiu tais proporções que, actualmente, existem protocolos em rede reconhecidos como *standard* para interligar microcontroladores em aplicações nas áreas de automação e controlo industrial.

As vantagens da utilização de microcontroladores dedicados são:

- Capacidade de memória e aritmética oferecem maior sofisticação do que a lógica discreta;
- Maior flexibilidade porque através de alteração do *software* e, eventualmente, dos sensores e/ou actuadores é possível controlar processos diferentes;
- Possuem uma melhor relação custo/qualidade do que os dispositivos analógicos ou digitais que substituem;
- Podem e devem ser utilizados sempre que o tamanho, peso, potência e custo forem factores preponderantes de escolha;
- Oferecem maior imunidade ao ruído;
- Permitem uma maior modularidade.

Actualmente a aplicação de redes de microcontroladores abrange desde o controlo de edificios inteligentes à automação industrial passando por aplicações na agricultura, existindo protocolos *standard*, interfaces e *software* que simplificam o desenvolvimento de aplicações de automação com controlo distribuído, como no caso o controlo ambiental de estufas agrícolas.

Protocolos *standard* para interligar microcontroladores, minimizam os esforços de desenvolvimento e, para além disso e mais importante, asseguram a compatibilidade com produtos e componentes de diferentes versões do mesmo protocolo o que permite a evolução dos sistemas.

Um dos protocolos com maior aceitação, é actualmente, o *Controller Area Network* (CAN) [4]. Este protocolo foi originalmente desenvolvido para aplicações, em tempo real, no campo da instrumentação e controlo de comunicações para a indústria automóvel e, posteriormente, aplicado noutras áreas.

O CAN foi adoptado pela indústria automóvel bem como por outros tipos de indústrias, devido à sua robustez e flexibilidade. A disponibilidade de circuitos integrados colocados no mercado por vários fabricantes, encoraja a sua utilização devido ao seu baixo custo.

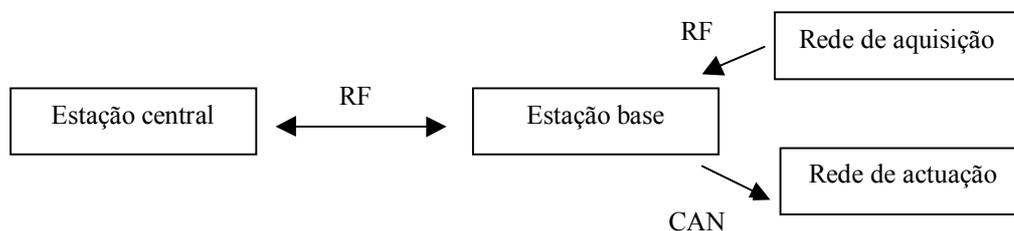
Algumas características deste protocolo de comunicações série são:

- É orientado à mensagem, sendo estas de tamanho curto e variável mas de formato fixo;
- Permite a exploração da implementação de redes multi-mestre;
- Permite taxas de transferência até 1Mbit/s;
- As mensagens transmitidas “alcançam” todos os nós do barramento, *multicast*. Após serem recebidas, as mensagens são filtradas com base nos identificadores que possuem;
- Implementa vários mecanismos de detecção e sinalização de erro que o tornam bastante fiável e robusto mesmo em ambientes bastante ruidosos.

### 1.3 O TRABALHO PROPOSTO

O trabalho que serve de base a esta dissertação pretende ser uma solução para um dos blocos de um SCMF, Figura 1.2, aplicando a electrónica e computação no controlo

ambiental de estufas agrícolas, no qual são utilizadas diversas plataformas de comunicação entre os diferentes blocos do sistema e aplicados algoritmos elaborados no sentido de atingir um controlo óptimo [1, 2].



**Figura 1.2 – Diagrama de blocos do SCMF computacional.**

Devido à mobilidade necessária dos sensores, no SCMF anteriormente mencionado, a aquisição é realizada através de uma rede própria que comunica com uma estação base através de ligação sem fios por rádio frequência (RF).

A estação base, composta por um PC e uma placa de interface, é a unidade de gestão integrando modelos, previamente validados, para o desenvolvimento das culturas e comportamento ambiental na estufa face às variáveis de controlo e às perturbações relevantes.

Este trabalho enquadra-se na componente de actuação do sistema anteriormente referido, permitindo a comunicação entre a estação base e microcontroladores ou outros dispositivos satélite, utilizados para o comando de diversos tipos de actuadores, possibilitando, assim, a alteração das variáveis cujo controlo seja considerado relevante para os algoritmos.

A solução proposta para a realização da componente de actuação do sistema é uma rede, composta por microcontroladores e outros dispositivos que “funcionam” como unidades remotas da estação base, utilizando como meio de comunicação o barramento série CAN. Assim, a rede de actuação proposta é composta pelas seguintes unidades:

- Unidades “inteligentes” com capacidade de desempenhar funções de *routing* e supervisão local;
- Unidades “não inteligentes” e mais económicas para operações simples de liga/desliga.

As unidades “inteligentes” da rede de actuação consistem em placas baseadas num microcontrolador, com entradas/saídas ligadas a sensores e/ou actuadores e com capacidade de armazenamento de dados. O *software* residente nestas unidades é composto pelos seguintes módulos:

- Módulo de iniciação;
- Módulo de actuação e alarme;
- Módulo de comunicações baseado no protocolo CAN.

Quanto às unidades “não inteligentes” estas são comandadas directamente por uma das unidades “inteligentes”, designada por *Host*, e são responsáveis pelas tarefas de liga/desliga dos actuadores.

As várias unidades constituirão uma rede, e cada uma comunicará com a placa de interface da estação base, através do barramento CAN. Esta placa de interface

comunicará, por sua vez, com o computador supervisor, através duma interface RS-232C e com os restantes nós da rede de actuação por CAN. Ao conjunto PC e placa supervisora atribuiu-se a designação de estação base ou PC-supervisor. Cada uma das estações base existentes, em cada estufa, poderão também comunicar por rádio frequência com uma estação central situada num local remoto, permitindo desta forma o controlo não só de uma estufa mas de um conjunto de estufas, tirando partido de todas as vantagens que daí podem advir [1].

O PC da estação base contém um programa onde reside a configuração de todo o sistema. Os parâmetros a controlar assim como os seus valores bem definidos e um módulo de comunicações, que lhe permite adquirir as leituras efectuadas pelas unidades remotas de aquisição do sistema, por rádio frequência, para compará-las com os valores de referência, *set-points*, e transmitir os modos de actuação/controlo às unidades que constituem a rede proposta no presente trabalho.

Devido à variação lenta dos parâmetros num sistema deste tipo, estufas agrícolas, a frequência de amostragem não é crítica. Como consequência da inércia das variáveis, é possível recorrer a um sistema de comunicações de tráfego periódico para com o computador central. A excepção a esta situação ocorre quando são actuados os sistemas de alarme, situação em que terá de haver a maior prioridade possível no processamento dos comandos adequados para resolução do problema.

Quanto ao tipo de protocolo de comunicações suportado pelo microcontrolador escolhido, o protocolo CAN é apropriado por ser de simples implementação, possuir elevada imunidade ao ruído e permitir a atribuição de prioridade às mensagens.

Em resumo: o presente trabalho descreve a utilização de uma rede CAN, para o comando de actuadores aplicados ao controlo climático de estufas agrícolas de acordo com ordens provenientes de uma estação base, a qual consiste num PC situado na estufa e devidamente protegido. Além da componente de comunicações foram também desenvolvidos circuitos de accionamento (interface de potência) para controlo liga/desliga de diversos actuadores e para o controlo de velocidade de um motor de indução.

## **1.4 ORGANIZAÇÃO DA DISSERTAÇÃO**

Com o objectivo de enquadrar o tema desta dissertação, o capítulo 1 constitui uma introdução ao trabalho desenvolvido, indicando os objectivos primários dos sistemas computacionais utilizados no controlo ambiental de estufas agrícolas, bem como uma abordagem muito resumida do sistema pretendido.

O capítulo 2 apresenta as acções geralmente utilizadas para controlar várias grandezas ambientais em estufas agrícolas, referindo os actuadores geralmente utilizados para o efeito.

O capítulo 3 apresenta o protocolo CAN (versão 2.0), referindo características do mesmo tais como: conceitos básicos, transferência de mensagens, suporte físico e controlo de erros. São também referidas as configurações de integrados vulgarmente utilizadas e os campos preferenciais de aplicação deste protocolo.

No capítulo 4 é feita uma abordagem ao sistema geral proposto para controlo de estufas agrícolas, no qual este trabalho se enquadra, justificando as plataformas de comunicação

utilizadas. É também apresentada a arquitectura da rede de actuação proposta, a qual constitui o objectivo desta dissertação.

Nos capítulos 5 e 6 é apresentada a implementação da rede de actuação proposta em termos de *hardware* e *software*, respectivamente. Assim, no capítulo 5 são apresentados os circuitos desenvolvidos para implementar os diversos nós da rede de actuação proposta, bem como os circuitos de accionamento projectados para soluções de controlo liga/desliga e para o controlo de velocidade de um motor de indução. No capítulo 6 é apresentado o *software* desenvolvido para a parte de comunicações do sistema bem como para o controlo de velocidade do motor de indução.

No capítulo 7 são apresentados e discutidos alguns testes e respectivos resultados, obtidos através de ensaios realizados, com o objectivo de justificar o sistema implementado.

No capítulo 8 são tecidas algumas considerações globais sobre o trabalho realizado, bem como sugestões que poderão ser efectuadas sobre o trabalho já desenvolvido.

As figuras e tabelas presentes neste relatório são numeradas com dois identificadores numéricos independentes, separados por um ponto. O primeiro identificador indica o capítulo ao qual essa figura ou tabela pertencem. O segundo identificador representa a numeração sequencial dentro de cada capítulo. A numeração sequencial das figuras e equações é independente. A figura 3.12, por exemplo, indica a figura 12 do capítulo 3.

O significado das abreviaturas utilizadas ao longo do texto está descrito após o índice.

As referências bibliográficas são indicadas por um número dentro de parênteses rectos. A numeração é efectuada de forma sequencial e associada a uma referência bibliográfica indicada na secção - Bibliografia.

## **2 ACCÇÕES DE ACTUAÇÃO EM ESTUFAS AGRÍCOLAS**

### **2.1 INTRODUÇÃO**

A crescente importância da utilização de estufas agrícolas, devido à necessidade de obter, com sucesso, culturas em ambientes e/ou épocas que à partida não seriam as mais favoráveis, bem como permitir que os produtos produzidos sejam de boa qualidade, impõem o desenvolvimento de sistemas para controlar grandezas consideradas importantes tais como a temperatura, humidade do ar e do solo, luminosidade, etc.

Assim, a acção que um sistema de controlo ambiental pode exercer sobre os factores ambientais de uma estufa agrícola, é limitada pelos equipamentos de actuação disponíveis na estufa onde está implementado, nomeadamente: sistemas de aquecimento, injectores de dióxido de carbono, ventiladores, cortinas de sombreamento, sistemas de rega, pulverizadores de pesticidas, etc. De seguida apresenta-se uma descrição de algumas das técnicas e equipamentos vulgarmente utilizados no controlo das grandezas físicas anteriormente referidas.

### **2.2 ESTUFAS**

A utilização de estufas agrícolas apresenta as seguintes vantagens e desvantagens:

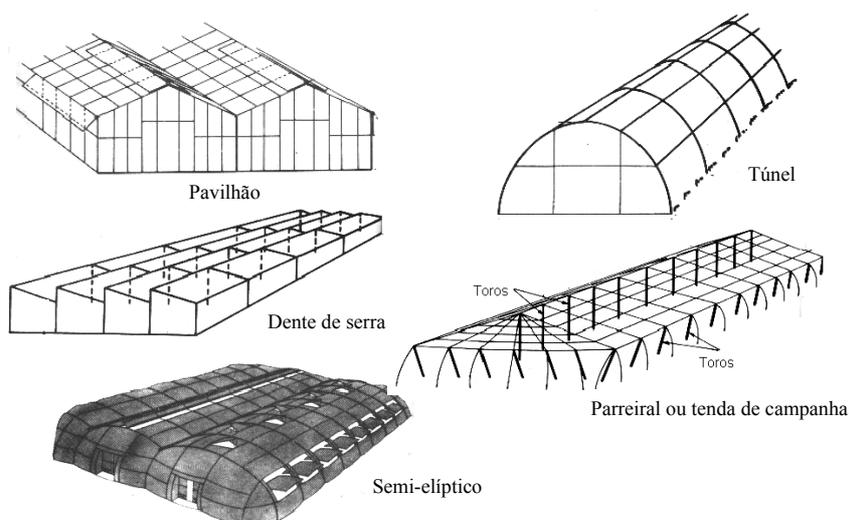
#### **2.2.1 Vantagens**

- Possibilidade de se produzirem produtos fora da época normal;
- Melhoria da qualidade;
- Maior precocidade;
- Melhor controlo das pragas e doenças;
- Aumento das produções;
- Economia da água de rega;
- Trabalho em melhores condições de comodidade e segurança;
- Melhor protecção contra os rigores do clima.

#### **2.2.2 Desvantagens**

- Encargos de produção;
- Exigência de maior especialização dos agricultores;
- Consumo energético;
- Maior utilização de pesticidas;
- Impacto ambiental.

### 2.2.3 Tipos de estufas



**Figura 2.1 - Classificação de estufas quanto à estrutura e ao perfil externo.**

A classificação mais corrente segundo Zoilo S. Cermeño [5] é a que atende à estrutura e ao perfil externo (ver Figura 2.1), isto é:

- Pavilhão – este tipo de estufas possuem tecto a formar um ou dois planos inclinados, ou seja a “uma água” ou a “duas águas”. Estas estufas são bastante utilizadas, em virtude da facilidade de construção e de conservação, por serem recomendáveis para a colocação de qualquer tipo de plástico na cobertura, pelas facilidades conseguidas na evacuação da água da chuva e, também, pela possibilidade de se unirem vários pavilhões em cascata;
- Dente de serra – este tipo de estufas é formado pela reunião em cascata de várias naves a uma água. Cada uma destas naves tem a cobertura formada por planos inclinados com cerca de  $30^\circ$ , com os lados mais altos apoiados sobre a extremidade superior das colunas mestras e com as bordas inferiores seguras também às colunas mestras mas a uma altura inferior. A largura destas naves pode ser considerável devido a ventilação zenital e ao facto da construção se fazer por adição de naves. Apresentam como vantagem a fácil ventilação mas possuem como desvantagem a evacuação da água da chuva, existente na vertente de cada um dos planos inclinados;
- Parreiral ou tenda de campanha – Neste tipo a estufa é construída à base de toros de madeira nos pés direitos e de arame galvanizado nas coberturas. A sua construção é idêntica à de uma tenda sendo das mais económicas. Os inconvenientes que apresenta são os seguintes: conservação e construção com necessidade de bastante especialização; rápido envelhecimento da instalação; pouco aconselhada para regiões pluviosas; dificuldade de trabalhos mecanizados devido ao número de toros, arames de ligação, etc.;
- Túnel ou semi-cilíndrico – A estrutura deste tipo de estufa é constituída por pés direitos e arcos. Este tipo de estufas é bastante aceitável pelas suas qualidades de diafanidade, controlo de temperatura, distribuição de luminosidade, etc.;

- Semi-elíptico – Este tipo de estufas apresenta grande diafanidade, considerável volume de ar, poucos obstáculos no interior e são bastante estanques. Trata-se de um tipo de muito interesse em zonas chuvosas.

Existe também um tipo de estufa, denominado hidropónico, que se distingue das estufas convencionais porque o meio onde as plantas vegetam é num ambiente inerte onde os nutrientes essenciais à planta são garantidos por uma solução nutritiva de composição variável [6, 7] e não pelo solo. Se convenientemente mantida, uma produção hidropónica é rápida quando comparada com uma cultura normal dependente do solo.

Os sistemas hidropónicos apresentam as seguintes vantagens quando comparados com outros sistemas [6]:

- Fazem reciclagem das soluções de nutrientes para serem utilizadas no próximo ciclo de rega, reduzindo desperdício de água e fertilizantes;
- Os nutrientes a fornecer às plantas podem ser fornecidos de forma precisa e alterados para satisfazer as necessidades das plantas;
- O controlo de doenças neste tipo de estufas é reforçado através da eliminação de uma das suas fontes mais importantes, o solo;
- Uma vez que o sistema de raízes não tem que competir pelo fornecimento de nutrientes, é possível a existência de um maior número de plantas por espaço disponível;
- Permite a adaptação em diversos locais, independentemente do tipo de solo;
- Melhor quantidade e qualidade;
- Redução da mão-de-obra;
- Obtenção de culturas com ciclo mais curto.

Na sua maioria, os sistemas hidropónicos são constituídos essencialmente por um reservatório de água com nutrientes, uma calha de crescimento, um caminho para entrega de água e nutrientes às raízes e por um meio que substitui o solo.

Alguns tipos de meios mais frequentemente utilizados para substituir o solo são: turfa, *perlite*, argila expandida, *vermiculite*, areias lavadas, britas e materiais sintéticos (ex. lã de vidro)[6].

Apesar das vantagens relativas, os sistemas hidropónicos tem de ser monitorados com regularidade. O controlo do pH e a concentração dos nutrientes são os factores mais críticos numa produção hidropónica de sucesso, recorrendo para isso a vários utensílios (sensores e actuadores).

As actuações, são executadas por diversos acessórios dedicados a este tipo específico de estufas [8]. Destes acessórios constam bombas submersíveis, bombas de ar, canalizações, temporizadores, medidores de pH, misturadores, soluções para ajuste do pH, entre outros.

Nas seguintes secções deste capítulo serão referidas as técnicas vulgarmente utilizadas no controlo da atmosfera de estufas convencionais, ou seja, nas quais a cultura depende do solo.

## **2.3 CONTROLO DA ATMOSFERA DUMA ESTUFA**

Nesta secção pretende-se abordar técnicas utilizadas no controlo de várias grandezas ambientais em estufas agrícolas.

### **2.3.1 Controlo de temperaturas elevadas**

Para o controlo de temperaturas elevadas dentro da estufa recorre-se a diversas técnicas para conseguir manter as mesmas dentro de limites considerados aceitáveis para uma determinada cultura. Assim, tendo como objectivo a utilização da estufa durante todo o ano, os produtores dispõem de vários sistemas de ventilação, arejamento e limitação de radiação solar incidente (ecrãs de sombreamento) em função da estufa e da região onde a mesma está implantada. Seguidamente serão focadas algumas técnicas utilizadas.

#### **2.3.1.1 VENTILAÇÃO**

A ventilação de uma estufa [9], consiste na remoção do ar interior e substituição por ar exterior, podendo ser natural ou forçada, recorrendo a última a ventiladores e outros equipamentos. A ventilação é necessária para controlar as altas temperaturas do Verão, devidas à radiação solar, manter a humidade relativa em níveis aceitáveis durante o Inverno, permitir um fluxo constante do ar pela totalidade da estufa e manter níveis aceitáveis da concentração de gases.

##### **2.3.1.1.1 Ventilação Estática ou natural**

#### **Princípio**

É a mais utilizada, fundamentando-se no facto de as diferenças de densidade (o ar quente é mais leve do que o ar frio) e de pressão, provocarem trocas de ar espontâneas entre o interior e o exterior. Dispondo de várias superfícies móveis (aberturas) em diferentes posições (laterais e zenitais) da estufa, é possível conseguir ventilação da estufa de uma forma natural.

#### **Técnicas**

##### **Aberturas zenitais**

Este tipo de aberturas são constituídas por painéis móveis e contínuos colocados no tecto ou parte superior da estufa. Em estufas de vidro estes painéis podem ser constituídos apenas por vidros móveis independentes uns dos outros. As aberturas são accionadas manualmente ou por meio de dispositivos mecanizados geralmente recorrendo ao motor eléctrico e uma cremalheira. Geralmente estas aberturas representam entre 10 a 15% [5] de abertura relativamente à superfície coberta dependendo este valor da existência ou não em simultâneo de aberturas laterais. A Figura 2.2 ilustra um sistema de comando para aberturas deste tipo.



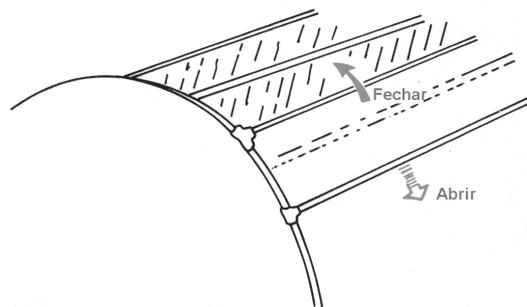
**Figura 2.2 - Fotografia de estufa com comando eléctrico de aberturas zenitais.**

### **Aberturas laterais**

Este tipo de aberturas são compostas por superfícies móveis colocadas nas paredes laterais das estufas, existindo vários tipos de implementação dos quais se indicam alguns vulgarmente utilizados [5]:

- Janelas enroláveis;
- Janelas pregueáveis;
- Janelas giratórias;
- Janelas deslizantes.

Exemplo: uma abertura lateral enrolável é uma abertura regulável, enrolando a parte de baixo do plástico em volta de um tubo, que é colocado sobre dois encaixes instalados a alturas diferentes conforme está ilustrado na Figura 2.3.

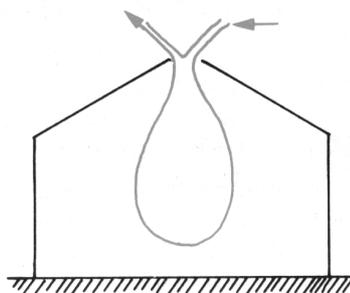


**Figura 2.3 – Estufa com técnica de abertura lateral enrolável.**

Dos tipos de aberturas referidos anteriormente, os dois últimos são os mais comuns em sistemas mecanizados.

## Eficácia dos diferentes sistemas

As aberturas zenitais situadas na cumeeira da estufa são mais eficientes, ventilando tanto como duas ou três janelas de fachada lateral de igual superfície. Cria-se um efeito de chaminé que favorece a circulação de ar. O ar quente ao evacuar-se provoca corrente de ar, sobretudo quando as aberturas zenitais são accionadas em simultâneo dos dois lados, como mostra a Figura 2.4.



**Figura 2.4 - Trajecto do ar utilizando ventilação estática por aberturas zenitais.**

O sistema de abertura zenital é eficaz, sobretudo se estiver associado a uma abertura lateral sendo, no entanto, de construção mais dispendiosa. Um sistema deste tipo é interessante para conseguir boa circulação de ar, sobretudo se as estufas não forem constituídas por mais do que três ou quatro capelas (problema da ventilação das capelas centrais) [10].

## Técnicas e equipamentos utilizados

Neste tipo de ventilação o mecanismo de actuação consiste geralmente em rodas dentadas que engrenam numa cremalheira colocada nas janelas; desta forma, quando as rodas giram, fazem elevar ou descer a cremalheira e, por consequência, a janela.

Este sistema pode accionar-se por processos manuais ou de uma forma mecânica por aplicação de um motor eléctrico.

No caso do motor eléctrico o comando é realizado utilizando relés ou contactores e fins de curso, sendo as potências dos motores da ordem de 0,25 a 1 HP.

## Vantagens e inconvenientes

A ventilação estática não necessita de grande investimento, permite a renovação do ar ambiente e diminuição do seu estado higrométrico.

A ventilação estática é na maioria dos casos suficiente para limitar as temperaturas excessivas em certas regiões, podendo diminuir a temperatura 3 ou 4 °C, no entanto, noutras regiões é necessário recorrer-se a outras técnicas (nebulização, painéis de sombreamento, etc.).

A eficácia deste tipo de ventilação está dependente da direcção e velocidade do vento, sendo mais aconselhada para regiões onde predominem ventos moderados. Utilizando estes sistemas, não se conseguem taxas elevadas de renovação de ar.

### 2.3.1.1.2 Ventilação Forçada

#### Princípio

Consiste na renovação de ar utilizando ventiladores-extractores, substituindo o ar quente do interior por ar da atmosfera exterior, conforme ilustrado na Figura 2.5.

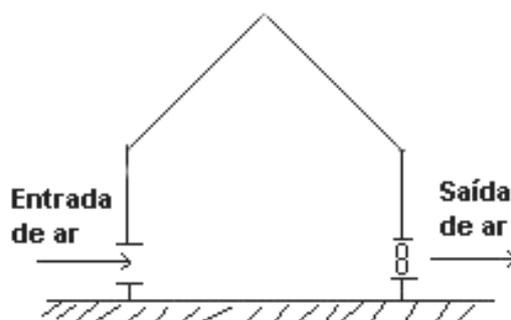


Figura 2.5 - Ventilação forçada.

#### Técnicas

A técnica mais utilizada é a ventilação por depressão, na qual são utilizados ventiladores-extractores para aspirar o ar da estufa para o exterior. A entrada de ar faz-se por entradas situadas no lado oposto ao dos ventiladores, as quais podem ser:

- Orifícios repartidos na parede que se fecham quando os ventiladores são desligados;
- Aberturas laterais comandadas de forma manual ou motorizada;

#### Equipamentos

Na escolha do ventilador, é necessária a seguinte informação que influencia o tipo e tamanho do mesmo [11]:

- Volume de ar que deve ser removido por unidade de tempo;
- Pressão estática – valor estimado e variações esperadas;
- Espaço disponível para instalar os ventiladores;
- Quantidade de ruído permitida;
- Eficiência – escolha do ventilador que permita a ventilação desejada com a menor potência possível;
- Considerações económicas.

Geralmente quando um ventilador é escolhido para determinada aplicação, de acordo com o exposto anteriormente, o motor recomendado é fornecido pelo fornecedor do equipamento. Para a maioria das aplicações onde por vezes as humidades são elevadas ou onde o ar é poluído, torna-se essencial que o motor seja estanque, para proteger as bobinas do motor dos efeitos corrosivos provocados pelos factores anteriormente referidos, os quais tendem a diminuir o tempo de serviço do motor. Os

motores dos ventiladores geralmente utilizados em estufas agrícolas dependem do tamanho das mesmas (por exemplo numa estufa com cerca de 200 m<sup>2</sup>, na U.T.A.D., é utilizado um motor com a potência de 0,75 HP). O tipo de controlo geralmente utilizado é do tipo liga/desliga utilizando contactores ou relés.

### **Vantagens e inconvenientes**

Este tipo de ventilação limita as temperaturas excessivas e assegura uma boa distribuição do ar. Permite ainda gerir a taxa de renovação de ar e limitar a higrometria dentro da estufa. As taxas de renovação são em geral na ordem de 40 renovações por hora [10], sendo uma renovação equivalente a retirar da estufa uma quantidade de ar igual ao seu volume.

Esta técnica apresenta como desvantagem os gastos energéticos, devendo a sua utilização ser alvo de estudo prévio de viabilidade.

### **2.3.1.2 REFRIGERAÇÃO POR HUMIDIFICAÇÃO - “COOLING SYSTEMS”**

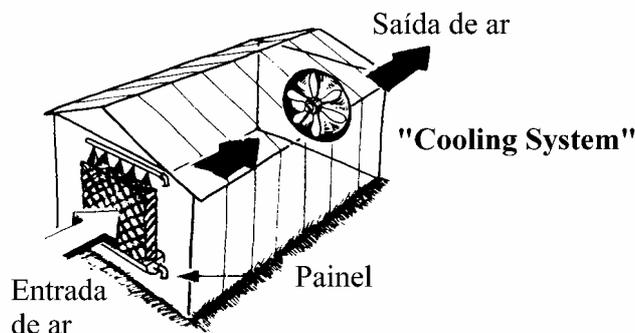
Este sistema tem como principal finalidade reduzir a temperatura da estufa; baseia-se na propriedade física da evaporação de uma determinada quantidade de água necessitar de absorver calor [5, 12].

### **Princípio**

Antes de atravessar a estufa, o ar exterior aspirado pelos ventiladores passa por um material humedecido libertando parte do seu calor por evaporação da água.

### **Técnicas**

Este sistema é geralmente constituído por um ou mais painéis, compostos por uma estrutura em cartão possuindo pequenos alvéolos [6]. Esta estrutura encontra-se normalmente colocada numa parede frontal da estufa. Na parede frontal oposta são instalados ventiladores-extractores, que provocam a entrada e passagem de ar através dos painéis “alvéolados”, onde sobre os quais existe circulação contínua de água. Os ventiladores ao serem ligados provocam a entrada de ar para a estufa. Ao passar pelos alvéolos, este fica carregado de humidade, refrescando o interior da estufa. A Figura 2.6 ilustra um sistema deste tipo.



**Figura 2.6 - Sistema de refrigeração e humificação.**

O painel (“colchão”) húmido situado num dos topos da estufa é uma superfície alveolar imputrescível mantida por armaduras em alumínio ou em aço galvanizado.

A existência de uma tubagem perfurada na parte superior assegura a repartição da água pelo painel. A água ao escorrer sobre o painel evapora-se parcialmente quando em contacto com o ar que o atravessa.

Na base do painel existe uma caleira que recolhe o líquido não evaporado e encaminha-o para um reservatório. A partir deste e utilizando uma pequena bomba elevatória a água é disponibilizada para participar num novo ciclo.

## **Equipamentos**

Segundo Zoilo S. Cermeño [5] a potência necessária do motor para mover a bomba de distribuição da água é de 2 HP por cada 1000 metros quadrados de superfície coberta.

Os ventiladores-extractores são do tipo helicoidal, com 1 metro de diâmetro aproximadamente. O motor que move cada um destes ventiladores é de 1.5 a 2 HP, com capacidade variável de 200 a 400 metros cúbicos de ar por minuto; a velocidade oscila entre 750 a 1000 r.p.m. São necessários três ventiladores-extractores por cada 1000 metros quadrados de superfície coberta da estufa.

A distância máxima recomendada entre o painel e os ventiladores é de  $3600/R$  [5], sendo R o número de renovações de ar por hora; o número de renovações de ar considerado óptimo situa-se próximo de 50 por hora. A distância óptima entre ventiladores e painéis deve estar compreendida entre 60 e 80 metros. A distância entre ventiladores deve ser de 7 a 10 metros. O arranque e a paragem automáticos dos ventiladores e bomba de água (electrobomba) são realizados através de contactos que abrem ou fecham (utilizando, por exemplo, contactores ou relés comandados por um termóstato).

Nas estufas da U.T.A.D e da Escola Superior Agrária de Castelo Branco com dimensões próximas dos 200 m<sup>2</sup>, são utilizados ventiladores-extractores de 0,75 HP, cada, num sistema deste tipo.

## **Vantagens e inconvenientes**

Este sistema, ao permitir a diminuição da temperatura e aumento da taxa de humidade, influencia favoravelmente o desenvolvimento de vegetais. Refira-se também que este sistema permite uma boa eficácia se o número de renovações de ar estiver compreendido entre 45 e 60 por hora [5] sendo melhorada durante períodos secos, onde a evaporação se realiza com maior facilidade.

Quanto às limitações ou desvantagens em utilizar um sistema destes referem-se as seguintes [10]:

- Investimento necessário para um sistema deste tipo é considerável e, tal como outros, deve ser alvo de um estudo prévio de viabilidade;
- A estufa onde se utilize um sistemas destes deve ser o mais estanque possível para conseguir dele o maior rendimento;
- Estabelece-se um gradiente de temperatura e humidade na estufa, estando mais húmido o ar do lado do painel humedecido do que do lado dos

extractores o que faz com que o comprimento das estufas, que utilizem este sistema, não deva exceder os 30 metros;

- Nas regiões onde a água contém elevado teor de sais minerais, os painéis humedecidos deterioram-se com facilidade.

### 2.3.1.3 ECRÃS DE SOMBREAMENTO

#### Princípio

Tem como objectivo criar uma sombra artificial fazendo com que a energia que entra na estufa, proveniente da radiação solar, diminua [13] evitando temperaturas demasiado elevadas.

#### Técnicas

As redes de sombreamento podem ser fixas ou móveis, sendo as últimas as que possuem interesse para os sistemas de automação. Estes sistemas móveis, que recorrem a motores e detectores de fim de curso, são normalmente instalados no interior da estufa. A colocação destes sistemas no exterior apresenta alguns inconvenientes, destacando-se o facto de poderem ser danificados pela acção de determinadas condições atmosféricas, tais como, ventos fortes e granizo. Os produtos mais utilizados são as redes de cor escura, em diversos materiais [5].

Os ecrãs de sombreamento podem aplicar-se a todas as culturas, e não só em regiões onde se verifiquem elevados níveis de radiação solar. A sua utilização é particularmente necessária em plantas que exijam baixos níveis de luminosidade e onde seja necessário reduzir o calor durante os dias mais solarengos. A Figura 2.7 mostra a utilização de ecrãs de sombreamento numa estufa.



**Figura 2.7 - Fotografia de estufa contendo ecrãs de sombreamento.**

Existem também aplicações onde se recorre a quebra-luzes opacos, os quais evitam a iluminação da cultura, empregando-se para essa finalidade material de polietileno negro que, colocado por cima da cobertura da estufa ou fazendo túneis no interior da instalação, permite controlar as horas de luz que se pretendem anular.

## Equipamentos

Neste tipo de sistemas a actuação é levada a cabo por motores colocados em posições estratégicas dentro da estufa e que movem os painéis de sombreamento, bem como por sensores de fim de curso. A potência destes motores depende dos painéis de sombreamento. Numa das estufas da Escola Superior Agrária de Castelo Branco são utilizados motores com a potência de 0,25 HP. Refira-se, também, que estes motores contêm geralmente um desmultiplicador para permitir melhor controlo no arrastar dos painéis.

## Vantagens e inconvenientes

O ecrã ou painel de sombreamento permite limitar e manter a temperatura, em determinadas estufas, 5 a 6 °C acima da temperatura exterior [10].

A escolha do limiar de radiação solar interior, deverá ser elevada, da ordem dos 600 w/m<sup>2</sup> (radiação solar global) para:

- Limitar a cobertura a curtas durações durante o dia (3 a 5 horas em dias de sol);
- Evitar desdobramento quando o tempo estiver encoberto;
- Não limitar a radiação útil à fotossíntese.

Os ecrãs de sombreamento limitam também as trocas convexas e diminuem a eficácia da ventilação estática sobretudo se os ventos forem fracos.

### 2.3.1.4 OUTRAS TÉCNICAS

Outras técnicas utilizadas para diminuir a temperatura em estufas são [5]:

- **Refrigeração da cobertura** utilizando uma cortina de água que desliza pela mesma;
- **Movimento do ar interior** utilizando ventoinhas que conseguem homogeneizar a temperatura, aumentar a evaporação da água do solo aumentando a transpiração das plantas, a humidade no ambiente da instalação e diminuir a temperatura.

## 2.3.2 Controlo de temperaturas baixas

Em muitos casos, a protecção fornecida pela estufa é suficiente para que a cultura se possa desenvolver durante o Inverno sem o recurso a sistemas de aquecimento. Contudo, existem regiões que, devido às suas baixas temperaturas, requerem a necessidade de utilização destes sistemas e/ou de isolamento adicional nas estufas.

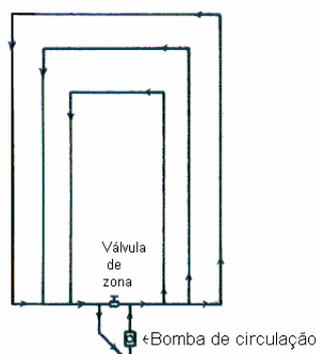
Uma estufa com aquecimento, quando comparada com uma que não o possui, apresenta as seguintes vantagens:

- Acelera a produção;
- Possibilita a produção de produtos fora de época;
- Melhor controlo das doenças, pois o aquecimento ajuda a evitar a ocorrência de humidade excessiva;

- Dependendo do combustível utilizado e do sistema de queima, permite dispor de dióxido de carbono resultante dos gases da combustão.

As estufas podem ser aquecidas utilizando diversas técnicas. O aquecimento pode ser exercido sobre a atmosfera interior da estufa, o solo ou ambos.

Geralmente o aquecimento do ar faz-se através da circulação de água quente em tubos, aquecimento indirecto, ou através da utilização de aquecedores que usam ar forçado, aquecimento directo. No solo é normalmente utilizada uma rede de tubos enterrada ou colocados à superfície (neste caso o ar também é aquecido) por onde circula água com uma temperatura que em regra é da ordem dos 20 a 25 °C (Figura 2.8). Em alternativa pode usar-se uma rede de resistências eléctricas enterrada no solo [5].



**Figura 2.8 – Sistema de aquecimento utilizando água quente.**

Os sistemas de aquecimento directo possuem a vantagem de ter menor tempo de resposta.

Alguns dos sistemas [14] de aquecimento normalmente utilizados para aplicação em estufas são:

- Aquecedores unitários;
- Sistemas a água quente;
- Sistemas de aquecimento a vapor.

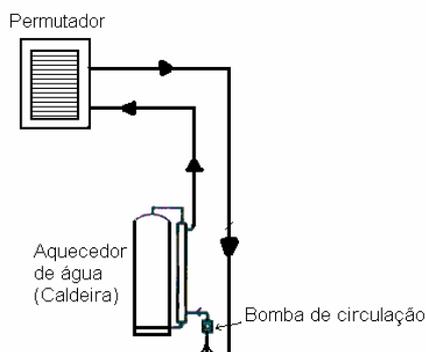
### **2.3.2.1 AQUECEDORES UNITÁRIOS**

Nestes sistemas as unidades aquecedoras são espaçadas convenientemente, sendo instaladas no solo ou suspensas e, geralmente, são alimentadas a gás natural armazenado em garrafas ou a gasóleo utilizando ventiladores para a distribuição do calor. Estes sistemas requerem um investimento relativamente moderado, são fáceis de instalar e expandir. Se utilizados, devem ser distribuídos por forma a cobrir toda a área com ar quente [14].

### **2.3.2.2 SISTEMAS A ÁGUA QUENTE**

Estes sistemas [14] utilizam canalizações que podem ser dispostas ao longo do perímetro da estufa a aquecer ou utilizando permutadores unitários a ar forçado por ventoinha, colocados numa estrutura aérea dentro dela. Sistemas como estes requerem

uma caldeira, electroválvulas e outros dispositivos de controlo (Figura 2.9). No entanto, um sistema a água quente é simples de instalar e geralmente requer menor manutenção do que um sistema a vapor. Existe um aquecimento e arrefecimento lento das canalizações, mas as temperaturas são geralmente mais uniformes.



**Figura 2.9 – Sistema utilizando água quente.**

### 2.3.2.3 SISTEMAS DE AQUECIMENTO A VAPOR

Um sistema de aquecimento a vapor necessita de uma caldeira, electroválvulas e outros dispositivos de controlo, dependentes do tipo e tamanho da caldeira utilizada. O vapor permite um aquecimento e arrefecimento rápido das linhas de vapor e geralmente necessita de menos canalização [14]. Um sistema de aquecimento a vapor pode ser utilizado também para a pasteurização do solo. Este tipo de sistemas requer um investimento inicial elevado, no entanto, apresenta uma considerável longevidade. Utiliza-se em grandes áreas, uma vez que o vapor pode ser transportado eficientemente através de longas distâncias.

### 2.3.2.4 FACTORES RELEVANTES PARA SELECÇÃO DO SISTEMA DE AQUECIMENTO

Conforme se pode depreender do que foi anteriormente referido, é imprescindível uma fonte de calor por forma a tornar possível o controlo de temperatura. A escolha dessa fonte de calor depende de vários factores alguns dos quais já foram referidos. Estes factores devem ser considerados antes do investimento inicial, uma vez que este depende do tipo de sistema a ser utilizado. Antes de escolher o tipo de sistema é necessário calcular a quantidade de calor requerida, que é determinada tendo em consideração as condições mais desfavoráveis às quais se presume o sistema vai ser submetido. A temperatura interna mínima depende da cultura em crescimento. Para efectuar o cálculo da margem de calor em °F que o sistema de controlo deve permitir, subtrai-se à temperatura que consideramos como ideal, a temperatura mínima nas condições mais adversas previstas. Um processo simples para determinar a quantidade de calor obtêm-se multiplicando a área da estufa pela diferença de temperatura máxima a ser mantida, e este resultado por um factor que depende da natureza da cobertura e factores de construção conforme traduz a Equação 2.1.

$$Q=KS(T_{\text{máx}} - T_{\text{mín}}) \quad (2.1)$$

- S – Área da superfície da estufa
- K – factor de transferência de calor que depende da cobertura e da qualidade de construção.
- $T_{\text{máx}}$  – temperatura máxima esperada
- $T_{\text{mín}}$  – temperatura mínima esperada

Um factor de transferência de calor de 1 ou 1,2 é utilizado [14] para estufas cobertas por uma única camada de polietileno ou plástico rígido, sendo 1 para uma estufa bem construída e 1,2 para estufas de pior qualidade de construção e com algumas falhas de isolamento (fugas de ar). Quanto maior for a velocidade do vento maiores são as perdas de calor. Um factor de 0 a 0,75 é utilizado para estufas cobertas com uma camada dupla de polietileno com espaço aberto de 7/10 cm mas não mais do que 10 cm. Um factor de 1,1 a 1,4 se a estufa for coberta por vidro. Um factor de 10% deve ser adicionado a todos os parâmetros anteriores se a estufa estiver localizada num sítio ventoso e se houver possibilidade de infiltração de ar nas estufas.

Em resumo, pode concluir-se que o sucesso no aquecimento de estufas está dependente do correcto dimensionamento e instalação do sistema de aquecimento, dos controlos adequados e dos métodos para obter uma distribuição uniforme. O tipo de construção da estufa, colheita a efectuar e níveis de temperatura a manter, são factores importantes a serem considerados na selecção e instalação de qualquer sistema de aquecimento.

Após a selecção do sistema de aquecimento, é necessário ter cuidados na manutenção para que se possa tirar o maior rendimento possível e garantir longevidade, justificando assim o investimento inicial. Os processos para a manutenção destes tipos de aquecimento [15, 16], devem ser cumpridos o mais escrupulosamente possível.

Também neste tipo de actuação o controlo utilizado é liga/desliga, porque, numa estufa, a temperatura não varia bruscamente. Assim, o tipo de acções a realizar, para conseguir o controlo referido, consiste em accionar e desligar aquecedores, electroválvulas, caldeiras, etc.

### **2.3.3 Irrigação**

Nesta secção pretende-se abordar, de uma forma resumida, algumas técnicas utilizadas em estufas para controlar as suas necessidades em termos de água e a sua importância para o desenvolvimento das plantas.

#### **O papel da água no desenvolvimento das plantas**

A água é o constituinte principal dos tecidos das plantas (70 a 95 %) desempenhando várias funções na vida das mesmas, destacando-se uma função mecânica responsável pela rigidez dos tecidos jovens e uma função fisiológica responsável pelo transporte de elementos nutritivos dissolvidos (seiva bruta) e dos produtos do metabolismo (seiva elaborada)[10].

Outra função importante da água é no processo da fotossíntese onde é necessária para a elaboração da matéria orgânica, que vai constituir os “materiais” da planta.

## **A água no solo**

As plantas encontram no solo a água que necessitam. Em culturas ao ar livre esta é conseguida de forma regular através da precipitação que depende das regiões. Nas estufas a água é fornecida por rega.

### **Transpiração, evaporação, evapotranspiração**

A água do solo dissipa-se através da acção da energia solar por:

- Evaporação ao nível do solo;
- Transpiração ao nível da planta. A transpiração favorece a absorção da água pelas raízes e a circulação da seiva, criando um caminho de água.

Estas duas perdas estão associadas no termo evapotranspiração.

A evapotranspiração potencial (ETP), é a quantidade máxima de água perdida pelo solo (evaporação) e pelas plantas (transpiração) quando estas encontram toda a água que necessitam e cobrem o solo na totalidade. A ETP exprime-se em milímetros de água por dia e depende dos seguintes factores:

- Humidade do solo, quanto mais húmido for o solo maior é a importância da evapotranspiração;
- Humidade do ar;
- Temperatura do ar. Um ar seco e quente aumenta a evapotranspiração;
- Agitação do ar, o vento favorece a evapotranspiração;
- Espécie vegetal.

#### **2.3.3.1 MÉTODOS DE DISTRIBUIÇÃO DE ÁGUA - REGA**

Os sistemas de rega mais utilizados em estufas são: rega por pé, rega por aspersão e rega gota-a-gota. Em casos especiais, a rega também pode ser subterrânea.

##### **2.3.3.1.1 Rega por pé**

Na rega por pé a água é conduzida ao solo da cultura a regar por meio de: regueiras, valas, mangueira, etc.; uma vez chegada a água inunda a parte do solo que deve ser regada. Este tipo de rega dá lugar a grande perda de líquido e a um controlo deficiente, porque origina excesso de humidade, tanto no solo como no ambiente da estufa.

##### **2.3.3.1.2 Rega por aspersão**

Este tipo de rega pode situar-se sobre a vegetação, molhando toda a folhagem, ou na forma baixa, de tal forma que só se molhem os caules e a parte baixa da folhagem.

Utilizando esta técnica a água é distribuída através de aspersores que difundem a água sob a forma de pequenas gotículas. Os aspersores são posicionados de forma a que os seus jactos de água se dividam e espalhem sobre o perímetro a regar.

## Utilidade

Algumas das vantagens que este tipo de irrigação permite são [5]:

- Economizar água;
- Possibilidade de regar sem efectuar o nivelamento do terreno;
- Humidificar o solo antes de uma sementeira, replantação de plantas ou plantação de “estacas”;
- Regar as plantas jovens que ainda não estão enraizadas no solo;
- Aumentar a higrometria e evitar temperaturas excessivas, permitindo um refrescamento das plantas quando as temperaturas são elevadas;
- Facilita a aplicação directa dos fertilizantes e pesticidas com a água;
- Possibilidade de aplicação nas horas do dia em que a estufa apresente o mais baixo nível de humidade; o que costuma coincidir com o meio-dia solar.

## Técnicas e Equipamentos utilizados

A instalação mais frequente consiste em tubos fixos munidos de aspersores, regularmente espaçados, colocados por cima das culturas.

Os equipamentos de rega por aspersão possuem os seguintes elementos [5]:

- Equipamento moto-bomba, composto por um motor eléctrico ou de explosão, e uma bomba para poder impulsionar a água à pressão necessária ao sistema de rega;
- Sistema de distribuição composto por uma tubagem principal que pode ser de diversos materiais (alumínio, ferro galvanizado, policloreto de vinilo, etc.) principais e por uma rede secundária onde estão ligadas as bocas ou aspersores;
- Na rede de distribuição instala-se uma série de válvulas de passagem e filtros. Os filtros evitam o entupimento das saídas com as impurezas que a água pode conter;
- Saídas e aspersores que podem ser de diferentes tipos e materiais;
- Válvulas de descarga para evitar o gotejamento após terminada a rega em instalações aéreas;
- Programadores que são os órgãos principais do sistema. Estes estipulam os intervalos de tempo em que as válvulas estão abertas ou fechadas;
- Válvulas automáticas, as quais estão ligadas a um programador por um tubo de controlo ou cabo eléctrico conforme sejam hidráulicas ou eléctricas, respectivamente. As válvulas abrirão ou fecharão a passagem da água para os aspersores da respectiva estação

## Vantagens e inconvenientes

A aspersão necessita de pouca vigilância. O problema das canalizações ou dos aspersores entupirem são raros, porque os orifícios de distribuição de água não estão expostos à projecção de terra.

Por outro lado, os requisitos de água são frequentes e significativos (débito de 100 l/h)[10]. O equilíbrio entre necessidades e consumos não se verifica, na maioria dos casos, fazendo com que o perigo da ausência ou excesso de água não seja desprezável.

Outro factor importante é a possibilidade de desenvolvimento de moléstias em consequência da humidade existente sobre a folhagem das plantas.

### 2.3.3.1.3 Rega localizada ou gota-a-gota

#### Princípio

A rega localizada fornece água à planta na região vizinha das raízes de cada planta (Figura 2.10), contrariamente à irrigação por aspersão que fornece água à totalidade da superfície cultivada. No solo existirão volumes bastante húmidos (perto das raízes) e outros bastante menos.

O objectivo é evitar o “*stress*” hídrico da planta e manter a humidade do solo na vizinhança das raízes dentro de limites considerados razoáveis.

A rega localizada é bastante utilizada em produções intensivas, permitindo o fornecimento de água em pequenas quantidades durante um longo período de tempo, sendo uma forma relativamente fácil de servir as necessidades das plantas.

A principal técnica de rega localizada consiste na rega vulgarmente designada por gota-a-gota a qual consiste no seguinte: uma rede de tubagem de material plástico flexível, com orifícios de saída calculados para que sejam atravessados por uma quantidade fixa e constante de água, é colocada por cima da parcela a regar. Nesta rede parte-se de uma tubagem principal da qual derivam tubagens secundárias, quase sempre uma por cada fila de cultivo. À entrada da rede de distribuição coloca-se um filtro, para evitar obstrução das saídas, e um manómetro, para regular a pressão da água que na maioria dos sistemas é de 1 atmosfera [5]. A quantidade de água que sai de cada furo de descarga é geralmente compreendida entre 2 e 24 litros por hora.

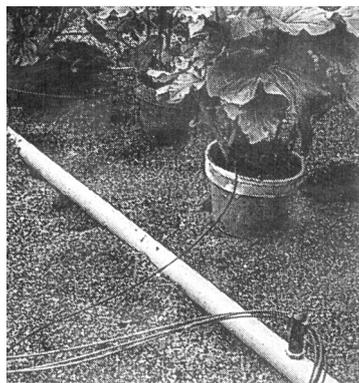


Figura 2.10 – Fotografia do sistema de rega gota-a-gota.

#### Equipamentos vulgarmente utilizados

Os equipamentos vulgarmente utilizados neste tipo de sistemas de rega são:

- Grupo moto-bomba para fornecer água sob pressão;
- Cabeça de controlo composta por válvula de fecho, com ou sem contador, manómetro, filtro e ligações ao tanque e saída para as tubagens;
- Rede de tubagem composta por um elemento primário do qual deriva o secundário, constituído por tubos de rega;
- Gotejadores os quais são fabricados em material de plástico ou em aço, variando o caudal entre 3 e 8 litros por hora [5].

A Figura 2.11 ilustra os constituintes normalmente utilizados num sistema de rega gota-a-gota.

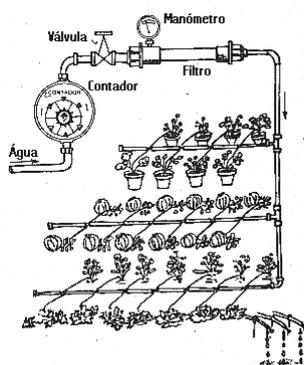


Figura 2.11 – Sistema completo de rega gota-a-gota.

Quando automatizado o controlo do sistema é do tipo liga/desliga sendo possível utilizar relés para o comando das electroválvulas e do grupo moto-bomba.

### Vantagens e inconvenientes

Algumas das vantagens oferecidas pela rega gota-a-gota são: economia de água e de mão-de-obra; melhor controlo da água no solo e no ambiente da estufa; permite automação e programação de regas

No entanto, este tipo de rega também apresenta algumas desvantagens nomeadamente: o risco de obstrução dos “gotejadores” e o facto de requerer um conhecimento maior, dos requisitos de rega, por parte do produtor.

### 2.3.4 Humidade

A água está presente no ar sob a forma de vapor. Existem duas formas de exprimir a quantidade de água no ar [10]:

- **Humidade absoluta:** é a quantidade de vapor de água contida no ar. Exprime-se em g por Kg de ar seco;
- **Humidade relativa ou higrometria:** refere-se à quantidade de vapor de água contida no ar e exprime-se em percentagem da quantidade máxima de vapor de água que o ar suporta àquela temperatura.

De facto, a quantidade máxima de água que o ar pode conter varia com a temperatura: quanto mais quente estiver o ar maior quantidade de água pode conter. Assim, quando

uma massa de ar arrefece, a sua humidade relativa aumenta mas a sua humidade absoluta não varia.

## **Influência da humidade sobre as plantas**

Um excesso de humidade relativa no ar pode limitar a respiração, a evapotranspiração, a circulação da seiva e como consequência o desenvolvimento das culturas. Uma higrometria excessiva aumenta os riscos de condensação sobre as plantas e em toda a cobertura da estufa, isto favorece o aparecimento de moléstias.

É difícil especificar o nível de higrometria mas em geral uma humidade relativa com valores compreendidos entre 85 a 90% é considerada nefasta para a maioria das culturas em produção [10].

Por outro lado, quando a humidade relativa diminui consideravelmente, as trocas gasosas abrandam, diminuindo a actividade da fotossíntese das plantas. Este fenómeno produz-se sobretudo a meio do dia, quando a temperatura é máxima.

Torna-se difícil controlar a higrometria numa estufa uma vez que está relacionada com a actividade fisiológica das plantas, do clima exterior, etc.

Devido à dificuldade em controlar a humidade, procuram evitar-se as variações bruscas entre valores extremos.

Para um bom desenvolvimento das plantas, a humidade relativa óptima do ar varia conforme a espécie e o seu estado vegetativo. Do ponto de vista prático a regulação da humidade relativa na estufa é baseada, na maioria das vezes, na experiência pessoal do produtor e não tomando como base resultado de estudos.

### **2.3.4.1 REGULAÇÃO DA HUMIDADE DO AR**

#### **2.3.4.1.1 Nebulização**

##### **Princípio**

A nebulização consiste em pulverizar o ar com finas gotículas de água. Estas devem manter-se suspensas no ar o maior tempo possível sem ferir os órgãos vegetais.

##### **Utilização**

A nebulização aplica-se sobretudo em multiplicação:

- Por sementeira;
- Por estaca;
- Por replantação.

A nebulização limita as perdas de água das plantas jovens. Permite uma aceleração e um aumento do desenvolvimento, encurtando o tempo de enraizamento.

A nebulização também se aplica em produção para refrescar o ar ambiente e limitar a transpiração durante os períodos quentes.

## Técnicas

### Sistema de nebulização por injector

Utilizando a nebulização por injector a superfície de acção do jacto de água fica bastante próxima do orifício. A água deve chegar a uma pressão elevada (4 a 5  $\text{Kg/cm}^2$ ) para que as gotas sejam suficientemente finas. Se a água for fornecida ao produtor a uma pressão inferior, este terá de aumentá-la recorrendo a uma electrobomba a pressão de ar. Este sistema de nebulização permite obter gotas com dimensão de 100  $\mu\text{m}$  aproximadamente [10]. A Figura 2.12 mostra a utilização de um sistema deste tipo.

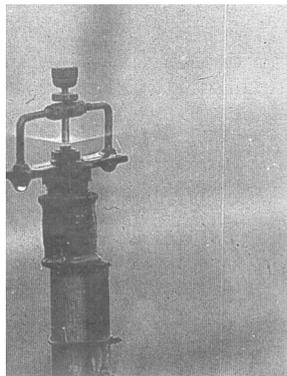


Figura 2.12 – Fotografia de nebulização por injector.

### Sistema de nebulização “Fog System”

Recorrendo a esta técnica, a água a alta pressão (40 a 100  $\text{Kg/cm}^2$ ) é forçada a atravessar um orifício de diâmetro microscópico. O jacto é imediatamente quebrado sobre uma agulha. Os diâmetros das pequenas gotas são de aproximadamente 10  $\mu\text{m}$  para uma pressão de 40  $\text{Kg/cm}^2$ . As pequenas gotas mantêm-se suspensas no ar durante algum tempo.

## Equipamentos

Este tipo de sistema é composto essencialmente por um grupo moto-bomba que pode ser eléctrico, a diesel ou a gasolina, para garantir a pressão da água necessária (quanto mais elevada, mais fina é a nebulização), por nebulizadores (saídas) e pelas tubagens [5]. Engloba ainda válvulas que controlam o fluxo da água, as quais podem habilitar ou inibir a nebulização, podendo estas ser de comando manual, hidráulico ou eléctrico, sendo as últimas as de maior interesse neste caso. Na Figura 2.13 é ilustrada a composição deste tipo de sistemas.

Também neste caso o controlo é do tipo liga/desliga podendo ser utilizados contactores ou relés para o efeito.

### Legenda

- 1 –Quadro eléctrico
  - 2 –Sonda para controlo de presença de água
  - 3- Pré-filtro
  - 4- Filtro
  - 5- Electroválvula
  - 6- Motor eléctrico
  - 7- Bomba a alta pressão
  - 8- Manómetro
  - 9- Regulador de pressão
  - 10- Suporte anti-vibratório
  - 11 –Ligação à canalização
  - 12- Válvula manual
  - 13- Difusor
- a) Nebulizador
  - b) Filtro
  - c) Suporte
  - d) Junta toroidal

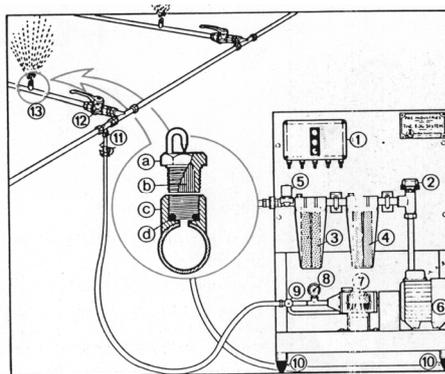


Figura 2.13 – Sistema de nebulização - “Fog System”.

## Vantagens e inconvenientes

Como vantagens o sistema de nebulização permite:

- Acelerar e aumentar a formação de raízes das jovens estacas;
- Limitar as perdas de água por transpiração;
- Limitar as temperaturas excessivas: as pequenas gotas ao evaporarem-se diminuem a temperatura ambiente.

No entanto, como qualquer técnica, também esta apresenta algumas características com as quais é necessário ter atenção:

- É mais fácil aumentar a humidade do que diminuí-la. A utilização de nebulização pressupõem ter capacidade de evacuar humidade excedentária quando esta não é necessária (sobretudo durante o período nocturno);
- As gotículas de maior dimensão podem ferir as pétalas de determinadas espécies de flores, o que faz com que exista uma diminuição do produto quando comercializado. É necessário verificar a pressão da água, principalmente nos sistemas que utilizam a técnica de injectar.

### 2.3.4.1.2 Desumidificação

#### Princípio

O ar carregado de humidade passa na vizinhança de um ponto frio, o *evaporador*. Uma parte do vapor de água que ele contém condensa-se. O ar arrefecido passa por uma bateria quente, o condensador, saindo quente e seco, misturando-se então com o ar ambiente.

#### Utilização

Uma bomba a calor (BAC) de desumidificação, Figura 2.14, pode ser utilizada quando o grau de higrometria é excessivo, sobretudo de manhã. Ela é particularmente

adaptada às estufas de vidro duplo ou de paredes revestidas, concebidas para limitar os desperdícios de energia. Devido ao isolamento deste tipo de estufas, uma condensação significativa corre o risco de se desenvolver sobre a face interna da parede e de cair sobre a forma de gotículas sobre o solo. Por outro lado, numa estufa de parede dupla as perdas são menores, limitando assim a renovação do ar.



Figura 2.14 – Fotografia de um desumidificador.

## Técnicas

Uma BAC de desumidificação é composta por [10]:

- Uma fonte fria sobre a qual a água se condensa;
- Uma fonte quente para aquecer o ar;
- Um ventilador para forçar o ar entre as baterias;
- Um compressor para assegurar o calor e o frio.

O ar seco e aquecido é ventilado na estufa. O arranjo dos componentes de uma BAC está ilustrado na Figura 2.15.

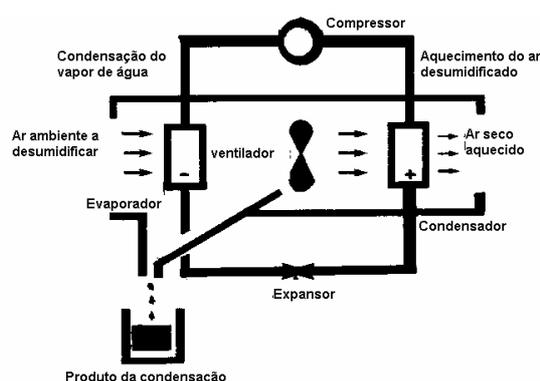


Figura 2.15 - Sistema de desumidificação.

A BAC é instalada por cima das culturas (fixa na estrutura) para assegurar uma boa circulação do ar ao evaporador.

A bomba de calor de desumidificação funciona essencialmente durante o período nocturno. Com efeito, durante o dia a ventilação e elevação das temperaturas pelos raios solares, contribuem para manter o estado higrométrico aceitável [10]. As BAC propostas pelos construtores têm uma potência compreendida entre 2 e 3 Kw. São

bombas de calor ar-ar, cujo débito é aproximadamente de 1500 m<sup>3</sup>/h. Estas permitem tratar entre 1000 a 1500 m<sup>2</sup> de estufa. Nestas condições elas condensam entre 5 a 6 litros de água por hora.

A humidade relativa pode ser mantida em valores próximos de 85%.

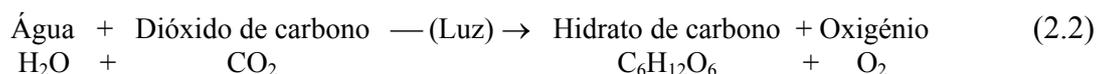
### **Vantagens e inconvenientes**

As BACs de desumidificação permitem diminuir a humidade relativa numa estufa de 1 a 3 %, e mantê-la a um nível aceitável para as culturas (80 a 85 %). Foram obtidos bons resultados sobre culturas de rosas em vasos [10].

As BAC de desumidificação representam um investimento que é preciso rentabilizar através de uma economia de energia, sobretudo por um ganho de qualidade da cultura e pela redução dos problemas fitossanitários.

### **2.3.5 Luminosidade**

A fotossíntese é um processo no qual as plantas utilizam a energia da luz para absorver dióxido de carbono da atmosfera e convertê-lo em energia química sob a forma de açúcar responsável pela sobrevivência e crescimento das mesmas. A fotossíntese é traduzida pela Equação 2.2.



A radiação activa para a fotossíntese é composta pelos raios compreendidos entre 400 e 700 nm (o que corresponde à gama visível do espectro) [10]. Para cada espécie existe um mínimo abaixo do qual a fotossíntese não é possível. Quando existe pouca ou nenhuma claridade, a respiração é predominante e a planta recorre às suas reservas. Por outro lado, quando existe um excesso de luminosidade a planta desenvolve-se. No entanto, acima de determinado patamar a intensidade luminosa não estimula mais a fotossíntese tendo que recorrer-se a outros factores, como por exemplo a concentração de CO<sub>2</sub> e irrigação.

O processo de fotossíntese pode também ocorrer em ambientes interiores, desde que seja providenciada uma fonte de luz artificial que forneça o espectro, intensidade e duração necessários à sua realização, tornando possível a utilização da estufa durante todo o ano.

Os principais factores que influenciam a intensidade luminosa numa estufa são:

- A posição do sol;
- O estado do tempo;
- A cobertura da estufa

#### **2.3.5.1 ILUMINAÇÃO**

Com o objectivo de conseguir o máximo rendimento num sistema de controlo ambiental agrícola, é necessário ter no interior da estufa um espectro e intensidade

semelhante ao da luz solar e desta forma aumentar a radiação útil à fotossíntese. Isto é conseguido através de lâmpadas próprias para o efeito e equipamentos que permitam medir a intensidade luminosa como são os casos do piranómetro e luxímetro. Estas lâmpadas, em conjunto com luminárias convenientemente projectadas, permitem a reflexão de luz para as plantas, a qual deve ser aproveitada ao máximo.

### 2.3.5.1.1 Lâmpadas fluorescentes

Antes de surgirem as luzes de descarga de alta intensidade, *High Intensity Discharge* (HID), os produtores de interior dependiam principalmente de luzes fluorescentes para conseguir os melhores resultados. Estas eram económicas, razoavelmente eficientes em termos energéticos, e a maioria delas emitiam um espectro suficiente para o crescimento das plantas. Existe uma grande variedade de lâmpadas disponíveis [17], que são classificadas por potência, comprimento e gama do espectro de cores. As fixações destas lâmpadas são geralmente compostas pelo suporte das lâmpadas, balastro e reflector. A Figura 2.16 mostra a utilização de lâmpadas fluorescentes em estufas. Com o surgimento das lâmpadas HID as lâmpadas fluorescentes são utilizadas usualmente para propagação (sementes) e em etapas iniciais do crescimento de vegetais. As luzes HID, mais eficientes e intensas, são actualmente a escolha preferida para o crescimento de plantas que requeiram elevada luminosidade.

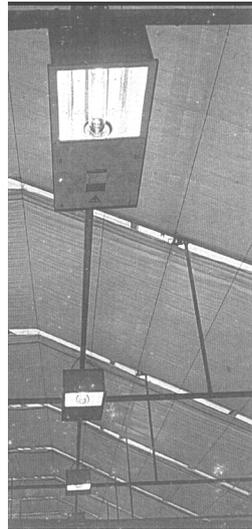


Figura 2.16 – Fotografia de estufa usando lâmpadas fluorescentes.

### 2.3.5.1.2 Luzes de crescimento HID

Existem vários tipos de lâmpadas HID [17]. Um dos tipos é constituído por metal *halide*, sendo estas desenvolvidas para fornecer um espectro o mais próximo possível da luz solar natural. Isto, conjugado com a sua eficiência energética e intensidade, torna-as ideais para crescimento de plantas em recintos fechados. A gama dos tipos de lâmpadas vai de 100 a 1000 watt sendo 400 e 1000 watt as mais utilizadas.

Outro tipo de lâmpadas HID são as de vapor de sódio a alta pressão, que, embora mais económicas, possuem um espectro mais reduzido, o que faz com que sejam utilizadas como um suplemento às lâmpadas de *metal-halide*. Estas lâmpadas têm um tempo de vida considerável (cerca de 12 000 horas) [10] e são por vezes utilizadas como suplemento e/ou prolongamento à luz solar, natural, em estufas. A Figura 2.17 mostra a aplicação de lâmpadas de vapor de sódio a alta pressão em estufas.

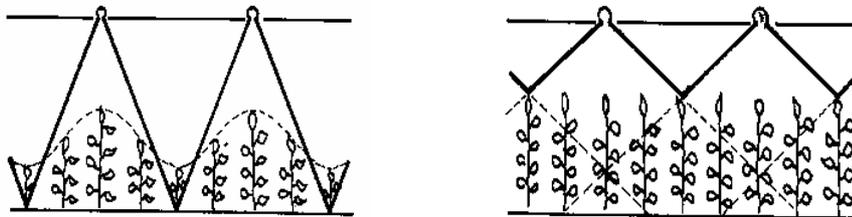


**Figura 2.17 – Fotografia de lâmpadas vapor de sódio a alta pressão em estufas.**

### **A instalação**

Na parte superior da estufa as luminárias são fixas à estrutura da estufa com a ajuda de correntes, as quais permitem regular a altura dos projectores.

A instalação deve permitir uma iluminação uniforme. Esta condição é indispensável para se conseguir uma colheita homogénea e de alta qualidade. Para tal é necessário encontrar a altura ideal dos projectores a qual depende do seu ângulo de reflexão (Figura 2.18). Existem no mercado diversos tipos de projectores em função da altura da instalação acima das culturas.



**Figura 2.18 – Instalação para distribuição uniforme de luminosidade.**

Com a ajuda de piranómetros é possível obter uma estimativa da iluminação, interior e exterior, presente em determinada estufa e de acordo com os valores lidos actuar no sentido de efectuar as correcções consideradas oportunas.

Mais uma vez o tipo de controlo geralmente aplicado neste tipo de acção, ou seja, controlo da iluminação, é do tipo liga/desliga.

### **Vantagens e inconvenientes**

O recurso a iluminação adicional permite as seguintes vantagens:

- Crescimento mais rápido da planta;
- Melhoria da qualidade e do rendimento da colheita;
- Produção programada e ajustada ao calendário de entregas;

- Programação das colheitas e uma melhor atribuição dos trabalhos durante o ano;
- Possibilidade de utilizar a estufa durante todo o ano.

Por outro lado a utilização destas técnicas deve ser alvo de um estudo prévio de viabilidade uma vez que são dispendiosas. Para que sejam rentáveis, o produtor deverá otimizar outros factores de crescimento da planta tais como: concentração de dióxido de carbono, irrigação e temperatura.

### **2.3.6 Controlo do teor de dióxido de carbono (CO<sub>2</sub>)**

Já foi referido que factores como a água, luz e temperatura são controláveis por forma a obter o crescimento óptimo. É também possível controlar a concentração de dióxido de carbono em estufas.

O dióxido de carbono contribui para o crescimento das plantas como elemento integrante dum processo natural designado por fotossíntese [18]. Quando o fornecimento de dióxido de carbono é cortado, ou reduzido, o desenvolvimento das plantas diminui.

A concentração normal de dióxido de carbono na atmosfera é aproximadamente de 0,03 % (300 partes por milhão (ppm)) [19]. Se for aumentada a concentração de CO<sub>2</sub> na estufa, quando em simultâneo se verificarem condições favoráveis de temperatura e de luminosidade, consegue-se um maior desenvolvimento das plantas.

A injeção artificial de CO<sub>2</sub> pode ser feita por dois processos. Num utilizam-se os gases resultantes da combustão do gás propano liquefeito, em dispositivos queimadores/difusores que espalham de forma uniforme o gás pelo recinto da estufa, no outro recorre-se à evaporação de dióxido de carbono puro, retido em garrafas[5].

O gás propano liquefeito produz 1500 litros de dióxido de carbono por cada quilograma desse combustível [5, 19]. Daí que seja mais utilizado o primeiro caso pela economia que representa; a segunda opção é bastante cara e de distribuição e manejo mais complicados.

Para distribuir dióxido de carbono pela estufa, podem aproveitar-se as condutas de aquecimento e/ou ventilação ou empregar-se tubos de polietileno perfurados que atravessem as estufas. A saída do gás é regulada pelo accionamento de electroválvulas e, mediante o emprego de ventiladores, força-se o dióxido de carbono e o ar a circular pelos tubos de modo a obter-se uma distribuição uniforme pela estufa.

Mais uma vez o tipo de controlo envolvido neste tipo de acção, ou seja, injeção de CO<sub>2</sub> e ventilação, é geralmente do tipo liga/desliga, utilizando-se: electroválvulas, ventiladores, reservatório de CO<sub>2</sub>, regulador de pressão do reservatório, medidor de fluxo, canalização, etc. A injeção de CO<sub>2</sub>, dependendo do plano de cultura, pode ser em períodos específicos do dia como por exemplo após o nascer do sol.

## **2.4 RESUMO**

Na maioria dos accionamentos em estufas agrícolas utilizando motores eléctricos, em geral estes são motores de indução com rotor em gaiola de esquilo, devido à sua

robustez, baixo custo e ao valor elevado de relação potência/peso. O facto de não possuir colector com escovas ou outro tipo de contactos confere-lhe uma robustez intrínseca, reduzindo a exigência de manutenção, o que torna este tipo de motores bastante populares em estufas agrícolas. Geralmente estes motores são monofásicos uma vez que na maioria dos casos a potência requerida não é elevada (inferior a 2 HP) e também porque a exigência de cablagem é menor do que a necessária numa solução trifásica.

Na Tabela 2.1 são indicados os tipos de actuação bem como alguns actuadores e tipos de controlo utilizáveis em estufas agrícolas.

<b>Tipo de actuação</b>	<b>Dispositivos actuadores</b>	<b>Tipos de comando</b>		<b>Tipo de controlo</b>
Iluminação	Lâmpadas, ecrãs de sombreamento automatizados.	Contactores / relés Fins de curso		Liga/desliga
Aquecimento	Electroválvulas, aquecedores, Caldeiras, permutadores.	Contactores / relés		Liga/desliga
Arrefecimento	Electroválvulas, ecrãs de sombreamento automatizados.	Contactores / relés Fins de curso		Liga/desliga
	Ventiladores	Variador de velocidade	Por variação de tensão e frequência.	Ajustável
			Por variação de tensão.	
		Contactores / relés		Liga/desliga
CO <sub>2</sub>	Electroválvulas, electrobombas.	Contactores / relés		Liga/desliga
Rega	Electroválvulas, electrobombas.	Contactores / relés		Liga/desliga

**Tabela 2.1 – Resumo das actuações e respectivo controlo normalmente utilizados em estufas agrícolas.**

## 3 PROTOCOLO DE COMUNICAÇÕES CAN

### 3.1 SISTEMAS EM REDE *vs* SISTEMAS CENTRALIZADOS

Com o aumento da utilização de sistemas electrónicos compostos por vários microcontroladores e periféricos inteligentes, como é o caso do controlo em veículos automóveis, ganharam bastante importância protocolos de rede entre estas unidades em relação a sistemas que utilizam um processador central responsável pelas tarefas de controlo.

A escolha por uma solução em rede permite um conjunto de benefícios, em relação a uma solução centralizada, tais como [20]:

- Menos fios no sistema o que torna a cablagem mais simples e barata;
- Ligações curtas a sensores analógicos sensíveis a ruído eléctrico, antes dos sinais serem convertidos em mensagens digitais “imunes” a ruído;
- Sistema flexível. Unidades funcionais podem ser adicionadas ou removidas de forma simples;
- Manutenção do sistema. A electrónica de várias unidades pode ser idêntica, permitindo troca entre elas;
- Cada unidade pode ser desenvolvida e testada individualmente de acordo com requisitos exigidos pelo sistema;
- Partilha de dados entre as várias unidades pode eliminar redundância de informação.

Após apresentar os motivos que levam à adopção de uma arquitectura em rede, serão abordadas, no decorrer das restantes secções deste capítulo, as principais propriedades do protocolo de comunicações em rede que serviu de base ao sistema implementado, o protocolo CAN.

### 3.2 INTRODUÇÃO

#### 3.2.1 Um pouco de história

O protocolo CAN foi desenvolvido por Robert Bosch em 1986 para aplicação na indústria automóvel, com o objectivo de simplificar os complexos sistemas de fios em veículos com sistemas de controlo compostos por múltiplos microcontroladores/micro-computadores para gestão do motor, sistema ABS, controlo da suspensão, etc. [21]. A sua especificação base anunciava elevada taxa de transmissão, grande imunidade a interferências eléctricas e capacidade de detectar erros [22].

A aplicação da tecnologia CAN, para partilha de dados e controlo em tempo real, tem vindo a tornar-se cada vez mais popular. Ao longo dos anos, o CAN evoluiu de aplicações dedicadas à indústria automóvel para outras de uso industrial e produtos envolvendo microcontroladores, com ligação por fio, e não só. Um dos benefícios do

CAN é suprimir a necessidade de sistemas complexos de fios substituindo-os por um simples cabo. É também considerado uma solução para implementar comunicação em rede de uma forma simples, barata e robusta, nomeadamente ao ruído electromagnético [23].

A utilização da electrónica, na melhoria da eficiência e funcionalidade dos sistemas actuais tornou, a multiplexagem uma necessidade para além de uma realidade, atingindo um nível onde a técnica de ligação ponto-a-ponto, tradicional, não consegue competir, economicamente, com o crescimento dos sistemas electrónicos. Actualmente utilizam-se sistemas interactivos onde a informação é partilhada através de um barramento comum. Uma vez que estas ligações aumentaram, a necessidade de interfaces série e de um protocolo de barramento aberto adquiriram bastante importância. O protocolo de comunicações CAN, descreve o método como a informação é transferida entre dispositivos e é assunto de *standards* internacionais aprovados pela *International Standard Organization* (ISO). Esta organização aprovou-o como barramento *standard* para redes de elevada velocidade ( $\geq 125$  Kbit/s) na partilha de informação em tempo real em veículos automóveis [4] e para taxas de transmissão menores ( $\leq 125$  Kbit/s) [24], estando em conformidade com o modelo de referência de sete camadas *Open Systems Interconnection* (OSI) para os dois níveis inferiores do mesmo: o nível de ligação de dados (nível 2) e o nível físico (nível 1).

Com a ISO, o CAN foi adoptado pela indústria automóvel bem como por outros tipos de indústrias, devido à sua robustez e flexibilidade. A disponibilidade de circuitos integrados colocados no mercado por vários fabricantes, de semicondutores, encoraja a sua utilização devido ao seu baixo custo.

### 3.2.2 Características do CAN

O CAN é um protocolo de comunicações série, que permite controlo distribuído em tempo real, com elevado nível de segurança [4]. É um sistema em barramento com capacidades multi-mestre, isto é, vários nós podem pedir acesso ao meio de transmissão em simultâneo. Este protocolo comporta também o conceito de *multicast*, isto é, permite que uma mensagem seja transmitida a um conjunto de receptores simultaneamente.

Nas redes CAN não existe o endereçamento dos destinatários no sentido convencional, em vez disso são transmitidas mensagens que possuem um determinado identificador. Assim, um emissor envia uma mensagem a todos os nós CAN e cada um por seu lado decide, com base no identificador recebido, se deve ou não processar a mensagem. O identificador determina também a prioridade intrínseca da mensagem, ao competir com outras pelo acesso ao barramento.

O CAN é considerado um sistema de barramento série, bom para ligar em rede subsistemas inteligentes, tais como sensores e actuadores. A informação transmitida possui tamanho curto. Assim, cada mensagem CAN pode conter um máximo de 8 *bytes* de informação útil, sendo no entanto possível transmitir blocos maiores de dados recorrendo a segmentação.

A taxa máxima de transmissão especificada é de 1 Mbit/s, correspondendo este valor a sistemas com comprimento de barramento até 40 m [4]. Para distâncias superiores a taxa de transmissão, recomendada, diminui. Alguns dos valores recomendados [25] são: 50 Kbit/s para distâncias até 1 Km e 125 Kbit/s para distâncias até 500 m. Se a distância

do barramento for superior a 1 Km pode ser necessária a utilização de dispositivos repetidores (*repeater*) ou ponte (*bridge*).

O número de elementos num sistema CAN está, teoricamente, limitado pelo número possível de identificadores diferentes. Este número limite é no entanto significativamente reduzido por limitações físicas do *hardware*. Existem no mercado de integrados *transceivers* que permitem ligar pelo menos 110 nós [26]. Refira-se também, neste contexto, que com módulos de entrada/saída (E/S) adequados é possível ter diversos sensores e actuadores por nó.

O CAN permite flexibilidade uma vez que podem ser adicionados novos nós a uma rede CAN sem requerer alterações do *software* ou *hardware* dos restantes nós, se o novo nó não for emissor, ou se o novo nó não necessitar da transmissão de dados adicionais [4].

Outra característica importante é o facto de o controlador CAN de cada estação, registar os erros, avaliando-os estatisticamente, por forma a desencadear acções com eles relacionadas. Estas acções podem corresponder ao desligar, ou não, da estação que provoca os erros, tornando este protocolo eficaz em ambientes ruidosos.

Utilizando o protocolo CAN, a ligação entre o nível físico (nível 1 OSI) e o de aplicação (nível 7 OSI) é feita utilizando vários protocolos emergentes ou através de *software* desenvolvido pelo utilizador [22]. O exemplo de um protocolo, baseado em CAN, *standard* industrial ao nível de aplicação é o DEVICenet<sup>TM</sup> da Allen-Bradley o qual é utilizado para ligar em rede controladores lógicos programáveis e sensores inteligentes.

Em resumo, o grande interesse pelo CAN por parte dos círculos da engenharia de automação industrial reside em diversas das suas características, nomeadamente as seguintes [27]:

- Ser um *standard* ISO;
- Considerável imunidade ao ruído;
- Capacidade multi-mestre;
- Capacidade *multicast*;
- Capacidade eficaz de detectar e sinalizar erros;
- Simplicidade;
- Retransmissão automática de mensagens “em espera” logo que o barramento esteja livre;
- Reduzido tempo de latência;
- Atribuição de prioridade às mensagens;
- Flexibilidade de configuração;
- Distinção entre erros temporários e erros permanentes dos nós;
- Elevadas taxas de transferência (1 Mbit/s);
- Redução de cabo a utilizar;
- Baixo preço;
- *Hardware standard*.

Nas próximas secções deste capítulo serão abordadas algumas regras que permitem ao protocolo CAN possuir as características anteriormente referidas e que o tornam actualmente num dos protocolos de comunicações com maior aceitação.

### 3.3 MÉTODO DE ENDEREÇAMENTO

Quando são transmitidos dados utilizando o CAN, não existe endereço fonte ou destino numa mensagem. Os identificadores únicos, das mensagens, servem para caracterizar o conteúdo da mensagem (ex. rpm ou temperatura do motor, no caso do controlo de veículos) sendo da competência de cada nó da rede decidir se a mensagem é ou não válida, para esse nó, realizando para isso um teste de aceitação ao identificador da mesma [28]. Este teste é designado por filtragem, “*Frame acceptance filtering*” [4], existindo vários dispositivos controladores que permitem diferentes níveis de sofisticação desta filtragem.

Outra característica importante do identificador, para além de definir o conteúdo da mensagem, é a de este estabelecer, também, a prioridade da mensagem. Isto é importante para a atribuição do barramento quando várias estações competem pelo acesso ao barramento.

O processo de transmissão e recepção de mensagens CAN, ilustrado na Figura 3.1, consiste no seguinte: se a unidade central de processamento, *Central Processing Unit* (CPU), de um dos nós desejar enviar uma mensagem para um ou mais nós da rede, este “passa” os dados a serem transmitidos e o respectivo identificador para o controlador CAN (“Preparar“) desse nó. Isto é tudo o que o CPU necessita realizar para iniciar a transferência de informação. A mensagem é então composta e transmitida pelo controlador CAN. Logo que o controlador consiga acesso ao barramento (“Enviar mensagem“) todos os outros nós na rede CAN tornam-se receptores da mesma (“Receber mensagem“). Cada estação na rede CAN, tendo recebido correctamente uma mensagem, realiza um teste de aceitação para determinar se os dados recebidos são ou não relevantes para essa estação (“Seleccção“). Se os dados tiverem significado são processados (“Aceite“), caso contrário são rejeitados (“Não Aceite“).

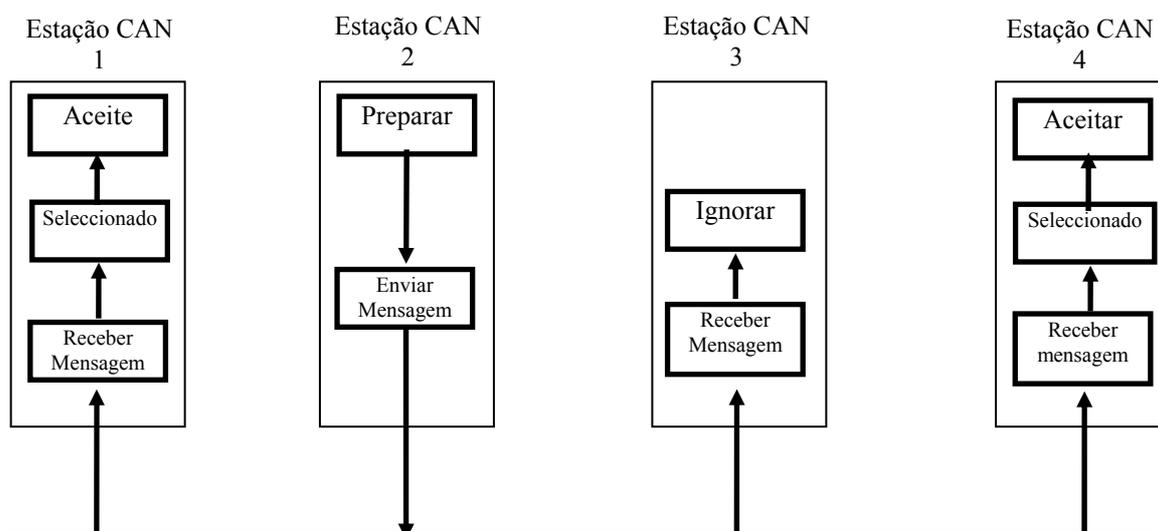


Figura 3.1 - Método de endereçamento.

É atingido, desta forma, um nível superior de flexibilidade do sistema e de configuração devido ao esquema de endereçamento orientado à contenção. É fácil adicionar estações

à rede CAN existente sem necessidade de alterações de *hardware* ou *software* dos nós existentes, desde que as novas estações sejam apenas receptoras. Uma vez que o protocolo de transmissão de dados não requer endereços físicos para os componentes individuais, suporta o conceito de electrónica modular e ao mesmo tempo permite recepção múltipla (*multicast, broadcast*) e sincronização de processos distribuídos. Medições, necessárias como informação por vários microcontroladores, podem ser transmitidas pela rede, de tal forma que não é necessário que cada microcontrolador possua o seu sensor, evitando redundância de sensores no caso de redes de aquisição.

### 3.4 PROCESSO DE ARBITRAGEM NÃO DESTRUTIVA

Para que os dados sejam processados em tempo real estes devem ser transferidos rapidamente. Isto exige um meio físico que permita elevada taxa de transmissão e chamadas rápidas à alocação do barramento quando várias estações tentam transmitir simultaneamente. No processamento em tempo real a urgência da troca de mensagens pela rede pode ser significativamente diferente: uma grandeza (ex. carga do motor) que varie rapidamente deve ser transmitida com maior frequência e menores atrasos do que outras que variem menos (ex. temperatura do motor).

No CAN, a prioridade com que uma mensagem é transmitida relativamente a outra é especificada pelo identificador da respectiva mensagem. A prioridade das mensagens é definida durante a fase de projecto do sistema sob a forma de valores binários. Para esta definição de prioridades é considerado que o identificador de menor valor numérico detém maior prioridade [22].

Por outro lado, a singularidade dos identificadores dos objectos de comunicação, mensagens, também é utilizada para arbitrar os pedidos de acesso ao barramento por parte dos nós concorrentes. O CAN é uma rede *Carrier Sense Multi-Acess with Deterministic Collision Resolution* (CSMA/DCR) [29], ou seja: os nós atrasam a transmissão se o barramento estiver ocupado; quando a condição de barramento livre for detectada, qualquer nó pode iniciar a transmissão; os conflitos de acesso ao barramento são solucionados por comparação orientada ao bit, *bitwise*, dos identificadores afectos aos objectos de comunicação, mensagens, e funciona da seguinte forma:

- Enquanto transmite o identificador do objecto de comunicação, cada nó monitora o barramento série;
- Se o *bit* transmitido for “recessivo”, nível lógico ‘1’, e for monitorizado um *bit* “dominante”, nível lógico ‘0’, o nó desiste da transmissão e inicia a recepção dos dados que chegam;
- O nó que transmite o objecto com o menor identificador, ganha acesso ao barramento e continua a transmissão.

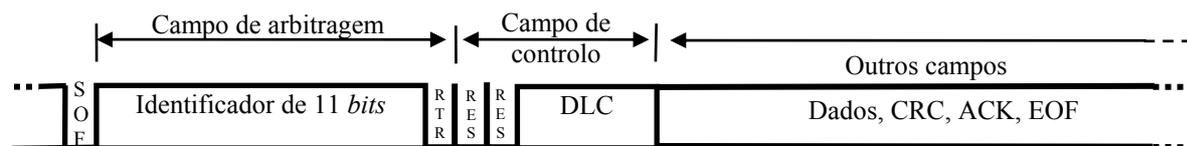
Isto significa que: a arbitragem é não destrutiva, uma vez que a transmissão do objecto de menor identificador não sofre atraso; o acesso ao barramento obedece a prioridades, permitindo que a informação mais urgente seja atendida em primeiro lugar. A retransmissão automática de um objecto de comunicação é tentada após uma perda no processo de arbitragem.

### 3.5 FORMATO DAS FRAMES

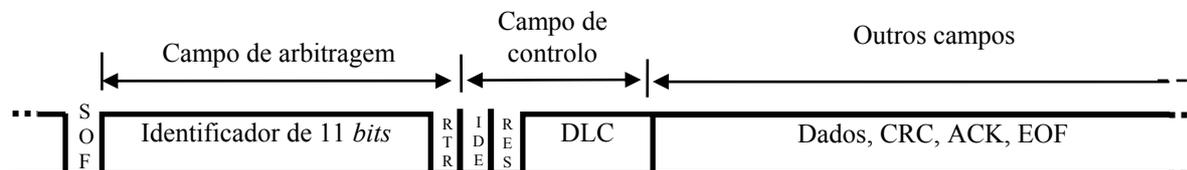
Originalmente as *frames* do protocolo CAN possuíam um identificador de 11 *bits*. Com a especificação 2.0 [27] foi definido um novo formato, o qual se baseia num identificador de 29 *bits* de comprimento, aumentando consideravelmente o número possível de identificadores únicos. Este novo formato é opcional, o que significa que as mensagens com identificadores de 11 *bits* continuarão a ser as mensagens *standard*. De acordo com a especificação 2.0 as *frames* com 11 *bits* identificadores designam-se por *frames standard* e as *frames* com 29 *bits* identificadores designam-se por *frames* estendidas.

Seguidamente serão abordadas as diferenças, mais significativas, entre os diversos formatos actualmente conhecidos [30].

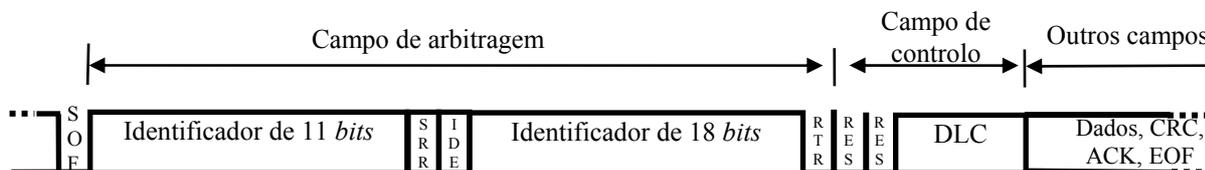
A Figura 3.2 ilustra os diferentes tipos de formato possíveis, onde se pode verificar que em todos os formatos, as mensagens começam com o *bit* de início de *frame*, *start of frame* (SOF), seguido dos *bits* do identificador. Com o identificador existem um ou três *bits* de controlo (RTR, SRR e IDE), no campo de arbitragem. Estes *bits* definem quando se trata de uma *frame standard* ou estendida e quando se trata de uma *frame* de dados ou remota.



(a) Formato dum *frame* de dados segundo as versões 1.0, 1.1, 1.2 e 2.0A.



(b) Formato dum *frame* de dados segundo a versão 2.0B (Formato *standard*).



(c) Formato dum *frame* de dados segundo a versão 2.0B (Formato estendido).

**Figura 3.2 – Formato dos diferentes tipos de *frames*.**

A Figura 3.2-a) ilustra uma *frame* de dados, de acordo com as especificações 1.0, 1.1, 1.2 e 2.0A do protocolo CAN. Totalmente compatível com aquele formato é o formato *standard* conforme é definido na especificação 2.0B, Figura 3.2-b).

Em contraste com o referido anteriormente, a Figura 3.2-c) representa o formato estendido, definido na versão 2.0 B, o qual diverge dos formatos anteriores no número

de *bits* do campo de arbitragem e valor dos *bits* de controlo.

O significado dos três *bits* de controlo é o seguinte:

- O *bit Remote Transmit Request* (RTR), distingue entre *frames* de dados e remotas. Em *frames* de dados este *bit* é “dominante”, sendo “recessivo” em *frames* remotas.
- O *bit Substitute Remote Request* (SRR), é um *bit* “recessivo”. Este *bit* é transmitido em *frames* no formato estendido na posição que o *bit* RTR ocupa em *frames standard*.
- O *bit Identifier Extension* (IDE), faz a distinção entre *frames standard* e estendidas. Em *frames standard* este *bit* é “dominante”, enquanto que em *frames* estendidas é “recessivo”.

Assim, num sistema onde vários nós iniciem a transmissão simultânea de *frames* com o mesmo identificador, as seguintes regras são aplicadas: *frames* de dados possuem maior prioridade do que *frames* remotas, e *frames standard* possuem maior prioridade do que *frames* estendidas. Isto significa que uma *frame standard* remota “vence”, no processo de arbitragem, uma *frame* de dados estendida, se os 11 *bits* mais significativos do identificador forem iguais [30]. No entanto, em sistemas onde tenham que coexistir os dois formatos é necessário ter alguns cuidados em termos de implementação.

A maioria dos novos controladores respeita a versão 2.0. Assim para permitir *frames* estendidas um controlador pode ser: CAN 2.0B passivo ou CAN 2.0B activo. Se for passivo, ignora *frames* estendidas ao contrário dos controladores 1.0 que transmitem *frames* de erro quando detectam *frames* com identificadores de 29 *bits*. Se for activo, o controlador permite a transmissão e recepção de *frames* estendidas. Existem portanto regras de compatibilidade que devem ser consideradas para transmitir e receber os dois tipos de *frames*:

- Controladores CAN 2.0B activos, transmitem e recebem ambos os formatos de *frames* estendidas ou *standard*;
- Controladores 2.0B passivos, transmitem e recebem *frames standard*, ignorando *frames* estendidas sem erros;
- Controladores CAN 1.0 geram erros quando detectam *frames* estendidas.

Assim, numa rede com controladores CAN 1.0 não podem existir mensagens com formato estendido obrigando a que todos os controladores utilizem o formato *standard*.

### 3.6 TRANSFERÊNCIA DE MENSAGENS

Conforme foi referido na secção anterior, as *frames* do protocolo CAN possuem dois formatos que diferem no tamanho do identificador:

- *Frames standard*, cujo identificador possui 11 *bits*;
- *Frames* estendidas, cujo identificador possui 29 *bits*.

Embora haja estes dois tipos de formato, só se irá considerar o formato de *frame standard* porque foi o utilizado nesta dissertação. A transmissão e recepção de informação, num sistema CAN são efectuadas e controladas através de quatro tipos diferentes de *frames*:

- *Frame* de Dados;
- *Frame* Remota;
- *Frame* de Erro;
- *Frame* de Sobrecarga.

### 3.6.1 *Frame* de Dados

Uma *frame* de dados, ilustrado na Figura 3.3, contém os dados do emissor para o receptor. Os sete campos que compõem este tipo de *frames* são os seguintes:

Início de <i>frame</i>	Campo de arbitragem	Campo de controlo	Campo de dados	Campo CRC	Campo ACK	Fim De <i>frame</i>
------------------------	---------------------	-------------------	----------------	-----------	-----------	---------------------

**Figura 3.3 - *Frame* de dados.**

- Início de *frame*, SOF. Este *bit* marca o início da transmissão de dados ou remotas. É um *bit* “dominante”. Todos os nós têm de se sincronizar, com a transição provocada pelo SOF do nó que iniciou primeiro a transmissão;
- Campo de arbitragem. Este campo depende do formato da *frame*, sendo no formato *standard*, constituído pelo identificador e pelo *bit* RTR. O *bit* RTR tem o valor “dominante” para *frames* de dados;
- Campo de controlo, o qual é composto por seis *bits*, sendo 2 *bits* reservados para futura expansão seguidos pelo campo *Data Length Code* (DLC). Os *bits* reservados são transmitidos como “dominante”. O campo DLC possui 4 *bits* que indicam o número de *bytes* do campo de dados;
- Campo de dados. Este campo indica o número admissível de *bytes* de dados para uma *frame* de dados varia num intervalo que pode ir de 0 a 8 [4, 27];
- Campo *Cyclic Redundancy Code* (CRC). Contém a sequência CRC seguida do delimitador CRC, o qual consiste num único *bit* “recessivo”. O cálculo da sequência CRC é feito tendo em conta uma polinomial geradora [4, 27];
- Campo *Acknowledge* (ACK), reconhecimento. Este campo é constituído por dois *bits*, o *slot* ACK e o delimitador ACK, sendo este último “recessivo”. Os nós receptores ao receberem correctamente a sequência CRC, enviam o reconhecimento substituindo o *bit* “recessivo” por um *bit* “dominante” na *slot* ACK. Como consequência do anteriormente exposto note-se que o *slot* ACK é rodeado por dois *bits* ‘recessivos’;
- Fim de *frame*, *end of frame* (EOF). Este campo serve de *flag* delimitadora das *frames* de dados ou remotas, sendo constituído por sete *bits* “recessivos”.

### 3.6.2 *Frame* Remota

Um nó que seja receptor de determinados dados, pode iniciar a transmissão dos

mesmos, através de pedido ao nó de origem, enviando uma *frame* remota, ou seja, um pedido de dados. A *frame* remota é constituída por seis campos, conforme ilustrado na Figura 3.4.

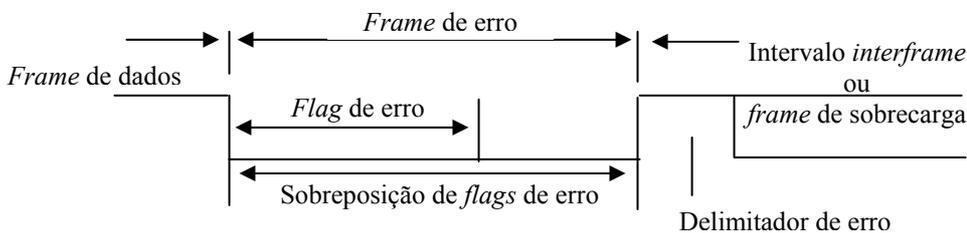
	Início de <i>frame</i>	Campo de arbitragem	Campo de controlo	Campo CRC	Campo ACK	Fim De <i>frame</i>	
--	------------------------	---------------------	-------------------	-----------	-----------	---------------------	--

**Figura 3.4 - Frame remota.**

Os campos constituintes duma *frame* remota são idênticos aos duma *frame* de dados, com a excepção do valor do *bit* RTR, do campo de arbitragem, que agora é “recessivo” e da inexistência de campo de dados. Os *bits* DLC do campo de controlo da *frame* remota devem possuir valor igual ao da *frame* de dados correspondente.

### 3.6.3 Frame de Erro

A *frame* de erro, cujo formato está ilustrado na Figura 3.5, [7], é transmitida por qualquer nó quando é detectado um erro no barramento e é constituída por dois campos distintos. O primeiro campo é dado pela sobreposição de *flags* de erro provenientes de diferentes estações. O segundo campo é o delimitador de erro.



**Figura 3.5 – Frame de erro.**

Um nó “activo-ao-erro” ao detectar um erro assinala-o pela transmissão de uma *flag* “activa-ao-erro”. O formato desta *flag* de erro viola a lei da inserção de *bits* aplicada a todos os campos desde o SOF até ao delimitador CRC. Como consequência, todos os outros nós detectam também uma condição de erro e iniciam por si a transmissão de uma *flag* de erro. A sequência de *bits* “dominantes”, monitorizada no barramento, resulta da sobreposição das diferentes *flags* de erro transmitidas por nós individuais. O comprimento total desta sequência varia entre um mínimo de seis e um máximo de doze *bits*.

Uma estação ‘passiva-ao-erro’ detectando uma condição de erro tenta sinalizar este facto através da transmissão de uma *flag* “passiva-ao-erro”. A estação “passiva-ao-erro” tem que aguardar sempre por seis *bits* subsequentes iguais após detectar uma *flag* “passiva-ao-erro”. Esta *flag* passiva ao erro é concluída quando forem detectados estes 6 *bits* iguais.

O segundo campo é o delimitador de erro, que consiste em oito *bits* “recessivos”.

Após transmissão de uma *flag* de erro, cada nó envia *bits* “recessivos” e monitoriza o barramento até detectar um *bit* “recessivo”. Após o que inicia a transmissão de mais sete *bits* “recessivos”.

### 3.6.4 *Frame* de Sobrecarga

A *frame* de sobrecarga, *overload*, contém dois campos de *bits*: *flag* de sobrecarga e delimitador de sobrecarga. O formato da *flag* de sobrecarga corresponde à da *flag* activa-ao-erro. O delimitador de sobrecarga tem formato idêntico ao do delimitador de erro [4, 27].

A *frame* de sobrecarga, ilustrado na Figura 3.6, é utilizada para provocar um atraso extra entre uma *frame* de dados ou remota e a *frame* posterior [7].

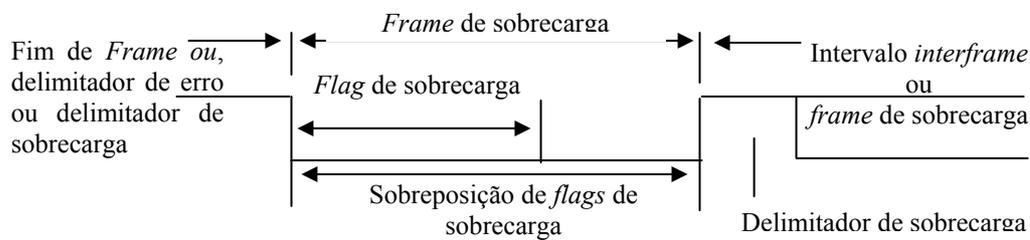


Figura 3.6 – *Frame* de sobrecarga.

A *flag* de sobrecarga consiste em seis *bits* “dominantes”. Esta *flag* destrói o formato fixo do campo de Intermissão, pelo que todos os outros nós também detectam uma condição de sobrecarga e iniciam a transmissão de uma *flag* de sobrecarga. As condições que originam a transmissão de uma *flag* de sobrecarga são [27]:

- Condições internas de um receptor, que requer um atraso da próxima *frame* de dados ou remota. Nestas condições a *frame* de sobrecarga só pode ter início durante o primeiro *bit* de uma intermissão;
- A detecção de um *bit* “dominante” durante a intermissão. Neste caso a *frame* de sobrecarga tem início depois de ser detectado o *bit* “dominante”.

Depois da transmissão de um *flag* de sobrecarga, qualquer nó “escuta” o barramento até detectar um *bit* “recessivo”.

#### O delimitador de sobrecarga

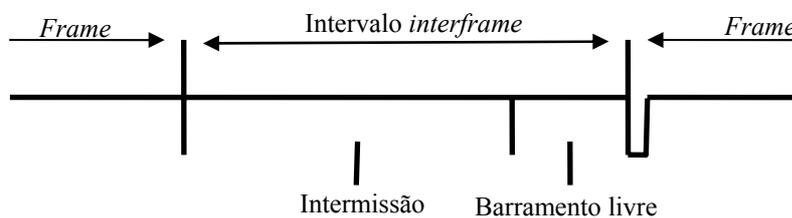
O delimitador de sobrecarga é composto por oito *bits* “recessivos”. Após a transmissão de uma *flag* de sobrecarga, todos os nós monitorizam o barramento até detectar uma transição de “dominante” para “recessivo”. Nesse instante todos os nós terminaram o envio da sua *flag* de sobrecarga e todos iniciam simultaneamente a transmissão de mais sete *bits* “recessivos” em simultâneo, para completar o delimitador de sobrecarga com o comprimento de oito *bits* [4].

### 3.6.5 Intervalo *interframe*

As *frames* de dados e as *frames* remotas são separadas de *frames* precedentes independentemente do tipo destas (dados, remota, erro ou sobrecarga), por um campo de *bits* designado por “intervalo *interframe*”, *interframe space*.

Ao contrário das anteriores, as *frames* de erro e de sobrecarga não são precedidas por um intervalo *interframe*.

O intervalo *interframe* contém os campos de Intermissão, de barramento livre, *Idle*, e ainda o campo de Suspensão de Transmissão para nós “passivos-ao-erro” que tenham sido emissores de *frames* anteriores. A Figura 3.7 e a Figura 3.8 ilustram o intervalo *interframe* para estações que não sejam “passiva-ao-erro” ou que tenham sido receptoras da mensagem anterior e para estações “passivas-ao-erro” que tenham transmitido a mensagem anterior, respectivamente.

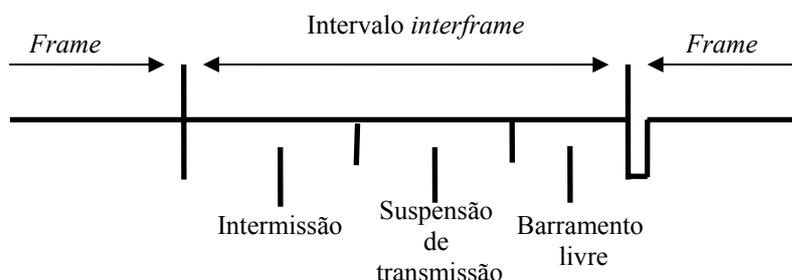


**Figura 3.7- Intervalo *interframe* para estações que não sejam “passivas-ao-erro” ou que tenham sido receptoras da mensagem anterior.**

O campo de intermissão consiste em três *bits* “recessivos”. Durante a intermissão, não é permitido a nenhum nó iniciar a transmissão de uma *frame* de dados ou remota. A única acção que pode ter é assinalar uma condição de sobrecarga.

O período de barramento livre (*idle*) pode ter um tamanho arbitrário. Uma *frame* que esteja pendente para transmissão é iniciada no primeiro *bit* posterior à intermissão. A detecção de um *bit* “dominante” durante o estado livre do barramento é interpretada como início da *frame*, SOF.

A suspensão de transmissão verifica-se quando um nó “passivo-ao-erro” após transmitir uma *frame*, enviar oito *bits* “recessivos” a seguir à intermissão, antes de lhe ser permitido iniciar a transmissão da *frame* posterior ou reconhecer o barramento como livre. Se entretanto outro nó iniciar uma transmissão, o nó tornar-se-á um receptor dessa mensagem.



**Figura 3.8 - Intervalo *interframe* para estações “passivas-ao-erro” que tenham transmitido a mensagem anterior.**

### 3.7 CODIFICAÇÃO

Os campos das *frames* de dados e remotas (SOF, arbitragem, controlo, dados e sequência CRC), são codificados pelo método da inserção do *bit* (*bit stuffing*) de comprimento cinco. Assim, sempre que um emissor detecte cinco *bits* consecutivos, de igual valor, na série de *bits* a transmitir, ele insere automaticamente um *bit* complementar na sequência de *bits* a transmitir. Os restantes campos têm forma fixa e não são sujeitos a este método. As *frames* de erro e sobrecarga possuem forma fixa pelo que também não são codificadas pela regra da inserção do *bit*. Este processo é representado na Figura 3.9.

Sequências a transmitir	100000abc	011111abc
Sequências transmitidas	1000001abc	0111110abc
a, b, c ∈ { 0, 1 }		

Figura 3.9 – Método de inserção de *bit*.

### 3.8 ORDEM DE TRANSMISSÃO DE BITS NUMA FRAME CAN

Uma *frame* é transferida campo a campo, iniciando no campo SOF. Dentro de um campo, o *bit* mais significativo é transmitido em primeiro lugar, conforme ilustrado na Figura 3.10, para uma *frame* de dados no formato *standard*.

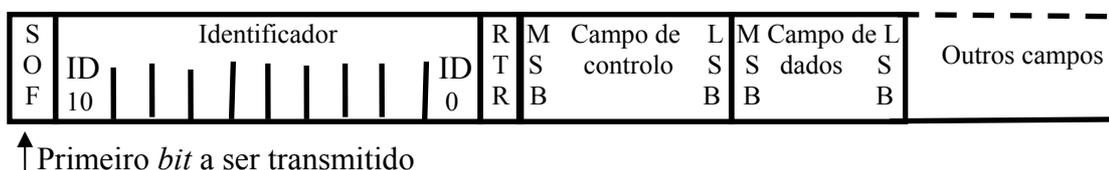


Figura 3.10 - Ordem de transmissão dos *bits*.

### 3.9 PRIORIDADE

Conforme foi referido na secção 3.4, as prioridades são definidas durante a fase de projecto do sistema, sendo tanto maior a prioridade de determinada mensagem, quanto menor for o valor numérico do seu identificador.

Utilizando a especificação 2.0 B [27], o identificador é constituído por 29 *bits* o que significa que o número possível de identificadores diferentes (mais de 500 milhões) para as mensagens é consideravelmente superior ao número de identificadores permitidos se for utilizada a especificação 2.0 A (2032). Como consequência disto, o número de níveis de prioridades também é significativamente superior. Não existe, no entanto, na maioria dos casos, necessidade de utilizar o CAN 2.0 B, uma vez que 2032 níveis de prioridade são suficientes para a generalidade das aplicações. Utilizando o

CAN 2.0B acaba por se desperdiçar largura de banda na maioria das aplicações.

Em termos de implementação existem controladores CAN que permitem apenas o formato 2.0A (Philips 80C200), enquanto que outros (Intel 82527) permitem os formatos CAN 2.0 A e CAN 2.0 B. A escolha do tipo de controlador a utilizar, deve ter em consideração, conforme foi referido na secção 3.5, a compatibilidade entre os diferentes formatos de *frame*.

### 3.10 TAXAS DE TRANSMISSÃO

No CAN a taxa de transmissão depende do comprimento de barramento e vice-versa. Esta limitação surge devido aos processos de arbitragem e *error-recovery*, o tempo de *bit* nunca deve ser inferior ao dobro do atraso de propagação no barramento. Segundo a especificação ISO 11989 [4], a taxa máxima corresponde a 1 Mbit/s para um tamanho de 40 metros. Para tamanhos de barramento maiores as recomendações segundo a CiA [25], *CAN in Automation*, são:

- 500 Kbit/s para distâncias até 100 metros;
- 250 Kbit/s para distâncias até 250 metros;
- 125 Kbit/s para distâncias até 500 metros ;
- 50 Kbit/s para distâncias até 1Km.

### 3.11 TAMANHO DUMA REDE CAN

O número de nós que podem existir numa única rede é, teoricamente, ilimitado [27]. No entanto, o número de identificadores e a capacidade dos *transceivers* existentes impõem restrições. Assim, dependendo do tipo de *transceiver*, até 32 ou 64 nós por rede é normal, existindo no entanto *transceivers* que permitem ligar pelo menos 110 nós por rede [26].

### 3.12 ARQUITECTURA DOS CONTROLADORES CAN

Existem duas versões de controladores CAN, uma com a designação de CAN básico, *Basic CAN*, e uma versão superior, possuindo um “filtro de aceitação” implementado em *hardware*, com a designação de CAN completo, *Full CAN* [22].

Na versão CAN básico, existe uma relação estreita entre o CPU e o controlador CAN pelo que todas as mensagens, que existam no barramento, são verificadas individualmente pelo microcontrolador. Isto resulta na ocupação do CPU para verificar todas as mensagens, em vez de processar apenas as relevantes, o que tende a limitar o *baud rate* a 250 Kbit/s. Por outro lado, na versão CAN completo, a introdução de um filtro de aceitação permite ao controlador recusar mensagens irrelevantes, através da verificação dos identificadores, libertando o CPU para processar apenas mensagens consideradas relevantes o que permite conseguir taxas de transmissão mais elevadas (1 Mbit/s).

## 3.13 MÉTODOS DE DETECÇÃO E SINALIZAÇÃO DE ERRO

Numa rede CAN há a garantia de que uma *frame* é simultaneamente aceite por todos os nós ou apenas por alguns. Assim a consistência de dados é, uma propriedade do sistema, alcançada através de conceitos de *multicast* e suporte de erros, *error handling* [4].

Para implementar o segundo conceito, estão definidos diversos tipos de erro e respectivos mecanismos de detecção.

Ao contrário de outros sistemas de barramento, o protocolo CAN não utiliza mensagens de confirmação mas detecta e assinala erros que ocorram [28].

### 3.13.1 Detecção de erros

Para a detecção de erros o protocolo CAN implementa três mecanismos ao nível da mensagem e dois ao nível do *bit*.

Esses mecanismos são [4]:

- Monitorização;
- Verificação da regra de inserção do *bit*;
- Verificação da *frame*;
- Verificação da sequência CRC de 15 *bits*;
- Verificação do reconhecimento (*ACK*).

Estes tipos de mecanismos, não mutuamente exclusivos, são utilizados para verificar a ocorrência dos seguintes tipos de erro:

#### 3.13.1.1 ERROS AO NÍVEL DA MENSAGEM

##### Erro de CRC

O CRC salvaguarda a informação contida na *frame*, sendo adicionados pelo emissor, no fim de transmissão desta, *bits* redundantes. O receptor recalcula estes *bits* e compara-os com os *bits* recebidos. Se não forem coincidentes é assinalado um erro de CRC.

##### Erro de formato

Este tipo de erro é detectado quando um campo de formato fixo contiver um ou mais *bits* ilegais.

##### Erro de reconhecimento

Este tipo de erro é detectado quando o emissor não detecta um *bit* “dominante” durante o *slot* ACK.

### 3.13.1.2 ERROS AO NÍVEL DO *BIT*

#### Erro de *bit*

O emissor tem capacidade para detectar erros de *bit* baseado na monitorização dos sinais do barramento. Assim cada nó que transmite também monitora o barramento, detectando, caso existam, diferenças entre o valor do *bit* transmitido e o valor do *bit* monitorizado.

#### Erro de inserção de *bit*

O erro de inserção de *bit* é detectado sempre que existam seis *bits* consecutivos com nível lógico igual, num campo da *frame* que seja codificado pelo método da inserção do *bit*.

### 3.13.2 Sinalização de erros

Se for detectado um erro de *bit*, de inserção de *bit*, de formato ou no reconhecimento, por qualquer nó, este inicia a transmissão de uma *flag* de erro durante o *bit* seguinte.

Se um erro na CRC for detectado, é iniciada a transmissão de uma *frame* de erro no *bit* seguinte ao delimitador ACK, a menos que já tenha sido iniciada uma *frame* de erro para qualquer outra condição de erro [4].

## 3.14 O NÍVEL FÍSICO

O nível físico é responsável pela transferência de *bits* entre os diferentes nós de uma rede CAN; este define o modo como os sinais são transmitidos, lidando, por isso, com parâmetros como temporização, codificação e sincronização das sequências de *bits* a serem transmitidos [29]. Na especificação inicial feita por Robert Bosh [27], nenhum meio foi definido, permitindo diferentes opções para o meio de transmissão e níveis dos sinais. Estas propriedades foram posteriormente contempladas pelo *standard* ISO [4] onde estão definidas características dos sinais.

O *standard* DS 102-1 da CiA [25], completa aquelas definições relativamente a especificações do meio físico e conectores. A CiA especifica também diversas ligações mecânicas (cabos e conectores).

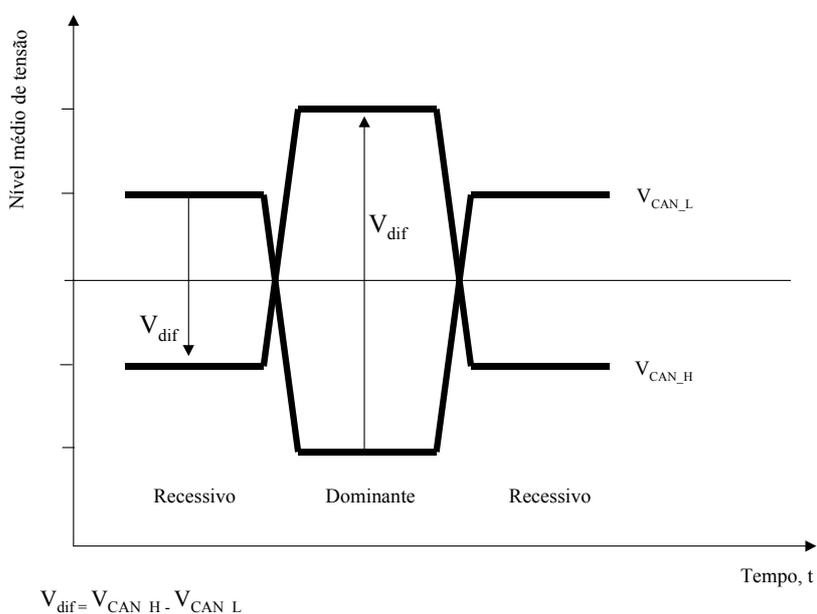
A rede CAN opera num modo quase estacionário: por cada transmissão de um *bit* é dado tempo suficiente para estabilizar o nível do sinal antes que seja feita a amostragem quase simultânea por todos os nós. Isto significa que a capacidade do barramento é de um *bit*. Devido à mencionada estabilidade requerida, o comprimento máximo da rede depende da taxa de transmissão. Alguns valores típicos para as taxas de transmissão foram referidos na secção 3.10.

### 3.14.1 Meio físico

Com o CAN é possível utilizar diversos meios físicos, tais como: par de fios entrelaçados, fibra óptica, rádio frequência, etc. Actualmente, a maioria das aplicações utiliza um barramento diferencial a dois fios [22].

### 3.14.2 Transmissão diferencial

No CAN os sinais são transmitidos utilizando tensões diferenciais, derivando daí muita da imunidade ao ruído e tolerância a falhas que o caracterizam. As duas linhas de sinal são designadas por 'CAN\_H' e 'CAN\_L'. Um '0' corresponde ao sinal CAN\_H superior ao CAN\_L e como tal designado por *bit* "dominante". A situação contrária, CAN\_L superior a CAN\_H, corresponde a um *bit* "recessivo" ou '1' conforme ilustra a Figura 3.11.



**Figura 3.11 - Representação física de *bit*.**

A utilização de diferenciais de tensão permite às redes CAN funcionar quando uma das linhas de sinal for danificada, ou em situações extremas de ruído. Recorrendo a um simples par entrelaçado, as entradas CAN diferenciais cancelam o ruído de forma efectiva, assegurando que estão compreendidas na gama de modo comum.

Existem interfaces económicas [26] disponíveis que fazem a translação dos níveis lógicos de 5 V para linha balanceada requerida pelo CAN e vice-versa.

Todos os *bits* são transmitidos de acordo com o método *Non-Return-to-Zero* (NRZ). Isto significa que o nível do *bit* é constante durante a sua duração, sendo "dominante" ou "recessivo". Este método apresenta uma densidade espectral baixa, possibilitando um bom aproveitamento da largura de banda de transmissão.

### 3.14.3 Tolerância a falhas no barramento

A especificação *standard* [4], define meios próprios para detecção de falhas, que permitem o funcionamento da rede em condições adversas, mesmo com uma relação sinal/ruído reduzida. A tolerância a falhas fundamenta-se na operação com apenas um fio e é capaz de garantir continuidade da operação da rede na presença de:

- Quebra de um fio;
- Curto circuito de um fio à massa, *ground*, ou à bateria de alimentação,  $V_{CC}$ ;
- Curto circuito entre as linhas CAN\_L e CAN\_H;

A especificação *standard* não prevê meios de tolerância no caso de anomalia simultânea nos dois fios do barramento.

## 3.15 TEMPO DE *BIT* E SINCRONIZAÇÃO

### 3.15.1 Representação e definição do tempo de *bit*

O tempo nominal do *bit*,  $t_B$ , é definido como sendo a duração de um *bit*, correspondendo ao inverso da taxa nominal de transmissão, número de *bit/s* transmitidos, de um emissor ideal na ausência de resincronização [4]. Os segmentos que constituem o tempo de *bit* são ilustrados na Figura 3.12.

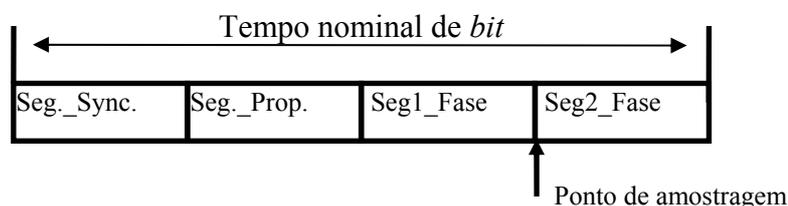


Figura 3.12 - Divisão do tempo de *bit*.

As funções de gestão do barramento executadas durante do tempo do *bit*, tais como o ajuste da sincronização de um nó, atraso de compensação da transmissão da rede e a posição dos pontos de amostragem, são definidas pela lógica programável do tempo de *bit* do controlador CAN [5, 7]. Os segmentos que constituem o tempo *bit* são:

- Segmento de Sincronização, *Seg\_Sync.*. Este segmento é usado para sincronizar os várias nós do barramento, sendo esperada uma transição durante o mesmo;
- Segmento do Tempo de Propagação, *Seg\_Prop.* Este segmento é utilizado para compensar os tempos de atraso físico na rede. Estes atrasos consistem nos tempos de propagação do sinal na linha do barramento e nos tempos de atraso interno aos nós;
- Segmentos 1 e 2 dos *buffers* de fase, *Seg1\_Fase* e *Seg2\_Fase*. Estes segmentos são utilizados para compensar erros de fase nas transições. Estes segmentos podem ser alongados ou encurtados por resincronização;

- Ponto de amostragem é o instante no qual é lido e interpretado o nível do barramento como sendo o valor do respectivo *bit*. Ocorre no final do segmento Seg1\_Fase.

O tempo de processamento de informação é o segmento de tempo que tem início no ponto de amostragem e é reservado para cálculos do nível de *bit* subsequente.

### 3.15.2 Sincronização

Existem duas formas de sincronização:

- Sincronização forçada (*Hard synchronization*), no início de uma mensagem;
- Ressincronização, durante a transferência da mensagem.

As regras que regem os processos de sincronização são as seguintes:

- Somente é permitida uma sincronização durante um tempo de *bit*;
- Uma transição só pode ser usada para sincronização se o valor detectado no ponto de amostragem anterior, diferir do valor do barramento imediatamente após a transição;
- A sincronização forçada é realizada durante o estado de barramento livre, sempre que haja uma transição de “recessivo” para “dominante”.

A ressincronização consiste em produzir um alongamento ou estreitar do tempo de *bit*, por forma a permitir que a localização do ponto de amostragem seja correcta.

## 3.16 CONFIGURAÇÕES BASE DE CIRCUITOS INTEGRADOS

Existem três configurações base no que diz respeito a utilização de integrados CAN [21]:

- Controlador CAN discreto (*Stand alone*);
- Microcontrolador com controlador CAN interno;
- *Serial Linked Input Output (SLIO)*.

### 3.16.1 Configuração utilizando controladores CAN discretos

No primeiro caso, o controlador CAN é ligado ao microcontrolador através dos barramentos de dados e endereços. Esta configuração é útil quando se pretende actualizar sistemas já existentes. Implica que seja possível efectuar a interface do controlador, ou seja, não pode ser utilizado um controlador apenas com E/S.

### 3.16.2 Configuração utilizando microcontrolador com controlador CAN

Se um sistema for desenvolvido de raiz, a segunda opção deve ser utilizada relativamente à primeira. Nesse caso, a interface de dados e endereços é feito internamente existindo vários destes dispositivos disponíveis no mercado.

### 3.16.3 Configuração SLIO CAN

O CAN, apesar das vantagens que permite, continua a ser considerado caro em alguns tipos de aplicações. Além disso, não é justificável utilizar microcontroladores para simples operações liga/desliga, *on/off*, tendo sido introduzida para esses casos, por alguns fabricantes, uma versão simples e barata de dispositivos designados vulgarmente por SLIO CAN. Trata-se de uma forma simples de reforçar a capacidade de E/S de um microcontrolador central através do barramento CAN. Os integrados SLIO detêm pouca “inteligência” necessitando de serem programados e calibrados por um microcontrolador, por eles responsável, funcionando sempre como “satélites” desse nó “inteligente” possuindo microprocessador.

Existe também a possibilidade de ter, no mesmo barramento, os três tipos de configurações anteriormente referidas. Mesmo tratando-se de integrados de diferentes fabricantes, com *software* desenvolvido de forma adequada, é possível interligar esses dispositivos.

Seguidamente, nesta secção, será justificada a utilização da tecnologia SLIO CAN em sistemas de controlo de reduzida velocidade (<125 Kbit/s), uma vez que foi esta a configuração base utilizada no âmbito desta dissertação.

Aplicações, como o caso de controlo de estufas, que envolvem, principalmente, operações liga/desliga de diversos dispositivos actuadores (lâmpadas, motores, electroválvulas, etc.) requerem tráfego reduzido de informação. Para aplicações deste tipo pode tornar-se caro e injustificado utilizar um barramento de dados de alta velocidade para informação de controlo, sendo por isso recomendada a utilização de um barramento de dados de menor velocidade.

Existem assim, duas configurações possíveis, sendo uma vantajosa para aplicações que exijam elevada velocidade e outra para aplicações de menor velocidade de transferência de informação. Do ponto de vista de engenharia é fácil fazer a passagem de elevada para baixa velocidade e vice-versa, se o mesmo protocolo for utilizado. No entanto, quando um sistema CAN é configurado para velocidades reduzidas, geralmente permanece o mesmo conjunto de circuitos integrados CAN. Isto não é económico uma vez que existe um desperdício de largura de banda de transmissão e recursos, utilizando a configuração composta por controladores CAN de elevada velocidade em aplicações de controlo que requerem velocidade reduzida, as quais, na maioria dos casos, envolvem funções simples de liga/desliga. Em resumo, o conjunto de circuitos integrados utilizado para aplicações de reduzida velocidade teria um custo igual ao gasto para aplicações de velocidade elevada.

O SLIO CAN foi desenvolvido para tirar partido da tecnologia CAN em aplicações de baixa velocidade com menor custo. Tipicamente, um integrado SLIO CAN custa cerca de 1/5 do preço de um microcontrolador com interface CAN.

#### **SLIO (*Serial Link Input/Output*)**

Na sua forma mais simples, o SLIO pode ser visto como um porto E/S com controlador CAN, respeitando a especificação 2.0 A e 2.0 B (passivo). Existem SLIOs que permitem uma taxa de transmissão máxima de 125 Kbit/s, utilizando um oscilador interno, sendo no entanto possível operar a 250 Kbit/s utilizando para tal um oscilador externo a cristal, o que aumenta no entanto o preço e complexidade da interface.

As Figuras 3.13 e 3.14 ilustram configurações típicas de sistemas CAN e SLIO CAN utilizadas em aplicações de alta e baixa velocidade, respectivamente.

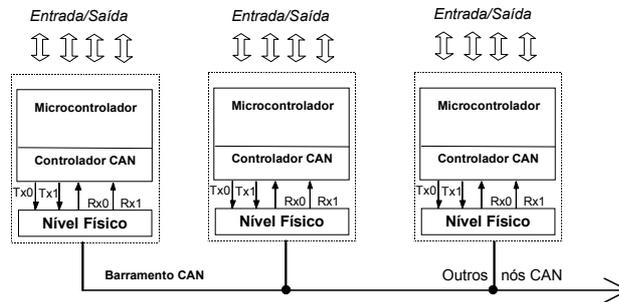


Figura 3.13 – Sistema CAN típico.

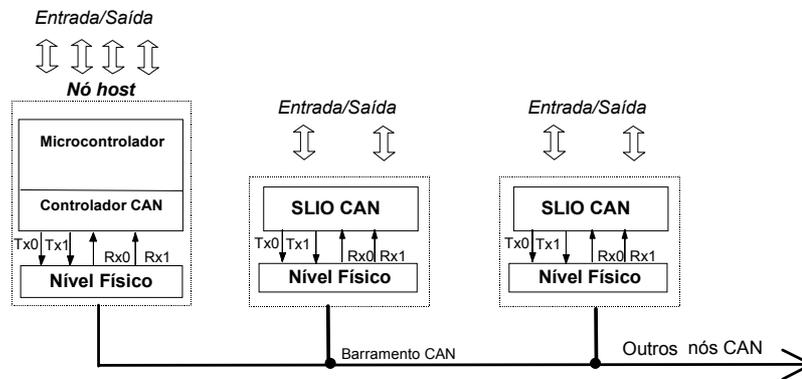


Figura 3.14 – Sistema SLIO CAN.

A diferença fundamental entre as duas configurações é a ausência do “dispendioso” microcontrolador nos nós SLIO CAN.

Conforme foi referido anteriormente, os dispositivos SLIO CAN são programados e calibrados por um nó “inteligente”, *host*, o qual consiste num nó possuindo microcontrolador. Um único nó *host* pode controlar um máximo de 16 SLIOs num barramento SLIO CAN, estando este número relacionado com a existência de 4 *bits* configuráveis no identificador em cada SLIO o que limita a 16 o número máximo de identificadores únicos possível, conforme é ilustrado na Figura 3.15.

Identificador <i>standard</i>	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
Identificador do SLIO	0	1	P3	1	0	P2	P1	P0	1	0	Dir

P0, P1, P2 e P3: pinos para configuração do identificador do SLIO  
Dir: *bit* de direcção para mensagens do SLIO CAN

Figura 3.15 – Relação entre o identificador do SLIO P82C150 e o identificador CAN *standard*.

Combinando dois fabricantes (Philips e National) de SLIOs, os quais possuem configurações diferentes para os identificadores (*bit* ID1, da Figura 3.15 diferente), é possível um máximo de 32 nós num sistema SLIO CAN [31]. Em ambos os casos o *bit*

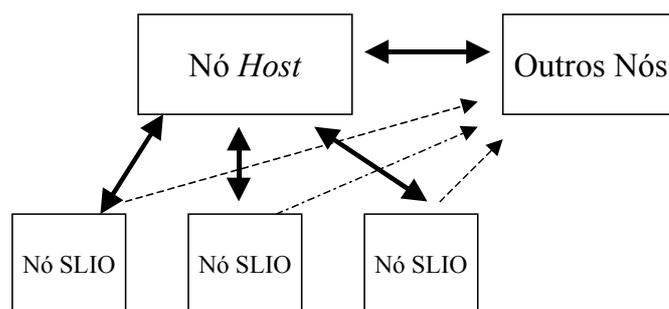
ID0 do identificador CAN, indica a direcção da transferência de dados, sendo ‘0’, em mensagens transmitidas do nó *host* para o(s) SLIO(s), e ‘1’ para *frames* com “sentido” oposto. O nó *host* pode fazer *polling* aos seus SLIOs transmitindo *frames* remotas para o sistema SLIO CAN verificando, periodicamente, a existência de todos os nós assegurando desta forma a integridade do sistema.

Outra característica importante na comunicação *host vs* nós SLIO CAN é a de o campo de dados do sistema SLIO CAN ser fixo. O primeiro *byte* do campo de dados funciona como registo de *status* e comando, correspondendo os restantes *bytes* de dados aos pinos de E/S do SLIO (16 *bits* E/S disponíveis no caso do PCA82C150 da Philips). Cada pino E/S pode ser programado individualmente.

## Endereço físico do SLIO

Devido aos 4 *bits* configuráveis do identificador, o SLIO CAN evidencia a técnica de endereço físico na qual a cada nó SLIO CAN é atribuído um identificador único por *hardware* durante o *reset*. Deixando de haver endereçamento funcional como no CAN em geral. Além disso, até certo nível actua como uma configuração mestre/escravo virtual. A relação SLIO-*host* está representada na Figura 3.16. Conforme foi referido anteriormente, todos os 16 SLIOs num barramento CAN devem ser controlados apenas por um *host*. No entanto, em algumas situações, podem ser divididos em grupos e controlados por vários *hosts*, não podendo no entanto, o número de SLIOs exceder 16 (ou 32) no mesmo barramento. No ambiente *multi-hosts* apenas é necessário um dos *hosts* para efeitos de calibração [31].

Devido à técnica de *broadcast* do CAN, todos os outros nós CAN (além do *host* e dos SLIOs) podem receber mensagens transmitidas pelos SLIOs (Figura 3.16). Sendo por isso, necessário garantir, através do teste de aceitação das mensagens, que todos os outros nós inteligentes (excepto o *host*) não actuem sobre os dados uma vez que tal situação poderia provocar erros e “confusão” no sistema.



**Figura 3.16 – SLIO e outros nós CAN partilhando o mesmo barramento.**

Nas próximas secções serão consideradas aplicações utilizando o SLIO P82C150 da Philips uma vez que foi o utilizado nesta dissertação.

## Iniciação do sistema SLIO CAN

Num sistema SLIO CAN, utilizando o SLIO P82C150, é necessário um nó *host* que envie uma mensagem de calibração com períodos de tempo iguais ou inferiores a 8000 tempos de *bit* para desta forma sincronizar o oscilador interno dos SLIOs com o tempo de *bit* do barramento [32]. Uma vez calibrados, os nós SLIO podem transmitir e receber mensagens como outros nós CAN. A Figura 3.17 ilustra o exemplo de uma mensagem de calibração [33].

SOF	Campo de arbitragem	Campo de controlo	Byte 1	Byte 2	Campo CRC
0	000 1010 1010 0	000 010	<u>1</u> 010 1010	0000 0100	000 0 1011 1000 00 <u>0</u>

| - *bit* inserido pelo controlador CAN para garantir a regra de inserção de *bit*.

**Figura 3.17 – Mensagem de calibração.**

A transmissão pelo SLIO é efectuada automaticamente pelo controlador CAN, que o mesmo possui. Durante a iniciação, o SLIO é configurado para realizar funções tais como entrada *event capture*, saída digital ou conversão analógico-digital. A iniciação é feita pelo nó *host* programando os registos dos nós SLIO através do barramento CAN. Do ponto de vista da recepção, o SLIO possui um controlador CAN com uma lógica interna que lhe permite reconhecer automaticamente o identificador único do nó. A *frame* de reconhecimento é composta pelo *status* corrente e pelo valor dos registos do SLIO, permitindo desta forma ao nó *host* uma verificação da mensagem transmitida e do estado do(s) SLIO(s).

Quando é incorporado um novo SLIO numa rede CAN, o novo nó é capaz de se fazer conhecer ao controlador *host* após detectar um mínimo de 3 *frames* CAN, num espaço de 8000 tempos de *bit* [33]. O novo SLIO envia uma mensagem própria de “assinalar presença” confirmando ao *host* a sua presença na rede. O único requisito que o novo elemento deve preencher é ter um identificador diferente de qualquer outro dos nós activos na rede CAN.

## O tamanho da rede SLIO CAN

Uma vez que o SLIO CAN não possui oscilador a cristal, a lógica interna de tempo de *bit* do SLIO é optimizada para tolerância máxima do oscilador [31]. Isto encurta o tamanho efectivo do sistema CAN uma vez que o ponto de amostragem deve estar o mais próximo possível do início do tempo de *bit* o que limita significativamente o tempo permitido para o atraso de propagação na linha de transmissão, resultando em comprimentos de barramento mais reduzidos do que num sistema CAN convencional. Na Tabela 3.1 são comparados os comprimentos de barramento de um sistema CAN com os de um sistema SLIO CAN [31]. Refira-se no entanto que os comprimentos para um sistema SLIO CAN são suficientes para diversas aplicações, como o caso do controlo de estufas agrícolas.

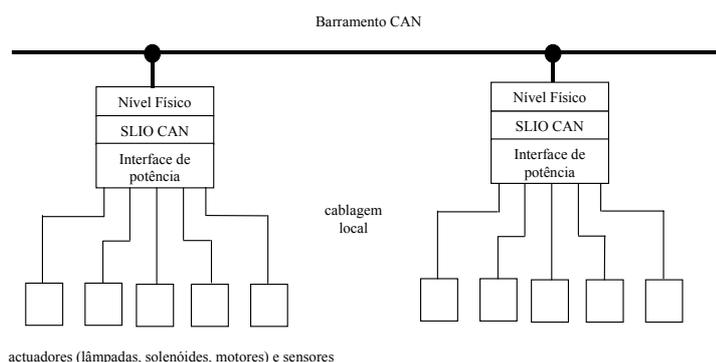
Taxa de transmissão	P82C150 (SLIO CAN)	P8xC592/PCA82C200 (CAN)
---------------------	-----------------------	----------------------------

125 Kbit/s	80 m	530 m
100 Kbit/s	120 m	620 m
50 Kbit/s	300 m	1300 m
20 Kbit/s	850 m	3300 m

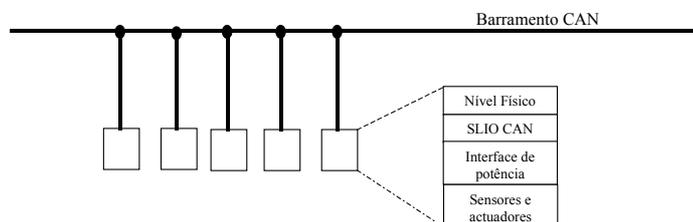
**Tabela 3.1– Distância máxima aconselhada, entre dois nós extremos, para uma rede CAN e SLIO CAN.**

## Futuro

Alguns autores [31] consideram que os semicondutores como o SLIO e circuitos de accionamento, de potência, podem ser incorporados em dispositivos “*plug-and-play*“. Esta aproximação modular elimina a cablagem local possibilitando uma melhoria da eficácia do sistema total, uma vez que existem poucos fios, que podem falhar, sendo também mais robusto em termos de compatibilidade electromagnética. Esta visão do futuro é ilustrada pelas Figuras 3.18 e 3.19.



**Figura 3.18 - Sistemas actuais.**



**Figura 3.19 – Sistemas futuros.**

## 3.17 PORQUE ESCOLHER O CAN?

O protocolo CAN, devido à sua fiabilidade e baixo custo, tem tido bastante aceitação,

existindo actualmente vários dispositivos acessíveis no mercado de circuitos integrados.

Uma das vantagens do protocolo CAN é a capacidade de suportar o erro. Esta capacidade consiste em retransmitir a mensagem que não aparece correctamente no barramento CAN.

O meio de transmissão é simples, sendo o mais comum um par de fios. Um sistema CAN pode trabalhar apenas com um fio. Dependendo da aplicação, existem outros tipos de ligações que podem ser escolhidos, como a via rádio, óptica, etc.

Num sistema de comunicações, quando um nó do sistema falha existe a possibilidade de este nó bloquear todo o sistema, mas com o protocolo CAN isto não sucede porque esse nó pode ser excluído de enviar e receber no barramento CAN.

Utilizando um relógio global é simples implementar aplicações em tempo real, porque é possível sincronizar um sistema e esta sincronização é mantida durante a latência de uma mensagem.

Uma vez que todos os módulos “escutam” todas as mensagens, é fácil implementar aplicações do tipo *event driven*. Assim, um módulo pode ser programado para reagir logo que qualquer mensagem surja no barramento CAN.

O protocolo CAN é bom para implementar sistemas de controlo distribuído. O método de arbitragem para determinar a prioridade das mensagens e a possibilidade de um determinado número de nós tentar o acesso ao barramento (multi-mestre) possibilita a construção de bons sistemas de controlo.

Pode-se concluir que a aceitação do protocolo CAN por parte da indústria de automação se deve a vários factores, designadamente:

- Baixo custo;
- Capacidade de funcionar em ambientes com condições eléctricas adversas;
- Elevado grau de capacidades de tempo real e controlo distribuído;
- Fácil utilização;
- Disponibilidade de componentes, controladores, com o protocolo CAN;
- Existência de *standards*.

## 4 DESCRIÇÃO DO SISTEMA GLOBAL

### 4.1 INTRODUÇÃO

Novos conceitos de métodos e técnicas para a gestão e controlo de sistemas agrícolas, como é o caso de estufas agrícolas, requerem a utilização de sistemas computacionais [1]. Tarefas, tais como gestão ambiental, irrigação e fornecimento de nutrientes, devem ser controladas de uma forma económica, por forma a que seja possível conseguir colheitas rentáveis. Para atingir estes objectivos são necessárias acções de controlo adequadas e executadas por um sistema integrado que coordene cada acção, de acordo com a combinação específica de diferentes entradas provenientes de sensores. Além disso este controlo tem, por vezes, de ser feito a partir de locais remotos.

Para escolher um barramento de campo ou infra-estrutura em rede é necessário ter em conta os requisitos e necessidades da aplicação a que se destinam, sendo alguns dos aspectos que determinam o cenário das aplicações agrícolas em causa os seguintes:

- Por vezes as estufas situam-se em locais remotos;
- O número de sensores aumenta rapidamente com o número de parâmetros mensuráveis e com a necessidade de redundância física;
- Os sensores são, por vezes, posicionados na área útil de cultivo ao contrário dos actuadores que se encontram em posições fixas e fora dela;
- O solo tem de sofrer alterações com alguma frequência, o que não se coaduna com a posição fixa de sensores na área de cultivo.

### 4.2 O SISTEMA GLOBAL

O trabalho que serve de base a esta dissertação, pretende ser uma solução para a componente de actuação de um SCMF que está a ser desenvolvido na Universidade de Trás-os-Montes e Alto Douro (U.T.A.D.), pelo Departamento de Engenharias da mesma, o qual é uma rede integrada que liga diversas plataformas de comunicação, tão diferentes como sejam o CAN, tecnologias de transmissão sem fios (RF) e *Transfer Control Protocol/Internet Protocol* (TCP/IP)[1]. O objectivo primordial é o de conceber um sistema de controlo distribuído a baixo custo, boa performance, flexibilidade e de elevada funcionalidade. Relacionado com estes factores e com a complexidade associada aos modelos e estratégias de controlo, o sistema proposto pretende ser o mais aberto possível e interagir com o utilizador. Isto significa que sempre que o utilizador defina um modelo de controlo, o *software* da aplicação deve permitir o retorno ao utilizador de informação relativa aos parâmetros mensuráveis que são monitorados com regularidade. Para obter bons resultados, o sistema dispõe de uma ampla variedade de sensores.

Actualmente, o objectivo de distribuir as tarefas de controlo é cada vez mais dependente do desenvolvimento da microelectrónica [34]. O custo de maior poder de processamento e ferramentas de comunicação está continuamente a diminuir, devido ao incremento da integração dos dispositivos na área da microelectrónica, como é o caso

de micro-processadores e microcontroladores que possuem diversos periféricos, nomeadamente para comunicações.

A utilização de sistemas computacionais em aplicações agrícolas, tais como estufas agrícolas, é possível e viável devido ao seu contínuo embaratecimento [34]. Se uma solução de controlo óptimo for pretendida para aumentar a eficiência, é necessário recorrer a este tipo de sistemas para lidar com a quantidade de dados e capacidade de cálculo requeridas para que tal objectivo seja conseguido.

Neste campo, no entanto, a maior preocupação prende-se com a definição dos parâmetros mais importantes a medir e controlar por forma a garantir uma complexidade controlável [3].

Porque as estufas podem estar situadas em locais remotos, foram previstos tipos diferentes de comunicações para realizar diferentes tarefas conforme a seguir se refere.

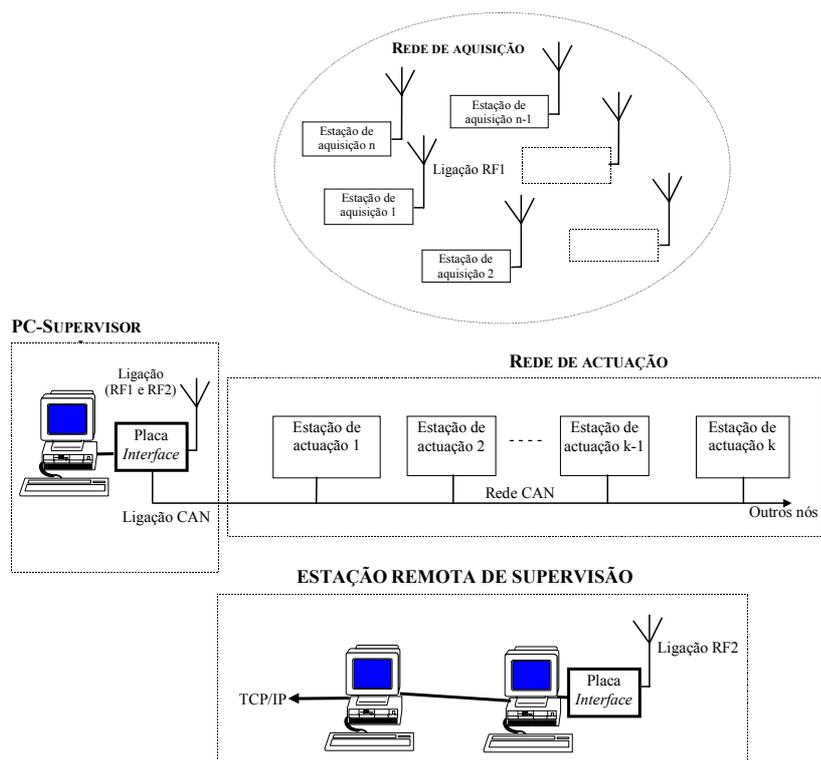
Uma vez que o número de sensores tende a aumentar, de acordo com o vasto número de parâmetros a serem controlados e à necessidade de redundância física, uma solução utilizando cabo seria incómoda para a rede de aquisição. Para resolver este problema é utilizada uma ligação sem fios. Assim, para a rede de aquisição foi implementada uma solução baseada em ligação por RF no interior de cada estufa, o que permite a mobilidade dos vários sensores instalados na mesma, de acordo com a colheita e condições ambientais, não prejudicando os trabalhos de cultivo.

A mobilidade dos actuadores, por outro lado, não é o problema maior, devido à posição fixa que estes ocupam e à necessidade de cablagem para a respectiva alimentação eléctrica, pelo que neste caso se justifica uma solução cableada para a rede de actuação. Assim, para o comando dos actuadores optou-se por comunicação através de uma rede CAN, a qual permite robustez e baixo custo ao sistema bem como capacidade de expansibilidade [21, 22].

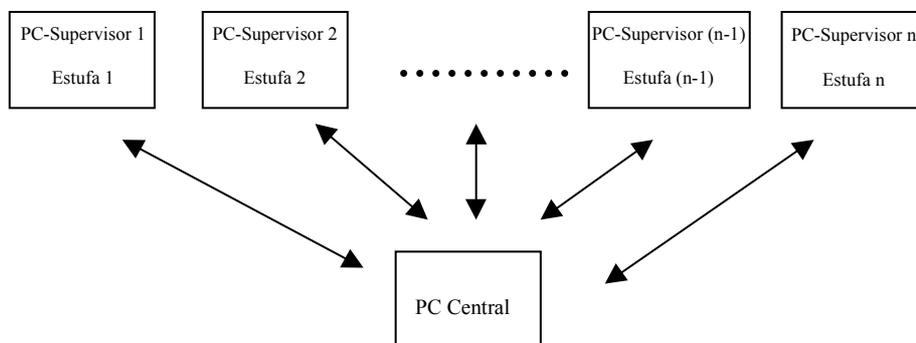
Ambas as redes, anteriormente referidas, podem comunicar com um PC instalado na estufa. Este PC destina-se essencialmente ao desempenho de tarefas de supervisão e “*routing*” sendo ligado por RF, CAN ou Ethernet a outros subsistemas (redes de aquisição e actuação bem como à estação remota de supervisão), permitindo ao utilizador interagir com o sistema a partir de um local remoto.

O funcionamento do sistema processa-se da forma seguinte [34]: numa primeira fase, do ciclo de controlo, é feita aquisição de valores dos sensores através da rede de aquisição, a qual transmite os mesmos, por RF (RF1=433,092 MHz), para uma estação base, composta por um PC e uma placa de interface, que tem funções de supervisão, PC-supervisor. No PC os dados são processados por forma a gerar acções de controlo, utilizando para tal algoritmos de controlo adequados [2]. Após o processamento, são enviados comandos para a rede de actuação, via barramento CAN, comandando diversos actuadores tais como electroválvulas, ventiladores, painéis de sombreamento, etc. Na Figura 4.1 é representado um esquema geral do sistema.

Através de outra ligação por RF (RF2=458 MHz), vários destes PCs de supervisão, instalados cada um em sua estufa, podem ser ligados a um PC central, permitindo o controlo de uma estufa ou conjunto de estufas a partir de um ponto central situado num local remoto [34], conforme ilustrado na Figura 4.2.



**Figura 4.1 – Configuração geral do sistema.**



**Figura 4.2 – Controlo de um conjunto de estufas.**

Em resumo do que foi dito, é proposto um sistema de controlo com as seguintes componentes:

- Rede de aquisição (com comunicação por RF);
- Rede de actuação (com comunicação por CAN);
- Acesso a sistemas remotos por Ethernet .

Um sistema destes possui maior importância sabendo que os operadores de estufas tradicionais não possuem o *know how* necessário para lidar com novas tecnologias. Este facto faz com que exista um aumento nos custos de produção sempre que for necessária assistência técnica, uma vez que esta exige na maioria dos casos a presença de um técnico no local. É, portanto, vantajoso tornar possível a monitorização do sistema a partir de um local remoto. Para conseguir este objectivo, o sistema permitirá uma

ligação com base no protocolo TCP/IP a uma rede local ou remota. Desta forma, a ligação do PC central à rede Ethernet/Internet possibilita a interacção com o sistema a partir de um local remoto, neste caso fora da UTAD, permitindo que o utilizador observe o comportamento do sistema ou altere a configuração do mesmo sem ter de se deslocar a cada uma das estufas.

Após terem sido apresentados os objectivos pretendidos para o sistema global de controlo de estufas agrícolas, será focada a partir daqui a atenção apenas sobre a rede de actuação, uma vez que é nesta que se centra a presente dissertação.

### 4.3 REDE DE ACTUAÇÃO

Conforme foi referido, os actuadores ocupam posições fixas e necessitam de fios para o fornecimento de energia eléctrica. Assim, para ligar os actuadores e formar uma rede de actuação, foi escolhida uma rede CAN devido à flexibilidade, robustez e baixo custo permitidos. Para além das propriedades anteriores e ter provado ser uma solução fiável para esta aplicação [34, 35], em trabalhos anteriormente desenvolvidos, o CAN possui também as seguintes características:

- Transferência de dados de forma rápida, robusta e segura para distâncias até 1 Km;
- Flexibilidade, o que significa que o sistema pode ser expandido sem necessidade de alterações do *hardware* existente;
- Utiliza o método *Non-Return-to-Zero* (NRZ) para codificação de *bit*, o que produz menos interferência electromagnética para taxas de transferência elevadas [1, 21, 36]. Isto é de grande importância, uma vez que a rede de aquisição utiliza ligação por RF para interligar os sensores à estação base.

As características do protocolo CAN foram abordadas no capítulo 3 que lhe foi dedicado.

Quanto às estações de actuação e restante *hardware* serão também apresentadas com detalhe em capítulos próprios. Realce-se, no entanto, que as mesmas foram projectadas sempre seguindo uma orientação de modularidade, por forma a simplificar o protótipo e reduzir custos de produção.

Uma vez que é objectivo do sistema proporcionar redundância relativamente às comunicações e dados, as estações da rede de actuação foram projectadas por forma a permitirem, para além de comandar actuadores, efectuar também se necessário, tarefas de aquisição de valores provenientes de sensores. Esta tarefa complementar de recolha de dados através da rede CAN, possibilita um caminho alternativo no caso de anomalia temporária da rede de aquisição que comunica por RF.

### 4.4 A ARQUITECTURA DA REDE DE ACTUAÇÃO PROPOSTA

A arquitectura proposta para a rede de actuação é composta por vários nós, os quais se encontram no interior da estufa alguns dos quais perto dos actuadores que comandam. Estes nós, destinam-se a executar comandos provenientes do PC-supervisor, sendo estes comandos resultado do processamento de dados, recolhidos pelos sensores, segundo

algoritmos de controlo adequados [2] à aplicação concreta. Para permitir a simplificação da arquitectura de todo o sistema as estações com microcontrolador são idênticas (*hardware*) entre si, divergindo apenas nas funções que desempenham (*software*).

O PC-supervisor possui entre as suas características, a possibilidade de configuração de todo o sistema, processamento dos dados com a respectiva optimização de constantes e *set-points* intrínsecas ao desenvolvimento mais adequado do algoritmo de controlo, isto para além de emitir pedidos de informação, pedidos de valores instantâneos de sensores e envio de comandos para actuação.

É importante referir também que a velocidade do barramento CAN da rede de actuação está condicionada pela taxa de transmissão das estações mais lentas, que nesta aplicação corresponde à de algumas estações escravas, *slave*, desta rede, as quais, conforme será referido, devido aos dispositivos que as constituem, limitam a velocidade de transmissão a um máximo de 125 Kbit/s. No entanto, devido à inércia do ambiente da estufa, este facto não é impeditivo para alcançar um bom desempenho do sistema.

A rede de actuação proposta é composta, essencialmente, por uma unidade que será designada por *Host*, e por dois tipos de unidades escravas que são distintos quanto à “inteligência” que possuem, preço e função que desempenham.

O nó *Host* é uma unidade “inteligente” que se baseia no microcontrolador P80C592 da Philips [37], sendo responsável pela calibração das unidades escravas menos “inteligentes” bem como pela comunicação entre o PC-supervisor situado na estufa e estas unidades, desempenhando assim funções de *router* de mensagens.

Quanto às unidades escravas, o tipo de unidade mais barato e menos “inteligente” é colocado junto dos actuadores executando comandos e respondendo a pedidos de informação emitidos pelo nó *Host*, sendo utilizado para as funções de controlo mais frequentes, (liga/desliga) numa estufa, e funcionando assim, como unidades remotas do nó *Host*. Estas unidades têm por base um dispositivo CAN conhecido por SLIO [33] e serão designadas por nós SLIO CAN.

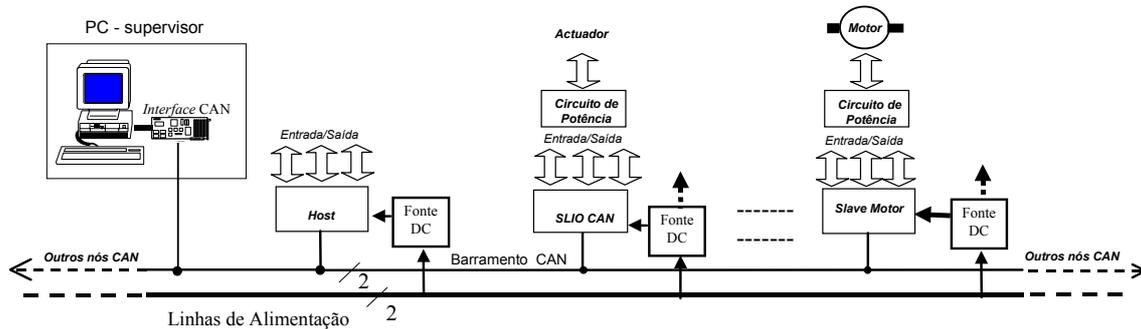
Outro tipo de nó escravo baseia-se, tal como o nó *Host*, nas potencialidades do microcontrolador P80C592 com controlador CAN permitindo maior número de soluções de actuação, nomeadamente duas saídas *Pulse Width Modulation* (PWM). Esta unidade escrava, é comandada pelo PC-supervisor via barramento CAN e é utilizada nesta aplicação para controlar a velocidade de um motor de corrente alternada (ventilador). Este tipo de nó escravo será designado por *Slave Motor* devido às funções que se pretende ele execute, sendo ainda possível, se necessário, aproveitar os portos não utilizados para realizar outras acções de actuação ou aquisição.

Para além dos tipos de nós referidos, é possível introduzir na rede outros nós CAN, desde que os mesmos respeitem a taxa de transferência e mensagens dos já existentes evitando desta forma conflitos na rede.

A implementação e características destas unidades serão oportunamente abordadas com maior detalhe num capítulo próprio, uma vez que aqui apenas se pretende referir de forma genérica e resumida a arquitectura proposta para o sistema.

Devido ao posicionamento dos actuadores numa estufa, e pelo facto de junto de cada um deles ter de existir tensão de alimentação, optou-se por utilizar esta fonte de energia para, através de conversão, obter as tensões de alimentação necessárias “*in loco*”, utilizando para tal fontes lineares de tensão.

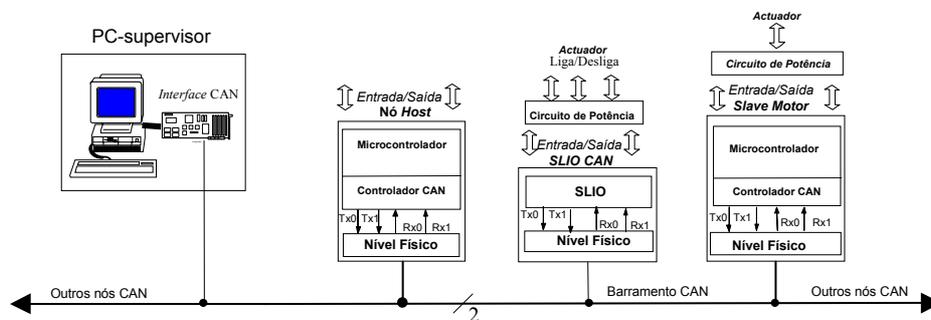
Na sequência do exposto, anteriormente, a arquitectura proposta para a rede de actuação está representada na Figura 4.3.



**Figura 4.3 – Arquitectura da rede de actuação proposta.**

O número de nós SLIO CAN desta aplicação está limitado a 16 por ter sido utilizado apenas um tipo de SLIO, P82C150 da Philips. Este limite de unidades SLIO CAN, está relacionado com limites do número de identificadores permitidos para os SLIOs, conforme foi referido na secção 3.16.3 do capítulo dedicado ao protocolo CAN.

A arquitectura da rede de actuação proposta para o sistema está também ilustrada na Figura 4.4, onde estão presentes os três tipos de nós anteriormente referidos, cujo *hardware* e *software* irão ser alvo de explicação detalhada nos capítulos 5 e 6 desta dissertação, ficando para já uma visão global das unidades fundamentais que compõem os diferentes nós que constituem a rede de actuação proposta.



**Figura 4.4 – Diagrama de blocos dos vários constituintes do sistema.**

## 5 HARDWARE DO SISTEMA

Este capítulo contém a descrição do *hardware* desenvolvido para a implementação da rede de actuação, cuja constituição foi definida no capítulo 4. Será apresentado o *hardware* elaborado para implementar as componentes que permitem a comunicação e actuação na rede proposta.

### 5.1 HARDWARE DE COMUNICAÇÕES

Nesta secção do capítulo pretende-se apresentar, em termos de *hardware*, os nós que compõem a rede de actuação implementada. Para conseguir esse objectivo apresentam-se os esquemas de cada bloco, acompanhados de comentários que traduzam de forma resumida o funcionamento de cada um deles.

Uma das considerações importantes no momento do projecto foi a de que o produto final seja compatível com os sistemas já existentes na U.T.A.D., com os quais irá interagir. O nó PC-supervisor foi desenvolvido no âmbito de outros trabalhos [1, 38] pelo que não será aqui descrito, sendo focada a atenção no *hardware* das unidades desenvolvidas durante a realização desta dissertação.

Conforme foi referido no capítulo 4 a rede de actuação é composta por um nó *Host*, vários nós (SLIO CAN) pelos quais é o responsável e por um nó que tal como o nó *Host* possui como unidade base um microcontrolador e é utilizado para o controlo de velocidade de um motor AC (ventilador da estufa), conforme ilustra a Figura 5.1.

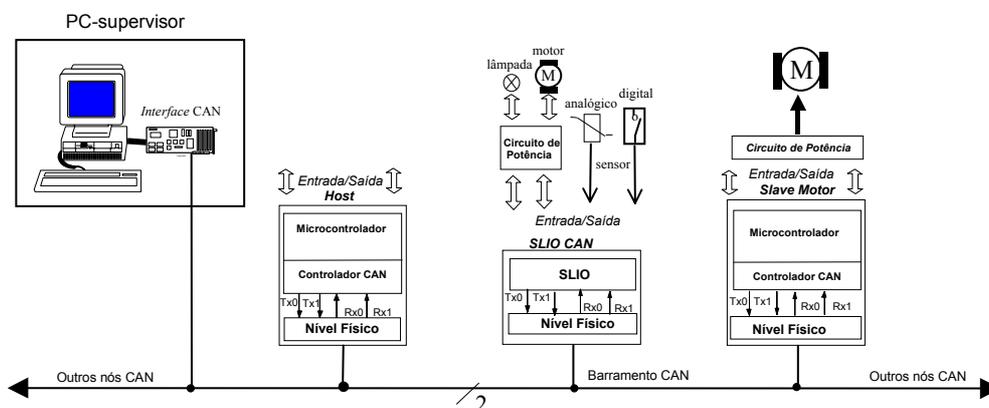


Figura 5.1 – Arquitectura da rede de actuação pretendida.

Seguidamente serão descritas as unidades da rede de actuação propostas e implementadas. As placas de circuito impresso (PCB) das unidades anteriormente referidas podem ser consultadas no Anexo 1.

#### 5.1.1 Nó *Host*

O nó *Host* é responsável pela componente da rede de actuação composta pelos vários nós SLIO CAN. Este nó efectua a calibração de todas as unidades SLIO CAN a seu

cargo e transmite, a estas, ordens ou pedidos de informação enviados pelo PC-supervisor (estação base), localizado na estufa, o qual é o responsável máximo pela gestão do sistema de controlo da mesma, nas suas componentes de aquisição e actuação.

Os blocos constituintes do circuito que compõem o nó *Host* são os seguintes:

- Microcontrolador;
- Relógio de tempo real, *Real Time Clock* (RTC);
- Alimentação e respectivo circuito de supervisão;
- Interface RS-232;
- Interface CAN;
- Expansão de memória (RAM e EPROM).

A seguir descrevem-se mais pormenorizadamente cada um dos blocos anteriormente enunciados.

### 5.1.1.1 O MICROCONTROLADOR

A família de microprocessadores MCS-51 da Intel, da qual o 8051 é o membro mais conhecido, é sem dúvida uma das mais utilizadas na indústria. Entretanto, com vista a serem utilizados em áreas de aplicação específica, vários fabricantes desenvolveram variantes do conceito originalmente desenvolvido. Actualmente existem no mercado versões do 8051 com características adicionais tais como: mais linhas entrada/saída, interface de comunicação, memória extra e conversores A/D, entre outras. Além disso, existem também variantes CMOS com reduzido consumo de energia. Assim, existem versões adaptadas a uma gama ampla de aplicações. O coração do nó *Host* da rede de actuação é o microcontrolador P80C592 da Philips o qual tem como CPU o 80C51.

#### 5.1.1.1.1 Características do microcontrolador

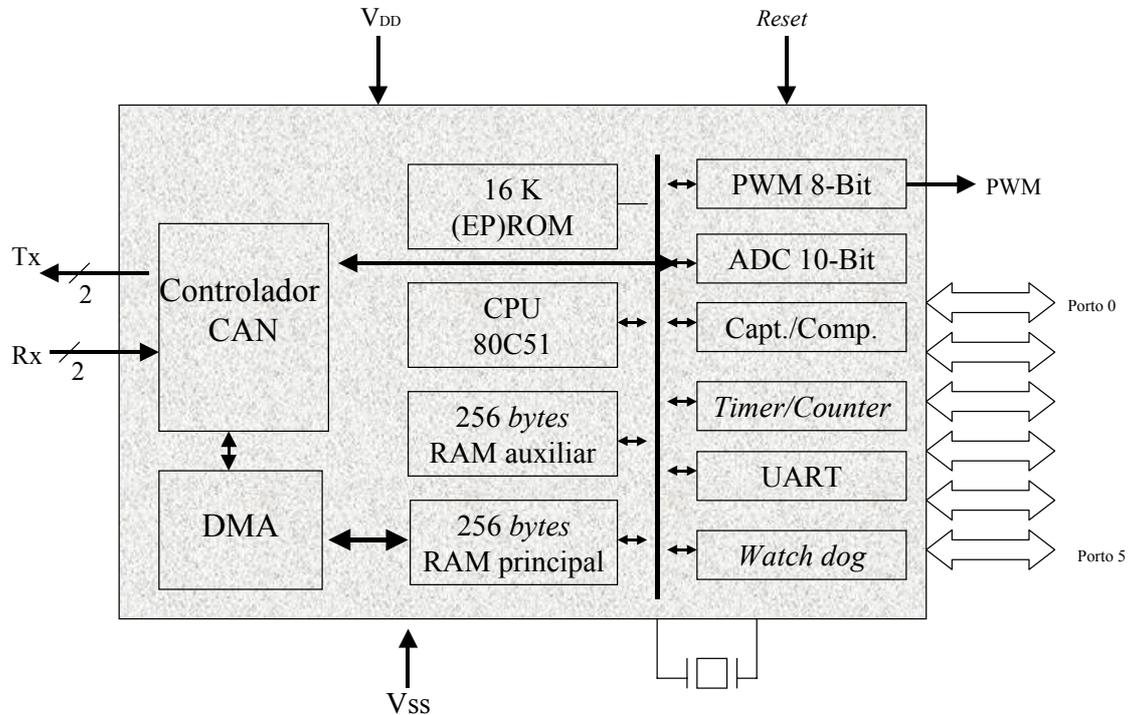
O nó *Host*, conforme foi referido, possui como unidade principal um microcontrolador, do qual se utilizaram várias características para realizar acções de actuação e controlo, utilizando o protocolo CAN nas comunicações. De entre as características do microcontrolador utilizado [37] destacam-se as seguintes:

- Unidade central de processamento 80C51;
- Um controlador CAN interno;
- Uma *universal asynchronous receiver-transmitter* (UART) *full-duplex*;
- Compatível com o *standard* 80C51;
- Duas saídas PWM com 8 *bits* de resolução;
- 2x256 bytes de RAM interna expansível externamente até 64 Kbytes;
- 16 Kbytes de ROM interna expansível externamente até 64 Kbytes;
- Um conversor A/D de 10 *bits* com 8 entradas analógicas multiplexadas;
- Um *watch dog timer*;
- Dois *timers/counters standard* de 16 *bits*;
- Quinze fontes de interrupção com dois níveis de prioridade;

- Cinco portos E/S de 8 *bits* e um porto de 8 *bits* partilhado com entradas analógicas (Porto 5).

A Figura 5.2 contém o diagrama de blocos do microcontrolador P8xC592.

**Figura 5.2- Diagrama de blocos do microcontrolador P8xC592.**



Além das características referidas anteriormente e do reduzido consumo de energia inerente à tecnologia CMOS, este microcontrolador permite ainda três modos de funcionamento, para reduzir ainda mais o consumo:

- Modo de espera (*Idle*)

Ao ser colocado neste modo de funcionamento através de escrita no registo próprio, *Power Control Register* (PCON) [37], os efeitos no microcontrolador são os seguintes:

- CPU é desactivado;
- *Timer 2* é parado e zerado;
- As saídas PWM são colocadas a zero;
- Conversor A/D é interrompido se estiver em progresso.

O microcontrolador abandona este modo de funcionamento quando for:

- Assinalada qualquer fonte de interrupção;
- Feito *reset* externo;
- Provocado *reset* pelo *watch dog timer*.

- Modo de desligado (*power-down*)

Ao ser colocado neste modo de funcionamento o microcontrolador é totalmente desactivado.

O microcontrolador abandona o estado de desligado quando for detectada actividade no barramento CAN ou feito pedido de *reset* externo.

- Modo adormecido (*sleep*)

Ao ser colocado neste modo de funcionamento o controlador CAN é desligado.

O microcontrolador abandona este modo, “acorda”, quando for detectada actividade no barramento CAN ou por *software* [37] através de escrita em registo próprio (PCON).

### 5.1.1.1.2 Expansão do microprocessador

A interface do microprocessador com o exterior é garantida por um barramento de 8 *bits* de dados (Porto 0) e um barramento de 16 *bits* (Portos 0 e 2) de endereços. O sinal de activação *Address Latch Enable* (ALE), é utilizado para distinguir ao nível do Porto 0 a informação de dados da informação de endereços, multiplexados. Quando o microprocessador pede acesso à memória externa de programa, a linha de controlo *Program Strobe Enable* (PSEN) é activada. As linhas de leitura (RD) e escrita (WR) são utilizadas para aceder à memória externa de dados.

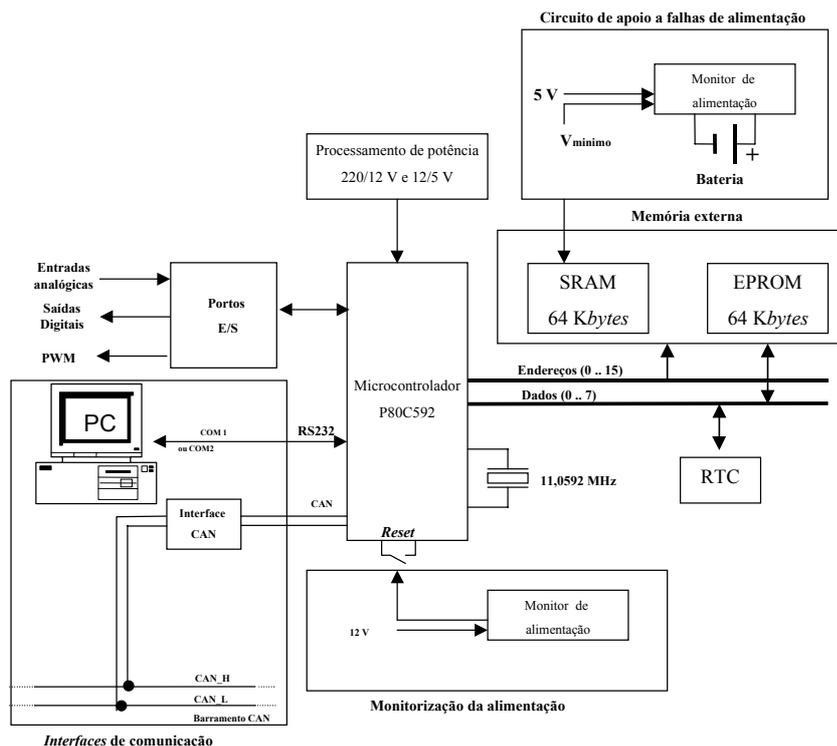
Todas as funções (portos E/S, portos série, registos de captura/comparação e conversor A/D) são geridas através de registos de funções especiais, *Special Function Register* (SFR). Esta solução flexível também é utilizada no 8051, embora com menos possibilidades. Finalmente vários registos SFR, tal como ponteiro de pilha, *Stack Pointer* (SP), ponteiro de dados, *Data Pointer* (DPTR), e outros são utilizados para coordenar a execução do programa.

### 5.1.1.1.3 Aspectos práticos

Na Figura 5.3 está representado o diagrama de blocos do circuito (*Host*) e na Figura 5.4 (página 71-A) o esquema completo do circuito implementado. Além do microcontrolador [37] e da memória, existe um relógio de tempo real, um circuito de reposição a zero (*reset*), uma interface CAN e uma interface RS-232. Foi projectado também um circuito de apoio, o qual possui uma pilha de lítio para garantir que os dados armazenados não se percam, quando a tensão de alimentação for desligada. Tanto a EPROM como a RAM externa possuem 64 Kbytes de capacidade, suficientes para a maioria das aplicações em causa.

Quanto à memória podem ser utilizados os 64 Kbytes de memória do programa bem como toda a memória de dados, sendo a selecção da capacidade pretendida, feita através de *jumpers* que inibem ou não linhas de endereço (AD13, AD14 e AD15).

Refira-se também que todos os portos E/S que não possuam função definida, nomeadamente os portos 4 e 5 na totalidade bem como as saídas PWM, estão acessíveis do exterior sendo possível a sua posterior utilização.



**Figura 5.3 — Diagrama de blocos da placa *Host*.**

O microcontrolador, conforme foi anteriormente referido, é o bloco central do nó *Host* interagindo com todos os outros blocos que compõem o nó, por isso, ao serem explicados os restantes blocos, será implicitamente explicado o funcionamento do microcontrolador.

Nas seguintes secções será exposto cada um dos restantes blocos que compõem a placa *Host*.

### 5.1.1.2 REAL TIME CLOCK (RTC)

Este bloco, importante em muitos sistemas de controlo, é responsável, neste nó da rede de actuação, pelas interrupções necessárias para calibrar a duração do tempo de *bit* dos SLIOs presentes na rede e pelo tempo de actuação, se necessário. Para desempenhar estas tarefas, o circuito integrado utilizado é o DS12887 [39] o qual foi programado para gerar interrupções com intervalos regulares de 62,5 ms, estando este período relacionado com requisitos de sincronismo dos SLIOs [33] conforme será explicado no capítulo 6.

O RTC utilizado é não volátil por um tempo de 10 anos, na ausência total de alimentação, pelo que não necessita de um circuito de *backup*. Este RTC possui também uma zona de memória não volátil de 114 bytes, que pode ser utilizada para guardar configurações ou mesmo para que o microcontrolador a utilize quando receber uma interrupção correspondente a falha de energia de alimentação (NMI), isto é, permite que nela se guarde o estado das variáveis, até que as condições sejam repostas. A selecção do RTC para alteração ou consulta de valores nele residentes é feita pelo microcontrolador através de um decodificador. Esta selecção é realizada por *software*, através do estado de um pino de saída (P1.0) do microcontrolador.

### 5.1.1.3 ALIMENTAÇÃO DAS PLACAS

A alimentação das placas *Host*, *SLIO CAN* e *Slave Motor* é conseguida em duas etapas. Assim, numa primeira etapa são obtidos 12 V DC a partir da rede (220 V AC) de alimentação que fornece energia aos actuadores. O circuito que realiza esta primeira etapa consiste num transformador, cujo sinal de saída após rectificação, filtragem e regulação [40], é de 12 V. Numa segunda fase utilizando um regulador de tensão, LM7805, obtêm-se os 5 V necessários para alimentar as placas, conforme mostra a Figura 5.5.

Os 12 V são necessários para garantir o funcionamento de determinados circuitos de accionamento de actuadores, conforme oportunamente será feita referência na parte deste capítulo dedicada a esse tipo de circuitos. Por questões de segurança foi utilizado um fusível de 1 ampere no circuito de alimentação.

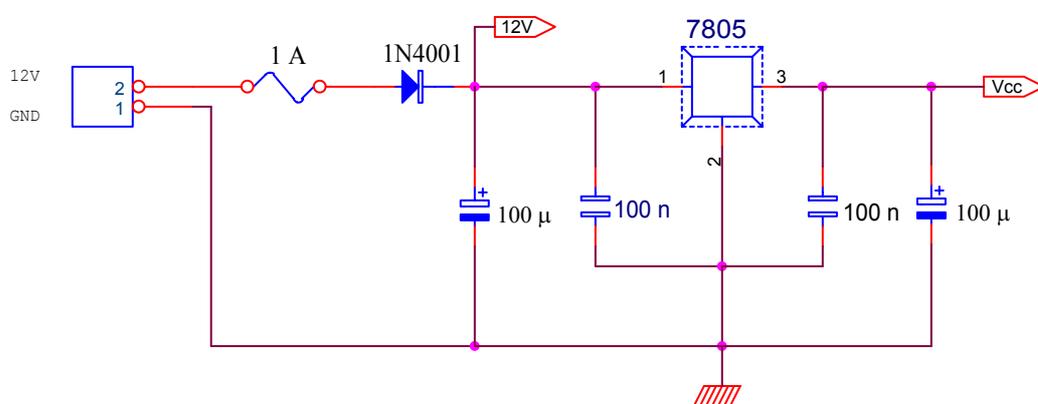


Figura 5.5 – Esquemático do circuito da fonte para 5 V DC.

Com a finalidade de precaver falhas de energia ou descida da tensão de alimentação, a níveis inaceitáveis ao correcto funcionamento dos circuitos, foi projectado um circuito para monitorização da tensão de alimentação, utilizando como base o circuito integrado DS1231 [41] conforme representado na Figura 5.6. O circuito integrado utilizado caracteriza-se por: sempre que a tensão de entrada que este está a monitorizar descer abaixo de determinado nível, ajustável, gera uma interrupção ( $\overline{INT0}$ ) ao microcontrolador para que este guarde o estado das variáveis nesse instante. Desta forma, quando a energia for reposta o microcontrolador pode continuar o seu funcionamento normal, com as variáveis no estado em que se encontravam quando a falha ocorreu. Para esta aplicação, o nível mínimo de tensão escolhido para monitorar foi de 8 V uma vez que, com esta tensão de entrada, o regulador de tensão utilizado [40] ainda funciona correctamente, conseguindo fornecer a tensão ( $V_{CC}=5$  V) esperada à saída. Assim, sempre que a tensão de alimentação for inferior a 8 V, nesta aplicação, o circuito integrado DS1231 gera uma interrupção  $\overline{INT0}$  (linha  $\overline{NMI}$ ) ao microcontrolador, para que este guarde as variáveis de estado, o ponteiro de dados e a hora do sistema. Se a tensão continuar a descer, quando no pino de entrada ( $V_{CC}$ ) for detectada uma tensão inferior a 4,75 V (porque  $TOL=GND$ ) [41], será gerado um sinal de *reset* (RST) ao microcontrolador.

O circuito integrado DS1231 reinicia automaticamente o microcontrolador logo que sejam restabelecidas as condições de alimentação. No entanto, durante o restabelecimento das condições de alimentação, o estado das saídas de *reset* (RST e



$\overline{RST}$ ) é mantido por um período mínimo de cerca de 150 ms. Este período mínimo serve para garantir que o processador esteja inactivo até que as condições sejam estáveis, protegendo assim o microcontrolador de fenómenos transitórios que por vezes se verificam durante o restabelecimento da alimentação.

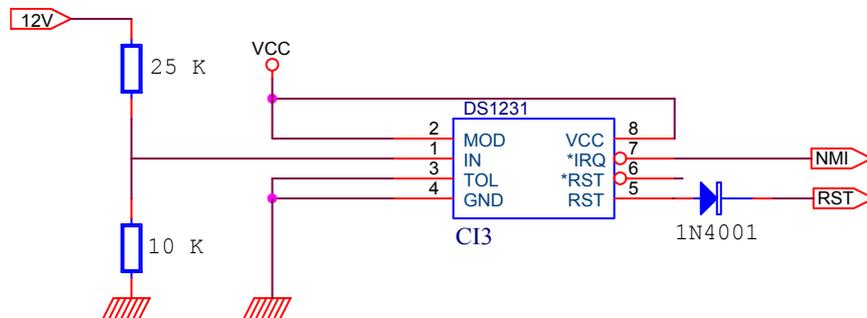


Figura 5.6 – Esquemático do circuito para monitorização da tensão de alimentação.

#### 5.1.1.4 INTERFACE RS-232

Para permitir a verificação, durante a fase de ensaios, da execução do programa residente neste nó, foi utilizada a UART que o microcontrolador possui para estabelecer comunicação via RS-232 deste com um PC.

Este bloco foi projectado e implementado à margem da placa *Host*. Foram no entanto deixadas nesta, acessíveis do exterior, as ligações necessárias à sua posterior implementação para que, caso seja necessário, o utilizador possa com a ajuda de um PC munido de *software* adequado, aceder a informação residente no microcontrolador. Para tornar possível esta opção programou-se a UART interna no modo 8-bits com *baud rate* de 4800 bit/s. Na implementação, em termos de *hardware* da interface RS-232, foi utilizado o *driver* MAX232 [42] conforme mostra o esquemático apresentado na Figura 5.7.

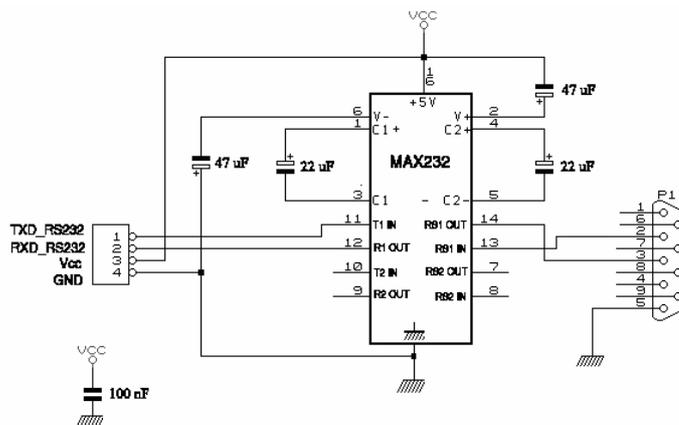


Figura 5.7 – Esquemático do circuito que permite comunicação RS-232 com o *Host*.

### 5.1.1.5 INTERFACE COM O BARRAMENTO CAN

O microcontrolador [37] utilizado nesta aplicação possui um controlador CAN interno. Assim, para estabelecer a comunicação por CAN é necessário implementar apenas o *transceiver*, isto é, a interface entre o controlador CAN e o meio de transmissão.

Foi utilizado o circuito integrado PCA82C250 [26] para implementar o *transceiver* CAN. Este circuito integrado apresenta algumas características importantes tais como:

- É totalmente compatível com o *standard* “ISO/DIS 11898”;
- Protege as linhas do barramento contra transitórios;
- Possui *slop control* para reduzir interferências de rádio frequência (RFI);
- Recepção diferencial com gama ampla de modo comum, permitindo elevada imunidade contra interferências electromagnéticas (EMI);
- Permite um número de ligações de, pelo menos, 110 nós;
- Oferece protecção térmica que é importante, principalmente, em situações de curto-circuito de uma das linhas do barramento;

Para além das características mencionadas anteriormente, o circuito integrado PCA82C250 permite três modos de funcionamento (elevada velocidade (*high speed*), *slope control* e *standby*), que são seleccionados neste caso por *hardware* e/ou *software*, através de alteração do estado de um dos pinos (RS) do mesmo. Nesta aplicação o modo escolhido foi o de alta velocidade por ser o aconselhado quando se utilizam os circuitos integrados SLIO [33]. Para o modo escolhido é recomendada [26], no caso de velocidades elevadas (ex. 1 Mbit/s), a utilização de um cabo blindado para evitar interferências de rádio frequência. Para velocidades menores, como no caso desta aplicação, ou comprimentos menores do barramento, um par entrelaçado ou par de fios paralelos pode ser utilizado como barramento sem necessidade de recorrer a blindagem.

A função do *transceiver* é a de permitir, ao controlador CAN do microcontrolador, transmissão e recepção diferencial ou seja:

- Converter sinais de saída do microcontrolador (CTX0 e CTX1) em níveis de tensão para as linhas do barramento;
- Converter níveis de tensão nas linhas do barramento para que sejam compatíveis com as entradas (CRX0 e CRX1) do microcontrolador P80C592.

As ligações físicas entre o microcontrolador e o *transceiver* são as seguintes:

#### Recepção

O receptor do microcontrolador consiste num comparador diferencial cujas duas linhas de entrada são CRX0 e CRX1.

Quando os sinais naquelas linhas tiverem valores tais que:

- $CRX0 > CRX1$  é interpretado como “recessivo” (nível lógico ‘1’);
- $CRX0 < CRX1$  é interpretado como “dominante” (nível lógico ‘0’).

Estas linhas foram ligadas ao *transceiver* conforme está representado no esquemático da Figura 5.8. Note-se que em vez de descodificar uma diferença entre dois sinais, as entradas do comparador diferencial descodificam o sinal de apenas uma linha (Rx/D)



NM27C512 [43]. Esta memória é seleccionada através do estado de três *jumper*s (S1, S2 e S3) que controlam, inibindo ou não, as linhas de endereço AD13, AD14 e AD15, conforme mostram a Tabela 5.1 e a Figura 5.9 (página 76-A).

<i>Jumper</i>	Memória disponível			
	64 KB	32 KB	16 KB	8 KB
S1 (AD13)	Ligado	Ligado	Ligado	Desligado
S2 (AD14)	Ligado	Ligado	Desligado	Desligado
S3 (AD15)	Ligado	Desligado	Desligado	Desligado

**Tabela 5.1 – Selecção de memória.**

O acesso à memória externa de programa por parte do microcontrolador é efectuado através da activação de uma das suas linhas de controlo (PSEN).

A memória externa de dados ou RAM é composta por dois circuitos integrados MT5C2568 [44] com capacidade de 32 Kbytes cada. A selecção entre memórias RAM ou RTC é feita através de um descodificador SN74HC139 [45], cujos pinos de entrada para selecção são ligados às linhas AD15 e IOM provenientes do microcontrolador. A activação do descodificador é feita pelos sinais das linhas (RD) e (WR), também provenientes do microcontrolador. As memórias RAM foram atribuídas às saídas mais significativas do descodificador, aproveitando o facto de por *default* os pinos do Porto 1 estarem no estado alto (IOM = '1'), ficando assim habilitada uma das memórias RAM (CI7 na Figura 5.9) utilizadas. A linha de endereço AD15 determina qual a RAM a ser acedida.

A multiplexagem de dados e endereços é efectuada através dos Portos 0 e 2, do microcontrolador do seguinte modo:

- Port0[0..7] – oito *bits* menos significativos do endereço e partilha do barramento de dados. A multiplexagem é feita utilizando uma *latch* SN74HC573 [46], que é controlada pelo pino de saída ALE do microcontrolador. Este sinal é gerado nos ciclos de leitura/escrita;
- Porto2[0..7] – oito *bits* mais significativos do endereço.

Para tornar as RAMs não voláteis, foi incluído um circuito para detecção de falhas de alimentação (CI9 e CI10 da Figura 5.9). Este circuito será explicado na próxima secção deste capítulo.

### 5.1.1.7 CIRCUITO PARA *BACKUP* DAS MEMÓRIAS

Este bloco do circuito utiliza o circuito integrado DS1210 [47] conforme mostra a Figura 5.9. O funcionamento resume-se ao seguinte: O circuito integrado DS1210 efectua uma monitorização constante da tensão de alimentação. Quando esta tensão de alimentação abandona determinados valores de tolerância, neste caso se for inferior ou igual a 4,5 volts porque foi ligada a entrada TOL a  $V_{CC}$ , um comparador de precisão detecta a falha de tensão de alimentação e inibe a saída  $\overline{CEO}$  (*chip enable*). Sempre que isto acontece a alimentação é comutada para a(s) bateria(s). Optou-se por utilizar apenas uma bateria de lítio recarregável. A bateria carrega através do circuito composto por uma resistência (R7 na Figura 5.9) e um diodo quando o circuito está em funcionamento

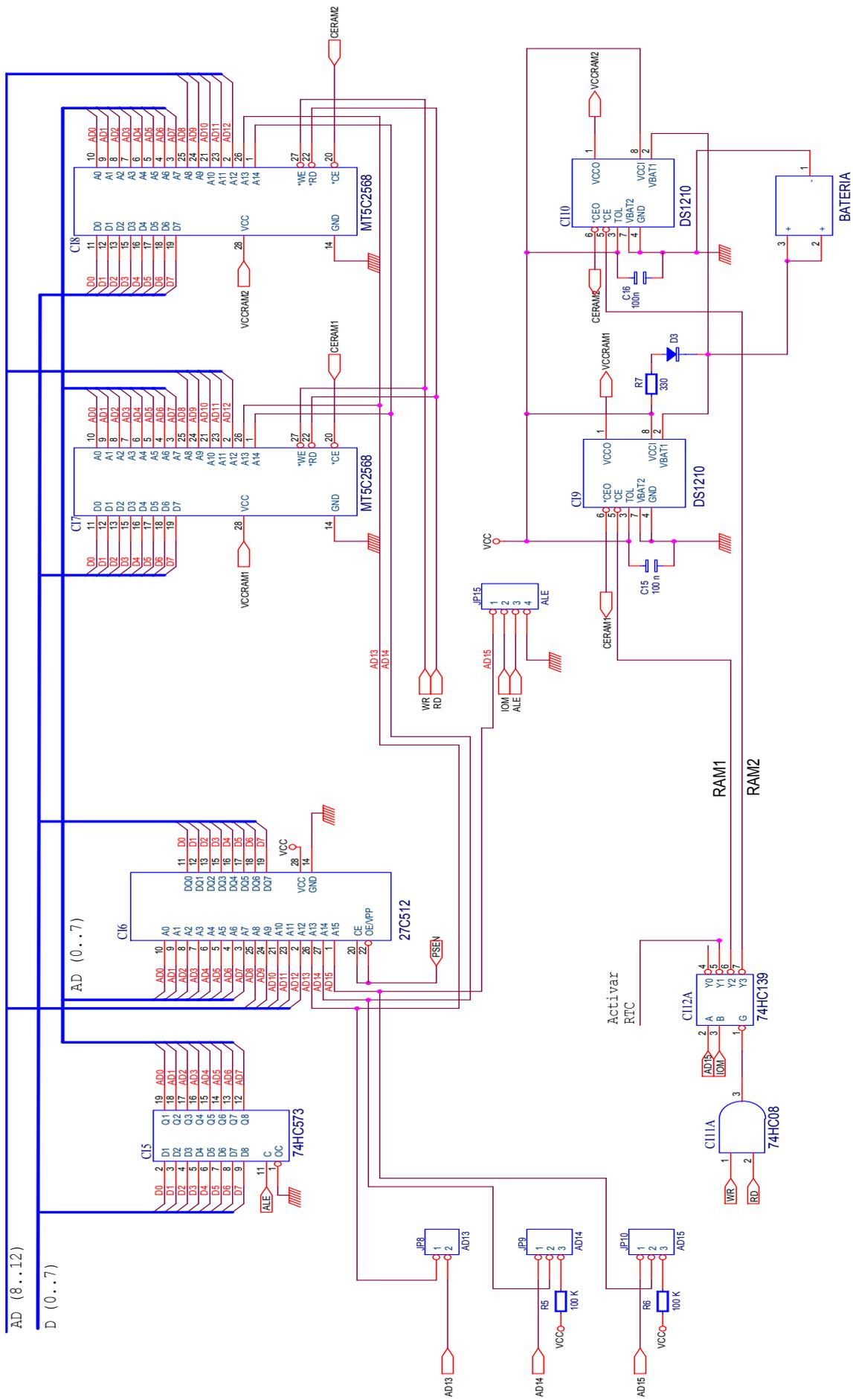


Figura 5.9 – Memória externa do nó Host.

normal, tendo o valor da resistência sido dimensionado de acordo com a corrente máxima que a bateria admite durante a carga.

Na implementação foram utilizados dois circuitos integrados DS1210 porque cada um tem capacidade de fornecer 50  $\mu\text{A}$ , e cada uma das memórias RAM utilizadas consome 40  $\mu\text{A}$ .

### 5.1.2 Nó *Slave Motor*

O nó *Slave Motor* utiliza também o microcontrolador P80C592 como elemento base e o mesmo diagrama de blocos que é utilizado na implementação do nó *Host*, conforme referido nas secções anteriores deste capítulo. A diferença entre estes dois nós da rede de actuação, reside apenas nas funções que cada um desempenha, isto é, no programa que cada um executa. Assim, o nó *Host* é responsável pelo sincronismo dos nós SLIO CAN e pela interface de mensagens entre esses nós e o PC-supervisor. Por outro lado o nó *Slave Motor* tem por função principal gerar numa das suas saídas PWM, uma onda sinusoidal sintetizada, com determinada frequência, através de modulação PWM para accionamento de um motor de indução. A frequência pretendida para a onda sinusoidal é um *set-point* transmitido pelo PC-supervisor e recebido pelo nó *Slave Motor*, via CAN, através de um *frame* de dados com formato que será referido numa das secções do capítulo 6. A onda sintetizada é posteriormente aplicada a um circuito próprio, para o controlo de velocidade de um motor de indução monofásico de um ventilador, conforme será oportunamente abordado com maior detalhe na secção 5.2.3.

Devido às semelhanças, em termos de *hardware*, existentes entre o nó *Slave Motor* e o nó *Host* abordar-se-á nas próximas secções as características do *hardware* que compõe as placas dos nós SLIO CAN.

### 5.1.3 Nós SLIO CAN

Conforme foi referido nos capítulos 3 e 4, as unidades designadas por SLIO CAN possuem como elemento principal o circuito integrado P82C150 da Philips.

A escolha pela utilização deste circuito integrado deve-se ao seu baixo preço, ao facto de ser considerado ideal para activar sensores e/ou actuadores remotos e por possuir a capacidade de recolher ou facultar dados digitais e/ou analógicos.

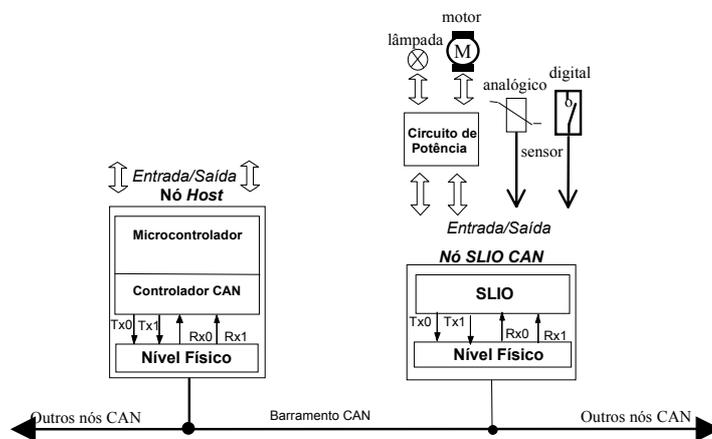
O SLIO é no entanto um circuito integrado com pouca “autonomia”, tendo de ser controlado e sincronizado por uma unidade “inteligente” que, no caso desta aplicação, é o nó *Host*. Os SLIOs podem, desta forma, ser vistos como unidades remotas do microcontrolador existente na placa do nó *Host*.

Além das características do SLIO P82C150, já referidas, destacam-se para esta aplicação as seguintes:

- Permite 16 portos de E/S configuráveis como analógicos ou digitais;
- Possui um oscilador de relógio local totalmente integrado (não requer cristal);
- Permite a configuração de 16 pinos, sendo cada pino configurável via barramento CAN (nomeadamente direcção do porto (entrada/saída) e o modo do porto (*polling*, *event*, etc));

- Permite duas saídas quase-analógicas com resolução de 10 *bits* (Anexo 2);
- Possui um conversor A/D de 10 *bits*, permitindo até seis canais analógicos de entrada multiplexados;
- Possui dois comparadores analógicos;
- Possui detecção automática da taxa de transmissão;
- Permite taxas de transmissão desde 20 Kbit/s até 125 Kbit/s utilizando o oscilador interno podendo atingir os 250 Kbit/s com recurso a oscilador externo;
- São permitidos até 16 nós SLIO por barramento CAN;
- Funções controladas por um único nó “inteligente” (neste caso o *Host*);
- Segue as versões 2.0 A e B (passivo) da especificação CAN [27];
- Permite o modo de *sleep* com *wake-up* via barramento CAN, para reduzir o consumo de energia.

Na Figura 5.10 estão representados, de forma genérica, os nós designados por SLIO CAN, os quais são responsáveis por interpretar comandos, mensagens CAN, enviados pelo *Host*, e, em função destes, gerar sinais nos portos de saída, para comandar circuitos de potência que dependem do tipo de actuador que se pretenda accionar. O SLIO permite também tarefas de aquisição utilizando para tal portos E/S os quais permitem entradas digitais (num máximo de 16), bem como ler até 6 entradas analógicas.



**Figura 5.10 – Diagrama de blocos do sistema SLIO CAN.**

Assim, este circuito integrado pode actuar como uma *gateway* discreta de entrada/saída numa rede CAN, capaz de transformar mensagens em sinais de saída digitais ou analógicos, bem como ler o estado de pinos de E/S digitais ou analógicos e transmitir os dados adquiridos como mensagens, possibilitando assim o conhecimento desses dados a outros nós da rede, designadamente ao nó *Host*.

A arquitectura implementada no que diz respeito à estrutura SLIO CAN está representada no diagrama de blocos da Figura 5.10, onde é possível observar a ausência do “dispendioso” microcontrolador nos nós SLIO CAN, os quais têm como unidade principal o circuito integrado SLIO, cujo custo é consideravelmente inferior ao do microcontrolador.

O SLIO possui também a característica de ter quatro *bits* do identificador programáveis através de um número igual de pinos E/S (pinos 1, 2, 27 e 28) os quais estando ligados a 0 ou 5 V ficam com o valor lógico ‘0’ ou ‘1’, respectivamente. O estado destes pinos é guardado no fim do *reset*, podendo de seguida ser utilizados como portos de E/S normais. Cada SLIO deve possuir um identificador único, o que faz que, com quatro *bits* configuráveis seja permitida a existência no máximo de 16 SLIOs por *Host* e por rede CAN, conforme foi referido na secção 3.16.3. A Figura 5.11 representa o formato do campo de arbitragem de uma *frame* CAN relacionada com os nós SLIO CAN.

Campo de arbitragem											
Identificador											RTR
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
0	1	P3	1	0	P2	P1	P0	1	0	DIR	X

**Figura 5.11 – Campo de arbitragem das *frames* do SLIO P82C150.**

Outra característica importante é a ausência do cristal nas placas dos nós SLIO CAN, este facto deve-se à capacidade que este integrado possui de auto-calibração a partir de mensagens transmitidas pelo nó *Host*, via barramento CAN, reduzindo o custo destes nós e por consequência da rede proposta.

Em resumo é possível afirmar que num sistema SLIO CAN:

## O SLIO

- É um dispositivo de custo reduzido que permite controlar portos de E/S numa rede CAN;
- Não possui a “inteligência” de um microcontrolador;
- Tem de ser controlado por um nó *Host*;
- Permite taxas de transmissão de [31]:
  - 80 m a 125 Kbit/s;
  - 120 m a 100 Kbit/s;
  - 300 m a 50 Kbit/s;
  - 850 m a 20 Kbit/s.

## O *Host*

- Controla o conteúdo dos registos do(s) SLIO(s);
- Envia mensagens de calibração para o(s) SLIO(s).

As mensagens de calibração são *frames* CAN com um formato próprio (que será explicado no capítulo 6) e que mantêm os SLIOs “acordados” ou seja prontos a comunicar.

Foi preocupação em todo o *hardware* desenvolvido nesta implementação, possibilitar o acesso do exterior a todos os portos, para que estes possam ser utilizados no comando de circuitos de accionamento, que servem de interface com os actuadores ou seja entre a

rede de comando e os actuadores. Os circuitos de accionamento serão abordados na secção 5.2 deste capítulo, que lhes é dedicada.

Quanto à implementação das placas SLIO CAN, cujo esquemático de encontra representado pela Figura 5.12, destacam-se nestas, como já foi referido, a presença do circuito integrado P82C150 bem como do *transceiver* P82C250, que tal como na placa *Host* tem a função de interpretar os sinais CAN existentes no barramento e disponibilizá-los ao controlador CAN do SLIO.

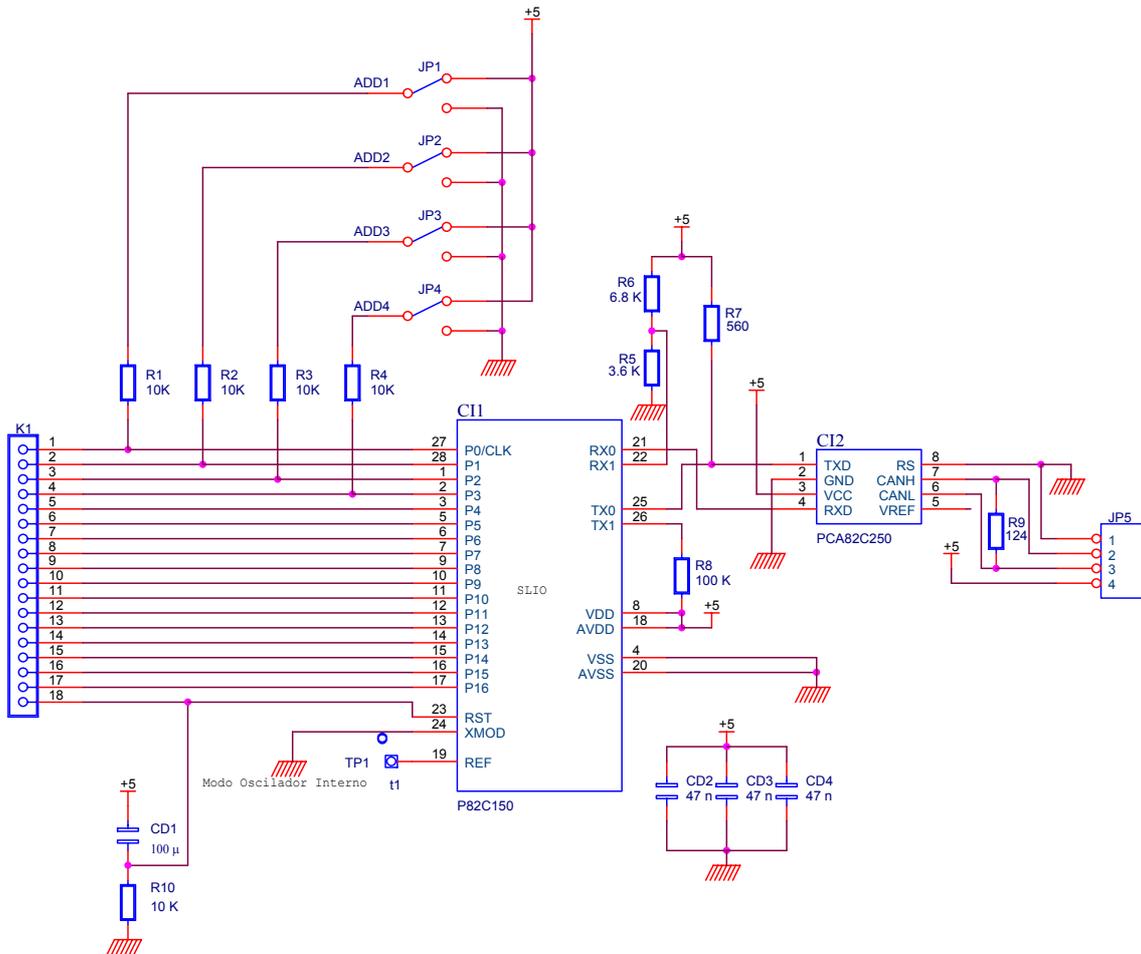


Figura 5.12 – Esquemático do circuito da placa SLIO CAN.

### 5.1.4 Circuito para aquisição de valores de temperatura

Para testar a funcionalidade do sistema proposto foi também necessário projectar um circuito que permitisse ler a temperatura ambiente. Para tal recorreu-se ao sensor de temperatura LM35DT [48], o qual apresenta saída em tensão proporcional à temperatura, sendo a relação de transferência 10 mV/°C. O circuito implementado é apresentado na Figura 5.13.



A presença do diodo D1 garante que a tensão de controlo é aplicada com polaridade correcta, enquanto que a presença do diodo D2 se prende com a protecção do relé.

É comum utilizar solenóides, relés e outros dispositivos, que utilizam energia magnética armazenada, para comandar válvulas, motores e outros actuadores. Quando estes dispositivos são comandados por tensões DC, criam problemas, especialmente os de estado sólido, porque quando estes são desligados a energia magnética armazenada tem de se libertar. Geralmente esta toma a forma de um pulso significativo de energia visto pelo dispositivo de comutação de estado sólido. Estes pulsos, “*inductive kick back*”, podem apresentar picos de tensão superiores aos da fonte de alimentação DC e provocar danos aos dispositivos de comutação de estado sólido [49]. Fabricantes de relés e solenóides recomendam protecção adicional, designada por “diodos de comutação”, aplicados perto dos referidos dispositivos.

Uma das principais vantagens dos relés de estado sólido é a sua compatibilidade com lógica de nível baixo. Qualquer porta lógica, isolada ou não, capaz de fornecer corrente necessária e tensão dentro da gama de dissipação de potência máxima pode ser utilizada para controlar relés de estado sólido.

Nesta aplicação foram utilizados os sinais com níveis TTL das saídas dos nós SLIO CAN para, através de relés de estado sólido, testar o accionamento, pelo sistema, de diversos actuadores em situações de comando liga/desliga, tendo sido obtidos resultados positivos. Os testes consistiram no accionamento de: ventilador, solenóide, electroválvula e lâmpadas.

## 5.2.2 Circuito de accionamento para actuadores DC

Para controlo liga/desliga de cargas DC utilizou-se o MOSFET de potência BUK101-50GS [50], que permite uma solução alternativa à utilização de relés para cargas com tensões até 50 V e correntes até 20 A como é o caso de alguns motores, solenóides e electroválvulas. O MOSFET utilizado possui nas suas propriedades as capacidades de protecção contra sobreaquecimento e contra curto-circuito, bem como permitir sinais de comando de 5 V. O esquemático do circuito implementado está ilustrado na Figura 5.15.

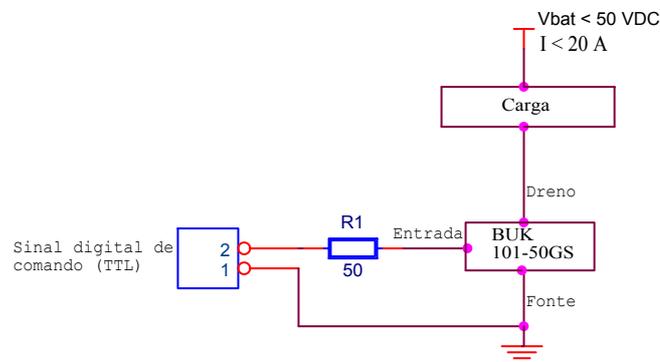


Figura 5.15- Esquemático de uma aplicação utilizando o transístor BUK101-50GS.

No circuito representado na Figura 5.15, quando o sinal digital de comando estiver ao nível lógico ‘1’ (5 V) é aplicada à carga a tensão fornecida pela bateria (liga) e quando estiver ao nível lógico ‘0’ (0 V) é retirada à carga a tensão da bateria (desliga).

## 5.2.3 Circuito para accionamento de um motor de indução

### 5.2.3.1 INTRODUÇÃO

Apesar da variedade de motores existentes no mercado, existem três tipos clássicos de motores: o de corrente contínua com comutadores e os motores de corrente alternada síncrono e assíncrono, os quais por sua vez englobam diversos tipos de motores de acordo com características próprias de cada um. A Figura 5.16 mostra uma possível classificação dos motores eléctricos [51].

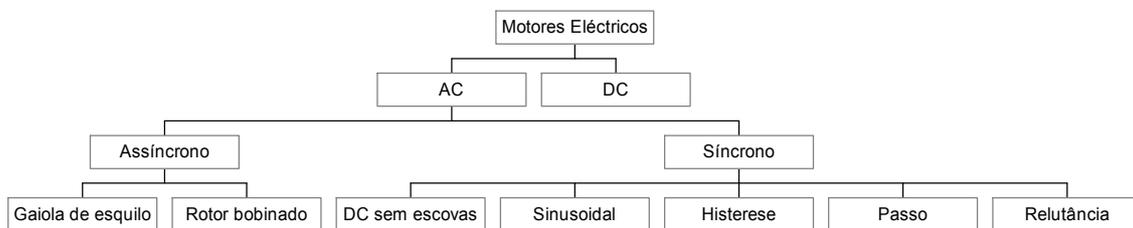


Figura 5.16 – Classificação de motores eléctricos.

A maioria dos accionamentos na indústria utilizam motores de indução com rotor em gaiola de esquilo, devido à sua robustez, baixo custo e ao valor elevado da relação potência/peso. O facto de não exigir manutenção, nem possuir colectores com escovas ou qualquer outro tipo de contactos, faz com que para algumas aplicações ele seja a única escolha possível. São os casos em que o motor funciona em locais dificilmente acessíveis ou em ambientes agressivos, como por exemplo, nas indústrias envolvendo processos químicos. No entanto, a utilização de motores de indução também apresenta inconvenientes, estes dizem respeito à dificuldade de controlo, devido ao seu complexo modelo matemático, ao comportamento não linear e à variação de alguns parâmetros eléctricos com a temperatura.

Quando alimentado directamente a partir da rede, com os valores nominais de tensão e frequência, a velocidade do motor de indução mantém-se praticamente constante. Por isso, durante muitos anos, ele foi tradicionalmente utilizado em aplicações requerendo velocidade fixa. São excepção os casos onde a obtenção de alguns valores discretos de velocidade que se conseguem alterando o número de pares de pólos do motor por processos electromecânicos, aplicações onde pequenas variações de velocidade são obtidas ajustando a tensão estatórica, ou a utilização do motor de indução de rotor bobinado, onde a velocidade é variada inserindo resistências de diferente valor em série com o rotor.

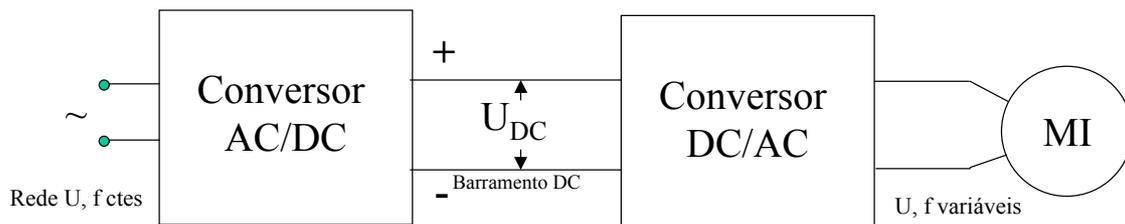
Para se alterar eficientemente e numa larga gama a velocidade do motor de indução, é indispensável actuar simultaneamente sobre a tensão e frequência de alimentação, o que nunca foi um processo simples, quer pela complexidade da electrónica envolvida, quer por se tratar do controlo dum estrutura multivariável e não linear [52].

A necessidade sentida pela indústria em aumentar a sua produtividade e a qualidade dos seus produtos e a preocupação crescente com a economia de energia levaram a que, sobretudo nas últimas décadas, se tenha feito um grande esforço no desenvolvimento de sistemas de accionamento baseados na utilização de máquinas de corrente alternada.

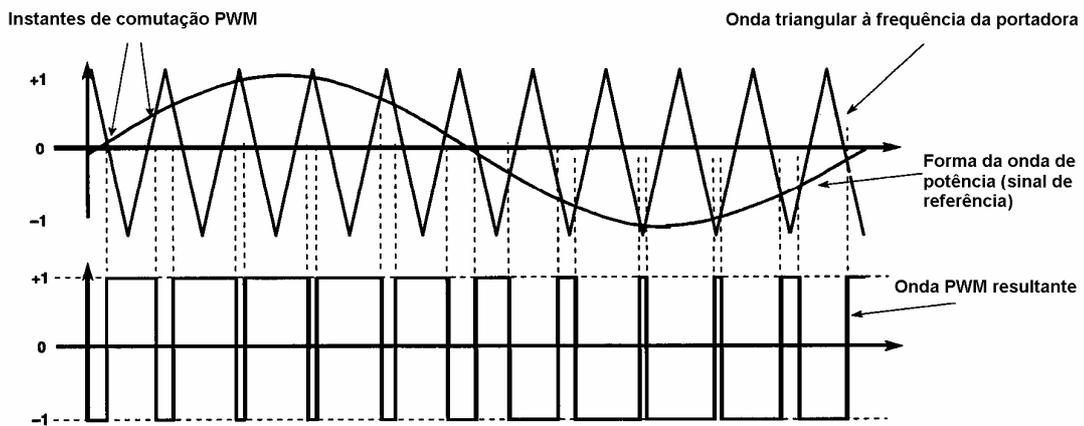
Esse desenvolvimento verificou-se em várias áreas: na microelectrónica, permitindo a utilização de *software* em vez de *hardware* na solução de problemas cada vez mais complicados por um preço aceitável; na tecnologia dos semicondutores, com o aperfeiçoamento dos componentes de potência já existentes e com o aparecimento de novos interruptores electrónicos, cada vez mais fáceis de comandar e com melhores desempenhos em termos de velocidade de comutação e eficiência; no controlo dos conversores de potência, nomeadamente, com o desenvolvimento das técnicas de modulação de largura de impulso; no estudo de novos modelos para o motor, tendo em vista a simplificação do seu controlo.

A topologia da maior parte dos sistemas de accionamento AC que utilizam conversores estáticos é a ilustrada na Figura 5.17. Excluem-se os cicloconversores ou conversores AC/AC, normalmente utilizados apenas para potências muito elevadas e para gamas de variação de velocidade restritas. Nos outros casos, a tensão alternada sinusoidal e com valores fixos de amplitude e frequência obtida a partir da rede é primeiro rectificada, obtendo-se um barramento DC. Segue-se o processo inverso, convertendo-se a tensão DC, na tensão de alimentação do motor, de amplitude e frequência variáveis.

**Figura 5.17 – Topologia dum sistema de accionamento AC.**

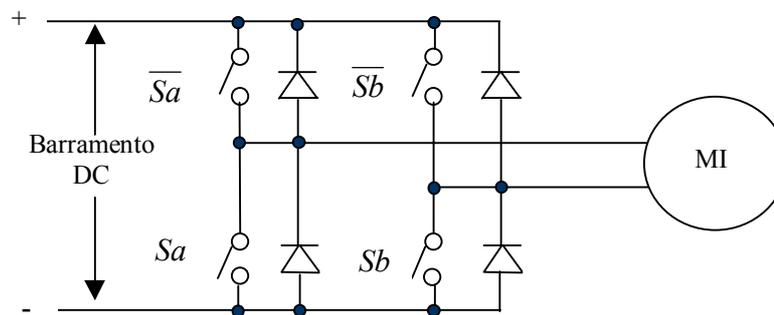


O ajuste de amplitude da tensão pode fazer-se ao nível do conversor AC/DC utilizando um rectificador controlado, ou ao nível dum terceiro conversor DC/DC (não representado na Figura 5.17), mas num sistema de accionamento moderno, tanto o valor da amplitude como o da frequência, são modificados no estágio de conversão DC/AC, através de um inversor e utilizando técnicas de modulação de largura de impulso (PWM). Em geral, os esquemas de PWM geram os padrões de comutação comparando a onda sinusoidal de uma fase (monofásico) ou das três fases (trifásico) com uma portadora triangular, Figura 5.18. A forma de onda produzida não é sinusoidal mas sim comutada. Contudo, utilizando as frequências de comutação (relativamente elevadas) admitidas pelos interruptores electrónicos disponíveis, e/ou recorrendo a técnicas de optimização dos valores dos ângulos de comutação, é possível conseguir que o conteúdo harmónico das ondas sintetizadas a partir do barramento DC seja pequeno, de tal modo que as perdas no motor são mínimas e o binário não é significativamente perturbado[52].



**Figura 5.18 – Modulação PWM de uma onda sinusoidal usando uma portadora triangular.**

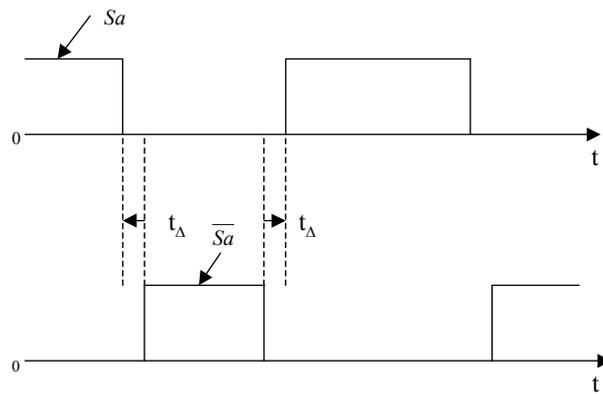
Para um melhor entendimento do processo de controlo do motor utilizando um inversor monofásico em ponte completa, ilustrado na Figura 5.19, bem como para representar os estados de comutação do inversor define-se a função  $S_a$  para a fase A como sendo:  $S_a = 0$  quando o semicondutor de potência inferior do braço ou fase A estiver “aberto”, e  $S_a = 1$  quando o mesmo semicondutor estiver “fechado”. Para a fase B as definições são idênticas.



**Figura 5.19 – Inversor monofásico em ponte completa.**

Os sinais ( $S_a$  e  $S_b$ ) que controlam os interruptores electrónicos inferiores de uma fase, são complementares dos sinais ( $\overline{S_a}$  e  $\overline{S_b}$ ) que controlam os interruptores electrónicos superiores da mesma fase, com adição de um “tempo morto” ( $t_{\Delta}$ ), conforme ilustrado na Figura 5.20 para o caso da fase A.

Tempo morto é o nome atribuído à diferença de tempos entre as comutações dos interruptores electrónicos inferior e superior numa das fases. Os dois interruptores electrónicos de cada fase nunca devem conduzir simultaneamente pois isso provocaria um curto-circuito do barramento DC que alimenta o inversor. Assim, o objectivo do tempo morto é o de proteger os interruptores electrónicos durante a comutação evitando simultaneidade de condução e por consequência elevadas correntes transitórias [53].



**Figura 5.20 – Efeito do “tempo morto” ( $t_{\Delta}$ ).**

A utilização dum inversor, do tipo do ilustrado na Figura 5.19 para o caso de uma ponte monofásica, comandado por técnicas de PWM é a solução que actualmente se aceita como *standard*, possuindo, entre outras, as seguintes vantagens:

- É a configuração mais económica, já que usa apenas dois conversores e só um deles é controlado (o inversor);
- O conversor do lado da rede é um simples rectificador, implementado com díodos e garantindo que o factor de potência à entrada é máximo;
- Um só barramento DC pode ser comum a vários inversores, minimizando-se assim o número de componentes de potência, a cablagem, o número de dispositivos de protecção, de conexão etc.

Os interruptores electrónicos utilizados na construção dos inversores podem ser de diferentes tipos em função das frequências de comutação, das potências, tensões e correntes envolvidas. Seguidamente referem-se as principais características de vários interruptores electrónicos utilizados na construção de inversores.

Assim, inicialmente a electrónica de potência foi dominada pelos díodos de silício e pelo tiristor. Estes componentes eram complementados pelos transístores bipolares mas apenas para potências muito pequenas.

Posteriormente houve um progresso evidente na tecnologia dos componentes de potência, o que facilitou imenso o projecto dos conversores em geral, e dos inversores em particular. Os transístores de potência aumentaram em capacidade e verificou-se o aparecimento de novos componentes como o gto (tirístores desligados pela *gate*), com possibilidade de interromper corrente e dispensando os circuitos de bloqueio forçado utilizados nos inversores a tirístores tradicionais. Transístores e gto's substituíram os tirístores em praticamente qualquer aplicação, se for excluído o controlo de potências muito elevadas. As perdas na comutação foram consideravelmente reduzidas e apesar do aumento do valor das frequências de comutação utilizadas. Tornou-se possível construir conversores muito robustos e com uma elevada relação KVA/volume.

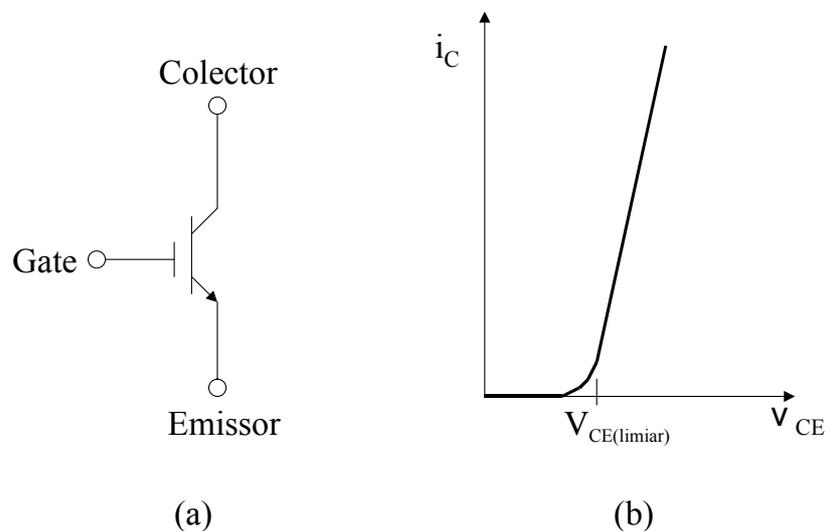
Os transístores bipolares podem ser utilizados para potências da ordem das centenas de KVA. Utilizando gtos é possível controlar potências da ordem dos vários MVA, existindo disponíveis comercialmente componentes para 3000A/4500V. Contudo, a sua utilização também é interessante para potências bastante inferiores.

Os transístores de potência de tecnologia unipolar (MOSFET, JFET, SIT e outros) também evoluíram. Os seus tempos de comutação são da ordem dos nanosegundos, pelo que podem ser utilizados com eficiência para frequências de comutação superiores a

100 KHz. Esta possibilidade não se traduz contudo em grandes vantagens no caso do accionamento de máquinas eléctricas, já que a maior parte dos inversores funcionam normalmente a frequências inferiores a 10 KHz. Por isso, e pelo facto de só se encontrarem disponíveis para potências pouco elevadas e valores de tensão limitados, os transístores MOS são adequados para outro tipo de aplicações, nomeadamente, fontes de alimentação comutadas. Apesar disso, pela simplicidade dos circuitos de comando, estes são por vezes utilizados para gerar ondas de PWM em inversores para pequenas potências e tensões DC inferiores a 500 V.

Os transístores bipolares e os MOSFETs possuem características que se complementam em alguns aspectos. Os transístores bipolares possuem baixas perdas de condução quando ligados, especialmente em dispositivos com elevadas tensões de bloqueio, mas têm tempos de comutação elevados, especialmente ao desligar. Os MOSFETs, por outro lado, podem ser ligados e desligados mais rapidamente, mas as perdas de condução quando ligados são maiores, especialmente em dispositivos para tensão de bloqueio elevadas. Estas observações levaram à tentativa de combinar os transístores bipolares e os MOSFETs num mesmo dispositivo, combinando as vantagens de ambos os tipos de dispositivos. Isto levou ao surgimento dos IGBTs (*Isolated Gate Bipolar Transistor*)[53].

Os IGBTs, cujo símbolo e curva característica estão representados na Figura 5.21, conseguiram sobrepor-se a outras tecnologias utilizadas em Electrónica de Potência na gama de tensões acima de 300 V e frequências de comutação até 20 kHz [54]. As vantagens técnicas dos IGBTs e o seu progressivo embaratecimento, fizeram destes semicondutores de potência componentes quase *standard* em andares de potência da ordem de 1 kW ou superiores.



**Figura 5.21 – Símbolo e característica (corrente - tensão) do IGBT: (a) símbolo de um IGBT canal-n; (b) característica de transferência.**

Actualmente existem módulos de IGBTs compostos por seis unidades para controlo de conversores de motores trifásicos, admitindo correntes que podem atingir 100 A por fase. Existem também módulos de IGBTs de potência para correntes de 1200 A e tensões inversas de 1700 a 3300 V.

No entanto, apesar do comando dos IGBTs ser simples, a preferência vem recaindo sobre a utilização de módulos incluindo funções de protecção no circuito de

accionamento, protegendo, desta forma, aqueles semicondutores de potência contra curto-circuitos e outras anomalias.

Das várias soluções existentes, o conceito de módulo “inteligente” tem vindo a ganhar cada vez maior importância. Estes são módulos de IGBTs que possuem, de forma integrada, funções de protecção bem como circuitos para comando de *gate*. Existe disponível no mercado uma gama considerável de módulos “inteligentes” de IGBTs. Estes módulos não são soluções completas porque necessitam de um número considerável de componentes externos requerendo nomeadamente, isolamento entre o circuito electrónico de comando e o andar de potência. Além disso, o projectista não pode influenciar as características de comutação dos módulos ou aceder aos terminais da *gate*, mesmo para efeitos de medição.

Os requisitos de isolamento, mencionados anteriormente, podem ser implementados de diferentes formas. A electrónica actual de comando com funções de protecção pode existir de forma integrada ou discreta. Os três componentes seguintes representam as soluções mais comuns de transmitir a informação de accionamento com isolamento eléctrico:

- Transformadores de impulsos;
- Optoacopladores;
- Fibras ópticas.

A solução utilizando transformadores de impulsos permite tempos de transição extremamente reduzidos, da ordem dos nanosegundos. Surgem dificuldades em elaborar uma solução que por um lado seja boa para várias frequências de comutação e ao mesmo tempo imune a valores elevados de  $dv/dt$  entre entrada e saída.

Uma solução utilizando optoacopladores é atractiva devido à sua simplicidade. A maior dificuldade que esta solução apresenta é a de encontrar uma forma de transferir rapidamente a informação de accionamento, assegurando estabilidade de  $dv/dt$ . Apenas alguns optoacopladores conseguem satisfazer estes requisitos de forma económica. Os optoacopladores são em princípio mais lentos do que os transformadores de impulsos. [54].

As fibras ópticas oferecem um grande número de vantagens tais como: ausência de acoplamento capacitivo entre entrada e saída; ausência de sensibilidade a  $dv/dt$ ; grande largura de banda; linhas compridas de accionamento sem interferência de acoplamento; isolamento de qualquer tensão e fácil aplicação. Algumas das suas desvantagens são: instabilidade; sensibilidade à humidade e custos relativamente elevados. Por estas razões, esta solução é utilizada em casos de tensões de operação, tensões de isolamento e potências elevadas e onde sejam necessárias distâncias compridas entre os andares de controlo e de potência.

Quanto às técnicas de controlo de sistemas de accionamento para motores, a escolha de arquitecturas baseadas na utilização de microprocessadores/microcontroladores e técnicas digitais, dum modo geral, é óbvia por razões que são conhecidas:

- Possibilidade de utilização de “*hardware standard*”;
- Versatilidade: muitas vezes, apenas o *software* é alterado;
- O número de placas de controlo é minimizado;
- A fiabilidade aumenta porque o número total de componentes diminui;

- A transmissão digital de sinal permite minimizar a cablagem e é extremamente imune ao ruído;
- É relativamente simples implementar por *software* algoritmos de controlo complexos;
- É fácil ajustar parâmetros, e a estabilidade dos seus valores é óptima;
- É fácil implementar esquemas de monitorização e diagnóstico.

Seguidamente será descrita a implementação, em termos de *hardware*, realizada no âmbito desta dissertação.

### 5.2.3.2 IMPLEMENTAÇÃO

Para este trabalho foi implementado um circuito para o controlo de velocidade de um motor de indução monofásico utilizável num ventilador de uma estufa. Pretende-se assim uma solução mais sofisticada do que a geralmente utilizada, ou seja, o controlo liga/desliga.

O diagrama de blocos do sistema implementado está representado na Figura 5.22.

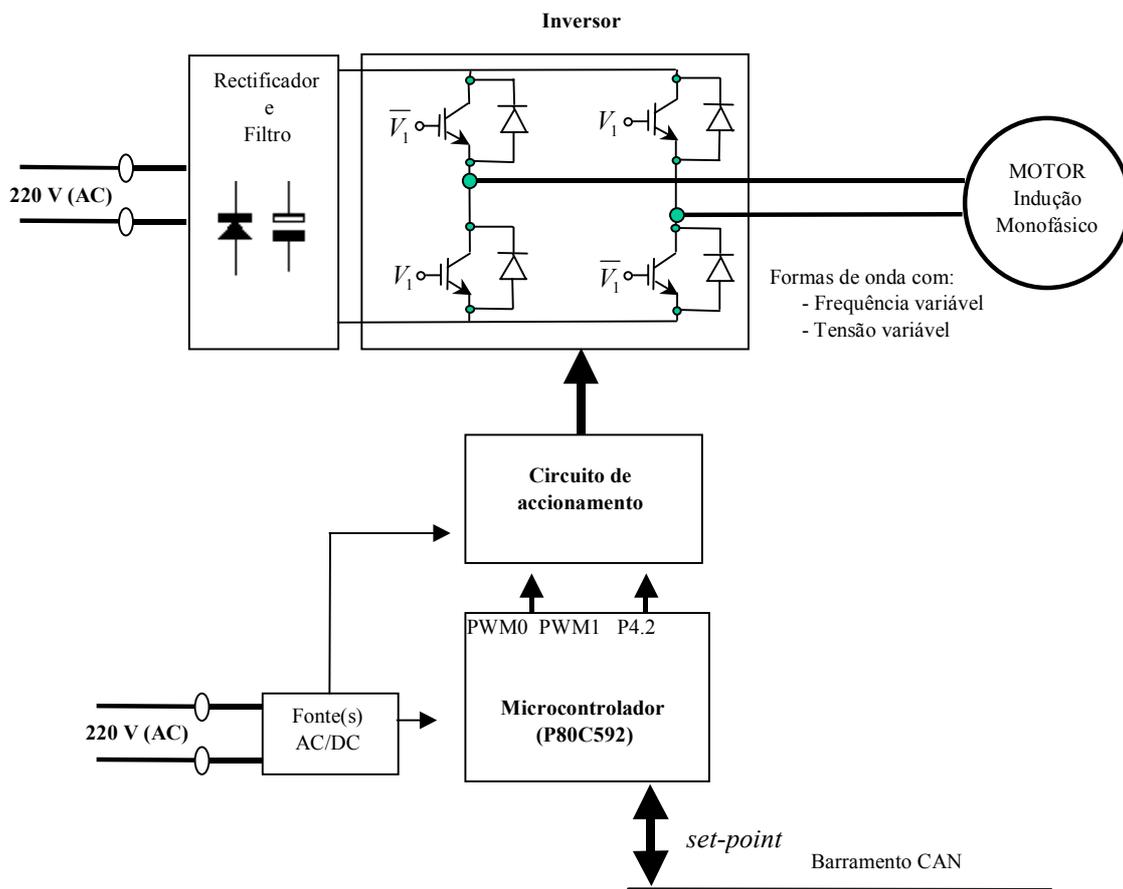


Figura 5.22 – Diagrama de blocos do sistema implementado.

Para gerar as tensões de fase com amplitude e frequência variável recorreu-se a um inversor, baseado em IGBTs. Como o controlo do motor se destina ao controlo da

velocidade de um ventilador, optou-se por um controlo em malha aberta evitando a necessidade de sensores, o que permite embaratecer e simplificar o referido controlo. Assim o controlador não possui entradas exceptuando o ajuste, *set-point*, da velocidade pretendida, que é recebido via barramento CAN.

O inversor foi implementado utilizando o módulo de potência “inteligente” PM20CSJ060 da Mitsubishi [55]. Nesta aplicação não é utilizado um dos braços deste módulo em virtude do motor controlado ser monofásico. A escolha do motor monofásico deveu-se a duas razões em particular: a primeira prende-se com o facto de os motores utilizados neste tipo de aplicação terem pouca potência; a segunda razão prende-se com o facto de o microcontrolador utilizado, P80C592, possuir apenas duas saídas PWM o que obrigaria à utilização, por exemplo, de um controlador PWM extra, como o SA828 [56], para gerar as três fases (desfasadas em 120°).

Conforme ilustra a Figura 5.22, existem várias componentes desta implementação que executam funções específicas. O esquemático do circuito implementado está representado na Figura 5.23 (página 90-A).

A seguir, é explicado cada um dos blocos constituintes do circuito de accionamento.

## **Rectificador e filtro**

Este bloco é responsável por converter a tensão alternada da rede com características conhecidas (220 V, 50 Hz) numa tensão contínua que, por controlo da comutação dos dispositivos que compõem o inversor, é modificada em frequência e amplitude, permitindo atingir o objectivo pretendido que é o de variar a velocidade do motor. Para implementar este bloco utilizou-se a solução tradicional que consiste em rectificar a tensão de entrada através de uma ponte rectificadora a quatro díodos, utilizando, de seguida, um condensador “tanque” ou de armazenamento, obtendo desta forma uma tensão contínua que em vazio tem um valor próximo de  $\sqrt{2} * 220$  V.

## **Inversor**

Conforme já foi referido, este bloco foi implementado utilizando o módulo inversor “inteligente” PM20CSJ060 da Mitsubishi [55], o qual permite correntes de 20 A e tensões máximas de 600 V. Este módulo possui ainda como características o facto de garantir um “tempo morto” de 2  $\mu$ s, permitir frequências do sinal de entrada PWM até 20 kHz e aceitar sinais de comando com amplitude máxima de 20 V (nível lógico 0, porque segue os níveis da lógica negativa) sendo, no entanto, recomendada a utilização de sinais com amplitude de 15 V. Seguidamente serão apresentados os requisitos cumpridos em função das indicações fornecidas pelo fabricante do módulo.

## **Alimentação do módulo Inversor**

Para alimentar o módulo inversor foi necessário providenciar quatro fontes (DC) isoladas de 15 V para alimentar cada um dos braços do inversor (não utilizados inclusive). Os transformadores (220/15) utilizados, na implementação destas fontes, possuem blindagem electrostática tendo uma capacidade de acoplamento entre primário e secundário inferior a 40 pF, por recomendação do fabricante do módulo de IGBTs,

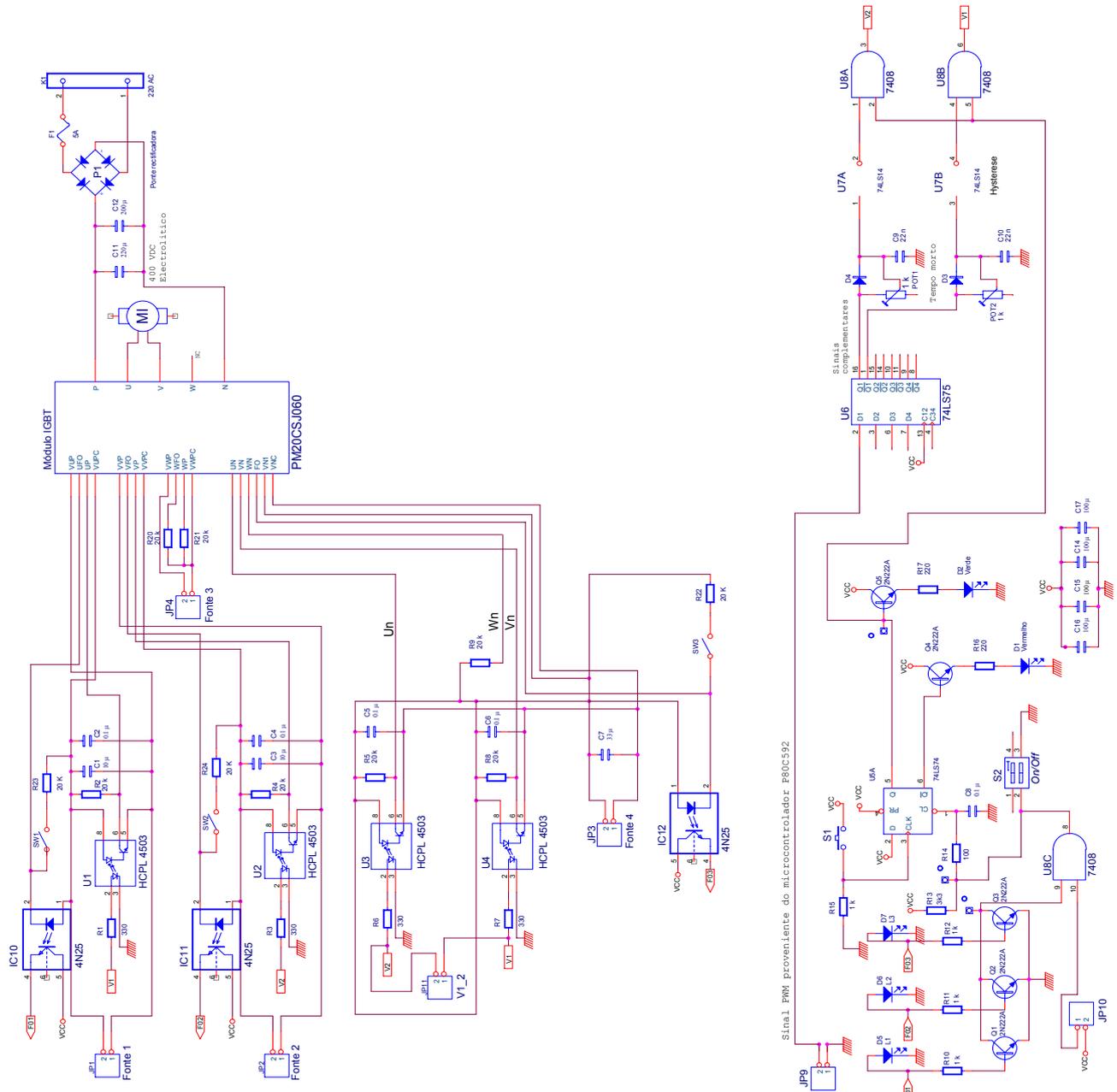


Figura 5.23 - Esquemático do circuito para accionamento do motor de indução monofásico.

sendo o valor aconselhado de 20 pF. Estes valores reduzidos de capacidade prendem-se com a necessidade de desacoplamento completo e de evitar ruído.

É também utilizada uma fonte auxiliar de 5 V para alimentar a placa que contém o microcontrolador e o circuito de accionamento.

## **Entradas e saídas não utilizadas**

Todas as entradas ou saídas do módulo não utilizadas, neste caso  $V_{WP}$ ,  $V_{WPC}$ ,  $W_{F0}$ ,  $W_P$  e  $W_N$ , foram, e devem ser, ligadas directamente ou através de uma resistência de *pull-up*, neste caso 20 k $\Omega$ , à respectiva fonte dedicada (fonte 3 na Figura 5.23), conforme recomendação do fabricante.

## **Sinais complementares para comutação do inversor**

Para permitir a comutação dos IGBTs que compõem o módulo inversor são necessários, por fase, dois sinais complementares. Com este propósito utilizou-se o circuito integrado 74LS75 [57] que permite com um sinal de entrada obter à saída dois sinais complementares, sendo um deles igual ao de entrada. Refira-se também que o módulo segue a lógica negativa relativamente aos níveis de tensão dos sinais de entrada [55], que neste caso são aplicados nos terminais  $U_P, V_P, W_P, U_N, V_N$  e  $W_N$  do módulo, conforme ilustrado na Figura 5.23.

## **Tempo de atraso**

Um requisito importante e imprescindível numa aplicação deste tipo é a garantia de um “tempo morto” com o objectivo de evitar curto-circuitos indesejáveis. O fabricante recomenda neste caso um “tempo morto” mínimo de 4 $\mu$ s, o qual é garantido, por *hardware*, através dos componentes POT1, POT2, D3, D4, C9 e C10, mostrados na Figura 5.23, cujos valores escolhidos garantem o atraso necessário de 7  $\mu$ s e pelo circuito integrado 74LS14 [58], que garante à sua saída os níveis TTL pretendidos.

## **Detecção de falhas**

O módulo “inteligente” tem a capacidade de detectar e assinalar alguns tipos de falhas, nomeadamente curto-circuitos, sobrecorrentes, sobreaquecimento e falta de tensão de alimentação, não fazendo contudo distinção entre elas. As falhas são assinaladas por saídas do módulo em colector aberto com resistência em série, sendo possível ligar directamente entre estas saídas e a tensão de alimentação, optoacopladores ou *leds*. Na implementação realizada foram aproveitadas estas saídas do módulo para activar um circuito de segurança e assim evitar danos no inversor, devido à repetição de falhas detectadas. Isto é conseguido inibindo o sinal de comando através de uma porta lógica AND, circuito integrado U8C da Figura 5.23.

## Isolamento

Para garantir o isolamento, entre a parte de comando e a parte de potência, foram utilizados optoacopladores com características especiais de acordo com o recomendado pelo fabricante. Assim, utilizaram-se para os sinais de entrada, terminal IN na Figura 5.23, optoacopladores de elevada velocidade de resposta, uma vez que a comutação se efectua com elevada frequência, e com elevada razão de rejeição de modo comum, *Common Mode Rejection* (CMR). Foi utilizado o integrado HCPL4503 [59], que possui as seguintes características:

### Elevado CMR

$CMR > 10 \text{ kV}/\mu\text{s}$  a  $V_{cm} > 1000 \text{ V}$ ,  $V_{cm}$  é a amplitude da subida de tensão aplicada.

### Elevada velocidade de resposta

$t_{PHL}, t_{PLH} < 0,8 \mu\text{s}$

### Tensão de isolamento

$V_{iso} > 2500 \text{ V}$

Para os sinais de saída do circuito de detecção de falhas, foram também utilizados optoacopladores, que não necessitam de ser tão rápidos, uma vez que as saídas de sinalização de falha (terminal F0 no circuito da Figura 5.23) mantêm o estado, pelo menos, durante 1,8 ms (típico) [55] e tiram vantagem de elevada taxa de transferência de corrente, *Current Transfer Ratio* (CTR). Como para tais optoacopladores o CTR não é tão elevado, passa pelo optoacoplador algum ruído de comutação, que pode ser eliminado adicionando um filtro com constante de tempo 10  $\mu\text{s}$ .

## 6 SOFTWARE DA REDE DE ACTUAÇÃO

A arquitectura de *software* do sistema global (Figura 6.1) é constituída pelas seguintes componentes:

- *Software* de configuração e optimização do sistema;
- *Software* de gestão da rede;
- *Software* de aquisição;
- *Software* de actuação.

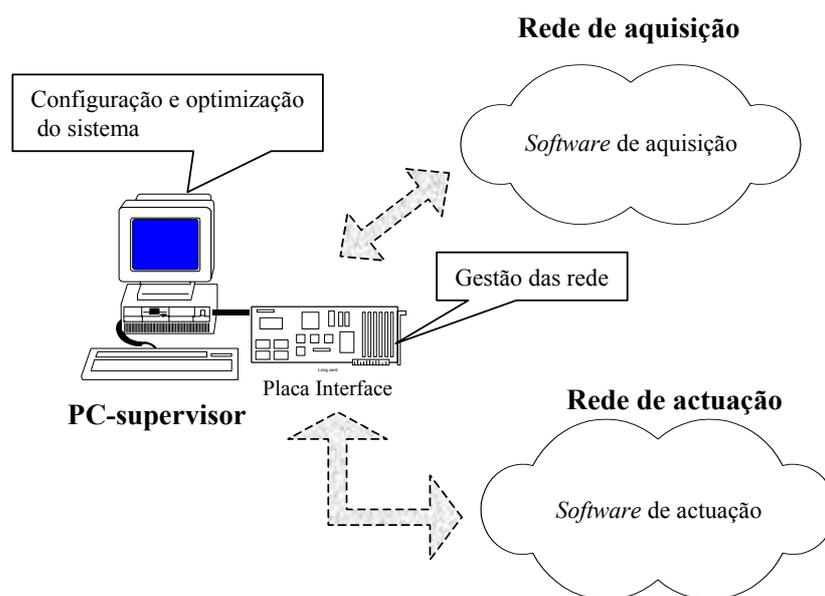


Figura 6.1 – Componentes de *software* do sistema.

O *software* de configuração e optimização do sistema está residente no PC da estação base, PC-supervisor, sendo um ponto de contacto do utilizador com o sistema. Entre as suas funções destacam-se a capacidade de configurar todo o sistema, processamento de dados para optimização de constantes intrínsecas ao desenvolvimento adequado do algoritmo de controlo, pedido de dados, ler e acertar o relógio de tempo real presente nas estações (com microcontrolador) de aquisição ou actuação [38].

O *software* de gestão da rede está implementado na placa de interface do PC-supervisor, sendo responsável pelo encaminhamento de todo o tráfego do PC para as estações de aquisição ou actuação e vice-versa, de um modo transparente para o utilizador. Para este processamento ser o mais simples possível, as *frames* que constituem as mensagens do PC já são compostas de acordo com o protocolo CAN [38].

O *software* de aquisição é executado nas estações de aquisição permitindo armazenar em memória e transmitir para o PC-supervisor os valores adquiridos, utilizando vários tipo de sensores distribuídos pela estufa.

De todas estas componentes apenas a de actuação foi desenvolvida no âmbito desta dissertação, sendo as restantes desenvolvidas no âmbito do sistema global de controlo [1, 38] de estufas agrícolas a ser desenvolvido na U.T.A.D., o qual conforme referido no capítulo 4, utiliza várias plataformas de comunicação aplicadas no controlo ambiental

das mesmas, e para o qual este trabalho pretende ser uma solução para a rede de actuação.

Seguidamente será analisado o *software* elaborado para a componente de actuação.

## 6.1 ESTRUTURA DE COMUNICAÇÕES DA REDE

A rede de actuação proposta é composta pelos seguintes elementos (Figura 6.2):

- Um PC-supervisor, responsável por tarefas de gestão, na estufa;
- Um nó (*Host*) responsável pelos nós SLIO CAN;
- Um nó designado por *Slave Motor*, dedicado ao controlo de velocidade de um motor;
- Vários nós SLIO CAN, dedicados ao controlo (liga/desliga) de actuadores;
- Outros nós CAN (com microcontrolador) adicionais.

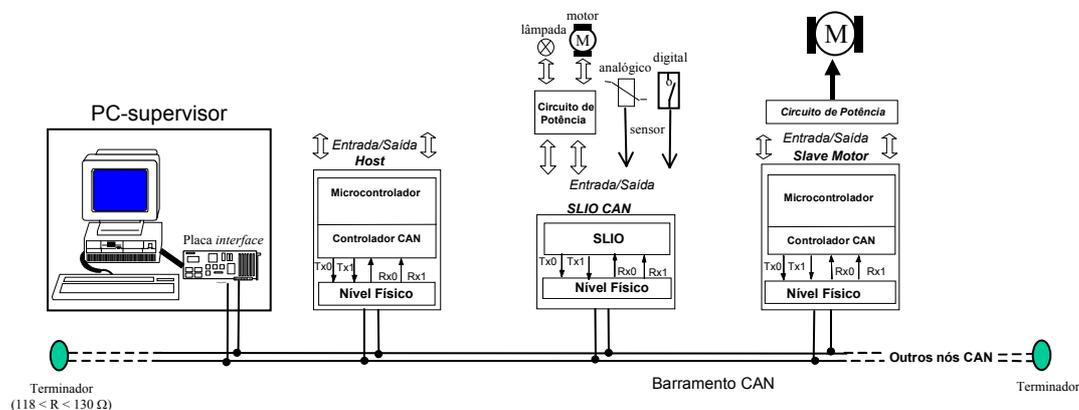


Figura 6.2 - Elementos que compõem a arquitectura da rede de actuação.

Posto isto, será abordado em secções deste capítulo o *software* residente nas placas *Host* e *Slave Motor*, o qual foi projectado em função das relações de comunicação existentes entre os diversos nós que constituem a rede implementada. O *software* residente no PC-Supervisor foi desenvolvido no âmbito de trabalhos anteriores [38] pelo que não será alvo de explicação nesta dissertação.

Quanto à comunicação existente entre os diversos nós da rede, esta foi projectada de forma a que o PC-supervisor comunique apenas com os nós “inteligentes” existentes na rede CAN, ou seja, com os nós que possuem um microcontrolador com controlador CAN. No que diz respeito aos nós SLIO CAN estes comunicam com o nó *Host*, o qual também é responsável pela sua calibração. A relação em termos de comunicação entre os vários nós da rede de actuação está representada no diagrama da Figura 6.3.

Antes de abordar o *software* residente em cada um dos nós (com microcontrolador), será apresentado, nas próximas secções, o formato e significado das mensagens que constituem a essência do sistema uma vez que é em função delas que é realizado determinado conjunto de acções de actuação.

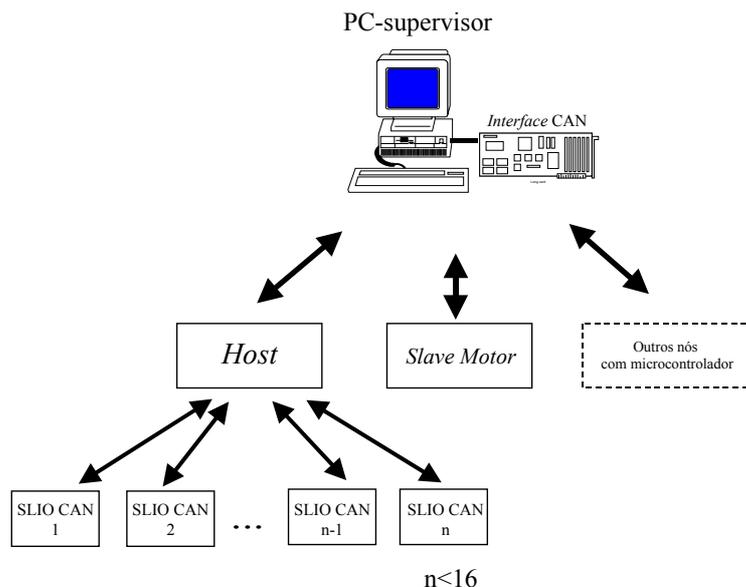


Figura 6.3 - Relação (em termos de comunicação) entre os vários nós da rede de actuação.

### 6.1.1 As mensagens CAN

É importante referir, antes de mais, que todas as mensagens utilizadas foram definidas seguindo as especificações 2.0 A e B (passivo) do protocolo CAN, estando por isso de acordo com os controladores CAN utilizados [33, 37].

No protocolo CAN, conforme foi referido no capítulo 3, sempre que se refere o termo *frame* de dados, este compreende sete campos distintos de informação: início de *frame*, campo de arbitragem, campo de controlo, campo de dados, campo CRC, campo de reconhecimento e fim de *frame*, conforme mostra a Figura 6.4. É o controlador do protocolo CAN, embebido no microcontrolador, que constrói o primeiro e os últimos três campos, anteriormente referidos, sendo os restantes construídos através de *software* desenvolvido pelo utilizador.

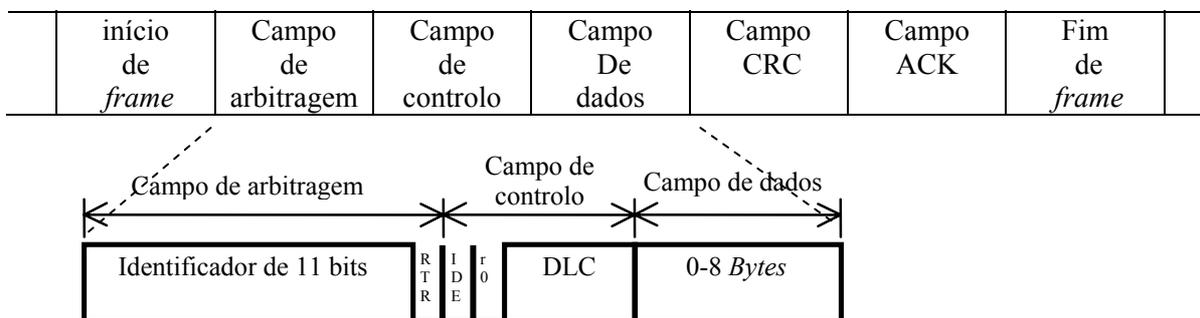


Figura 6.4 – *Frame* de dados CAN.

Conforme já referido no capítulo 3, cada mensagem possui no campo de arbitragem um conjunto de *bits* designado por identificador que no caso das versões CAN 2.0 A ou 2.0 B (passivo) é constituído por 11 *bits*, representado na Figura 6.4, os quais caracterizam a mensagem, indicando qual o seu significado e prioridade.

No entanto, do ponto de vista do utilizador, os campos de arbitragem e controlo para as versões CAN 2.0A e B (passivo) têm o formato representado pela Figura 6.5 onde o campo de controlo é composto apenas pelos *bits* do código de comprimento de dados, DLC, sendo inacessíveis, para o utilizador, os dois *bits* reservados a aplicações futuras os quais no entanto são considerados pelo (s) controlador (s) CAN.

Campo de arbitragem											Campo de controlo				
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0
← Byte 1 →								← Byte 2 →							

**Figura 6.5 — Formato dos dois bytes que constituem os campos de arbitragem e controlo de uma frame de dados.**

Os 11 *bits* do identificador CAN pertencentes ao campo de arbitragem são utilizados para duas funções distintas:

- Os 8 *bits* mais significativos (ID10..ID3), são utilizados no filtro de aceitação do controlador CAN, ou seja, servem para identificar os nós CAN para os quais a mensagem tem significado;
- Os 3 *bits* menos significativos (ID2..ID0), são irrelevantes no processo de aceitação, sendo utilizados nesta aplicação para identificar o tipo de mensagem.

A atribuição dos códigos de aceitação é feita através da programação do registo de Aceitação, *Acceptance Code Register* (ACR), do microcontrolador utilizado [37].

Para além do código de aceitação, cada nó (com microcontrolador) utiliza também uma máscara para realizar o processo de filtragem, permitindo desta forma que cada nó verifique se a mensagem é ou não relevante para ele. A máscara é atribuída através da programação do registo de Máscara de Aceitação, *Acceptance Mask Register* (AMR) sendo o complemento do ACR para cada nó.

Se uma mensagem passar o teste de aceitação e se existir um *buffer* de recepção livre, então os campos de arbitragem, controlo e de dados dessa mensagem são sequencialmente armazenados no *buffer* para posterior processamento. Os 8 *bits* mais significativos do identificador da mensagem participam no processo de filtragem da seguinte forma:

Cada nó realiza a monitorização do barramento e, ao receber o 1º *byte* (ID10..ID3), verifica se a seguinte equação é válida:

$$(ID10...ID3) = [(AC.7 \text{ a } AC.0) \vee (AM.7 \text{ a } AM.0)] = 1111 \ 1111_b$$

Em caso afirmativo, a mensagem é considerada válida para esse nó, caso contrário continua a monitorar o barramento até receber uma mensagem válida.

Em resumo, os 8 *bits* (AC.7 a AC.0) do registo ACR e os 8 *bits* mais significativos do identificador da mensagem (ID10..ID3) devem ser iguais nas posições marcadas como relevantes pelos *bits* (AM.7 a AM.0) do registo AMR. São consideradas relevantes as posições dos *bits* colocados a '0' e irrelevantes as posições dos *bits* colocados a '1' no registo AMR.

Para os nós com microcontrolador constituintes do sistema foram atribuídos os códigos

de aceitação que mostra a Tabela 6.1.

Nó da rede	Filtro de aceitação
Supervisor	0100 1000
<i>Host</i>	0101 0000
<i>Slave Motor</i>	1001 0000
<i>Host e Slave Motor</i>	1101 0000

**Tabela 6.1 — Códigos de aceitação dos nós (com microcontrolador) que constituem a rede de actuação.**

Para o nó *Host* os valores escolhidos para o processo de filtragem foram os de ACR= 0x50H e AMR= 0xAFH o que faz com que este nó receba todas as mensagens com os *bits* ID7 e ID9 do identificador a '1', e rejeite todas as mensagens na rede que tenham *bits* de valor diferente naquelas posições.

Quanto aos três *bits* menos significativos do identificador (ID2..ID0) pertencentes ao 2º *byte*, estes são utilizados para distinguir o tipo de mensagem para a rede de actuação, conforme mostra a Tabela 6.2.

Tipos de Mensagem	Identificador		
	ID2	ID1	ID0
Alarme	0	0	0
Configuração	0	0	1
RTC	0	1	0
Reservado	0	1	1
Actuação	1	0	0
De SLIO	1	0	1
<i>Set-point</i>	1	1	0
Aquisição	1	1	1

**Tabela 6.2 – Significado dos três *bits* menos significativos do identificador.**

O significado dos restantes *bits* do campo de arbitragem e de controlo das mensagens CAN, conforme descrito no capítulo 3, é o seguinte:

- O *bit* RTR distingue as mensagens de dados (RTR=0) das mensagens de pedido remoto de dados (RTR=1);
- Os *bits* DLC indicam o comprimento em número de *bytes*, do campo de dados que compõe a mensagem.

A Figura 6.6 ilustra o sentido e tipo das mensagens CAN na rede de actuação.

Quanto ao significado e nós da rede, envolvidos na comunicação, as mensagens podem ser:

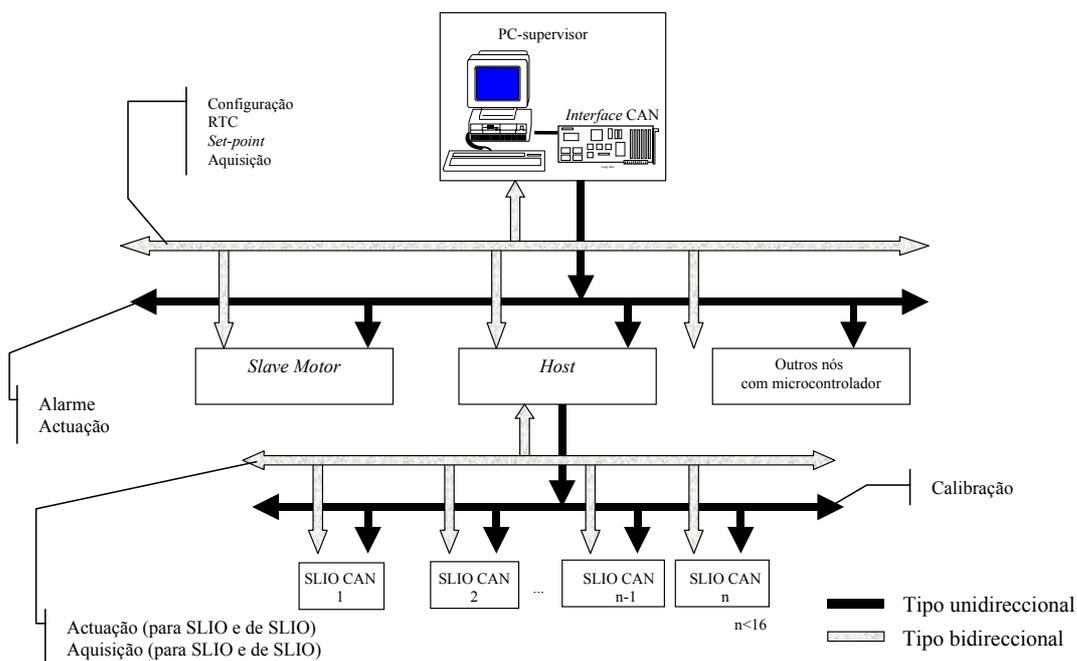
1- Mensagens entre o PC- supervisor e as estações (com microcontrolador)

- Mensagens de alarme;
- Mensagens de configuração;
- Acerto e consulta do relógio (RTC);
- Envio de *set-points* (ex. frequência de uma onda a sintetizar);
- Envio de comandos para actuação;

- Mensagens de aquisição.

## 2- Mensagens entre o nó *Host* e os nós SLIO CAN

- Mensagens destinadas aos nós SLIO CAN responsáveis por diversos tipos de actuadores na estufa (ex. motores de janelas, electroválvulas, etc). Estas mensagens podem ser comandos ou pedidos de informação enviados aos nós SLIO CAN por forma a permitir uma avaliação do estado da rede SLIO CAN por parte do nó *Host* e, se necessário, por parte do PC-supervisor;
- Mensagens de calibração.



**Figura 6.6 - Sentido(s) do fluxo de mensagens em função do tipo de mensagens na rede de actuação.**

Nas próximas secções deste capítulo será referido o significado e formato das mensagens referidas.

## 6.1.2 Mensagens entre o PC-supervisor e as estações com microcontrolador

### 6.1.2.1 MENSAGEM DE CONFIGURAÇÃO

Para permitir configurar vários nós “inteligentes” da rede a partir do PC-supervisor, via CAN, foi prevista a existência de mensagens de configuração para essa finalidade. Este tipo de mensagem tem características próprias e é diferente conforme o tipo de nó a que se destina.

## i) Nó *Host*

No caso do nó *Host*, para configurar o sistema SLIO CAN a partir de uma estação remota (PC-supervisor), a mensagem de configuração deve permitir alterar os dispositivos accionadores (SLIOs) e/ou pinos dos mesmos (tanto para activar como inibir), atribuídos à realização de determinada tarefa de actuação (ex. aquecimento, arrefecimento, janelas, ventilador, rega, CO<sub>2</sub>, etc.). Desta forma será melhorada a flexibilidade do sistema uma vez que é possível a redistribuição de recursos (nós SLIO CAN), para além da capacidade de adicionar nós ao sistema que o CAN permite.

Para implementar esta capacidade de configuração, foi reservado na memória do nó *Host* um *buffer* de configuração, Anexo 3, o qual armazena uma configuração *default* conforme ilustrado na Tabela 6.3, sendo a atribuição do número de tarefa por actuação representada na Tabela 6.4. Assim, através de mensagens CAN que permitam alterar o *buffer* de configuração do nó *Host* é possível redistribuir recursos (nós SLIO CAN) do sistema, atribuídos a determinado tipo de actuação.

Nº Tarefa	<i>Buffer</i> de Configuração				
	Accionador	MSB 1	LSB 1	MSB 2	LSB 2
0 →	<b>Slio 0</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
1 →	<b>Slio 1</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
2 →	<b>Slio 2</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
3 →	<b>Slio 3</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
4 →	<b>Slio 4</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
5 →	<b>Slio 5</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
6 →	<b>Slio 6</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
7 →	<b>Slio 7</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
8 →	<b>Slio 8</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
9 →	<b>Slio 9</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
10 →	<b>Slio 10</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
11 →	<b>Slio 11</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
12 →	<b>Slio 12</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
13 →	<b>Slio 13</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
14 →	<b>Slio 14</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>
15 →	<b>Slio 15</b>	<b>0xFF</b>	<b>0xFF</b>	<b>0x00</b>	<b>0x00</b>

Tabela 6.3 - Configuração existente por *default* no *buffer* de configuração do nó *Host*.

Nº Tarefa	Actuação
0 a 3	Aquecimento
4 a 5	Arrefecimento
6 a 9	CO <sub>2</sub>
10 a 14	Rega
15	Alarme

Tabela 6.4 – Atribuição de nº tarefa/actuação por *default*.

As mensagens de configuração permitem duas variantes:

## 1 - *Frame* de dados para configuração

Sempre que for necessário alterar a configuração de qualquer *tarefa* do sistema SLIO CAN, a *frame* que o PC-supervisor deve transmitir para o nó *Host* pertencente à rede de actuação deve ter o formato que ilustra a Figura 6.7.

Campo de arbitragem e controlo		Campo de dados					
Identificador	Msg+Rtr+DLC	Nº da tarefa a configurar	Endereço do Accionador	MSB1	LSB1	MSB2	LSB2
<i>Byte 1</i>	<i>Byte 2</i>	<i>Byte 3</i>	<i>Byte 4</i>	<i>Byte 5</i>	<i>Byte 6</i>	<i>Byte 7</i>	<i>Byte 8</i>

**Figura 6.7– Formato de uma mensagem de configuração.**

Os *bytes* que compõem a mensagem de configuração têm o seguinte conteúdo:

**1º byte (identificador)** – Este *byte* é composto pelos 8 *bits* mais significativos do identificador os quais são utilizados pelo controlador CAN no processo de validação da mensagem. Os *bits* deste campo devem permitir a validação da mensagem apenas pelo nó *Host*. Assim sendo foi escolhido para este campo o valor ID10..ID3=0101 0000.

**2º byte (Msg+RTR+DLC)** – O conteúdo deste *byte* é fixo e composto pelos seguintes valores:

Msg - valor 001 por se referir a uma mensagem de configuração (conforme referido na Tabela 6.2);

RTR - valor 0 por se tratar de um mensagem de dados;

DLC - valor 0110 por possuir um campo de dados com 6 *bytes*.

**3º byte (Número da tarefa a configurar)** – É o primeiro *byte* de dados da mensagem CAN e contém o número da tarefa (no máximo 15) à qual se destina a configuração. O número da tarefa é fixo e associado a determinado tipo de actuação no programa executado no nó *Host* (Tabela 6.4). Por exemplo, por defeito as tarefas 0, 1, 2 e 3 estão relacionadas com funções de aquecimento. Ou seja é possível utilizar quatro nós SLIO CAN para funções deste tipo sendo as configurações associadas às tarefas 0, 1, 2 e 3 do *buffer* de configuração. Com este processo de configuração é possível configurar todos os dispositivos SLIO CAN e permitir vários tipos de actuação por dispositivo até um máximo de 16, por ser este o número de saídas possível para cada nó SLIO CAN [33].

**4º byte (Endereço do accionador)** — Este *byte* da mensagem contém o identificador do dispositivo (nó SLIO CAN) que irá realizar a tarefa indicada pelo *byte* anterior.

**5º e 6º bytes (MSB1 e LSB1)** — Estes dois *bytes* da mensagem significam os pinos do dispositivo (SLIO CAN) que terão relevância quando o mesmo for solicitado para activar as suas saídas. Por *default* os nós SLIO CAN possuem estes portos a “1” (MSB2=0xFF e LSB2=0xFF) quando chamados a activar as saídas, o que consiste em, ficarem todas com o nível alto (5 V).

**7º e 8º bytes (MSB2 e LSB2)** — Estes dois *bytes* da mensagem significam os pinos do dispositivo (SLIO CAN) que terão relevância quando o mesmo for solicitado a inibir os sinais que tem à saída por *default* todos os SLIOs possuem estes portos a ‘0’

(MSB2=0x00 e LSB2=0x00) quando chamados a inibir as saídas, ficando todas ao nível baixo (0 V).

## 2 - *Frame* remota ou pedido de configuração

Sempre que o PC-supervisor pretenda conhecer o valor corrente da configuração guardada no nó *Host*, relacionado com determinada tarefa, é possível através do envio de uma *frame* remota ter acesso a essa informação.

A *frame* remota que o PC-supervisor deve enviar tem o formato representado na Figura 6.8 a qual é idêntica ao da *frame* de dados da Figura 6.7, apresentando, no entanto, algumas diferenças.

Campos de arbitragem e controlo		Campo e dados
Identificador	Msg+Rtr+DLC	Nº da tarefa
<i>Byte 1</i>	<i>Byte 2</i>	<i>Byte 3</i>

Figura 6.8 – Formato de uma mensagem para pedido de configuração.

O conteúdo dos *bytes* que constituem este tipo de *frame* é o seguinte:

**1º *byte* (identificador)** – É composto pelos 8 *bits* mais significativos do identificador e que deve possuir um valor igual ao utilizado para a *frame* de dados.

**2º *byte* (Msg+RTR+DLC)** – O conteúdo deste *byte* é fixo e bem definido sendo composto pelos seguintes constituintes:

Msg - de valor 001 por se tratar de uma mensagem de configuração;

RTR - de valor 1 porque se trata de um pedido remoto de dados;

DLC – valor 0110 por ser este o número de *bytes* de dados que corresponde à resposta.

**3º *byte* (Número da tarefa)** - Este *byte* da mensagem deve conter o número (no máximo 15) da tarefa, sobre a qual se pretende saber a configuração.

A resposta correspondente a este pedido, consiste no envio, por parte do *Host*, de uma mensagem de dados (RTR=0) com formato igual ao representado na Figura 6.8. Esta mensagem possui, no entanto, um identificador que será utilizado por todos os nós com microcontrolador, no processo de filtragem, mas que será validado apenas pelo controlador do nó PC-supervisor, o qual utiliza o AMR com o valor indicado na Tabela 6.1. O valor utilizado para os 8 *bits* mais significativos do identificador desta mensagem é de ID10..ID3=01001000, ou seja, 0x48H.

### ii) Outros nós (com microcontrolador)

É possível implementar um processo idêntico para configurar outros nós CAN (com microcontrolador) a partir do PC-supervisor desde que sejam tidos em consideração o *software* e valor dos registos AMR e ACR desses nós. Ou seja, no caso de se pretender configurar remotamente outros nós CAN, as mensagens a enviar devem especificar nos 8 *bits* mais significativos do identificador, qual ou quais os tipos de nós para os quais têm significado, por forma a que o campo de dados dessas *frames* seja convenientemente interpretado. Este campo de dados será diferente do utilizado no caso

do nó *Host*, uma vez que este está relacionado com características próprias daquele nó (gestor do sistema SLIO CAN).

### 6.1.2.2 MENSAGEM DE ALARME

Para situações de anomalia no sistema, foi definido um tipo de mensagem que será transmitida pelo PC-supervisor, para desencadear um conjunto de acções, em função das anomalias registadas. Este tipo de mensagem é caracterizado por possuir os três *bits* menos significativos do identificador com o valor zero (Tabela 6.2 e Figura 6.9). Os oito *bits* mais significativos do identificador participam no processo de aceitação, realizado pelos diversos controladores existentes na rede CAN.

Uma mensagem de alarme, quando recebida pelo nó *Host* da rede de actuação, faz com que sejam transmitidas mensagens aos nós SLIO CAN pelos quais é responsável, para que sejam desligados todos os actuadores da estufa e activado um sinal de alarme.

Quando recebido por outros nós CAN (com microcontrolador), este tipo de mensagem poderá desencadear um conjunto de acções similares às referidas para o nó *Host* e que dependem do *software* residente nesses nós.

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0
X	x	X	x	X	X	x	x	0	0	0	0	x	x	x	X

**Figura 6.9 – Formato dos campos de arbitragem e controlo numa mensagem de alarme.**

### 6.1.2.3 MENSAGEM DE SET-POINT(S)

Para tornar possível, ao PC-supervisor do sistema, enviar *set-points* para os nós CAN com microcontrolador, na rede de actuação, foi definido um tipo de mensagem para o efeito. Este tipo de mensagem é caracterizado pelo valor dos *bits* menos significativos do identificador terem o valor ID2ID1ID0=110.

Este tipo de *frame* permite duas variantes:

- *Frame* de dados;
- *Frame* remota.

No primeiro caso a *frame* tem o formato representado na Figura 6.10 onde se destacam:

- Os 8 *bits* mais significativos do identificador, tal como nos tipos de mensagem anteriores, participam no processo de filtragem realizado pelos controladores dos vários nós CAN da rede, sendo utilizados para identificar o(s) tipo(s) de nó(s) para os quais o conteúdo da mensagem possui significado;
- *Bit* RTR do campo de arbitragem com o valor lógico ‘0’;
- Campo de dados que corresponde aos valores dos *set-points*, o qual deve ser correctamente interpretado e utilizado pelo nó receptor de acordo com o código nele residente.

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0
x	x	X	x	x	x	x	x	1	1	0	0	0	0	0	1

Campo de Dados							
Byte 1							
Liga/desliga		V5	V4	V3	V2	V1	V0
	x						

**Figura 6.10 – Formato da *frame* de dados para envio de *set-points* por parte do nó PC-supervisor.**

Exemplo: No caso de mensagens de *set-point* enviadas pelo PC-supervisor para o nó *Slave Motor* o formato corresponde ao apresentado na Figura 6.10. Neste caso o 1º *byte* de dados assinala a intenção de activar ou inibir uma carga, sendo utilizado o *bit* mais significativo (Liga/desliga) desse *byte*, que assinala o estado desejado por parte do PC-supervisor para essa carga. Existem também nesse *byte* de dados, 6 *bits* (V5..V0) que se destinam, conforme será referido na secção 6.3, à indicação por parte do PC-supervisor do valor pretendido para a frequência (0 a 60 Hz) de uma onda sinusoidal a sintetizar, pelo nó *Slave Motor*, utilizando para tal técnicas de modulação PWM para controlo de velocidade de um motor de indução monofásico.

Para conhecer o valor do(s) *set-point*(s) de determinado(s) nó(s), a *frame* remota a enviar pelo PC-supervisor deve ter o formato ilustrado na Figura 6.11, a qual se distingue da *frame* de dados no valor do *bit* RTR e pela ausência de campo de dados.

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0
x	x	x	x	X	X	x	x	1	1	0	1	0	0	0	1

**Figura 6.11 – Formato da *frame* remota para pedido de *set-points*.**

Exemplo: para saber os valores da frequência e se está ligada ou desligada a carga do nó *Slave Motor*, a *frame* remota (RTR=1) que o PC-supervisor precisa enviar tem um formato idêntico ao apresentado na Figura 6.11. O valor dos bits ID10..ID3 deve estar de acordo com o valor dos registos AMR e ACR do nó *Slave Motor*.

A resposta a este pedido corresponde ao envio, pelo nó “interrogado”, de uma mensagem de dados (RTR=0) com formato igual ao representado na Figura 6.10. Esta mensagem deverá possuir, um identificador com os 8 *bits* mais significativos de valor ID10..ID3=01001000 conforme explicado anteriormente para resposta a mensagens remotas provenientes do PC-supervisor. O campo de dados desta mensagem é composto pelos valores existentes no nó emissor.

#### 6.1.2.4 MENSAGEM DE AQUISIÇÃO

Apesar de não ser objectivo desta aplicação desempenhar tarefas de aquisição mas sim de actuação, foi considerada também a possibilidade de realizar aquisição através dos nós que compõem a rede de actuação, sendo importante no caso de anomalia temporária da rede de aquisição, por RF. O formato da mensagem CAN de aquisição a ser transmitida pelo nó PC-supervisor, para a rede de actuação, é o apresentado na Figura 6.12.

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0
x	x	x	x	X	X	x	X	1	1	1	0	X	x	x	x

**Figura 6.12 – Formato da *frame* de dados para realizar aquisição.**

Uma mensagem deste tipo, quando recebida pelo nó *Host*, faz com que este envie mensagens a vários nós SLIO CAN preparados para tal tarefa, os quais por sua vez lhe responderão, de acordo com a prioridade de cada nó, com uma mensagem contendo o valor adquirido através de um conversor A/D de 10 *bits* [33], segundo um formato que será explicado na secção 6.1.3, dedicada a mensagens entre os nós SLIO e o nó *Host*.

Após serem processados no nó *Host*, os dados resultantes da aquisição efectuada por vários nós SLIO CAN, são enviados pelo mesmo ao nó PC-supervisor, através de uma mensagem de dados com o formato e conteúdo representado na Figura 6.13.

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	DIR	RTR	DLC3	DLC2	DLC1	DLC0
0	1	x	0	1	X	x	x	1	1	1	0	0	0	1	1

Campo de dados															
<i>Byte 1</i>								<i>Byte 2</i>							
0	0	0	0	S3	S2	S1	S0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

Campo de dados							
<i>Byte 3</i>							
AD1	AD0	0	0	0	0	0	0

**Figura 6.13 – Formato da mensagem enviada pelo nó *Host* após aquisição (A/D) realizada pelos nós SLIO CAN.**

Nesta mensagem os *bits* menos significativos do primeiro *byte* de dados (S3..S0) identificam o SLIO onde foi efectuada a aquisição, enquanto que os *bits* dos segundo e terceiro *bytes* (AD9...AD0) correspondem ao valor da mesma.

Uma mensagem de aquisição (Figura 6.12), quando recebida por um nó com microcontrolador que não o nó *Host*, permite através do *software* residente no nó receptor, executar uma ou várias aquisições A/D. O resultado da aquisição deve ser transmitido utilizando uma *frame* idêntica à enviada pelo nó *Host*, porém com diferença nos *bits* do 1º *byte* de dados que indicam a origem dos dados. O formato da mensagem a enviar pelos nós com microcontrolador (excepto o *Host*) está representado na Figura 6.14.

A diferença entre estas mensagens e as enviadas pelo nó *Host* reside no 4º *bit* mais significativo do 1º *byte* de dados que é colocado ao nível lógico ‘1’.

Os *bits* (N3..N0) identificam o nó (com microcontrolador) onde foi efectuada a aquisição enquanto que os *bits* (AD9...AD0) correspondem ao valor da mesma.

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	DIR	RTR	DLC3	DLC2	DLC1	DLC0
0	1	x	0	1	x	x	x	1	1	1	0	0	0	1	1

Campo de dados															
Byte 1								Byte 2							
0	0	0	1	N3	N2	N1	N0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

Campo de dados							
Byte 3							
AD1	AD0	0	0	0	0	0	0

**Figura 6.14 – Formato da mensagem enviada por um nó com microcontrolador (excepto o *Host*) após aquisição A/D.**

### 6.1.2.5 MENSAGEM PARA RELÓGIO DE TEMPO REAL (RTC)

Para que seja possível configurar o relógio de tempo real do sistema, RTC, foi definido um tipo de mensagem para o efeito. Este tipo de mensagem é importante uma vez que permite configurar e consultar os dados do RTC de um nó (com microcontrolador) a partir do PC-supervisor.

Tal como para outros tipos de mensagens também este permite duas variantes na troca de informação entre o PC-Supervisor e o *Host*, ou outro nó da rede que possua microcontrolador. Assim, as variantes possíveis para este tipo de mensagem são:

- Configurar os dados do RTC de determinado nó;
- Consultar os dados do RTC de determinado nó.

A *frame* para configurar o RTC de determinado nó com microcontrolador (*Host* inclusive) deve ser enviada pelo PC-supervisor com o formato representado na Figura 6.15. Nesta mensagem, para além dos 8 *bits* mais significativos do campo de arbitragem que permitem definir os nós para os quais ela tem relevância (Tabela 6.1), destaca-se o valor do *bit* RTR indicando que se trata de uma *frame* de dados bem como os campos que compõem a informação transmitida (campo de dados da mensagem CAN) e que são os valores desejados para acerto das grandezas temporais do sistema ou seja, horas, minutos, etc.

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0
x	x	x	x	X	x	x	x	0	1	0	0	0	1	1	0

Campo de dados					
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Segundos	Minutos	Horas	Dia	Mês	Ano

**Figura 6.15 – Formato da *frame* para configuração do RTC.**

A *frame* para pedido de informação sobre o estado do RTC é idêntica à anterior com a excepção do *bit* RTR=1 e da inexistência de campo de dados.

O campo DLC possui neste caso o valor sete (DLC3..DLC0=0111), uma vez que é este o número de *bytes* esperados, respeitando a especificação do protocolo CAN [27] utilizada.

A resposta a este pedido corresponde ao envio, pelo nó “interrogado”, de uma mensagem de dados (RTR=0) destinada ao nó PC-supervisor com formato igual ao representado na Figura 6.16. Assim, é possível por parte do PC-supervisor conhecer o estado das grandezas do relógio de tempo real de determinado nó da rede (indicado no 1º *byte* de dados).

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0
0	1	x	1	0	x	x	x	0	1	0	0	0	1	1	1

Campo de dados						
Byte1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Nó	Segundos	Minutos	Horas	Dia	Mês	Ano

**Figura 6.16 – Formato da *frame* contendo os valores do RTC.**

### 6.1.2.6 MENSAGENS DO PC-SUPERVISOR PARA ACTUAÇÃO

Por forma a possibilitar o envio de comandos por parte do PC-Supervisor para actuação, foi definido um tipo de mensagem com esse propósito.

Neste tipo de mensagem de dados, Figura 6.17, destaca-se a importância dos três *bits* menos significativos do campo identificador os quais indicam aos nós da rede (com microcontrolador) que se trata de uma mensagem para actuação (Tabela 6.2) e do primeiro *byte* do campo de dados, o qual indica o tipo de actuação e estado pretendido para os actuadores. O significado dos *bits* do primeiro *byte* de dados é o seguinte:

- Os quatro *bits* menos significativos (b3...b0) identificam o tipo de actuação, permitindo distinguir 16 tipos diferentes, estando alguns representados na Tabela 6.5;
- O *bit* b4 indica o estado pretendido para a actuação da seguinte forma:
  - Activar b4=1;
  - Inibir b4=0.

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	1	0	0	0	0	0	0	1

Campo de dados							
Byte 1 – Tipo de actuação e comando							
X	X	x	b4	b3	b2	b1	b0

**Figura 6.17 – Formato das *frames* de dados para comunicação entre o PC-Supervisor e os nós (com microcontrolador) da rede para actuação.**

Actuação	b3	b2	b1	b0
Arrefecimento	0	0	1	1
Aquecimento	0	1	0	0
CO2	0	1	0	1
Rega	0	1	1	0

**Tabela 6.5 –Tipos de actuação.**

Exemplo: Se for pretendido activar o sistema de arrefecimento, a mensagem a enviar pelo PC-supervisor deve ser composta conforme mostra a Figura 6.18.

Campo de arbitragem												Campo de controlo				
X	X	X	X	X	X	X	X	1	0	0	0	0	0	0	0	1

Campo de dados							
Byte 1 – Tipo de actuação e comando							
X	X	0	1	0	0	1	1

**Figura 6.18 – Mensagem para arrefecimento.**

O nó *Host* ou outro nó “inteligente”, dependendo dos oito *bits* mais significativos do identificador, ao receber esta mensagem irá, por sua vez, desencadear um conjunto de acções, de acordo com o *software* nele residente, relacionadas com o controlo do sistema de arrefecimento. No caso do nó *Host* o conjunto de acções passa pelo envio de mensagens aos SLIOs responsáveis por tarefas de arrefecimento, para activação dos respectivos actuadores (porque b4=1), o que pode significar por exemplo: abrir electroválvulas, ligar ventiladores, ou accionar motores para abertura de janelas.

## 6.1.3 Mensagens entre o nó *Host* e os nós SLIO CAN

### 6.1.3.1 MENSAGENS DO NÓ *HOST* PARA OS SLIOS

Na sequência do que foi referido para as mensagens do PC-supervisor para actuação, falta referir o formato das mensagens utilizadas para implementar a comunicação entre o nó *Host* e os nós SLIO CAN pelos quais é responsável. O nó *Host* desempenha funções de *router* no sistema uma vez que, após receber comandos provenientes do PC-supervisor do sistema para executar determinado tipo de actuação (por exemplo arrefecimento da estufa), cabe-lhe a ele ordenar aos nós SLIO CAN, pelos quais é responsável, o activar ou inibir das saídas destes, o que por sua vez está relacionado com o ligar ou desligar de dispositivos actuadores (ex. Electroválvulas, ventiladores, injectores de CO2, etc.) relativos ao tipo de actuação pretendida.

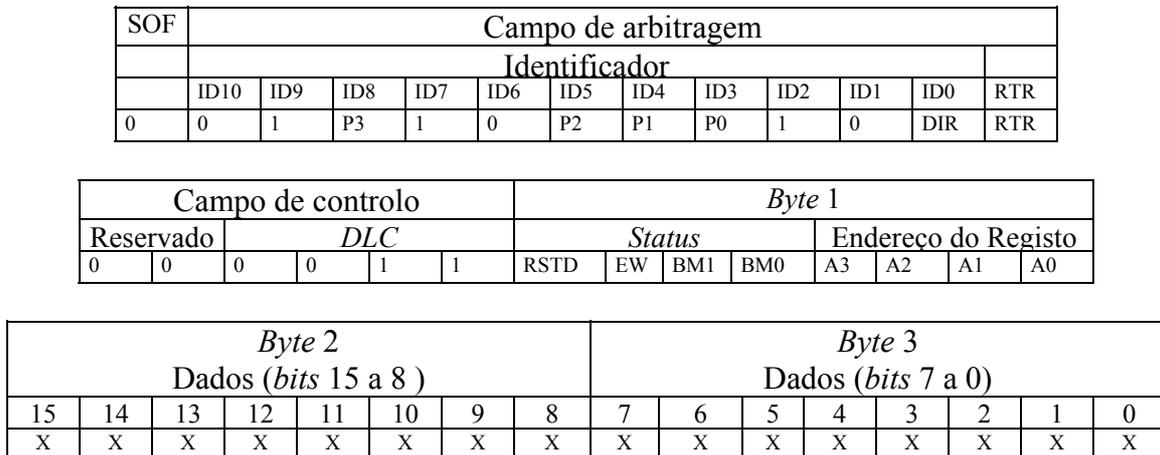
As mensagens do nó *Host* para os nós SLIO CAN foram implementadas de acordo com as especificações do SLIO P82C150 [33], sendo caracterizadas por possuírem o formato representado na Figura 6.19.

Note-se que o campo identificador só tem 4 *bits* configuráveis (P2 a P0), os quais possuem um valor único para determinado nó SLIO CAN, sendo este valor correspondente ao nível lógico dos sinais colocados nos respectivos pinos durante a operação de *reset* por *hardware* [33]. O campo identificador possui ainda o *bit* ID0

(DIR) que deve ser colocado ao nível lógico ‘1’ no caso de *frames* de dados destinadas ao nó *Host* e em pedidos de *frames* remotas enviados pelo nó *Host* para os nós SLIO e deve ser colocado ao nível lógico ‘0’ no caso de *frames* de dados enviadas do nó *Host* para os nós SLIO CAN. Ou seja:

ID2ID1ID0 = 100 ; *frames* do *Host* para os SLIOs

ID2ID1ID0 = 101 ; *frames* dos SLIOs para o *Host*



**Figura 6.19 – Formato de uma *frame* de dados do SLIO P82C150.**

Na Figura 6.19 também está representado o campo de controlo da mensagem, no qual é possível verificar que o comprimento de dados é fixo e igual a 3 *bytes*. O primeiro *byte* do campo de dados contém o *status* da informação e o endereço do registo pretendido, quanto aos segundo e terceiro *bytes* de dados estes dizem respeito ao conteúdo pretendido para o registo endereçado. O significado dos *bits* de *status* existentes no 1º *byte* de dados é referido no Anexo 4.

Quanto aos 4 *bits* menos significativos do primeiro *byte* de dados, estes representam o endereço do registo ao qual se destinam os dois *bytes* de dados seguintes, existindo 8 registos possíveis com endereço, designação e significado referidos no Anexo 4.

É através da identificação do SLIO e do estado dos pinos E/S do mesmo que foi implementado o controlo de algumas actuações, nomeadamente as do tipo liga/desliga, utilizando para tal circuitos que permitem a interface entre as saídas dos nós SLIO CAN e os actuadores que os mesmos pretendem controlar, recorrendo para o efeito a relés de estado sólido ou outros dispositivos.

Exemplo: Activar o pino E/S 4 do SLIO 7, ou seja, do SLIO que possui os *bits* configuráveis do identificador com o valor P3P2P1P0=0111.

Para realizar esta tarefa o nó *Host* transmite duas mensagens:

i) Uma primeira mensagem para activar o pino de saída pretendido, tendo para tal que escrever no registo *Output Enable* cujo endereço é o 04H (Anexo 4), atribuindo ao *bit* correspondente ao pino E/S pretendido, o nível lógico ‘1’. Ou seja:

- P3P2P1P0=0111 ; SLIO pretendido
- DIR=0 ; mensagem do *Host* para o SLIO
- RTR=0 ; uma *frame* de dados

- A3A2A1A0 = 0100<sub>(binário)</sub>=04<sub>(décimal)</sub> ; Endereço do registo
- primeiro *byte* (saídas de 15 a 8) com os *bits* todos a zero lógico;
- segundo *byte* (saídas de 7 a 0) com o *bit* pretendido a '1' lógico.

O formato e conteúdo da mensagem a enviar pelo nó *Host* está representado na Figura 6.20.

SOF	Campo de arbitragem												
	Identificador												
	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	
0	0	1	P3	1	0	P2	P1	P0	1	0	DIR	RTR	
0	0	1	0	1	0	1	1	1	1	0	0	0	

Campo de controlo						Byte 1							
Reservado		DLC				Status				Endereço do Registo			
0	0	0	0	1	1	RSTD	EW	BM1	BM0	A3	A2	A1	A0
0	0	0	0	1	1	X	X	X	X	0	1	0	0

Byte 2								Byte 3							
Dados (bits 15 a 8) do registo de E/S								Dados (bits 7 a 0) do registo de E/S							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Figura 6.20 – Mensagem para habilitar o pino E/S.

ii) Após ter habilitado o pino de saída pretendido deve ser escrito (via CAN) no *Output Register* (endereço 03) o estado exigido para esse porto E/S do SLIO, que neste caso é o nível lógico alto ou '1'. Esta mensagem está representada pela Figura 6.21.

SOF	Campo de arbitragem												
	Identificador												
	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	
0	0	1	P3	1	0	P2	P1	P0	1	0	DIR	RTR	
0	0	1	0	1	0	1	1	1	1	0	0	0	

Campo de controlo						Byte 1							
Reservado		DLC				Status				Endereço do Registo			
0	0	0	0	1	1	RSTD	EW	BM1	BM0	A3	A2	A1	A0
0	0	0	0	1	1	X	X	X	X	0	0	1	1

Byte 2								Byte 3							
Dados (bits 15 a 8) do registo de E/S								Dados (bits 7 a 0) do registo de E/S							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Figura 6.21 – Mensagem para activar o pino de E/S.

### 6.1.3.2 MENSAGENS DOS SLIOS PARA O HOST

As mensagens dos nós SLIOs para a rede de actuação consistem em respostas a ordens ou pedidos de informação transmitidos pelo nó *Host* responsável por eles. O formato

destas mensagens é igual ao das transmitidas do *Host* para os SLIOs, divergindo apenas nos valores dos *bits* RTR e DIR, os quais possuem os valores RTR=0 e DIR=1, porque os SLIOS não podem pedir informação, limitando-se a executar ordens ou responder a pedidos de informação do *Host* que os comanda, pelo que não será abordado novamente o formato destas mensagens. As diferenças entre os vários tipos de *frames* possíveis entre o nó *Host* e os nós SLIO CAN estão resumidas no Anexo 4. Assim, este tipo de mensagem é identificado pelos três *bits* menos significativos conterem os valores ID2ID1ID0=101 conforme referido na Tabela 6.2.

Um exemplo de mensagem enviada por um nó SLIO CAN para o nó *Host* é a que é transmitida após conversão A/D realizada por esse nó a pedido do *Host*. O pedido de conversão por parte do *Host* é feito através de uma mensagem com o formato representado na Figura 6.19 e referindo o registo *Analogue Configuration Register* (ACR), explicado no Anexo 5, com o *bit* mais significativo do 2º *Byte* de dados (*bit* ADC), ao nível lógico ‘1’. A recepção desta mensagem por parte do nó SLIO cujo endereço físico corresponde ao valor dos *bits* (P3..P0) do campo de arbitragem, desencadeia o início de uma conversão A/D que termina com a transmissão de uma mensagem contendo o valor convertido, com formato igual ao representado na Figura 6.22.

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	DIR	RTR	DLC3	DLC2	DLC1	DLC0
0	1	P3	1	0	P2	P1	P0	1	0	1	0	0	0	1	1

Campo de dados															
Byte 1								Byte 2							
RSTD	EW	BM1	BM0	1	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

Campo de dados							
Byte 3							
AD1	AD0	0	0	0	0	0	0

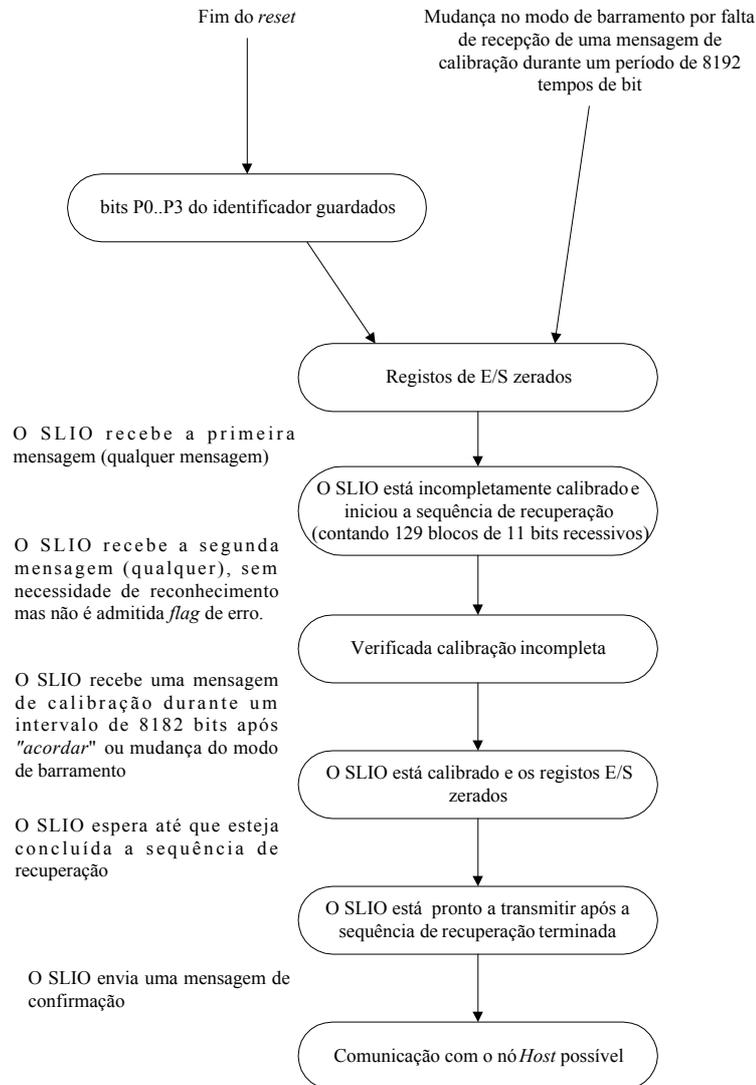
Figura 6.22 – Formato de uma mensagem do SLIO com o valor de uma conversão A/D.

## 6.1.4 Calibração dos SLIOS

Uma das principais características dos dispositivos SLIO CAN é a capacidade que estes possuem de auto-calibração, tomando como base mensagens que recebem do nó *Host* via barramento CAN. Durante a iniciação (após *reset* ou *wake-up*) qualquer mensagem é utilizada para calibrar o tempo de *bit* até que a calibração seja suficiente para receber mensagens correctamente. Posteriormente, o tempo de *bit* é calibrado e ajustado através de mensagens de calibração que possuem um identificador próprio e são transmitidas pelo nó *Host*.

O processo de calibração é composto pelas seguintes etapas: após *reset* ou mudança de modo, o SLIO deve receber pelo menos três mensagens para que possa calibrar o tempo de *bit*. A primeira mensagem é utilizada para determinar a duração do tempo de *bit* (calibração incompleta) entre duas transições consecutivas pelo que a sequência de *bits* deve conter ‘1010’. Após estar incompletamente calibrado o SLIO pode receber uma mensagem CAN válida, qualquer, e executar o comando sem dar conhecimento ao *Host*.

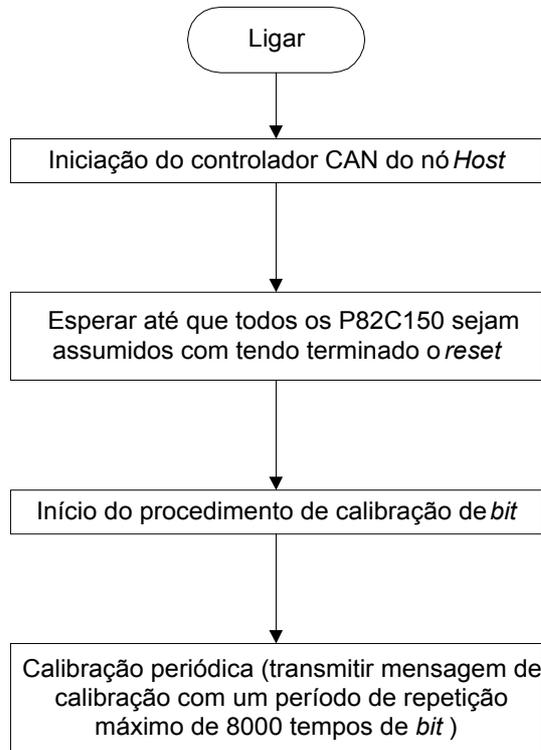
Com uma terceira mensagem CAN válida e uma mensagem de calibração o SLIO fica completamente calibrado e envia uma mensagem a “assinalar presença” ao *Host*. A mensagem de “assinalar presença” devolve o conteúdo dos dados do registo com endereço 0 (Anexo 4) podendo ser reconhecida pelo *Host* através de verificação do *bit* de *status* (RSTD) que é igual a ‘1’ para a mensagem de “assinalar presença”, e igual a ‘0’ para as outras *frames* de dados. Enquanto o SLIO estiver calibrado comporta-se como um nó CAN activo. Esta fase do processo de calibração está ilustrada na Figura 6.23.



**Figura 6.23 - Processo completo de calibração dos nós SLIO (P82C150) do sistema.**

Apenas nós SLIO CAN sendo calibrados por mensagens de calibração podem transmitir mensagens CAN. Uma vez que o SLIO não consegue transmitir sem estar correctamente calibrado, não consegue também “acordar” outros nós CAN ao longo do barramento do qual faz parte. Assim, para manter a rede “acordada”, deve ser transmitida regularmente pelo *Host* uma mensagem de calibração, com um período máximo de repetição de 8192 tempos de *bit* (duração de um *bit* medido pelo SLIO), sendo recomendada, pelo fabricante, a utilização de um período de repetição compreendido entre 3800 e 8000 tempos de bit [33].

Nesta aplicação foi utilizado um tempo de *bit* de 9  $\mu$ s (taxa de transferência de 110 Kbit/s), Anexo 6, e um período de calibração de 62,5 ms de acordo com os limites recomendados. Este tempo foi implementado por programação do período de interrupção do RTC [39], e da execução da respectiva rotina de serviço à interrupção a qual inclui o envio da mensagem de calibração pelo *Host* conforme será posteriormente referido numa secção dedicada a esta rotina. O procedimento completo de calibração executado pelo nó *Host* é ilustrado no diagrama da Figura 6.24.



**Figura 6.24 - Procedimento geral para sistemas em barramento CAN com nós possuindo o SLIO P82C150.**

O procedimento de calibração, ilustrado pela Figura 6.23, é executado pelo controlador CAN do *Host* (P80C592), sendo transparente para o programa executado no *Host*, excepto o carregar da mensagem para transmissão, verificação e confirmação, por parte do *Host*, de calibração com sucesso.

#### **6.1.4.1 MENSAGEM DE CALIBRAÇÃO**

De acordo com a recomendação do fabricante dos SLIOs utilizados, P82C150 da Philips, a mensagem de calibração deve possuir um formato próprio respeitando os seguintes itens:

- Ser transmitida por um nó controlado por cristal (*Host*);
- Possuir identificador: 000 1010 1010 (1 = recessivo; 0 = dominante);
- Possuir *bit* RTR ao nível lógico “0”;
- Possuir um campo de controlo com: DLC de 2 a 8;

- A primeira transição de recessivo para dominante, após o campo de controlo, deve ser seguida de outra transição de recessivo para dominante numa distância de exactamente 32 *bits* (com *stuff bits* inclusive);
- Exemplo de uma mensagem aconselhada [33] e que foi a utilizada:
  - DLC = 0010;
  - Primeiro *byte*: 1010 1010 (AAH);
  - Segundo *byte*: 0000 0100 (04H).

A mensagem utilizada no âmbito desta aplicação está representada na Figura 6.25.

SOF	Campo de arbitragem	Campo de controlo	Byte 1	Byte 2	Campo CRC
0	000 1010 1010 0	000 010	<u>1</u> 010 1010	0000 0100	000 0 1011 1000 00  <u>0</u>

Figura 6.25 – Mensagem de calibração.

**Nota:** estão sublinhadas as transições consideradas de maior importância no processo de calibração.

| - *stuff bit* inserido pelo controlador CAN.

## 6.2 SOFTWARE RESIDENTE NO HOST

Do *software* implementado na placa *Host* destacam-se as seguintes componentes:

- Iniciação do microcontrolador e controlador CAN interno;
- Atendimento a interrupções relacionadas com falha de energia, calibração dos nós SLIO CAN e comunicação CAN;
- Selecção do modo de controlo (local/remoto);
- Programa principal.

### 6.2.1 Iniciação

Durante a fase de iniciação do programa são definidas as fontes de interrupção “autorizadas” no decorrer do mesmo. Isto é realizado através de escrita em registos próprios para o efeito (IEN0, IEN1, IP0 e IP1) do microcontrolador conforme é descrito no Anexo 6.

Durante o procedimento de iniciação é configurado, também, o controlador CAN interno do microcontrolador P80C592, de acordo com os parâmetros de comunicação necessários. Este procedimento é realizado através de programação dos registos do Segmento de Controlo (Anexo 6) e inclui a definição dos seguintes itens:

- Filtro de aceitação;
- Tempo de barramento;
- Modo de saída;

- Interrupções (do controlador CAN);
- Modo da tensão de referência.

Os registos e respectivos valores utilizados no âmbito desta dissertação estão descritos no Anexo 6.

Após concluído o processo de configuração do controlador, são possíveis operações de recepção e transmissão via barramento CAN.

## 6.2.2 Atendimento a interrupções

Para cada tipo particular de interrupção foi desenvolvida uma rotina, tendo em conta os requisitos de cada uma das fontes de interrupção.

As fontes de interrupção utilizadas destinam-se a:

- Assinalar falhas de energia;
- Assinalar a recepção de mensagens CAN;
- Permitir a calibração do sistema SLIO CAN e acções de controlo temporizadas.

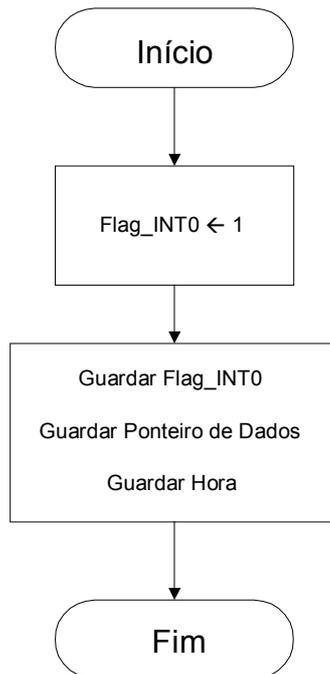
De seguida será explicada cada uma das rotinas de serviço à interrupção utilizadas pela ordem de prioridade que foi estabelecida, através de programação dos registos próprios do microcontrolador durante a iniciação, conforme descrito no Anexo 6.

### 6.2.2.1 ROTINA DE SERVIÇO À INTERRUPÇÃO NÃO MASCARÁVEL (NMI)

Para precaver situações em que haja falha de energia no sistema, foi implementado um processo que permite salvaguardar o estado das variáveis quando a falha ocorre e desta forma retomar o funcionamento, quando a anomalia for solucionada, em condições idênticas às verificadas na altura em que ocorreu a falha. Esta rotina é responsável pelo atendimento de interrupções provenientes do circuito de monitorização de tensão de alimentação explicado no capítulo 5, secção 5.1.1.3, e que tem por base o circuito integrado DS1231 [41].

A sequência de instruções executadas sempre que esta rotina é solicitada, está representada no fluxograma da Figura 6.26 e consiste nos seguintes pontos:

1. Sempre que ocorra esta interrupção é activada uma *flag*, que a identifica no decorrer do programa, permitindo que sejam realizadas tarefas com ela relacionadas;
2. É guardada em memória a hora, a *flag* anteriormente referida e o ponteiro de dados para que, após a reposição da alimentação, o sistema inicie com os dados que possuía antes da interrupção ter sido detectada;
3. O microcontrolador passa ao modo livre, *IDLE*, do qual sai quando for feito o *reset* do mesmo por *hardware* [41], conforme explicado no capítulo 5.



**Figura 6.26 – Fluxograma para a rotina de serviço à interrupção NMI.**

### 6.2.2.2 ROTINA DE SERVIÇO À INTERRUPTÃO CAN

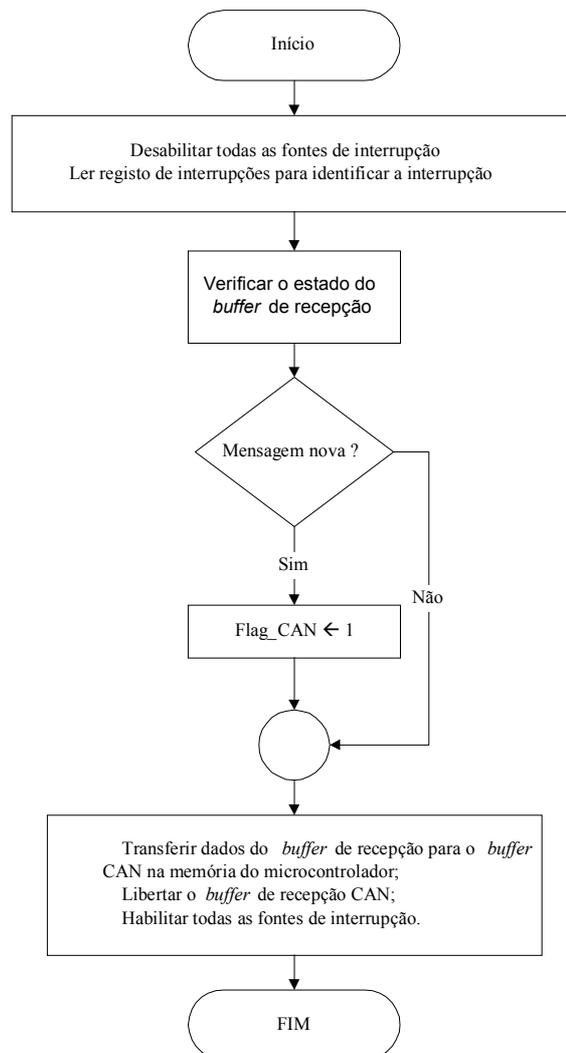
Esta rotina é responsável pelo atendimento a interrupções provenientes do controlador CAN.

Assim como a anterior, esta interrupção é de extrema importância uma vez que é ela que assinala a existência de novas mensagens CAN destinadas ao nó onde é gerada. A sequência de operações que esta rotina executa é representada no fluxograma da Figura 6.27 e é a seguinte:

1. Sempre que ocorre uma interrupção destas, são inibidos os outros tipos de interrupção e é verificado o conteúdo do registo interno de interrupções, para que assim seja possível actuar sobre a linha de interrupção.
2. É verificado se a interrupção é devida a uma recepção de dados e se o bloco de dados em questão é o primeiro.
  - Se o bloco de dados for o primeiro, activa-se uma *flag* que identifica a ocorrência de uma interrupção CAN, e o registo do *buffer* de recepção é consultado para se verificar quantos *bytes* compõem a mensagem. Posteriormente, é transferida esta informação do *buffer* de recepção para um *buffer* de dados previamente reservado na memória do microcontrolador. Por fim, o *buffer* de recepção é libertado para que possa ser utilizado na recepção da próxima mensagem.
  - Se o bloco de dados não for o primeiro, isto é, se for uma mensagem segmentada, é consultado o registo do *buffer* de recepção para se

verificar quantos *bytes* da mensagem ainda faltam receber. Posteriormente é transferida esta informação do *buffer* de recepção para um *buffer* de dados previamente reservado na memória do microcontrolador, decrementando uma variável global que indica o número de blocos ainda por receber.

3. São habilitados os outros tipos de interrupção.



**Figura 6.27 – Fluxograma da rotina de serviço a uma interrupção CAN.**

### 6.2.2.3 ROTINA DE SERVIÇO À INTERRUPÇÃO DO RELÓGIO DE TEMPO REAL (RTC)

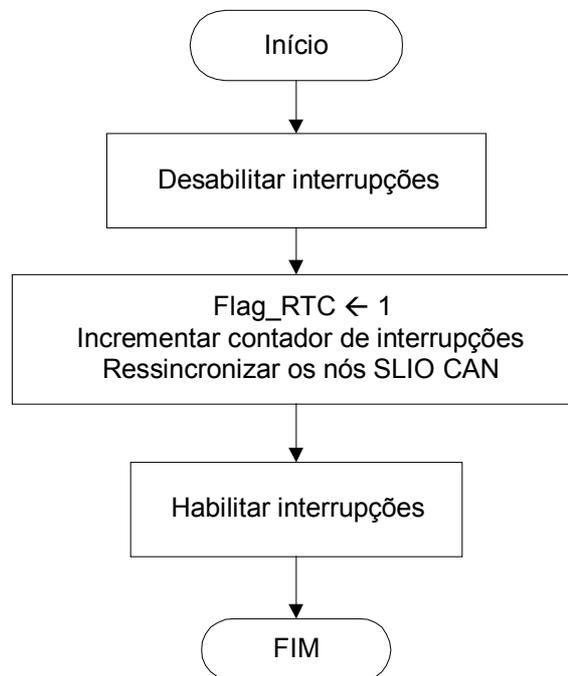
Esta rotina é de extrema importância uma vez que é recorrendo a ela que se implementa o processo de calibração dos nós SLIO CAN pelos quais o nó *Host* é responsável. Assim, esta rotina é chamada a ser executada em intervalos fixos de tempo (62,5 ms) que são assinalados por interrupções geradas pelo RTC DS1287, os quais são programados durante a iniciação do microcontrolador de acordo com os requisitos de

calibração dos nós SLIO CAN.

Esta rotina possibilita, também, o desencadear de tarefas de actuação de natureza temporal, recorrendo a contadores de interrupções conseguindo-se assim tempos de actuação múltiplos do período de interrupção gerado pelo RTC.

A sequência de instruções executadas, sempre que esta rotina é solicitada, está representada no fluxograma da Figura 6.28 e consiste nos seguintes pontos:

1. Sempre que ocorra esta interrupção são desabilitados os outros tipos de interrupção e é activada uma *flag*, que a identificará no decorrer do programa, permitindo que sejam realizadas tarefas com ela relacionadas;
2. É incrementado um contador de interrupções para o caso de necessidade de actuações de natureza temporal, permitindo assim tempos múltiplos do período com que a RTC gera uma interrupção; O *reset* deste contador é efectuado durante o decorrer do programa principal residente no *Host*;
3. É enviada uma mensagem de calibração para a rede CAN por forma a manter os nós SLIO CAN activos na rede e conseqüentemente prontos a comunicar via barramento CAN. Este requisito foi abordado anteriormente na secção 6.1.4, dedicada ao processo de calibração dos nós SLIO CAN;
4. Por fim são habilitados todos os tipos de interrupção prosseguindo o funcionamento normal do programa.



**Figura 6.28 – Fluxograma para a rotina de serviço à interrupção RTC.**

## 6.2.3 Selecção do modo de Controlo (Local/Remoto)

Para precaver situações em que possa ocorrer erro ou necessidade de ser realizada manutenção da estação supervisora (PC-supervisor), foi implementada uma forma de controlo do sistema, durante a anomalia, ser transferido temporariamente para o nó *Host*.

Para que o nó PC-supervisor tenha o controlo da rede de actuação, deve transmitir uma mensagem com identificador CAN bem definido, que quando recebida pelo nó *Host* faz com que este execute tarefas de acordo com as mensagens CAN recebidas a partir do nó PC-supervisor (controlo remoto), o que corresponde à situação normal de funcionamento. Existe no entanto outra mensagem, também com um identificador bem definido, que quando recebida pelo nó *Host* provoca um efeito contrário ao anterior (controlo local) fazendo com que o controlo da rede SLIO CAN seja realizado pelo nó *Host*.

O formato das mensagens que devem ser enviadas pelo PC-supervisor, para o nó *Host*, é igual ao das mensagens de *set-point* referido na secção 6.1.2.3, utilizando o primeiro *byte* do campo de dados para indicar o modo pretendido. Assim, o primeiro *byte* de dados com o valor hexadecimal 55h corresponde a controlo local e o valor AAh a controlo remoto conforme ilustra a Figura 6.29.

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0
X	1	X	1	x	x	x	x	1	1	0	0	0	0	0	1

Campo de dados							
Byte 1 – Local/Remoto							
1	0	1	0	1	0	1	0

**Figura 6. 29 - Formato da mensagem para execução do programa em modo remoto.**

Após recepção de uma destas mensagens é actualizada uma *flag* global que assinala o modo de funcionamento que deve ser executado. Considerou-se Flag\_Global=1 para modo de controlo local e Flag\_Global=0 para o modo de controlo remoto.

Por *default*, após o *reset*, o *Host* executa o programa correspondente ao controlo remoto. Quando está a funcionar no modo de controlo local o nó *Host* realiza controlo de temperatura do tipo liga/desliga efectuando aquisição de temperatura e comandando vários actuadores (Ventilador, janelas, etc.) através dos nós SLIO CAN. O intervalo de temperaturas ( $T_{\min}$  a  $T_{\max}$ ) admissíveis bem como o intervalo de tempo entre actualizações consecutivas são definidos durante a iniciação e constantes durante o decorrer do programa no caso de controlo local. O intervalo de tempo entre actualizações consecutivas é implementado conforme referido na rotina de serviço á interrupção RTC (secção 6.2.2.3).

## 6.2.4 Programa principal

O programa residente no nó *host* é executado em ciclo fechado, respondendo a eventos que são assinalados pelas interrupções anteriormente referidas. Assim, o que acontece são respostas a eventos, sendo estas desencadeadas numa fase inicial pelas rotinas de

serviço à interrupção com o correspondente assinalar de uma *flag*, em função da qual é executado um conjunto de acções.

O conjunto de acções a serem realizadas pelo nó *Host* depende do modo de controlo (local/remoto), do tipo de interrupção recebido e no caso de interrupção CAN do conteúdo da mensagem, o qual pode estar relacionado com comandos ou pedidos de informação transmitidos pelo nó Supervisor (PC) ou com mensagens provenientes dos nós SLIO CAN. A Figura 6.30 (página 119-A) representa o fluxograma do programa.

## 6.3 SOFTWARE RESIDENTE NO NÓ *SLAVE MOTOR*

### 6.3.1 Programa principal

Apesar de o hardware do nó *Slave Motor* ser idêntico ao do nó *Host*, conforme foi referido na secção 5.1.2, em termos de *software* existem diferenças importantes, uma vez que estes nós realizam funções distintas. O nó *Slave Motor* foi programado para desempenhar funções relacionadas com o controlo de velocidade de um motor de indução monofásico, neste caso de um ventilador. O programa residente no nó *Slave Motor* é, no entanto, idêntico ao descrito para o nó *Host*, relativamente aos tipos de interrupções e respectivas rotinas de serviço, sendo no entanto diferente o tempo da interrupção da RTC, que neste caso é de aproximadamente 1 ms (976,5625  $\mu$ s) e a inexistência de reconfiguração dos SLIOs.

O código residente no nó *Slave Motor* foi desenvolvido com o propósito de, com base na recepção de um tipo de mensagem CAN (*set-point*, Tabela 6.2), contendo um conjunto de *set-points*, modular numa das saídas PWM do microcontrolador, uma onda sinusoidal com determinada frequência e amplitude, em função dos valores recebidos. A Figura 6.31 (página 119-B) contém o fluxograma do programa residente neste nó.

Quanto aos tipos de mensagens CAN de Alarme e RTC, o formato e interpretação por parte do nó *Slave Motor* é semelhante ao explicado para o nó *Host*. A comunicação com nós SLIO CAN, neste caso é inexistente. As mensagens provenientes dos nós SLIO CAN (tipo De\_SLIO) e as mensagens de configuração são ignoradas por este nó. A configuração é feita através da mensagem do tipo *set-point* uma vez que esta é suficiente para realizar a tarefa principal do nó, que é gerar um sinal PWM para controlo de velocidade de um motor de indução.

O formato e conteúdo da mensagem que deve ser enviada pelo PC-supervisor, com o objectivo de gerar determinada saída PWM através do nó *Slave Motor* estão ilustrados na Figura 6.32.

Campo de arbitragem												Campo de controlo			
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0
x	x	X	x	x	x	x	x	1	1	0	0	0	0	0	1

Campo de dados						
Byte 1						
Liga/desliga	V5	V4	V3	V2	V1	V0
x	V5	V4	V3	V2	V1	V0

Figura 6.32 – Formato da *frame* para ajuste da frequência a sintetizar.

Nesta mensagem destaca-se o significado do 1º byte do campo de dados, o qual indica a frequência pretendida para a onda sinusoidal a sintetizar (V5..V0) permitindo valores de frequência entre 0 e 63 Hz. Além do valor da frequência pretendida para a onda sinusoidal a sintetizar, o 1º byte indica também o estado pretendido para a carga, neste caso ventilador, ou seja se ligado (Liga/desliga=1) ou desligado (Liga/desliga=0). Com a recepção destes *set-points* e através do programa residente no microcontrolador é possível obter à saída do canal PWM pretendido (PWM0) a onda desejada, que consiste numa onda sinusoidal modulada em largura de impulso

Quanto à interpretação das mensagens de actuação e aquisição, apesar de não ter sido implementado, é possível actuar diversos portos ou adquirir sinais utilizando para tal o conversor A/D que o microcontrolador possui, tornando os valores dessas leituras acessíveis através do barramento CAN ao PC-supervisor ou outros nós (com microcontrolador) no barramento.

Nas próximas secções será feita uma abordagem à forma como é gerado, pelo microcontrolador, o sinal PWM que permite o accionamento do módulo de IGBTs, PM20CSJ060, e conseqüentemente o controlo de velocidade do motor ligado à saída deste módulo.

## 6.3.2 Modulador PWM

O modulador, a partir de informação sobre a amplitude e frequência pretendidas para a onda a sintetizar, produz os sinais nas saídas PWM do microcontrolador necessários para accionar os semicondutores de potência do inversor monofásico (ou trifásico), de acordo com a técnica de modulação pretendida.

Numa onda sinusoidal monofásica é necessário conhecer a informação relativa a pelo menos 90°, da forma de onda, para reconstruir com exactidão os restantes 270°. Num sistema trifásico, basta conhecer a informação relativa aos primeiros 30° das três fases para conseguir reconstruir os restantes 330° [60]. Explorando estas simetrias, é possível simplificar os algoritmos que implementam as técnicas de PWM, reduzir o tamanho de tabelas que guardam a informação necessária à execução dos algoritmos e representar o valor dos ângulos com maior precisão para o mesmo número de *bits*.

### 6.3.2.1 MODULAÇÃO DE LARGURA DE IMPULSO SINUSOIDAL

No método de modulação sinusoidal, é possível determinar os ângulos de comutação em tempo real, desde que se armazene um número conveniente de amostras em memória. Essas amostras são calculadas com base na seguinte expressão:

$$F = \text{sen}(\omega t_k) \quad (6.1)$$

A equação utilizada para calcular a largura dos pulsos neste tipo de modulação a qual foi deduzida em [61, 62] e utilizada noutras aplicações [60] é a seguinte:

$$T_k = \frac{T_s}{2} [1 + m_a \times F(\omega t_k)] \quad (6.2)$$

Sendo  $m_a$  o índice de modulação (variável entre 0 e 1),  $T_k$  a largura de pulso,  $T_s$  a largura de pulso máxima admissível, neste caso 255 [37],  $\omega$  a frequência angular da

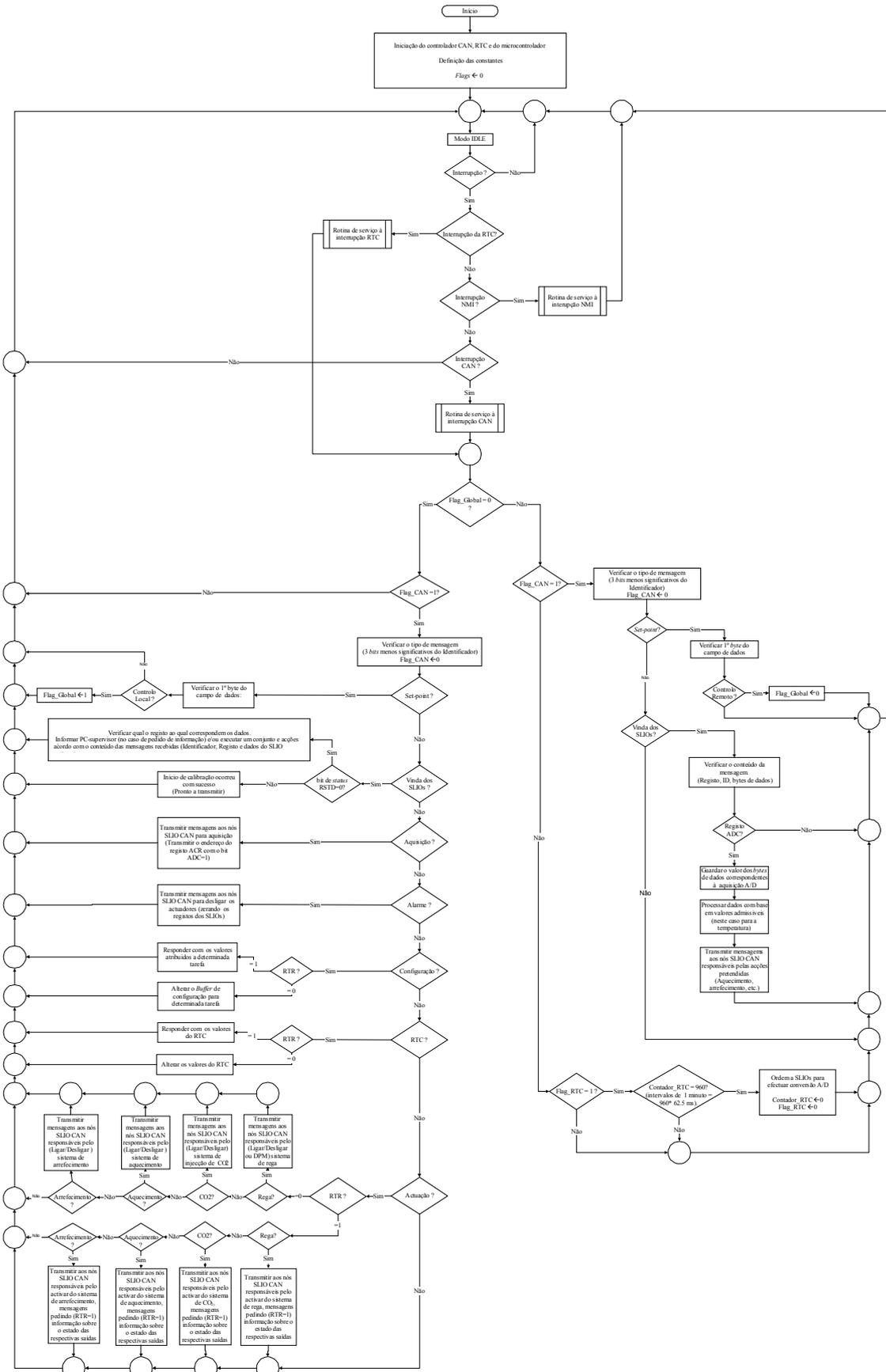


Figura 6.30 – Fluxograma do programa residente no nó Host.

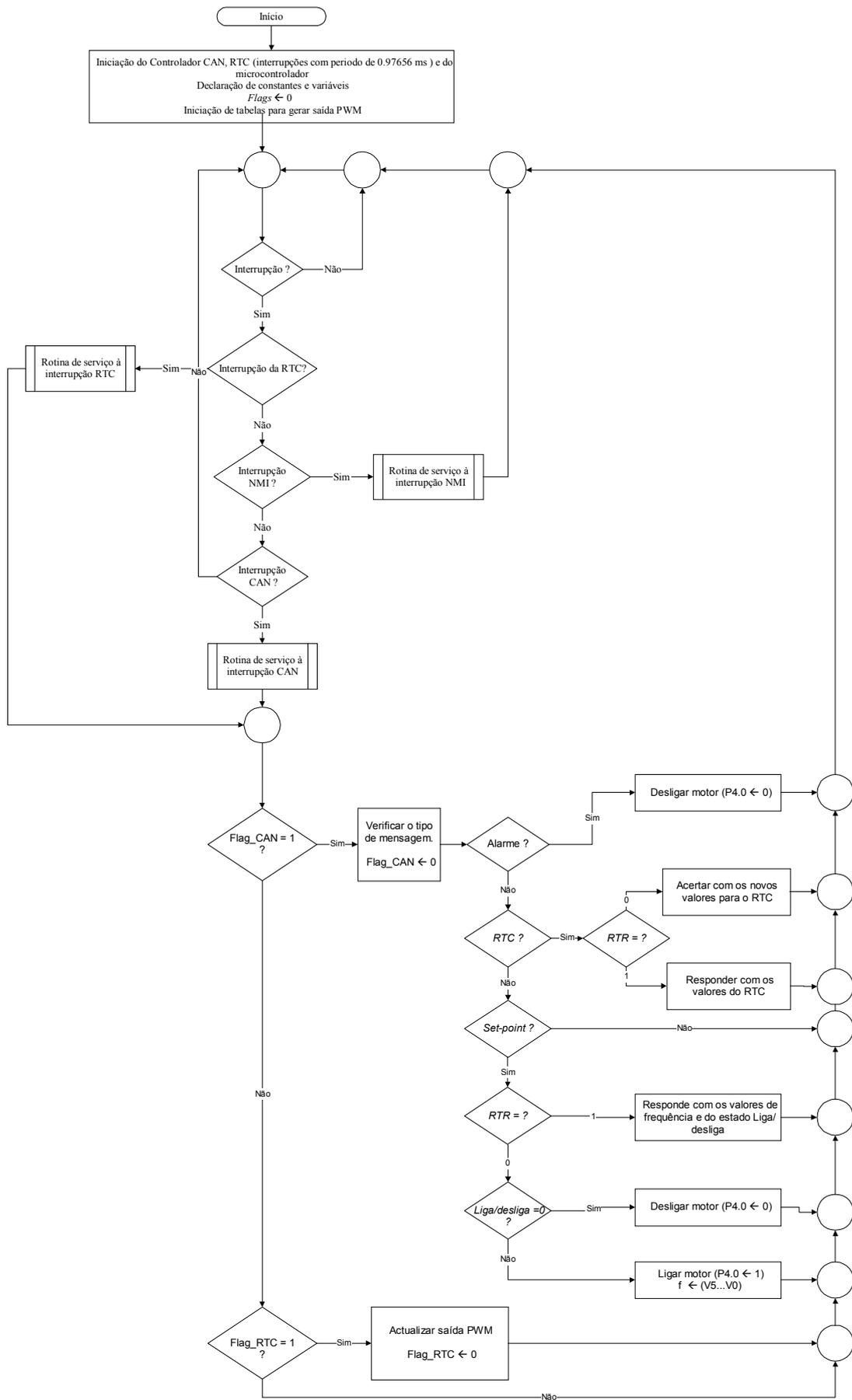
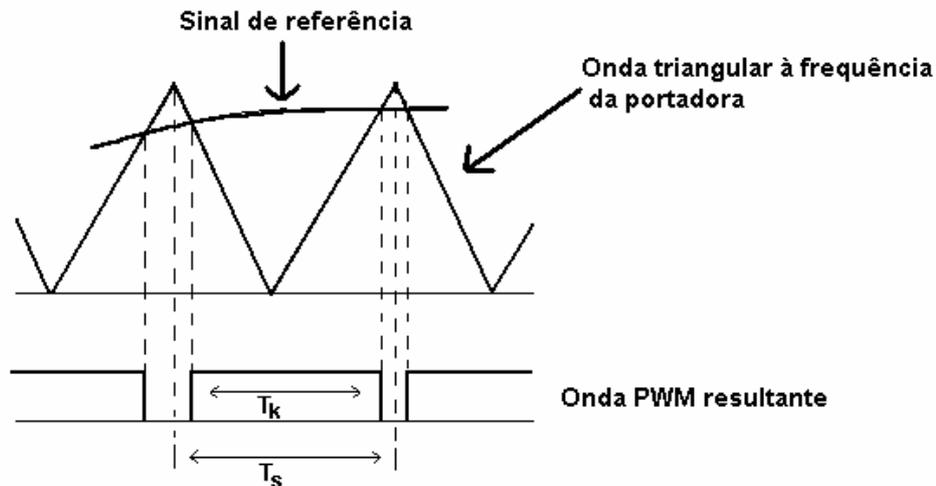


Figura 6.31 – Fluxograma do programa residente na placa Slave Motor.

onda sintetizada e 'F' a função que representa o sinal de referência equação (6.1) e que depende da técnica de modulação que se pretende sintetizar. A Figura 6.33 ilustra o significado e a relação entre  $T_k$  e  $T_s$ .



**Figura 6.33 – Comutação do sinal PWM.**

Para um dado pedido de frequência ( $\omega=2\pi f$ ), o ângulo cujo seno se pretende modular  $\alpha_k=\omega t_k$ , é incrementado de acordo com a seguinte expressão [60]:

$$\alpha_{(k+1)} = \alpha_k + \omega T_p = \alpha_k + 2\pi f/f_p \quad (6.3)$$

sendo  $f_p (=1/T_p)$  a frequência de comutação também designada por frequência da portadora e  $f$  a frequência a sintetizar.

### 6.3.2.2 FREQUÊNCIA DO SINAL PWM

O microcontrolador utilizado neste trabalho, P80C592 [37], possui duas saídas PWM, cuja frequência é definida através de escrita no registo, *Prescaler Frequency Control Register* (PWMP), que o mesmo possui para o efeito. O valor deste registo está relacionado com a frequência dos sinais PWM de acordo com a equação (6.4).

$$f_{\text{PWM}} = \frac{f_{\text{CLK}}}{2 \times (1 + \text{PWMP}) \times 255} \quad (6.4)$$

Assim, utilizando um cristal de  $f_{\text{CLK}}=11,0592$  MHz a gama de frequências possível para a portadora varia desde 84,7 Hz até 21,6 kHz.

Nesta aplicação escolheu-se  $f_{\text{PWM}} = 2409$  Hz respeitando os requisitos do módulo de IGBTs e tendo em atenção o fim em vista, ou seja, accionamento de máquinas eléctricas.

### 6.3.2.3 VALOR DOS IMPULSOS

O microcontrolador utilizado possui também dois registos, um para cada canal PWM, que permitem definir a largura dos pulsos. Estes registos de 8 bits, cujas designações são *Pulse Width Register 0* (PWM0) e *Pulse Width Register 1* (PWM1), contêm o valor da largura de pulso instantânea que pode variar entre 0 a 255. Correspondendo o valor 0 ao sinal de nível alto, 5 V, e o valor 255 ao de nível baixo, 0 V, durante todo o período,

( $T_p = 1/f_{PWM}$ ). Assim, utilizando 8 bits para representar a largura dos impulsos ( $N_p=255$ ), a resolução em termos de amplitude da onda sintetizada é de  $T_p/256$ . Quanto maior for o número de bits utilizado na representação dos impulsos, melhor será a resolução em termos de amplitude e menor será a frequência da portadora [60]. De seguida é reescrita a equação (6.2) em termos de valor do registo, PWM0, utilizado:

$$N_k = \frac{N_p}{2} [1 + m_a \times F(\omega t_k)] = \frac{255}{2} [1 + m_a \times F(\omega t_k)] \quad (6.5)$$

#### 6.3.2.4 A MEMÓRIA

Tendo como objectivo acelerar o algoritmo que implementa a técnica de modulação, foram guardadas duas tabelas em memória com conteúdos distintos, estando no entanto relacionadas.

Uma das tabelas (**Tabela\_Senos**), definida durante o início do programa e fixa durante o decorrer do mesmo, é composta por elementos com o valor da função *seno* dos ângulos correspondentes a uma onda a sintetizar com frequência,  $f$ , de 1 Hz. Foi também estipulada uma frequência para leitura da tabela e correspondente actualização do registo PWM,  $f_p$ , de 1024 Hz a qual é implementada através de programação do RTC [39], para gerar interrupções com intervalos de 976,5625 ms. Assim, para uma frequência de 1 Hz, são possíveis durante um segundo 1024 amostras, ou seja, 1023 incrementos de ângulo obedecendo à equação (6.3).

Não se deve confundir entre a frequência de actualização do registo PWM ( $f_p=1024$  Hz), e a frequência do sinal PWM ( $f_{PWM}=2409$  Hz). Assim, existe um número, ( $f_{PWM}/f_p$ ), de períodos do sinal PWM consecutivos de igual largura de pulso, por não ser feita a actualização do registo PWM0 durante um período do sinal PWM, mas sim, durante um múltiplo desse período ( $T_p/T_{PWM}$ ).

Uma segunda tabela (**Tabela\_PWM**), com o mesmo número de elementos da anterior, armazena o valor dos impulsos PWM e é lida com intervalos de tempo de  $T_p = 976,5625$  ms. Os elementos guardados nesta tabela são calculados de acordo com a equação (6.5) e correspondem aos valores pelos quais o registo PWM utilizado, PWM0, é actualizado sempre que ocorra a interrupção do RTC.

Ao contrário da tabela de senos, esta, para além de ser actualizada no início com uma frequência de  $f=50$  Hz, é actualizável em qualquer momento, desde que para tal o microcontrolador receba uma mensagem CAN, contendo um novo valor de frequência ( $f$ ). O formato desta mensagem CAN foi abordado na secção 6.1.2.3.

Após receber a mensagem, indicando novo valor de frequência, é feita a actualização da tabela que armazena o valor dos impulsos apenas nas posições necessárias. Por exemplo se a frequência desejada for  $f=25$  Hz, apenas serão actualizadas as posições, de 0 a 1023, com incrementos de 25, ou seja, as posições 0, 25, 50, 75, ... 975, 1000. Posteriormente sempre que ocorram interrupções da RTC, com intervalos de 976,5625 ms, apenas serão lidas estas posições, ignorando as restantes. Utilizando este método para gerar sinais PWM é possível sintetizar ondas sinusoidais com frequências que podem variar de 1 Hz até 64 Hz, devido aos 6 *bits* da mensagem CAN, tendo este limite máximo sido estipulado de acordo com as características do motor utilizado (frequência nominal de 50 Hz).

### 6.3.2.5 PROGRAMA PARA IMPLEMENTAR O MODULADOR PWM

Na Figura 6.34 está representado o fluxograma do programa implementado para gerar o sinal PWM, numa das saídas (PWM0) do microcontrolador P80C592 residente no nó *Slave Motor*, para a frequência que se pretenda sintetizar:

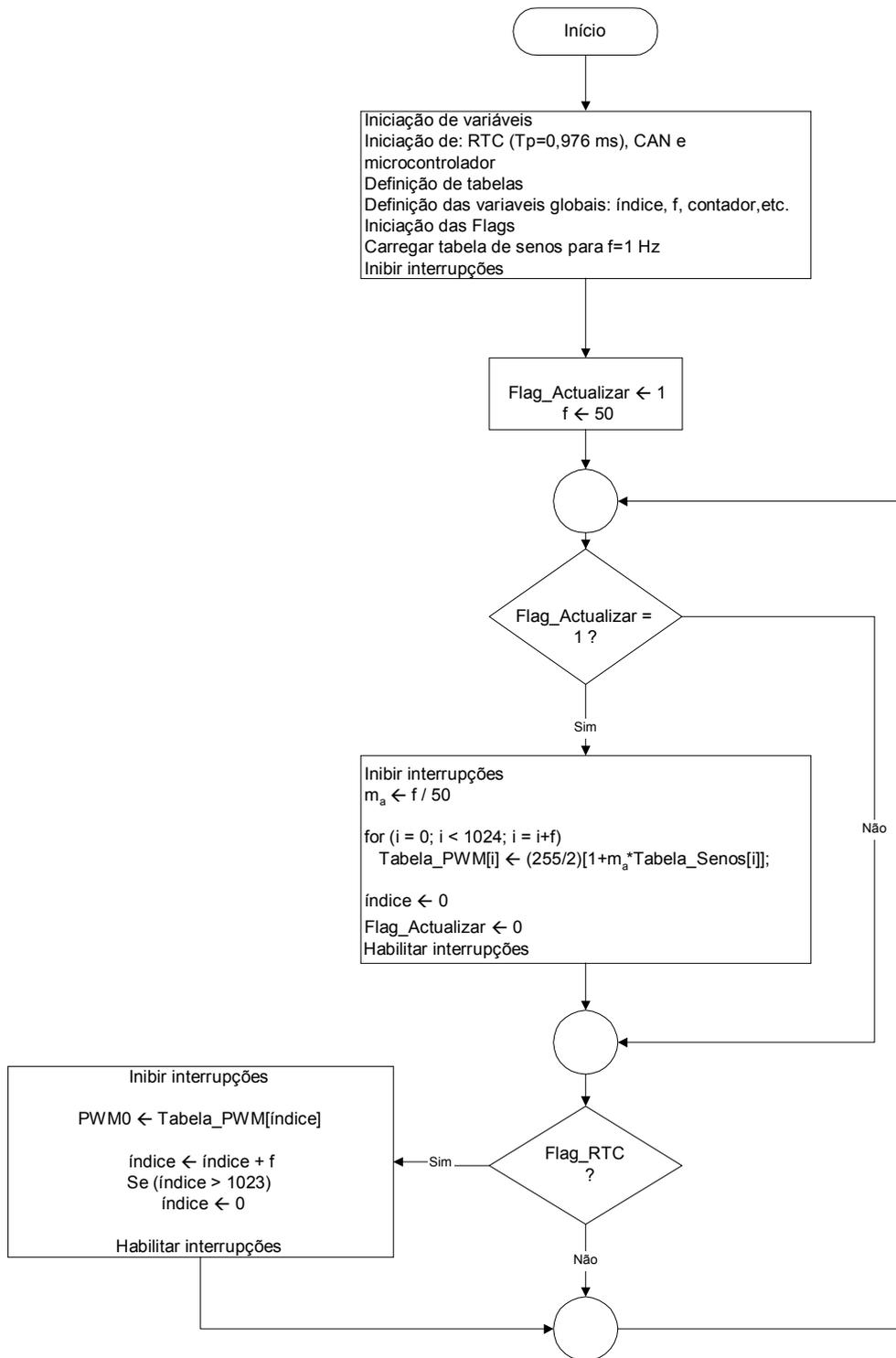


Figura 6.34 – Fluxograma do código para sintetizar uma onda sinusoidal com frequência  $f$ .

## 7 TESTES E RESULTADOS

Pelo facto de no momento o sistema desenvolvido ainda não ter sido implementado numa estufa real, as conclusões aqui apresentadas fundamentam-se em ensaios preliminares.

### 7.1 REDE CAN

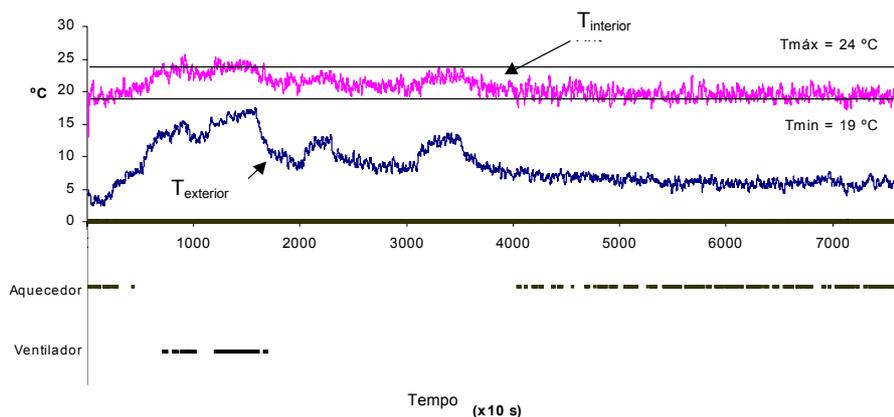
Nos ensaios realizados com a rede SLIO CAN verificou-se que a comunicação diferencial, utilizando uma taxa de transferência 110 Kbit/s (Anexo 6), foi conseguida de forma bastante satisfatória. Os nós SLIO CAN não “perderam” o sincronismo e as acções de controlo não foram postas em causa porque, prevendo esta situação, o *Host* (emissor) reenviava o comando “perdido” sempre que não obtinha uma mensagem de confirmação por parte dos nós SLIO CAN (receptores).

Os ensaios consistiram no registo de dados, via comunicação por RS-232C com um PC, referentes ao controlo de temperatura efectuado pelo nó *Host* (modo local), usando a rede SLIO CAN (dois nós) através da qual efectuou aquisição e controlo (accionamento, por relés, de um ventilador para arrefecimento e de um aquecedor eléctrico) do ambiente de uma estufa, propositadamente construída para a realização destes ensaios, Figura 7.1. Quanto às características da estufa utilizada, esta é do tipo túnel e as suas dimensões são: largura = 55 cm; altura = 50 cm; comprimento = 1 m.



**Figura 7.1 – Fotografia da estufa utilizada durante os testes da rede CAN.**

Os valores foram registados com amostragens separadas por intervalos de tempo de 10 segundos (teste mais exigente do que as condições normais em estufas reais onde o intervalo de tempo é, em muitas aplicações, cerca de 1 minuto). O sistema funcionou durante um período ininterrupto de 7 dias sem falha de sincronismo, tendo sido desligado (*reset*) por ordem expressa do operador. A Figura 7.2 mostra valores adquiridos durante o período de ensaios do sistema SLIO CAN.



**Figura 7.2 - Amostras da actuação feita utilizando a rede SLIO CAN implementada no âmbito desta dissertação (temperatura e estado dos actuadores).**

O tráfego de mensagens, na rede, durante os ensaios do sistema é composto pelas mensagens de calibração para os nós SLIO CAN (enviadas pelo nó *Host* com intervalos de 62,5 ms) e pelas mensagens de comando enviadas pelo nó *Host* com intervalos de 10 segundos após aquisição e comparação do valor da temperatura lida ( $T_{lida}$ ) na estufa com os *set-points* de  $T_{máx} = 24\text{ °C}$  e  $T_{mín} = 19\text{ °C}$ , definidos para as temperaturas máxima e mínima, respectivamente. Durante os ensaios foram utilizadas as seguintes mensagens de comando:

- Aquecer a estufa (activar o porto E/S 4 do SLIO 0 e desactivar o porto 4 do SLIO 1) se  $T_{lida} < T_{mín}$ ;
- Arrefecer/ventilar a estufa (desactivar o porto E/S 4 do SLIO 0 e activar o porto 4 do SLIO 1) se  $T_{lida} > T_{máx}$ ;
- Anular actuações (desactivar o porto E/S 4 do SLIO 0 e o porto 4 do SLIO 1) se  $T_{mín} < T_{lida} < T_{máx}$ .

Os actuadores utilizados nestes ensaios foram um ventilador e um aquecedor eléctrico, uma vez que os mesmos foram feitos tendo em atenção apenas o controlo de temperatura e foram realizados durante a estação de Inverno (Janeiro de 1999).

Para testar a robustez do sistema em condições de ruído foi ligada, no ponto de alimentação do nó *Host*, uma extensão com aproximadamente 50 metros de comprimento no extremo oposto da qual, foram ligados vários dispositivos eléctricos de natureza indutiva, nomeadamente: uma rebarbadora (720 W, 11000 rpm) e um berbequim (400 W) bem como dispositivos de natureza não indutiva tais como uma lâmpada fluorescente (18 W). Estes dispositivos foram ligados e desligados várias vezes, em simultâneo ou alternadamente, sem que a rede SLIO CAN manifestasse qualquer alteração de comportamento por esse facto.

Quanto ao consumo de energia, verificou-se que as placas “inteligentes” consomem um valor de corrente próximo de 100 mA e as placas SLIO CAN valores próximos de 50 mA e 100 mA, actuando sem e com carga, respectivamente.

## 7.2 CIRCUITOS DE ACCIONAMENTO

### 7.2.1 Solução utilizando relés

Utilizando a solução de accionamento por relés, o comportamento foi o esperado porque o accionamento dos actuadores foi desempenhado com sucesso.

### 7.2.2 Accionamento do motor de indução monofásico

#### 7.2.2.1 AMBIENTE DE TESTE

Os ensaios do accionamento do motor de indução monofásico foram realizados com base na infra-estrutura ilustrada na Figura 7.3. Assim, os dados obtidos durante os ensaios foram registados em tempo real e transferidos do osciloscópio para o PC via RS-232C utilizando o programa PROCOMM. Posteriormente os ficheiros contendo esses dados foram convertidos num formato próprio e importados para o *package* de processamento de sinal – DADiSP para que fosse possível fazer a respectiva análise, nomeadamente a construção do espectro de frequências dos sinais resultantes.

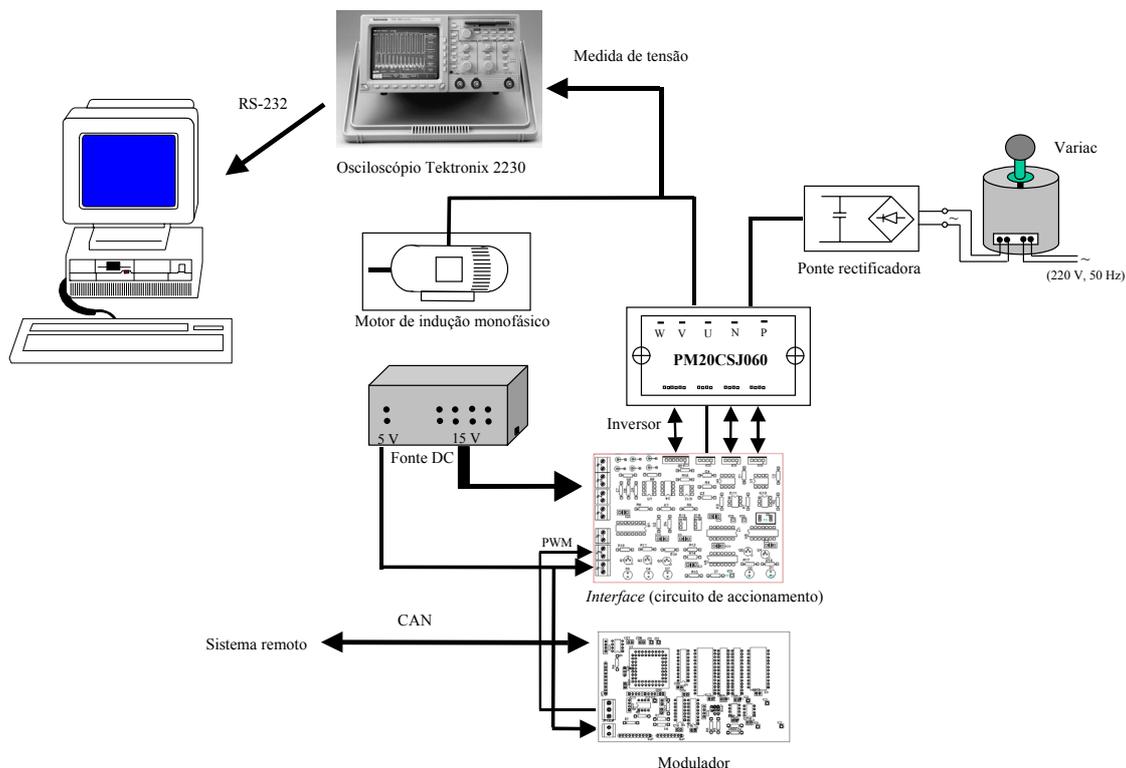


Figura 7.3 – Infra-estrutura utilizada na fase de ensaios.

O motor monofásico utilizado é de pequena potência apresentando os seguintes valores na sua chapa de características:

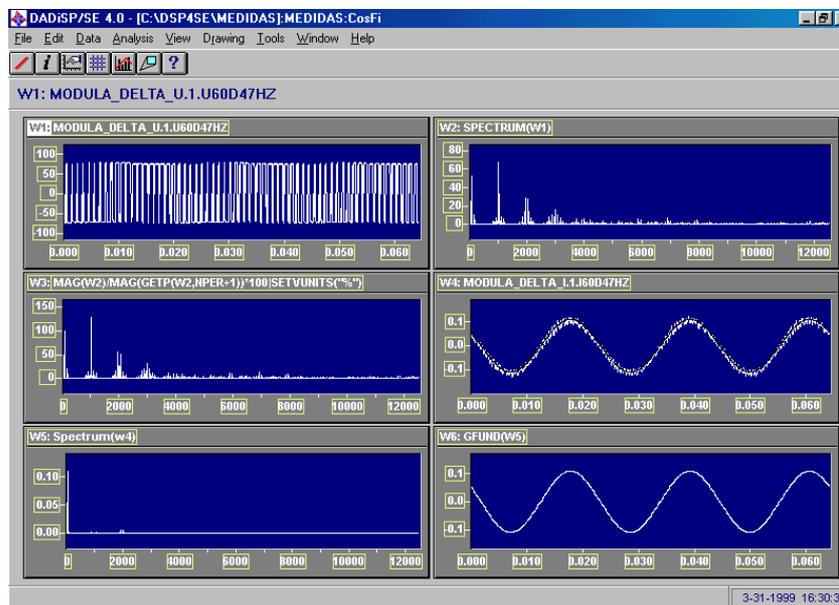
- 0,25 HP;
- 1370 rpm;
- 220 V / 50 Hz;
- 1,6 A.

Antes de fazer referência aos ensaios realizados, será feita uma descrição de algumas das ferramentas utilizadas durante a realização dos mesmos.

## O DADiSP

O DADiSP (produto da DSP *Development Corporation*) é uma ferramenta de *software* gráfica largamente utilizada para visualização, análise e manipulação de dados que podem estar dispostos em séries, matrizes, tabelas e imagens gráficas.

Corre em ambiente *Windows* apresentando uma organização celular, onde cada célula contém a definição de um sinal e uma imagem (Figura 7.4). O sinal pode ser especificado basicamente de dois modos: a partir de ficheiros de amostras armazenadas em disco e através de expressões analíticas, que relacionam os dados das diferentes células de uma forma idêntica a uma folha de cálculo (sem necessidade de recurso a programação). Existe um grande número de funções pré-definidas (matemáticas, estatísticas, trigonométricas, matriciais, para análise de Fourier e para gráficos de duas, três ou quatro dimensões), podendo ainda o utilizador criar menus e macros, para além de outras funções necessárias para a sua análise.



**Figura 7.4 – Aspecto gráfico do DADiSP 4.0.**

O DADisp serviu essencialmente para realizar a análise harmónica das formas de onda da tensão do motor.

## O Osciloscópio

Para visualizar, analisar e registar as formas de onda da tensão foi utilizado um osciloscópio – Tektronix2230. Trata-se de um osciloscópio de 100 MHz de largura de banda, memória digital, frequência de amostragem até 20 MHz, que inclui todas as capacidades de um osciloscópio convencional. Como características destacam-se: gravação de sinais do canal 1 e/ou 2 em memória não volátil; escolha do número de amostras de um sinal a armazenar (1 K ou 4 K); cursores para obter medidas de tensão, diferença de tempos, atrasos, etc.; visualização no painel frontal das escalas V/DIV e S/DIV e das medições directas dos cursores; comunicação via RS-232C, permitindo a transferência de ficheiros contendo informação relativa às formas de onda armazenadas e permitindo ainda o comando do aparelho através do PC.

Nas secções seguintes deste capítulo serão apresentados os ensaios e resultados obtidos após a implementação do algoritmo de modulação de largura de impulso bem como do *hardware* projectado para o accionamento do motor de indução monofásico, conforme foi descrito nos capítulos 5 e 6.

### 7.2.2.2 TESTE DO *HARDWARE* IMPLEMENTADO

O *hardware* implementado para controlo de velocidade do motor foi testado, numa primeira fase, usando modulação de onda quadrada [60], tendo sido utilizado o circuito integrado NE555N conforme mostra o esquemático ilustrado na Figura 7.5.

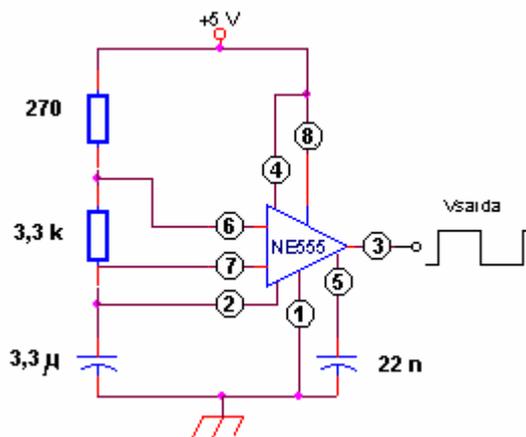
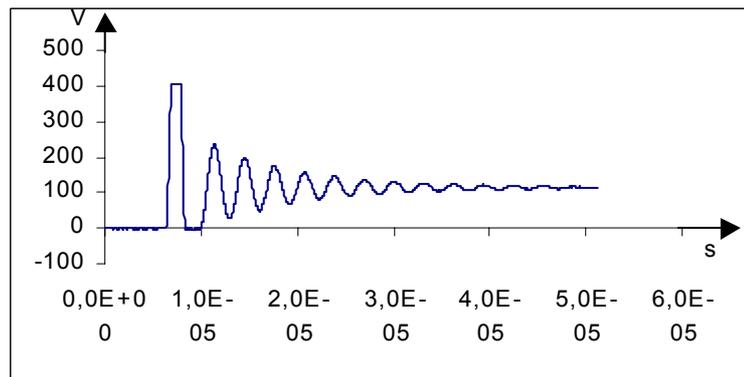


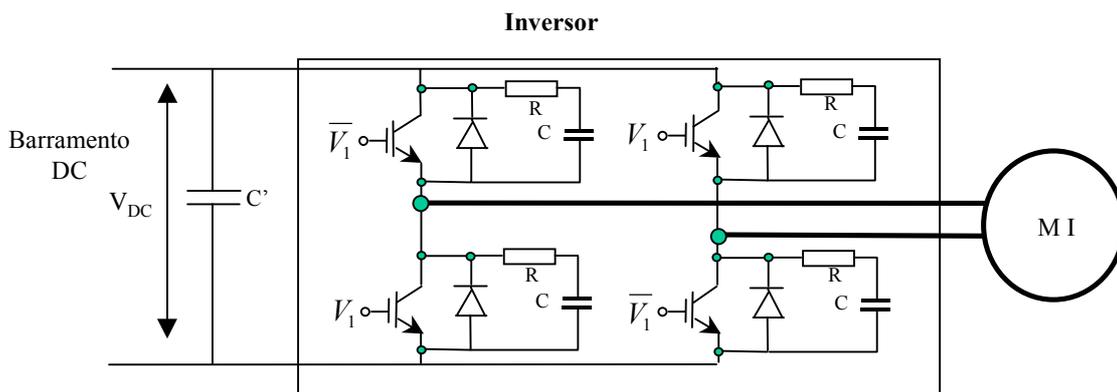
Figura 7.5 – Gerador de onda quadrada.

A saída do circuito (pino 3 do integrado NE555) consiste numa onda quadrada com frequência de 50 Hz. Com este tipo de modulação e apesar do motor funcionar, foi no entanto verificada a existência de uma elevada variação  $dv/dt$  durante os instantes de comutação (ao desligar os IGBTs). A Figura 7.6 ilustra a forma de onda da tensão correspondente a uma das fases do motor com uma tensão no barramento DC de 110 V.



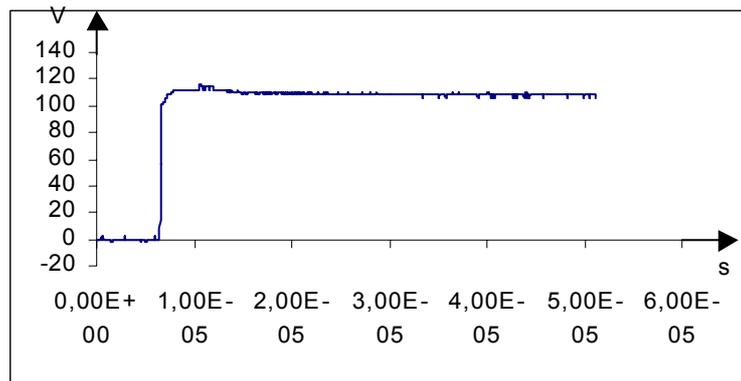
**Figura 7.6– Forma de onda lida antes da colocação dos circuitos de “amortecimento”.**

Para melhorar a comutação deficiente, anteriormente referida, foi adicionado um circuito de “amortecimento”, *snubber*, composto por uma resistência ( $R=470\ \Omega$ ) e um condensador de polipropileno ( $C=1\ \mu\text{F}$ ) colocados em paralelo com os IGBTs do módulo PM20CSJ060, conforme aconselhado em [53, 55] e ilustrado na Figura 7.7. Além do circuito anteriormente referido foi também adicionado um condensador de polipropileno ( $C'=4,4\ \mu\text{F}$ ) em paralelo com a entrada do inversor.



**Figura 7.7 – Adição de circuitos de “amortecimento”.**

A forma de onda obtida após aplicar os circuitos de “amortecimento” anteriormente referidos está ilustrada na Figura 7.8.



**Figura 7.8 - Forma de onda lida após colocação dos circuitos de “amortecimento”.**

Outro ponto importante diz respeito ao facto de durante a comutação as fontes isoladas de tensão DC (de 15 V), que alimentam os circuitos de accionamento dos IGBTs do inversor, apresentarem um ruído de alta frequência, coincidente no tempo com as alturas de comutação do sinal de comando, que era crescente com o aumento da tensão no barramento DC. Este ruído provocava a detecção por parte do módulo PM20CSJ060 e correspondente assinalar de erro ( $F_0$ ) por subtensão nos seus terminais de entrada (sempre que a tensão nas fontes for inferior a 13,5 V). Para resolver este problema foram incorporados condensadores de polipropileno em paralelo com as saídas das fontes de alimentação bem como aumentada a capacidade dos condensadores já existentes no circuito de accionamento.

### 7.2.2.3 TESTE DO MODULADOR PWM

Nesta secção é feita uma análise ao funcionamento do modulador. Todos os resultados aqui apresentados foram analisados através do DADiSP tendo como origem ondas captadas em tempo real. As formas de onda dizem respeito à tensão de alimentação do motor, medida entre as fases do mesmo. A análise consiste na observação das ondas de tensão e do respectivo espectro de frequências.

A frequência de comutação foi sempre de 2409 Hz. O índice de modulação ( $m_a$ ) foi limitado a 98%, para que a limitação da largura mínima do pulso não afectasse a distorção das ondas sintetizadas.

Um parâmetro importante, quando se fala desta técnica de modulação, por largura de impulso sinusoidal, é a razão entre as frequências da portadora ( $f_p$ ) e da referência ( $f$ ), frequentemente chamada de “razão de frequência” ( $m_f$ ) [60].

Nas Figuras 7.9 e 7.10 são ilustrados os sinais medidos com o recurso ao osciloscópio. É também representado o espectro de frequências do referido sinal, no qual a amplitude dos diferentes harmónicos é expressa em percentagem da amplitude da componente fundamental. Assim a onda gerada e sobre a qual são mostrados os resultados ilustrados na Figura 7.9 possui as seguintes características:  $f=50$  Hz,  $f_p=2409$  Hz,  $m_a=0,98$  e  $m_f=f_p/f=48,18$  e garantia, por *software*, de um pulso mínimo de 50  $\mu$ s. A onda da Figura 7.10 tem as seguintes características:  $f=25$  Hz,  $f_p=2409$  Hz,  $m_a=0,5$  e  $m_f=96,36$ .

É possível concluir que o aspecto dos sinais de tensão obtidos através de modulação por largura de impulso sinusoidal, são os esperados para este tipo de modulação, ou seja, as componentes mais significativas surgem na vizinhança da frequência da portadora e

seus múltiplos, sendo facilmente filtrados por cargas indutivas como é o caso dos motores eléctricos.

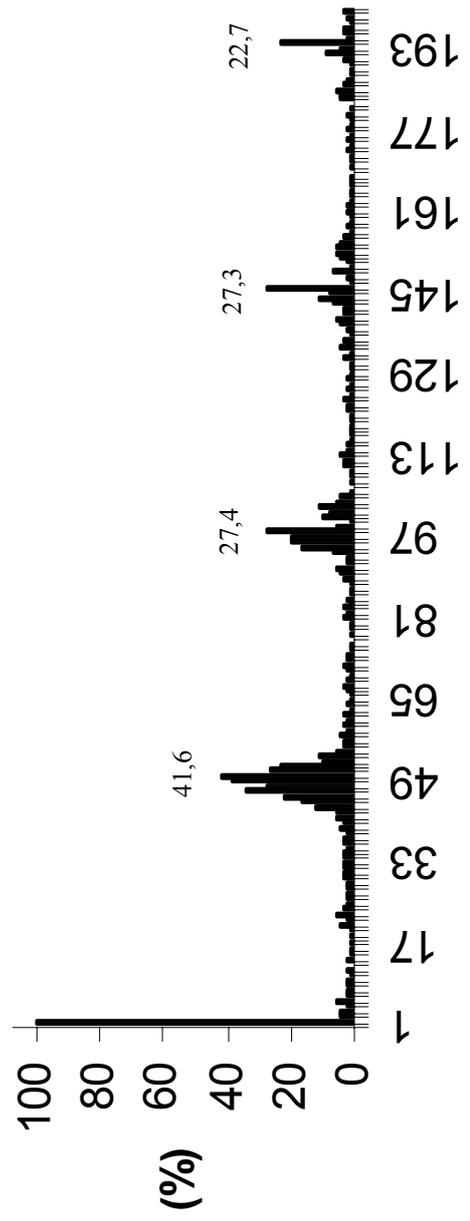
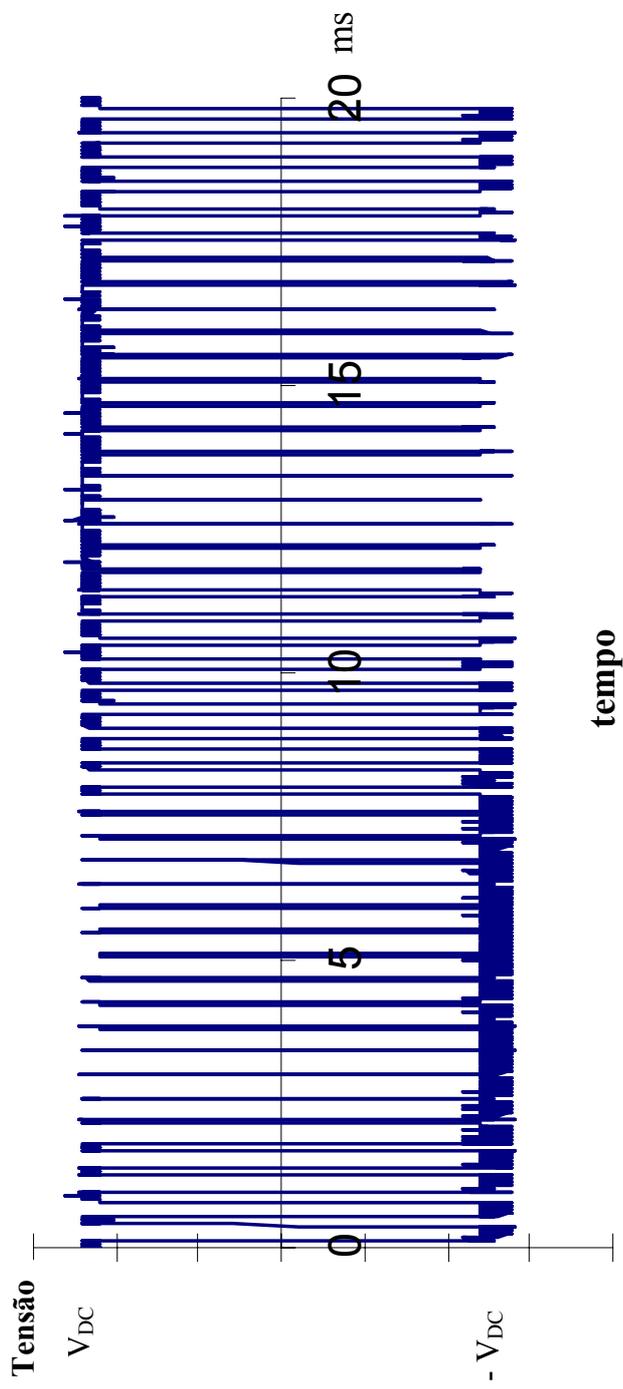


Figura 7.9 – Modulação PWM ( $f=50$  Hz): a) forma de onda da tensão; b) espectro da tensão.

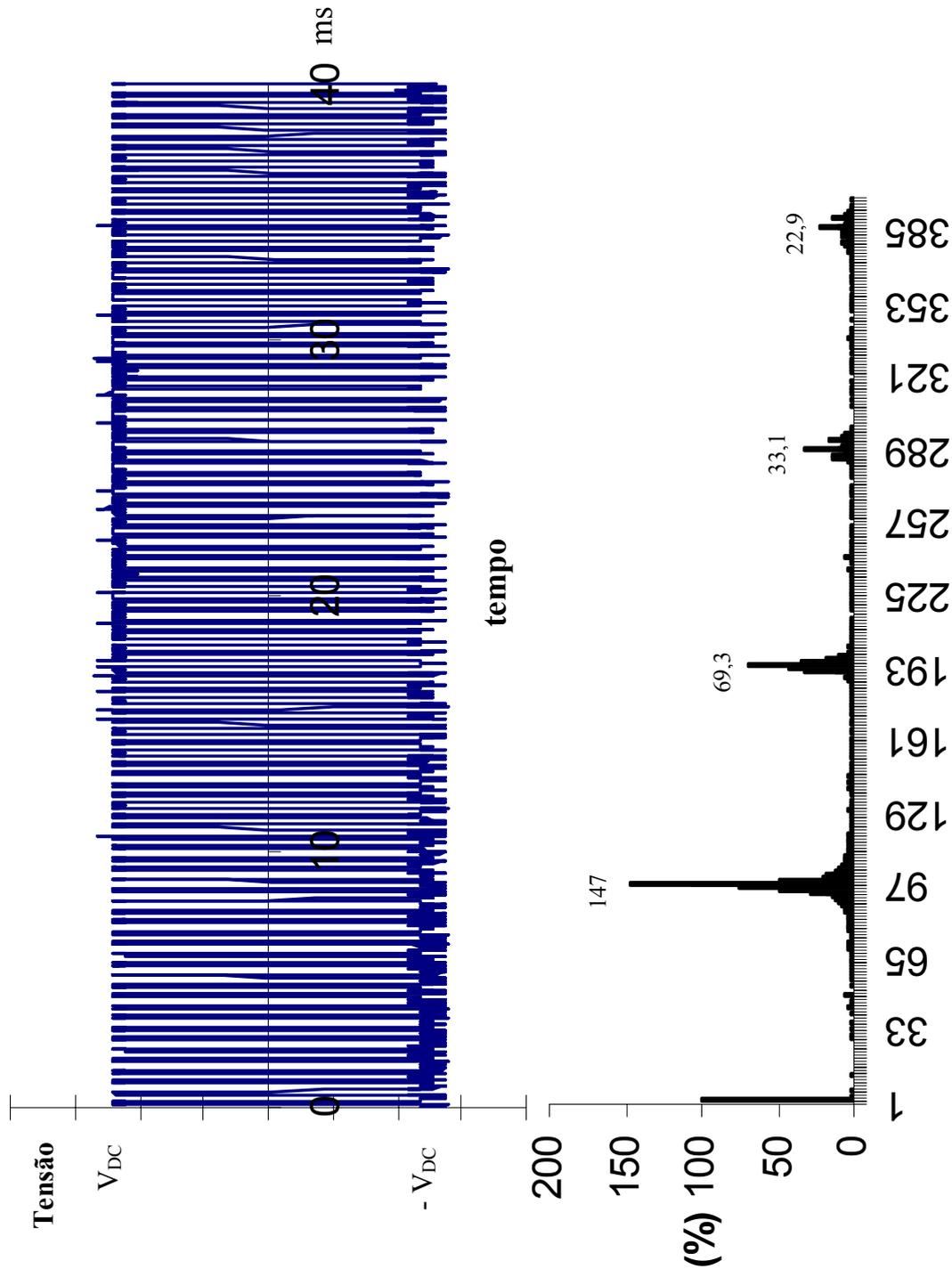


Figura 7.10– Modulação PWM ( $f=25$  Hz): a) forma de onda da tensão; b) espectro da tensão.

## 8 CONCLUSÕES E SUGESTÕES PARA TRABALHO FUTURO

### 8.1 CONCLUSÕES

Esta dissertação descreve, basicamente, o estudo e implementação de uma solução para o comando e accionamento de actuadores utilizados em estufas agrícolas.

Inicialmente foi realizada pesquisa sobre as grandezas físicas relevantes no controlo ambiental de estufas agrícolas, procurando as estratégias vulgarmente utilizadas e os actuadores que as permitem implementar.

Após conhecer as particularidades do problema que se pretende solucionar e tendo este trabalho como objectivo ser uma solução eficaz e económica para a componente de actuação de um sistema computacional já existente, foi estudada uma forma de implementar a plataforma de comunicação entre os diferentes dispositivos que se pretendem controlar, tendo a escolha sido pelo protocolo de comunicações CAN, por este permitir implementar soluções económicas, robustas e simples e além disso por o mesmo já ter sido a escolha para a componente de aquisição do sistema onde o trabalho desta dissertação se pretende enquadrar.

Devido à inércia deste tipo de sistemas, optou-se por uma solução que privilegiasse o aspecto económico e a fiabilidade em detrimento da velocidade de resposta. Assim, optou-se por aderir ao conceito de uma implementação SLIO CAN, a qual consiste na utilização de circuitos integrados de preço reduzido adequados a tarefas simples de liga/desliga, permitindo no entanto a realização de conversão A/D e a possibilidade de gerar saídas analógicas. Ao enveredar por uma solução destas, a velocidade de transmissão fica limitada a 125 Kbit/s e as distâncias a cerca de 830 metros, sendo no entanto estes valores suficientes para a aplicação em causa.

Para implementar a solução projectada foi desenvolvido *hardware* e *software* que permite, após ordem do sistema central, executar um determinado número de acções de comando e monitorização do sistema.

Após implementar a plataforma de comunicações e comando do sistema, foram também desenvolvidos e implementados circuitos de accionamento os quais servem de *interface* entre as unidades de comando e os dispositivos actuadores. Para estes circuitos de accionamento foram providenciadas soluções para dois tipos diferentes de actuação: uma consiste em circuitos para comando do tipo liga/desliga, o qual constitui a maioria dos casos em estufas agrícolas; outro tipo de circuito implementado para o controlo de velocidade de um motor de indução monofásico, utilizando modulação PWM. Todos os dispositivos foram projectados e implementados para executarem comandos transmitidos a partir de uma estação remota via CAN.

Os resultados obtidos através de ensaios realizados em laboratório, numa estufa em modelo reduzido propositadamente construída para efeito, permitem concluir que o sistema desenvolvido oferece garantias de aplicabilidade em estufas agrícolas reais, não pondo em causa as acções de controlo nas mesmas.

Quanto ao controlo de velocidade do motor os resultados alcançados foram os previstos uma vez que o motor funciona por comando electrónico (modulação PWM através do inversor de IGBTs) sendo possível a alteração da velocidade de rotação do mesmo a partir de um local remoto.

## 8.2 SUGESTÕES PARA TRABALHO FUTURO

Em trabalho futuro este sistema poderá ser ligado ao sistema desenvolvido na UTAD e instalado numa estufa real, para análise do seu comportamento em condições que não as de laboratório.

Em futuras evoluções desta aplicação o SLIO P82C150 da Phillips deverá ser substituído pelo seu equivalente baseado no microcontrolador COP884BC da National Semiconductors, garantindo assim, a continuidade do projecto, uma vez que a Philips anunciou que o SLIO P82C150 está em regime de manutenção.

Outra possível evolução deste trabalho poderá consistir na melhoria das capacidades do nó *Slave Motor* por substituição do microcontrolador P80C592 da Philips por um dos DSPs TMS320F241/243 da Texas Instruments os quais são optimizados para aplicações com controlo digital de motores.

Outra evolução de grande interesse seria a união do circuitos de comando e accionamento num único circuito, com as vantagens de espaço e modularidade que daí se poderiam retirar.

Para aplicações onde, devido à potência exigida, seja necessário um motor trifásico, seria interessante alterar o *layout* do circuito de accionamento do motor de indução no sentido de aproveitar o mesmo inversor IGBT, para efectuar o controlo de velocidade de um motor trifásico.

Seria também interessante, extrapolar esta aplicação para outros ambientes como por exemplo na área da Domótica onde a velocidade e distâncias sejam comportáveis.

## BIBLIOGRAFIA

- [1] Carlos M. J. Alves Serôdio, Carlos A. C. Couto, João L. Monteiro, "An Integrated Network for Agricultural Management Applications", 1998 IEEE.
- [2] J. Cunha, C. Couto, A. E. B. Ruano, "Real Time Parameter Estimation of Dynamic Temperature and Humidity Models for Greenhouse Adaptive Climate Control", in proceedings of the 4th IFAC Workshop on Algorithms and Architectures for Real-Time Control, Vilamoura, Abr. 1997, pp. 147-152.
- [3] F. S. Zazueta, R. Bucklin, P. H. Jones e A. G. Smajstrla, "Basic Concepts in Environmental Computer Control of Agricultural Systems", Circular 1029, Florida Cooperative Extension Services, Institute of Food and Agricultural Sciences, University of Florida, Dez. 1991.
- [4] "Road Vehicles – Interchange of Digital Information – Controller Area Network (CAN) for High Speed Communication (ISO 11898)", ISO, Nov. 1993.
- [5] Zoilo Serrano Cermeño, "Estufas Instalações e Maneio", Litexa Editora, Lda, 1990.
- [6] Cristina Maria Capucho Portelinha, "Estudo Comparativo da Cultura de duas espécies Phaseoleae em Hidroponia (N.F.T.)", Relatório do Trabalho de Fim de Curso, Escola Superior Agrária de Castelo Branco, 1993.
- [7] Home Harvest<sup>®</sup>, "Hydroponics Systems", <http://www.homeharvest.com/hydrosys.html>, Out. 1997.
- [8] Home Harvest<sup>®</sup>, "Hydroponics Accessories & Supplies", <http://www.homeharvest.com/hydrosup.html>, Out. 1997.
- [9] D. E. Buffington, R. A. Bucklin, R. W. Henley and D.B. McConnell, "Greenhouse Ventilation", Fact Sheet AE-10, Florida Cooperative Extension Service, Institute of Food and Agricultural Sciences, University of Florida, Jan. 1993.
- [10] "Equipements electriques em horticulture et maraichage", Electricité de France, Paris, 1988.
- [11] D. E Buffington, R. A. Bucklin, R. W. Henley e D. B. McConnel, "Fans For Greenhouses", Fact Sheet AE-12, Florida Cooperative Extension Service, 1992.
- [12] R. A Bucklin, R. W. Henley and D.B. McConnell, "Fan and Pad Greenhouse Evaporative Cooling Systems", Circular 1135, Florida Cooperative Extension Service, Institute of Food and Agricultural Sciences, University of Florida, Dez. 1993.
- [13] R. A Bucklin, "Florida Greenhouse Design", Bulletin 235, Florida Cooperative Extension Service, Out. 1988.

- [14] D. E. Buffington, R. A. Bucklin, R. W. Henley and D. B. McConnell, "Heating Greenhouses", Fact Sheet AE-11, Florida Cooperative Extension Service, 1992.
- [15] D. E. Buffington, R. A. Bucklin, R. W. Henley e D. B. McConnel, "Maintenance Guide for Greenhouse Ventilation, Evaporative Cooling and Heating Systems", Fact Sheet AE-26, Florida Cooperative Extension Service, 1992.
- [16] R. A. Bucklin, P.H. Jones, B. A. Barmby, D.B. McConnell e R.W.Henley, "Greenhouse Heating Checklist", Circular 791, Florida Cooperative Extension Service, Jun. 1992.
- [17] "Plants and light", Homegrown Hydroponics, <http://www.hydroponics.com/c-light.html>, Jan. 1998.
- [18] "The Importance of Carbon Dioxide for Healthy Plant Growth", Home Harvest, <http://www.homeharvest.com/envctrlco2.html>, Out. 1997.
- [19] Pierce H. Jones, "Environmental Requirement for Greenhouses", Fact Sheet RFAA95, Florida Cooperative Extension Service, Institute of Food and Agricultural Sciences, University of Florida, Out. 1994.
- [20] Lars-Berno Fredriksson, "Controller Area Networks and the protocol CAN for Machine Control Systems", KVASER AB, Jan. 1997.
- [21] S. B. Khoh & R. T. McLaughlin, "Controller Area Network (CAN) for Industrial Applications", Controller Area Network *press release*, Advanced Technology Centre, University of Warwick, 1996.
- [22] "Controller Area networking, The Future Of Industrial Microprocessor Communications?", C51/166 newsletter, Hitex UK Ltd, Jan. 1995.
- [23] R. T. McLaughlin, "EMC Susceptibility Testing of a CAN Car ", sae paper 932866.
- [24] "Road vehicles-Serial data communication for automotive application. Part 1: Controller Area Network (CAN).", ISO/DIS -11519 part 1, International Organisation for Standardisation, 1994.
- [25] "Cia Draft Standard 102 Version 2.0", CAN Physical Layer for Industrial Applications, Two-Wire Differential Transmission, Abr. 1994.
- [26] "CAN controller interface-PCA82C250", Philips Semiconductors objective specification, Abr. 1994.
- [27] "CAN specification, Version 2.0", Philips Semiconductors, 2000 Hamburg 1, 1991.
- [28] "Controller Area Network, CAN, a Serial Bus System - Not Just For Vehicles", CAN in Automation(CiA) e.V.

- [29] José Rufino, “An Overview of the Controller Area Network”, Proceedings of the Cia Forum – CAN for Newcomers, Braga, Portugal, Jan. 1997.
- [30] Hans-Christian Reuss, “Extended *Frame* Format – A New Option of CAN Protocol”, Product Concept & Application Laboratory Hamburg, F. R. Germany, HAI/AN 92002, Philips Semiconductors, 1993.
- [31] R. T. McLaughlin, S. B. Khoh, “A SLIO (Serial Link Input/Output) CAN implementation”, University of Warwick .
- [32] Dr. K. W. Young, R. T. McLaughlin & S. B. Khoh, “Low Cost SLIO CAN Based Body Control System”, University of Warwick.
- [33] “P82C150 CAN Serial Linked I/O device (SLIO) with digital and analog port functions”, Preliminary specification, Philips Semiconductors, 1996.
- [34] Carlos Serôdio, J. Boaventura Cunha, Raul Morais, Carlos Couto, João Monteiro,” A Network for Agricultural Management Systems: The Communications and Control Platforms”, 1st IFAC Workshop on Control Applications and Ergonomics in Agriculture, Athens, Greece, Jun. 1998, pp. 333-338.
- [35] Carlos Serôdio, J. Boaventura Cunha, Manuel Cordeiro, António Valente, Raul Morais, Paulo Salgado, Carlos Couto, “Mnet-DACS: Multi-level Network Data Acquisition and Control Systems, ISIE97’-Proc. IEEE International Symposium on Industrial Electronics, vol. Nº 1, pp. 39-43, Guimarães, Jun. 97, ISBN 0-7803-3936-3.
- [36] Kienckel, U., Cao C. T., e Litschel M., “The Impact of Bit Representation on Noise Emissions in Automotive Networks”, in Proc. IEEE 6th Int. Conf. On Automotive Electronics, pp 200-203, 1987.
- [37] “P8xC592 8-bit microcontroller with on-chip CAN”, Product specification, Philips Semiconductors, Jun. 1996.
- [38] Carlos Manuel José Alves Serôdio,”Arquitetura CAN, Aplicada ao Controlo Ambiental de Estufas Agrícolas”, Provas de Aptidão Pedagógica e Capacidade Científica, Universidade de Trás-os-Montes e Alto Douro, Vila Real, Mar. 1996.
- [39] “DS12887 Real Time Clock”, Dallas Semiconductors, Fev. 1998.
- [40] “LM78xx Series Voltage Regulators”, National Semiconductor, Fev. 1995.
- [41] “DS1231 Power Monitor Chip”, Dallas Semiconductors, Fev. 1998.
- [42] “MAX220 - MAX249, +5V-Powered, Multi-channel RS-232 Drivers/Receivers” , MAXIM Integrated Products, Nov. 1997.
- [43] “NM27C512 512 Kbits (64 Kb x 8) UV EPROM and OTP EPROM”, SGS-Thomson, Nov. 1998.

- [44] “MT5C2568 883C 32K x 8 SRAM”, Austin Semiconductor, Inc., Nov. 1997.
- [45] “Dual 2-Line to 4-Line Decoders/Demultiplexers SN74HC139”, AHC/AHCT Logic Advanced High-speed CMOS, Logic Products Data Book, Texas Instruments, 1997.
- [46] “Octal Transparent D-Type Latches With 3-State Outputs SN74AHC573”, AHC/AHCT Logic Advanced High-speed CMOS, Logic Products Data Book, Texas Instruments, 1997.
- [47] “DS1210 Nonvolatile Controller Chip”, Dallas Semiconductor.
- [48] “LM35/LM35A/LM35C/LM35CA/LM35D Precision Centigrade Temperature Sensors”, National Semiconductors, Set. 1997.
- [49] “DC Solenoid application”, Western Reserv Catalog (WRC) 1997, <http://www.wrcakron.com/>, 1997.
- [50] “PowerMOS transistor TOPFET, BUK101-50GS”, product specification, Philips Semiconductors, Jan. 1993.
- [51] “DSP Solution for AC Induction Motor”, Application Report BPRA043, Texas Instruments, 1996.
- [52] Júlio Manuel de Sousa Barreiros Martins, “Controlo de velocidade do motor de indução de rotor em gaiola de esquilo”, Dissertação submetida na Universidade do Minho para obtenção do grau de Doutor em Engenharia de Sistemas e Informática, Universidade do Minho, 1993.
- [53] Ned Mohan, Tore M. Undeland, William P. Robbins, “Power Electronics Converters, Applications and Design”, 2ª Ed., John Wiley & Sons, Inc., 1995.
- [54] Heinz Ruedi , “Intelligent interfaces between power and control, Gate drives for IGBTs”, Applications Power Semiconductors, Siemens, Components XXXI, No.3, 1996.
- [55] “Mitsubishi Intelligent Power Modules - PM20CSJ060”, Power Module MOS Databook, Mitsubishi Semiconductors, 1995, pag. 3-77/82.
- [56] “Motion Control & Power Conditioning”, Product Handbook, Mitel Semiconductor, Mai. 1997.
- [57] “SN74LS75 4-bit bistable latches”, Production data documents, Texas Instruments, Mar. 1988.
- [58] “74HCT14 Hex inverting Schmitt trigger”, Product specification, IC06, Philips Semiconductors, Set. 1993.
- [59] HCPL4503, “High CMR, High Speed Optocouplers”, Technical Data, Hewlett Packard.

- [60] MINAS, Graça Maria Henriques, “Técnicas de modulação de largura de impulso para accionamento de máquinas eléctricas de corrente alternada”, Braga: Universidade do Minho, 1998. Dissertação de Mestrado.
- [61] S. R. Bowes, R. R. Clements, “Computer-aided design of PWM inverter systems”, IEE Proc., Vol. 129, Pt. B, No. 1, Jan. 1982, p. 1-17.
- [62] Sidney Robert Bowes, “Novel Real-Time Harmonic Minimized PWM Control for Drives and Static Power Converters”, IEEE transactions on Power Electronics, Vol. 9, No. 3, Mai. 1994, p. 256-262
- [63] “P80CL31; P80CL51 Low voltage 8-bit microcontrollers with UART”, Product specification, Philips Semiconductors, Jan. 1995.

**ANEXO 1 – PLACAS DE CIRCUITO IMPRESSO**

- **PLACA DE CIRCUITO IMPRESSO DO NÓ *HOST* OU DO NÓ *SLAVE MOTOR***

**Figura A1.1 - Aspecto do *layout* - componentes**

**Figura A1.2 - Aspecto do circuito impresso (lado superior)**

**Figura A1.3 - Aspecto do circuito impresso (lado inferior)**

- **PLACA DE CIRCUITO IMPRESSO DO NÓ SLIO CAN**

**Figura A1.4 - *Layout* dos componentes**

**Figura A1.5 - Aspecto do circuito impresso**

- **PLACA DE CIRCUITO IMPRESSO DA INTERFACE RS-232**

**Figura A1.6 - *Layout* dos componentes**

**Figura A1.7 - Aspecto do circuito impresso**

- **PLACAS DE CIRCUITO IMPRESSO DOS CIRCUITOS UTILIZADOS PARA ACCIONAMENTO DO MOTOR DE INDUÇÃO MONOFÁSICO**

**Figura A1.8 - Aspecto do *layout* do circuito de accionamento - componentes**

**Figura A1.9 - Aspecto do circuito impresso (lado superior)**

**Figura A1.10 - Aspecto do circuito impresso (lado inferior)**

**Figura A1.11 - Aspecto do *layout* do bloco rectificador – componentes.**

**Figura A1.12 - Aspecto do circuito impresso do bloco rectificador.**

- **PLACA DE CIRCUITO IMPRESSO DO CIRCUITO UTILIZADO PARA ACCIONAMENTO UTILIZANDO RELÉS**

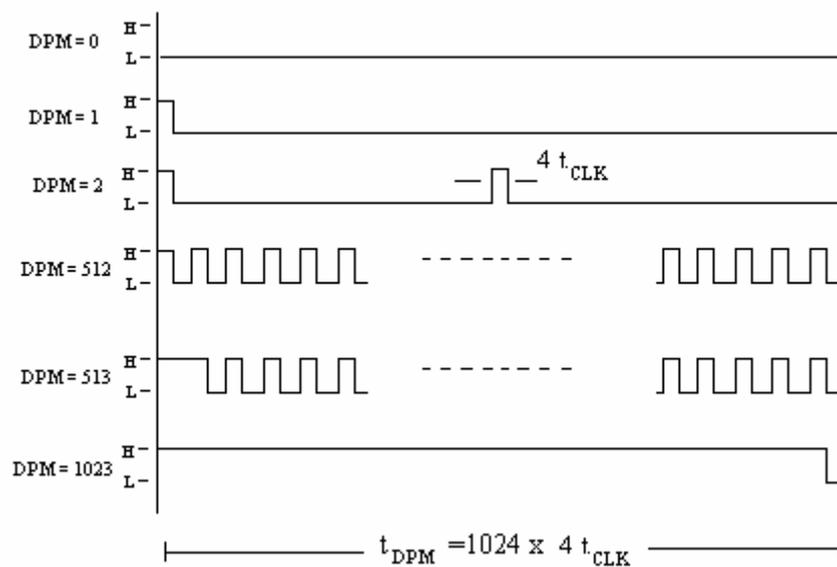
**Figura A1.13 - Aspecto do *layout* - componentes**

**Figura A1.14 - Aspecto do circuito impresso**

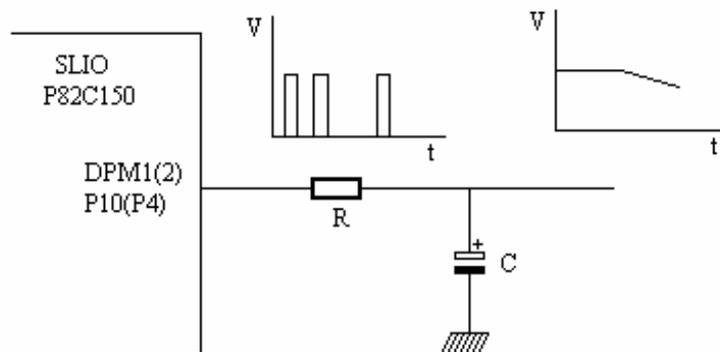
## **ANEXO 2 - UTILIZAÇÃO DAS SAÍDAS ANALÓGICAS**

## Registos DMP1 e DPM2

Os registos, *Distribute Pulse Modulation*, DPM1 e DPM2 são programados com os valores necessários para gerar um sinal de saída quase analógico nos pinos 10 e 4 do SLIO, respectivamente, os quais são gerados por DPM (Figura A2.1). Os pinos 10 e 4 do SLIO P82C150 devem ser previamente activados por escrita no registo, *Output Enable Register* (OER), para que este possa gerar as saídas cujo número de pulsos durante um período DPM corresponde ao valor dos registos DPM1 ou DPM2. Estes pulsos tem a duração de  $4t_{CLK}$  ( $t_{CLK}$  é o período do relógio) e são distribuídos pelo período DPM. É possível obter o valor analógico após submeter a saída a um filtro RC conforme ilustra a Figura A2.2.



**Figura A2.1 - Pulsos na saída DPM (P10 ou P4 ) do SLIO.**



**Figura A2.2 - Exemplo de uma aplicação utilizando as saídas DPM.**

### **ANEXO 3 - MAPEAMENTO DE MEMÓRIA**

O mapeamento de memória do nó *Host* foi escolhido por forma a reservar espaço de memória para todas as variáveis e assim evitar conflitos devidos a sobreposição de posições de memória. O esquema da Figura A3.1 representa a forma como foi elaborado o mapeamento.

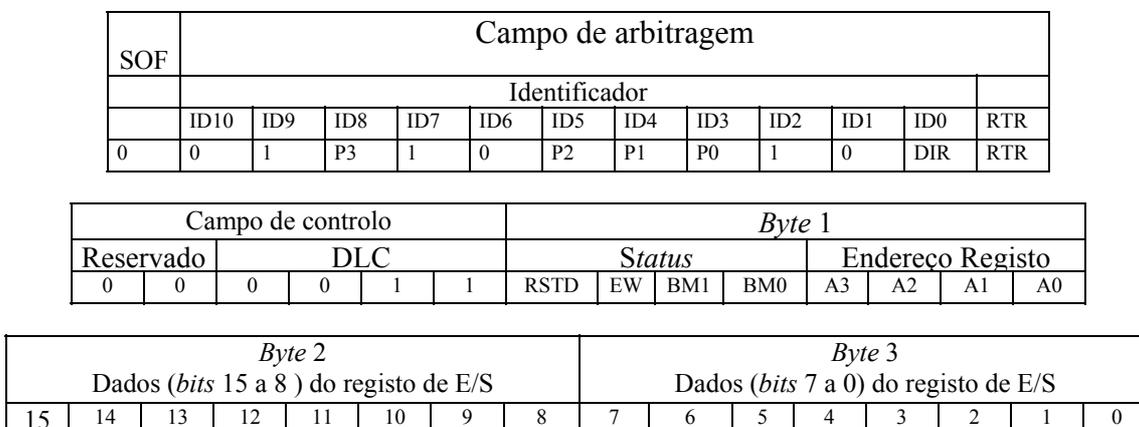
Endereço	Conteúdo
0x010907 0x010906	Área de dados
0x010901 0x010815 0x010801	RTC
	CAN
	RESERVADO
0x01016E 0x01015A 0x010159	Funções
0x010109 0x010108	Configuração
0x010000	Ponteiro de dados, <i>Flags</i>

**Figura A3.1 – Mapeamento da memória de dados do *Host*.**

**ANEXO 4 - TIPOS DE *FRAME* E REGISTOS DO SLIO P82C150**

## Frame de dados num sistema SLIO CAN

As *frames* de dados do SLIO P82C150 possuem o formato representado na Figura A4.1.



**Figura A4.1 – Formato das *frames* do SLIO P82C150.**

No campo identificador existem apenas 4 *bits* configuráveis P3 a P0, os quais caracterizam o valor único para determinado SLIO [33], sendo o valor destes *bits* igual ao nível lógico dos sinais colocados nos respectivos pinos durante a operação de *reset* por *hardware*. O campo identificador possui ainda o *bit* DIR que deve ser colocado ao nível ‘1’ para *frames* de dados destinadas ao nó *Host* e em pedidos de *frames* remotas enviados pelo nó *Host* para os nós SLIO e colocado ao nível lógico 0 para *frames* de dados enviadas do nó *Host* para os nós SLIO CAN.

Na Figura A4.1 é possível verificar que o comprimento de dados é fixo e igual a 3 *bytes*, os quais contêm no primeiro *byte* o *status* da informação e endereço do registo pretendido e nos segundo e terceiro *bytes* de dados o valor desejado para o registo endereçado. Isto é tudo o que a mensagem destinada a um nó SLIO CAN deve possuir para definir o estado para as 16 saídas possíveis dos SLIO. Os quatro *bits* mais significativos do primeiro *byte* de dados são apenas de leitura sendo os significados destes *bits* de *status* os seguintes:

- RSTD - Ao nível lógico ‘1’ na primeira mensagem após calibração conseguida com sucesso, ou seja, após a sincronização do tempo de *bit*.
- EW – Tal como o anterior tem nível lógico ‘1’ durante a primeira mensagem de calibração e quando ocorre um erro.
- BM1 e BM0 - Indicam em que modo se encontra o barramento da forma que é ilustrada pela Tabela A4.1.

<i>Modo do barramento</i>	<i>BM1</i>	<i>BM0</i>
Diferencial	0	0
Um fio – RX1	0	1
Um fio – RX0	1	0
Sleep	1	1

**Tabela A4.1 – Modos de barramento.**

## Registos do SLIO P82C150

Quanto aos 4 *bits* menos significativos do primeiro *byte* de dados, estes representam o endereço do registo ao qual se destinam os 2º e 3º *bytes* de dados. Existem 8 registos possíveis com os endereços e designações referidos na Tabela A4.2.

Registo	Endereço	Observações
<i>Data Input Register</i>	00	Registo que contém o estado dos 16 pinos do SLIO.
<i>Positive Edge Register</i>	01	Registo só de escrita que permite colocar determinado pino no modo <i>event capture</i> (pulso crescente).
<i>Negative Edge Register</i>	02	Registo só de escrita que permite colocar determinado pino no modo <i>event capture</i> (pulso decrescente).
<i>Data Output Register</i>	03	Registo só de escrita que possui os dados a colocar nos pinos de saída. Tem de ter o correspondente <i>Output enable</i> a '1' para ser levado em consideração.
<i>Output Enable Register</i>	04	Este registo de saída controla os <i>drivers</i> de saída dos vários pinos de saída.
<i>Analog Configuration Register</i>	05	Registo de leitura e escrita que contém a informação necessária para iniciar, e ler o resultado de uma conversão A/D.
<i>DPM1 Register</i>	06	Este registo só de escrita contém os dados para a saída quase analógica (pino 10). Tem de ter o correspondente <i>Output Enable</i> a '1' para ser levado em consideração.
<i>DPM2 Register</i>	07	Igual ao anterior para o pino 4.
<i>Analog-to-digital Conversion (ADC) Register</i>	08	Registo só de leitura contém o resultado da conversão A/D feita conforme configurado pelo registo ACR (endereço 5).

**Tabela A4.2 – Endereço dos registos do SLIO P82C150.**

Quanto aos segundo e terceiros *bytes* de dados estes contêm o estado dos 16 pinos E/S do SLIO correspondente ao identificador e registo corrente.

## Diferença entre mensagens de dados e repostas num sistema SLIO CAN

As diferenças entre os tipos de *frames* existentes entre o nó *host* e os nós SLIO dizem respeito ao valor dos *bits* RTR e DIR e são resumidas na Tabela A4.3.

<i>Frame</i>	Transmissão pelo SLIO	Recepção pelo SLIO
Dados	Sim (DLC=3;DIR=1)	Sim (DLC=3;DIR=0; no caso da mensagem de sincronização) 2≤DLC≤8
Remota	Não	Sim (DLC=3;DIR=1)
Erro	Sim	Sim
<i>Overload</i>	Sim (só como resposta)	Sim

**Tabela A4.3 – Comunicação *Host*-SLIO.**

## **ANEXO 5 - AQUISIÇÃO ANALÓGICO-DIGITAL UTILIZANDO OS SLIOS**

## Registo *Analog Configuration Register* (ACR)

Este registo de leitura/escrita serve para definir a conversão A/D e é composto por 16 *bits* com significados diferentes conforme mostra a Figura A5.1

ACR (MSB)							
ADC	OC3	OC2	OC1	0	M3	M2	M1

ACR (LSB)							
SW3	SW2	SW1	0	0	0	0	0

**Figura A5.1 – Registo ACR.**

O significado dos *bits* que compõem o registo ACR é o seguinte:

- **O *bit* ADC** - Este *bit* é apenas de escrita. Quando ADC=1 o SLIO [33] inicia um ciclo de conversão A/D que termina com o envio de uma mensagem com o resultado da conversão por parte do SLIO. No fim da conversão este *bit* é zerado automaticamente.
- **OC3-OC1** - Estes *bits*, apenas de leitura, representam o valor lógico das saídas de três comparadores analógicos cujas saídas se podem verificar nos pinos P7, P8 e P9 e cujas entradas correspondem aos pinos P10, P11, P12, P13 e P15 do SLIO [33].
- **M3 a M1** - Estes *bits*, apenas de escrita, dizem respeito à capacidade de activar ou não as saídas dos comparadores anteriormente referidos.
- **SW3 a SW1** - Estes *bits*, apenas de escrita, permitem seleccionar um de seis canais possíveis para aquisição.

## Registo *Analog-to-Digital Conversion* (ADC)

Este registo de leitura contém o resultado da aquisição A/D, de 10 *bits*, dos sinais analógicos aplicados no pino I/O seleccionado pelos *bits* SWx do registo ACR. A conversão é iniciada quando o *bit* ADC=1 do ACR ou pela transmissão duma *frame* de dados endereçando o registo ADC do SLIO.

O formato e conteúdo do registo ADC é representado pela Figura A5.2.

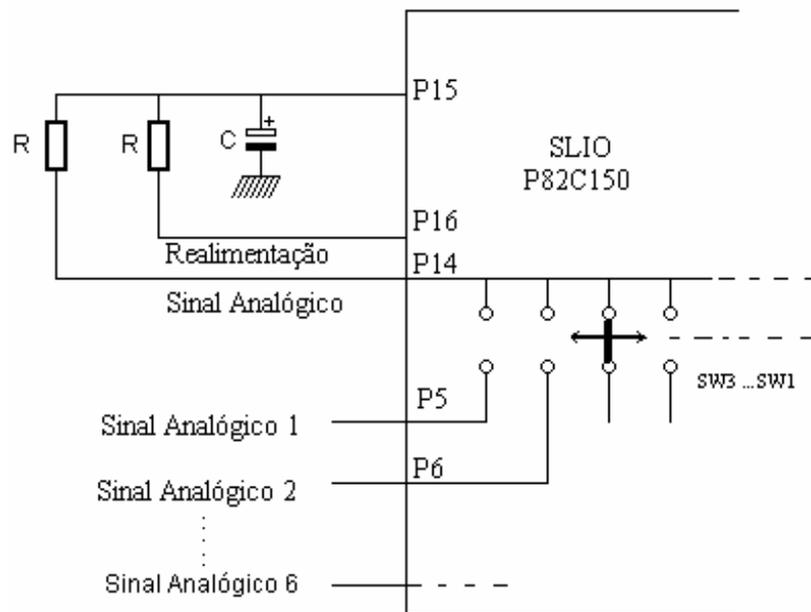
ADC (MSB)							
AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

ADC (LSB)							
AD1	AD0	0	0	0	0	0	0

**Figura A5.2 – Registo ADC.**

A Figura A5.3 mostra um exemplo de aquisição multiplexada de seis sinais analógicos, os quais são seleccionados pelos *bits* SWx do registo ACR.



**Figura A5.3 – Aquisição usando 6 canais multiplexados.**

## **ANEXO 6 – INICIAÇÃO DO MICROCONTROLADOR E CONTROLADOR CAN**

## Procedimento de Iniciação

Durante o procedimento de iniciação são configurados o microcontrolador, controlador CAN, RTC e a UART interna.

## Iniciação do microcontrolador

Para iniciar o microcontrolador é necessário definir e identificar as fontes de interrupção pretendidas, para que sejam desactivadas as restantes.

Para habilitar as interrupções pretendidas é necessário actuar sobre o registo de interrupção IEN0 (endereço A8H) [37], activando cada fonte de interrupção individualmente, o que é conseguido colocando a '1' o *bit* que lhe corresponde nesse registo.

Os conteúdos do registo IEN0 está representado na Figura A6.1.

IEN0 7	IEN0 6	IEN0 5	IEN0 4	IEN0 3	IEN0 2	IEN0 1	IEN0 0
EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0
1	0	1	0	0	1	0	1

**Figura A6.1 - Conteúdo do registo IEN0.**

De acordo com as fontes de interrupção necessárias foi atribuído o valor IEN0=A5H o que corresponde a habilitar os *bits* cujo significado é o seguinte:

- EA: habilitar interrupções individuais
- ES1: Habilita interrupções SIO1 (CAN)
- EX1: Habilita interrupções externa 1 (RTC)
- EX0: Habilita interrupções externa 2 (NMI)

Quanto à prioridade das interrupções, foram mantidas as prioridades que existem por defeito colocando a zero os *bits* que compõem os registos IP0 e IP1 ficando, desta forma, estabelecida a seguinte ordem de prioridades:

- EX0: Habilita interrupções externa 2 (NMI) (maior prioridade)
- ES1: Habilita interrupções SIO1 (CAN)
- EX1: Habilita interrupções externa 1 (RTC)
- ES0: Habilita interrupções SIO0 (UART) (menor prioridade)

## Iniciação do controlador CAN

### Registos do CAN

O controlador CAN interno do microcontrolador P80C592 é visto pelo CPU como um dispositivo mapeado em memória o qual é constituído por um segmento de Controlo, um *buffer* de transmissão e dois *buffers* de recepção conforme ilustra a Figura A6.2.

Endereço (dec/hex)	REGISTOS CAN	MODOS DE ACESSO DO CPU	
	<b>SEGMENTO DE CONTROLO</b>		
0	Controlo (CR)	Escrita	
1	Comando (CMR)	←	CANCON Acesso
2	Status (SR)	→	CANSTA Directo
3	Interrupções (IR)	→	CANCON Com SFR
4	Código de aceitação(ACR)	Leitura	
5	Máscara de aceitação(AMR)		
6	Bus Timing 0(BTR0)		
7	Bus Timing 1(BTR1)		
8	Controlo de Saída (OCR)		
9	Teste: não utilizado		
			Acesso usando os registos CANADR/CANDAT
			Leitura/escrita ↔
	<b>BUFFER DE TRANSMISSÃO</b>		
10 A	Identificador (10..3)		
11 B	Identificador (2..0), RTR, DLC		
12 C	Byte de dados #1	Escrita ←	Acesso Com DMA
19 13	Byte de dados #8		
	<b>BUFFER DE RECEPÇÃO 0/1</b>		
20 14	Identificador (10..3)		
21 15	Identificador (2..0), RTR, DLC		
22 16	Byte de dados #1		
		Leitura →	
29 1D	Byte de dados #8		

**Figura A6.2 – Modos de acesso do CPU ao controlador CAN do microcontrolador P80C592.**

A troca de sinais de comando, controlo e de *status* entre o CPU e o controlador CAN é realizada através do Segmento de Controlo, o qual é composto por 10 *bytes*. Este Segmento é configurado durante a iniciação para definição dos parâmetros de comunicação. Além disso, este Segmento é utilizado também, pelo CPU, para controlar as comunicações CAN.

Os dados a serem transmitidos são transferidos, pelo CPU, para o *buffer* de transmissão. Este *buffer* armazena o *descriptor* (Identificador, *bit* RTR e DLC) e até 8 *bytes* de dados.

Após recepção com sucesso, as mensagens são lidas do *buffer* de recepção. Este *buffer* consiste em duas memórias de 10 *bytes* de capacidade que são utilizados alternadamente para armazenar mensagens. Desta forma o CPU pode processar uma mensagem enquanto outra está a ser recebida

Para aceder aos registos que constituem o Segmento de Controlo, referido anteriormente, o microcontrolador possui quatro registos de funções especiais (SFR) designados por CANADR, CANDAT, CANCON e CANSTA. Todos os registos CAN do Segmento de Controlo que são utilizados durante a comunicação aparecem, como sendo directamente endereçáveis uma vez que são representados por estes quatro registo de funções especiais.

Os significados dos registos SFR que o microcontrolador possui para configurar o controlador CAN interno são:

### CANADR

Os 5 *bits* menos significativos (CANA4...CANA0) deste registo, permitem endereçar qualquer um dos registos internos do controlador CAN através de um acesso para escrita ao registo CANADR. Ler ou alterar o registo CAN particular é feito através de leitura ou escrita no registo CANDAT. O CANADR é implementado como registo de leitura/escrita o qual também possui *bits* de controlo para seleccionar endereçamento de auto incremento (AUTOINC) e iniciar transmissão *Direct Memory Access* (DMA). O conteúdo deste registo é representado na Figura A6.3.

DMA	RESERV.	AUTOINC	CANA4	CANA3	CANA2	CANA1	CANA0
-----	---------	---------	-------	-------	-------	-------	-------

**Figura A6.3 – Formato do registo CANADR.**

### CANDAT

Ler ou escrever neste registo de 8 *bits* significa acesso ao registo interno, do controlador CAN, cujo endereço está em CANADR. O formato deste registo é representado na Figura A6.4.

CAND7	CAND6	CAND5	CAND4	CAND3	CAND2	CAND1	CAND0
-------	-------	-------	-------	-------	-------	-------	-------

**Figura A6.4 – Formato do registo CANDAT.**

### CANCON

É um registo de leitura e escrita embora contendo significados distintos. Ler é o mesmo que ler o registo interno de interrupção do controlador CAN. Escrever é o mesmo que escrever no registo interno de comando do controlador CAN. O conteúdo do registo para as das duas acções está representado na Figura A6.5.

LEITURA							
RESERV	RESERV	RESERV	WAKE UP	OVERR	ERROR	TRAN	RECEI
ESCRITA							
RX0 act	RX1 act	WU mode	SLEEP	CLEAR OVERR	RELEASE RXB	ABORT TRANSM	TRANSM. REQUEST

**Figura A6.5 – Formato do registo CANCON.**

## CANSTA

É um registo de leitura e escrita. A leitura é equivalente a um acesso ao registo interno de estado (*status*) do controlador CAN. Durante a escrita selecciona a posição da RAM interna para transferência DMA. O conteúdo deste SFR é representado na Figura A6.6.

LEITURA							
BUS STATUS	ERROR STATUS	TRANS STATUS	RECEI STATUS	TXCOMP1 STATUS	TxBUF ACESS	DATA OVERR	RxBUF STATUS
ESCRITA							
RAM7	RAM6	RAM5	RAM4	RAM3	RAM2	RAM1	RAM0

**Figura A6.6 – Formato do registo CANSTA.**

Após abordar os registos de funções especiais, será apresentado o procedimento de iniciação do controlador CAN. Este procedimento consiste na configuração dos seguintes registos internos do controlador CAN:

- Registo Acceptance Code Register (ACR);
- Registo *Acceptance Mask Register* (AMR);
- Registos *Bus Timig Register 0* (BTR0) e *Bus Timig Register 1* (BTR1);
- Registo de controlo de saída *Output Control Register* (OCR).

O procedimento levado a cabo para a configuração do controlador CAN é o seguinte:

1. Escrever no registo de Controlo a palavra 03H, para que o *bit Reset Request* (RR) fique activo, e desta forma se torne possível aceder aos registos internos ACR, AMR, BTR0, BTR1 e OCR;
2. Escrever nos registo ACR e AMR palavras de 8 *bits* de acordo com o filtro de aceitação pretendido;
3. Escrever nos registos BTR0 e BTR1 os valores para definir o relógio do sistema, os segmentos constituintes do período de *bit*, e conseqüentemente a taxa de transmissão.

De seguida apresentam-se os cálculos realizados para a presente aplicação.

## Estabelecimento do tempo de *bit*, tempo de propagação e taxa de transferência

A configuração do tempo de *bit*, taxa de transmissão e tempo de propagação, é efectuada através de escrita nos registos, *Bus Timing Register*, BTR0 e BTR1, cujos *bits* possuem significados diferentes, conforme a seguir se refere:

### BTR0

BTR0							
SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0

**Figura A6.7 - Registo BTR0.**

SJW.x : Por forma a compensar deslocamentos de fase entre osciladores de relógio de diferentes controladores do barramento, qualquer destes controladores deve ressincronizar a partir da transição de um sinal relevante na transmissão corrente. Assim os *bits* SJW.x definem o número máximo de ciclos de relógio ( $t_{SCL}$ ) que o período de um *bit* pode ser alongado ou diminuído por uma ressincronização.

$$t_{SJW} = t_{SCL} ( 2SJW.1 + SJW.0 + 1 )$$

BRP.x : O período do relógio do sistema  $t_{SCL}$  é programável e define o tempo de *bit* individual. O relógio do sistema é calculado a partir da seguinte equação:

$$t_{SCL} = 2t_{CLK}(32BRP.5 + 16 BRP.4 + 8BRP.3 + 4 BRP.2 + 2BRP.1 + BRP.0 + 1 )$$

em que :  $t_{CLK}$  é o período utilizado pelo oscilador do P80C592.

### BTR1

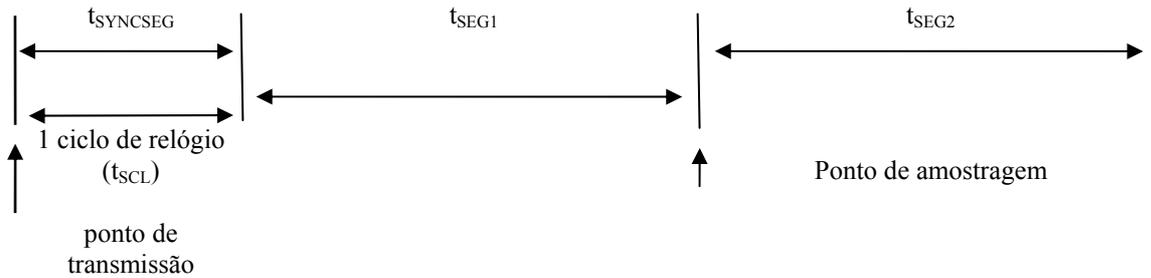
O conteúdo do registo BTR1 define o comprimento do período do *bit*, a localização do ponto de amostragem e o número de amostras que devem ser lidas em cada ponto de amostragem.

BTR1							
SAM	TSEG2.2	TSEG2.1	TSEG2.0	TSEG1.3	TSEG1.2	TSEG1.1	TSEG1.0

**Figura A6.8 - Registo BTR1.**

Se o nível lógico do *bit* mais significativo deste registo (SAM) for:

- ‘0’ (1 amostra), o barramento é amostrado uma vez. Isto é recomendado para barramentos de elevada velocidade;
- ‘1’ (3 amostras), são lidas três amostras. Isto é recomendado para barramentos de baixa/média velocidade.



**Figura A6.9 – Constituição do tempo de *bit*.**

A forma como é implementada a segmentação do tempo de *bit*,  $t_{bit}$ , no controlador CAN é ilustrada na Figura A6.9 e está de acordo com a seguinte expressão:

$$t_{bit} = t_{SYNCSEG} + t_{SEG1} + t_{SEG2}$$

Os segmentos  $t_{SEG1}$  e  $t_{SEG2}$ , determinam o número de ciclos de relógio, por período de *bit* e a localização do ponto de amostragem. Estes segmentos são configuráveis através de escrita no registo BTR1, de acordo com as seguintes expressões:

$$t_{SEG1} = t_{SCL}(8TSEG1.3 + 4TSEG1.2 + 2TSEG1.1 + TSEG1.0 + 1)$$

$$t_{SEG2} = t_{SCL}(4TSEG2.2 + 2TSEG2.1 + TSEG2.0 + 1)$$

Existem no entanto algumas condições relacionadas com o tempo de *bit*, as quais se devem a processamento interno e que devem ser também respeitadas [37]. Essas condições são:

- $t_{SEG1} \geq 2t_{SCL}$
- $t_{SEG2} \geq t_{SJW}$
- $t_{SEG1} \geq t_{SEG2}$
- $t_{SEG1} \geq t_{SJW} + t_{prop}$

Para determinar o valor dos registos cumprindo as desigualdades anteriormente referidas, foi necessário efectuar alguns cálculos.

Assim utilizando um cristal de 11,0592 Mhz, pretendendo um *baud rate* próximo de 125 Kbit/s e sabendo que nos SLIOS [33] um tempo de *bit* corresponde a  $10 \cdot t_{SCL}$ , vem que:

$$t_{bit} = 1/(125 \text{ Kbit/s}) = 8\mu\text{s}$$

$$\text{como } t_{bit} = 10 t_{SCL} \Rightarrow t_{SCL} = 8 \mu\text{s}/10 = 0,8 \mu\text{s}$$

sabendo (BTR0) que  $t_{SCL} = 2 t_{CLK} \cdot (x) = 0,8 \mu\text{s}$  e que  $t_{CLK} = 1/(11,059 \text{ MHz})$  obtém-se  $x = 4,4236$ , escolhendo o número inteiro imediatamente superior ou seja  $x=5$ .

Para os registos BTR0 e BTR1 foram escolhidos os valores:

$$\text{BTR0} = \text{C4H}$$

$$\text{BTR1} = \text{34H}$$

Correspondendo a :

$$\begin{aligned}
 t_{SEG1} &= 5 t_{SCL} \\
 t_{SEG2} &= 4 t_{SCL} \\
 t_{SJW} &= 4 t_{SCL} \\
 t_{SCL} &= 2 t_{CLK} (5) = 10 t_{CLK}
 \end{aligned}$$

Com estes valores de BTR0 e BTR1, são respeitadas as condições requeridas ao bom dimensionamento destes registos e garantido um tempo de *bit* de  $t_{BIT} = 10t_{SCL} = 10(10 t_{CLK}) = 9 \mu s$ , o que está dentro dos limites previstos [33], bem como um tempo de propagação máximo de  $t_{prop(Máx)} = 0,9 \mu s$ . Com a escolha de  $x=5$  é garantida uma taxa de transmissão de 110,590 Kbit/s.

4. Para iniciar o controlador CAN falta ainda falar de uma acção importante a qual consiste na definição da configuração dos transístores de saída. Esta escolha é feita através escrita no registo de controlo de saída, OCR, o qual é interno ao controlador CAN e permite, por *software*, configurar os *drivers* de saída. Existem 4 tipos de *driver* de saída possíveis:

- *float*;
- *pull-down*;
- *pull-up*;
- *push-pull*;

A configuração escolhida foi a de *push-pull*, com polaridade 0, escrevendo para tal no registo OCR a palavra DAH e cujo significado está ilustrado na Figura A6.10.

OCTP1	OCTN1	OCPOL1	OCTP0	OCTN0	OCPOL0	OCMOD1	OCMOD0
1	1	0	1	1	0	1	0

**Figura A6.10 - Registo OCR.**

OCTP1, OCTN1, OCTP0 e OCTN0 = "1111" seleccionam a configuração *push-pull*  
 OCPOL1 e OCPOL0 = "00" seleccionam polaridade 0  
 OCMOD1 e OCMOD0 = "10" seleccionam o modo de saída normal

Para completar a configuração dos transístores de saída, são ainda importantes os seguintes *bits*:

CR.6="S"=0 e CR.5="RA"=0, *bits* do registo interno de controlo, que significam:  
 "S"=0, a ressincronização é feita na transição de recessivo para dominante;  
 "RA"=0, não é utilizado o pino REF do microcontrolador.

CMR.7="Rx0 active"=1 e CMR.6="Rx1 active"=1, *bits* do registo interno de comando, que significam:

"Rx0 active"=1, Rx0 = CRx0;  
 "Rx1 active"=1, Rx1 =  $A_{VDD}/2$ , permitindo uma tensão de referência igual a  $A_{VDD}/2$ , para ligar a referência do *driver* de linha P82C250, ficando o pino CRx0 como linha de dados.

Por último e para concluir o procedimento de calibração do controlador CAN, é escrito no registo de controlo a palavra 0x02H, para que o *bit* RR fique ao nível baixo e assim o controlador volte ao estado normal de funcionamento.

## Iniciação do RTC (DS12887)

A programação do RTC é efectuada através de escrita nos registos A, B e C do mesmo [39]. Seguidamente é referido o significado dos *bits* consultados e alterados, referindo os valores utilizados.

### Registo A

Registo A =0x2C							
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0
0	0	1	0	1	1	0	0

**Figura A6.11 – Registo A do RTC DS12887.**

- *Update in progress (UIP)*: Flag de status que pode ser monitorizada. Quando ao nível lógico ‘1’ assinala uma actualização em progresso.
- *Bits (DV2..DV0)*: estes *bits* permitem activar ou inibir o oscilador interno do RTC. O valor DV2..DV0=010 é a única combinação de *bits* que habilita o oscilador interno e permite ao RTC manter o tempo.
- *Bits (RS3..RS0)*: estes 4 *bits* de selecção permitem escolher o período de interrupção (RS3..RS0=1100 para 62,5 ms).

### Registo B

Registo B=0xCA							
SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE
1	1	0	0	1	0	1	0

**Figura A6.12 – Registo B do RTC DS12887.**

- **SET**: Este *bit* quando colocado a ‘1’ permite actualização dos *bytes* correspondentes ao calendário e tempo sem interrupção a meio da iniciação.
- *Periodicly Interrupt Enable (PIE)*: É um *bit* de leitura/escrita que permite que o *bit Periodic Interrupt flag (PF)* do registo C faça com que o pino IRQ fique com o nível lógico ‘0’. Quando este *bit* é colocado a ‘1’, são geradas interrupções periódicas levando o valor de IRQ ao nível lógico ‘0’, sendo o período especificado por configuração dos *bits* (RS3..RS0) do registo A.
- *Alarm Interrupt Enable (AIE)*: É um *bit* de leitura/escrita, que, quando colocado a ‘0’, o *bit Alarm Flag (AF)* do registo C não inicia o sinal de IRQ.
- *Square Wave Enable (SQWE)*: Este *bit* serve para activar uma saída (SQWE) que gera uma onda quadrada com período igual ao período de interrupção escolhido (62.5 ms neste caso) por escrita nos *bits* RS3..RS0 do registo A.

- **Data Mode (DM)** : Este *bit* permite a escolher se a informação referente ao calendário ou tempo está no formato binário (1) ou BCD (0).
- **24/12**: Este *bit* de controlo permite seleccionar o formato para o *byte* das horas ('1' para o modo 24 horas e '0' para o modo de 12 horas).
- **Daylight Savings Enable (DSE)**: Quando colocado a '0' não executa as actualizações devido a alterações de hora originadas por horário de Verão ou Inverno.

Após estabelecidos os *set-points* anteriormente referidos, é colocado o *bit* SET com o valor '0' para que o RTC possa funcionar normalmente.

## Registo C

Registo C							
IRQF	PF	AF	UF	0	0	0	0

**Figura A6.13 – Registo C do RTC DS12887.**

O registo C é composto por *flags* cujo significado é o seguinte:

- **Interrupt Request Flag (IRQF)**: Esta *flag* é colocada a '1' quando uma das seguintes condições se verificar:  
 PF=PIE=1;  
 AF=AIE=1;  
 UF=UIE=1;

Quando esta *flag* tiver o valor lógico '1' o pino  $\overline{IRQ}$  toma o valor lógico '0'.

- **Periodic interrupt flag (PF)**: Esta *flag* tem o valor lógico '1' quando é detectada uma transição com período definido através dos valores dos bits RS3..RS0 do registo A.
- **Alarm Flag (AF)**: Indica que o tempo actual coincide com o tempo de alarme escolhido (não utilizado nesta aplicação).
- **Update Ended Interrupt Flag (UF)**: Esta *flag* tem o valor '1' após cada ciclo de actualização.

## Iniciação da SIO

A comunicação via RS232 com o microcontrolador, é configurada através de escrita nos seguintes registos [63]:

- *Serial Port Control (S0CON)*;
- *Power Control (PCON)*;
- *Timer Control (TCON)*;
- *Timer 0 and 1 Mode Control (TMOD)*;
- *Timer 1 High byte (TH1)*;
- *Timer 1 Low byte (TL1)*.

Os procedimentos seguidos para iniciar a UART interna do microcontrolador são os seguintes:

A escolha do modo de operação é realizada por escrita no registo S0CON. Assim, foi escrita a palavra 0x50H neste registo, uma vez que é pretendido o modo 1 (SM0=0 e SM1=1) de funcionamento, ou seja, UART de 8 *bits* com taxa de transmissão programável, bem como habilitar recepção série (REN=1).

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
0	1	0	1	0	0	0	0

**Figura A6.14 – Registo S0CON.**

A programação do *baud rate* é conseguida por escrita nos registos TMOD, TH1, TL1, TCON e PCON.

Como a UART no modo 1, o *baud rate* é gerado pelo *Timer 1*, devendo este ser programado para o modo 2 de funcionamento [63]. Por isso, foi escrita, no registo TMOD, a palavra 0x20H, conforme ilustra a Figura A6.16.

<i>Timer 1</i>				<i>Timer 0</i>			
Gate	C/T	M1	M0	Gate	C/T	M1	M0
0	0	1	0	0	0	0	0

**Figura A6.15 – Registo TMOD.**

Como, após as definições anteriores, a taxa de transmissão é dada pela seguinte fórmula [63]:

$$Baud\ rate = \frac{2^{SMOD}}{32} \times \frac{f_{osc}}{[12 \times (256 - TH1)]}$$

Foram, posteriormente, alterados os valores dos seguintes registos:

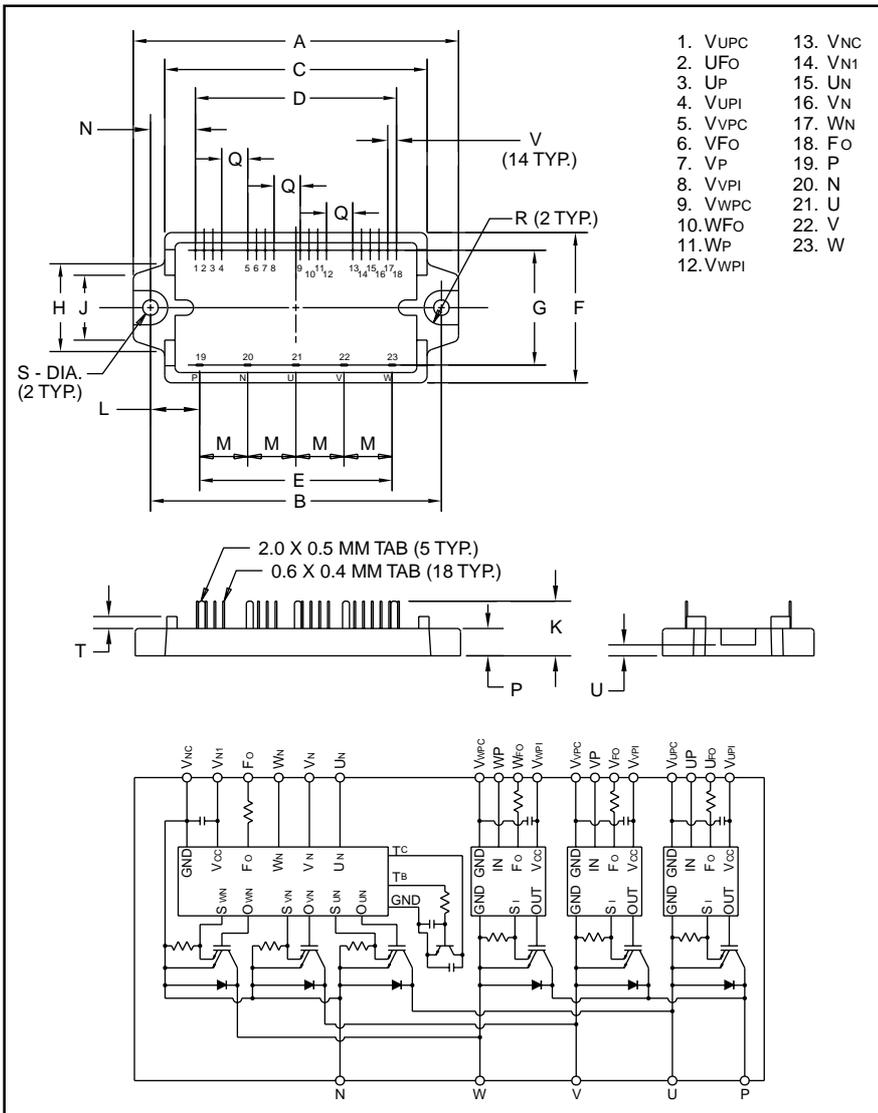
- Registo TCON, colocando o *bit* TR1 a 1 para activar o *Timer 1*;
- Registo PCON, para que o *baud rate* não seja o dobro do programado, foi escrito PCON=0x00H, garantindo assim que SMOD=0;
- Registos TH1 e TL1, atribuindo-lhes o valor 0xFAH, uma vez que é pretendido um *baud rate* de 4800 *bit/s*.

Por fim, são activadas a interrupção da SIO (UART) e a recepção de dados, concluindo assim o respectivo processo de iniciação.

**ANEXO 7 – ESPECIFICAÇÕES TÉCNICAS DO MÓDULO PM20CSJ060**

# PM20CSJ060

FLAT-BASE TYPE  
INSULATED PACKAGE



- 1. V<sub>UPC</sub>
- 2. U<sub>FO</sub>
- 3. U<sub>P</sub>
- 4. V<sub>UPI</sub>
- 5. V<sub>VPC</sub>
- 6. V<sub>FO</sub>
- 7. V<sub>P</sub>
- 8. V<sub>VPI</sub>
- 9. V<sub>WPC</sub>
- 10. W<sub>FO</sub>
- 11. W<sub>P</sub>
- 12. V<sub>WPI</sub>
- 13. V<sub>NC</sub>
- 14. V<sub>N1</sub>
- 15. U<sub>N</sub>
- 16. V<sub>N</sub>
- 17. W<sub>N</sub>
- 18. F<sub>O</sub>
- 19. P
- 20. N
- 21. U
- 22. V
- 23. W



**Description:**

Mitsubishi Intelligent Power Modules are isolated base modules designed for power switching applications operating at frequencies to 20kHz. Built-in control circuits provide optimum gate drive and protection for the IGBT and free-wheel diode power devices.

**Features:**

- Complete Output Power Circuit
- Gate Drive Circuit
- Protection Logic
  - Short Circuit
  - Over Current
  - Over Temperature
  - Under Voltage

**Applications:**

- Inverters
- UPS
- Motion/Servo Control
- Power Supplies

**Ordering Information:**

Example: Select the complete part number from the table below -i.e. PM20CSJ060 is a 600V, 20 Ampere Intelligent Power Module.

Type	Current Rating Amperes	V <sub>CES</sub> Volts (x 10)
PM	20	60

Outline Drawing and Circuit Diagram

Dimensions	Inches	Millimeters
A	3.72±0.04	94.5±1.0
B	3.33±0.02	84.5±0.5
C	2.99	76.0
D	2.300±0.02	58.42±0.5
E	2.20±0.02	56.0±0.5
F	1.73±0.04	44.0±1.0
G	1.32±0.03	33.6±0.8
H	1.01	25.7
J	0.75	19.0
K	0.71±0.04	18.0±1.0

Dimensions	Inches	Millimeters
L	0.561	14.25
M	0.55±0.01	14.0±0.25
N	0.513	13.04
P	0.31±0.02	8.0±0.5
Q	0.300	7.62
R	0.20 Rad.	Rad. 5.0
S	0.18 Dia.	Dia. 4.5
T	0.14	3.5
U	0.13±0.02	3.2±0.5
V	0.100±0.01	2.54±0.25

# PM20CSJ060

FLAT-BASE TYPE  
INSULATED PACKAGE

## Absolute Maximum Ratings, $T_j = 25^\circ\text{C}$ unless otherwise specified

Ratings	Symbol	PM20CSJ060	Units
Power Device Junction Temperature	$T_j$	-20 to 150	$^\circ\text{C}$
Storage Temperature	$T_{\text{stg}}$	-40 to 125	$^\circ\text{C}$
Case Operating Temperature	$T_C$	-20 to 100	$^\circ\text{C}$
Mounting Torque, M4 Mounting Screws	—	0.98 ~ 1.47	N · m
Module Weight (Typical)	—	60	Grams
Supply Voltage Protected by OC and SC ( $V_D = 13.5 - 16.5\text{V}$ , Inverter Part)	$V_{\text{CC(prot.)}}$	400	Volts
Isolation Voltage (Main Terminal to Baseplate, AC 1 min.)	$V_{\text{iso}}$	2500	Vrms

## Control Sector

Supply Voltage (Applied between $V_{\text{UP1}}-V_{\text{UPC}}$ , $V_{\text{VP1}}-V_{\text{VPC}}$ , $V_{\text{WP1}}-V_{\text{WPC}}$ , $V_{\text{N1}}-V_{\text{NC}}$ )	$V_D$	20	Volts
Input Voltage (Applied between $U_P-V_{\text{UPC}}$ , $V_P-V_{\text{VPC}}$ , $W_P-V_{\text{WPC}}$ , $U_N \cdot V_N \cdot W_N-V_{\text{NC}}$ )	$V_{\text{CIN}}$	20	Volts
Fault Output Supply Voltage (Applied between $U_{\text{FO}}-V_{\text{UPC}}$ , $V_{\text{FO}}-V_{\text{VPC}}$ , $W_{\text{FO}}-V_{\text{WPC}}$ , $F_O-V_{\text{NC}}$ )	$V_{\text{FO}}$	20	Volts
Fault Output Current (Sink Current of $U_{\text{FO}}$ , $V_{\text{FO}}$ , $W_{\text{FO}}$ and $F_O$ Terminal)	$I_{\text{FO}}$	20	mA

## IGBT Inverter Sector

Collector-Emitter Voltage ( $V_D = 15\text{V}$ , $V_{\text{CIN}} = 15\text{V}$ )	$V_{\text{CES}}$	600	Volts
Collector Current, ( $T_C = 25^\circ\text{C}$ )	$I_C$	20	Amperes
Peak Collector Current, ( $T_C = 25^\circ\text{C}$ )	$I_{\text{CP}}$	40	Amperes
Supply Voltage (Applied between P - N)	$V_{\text{CC}}$	450	Volts
Supply Voltage, Surge (Applied between P - N)	$V_{\text{CC(surge)}}$	500	Volts
Collector Dissipation	$P_C$	56	Watts

**PM20CSJ060**FLAT-BASE TYPE  
INSULATED PACKAGE**Electrical and Mechanical Characteristics,  $T_j = 25^\circ\text{C}$  unless otherwise specified**

Characteristics	Symbol	Test Conditions	Min.	Typ.	Max.	Units
<b>Control Sector</b>						
Over Current Trip Level Inverter Part	OC	$-20^\circ\text{C} \leq T \leq 125^\circ\text{C}$ , $V_D = 15\text{V}$	28	38	—	Amperes
Short Circuit Trip Level Inverter Part	SC	$-20^\circ\text{C} \leq T \leq 125^\circ\text{C}$ , $V_D = 15\text{V}$	—	57	—	Amperes
Over Current Delay Time	$t_{\text{off}}(\text{OC})$	$V_D = 15\text{V}$	—	10	—	$\mu\text{s}$
Over Temperature Protection	OT	Trip Level	100	110	120	$^\circ\text{C}$
	$\text{OT}_r$	Reset Level	—	90	—	$^\circ\text{C}$
Supply Circuit Under Voltage Protection	UV	Trip Level	11.5	12.0	12.5	Volts
	$\text{UV}_r$	Reset Level	—	12.5	—	Volts
Supply Voltage	$V_D$	Applied between $V_{\text{UP1}}-V_{\text{UPC}}$ , $V_{\text{VP1}}-V_{\text{VPC}}$ , $V_{\text{WP1}}-V_{\text{WPC}}$ , $V_{\text{N1}}-V_{\text{NC}}$	13.5	15	16.5	Volts
Circuit Current	$I_D$	$V_D = 15\text{V}$ , $V_{\text{CIN}} = 15\text{V}$ , $V_{\text{N1}}-V_{\text{NC}}$	—	18	25	mA
		$V_D = 15\text{V}$ , $V_{\text{CIN}} = 15\text{V}$ , $V_{\text{XP1}}-V_{\text{XPC}}$	—	7	10	mA
Input ON Threshold Voltage	$V_{\text{th}}(\text{on})$	Applied between	1.2	1.5	1.8	Volts
Input OFF Threshold Voltage	$V_{\text{th}}(\text{off})$	$U_P-V_{\text{UPC}}$ , $V_P-V_{\text{VPC}}$ , $W_P-V_{\text{WPC}}$ , $U_N \cdot V_N \cdot W_N-V_{\text{NC}}$	1.7	2.0	2.3	Volts
PWM Input Frequency	$f_{\text{PWM}}$	3- $\phi$ Sinusoidal	—	15	20	kHz
Fault Output Current	$I_{\text{FO}}(\text{H})$	$V_D = 15\text{V}$ , $V_{\text{FO}} = 15\text{V}$	—	—	0.01	mA
	$I_{\text{FO}}(\text{L})$	$V_D = 15\text{V}$ , $V_{\text{FO}} = 15\text{V}$	—	10	15	mA
Minimum Fault Output Pulse Width	$t_{\text{FO}}$	$V_D = 15\text{V}$	1.0	1.8	—	ms

# PM20CSJ060

FLAT-BASE TYPE  
INSULATED PACKAGE

## Electrical and Mechanical Characteristics, $T_j = 25^\circ\text{C}$ unless otherwise specified

Characteristics	Symbol	Test Conditions	Min.	Typ.	Max.	Units
<b>IGBT Inverter Sector</b>						
Collector Cutoff Current	$I_{CES}$	$V_{CIN} = 15\text{V}, V_{CE} = V_{CES}, T_j = 25^\circ\text{C}$	—	—	1.0	mA
		$V_{CIN} = 15\text{V}, V_{CE} = V_{CES}, T_j = 125^\circ\text{C}$	—	—	10	mA
Diode Forward Voltage	$V_{EC}$	$-I_C = 20\text{A}, V_D = 15\text{V}, V_{CIN} = 15\text{V}$	—	2.5	3.5	Volts
Collector-Emitter Saturation Voltage	$V_{CE(sat)}$	$V_D = 15\text{V}, V_{CIN} = 0\text{V}, I_C = 20\text{A}$	—	1.8	2.5	Volts
		$V_D = 15\text{V}, V_{CIN} = 0\text{V}, I_C = 20\text{A}, T_j = 125^\circ\text{C}$	—	1.9	2.6	Volts
Inductive Load Switching Times	$t_{on}$		0.3	0.6	1.5	$\mu\text{s}$
	$t_{rr}$	$V_D = 15\text{V}, V_{CIN} = 0 \leftrightarrow 15\text{V}$	—	0.12	0.3	$\mu\text{s}$
	$t_{C(on)}$	$V_{CC} = 300\text{V}, I_C = 20\text{A}$	—	0.2	0.8	$\mu\text{s}$
	$t_{off}$	$T_j = 125^\circ\text{C}$	—	1.5	2.3	$\mu\text{s}$
	$t_{C(off)}$		—	0.5	1.5	$\mu\text{s}$

## Thermal Characteristics

Characteristic	Symbol	Condition	Min.	Typ.	Max.	Units
Junction to Case Thermal Resistance	$R_{th(j-c)Q}$	Each IGBT	—	—	2.2	$^\circ\text{C/Watt}$
	$R_{th(j-c)F}$	Each FWDi	—	—	4.5	$^\circ\text{C/Watt}$
Contact Thermal Resistance	$R_{th(c-f)}$	Case to Fin Per Module, Thermal Grease Applied	—	—	0.083	$^\circ\text{C/Watt}$

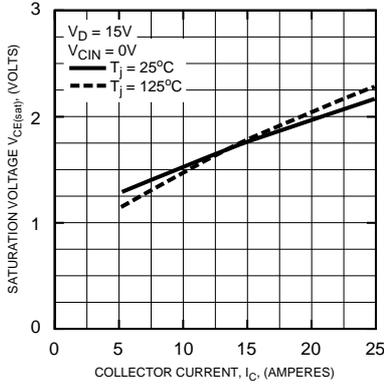
## Recommended Conditions for Use

Characteristic	Symbol	Condition	Value	Units
Supply Voltage	$V_{CC}$	Applied across P-N Terminals	0 ~ 400	Volts
	$V_D$	Applied between $V_{UP1}-V_{UPC}, V_{N1}-V_{NC}, V_{VP1}-V_{VPC}, V_{WP1}-V_{WPC}$	$15 \pm 1.5$	Volts
Input ON Voltage	$V_{CIN(on)}$	Applied between	0 ~ 0.8	Volts
Input OFF Voltage	$V_{CIN(off)}$	$U_P-V_{UPC}, V_P-V_{VPC}, W_P-V_{WPC}, U_N \cdot V_N \cdot W_N-V_{NC}$	$4.0 \sim V_D$	Volts
PWM Input Frequency	$f_{PWM}$	Using Application Circuit	5 ~ 20	kHz
Minimum Dead Time	$t_{dead}$	Input Signal	$\geq 2.0$	$\mu\text{s}$

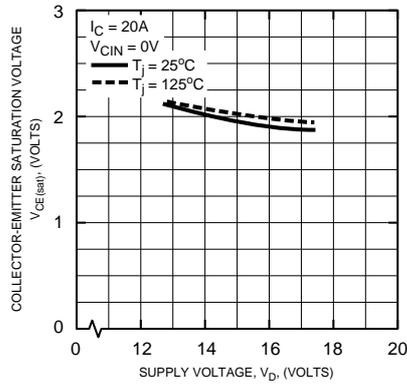
# PM20CSJ060

FLAT-BASE TYPE  
INSULATED PACKAGE

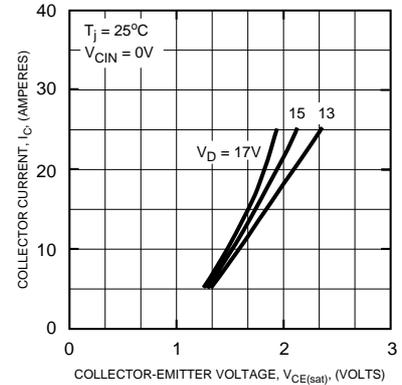
**SATURATION VOLTAGE CHARACTERISTICS (TYPICAL)**



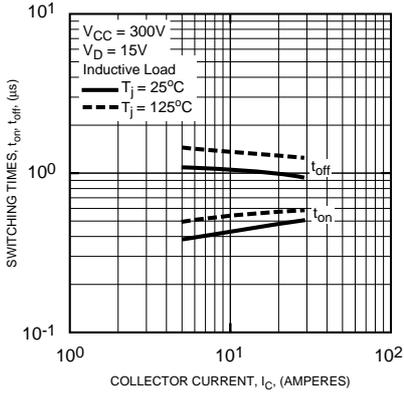
**COLLECTOR-EMITTER SATURATION VOLTAGE CHARACTERISTICS (TYPICAL)**



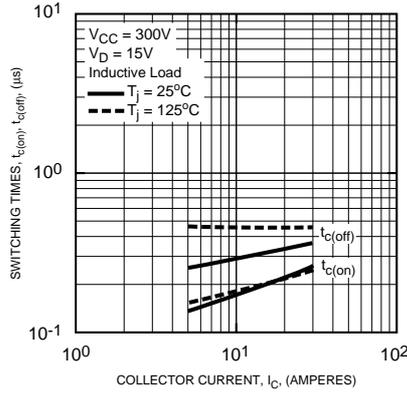
**OUTPUT CHARACTERISTICS (TYPICAL)**



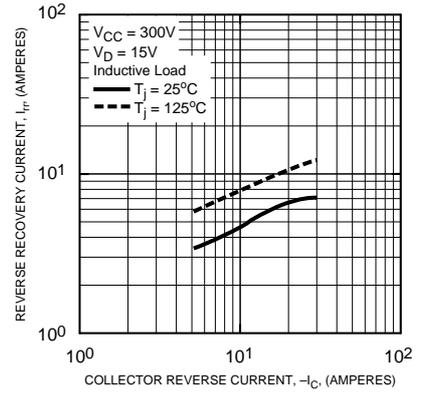
**SWITCHING TIME VS. COLLECTOR CURRENT (TYPICAL)**



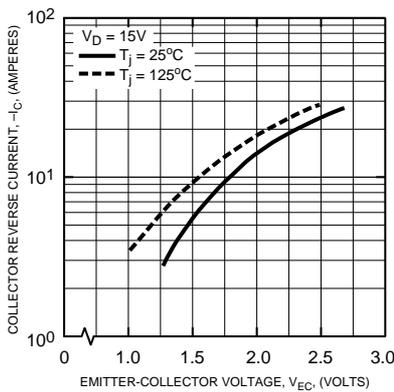
**SWITCHING TIME VS. COLLECTOR CURRENT (TYPICAL)**



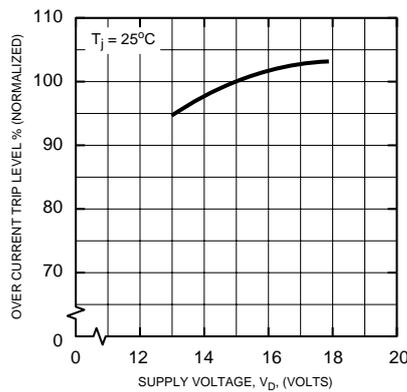
**REVERSE RECOVERY CURRENT VS. COLLECTOR CURRENT (TYPICAL)**



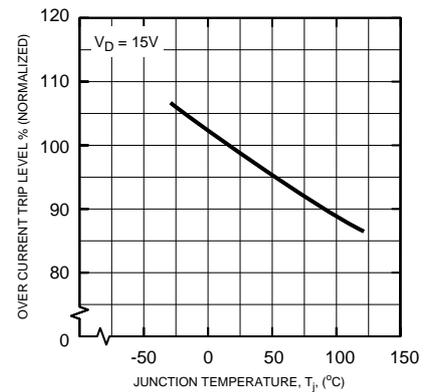
**DIODE FORWARD CHARACTERISTICS**



**OVER CURRENT TRIP LEVEL VS. SUPPLY VOLTAGE (TYPICAL)**



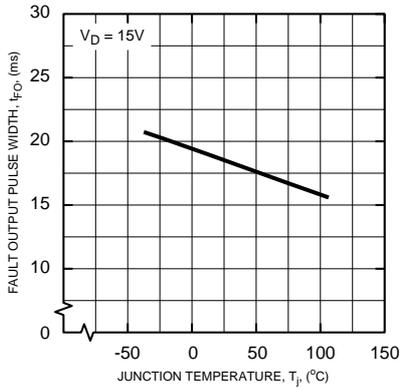
**OVER CURRENT TRIP LEVEL TEMPERATURE DEPENDENCY (TYPICAL)**



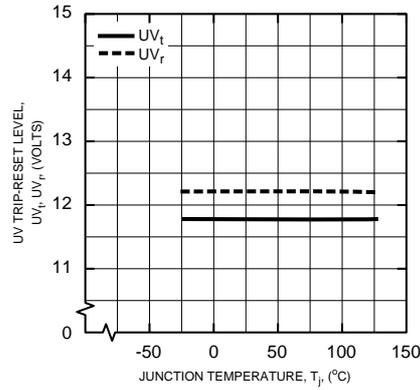
# PM20CSJ060

FLAT-BASE TYPE  
INSULATED PACKAGE

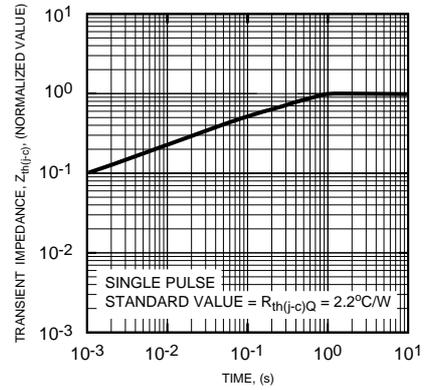
FAULT OUTPUT PULSE WIDTH VS. TEMPERATURE (TYPICAL)



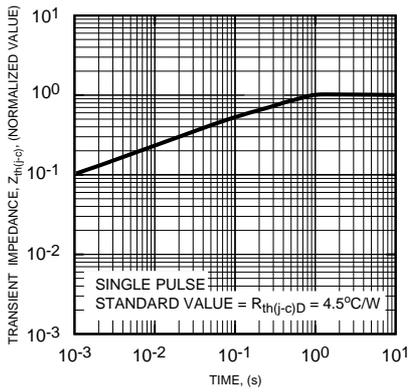
CONTROL SUPPLY VOLTAGE TRIP-RESET LEVEL TEMPERATURE DEPENDENCY (TYPICAL)



TRANSIENT THERMAL IMPEDANCE CHARACTERISTICS (Each IGBT)



TRANSIENT THERMAL IMPEDANCE CHARACTERISTICS (Each FWDi)



# GLOSSÁRIO

**Algoritmo:** Processo de resolver um problema. Normalmente expresso como uma fórmula matemática ou uma declaração lógica.

**Arquitectura:** A forma como os circuitos lógicos de um microprocessador são colocados para construir a lógica de cálculo, circuitos de memória, e estrutura de entrada/saída. A forma como determinado conjunto de unidades é utilizado para conseguir um objectivo.

**Barramento; Bus:** Topologia de uma rede de comunicações, onde todos os nós são alcançáveis por ligações passivas que permitem transmissão em ambos os sentidos. O arranjo de ligações eléctricas que permitem que várias partes do microprocessador ou microcontrolador comuniquem. Um barramento possui sinais de endereços, dados, e controlo.

**Baud rate:** Velocidade de transmissão de dados em transmissão serie medida em *bits* por segundo (bit/s).

**Binário:** Sistema numérico com dois caracteres (representados geralmente por 0 e 1). É o sistema numérico mais simples de implementar utilizando sistemas digitais.

**Bit:** Um dígito binário. Um elemento de informação numa palavra binária. Pode ser verdadeiro (nível lógico 1) ou falso (nível lógico 0).

**Bit rate:** Número de *bits* por tempo durante a transmissão, independentemente da representação do *bit*.

**Bit stuffing:** Técnica utilizada em protocolos orientados ao *bit* onde quando o emissor encontrar um certo número de *bits* consecutivos de igual valor nos dados, automaticamente insira um *bit* de valor complementar – *stuff bit* – na *frame* a enviar. Os receptores levam a cabo um processo contrário.

**Bridge:** Um dispositivo que liga duas redes locais, *Local Area Networks* (LANs), ou dois segmentos da mesma LAN. As duas redes a serem ligadas podem ser de tipo semelhante ou dissimilar. Ao contrário dos *routers*, as *briges* são independentes dos protocolos. As *bridges* operam ao nível 2 do modelo de referência OSI.

**Buffer:** Registo auxiliar; armazenamento intermédio temporário.

**Byte:** Uma palavra digital composta por 8 *bits*.

**Cablagem:** Nome atribuído, na gíria, ao sistema de cabos e fios de determinado sistema.

**Central Processing Unit (CPU):** A lógica que executa a computação e suporte de dados num microprocessador ou computador.

**Chip:** Um circuito integrado.

**Contention-based arbitration:** Processo de arbitragem *Carrier Sense Multiple Access* (CSMA), onde acesso simultâneo de múltiplos nós resulta em contenção. Uma *frame* irá sobreviver à contenção.

**Controlo liga/desliga; On/Off:** Tipo de controlo que se resume ao ligar (activar) ou desligar (anular) determinado tipo de acções.

**Conversor Analógico-digital (A/D):** Circuito eléctrico que converte um sinal analógico (tensão ou corrente) num número binário proporcional (palavra digital).

**Conversor Digital-analógico (D/A):** Circuito eléctrico que converte uma palavra digital (binário) numa tensão ou corrente proporcionais (sinal analógico).

**Counter/timer:** Um circuito digital utilizado para permitir funções de contagem (*counter*) ou temporização (*timer*) num microprocessador ou num dos seus dispositivos de E/S.

**Cyclical Redundancy Check (CRC):** Um algoritmo de verificação de erro que trata os dados como coeficientes de um polinómio. O resto é transmitido com os dados e comparado com um resto recalculado no lado receptor.

**Databook:** Manual elaborado pelo fabricante onde se podem encontrar os dados técnicos referentes a determinado componente (IC).

**Diagrama de blocos:** Uma ferramenta de representação onde uma série de blocos interligados mostra o fluxo do sinal (processamento) e controlo num circuito eléctrico.

**Diagrama esquemático:** Um documento que mostra cada componente electrónico num circuito e mostra como este se liga a todos os outros no circuito.

**Emissor:** Dispositivo que converte sinais informação ou sinais de dados em sinais eléctricos ou ópticos para que estes possam ser transferidos através do meio de comunicação.

**Especificação:** Uma declaração detalhada de exactamente o que deve ser feito para resolver um problema.

**Flag:** Indicador de estado.

**Frame:** Unidade de dados pertencente ao nível de ligação do protocolo, especificando o arranjo e significado de *bits* ou campos de *bits* na sequência de transferência através do meio de transmissão

**Full-duplex:** Um sistema de transmissão de dados que permite que os dados sejam transmitidos em ambas direcções simultaneamente.

**Gateway:** É uma combinação de *hardware* e *software* para ligar dois tipos diferentes de redes. As *gateways* operam ao nível 7 do modelo OSI. No uso moderno, o termo *Gateway* refere-se a sistemas que fazem a tradução de algum formato nativo para outro.

**Ground:** Terra.

**Half-duplex:** Um sistema de transmissão de dados que permite transmissão de dados em apenas uma direcção de cada vez.

**Hardware:** Componente físico; equipamento.

**Hex:** Hexadecimal. Sistema de 16 caracteres ( 0 a 9 e de A a F). Utilizado como abreviatura de números binários 0000 a 1111.

**High Byte:** O *byte* mais significativo de uma palavra composta por vários *bytes*.

**Idle:** Inactivo.

**Implementação:** Processo de converter o projecto num produto funcional.

**Integrated Circuit (IC):** Um circuito implementado em estado sólido criado por interligação e um número de transístores numa pastilha de silício. Também vulgarmente designado como *chip*.

**Interrupção:** Um pedido para iniciar a execução de um programa específico (rotina de serviço) causado por um sinal eléctrico exterior.

**Interrupção mascarável:** Uma entrada de interrupção que pode ser desactivada pelo programador.

**Interrupção não mascarável; Nonmaskable interrupt (NMI):** Uma entrada de interrupção que está sempre activa, não podendo ser desactivada pelo programador.

**Least significant bit (LSB):** A posição de *bit* numa palavra binária com menor peso.

**Low byte:** O *byte* menos significativo de uma palavra composta por vários *bytes*.

**Microcontrolador:** Um circuito integrado que possui um microprocessador, RAM, ROM e portos de E/S .

**Microprocessador:** Um circuito integrado que implementa a arquitectura de um CPU digital de um computador.

**Multicast:** Uma *frame* de dados pode ser endereçada a um grupo de nós em simultâneo. *Broadcast* é um caso especial de *multicast*, onde uma *frame* é endereçada a todos os nós simultaneamente.

**Multi-mestre:** Sistema dividido em vários nós onde cada nó pode temporariamente controlar a acção dos outros nós.

**Nível do barramento:** Um dos níveis lógicos complementares: No CAN estes níveis designam-se por “dominante” ou “recessivo”. O valor “dominante” representa o nível lógico “0”, e o “recessivo” o nível lógico “1”. Durante a transmissão simultânea de *bits* “dominantes” e “recessivos”, o valor resultante no barramento será o “dominante”.

**Nó:** Qualquer elemento, ligado a uma linha de comunicação, capaz de comunicar através da rede de acordo com uma especificação de um protocolo de comunicação.

**Non-return-to-zero (NRZ):** Método de representar sinais binários. Dentro de um e ao mesmo tempo, o nível do sinal não sofre alteração, i.e. uma sequência de *bits* contendo o mesmo valor lógico provoca ausência de transições.

**Polling:** Um método de servir um porto E/S. Quando um porto E/S é servido com uma rotina de *polling*, a rotina verifica todos os portos para verificar se estes necessitam de serviço e então serve os que requererem o mesmo.

**Prioridade:** Atributo a uma *frame* controlando a sua posição durante a arbitragem. Uma elevada prioridade incrementa a possibilidade de uma *frame* ganhar o processo de arbitragem.

**Personal Computer (PC):** Um computador com uma arquitectura semelhante a um IBM *Personal Computer* (baseado no microprocessador 8080, 80286, 80386, etc.).

**Programa:** Um conjunto de instruções que informam o microprocessador como processar os dados para conseguir realizar determinada tarefa.

**Projecto:** Descrição de como um problema será solucionado usando dispositivos específicos (*hardware*) ou código de programas (*software*). O projecto refere a especificação que detalha como o projecto deve actuar.

**Protocolo:** Conjunto formal de convenções e regras para troca de informação entre dois nós, incluindo da especificação da administração de *frames*, transferência de *frames* e nível físico.

**Porto:** Circuitos num microprocessador utilizados para transferir dados de e para o exterior do sistema.

**Porto de entrada/saída (E/S):** Um porto num sistema baseado num microprocessador utilizado especificamente para transferir dados de e para o sistema.

**Programa residente:** Um programa que é carregado na memória principal e permanece em memória por forma a permitir que seja de rápido acesso.

**Random-Access Memory (RAM):** Um sistema de memória onde cada elemento (*bit*, *byte* ou *word*) pode ser acedido tão facilmente como os restantes.

**Read Only Memory (ROM):** Um dispositivo de memória que guarda instruções do programa ou dados. Uma vez programada uma ROM não pode ser alterada com facilidade.

**Receptor:** Dispositivo que converte sinais físicos utilizados na transmissão em sinais lógicos de informação ou sinais de dados.

**Registo:** Parte de um microprocessador utilizada para guardar temporariamente uma palavra digital.

**Relógio; Clock:** Sinal gerado por um oscilador e que é utilizado para permitir o tempo base para um microprocessador e/ou circuitos a ele associados.

**Repeater; repetidores:** Dispositivo utilizado em redes para regenerar ou replicar um sinal. Os repetidores são utilizados em sistemas de transmissão para regenerar sinais analógicos ou digitais alterados durante a transmissão. Os repetidores analógicos frequentemente podem apenas amplificar o sinal enquanto que repetidores digitais podem restituir ao sinal a sua qualidade original. (nível 1 do modelo OSI).

**Reset:** Uma interrupção especial utilizada para iniciar o microprocessador. O vector de *reset* guarda o primeiro passo do programa.

**Ripple:** Ondulação residual.

**Rotina de serviço à interrupção:** Um programa especial chamado para servir um pedido de interrupção.

**Router:** Dispositivo que liga duas redes locais. Os *routers* são idênticos às *bridges*, mas fornecem funcionalidade adicional, tal como a habilidade de filtrar mensagens e enviá-las para diferentes locais baseado em vários critérios (nível 3 do modelo OSI).

**RS-232:** Um *standard* eléctrico para transmissão de dados via série. Este *standard* define um nível lógico 1 como sendo maior do que +3 V DC e um nível lógico 0 como sendo menor do que -3 V DC.

**Serial I/O:** Uma técnica de comunicação que converte palavras binárias numa sequência de *bits* enviados um após o outro através de um fio.

**Set-point:** Ponto de acerto.

**Software:** Conjunto de programas; programação geral; documentação.

**Standard:** Padrão; norma.

**Start bit:** Um *bit* de dados colocado antes de uma palavra de dados para indicar o início de uma transmissão série.

**Start of Frame (SOF):** *Bit* que assinala o início de uma nova *frame*.

**Status:** Estado; condição.

**Stop bit:** Um *bit* de dados colocado após uma palavra de dados para indicar o fim de uma transmissão série.

**Synchronous data:** Transmissão de dados onde o tempo entre elementos de dados (palavras ou *bits*) é conhecido (fixo).

**Transceiver do barramento de dados:** Um amplificador bidireccional utilizado para colocar ou receber dados de um barramento. Quando não utilizado no barramento, as saídas do *transceiver* estão no estado de alta impedância.

**Universal asynchronous receiver trasnmitter (UART):** Dispositivo que converte dados paralelos em dados série e dados série para dados paralelo.

**Unidade Host:** Unidade central; unidade principal; unidade hospedeira.

**Zénite:** Ponto mais elevado.