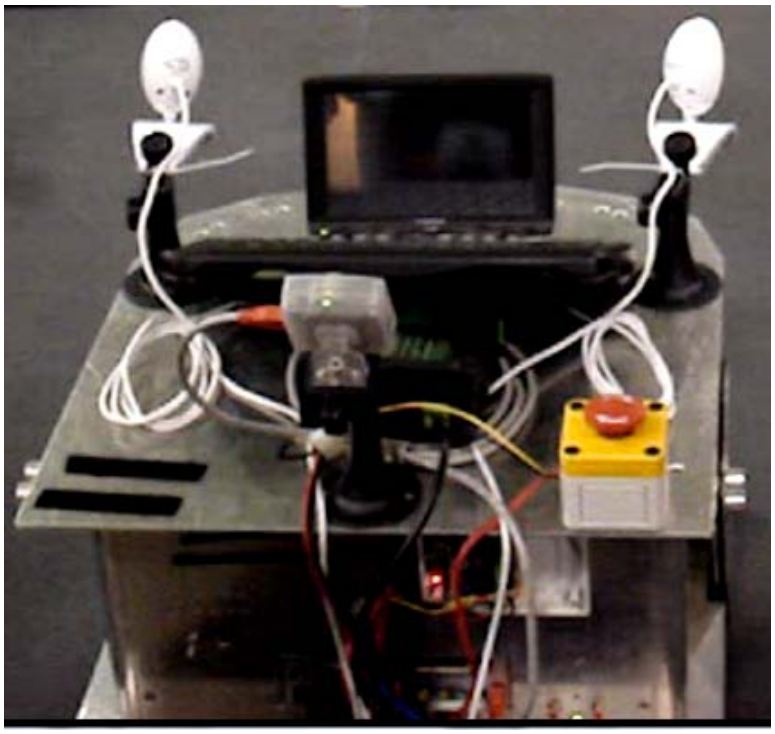
INSTITUTO DE SISTEMAS E ROBÓTICA
Departamento de Engenharia Electrotécnica – Universidade de Coimbra
Pólo II da Universidade de Coimbra
3030-290 COIMBRA, PORTUGAL
Tel: +351 239 796 200

Manual Técnico do ISRobot



Luís Filipe Rodrigues Alves

Coimbra, Julho de 2008

Conteúdo

Lista de Figuras

Capítulo 1

1 Introdução

O ***ISROBOT*** é um robô móvel diferencial de dimensão reduzida, destinado ao teste e desenvolvimento de novos algoritmos de controlo e navegação, com vista à participação em provas de Condução Autónoma que se realizam todos os anos no âmbito do Festival Nacional de Robótica. Esta tem de ser totalmente autónoma e tem de ter a capacidade de percepção do meio semi-estruturado que compõe os vários desafios que a prova apresenta.

A sua arquitectura é baseada num sistema distribuído e modular, tanto ao nível do hardware como do software. Basicamente esta assenta sobre um barramento de comunicação CAN, onde se encontram ligados os diversos módulos de hardware/software que irão ser descritos mais em detalhe nos capítulos seguintes.

Capítulo 2

2 Descrição do Sistema

Para além de uma estrutura de alumínio dotada de tracção diferencial e com dimensões reduzidas, o ISRobot foi equipado com vários sensores para permitir o desenvolvimento de algoritmos no contexto da prova de condução autónoma.

2.1 Estrutura Física

O ISRobot consiste numa estrutura em alumínio com duas rodas motrizes (tracção diferencial) na parte da frente e uma roda castor de apoio na parte de trás.
Temos no seu interior grande parte do sistema que o compõem nomeadamente: sistema de controlo (PC e sistema distribuído de CAN), actuação e alimentação. Só os sensores e o interface com o computador interno se encontram no exterior da estrutura.

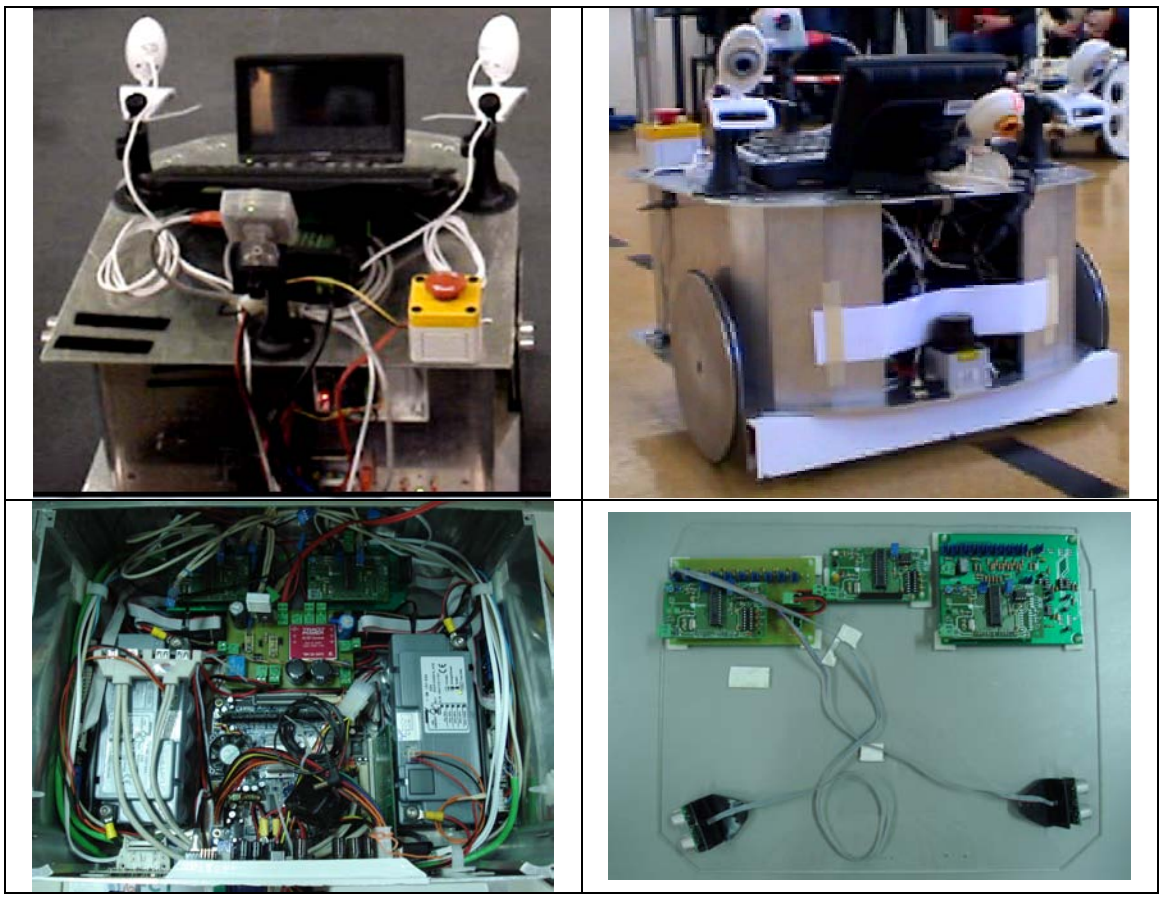


Figura 2.1: Sistemas que compõem o ISRobot

Em termos de dimensões a estrutura apresenta aproximadamente 50cm de largura, 50cm de comprimento e 30cm de altura, estando o seu peso distribuído simetricamente e pesando aproximadamente 20kg. É actuado recorrendo a dois motores DC de 24V com respectivo desmultiplicador de velocidade com factor de "1:33", alimentados por duas baterias de 12V e controlados por dois *drivers* de potência que garantem o seu efectivo controlo de forma independente. O binário de cada roda é de sensivelmente 2 Nm.

2.2 Constituição do Sistema

De seguida apresenta-se o sistema do ISRobot em termos de hardware, ou seja, são apresentados todos os componentes que constituem o sistema de navegação autónoma e a forma como estes estão interligados. É ainda descrito o fluxo de dados entre os diversos componentes.

2.2.1 Constituintes do Sistema

O sistema de navegação autónoma é então constituído por:

- ✓ 1 - PC Embutido EPIA MII

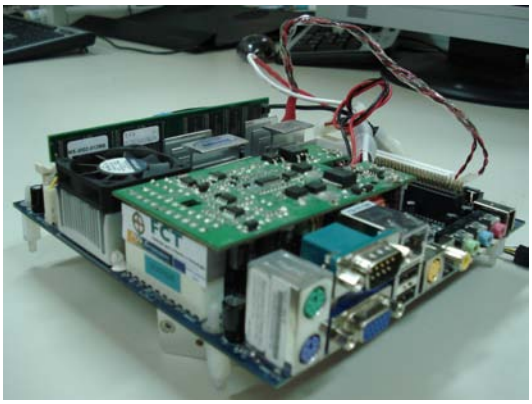


Figura 2.2: PC Embutido

- ✓ 1 - Placa de CAN PCI

Figura 2.3: Placa de CAN

- ✓ 2 - Bateria 12V (7 Ah)

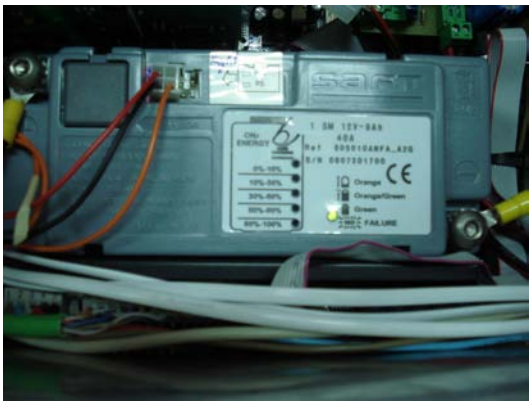


Figura 2.4: Bateria

- ✓ 5 - Módulos de interface com PIC18F258 (PIC_Base)

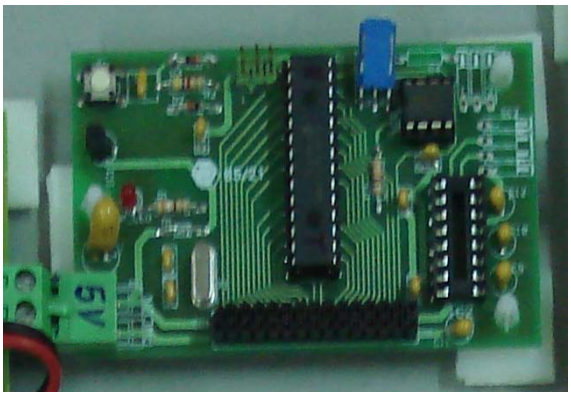


Figura 2.5: Pic Base

- ✓ 2 - Motor / Driver de Potência /Módulo de Hardware de interface com PIC_Base

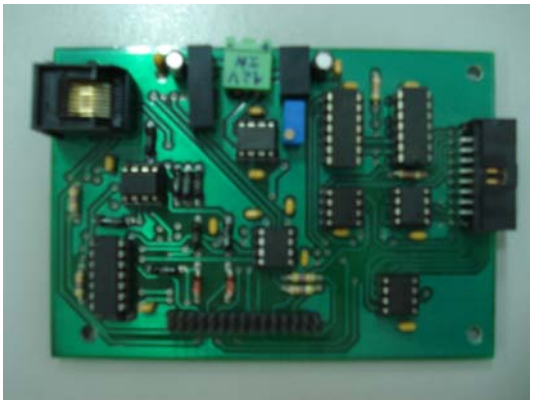


Figura 2.6: Módulo de interface com o *driver* de potência

✓ 1 - Ultra Sons / Módulo de Hardware de interface com PIC_Base

Figura 2.7: Ultra Som

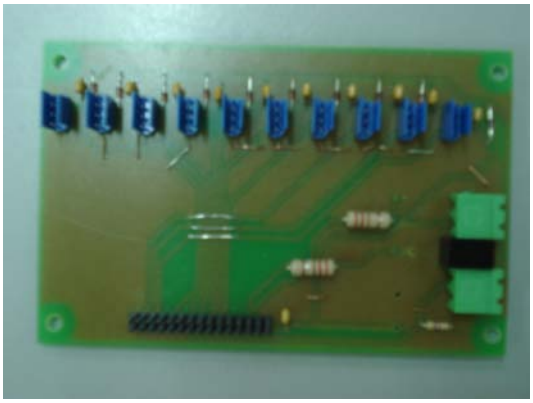


Figura 2.8: Módulo de interface com os ultra-sons

✓ 1 - Infravermelhos/ Módulo de Hardware de interface com PIC_Base

Figura 2.9: Infravermelhos

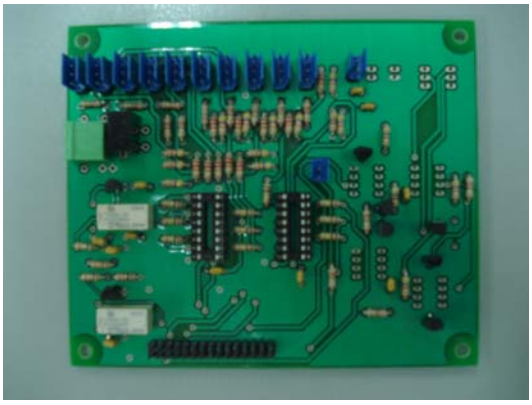


Figura 2.10: Módulo de interface com os Infravermelhos

- ✓ 1 – Sistema de alimentação e protecção

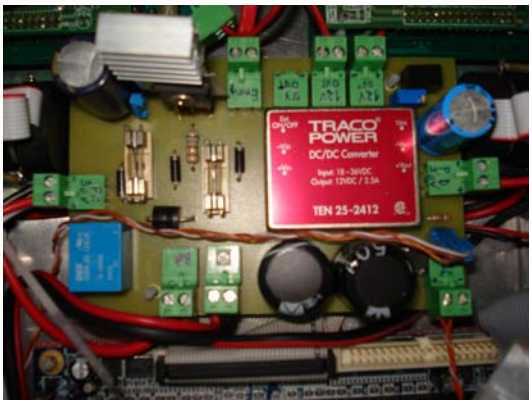


Figura 2.11: Módulo de distribuição da alimentação e protecção

2.3 Fluxo de Dados

A Figura 2.12 mostra a forma como se interligam os vários constituintes que foram apresentados anteriormente e a forma como se faz a troca de informação entre estes.

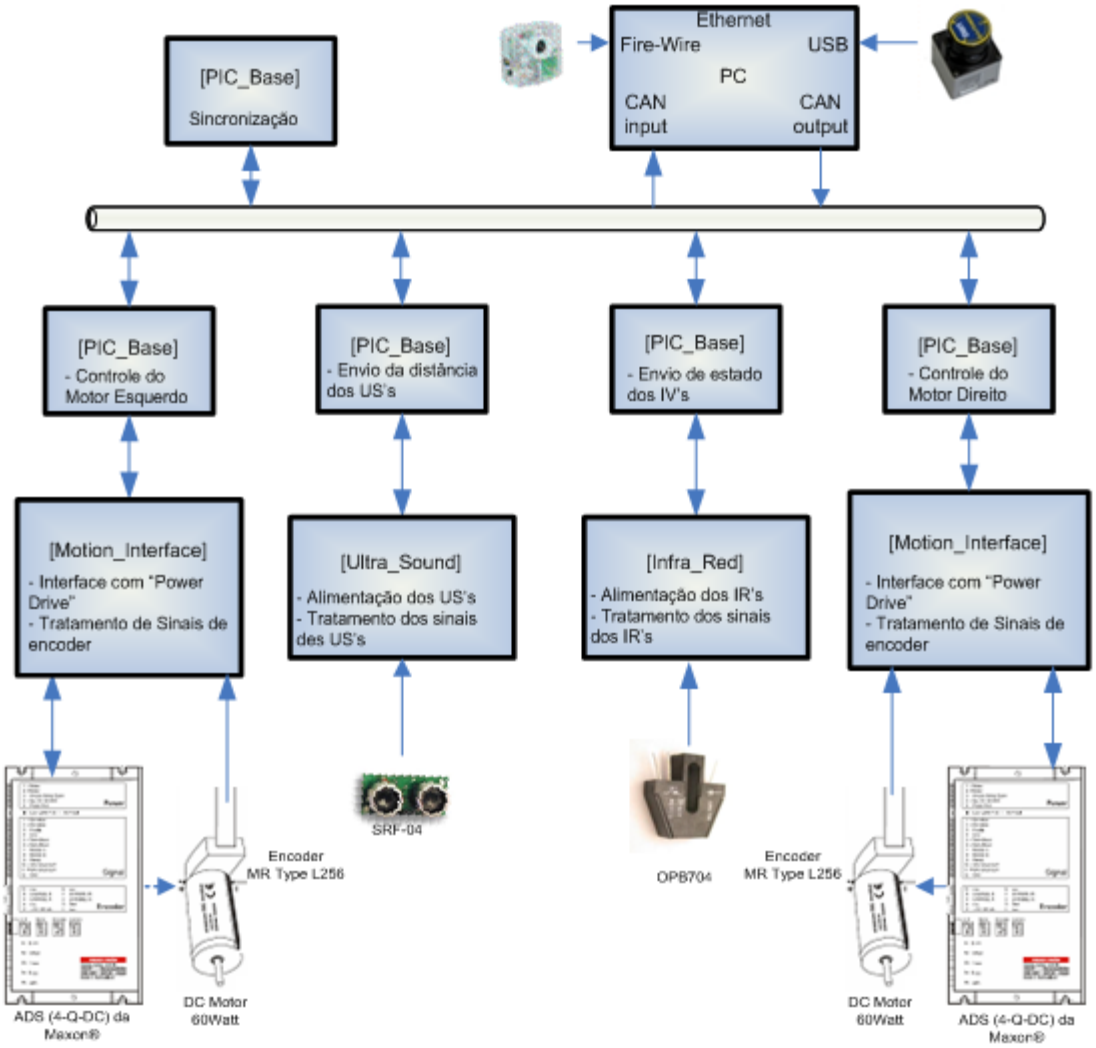


Figura 2.12: Fluxo de Dados

Como é perceptível, toda a informação circula entre os vários PIC's (*Programmable Interface Controller*) e o PC (*Personal Computer*), através de um barramento CAN (*Controller Area Network*). Funcionando o PC como supervisor/planeador, este encontra-se num nível hierárquico superior ao dos PICs, recebendo informação dos mesmos e transmitindo-lhes ordens para controlo de todo o sistema.

Neste caso como temos uma placa de CAN com duas portas no PC, temos uma disposição do barramento de CAN em forma de anel, começando e terminando no PC. Neste momento é possível usar apenas uma porta CAN, para leitura e escrita do sistema

e seu controlo. A outra é usada mais para *debug* e futuramente poderá ser usada num sistema mais complexo em que se pretenda mais robustez e fiabilidade do sistema.

Como a identificação de cada dispositivo de CAN é feita de forma inequívoca, a sua ordem de ligação nesse mesmo barramento é irrelevante.

Neste momento essa disposição é a seguinte:

PC(CAN_0)[1] → ControloMotorDireito[2] → IR[3] → Sincronismo[4] → US[5] → ControloMotorEsquerdo[6] → PC(CAN_1)[7]

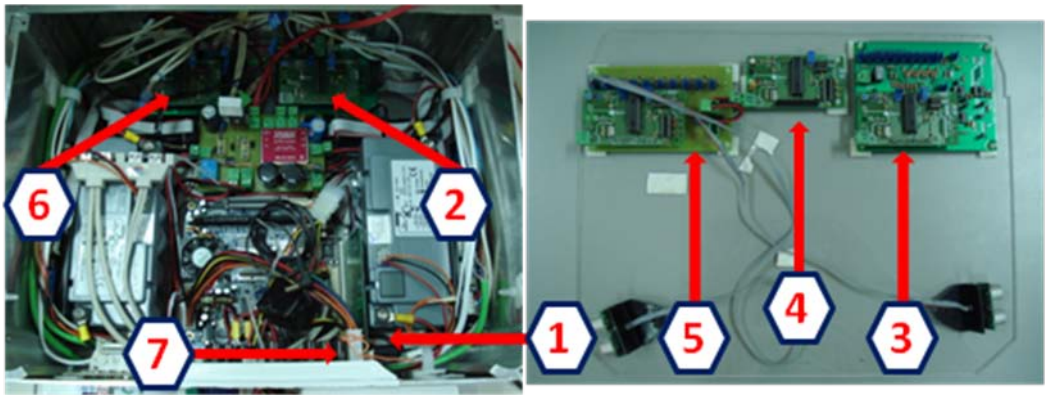


Figura 2.13: Disposição dos componentes no ISRobot

2.4 Sistema de Alimentação

O sistema de alimentação é composto por três níveis de tenções distintas. Como fonte de energia temos portanto as duas baterias de 12V em série para obter os 24V utilizados pelos *PowerDrives* que alimentam e controlam os motores respectivamente assim como o PC e os *PowerConverters* e que depois disponibiliza os 12V e 5V.

Quando se liga o interruptor das baterias ou conectamos a fonte externa, passamos a ter ~24V a entrar na placa de alimentação e protecção, presente no meio do robô e que distribui depois a alimentação pelo sistema. Esta placa tem três níveis de corte/limitação da distribuição de energia pelo sistema. O primeiro nível alimenta o PC e tem uma protecção de limitação da tensão máxima que alimenta a fonte do PC em 28V. O sistema tem um botão de corte de emergência (botão redondo vermelho) que corta a alimentação de todo o sistema à excepção do PC. Quando este está ligado temos os *drivers* de potência alimentados e os conversores DC-DC de 24V→12V→5V, no entanto as saídas dos conversores são controladas ainda por um outro interruptor (mini-interruptor de três posições situado no painel traseiro junto dos leds de sinalização do

lado esquerdo). As três posições são ON-OFF-Externo, em que na posição externa é ligado um controlo por rádio frequência de forma a desligar o sistema em fase de testes.

Os conversores são responsáveis pela alimentação do sistema distribuído de CAN, enquanto o laser, as câmaras, o monitor e o controlo por rádio frequência são alimentadas directamente da fonte do PC.

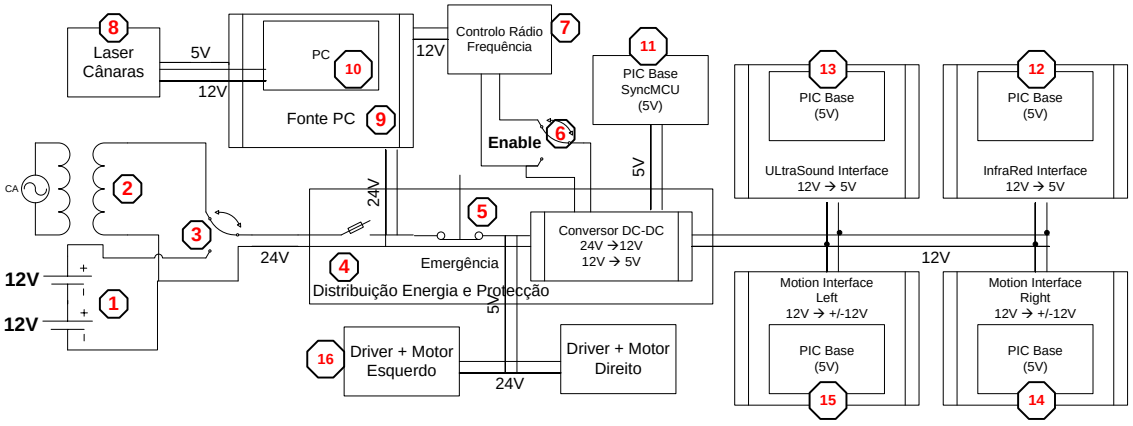


Figura 2.14: Esquema eléctrico

- (1) →Bateria de 12V

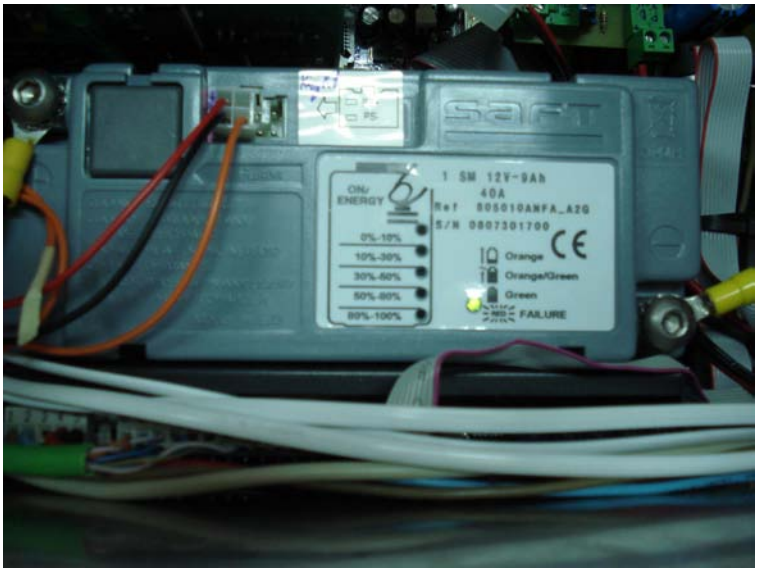


Figura 2.15: Bateria

- (2) →Alimentador Externo

Figura 2.16: Alimentador externo

➢(3) →Selector da fonte de energia em uso (Bateria ou Externa)

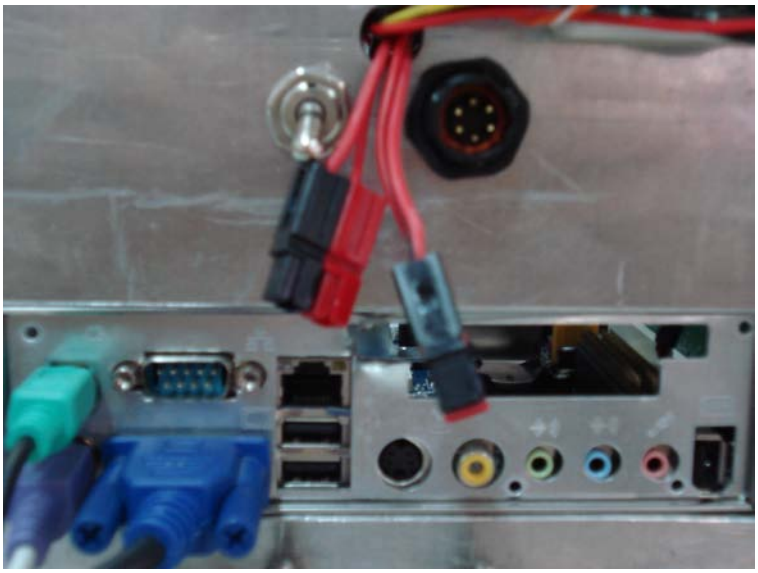


Figura 2.17: Selector da fonte de energia

➢(4) →Placa de protecção e distribuição de energia ao sistema

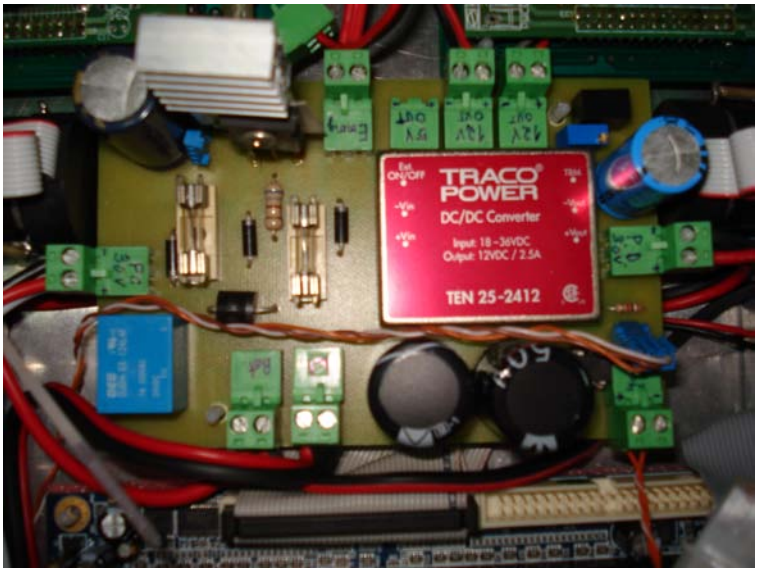


Figura 2.18: Placa de protecção e distribuição de energia ao sistema

- (5) →Corte da alimentação do sistema de controlo e actuação (Interruptor de pressão vermelho)

Figura 2.19: Corte da alimentação do sistema de controlo e actuação

- (6) →Botão de activação do conversor DC-DC 24V-12V

Figura 2.20: Botão de activação do conversor DC-DC 24V-12V

➢(7) →Controlo rádio frequência

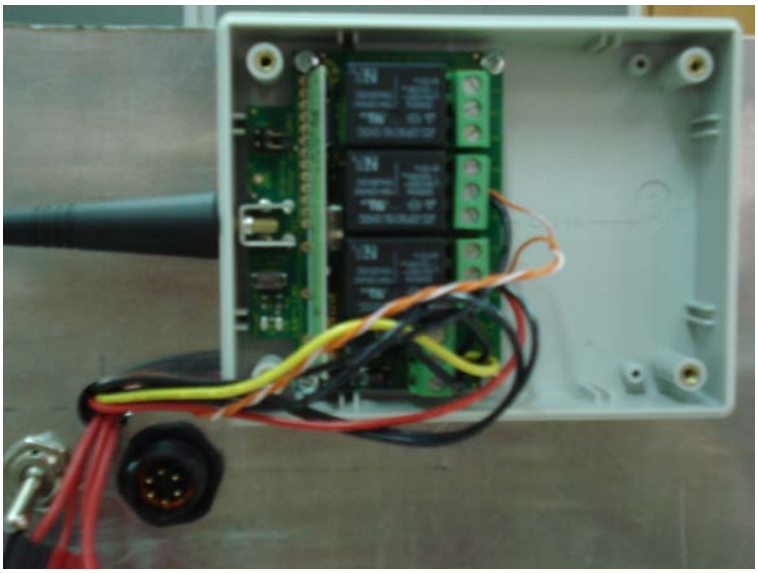


Figura 2.21: Controlo rádio frequência

➢(8) →Laser, Câmaras e Mini-TFT

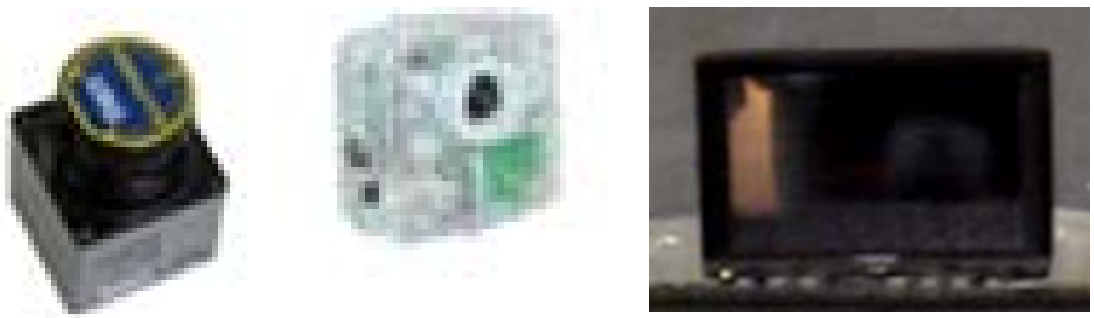


Figura 2.22: Laser, Câmaras e Mini-TFT

➢(9)(10) →Fonte do PC Epia mini-ITX MII e periféricos ligados ao mesmo (Laser, Câmaras e TFT)

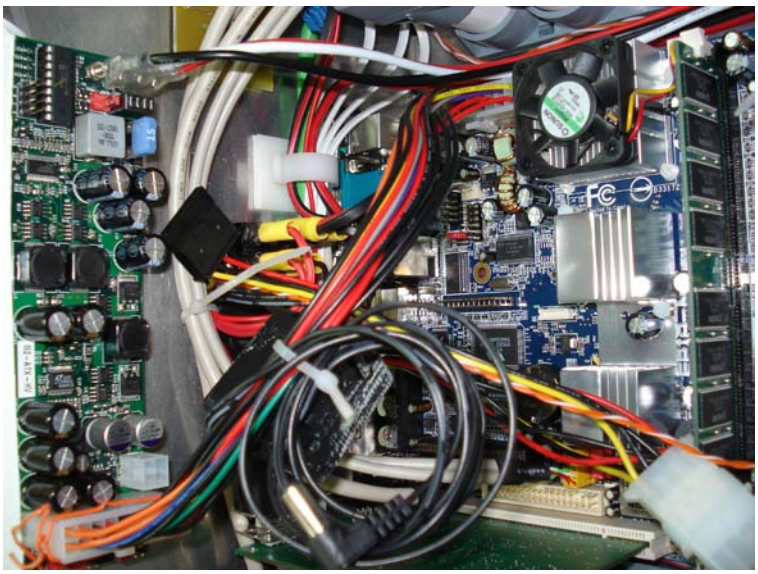


Figura 2.23: Fonte do PC e PC

➢(11) (12) (13)→PIC_Base, Placa de infravermelhos e Placa de ultra-sons

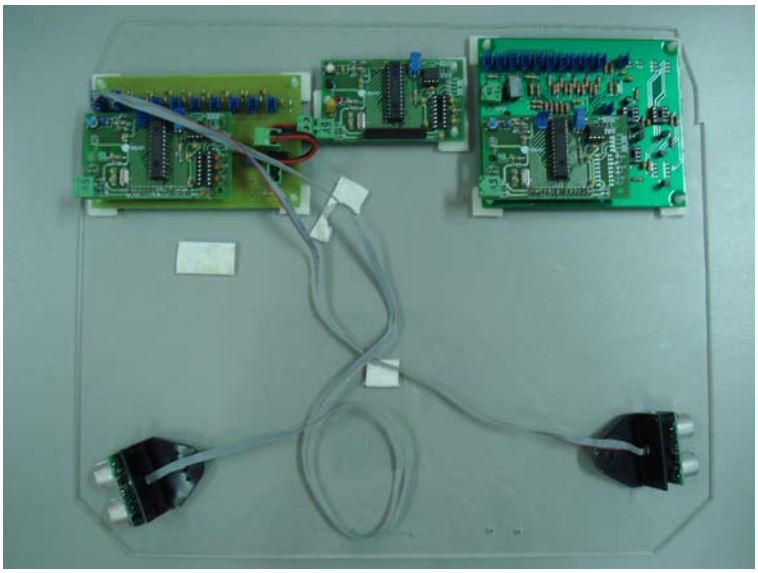


Figura 2.24: Placa dos infravermelhos e dos ultra-sons com as PIC_Base

➢(14)(15) →Placa de interface com o módulo de potência do motor

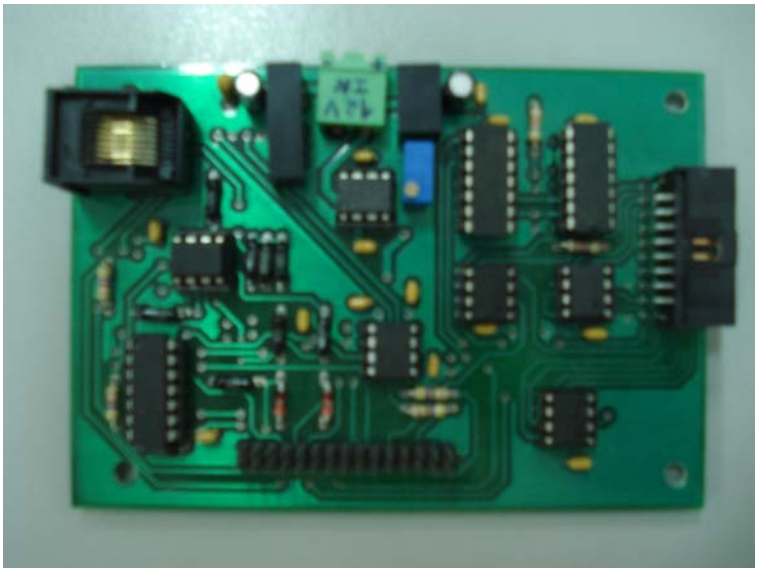


Figura 2.25: Placa de interface com o módulo de potência do motor

- (16) → Módulo de potência e motor

Figura 2.26: Módulo de potência e motor

2.5 Recomendações de Uso

O sistema de alimentação é feito com o recurso a duas baterias de 12V, de forma ao sistema ser autónomo. No entanto em fases de testes é preferível usar um alimentador externo de 24V para não descarregar as baterias sempre que possível. Não é possível estar a carregar as baterias e a usá-las ao mesmo tempo, uma vez que estas são carregadas de forma independente e portanto não podem estar ligadas em série quando estão em carga. Para efectuar a recarga das baterias é usada uma fonte de laboratório com regulação da corrente máxima. Carga normal é feita com 15V e 3A, se for necessária uma carga rápida pode usar 15.5V e 4A sendo necessária uma fonte para cada bateria ou uma fonte dupla.

Existe um interruptor central no painel traseiro de três posições que serve para seleccionar o tipo de uso que se quer fazer das baterias. Posição do meio as baterias estão desligadas, para baixo permite a sua carga e para cima liga as baterias em série para alimentar o sistema. É possível estar a trabalhar com as baterias e ligar o carregador externo e vice-versa, sendo que ao desligar o carregador externo deve ser feito desligando o conector ao robô em primeiro lugar, porque se desligam o carregador

da tomada o sistema de comutação não reage em tempo útil e o computador vai abaixo o que não é recomendável.

Em fase de testes no chão é recomendável colocar o controlo do *enable* do conversor na posição de controlo externo por comando de rádio frequência para poder desligar o sistema de tracção em caso de necessidade. Para tal existe um interruptor perto do sistema de sinalização do sistema composto por vários leds do lado direito do painel traseiro. Na posição central o sistema de controlo está desligado, para baixo está ligado e para cima é controlado pelo controlo remoto. Em caso de emergência ou para desligar para além do sistema de controlo desligar também os módulos de potência dos motores é pressionar o botão de emergência vermelho.

2.6 Colocação em funcionamento/diagnóstico do sistema

Para colocação do sistema em funcionamento deve-se proceder à ligação da alimentação de todo o sistema (ligando para tal os três interruptores existentes nas posições de acordo com o pretendido). Em condições normais o sistema distribuído está neste momento operacional, sendo possível verificar esse funcionamento fazendo a leitura simples do barramento de CAN, tanto na porta can0 como na can1 do PC. E necessário no entanto carregar os módulos de CAN e RTAI após o arranque do sistema operativo linux de tempo real usando o script "./initAll.sh" que se encontra em "cd /root".

O sistema de CAN funciona a 250 kbits/s, como essa é a baud-rate por defeito da placa de CAN do PC, não é necessário reconfigurá-la. Basta portanto executar o programa "./receive can0" ou "./receive can1" para que surja no ecrã várias mensagens, uma por cada pic presente no sistema. Existem pelo menos três mensagens (*motion and sincMCU*) com id's de mensagem diferentes e que se repetem ciclicamente. Caso isso não ocorra significa que algo não está correcto.

Na possibilidade de por algum motivo haver a omissão desta mensagem inicial do pic, que corresponde ao seu estado de boot-loader, como é explicado no capítulo 3.1.2 mais em pormenor, tal pode significar em último caso a necessidade de substituir/reprogramar o boot-loader do pic. A programação do boot-loader do pic, é feita recorrendo ao programador da microchip ICD2 e o interface gráfico MPLab, fazendo a importação do ficheiro ".hex" correspondente ao BootFirmware do pic pretendido, que se encontra na pasta referente ao *firmware*.dos pic's. Não fazer

confusão com o programa principal, o *firmware* serve apenas de identificação do pic, para posterior programação do código principal via CAN, existindo portanto dois ficheiros ".hex" distintos para cada pic.

Outra situação possível de ocorrer é o código que programamos via CAN (código principal), necessitar de ser reprogramado novamente, esta situação é facilmente detectável da seguinte forma, caso após a ordem via CAN de mudar o estado de "boot" para "run" do pic, este continuar a apresentar a mensagem inicial de boot.

Para programação via CAN e dar ordem ao pic para mudar o seu estado de "boot" para "run", tal é explicado no capítulo 3.1.2.

Estando tudo operacional no barramento de CAN inicia-se o programa de RTAI, para tal entra-se numa consola (user: root, passwd: rob07) e na directoria *"cd /home/isrobot/exec"* está o programa de leitura de CAN e o programa de RTAI.

O código fonte encontra-se na pasta *"cd /home/isrobot/RTisrobot"*, para compilar execute *"./autogen.sh;make"*, sendo o executável gerado com o nome de "rtisrobot". Para correr o programa basta o comando *"./rtisrobot"*.

Capítulo 3

3 Arquitectura Distribuída

A Arquitectura Distribuída é composta por vários componentes que visam a sua modularidade e versatilidade, levando no entanto à necessidade de obedecer a algumas regras para que haja determinismo e coerência na informação recebida e transmitida por cada módulo do sistema, sendo para tal a arquitectura baseada num sistema de tempo real.

O sistema global é composto por três camadas, presentes na Figura 3.1, que são: comunicação de baixo nível, camada de controlo de baixo nível e no topo a camada responsável pela navegação e percepção do meio em seu redor.

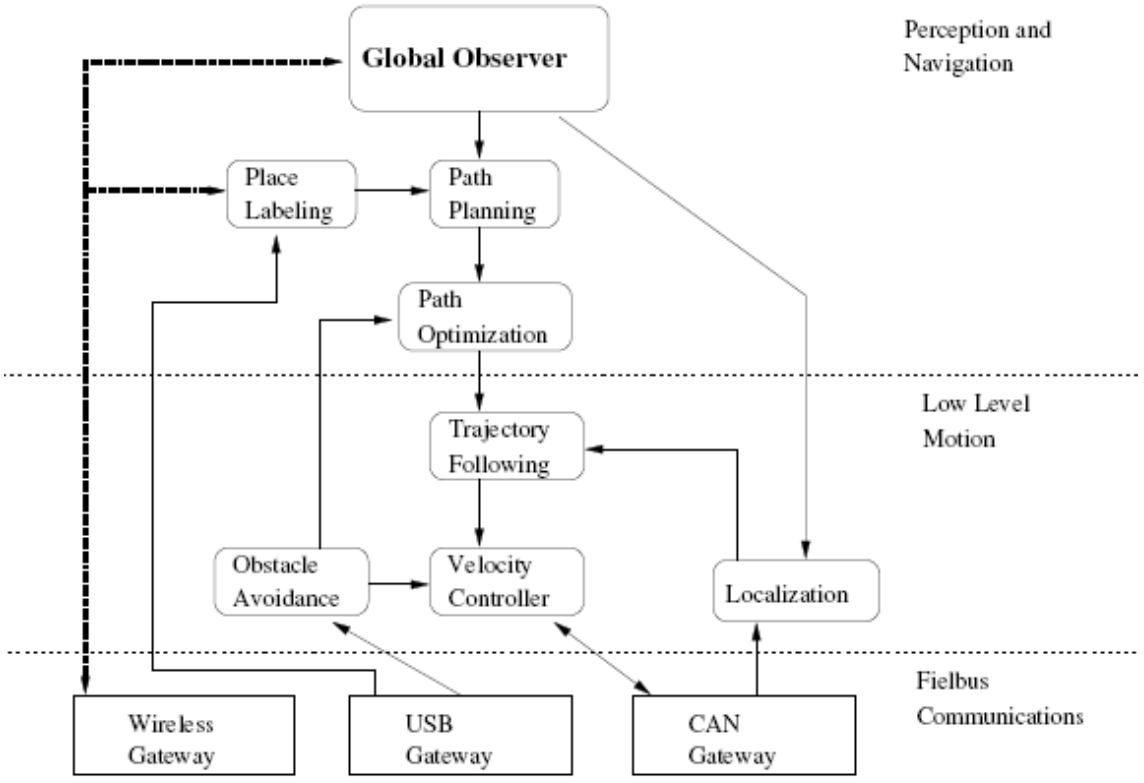


Figura 3.1: Arquitectura de Software

Dada a modularidade da arquitectura tanto a nível de software como de hardware, existe portanto diversos módulos. Cada módulo é composto por hardware projectado para

determinada função e um microcontrolador para permitir a interacção com todo o sistema via CAN e o hardware que tem a seu cargo. A arquitectura de hardware é ilustrada na Figura 3.2, onde é perceptível os diversos componentes de hardware que a compõem.

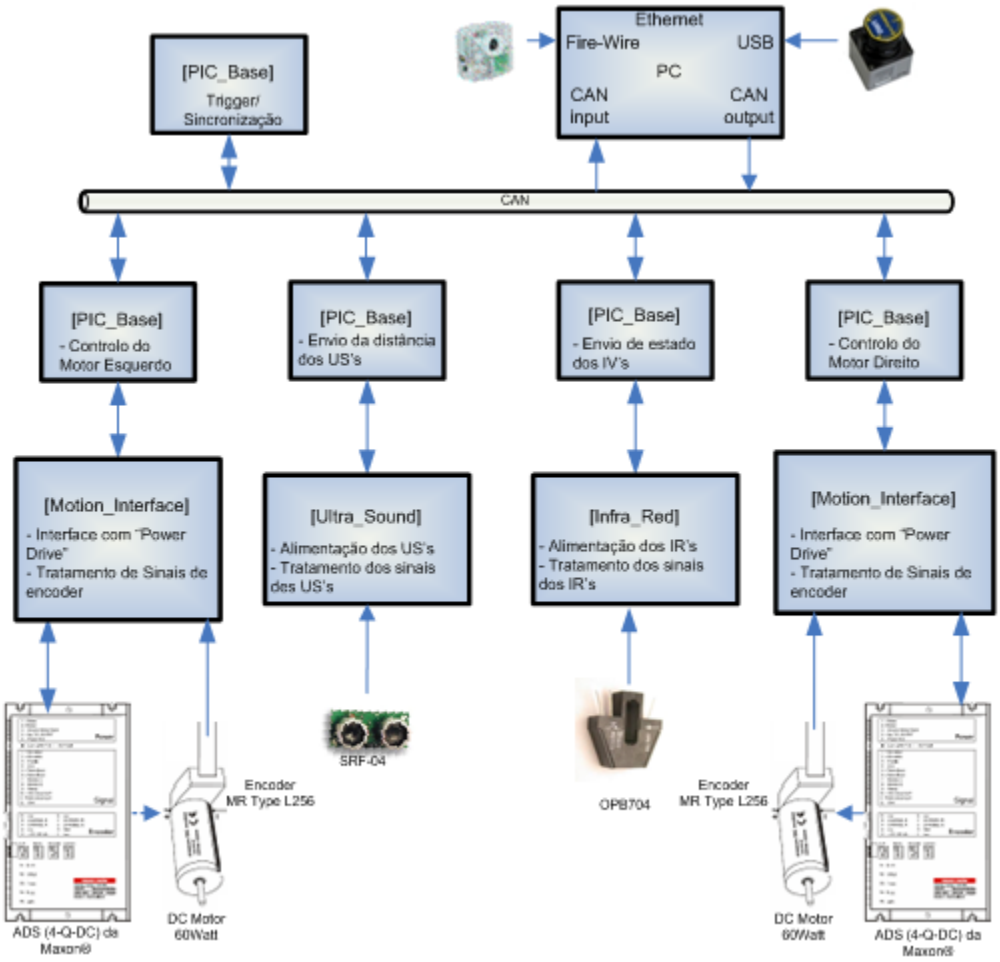


Figura 3.2: Estrutura de hardware de aquisição, controlo e actuação do ISRobot

O Programa utilizado para desenvolver o código para os microcontroladores é o HITECH, uma vez que este trabalha em Linux, ao contrário do MPLab®, utilizando-se a linguagem C, como linguagem de programação. Na camada superior de tempo real presente no PC é usada a linguagem C++, sendo cada módulo uma classe de forma a ter o sistema modular.

3.1 Módulo de Processamento "PIC_Base"

O módulo de processamento "PicBase", é composto pelos componentes necessários ao funcionamento do microcontrolador, à comunicação no barramento de CAN e à

expansão das suas capacidades através do interface com o módulo de hardware para o qual está programado para gerir.

O uso dos microcontroladores traz muitas vantagens para este tipo de sistemas onde existe a necessidade de constante evolução, permitindo flexibilidade. Estes oferecem inúmeras capacidades tais como de tratamento, análise e comunicação de dados/sinais, e uma panóplia de periféricos integrados que evita assim muita electrónica externa. Este facto confere mais flexibilidade às aplicações que integram, recaindo sobre este tipo de dispositivo a escolha para o desenvolvimento dos vários componentes que constituem o sistema.

3.1.1 Características Gerais do PIC Utilizado

Os microcontroladores utilizados neste sistema são da família PIC18FXX8, sendo o modelo utilizado neste trabalho o PIC18F258, o qual apresenta já bastantes periféricos integrados, como comunicação via RS232, CAN, SPI, ou mesmo ADC e geradores de PWM, (os periféricos utilizados serão apresentados mais à frente). Pode ver-se na Figura 3.3 o diagrama de pinos do Microcontrolador:

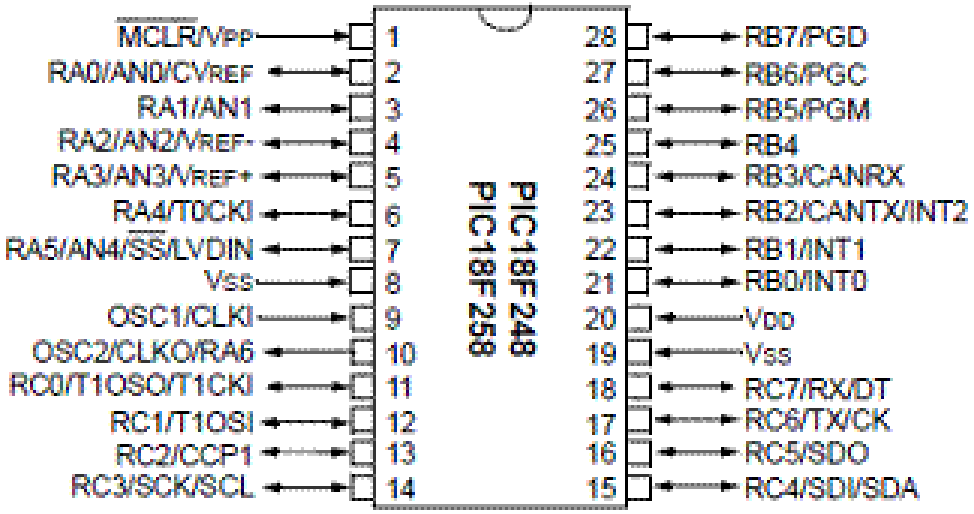


Figura 3.3: "pin diagram" do PIC18F2x8

Como é perceptível pelo "*pin diagram*", este microcontrolador apresenta 3 portos de I/O (Porto A, B e C), sendo que estes se encontram quase na sua totalidade multiplexados com outras funções associadas aos vários periféricos presentes no microcontrolador.

Regra geral quando um determinado periférico está activo os pinos associados a este não podem ser utilizados como pinos genéricos de I/O.

3.1.2 Módulo de Hardware "*PIC_Base*"

A "*PIC_Base*" é um módulo de hardware, utilizado para fazer a interface entre os PIC's e os vários módulos de hardware desenvolvidos. Este módulo de interface tem várias capacidades. Destaca-se, a possibilidade de programação do PIC através do programador disponível da Microchip(R), o MPLab ICD 2(R) sendo este utilizado na fase inicial para programação do *boot-firmware* de cada PIC, que incorpora a sua identificação única no sistema. Para além da possibilidade de programação, o módulo dispõe ainda de uma ligação fácil a todos os portos por meio de uma ficha de 20 pinos e do hardware periférico necessário para comunicação RS232 e CAN, sendo esta última também explorada para a programação do microcontrolador, recorrendo para tal ao programa "canbootmngr_v2", uma vez já residindo no módulo o seu respectivo *boot-firmware.*

A identificação dos diversos módulos nesta fase é expressa de seguida assim como um exemplo de utilização.

Tabela 3.1: ID's utilizados pelo programa "canbootmngr_v2"

CanBootManager	Group	PIC	Msg Boot
IRNode	1	1	0x21
USNode	1	2	0x22
MotionNode	2	1 (Left) / 2 (Right)	0x41/0x42
ControlNode	4	1	0x81
TriggerNode	6	1	0xC1

O programa "canbootmngr_v2" é usado essencialmente para programação das funcionalidades de cada PIC via CAN, permitindo desta forma programar todos os PICs sem ter de conectá-los um a um ao programador da Microchip. Para além de programar é possível transmitir uma ordem aos PIC´s para saírem do estado *"Boot Loader Mode"* e passarem para *"Idle Mode"* através do programa "canbootmngr_v2". O comando completo para programar um determinado PIC com o ficheiro ".hex" criado pelo

compilador, neste caso o TriggerNode é o seguinte: "./canbootmngr_v2 -g 6 -p 1 -d -f TriggerNode.hex". A interpretação do comando significa fazer o *download* do código para o PIC com a identificação expressa.

Os microcontroladores presentes em cada módulo possuem três modos de operação, sendo dois deles já enumerados anteriormente, nomeadamente *"Boot Loader Mode"*, *"Idle Mode"* e também *"Running Mode*". O *"Boot Loader Mode"* é o primeiro estado em que este fica quando é alimentado, permitindo a sua reprogramação de forma simples e acessível sem ter de mexer no hardware. Passando para *"Idle Mode"* este fica num estado de espera, em que o seu normal funcionamento é suspenso. Enquanto em "*Running Mode*" este efectua o devido controlo do que o rodeia e adquire a devida informação do sistema para posterior envio para o sistema central. Existindo no entanto uma ordem pela qual os diferentes estados podem passar, a qual é expressa no diagrama seguinte.

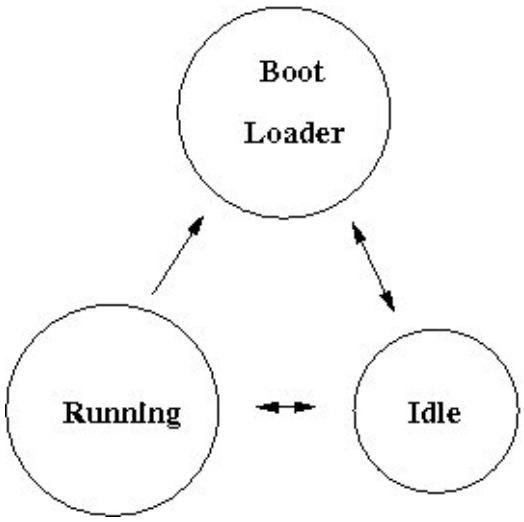


Figura 3.4: Modos de operação

3.1.3 Detalhes da Comunicação CAN

Para que os vários dispositivos troquem a informação correctamente sob o protocolo CAN, existe a necessidade da definição da estrutura das mensagens. Sendo a estrutura global de uma mensagem segundo o protocolo CAN como se retrata na Figura 3.5, definiu-se a estrutura específica das mensagens, a qual é adequada às necessidades existentes neste sistema.

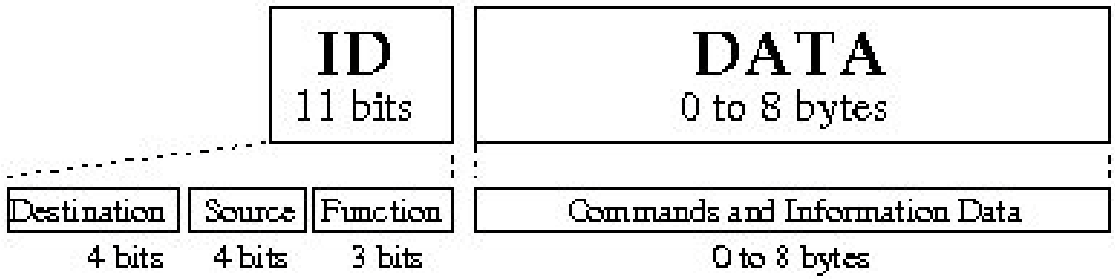


Figura 3.5: Protocolo CAN utilizado

Como pode ser observado, o ID da mensagem é único uma vez que é composto pelo destino, origem e a função pretendida para o módulo de destino. Em termos de ID dos módulos estes estão descritos na tabela seguinte, no caso da função esta é dependente do módulo em questão sendo portanto apresentada na secção do módulo a que respeita, assim como o campo de dados, cujo tamanho também ira depender dessa mesma função.

Tabela 3.2: ID's de *"Destination/Source"* usados no protocolo CAN

ID		Módulo
0		PC
1		*UltraSound*
2		*InfraRed*
4		*Right PDriveEncoder*
5		*Left PDriveEncoder*
6		*Both PDriveEncoder*
15		*syncMCU*

Como se poderá verificar mais à frente, existem duas funções comuns a todos os módulos, que são nomeadamente o "*Turn Node OFF → Idle Mode*" e "*Turn Node ON → Running Mode*", as quais são essenciais, correspondendo o ID "0" e "1" respectivamente.

Para além da identificação de cada módulo no barramento de CAN, para que tenhamos um sistema fiável e determinista é necessário que todas ou a sua maioria obedeça a determinados instantes de tempo na comunicação dos dados e operações de controlo.

Existe portanto uma sequência temporal das várias acções a serem executadas no sistema, como é ilustrado na figura seguinte.

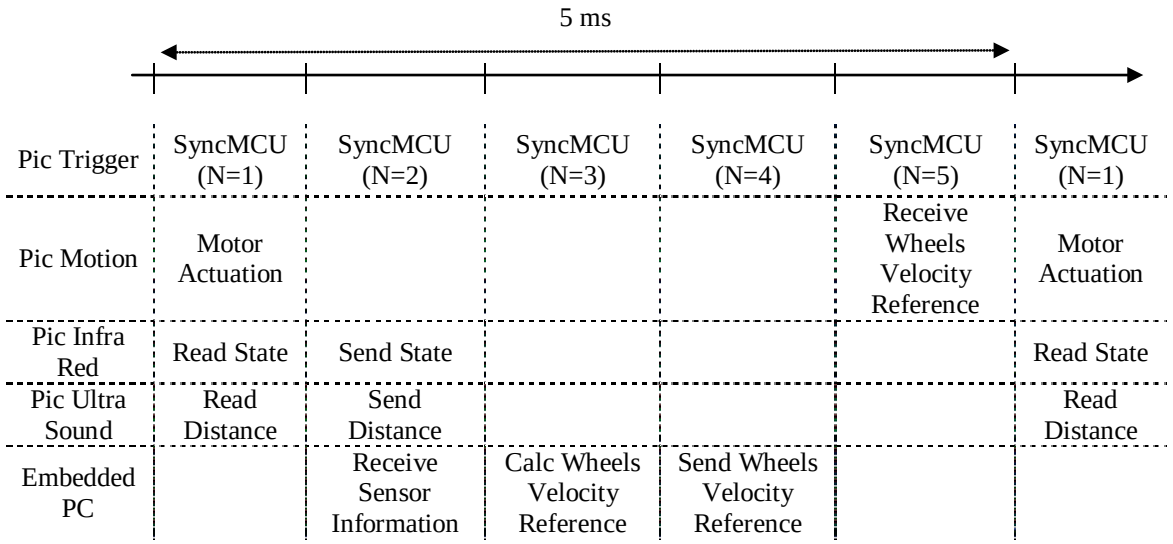


Figura 3.6: Diagrama temporal de acções

*3.2 **Módulo de Software/Hardware** "Trigger"*

3.2.1 Software

As funções de CAN para este módulo, são as identificadas na Tabela 3.3.

Tabela 3.3: ID's das funções presentes no módulo Trigger

ID		Função
0		*Turn Node OFF*
1		*Turn Node ON*
15		*Synchronize Nodes*

A função "Synchronize All Nodes", é usada para sincronizar a malha de controlo do sistema. Sendo o tempo do ciclo de controlo de 5ms, é enviada uma mensagem de sincronismo a cada 1ms, com a indicação da fase do ciclo em que o sistema se encontra, para que, sejam efectuadas todas as acções dos diversos módulos de uma forma cíclica e sincronizada. Tal mensagem é elaborada da seguinte forma:

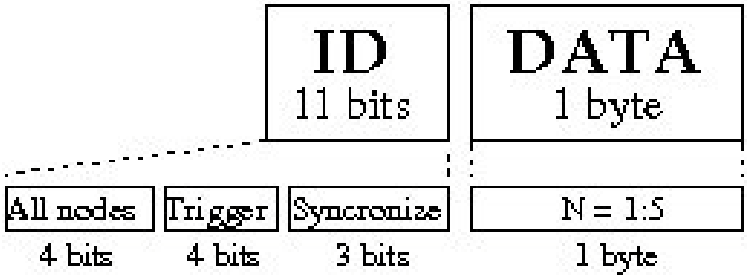


Figura 3.7: Mensagem de sincronização do sistema

Módulo do Trigger

Como foi possível ver nos outros módulos, estes efectuam as suas acções com base numa determinada ordem e respeitando determinado ciclo, sendo este módulo o responsável por essa sincronização, a qual é feita de 1 em 1ms, sendo o ciclo total de 5ms como referido anteriormente.

Visto isto, apresenta-se de seguida o código que permite esta sincronização dos módulos.

```
void interrupt HighPriorityInterrupts(void){
  if(TMR3IF)
    {
      TMR3IF = 0;
      // WriteTimer3( 55535 );  55535d = D8EFh  // Resets Cycle Timer (1ms/40Mhz)
      TMR3H = 0xD8;  // Write high byte to Timer3
      TMR3L = 0xEF;  // Write low byte to Timer3

      if (Cycle > 4)
      {
         Cycle = 0;
      }
      Cycle++;

      if (TriggerState==ON)
      {
        CANMsgOut.stdID   = Sincronize;
        CANMsgOut.len     = 1;
        CANMsgOut.data[0] = Cycle;
        sendCanMessage(CANMsgOut);             // Send Sincronization message to CAN BUS

        if (StateChanged == TRUE ){
          CANMsgOut.stdID = 0;                 // Informs robcan that
          CANMsgOut.stdID = TRIGGER << 3;      // trigger node is now on
          CANMsgOut.stdID += 1;
          CANMsgOut.len = 1;
          sendCanMessage(CANMsgOut);
          StateChanged = FALSE;
        }
      }
      else
      {
        if (StateChanged == TRUE ){
          CANMsgOut.stdID = 0;                // Informs robcan that
          CANMsgOut.stdID = TRIGGER << 3;     // trigger node is now off
          CANMsgOut.len = 1;
          sendCanMessage(CANMsgOut);
          StateChanged = FALSE;
        }
      }
    }
```

Este módulo envia uma mensagem de CAN de forma periódica, baseado num temporizador interno que conta 1ms, quando o módulo se encontra no estado activo. Na mensagem de sincronismo temos um byte que determina a fase do ciclo de operações. Temos cinco fases distintas de operações correspondentes a diferentes tarefas dos módulos, perfazendo o ciclo de 5ms e que se repete de forma cíclica.

3.2.2 Hardware

Este módulo é essencialmente apenas o módulo de processamento PIC_Base, uma vez que este não necessita de mais nenhum hardware de interface, utilizando apenas a comunicação de CAN do módulo, para efectuar a sincronização (tipo maestro numa banda) de todo o sistema e sinalizando todas as acções dos outros módulos no devido e exacto momento em que tal deve ser efectuado, como foi descrito anteriormente.

Módulo de Hardware "PIC_Base"

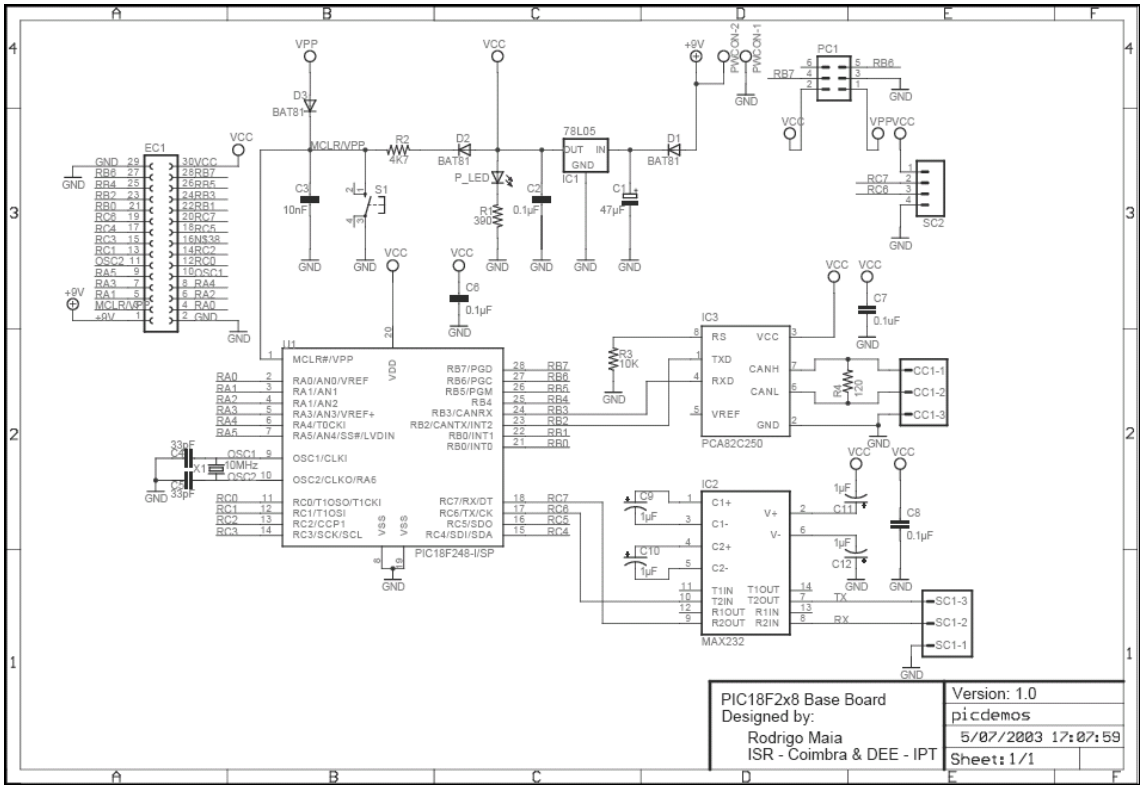


Figura 3.8: Esquemático do módulo

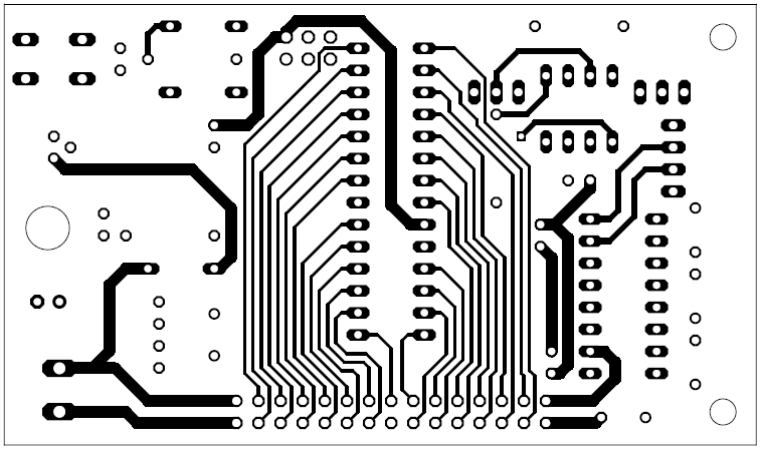


Figura 3.9: Layout do PCB – Top Layer

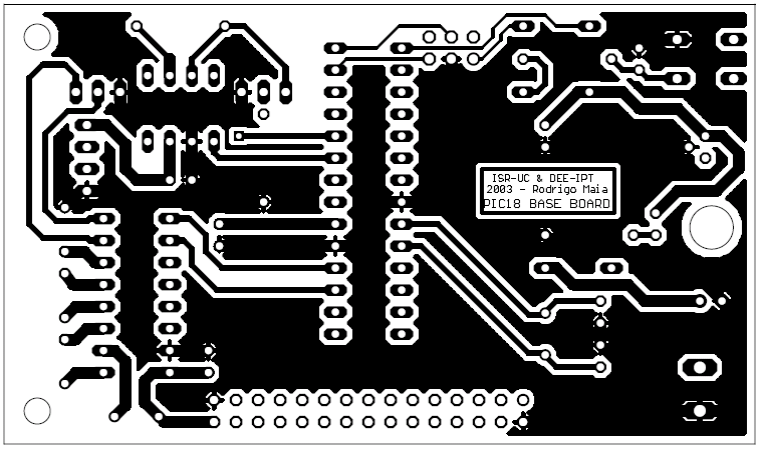


Figura 3.10: Layout do PCB – Bottom Layer

*3.3 **Módulo de Software/Hardware** "Motion_Interface"*

3.3.1 Software

As funções de CAN para este módulo, são as identificadas na Tabela 3.4.

Tabela 3.4: ID's das funções presentes no módulo *Motion*

ID		Função
0		*Turn Node OFF*
1		*Turn Node ON*
2		*Set DAC Command*
4		*Data from Motion*

A função "*Set DAC Command*", é usada para definir o valor a transmitir ao DAC que por sua vez irá corresponder a uma velocidade do motor, sendo esse valor composto por 2 bytes (DAC de 10 bits de resolução).

Os valores monitorizados pelo módulo totalizam oito bytes, em que quatro bytes correspondem ao valor acumulado dos pulsos do encoder, seguido de mais dois bytes referentes à velocidade do motor e por fim mais dois bytes correspondendo ao valor de comando recebido pelo módulo. Estes são os dados transmitidos sobre a função "*Data from Motion*".

O fluxograma presente na Figura 3.11 mostra resumidamente a estrutura do código desenvolvido para o PIC responsável pelo envio de comandos para o motor, o qual irá corresponder a uma determinada velocidade.

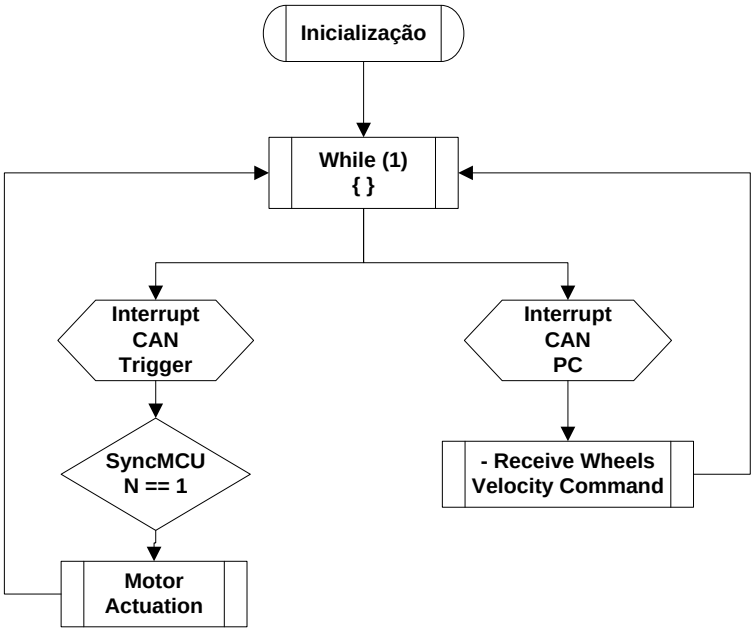


Figura 3.11: Fluxograma do código implementado no *Motion Node*

Como é constatado pelo fluxograma presente na Figura 3.11, o *Motion Node* recebe pelo barramento de CAN da camada de alto nível, comandos para controlo da velocidade do motor. Esse comando é concretizado em acção sobre o motor quando o *Motion* recebe uma mensagem de sincronismo sinalizando o instante N=1 do ciclo.

3.3.2 Hardware

Dadas as necessidades que se apresentavam ao tratamento dos sinais, o módulo *Motion_Interface* foi desenvolvido de modo a servir de interface entre o *driver* de potência (Servoamplifier ADS) e o módulo "*PIC_Base*", que faz o interface com o PIC responsável pelo controlo de velocidade de cada um dos motores.

As funções principais deste módulo são então:

1. Tratar o sinal de comando enviado pelo PIC segundo o protocolo SPI, de modo a transformar posteriormente num sinal analógico de -10V a 10V com recurso a um DAC, para colocar na entrada de comando do "drive de potência". É ainda enviado para o PIC um sinal analógico de 0V a 5V, resultado da conversão D/A, que serve de auto-regulação do sistema de controlo.

2. Tratar os sinais enviados pelo encoder (3 canais em modo diferencial, Canal A, Canal B e Canal Index), de modo a enviar para o PIC dois sinais distintos:

2.1. Count-Up e Count-Down, a partir do Canal A e Canal B que se encontram em quadratura.

2.2. Sinal de contagem de volta a partir do Canal Index.

3. Tratar os sinais de informação de corrente e velocidade disponíveis no drive de potência, que se trata de um sinal analógico que varia entre -10V e 10V, de modo a enviar para o PIC um sinal analógico de 0 a 5V.

4. Tratar o sinal de Status, disponível no drive de potência de potência de modo a enviar para o módulo PIC_Base.

5. Tratar o sinal de Enable a enviar da PIC_Base para o drive de potência para colocar o drive ON/OFF, remotamente.

Tratamento do Sinal de Comando

O sinal de comando (velocidade pretendida) é enviado pelo PC via CAN. Depois de recebido e tratado é enviado usando o protocolo SPI. Existe então a necessidade da utilização de um conversor Digital/Analógico (DAC) para a conversão desse sinal digital num sinal analógico. Utilizou-se o DAC MCP4921 (*12-bit voltage output digital-to-analog converter*). Este coloca na saída um sinal analógico VDAC, de 0V a 5V.

Este sinal tem ainda de ser tratado antes de ser enviado para o módulo de potência, uma vez que o sinal de comando a enviar tem de ser um sinal analógico de -10V a 10V. Com o objectivo de ter essa gama de valores com precisão e de igual forma em ambos os módulos que controlam cada motor, recorreu-se a uma montagem amplificadora com ganho variável para ajuste fino desse ganho.

Esse auto-ajuste do ganho do amplificador é realizado recorrendo a potenciómetros digitais, representados na Figura 3.12. P_1 e P_3 são dotados de comunicação SPI, e os seus valores são controlados pelos PIC's associados a cada um dos módulos, em função da comparação do sinal de comando à saída dos módulos com um dado sinal de referência.
A montagem escolhida é o amplificador de diferença porque apresenta uma equação simples num caso particular. Esta é apresentada na Figura 3.12, e o seu ganho é dado

pelas eq. 3.1 e 3.2. A primeira é a equação genérica e a segunda um caso particular, aplicado quando se verifica $P_1=P_3$ e $R_2=R_4$.

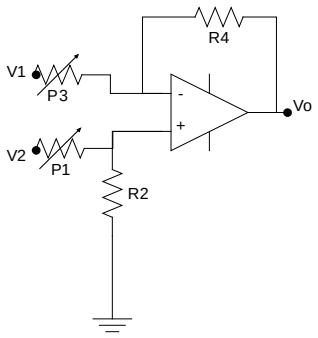


Figura 3.12: Amplificador de diferença

$$v_0 = \frac{\left(1+\frac{R_4}{P_3}\right)}{\left(1+\frac{P_1}{R_2}\right)} \cdot v_1 - \frac{R_4}{P_3} \cdot v_2 \tag{3.1}$$

$$v_0 = \frac{R_4}{P_3} \cdot (v_1 - v_2) \tag{3.2}$$

O circuito usado em simulação é apresentado na Figura 3.13 e o seu resultado pode ser observado na Figura 3.14, em que se verifica que a gama de valores pretendida é obtida. Para valores de saída do DAC entre 0V e 5V corresponde a toda a gama de entrada de referência do *PowerDrive* que é de -10V a 10V, como se verifica pela Figura 3.14 os sinais a cor verde e vermelho respectivamente.

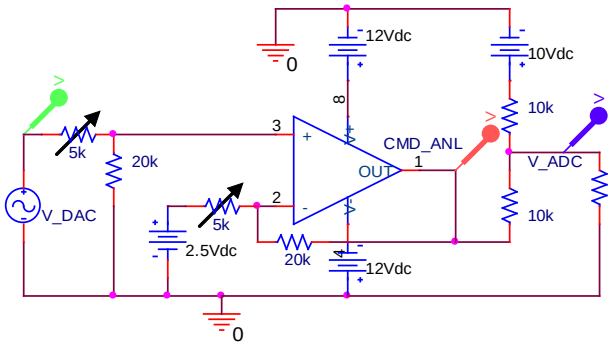


Figura 3.13: Montagem amplificadora de diferença para simulação

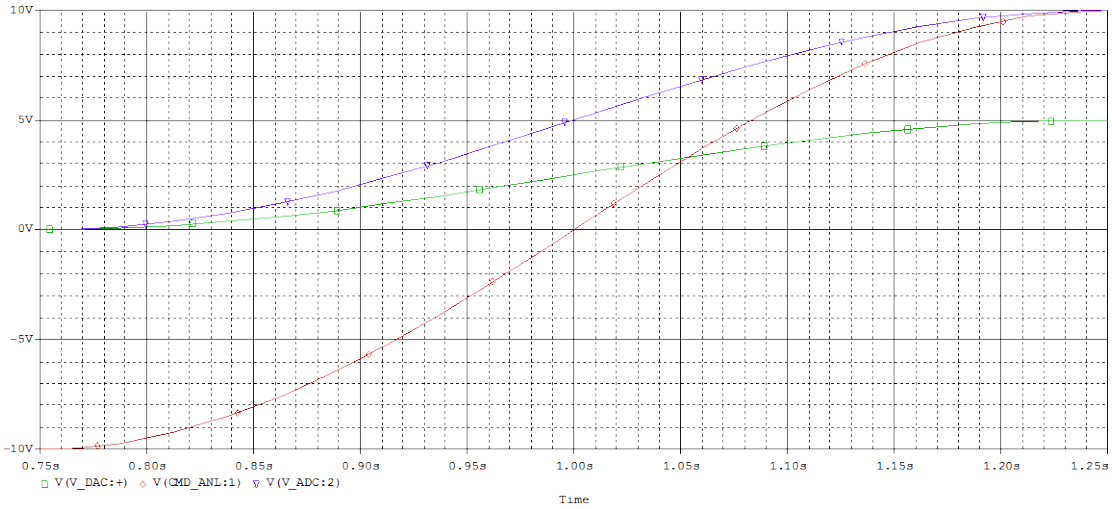


Figura 3.14: Resultado da simulação

Tratamento dos Sinais de Encoder

Como já foi referido o encoder possui 3 canais, Canal A, Canal B e Canal Índex. Cada um desses canais disponibiliza o sinal em modo diferencial, pelo que, existe a necessidade de transformar esse sinal diferencial em TTL. Utiliza-se para o efeito o integrado SN75179B (*differential driver and receiver pair*), como se observa na Figura 3.15.

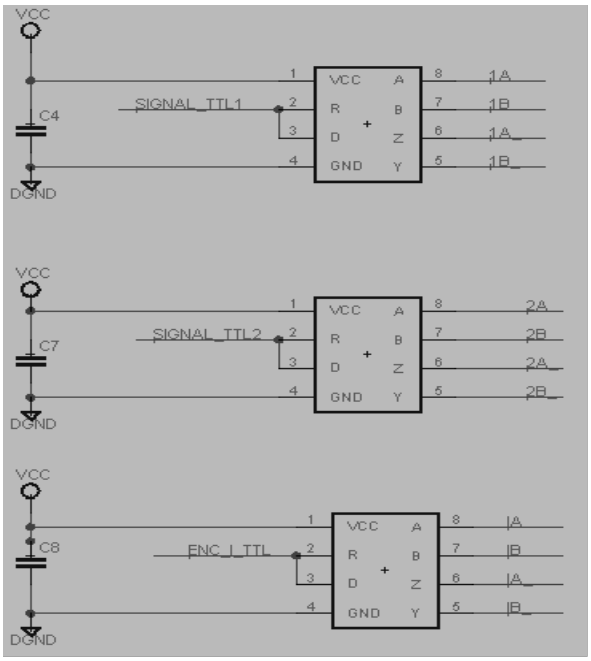


Figura 3.15: Conversão Diferencial/TTL dos sinais vindo do encoder

O sinal TTL obtido a partir do canal Índex é um sinal que é enviado para o PIC. No que respeita aos sinais TTL obtidos dos Canais A e B, estes encontram-se em quadratura,

existindo a necessidade de transformar esses sinais em dois sinais distintos de *Count-UP* para enviar para o PIC. Para tal utiliza-se a montagem presente na Figura 3.16, recorrendo à utilização de dois Flip-Flops do tipo D e duas portas NOT, cujo resultado em simulação pode ser observado na Figura 3.17 e Figura 3.18.

O objectivo é identificar se é o canal A ou o canal B que está em avanço em relação ao outro, obtendo-se portanto os sinais de *Count-UP* e *Count-Down*, respectivamente.

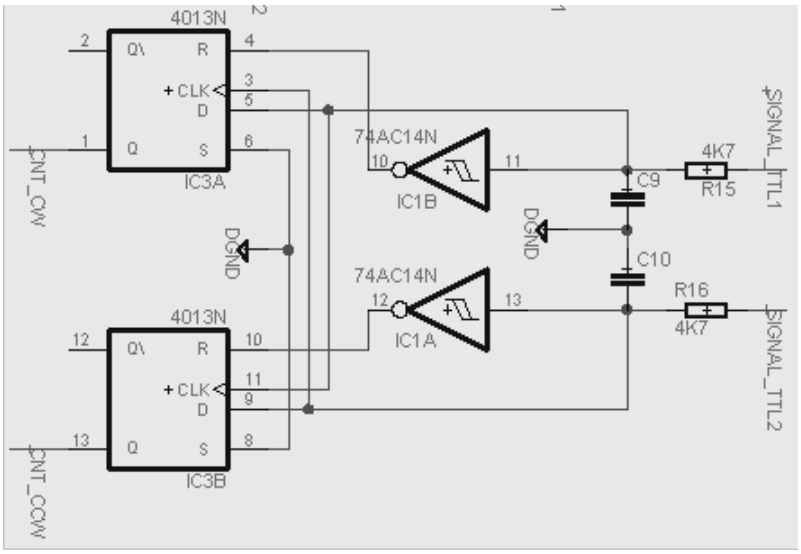


Figura 3.16: Detecção do sentido de rotação

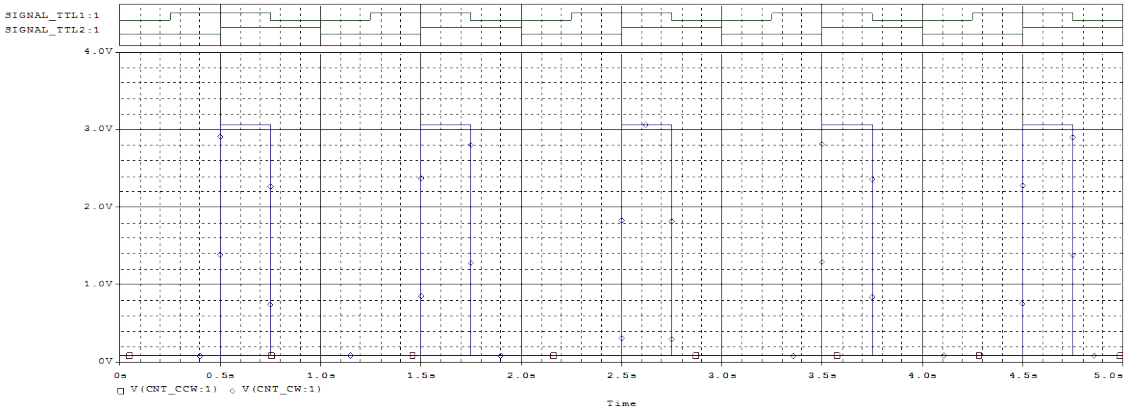


Figura 3.17: Detecção de *Count-Up*

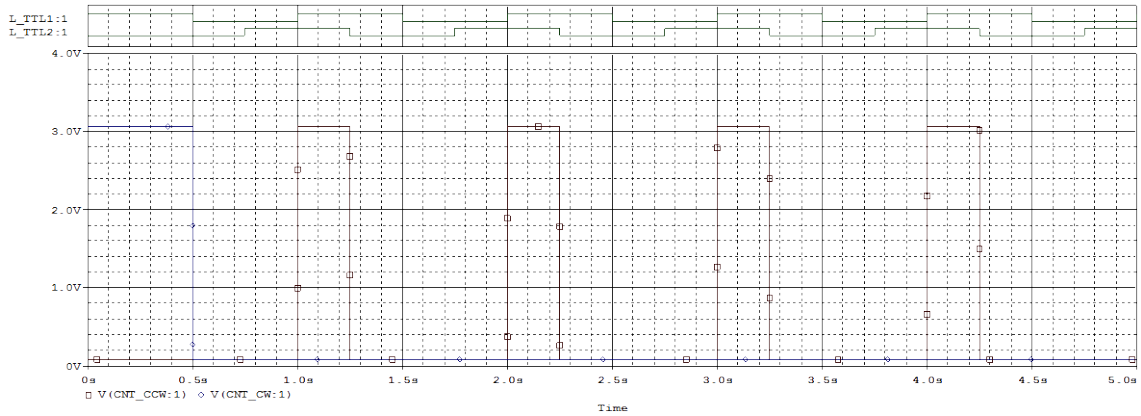


Figura 3.18: Detecção de *Count-Down*

Módulo de Hardware "Motion_Interface"

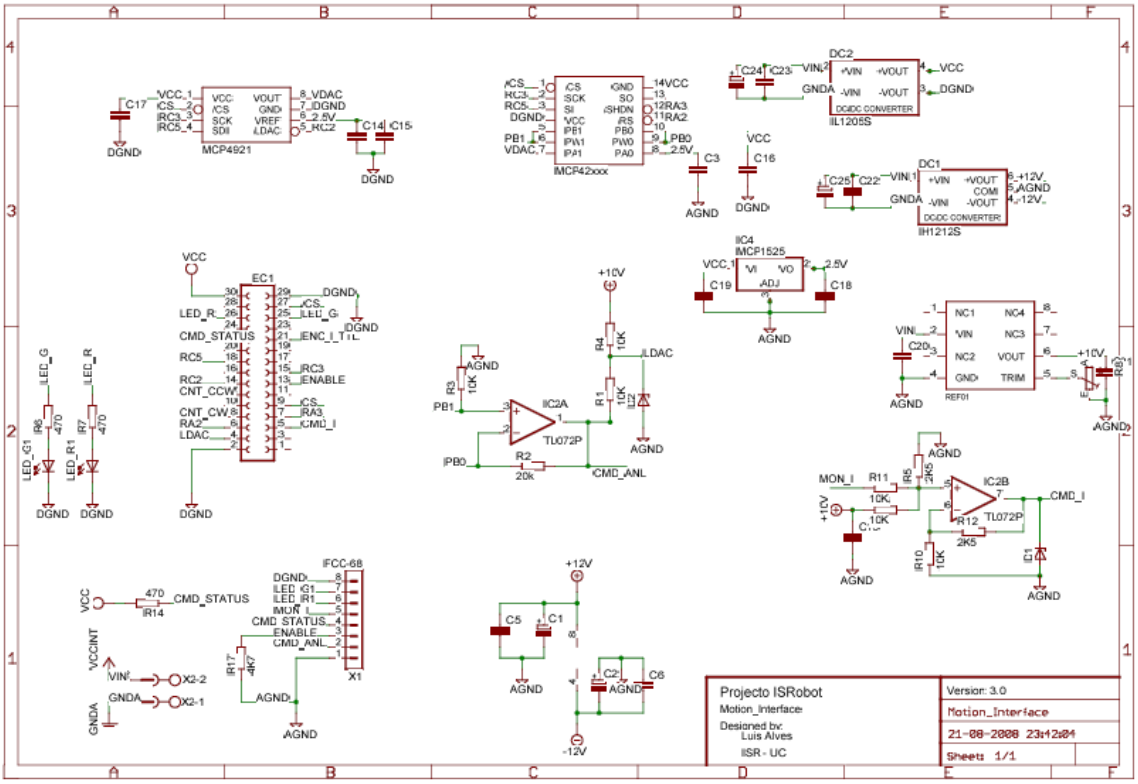


Figura 3.19: Esquemático do módulo *Motion Interface (1/2)*

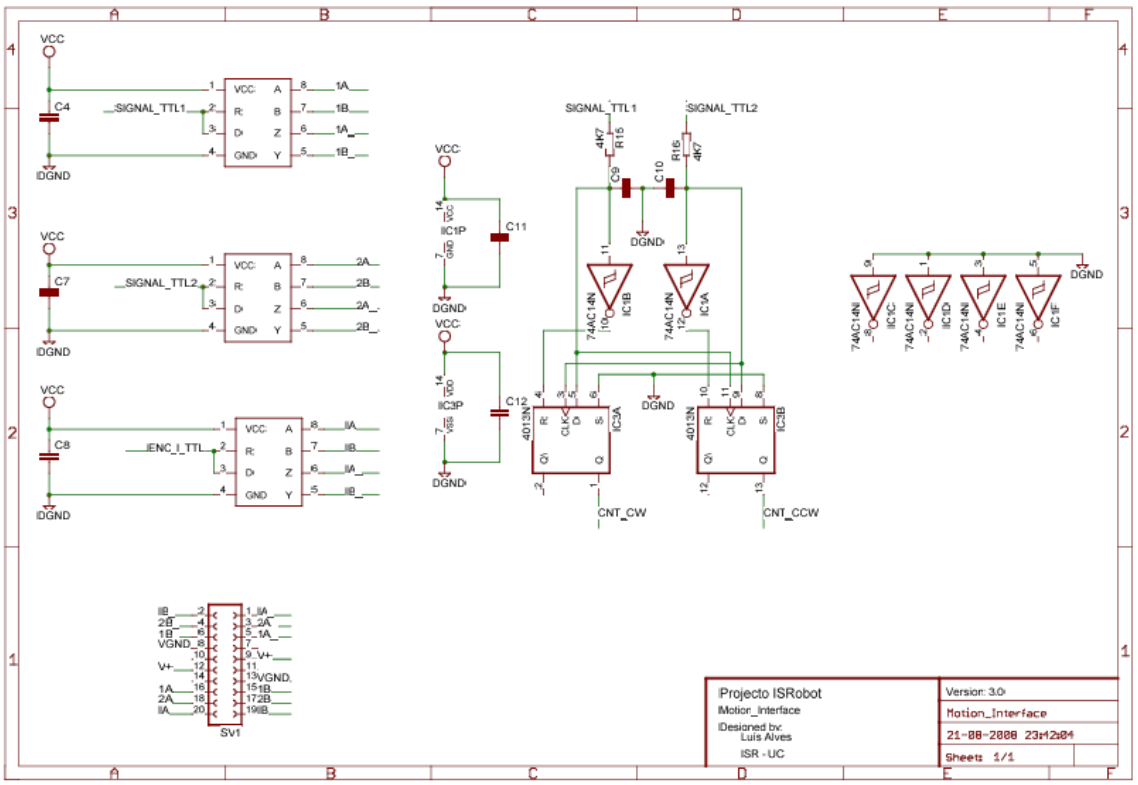


Figura 3.20: Esquemático do módulo *Motion Interface (2/2)*

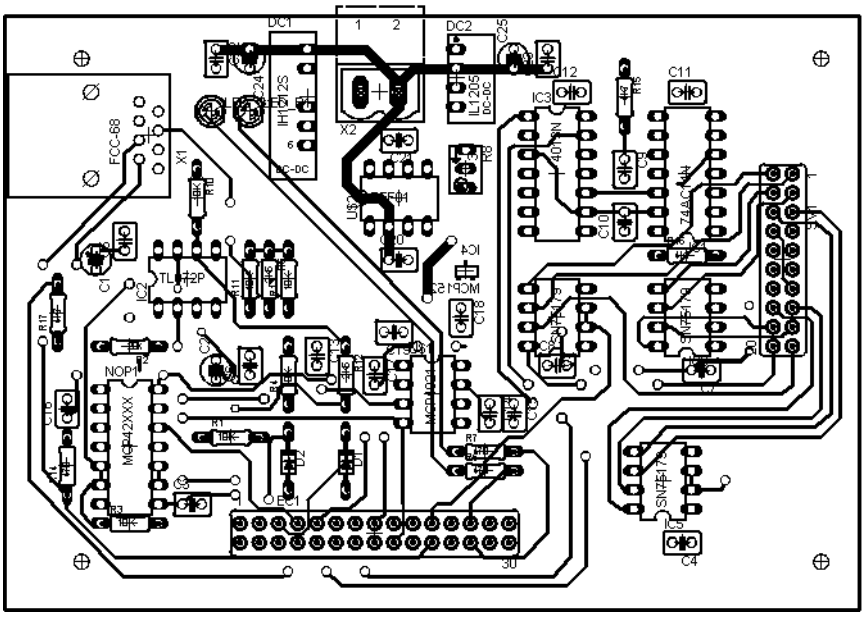


Figura 3.21: Layout do PCB *Motion Interface* (top layer)

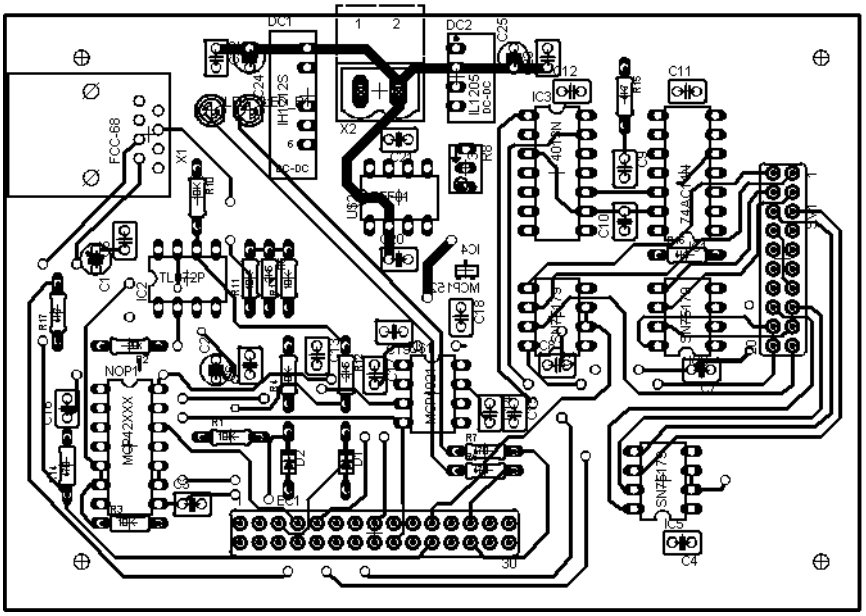


Figura 3.22: Layout do PCB *Motion Interface* (bottom layer)

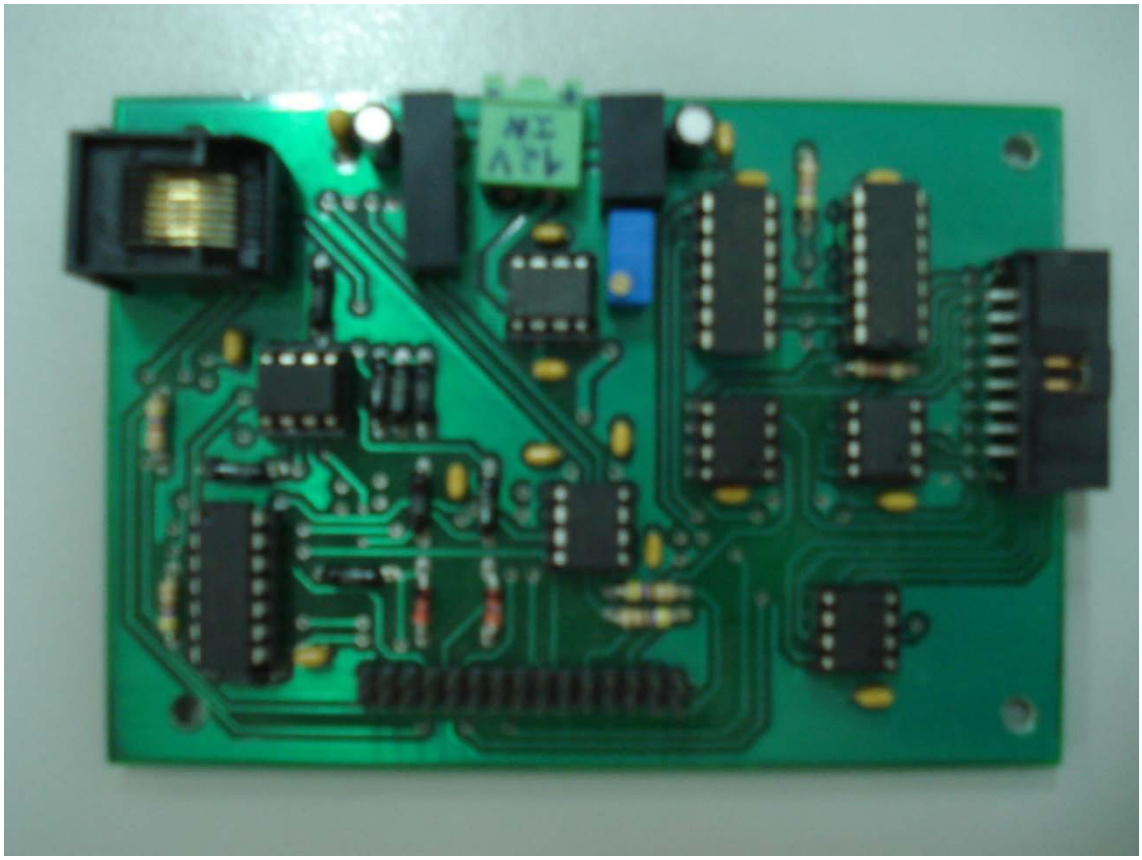


Figura 3.23: PCB do módulo *Motion Interface*

*3.4 **Módulo de Software/Hardware** "Ultra_Sound_Interface"*

3.4.1 Software

As funções de CAN para este módulo, são as identificadas na Tabela 3.5.

Tabela 3.5: ID's das funções presentes no módulo *Ultra Sound*

ID		Função
0		*Turn Node OFF*
1		*Turn Node ON*
4		*Data from US*

Os valores monitorizados pelo módulo totalizam no máximo oito bytes, em que cada byte corresponde ao valor medido por cada ultra-som, sendo estes os dados transmitidos sobre a função "*Data from US*". É possível monitorizar oito ultra-sons no máximo, sendo que se estiverem apenas dois ligados, apenas dois bytes de dados serão transmitidos.

O fluxograma presente na Figura 3.24 mostra resumidamente a estrutura do código desenvolvido para o PIC responsável pelo envio do estado dos infra-vermelhos.

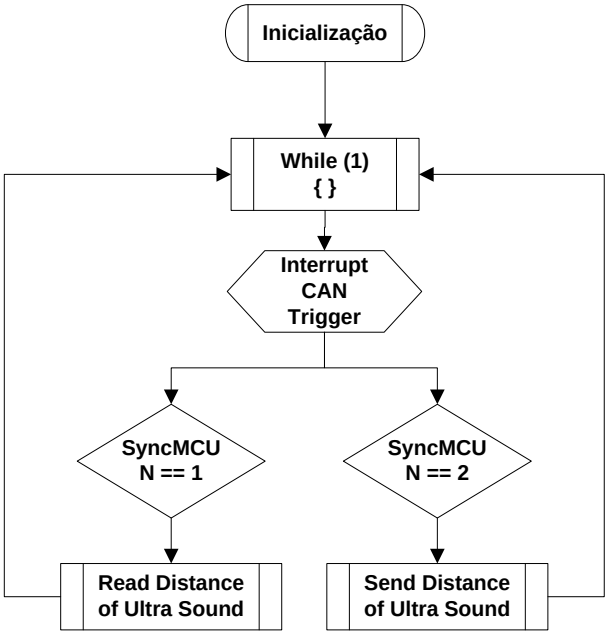


Figura 3.24: Fluxograma do código implementado no Ultra Sound *Node*

Como é constatado pelo fluxograma presente na Figura 3.24, o *Ultra Sound Node* envia pelo barramento de CAN para a camada de alto nível, o valor de distância medido por cada ultra-som de acordo com as mensagens de sincronismo.

3.4.2 Hardware

Este módulo de hardware tem como objectivo fazer a interface dos sinais que são enviados e recebidos entre o PIC responsável por monitorizar os ultra-sons e os próprios, não possuindo quaisquer outras tarefas adicionais, tornando-se como tal um circuito bastante simples. Este módulo encontra-se preparado para poder ligar até um máximo de oito ultra-sons.

A escolha dos ultra-sons a utilizar no projecto recaiu sobre o modelo SRF-04 cujas características se adaptam bem às necessidades do projecto. As características principais do SRF-04, encontram-se descritas na Tabela 3.6.

Tabela 3.6: Características dos ultra-sons

Sinal	Unidades S.I.	Valor/Condição de funcionamento
Tensão	V	5
Corrente	mA	30 Tip. (50 Max.)
Frequência	KHz	40
Distância Max.	m	3
Distância Min.	cm	3
Disparo (Trigger)	µS	10 (Min. nível TTL)
Pulso de eco		- Sinal TTL positivo, proporcional à distância ao obstáculo
Dimensões		- 43mm x 20mm x 17mm

Este sonar responde a um impulso de nível lógico 1 com a duração de 10µS, que serve de sinal de disparo. Quando este sinal é recebido pelo sonar, este envia então um sinal acústico ultra-sónico e coloca a linha de eco em modo de escuta durante um período entre 100 µs a 18 ms aproximadamente. Só após 10 µs do último eco recebido se pode enviar um novo pedido de leitura. O que se referiu anteriormente constata-se no diagrama temporal representado na Figura 3.25.

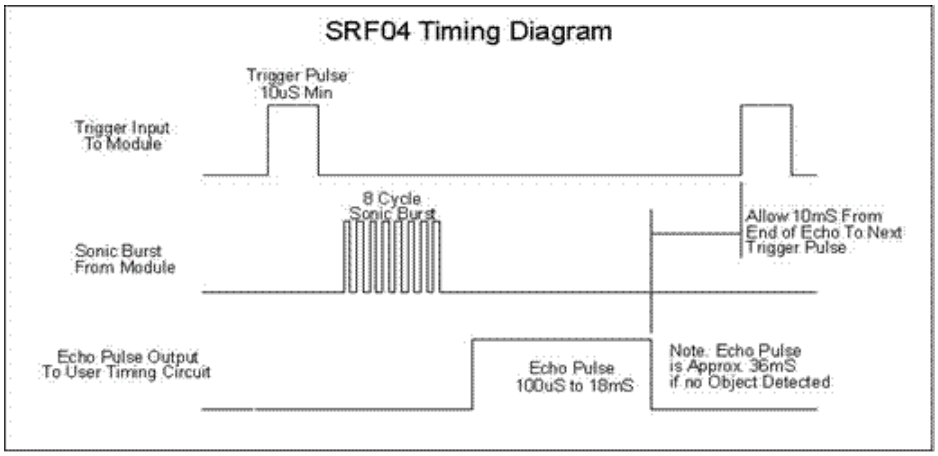


Figura 3.25: Diagrama temporal do funcionamento do SRF-04

A utilização deste tipo de sonar deve-se também à sua sensibilidade, sendo esse um factor importante para a detecção de obstáculos com áreas reduzidas, como acontece com a fita indicadora de zona de obras que tem apenas alguns centímetros de largura. No entanto, a dispersão do sinal acústico não é uniforme, embora seja especificado que o SRF-04 utiliza um ângulo de dispersão de 90º, a sua precisão melhora consideravelmente à medida que esse ângulo é reduzido, como se pode observar no diagrama de potência de radiação de dispersão acústica do SRF-04, na Figura 3.26.

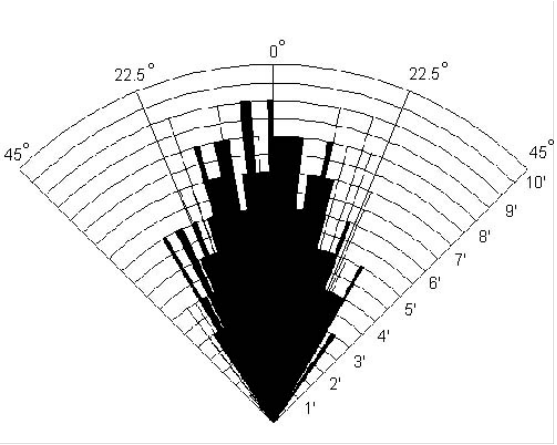


Figura 3.26: Dispersão acústica (Diagrama de potência de radiação) do SRF-04

Graças ao apresentado é comum adoptar-se a solução de usar vários sensores com diferentes orientações de forma a cobrir ângulos de dispersão na ordem de 60º, conseguindo desta forma medições mais precisas.

Módulo de Hardware "Ultra_Sound_Interface"

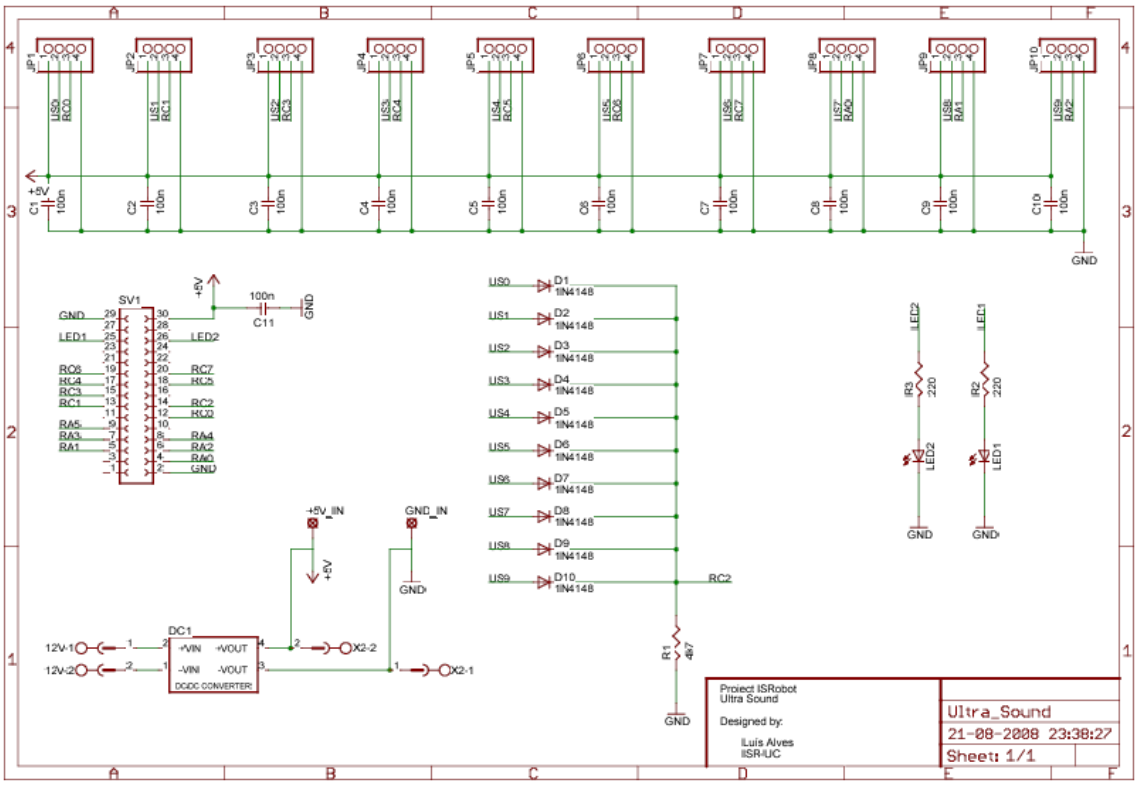


Figura 3.27: Esquemático do módulo *Ultra Sound*

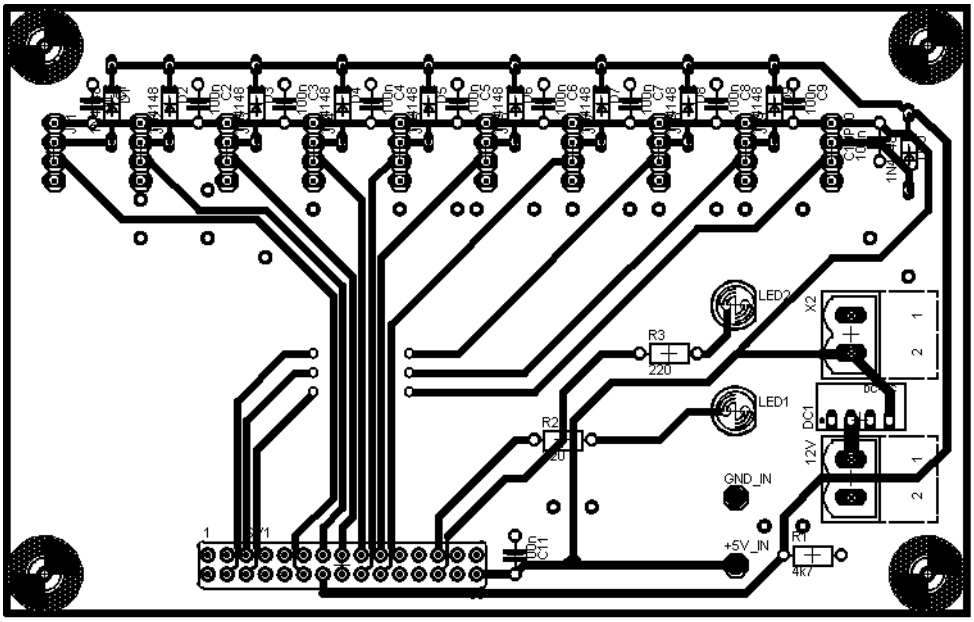


Figura 3.28: Layout do PCB *Ultra Sound Interface* (bottom layer)

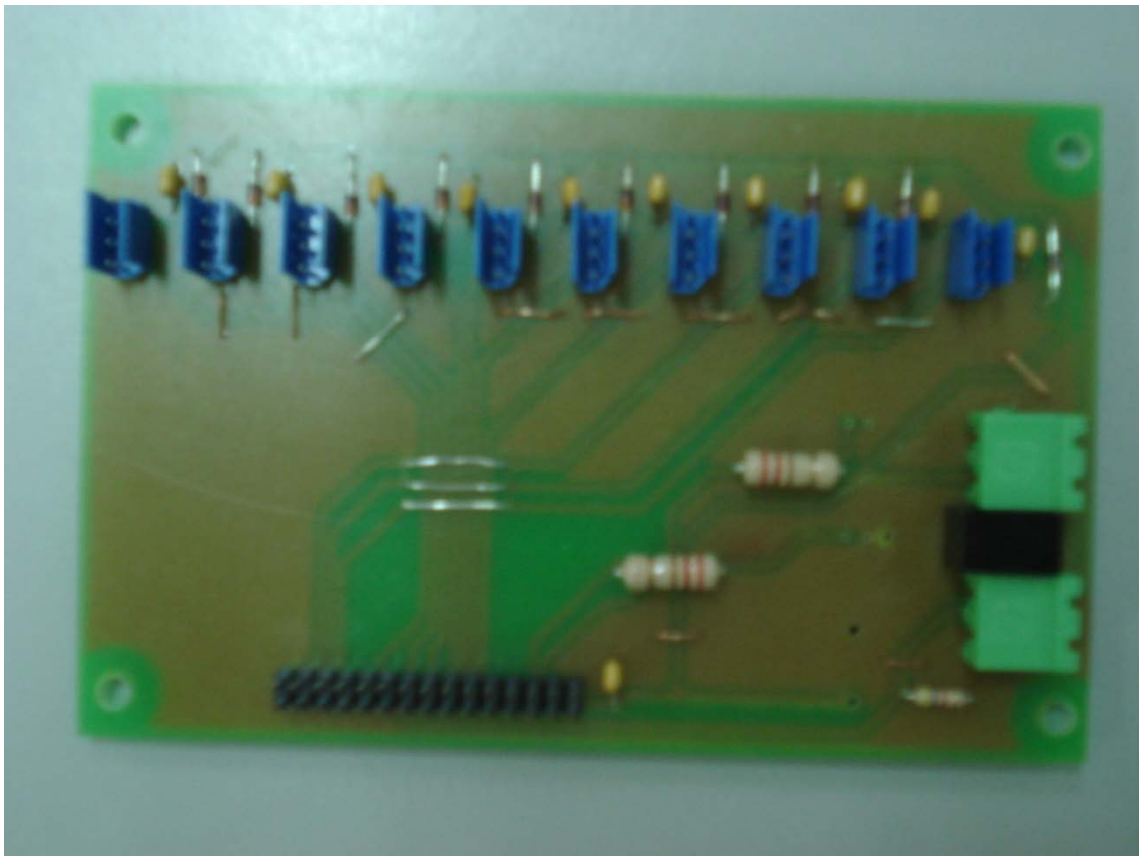


Figura 3.29: PCB do módulo *Ultra Sound Interface*

*3.5 **Módulo de Software/Hardware** "Infra_Red_Interface"*

3.5.1 Software

As funções de CAN para este módulo, são as identificadas na Tabela 3.7.

Tabela 3.7: ID's das funções presentes no módulo *Infra Red*

ID		Função
0		*Turn Node OFF*
1		*Turn Node ON*
4		*Data from Infra Red*

Os valores monitorizados pelo módulo totalizam dois bytes, sendo usados dez bits alinhados à direita, com a informação do estado de cada infravermelho sobre a função "*Data from Infra Red*".

O fluxograma presente na Figura 3.30 mostra resumidamente a estrutura do código desenvolvido para o PIC responsável pelo envio do estado dos infra-vermelhos.

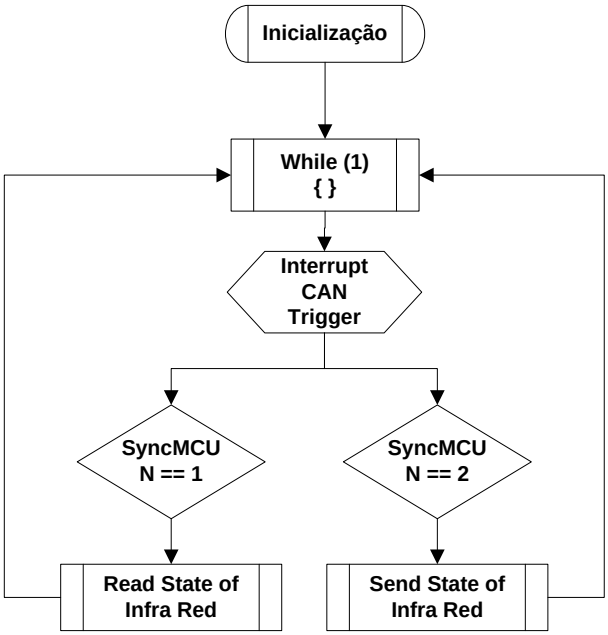


Figura 3.30: Fluxograma do código implementado no Infra Red *Node*

Como é constatado pelo fluxograma presente na Figura 3.30, o *Infra Red Node* envia pelo barramento de CAN para a camada de alto nível, o estado dos infra-vermelhos anteriormente lidos de acordo com as mensagens de sincronismo.

3.5.2 Hardware

Este módulo de hardware foi desenvolvido com o intuito de fazer de interface entre o PIC responsável pela monitorização do estado dos sensores IV. Para os sensores IV existem buffers que garantem a estabilidade dos sinais à entrada do PIC, servindo também de protecção. Este módulo encontra-se preparado para poder ligar até um máximo de dez sensores de infravermelhos.

Características dos IR's

Este tipo de sensores consiste num díodo emissor de infravermelhos e de um transístor fotossensível ("*Phototransistor*"). No caso do IR utilizado, OPB704 encontram-se colocados lado a lado no encapsulamento, como mostra na Figura 3.31.

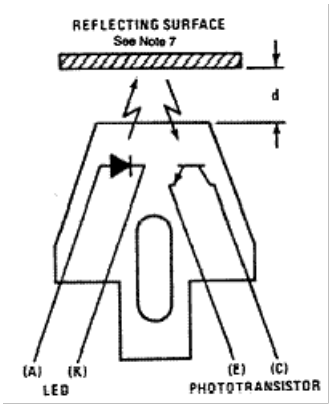


Figura 3.31: Infravermelho OPB704

A distância *d* representada é de cerca de 1,5 centímetros. Esta distância está relacionada com a superfície utilizada na pista presente no concurso, embora este valor varie em função da superfície de reflexão e da luminosidade ambiente.

Módulo de Hardware "Infra_Red_Interface"

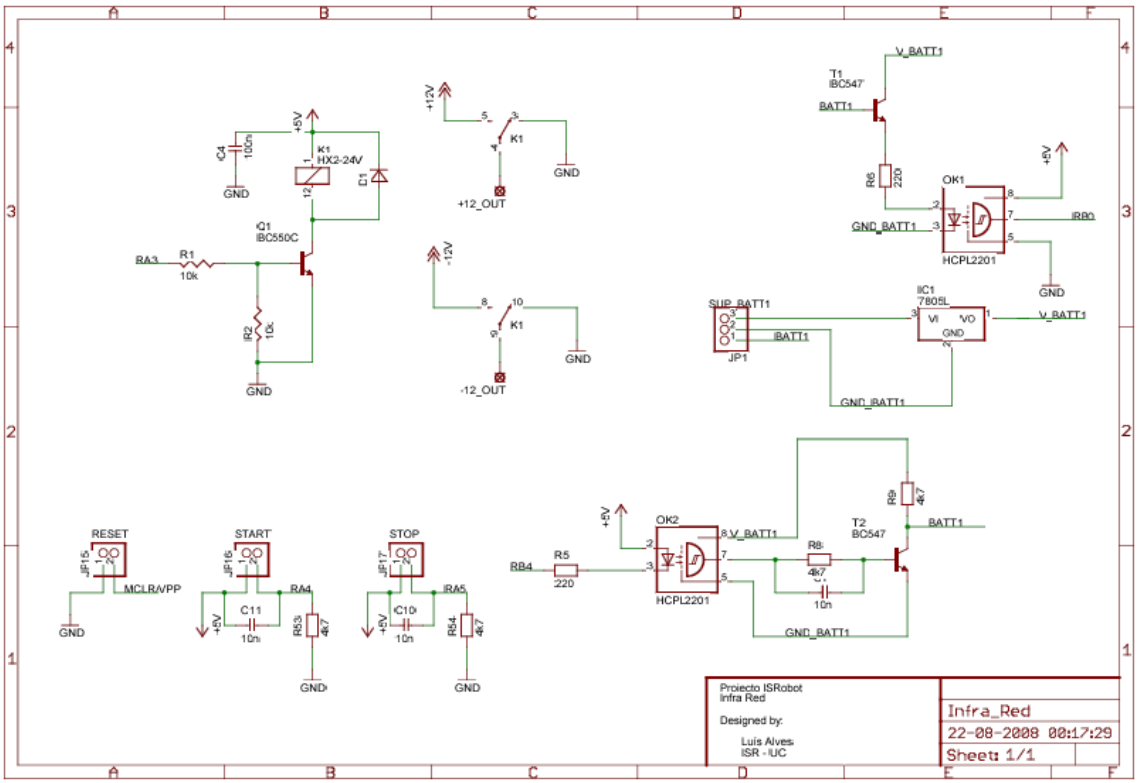


Figura 3.32: Esquemático do módulo *Infra Red* (1/4)

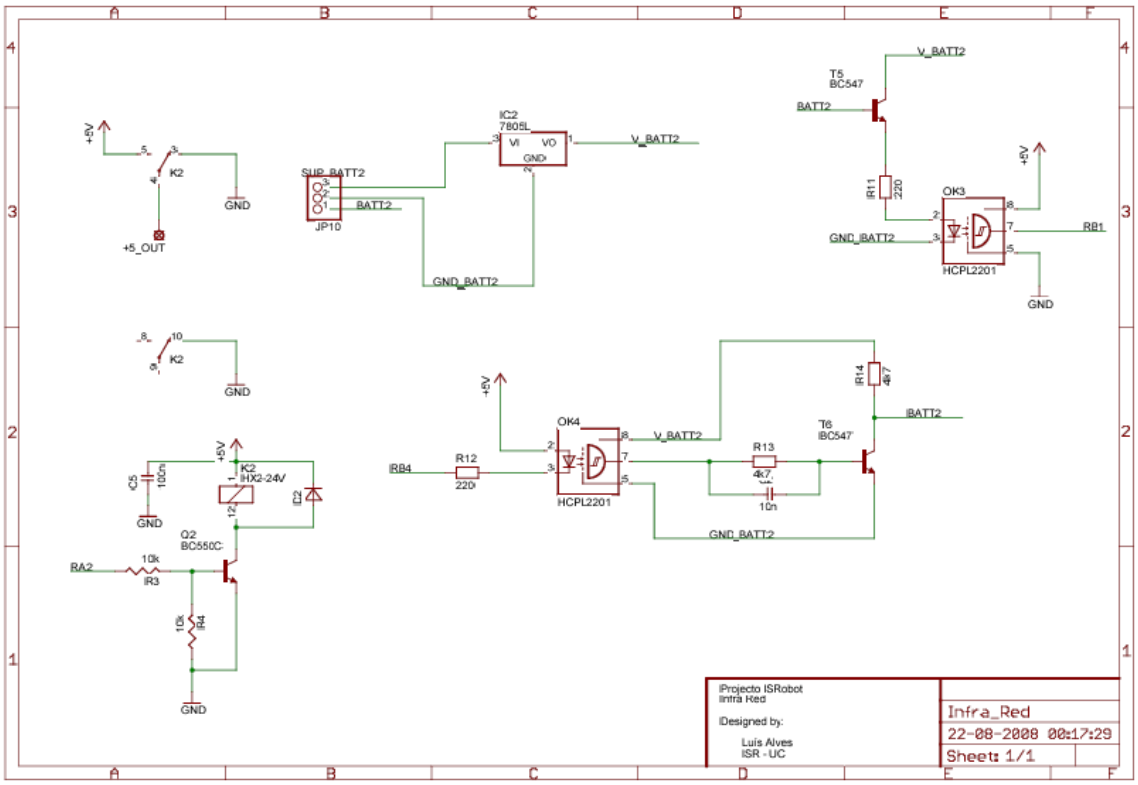


Figura 3.33: Esquemático do módulo *Infra Red* (2/4)

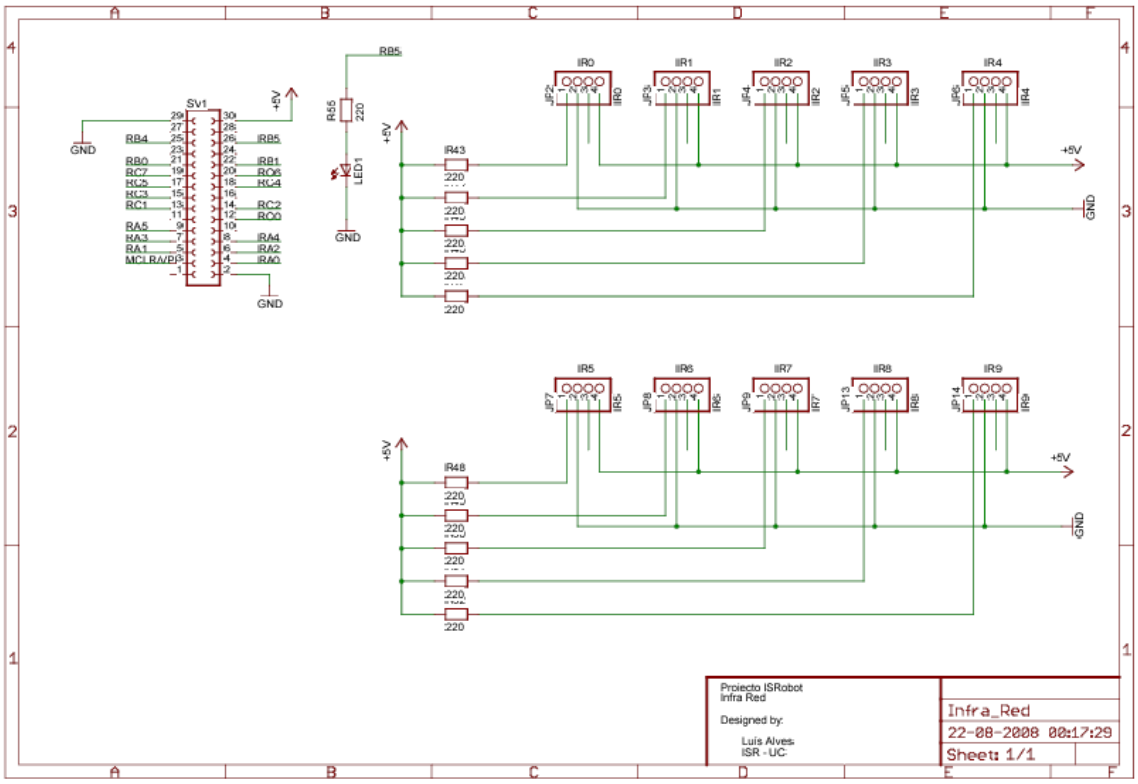


Figura 3.34: Esquemático do módulo *Infra Red* (3/4)

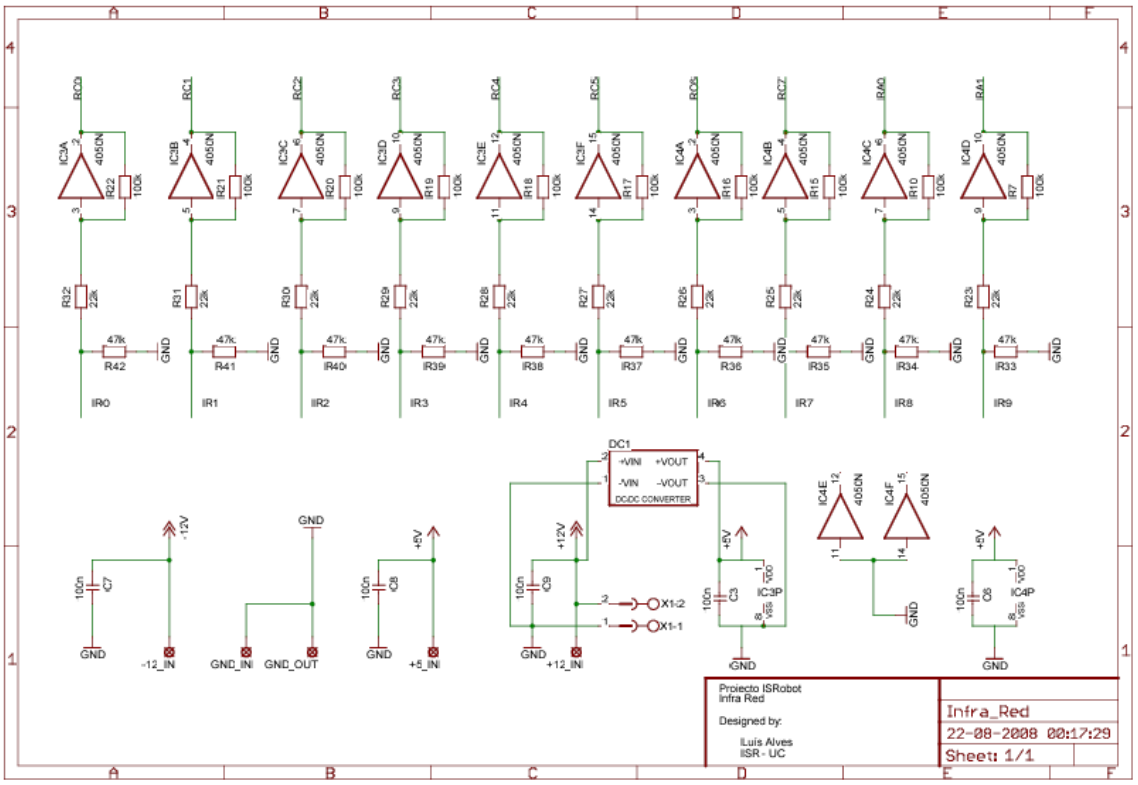


Figura 3.35: Esquemático do módulo *Infra Red* (4/4)

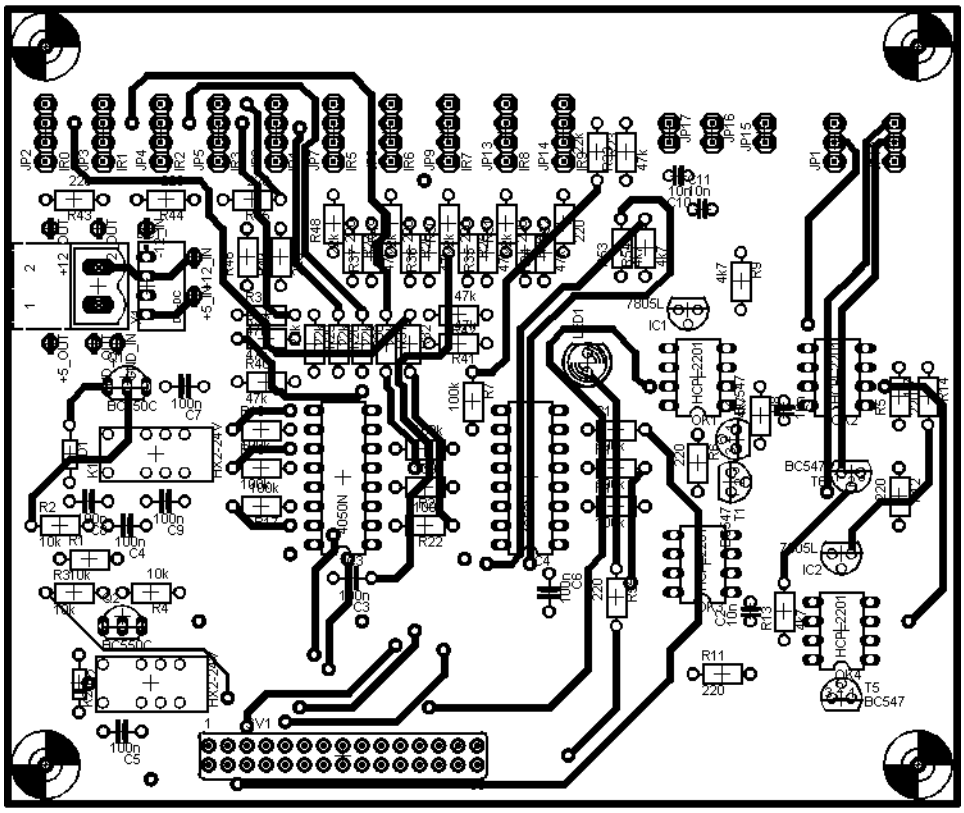


Figura 3.36: Layout do PCB *Infra Red Interface* (top layer)

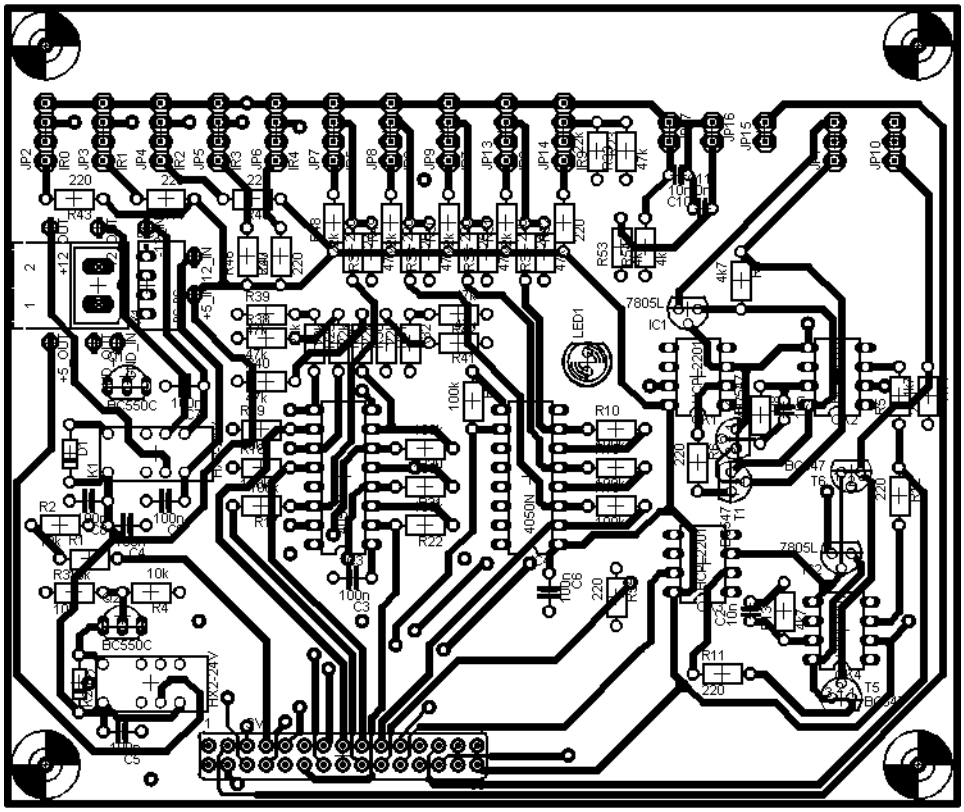


Figura 3.37: Layout do PCB *Infra Red Interface* (bottom layer)

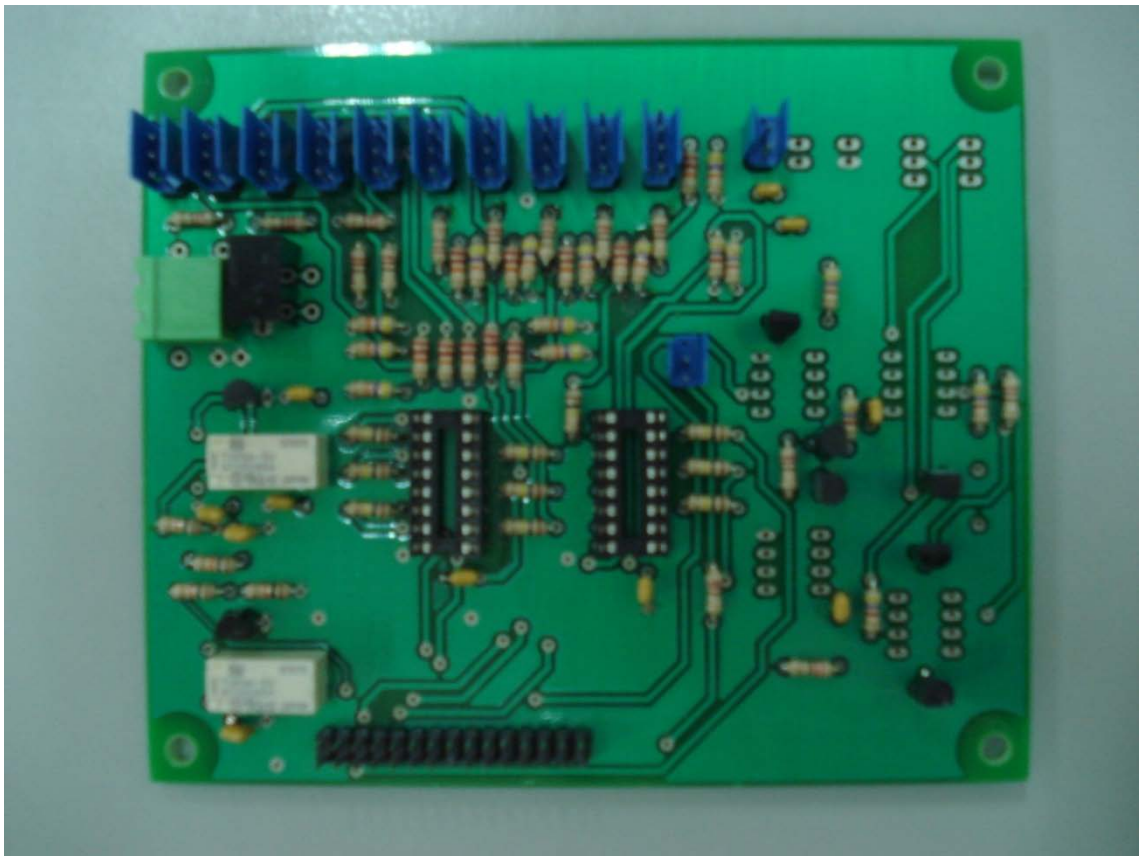


Figura 3.38: PCB do módulo *Infra Red Interface*

3.6 Módulos de Software Presentes no PC

O fluxograma presente na Figura 3.39 representa resumidamente a estrutura do código implementada no PC, para a recolha da informação proveniente do barramento de CAN.

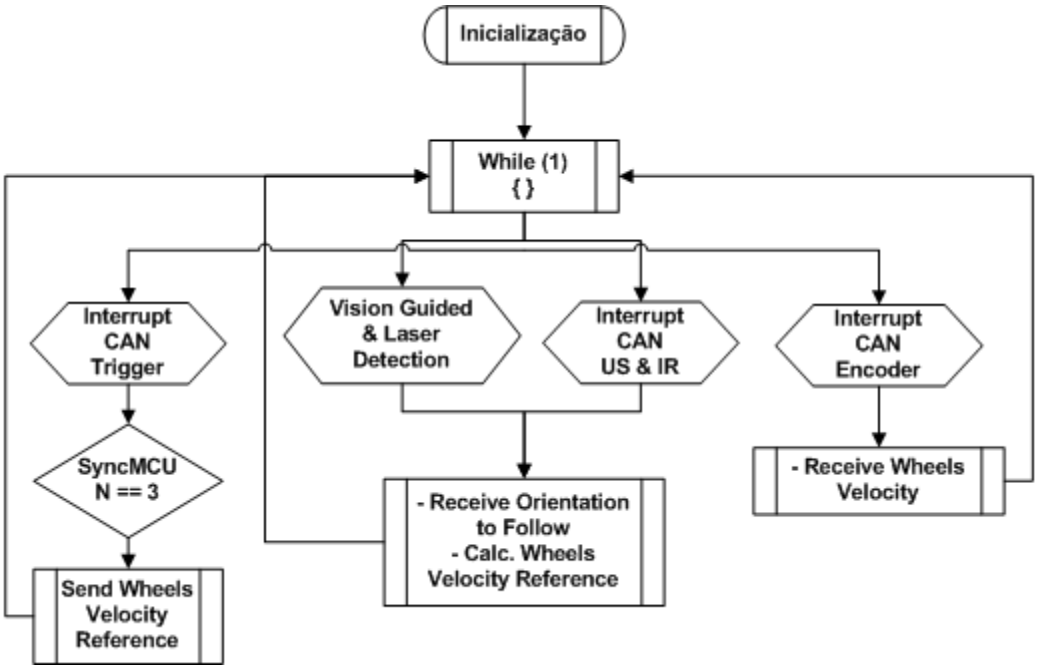


Figura 3.39: Fluxograma dos módulos de controlo no PC

Como é constatado pelo fluxograma presente na Figura 3.39, o PC recebe pelo barramento de CAN da camada de baixo nível, informação do processo o qual este tem de controlar, sendo a recolha da informação respeitante a cargo da camada de interligação. É com base na informação recolhida tanto do CAN como do laser e visão que é calculada a orientação a seguir. Com base nessa orientação é calculada a velocidade desejada para os motores e é posteriormente enviada respeitando o sincronismo e instante alocado para essa tarefa.

3.6.1 Sistema de Tempo Real - RTAI

Com este sistema de tempo real, o programa pode ter várias tarefas (*threads*) com periodicidade e prioridade definidas de acordo com a importância da tarefa.

Neste caso em concreto existe a tarefa principal que faz a gestão dos recursos a serem usados (AutoDrive). É esta que, de acordo com as necessidades cria outras para gerir determinados recursos em concreto. Estes recursos são o CAN, o Laser e a Visão, existindo as tarefas *CAN_Module*, *Laser_Module* e *Vision_Module* respectivamente. Estas tarefas têm como função gerir os respectivos recursos (dispositivos) e fazer o

tratamento dos dados provenientes dos mesmos. Como é desejado que exista um processamento paralelo entre a aquisição e processamento dos dados, ou seja, que se possa fazer uma nova aquisição de dados antes de ter terminado a análise da amostra anterior, temos outras tarefas dedicadas à aquisição dos dados.

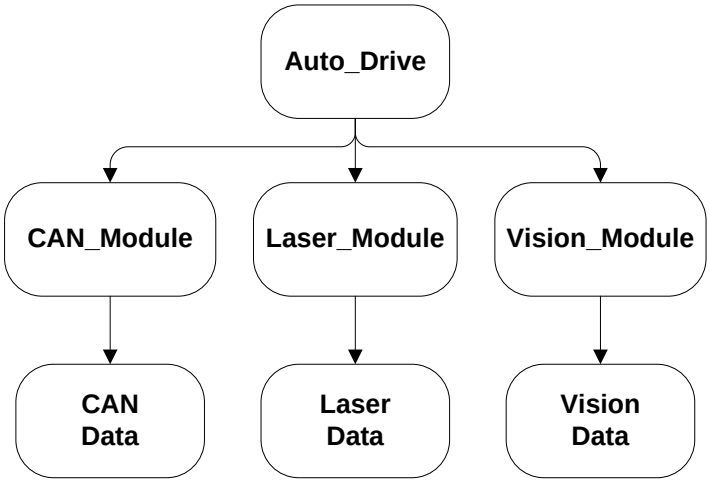


Figura 3.40: Sub-tarefas do programa principal de RTAI

3.6.2 Navegação com recurso a Laser e Visão

Para termos um robô/veículo autónomo aplicado à condução autónoma, este tem de interagir com o meio, sendo esse mundo complexo é usual usar-se laser e visão à semelhança do DARPA Grand Challenge.

Neste caso o laser foi usado apenas como dispositivo de segurança não tendo sido feito a segmentação e classificação de qualquer ordem. Funcionando como laser de segurança quer dizer que é definida uma zona próxima do mesmo onde caso seja detectado algo é despoletada uma acção de segurança. O alcance do mesmo é de cerca de 4m, tendo a zona de segurança sido definida nos 0.3m para imobilização do robô.

O laser usado (Hokuyo) faz varrimentos de 240º com uma resolução de 0.3516º. Esta amplitude foi subdividida em cinco zonas distintas (30º+50º+80º+50º+30º), tendo sido dado importância apenas às três zonas centrais (180º), e que permite identificar se o obstáculo se encontra na frente ou num dos lados do robô. Desta forma é possível, no caso particular da prova, decidir se o robô pode prosseguir na faixa de rodagem actual ou se tem de mudar para a outra.

Neste caso particular o objectivo do uso da visão é identificar as linhas delimitadoras das duas faixas de rodagem e respectiva linha central tracejada. Com este objectivo foram usadas duas câmaras (uma de cada lado da estrutura frontal do robô), com vista a detectar a linha mais próxima do robô, sendo que nesses caso o varrimento da imagem da câmara direita faz-se da esquerda para a direita e a outra faz no sentido contrário. A ideia com esta solução era estando o robô na faixa direita orientar-se pela linha direita continua, e caso passasse para a faixa esquerda orientar-se pela linha esquerda contínua. Devido à limitação de processamento do PC a aquisição de imagens é feita apenas numa câmara. Infelizmente esta solução não foi devidamente explorada, tendo-se optado por usar apenas uma câmara central a seguir a linha tracejado como alternativa mais simples. Neste caso o varrimento da imagem faz-se da zona central da imagem para as extremidades e segue-se a linha tracejado, tendo sido atingido uma solução minimamente aceitável tendo em conta as limitações existentes neste vasto leque de soluções passíveis de serem implementadas futuramente.

Capítulo 4

4 Sistema Motriz e de Alimentação

4.1 Motores

Os motores usados permitem uma velocidade (*v*) e uma aceleração (*a*) máxima de 2m/s e 2m/s^2 respectivamente, ou seja, que o robô acelere até à velocidade máxima em apenas um segundo. Sendo que a massa do robô é aproximadamente 20 Kg.

O motor DC é alimentado a 24 V e está acoplado a um sistema de engrenagem de redução de velocidade, permitindo desta forma ter um binário mais elevado e velocidade perto da gama especificada. A curva velocidade/binário para o motor RE 30 (Graphite Brushes, 60Watt) da Maxon® é a apresentada na Figura 4.1.

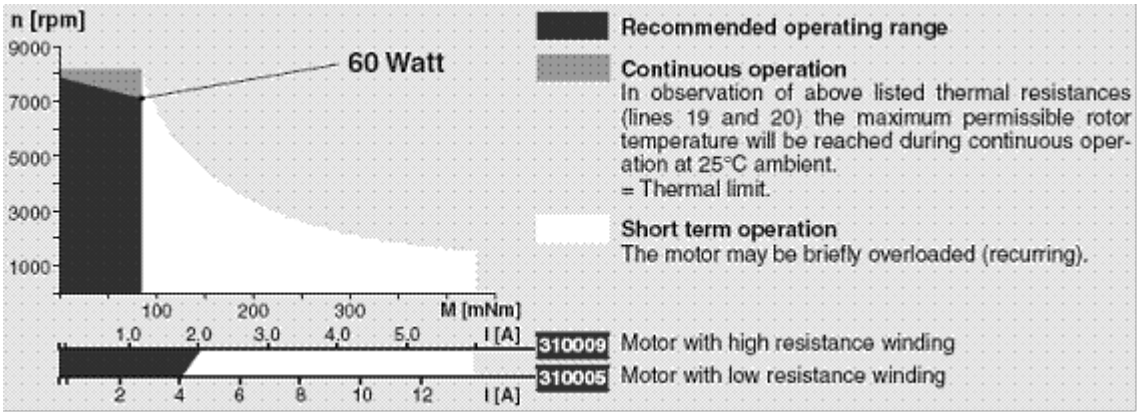


Figura 4.1: Curva característica do motor

A este motor é acoplado uma caixa de engrenagem GP32A (*Planetary Gearhead* 0,75-4,5Nm) da Maxon®, com uma redução aproximada de 1/33 (redução real 529/16), que permite desta forma, e de acordo com a curva característica do motor, que a velocidade máxima das rodas seja dada pela Equação 4.1:

$$\omega_{Rodas} = \frac{\omega_{motor}}{33} \cong \frac{7000}{33} = 212{,}12\ rpm \tag{4.1}$$

Para um binário máximo, por motor, dado pela Equação 4.2.

$$\tau_{Rodas} = \tau_{motor}.33 \cong 0{,}08 \times 33 = 2.64\ Nm \quad (4.2)$$

Como o sistema é movido com recurso a dois motores, cada um acoplado a uma das rodas, então obtém-se um binário resultante que será o dobro do apresentado na Equação 4.2, ou seja, um binário resultante total de aproximadamente *5,28 Nm*.

4.2 Codificador Óptico

O codificador óptico está acoplado ao motor como se mostra na Figura 4.2.

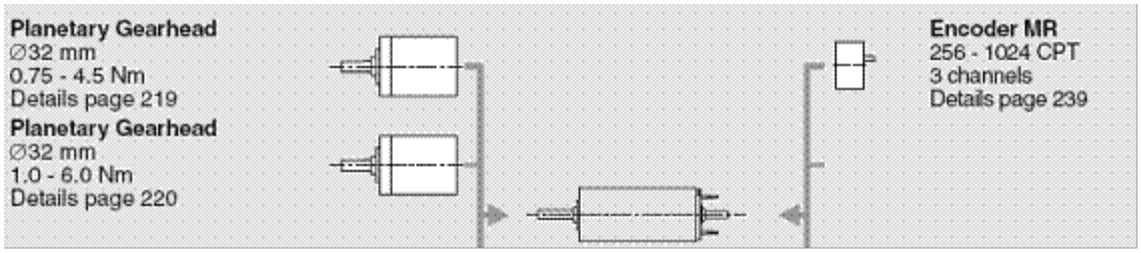


Figura 4.2: Composição motor+caixa de engrenagem+encoder

O codificador óptico é o Encoder MR TypeL da Maxon®, tratando-se de um codificador óptico de 3 canais em modo diferencial com a resolução de 256 pulsos por volta. O canal A e B têm uma resolução de 256 pulsos por volta, encontrando-se estes dois canais em quadratura sendo o terceiro um canal de Índex que envia apenas um pulso por volta. É alimentado com uma tensão de 5V, sendo esta feita a partir do módulo de potência Servoamplifier ADS (4-Q-DC) da Maxon®, sendo as suas características principais apresentadas na próxima secção. O conector do codificador óptico tem a configuração apresentada na Figura 4.3:

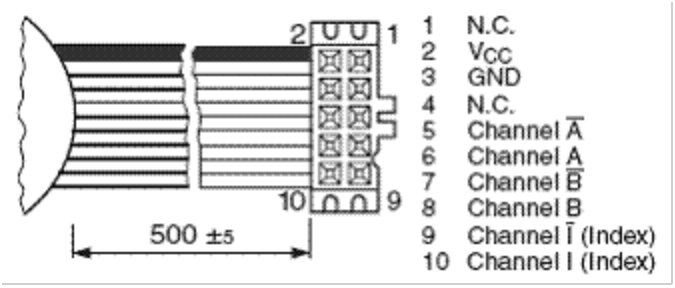


Figura 4.3: Configuração do conector do encoder

4.3 Módulo de Potência

Neste sub capítulo pretende-se essencialmente dar a conhecer o módulo de potência ("*Power Drive*") utilizado e as suas características mais importantes.

O módulo de potência utilizado é o Servoamplifier ADS (4-Q-DC) da Maxon®, que dispõe de 4 modos de controlo para o motor.

As características dos principais sinais de entrada e saída do módulo são apresentadas de seguida na Tabela 4.1, onde se pode encontrar informações relativas à alimentação do módulo bem como em relação aos sinais de saída.

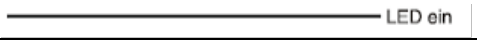
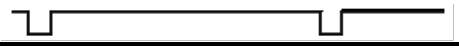
Tabela 4.1: Características principais do módulo

Sinal		Valor
Tensão de alimentação		12 a 50 V
Tensão máxima de Saída		0.9 * V_{CC}
Corrente máxima de Saída		10 A
Corrente continua de Saída		5 A
Frequência de comutação		50 kHz
Eficiência		95 %
Entradas	Set Value	-10 a 10 V (R_i=20kΩ)
	Enable	4 a 50 V (R_i=15kΩ)
	Sinais de Encoder	f_{Max}=100kHz (Níveis TTL)
Saídas	Monitorização de Corrente	-10 a 10 V (R_o=100Ω)
	Monitorização de Velocidade	-10 a 10 V (R_o=100Ω)
	Indicador de Estado “Ready”	Montagem em colector aberto

4.3.1 Estado do Módulo

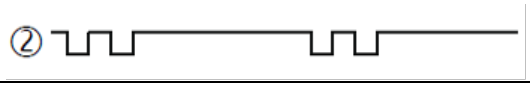
Para facilitar o diagnóstico de eventuais problemas de funcionamento que possam ocorrer no módulo, este encontra-se munido de 2 led’s que indicam o seu estado e a ocorrência de algum erro. O estado do módulo, bem como os erros que possam surgir no seu funcionamento podem ser então definidos de acordo com o estado dos led’s descritos na Tabela 4.2 e Tabela 4.3.

Tabela 4.2: Estado do módulo de potência (led verde)

Led Verde	
Estado do Led	Estado do Módulo
	Módulo Activado
	Módulo Alimentado mas Inactivo

Como é perceptível, a cor verde do Led indica apenas se o módulo está activado ou desactivado, o que implica que só quando este está activado os sinais de tensão colocado nas entradas "+/-*Set value*" serão tidos em conta pelo módulo.

Tabela 4.3: Estado do módulo de potência (led vermelho)

Led Vermelho	
Estado do Led	Estado do Módulo
	Indicação de que a temperatura dos Transístores de Potência (Andar de Potência) foi excedida, 90ºC. Módulo desactivado.
	Indicação que a corrente máxima admitida pelo motor foi atingida, aprox.12.5A. Módulo desactivado.
	Indicação que a fonte de tensão interna não consegue atingir o valor de tensão pretendido. Módulo desactivado.
	Indicação que os sinais de entrada do codificador óptico excedeu a frequência máxima admissível, 150 KHz. Módulo desactivado.

Quando ocorrem os erros descritos anteriormente, o módulo passa ao estado de inactivo, sendo necessário reactivá-lo novamente, sempre que tal aconteça. Ou seja, não basta que o problema desapareça para ele passar ao modo activo. Se o problema persistir, quando a activação do módulo é iniciada, este coloca-se imediatamente no estado inactivo até que o problema seja resolvido.

As causas associadas ao primeiro tipo de erro são as mais variadas, podendo por exemplo dever-se a uma temperatura ambiente muito elevada, que pode ocorrer em aplicações especiais ou por existir uma má dissipação/ventilação do módulo. Pode ainda dever-se a um máximo de corrente contínua permitida atingida, situação essa que será

aprofundada mais à frente quando forem abordados os potenciómetros de ajuste que o módulo possui. Essa corrente é normalmente de 5A.

4.3.2 Entradas e Saídas

Os sinais de entrada e saída deste módulo estão divididos em três tipos:

- ***Power***, onde se encontra a alimentação do módulo e as saídas de alimentação para o motor.

- ***Signal***, onde se encontram sinais de entrada de controlo, e sinais de saída indicando o estado de funcionamento do módulo.

    i. *+/- Set Value*, sinal de entrada em tensão (-10V a 10V) para controlo de sentido e velocidade do motor.

    ii. *Enable*, sinal de entrada em tensão, activa (com tensões entre 4V a 40V) a alimentação do Motor. Inactivo (com tensões de 0V a 2,5V) cortando a alimentação do motor.

    iii. *DC Tacho*, Sinal de tensão (2V a 50V), para ligação de Tacómetros, para controlo de velocidade.

    iv. *Monitor n*, Sinal de saída em tensão (-10V a 10V), equivalente à velocidade que o motor está a rodar.

    v. *Monitor I*, Sinal de saída em tensão (-10V a 10V), equivalente à corrente que o motor está a consumir.

    vi. *Ready*, Coloca a entrada ligada à massa quando o módulo se encontra a funcionar correctamente, coloca a saída em alta impedância quando o módulo não se encontra a funcionar correctamente, cujo esquema é apresentado na Figura 4.4.

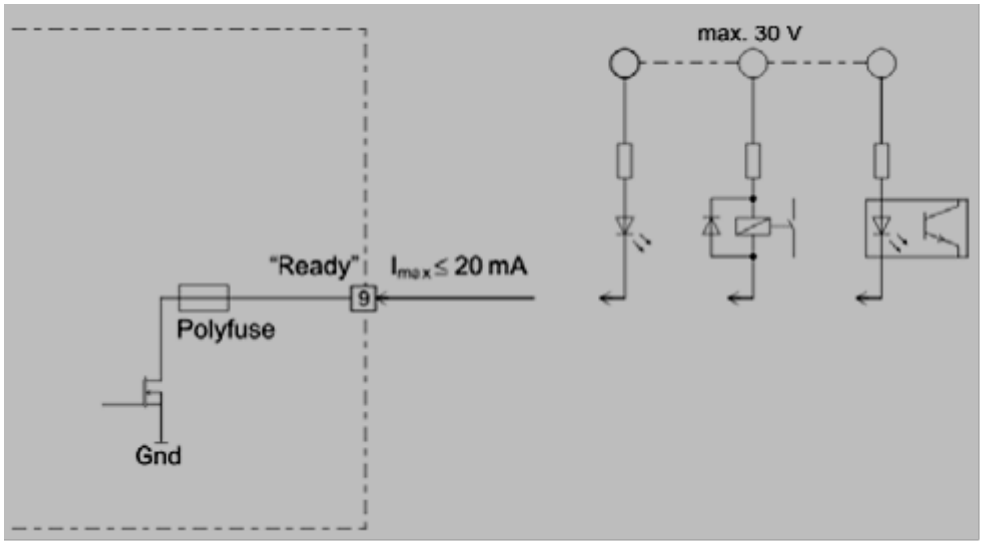
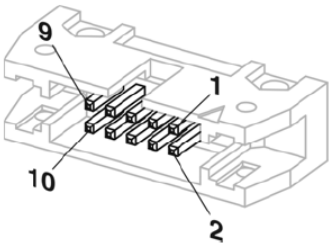


Figura 4.4: Configuração do circuito interno para o sinal "*ready*"

- **Encoder**, Sinais vindos do encoder de acordo com o esquema na Figura 4.5.



Pin configuration at "Encoder" input:

1	n.c.	Not connected
2	+5 V	+ 5 VDC max. 80 mA
3	Gnd	Ground
4	n.c.	Not connected
5	A\	Inverted Channel A
6	A	Channel A
7	B\	Inverted Channel B
8	B	Channel B
9	n.c.	Not connected
10	n.c.	Not connected

Figura 4.5: Ligações dos sinais de encoder

4.3.3 Modos de Controlo

De seguida apresentam-se os modos de controlo que se mostraram mais adequados para o controlo motriz do robô.

Os modos de controlo, disponíveis no módulo são, os seguintes:

1. Controlo de velocidade, a partir de sinais de Tacómetro.

2. Controlo de Velocidade, a partir de sinais de Encoder.

3. Controlo de velocidade por compensação de *IxR*

4. Controlo de Binário ou Corrente

A selecção do modo de controlo é feita com recurso a um conjunto de DIP-Switchs que se encontram num local de fácil acesso do exterior, sendo que as configurações de selecção são as representadas na Figura 4.6.

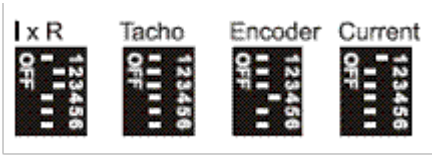
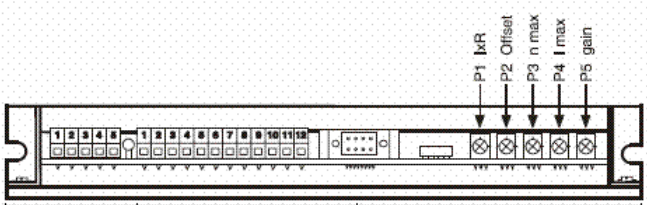


Figura 4.6: Configuração dos *DIP-Switchs* de selecção de modo

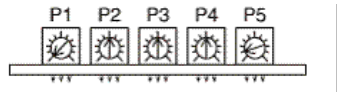
4.3.4 Potenciómetros de Ajuste de Funcionamento

Antes de abordar o modo de controlo usado há que dar algumas notas introdutórias acerca do funcionamento do módulo. Este módulo possui alguns potenciómetros que permitem fazer ajustes no seu funcionamento. Esses potenciómetros encontram-se no exterior do módulo de modo a ser fácil a sua manipulação, estando estes pré-ajustados (ver Figura 4.8) para o tipo de utilização mais comum do módulo. No quadro apresentado na Figura 4.7, são apresentadas as funções dos vários potenciómetros e qual o tipo de ajuste que estes permitem, bem como a posição dos potenciómetros tendo em conta a perspectiva lateral do módulo.



Potentiometer		Function	Turn to the left ↺	Turn to the right ↻
P1	IxR	IxR compensation	weak compensation	strong compensation
P2	Offset	Adjustment n = 0 / I = 0 at set value 0 V	motor turns CCW	motor turns CW
P3	n_{max}	max. speed at 10 V set value	speed slower	speed faster
P4	I_{max}	current limit	lower min. 0.5 A	higher max. 10 A
P5	gain	amplification	lower	higher

Figura 4.7: Potenciómetros de ajuste



Pre-adjustment of potentiometers		
P1	IxR	0 %
P2	Offset	50 %
P3	nmax	50 %
P4	Imax	50 %
P5	gain	10 %

Figura 4.8: Pré ajuste dos potenciómetros

Para além destes potenciómetros existe ainda a possibilidade de ajustes mais específicos, encontrando-se, no entanto, os potenciómetros respectivos no interior do módulo. A sua manipulação não é permitida sem que o módulo seja aberto.

Estes potenciómetros, P6, P7 e P8 cuja localização se apresenta na Figura 4.9, servem respectivamente para controlar o ganho em velocidade, o ganho em corrente e o valor de corrente de saída máximo em regime contínuo.

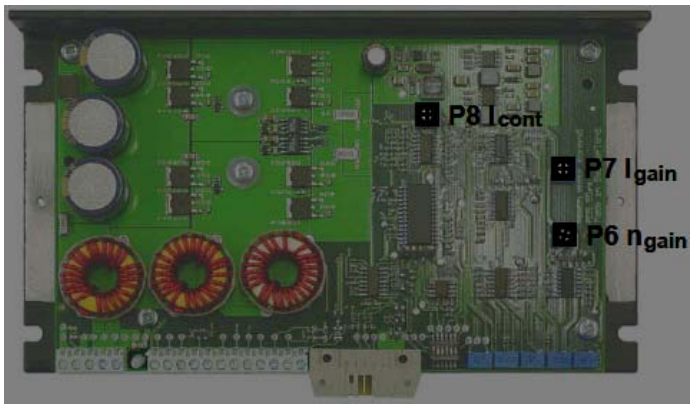


Figura 4.9: Localização dos potenciómetros

Os prés-ajustes dos potenciómetros P6 e P7, são respectivamente 25% e 40%. No que respeita ao potenciómetro P8 este serve para regular o valor de corrente em modo contínuo admissível, sendo que o valor de pico, regulado através do potenciómetro P4 será aceite durante um período de 0,1s em cada 1s. No tempo restante encontra-se limitado ao valor de corrente em modo contínuo regulado pelo potenciómetro P8, como se mostra na Figura 4.10. No entanto, este modo de controlo do valor máximo da corrente só é realizado quando o DIP switch 6 se encontra activo.

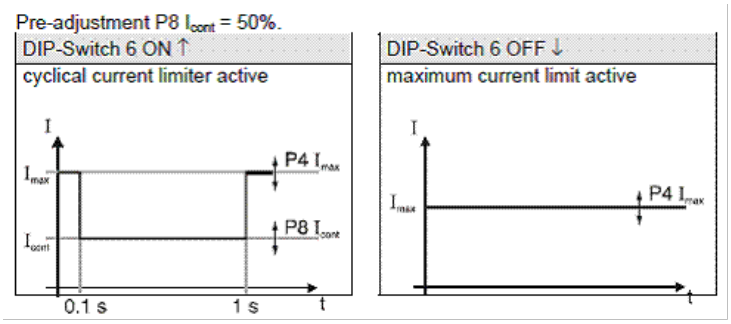


Figura 4.10: Acção do potenciómetro P8

4.3.5 Controlo em Modo de Encoder

O controlo em modo de encoder é realizado com base na informação que recebe do encoder acoplado ao motor. O valor que se pretende para a velocidade corresponde a uma dada tensão colocado na entrada "*set value*", que irá entrar no controlador de acordo com a informação que o próprio módulo de potência recebe do encoder, controlando o motor à velocidade desejada.

O controlo de velocidade é feito com base num controlador analógico que se encontra implementado em hardware, consistindo num amplificador inversor e numa montagem integradora, cuja dinâmica pode ser ajustada recorrendo à regulação dos potenciómetros de ajuste P5 e P6.

O esquema interno do módulo de potência pode ser consultado no catálogo do Servoamplifier ADS50/5 da Maxon® [10], estando de seguida na Figura 4.11 descrito de forma esquemática toda a cascata de montagens amplificadoras que implementam a malha responsável pelo controlo de velocidade do módulo.

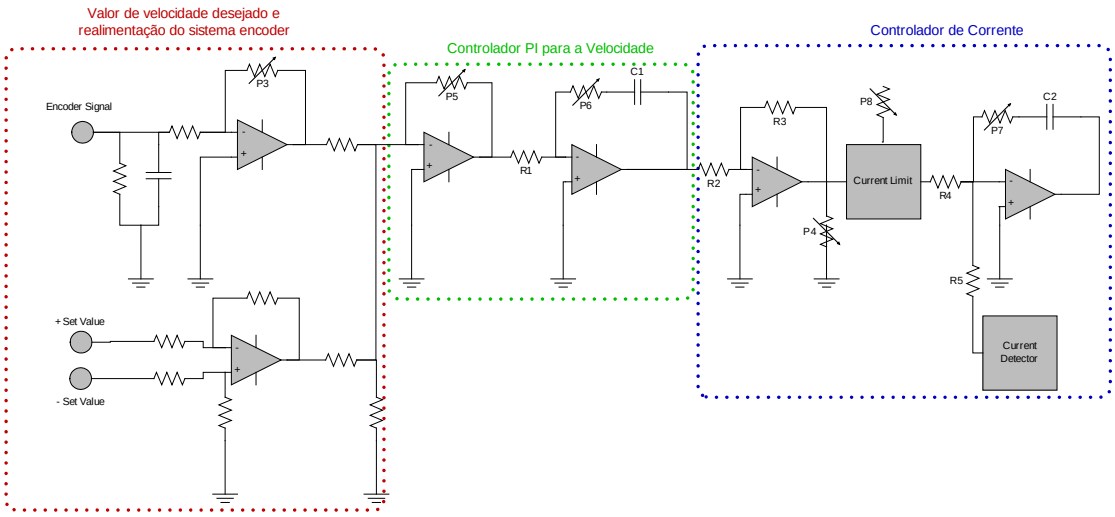


Figura 4.11: Esquema da malha de controlo em modo de encoder

A função de transferência da malha do controlador PI (Proporcional Integral) em velocidade é dada por $H_1(s)*H_2(s)$, com $H_1(s)$ a função de transferência de um amplificador inversor e $H_2(s)$ a função de transferência de um amplificador integrador como mostra a Figura 4.12.

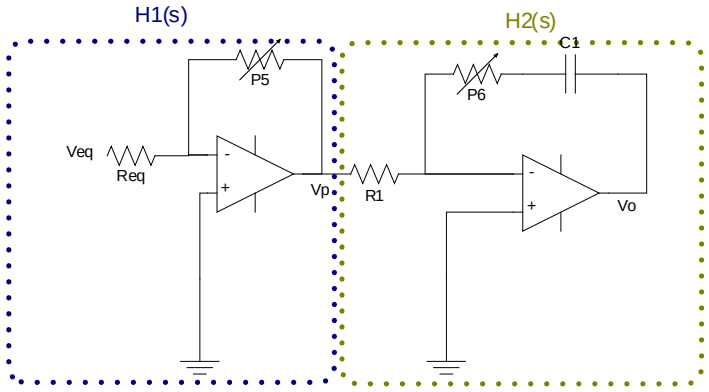


Figura 4.12: Controlador PI de velocidade do módulo

No que respeita ao amplificador inversor, considera-se que a impedância de entrada da montagem R_{eq} é elevada, ficando a sua função de transferência dada pela Equação 4.3:

$$H_1(s) = \frac{V_p}{V_{eq}} = -\frac{P_5}{R_{eq}} \tag{4.3}$$

O integrador tem a função de transferência dada pela Equação 4.4.

$$H_2(s) = \frac{P_6 + \frac{1}{sC_1}}{R_1} = -\frac{1}{R_1} \cdot \left(1 + \frac{1}{sP_6C_1}\right) \tag{4.4}$$

Variando o valor do potenciómetro P5 e o potenciómetro P6 pode-se então ajustar a sintonização do controlador PI, sendo a sua função de transferência dada por:

$$\begin{aligned} H(s) &= H_1(s) \cdot H_2(s) = \left(-\frac{P_5}{R_{eq}}\right) \cdot \left(-\frac{1}{R_1} \cdot \left(1 + \frac{1}{sP_6C_1}\right)\right) \\ \Leftrightarrow H(s) &= \overbrace{\frac{P_5}{R_{eq}R_1}}^{K_{si}} \cdot \left(1 + \frac{1}{s\underbrace{P_6C_1}_{\tau_{si}}}\right) \end{aligned} \tag{4.5}$$

O ajuste do potenciómetro P5 controla então a acção proporcional ao variar o ganho K_{si}, e o ajuste do potenciómetro P6 controla a acção integral ao variar o valor da constante de tempo τ_{si}.

4.4 Baterias

A autonomia de um robô móvel autónomo está intimamente ligada às características das baterias que suportam todo o sistema e este é, obviamente, concebido tendo em vista a minimização do consumo de energia.

Tabela 4.4: Consumos do sistema

Equipamento/Modulo	Corrente[(1)] (A)	Potência Consumida[(1)] (W)
PC	2	48
Conjunto (Motor + Power Drive)[(2)]	1,5	36
Restantes módulos de hardware	0,5	12

(1)Valores Aproximados; (2) Para os dois conjuntos

No que respeita ao consumo do sistema, são apresentados valores aproximados na Tabela 4.4 para os vários constituintes, conseguindo dessa forma chegar a um valor próximo do consumo do sistema, que se estima em 4 A/h, garantindo as duas baterias aproximadamente 2 horas de autonomia. As baterias para aplicações em robótica móvel autónoma têm que preencher alguns requisitos no que respeita ao seu tamanho e peso. Tal facto contribui por optar por baterias secas (mais leves e compactas).

Pretendendo-se um sistema distribuído simetricamente, então a opção mais certa é utilizar duas baterias, e dadas, as características dos vários módulos de hardware, a tensão nominal mais apropriada seria de 12V. Ligando estas em série obtém-se então a tensão nominal dos motores 24V.

São utilizadas as baterias da SAFT. As suas principais características apresentam-se resumidamente na Tabela 4.5:

Tabela 4.5: Características principais da bateria

Caracteristica	Valor/Especificação
Designação	VH Module D
Tensão nominal	12V
Capacidade mínima	8 Ah

Capacidade tipica		8.4 Ah
Corrente de Carga	Main	3 A
	Balancing	180 mA
	Trickle	90 mA
Duração de Carga	100%*	6 Horas
	90%	3 Horas

* - É necessário efectuar uma carga de 100% a cada 20 ciclos de carga/descarga ou de 3 em 3 meses.

O aspecto das baterias bem como uma ideia das dimensões desta podem ser observadas na Figura 4.13.

Figura 4.13: Aspecto da bateria

*4.5 **Módulo de Hardware** "Power_Convertion"*

Este módulo de hardware ao contrário dos outros módulos não é activo, não possui ligação com nenhum módulo "*PIC_Base*", nem com o PC. Serve apenas para centralizar e proteger todas as entradas e saídas de alimentação e gerir a fonte de energia a usar de forma electromecânica.

4.5.1 Esquema de Alimentações do Sistema

Tendo em vista a minimização da cablagem, a alimentação de todos os módulos de hardware do sistema distribuído é feita a 12V.

Dado o sistema de alimentação ser composto por duas baterias ligadas em série que alimenta os módulos de potência e respectivos motores, é no módulo "*Power_Convertion*" que se obtém os 12V. Neste módulo são obtidos os 12V recorrendo a um conversor DC-DC, TRACO® POWER TEM 25-2412, que apresenta a capacidade de ser desligado externamente.

Dado que grande parte dos módulos necessita também de alimentação a 5V estes tem incorporado um mini conversor que satisfaz essa necessidade.

No que respeita a tensões de referência (precisão) necessárias para as montagens amplificadorasutiliza-se IC's dedicados para o efeito que garantem a não flutuação desses sinais de referência em função da carga imposta à alimentação do módulo de hardware.

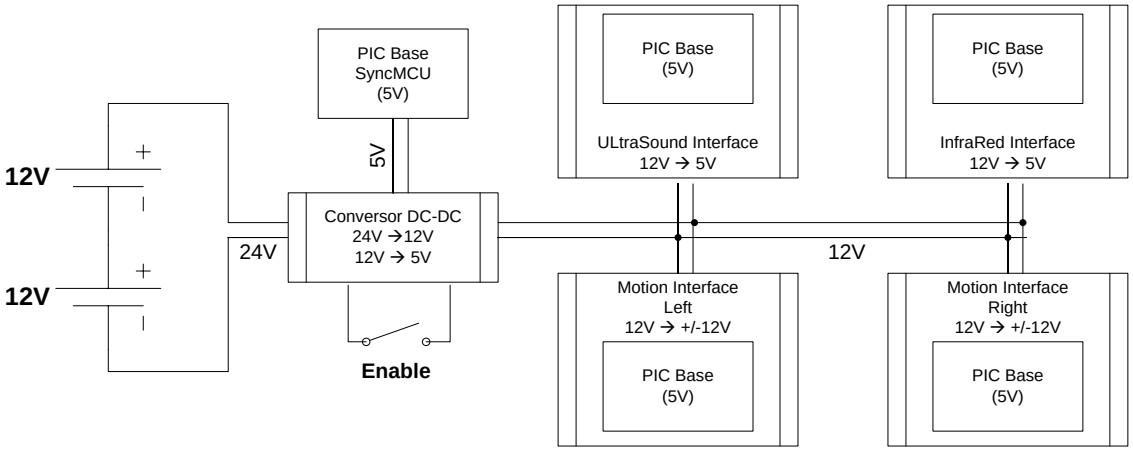


Figura 4.14: Esquematização das alimentações dos vários módulos

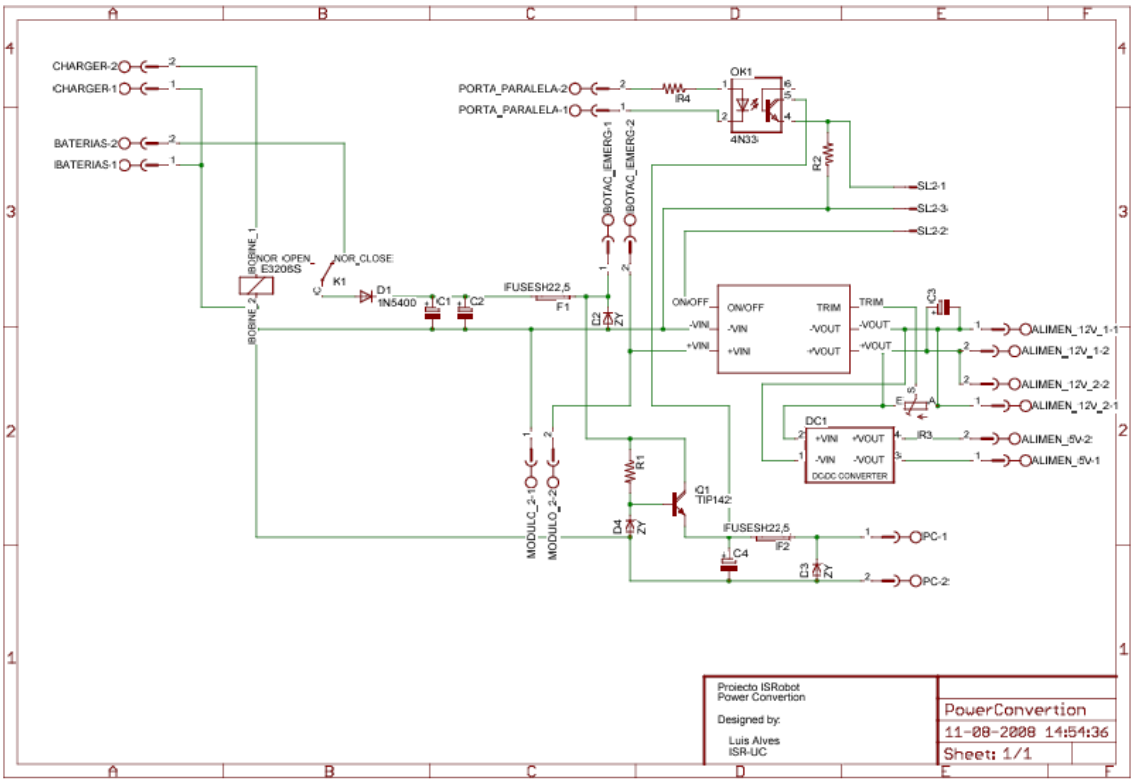


Figura 4.15: Esquemático do módulo *Power Convertion*

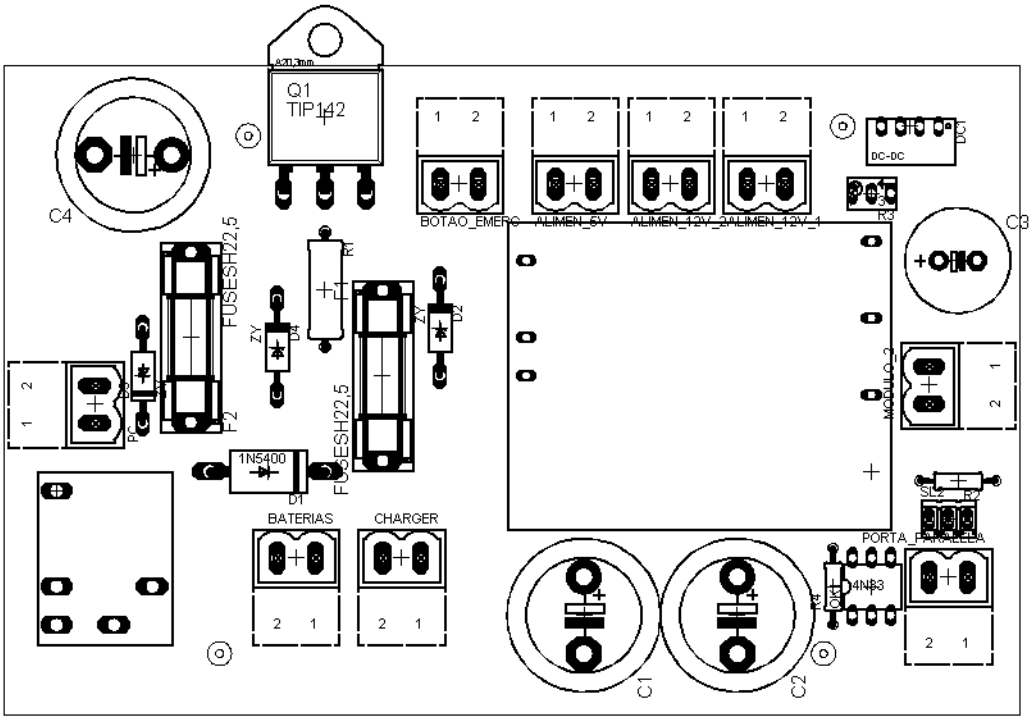


Figura 4.16: Layout do PCB *Power Convertion Interface* (top layer)

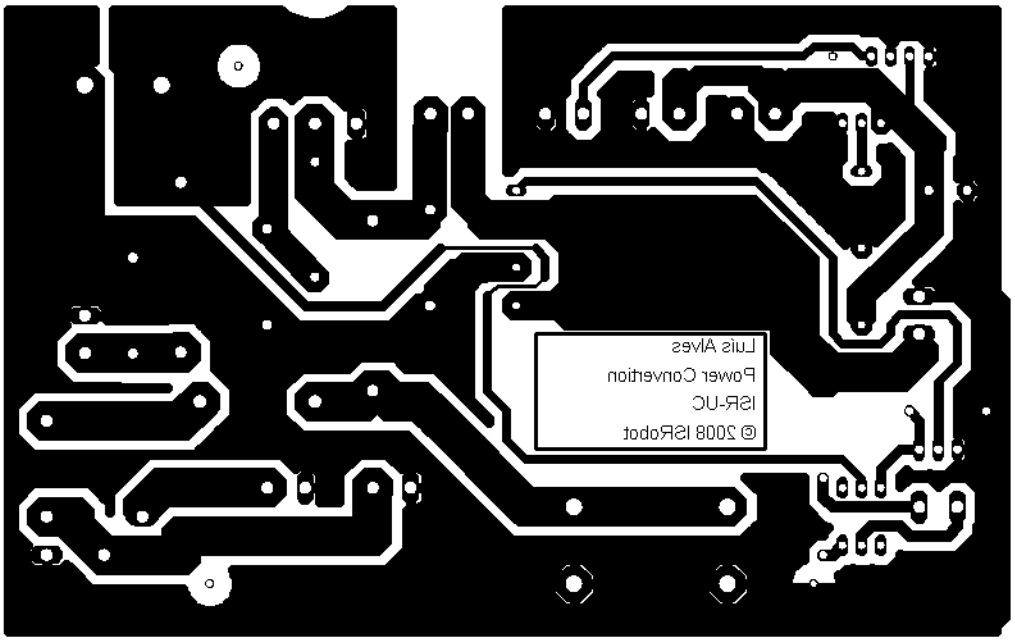


Figura 4.17: Layout do PCB *Power Convertion Interface* (top layer)

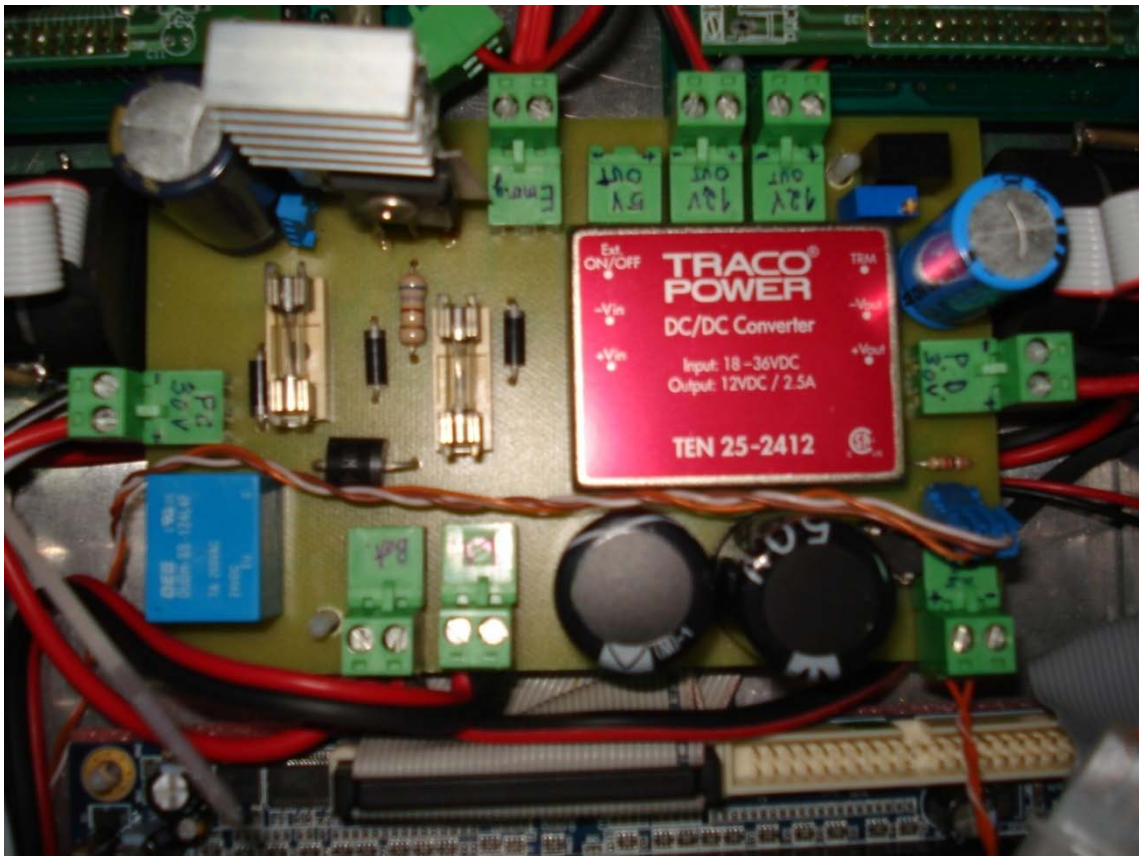


Figura 4.18: PCB do módulo *Power Convertion*

4.6 Estrutura de Suporte do Robô

A estrutura do Robô é constituída por uma plataforma robusta toda ela em alumínio trabalhado mecanicamente, possui duas rodas motrizes frontais de alumínio com um oringue de borracha, e uma roda livre de apoio na parte de trás. A Figura 4.19 apresenta o robô de perfil onde se verificam essas características.

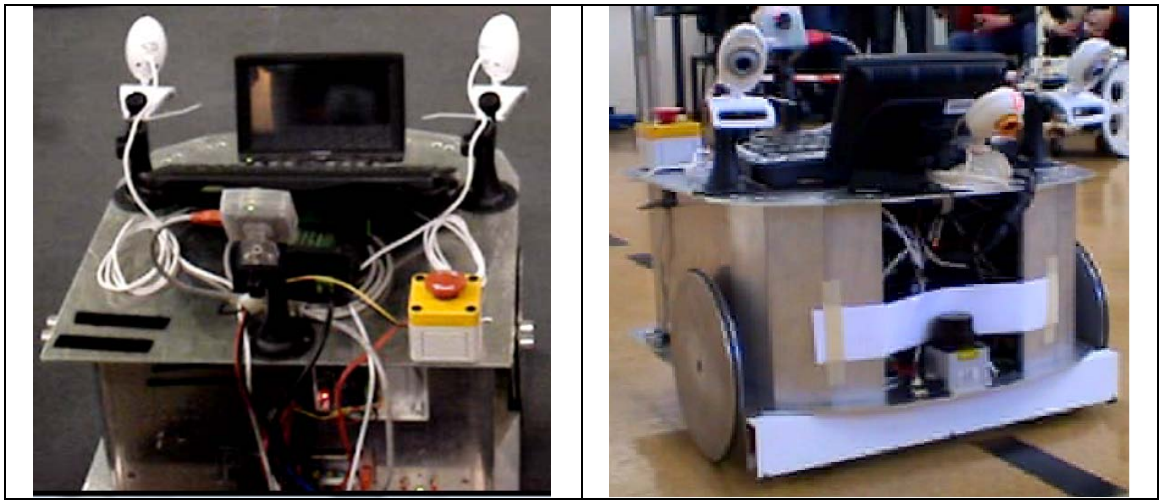


Figura 4.19: Perfil do robô

Esta plataforma foi concebida de forma a poder integrar, no seu interior, todos os componentes necessários à sua autonomia. De seguida, apresentam-se as dimensões reais da plataforma e a localização dos vários componentes que compõem a mesma.

A localização dos vários componentes como, baterias, motores, módulos de potência, placa do PC e os outros módulos de hardware, está representada na Figura 4.20. Sendo que a projecção do robô ocupa sensivelmente 0,5m por 0,5m.

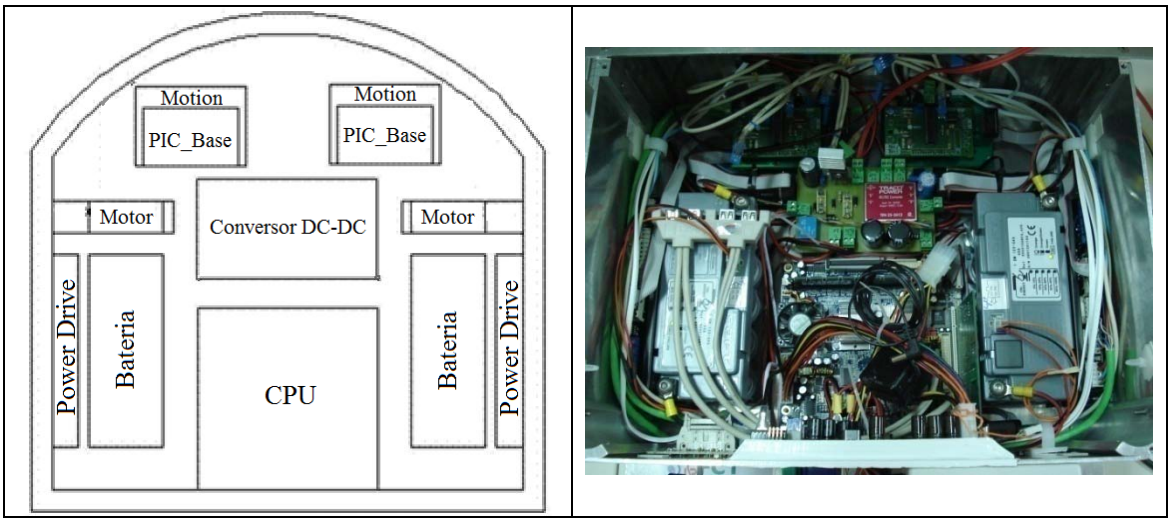


Figura 4.20: Localização do hardware dentro da estrutura

Um dos aspectos importantes, para além das dimensões do robô, é o seu peso, e a distribuição da carga (centro de massa). Esta última já se encontra com uma distribuição optimizada dado a simetria de ambos os lados do robô, sendo que o peso total do robô é aproximado em cerca de 20 kg.

4.7 Sistemas Periféricos do Robô

Como é perceptível, toda a informação circula entre os vários PIC's e o PC, que é o "cérebro" do sistema, através de um barramento CAN. Os PIC's comportam-se como escravos (*Slaves*), cumprindo as ordens do PC, que vai ser o mestre (*Master*).

Os PIC's associados aos sensores de IV e Ultra-Sons, têm como tarefa enviar o estado dos sensores periodicamente, para tomada de decisões por parte do PC. Temos portanto ligado à rede distribuída de CAN alguns sensores, que em termos de quantidade de dados é suportável para este tipo de rede.

Outros tipos de sensores tais como o Laser e a Visão são sistemas periféricos que são ligados directamente ao PC, visto usarem outro tipo de comunicação que suporta uma taxa de transferência de dados maior, que é o USB e FireWire respectivamente.

Para além destes sensores periféricos, existem alguns mecanismos de segurança inerentes ao robô tais como: um botão de emergência local para parar o sistema ou por rádio frequência (telecomando) que se revelou indispensável em fase de testes.

Associado ao PC temos os periféricos normais para interacção com o mesmo, que é um mini-tft com *touch-screen* e teclado.

Capítulo 5

5 Conteúdo do CD

- Documentos

  Documentos de trabalhos realizados com a plataforma ISRobot.

- Hardware

  - Dispositivos

  Manual dos diversos componentes para consulta de informação mais detalhada.

  - Pic's

  Esquemático das diversas placas que estão ligadas ao barramento de can e que servem de interface entre a placa do microcontrolador e os dispositivos de hardware. É usado o programa Eagle para visualizar e alterar os esquemáticos. Ficheiros Grebber também presentes para a replicação das placas caso necessário.

- Filmes

  Resultados experimentais do ISRobot em acção.

- Imagens

  Imagens dos diversos componentes existentes no ISRobot.

- Software

  - Código PC

  Programa de RTAI, necessário ter privilégios de super-utilizador para executar e carregar os módulos de RTAI e CAN.

  - Código Pic's

Programas dos diversos microcontroladores, para compilar é usado o compilador da Hitech “picc18”. Para programar é necessário usar o programador por CAN “canbootmngr_v2”. Modo de uso “./canbootmngr_v2 -h” para ver a ajuda. Modo de uso mais aprofundado no capitulo 3.1.2.