## Arquitetura Escalar de Aquisição de Dados Sensoriais para um Veículo Autônomo

por

Henrique Serdeira

DCC/IM/NCE - UFRJ

2005

### Henrique Serdeira

## Arquitetura Escalar de Aquisição de Dados Sensoriais para um Veículo Autônomo

Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática do Instituto de Matemática e do Núcleo de Computação Eletrônica, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Ciências em Informática.

Orientadora:

Prof<sup>a</sup>. Eliana Prado Lopes Aude, Ph.D.

Rio de Janeiro 2005

### FICHA CATALOGRÁFICA

S482 Serdeira, Henrique.

Arquitetura escalar de aquisição de dados sensoriais para um veículo autônomo / Henrique Serdeira. - Rio de Janeiro: UFRJ/IM/NCE, 2005.

105 f.:il.

Dissertação (Mestrado em Informática) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Núcleo de Computação Eletrônica, 2005.

Orientadora: Eliana Prado Lopes Aude

- 1. Arquitetura de Aquisição de Dados Teses.
- 2. Veículo Autônomo Teses.
- 3. Interface Teses. I. Eliana Prado Lopes Aude (Orient.).
- II. Universidade Federal do Rio de Janeiro. Instituto de (Matemática. Núcleo de Computação Eletrônica. III. Título

CCD

## Arquitetura Escalar de Aquisição de Dados Sensoriais para um Veículo Autônomo

### Henrique Serdeira

Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática do Instituto de Matemática e do Núcleo de Computação Eletrônica, da Universidade Federal do Rio de Janeiro - UFRJ, como parte dos requisitos necessários à obtenção do título de Mestre em Ciências em Informática.

	Henrique Serdeira
provado por:	
	Prof <sup>a</sup> . Eliana Prado Lopes Aude, Ph.D.
	(Universidade Federal do Rio de Janeiro)
	(Presidente)
	Prof. Ernesto Prado Lopes, Ph.D.
	(Universidade Federal do Rio de Janeiro)
	Prof. João Carlos Machado, Ph.D.
	(Universidade Federal do Rio de Janeiro)
	Prof. Mauro Antonio Rincon, Ph.D.
	(Universidade Federal do Rio de Janeiro)

Rio de Janeiro, RJ – Brasil Setembro – 2005

A Deus, aos meus pais, à minha esposa Silvia e aos meus filhos Bruno e André.

## Agradecimentos

Ao NCE/UFRJ, Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro, que com seu ambiente de fomento a pesquisa forneceu a estrutura para a realização deste projeto.

À FINEP e ao CNPq, pelo financiamento e apoio dados ao projeto.

À minha orientadora, Professora Eliana Prado Lopes Aude, por me oferecer a oportunidade de trabalhar neste projeto, transmitindo suas experiências e conhecimentos preciosos, e nunca permitindo que eu desistisse diante das dificuldades encontradas.

Aos colegas de trabalho do projeto CONTROLAB, Eliana Prado Lopes Aude, Mario Ferreira Martins, Júlio Tadeu Carvalho da Silveira e Ernesto Prado Lopes, que me propiciaram uma excelente convivência para o meu aprimoramento profissional.

A minha esposa, Silvia Gavinho, que com indescritível ternura e paciência soube aguardar a realização desta etapa.

Aos meus filhos, Bruno e André, que durante longo tempo presenciaram um pai ocasionalmente impaciente e ausente.

Aos meus pais, Elso e Lecy, e a minha irmã, Claudia, que sempre me apoiaram em minhas decisões, oferecendo o amor irrestrito de uma família.

Aos colegas do NCE e da Área de Ensino e Pesquisa (AEP/NCE), em especial: Marcio Thadeu, José Maria, Flávio Signorelli, Marcela, Cristiano Aguiar, Marcelo Signorelli e Rafael Carvalho, pelo apoio durante este trabalho.

As funcionárias da secretaria do Departamento de Computação Científica, em especial a "Tia Daise", por resolver todas as pendências burocráticas.

### Resumo

SERDEIRA, Henrique. Arquitetura Escalar de Aquisição de Dados Sensoriais para um Veículo Autônomo. Rio de Janeiro, 2005. Dissertação (Mestrado em Informática) - Instituto de Matemática e Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

Este trabalho descreve uma arquitetura escalar capaz de aquisitar dados sensoriais a serem utilizados pelos sistemas de controle do Veículo Autônomo do CONTRO-LAB desenvolvido no Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro (NCE/UFRJ). Tanto o "hardware" como o "software" dessa arquitetura foram desenvolvidos para permitirem fácil expansão da capacidade sensorial do robot.

Inicialmente é apresentada a arquitetura geral do sistema. Em seguida é mostrada a descrição detalhada do *hardware* do sistema de aquisição de dados dos sensores e acionamento dos dispositivos dos motores. A seguir, são apresentadas as rotinas desenvolvidas para o comando da interface.

Finalmente, são apresentadas as conclusões obtidas através de experiências práticas realizadas com um conjunto de sensores de infravermelhos que constituem a bengala eletrônica utilizada pelo Veículo Autônomo para desviar de obstáculos.

## Abstract

SERDEIRA, Henrique. Arquitetura Escalar de Aquisição de Dados Sensoriais para um Veículo Autônomo. Rio de Janeiro, 2005. Dissertação (Mestrado em Informática) - Instituto de Matemática e Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

This work describes a scalable architecture capable of acquiring data from an environment to be used by the CONTROLAB AGV's (Autonomous Guided Vehicle) control systems developed at Computer Center of Federal University of Rio de Janeiro. Both the hardware and the software are designed to allow future expansion of the robot's sensing capabilities.

Initially, the architecture's overview is presented. Then, the detailed description of the hardware of the sensor data acquisition and motor actuators is shown. Following, the routines developed to command the interface are presented. Finally, results obtained from practical experiments with a set of infrared sensors that form the Electronic Stick used by the AGV to avoid obstacles are discussed.

# Sumário

$\mathbf{R}$	esum	O	5
$\mathbf{A}$	bstra	act	6
Sı	ımár	io	7
Li	sta d	le Figuras	9
Li	sta d	le Tabelas	11
Li	stas	de Abreviaturas e Siglas	12
1	Intr 1.1	rodução Requisitos impostos pelos sistemas do Veículo Autônomo à aquisição	13
	1.2 1.3	de dados	13 14 15
2	2.1 2.2	sores e atuadores estudados Introdução	17 17 18 20 24 25
	2.3	Sensores de detecção de obstáculos	27 28 32 34
3	Ana 3.1 3.2 3.3	Alise de arquiteturas   Introdução	37 37 39 44
4	4.1 4.2	Introdução	49 49 49

	4.4 4.5 4.6 4.7 4.8	Atuação nos motores CC	66 70 73
5	Rot	nas do sistema de aquisição de dados	30
	5.1	,	80
	5.2	Funcionamento das rotinas da interface de comandos do Veículo Au-	
		ônomo	80
		1 7	83
		5.2.2 Rotinas de tratamento os motores CC	85
		5.2.3 Rotinas de tratamento dos sensores infravermelhos	91
		5.2.4 Rotinas de tratamento dos sensores de ultra-som	93
		S.2.5 Rotinas de tratamento de outros sensores	93
6	Res	Itados experimentais	94
	6.1	ntrodução	94
	6.2	Experimentos de aquisição	
		6.2.1 Encoder	
		5.2.2 Bengala	
		3.2.3 Ultra-som	
7	Cor	lusões e trabalhos futuros 10	)1
-	7.1	Considerações Finais	
$R\epsilon$	eferê	cias Bibliográficas 10	)3

# Lista de Figuras

2.1	Encoder óptico incremental com dois canais (A e B)	21
2.2	Saída do encoder incremental, em quadratura de fase, mostrando o	
	sinal para sentido horário (H) e anti-horário (A)	22
2.3	Exemplo de encoder óptico absoluto de oito bits codificado em código	
	Gray	23
2.4	Diagrama de funcionamento de medidor de velocidade com utilização	
	de efeito Doppler	25
2.5	Propagação do pulso de ultra-som (a) emissor e sensor em pontos	
	diferentes (b) emissor e sensor no mesmo ponto	28
2.6	Esquema básico de sensor de ultra-som. O transdutor utilizado neste	
	esquema funciona como emissor e receptor de ultra-som	30
2.7	Forma de onda dos pulsos de ultra-som. Os vários pulsos de eco	00
	referem-se a reflexões de objetos em posições diferentes	31
2.8	Diagrama de medidor de distância utilizando infravermelho através	01
2.0	da medida do deslocamento de fase $\varphi$ de um pulso de refletido de	
	infravermelho	33
2.9	Implementação da bengala eletrônica no Veículo Autônomo composta	00
2.0	de sete sensores GP2D02	35
2 10	Funcionamento da triangulação no sensor GP2D02	36
2.10	Tuncionamento da triangulação no sensor en 2002.	50
3.1	Classificação das tecnologias digitais nos circuitos integrados	41
3.2	Esquema interno de uma PAL	42
3.3	Diagrama de uma CPLD da série MAX7000 da ALTERA	43
3.4	Bloco lógico de uma FPGA da Xilinx	44
3.5	Esquema de uma FPGA da série 10k da ALTERA	45
3.6	Etapas do processo de projeto de um dispositivo programável da AL-	
	TERA	47
3.7	LAB da família MAX7000 da ALTERA	48
3.8	Diagrama interno de uma pastilha da família MAX7000	48
4.1	Diagrama em blocos da arquitetura do Veículo Autônomo CONTRO-	
	LAB-AGV.	51
4.2	Estrutura internas dos barramentos da interface SIGA	54
4.3	Sinais do sistema de gerenciamento da interface SIGA	55
4.4	Forma de onda PWM com $duty\ ratio$ igual a $\delta$ e período igual a $T.$ .	62
4.5	Diagrama básico do circuito acionador PWM para motor CC	63
4.6	Circuito amplificador em ponte H para permitir mudança na direção	
	dos motores CC	63

4.7 4.8 4.9	Sinais do circuito para atuação de um motor CC por PWM Sinais do circuito do sistema de atuação de 5 motores CC Saída do encoder incremental, em quadratura de fase, mostrando o	64 65
1.0	sinal para sentido horário (H) e anti-horário (A)	67
4.10	Máquina de estados de discriminação do encoder incremental para detección de sentido herério (H) e enti herério (A)	68
4.11	detecção de sentido horário (H) e anti-horário (A)	69
	Sinais do circuito do sistema para 3 contadores de posição	70
	Sinais do módulo de ultra-som utilizados pelo Veículo Autônomo. O	
	sinal início determina a emissão de pulsos ultra-sônicos com o sinal	
	saída_dig acusando a recepção somente do primeiro eco	71
	Sinais do circuito do medidor por ultra-som	72
	Sinais do circuito do sistema para 4 medidores por ultra-som	73
	GP2D02 - Diagrama interno do sensor	74
1.11	e Saída	75
4.18	Sinais do circuito para leitura do sensor GP2D02	75
	Máquina de estados para leitura da medida do sensor GP2D02 dis-	
	ponibilizada via serial	76
4.20	GP2D02 - Sinais do circuito do sistema para leitura de 8 sensores	
1 91	GP2D02	77 79
4.21	Smais do circuito do sistema para entrada e saida paraieras	79
5.1	Diagrama de blocos das rotinas de manipulação da interface do Veí-	
	culo Autônomo.	82
5.2	Estrutura da variável MOT_DRV_atual, responsável pelos dados da	0.5
5.3	odometria	87
0.0	distância medido pela bengala eletrônica	91
C 1		
6.1	Circuito para fornecer histerese de forma a impedir oscilações nas transições da saída dos sensores dos encoders	95
6.2	Gráfico dos dados de calibração dos sensores IR relacionando o valor	90
0.2	no sensor e a distância real ao objeto detectado	98
6.3	Tela de saída da bengala eletrônica detectando um objeto. Os valores	- 0
	nas colunas mostram as distâncias, em metros, dos sensores ao objeto,	
	tomadas em vários momentos	100

# Lista de Tabelas

3.1	Família de dispositivos da ALTERA	45
4.1	Endereços das portas de E/S utilizadas pela interface	56
4.2	Descrição do registro de estados do sistema de gerenciamento da interface	58
4.3	Relação dos endereços dos registros de dispositivos internos da interface SIGA	61
5.1	Parâmetros dos motores utilizados no Veículo Autônomo necessários para os cálculos de odometria	86
5.2	Funções de conversão dos dados de posição, velocidade e aceleração dos motores	90
6.1	Dados de calibração dos sensores IR contendo os valores obtidos pelos sensores (vs) em diversas distâncias e seus respectivos desvios padrão	
	(dp)	97

# Listas de Abreviaturas e Siglas

AGV Automatic Guided Vehicle

AHDL ALTERA Hardware Description Language
ASIC Application Specific Integrated Circuits

CC Corrente Contínua CI Circuito Integrado

CMOS Complementary Metal-Oxide Silicon CPLD Complex Programmable Logic Device

 $\begin{array}{lll} {\rm DMA} & {\it Direct\ Memory\ Access} \\ {\rm DOS} & {\it Disk\ Operational\ System} \\ {\rm DSP} & {\it Digital\ Signal\ Processing} \\ {\rm EDA} & {\it Electronic\ Design\ Automation} \\ {\rm EPLD} & {\it Erasable\ Programmable\ Logic\ Device} \end{array}$ 

E/S Entrada e Saída

 $\begin{array}{ll} {\rm FPGA} & {\it Field-Programmable \ Gate \ Array} \\ {\rm FPLD} & {\it Field \ Programmable \ Logic \ Device} \end{array}$ 

GCC GNU Compiler Collection

GNU Gnu's not Unix

GPS Global Positioning System
HLD Hardware Description Language

ICBM Interface Controladora de um Braço Mecânico

IR Infrared

ISA Industry Standard Architecture

LAB Logic Array Block
LSI Large-Scale Integration

MOSFET Metal-Oxide Silicon Field Effect Transistor

MSI Medium-Scale Integration

NCE Núcleo de Computação Eletrônica

PAL Programmable Array Logic

PC Personal Computer

PCI Peripheral Component Interconnect
PIA Programmable Interconnect Array
PLA Programmable Logic Array
PLD Programmable Logic Device
PSD Position Sensitive Detector

PWM Pulse Width Modulation

PXI PCI Extensions for Instrumentation

SIGA Sistema de Interfaceamento Global do AGV

SOP Sum Of Products

SPLD Simple Programmable Logic Device

SSI Small-Scale Integration

TOF Time Of Flight

TTL Transistor-Transistor Logic

UFRJ Universidade Federal do Rio de Janeiro

USB Universal Serial Bus

VHDL Very High Speed Integrated Circuit Hardware Description Language

VLSI Very Large-Scale Integration

# Capítulo 1

# Introdução

# 1.1 Requisitos impostos pelos sistemas do Veículo Autônomo à aquisição de dados

O projeto CONTROLAB (AUDE, 1995) consiste em um laboratório de controle que utiliza sistemas inteligentes que são desenvolvidos para trabalhar de forma autônoma, porém cooperando entre si. Estes sistemas são baseados em equipamentos do tipo PC compatíveis e desenvolvidos em linguagem C.

Atualmente o projeto CONTROLAB está desenvolvendo um projeto de Veículo Autônomo (CONTROLAB-AGV) (AUDE, 1999a), que utiliza a planta baixa representando o ambiente em que ele se encontra e sistemas inteligentes para a navegação neste ambiente. Através da rede Internet, ele recebe comandos para se deslocar no ambiente especificado pelo mapa. Um sistema de visão em cooperação com um sistema de sensores de ultra-som e sensores de infravermelho detecta em tempo real obstáculos no trajeto do veículo. Os sistemas de controle e de orientação definem a trajetória a ser seguida. Como visto, este projeto consiste de diversos sistemas inteligentes, que em conjunto, determinam a finalização de uma tarefa.

Estes sistemas inteligentes necessitam de diversos tipos de sensores para perceber

o ambiente ao redor do Veículo Autônomo. A determinação dos sensores necessários é função da aplicação que o Veículo Autônomo executará. De qualquer forma estes sensores devem interfacear com uma plataforma de processamento de dados de forma a receber suas informações de forma exata e rápida, para que possam ser analisadas pelos sistemas inteligentes do Veículo Autônomo.

Dada a complexidade que os diversos sistemas inteligentes podem assumir, também se deve otimizar a utilização dos sistemas de processamento enquanto as coletas de informações dos sensores são efetuadas. Isto significa que a aquisição de dados não pode se tornar uma tarefa que demande muito poder computacional da plataforma de processamento. O máximo de pré-processamento deve ser efetuado nos dispositivos que interfaceiam os sensores. Além disso, como se presume que o Veículo Autônomo do CONTROLAB deve funcionar de forma autônoma, inclusive quanto à questão de ter sua fonte de alimentação independente de fontes externas, qualquer dispositivo ou circuito projetado para este ambiente deve consumir o mínimo possível de energia.

### 1.2 Objetivos deste trabalho

O objetivo deste trabalho é projetar uma arquitetura de aquisição de dados de sensores para que o sistema de controle do Veículo Autônomo possa realizar as tarefas comandadas. Esta arquitetura deve permitir que vários tipos de sensores possam ser conectados ao Veículo Autônomo, sendo que atualmente estão operando sensores para determinação de posição e velocidade do Veículo Autônomo e sensores de detecção de obstáculos. Esta arquitetura é denominada interface SIGA.

A interface SIGA será a responsável pela comunicação dos vários subsistemas do Veículo Autônomo com o ambiente. Esta interface é composta de módulos para aquisição de posição e velocidade de motores CC, módulos de atuação em motores

CC, módulos de medição de distância através de ultra-som, e módulos de medição de distância utilizando sensores infravermelhos. Serão implementados ainda módulos para a leitura de dados de dispositivos genéricos que possam futuramente ser adicionados ao Veículo Autônomo.

Considerando os avanços tecnológicos, o projeto desta arquitetura deve permitir fácil migração de plataformas, minimizando o custo de reprojeto caso necessário. Também deve utilizar o estado da arte, evitando o uso de tecnologias que possam ser ultrapassadas em pouco tempo.

Rotinas de leitura e escrita de dados provenientes dos sensores devem ser projetadas para serem executadas de forma independente da plataforma utilizada. Estas rotinas devem permitir a conexão com os sistemas de controle projetados de forma que o usuário/programador dos sistemas de controle tenha o menor esforço na sua manipulação. Também devem permitir a migração destas rotinas para vários sistemas operacionais e plataformas de processamento sem a necessidade de grandes modificações serem efetuadas.

### 1.3 Conteúdo da dissertação

Esta dissertação está organizada em sete capítulos. O capítulo 2 apresenta um resumo dos diversos sensores e atuadores utilizados em robótica, dando ênfase nos sensores que foram escolhidos para a construção do Veículo Autônomo. O capítulo 3 aborda diversas arquiteturas possíveis que foram estudadas para o desenvolvimento da interface com o Veículo Autônomo. O capítulo 4 apresenta a arquitetura escolhida, descrevendo seu funcionamento. No capítulo 5 as rotinas que foram desenvolvidas para uso com a interface são detalhadas. O capítulo 6 discute os resultados experimentais da utilização dos sistemas embarcados no Veículo Autônomo, mostrando seu desempenho em alguns testes. Finalmente o capítulo 7 apresenta as

conclusões e aborda possíveis desdobramentos deste trabalho.

## Capítulo 2

## Sensores e atuadores estudados

### 2.1 Introdução

Um equipamento de robótica, por exemplo, um braço mecânico ou um veículo autônomo, é constituído de uma série de elementos necessários ao seu funcionamento. Em princípio estes equipamentos devem exercer alguma atuação no ambiente em que se encontram baseados devido às reações que o ambiente exerce sobre eles. Para que estas funções sejam realizadas é necessário que os elementos sensores e atuadores sejam minuciosamente escolhidos, já que a performance do equipamento depende da precisão da atuação e da qualidade do sensoriamento do ambiente.

Para a construção de um Veículo Autônomo capaz de se deslocar em um ambiente composto por salas, corredores, porta e paredes, e que ainda possa conter mobiliário espalhado além de objetos móveis, a escolha de atuadores e sensores apropriados para sua locomoção torna-se primordial. Os atuadores devem permitir a possibilidade de ações rápidas, de forma que manobras complexas possam ser executadas. Em contrapartida, os sensores devem captar variações no ambiente de forma rápida e precisa, de forma que decisões possam ser tomadas com precisão. Tanto nos atuadores como nos sensores serão necessários também que, aliada à precisão, haja simplicidade no interfaceamento com os sistemas de processamento de decisão, o

que proporciona baixo custo computacional neste interfaceamento e economia de recursos.

Como na maioria dos veículos autônomos, a proposta do Veículo Autônomo CONTROLAB-AGV é baseada em locomoção através de rodas. O CONTROLAB-AGV é basicamente um triciclo com uma roda na sua frente, que pode ser controlada de forma a orientar o veículo em uma direção e duas rodas traseiras com tração independentes, o que permite o Veículo Autônomo girar em torno de seu eixo.

Este capítulo descreve os tipos de sensores que atendem aos requisitos impostos pelo Veículo Autônomo proposto, bem como faz um estudo de quais sensores são mais adequados a sua implementação. Será descrito o funcionamento básico dos sensores de leitura dos motores utilizados na locomoção do Veículo Autônomo e dos sensores utilizados na detecção de obstáculos.

### 2.2 Sistema de leitura de posição do AGV

O sistema de medição de posição do Veículo Autônomo é baseado na odometria, ou seja, na medida de deslocamento angular de suas rodas. Este é o sistema de medição de posição mais comumente utilizado em veículos autônomos. A vantagem principal deste sistema é sua simplicidade, permitindo que a posição do veículo seja sempre estimada. A desvantagem é que o erro de posicionamento cresce sem limites, necessitando de constantes reavaliações de posição em relação a uma referência fixa. Os sistemas baseados em odometria usam sensores rápidos, o que facilita o tratamento de seus dados.

Outro sistema de medição utilizado é baseado em navegação inercial, onde a posição é estimada com base nas variações de rotação e aceleração. Estas variações são integradas, obtendo-se então o posicionamento do veículo. Os sensores normalmente utilizados são giroscópios e acelerômetros. Estes sensores têm custo elevados

o que se torna uma desvantagem em relação à medição por odometria. Além disso, o erro de posicionamento também cresce sem limites, já que estes sensores apresentam desvios de medida com o tempo, e também por histerese na medida.

Os sistemas de determinação da posição descritos acima são conhecidos como medidores de posição relativa. Devido à facilidade de integração e precisão, estes sistemas são os mais comumente implementados em veículos autônomos. A maior desvantagem destes sistemas de determinação da posição é a necessidade da obtenção da posição de origem inicial do veículo, além da constante obtenção de uma referência externa para a correção da posição real do veículo.

Existem também medidores de posição absoluta para a determinação da posição de veículos. A grande vantagem destes sistemas é a apresentação da posição absoluta real sem a necessidade de reavaliações. Porém, o custo dos equipamentos e a complexidade na obtenção das medidas tornam a utilização de tais sistemas por vezes impraticáveis. Para viabilizar o uso de tais sistemas de medição são necessárias, na maioria das vezes, instalações de equipamentos fora do veículo, o que eleva os custos e restringe o uso para determinadas aplicações e ambientes. Exemplos de medidores absolutos de posição são: posicionamento através da utilização de padrões de imagem, padrões de sinais (active beacon), e GPS (Global Positioning System). No caso específico do GPS, ele necessita visada direta para um conjunto de satélites, o que não é possível para aplicações em ambientes urbanos, no interior de prédios e em ambientes subaquáticos.

Neste capítulo será dada atenção aos sensores de posição relativa, pois os mesmos são mais adequados às características do projeto do Veículo Autônomo CONTROLAB-AGV.

Os sensores mais populares utilizados em equipamentos que necessitam medir sua posição relativa são do tipo *encoder*. Basicamente são dispositivos que geram uma seqüência de pulsos de forma proporcional ao seu movimento, isto é, quanto maior

o deslocamento maior a quantidade de pulsos gerados. Atualmente, a maioria dos equipamentos utiliza encoders para a obtenção das medidas de posição e velocidade, sendo os de tipo ópticos os mais comumente encontrados. Existem outros tipos de encoders (BORENSTEIN, 1996a), como magnéticos, indutivos e capacitivos, que não serão abordados neste trabalho, devido tanto aos seus custos e disponibilidades, quanto a características específicas.

#### 2.2.1 Encoders ópticos

Os encoders ópticos são sensores destinados a obter medidas de posição e velocidade. Normalmente são encoders rotativos, que fazem medidas de ângulos, apesar de existirem encoders lineares para medidas de comprimento. Os encoders rotativos ópticos, são compostos basicamente por um disco rotativo com partes opacas e transparentes codificadas em sua superfície, e um sistema óptico formado por um ou mais dispositivos emissores de luz, e outros receptores direcionados para as partes transparentes e opacas. Este disco é acoplado ao eixo no qual se deseja medir o deslocamento. Conforme o disco é posto para girar, as partes transparentes/opacas do disco periodicamente interrompem ou deixam passar a luz emitida. Esta luz é então coletada no receptor de luz, gerando um padrão de pulsos elétricos relacionados a este processo de interrupção e não interrupção. Este padrão de pulsos representa o deslocamento do disco e é relacionado com a posição e velocidades angulares do eixo em questão (Figura 2.1). Os encoders ópticos podem ser divididos em duas categorias: encoders incrementais e encoders absolutos.

O encoder óptico incremental é composto por um disco codificado em padrões de superfície transparentes e opacas. Um emissor de luz é colocado em um dos lados desta superfície enquanto no outro lado um fotodetector é alinhado ao emissor de luz, de tal maneira que, quando a superfície transparente está entre o emissor e o detector, a luz é capaz de atravessá-la. Uma tensão é gerada no fotodetector e

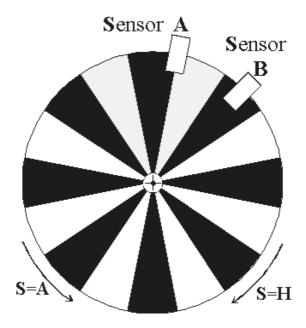


Figura 2.1: Encoder óptico incremental com dois canais (A e B).

utilizada para acusar movimento no disco. Enquanto o disco roda, uma seqüência de tensões altas e baixas são geradas, indicando que o mesmo está se deslocando.

Quando o encoder tem apenas um dispositivo emissor/detector ele é chamado de encoder de um canal. Este encoder serve apenas para medir velocidade já que não consegue discriminar a direção do movimento, mas somente a variação do movimento. Este problema é resolvido com a adição de mais um conjunto emissor/detector, formando um segundo canal. Este segundo canal é colocado de tal forma que o sinal gerado tem uma defasagem de 90° em relação ao sinal do primeiro canal, conforme ilustrado na Figura 2.1. A saída do encoder é mostrada na Figura 2.2. Uma decodificação destes sinais pode determinar a direção da rotação. Desta forma a posição do motor pode ser determinada, incrementando ou decrementando a posição de acordo com a direção discriminada. Este tipo de encoder é chamado de encoder de quadratura de fase. Uma forma de implementação desta decodificação pode ser vista adiante no capítulo 4.

Uma característica evidente dos encoders de quadratura de fase é a necessidade de uma referência externa para definição da origem de posição. Esta referência

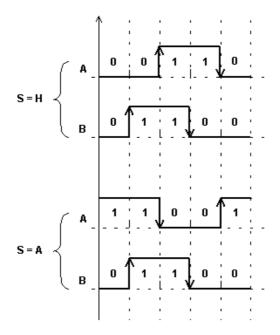


Figura 2.2: Saída do encoder incremental, em quadratura de fase, mostrando o sinal para sentido horário (H) e anti-horário (A).

externa é obtida através da adição de um novo conjunto emissor/detector, ou seja, um terceiro canal, também chamado de índice ou origem. Apesar da inclusão deste índice solucionar o problema de obter a referência, um novo problema é acrescentado, que é a necessidade de realização do procedimento de procura da referência ou **reset**. No caso específico de movimentos repetitivos de ir para frente e para trás basta acionar o motor até que o sensor de índice seja encontrado.

Os encoders ópticos absolutos diferem dos encoders incrementais pelo fato de não necessitarem do canal de índice. Os encoders absolutos têm a todo instante a sua posição absoluta, mesmo no caso de interrupção de energia. Esta característica confere aos encoders absolutos a desnecessidade de procedimento de busca de referência, simplificando o sistema de medida.

Estes encoders têm várias trilhas concêntricas, uma para cada bit de resolução (Figura 2.3). Em cada trilha há um conjunto emissor/detector que fornece os sinais de posicionamento. Estes conjuntos são dispostos de tal forma que para cada posição do encoder há uma única codificação possível. A codificação normalmente utilizada

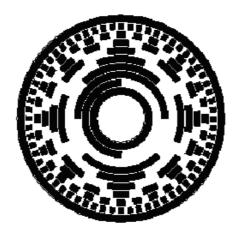


Figura 2.3: Exemplo de encoder óptico absoluto de oito *bits* codificado em código Gray.

é o código Gray que tem como principal característica a alteração de apenas um bit em relação a qualquer vizinho da posição anterior, o que diminui significativamente o ruído elétrico. Ao contrário dos encoders incrementais, em que a saída deve ser processada para obter a variação de posição do eixo, os encoders absolutos fornecem diretamente a medida de posição do eixo.

Este encoder deve ser aplicado quando há interesse na medida da posição angular em relação a uma volta do eixo. Quanto maior for a precisão desta medida maior deve ser o número de trilhas, tornando cada vez mais difícil a sua confecção, elevando o preço final.

Apesar da grande vantagem que os encoders apresentam, algumas considerações devem ser feitas. Apesar da precisão da medida de rotação dos motores ou das rodas a eles conectadas, a utilização de encoders implica que quaisquer problemas ligados ao contato com as rodas no solo impliquem em resultados erráticos.

As folgas existentes entre os acoplamentos mecânicos dos motores e rodas também são responsáveis por erros imprevisíveis na medição de posição e velocidades do Veículo Autônomo. Quando o encoder é colocado no eixo do motor, a precisão na medida do movimento de rotação das rodas será aumentada pela razão de redução da caixa de acoplamento das rodas, desta forma podendo utilizar um encoder de

menor resolução. Neste caso, porém a folga existente na caixa de acoplamento irá gerar erros na medida de deslocamento das rodas. Quando o encoder é colocado diretamente no eixo das rodas, o movimento real das rodas será o mesmo movimento do encoder, entretanto neste caso um encoder de maior resolução se tornará necessário, elevando o custo o projeto.

Medidas erradas dos diâmetros das rodas, ponto de contato incerto com o solo e desalinhamemto das rodas em relação ao eixo são erros classificados como sistemáticos. Estes erros são previsíveis, porém acumulam-se com o tempo. Em solos regulares podem ser a principal fonte de erros, se não corrigidos.

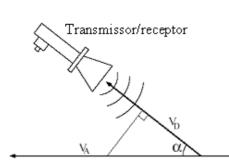
Escorregamentos traduzem deslocamentos das rodas que não implicam em movimento do Veículo Autônomo. De outra forma, deslizamentos implicam em movimento do Veículo Autônomo sem rotação das rodas. Estes erros são classificados como erros não sistemáticos. Outros erros não sistemáticos são imperfeições no solo e obstáculos inesperados sob as rodas. De forma geral estes erros são imprevisíveis e dominantes em solos irregulares.

A existência de tais erros força a necessidade do uso de outros tipos de sensores de posição.

### 2.2.2 Sensores com Doppler

Os sensores que utilizam o efeito Doppler baseiam-se no deslocamento de freqüência da onda emitida, que é observado quando a mesma é refletida por um obstáculo em movimento em relação ao emissor. Uma implementação típica pode ser observada na Figura 2.4.

Erros na medição podem surgir devido a irregularidades do solo e componente vertical da velocidade. Um fato curioso é quando o veículo estiver parado sobre um curso d'água. O efeito Doppler sobre o curso d'água faria uma falsa detecção de



$$V_A = \frac{V_D}{\cos \alpha} = \frac{cF_D}{2F_0 \cos \alpha}$$

Onde:

 $V_A$  — V elocidade real do chão  $V_D$  — V elocidade D oppler medida

α – Ângulo de inclinação

c -velocidade da luz

 $oldsymbol{F_D}$  — Deslocamento da freqüência observada

 $F_{\mathcal{O}}$  – Freqüência transmitida.

Figura 2.4: Diagrama de funcionamento de medidor de velocidade com utilização de efeito Doppler.

velocidade do veículo. Apesar do mesmo estar parado, o sensor mediria a velocidade do curso d'água e não a velocidade do veiculo.

#### 2.2.3 Acelerômetros, giroscópios e bússolas

Acelerômetros suprem informação sobre a aceleração de um objeto. De forma geral têm a relação sinal-ruído baixa para medir baixar acelerações. Também são muito sensíveis à irregularidades do solo, já que pequenas irregularidades mediriam a aceleração gravitacional. Adicionado a estes problemas, um alto drift é encontrado, mesmo em sistemas mecanicamente compensados. Drifts de até oito cm/s são encontrados em alguns tipos de sensores (BARSHAN, 1993), o que torna seu uso intolerável para Veículo Autônomo pequenos.

Giroscópios fornecem informação de velocidade angular. Os giroscópios são basicamente classificados em mecânicos e ópticos. Os giroscópios mecânicos são influenciados pela rotação da Terra, sendo seu valor dependente da posição que o mesmo se encontra em relação aos pólos da Terra, além de terem vários problemas de manutenção mecânica.

Os giroscópios ópticos têm poucas e eventualmente nenhuma parte móvel, além de serem insensíveis ao campo gravitacional. Também têm um baixo drift se comparados aos giroscópios mecânicos, o que os tornam mais atrativos. O maior problema

com os giroscópios são seus custos, o que os tornam impraticáveis em muitas implementações. Também necessitam de uma base estável para evitar que medidas erráticas sejam obtidas.

Giroscópios e acelerômetros são intensamente utilizados em navegação inercial. As medidas de velocidade e aceleração são obtidas diretamente. Já a posição pode ser obtida através da integração dos resultados de medidas da velocidade angular feita pelos giroscópios. Para o acelerômetro é necessário a dupla integração dos resultados. Isso significa que pequenos erros na medida podem causar erros relativamente grandes na obtenção da posição, o que é a principal desvantagem no uso destes sensores.

Bússolas informam o posicionamento angular tomando como base o pólo magnético da Terra. São divididas em várias categorias, sendo as principais: as magnético-mecânicas, fluxgate e as de efeito Hall. A principal causa de erro no uso de bússolas é a interferência causada por fontes de alimentação, campos magnéticos e estruturas de ferro próximas aos sensores. Estas interferências são difíceis de evitar por serem imprevisíveis no caso de um Veículo Autônomo que se movimenta por um ambiente desconhecido.

O uso simultâneo de giroscópios e acelerômetros combinados com medição de movimento das rodas pelos encoders (odometria) pode ser utilizado para melhorar os resultados de medidas de posição (BORENSTEIN, 1996b). Este método de fusão de sensores utiliza modelos estatísticos para representar o comportamento dos equipamentos de medida.

Neste trabalho não será abordado um estudo mais detalhado destes dispositivos, já que não fazem parte da implementação inicial do Veículo Autônomo.

### 2.3 Sensores de detecção de obstáculos

A detecção de obstáculos é determinante no projeto de um Veículo Autônomo que pretende desviar de objetos espalhados no ambiente. Existem vários sensores capazes de "sentir" o ambiente de forma a reconhecer se um objeto situado na trajetória de um Veículo Autônomo se trata de um obstáculo. O sensor mais apropriado dependerá da aplicação e do ambiente em que o veículo estiver inserido.

De forma geral os detectores de obstáculos podem ser classificados em duas categorias: detectores que emitem um sinal e esperam a reação do objeto a este sinal, chamados de sensores ativos; e sensores que medem o ambiente de forma a reconhecer sinais que caracterizam objetos, chamados de sensores passivos.

Como exemplos de sensores passivos podem ser citados os que reconhecem objetos através de imagens ou detectores de campo magnético, neste caso tratando-se de obstáculos sabidamente metálicos. No caso do primeiro exemplo, um grande poder computacional se faz necessário para tratar o reconhecimento de objetos, o que torna este sensor restrito a aplicações específicas e de baixa velocidade. No segundo exemplo, o objeto a ser detectado deve ter determinadas características, o que não é típico do mundo real.

Como sensores ativos, podem ser citados os que utilizam os medidores de distância baseados em tempo de vôo (time of flight - TOF) ou os que utilizam triangulação. Os TOFs são os sensores que medem a distância ao objeto baseado no tempo que o sinal emitido leva para percorrer esta distância. Nesta categoria encontram-se os sensores ultra-sônicos, infravermelhos e laser. Os que utilizam a triangulação para obter a distância emitem um sinal ao obstáculo e determinam o ponto onde o sinal emitido foi recebido. A distância entre o ponto da recepção e o emissor formam a base de um triângulo cuja altura é a distância a ser determinada. Sensores que utilizam laser e infravermelho podem ser utilizar este método de avaliação de distância.

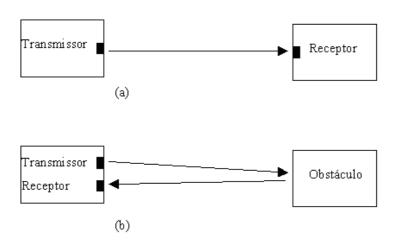


Figura 2.5: Propagação do pulso de ultra-som (a) emissor e sensor em pontos diferentes (b) emissor e sensor no mesmo ponto.

Os sensores ativos serão estudados neste trabalho. Por precisarem de recursos computacionais ou serem apropriados para trabalhar em ambientes específicos os sensores passivos não serão estudados neste trabalho.

#### 2.3.1 Descrição do sistema de ultra-som

A medição de distância utilizando sensores de ultra-som e infravermelho utilizam o mesmo princípio. Este princípio é o de medir o tempo que um sinal (pulso) emitido leva para percorrer uma distância entre o agente transmissor e um agente receptor. Os sistemas que utilizam estes princípios são conhecidos como tempo de vôo (TOF). O agente receptor pode estar localizado ou no ponto em que se quer medir ou no mesmo ponto que se encontra o agente transmissor, sendo que neste último caso a distância percorrida pelo pulso emitido é igual ao dobro da distância que efetivamente deseja-se medir. Estes esquemas estão ilustrados na Figura 2.5. No caso específico de Veículo Autônomo a segunda opção é a mais utilizada, já que nesta forma não há necessidade de colocar transdutores no ambiente em que o Veículo Autônomo se encontra.

O tempo t que o pulso emitido leva para percorrer a distância entre o transmissor

e o receptor é proporcional a distância d entre estes e dependente do meio utilizado para a transmissão. Considerando v como a velocidade de propagação do som no meio tem-se que para o caso (a) da Figura 2.5 a distância é dada por:

$$d = v * t$$

e para o caso (b) da Figura 2.5 a distância é dada por:

$$d = \frac{v * t}{2}$$

O ultra-som é da mesma natureza do som audível, isto é, é a propagação de ondas mecânicas através de um meio físico, diferenciando-se apenas pela faixa de freqüência, abrangendo freqüências acima de 20 kHz. Freqüências comumente utilizadas para medidas de distância no ar estão na faixa entre 40 e 50 kHz.

Para medidas de distância no ar, a velocidade de propagação  $v_T$  do ultra-som a temperatura de 25 °C é de aproximadamente 346,15 m/s. A velocidade do som  $v_T$ , em metros por segundo, muda com a temperatura T, em graus Celsius, de acordo com a fórmula (SHIRLEY, 1989):

$$v_t = v\sqrt{1 + \frac{T}{273}}$$

Onde  $v_0 = 331,3176 \text{ m/s},$  é a velocidade do som a 0 °C.

Os transdutores de ultra-som para aplicação de medidas de distância podem ser de dois tipos:

• piezelétricos - um cristal apresenta deformação mecânica quando submetido uma tensão elétrica e vice-versa.

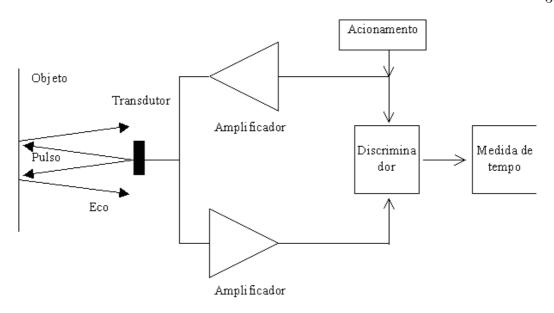


Figura 2.6: Esquema básico de sensor de ultra-som. O transdutor utilizado neste esquema funciona como emissor e receptor de ultra-som.

• eletrostáticos - duas placas metálicas paralelas, semelhantes a um capacitor, têm sua distância modificada quando submetidas a uma tensão elétrica.

Os transdutores piezelétricos oferecem menor potência e com isso, medem distâncias menores, porém são de acionamento mais fácil. Já os eletrostáticos oferecem maior potência acústica, oferecendo maiores distâncias. Porém seu acionamento requer circuitos mais complexos.

Pode-se utilizar um sistema com um transdutor servindo de emissor de ultra-som e outro servindo de receptor. Existe a possibilidade de apenas um transdutor ser utilizado, servindo este de emissor e receptor. O esquema apresentado na Figura 2.6 pode ser montado utilizando-se um único transdutor para a transmissão/recepção do pulso de ultra-som.

Um grande problema com a utilização de ultra-som para medição de distância é a largura do cone de emissão. Um cone muito largo implica que parte do sinal está sendo emitida para regiões não desejadas. Estes fatos aliados à absorção do sinal pelos objetos resultam na baixa intensidade do sinal retornado no transdutor. Esta

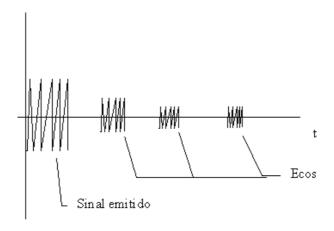


Figura 2.7: Forma de onda dos pulsos de ultra-som. Os vários pulsos de eco referemse a reflexões de objetos em posições diferentes.

intensidade ainda é função da distância percorrida pelo sinal, tanto menor quanto maior a distância. Os amplificadores de recepção do sinal devem ter um ganho alto, e normalmente este ganho deve ser programável em função do tempo, uma vez que quanto maior o tempo desde a emissão do pulso de ultra-som, maior é a distância percorrida e, conseqüentemente, maior a atenuação do sinal.

Normalmente utilizam-se 40 kHz ou 50 kHz como freqüência para a medição de distância. Emite-se então um pulso de duração entre 4 a 20 ciclos na freqüência utilizada. Após o eco deste pulso num obstáculo, um pulso é recebido, amplificado e é então medido o tempo decorrido entre este pulso de eco e o pulso anteriormente emitido. Os sinais podem se comportar da seguinte forma (Figura 2.7).

Dentre as vantagens do uso deste tipo de sensor vale ressaltar que as medidas de distâncias se mantêm com a mesma precisão em toda a faixa de distância desde que o sinal de eco seja detectado, ao contrário dos sensores que utilizam triangulação, que perdem precisão conforme a distância aumenta.

Como desvantagens deste método pode-se citar a sensibilidade com a velocidade da onda no meio, que no caso do ultra-som é dependente da temperatura ambiente. O ângulo de incidência do pulso de ultra-som com o obstáculo também é motivo de erros, já que a falta de incidência normal causa a atenuação do sinal de eco e pode

gerar inclusive reflexos no sinal, causando trajetórias erráticas e consequentemente medidas erradas.

#### 2.3.2 Descrição do infravermelho

A luz infravermelha faz parte do espectro de radiação eletromagnética, abaixo da faixa do vermelho visível, tendo comprimento de onda situado entre 650 nm a 1  $\mu$ m. Dado a natureza eletromagnética, a velocidade de propagação do infravermelho é a da velocidade da luz,  $c=2,998*10^8$  m/s. Como valor prático utiliza-se  $c=3*10^8$  m/s

Nesta velocidade percebe-se que a medida de tempo deveria ser muito precisa, visto que o pulso de infravermelho percorreria 1 metro em apenas 3,3 ns. Para medidas com precisão de centímetros precisaríamos de um relógio com precisão de 33 ps.

Utiliza-se então uma técnica alternativa para medição da distância que é o deslocamento de fase (TEXAS, 1978). Através desta técnica não se medirá simplesmente o tempo entre o pulso emitido e o recebido, mas sim quanto tempo o pulso recebido está defasado do pulso emitido. Utiliza-se o circuito esquematizado na Figura 2.8 para medir o deslocamento de fase  $\varphi$ .

O transmissor de infravermelho é modulado em uma determinada freqüência com grande precisão. Supondo uma freqüência f de 4,433 MHz como exemplo. Tem-se com tempo t para percorrer uma distancia d dado por:

$$t = 2 * \frac{d}{c}$$

e a fase  $\varphi$  é dada por:

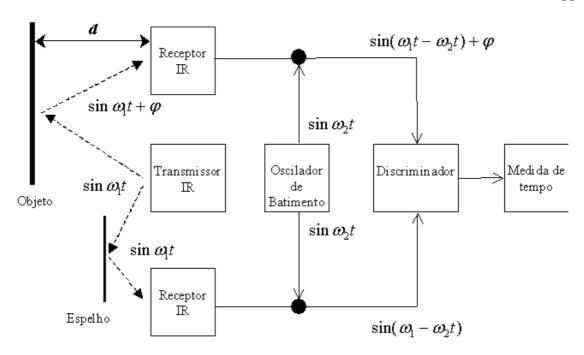


Figura 2.8: Diagrama de medidor de distância utilizando infravermelho através da medida do deslocamento de fase  $\varphi$  de um pulso de refletido de infravermelho.

$$\varphi = 2 * \pi * f * t$$

substituindo t obtém-se:

$$\varphi = \frac{4 * \pi * d * f}{c}$$

Supondo um deslocamento de fase de 0,25 graus e utilizando as fórmulas anteriores, obtém-se d=2,1 cm e t=138 ps. Este intervalo de tempo ainda é muito pequeno para ser medido.

Utiliza-se então um oscilador de batimento com frequência ligeiramente maior que a frequência do transmissor (por exemplo 80 Hz). Neste caso, através da multiplicação deste dois sinais (isto é, misturando-se os sinais), obtém-se:

$$seno\varphi_{1}*\ seno\varphi_{2}=\frac{seno\left(\varphi_{1}+\varphi_{2}\right)}{2}+\frac{seno\left(\varphi_{1}-\varphi_{2}\right)}{2}$$

onde:

$$\varphi=2*\pi*f_1$$
 , onde  $f_1$  é 4,433 MHz e 
$$\varphi=2*\pi*f_2 \text{ , onde } f_2 \text{ é 4,43308 MHz}.$$

de um filtro passa-baixa) e uma freqüência baixa de 80 Hz.

Nota-se que agora obtém-se uma freqüência alta (que pode ser descartada através

Supondo o mesmo deslocamento de fase de  $0.25^{\circ}$  e utilizando as fórmulas anteriores para a freqüência de 80 Hz, obteríamos d=2,1 cm e t=8,6  $\mu$ s, que é um tempo que pode ser facilmente medido.

No caso da implementação deve-se montar dois circuitos receptores para que o atraso na propagação dos sinais pelo circuito sejam iguais tanto do sinal refletido pelo objeto como o sinal refletido pelo espelho (referência).

Outro fato a ser relevado refere-se a maior distância que pode ser medida. Quando o sinal estiver defasado de 180°, para o exemplo esta distância seria de 16,9 m.

### 2.3.3 Descrição da bengala

A "bengala eletrônica" utilizada pelo CONTROLAB-AGV foi implementada através de um medidor de distância capaz de detectar obstáculos que estejam na frente do veículo. O alcance desta "bengala" pode ser pequeno, no caso do Veículo Autônomo proposto compreendido entre 20 cm e 80 cm, formando um ângulo de 30° (Figura 2.9), utilizando para isto, sensores infravermelho. É indicado o uso do sensor da Sharp GP2D02 (SHARP, 2003), que já incorpora o transmissor e receptor IR, o sistema ótico, filtros, circuitos detectores e amplificadores.

O sensor GP2D02 é classificado como PSD (*Position Sensitive Detector*), ou seja, um detector de posição. O seu funcionamento baseia-se na triangulação, conforme

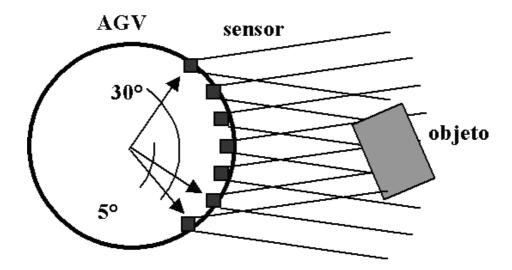


Figura 2.9: Implementação da bengala eletrônica no Veículo Autônomo composta de sete sensores GP2D02.

pode ser observado na Figura 2.10. Um pulso de luz infravermelha é emitido por um emissor infravermelho que tem uma lente colimadora. A luz é então refletida pelo objeto postado a sua frente e recebida pelo sensor PSD. Este dispositivo é um sensor que tem um sinal elétrico de saída cuja intensidade é proporcional à posição que a luz incidente é recebida. Este sinal é então convertido em um valor binário de oito bits e disponibilizado na saída de forma serial para que possa ser lido.

A utilização deste sensor oferece as seguintes vantagens:

- Imunidade ao nível de luz ambiente, independente da fonte (fluorescente, incandescente ou natural).
- Cor e reflexibilidade dos objetos são praticamente irrelevantes.
- Imunidade às várias fontes de IR externa.
- Baixo consumo, quando comparado com medidores de distância por ultra-som.
- Fácil interfaceamento com circuitos digitais.

O sistema de acionamento e leitura de sensores infravermelhos (IR) tem a função

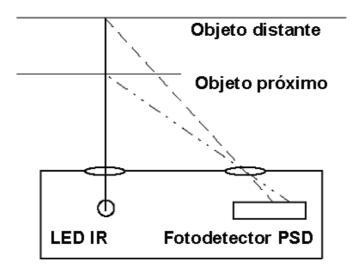


Figura 2.10: Funcionamento da triangulação no sensor GP2D02.

de medir distâncias curtas, situadas entre 20 cm e 80 cm. É através deste sistema que está sendo implementada uma "bengala eletrônica".

Este sensor projeta um cone de visão cuja largura da base varia conforme a distância ao objeto. Para objetos situados a 40 cm, a largura da base é de aproximadamente 10 cm de um extremo ao outro, o que representa um cone de aproximadamente 10°.

Uma limitação já encontrada nestes dispositivos sensores é que cada dispositivo apresenta resposta de saída distinta. Esta limitação pode ser corrigida estabelecendo-se um procedimento de calibração para o dispositivo sensor. Pretende-se utilizar procedimentos estatísticos, ainda a serem definidos, para caracterizar a função de saída, e assim calibrar o dispositivo sensor. Este procedimento deverá ser usado para cada um dos sensores utilizados.

## Capítulo 3

## Análise de arquiteturas

#### 3.1 Introdução

Uma interface que visa aquisição de dados de sensores tem o propósito fundamental de fornecer meios para que os sinais dos sensores conectados possam ser enviados e entendidos pela plataforma que suporta estes dispositivos que no caso específico deste trabalho é o Veículo Autônomo.

A seguir algumas alternativas de projeto e suas características principais são descritas:

- Uma interface baseada no uso de circuitos integrados TTL LSI na implementação dos módulos;
- A utilização de microcontroladores programáveis;
- O uso de EPLDs (*Erasable Programmable Logic Devices*) na implementação dos módulos.

A primeira opção, o uso de circuitos integrados TTL LSI, foi utilizada no projeto e implementação de uma interface, conhecida como ICBM (Interface Controladora de um Braço Mecânico) no ambiente CONTROLAB, que tinha como objetivo a

leitura dos sensores e acionamento dos motores que compunha o braço mecânico (SERDEIRA, 1991). Esta interface permitia a atuação de até seis motores CC, além da aquisição de dados de posição e velocidade de até seis encoders incrementais. Permitia ainda o comando e leitura de dispositivos simples do tipo liga-desliga. Basicamente esta interface era um sistema de atuação e leitura de posição de motores CC. A vantagem desta interface era a sua versatilidade, já que por ser composta de elementos discretos oferecia maior liberdade de configuração. Apesar de mostrar ser satisfatória no ambiente proposto, esta interface tinha como desvantagem o grande tamanho e o consumo elevado, detalhes oriundos da implementação com TTL LSI. Outra desvantagem relevante foi a enorme dificuldade para realizar alterações, seja devido a erros de projeto, ou por ampliações ou reestruturações da interface.

O uso de microcontroladores na implementação, conforme a segunda opção, facilita a manutenção da interface, além de diminuir seu tamanho. Pode-se dizer que cada microcontrolador implementa um ou mais módulos e que estes módulos são ligados ao PC através de um barramento próprio. Tem como principal desvantagem a velocidade que algumas funções podem requerer, o que implica na utilização de microcontroladores específicos. Uma vantagem interessante na utilização de microcontroladores é que funções mais complexas podem ser implementadas na própria interface, liberando o PC para outros processamentos.

Finalmente, o uso de EPLDs, oferece as vantagens de ambas as soluções anteriores. O projeto utilizando EPLDs tem tamanho reduzido além de oferecer flexibilidade de configuração.

A escolha por EPLDs do sistema ALTERA (ALTERA, 1996) para o projeto e implementação desta interface foi baseada, não só pelas suas características atraentes como disponibilidade comercial e preços, mas também pelo seu ótimo desempenho na sua utilização pelo projeto MULTIPLUS (AUDE, 1996), grupo de trabalho do NCE/UFRJ, responsável pelo desenvolvimento de um computador paralelo.

#### 3.2 Evolução dos circuitos integrados

Na resolução de problemas em computação deve-se tomar a decisão de qual é a melhor solução a ser implementada. No modelo atualmente utilizado a decisão reside em utilização de *hardware* e de *software* para a realização desta tarefa. Depois de definido um algoritmo, que deve ser implementado para a solução do problema, a análise de quais pontos são críticos quanto à performance desejada deve ser feita.

Quanto ao uso de *software*, enumeram-se as seguintes características:

- Flexibilidade proporcionada pela programação rápida do algoritmo desejado;
- Portabilidade para outro ambiente mais rápida, se utilizado linguagem de alto nível;
- Lentidão devido à execução sequencial do algoritmo implementado.

Ao hardware suas características seguem:

- Rapidez, uma vez que não depende de execução de instruções e permite que tarefas sejam executadas em paralelo;
- Pouco flexíveis, já que são projetados para uma utilização específica.

Na linha de projetos de circuitos digitais, uma grande evolução surgiu nos últimos anos. Inicialmente os circuitos eram projetados com transistores discretos, não oferecendo nenhuma flexibilidade ao projeto, além de ter grande consumo e tamanho. O desenvolvimento na manufatura de circuitos permitiu a criação de circuitos SSI (Small-Scale Integration), MSI (Medium-Scale Integration) e LSI (Large-Scale Integration), onde vários transistores eram colocados em uma pastilha semicondutora desempenhando uma função básica (HAMBLEM, 2001). Nesta classe encontram-se

os circuitos integrados TTL da série 74xx e CMOS da série 4xxx, que permitiram um grande avanço em projetos de circuitos digitais, onde era necessário que o projetista conectasse vários desses circuitos integrados para realização do projeto. Nesta fase houve drástica diminuição do consumo e tamanho dos projetos, porém modificar o projeto ainda tornava-se uma tarefa árdua.

A tecnologia de circuitos integrados VLSI (Very Large-Scale Integration) permitiu que circuitos mais complexos fossem colocados dentro de uma única pastilha, ainda com projetos configurados somente em tempo de manufatura. O desenvolvimento de ferramentas EDA (Electronic Design Automation) e o barateamento dos processos de fabricação de circuitos integrados permitiram que o usuário desenvolvesse seus próprios projetos de chips conduzindo o desenvolvimento de circuitos de acordo com suas necessidades. A rapidez no desenvolvimento de circuitos integrados é devido também a difusão das linguagens descritivas de hardware, conhecidas como HDL (Hardware Description Language), permitindo que houvesse a padronização de projetos. Esta padronização permite adaptações mais rápidas do projeto à migração para novas tecnologias.

A Figura 3.1 mostra a classificação dos circuitos integrados digitais segundo a sua forma de implementação. A família de circuitos de unidades lógicas padrões (Standard Logic) é aquela definida por circuitos integrados SSI e MSI, que definiam funções básicas em cada chip, exigindo que fossem conectados externamente para que executassem funções mais complexas. Atualmente é utilizado em prototipagem de pequeno porte e para interfaceamento de circuitos.

Os circuitos que necessitam de processo especiais de fabricação para a aplicação desejada são denominados totalmente customizados (Full Custom). Normalmente os circuitos são projetados ao nível de transistor. Por serem específicos, o tempo de desenvolvimento destes CIs associado com o custo de manufatura ainda são um obstáculo em projetos que necessitam prototipagem rápida e poucas unidades. Este

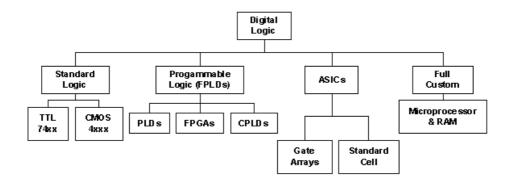


Figura 3.1: Classificação das tecnologias digitais nos circuitos integrados.

processo normalmente só é utilizado quando o tempo do produto no mercado e a quantidade unidades produzidas justificam seus custos. A maior vantagem na utilização deste processo está na alta performance que o produto final pode alcançar.

O processo denominado ASIC (Application Specific Integrated Circuits) é semelhante ao full custom quanto à especificidade do projeto, porém são utilizados conjuntos lógicos padronizados para sua confecção. Pode conter um conjunto de células lógicas (gate arrays) implementando funções básicas (portas lógicas e flip-flops) ou conter células padrões (standard cells) compostas de memórias, controladores e até microprocessadores. Neste processo ainda é exigido projeto personalizado, envolvendo custo e tempo elevados. Também é exigido do projetista um grande conhecimento das várias etapas do processo (WESTE, 1985). Tanto no caso de full custom quanto nos ASICs, ao tempo de projeto soma-se o tempo de testes e de depurações do produto.

Finalmente as lógicas programáveis FPLDs (Field Programmable Logic Devices) formam o último grupo de circuitos integrados digitais. Estes dispositivos permitem que o projeto seja implementado em um dispositivo programável pelo usuário, não necessitando o processo de fabricação do circuito. Normalmente os dispositivos podem ser reprogramáveis, tornando desta forma eventuais mudanças no projeto original muito mais fácil. Esta característica das PLDs tornam a implementação de

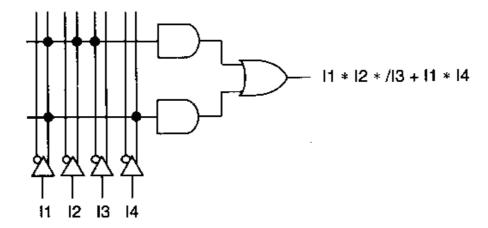


Figura 3.2: Esquema interno de uma PAL.

projetos mais curtas e com custos mais baixos. A principal desvantagem da utilização deste processo de projeto de circuitos é que a performance final do circuito normalmente é menor que projetos full custom ou ASICs. Porém em projetos onde o número de unidades produzidas é menor e que exigem modificações durante a vida do produto é uma opção interessante. As FPLDs podem ser classificas em SPLDs (Simple Programmable Logic Devices), CPLDs (Complex Programmable Logic Devices) e FPGAs (Field-Programmable Gate Arrays) (BROWN, 1996).

As SPLDs normalmente consistem em dispositivos que utilizam funções NOT, AND, OR básicas. As entradas das portas AND estão configuradas de tal forma que podem ser programadas, assumindo entradas normais ou invertidas das entradas do circuito. As saídas nos ANDs são então conectadas a entradas de ORs. A saída destes ORs compõem a saída do circuito final. Este configuração permite que mintermos, ou seja, uma soma de produtos (SOP) das entradas possa ser produzida. Exemplos destes dispositivos são PLAs (*Programmable Logic Arrays*) e PALs (*Programmable Array Logics*). A Figura 3.2 mostra a representação de uma PAL (MONOLITHIC, 1983). A principal vantagem das SPLDs é o baixo custo, porém a necessidade de circuitos com lógicas muito complexas e várias entradas esgota rapidamente o espaço e a quantidade de funções implementadas diminui consideravelmente.

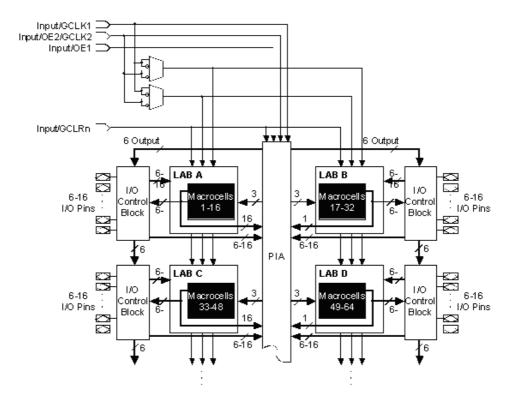


Figura 3.3: Diagrama de uma CPLD da série MAX7000 da ALTERA.

Os CPLDs são basicamente dispositivos que implementam várias SPLDs e uma malha de interconexão que pode interligá-las. As SPLDs contidas numa CPLD podem executar funções mais complexas do que simples SOP, podendo adicionar *flip-flops*, registros e outros dispositivos, definindo blocos funcionais programáveis. Estes blocos funcionais são interligados através de uma malha de interconexão, também programáveis. Esta malha também pode interligar as entradas e saídas dos circuitos para o exterior do integrado, como mostrado na Figura 3.3.

Exemplos de CPLDs podem ser encontrados nos dispositivos da ALTERA Corporation, pioneira na comercialização desta tecnologia. ALTERA lançou o que ela denominou de EPLDs (*Erasable Programmable Logic Devices*) ganhando o mercado e sendo seguido por outros fabricantes.

Enquanto que as CPLDs trabalham com blocos de SPLDs, as FPGA utilizam blocos lógicos para esta função (Figura 3.4). Estes blocos lógicos podem ser programados executando uma função lógica definida. Estas funções são roteadas através

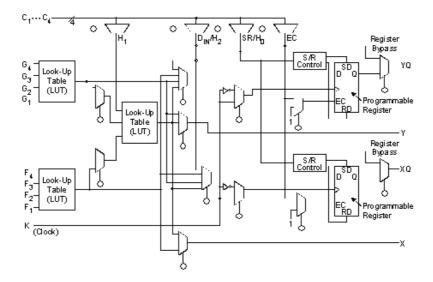


Figura 3.4: Bloco lógico de uma FPGA da Xilinx.

de uma lógica de interconexão programável. Esta configuração permite maior capacidade de funções lógicas que podem ser implementadas em comparação com as CPLDs. A pioneira neste tipo de dispositivo foi a XILINX Inc. A Figura 3.5 mostra um destes dispositivos.

#### 3.3 O sistema da ALTERA

A ALTERA Corp. (ALTERA, 1996) oferece uma vasta gama de dispositivos PLDs, tanto CPLDs, quanto FPGAs. Também oferece um ambiente de desenvolvimento de projetos utilizando estes dispositivos, oferecendo além de ferramentas de programação e análise, ferramentas de síntese e simulação.

Atualmente a família de dispositivos ALTERA é composta de várias linhas (ALTERA, 2005), cada uma com características de densidade e velocidade determinadas. As principais estão descritas na Tabela 3.1.

Os ambientes de desenvolvimento oferecidos pela ALTERA são o MAX+PLUS II (ALTERA, 1997) e mais recentemente o QUARTUS II. Estes ambientes permitem que as várias fases envolvidas num projeto sejam executadas de forma integrada. O

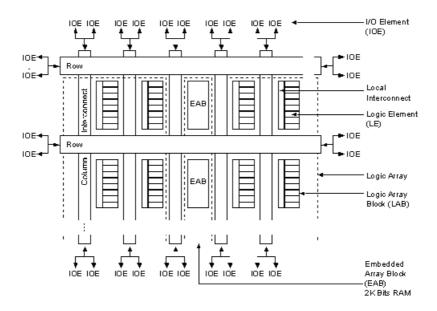


Figura 3.5: Esquema de uma FPGA da série 10k da ALTERA.

Tabela 3.1: Família de dispositivos da ALTERA.

Família	Tipo	Descrição
MAX 7000	CPLD	CPLD de baixa densidade e baixo custo
MAX II		CPLD de alta densidade e alta performance
FLEX 10K	FPGA	FPGA de baixa densidade
APEX 20K	FPGA	FPGA de baixa densidade
Cyclone	FPGA	FPGA de média densidade
Cyclone II	FPGA	FPGA de média densidade e baixo custo
Stratix	FPGA	FPGA de alta densidade
Stratix II	FPGA	FPGA de alta densidade e alta performance

projeto pode ser introduzido de várias formas, desde edição de gráfica do circuito até a programação em linguagem de descrição de hardware (HDL). Além das linguagems VHDL (Very High Speed Integrated Circuit Hardware Description Language) e Verilog HDL, consolidadas na indústria em ambientes de programação de circuitos a ALTERA oferece uma linguagem própria denominada AHDL (ALTERA Hardware Description Language).

Após a descrição do projeto através de um dos métodos de entrada descritos acima, o projeto deve passar por uma compilação onde será feita a consistência do código fornecido e será executada a síntese do circuito desejado. O projeto pode ser compilado para uma família de dispositivos específica, podendo o compilador, desta forma, extrair informações temporais sobre o circuito gerado.

Com a saída da compilação do circuito é possível executar uma simulação do projeto. O ambiente ALTERA oferece um editor de formas de ondas que pode ser utilizado para gerar as entradas da simulação. Através do resultado da simulação a funcionalidade do circuito projetado pode ser conferida. Uma análise temporal também pode ser executada. Caso qualquer erro ou inadequação seja encontrado o projeto pode ser novamente editado e compilado.

Estando tudo em ordem, o projeto pode ser programado no dispositivo adequado através de um programador de dispositivos apropriado. A ALTERA oferece uma gama extensa de dispositivos para programação, desde interfaces USB (*Universal Serial Bus*), paralelas e seriais até programadores dedicados. Estando o dispositivo corretamente programado, este pode ser conectado no circuito final para verificação e depuração. Todas etapas podem ser visualizadas na Figura 3.6.

Uma das séries de EPLDs da ALTERA que fez bastante sucesso foi a família MAX7000. Estas EPLDs eram compostas por um conjunto de células lógicas contendo circuitos programáveis capazes de realizar funções lógicas com ou sem registrador, conhecidas como macrocélulas. Uma macrocélula típica desta família é ilustrada

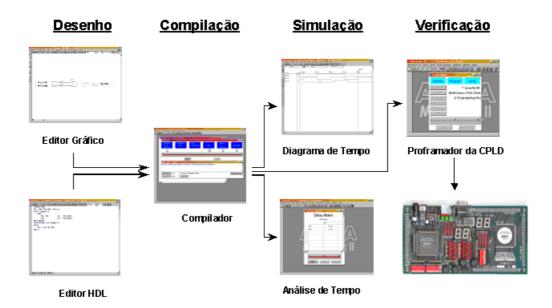


Figura 3.6: Etapas do processo de projeto de um dispositivo programável da ALTERA.

na Figura 3.7. Oito macrocélulas são agrupadas em blocos maiores formando um LAB (*Logic Array Block*). Em cada um destes LABs, visto na Figura 3.3, funções mais complexas podem ser efetuadas, utilizando como entradas pinos de entrada e saída do integrado ou saídas de qualquer macrocélulas do mesmo LAB, além de uma seleção de macrocélulas vizinhas.

Finalmente, uma pastilha desta família é composto por uma série de LABs que podem ser interligados através de uma rede de interconexão programável, conhecida como PIA (*Programmable Interconnect Array*), conforme ilustrado na Figura 3.8. Existem dispositivos com densidades de 32, 64, 96, 128, 160, 256 e 512 macrocélulas.

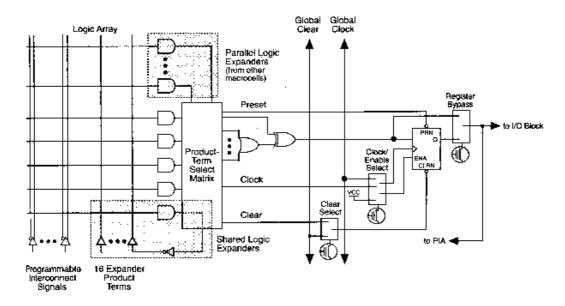


Figura 3.7: LAB da família MAX7000 da ALTERA.

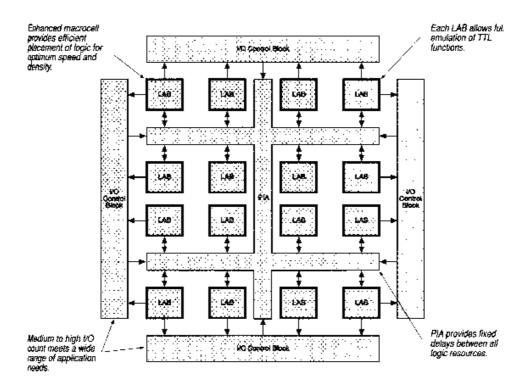


Figura 3.8: Diagrama interno de uma pastilha da família MAX7000.

## Capítulo 4

# Arquitetura do sistema de aquisição de dados

#### 4.1 Introdução

Neste capítulo será apresentada a arquitetura desenvolvida para o sistema de aquisição de dados dos sensores e a atuação sobre os diversos dispositivos e equipamentos que compõem o Veículo Autônomo proposto. A primeira seção descreve o funcionamento da interface desenvolvida. A segunda seção descreve o módulo de atuação nos motores CC de tração e direção do veículo. A terceira seção apresenta o módulo de medida e leitura de posição e velocidade dos motores. A quarta seção descreve o módulo referente aos sensores infravermelhos. A quinta seção apresenta o módulo de leitura de medidas dos sonares. Finalmente a sexta seção tratará do módulo de leitura de outros vários sistemas que o Veículo Autônomo pode adotar, tais como bússola, giroscópio e outros sensores.

### 4.2 Funcionamento da interface desenvolvida

Para o processamento de dados e sinais do Veículo decidiu-se pelo uso de um sistema utilizando um microcomputador da linha PC. Os micros desta linha mostraram-se

mais atrativos por diversos fatores, entre os quais seu custo atual baixo; sua grande capacidade de processamento - que cada dia torna-se mais veloz; a grande variedade de interfaces de coprocessamento (DSP's (Digital Signal Processing) e interfaces similares) disponíveis no mercado; e principalmente por ter uma arquitetura aberta e com vasta documentação, o que permitiria projetar uma interface que utilizasse o máximo de recursos que a arquitetura oferecesse.

A aquisição de dados dos sensores e atuação nos dispositivos que integram o Veículo Autônomo é feita de forma periódica e relativamente simples, apesar da complexidade eventual de alguns desses dispositivos. Desta forma, minimizando a carga de processamento para a leitura/escrita de dados nos sensores/atuadores, projetou-se uma interface capaz de receber e enviar dados a vários sensores e atuadores em paralelo, de forma transparente e rápida. Esta interface, denominada de SIGA é formada basicamente por uma série de blocos de entradas e/ou saídas responsáveis pelo tratamento de sinais digitais oriundos dos sensores e atuadores pertencentes ao Veículo Autônomo, e por registradores de entrada e/ou saída de dados que se ligam ao barramento de entrada e saída de periféricos de um micro da linha PC para que a .comunicação entre os dois sistemas, interface e PC, se estabeleça.

Para o funcionamento do Veículo Autônomo são necessários diversos dispositivos sensores e atuadores, conforme pode ser visualizado no diagrama em blocos da Figura 4.1.

Estes conjuntos de dispositivos podem ser divididos em cinco blocos básicos com funcionamento independente com relação a sua utilização no Veículo:

- Sistema de gerenciamento da interface;
- Sistema de atuação e leitura de posição das rodas;
- Sistema de acionamento e leitura dos sonares;

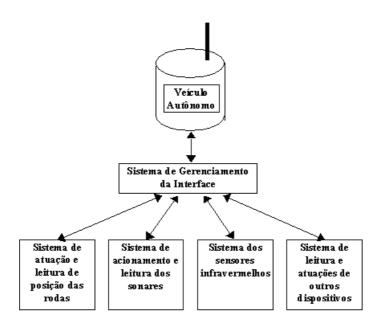


Figura 4.1: Diagrama em blocos da arquitetura do Veículo Autônomo CONTRO-LAB-AGV.

- Sistema de acionamento e leitura dos sensores infravermelhos (IR);
- Sistema de leitura e atuações de outros dispositivos.

O sistema de atuação e posicionamento das rodas é o responsável pelo acionamento dos motores CC que comandam as rodas de tração e direção, bem como pela leitura de suas posições atuais medidas por encoders incrementais. De acordo com a arquitetura escolhida para o Veículo, serão necessários três módulos de atuação e leitura da posição para as rodas, sendo dois módulos para as rodas de tração e um para a roda de auxílio à direção. A potência a ser entregue aos motores CC será determinada através de um algoritmo de controle (AUDE, 2003) que estará sendo executado em algum sistema de processamento, tendo como base os dados fornecidos pela leitura de posição das rodas e/ou outro dado externo, por exemplo, o posicionamento global do veículo. O tempo de aquisição de dados depende de todo sistema utilizado (motores, acoplamentos, rodas, peso do veículo, etc). No projeto do CONTROLAB-AGV o tempo para aquisição de dados dos motores é de 20 milissegundos.

O sistema de acionamento e leitura dos sonares é o responsável pelo disparo do pulso de ultra-som de um sonar, bem como a medição de tempo entre este pulso e seu respectivo eco, cujo tempo é diretamente proporcional à distância percorrida. No modelo proposto para este Veículo, serão utilizados quatro sonares, sendo que três, fixados sobre a bandeja giratória serão responsáveis pelo sistema de posicionamento/detecção horizontal e um, fixo orientado para o alto, pela determinação do pé-direito do ambiente em que se encontra. Este sistema deve permitir tanto um sonar ser disparado, com seu retorno esperado pelo mesmo, como também um sonar ser disparado enquanto que os outros sonares ficam esperando seu eco.

O sistema de acionamento e leitura dos sensores de infravermelho tem como função medir a distância de obstáculos situados entre 20 cm e 80 cm à frente do AGV. É também responsável por determinar o ângulo em que este obstáculo se encontra em relação à frente do veículo, semelhante a uma "bengala eletrônica" (LOPES, 2001).

Os sistemas de leitura e atuação de outros dispositivos e composto por portas paralelas de dados que suportam leitura e escritas de sensores e/ou atuadores que o Veículo Autônomo possa necessitar durante o seu desenvolvimento, tais como giroscópios, bússolas, solenóides, etc.

Esta interface utiliza 16 bits de dados para comunicação com o PC, que podem ser transferidos tanto por endereçamento de portas de E/S como através de programação de um canal livre de 16 bits de DMA (Direct Memory Access), obtendo-se com este ultimo, taxas de transferência mais rápidas.

A interface também permite interrupções periódicas através de uma linha não utilizada de interrupções do PC, que podem ser programadas e habilitadas através de um relógio interno da interface. O intervalo de tempo utilizado entre interrupções será o tempo de aquisição de dados.

#### 4.3 Gerenciamento da interface

O hardware para manipulação de dispositivos e gerenciamento da interface com o Veículo Autônomo foi projetado para trabalhar com o barramento ISA (*Industry Standard Architecture*) de um sistema microcomputador PC. Além do uso do barramento de endereços e dados dos sinais de controle, a interface também deve utilizar os sinais de interrupção e de transferência via DMA.

Os vários módulos da interface são conectados através de um barramento interno de dados de 16 bits e um barramento interno de endereços de dispositivos de seis bits, conforme ilustrado na Figura 4.2. O barramento de dados da interface conectase ao barramento do PC através de um buffer bidirecional controlado pelo módulo gerenciador da interface.

O módulo de gerenciamento da interface pode ser dividido em três blocos básicos, conforme visualizado na Figura 4.3. O primeiro bloco constitui a decodificação e sincronização dos sinais oriundos do barramento do microcomputador PC, conforme descritos a seguir. O segundo bloco agrega os diversos relógios utilizados pela interface, necessários para a contagem e sincronização de eventos dos dispositivos controlados. O terceiro, é composto pela decodificação dos endereços internos utilizados pela interface SIGA para endereçar seus vários módulos controladores de dispositivos.

Para a codificação e sincronização com o PC os sinais utilizados foram os seguintes sinais do barramento de controle e dados:

- /ior\_pc sinaliza que uma operação de leitura de porta será efetuada;
- /iow\_pc sinaliza que uma operação de escrita de porta será efetuada;
- /dack\_pc sinal que identifica que uma operação de DMA será realizada;
- sbhe uma operação com transferência de 16 bits será realizada;

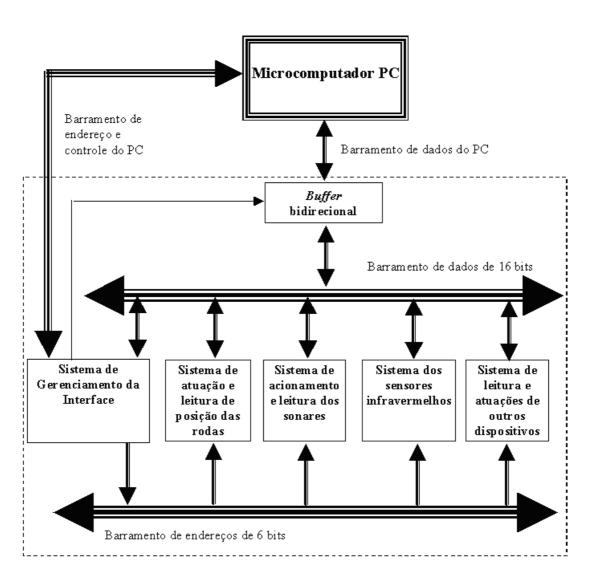


Figura 4.2: Estrutura internas dos barramentos da interface SIGA.

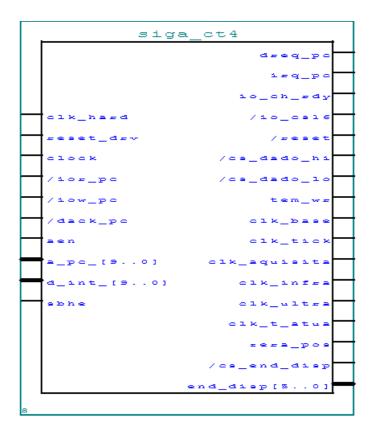


Figura 4.3: Sinais do sistema de gerenciamento da interface SIGA.

- reset\_drv sinaliza que o PC foi resetado;
- clock sinal de relógio do barramento do PC;
- aen acusa a existência de dados de endereço quando há DMA;
- a\_pc\_[9..0] barramento de endereços utilizados na decodificação da interface;
- d\_int\_[15..0] barramento de 16 bits de dados;
- dreq\_pc utilizado para pedir uma operação de DMA;
- irq\_pc utilizado para requisitar uma interrupção do processamento do PC;
- io\_ch\_rdy sinaliza que o pedido de wait states no barramento do PC e
- /io\_cs16 sinaliza que o dispositivo utilizará uma transferència de 16 bits.

		3 1 1
Porta	Direção	Descrição
0x390	escrita	Registro do relógio de tempo de tick
0x392	escrita	Registro do relógio de tempo de aquisição
0x394	escrita	Registro do relógio de tempo do infravermelho
0x396	escrita	Registro do relógio de tempo do ultra-som
0x398	escrita	Registro dos estados da interface
0x39A	escrita	Registro do endereço do dado
0x39C	leitura	Registro do dado
0x39C	escrita	Registro do dado

Tabela 4.1: Endereços das portas de E/S utilizadas pela interface.

A interface foi programada para ser acessada através das portas de entrada/saída a partir do endereço base 0x390 até o endereço 0x39C, normalmente não utilizados pelos periféricos do PC atualmente. Caso necessário qualquer outra porta livre no PC poderá ser utilizada, bastando para tanto modificar o endereço base. A decodificação dos endereços das portas utilizadas para entrada/saída de dados é realizada utilizando os sinais do barramento de endereço do PC a\_pc\_[9..0] e os sinais /ior\_pc, /iow\_pc e aen. A utilização das portas é descrita na Tabela 4.1.

Para o correto funcionamento da interface, os diversos relógios que sincronizam a interface e temporizam os dispositivos atuadores e sensores devem ser ajustados. Estes ajustes são efetuados escrevendo o valor adequado de tempo de contagem em cada registro destes contadores. A seguir cada um desses relógios é descrito.

O relógio base de 5 MHz utilizado como entrada de outros relógios da interface é obtido através da divisão de um oscilador de 20 MHz. A utilização de um relógio de freqüência alta permite uma maior precisão na contagem de tempo da interface.

O relógio responsável pela sincronização da aquisição de dados e atuação dos motores da interface é o relógio de tempo de aquisição. Este relógio é incrementado pelo relógio básico da interface, ou seja, o relógio de tempo de tick. O relógio de tempo de aquisição é responsável pelo sinal que interromperá o microcomputador PC para que o mesmo possa processar as rotinas que controlam os dispositivos ligados ao

Veículo Autônomo. Este relógio é um contador de 10 bits, permitindo uma resolução de um milésimo do tempo básico, que é o tempo de tick. A programação deste relógio dá-se através da escrita na porta 0x392 de entrada e saída com uma palavra de 10 bits com o valor adequado.

O relógio de tempo de tick define a menor divisão de tempo da interface. Além de ser o tempo básico de aquisição de dados da interface, ele também compõem o tempo de motor ligado, que é o período em que os motores CC devem ser acionados, como será mostrado na seção 4.4. O relógio de tick também é definido por um contador de 10 bits e tem como relógio base um relógio interno programável com a freqüência de 5 MHz. Com esta freqüência base é possível obter tempo de tick desde 200 nanossegundos, quando o contador é carregado com o valor "1", até 204,8 microssegundos, quando o contador é carregado com o valor 1023. O que determinará o tempo de tick será principalmente a resolução desejada para o tempo de motor ligado. O registro do contador de tick é carregado através da porta 0x390 com a escrita de uma palavra inteira de 10 bits no barramento de dados do PC.

O relógio de tempo de infravermelho é utilizado para temporizar a máquina de estados responsável pelo interfaceamento dos sensores infravermelhos conforme será descrito na seção 4.7. Esse tempo é contabilizado por um contador de 10 bits, cujo clock é o relógio base de 5 MHz. Sua programação é efetuada através da porta de escrita 0x394. O tempo de infravermelho é definido atualmente como sendo um relógio de período igual a 100 microssegundos.

O relógio de tempo de ultra-som é utilizado para medir a duração dos pulsos emitidos pelos dispositivos ultra-sônicos do Veículo Autônomo, conforme descrito na seção 4.6. Este relógio foi construído com um contador de dez bits com relógio base de 5 MHz. A porta de programação utilizada com o PC é a 0x396, sendo de escrita.

A interface efetivamente funciona somente após o correto preenchimento do re-

Tabela 4.2: Descrição do registro de estados do sistema de gerenciamento da interface.

bit	Descrição
0	Habilita a contagem dos relógios
1	Reiniciar a interface
2	Permite zerar os contadores de posição
3	Limpa o registro de interrupção
4	Permite interrupção na interface
5	Permite utilização de DMA na interface
6	Faz pedido de DMA

gistro de estados. Este registro de oito *bits* determina quando a interface dispara a contagem dos relógios e quais recursos do PC serão utilizados, por exemplo, a utilização do sinal de pedido de interrupção por um dispositivo externo ao PC, conforme descrito na Tabela 4.2. Sua escrita é feita através da porta 0x398 do PC.

Os contadores dos relógios só começarão a contar os tempos programados a partir do momento que o bit "0" do registro de estados for escrito com o valor "1". O bit "1" do registro é responsável por "resetar" toda interface após a escrita de um valor "1", reiniciando todos os contadores e máquinas de estados. O bit "2" permite que os contadores de posição dos motores CC sejam zerados (valor "1") ou não zerados (valor "0") após passar por um sensor de origem (ver seção 4.5).

Para que as interrupções do relógio de tempo de aquisição, no PC, sejam permitidas pela interface, o valor "1" deve ser escrito no bit "4" do registro de estados, através de uma rotina. No momento que uma interrupção é pedida para o PC, quando acontece um tempo de aquisição, o sinal ireq\_pc vai para nível alto. Após o atendimento desta interrupção pelo PC através de rotinas de atendimento de interrupção, é necessário que o sinal ireq\_pc restabeleça o nível baixo. Isto é obtido pulsando o bit "3" negativamente, ou seja, este bit deve ser zerado e novamente colocado em "1" para que novas interrupções possam ser acionadas e posteriormente atendidas pelo PC.

A habilitação de DMA é efetivada com a escrita de "1" no bit "5". O pedido de DMA poderá ser realizado de quatro formas: a primeira é através de um evento externo via sinal dreq\_pc; o segundo através de programação das portas do DMA; o terceiro pela conclusão de um tempo de aquisição; e finalmente através da geração de pedido de DMA pela própria interface com a escrita de "1" no bit "6" do registro de estados.

Quando o DMA é habilitado na interface, toda vez que um tempo de aquisição é completado, automaticamente um pedido de DMA é feito através da linha de sinal de controle dreq\_pc. Isto implicará em um pedido de DMA para o barramento do PC, que será atendido caso a programação de DMA tenha sido efetuada corretamente. Esta programação será vista posteriormente no capítulo 5. Após o aceite do pedido de DMA pelo sistema de árbitro do barramento do PC será dado o início da transferência automática de dados entre a memória do PC e a interface, onde o valor disponível nos registros dos diversos dispositivos sensores da interface SIGA serão transferidos para a memória do PC. Este pedido de leitura de dados da interface via DMA também pode ser efetuado através do registro de estados com a escrita no bit "6".

Durante o atendimento da interrupção, após terem sido feitos os processamentos e as análises dos dados dos dispositivos, as atuações nos dispositivos podem ser efetuadas através da programação do DMA do PC para uma transferência de escrita de dados, quando os dados da memória do PC são transferidos automaticamente para a interface nos registros apropriados.

O acesso aos vários registros dos dispositivos ligados na interface SIGA pode ser feito através de DMA da forma descrita acima, ou diretamente, acessando cada um dos registros desejados individualmente. Este acesso deve ser feito primeiro escrevendo-se no registro de endereço de dado, localizado na porta 0x39A, o endereço do registro desejado. Após esta escrita, as linhas do barramento de endereço interno

end\_disp[], de seis bits, terão o valor do registro desejado, de acordo com o endereço de dispositivos na Tabela 4.3. Caso uma leitura de registro seja desejada, uma leitura na porta 0x39C deverá ser efetuada, quando então os dados do registro desejado passarão do barramento interno de dados d\_int[] para o barramente de dados do PC. De outra forma, caso seja necessário uma escrita de dados em um registro deverá ser escrita na porta 0x39C o dado desejado. Em ambos os casos o sinal /cs\_end\_disp será acionado para que os circuitos dos outros dispositivos sejam preparados para a transferência de dados.

A lógica de interfaceamento com o PC e os contadores dos relógios foi implementada em uma EPLD ALTERA 7160. A taxa de ocupação deste circuito foi de 78% com 50 pinos de entrada/saída utilizados.

### 4.4 Atuação nos motores CC

Os motores CC comandam o movimento das rodas de tração do Veículo Autônomo e o movimento angular da roda de direção, facilitando a realização de curvas que o Veículo Autônomo foi comandado a fazer. Para a atuação nos motores CC foi escolhido o acionamento através de PWM (*Pulse Width Modulation*). Esta técnica apresenta como vantagem principal sua simplicidade de operação via computador, permitindo o uso de circuitos de potência simples, pequenos, leves e de grande eficiência (DUBEY, 1989). O controle de potência através de PWM foi utilizado com êxito no desenvolvimento da ICBM (AUDE, 1995), colaborando para a adoção deste método no presente projeto.

A utilização de PWM facilita bastante o acionamento de motores através de circuitos digitais. Através desta técnica o motor é acionado com tensão constante durante um determinado tempo  $\delta T$  a cada intervalo fixo de tempo T (Figura 4.4). O fator  $\delta$  é chamado de duty ratio e varia entre 0 e 1. Um diagrama básico de

Tabela 4.3: Relação dos endereços dos registros de dispositivos internos da interface SIGA.

Endereço	End_disp[]	Descrição
0x00	0x00	Posição do motor 1
0x02	0x01	Velocidade do motor 1
0x04	0x02	Posição do motor 2
0x06	0x03	Velocidade do motor 2
0x08	0x04	Posição do motor 3
0x0A	0x05	Velocidade do motor 3
0x0C	0x06	Posição do motor 4
0x0E	0x07	Velocidade do motor 4
0x10	0x08	Posição do motor 5
0x12	0x09	Velocidade do motor 5
0x14	0x0A	Registro de origens dos motores
0x16	0x0B	Registro de leitura paralela
0x18	0x0C	Contador do ultra-som 1
0x1A	0x0D	Contador do ultra-som 2
0x1C	0x0E	Contador do ultra-som 3
0x1E	0x0F	Contador do ultra-som 4
0x20	0x10	Atuação do motor 1
0x22	0x11	Atuação do motor 2
0x24	0x12	Atuação do motor 3
0x26	0x13	Atuação do motor 4
0x28	0x14	Atuação do motor 5
0x2A	0x15	Registro de escrita paralela
0x2C	0x16	Registro de ativação e inibição do ultra-som
0x2E	0x17	Registro de ativação do infravermelho
0x30	0x18	Dado do infravermelho 1
0x32	0x19	Dado do infravermelho 2
0x34	0x1A	Dado do infravermelho 3
0x36	0x1B	Dado do infravermelho 4
0x38	0x1C	Dado do infravermelho 5
0x3A	0x1D	Dado do infravermelho 6
0x3C	0x1E	Dado do infravermelho 7
0x3E	0x1F	Dado do infravermelho 8

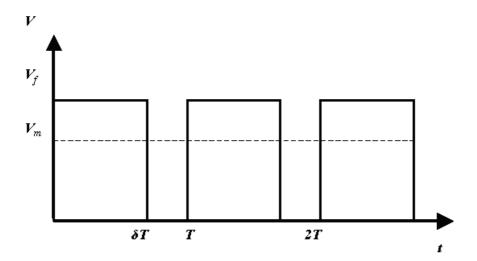


Figura 4.4: Forma de onda PWM com duty ratio igual a  $\delta$  e período igual a T.

um circuito acionador utilizando PWM pode ser visto na Figura 4.5. Um circuito que permite a inversão do motor é o amplificador em ponte H, e pode ser visto na Figura 4.6. A tensão média  $V_m$  nos motores é função da tensão de alimentação  $V_f$  e é dada por:

$$V_m = \int_0^T V_f dt = \frac{1}{T} \int_0^{\delta T} V_f dt = \delta V_f \tag{4.1}$$

Desta forma, podem-se obter tensões contínuas médias de valores entre 0 e  $V_f$  a partir de uma fonte de tensão fixa de alimentação, bastando para isto variar o duty ratio, ou seja, o tempo em que o motor é alimentado em determinado intervalo de tempo através da entrada controladora  $V_c$ . Algumas considerações de ordem prática devem ser feitas. A chave controlada SW idealmente deve ter tensão zero entre seus terminais quando estes são acionados, de forma que toda a tensão de alimentação  $V_f$  esteja disponível nos terminais dos motores para que a Equação 4.1 seja verdadeira. Da mesma forma a tensão de alimentação  $V_f$  deve se manter constante. Na pratica, para obter estes resultados, foram utilizados como chaves controladas transistores MOSFETs de alta corrente e baixa resistência de condução para obter tensões baixas na chave quando em condução. Também foi utilizada uma fonte regulada de

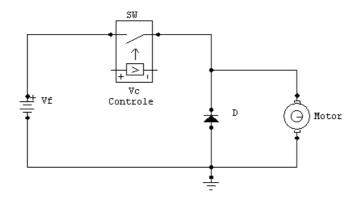


Figura 4.5: Diagrama básico do circuito acionador PWM para motor CC.

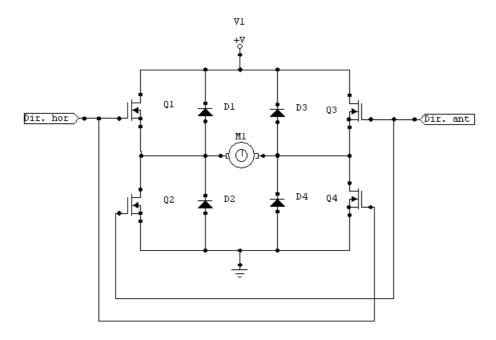


Figura 4.6: Circuito amplificador em ponte H para permitir mudança na direção dos motores CC.

alimentação CC para alimentação dos motores, o que permite manter a tensão de alimentação constante.

São necessárias para a atuação em um motor CC a quantidade de potência desejada no motor e o sentido de rotação do motor. Na utilização do PWM, a quantidade de potência é determinada por um valor proporcional ao tempo em que a tensão de alimentação será aplicada no motor. Este valor, que é denominado tempo de motor ligado é constituído por uma unidade chamada de tick. O número de ticks máximo que poderá ser dado em um motor não poderá ultrapassar um

```
Siga_at

GLK_TICK

CLK_T_ATUA

/RESET

LIG_MOT_DC_HOR

/CS_ATUA

D_INT_[(LARG_CONT_AT) - (1)..0]
```

Figura 4.7: Sinais do circuito para atuação de um motor CC por PWM.

valor predeterminado, denominado de tempo de atuação. Após um tempo de motor ligado, o motor será desligado até que um novo tempo de motor ligado seja novamente programado. Para a determinação do sentido é necessária apenas a programação de sentido horário ou anti-horário. O valor de tempo de motor ligado em tick está associado ao valor de  $\delta T$  assim como o valor de tempo de atuação em tick está associado ao valor T da Equação 4.1. A razão entre tempo de motor ligado e tempo de tempo

O tempo de motor ligado é medido por um contador de onze bits sendo um bit para a determinação da direção e dez bits para o valor da potência. Desta forma, é permitido a programação de potências no motor, desde a potência total (cem por cento da potência) até potência zero (motor desligado), em incrementos de milésimos de potência, sendo que o bit mais significativo é a direção e determina se o motor deverá girar no sentido horário ou anti-horário. O valor desejado em ticks deve ser inicialmente colocado neste contador. Esta tarefa deve ser feita pondo o valor do tempo de motor ligado no barramento interno de dados d\_int. Após uma transição positiva do sinal /cs\_atua, que determina que uma atuação será escrita, este valor será carregado no contador. O diagrama do circuito responsável pela atuação nos motores pode ser visualizado na Figura 4.7.

Um pulso do sinal clk\_t\_atua, que tem o período do tempo de atuação, deve ser o responsável pelo acionamento dos motores quando o tempo de motor ligado é diferente de zero atuando nos sinais lig\_mot\_dc\_hor ou lig\_mot\_dc\_ant. A partir da

```
siga_at4

clk_tick

clk_t_atua lig_mot_dc_hor[5..1]

/reset lig_mot_dc_ant[5..1]

/cs_end_disp sai_dir[5..1]

end_disp[5..0]

d_int_[15..0]
```

Figura 4.8: Sinais do circuito do sistema de atuação de 5 motores CC.

atuação do sinal clk\_t\_atua, a cada pulso positivo do sinal clk\_tick, tendo este um período de tick, há o decremento do contador de tempo de motor ligado. Quando este contador chega a zero a saída é desativada, ou seja, os motores são desligados. Os sinais lig\_mot\_dc\_hor e lig\_mot\_dc\_ant são as saídas que comandam se os motores estarão ligados no sentido horário ou anti-horário respectivamente. Estas saídas deverão ser ligadas ao módulo de potência dos motores para que estes possam ser acionados.

O sinal /reset serve para reiniciar os circuitos da interface no momento em que o mesmo é ligado ou na eventualidade de novas programações.

Esta lógica de atuação nos motores CC foi implementada utilizando-se uma EPLD ALTERA 7128. Neste dispositivo foi possível programar cinco módulos de atuação de motores, todos independentes. A ocupação do dispositivo foi de 74% com 41 pinos sendo utilizados por sinais de entrada e saída. O diagrama desta implementação pode ser visualizado na Figura 4.8. Cada módulo de atuação é acessado através do barramento de endereços de dispositivos nos sinais end\_disp[] e do sinal de controle /cs\_end\_disp oriundos do módulo de controle. Os endereços 0x10, 0x11, 0x12, 0x13 e 0x14 acessam cada um dos registros potência de atuação dos motores. Estes têm onze bits e são somente de escrita.

O circuito de potência para o acionamento dos motores foi desenvolvido através de um circuito em ponte implementado com transistores TMOS de grande capacidade de corrente. Por apresentarem baixa resistência no estado ligado e, em conseqüência, baixa queda de tensão, estes transistores foram escolhidos como melhor opção que os tradicionais transistores bipolares e Darlington. Este é um aspecto de grande importância, já que, devido à alta corrente nos motores CC utilizados, na ordem de cinco ampères quando em regime, aliado à baixa tensão de operação dos motores CC, doze Volts, permite melhor aproveitamento de energia, o que é crucial no desenvolvimento de projetos que fazem uso de baterias.

# 4.5 Leitura de posição e velocidade dos motores CC

A leitura de posição e velocidade dos motores CC tem por finalidade a obtenção da posição e velocidade angulares das rodas de tração e da posição angular da roda de direção. Estes dados são obtidos através da relação de redução entre a roda e o motor CC. O módulo de leitura de posição e velocidade dos motores CC foi implementado através da utilização de encoders incrementais como sensores (SNYDER, 1985). A escolha de encoder como sensor foi feita por ser de baixo custo e pela facilidade de implementação.

Cada encoder, composto por dois conjuntos de sistemas óticos para medir as transições de raias, gera sinais que são discriminados, obtendo pulsos que deverão ser contados, conforme abordado no capítulo 2. A posição do motor, e conseqüentemente da roda a ele conectada, é diretamente proporcional à quantidade de pulsos contados. A velocidade pode ser obtida através da medida de tempo entre dois pulsos consecutivos.

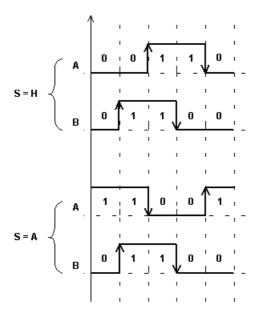


Figura 4.9: Saída do encoder incremental, em quadratura de fase, mostrando o sinal para sentido horário (H) e anti-horário (A).

Os sinais vindo dos sensores do encoders normalmente estão em quadratura, ou seja, separados em fase de 900 (Figura 4.9). Esta discriminação é obtida através da utilização de uma máquina de estados simples em que cada mudança observada nos sinais fase\_0 e fase\_90 na saída de um encoder, amostrada pelo sinal de relógio clk\_hard é comparada com seu estado anterior (Figura 4.10). Caso as entradas atuais levem o sistema a um próximo estado característico de rotação horária, o sinal de direção dir\_encoder será ativado para o nível lógico "1"e é gerado um pulso, clk\_encoder, que incrementará o contador. No caso contrário, o sinal de direção será ativado para o nível lógico "0", sentido anti-horário, fazendo com que o sinal de relógio decremente o contador.

O número de transições possíveis em uma revolução do motor depende do número de raias que o encoder tem e de que forma a discriminação dos sinais é feita. Tomando como base um encoder com N raias escuras, para o caso de uso de um encoder incremental óptico, e considerando que só será realizada a comparação de estados quando houver a mudança de raia de claro para escura de apenas uma saída do sensor do encoder, têm-se N pulsos do sinal clk\_encoder para uma volta completa do

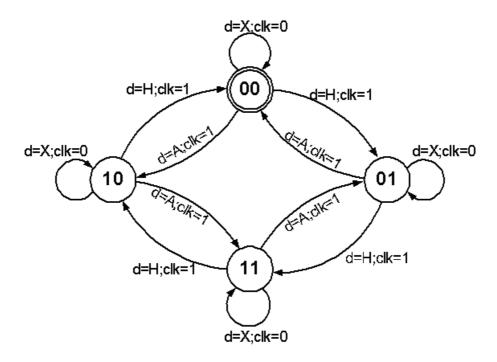


Figura 4.10: Máquina de estados de discriminação do encoder incremental para detecção de sentido horário (H) e anti-horário (A).

encoder. Para o caso de mudanças tanto de claro para escuro quanto de escuro para claro de uma entrada do sensor do encoder, o número de pulsos por volta completa do encoder passa a ser o dobro do número de raias escuras do encoder, ou seja, 2N. Finalmente quando qualquer transição de claro/escuro e escuro/claro for analisada em qualquer dos canais do encoder, o número de pulsos por volta do encoder passará a ser 4N, o quádruplo do número de raias escuras do encoder. Esta característica de poder modificar a resolução do encoder também foi implementada, bastando ativar o sinal X4 e X2, caso seja desejado o quádruplo ou o dobro da resolução respectivamente. Caso o desejado seja a resolução normal basta deixar estes sinais em nível lógico "0".

Os sinais clk\_encoder e dir\_encoder são conectados a um contador de treze bits para a contagem da posição do encoder, representado na Figura 4.11. Quando o sinal de direção é horário, ou seja, nível lógico "1", este contador é incrementado e, decrementado quando o sentido é anti-horário.

```
Siga_p

CLK_HARD

ORIGEM

/CS_POS D_INT_[(LARG_CONT_P) - (1)..0]

/RESET

CLK_ENCODER

DIR_ENCODER
```

Figura 4.11: Sinais do circuito para um contador de posição.

Como já visto no capítulo 2, os encoders incrementais não determinam o referencial da posição de origem. Neste caso é necessária uma referência externa para que seja determinada a posição da origem. Um sinal origem é disponibilizado para que o contador possa ser iniciado com zero quando houver a ativação de origem por um sensor. Esta ativação poderá ser mascarada quando for desativado o sinal zera\_pos, ou seja, caso o sinal zera\_pos esteja em nível lógico "0", o contador não será zerado, independente de uma origem ser detectada ou não. O contador também será zerado na ativação de um sinal /reset gerado externamente à interface.

Após cada contagem do sinal do encoder, o conteúdo atual do contador é armazenado em um registro da posição para leitura. A saída de dados com o conteúdo do registro da posição é disponibilizada para leitura através do barramento internos de dados d\_int quando o sinal /cs\_pos é acionado. No momento desta leitura qualquer atualização no contador é adiada para o momento seguinte a leitura, não ocorrendo desta forma erros de sincronização de dados.

A lógica necessária para a leitura de posição foi implementada em uma EPLD ALTERA 7160, permitindo a leitura de três encoders incrementais e a contagem de posição através de três registradores de treze bits (Figura 4.12). A taxa de ocupação do dispositivo foi de 81%, utilizando-se 43 pinos para entradas e saídas de sinais. Cada um destes módulos de medição de posição é acessado através do barramento de endereços de dispositivos nos sinais end\_disp[] e do sinal de controle

```
siga_p3

clk_hard

clk_veloc

origem[3..1]

zera_pos

/cs_end_disp clk_encoder[3..1]

end_disp[5..0] dir_encoder[3..1]

/reset d_int_[15..0]

fase_0_[3..1]

fase_90_[3..1]

x2

x4
```

Figura 4.12: Sinais do circuito do sistema para 3 contadores de posição.

/cs\_end\_disp oriundos do módulo de controle. Os registros que contêm a posição atual dos encoders são obtidos através dos endereços internos 0x00, 0x02 e 0x04. Estes registros são somente para leitura.

#### 4.6 Leitura das medidas de ultra-som

A medida de distância do Veículo Autônomo até um obstáculo pode ser efetuada via sensores ultra-sônicos. Esta medida pode ser obtida através da contagem de tempo entre um sinal de emissão de ultra-som e o sinal de recepção do eco refletido pelo obstáculo encontrado. A distância do Veículo Autônomo ao obstáculo será proporcional ao tempo que o pulso da onda ultra-sônica demorou a percorrer, conforme explicado no capítulo 2.

O projeto do medidor dos sinais de ultra-som tomou por base um aparelho me-

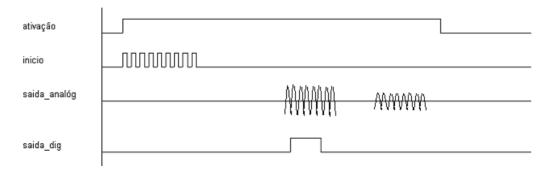


Figura 4.13: Sinais do módulo de ultra-som utilizados pelo Veículo Autônomo. O sinal início determina a emissão de pulsos ultra-sônicos com o sinal saída\_dig acusando a recepção somente do primeiro eco.

didor de distâncias por ultra-som disponível no mercado, COMPUMETER, modelo SW-104J, fabricante ARAX, capaz de medir objetos distando de 55 centímetros até 12,5 metros. Estes aparelhos tinham um conjunto emissor e receptor independentes e um circuito eletrônico embarcados responsável por emitir um pulso ultra-sônico, receber seu eco e visualizar a medida em seu visor. Após análise do circuito eletrônico chegou-se a conclusão que os sinais essenciais para o interfaceamento com o medidor do Veículo Autônomo são o pulso de excitação do transdutor emissor e o sinal de eco (Figura 4.13). Estes sinais não são de natureza muito diferentes dos sinais encontrados no módulo sensor da Polaroid 6500 (SENSCOMP, 2005), onde os sinais INIT e ECHO têm funções semelhantes.

O circuito de medição dos sonares tem dois sinais digitais de entrada: pulso\_t que indica o momento do disparo do pulso de ultra-som e pulso\_r que representa a recepção do pulso de eco. O diagrama em blocos do circuito pode ser visualizado na Figura 4.14. A saída pulso\_tof é disponibilizada para visualização do tempo total do sinal. Esta saída é ativada desde o momento que o pulso\_t é disparado até o momento em que pulso\_r é ativado, quando o eco é recebido, permitindo a visualização deste tempo em osciloscópio.

No momento do disparo de um sinal de ultra-som, um contador de quinze bits

```
LARG_CONT_UL=10

siga_ul

CLK_HARD

/RESET

CLK_ULTRA D_INT_[(LARG_CONT_UL) - (1)..0]

/CS_ULTRA PULSO_TOF

/CS_AI_ULTRA ATIVA_ULT

PULSO_T INIBE_ULT

PULSO_R

D_INT_ATIVA

D_INT_INIBE
```

Figura 4.14: Sinais do circuito do medidor por ultra-som.

começa sua contagem de tempo, com um sinal de relógio clk\_ultra. O relógio programável que gera o sinal clk\_ultra, descrito na seção 4.3, pode assumir períodos entre 200 nanossegundos e 204,8 microssegundos. Desta forma, o contador de quinze bits pode assumir um valor máximo maior que 6 segundos. A contagem termina na chegada do eco, quando o valor do contador é armazenado em um registro para posterior leitura. Após armazenar o valor do contador, este é zerado para que nova contagem possa ocorrer. O contador também é reiniciado quando é ativado um sinal /reset. O valor do registro de tempo do ultra-som é obtido através do barramento internos de dados d\_int quando o sinal /cs\_ultra é acionado para leitura.

O sinal ativa\_ult deve ser ativado quando for desejado emitir um pulso de ultrasom. É prevista a inibição da transmissão do pulso, porém com a medição do tempo
de eco através do sinal inibe\_ult. Desta forma pode-se medir o eco proveniente
de um pulso ultra-sônico de outro sensor. Ambos sinais são ativados utilizando
os sinais do barramento internos de dados d\_int quando o sinal /cs\_ai\_ultra é
acionado para escrita.

Este módulo foi implementado através de uma EPLD ALTERA 7160, permitindo o acesso de quatro dispositivos ultra-sônicos com registradores de quinze *bits* cada um. A taxa de ocupação do dispositivo foi de 84%, utilizando 46 pinos para sinais.

Figura 4.15: Sinais do circuito do sistema para 4 medidores por ultra-som.

O diagrama em blocos deste pode ser visualizado na Figura 4.15. Cada um destes módulos é acessado através do barramento de endereços de dispositivos nos sinais end\_disp[] e do sinal de controle /cs\_end\_disp oriundos do módulo de controle. Os registros que contêm os valores dos contadores dos sensores são obtidos através dos endereços internos 0x0C, 0x0D, 0x0E e 0x0F. Estes registros são somente para leitura. O endereço interno 0x16 é utilizado para a ativação dos sinais ativa\_ult e inibe\_ult. Os quatro primeiros bits do barramento de dados d\_int[] (bits 0 a 3) são referentes aos sinais de ativação e os quatros próximos bits (bits 4 a 7) são referentes aos sinais de inibição.

### 4.7 Leitura das medidas dos infravermelhos

Os sensores infravermelhos utilizados no Veículo Autônomo têm o objetivo de fazer a medida de distância de objetos que se encontram a menos de oitenta centímetros, uma vez que os sensores ultra-sônicos utilizados não permitem medidas dedistância menores que cinqüenta centímetros. O sensor infravermelho escolhido para este Veículo Autônomo foi o modelo GP2D02 da Sharp. Este sensor tem como característica a entrega de um resultado cujo valor é função da distância do objeto posto

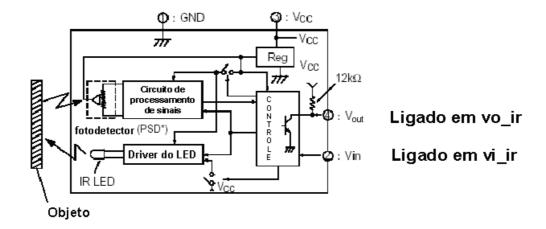


Figura 4.16: GP2D02 - Diagrama interno do sensor.

a sua frente. Uma grande vantagem encontrada neste sensor é o fato de o mesmo já possuir uma interface digital de seus sinais, o que facilita enormemente o projeto de uma interface entre este sensor e o sistema do Veículo Autônomo.

Conforme visto, este dispositivo serializa os dados, dispondo de uma linha para controle e sinalização de relógio e outra linha para status e saída dos dados Figura 4.16. A forma de leitura dos dados deve seguir o diagrama de tempos fornecido pelas Sharp para temporização dos sinais, visualizado na Figura 4.17. O bit mais significativo (MSB) é colocado primeiro na saída após o fim da medida pelo sensor. Os outros bits se seguem ao passo que a entrada Vin é acionada, até o menos significativo (LSB). O roteiro completo para a leitura deste sensor é visto a seguir.

O diagrama em blocos do circuito pode ser visualizado na Figura 4.18. Uma máquina de estados, ilustrada na Figura 4.19, foi montada para permitir o funcionamento do sensor conforme descrito a seguir. Inicialmente a entrada Vin do sensor GP2D02 é mantida em nível lógico alto (nível "1") e a saída Vout tem nível "1". Para iniciar uma medida o sinal Vout deve ser colocado em nível "0". Isto é obtido acionando o sinal /cs\_a\_infra, do módulo infravermelho, com o sinal d\_int\_ativa em "1", o que levará a saída vi\_ir para "0". Com isso, a saída Vout do sensor passa para "0", o que é sentido através da entrada vo\_ir do circuito.

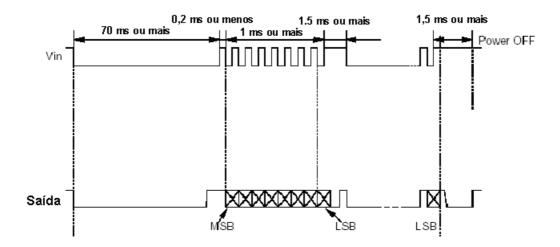


Figura 4.17: GP2D02 - Diagrama de temporização utilizando os sinais seriais Vin e Saída.

```
Siga_ir

/RESET

CLK_INFRA

/CS_INFRA

/CS_A_INFRA D_INT_[(LARG_DADOS_IR) - (1)..0]

VO_IR

D_INT_ATIVA
```

Figura 4.18: Sinais do circuito para leitura do sensor GP2D02.

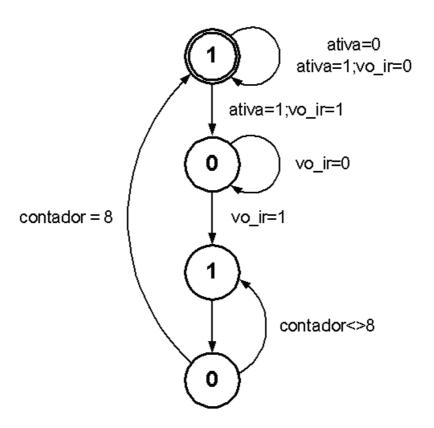


Figura 4.19: Máquina de estados para leitura da medida do sensor GP2D02 disponibilizada via serial.

```
siga_ir8

/reset
clk_infra vi_ir[7..1]

/cs_end_disp d_int_[15..0]

end_disp[5..0]

vo_ir[7..1]
```

Figura 4.20: GP2D02 - Sinais do circuito do sistema para leitura de 8 sensores GP2D02.

O sensor GP2D02 faz várias medidas para a obtenção da distância do objeto, resultando um valor médio para a medida (KÜHNEL, 2003). Este procedimento é seguido de um processamento para a obtenção do valor da medida e a discretização deste resultado. Estas operações podem durar aproximadamente 70 milissegundos, segundo o manual do sensor. Quando o resultado da medida estiver disponível, o sinal Vout do sensor é levado a nível alto "1". Essa transição é sentida por vo\_ir, e a máquina de estados prepara-se para aquisitar o resultado da medida do sensor. Cada um dos oito bits de dados da medida é transferido no sinal vo\_ir na transição de "1"para "0" do sinal vi\_ir. Após receber todos os bits do resultado da medida o sinal Vin do sensor é colocado em "1"indicando que não será feita nova medida e o dispositivo GP2D02 entra no estado de power-off, reduzindo seu consumo de energia.

Os dados recebidos através da saída serial Vout do sensor são armazenados, na interface, em um registro de oito bits com o resultado da medida. Este resultado é disponibilizado no barramento interno d\_int[] quando o sinal /cs\_infra for acionado. O sinal /reset permite reinicializar a máquina de estados e os contadores responsáveis pelo funcionamento deste circuito.

A lógica necessária para o processamento dos sensores infravermelhos foi implementada em uma EPLD ALTERA 7160, permitindo o controle de oito sensores do dispositivo GP2D02. A taxa de ocupação do dispositivo foi de 90%, utilizando

39 pinos para sinais. O diagrama em blocos completo é ilustrado na Figura 4.20. Cada um destes módulos dos sensores PSD é acessado através do barramento de endereços de dispositivos nos sinais end\_disp[] e do sinal de controle /cs\_end\_disp oriundos do módulo de controle. Os registros que contêm os valores dos contadores dos sensores são obtidos no barramento de dados interno d\_int[] através dos endereços internos 0x18 até 0x1F, responsáveis pelo endereçamento de cada um dos sensores interligados. Estes registros são somente para leitura. Para o acionamento do sensor, de forma a iniciar um processo de medida, o endereço 0x17 deve ser colocado no barramento de endereços de dispositivos internos e o sinal /cs\_end\_disp ativado para escrita. O primeiro sensor é ativado escrevendo-se "1"no bit "0"do barramento de dados d\_int[]. De forma semelhante o segundo sensor é ativado no bit "1", e assim sucessivamente, até o oitavo sensor que deve ser escrito no bit "7"do barramento d\_int[].

### 4.8 Leitura de outros sensores

É prevista a utilização de portas paralelas para atuação e aquisição de dados de outros circuitos que possam ser conectados ao Veículo Autônomo. Estas portas são implementadas através de registros de entrada e saída de dados (Figura 4.21). Atualmente foram implementados dois registros de 16 bits para esta finalidade, um para leitura e outro para escrita. A leitura é feita através do endereço interno 0x0B e a escrita no endereço 0x15. Os dados são disponibilizados no barramento interno d\_int[]. A entrada de dados é realizada através dos sinais disp\_in[], enquanto a saída é disponibilizada pelos sinais disp\_out[]. Esta lógica foi implementada em espaços vazios nos vários dispositivos ALTERA instalados no Veículo Autônomo.

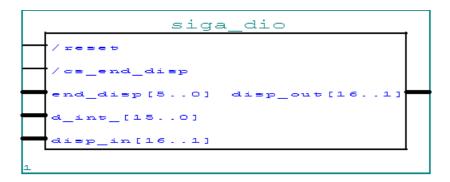


Figura 4.21: Sinais do circuito do sistema para entrada e saída paralelas.

# Capítulo 5

# Rotinas do sistema de aquisição de dados

### 5.1 Introdução

As rotinas da interface de comandos permitem ao usuário/programador acesso aos vários sistemas de processamento de sinais, atuações e aquisições de dados do Veículo Autônomo. O desenvolvimento destas rotinas tem como alvo principal a performance de processamento, já que o objetivo maior da interface de comandos e aquisição de dados do Veículo Autônomo é funcionar como uma unidade de co-processamento, liberando desta forma o computador para processamento de tarefas de outros sistemas que tenha maior custo computacional, por exemplo, processamento de imagens e planejamento de trajetória.

# 5.2 Funcionamento das rotinas da interface de comandos do Veículo Autônomo

De acordo com a arquitetura definida no capítulo anterior, a interface de comandos e aquisição de dados do Veículo Autônomo é dividida em sistemas menores, cada um

realizando processamento de comandos e dados de dispositivos sensores e atuadores diferentes, seguindo uma hierarquia onde existe um módulo principal que coordena a atuação dos demais. De forma semelhante, as rotinas da interface seguem uma estrutura hierárquica, em que temos sistemas de funcionamento independentes coordenados por uma rotina principal. A representação desta hierarquia é mostrada na Figura 5.1 através de um diagrama em blocos dos diversos módulos.

Durante todo projeto das rotinas da interface de comandos e aquisição de dados foi levado em consideração a portabilidade das mesmas para vários tipos de sistemas operacionais, além de permitir o uso de vários compiladores diferentes, sem que para isso fossem necessárias grandes modificações no mesmo. Desta forma foi evitado ao máximo o uso de soluções mirabolantes que utilizassem funções ou diretivas específicas de compiladores ou sistemas operacionais. Em casos onde se tornou necessária a utilização de soluções específicas, foram usadas macros e/ou funções como solução, centralizando em um único ponto o local de possíveis modificações. A linguagem de programação utilizada no desenvolvimento das rotinas é a linguagem C, já que além da simplicidade, permite acesso fácil e rápido ao hardware desenvolvido, não necessitando, desta forma, do uso de rotinas feitas em assembly, o que dificultaria a portabilidade do sistema para outras arquiteturas. Apesar de todo o sistema do Veículo Autônomo, nesta primeira etapa, estar baseado no sistema operacional MSDOS, uma próxima versão, no futuro, poderá ser implementada em um ambiente que tenha suporte nativo em processamento em tempo real, por exemplo, o sistema operacional QNX.

O conjunto de rotinas da interface do Veículo Autônomo pode ser encarado como um driver entre o sistema operacional, e as aplicações de alto nível do Veículo Autônomo, por exemplo, o sistema de controle dos motores CC ou a avaliação de obstáculos detectados e controle de trajetória. A utilização desta estrutura permite vantagens importantes como isolar a necessidade de conhecimento do programa-

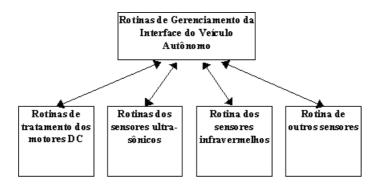


Figura 5.1: Diagrama de blocos das rotinas de manipulação da interface do Veículo Autônomo.

dor/usuário sobre detalhes de implementação do hardware e do sistema operacional. Basicamente, para cada um dos subsistemas da interface do Veículo Autônomo foram desenvolvidas rotinas de inicialização e finalização. Após a inicialização de determinado subsistema a utilização dos recursos do mesmo torna-se simples, bastando chamar a rotina adequada ao serviço desejado.

A primeira atividade que o driver das rotinas de comandos do Veículo Autônomo deve executar é definir o tempo de aquisição dos motores CC e tempo de tick. Estes serão os tempos que irão sincronizar todas as atividades e tomadas de decisões da interface do Veículo Autônomo durante seu funcionamento. O tempo de tick é a menor divisão de tempo de toda a interface, sendo que todas as outras temporizações são definidas a partir dele. Sob o ponto de vista de execução das rotinas da interface pelo processador, o tempo de tick não tem significado, sendo apenas um parâmetro para programação dos registros dos temporizadores da interface. Já o tempo de aquisição equivale ao período de tempo em que serão lidos os dados dos motores CC. O tempo de aquisição é o temporizador mais importante de toda a interface sob o ponto de vista do processamento de rotinas do Veículo Autônomo. É a cada tempo de aquisição que é acionada uma interrupção no processador, e por isso também é chamado de tempo de interrupção. Estes tempos são passados para a interface através de chamada da rotina SIGA\_Define\_Aqui\_Tick(tempo\_de\_aquisição,tempo\_de\_tick).

Após a definição e programação dos tempos segue-se a instalação do driver em si, executada pela rotina SIGA\_Inicia(). Esta rotina é responsável pela programação das pastilhas de DMA e interrupção do PC, programação dos contadores de relógio da interface, programação do registro de estados da interface, instala a rotina e libera a interrupção do PC. Com esta liberação, ao final de cada **tempo de aquisição**, esta rotina será executada. Quando não for mais necessária a utilização da interface, deverá ser executada a rotina SIGA\_Finaliza(), que desinstalará a rotina de interrupção e irá restaurar a programação anterior das pastilhas de interrupção de DMA do PC.

### 5.2.1 Rotinas de tratamento de interrupção

Os sistemas de alto nível que compõem o Veículo Autônomo utilizam a rotina de interrupção para medir e atuar no ambiente. Isto é realizado eficientemente com o auxílio das diversas rotinas de aquisição de dados dos sensores e atuação nos dispositivos oferecidas pela interface do Veículo Autônomo. O endereço da função do sistema a ser chamada dentro da rotina de interrupção é passada para interface de forma que, quando o PC for interrompido pela interface, esta função é executada. Um exemplo de um sistema de alto nível que utiliza as funções da interface é o sistema de controle ótimo do Veículo Autônomo (AUDE, 2003).

No início da rotina de interrupção, os dados dos sensores disponibilizados pela interface são lidos. Como mencionado no capítulo 4, estes dados podem ser obtidos através de transferência via DMA ou através de operações diretas de entrada/saída. A escolha do método utilizado será baseado em características da interface utilizada, na forma que os dados originados nos sensores são organizados e na quantidade de dados a serem transferidos. Uma análise sobre esta escolha se segue.

A transferência via DMA tem base na utilização de circuitos apropriados no sistema de processamento computacional. No caso da arquitetura do PC ela é

baseada no circuito integrado 8237 da Intel, programado de forma a permitir uma transferência de dados após cada execução de instrução do processador. No caso de operações diretas de entrada/saída, através da execução de instruções do tipo input e output, é escrito uma rotina para cada conjunto de dispositivos sensores, ordenando uma transferência de dados.

A principal vantagem de utilizar a transferência de dados via DMA é o "paralelismo" obtido com a execução de instruções e transferência de dados intercalados. Para o uso de transferência via DMA é necessário além de uma programação inicial dos circuitos responsáveis pela transferência, uma outra programação dos mesmos circuitos a cada interrupção. Desta forma, quando existem poucos dados a serem transferidos da interface via DMA, pode não ser vantajosa sua utilização, já que o número de instruções na rotina de programação da transferência necessárias mais o número de transferências de dados em si pode tornar-se maior que o número de instruções utilizados no método de operações diretas de entrada e saída. No caso específico desta interface, com o número atual de dispositivos, a melhor opção é a de operações diretas de entrada/saída. Para qualquer um dos métodos descritos uma área para transferência de dados deve ser reservada na memória do PC.

A transferência de dados dos diversos sensores no início do tempo de interrupção, como primeiro passo, é decorrente da necessidade de sincronização dos mesmos, ou seja, desta forma é assegurado que os resultados dos tratamentos dos dados dos sensores e suas análises referem-se a uma mesma amostragem no tempo, tanto para sensores do mesmo tipo, como no caso entre sensores infravermelhos, para sensores de tipos diferentes, entre sensores de posição e velocidade de motores e bússolas Cabe ressaltar que a interface está continuamente coletando dados e disponibilizando-os para serem lidos por suas rotinas.

O próximo passo após a coleta de dados dos sensores passa a ser o tratamento dos mesmos. Nesta fase, a rotina de interrupção irá executar chamada a funções

específicas de tratamento para cada sensor utilizado no Veículo Autônomo. Somente os tipos de sensores que forem utilizados necessitam ser inicializados e terão suas funções de tratamento sendo executadas. Isto significa que se determinada aplicação do Veículo Autônomo nunca necessitar, por exemplo, dos sensores infravermelho, não será necessário iniciá-los (estes procedimentos de inicialização dos diversos sensores serão vistos mais adiante) e nem serem executados na rotina de interrupção. Esta prática permite o término mais rápido da rotina de interrupção e, desta forma, mais tempo para o processamento de tarefas dos sistemas de mais alto nível. Atualmente existem chamadas para funções de tratamento dos sensores dos motores, dos sensores infravermelhos e dos sensores de ultra-som.

Finalmente, é chamada a função responsável pela execução das tarefas de alto nível (função do sistema de alto nível). Esta função é a última atividade da rotina de interrupção nesta interface de comandos e é conhecida como **função do cliente**. A função do cliente define como a interface deve atuar no ambiente.

Fato relevante a se destacar é o cuidado de que todas as funções chamadas na rotina de interrupção terem que executar no menor tempo possível, ou porque o sistema operacional pode estar precisando que o processador execute uma de sua funções, ou devido a urgência na execução das funções de alto nível dos sistemas do Veículo Autônomo..

#### 5.2.2 Rotinas de tratamento os motores CC

As rotinas dos motores CC são utilizadas para tratamento dos dados relativos aos motores CC utilizados no Veículo Autônomo. Elas podem ser divididas em duas categorias: rotinas que tratam de dados de leitura dos sensores acoplados aos motores CC, como medidores de posição e velocidade; e rotinas que tratam da atuação nos motores CC, aplicando-lhes alguma potência.

Tabela 5.1: Parâmetros dos motores utilizados no Veículo Autônomo necessários para os cálculos de odometria.

Parâmetro	Descrição
MOT_DRV_pulsos_volta	Pulsos do encoder por rotação dos motores
MOT_DRV_pulsos_segundo	Velocidade em pulsos por segundo do encoder por
	rotação dos motores
MOT_DRV_pulsos_seg_pico	Velocidade em pulsos por segundo do encoder por
	rotação dos motores (valores de pico)
MOT_DRV_pulsos_seg_picol	Velocidade em pulsos por segundo do encoder por
	rotação dos motores (valores de pico - motor livre)
MOT_DRV_pulsos_segundo_2	Aceleração em pulsos por segundo do encoder por
	rotação dos motores
MOT_DRV_pulsos_seg_pico_2	Aceleração em pulsos por segundo do encoder por
	rotação dos motores (valores de pico)
MOT_DRV_pulsos_seg_picol_2	Aceleração em pulsos por segundo do encoder por
	rotação dos motores (valores de pico - motor livre)
MOT_DRV_reducao	Redução total dos motores para o movimento das rodas
MOT_DRV_raio_nominal	Raios nominais das rodas (metros)
MOT_DRV_raio	Raios reais das rodas (metros)
MOT_DRV_offset_reset	Offset da posição de reset em relação ao reset no sensor

Para começar a utilizar os dados dos motores CC é necessário antes instalar o módulo de tratamento dos motores. A instalação é feita através da chamada da rotina MOT\_DRV\_Inicia(). Esta função define uma chamada da rotina de tratamento de dados do motor dentro da função de tratamento de interrupção da interface SIGA, após a coleta de dados dos motores. Além disso, esta rotina comanda a leitura dos parâmetros dos motores utilizados no Veículo Autônomo. Estes parâmetros são constantes relativas à construção dos motores e do Veículo Autônomo. Um resumo destes parâmetros atualmente utilizados pode ser observado na Tabela 5.1. Estes parâmetros devem ser definidos para cada um dos motores utilizados.

Como visto na seção 5.2.1, a rotina principal da interface do Veículo Autônomo é uma rotina de interrupção que percorre todos os tipos de dispositivos disponíveis e faz um tratamento de seus respectivos dados para posterior análise dos sistemas de mais alto nível. O primeiro passo da rotina de interrupção é coletar os dados das interfaces dos dispositivos instalados, que no caso dos motores CC são os sensores de posição e velocidade. Após este procedimento é executado, ainda dentro da mesma rotina de interrupção, o tratamento dos dados dos motores.

```
typedef struct
  unsigned long
                                // tempo da aquisição dos dados
               pos[MOTOR DC];
                                  posição dos motores em pulsos
  long
                                   deslocamento dos motores em pulsos
  int
               des[MOTOR DC];
  float
               vel [MOTOR DC] ;
                                  velocidade dos motores em pulsos
                                  por segundo
  float
               ace[MOTOR DC];
                                  aceleração dos motores em pulsos
                               // por segundo ao guadrado
                 estrutura de dados dos motores
 MOT DRV T:
```

Figura 5.2: Estrutura da variável MOT\_DRV\_atual, responsável pelos dados da odometria.

O primeiro passo da rotina de tratamento dos motores é verificar se está sendo executado um procedimento de **reset** do motor, comandado por algum sistema de alto nível do veículo. Neste caso, a interface tentará localizar a origem para referenciar a contagem de posição deste motor. Isto é realizado testando se o motor acionou o sensor de origem. No momento que o motor passou pela origem o valor do registro de posição é zerado, e a partir daí, é incrementado até a parada total do motor. Este valor é então é armazenado na variavel MOT\_DRV\_atual, descrita a seguir. O bit "2" do registro de estados da interface é desativado para evitar que novas passagens pelo sensor de origem zerem novamente o contador de posição.

Seguinte a este procedimento, a atualização dos medidores de posição, velocidade e aceleração dos motores é realizada. Conforme foi descrito no capítulo 4, os contadores de posição implementados no hardware têm treze bits de largura. Este valor é suficiente para uma contagem de 8192 pulsos de encoder, o que na maioria das vezes é insuficiente para representar a rotação dos motores por longo tempo. A situação é mais complicada ainda para o caso de movimentos contínuos dos motores, quando se está interessado não somente em saber a posição do motor (e conseqüentemente da roda do Veículo Autônomo) como também quantas voltas o motor já realizou. Para resolver esta situação é implementado na rotina de tratamento dos motores contadores de 32 bits de largura. Desta forma o contador de posição dos motores pode chegar a uma contagem superior a quatro milhões de pulsos antes de ter um overflow.

A variavel MOT\_DRV\_atual é composta por uma estrutura, MOT\_DRV\_T, mostrada na Figura 5.2. Ela armazena uma variável inteira de 32 bits para a posição. A diferença entre a medida da posição atual obtida do registrador de posição do motor com a medida de posição obtida na interrupção anterior é definida como deslocamento e é armazenada em uma variável em MOT\_DRV\_atual. Este deslocamento é adicionado ao valor de posição do motor armazenado na variável de posição de MOT\_DRV\_atual e seu valor é atualizado a cada interrupção que ocorre na interface.

A velocidade dos motores é obtida efetuando a divisão do deslocamento com o tempo de aquisição programado na interface, em segundos. Este resultado é armazenado em uma variável de velocidade, do tipo ponto flutuante, em MOT\_DRV\_atual. Este valor representa a quantidade de pulsos do encoder realizados por segundo.

De forma semelhante, a aceleração dos motores é obtida com a diferença entre a velocidade atual e a da interrupção anterior e dividindo este resultado pelo tempo de aquisição programado na interface, em segundos. Este resultado é armazenado em uma variável de aceleração, do tipo ponto flutuante, em MOT\_DRV\_atual. Este valor representa a quantidade de pulsos do encoder realizados por segundo ao quadrado.

Os dados dos motores atualizados na variável MOT\_DRV\_atual podem ser obtidos através da função MOT\_DRV\_Le(var\_m). Esta função tem como parâmetro var\_m que é um ponteiro para uma variável do tipo de MOT\_DRV\_atual, que será utilizada na rotina do cliente.

Nota-se que a variável de dados dos motores MOT\_DRV\_atual se trata de um vetor e que cada posição do mesmo contem dados de deslocamento, posição, velocidade e aceleração de cada um dos motores colocados no Veículo Autônomo.

No caso específico desta implementação de Veículo Autônomo, onde foram usados encoders incrementais para medição de posição e velocidade, o tratamento dos dados é caracterizado pela contagem dos pulsos dos encoders e diversos cálculos que transformam esta contagem em valores de deslocamento, posição, velocidade e aceleração dos motores. Inicialmente estes valores não estão em unidades de engenharia, ou seja, metros, metros por segundo e metros por segundo ao quadrado. A conversão em unidades de engenharia dar-se-á somente nas etapas de análise desses dados pelos sistemas de mais alto nível através de uso de funções de conversão. O conjunto destas funções pode ser visto na Tabela5.2.

As atuações nos motores do Veículo Autônomo são normalmente executadas na rotina do cliente, já que os valores da potência que devem ser dados aos motores normalmente partem de um procedimento de análise dos dados coletados pelos sensores do Veículo Autônomo. A exceção é o procedimento de reset de motores, onde a atuação é executada durante o tratamento dos dados, conforme explicado anteriormente. Para atuar nos motores é necessário utilizar a rotina MOT\_DRV\_Atua(mot,pot), onde o parâmetro mot é o número do motor desejado e o parâmetro pot é o tempo que o motor deve ficar ligado em ticks, para cada um dos motores do Veículo Autônomo. Quando todos os motores tiverem sido programados a rotina MOT\_DRV\_Dispara\_Atuação() será responsável por comandar o funcionamento dos motores, acionando o relógio de tempo de motor ligado.

A execução de *reset* dos motores descrito anteriormente é um comando que pode ser dado pelo sistema de comandos do Veículo Autônomo a qualquer momento que se fizer necessário um reinício dos contadores dos motores. A rotina MOT\_DRV\_Dispara\_Reset() é responsável pelo realização deste procedimento.

Quando não for mais necessário utilizar nenhuma função com os motores no Veículo Autônomo as rotinas de tratamento dos motores deverá ser finalizada, bastando para isto executar a rotina MOT\_DRV\_Finaliza().

Tabela 5.2: Funções de conversão dos dados de posição, velocidade e aceleração dos motores.

Função	Descrição
MOT_DRV_Pos_P2Rad_Motor	Conversão de posição em pulsos para radianos no motor
MOT_DRV_Pos_P2Grau_Motor	Conversão de posição em pulsos para graus no motor
MOT_DRV_Pos_P2Rad_Roda	Conversão de posição em pulsos para radianos na roda
MOT_DRV_Pos_P2Grau_Roda	Conversão de posição em pulsos para graus na roda
MOT_DRV_Pos_P2Metros	Conversão de posição em pulsos para metros na roda
MOT_DRV_Vel_P2Rad_Motor	Conversão de velocidade em pulsos para radianos no motor
MOT_DRV_Vel_P2Grau_Motor	Conversão de velocidade em pulsos para graus no motor
MOT_DRV_Vel_P2Rad_Roda	Conversão de velocidade em pulsos para radianos na roda
MOT_DRV_Vel_P2Grau_Roda	Conversão de velocidade em pulsos para graus na roda
MOT_DRV_Vel_P2Metros	Conversão de velocidade em pulsos para metros na roda
MOT_DRV_Ace_P2Rad_Motor	Conversão de aceleração em pulsos para radianos no motor
MOT_DRV_Ace_P2Grau_Motor	Conversão de aceleração em pulsos para graus no motor
MOT_DRV_Ace_P2Rad_Roda	Conversão de aceleração em pulsos para radianos na roda
MOT_DRV_Ace_P2Grau_Roda	Conversão de aceleração em pulsos para graus na roda
MOT_DRV_Ace_P2Metros	Conversão de aceleração em pulsos para metros na roda
MOT_DRV_Pos_Rad_Motor_P2	Conversão de posição em radianos no motor para pulsos
MOT_DRV_Pos_Grau_Motor_P2	Conversão de posição em graus no motor para pulsos
MOT_DRV_Pos_Rad_Roda_P2	Conversão de posição em radianos na roda para pulsos
MOT_DRV_Pos_Grau_Roda_P2	Conversão de posição em graus na roda para pulsos
MOT_DRV_Pos_Metros_P2	Conversão de posição em metros na roda para pulsos
MOT_DRV_Vel_Rad_Motor_P2	Conversão de velocidade em radianos no motor para pulsos
MOT_DRV_Vel_Grau_Motor_P2	Conversão de velocidade em graus no motor para pulsos
MOT_DRV_Vel_Rad_Roda_P2	Conversão de velocidade em radianos na roda para pulsos
MOT_DRV_Vel_Grau_Roda_P2	Conversão de velocidade em graus na roda para pulsos
MOT_DRV_Vel_Metros_P2	Conversão de velocidade em metros na roda para pulsos
MOT_DRV_Ace_Rad_Motor_P2	Conversão de aceleração em radianos no motor para pulsos
MOT_DRV_Ace_Grau_Motor_P2	Conversão de aceleração em graus no motor para pulsos
MOT_DRV_Ace_Rad_Roda_P2	Conversão de aceleração em radianos na roda para pulsos
MOT_DRV_Ace_Grau_Roda_P2	Conversão de aceleração em graus na roda para pulsos
MOT_DRV_Ace_Metros_P2	Conversão de aceleração em metros na roda para pulsos

Figura 5.3: Estrutura da variável BENG\_DRV\_atual, responsável pelo valor de distância medido pela bengala eletrônica.

#### 5.2.3 Rotinas de tratamento dos sensores infravermelhos

Os sensores infravermelhos são os responsáveis pela detecção de obstáculos próximos ao Veículo Autônomo. As rotinas que tratam estes sensores podem ser divididas em duas partes: rotinas que tratam dos dados resultantes da emissão e recepção do sensor infravermelho e que resultaram em um valor de distância do possível objeto detectado e rotinas que atuam no sensor infravermelho, iniciando um procedimento de medida no sensor.

Para que o tratamento de dados dos sensores infravermelhos seja iniciado, é necessário que seja executada a rotina BENG\_DRV\_Inicia() para que a mesma instale os procedimentos de tratamento adequados. Neste procedimento de iniciação, é lido o arquivo que contém as estatísticas das características de cada sensor. Também é inserida a chamada da rotina de tratamento dos sensores na rotina de interrupção da interface SIGA.

A leitura dos sensores de infravermelho é realizada com a utilização da rotina BENG\_DRV\_Le(var\_beng), onde o parâmetro var\_beng é uma variável estruturada, semelhante a variável BENG\_DRV\_atual, descrita na Figura 5.3.

Devido às características dos sensores utilizados, em que cada sensor infravermelho tem parâmetros diferentes, devido ao processo de fabricação, levantamentos estatísticos com os dados obtidos pelo sensor devem ser realizados, de forma que o resultado mais provável da medida é obtido. O valor disponibilizado pelo sensor tem uma relação com a medida real de distância que deve ser aferido para cada dispositivo instalado. Os valores destas estatísticas são armazenados como arquivos de calibração que devem ser lidos no procedimento de iniciação da rotina de tratamento dos sensores infravermelhos. O processo de calibração será detalhado mais adiante no capítulo 6.

A atuação nos sensores infravermelhos é feita nas rotinas de alto nível, já que o momento em que o disparo de um sensor deve ser realizado depende da análise de todos os dados já coletados pelo Veículo Autônomo. A atuação nos sensores infravermelhos é realizada através da rotina BEND\_DRV\_Dispara(bengs), onde bengs é um inteiro sem sinal em que cada bit representa um sensor infravermelho da bengala. Quando este bit está ligado, ou seja, em nível "1", o sensor correspondente é acionado, caso contrário, ele permanece inativo.

Outro fator importante realizado pela rotina de tratamento dos sensores infravermelhos é o tempo de resposta dos sensores. Como o tempo de resposta do sensor infravermelho utilizado no Veículo Autônomo é de setenta milissegundos, tempo este superior ao tempo de aquisição dos motores CC, que é de vinte milissegundos, uma sincronização nas rotinas deve ser realizada. Após o disparo de uma medida de distância por infravermelho os sensores são testados para ver se os dados de leitura já estão prontos e só então realizar seu tratamento. Esta teste é feito utilizando-se a função BENG\_DRV\_Le(var\_beng). Se ela retornar um valor diferente de nulo (NULL), existem novos dados da bengala disponíveis.

Da mesma forma que nas rotinas dos motores, após o uso dos sensores infravermelhos, uma rotina de finalização, BENG\_DRV\_Finaliza(), deve ser executada, sinalizando o fim do uso dos sensores infravermelhos.

#### 5.2.4 Rotinas de tratamento dos sensores de ultra-som

Da mesma forma que nos sensores de infravermelho, os sensores ultra-sônicos têm rotinas que tratam os dados de forma estatística, tentando com isto apresentar um valor mais provável para a medida. A rotina ULTRA\_DRV\_Le\_Valor(ultra) faz a leitura do sensor ultra-sônico ultra, sendo este valor em pulsos do relógio de ultra-som. A rotina que converte o valor lido nos sensores em valores de engenharia, ou seja, valores da distância do objeto detectado em metros é ULT\_DRV\_Dist\_Metros(tempo), onde tempo é a quantidade de pulsos do contador de ultra-som.

Também a atuação nos sensores ultra-sônicos é realizada nas rotinas de nível mais alto, através das rotinas ULTRA\_DRV\_Ativa(ultra) e ULTRA\_DRV\_Desativa(Ultra). A primeira comanda o sensor ultra a emitir pulsos de ultra-som para realização da medida, enquanto que a segunda comanda que o ultra-som ultra permaneça em silêncio. Quando existe um dado disponível do sensor de ultra-som a função ULTRA\_DRV\_Le\_Valor(ultra) retorna um valor diferente de nulo (NULL).

Como a velocidade do som é função da temperatura, conforme descrito no capítulo 2, a temperatura ambiente deve ser fornecida para que o calculo da distância seja efetuado. A rotina ULT\_DRV\_Vel\_Temp(temp) deve ser utilizada para fornecer a temperatura do ambiente do Veículo Autônomo, passando-a pelo parâmetro temp.

#### 5.2.5 Rotinas de tratamento de outros sensores

As rotinas referentes a outros sensores que possam integrar a interface do Veículo Autônomo devem criar um módulo para seu tratamento e incluir na rotina de tratamento de interrupção uma chamada para a execução deste módulo. Deve-se dar atenção à ordem em que estes vários módulos estarão agrupados, de modo a continuar garantindo a sincronização de tempos que a interface necessita, evitando também inter-relacionamento entre módulos distintos.

# Capítulo 6

# Resultados experimentais

### 6.1 Introdução

A arquitetura projetada foi implementada utilizando componentes da família MAX-7000 da ALTERA. A interligação dos componentes foi realizada em uma placa de prototipagem para *wire-wrap*, o que permitia a modificação rápida e acesso fácil aos pontos de teste.

Os algoritmos de programação da interface inicialmente foram desenvolvidos com a utilização do Borland C, sobre um ambiente DOS 6.22. Atualmente o desenvolvimento está sendo feito com o compilador GCC (Gnu C Compiler) 2.95.3 sendo executado no ambiente DOS 6.22, porém já prevendo migração para o QNX Neutrino versão 6.x..

### 6.2 Experimentos de aquisição

#### 6.2.1 Encoder

O módulo responsável pela leitura dos encoder foi implementado utilizando um dispositivo MAX 7160 da ALTERA para execução das funções discriminação do

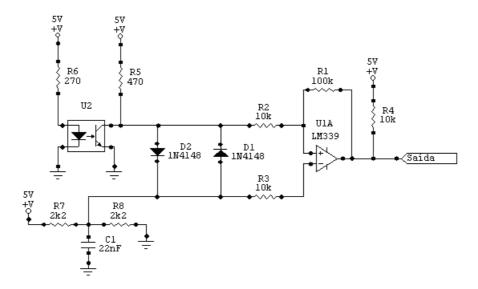


Figura 6.1: Circuito para fornecer histerese de forma a impedir oscilações nas transições da saída dos sensores dos encoders.

encoder e contagem dos pulsos.

Os encoders utilizados no Veículo Autônomo foram confeccionados no próprio laboratório do CONTROLAB e são compostos de discos de acrílico com 32 raias pretas e 32 raias transparentes, desta forma permitindo a contagem de até 128 pulsos por rotação completa do eixo a eles acoplados.

Os sensores utilizados como emissor/detector dos encoders foram fotodetectores de passagem normalmente utilizados em projetos deste tipo. Na primeira tentativa de acoplamento entre os fotodetectores e o módulo contadores viu-se a necessidade de melhor condicionamento dos sinais provenientes dos fotodetectores. Estes se mostravam extremamente ruidosos e apresentavam oscilações quando estavam próximas a regiões de transição de uma raia do encoder. Desta forma houve um projeto de circuito que visava dar certo grau de histerese nos fotodetectores (Figura 6.1), de tal forma que quando houvesse a transição de raias não houvesse oscilações.

Comparado com sistemas de aquisição de posição disponíveis no comércio, o sistema implementado mostra-se extremamente eficiente, fornecendo medidas de deslocamento dos motores com grau de precisão necessários ao funcionamento do

### 6.2.2 Bengala

O sistema de detecção de obstáculos utilizando infravermelho foi implementado utilizando um dispositivo MAX 7160 da ALTERA, permitindo o acionamento de até oito unidades de sensores GP2D02. Não houve dificuldades na interface destes sensores, bastando apenas a adição de um buffer para condicionamento dos sinais entre o sensor e o dispositivo ALTERA, já que as distâncias entre a interface de aquisição e os sensores eram relativamente grandes e passavam por pontos críticos quanto a ruído no Veículo Autônomo.

Para calibração dos sensores foi desenvolvido um programa para coletar medidas dos sensores. A metodologia de calibração consiste na obtenção dos valores reais da posição de um obstáculo colocado em frente de cada sensor. Várias medidas são efetuadas, variando-se a distância do obstáculo e também o material do mesmo. Após estas coletas de dados, os valores prováveis para a medida são calculados obtendo-se a média e o desvio padrão das diversas medidas realizadas. Este processo é realizado para cada um dos sensores colocados no Veículo Autônomo. Estas médias e desvios padrões são então armazenadas em um arquivo, que servirá para caracterizar os sensores utilizados no Veículo Autônomo. Os dados de calibração dos sensores utilizados no CONTROLAB-AGV nesta experiência são mostrados na Tabela 6.1.

Nota-se que o desvio padrão encontrado nas faixas abaixo de 20 centímetros e acima de 70 centímetros são relativamente altos na maioria dos sensores. De acordo com o fabricante, a faixa de utilização deste sensor está compreendida entre 10 centímetros e 80 centímetros. Na prática, porém, valores entre 20 centímetros e 60 centímetros parecem ser mais confiáveis.

Outra característica interessante observada nos sensores GP2D02 é que medidas

believe (vs) em diversus distancias e seus respectivos desvies padrao (dp).														
Dist.	IR1		IR2		IR3		IR4		IR5		IR6		IR7	
(m)	vs	dp												
0,10	232	1,527	230	0,537	210	0,325	234	0,927	240	0,593	214	0,787	217	0,567
0,20	164	0,673	155	0,549	130	0,634	156	0,711	170	0,598	148	0,474	140	0,480
0,30	136	0,302	126	0,502	101	0,238	128	0,000	140	0,372	121	0,137	112	0,241
0,40	122	0,304	111	0,000	86	0,000	114	0,303	124	0,478	109	0,144	105	0,000
0,50	115	0,139	103	0,418	78	0,329	106	0,000	116	0,282	101	0,498	100	0,505
0,60	110	0,239	98	0,245	73	0,000	101	0,454	113	0,558	96	0,283	85	0,197
0,70	106	0,416	93	0,471	69	0,471	97	0,499	109	0,635	92	0,445	82	0,473
0,80	101	0,570	86	0,545	66	0,535	92	0,507	106	0,855	88	0,452	79	0,444
0,90	98	0,646	83	0,954	65	0,591	90	0,670	104	1,132	85	0,761	77	0,557
1,00	95	0,700	78	0,921	62	0,582	85	0,774	101	1,209	83	0,756	75	0,756
2,00	90	4,997	65	5,219	57	2,519	80	3,982	96	3,866	78	5,561	70	5,140

Tabela 6.1: Dados de calibração dos sensores IR contendo os valores obtidos pelos sensores (vs) em diversas distâncias e seus respectivos desvios padrão (dp).

de objetos mais próximos apresentam melhor precisão, enquanto que objetos mais distantes oferecem pequenas variações no valor de distância. Isto é uma propriedade dos sensores baseados em triangulação, que para objetos mais distantes têm uma variação de ângulo menor, enquanto que para distâncias mais curtas esta variação torna-se mais significativa. Esta característica pode ser melhor observada na Figura 6.2.

No processo de calibração notou-se que os sensores são praticamente insensíveis a variação de texturas, ou seja, pouca variação da medida ocorre para diferentes tipos de materiais, tais como madeira, cortiça, pano, vidro, papel e plásticos. Para os vários materiais utilizados nos testes, também foi testada a sensibilidade quanto à cor dos objetos e a iluminação ambiente. Foi testado desde o ambiente totalmente escuro até com sol forte. Apenas em situações de sol forte foi notada alguma variação significativa na medida. Quanto à cor, existe pequena variação entre objetos muito claros e muito escuros. Também foi notada a sensibilidade da inclinação dos objetos em relação ao sensor. Dependendo da inclinação o objeto pode não ser detectado, e em alguns casos ele pode ser detectado como estando mais distante.

Um método de interpolação, descrito adiante, fez-se necessária para obter valores intermediários, uma vez que não é pratico obter todas as medidas possíveis para a calibração em todas as situações com todos os sensores. No processo de calibração foi

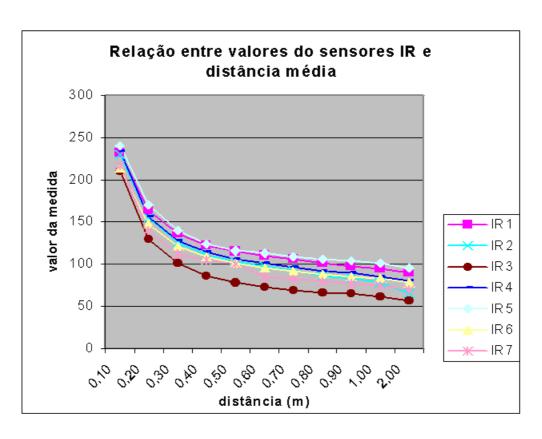


Figura 6.2: Gráfico dos dados de calibração dos sensores IR relacionando o valor no sensor e a distância real ao objeto detectado.

utilizado espaçamento de 10 centímetros entre uma medida e outra em um intervalo de 10 centímetros até 1 metro. Os pontos obtidos para estas medidas entraram numa fórmula de interpolação, de tal modo que, quando um novo valor de medida do sensor é coletado, os pontos que mais se aproximam dele são buscados. É então avaliada uma proporção linear entre as distâncias que estes pontos próximos representam e o valor da medida do sensor, resultando então uma distância provável para a medida. Considerando V como sendo o valor da medida realizada encontramos a distância D, em metros, utilizando a seguinte fórmula:

$$D = \frac{d_i - d_{i-1}}{v_i - v_{i-1}} * (V - v_{i-1})$$

Onde  $v_i$  e  $v_{i-1}$  são os valores de medidas do infravermelho encontrados na Tabela 6.1, mais próximos de V.  $d_i$  e  $d_{i-1}$  são os valores de distância correspondentes a  $v_i$  e  $v_{i-1}$  encontrados na tabela.

Para teste e aferição da bengala eletrônica foi implementado um programa que dispara os sensores infravermelhos e coleta suas medidas, mostrando o ponto de detecção de obstáculos na tela, como se fosse uma tela de radar. Uma tela de saída deste programa pode ser vista na Figura 6.3, onde um obstáculo foi detectado por vários sensores. Neste caso o objeto era uma caixa de papelão de 50 centímetros de lado que estava com uma de suas arestas em frente aos sensores. O erro na medida de distância da quina deste objeto foi de aproximadamente 2 centímetros para uma distância de 24 centímetros. Cabe ressaltar que medida de distância a quinas é de difícil avaliação.

#### 6.2.3 Ultra-som

Os sensores de ultra-som foram pouco testados nesta implementação do Veículo Autônomo por motivos de mudança de filosofia no algoritmo de determinação da

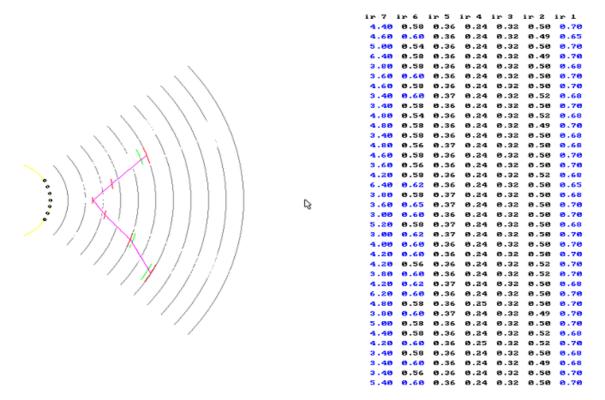


Figura 6.3: Tela de saída da bengala eletrônica detectando um objeto. Os valores nas colunas mostram as distâncias, em metros, dos sensores ao objeto, tomadas em vários momentos.

trajetória com desvio de obstáculos. Os testes realizados utilizaram modelos de trenas ultra-sônicas comerciais que tiveram seus circuitos internos analisados e depois interfaceados com o módulo de ultra-som desenvolvido.

Os testes executados foram suficientes para comprovar o funcionamento da interface de tratamento de dados do ultra-som. Porém, os sensores de detecção de obstáculos que compunham o projeto tinham que satisfazer as seguintes especificações: custo baixo, baixo consumo, e alcance determinado (entre 10 centímetros e 80 centímetros). Os sensores de ultra-som de custo baixo disponíveis no mercado não satisfizeram nem o consumo baixo nem a faixa de alcance requisitada.

## Capítulo 7

### Conclusões e trabalhos futuros

### 7.1 Considerações Finais

Este trabalho teve como motivação principal o estudo, projeto e implementação de uma arquitetura escalar de aquisição de dados de um Veículo Autônomo. Esta arquitetura permitiu a todos os sistemas que compõem o veículo receberem informações sobre o ambiente e atuarem no mesmo de forma simples, rápida e organizada. Estes objetivos foram obtidos com o desenvolvimento de uma interface dedicada a atender os sistemas que compunham o CONTROLAB-AGV.

Esta interface utilizou circuitos modulares implementados em uma tecnologia de hardware reconfigurável, utilizando EPLDs. Conforme descrito no capítulo 4, isto permite que alterações possam ser facilmente implementadas.

Outra decisão importante foi o desenvolvimento de rotinas, descritas no capítulo 5, utilizadas pelos sistemas de alto nível, para interagir com o ambiente. Estas rotinas foram implementadas em linguagem C, utilizando o DOS como sistema operacional. Inicialmente foram compiladas com o compilador Borland C, porém atualmente o compilador GCC está sendo utilizado por ser mais fácil de portar para outros sistemas operacionais.

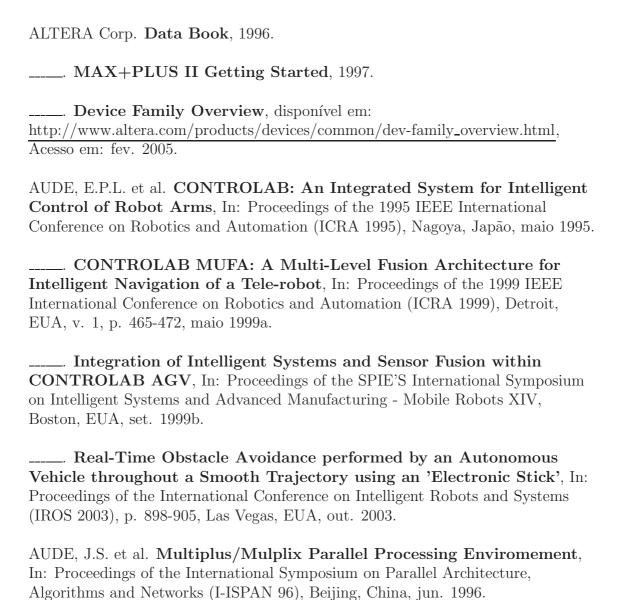
Os resultados quanto à performance da arquitetura de aquisição de dados foram demonstrados pelo sistema de utilização da bengala eletrônica do CONTROLAB-AGV discutidos na seção 6.2.2.

Esta arquitetura tem sido exaustivamente utilizada pelos diversos sistemas de alto nível do CONTROLAB-AGV (AUDE, 2003, LOPES, 2005) que não fazem parte do escopo deste projeto.

Para acompanhar o desenvolvimento tecnológico do CONTROLAB-AGV, certamente será necessária a migração do barramento ISA, utilizado atualmente na interface, para os atuais barramentos PCI (*Peripheral Component Interconnect*) ou PXI (*PCI Extensions for Instrumentation*).

Para permitir maiores velocidades do CONTROLAB-AGV, será necessária a utilização de sensores mais rápidos na implementação da bengala eletrônica.

## Referências Bibliográficas



BARSHAN, B.; DURRANT-WHYTE, H.F. An Inertial Navigation System for a Mobile Robot, In: Proceedings of the 1999 IEEE International Conference

on Robotics and Systems, Yokohama, Japão, p. 2243-2248, jul. 1993.

BORENSTEIN, J.; EVERETT, B.; FENG, L. Where am I? Sensors and Methods for Mobile Robot Positioning, University of Michigan, mar. 1996a.

BORENSTEIN, J.; FENG, L. Gyrodometry: A New Method for Combining Data from Gyros and Odometry in Mobile Robots, In: Proceedings of the 1996 IEEE International Conference on Robotics and Automation (ICRA 1996), Minneapolis, EUA, abr. 1996b.

BROWN, S.; RODE, J. Architeture of FPGAs and CPLDs: A Tutorial, IEEE Design & Test of Computers, v. 13, n. 2, p. 42-57, Summer, 1996.

CARNEIRO, G.H.M.B. et al. An Integrated Request Server Architecture for Tele-commanding the CONTROLAB AGV through Real Time Data and Image, In: Proceedings of the IEEE 42nd Midwest Symposium on Circuits and Systems (MWSCAS), Las Cruces, New México, EUA, ago. 1999.

DUBEY, G.K. Power Semiconductor Controlled Drives, Prentice-Hall International, 1989.

EGGEBRECHT, L.C. Interfacing to the I—BM Personal Computer, 2 ed., SAMS, Estados Unidos, 2001.

HAMBLEM, J.O.; FURMAN, M.D. Rapid Prototyping of Digital Systems: A Tutorial Approach, 2 ed., Kluwer Academic Publishers, 2001.

KÜHNEL, C. **GP2D02 - A Sensor for Distance Depending Control**, Disponível em: <a href="http://www.mcselec.com/download/appnotes/gp2d02.pdf">http://www.mcselec.com/download/appnotes/gp2d02.pdf</a>, Acesso em: <a href="jan.2003">jan.2003</a>

LOPES, E.P. et al. **Application of a Blind Person Strategy for Obstacle Avoidance with the use of Potential Fields**, In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA 2001), Seoul, Coréia, maio 2001.

——. Obstacle Avoidance Strategy Based on Adaptative Potential Fields Generated by an Electronic Stick, In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, Canada, ago. 2005.

MONOLITHIC MEMORIES INC. PAL Handbook, 1983.

SENSCOMP INC. **6500 Series Ranging Modules**, Disponível em:  $\frac{\text{http://www.senscomp.com/specs/6500\%20module\%20spec.pdf}}{2005}, Acesso em: jan.$ 

SERDEIRA, H. ICBM - Interface Controladora de um Braço Mecânico,

Projeto final de Curso, Departamento de Engenharia Eletrônica/UFRJ, Rio de janeiro, RJ, Brasil, maio 1991.

SHARP CORPORATION **GP2D02**, **Compact**, **High Sensitive Distance Measuring Sensor**, disponível em

http://www.acroname.com/robotics/parts/gp2d02.pdf, Acesso em: jan. 2003

SHIRLEY, P.A. An Introduction to Ultrasonic Sensing, Sensors, v. 6, n. 11, nov. 1989.

SNYDER, W. E. Industrial Robots - Computer Interfacing and Control, 1 ed., Prentice-Hall, 1985.

TEXAS INSTRUMENT LTD. Optoeletronics Theory and Practice, McGraw-Hill, 1978.

WESTE, NEIL H. E.; ESHRAGHIAN, KAMRAN Principles of CMOS VLSI Design: A Systems Perspective, 1 ed., Addison-Wesley Publishing Co., 1985.