

Faculdade de Engenharia da Universidade do Porto



**Avaliação de uma Estratégia para o Controlo
Remoto de Experiências**

Vítor Duarte Oliveira Silva

Relatório de Projecto realizado no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Automação

Orientador: Investigadora Principal Maria Teresa Restivo
Co-orientador: Professor Joaquim Gabriel Mendes

Julho de 2008

© Vítor Silva, 2008

A Dissertação intitulada

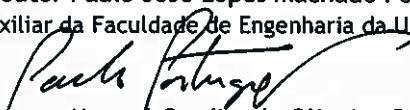
“Avaliação de uma estratégia para o controlo remoto de experiências”

foi aprovada em provas realizadas em 17/Julho/2008

o júri

Presidente

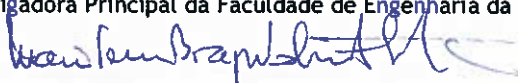
Professor Doutor Paulo José Lopes Machado Portugal
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Manuel Gradim de Oliveira Gericota
Equiparado a Professor Adjunto do Instituto Superior de Engenharia do Porto



Professora Doutora Maria Teresa Braga Valente de Almeida Restivo
Investigadora Principal da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Joaquim Gabriel Magalhães Mendes
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto



O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

Autor - Vitor Duarte Oliveira da Silva



Faculdade de Engenharia da Universidade do Porto

Resumo

O presente relatório de projecto apresenta uma solução original a nível de sistemas de aquisição e de controlo remoto de experiências com interesse académico, procurando criar um exemplo de utilidade para o ensino secundário. Deste modo procura também contribuir para a difusão da experimentação nas camadas jovens, esperando que esta nova contribuição permita criar algum incentivo na sua utilização e exploração.

Hoje em dia a *Internet* oferece inúmeras capacidades de interacção com o utilizador, permite a troca de informação em grande escala e possibilita o controlo remoto de um número elevado de sistemas. Ao nível industrial, é já possível aceder remotamente a uma máquina específica e observar o seu estado e o seu funcionamento, podendo actuar sobre este. Estas capacidades apresentam os seus custos: embora a oferta de sistemas de aquisição de dados e controlo remoto seja elevada normalmente são desenvolvidos para o controlo em grande escala e para a aplicação industrial, atingindo preços elevados.

Por outro lado os sistemas a uma escala mais pequena normalmente estão associados a uma máquina local que faz o seu processamento e a ligação à *World Wide Web*. Neste processo, recorre-se normalmente a um *software* específico para aquisição de dados, geralmente de preço considerável, podendo apresentar incompatibilidades entre sistemas, e não oferecendo a robustez necessária.

Com base nas potencialidades da *Internet* e no aumento das funcionalidades dos sistemas de informação pretende-se pois desenvolver um sistema capaz de publicar, na *Internet*, o estado de um sistema físico e de permitir a actuação do mesmo, de um modo autónomo, flexível e económico, numa alternativa aos sistemas industriais e profissionais já existentes. É pois fundamental apresentar uma solução baseada em *software* livre, como o *Lazarus*, para desenvolver o controlador de uma experiência que se caracterize por ter uma grande compatibilidade com vários sistemas. Foi escolhido um microcontrolador como cérebro do sistema de aquisição.

Com vista a alargar a flexibilidade do sistema, foi implementada uma interface física *Ethernet* que permite a centralização da informação de experiências para permitir o acesso posterior pela *Internet*. A interface a disponibilizar ao utilizador foi realizada com ferramentas de programação *Web*, como *PHP*, *Java Applet*, *Ajax*, *Jquery* e *HTML*, garantindo uma interface intuitiva com uma apresentação convidativa e clara da informação. A solução adoptada para a apresentação de dados, dada a sua grande universalidade, garante o seu suporte e a sua compatibilidade a longo prazo, ao contrário de *software* como é o caso do *LabVIEW*, que necessita de “*plugins*” e actualizações permanentes, não gratuitas, para manter o seu bom funcionamento.

Este trabalho pretende descrever a ideia, o sistema, o seu desenvolvimento e avaliar o desempenho por comparação com outros sistemas existentes, bem como a sua validação junto dos alunos do ensino secundário. O teste comparativo do seu funcionamento tendo por base o

conhecimento existente de outras soluções será feito no LIM, na Faculdade de Engenharia da Universidade do Porto, tendo por referência outros sistemas já implementados com recurso a outras tecnologias.

Abstract

The present report describes a novel solution for remote data acquisition and system control for academic purposes, using a case study of interest for high school physics. It also intends to contribute for the experimental dissemination among the youngest, trying to encourage its use and exploration.

Nowadays, the *Internet* offers many different possibilities for the user interaction, for the large scale information exchange and for the remote control of a huge number of systems. In industrial applications it is even possible to have remote access to equipment, for monitoring its working performance and for actuating on it. But there are costs. Although there is a large supply of data acquisition and remote control systems for industrial application, their prices are considerable. On the other hand systems for applications at reduced scale are associated to a local computer for processing and connection to the *Internet*. In such cases specific *software* for data acquisition and system control is used, at a considerable cost and, very often, causing system incompatibility without offering a robust performance.

Based on the constant new *Internet* technologies and on the increasing information systems performance this work aims at developing a solution for publishing on the *Internet* the state of a physical system with a autonomous, flexible and low cost solution, as an alternative to the existing industrial and commercial ones.

The solution is based on free *software* as is the case of *Lazarus*, for example, for developing a controller for a remote experiment, offering a strong compatibility with other systems. A microcontroller is used.

An Ethernet interface was added for communication with the web server. The user interface was developed with web programming tools such as PHP, *Java* Applet, Ajax, JQuery e *HTML*, providing an intuitive interface. This solution is universally compatible, unlike *software* such as LabVIEW which needs heavy special plug-ins and costly frequent updates.

This work describes the idea, the system and its development and tries to make a preliminary performance evaluation, using other available systems at the Laboratory of Instrumentation for Measurement from Faculty of Engineering of University of Porto.

Agradecimentos

Em primeiro lugar queria agradecer à minha namorada, Ana Costa, pelo incentivo e companhia, dentro e fora da faculdade.

Aos meus orientadores, Investigadora Principal Maria Teresa Restivo e Professor Joaquim Gabriel Mendes, pelo incentivo, orientação e recursos disponibilizados durante a realização deste projecto, nomeadamente o LIM.

A oportunidade concedida pelo projecto Experimenta@FEUP para a realização deste trabalho.

Aos meus colegas de curso que sempre me acompanharam, ajudaram e divertiram enquanto aluno desta mui nobre faculdade, Hugo Faria, Rui Pena, Rui Cachorreiro.

Aos frequentadores da sala Chefax, Carlos Oliveira, Nuno Ildefonso, Daniel Pina, Luís Rei, Filipe Carvalho, André Moreira, António João, Eduardo Sousa, André Tavares, João Leal e claro à Natália Costa que sempre tornaram os momentos passados na faculdade um verdadeiro convívio.

Agradeço também aos meus companheiros de trabalho do LIM, Danilo, Helena, Tânia, Carmen, Nuno e Isa pelo bom ambiente de trabalho que proporcionaram.

Índice

Capítulo 1	1
Introdução	1
1.1 Relevância dos Laboratórios Remotos.....	1
1.2 Motivação	2
1.3 Organização do trabalho.....	2
Capítulo 2	5
Situação Actual	5
2.1 Sistemas actualmente usados na FEUP.....	5
2.2 Outros sistemas	8
2.3 Sistemas alternativos de aquisição de dados	10
2.4 Conclusão	14
Capítulo 3	15
Experiência Remota - Pêndulo Gravítico	15
3.1 Introdução.....	15
3.2 Pêndulo Gravítico.....	16
3.3 Requisitos para actuação do Pêndulo Gravítico	18
Capítulo 4	21
Desenvolvimento do Protótipo	21
4.1 Protótipo Mecânico	21
4.2 Equipamento de sensorização e actuação.....	26
4.3 Solução de <i>hardware</i> de aquisição de dados.....	32
4.4 <i>Software</i>	39
Capítulo 5	49
Conclusões e trabalho futuro	49
Referências	53

Lista de figuras

Figura 2.1 - Interface <i>Web</i> de uma experiência, desenvolvida em <i>Labview</i>	6
Figura 2.2 - Estrutura actual.....	6
Figura 2.3 - Interface para programação remota de microprocessador.....	7
Figura 2.4 - Incompatibilidade da interface de <i>Booking</i> com o <i>browser Firefox</i>	8
Figura 2.5 - Diagrama de blocos do sistema pretendido	8
Figura 2.6 - Oscilador Harmónico remoto (Univ. <i>Purdue</i> , USA).....	9
Figura 2.7 - <i>PICDEM.net 2</i> , placa de desenvolvimento da <i>Microchip</i>	10
Figura 2.8 - Controlador <i>ADAM - 6022</i>	11
Figura 2.9 - <i>ADAM Ethernet Utility</i>	11
Figura 2.10 - <i>Labjack UE9</i>	12
Figura 2.11 - Placa do <i>ETHERNUT</i>	13
Figura 2.12 - Estrutura a usada para o sistema de aquisição de dados	14
Figura 3.1- Principais posições do pêndulo.....	16
Figura 3.2- Relação período - comprimento	17
Figura 4.1 - Protótipo virtual V1.0.....	21
Figura 4.2 - Protótipo experimental provisório.....	22
Figura 4.3 - Modelização do pêndulo em <i>Solid Works</i>	23
Figura 4.4 - Suportes de <i>drive</i> e de travão electromecânico.	23
Figura 4.5 - Desenho do protótipo final.....	23
Figura 4.6 - Protótipo: suporte do pêndulo, sensorização e sistema de actuação.	24
Figura 4.7 - Travão electromecânico - suporte para ajuste de precisão.....	25

Figura 4.8 - Actuador para transporte do pêndulo para posição inicial: a) suporte e ajuste vertical, b) ajuste horizontal	25
Figura 4.9 - Pormenor do sistema de amortecimento.....	26
Figura 4.10 - Efeito de Hall	26
Figura 4.11 - Sensor Potenciômetro.....	27
Figura 4.12 - Erro de linearidade	27
Figura 4.13 - Codificador Óptico.	28
Figura 4.14 - <i>Encoder Omron</i>	28
Figura 4.15 - Esquema de funcionamento da <i>latch</i>	29
Figura 4.16 - Actuador magnético	29
Figura 4.17 - Circuito de accionamento da <i>latch</i>	30
Figura 4.18 - Circuito de accionamento da <i>Drive</i>	31
Figura 4.19 - Diagrama de blocos do SAD.....	32
Figura 4.20 - Microcontrolador <i>ATmega8</i>	33
Figura 4.21 - Trama USART	35
Figura 4.22 - <i>Xport Direct</i>	36
Figura 4.23 - Sinais do <i>encoder OMRON E6B2</i>	37
Figura 4.24 - Circuito de isolamento entre sinais do <i>encoder</i> /circuito de aquisição.....	37
Figura 4.25 - Sinais provenientes do <i>LS7184</i> ; em cima o sinal de <i>clock</i> ; em baixo o sinal de sentido	38
Figura 4.26 - Circuito Electrónico associado ao <i>ATmega8</i>	38
Figura 4.27 - Diagrama de entradas e saídas do microcontrolador	39
Figura 4.28 - Estrutura de <i>software</i>	40
Figura 4.29 - Fluxograma de funcionamento do programa principal do microcontrolador.	41
Figura 4.31 - Estrutura da trama enviada do microcontrolador para o <i>Lazarus</i>	43
Figura 4.30 - Funcionamento das rotinas de interrupção do microcontrolador	42
Figura 4.32 - Algoritmo de “ <i>Parsing</i> ” implementado em <i>Lazarus</i>	43
Figura 4.33 - Servidor e interface local da experiência.....	44
Figura 4.34 - Algoritmo de implementação da aplicação <i>Lazarus</i>	46
Figura 4.35 - Interface <i>Java</i>	47
Figura 4.36 - Gráfico da interface <i>Java</i>	47

Lista de tabelas

Tabela 3-1 - Módulo “Movimentos Oscilatórios” do programa do 12º ano de escolaridade	16
Tabela 3-2 - Variáveis.....	18
Tabela 3-3 - Actuadores, Detectores e Transdutores	18
Tabela 4-1 - Valores dos ganhos em corrente dos transístores.	30
Tabela 4-2 - Tabela de funcionamento do motor da <i>Drive</i> de <i>cd</i>	31
Tabela 4-3 - Características do protocolo <i>RS-232C</i>	35
Tabela 4-4 - Tabela de associação dos pinos com os respectivos elementos do sistema	39
Tabela 5-1 - Tabela de análise de custo.....	50

Abreviaturas e Símbolos

Lista de abreviaturas (ordenadas por ordem alfabética)

A/D	Conversor Analógico Digital
CICA	Centro de Informática Professor Correia de Araújo
FEUP	Faculdade de Engenharia da Universidade do Porto
LIM	Laboratório de Instrumentação para a Medição
HTML	HyperText Markup Language
PCI	Peripheral Component Interface, um interface físico de comunicação
ROV	Remotely Operated Vehicles
USB	Universal Serial Bus, um interface físico de comunicação

Lista de símbolos

α	Ângulo
T	Período
g	Aceleração da gravidade

Capítulo 1

Introdução

O interesse dos laboratórios remotos ao longo da última década tem vindo a aumentar, principalmente a nível do ensino, não só no superior, mas também do secundário.

Com a facilidade de acesso às tecnologias de informação e comunicação (TIC) hoje verifica-se estar ao alcance das instituições de ensino aquilo que antigamente era só possível em grandes laboratórios e instituições. São exemplos, a utilização do telescópio espacial *Hubble* e a dos veículos remotamente comandados, *ROV*. Para isso basta que o utilizador esteja familiarizado com um computador e tenha acesso à *Internet* de banda larga.

Nesta perspectiva hoje torna-se possível disponibilizar, em qualquer lugar e a qualquer hora, uma experiência.

1.1 Relevância dos Laboratórios Remotos

Os laboratórios remotos surgiram como resultado de três correntes, das quais a primeira é a evolução tecnológica. Até há uns anos atrás, não era possível transmitir vídeo e outros dados de grande volume de informação pela *Internet* sem elevados custos associados. Por essa razão, existia uma grande dificuldade de implementação deste tipo de sistemas. Com o aparecimento da banda larga e, por consequência, com a vídeo-conferência por exemplo, tornou-se possível essa transferência sem grandes custos. A integração de comunicação digital na aparelhagem (porta série, *Ethernet*, USB, entre outros) também contribuiu bastante para facilitar a construção de laboratórios controlados à distância.

A outra grande motivação foi o elevado custo e a fragilidade de certa aparelhagem face aos limites orçamentais disponíveis para a sua compra. Sendo assim, tornou-se necessário garantir a sua integridade e durabilidade. Através de rotinas de utilização automatizadas, é possível garantir que o processo de utilização seja desempenhado dentro das condições suportadas pelos equipamentos. Desta forma, é garantida uma reduzida taxa de avarias por negligência do utilizador que, nesse caso, apenas precisa de saber como se deve fazer e não como “o fazer”.

Outro factor que tem contribuído para o crescimento dos laboratórios remotos tem a ver com as comunidades científicas. Neste contexto, elementos destas comunidades, mesmo que geograficamente distantes, podem partilhar infra-estruturas.

No âmbito deste trabalho de projecto, pretende-se desenvolver uma solução que ofereça uma estrutura robusta e de baixo custo para acesso remoto à experimentação. A escolha de um tema no âmbito do ensino secundário teve por objectivo disponibilizar experimentação remota e incentivar o seu uso pelos jovens pré-universitários.

Esta solução permite também mostrar como podem os estabelecimentos de ensino, mesmo com recursos limitados, disponibilizar experimentação on-line, contribuindo para o aumento de meios e partilha dos mesmos no ensino experimental. Uma vez que as experiências estão disponíveis 24 horas por dia, 7 dias da semana, o horário de funcionamento está totalmente alargado sem obrigar a novos recursos humanos nos locais onde as mesmas se encontrem.

Outra vantagem que pode estar associada ao uso de laboratórios remotos reside exactamente no seu acesso à distância, facilitando-o assim a pessoas com dificuldades físicas motoras.

O autor deste trabalho não defende, de modo algum, a substituição da vivência real em laboratório, indispensável para a formação em muitas das áreas do conhecimento, mas sim a utilização destes recursos experimentais on-line como um complemento ao ensino experimental presencial, fortalecendo-o, [1].

Conjugando as perspectivas referidas, pretendeu-se desenvolver um sistema que fosse capaz de adquirir dados sobre o funcionamento do pêndulo gravítico, nomeadamente no que toca à sua trajectória quando em movimento. Com os dados recolhidos podem ser também determinadas as outras grandezas (velocidade e aceleração angulares) bem como o valor da aceleração da gravidade.

1.2 Motivação

O autor do trabalho empenhou-se na sua realização convencido de que o seu contributo pode vir a ajudar a cumprir vários dos aspectos acima referidos. Assim, trabalhando uma experiência de interesse no âmbito do ensino secundário espera ajudar a motivar os mais jovens na área experimental, disponibilizar uma solução de custo modesto e, simultaneamente, contribuir para demonstrar as potencialidades da reutilização de equipamento informático obsoleto, [2].

O desenvolvimento deste trabalho revelou-se também um grande desafio dado o seu carácter multidisciplinar, apresentando alguns temas antes desconhecidos para o autor e tendo-lhe permitido, também, integrar-se num ambiente de trabalho de um outro Departamento da Faculdade onde realizou a sua formação.

1.3 Organização do trabalho

O relatório do projecto está dividido em 5 capítulos.

No capítulo 1 é feita uma descrição sumária do interesse dos laboratórios remotos no contexto do ensino, traduzindo o interesse pessoal que o tema despertou e apresentando a organização do relatório.

No capítulo 2 é realizada uma breve descrição da estrutura utilizada pelas experiências on-line disponibilizadas pelo LIM, Laboratório de Instrumentação para Medição da Faculdade de Engenharia da Universidade do Porto. Segue-se uma breve descrição de várias soluções

para aquisição e controlo de dados para experiências remotas, tentando encontrar uma solução baseada em *freeware* que ofereça características de robustez e baixo custo.

No capítulo 3 procura-se justificar a escolha da experiência, o pêndulo gravítico, tendo em conta a importância da física envolvida associada à sua simplicidade, de modo a permitir concentrar a atenção no desenvolvimento da solução *on-line* a realizar num tempo limitado.

No capítulo 4 é descrito o trabalho desenvolvido do ponto de vista do seu projecto mecânico, da respectiva sensorização e do *hardware* e *software* de comunicação e controlo de dados para acesso remoto.

No capítulo 5, em forma de resumo, percorrem-se os pontos fundamentais do trabalho realizado no âmbito da solução apontada. Referem-se melhoramentos a incluir no futuro próximo decorrentes dos testes de validação realizados e da avaliação que ainda foi possível levar a cabo com uma pequena amostra de utilizadores.

Capítulo 2

Situação Actual

Neste capítulo é feita uma descrição resumida da estrutura utilizada para as experiências *on-line* do LIM. Partindo da perspectiva de utilização de *software* livre, segue-se uma breve avaliação de várias soluções para aquisição e controlo de experiências e comunicação para respectivo acesso remoto. Será assim apontada a solução que se mostrou mais conveniente e possível.

2.1 Sistemas actualmente usados na FEUP

Na Faculdade de Engenharia da Universidade do Porto têm sido realizados esforços vários no sentido de disponibilizar laboratórios remotos. A maioria das experiências concebidas no âmbito do LIM estão disponíveis em REMOTELAB (LIM), no endereço <http://elabs.fe.up.pt>.

O conjunto de aplicações, alojadas em REMOTELAB (LIM), [3], oferece ao utilizador 6 experiências:

- *Mechanical Material Characterization;*
- *Straightness Evaluation;*
- *Level Measurement and Control;*
- *Temperature Calibration Procedure;*
- *Meteorological Station Monitorization;*
- *Remote Control of a Porsche.*

A figura 2.1 mostra um exemplo de interface do utilizador disponibilizada, no modo de utilização automática, para a primeira experiência, pertencente ao conjunto anterior.

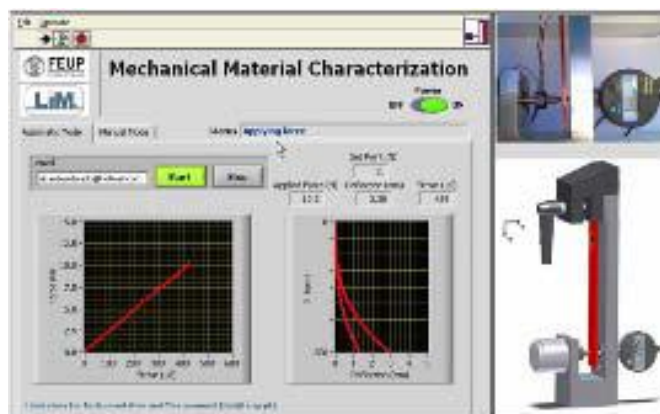


Figura 2.1 - Interface Web de uma experiência, desenvolvida em Labview

Na generalidade dos sistemas o *hardware* de aquisição de dados e de actuação no *set-up* experimental está baseado em vários tipos de placas da *National Instruments* (PCI, USB) e em *software LabVIEW*. Este *software* oferece já a funcionalidade de servidor Web, permitindo disponibilizar automaticamente a interface de utilizador numa página *HTML*, [4].



Figura 2.2 - Estrutura actual

Cada experiência tem um computador dedicado, o servidor local, para correr as aplicações e receber os dados adquiridos pelo *hardware* específico da sensorização de cada *set-up* experimental. Estes dados são processados pela aplicação dedicada e apresentados na interface do utilizador. Todos os sistemas físicos são observados através de vídeo em tempo real dando ao utilizador um maior realismo da experiência, figura 2.2. Tem sido uma preocupação constante o desenvolvimento de experiências em que possa ser observada a mudança de estado de algum dos seus componentes de forma a aumentar o realismo da experiência e o interesse do utilizador.

Para a generalidade das experiências concebeu-se um sistema de envio de dados por *email*, sempre que o utilizador tenha concluído a realização de uma experiência e introduzido o seu endereço. Esses dados devem posteriormente servir para tratamento e validação de resultados e conceitos.

Para envio de imagem existem duas soluções; numa delas utiliza-se o recurso a câmaras *IP*, na outra, câmaras *Web*. As câmaras *IP*, devido à sua qualidade, são utilizadas em experiências que exibam maior dinâmica.

Existe ainda um servidor principal alojado no Centro de Informática Professor Correia de Araújo (CICA), que gere os acessos às experiências e à apresentação das páginas *HTML* das

interfaces, lançadas pelo servidor local. Uma aplicação de *Booking* desenvolvida para a plataforma *Moodle* controla os acessos permitindo um sistema de reservas de intervalos de tempo de uma hora, [5].

Outra plataforma de controlo remoto disponível, integrada no projecto *Labs-on-the-Web*, é a bancada virtual de microprocessadores.

Essa plataforma permite aos utilizadores fazerem *upload* de um programa para um microprocessador, da família 80C51, com o intuito de testar e validar a sua funcionalidade. A interface de utilizador deste sistema é apresentada na figura 2.3.

Os utilizadores podem assim preparar um trabalho para depois ser terminado durante a formação (acção de formação, aulas, etc.), ou então terminar algo que não tenha sido conseguido na situação inversa. Esta plataforma também recorre a *software LabVIEW* para desenvolver a interface do utilizador. O *hardware* que implementa a experiência está baseado nas plataformas *NI Elvis*, da *National Instruments*.

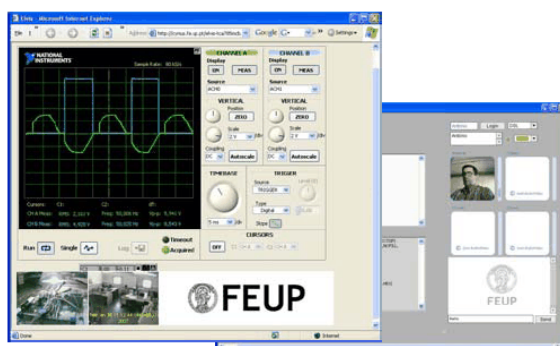


Figura 2.3 - Interface para programação remota de microprocessador.

Os sistemas baseados em *LabVIEW* apresentam a vantagem de permitir facilmente projectar e desenvolver, do ponto de vista do *software*, as aplicações que se pretendam desde que utilizando o *hardware* específico em simultâneo com as respectivas interfaces. Visto ser um *software* que utiliza uma linguagem de programação por blocos de alto nível, permite adicionar vários módulos para tratar e apresentar os dados, e recorrer a um grande número de instrumentos virtuais da sua biblioteca residente, que facilita múltiplas funções como a de representação gráfica, de tratamento matemático, etc. A sua estrutura permite exportar dados para uma vasta gama de formatos, que vão desde o *Office* até ao *HTML*.

Devido a todas estas potencialidades o *software LabVIEW*, como outros *software's* comerciais, exige grandes capacidades de processamento, obrigando o utilizador a instalar ferramentas específicas para uso das suas potencialidades. Requer também constante atenção às funcionalidades de servidor *Web*, relativamente a problemas de compatibilidade com os diferentes *browsers*, com os diferentes antivírus e as suas actualizações, etc. Presentemente estes problemas estão a ser estudados em colaboração com os serviços técnicos do CICA.

Por fim, refira-se ainda os problemas de incompatibilidade, com certos *browser's*, que se têm verificado a nível da interface de *Booking*, de que é um exemplo o representado na figura 2.4.

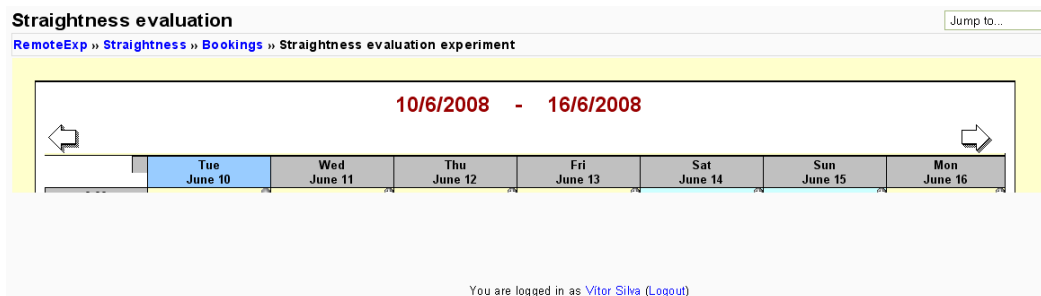


Figura 2.4 - Incompatibilidade da interface de *Booking* com o browser *Firefox*.

O sistema que se idealizou como alternativo aos apresentados anteriormente, não necessita de um computador local para adquirir os dados da experiência. Dispensa também o recurso ao *software LabVIEW*, visto que se pretende fundamentar em ferramentas *open source* e *freeware*. Na figura 2.5 pode-se observar a estrutura pretendida para a colocação de experiências *on-line*.



Figura 2.5 - Diagrama de blocos do sistema pretendido

Os dados da experiência são processados pelo sistema de aquisição e controlo que os converte para o formato *TCP/IP*, formato usado para transmitir os dados até ao utilizador. O *software* que será executado no servidor será desenvolvido em *Lazarus*, efectuando o tratamento dos dados. Estes serão depois fornecidos à aplicação *Java Applet* que os apresentará ao utilizador.

2.2 Outros sistemas

Existem outros Laboratórios Remotos na *Web* que disponibilizam variadas experiências que, no entanto, revelaram problemas de execução. Um exemplo concreto é o “*Elab*” da Universidade Técnica de Lisboa. Este laboratório remoto utiliza ferramentas como o “*Java Web Start*”, como plataforma de acesso às experiências. Contudo, após descarregar os ficheiros e instalar o programa não foi possível aceder às experiências, não tendo sido possível analisar o modo de funcionamento deste tipo de plataforma, [6].

Outro laboratório remoto analisado foi o do Departamento de Física da **Universidade Purdue**, sediada nos Estados Unidos. Este laboratório disponibiliza uma série de experiências em que o seu acesso não é em tempo real mas baseado em vídeos e dados previamente adquiridos e armazenados em base de dados. A figura 2.6 dá um exemplo das experiências disponíveis.

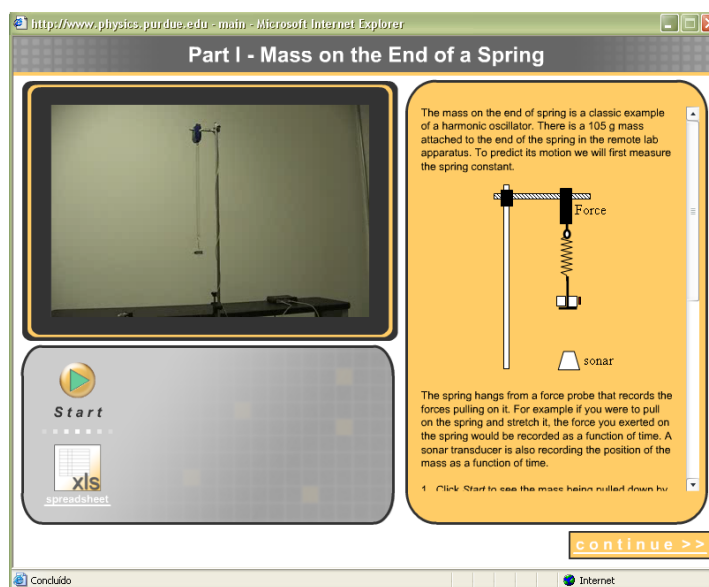


Figura 2.6 - Oscilador Harmónico remoto (Univ. *Purdue*, USA).

A partir da informação recolhida torna-se notório que é necessário o desenvolvimento de novas soluções de baixo custo, elevada fiabilidade e preferencialmente em *software* livre, que exijam poucos recursos do lado do utilizador a nível de processamento e troca de dados, [7].

2.3 Sistemas alternativos de aquisição de dados

Dada a evolução tecnológica, o número de Sistemas de Aquisição de Dados (SAD) tem vindo a aumentar. De acordo com os objectivos pretendidos, efectuou-se uma contextualização das alternativas existentes no mercado, capazes de cumprir o conjunto de requisitos especificados:

- Compacto
- Autónomo (*Stand Alone*)
- Baixo custo
- Robusto
- Modular
- Comunicação *Ethernet*

Apresentam-se seguidamente algumas das soluções mais significativas encontradas.

2.3.1 Microchip

A *Microchip* é uma empresa que produz microcontroladores de baixo custo, sendo conhecida principalmente pelos seus produtos *PIC*.

O produto *PICDEM.net 2*, apresentado na figura 2.7, está equipado com um microcontrolador 18F97J60, porta *Ethernet* embutida, permitindo implementar a comunicação *TCP/IP* (inclui *stack TCP/IP*) e um servidor *Web*, [8]. Apresentando as seguintes características:

- Conector RJ-11 - conector para programação da placa via *MPLAB ICD 2* para programação *IN-SYSTEM*.
- 70 portas *I/O*;
- 16 Entradas de *A/D*;
- Protocolo *I²C*;
- 2 Temporizadores 8 *bits* e 3 de 16 *bits*.



Figura 2.7 - *PICDEM.net 2*, placa de desenvolvimento da *Microchip*

Comentários

Existe muita informação disponível na *Web*, tornando mais fácil o desenvolvimento de uma aplicação para este tipo de sistemas. Estes microcontroladores suportam diversos protocolos de comunicação (*RS-485*, *RS-232*, *SPI* e *Ethernet*). Esta placa inclui também uma

área de prototipagem o que facilita bastante a implementação, com vantagens a nível de ruído. Permite a programação em diferentes tipos de linguagem (C, Basic, etc.).

2.3.2 Advantech

A Advantech tem alguns dispositivos com interface *Ethernet* bastante robustos para aplicações industriais.

O produto ADAM-6022, figura 2.8, é módulo de I/O que inclui um controlador PID. As características do dispositivo são as seguintes:

- 3 Entradas analógicas diferenciais (± 10 VDC, 0 ~ 20 mA, 4 ~ 20 mA);
- 2 Saídas analógicas, com as gamas: 0 ~ 10 VDC, 4 ~ 20 mA, 0 ~ 20 mA;
- 2 Entradas digitais;
- 2 Saídas digitais;
- Interface *Ethernet* 10/100 Base T
- Suporta MODBUS/TCP, TCP/IP, UDP e HTTP;

O dispositivo vem acompanhado com o software de configuração e monitorização, o ADAM Ethernet Utility (figura 2.9), [9].



Figura 2.8 - Controlador ADAM - 6022

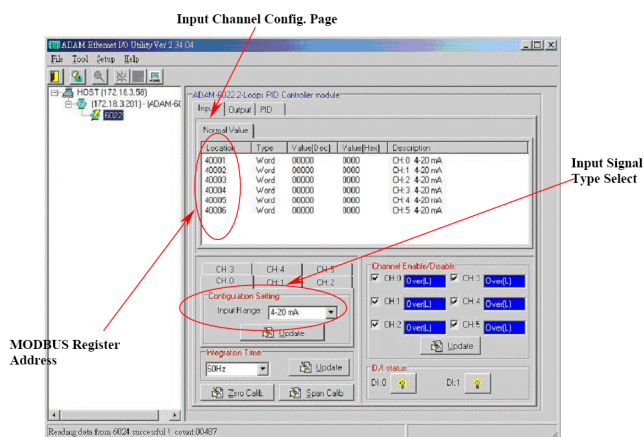


Figura 2.9 - ADAM Ethernet Utility

Comentários

Esta solução revela potencialidades para aplicações de aquisição remota, apresentando, no entanto, como desvantagem, um número limitado de entradas e saídas.

Também é de notar a facilidade de utilização por parte do utilizador, que apenas necessita de instalar o *software ADAM Ethernet Utility*, e executá-lo. O armazenamento de dados é feito em ficheiros de texto.

2.3.3 Labjack

A empresa LABJACK comercializa uma série de módulos de aquisição com ligação *USB* ou *Ethernet*. O produto designado por UE9, apresentado na figura 2.10, revela as seguintes características:

- 14 Entradas analógicas (12-16 Bits);
- ± 5 ou 0-5 V no máximo, para Entradas Analógicas;
- 2 Saídas Analógicas (12-Bit, 0-5 Volts);
- 23 I/O Digitais;
- Até 2 Contadores (32-Bits cada);
- Suporta Modbus TCP/IP.



Figura 2.10 - Labjack UE9

As entradas analógicas podem ser configuradas individualmente como unipolares, em 4 gamas diferentes (0-5 V até 0-0.625 V), ou bipolares, de ± 5 V, com uma resolução máxima de 16 bits e uma taxa de amostragem de 370 amostras por segundo.

As portas digitais podem ser configuradas individualmente como entradas ou saídas, entre 0 e 5V. Possui ainda *FIO*, *Flexible IO*, que podem ser convertidas em

temporizadores ou contadores por *software*.

A alimentação pode ser fornecida via *USB* ou, no caso de uso de *Ethernet*, através de uma fonte *DC*.

Este equipamento suporta o protocolo *MODBUS TCP*, podendo ligar-se facilmente a um equipamento que faça uso desta linguagem sem recurso a qualquer *driver*.

O dispositivo vem equipado com o *software* que permite configurar e testar o UE9, denominado *LJ Control Panel*. No que concerne à aquisição dos dados, são disponibilizadas 3 opções de recolha: uma via *MODBUS*, usando o protocolo desenvolvido na indústria. Outra opção usando funções em linguagem C, incluindo a biblioteca *LabJackUD.h*, que permite ao programador aceder ao conjunto de funções que efectuem a aquisição dos dados das diferentes entradas. A última opção é baseada no recurso a um conjunto de comandos enviados directamente para o processador do dispositivo, ditas tramas de 18 bytes. Estas tramas contêm comandos reconhecidos pelo processador que fazem com que este responda

de forma a devolver os dados pretendidos. Estes comandos permitem configurar a porta do dispositivo, limpar o *buffer*, detectar aparelhos ligados na rede, ler ou escrever numa porta, ler um temporizador ou contador, ler múltiplas entradas ou escrever múltiplas saídas (apenas por USB), ler/escrever/apagar em posições de memória, fazer o *reset* ao sistema. O dispositivo suporta ainda protocolos série, tais como *SPI*, *I²C* e *RS-232C*, permitindo a comunicação com outros micros se necessário [10].

Comentários

De todos os dispositivos analisados acima, este é o que apresenta a melhor configuração e um preço bastante acessível para a oferta que disponibiliza. Tem a grande vantagem de suportar o protocolo *MODBUS TCP/IP*, bastante documentado e suportado por uma vasta gama de produtos existentes no mercado. A possibilidade de incluir a biblioteca *LabJackUD.h*, permite o desenvolvimento de programas de aquisição baseado em qualquer *software*, desde que utilize a linguagem *C* como base, dando flexibilidade ao sistema.

Apresenta um elevado número de entradas, saídas e contadores, com resoluções de 16 *bits* na entrada analógica e 32 *bits* no contador. O tempo de amostragem varia com o número de entradas utilizadas (1,2 a 45,6 *ms*).

2.3.4 Ethernut

O ETHERNUT é baseado em *software open source*, desenhado para aplicações com *Ethernet* embutida, figura 2.11. O sistema utiliza um microcontrolador da *ATMEL*, o *ATmega128*. Possui uma interface de expansão onde podem ser ligados outros dispositivos.

Para conseguir realizar todas estas funções, o sistema é constituído por:

- Um sistema operativo (*Nut/OS*) e uma *stack* (*Nut/Net TCP/IP stack*);
- Porta de expansão de 64-pinos, com bus de 8-bit;
- 3 portas *I/O* de 8 *bits* programáveis;
- 2 Encaixes de 10 pinos para *in-system programming*, de forma a programar a *Flash* e a *EEPROM* (*JTAG* ou *ISP*);
- 1 Ranhura de 20 pinos fêmea com 8- conversores A/D de 10-bits para entradas analógicas.

A sua aplicação foca-se em áreas como:

- Sensores em rede;
- Equipamento de monitorização remota;
- Diagnóstico remoto;
- Aplicações *Ethernet* Industriais;
- Controlo de edifícios e de habitações.

Apesar de oferecer mais do que um *bootloader*, por *RS-232* ou *Ethernet*,



Figura 2.11 - Placa do ETHERNUT

inicialmente o programa tem que ser carregado via JTAG ou ISP.

O *ATmega128* fornece 128 Kbytes de memória não volátil que pode ser usada como memória de programa ou memória *read-only*. A placa está ainda equipada com uma memória *SRAM* de 32 Kbytes para armazenamento de dados, e uma *EEPROM* de 4 Kbytes, tipicamente usada para guardar informação relativa a configurações, mas que pode ser usada para guardar dados, [11].

Comentários

Trata-se de um sistema compacto e de baixo custo, utilizando um integrado bastante familiar e bem documentado. A placa revela robustez e apresenta um número elevado de portas de entrada e saída. Os conversores A/D oferecem uma resolução de 10 *bits*, o que é relativamente reduzida para a generalidade das aplicações. Quanto aos custos associados, são relativamente baixos se tivermos em conta alguns sistemas analisados anteriormente.

2.4 Conclusão

Os sistemas de aquisição de dados apresentados anteriormente constituem uma alternativa à solução baseada em *software LabVIEW* e placas proprietárias referidas no subcapítulo 2.2. Apresentam como vantagem o seu baixo custo, embora necessitem de um maior tempo de desenvolvimento de *software*.

Neste projecto optou-se por seguir uma estrutura de aquisição de dados semelhante à do produto *ETHERNUT*. Tendo características à medida das necessidades da experiência a solução escolhida apresenta um baixo custo, pequena dimensão, baixo consumo, programável em *software* livre e o autor do trabalho tem já algum conhecimento do produto da família *ATmega*, constituinte do produto *ETHERNUT*. A estrutura usada pode ser vista na figura 2.12.

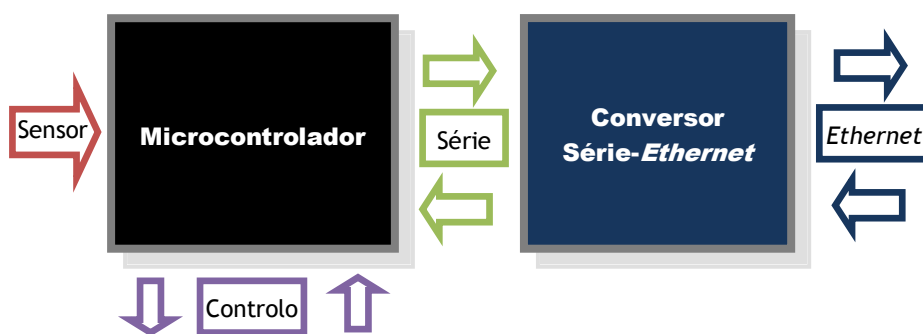


Figura 2.12 - Estrutura a usada para o sistema de aquisição de dados

Capítulo 3

Experiência Remota - Pêndulo Gravítico

Neste capítulo é fundamentada a escolha da experiência e feita uma breve abordagem ao tipo de ensaio experimental que se procura colocar on-line, utilizando a solução para a qual tenta contribuir, o presente trabalho de projecto.

3.1 Introdução

No momento presente e nas actividades do LIM, na vertente de Laboratórios Remotos, existe a preocupação de encontrar soluções alternativas baseadas em *hardware* de baixo custo e *software* gratuito, robustas e estáveis, que conduzam a interfaces amigáveis e leves. A FEUP tem também desenvolvido algum esforço na criação de um portal educacional, voltado para os alunos do secundário, onde se procuram reflectir resultados experimentais na área das engenharias que possam ser apelativos para esses jovens e que promovam o conhecimento e o gosto pela experimentação, [12], [13].

O período de tempo para realização do projecto, teste, validação e a escrita do presente trabalho apresentava-se manifestamente curto e por esse motivo era imperativa uma escolha criteriosa da experiência.

No LIM muitas das actividades experimentais contemplam a medição de grandezas físicas. Por esse motivo, também porque a experiência do pêndulo gravítico envolve princípios bem conhecidos e importantes nas bases dos conhecimentos da Mecânica e finalmente porque o seu *set-up* experimental não se afigurava de elevada complexidade, foi a experiência seleccionada.

Resumindo, a importância do pêndulo gravítico na história, pela sua natureza mecânica, a aparente facilidade na sua automatização sem se preverem grandes custos e o facto desse tema fazer parte do programa escolar do secundário, no 12.º ano, na unidade I - Mecânica, da disciplina de Física, foram os motivos fortes para a escolha.

No referido programa, [14], estão previstas 4 aulas para o módulo sobre “Movimentos Oscilatórios”, que colmata com trabalho laboratorial sobre o Pêndulo Gravítico, tabela 3.1.

Tabela 3-1 - Módulo “Movimentos Oscilatórios” do programa do 12º ano de escolaridade

Movimentos Oscilatórios (4h)	Temas
	Lei de <i>Hooke</i> e equação do movimento harmónico simples. Características de um oscilador harmónico simples: período, frequência e frequência angular; elongação e amplitude. Velocidade e aceleração de um oscilador harmónico simples. Energia de um oscilador harmónico simples. Movimento harmónico amortecido. Física em acção.
Trabalho Laboratorial	
	Pêndulo gravítico

Genericamente os cálculos envolvidos são pouco exigentes permitindo ao utilizador verificar facilmente a validade dos resultados e aperceber-se da sua real contextualização. Em termos de sinais a adquirir apenas é relevante medir a posição angular do pêndulo ao longo do tempo quando colocado em movimento, o que pode ser realizado instrumentação pouco sofisticada. Os restantes componentes para automatizar o sistema, como veremos mais adiante, são tão elementares que podem ser seleccionados no âmbito de componentes electrónicos obsoletos. Este será ainda um aspecto de relevo da experiência, procurando-se através deste exemplo sensibilizar os jovens para a perspectiva de reutilização de equipamentos e assim incentivar a vontade de colocar a imaginação ao serviço da resolução das dificuldades.

3.2 Pêndulo Gravítico

Um pêndulo é constituído por uma massa solidária com um ponto fixo através de uma haste, livre de oscilar. Quando deslocado da sua posição de equilíbrio fica sujeito a uma força resultante do efeito da gravidade e que o tende a levar, de novo, para a sua posição de equilíbrio, causando a sua constante oscilação em torno desta.



Figura 3.1- Principais posições do pêndulo

Galileu realiza os seus estudos sobre pêndulos no início de 1600 inspirado pelo movimento do candelabro da Catedral de Pisa. Observa então que o período do movimento angular do pêndulo é praticamente independente da amplitude de oscilação bem como da sua massa. Varia sim proporcionalmente à raiz quadrada do seu comprimento. Estabeleceu assim a lei do isocronismo do pêndulo e traduziu esta e outras leis físicas de um modo matemático tendo por isso sido considerado o fundador da Física moderna. Mais tarde, em 1656 Christiaan Huggens patenteia os relógios de pêndulo [15], [16].

Esta experiência deverá pois potenciar a percepção e consolidação dos conhecimentos sobre o pêndulo gravítico adquiridos nas aulas de física. Por isso, de um modo geral, espera-se assim contribuir para dar cumprimento a objectivos fixados no Programa de Física do 12º ano de escolaridade reforçando até alguns deles [14]. Assim:

- Promover o conhecimento de conceitos, leis e teorias físicas;
- Realçar as relações entre ciência e tecnologia aqui potenciada pelo recurso a toda a tecnologia envolvida;
- Desenvolver capacidades de observação, experimentação e abstracção;
- Desenvolver a imaginação e a criatividade na elaboração de trabalhos relacionados com a ciência, esperando que este exemplo de reutilização de equipamentos obsoletos seja uma boa fonte de inspiração.

Uma vez que se pretende que o projecto aqui desenvolvido sirva como um complemento ao ensino, a questão que se coloca primeiramente é: “Como se poderá calcular o valor da aceleração da gravidade com um pêndulo simples?”. Para dar resposta a esta pergunta, e para saber que dados se devem adquirir, temos que ter em consideração o seguinte:

- Que forças actuam no pêndulo;
- Que tipo de movimento o pêndulo executa;
- Qual a variação da velocidade com o tempo;

Está provado que o período de oscilação não depende, nem do ângulo de afastamento do ponto de equilíbrio (para ângulos inferiores a 10°), nem da massa do pêndulo. Esse período depende apenas do comprimento da haste do pêndulo, através de uma relação quadrática, tal como nos é mostrado na figura 3.2, [17].

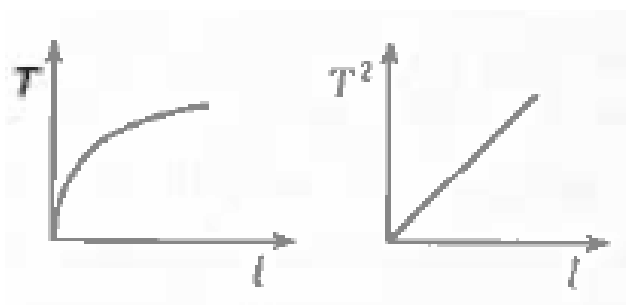


Figura 3.2- Relação período - comprimento

Para pequenas oscilações, o período do movimento pendular é dado pela expressão 3-1.

$$T = 2\pi \sqrt{\frac{l}{g}} \quad (3-1)$$

Na sequência da lógica anterior, o que se pretende saber é o valor da aceleração da gravidade, a variação das suas componentes (radial e normal), a variação da velocidade e a variação da posição do pêndulo ao longo de todo o tempo da experiência. Numa primeira fase apenas será determinado o valor da aceleração gravítica e representada a trajectória do pêndulo. Para isso, é necessário adquirir um conjunto de variáveis que permitam fazer as leituras bem como automatizar o processo. Na tabela 3-2 reúnem-se as variáveis em questão.

Tabela 3-2 - Variáveis

Variáveis
Posição em cada instante
Sentido
Tempo

Neste momento torna-se essencial definir as automatizações que devem ser introduzidas para tornar o trabalho remotamente acessível.

3.3 Requisitos para actuação do Pêndulo Gravítico

A realização da experiência de um modo remoto deverá prever as seguintes acções:

- Transportar o pêndulo da sua posição de equilíbrio até uma posição que corresponda a um afastamento angular de cerca de 10°. Para esse feito é necessário calcular a distância (relativamente à posição de equilíbrio) a que este deve ser levado: $d = l \cdot \sin(10^\circ)$;
- Levar o pêndulo à distância calculada anteriormente e mantê-lo nesta posição;
- Libertar o pêndulo e iniciar a contagem de tempo relativa a um dado número de oscilações;
- Calcular o período médio de cada oscilação;
- A partir deste valor pode ser calculado o valor de g .

Para além da automatização essencial para permitir este procedimento é necessário sensorizar o sistema do pêndulo com um transdutor capaz de adquirir a sua posição em cada instante, e que caracterizaremos mais à frente. A tabela 3-3 resume as necessidades acima enunciadas.

Tabela 3-3 - Actuadores, Detectores e Transdutores

Actuadores	Descrição
Motor	Levar/parar o pêndulo para posição inicial
Travão	Segurar o pêndulo
Detectores	
Fim de curso	Detectar os limites do sistema de transporte
Transdutor	
Posição	Fornecer a posição do pêndulo em cada instante
Sentido	Fornecer o sentido do deslocamento angular

Em suma, neste capítulo procurou-se justificar:

- A escolha da experiência sobre a qual recai o desenvolvimento;
- O modo como será explorada.

Assim, o utilizador ao iniciar a experiência do pêndulo gravítico deverá encontrá-lo já na posição de início de oscilação e em repouso (afastado cerca de 10° da posição de equilíbrio); o movimento inicia-se e também a contagem das oscilações. O transdutor de posição angular fornecerá os valores da posição angular em cada momento durante 10 oscilações e estes serão gravados num ficheiro de texto que se procurará enviar ao utilizador por *email*; no fim da experiência o actuador de deslocamento deverá recolocar o pêndulo na sua posição inicial onde deverá ficar em repouso até nova solicitação do utilizador.

Capítulo 4

Desenvolvimento do Protótipo

Este capítulo procura descrever o trabalho desenvolvido no âmbito do sistema mecânico, das soluções da sensorização do pêndulo e da escolhida para o *hardware* e *software* necessários para a comunicação e controlo de dados permitindo o acesso remoto a este *set-up*.

4.1 Protótipo Mecânico

Para desenvolver o protótipo mecânico, e para definir algumas soluções, analisou-se o funcionamento de um pêndulo improvisado para o efeito. Esta fase exploratória foi complementada com a observação de conteúdos (texto e multimédia) de uma publicação da Porto Editora, [17].

Com base no conjunto de pressupostos resultantes realizou-se o primeiro modelo numa aplicação *Google Sketchup*, figura 4.1.

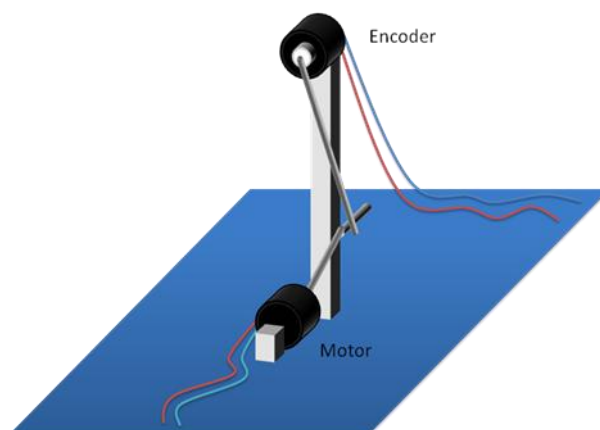


Figura 4.1 - Protótipo virtual V1.0

A solução que se descreverá utiliza meios bastante elementares, muito baseado em componentes retirados de equipamentos obsoletos. No tempo disponível não foi também possível dispor do projecto mecânico realizado nas oficinas do Departamento de Engenharia Mecânica e Gestão Industrial da FEUP. A premência para solucionar esta etapa levou o autor deste trabalho a optar por realizar o *set-up* mecânico pelos seus próprios meios, com base no modelo referido.

Enquanto as peças mecânicas foram sendo elaboradas a solução improvisada, figura 4.2, foi permitindo testar a evolução da solução de *hardware* e *software* para aquisição e dados e disponibilização *on-line*.



Figura 4.2 - Protótipo experimental provisório

Assim foi usada uma barra metálica para suporte. No topo desta está fixado o transdutor de posição angular usado no desenvolvimento do projecto. No seu eixo foi suspenso o pêndulo, constituído por uma barra de material polímero com uma massa no seu extremo. Para actuador escolheu-se um *drive* de *cd* existente num conjunto de componentes eléctricos e electrónico proveniente de equipamento obsoleto.

O projecto do pêndulo foi realizado em *Solid Works* tendo-se procurado estudar as peças que melhor se adaptavam à dinâmica pretendida de acordo com os requisitos da actuação definidos em 3.3. A primeira peça desenhada foi o pêndulo e o seu acoplamento com o eixo do transdutor de posição angular, figura 4.3. Aproveitando o facto do veio do transdutor ser facetado foi fabricada uma peça em latão que permite acoplar simultaneamente o pêndulo ao veio dispensando o uso de uma chaveta. Este método impede que exista movimento relativo (deslize) durante a oscilação do pêndulo.



Figura 4.3 - Modelização do pêndulo em *Solid Works*.

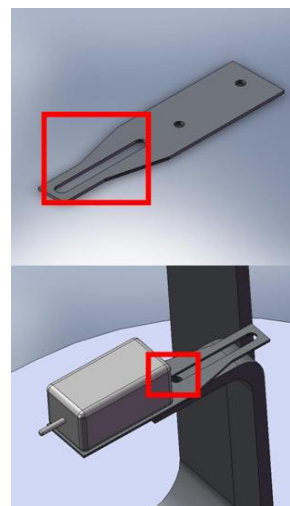


Figura 4.4 - Suportes de *drive* e de travão electromecânico.

Na extremidade inferior da haste do pêndulo foi roscada a massa que foi estimada de em cerca de 450 g, permitindo assim desprezar o peso da barra no cálculo do centro de massa.

Uma outra peça para suporte e para ajuste mecânico foi seguidamente projectada para posicionar com precisão vários elementos (*drive* e travão electromecânico), figura 4.4.

Por último foram definidas as barras de suporte dos elementos, uma para o pêndulo e para o *encoder*, outra para o travão electromecânico, figura 4.5.

Após a escrita deste relatório serão introduzidos elementos mecânicos nos suportes do pêndulo para aumentar a rigidez do mesmo.

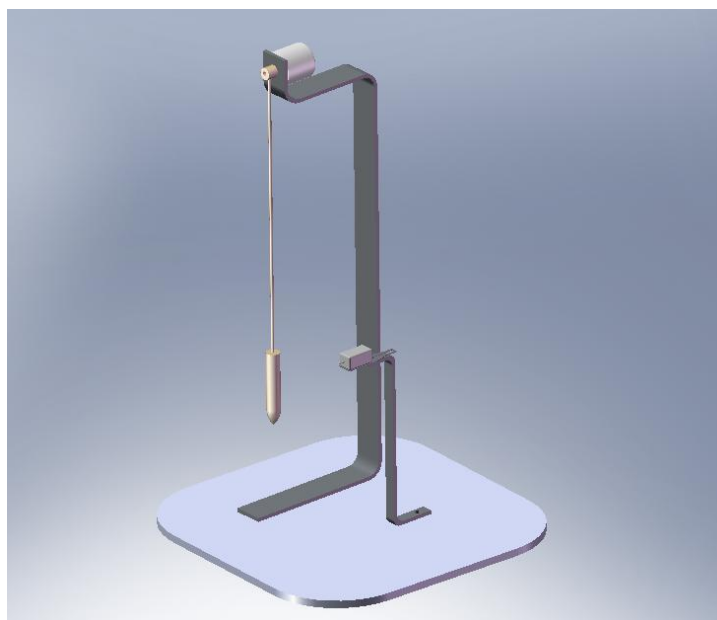


Figura 4.5 - Desenho do protótipo final

Seguiu-se a fase de construção de que resultou o protótipo final apresentado na figura 4.6.

Numa visualização geral do protótipo do pêndulo pode dizer-se que este é constituído por um transferidor, colocado no topo, para dar ao utilizador a noção da grandeza do ângulo de afastamento do pêndulo; um suporte para o pêndulo do tipo barra, em ferro, (750x40x5mm) e outro do mesmo tipo mais pequeno (270x15x5mm), para suporte do travão electromecânico. Do lado esquerdo inferior da figura 4.6 pode observar o actuador que transporta o pêndulo para a sua posição inicial de funcionamento - um *drive* de *cd*'s.

As figuras 4.6, 4.7, 4.8 contêm pormenores de peças de acoplamento utilizadas na montagem.

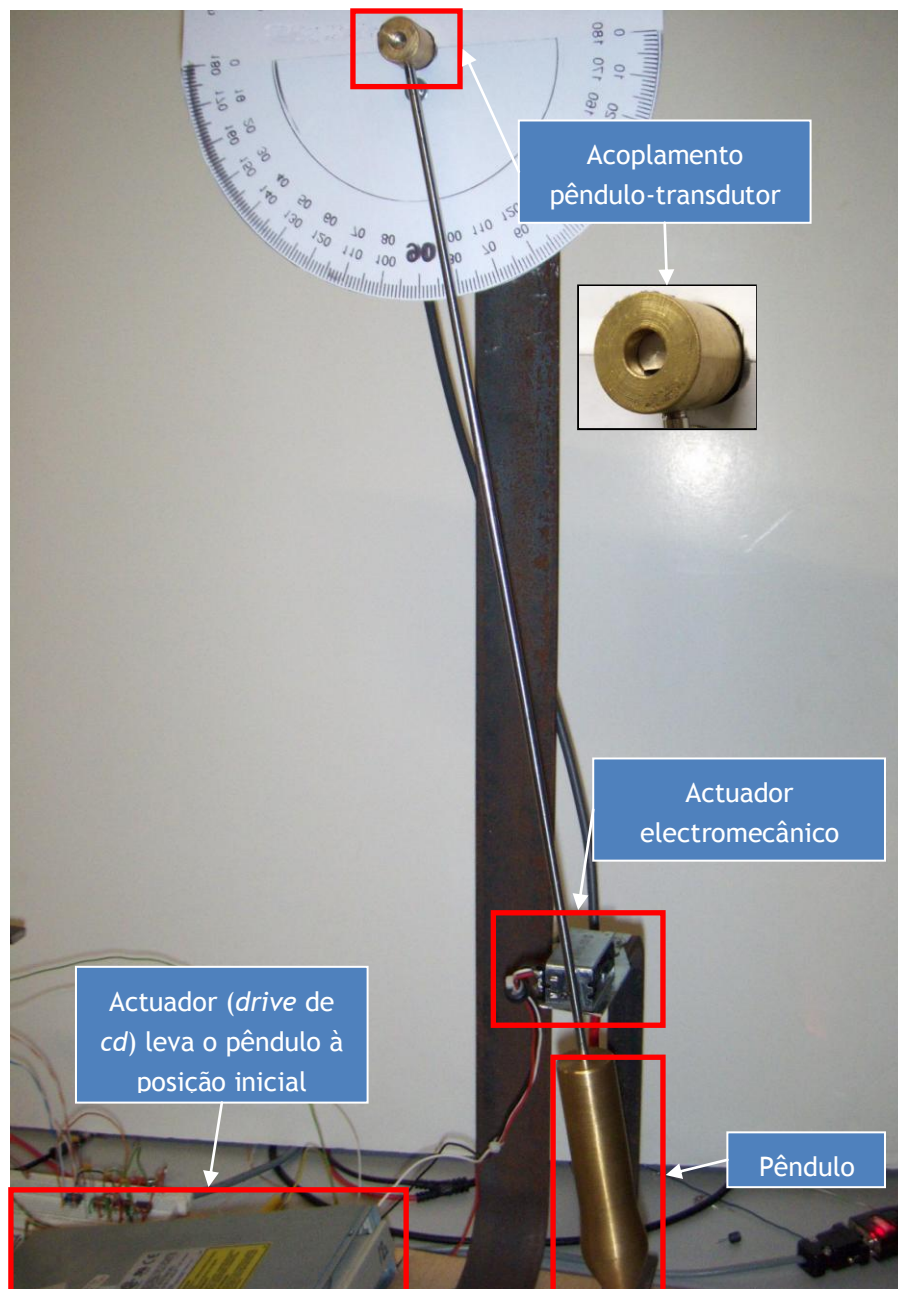


Figura 4.6 - Protótipo: suporte do pêndulo, sensorização e sistema de actuação.

Observe-se a caixa a vermelho na figura 4.7. Este tipo de peça permite afinar a posição do travão electromecânico para que o seu disparo seja optimizado, introduzindo o menor erro possível no movimento do pêndulo.

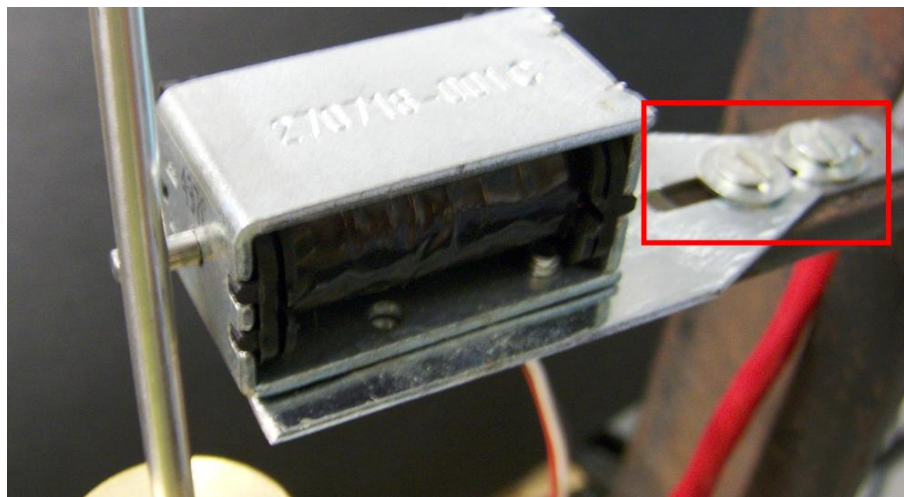


Figura 4.7 - Travão electromecânico - suporte para ajuste de precisão

No caso do segundo actuador, também foi implementado um sistema com capacidade de afinação, baseado em duas peças semelhantes à anterior. Este método permitiu proceder ao posicionamento mecânico deste actuador com dois graus de liberdade, vertical e horizontal, figura 4.8.



a) b)
Figura 4.8 - Actuador para transporte do pêndulo para posição inicial: a) suporte e ajuste vertical, b) ajuste horizontal

A opção de reutilizar equipamentos acarreta sempre dificuldades. De facto, a decisão em aumentar a massa do pêndulo trouxe dificuldades de utilização do *drive* de cd's como posicionador do pêndulo. Assim tornou-se fundamental criar um sistema que amortecesse a pancada do pêndulo no elemento móvel do actuador no processo de paragem. Dado o pouco tempo disponível optou-se por uma solução prática e eficiente: utilizar uns elásticos de borracha com a função de barreira amortecedora, figura 4.9, [18].



Figura 4.9 - Pormenor do sistema de amortecimento.

O sistema foi testado e revelou as funcionalidades necessárias para viabilizar a experiência idealizada. No entanto, está-se consciente da necessidade de alguns melhoramentos a nível de montagem mecânica.

4.2 Equipamento de sensorização e actuação

A disponibilização desta experiência on-line implica dotá-la de elementos (sensores, detectores e actuadores) que permitam a recolha de informação e a actuação através de sinais eléctricos.

4.2.1 Sensorização

Outra etapa importante neste projecto é a que diz respeito à selecção dos transdutor para medição da posição angular, uma vez que se pretende medir posição e desta medição, caso o sinal seja de qualidade, poder-se-á determinar a velocidade e a aceleração angulares. Para isso, recolheu-se alguma documentação que se resume seguidamente.

Sensores de Efeito de Hall

Quando uma corrente eléctrica atravessa um condutor ou semicondutor que se encontra na presença de campo magnético com uma componente normal à direcção da corrente, é gerada uma tensão eléctrica, designada por tensão Hall, normal ao plano definido pela direcção da corrente e da componente do campo magnético normal à da direcção da corrente, figura 4.10.

O uso destes sensores no campo da medição é muito diversificada, podendo ser

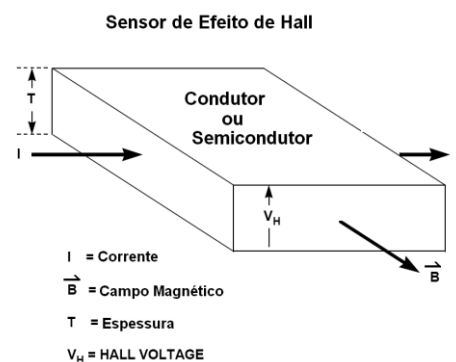


Figura 4.10 - Efeito de Hall

usados na medição de deslocamento, de velocidade angular, etc., [19], [20].

Potenciômetros rotativos

Este sensor funciona como um simples divisor de tensão em que ponto móvel está solidário com o corpo em movimento, cuja rotação se deseja medir: obtém-se uma tensão de saída, e_o , proporcional à tensão de alimentação, e_i , e à deslocação θ do ponto móvel, figura 4.11.

O erro mais significativo com este tipo de transdutores provém do "efeito de carga" e da não linearidade resultante do valor finito da impedância de carga R_L (associada ao aparelho de medição), figura 4.12, [19], [20].

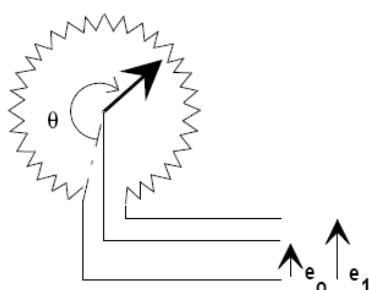


Figura 4.11 - Sensor Potenciométrico

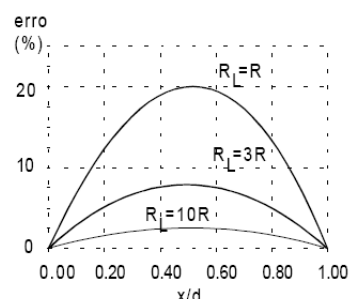


Figura 4.12 - Erro de linearidade

Codificadores Ópticos

Entre os sensores de posição angular mais populares surgem os codificadores ópticos (absolutos e incrementais). Há também dois tipos distintos de acordo com o modo como a informação óptica passa do elemento emissor (fotodíodo) para o elemento detector fototransistor: do tipo barreira e de reflexão.

Por exemplo, no caso de um codificador angular do tipo barreira, um disco perfurado roda solidário com o corpo em rotação, encontrando-se de cada lado deste o elemento emissor e receptor, respectivamente. À medida que o disco roda, a saída do detector alterna entre dois estados de funcionamento de um fototransistor (corte/saturação), gerando um trem de impulsos de acordo com a velocidade do disco.

Os codificadores incrementais têm ainda um segundo par de emissor/detector óptico com o objectivo de definir o sentido de rotação. Em alguns casos existe um terceiro par de emissor/detector óptico utilizado para definir a posição zero, figura 4.13, [19], [20], [21].

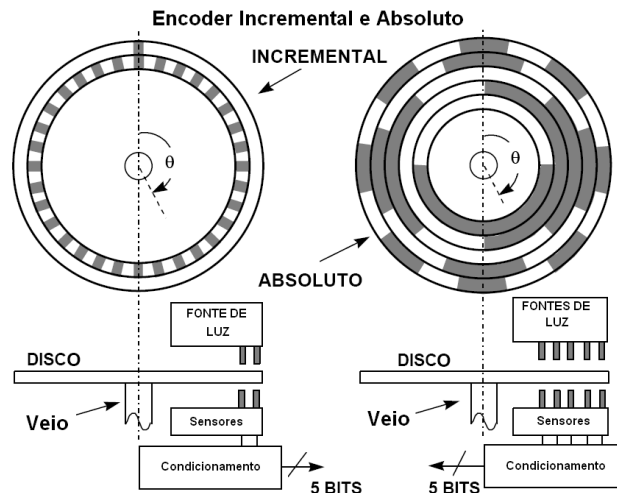


Figura 4.13 - Codificador Óptico.

Escolha do dispositivo

A principal característica a exigir ao dispositivo que será usado para medição angular é a de apresentar um baixo binário, de modo a não se tornar limitativo ao movimento do pêndulo. Em face da aplicação e das solicitações de esforço a que ficará submetido a relação atravancamento/robustez é também uma característica a ser considerada. A qualidade do sinal é também importante para permitir determinar, a partir da medição de posição angular, as respectivas velocidade e aceleração através de uma simples ou dupla diferenciação. Face a todas estas características desejáveis a decisão terá que recair então num dispositivo tipo codificador angular, contrariamente a outras escolhas de mais baixo custo, que nortearam as restantes opções até ao momento.

O codificador incremental *Omron E6B2-CWZ6C*, figura 4.14, apresenta uma resolução de 360 impulsos por revolução, ou seja, um impulso por grau. Possui dois canais (A e B), desfasados, para determinação do sentido. Possui um sinal de zero que é activado sempre que passa num ponto de referência, permitindo eliminação de *offset*.

A grande vantagem deste *encoder* é o facto de ser óptico, não introduzindo nenhuma resistência considerável ao sistema. Aceita uma alimentação entre 5V a 24V DC, com saída em colector aberto. Suporta uma carga máxima, no veio, de 3Kg. A análise de vários codificadores considerados conduziu a esta escolha que satisfaz as características acima consideradas fundamentais. Foi acrescido do facto de a sua aquisição ter sido imediata.



Figura 4.14 - Encoder Omron

4.2.2 Actuação

Contrariamente ao que se verificou no parágrafo anterior, no que diz respeito ao custo da solução escolhida para sistemas de actuação recorreu-se a soluções mais simples e económicas.

Para sistema de travão electromecânico foi usado um solenóide de duplo estado. Uma informação do tipo impulso permite alterar o seu estado, poupando energia e prolongando o tempo de vida.

Para o posicionamento do pêndulo na sua posição inicial de funcionamento (mantida pela acção do travão electromecânico), foi usada uma *drive* de *cd*, proveniente de equipamento obsoleto.

Travão Electromecânico

Na figura 4.15, pode ser observado o esquema de funcionamento deste actuador (*latch* magnética), figura 4.16. Este elemento consome uma corrente de 1 A (12 VDC) em funcionamento.

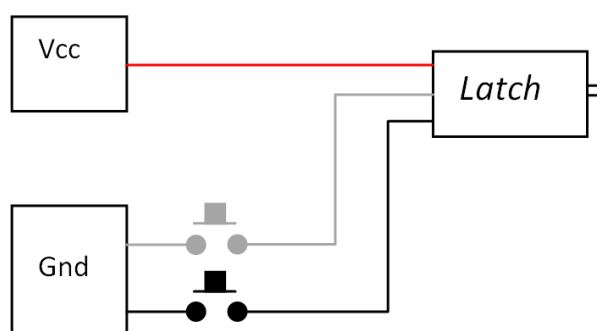


Figura 4.15 - Esquema de funcionamento da *latch*

Como no caso do microcontrolador que será exposto mais à frente a corrente disponível é limitada a 40 mA, foi necessário desenhar um circuito de amplificação auxiliar, com 2 transístores. Um realiza o isolamento da porta do microcontrolador do circuito e fornece a corrente necessária para activar o segundo transístor. Este segundo é um transístor de potência, responsável pela estimulação do actuador magnético.

Uma vez que ao actuar na *latch* é fornecido um impulso, a corrente de pico será maior do que a corrente de funcionamento normal, 1A. Sendo assim escolheu-se o transístor TIP31C para actuar directamente na *latch* uma vez que este suporta um pulso de corrente elevado para baixas tensões. Para isolar e activar este transístor escolheu-se o BC337. Ambos os transístores são do tipo *npn*. Para

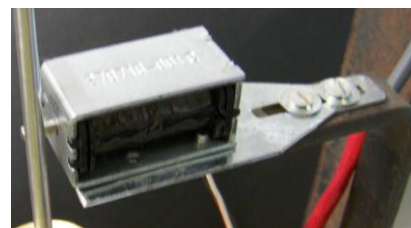


Figura 4.16 - Actuador magnético

dimensionar o sistema retirou-se os seguintes valores das características dos integrados, tabela 4-1.

Tabela 4-1 - Valores dos ganhos em corrente dos transístores.

Transístores	hfe
TIP31C	25
BC337	100

Para garantir um valor de corrente de, pelo menos, 1 A no transístor TIP31C, a corrente de base do mesmo deverá ser:

$$i_b = \frac{I_c}{hfe} = \frac{1}{25} = 0,040 A \quad (4-1)$$

Conhecido este valor de corrente, foi calculada a resistência da base deste transístor, sabendo que a sua alimentação na base será de 5V, [22].

Um procedimento semelhante foi seguido para os cálculos relativos ao transístor BC337. Obteve-se para resistência de base um valor de 330Ω. O microcontrolador fornece o sinal de 5VDC que activa este transístor.

Foram ainda colocadas resistências de *pulldown*, de 1KΩ, no emissor e na base do BC337. O sistema de accionamento da *latch* é composto por dois circuitos conforme a figura 4.17.

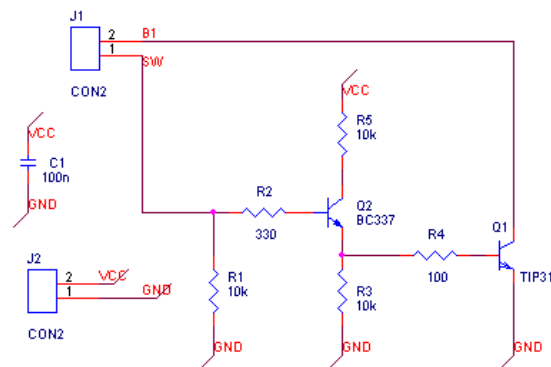


Figura 4.17 - Circuito de accionamento da *latch*

Sistema de posicionamento do pêndulo

Para a actuação deste elemento recorreu-se ao integrado L298, um *driver* de ponte completa, desenhado propositadamente para controlo bidireccional de rotação de um motor. Apesar de este integrado ser sobredimensionado, o seu custo é relativamente baixo assegurando uma maior durabilidade do sistema.

Para efectuar o controlo da *drive* implementou-se um circuito predefinido, figura 4.18, disponibilizado pela folha de características do produto, em que V_{cc} é a alimentação, V_{ss} é o nível lógico 1 e V_{en} activa a ponte.

4.3 Solução de *hardware* de aquisição de dados

A disponibilização on-line de uma experiência implica a disponibilização de todos os parâmetros físicos necessários para a realização da experiência e de transmitir à experiência ordens que permitam a concretização de um procedimento. Isto é, o *hardware* que permite um ‘diálogo’ entre a interface do utilizador e o *set-up* experimental.

Para a constituição de um sistema de aquisição é preciso estabelecer uma estrutura. O autor deste trabalho decidiu escolher a estrutura apresentada na figura 4.19, explicando de seguida o trabalho desenvolvido em cada bloco dessa mesma estrutura.

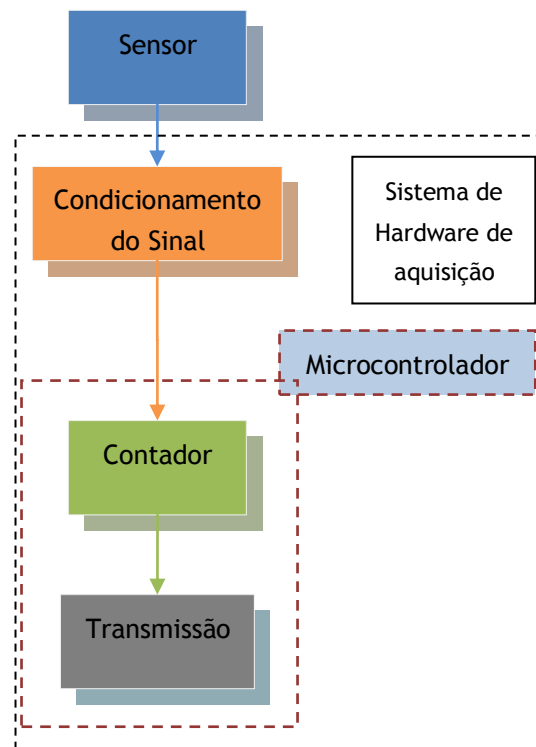


Figura 4.19 - Diagrama de blocos do SAD

4.3.1 Microcontrolador

No parágrafo 2.3 viram-se algumas soluções de baixo custo. Entre elas estava o tipo de solução que se utilizará devido ao tempo limitado para a realização deste trabalho e à familiarização com o funcionamento dos microcontroladores da família *ATmega*. Decidiu-se optar assim microcontrolador *ATmega8*, figura 4.20.

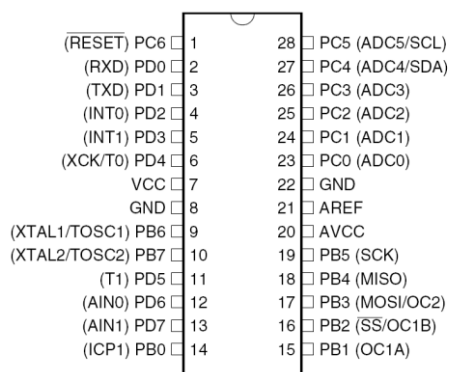


Figura 4.20 - Microcontrolador *ATmega8*

Trata-se de um microcontrolador de estrutura física *PDIP*, de 28 pinos. É composto por um *bus* de 8 *bits* baseado numa arquitectura *AVR RISC*, realizando taxas de transferência de 1 *MIPS*@1 *MHz*. Esta capacidade de processamento deve-se à sua arquitectura que permite executar instruções num único ciclo de relógio, oferecendo simultaneamente um sistema de baixo custo e baixo consumo, como referido no subcapítulo 2.4.

Resume-se seguidamente as suas principais características:

- 130 operações diferentes realizáveis em ciclos únicos de relógio;
- Taxa de processamento (*Throughput*) de 16 *MIPS*@16 *MHz*;
- 8 *Kbytes* de memória *Flash*;
- 512 *Bytes* de *EEPROM*;
- 1 *Kbyte* de *SRAM* interna;
- 1 Contador de 16 *bits*;
- 6 Conversores *A/D* de 10 *bits*;
- 23 Linhas *I/O* programáveis;
- Tensão de alimentação 4,5 a 5,5 *V*;
- Máxima frequência de relógio 16 *MHz*.

Este integrado suporta interrupções, tanto internas como externas (*INT0*, *INT1*) e uma porta *USART*. A baixa resolução do conversor *A/D* não é limitativa para a aplicação em causa.

A memória *FLASH* pode ser reprogramada através da porta *SPI*, ou através de um *bootloader*, instalado no núcleo do chip, via RS-232.

De seguida, será feita uma pequena descrição do funcionamento do *ATmega8* e das principais funções usadas no desenvolvimento do protótipo, [23].

Interrupções

O micro disponibiliza várias interrupções. Cada conjunto de interrupções tem associado um conjunto de *bits* que necessitam estar ao nível lógico 1, juntamente com o *bit Global*

Interrupt Enable, para habilitar as interrupções, em geral. O objectivo é permitir a pronta resposta do micro ao evento que desencadeia a interrupção, [23].

Existem dois tipos de interrupções.

- Interrupções desencadeadas por um evento que activa a *flag* de interrupção. Neste caso, é chamada pelo micro a rotina associada àquela interrupção e a *flag* é libertada por *hardware* ou também por *software*. Quando o CPU retorna de uma interrupção, regressa ao programa principal e executa uma ou duas instruções antes de atender outras interrupções pendentes.
- Interrupções designadas por externas, desencadeadas pelos pinos INT0 e INT1, mesmo que estes estejam definidos como saídas. Este método possibilita a criação de interrupções por *software*. As externas podem ser accionadas por uma passagem de 0-> 1, ou inversamente, de 1-> 0. Isto é configurado no registo MCUCR.

Temporizador/Contador

O micro possui dois temporizadores/contadores de 8 *bits* e um de 16 *bits*. O primeiro, *Timer*, de 8 *bits*, é destinado ao uso geral, podendo funcionar como contador, gerador de frequências, contador de eventos externos, *PRESCALER* de 10-*bits*. O registo responsável pela configuração do *Timer* é o TCCR, onde é possível activar ou desactivar os *bits* necessários para que o temporizador funcione, da maneira e do modo pretendidos. O temporizador pode ser gerado internamente, por via *PRESCALER* ou mesmo através de um relógio externo ligado ao respectivo pino.

A contagem é sempre incremental, não existindo nenhum *reset* do seu valor. O contador atinge o seu valor máximo de contagem e recomeça automaticamente [23].

Porta série

O microcontrolador vem equipado com uma porta *USART* (*Universal Synchronous and Asynchronous serial Receiver and Transmitter*), que permite a troca de dados com um outro microcontrolador ou computador via RS-232.

Suporta tramas até 9 *bits*, com um ou dois *bits* de paragem, e permite a comunicação *full-duplex*, com uma boa taxa de transferência. O relógio que controla o *UBRR* (*USART Baud Rate Register*) funciona como um *prescaler* do relógio interno do microcontrolador (f_{osc}). Para calcular o valor de *UBRR* para uma determinada taxa de transferência (*BAUD*) pretendida utiliza-se a seguinte fórmula:

$$UBRR = \frac{f_{osc}}{16 \text{ BAUD}} - 1 \quad (4-2)$$

Nesta aplicação é usada uma *baud rate* de 38400 bps, ou seja, obtendo-se um de 23 para o registo *UBRR* (equação 4-2).

As tramas têm o seguinte formato, de acordo com as opções escolhidas, figura 4.21.

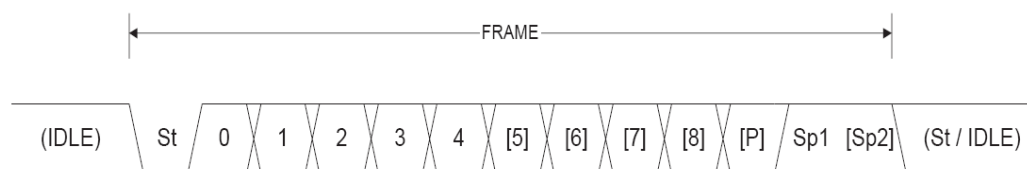


Figura 4.21 - Trama USART

Ao usar esta BAUDRATE (38400 bps) foi tido em conta a frequência do cristal do microcontrolador, 14.7456 MHz, que sendo múltipla da frequência típica de computadores com arquitectura IBM-PC, traduz-se por uma taxa de erros muito reduzida entre os dois elementos [23].

O microcontrolador apresenta ainda outras funcionalidades que não são relevantes para a aplicação em questão.

4.3.2 Comunicação de dados

Para colocar uma experiência on-line é essencial dotar o sistema de aquisição com capacidades de comunicação por Web com o utilizador. O sistema de aquisição e controlo de dados deverá comunicar com o servidor Web através de diferentes protocolos. Neste caso utilizam-se dois tipos de comunicação: RS-232C e protocolo Ethernet.

O protocolo de comunicação RS-232C surgiu nos anos 60. Na tabela 4-3 pode-se ver as características do protocolo RS-232C.

Tabela 4-3 - Características do protocolo RS-232C

Especificação	RS-232C
Drivers por linha	1
Receptores por linha	1
Comp. Max do cabo, mt	150
Taxa de transmissão Max.	1 Mb/s
Tensão de saída Max., Volt	+/-25Vdc
Sensibilidade	+/-3 Vdc
Resistência de entrada, k-Ohm	(3 to 7)

Por último temos o protocolo *Ethernet*, bem conhecido e provavelmente um dos mais usados nos nossos dias. Este protocolo surgiu nos anos 70 na *Xerox Corporation*. Nos anos 80, a *Digital Equipment Corporation*, a *Intel Corporation* e a *Xerox Corporation* formalizaram a descrição do protocolo num documento intitulado “The Ethernet, a *Local Area Network: Data Link Layer and Physical Layer Specifications*”.

Em 1985, o IEEE lançou a sua própria versão do protocolo, muito similar ao anterior, garantindo a compatibilidade entre eles, [24].

Hoje em dia uma grande diversidade dos equipamentos incorpora uma porta *Ethernet*, tornando a sua integração neste ambiente bastante mais fácil.

O sistema *Ethernet* tem 3 elementos básicos:

- O meio físico usado para a troca de sinais entre dois computadores;
- Um conjunto de regras de controlo de acesso ao meio;
- Uma trama *Ethernet*, que é constituída por um conjunto normalizado de *bits*, usada para transportar os dados entre computadores.

Neste trabalho é essencial que a comunicação *Ethernet* garanta a posterior ligação directa à *Internet*. Contudo, o microcontrolador usado, o *ATmega8*, não possui essa interface, tendo apenas uma porta RS-232.

Revelou-se necessário encontrar uma solução para este problema. Uma pesquisa de mercado conduziu à solução baseada num integrado da *Lantronix* capaz de realizar a conversão desejada, a placa *Xport Direct*, apresentada na figura 4.22, [25].



Figura 4.22 - *Xport Direct*

Esta solução possibilita a inserção fácil da comunicação *Ethernet* em circuitos electrónicos através da porta série. Apresenta um custo reduzido e é ideal para transmissões grandes quantidades de dados, onde os microcontroladores se revelam insuficientes devido à sua limitada capacidade de memória e de velocidade de comunicação.

Sucintamente, esta solução reúne os dados recebidos pela porta série, empacota-os e envia-os para a rede *Ethernet* com tramas *TCP* ou *UDP*. Do mesmo modo, quando este equipamento recebe pacotes *TCP* ou *UDP* separa-os correctamente, retirando os dados, e envia-os ao microcontrolador pela sua porta série. Para isso basta definir os parâmetros da comunicação série, num extremo, e os do servidor e a porta, no outro.

A placa suporta ainda protocolos como *TCP/IP*, *UDP/IP*, *ARP*, *ICMP*, *TFTP*, *Telnet*, *DHCP*.

Pode ser configurada por *Telnet* ou porta série, permitindo alterar as configurações, da comunicação série (baudrate, stop *bits*, paridade) ou da comunicação *ethernet*. Permite ainda implementar protecção por *password*.

Algumas características da placa relativamente ao processamento:

- **CPU:** Baseado no micro *DSTni-EX* optimizado para 16-bit, arquitectura X86@48MHz com *MAC/PHY* 10/100 integrado;
- **Memory:** 256 KB *SRAM* e 128 KB *flash*;
- **Alimentação:** 3.3 VDC, 200mA;
- **Firmware:** upgrade via *TFTP* e porta série;

4.3.3 Aquisição dos Sinais do *Encoder*

O *encoder* utilizado na experiência é o modelo *E6B2* da *Omron*. Este disponibiliza 3 sinais de saída: canal A, canal B e um terceiro Z, o sinal de zero, como se vê na figura 4.23.

De modo a isolar os sinais e baixar o nível para as tensões TTL, foram usados os optocopladores *6N135*, que apresentam as seguintes características:

- Elevada Velocidade - 1MBit/s ;
- Encapsulamento PDIP;
- Resposta ao degrau - $0,45\mu\text{s}$;
- Boa estabilidade com a temperatura;

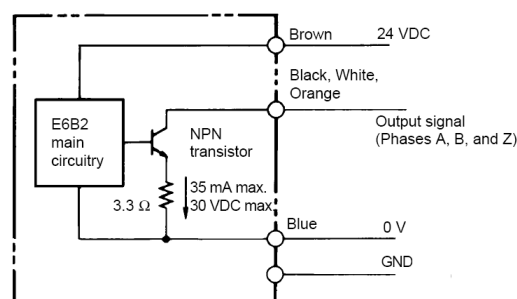


Figura 4.23 - Sinais do *encoder OMRON E6B2*

Usando a montagem típica presente na folha de características, obteve-se o circuito apresentado na figura 4.24.

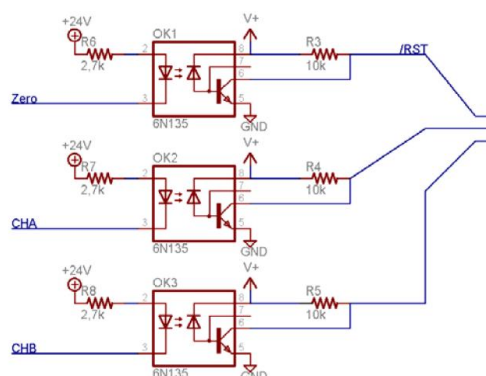


Figura 4.24 - Circuito de isolamento entre sinais do *encoder*/circuito de aquisição

Para permitir uma leitura mais fácil dos sinais do *encoder*, utilizou-se o integrado *LS7184* que recebendo os três sinais acima referidos, disponibiliza nas suas saídas o sinal de clock resultante da conjugação dos sinais A e B, bem como o respectivo sentido, figura 4.25. Estes sinais são então introduzidos no micro *ATmega8*.

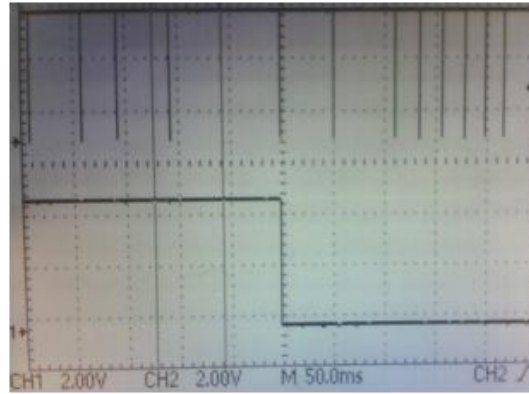


Figura 4.25 - Sinais provenientes do LS7184; em cima o sinal de *clock*; em baixo o sinal de sentido

Na figura 4.26 podemos ver o esquema do circuito electrónico associado ao micro. O integrado MAX232 (IC3) é um *driver* que gera as tensões especificadas nas normas TIA/EIA-232-F a partir de uma fonte de alimentação de 5VDC, para adaptação dos níveis de tensão CMOS vindas do microcontrolador para a porta série.

Existe também um circuito de *reset* (DS1813) que permite reiniciar o microcontrolador, no caso de falha de alimentação.

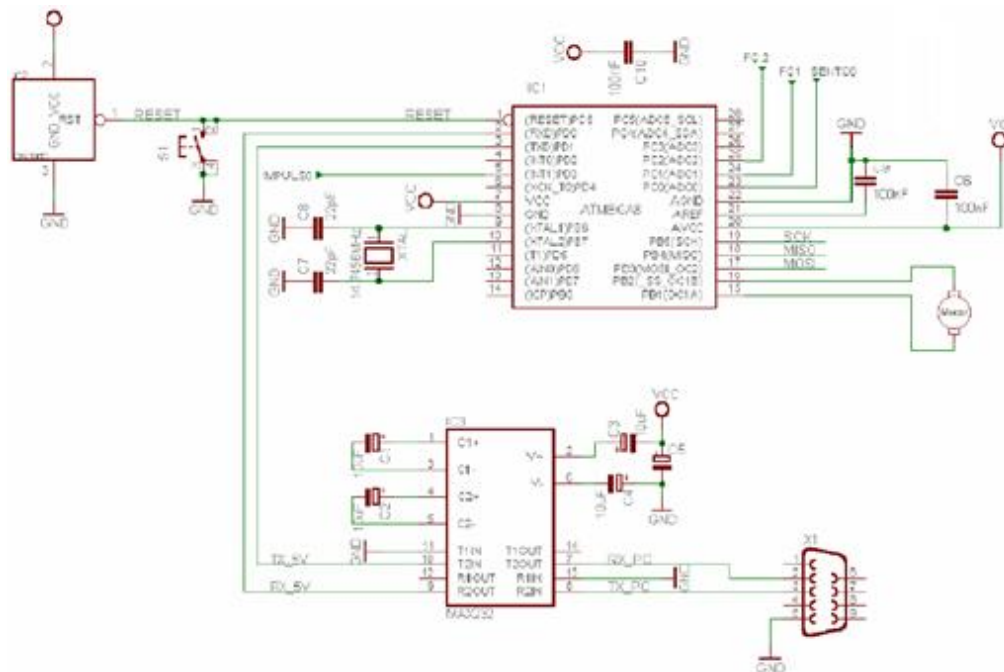


Figura 4.26 - Circuito Electrónico associado ao ATmega8

Através deste circuito o microcontrolador ATmega8 efectua o controlo deste sistema de acordo com as variáveis das tabelas 3-2 e 3-3.

Tabela 4-4 - Tabela de associação dos pinos com os respectivos elementos do sistema

Actuadores	Saídas ATmega8
Motor	PB1 e PB2
Travão Electromecânico	PB3
Detectores	Entradas ATmega8
Fim de curso Esq.	PC1
Fim de curso Dir.	PC2
Sensores	
Posição	INT1
Sentido	PC0 e INT0

Através do conjunto de entradas e saídas, apresentados na tabela 4-5, foi possível controlar todo o procedimento de aquisição dos sinais do *encoder*, como era pretendido. Na figura 4.27 apresenta-se o diagrama de entradas e saídas do sistema.

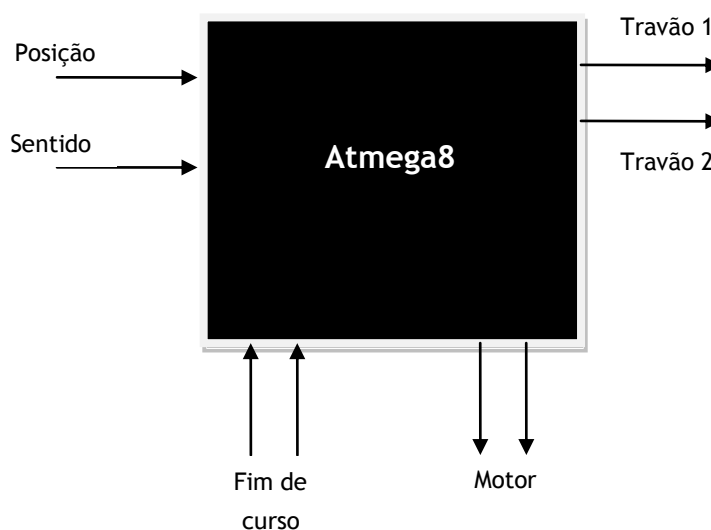


Figura 4.27 - Diagrama de entradas e saídas do microcontrolador

4.4 Software

Para a implementar a comunicação com o *set-up* experimental foi necessária a programação em diferentes linguagens: C, para o microcontrolador; *Freepascal*, para a aplicação *Lazarus* do servidor; *Java*, usada no desenvolvimento da interface do utilizador remoto, figura 4.28. Assim, o *software* está dividido em três módulos principais:

- *Software* de aquisição e controlo - executado no microcontrolador, adquire os sinais do *encoder* e dos fins de curso, controla os actuadores do *set-up* e a comunicação com a *Xport*.

- Aplicação *Lazarus* - instalado no servidor, é responsável pelo processamento dos dados, seu armazenamento e comunicação com a aplicação *Java*.
- *Software Java, HTML* - responsável pela apresentação remota dos resultados da experiência recebidos da aplicação anterior, não tendo qualquer capacidade de processamento o que a torna mais “leve”.

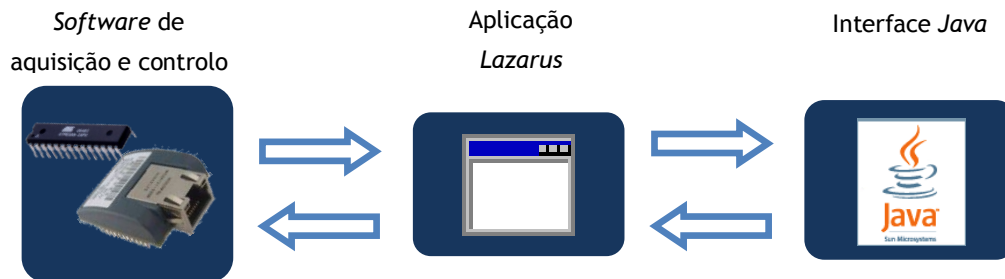


Figura 4.28 - Estrutura de *software*

Nas secções seguintes serão descritas detalhadamente cada uma das aplicações, que constituem a estrutura de *software*.

4.4.1 *Software* do Microcontrolador

O *software* do microcontrolador está dividido em 4 partes, inicialização, actuação, comunicação e aquisição. Na figura 4.29 é apresentado o fluxograma da *inicialização*, *actuação* e *comunicação*, sendo a aquisição explicada mais à frente, na figura 4.30.

O *software* que controla a experiência começa com uma rotina de *inicialização* (inicializa) que tem como função colocar o microcontrolador num modo de espera efectuando um *reset* a todas as variáveis. De seguida um ciclo contínuo chama repetidamente, a função de leitura do *buffer* da porta série e verifica o estado dos fins de curso da *drive* de *cd*, para desligar o respectivo motor. Sempre que a função *receive_byte()* retorne um caracter diferente de *null*, executa o comando associado. Quando é recebido o caracter 'i', é chamada a rotina de *inicialização* que prepara o sistema para uma nova execução da experiência. Se receber um 'p', significa que se deseja parar a experiência, activando a *flag* de paragem.

Se a *flag* de paragem está activa significa que a aquisição de dados já terminou ou foi cancelada. É então necessário monitorizar a posição actual do pêndulo para que a *drive* de *cd* seja aberta no momento certo, garantindo assim que o contacto se faz com o mínimo de impacto.

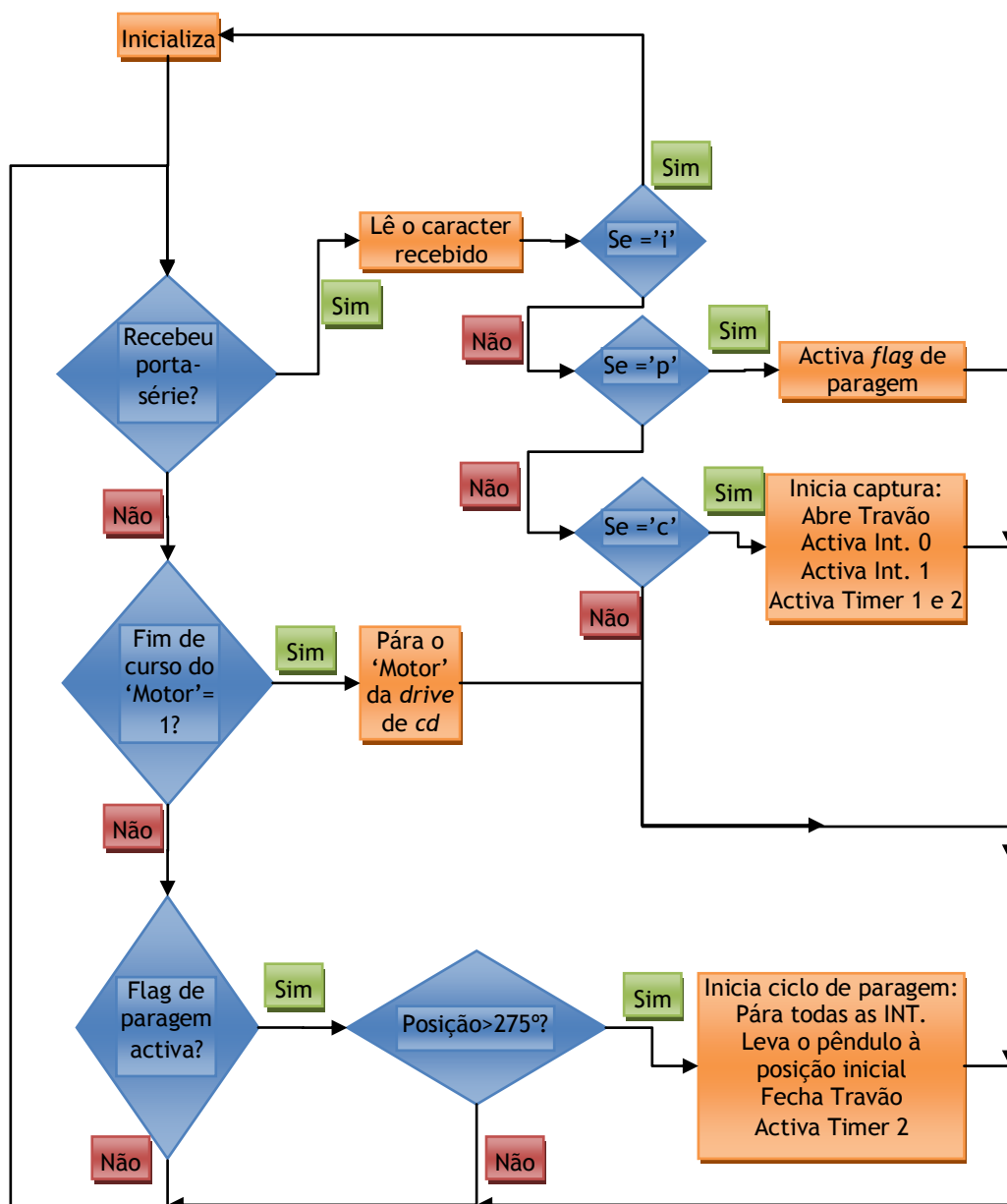


Figura 4.29 - Fluxograma de funcionamento do programa principal do microcontrolador.

Por último, o comando 'c', dá início à aquisição de dados (valor da posição angular do pêndulo e intervalo de tempo entre cada de posição) implementada pelas rotinas de interrupção, mostradas na figura 4.30.

As funções *send_byte()* e *sendstring()* são responsáveis pelo envio de informação para o servidor, a função *sendstring()* chama a função *send_byte()*; recursivamente para enviar uma *string*.

Cada vez que se dá uma mudança da posição actual do pêndulo, o sinal vindo do *encoder* activa a INT1, fazendo com que esta actualize o valor da variável 'Posição', sendo esta enviada para o servidor.

A experiência é dada por terminada ao fim de 10 ciclos de oscilação estável. Para isso, o sinal de 'Sentido', proveniente do integrado LS7184, activa a interrupção, INT 0, permitindo contar assim o número de ciclos de oscilação estável. O termo oscilação estável surge, depois

de se ter verificado que o disparo do travão causa a vibração do suporte levando a que a INT 0 seja actuada erroneamente. Para corrigir este problema, a rotina de aquisição só começa alguns segundos depois de iniciado o movimento oscilatório, quando a vibração é pequena.

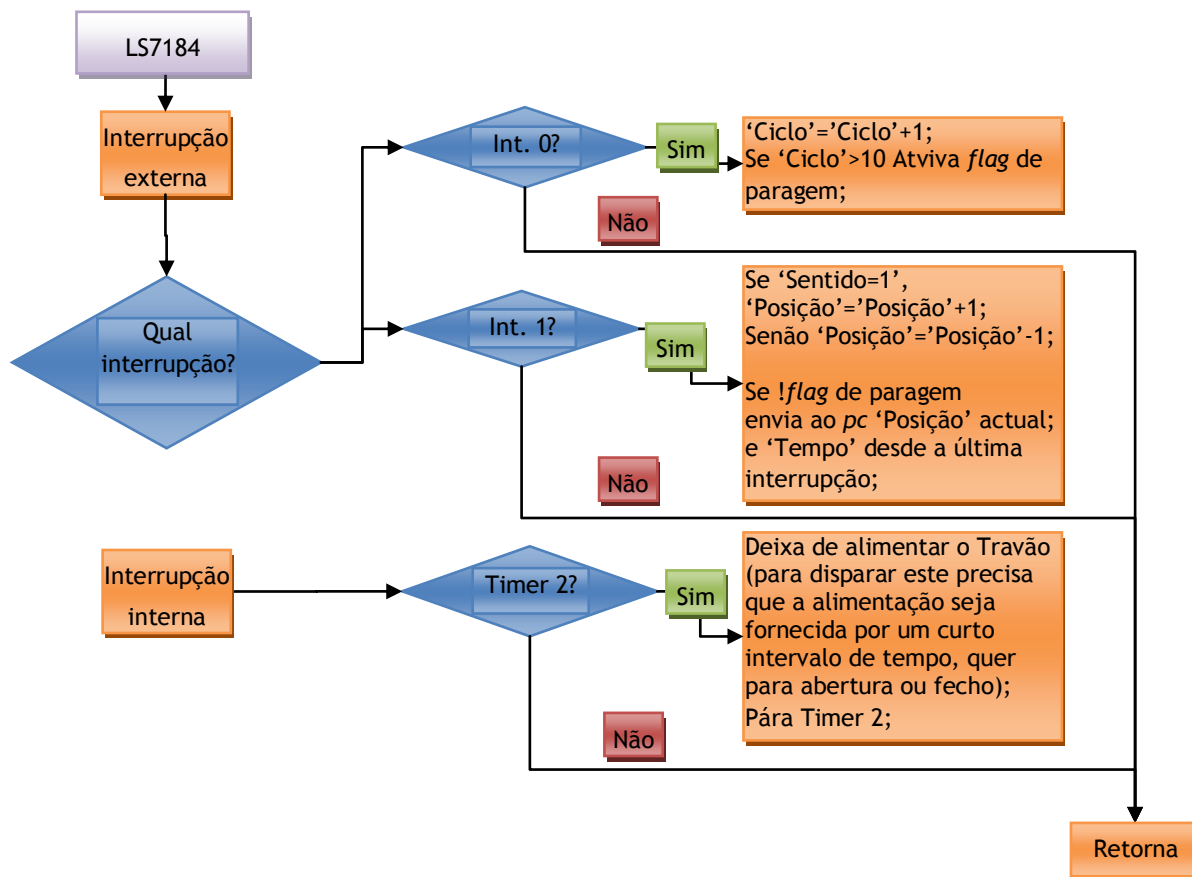


Figura 4.30 - Funcionamento das rotinas de interrupção do microcontrolador

A última interrupção implementada é a correspondente ao *Timer 2*. Ao accionar o travão, quer para prender ou largar o pêndulo, este necessita que o sinal seja fornecido durante um intervalo de tempo de cerca de 100 ms. O *Timer 2* inicia a sua contagem aquando da activação do comando do travão, uma vez chamada a rotina de interrupção do *Timer 2*, deixa de fornecer energia ao travão e desliga-se a si próprio.

4.4.2 Aplicação *Lazarus*

A aplicação *Lazarus* [29] funciona como intermediário entre o microcontrolador e a interface *Java*, permitindo ainda executar localmente a experiência (dispensando o *Java*).

O programa recebe uma trama com o valor actual da posição angular e outra variável que dá o tempo entre cada interrupção. A trama tem a estrutura apresentada na figura 4.31.

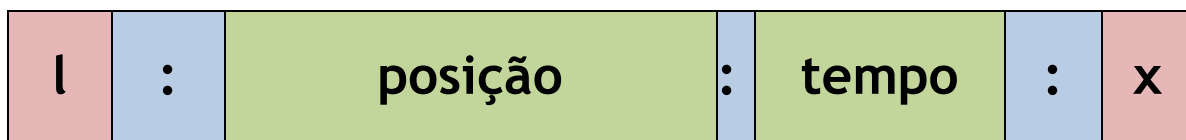


Figura 4.31 - Estrutura da trama enviada do microcontrolador para o *Lazarus*

Este tipo de estrutura permite à aplicação *Lazarus* reconstruir a trama mesmo que não a receba de uma vez, através de um ciclo que verifica, caracter a caracter, as tramas recebidas (constrói_trama). No fim da aquisição o microcontrolador envia um caracter 'f', de fim, para que a aplicação *Lazarus* saiba que a aquisição terminou.

As tramas reconstruídas são então submetidas a um sistema de *parsing*, explicado na figura 4.32.

Ao estar constantemente a receber os dados da porta *TCP*, a aplicação não conseguia processá-los a tempo de atender a uma nova chegada de dados. Isto fazia com que os valores apresentados no gráfico não acompanhassem a situação real.

Para resolver este problema criou-se uma variável para guardar todos os valores recebidos pela porta *TCP*. Essa variável vai sendo submetida a um tratamento que extrai as tramas e consequentemente os dados necessários para a representação da trajetória do pêndulo.

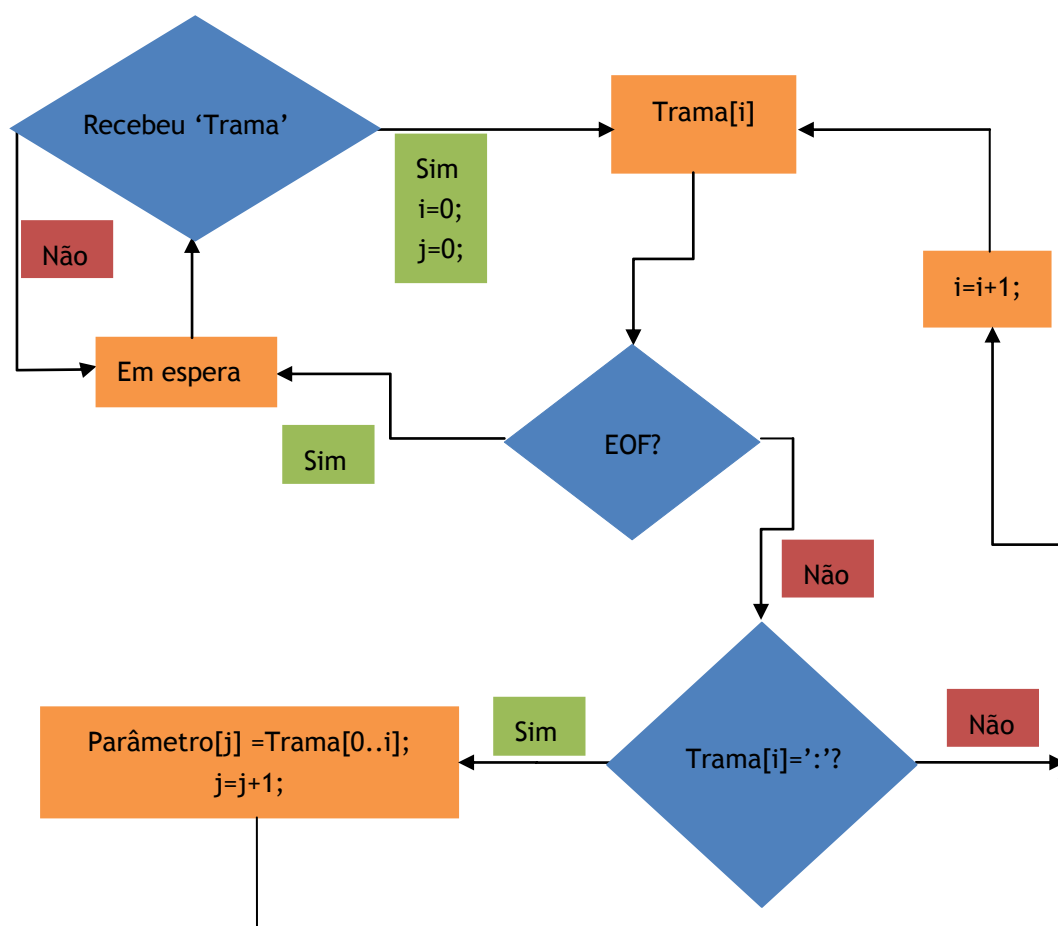


Figura 4.32 - Algoritmo de “*Parsing*” implementado em *Lazarus*.

Na figura 4.33 podemos visualizar a interface da aplicação após uma aquisição.

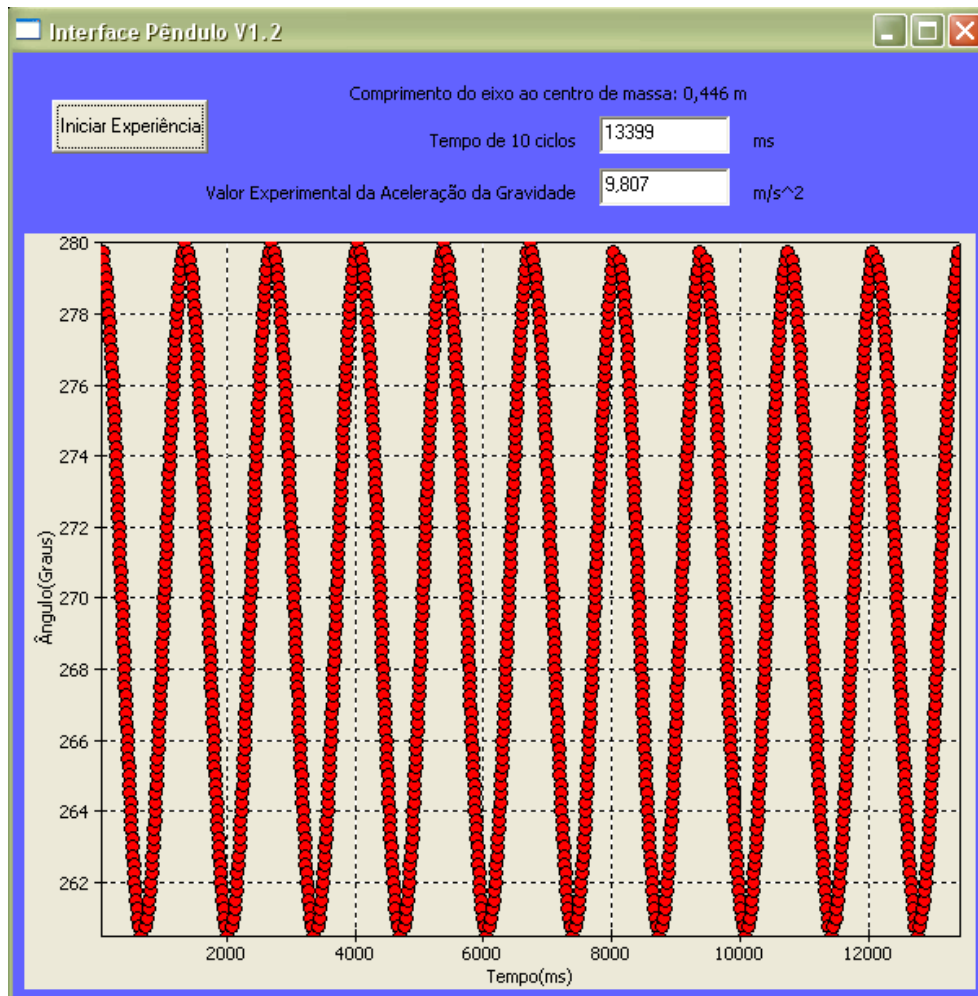


Figura 4.33 - Servidor e interface local da experiência

Passamos então a expor o funcionamento da aplicação no decorrer da experiência. A aplicação começa por fazer um *reset* ao sistema, enviando um comando ao microcontrolador para que este também inicialize as suas variáveis, e fica à escuta da porta TCP destinada a receber os comandos *Java*.

Quando recebe um comando da aplicação *Java* (ou é premido o botão de iniciar experiência) é enviado um comando de início para o microcontrolador. A aplicação fica em espera, para receber os dados. Após um período de estabilização o microcontrolador inicia o envio dos dados. Após um número considerável de recepções inicia-se então a fase de construção de tramas, após a qual a aplicação volta a receber novos dados.

Enquanto não é recebido o carácter de paragem ('f') o ciclo continua, submetendo a trama ao *parsing*, extraíndo os valores da posição e tempo. Os valores tratados são apresentados no gráfico local e enviados para a interface remota, em *Java*.

Aquando do término da experiência, pode acontecer que o número de tramas vindas do microcontrolador não atinja o número necessário para despoletar a etapa seguinte do ciclo, para impedir que isso suceda, um temporizador (*Timer 1*), é activado e reinicializado cada vez que ocorre uma recepção. Quando esse temporizador atinge o *timeout*, chama o ciclo de

construção de tramas. Um segundo temporizador (*Timer2*), com um *timeout* superior ao *Timer1*, é introduzido para resolver problemas de comunicação ou alienação da experiência em que a transmissão de dados é interrompida. Assim se o temporizador 1 lançar a interrupção e o segundo temporizador também, o sistema é reiniciado.

A aplicação implementa ainda o controlo de erros (valores nulos, falha de ligação) permitindo corrigir falhas e reinicializar o microcontrolador se necessário.

Na figura 4.34 apresenta-se o algoritmo que controla a aplicação.

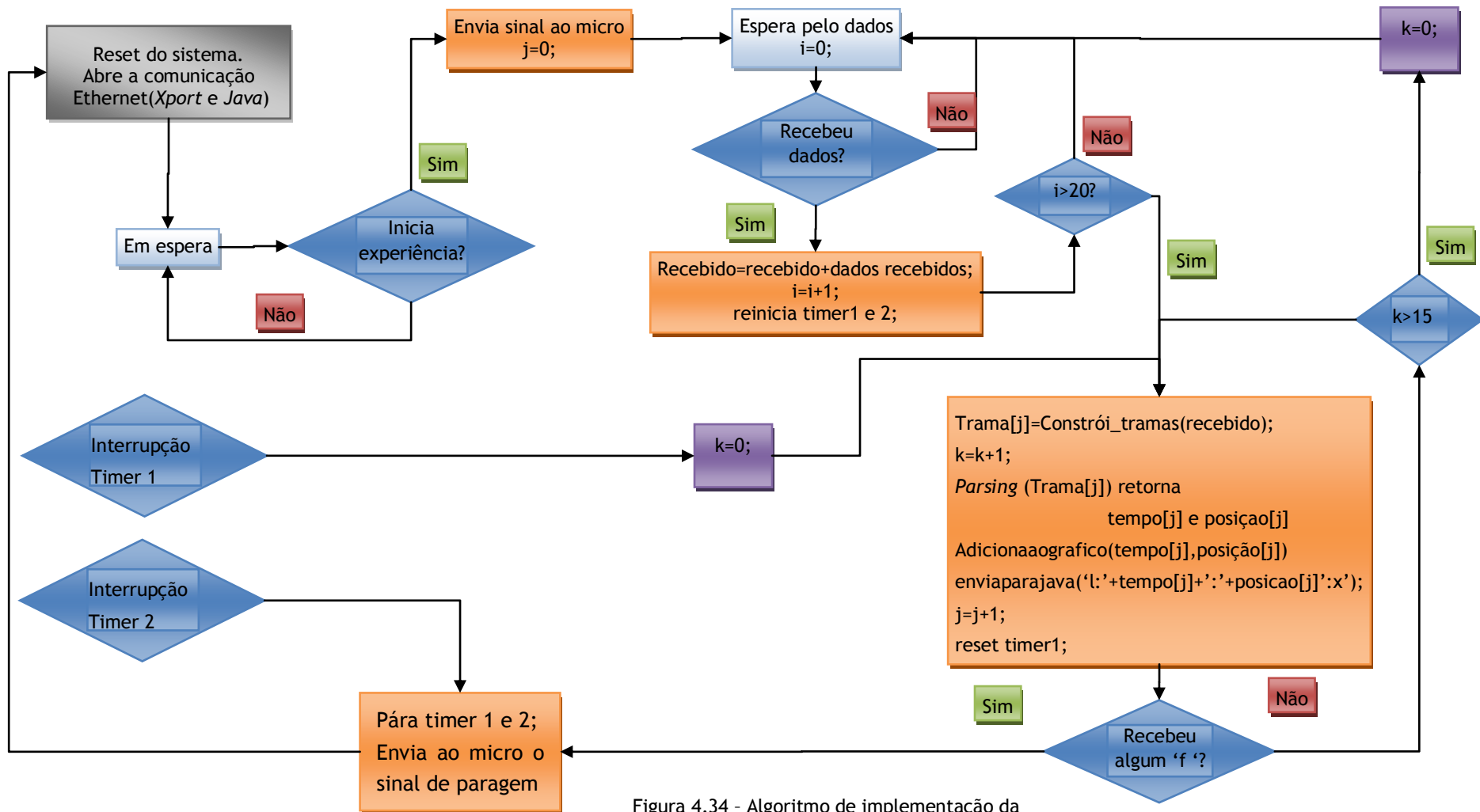


Figura 4.34 - Algoritmo de implementação da aplicação *Lazarus*

4.4.3 Java Applet

Para interagir com a experiência desenvolveu-se uma interface, figura 4.35, composta por duas áreas uma contendo a interface *Java Applet* [30] que permite ao utilizador interagir com ao experiência e uma outra que disponibiliza o vídeo em tempo real, proveniente de uma câmara *IP*.

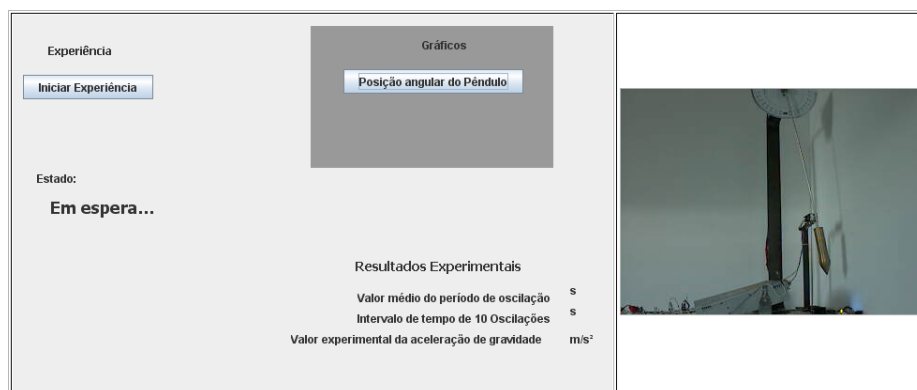


Figura 4.35 - Interface Java

A partir deste interface é possível iniciar a experiência, recebendo informação sobre os possíveis estados da mesma: em espera, em estabilização e aquisição.

Recorrendo a uma biblioteca *open source*, foi implementado o gráfico da posição angular do pêndulo ao longo do tempo, figura 4.36, com os dados adquiridos pelo sistema descrito nos capítulos anteriores.

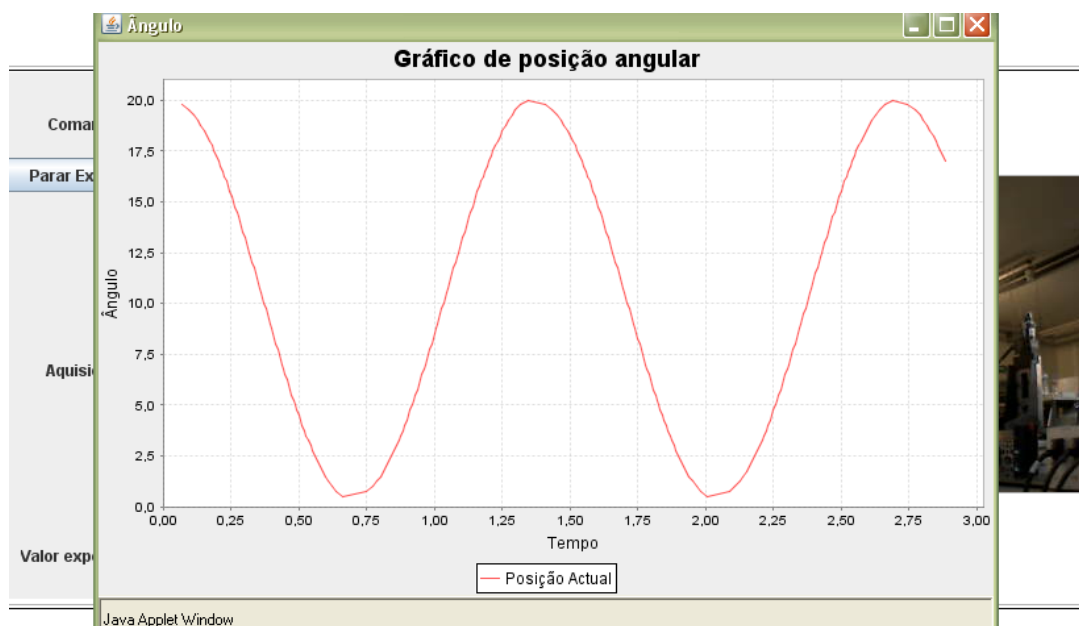


Figura 4.36 - Gráfico da interface Java

A função principal da interface *Java Applet* é apresentar ao utilizador os valores resultantes da experiência, sendo que, também é possível iniciar e parar a execução da experiência através de comandos enviados para o servidor. Os resultados que podem ser visualizados na interface são: o valor experimental da aceleração da gravidade, o intervalo de tempo de 10 oscilações e o valor médio de um período de oscilação.

Neste capítulo procurou-se explicitar a construção e constituição mecânica do pêndulo, salientando os pormenores de ajuste e suporte, relativamente aos actuadores, a *drive* de *cd* e travão electromecânico.

De seguida justificou-se a escolha do sensor de aquisição de posição angular, o *encoder* óptico, tendo em conta as suas características de baixo binário, robustez, qualidade do sinal e imediata disponibilidade.

Relatou-se as soluções de baixo custo para actuação, uma baseada em material informático obsoleto (*drive* de *cd*) e outra constituída por um actuador electromecânico com memória. Esta permite manter o seu estado sem gasto de energia.

Por último descreve-se as arquitecturas de *hardware* e *software* responsáveis pela aquisição, comunicação e apresentação dos resultados. O controlo e a aquisição, feitos no microcontrolador, foram implementados em linguagem C.

A aplicação que corre no servidor foi implementada em *Lazarus*, realizando o intercâmbio entre a interface *Java Applet* e o microcontrolador, e efectua também o tratamento dos dados que chegam do microcontrolador, enviando-os depois para a interface *Java Applet*.

Capítulo 5

Conclusões e trabalho futuro

As tecnologias de informação e comunicação (TIC) são cada vez mais parte integrante do dia-a-dia dos jovens. Numa altura em que o insucesso escolar ligado às áreas da matemática e física tem vindo a aumentar [26], [27] pretendeu-se com o desenvolvimento deste trabalho dar um pequeno contributo para incentivar e cativar os estudantes pré-universitários na área da experimentação da Física.

Utilizando um tema integrado na unidade curricular do secundário, mais propriamente na disciplina de Física do 12.º ano, desenvolveu-se um protótipo que permite a execução remota de uma experiência para medição da aceleração da gravidade, utilizando um simples pêndulo, à semelhança do que é realizado nas escolas secundárias.

A linha de desenvolvimento do projecto apontou, sempre que possível, para o uso de ferramentas *freeware*, *hardware* de baixo custo e mesmo para a reutilização de material informático obsoleto, demonstrando que com poucos meios e imaginação é possível construir dispositivos com utilidade.

Ao mesmo tempo pretende-se mostrar que é possível desenvolver um sistema de aquisição e controlo à distância sem utilizar ferramentas de *software* comerciais (por exemplo, *LabVIEW*, *Delphi*, *etc.*), nem *hardware* dispendioso.

O sistema desenvolvido apresenta uma estrutura baseada num microcontrolador, capaz de realizar a aquisição de dados da experiência e de transmiti-los para um servidor onde os dados são tratados e apresentados ao utilizador através de uma aplicação gráfica em formato *HTML*.

A construção mecânica foi desenvolvida e executada pelo autor do trabalho. Para facilitar o seu dimensionamento foi utilizado o *software* 3D *Solid Works*. A montagem executada não introduz erros significativos ao nível do cálculo que se pretende realizar. Ao longo dos testes realizados o erro na determinação do valor médio da aceleração da gravidade foi da ordem de 0,1%, em relação ao seu valor normalizado, de $9,80665 \text{ m/s}^2$ [28]. Em qualquer caso melhoramentos da montagem mecânica serão referidos como trabalhos futuros.

No que concerne à sensorização do sistema, esta resume-se à medição da posição em cada instante. Optou-se assim por uma solução mecanicamente robusta (encastrado, incorporação de veio), baixo binário em funcionamento, boa qualidade de sinal (saída digital)

e boa resolução. A posterior integração do componente LS7184 permitiu dotar o sistema de medição de posição angular de uma resolução de 1/4 de grau.

A solução de *hardware* para aquisição e controlo de dados baseou-se no microcontrolador *ATmega8*, que reúne um conjunto de características que satisfazem a realização deste trabalho. Assim apresenta uma boa velocidade de aquisição, um conjunto de entradas e saídas digitais e entradas analógicas que satisfaz largamente as necessidades da aplicação, baixo consumo, compacto e de baixo custo.

Para resolver os problemas de comunicação apresentados pelo microcontrolador utilizou-se um módulo capaz de implementar a conversão *série-Ethernet*, *Xport*, também de baixo custo e disponibilizando múltiplos protocolos de comunicação num bloco compacto.

Os sistemas para actuação do pêndulo gravítico resumem-se a um *drive* de *cd's* para posicionamento da sua massa na posição inicial da experiência e a um solenóide do tipo *latch* magnética responsável por dar início ao movimento do pêndulo.

A aplicação para tratamento de dados foi concebida em *software* livre, *Lazarus*. Esta aplicação serve de servidor local para a aplicação *Java* e como interface local da experiência.

A aplicação desenvolvida em linguagem *Java*, permite disponibilizar os dados on-line, via *Java Applet*, bem como receber os comandos do utilizador. Para a representação gráfica de posição angular recorreu-se a uma biblioteca *open source*, *JfreeChart*. A utilização da interface do cliente exige o *plugin Java Runtime Environment*. Este requisito é disponibilizado na página *Web* da experiência. Nas versões mais recentes dos *browsers* já não é necessária a sua instalação, pois já vem incluído.

A par da apresentação dos dados é possível ainda ao utilizador visualizar o decorrer da experiência através de uma transmissão vídeo, em directo, recorrendo a uma câmara *IP*.

Foi realizado um estudo do custo estimado para o desenvolvimento deste sistema de aquisição e comunicação que se apresenta na Tabela 5.1.

Tabela 5-1 - Tabela de análise de custo

<i>ATmega8</i>	3,75 €
<i>Xport</i>	31,50 €
L298	6,00 €
Max232	1,41 €
TIP31C	1,40 €
DS1813	1,80 €
Solenóide	9,50 €
Componentes vários	10,00€
Total	65,36

A solução final deverá adicionar ao valor anterior cerca de 10 € para uma placa PCB e ainda cerca de 30 € para a alimentação do microcontrolador. A alimentação dos actuadores será realizada, mais uma vez, com recurso a material obsoleto.

Tendo em conta o conhecimento adquirido durante este desenvolvimento estima-se que o tempo requerido para implementar a interface do utilizador e a comunicação de dados seja da ordem de um mês.

Tendo em conta os preços de alguns sistemas usados actualmente em laboratórios remotos, académicos e não só, esta solução revela-se muito interessante para o uso em experiências sem grande complexidade, pelo seu baixo custo e robustez do *software* e pela sua característica multi-plataforma. Tal como neste caso, em cada nova situação, este tipo de abordagem permite escolher a solução *hardware* à medida da aplicação experimental.

Apesar da solução de estratégia de controlo remoto de experiências on-line aqui apresentada cumprir uma grande parte dos requisitos estabelecidos existe ainda um conjunto de dificuldades que não foi possível ultrapassar até à data de entrega deste documento, nomeadamente:

- Disponibilização de um método de controlo de acesso não permitindo que mais do que um utilizador aceda simultaneamente à interface, em *Java*;
- Integração de sistema de envio automático de resultados experimentais para o utilizador, por e-mail;
- Integração do gráfico na interface do utilizador;
- Visualização de 3 grandezas, posição, velocidade e aceleração angular, no fim da experiência, em simultâneo ou por escolha do utilizador;
- Melhoramentos na usabilidade e na apresentação da interface, nomeadamente ao nível das normas básicas W3C.

A nível do *set-up* estrutural do pêndulo há alterações a introduzir para melhorar a rigidez e, desse modo, aumentar a estabilidade do sistema. Há ainda que realizar a montagem do *hardware* em placa PCB.

Como último apontamento refere-se que o sistema, com as funcionalidades que apresenta neste momento, foi avaliado por um conjunto de docentes (3 FEUP, 5 FCTUC, 1 ISEL) e uma amostra de 12 alunos do secundário (no âmbito de actividades da Ujr). Como observações fundamentais, foi referido que:

- Não exige um tempo elevado para descarregar a interface;
- A interface é intuitiva e fácil de usar;
- O seu aspecto gráfico é agradável;
- A experiência contribui para ajudar a introduzir um tópico do ensino da disciplina de Física do 12.º ano.

Como crítica fundamental foi referido o facto de a representação gráfica não estar ainda integrada na interface com o utilizador.

Referências

- [1] Thais A. B. Pereira, Bruno R. A. Sales, Daniel F. L. Souza, Liliane S. Machado, Ronei M. Moraes, Joaquim, Maria T. Restivo, Antonio M. Lopes, Sistemas Hápticos na Educação à Distância Através de Laboratórios Remotos, X Symposium on Virtual and Augmented Reality, SRV 2008, João Pessoa (PB), 13 a 16 de Maio 2008, Brasil
- [2] Joaquim Mendes, Maria Teresa Restivo, Carlos Moreira da Silva e Jorge Manuel Reis, “Reaproveitamento de Equipamentos Obsoletos no Desenvolvimento de um Sistema de Medição Dimensional para Apoio ao Ensino da Engenharia”, Proceedings do 3º Congresso Luso-Moçambicano de Engenharia, Maputo, Moçambique, pp. 161-168, Agosto de 2003.
- [3] Maria Teresa Restivo, Isabel Carvalho, Joaquim Mendes, Ricardo Magalhães, Gyula Gróf, Remotely visiting academic labs using ICTs, Proceedings of SEFI 34 th Annual Conference, Session 4, pp. 64, Upsala, Sweden, 28th June/1st July, 2006.
- [4] Maria Teresa Restivo, Jaime Villate, Fernando Almeida, Fátima Chouzal, Ricardo Magalhães and Isabel Menezes, The Michelson Interferometer: a learning object, CD-Rom Proceedings of the International Conference on Remote Engineering and Virtual Instrumentation - REV2007, 24-26 June 2007, FEUP, Portugal.
- [5] J.M. Ferreira, and A.M. Cardoso, “A Moodle extension to book online labs,” Proceedings of Remote Engineering and Virtual Instrumentation International Symposium, REV’05, 2005.
- [6] Gravidade - Protocolo Avançado, e-escola, http://www.e-escola.pt/site/topico_elab.asp?topico=120&exp=11
- [7] <http://www.physics.purdue.edu/class/phys152l/#>
- [8] Microchip. (2007). Microchip Technology Inc. Retrieved from PICDEM.net 2 Internet/Ethernet Development Board User's Guide: <http://ww1.microchip.com/downloads/en/DeviceDoc/51623b.pdf>
- [9] Advantech. (2008). Ethernet-based Dual-loop PID Controller. Retrieved from Advantech: http://taiwan.advantech.com.tw/unzipfunc/Unzip/1-2LKIY1/ADAM-6022_DS.pdf
- [10] Labjack. (2008, 2 12). LabJack - LabJack UE9 USB/Ethernet DAQ. Retrieved from LabJack - USB/Ethernet based data acquisition and control: http://www.labjack.com/files/LabJack_UE9_Users_Guide.pdf
- [11] Ethernut. (2005, 11). Ethernut 1 Hardware. Retrieved from Embedded Ethernet: <http://www.ethernut.de/pdf/enhwm13e.pdf>
- [12] Jorge Almeida, Pedro Portela, Maria Teresa Restivo and José Couto Marques, “Complementary Learning Through Multidisciplinary in Engineering Education”,

- Proceedings do World Congress on Engineering and Technology Education, WCETE'2004, pp. 270-273, Brazil, March 2004.
- [13] Maria Teresa Restivo, José Couto Marques and Tito Vieira, Experimenta@FEUP, CD-Rom Proceedings of the International Conference on Engineering Education, ICEE 2007, Universidade de Coimbra, Setembro 2007.
- [14] Programa de Física. (2004). Ministério de Educação.
- [15] Baker, G. L., & Blackburn, J. A. (2005). The Pendulum: A Case Study in Physics. USA: Oxford University Press, ISBN10: 0198567545.
- [16] Enciclopédia Verbo Edição Século XXI, 2006, ISBN: 972-22-1850-6.
- [17] Noémia Maciel, Maria Manuela Gradim, Jaime E. Villate, Maria José Campante, Eu e a Física, Agosto de 2005, Porto Editora, ISBN-13: 978-972-0-42208-8.
- [18] Victor A. Greenhut, J. D. The CRC HANDBOOK of MECHANICAL ENGINEERING Second Edition, CRC Press, ISBN 0-8493-0866-6.
- [19] Pallás-Areny, Ramón, Webster, John G., Sensors and Signal Conditioning, 2nd ed, New York [etc] : John Wiley & Sons, 2001, ISBN 0-471-33232-1.
- [20] Maria Teresa Restivo, Fernando Gomes de Almeida, Maria de Fátima Chouzal, Joaquim Mendes e António Mendes Lopes, Laboratórios de Instrumentação para medição/Instrumentation for Measurement Laboratories, bilingual edition, ISBN: 978-972-8025-67-0, Editora UPorto, Março 2008.
- [21] Trietley, H. L. (1986), Transducers in Mechanical and Electronic Design, Marcel Dekker Inc, ISBN 0-8247-7598-8.
- [22] Sedra, Adel S., Smith, Kenneth C., Microelectrónica, 4ª ed, Rio de Janeiro : Makron, cop. 2000, ISBN 85-346-1044-4.
- [23] ATMEL. (2004). ATMEGA8 Complete Reference Manual
- [24] Axelson, J. (2003). Embedded Ethernet and Internet Complete. 9/2003 USA: Lakeview Research, ISBN 1-931448-00-0.
- [25] Lantronix. (2008). XPort Direct - Embedded Ethernet, Embedded Device Server, Serial To Ethernet. Retrieved from Lantronix: <http://www.lantronix.com/device-networking/embedded-device-servers/xport-direct.html>
- [26] Eneas - European Network for Environmental Assessment and Service.
- [27] CONFERÊNCIA SUCESSO E INSUCESSO: ESCOLA, ECONOMIA E SOCIEDADE, Fundação Calouste Gulbenkian, 19-20 Novembro, Lisboa, 2007.
- [28] The Internet Encyclopedia of Science, acceleration due to gravity (g). Retrieved from Lantronix: <http://www.daviddarling.info/encyclopedia/A/accgrav.html>
- [29] Free Pascal : Reference guide. Reference guide for Free Pascal, version 2.2. Retrieved from: <ftp://ftp.freepascal.org/pub/fpc/docs-pdf/ref.pdf>.
- [30] Deitel, Harvey M., Deitel, Paul J. , Java: How to program, Upper Saddle River, NJ : Prentice Hall, cop. 1997, ISBN 0-13-263401-5.