

Criação de interface gráfica virtual aplicada a um sistema de controlo de processos

Relatório de Projeto para a obtenção do grau de Mestre em Engenharia
Mecânica

Autor

Rui Pedro Ferreira Fonseca

Orientador

Professor Doutor Pedro Jorge Borges Fontes Negrão Beirão

Professor Adjunto

Coimbra, dezembro de 2016

AGRADECIMENTOS

Gostaria de agradecer principalmente ao meu orientador, Doutor Pedro Beirão, pela sua ajuda, paciência e disponibilidade demonstrada no decorrer do desenvolvimento desta tese.

Também gostaria de agradecer ao professor Frederico Santos do Departamento de Engenharia Electrotécnica pela preciosa ajuda na configuração do circuito retificador.

À minha família pela motivação, compreensão, força e ajuda com que sempre me acompanharam ao longo desta caminhada.

Quero também agradecer aos meus colegas de curso e de trabalho pelo seu companheirismo e amizade.

Finalmente, a todos os que tornaram, directa ou indirectamente, possível a execução deste desafio.

RESUMO

Esta tese, desenvolvida no âmbito da unidade curricular de Projeto do Mestrado em Engenharia Mecânica do Departamento de Engenharia Mecânica (DEM) do Instituto Superior de Engenharia de Coimbra (ISEC), tem como objetivo o estudo, desenvolvimento e implementação de uma possível solução para um problema específico com que as instituições de ensino superior de engenharia se deparam atualmente: o de fornecer aos alunos experiências e exemplos práticos estando ao mesmo tempo limitados pela natureza finita de recursos materiais, nomeadamente ao nível de equipamentos laboratoriais.

Uma resposta viável para o problema descrito baseia-se na utilização de um *software–Laboratory Virtual Instrument Engineering Workbench* (LabVIEW) – que permite a simulação virtual de equipamentos reais existentes em laboratório.

A tese divide-se em duas partes. Na primeira parte descreve-se o desenvolvimento e implementação de um sistema híbrido Arduino/LabVIEW através de uma plataforma de prototipagem electrónica *open-source* – placa Arduino – recorrendo a sensores já existentes no sistema de controlo de processos. A combinação do LabVIEW com a placa Arduino permitirá a visualização dos dados adquiridos bem como a sua utilização como forma de controlar processos. Na segunda parte será efetuada a simulação virtual, recorrendo ao LabVIEW, de um exercício didáctico associado a um sistema de controlo de processos.

Palavras-Chave: exercícios didácticos, simulação virtual, LabVIEW, sensores, Arduino

ABSTRACT

This thesis was developed within the Project course unit of the Master in Mechanical Engineering, lectured in the Department of Mechanical Engineering (DEM) of the Instituto Superior de Engenharia de Coimbra (ISEC). It describes the study, development and implementation of a possible solution to a specific problem the higher engineering education institutions are facing nowadays: to provide students with experiences and practical examples while, simultaneously, being limited by the finite nature of material resources, particularly in terms of laboratory equipment.

A viable solution to the problem described above is based on the use of specific software – *Laboratory Virtual Instrument Engineering Workbench* (LabVIEW) – which allows the virtual simulation of real laboratory equipment's.

The thesis is divided into two parts. The first part describes the development and implementation of an hybrid system Arduino/LabVIEW by means of an open-source electronic prototyping platform– Arduino – using some of the sensors available in the process control system. The combination between LabVIEW and Arduino will allow the visualization of the acquired data as well as its utilization as a means of controlling processes. The second part describes the virtual simulation, using LabVIEW software, of one didactic exercise based on a process control system.

Keywords: didactic exercises, virtual simulation, LabVIEW, sensors, Arduino

ÍNDICE GERAL

AGRADECIMENTOS	V
RESUMO	VII
ABSTRACT	IX
ÍNDICE GERAL	XI
LISTA DE FIGURAS	XIII
LISTA DE TABELAS	XV
1 INTRODUÇÃO	1
1.1 INTRODUÇÃO GERAL.....	1
1.2 OBJECTIVOS.....	2
2 INSTRUMENTAÇÃO VIRTUAL	3
2.1 ESTADO DE ARTE.....	3
3 SOFTWARE LABVIEW	5
3.1 INTRODUÇÃO	5
3.2 PROGRAMAÇÃO.....	5
4 SISTEMA DE CONTROLO DE PROCESSOS	8
4.1 INTRODUÇÃO	8
4.2 CARACTERÍSTICAS TÉCNICAS	9
5 PLACA ARDUÍNO	10
5.1 INTRODUÇÃO	10
5.2 CARACTERÍSTICAS.....	11
5.2.1 <i>Memória</i>	12
5.2.2 <i>Entradas e saídas</i>	12
5.2.3 <i>Comunicação</i>	13
5.2.4 <i>Programação</i>	13
5.2.5 <i>Reset automático</i>	14
5.2.6 <i>Proteção contra sobrecarga USB</i>	14
6 ADAPTAÇÃO DO SISTEMA DE CONTROLO DE PROCESSOS	15
6.1 INTRODUÇÃO	15
6.2 SENSOR DE NÍVEL DE ÁGUA	16
6.3 BOMBA DE RECIRCULAÇÃO.....	18
6.4 SENSOR LVDT.....	23
6.5 VÁLVULA DE DRENAGEM.....	29

7	EXERCÍCIO DIDÁTICO	33
7.1	INTRODUÇÃO.....	33
7.2	PROCEDIMENTO.....	33
8	CONCLUSÃO	37
9	TRABALHO FUTURO.....	39
10	BIBLIOGRAFIA.....	40

LISTA DE FIGURAS

Figura 1- Implementação da instrumentação virtual no ensino.....	2
Figura 2- Exemplo de um painel frontal.....	6
Figura 3- Excerto de um diagrama de blocos	6
Figura 4-Exemplo de indicador e controlos no painel frontal	7
Figura 5- Exemplo de indicador e controlos no diagrama de blocos	7
Figura 6- Módulos didático (esquerda) e de controlo (direita) do sistema de controlo de processos.....	8
Figura 7- Ligações e medições efetuadas no módulo de controlo.....	9
Figura 8- Placa Arduíno Duemilanove	11
Figura 9- Identificação das fontes de alimentação	11
Figura10 - Placa Arduíno devidamente identificada	13
Figura 11- IDE para programação da placa Arduíno	14
Figura 12- Placa Arduíno e caixa de barramentos.....	15
Figura 13- Sensor de nível de água	16
Figura 14- Circuito do sensor de nível de água	17
Figura 15- Excerto do diagrama de blocos em LabVIEW	18
Figura 16- Bomba de recirculação.....	18
Figura 17- Duty-Cycle.....	19
Figura 18- Ponte H	20
Figura 19- Esquema do circuito da ponte H.....	20
Figura 20-Esquemática do circuito de accionamento da bomba de reirculação	21
Figura 21- Excerto do diagrama de blocos em LabVIEW	22
Figura 22- Circuito de comando em fase de testes	22
Figura 23- Sensor LVDT.....	23
Figura 24- Tensão nas bobinas de um LVDT	24
Figura 25- Esquema de um LVDT	24
Figura 26- Variação da tensão e linearidade função da posição do núcleo.....	25
Figura 27- Leitura do sinal com ruído	26
Figura 28- Circuito retificador.....	26
Figura 29- Compensação da corrente AC feita pelo condensador e conversão em DC.....	26
Figura 30- Placa electrónica com circuito rectificador.....	27

Figura 31- Leitura do sinal após retificação.....	27
Figura 32- Esquematização da montagem do circuito	27
Figura 33- Excerto do diagrama de blocos em LabVIEW	28
Figura 34- Eletroválvula de drenagem.....	29
Figura 35- Esquema da eletroválvula de drenagem	29
Figura 36- Esquema de um relé	30
Figura 37- Relé usado	30
Figura 38- Esquema do circuito de acionamento da eletroválvula de drenagem.....	31
Figura 39- Excerto do diagrama de blocos em LabVIEW	31
Figura 40- Ilustração do botão de comando no LabVIEW	32
Figura 41- Montagem final do sistema híbrido Arduino/LabVIEW.....	34
Figura 42- Iniciação da interface e activação da bomba de recirculação.....	35
Figura 43- Subida do nível de água e do nível de tensão do sensor LVDT.....	35
Figura 44- Sensor de nível activado e activação da válvula de drenagem.....	36

LISTA DE TABELAS

Tabela 1- Características técnicas dos sensores	9
Tabela 2- Outras características técnicas do equipamento	9

1 Introdução

1.1 Introdução Geral

A inovação tecnológica, considerada uma consequência relevante da investigação académica, pode alterar a forma como as instituições de ensino transmitem conhecimento e os alunos aprendem. Para as instituições de ensino os desafios são grandes e variados. O ensino à distância, os sistemas de gestão de aprendizagem e a oportunidade de colaborar com as outras instituições são alguns exemplos do que as instituições de ensino estão actualmente a adoptar.

O ensino e a inovação tecnológica estão relacionados entre si, isto é, com a evolução tecnológica o ensino evolui e vice-versa. Porém, as rápidas mudanças no campo da tecnologia criam problemas às instituições de ensino, sendo que o ensino da engenharia não é excepção.

Há uma necessidade contínua de actualizar e diversificar os conteúdos programáticos das diferentes disciplinas de modo a acompanhar o ritmo da evolução tecnológica. Na área do ensino da engenharia, o qual contém uma percentagem importante de trabalhos laboratoriais e/ou de cariz experimental, surgem dificuldades acrescidas.

Segundo dados publicados pela Direcção Geral de Ensino Superior (DGES, 2010), com o aumento do número de alunos a frequentar as instituições de ensino nos últimos anos, surge um problema específico: como fornecer aos alunos importantes e relevantes experiências práticas, estando ao mesmo limitado pela natureza finita de recursos, sejam materiais, humanos, ou de infra-estruturas?

O número de bancadas/mesas de trabalho ou o número de instrumentos ou equipamentos é insuficiente para o número de alunos, não permitindo a sua utilização por todos em condições ideais. A esta situação acresce o facto de que o uso inapropriado dos equipamentos por parte de alunos, menos experientes, pode provocar a sua inutilização. Para além dos problemas acima mencionados, os instrumentos laboratoriais tradicionais são bastante dispendiosos e projetados para efectuar operações pré-concebidas pelos fabricantes. Muitas vezes não podem ser atualizados ou adaptados às necessidades actuais. Como corolário desse facto, por vezes os alunos deparam-se com equipamentos laboratoriais obsoletos e que já não respondem cabalmente às evoluções tecnológicas.

Uma possível solução para a questão anterior baseia-se na utilização de um *software* que permita simular virtualmente alguns dos equipamentos que existam em laboratório.

Actualmente a instrumentação virtual é usada em várias áreas. O ensino não é excepção, sendo empregue como uma ferramenta educacional. Pode ser usada para projectar sistemas de medição e controlo, como pode ser uma ferramenta de simulação, permitindo ao utilizador

observar o comportamento de sistemas complexos em diferentes contextos, com variáveis diversas e compreender melhor determinados fenómenos.

Alguns estudos (Gillich *et al.*, 2010; Olteanu *et al.*, 2009) demonstram que os alunos aceitam bem o uso da instrumentação virtual nas aulas, permitindo-lhes compreender melhor determinados temas. Como consequência, os problemas são resolvidos mais facilmente. Além disso, os conceitos teóricos leccionados nas aulas são melhor assimilados e a auto-aprendizagem é facilitada.

A Figura 1 ilustra a opinião dos alunos sobre a utilidade da instrumentação virtual para entender os conceitos da ciência (a) e a implementação das ferramentas da instrumentação virtual no futuro (b).

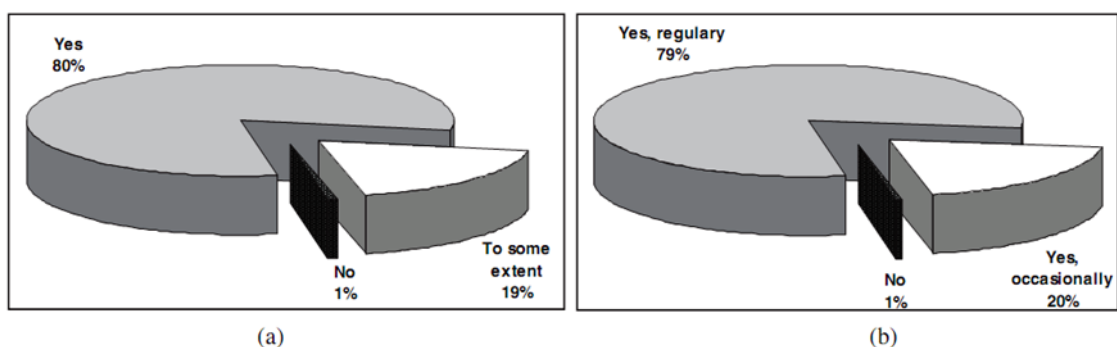


Figura 1- Implementação da instrumentação virtual no ensino

1.2 Objectivos

Pretende-se simular, de forma virtual recorrendo a um *software* específico – *Laboratory Virtual Instrument Engineering Workbench* (LabVIEW) – um conjunto de exercícios didáticos associados ao sistema de controlo de processos De Lorenzo DL 2314 existente no Laboratório de Instrumentação e Controlo do Departamento de Engenharia Mecânica (DEM) do Instituto Superior de Engenharia de Coimbra (ISEC). Adicionalmente, pretende-se ainda, recorrendo à combinação do mesmo *software* com uma plataforma de prototipagem electrónica *open-source* – placa Arduino – proceder à aquisição de dados de sensores já existentes no referido sistema de controlo de processos, por forma permitir a visualização desses mesmos dados, bem como a sua utilização como forma de controlar certos processos.

2 Instrumentação virtual

2.1 Estado de arte

Um sistema de instrumentação virtual consiste num *software* utilizado para desenvolver testes computadorizados e sistemas de medição; para controlar dispositivos de *hardware* de medição recorrendo a um computador; e para exibir de modo gráfico os resultados da medição de dados no monitor do computador. Os dados resultantes do ensaio de medição são adquiridos por um dispositivo externo ligado ao computador (Goldberg, 2000).

Existem várias definições para instrumento virtual. Santori define um instrumento virtual como “um instrumento cuja função geral e recursos são determinados em *software*” (Santori, 1990). Goldberg descreve que “um instrumento virtual é composto de algumas subunidades, alguns computadores de uso geral, alguns *softwares*, e um pouco de conhecimento” (Goldberg, 2000).

De uma maneira geral, estes autores captam a definição da ideia básica de instrumentação virtual, ou seja, um computador pode simular qualquer outro, desde que tenha instalado um *software* que simula o outro computador (Sumathi *et al.*, 2007). Esta universalidade apresenta um dos princípios básicos de um instrumento virtual: a sua capacidade de mudar de forma através de *software*, permitindo ao utilizador alterar livremente a sua função para se adequar a uma ampla gama de aplicações.

O recurso à instrumentação virtual insere-se nas novas técnicas que podem ser utilizadas para projectar diferentes simulações de processos reais. A instrumentação virtual tem vindo a conquistar um lugar de destaque em áreas que lidam com sistemas de medição, aquisição e teste, bem como de automação e controlo, permitindo uma fácil integração da sua utilização e uma adaptação às necessidades do programador. Os principais factores responsáveis pelo seu sucesso assentam no rápido desenvolvimento e disseminação dos computadores (o que teve consequências a nível económico), e ainda na elevada rapidez de conversão de dados (Sumathi *et al.*, 2007). Adicionalmente, o desenvolvimento e proliferação da simulação de sensores, processamento de dados e actuadores e a sua simulação através de computadores, permitiram a rápida expansão da instrumentação virtual e de interfaces de comunicação com o utilizador.

A instrumentação virtual veio revolucionar todo o paradigma da indústria, permitindo que equipamentos ou processos industriais reais sejam simulados virtualmente. Torna-se possível criar variáveis que simulam sensores que recolhem informação. Os valores provenientes desses sensores são tratados por um algoritmo previamente programado num *software* específico que imita de forma fiel um determinado equipamento ou processo. Esses

valores são posteriormente enviados para uma interface que comunica com o utilizador. Depois de lidos e interpretados pelo referido algoritmo torna-se possível accionar atuadores, controlando assim todo o equipamento ou o processo e mantendo-o dentro dos parâmetros estabelecidos. Este procedimento acontece sem implicações no processo ou equipamento real.

É possível transpor, de certa forma, o que foi descrito no parágrafo anterior, para o caso particular do ensino de engenharia – os alunos podem adquirir conhecimentos práticos de forma a simular a realidade, aumentando os seus conhecimentos e reduzindo as hipóteses de erro em ensaios que recorram a equipamentos laboratoriais reais que possam resultar na danificação desses equipamentos. A flexibilidade da instrumentação virtual permite ainda a auto-aprendizagem ou a aprendizagem à distância.

3 Software LabVIEW

3.1 Introdução

O LabVIEW (Laboratory Virtual Instrument Engineering Workbench) é uma linguagem de programação gráfica por blocos, criada pela National Instruments. Este *software* foi lançado em 1986 para Macintosh, mas dado ao seu sucesso e vulgarização está presente em todos os sistemas operativos.

O princípio do LabVIEW é a realização de medições e a automação. A programação é feita pelo modelo de fluxo de dados, o que oferece a esta linguagem benefícios como a facilidade de aprendizagem, a aquisição de dados e a sua manipulação.

Os programas em LabVIEW são chamados de instrumentos virtuais (VI's). São compostos pelo painel frontal, que contém a interface, e pelo diagrama de blocos, que contém o código gráfico do programa. O programa não é processado por um interpretador, mas sim compilado. A linguagem de programação tendo em conta as suas especificidades, é considerada de alto nível.

3.2 Programação

A linguagem de programação usada no *software* LabVIEW baseia-se numa programação gráfica em fluxo de dados. Os dados são transportados através de fios, reunindo-se depois em nós. Os nós são blocos onde é processada a informação, com entrada e saída de dados, segundo a programação específica do bloco. Estes nós são no fundo objectos associados em bibliotecas, tornado a programação mais intuitiva. Cada programa é designado por VI (Instrumentação Virtual) e contém duas janelas: o painel frontal (exemplo visível na Figura 2) é uma interface visual com a qual o utilizador interage em ambiente virtual e o diagrama de blocos (exemplo visível na Figura 3), onde é feita a programação propriamente dita.



Figura 2- Exemplo de um painel frontal

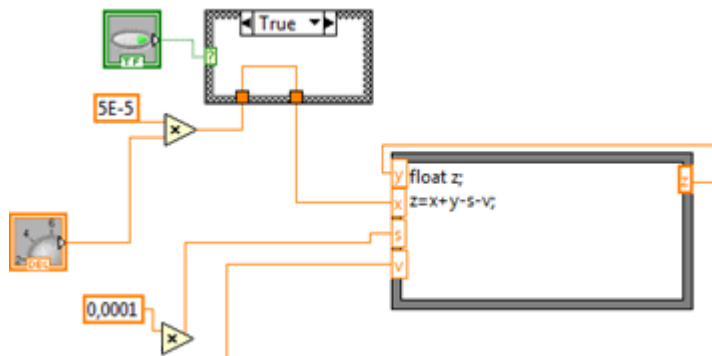


Figura 3- Excerto de um diagrama de blocos

Uma das particularidades, à semelhança de outras linguagens de programação mais tradicionais, é a possibilidade de criar um sub-programa – sub-VI – dentro do próprio programa, escolhendo um conjunto de ícones e agrupando-os em seguida. No fundo é uma sub-rotina, que depois de criada fica reduzida a um só ícone com entradas e saídas, mas que concentra mais blocos de programação de forma a complementar, com tratamento de dados secundários, o programa principal.

Para interagir e programar de forma mais intuitiva, o LabVIEW dispõe de variadas bibliotecas repletas de ícones de controlos (entradas) e indicadores (saídas), que permitem construir programas que abrangem quase todas as aplicações possíveis. Os indicadores e os controlos (exemplos visíveis na Figura 4) estão representados no painel frontal e no diagrama de blocos.

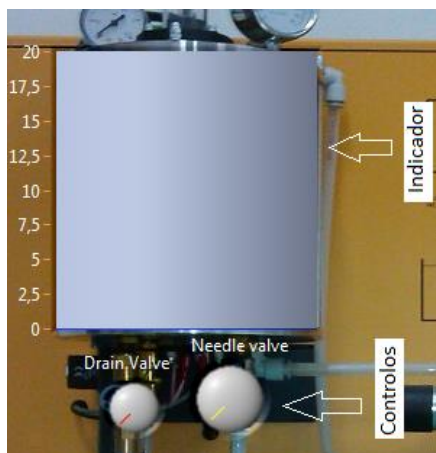


Figura 4- Exemplo de indicador e controlos no painel frontal

No painel frontal, estes dois elementos fazem a interação do processo com o utilizador, informando-o do que está a correr no programa através de indicadores. Para fazer os ajustes desejados o utilizador dispõe de controlos.

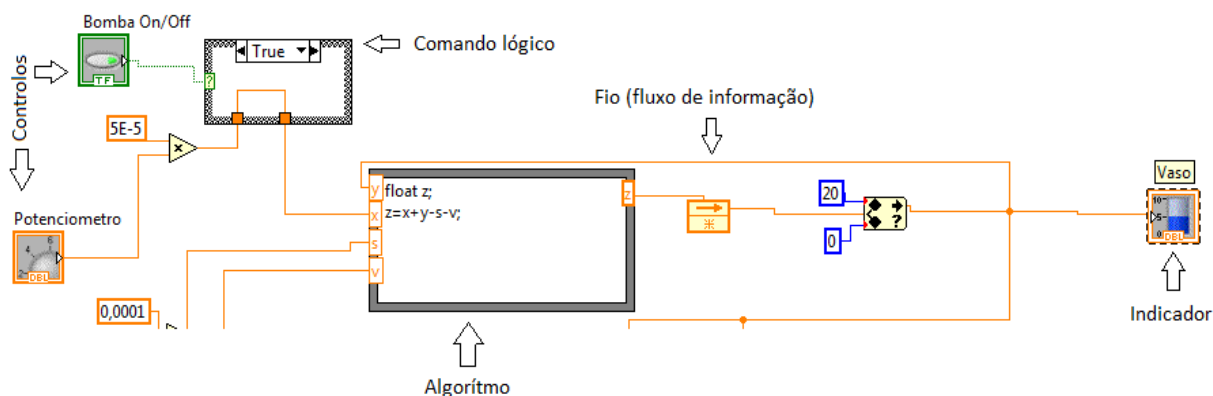


Figura 5- Exemplo de indicador e controlos no diagrama de blocos

No diagrama de blocos estes elementos (exemplo visível na Figura 5) são ligados e programados para produzir os resultados desejados. A ligação destes elementos é feita por fios que transportam a informação (fluxo de dados) entre blocos. Esses fios podem ter várias cores e espessuras, consoante o tipo de informação que transportam.

Este tipo de interface permite aos programadores menos habituados a lidar com o LabVIEW conseguirem, de forma simples, construir um programa.

Sobre este assunto é possível encontrar uma vasta informação disponibilizada em variadas fontes (como por exemplo a página *web* do fabricante do *software*, que auxilia os utilizadores na análise e solução dos problemas e na potencialização dos projetos que pretendam desenvolver.

4 Sistema de controlo de processos

4.1 Introdução

O sistema de controlo de processos DL 2314, visível na Figura 6, permite simular e controlar vários processos, permitindo aos alunos aplicar os conceitos teóricos adquiridos mediante a realização de exercícios didáticos experimentais que constam de um manual que acompanha o equipamento.

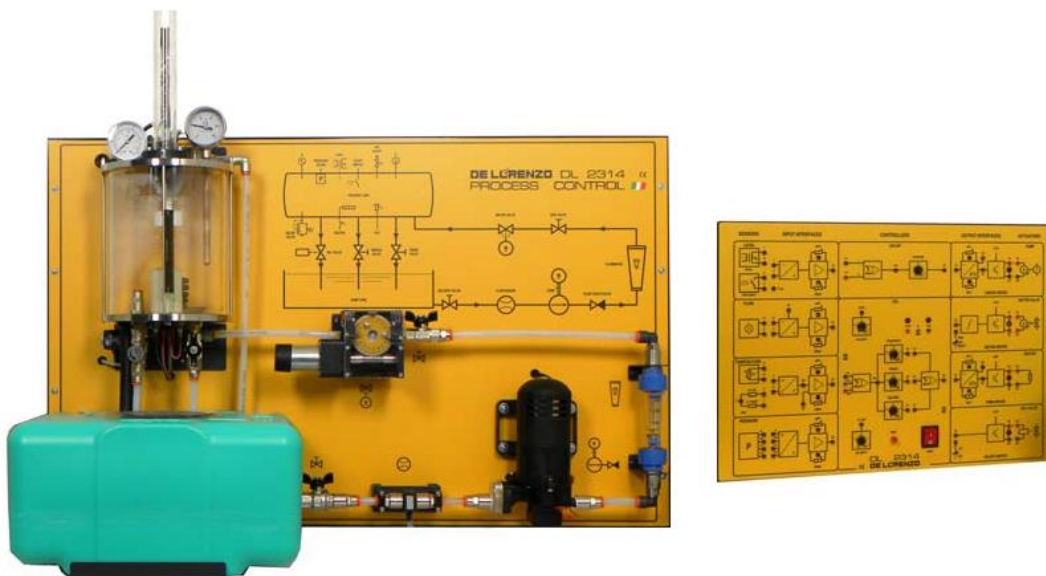


Figura 6- Módulos didático (esquerda) e de controlo (direita) do sistema de controlo de processos

Este equipamento é constituído por um módulo didático que inclui um reservatório cilíndrico, um tanque para armazenar água e vários atuadores e sensores (nível, pressão, temperatura e caudal). É ainda constituído por um módulo de controlo, responsável pelo controlo *on/off*, integral e proporcional dos vários atuadores e sensores e pela obtenção dos valores provenientes dos sensores.

Consoante o exercício que se pretenda fazer, é necessário efetuar previamente as ligações especificadas no manual que acompanha o equipamento, mediante a utilização de pequenos cabos eléctricos fornecidos pelo fabricante. Os resultados são geralmente obtidos mediante o registo manual dos valores lidos num multímetro ou eventualmente num osciloscópio, conforme a Figura 7 mostra.

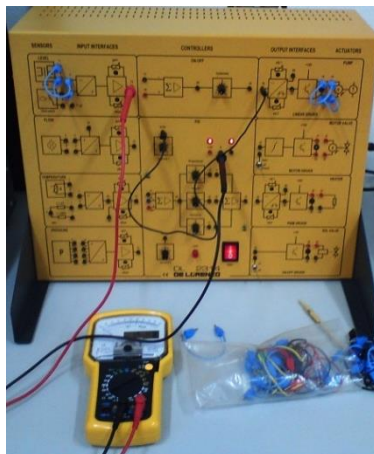


Figura 7- Ligações e medições efetuadas no módulo de controlo

4.2 Características técnicas

A Tabela 1 e a Tabela 2 resumem, respectivamente, as características técnicas dos sensores e as restantes características técnicas do equipamento.

Tabela 1- Características técnicas dos sensores

Temperatura	Nível	Caudal	Pressão
Termoresistência de platina Pt100	Transdutor Diferencial Linear Variável (LVDT)	Caudalímetro 8000 impulsos/l	Extensómetro
Termómetro bimetálico	Sensor <i>On/Off</i>	Caudalímetro (leitura direta)	Manómetro (leitura direta)

Tabela 2- Outras características técnicas do equipamento

	Características
Capacidade do reservatório cilíndrico	5 l
Capacidade do tanque	20 l
Bomba de recirculação	6 l/m; 12V; 1,5 A
Válvulas de escoamento	-
Eletroválvula de escoamento	24 V
Resistência eléctrica	48 V; 200 W
Válvula de segurança	2,4 bar
Termóstato de segurança	-

5 Placa Arduino

5.1 Introdução

Esta plataforma de prototipagem electrónica *open-source* foi desenvolvida com o intuito de democratizar a electrónica e a robótica. Sendo *open-source* permite que programadores profissionais e amadores possam ter acesso ao código fonte, alterá-lo e/ou desenvolvê-lo de acordo com as necessidades específicas de cada projecto.

Uma das características fundamentais é a facilidade de utilização, a qual permite uma grande diversidade de aplicações, tanto a nível de *software* bem como de *hardware*, tornando a automação muito mais acessível e economicamente viável. É aqui que reside o interesse neste componente (Banzi, 2011).

O elevado custo das placas de aquisição de dados da National Instruments aliado à necessidade de utilizar o LabVIEW como ferramenta didáctica, levou a própria National Instruments a incentivar um projecto conhecido por LabVIEW Interface for Arduino (LIFA). Através da programação de um código fonte no Ambiente de Desenvolvimento Integrado (IDE), que permite escrever e descarregar o código na placa Arduino, processa-se a comunicação com a programação de blocos do LabVIEW, permitindo que a placa Arduino seja programada directamente pelos blocos do LabVIEW, e ainda usar este para ler e comandar os sinais na placa Arduino. Bastando apenas ter a placa Arduino, bastante mais acessível face às soluções da National Instruments, pode-se, ainda que de forma didáctica, estudar a aprofundar o LabVIEW.

Existem vários modelos de placas Arduino aptas a responder a diferentes necessidades de processamento, alimentação e memória, entre outras características. A placa usada é uma Arduino Duemilanove, visível na Figura 8, de características simples, mas de uma enorme robustez, capaz de suprir quase todas as necessidades da electrónica actual.

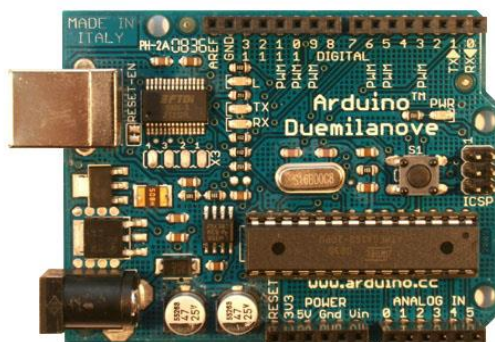


Figura 8- Placa Arduino Duemilanove

5.2 Características

A placa Arduino usada é baseada no microcontrolador Atmega328. É constituída por 14 portas (entradas/saídas) digitais, 6 entradas analógicas, um oscilador que permite uma frequência de 16 MHz, uma ligação USB, uma ficha de alimentação, um *In Circuit Serial Programming* (ICSP) que são 6 pinos que servem para programar o microprocessador Atmega e um botão de *reset*.

A alimentação externa da placa suporta uma tensão de 6 V até 20 V. Tal como a Figura 9 esquematiza, existem várias fontes de alimentação na placa Arduino:

V_{in}: entrada de alimentação para a fonte de alimentação externa.

5V: fonte de alimentação usada pelo microcontrolador e outros componentes e que pode ser gerada pelo V_{in}, pela ligação USB (*Universal Serial Bus*) ou outra fonte de 5 V.

3V3: alimentação de 3,3 V fornecida pelo microcontrolador *Future Technology Devices International* (FTDI) que é um circuito integrado comum em placas Arduino com o objectivo de compatibilizar a placa Arduino com os mais variáveis sistemas informáticos.

GND: terra ou massa.

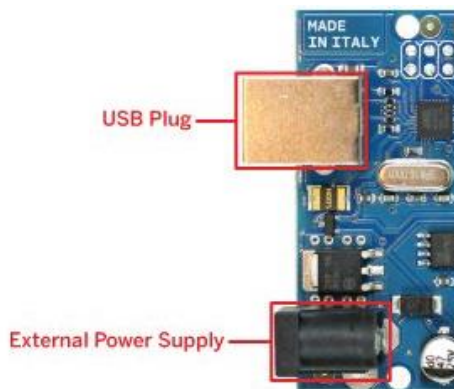


Figura 9- Identificação das fontes de alimentação

5.2.1 Memória

Em termos de memória o microcontrolador ATmega328 tem 32 Kb de memória *flash* para armazenar códigos, além dos 2 Kb de *Static Random Access Memory* (SRAM) e 1 Kb de *Erasable Programmable Read Only Memory* (EPROM).

5.2.2 Entradas e saídas

Outra característica importante que define a natureza da placa Arduino Duemilanove é a quantidade de entradas e saídas e suas características. As 14 portas digitais que a placa Arduino dispõe podem ser usados como entradas ou como saídas, dependendo apenas da forma como se programa, sendo a tensão de funcionamento de 5 V e a corrente que recebe ou envia de 40 mA. Alguns destes pinos têm funções específicas adicionais tais como:

Serial: as portas 0 (RX) e 1 (TX) são usadas para receber e transmitir dados, respectivamente.

Interruptores externos: as portas 2 e 3 podem ser programadas para activar uma interrupção ou uma mudança de valor.

PWM: as portas 3, 5, 6, 9, 10 e 11 têm a capacidade de fornecer um sinal PWM de 8bit, que permite controlar accionamentos de forma pulsada, controlando esse mesmo pulso.

LED: a porta 13 possui um indicador luminoso. Quando assume o valor HIGH, o indicador acende e quando está em LOW fica desligado.

Em relação às portas analógicas, a placa Arduino Duemilanove possui 6 entradas analógicas sendo que cada uma tem 10 bits de resolução.

I²C: as portas analógicas A4 e A5 suportam o protocolo de comunicação I²C, conhecido pela comunicação mestre-escravo, com outros dispositivos usando para tal a biblioteca Wire (nome de um conjunto de algoritmos já inseridos no sistema que facilitam a programação da comunicação destes dispositivos).

AREF: pode-se definir uma tensão de referência para as entradas analógicas de forma a iniciar ou terminar leituras.

Reset: botão que permite enviar o valor LOW para o microcontrolador de forma a reiniciá-lo.

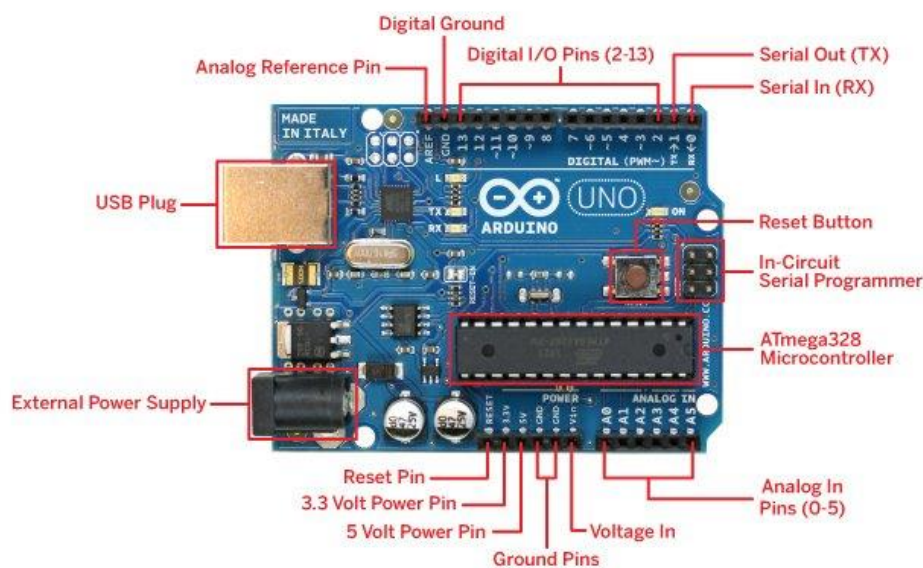


Figura 10- Placa Arduino devidamente identificada

5.2.3 Comunicação

A placa Arduino possui várias formas de comunicar com o computador, mediante USB e com outras placas Arduino, através de protocolos já descritos anteriormente, nomeadamente através do I²C e através dos pinos I/O.

5.2.4 Programação

Tal como mencionado anteriormente, a placa Arduino possui uma plataforma IDE (exemplo visível na Figura 11) que permite programar e carregar o código no microcontrolador. Foi usado um código feito por um grupo de programadores de código aberto. Este código comunica com o LabVIEW, interpretando os valores físicos de tensão recebidos e colocando-os na interface do LabVIEW. Também quando se programa no LabVIEW, o código interpreta essa linguagem e faz com a placa Arduino envie o sinal físico correspondente.

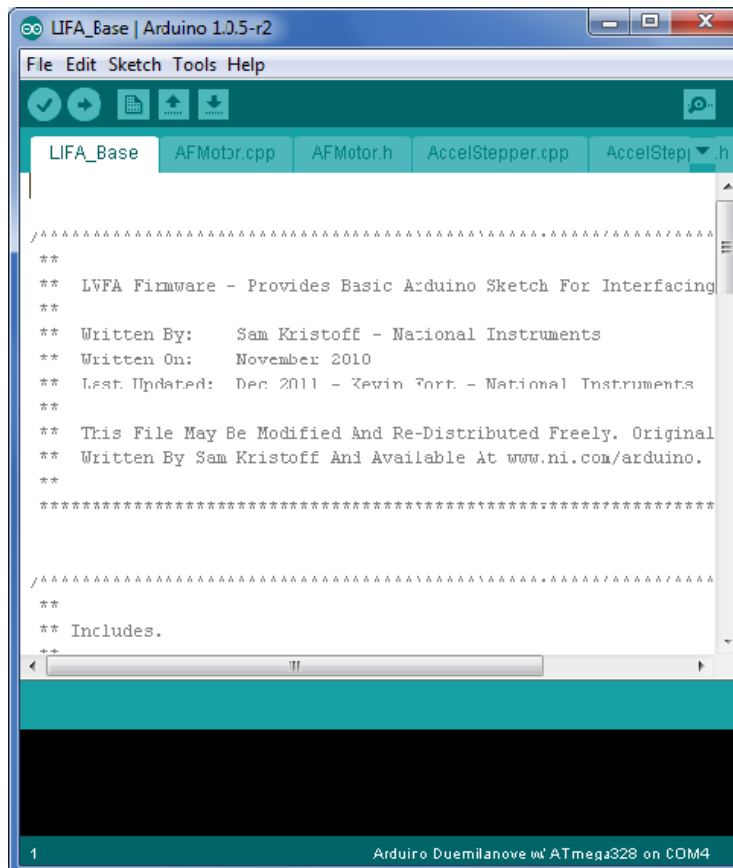


Figura 11- IDE para programação da placa Arduíno

O microcontrolador Atmega328 vem com BIOS embutida, que permite ao computador enviar o código sem recorrer a *hardware* externo para fazer essa gravação. Outra forma de programar a placa sem usar a ligação USB consiste em recorrer ao protocolo de comunicação ICSP mediante os pinos com esta característica.

5.2.5 *Reset* automático

A placa Arduíno aciona a função *reset* quando faz a ligação com o computador. Além disso a placa possui também um botão físico que permite fazer a mesma função.

5.2.6 Proteção contra sobrecarga USB

A placa Arduíno possui um fusível que protege de possíveis sobretensões provenientes da ligação USB.

6 Adaptação do sistema de controlo de processos

6.1 Introdução

Neste capítulo descrevem-se todas as adaptações efetuadas no sistema de controlo de processos de forma a incorporar novas funcionalidades, mas sem anular as especificações de origem. Para poder comandar o sistema de controlo de processos e adquirir os seus sinais, sem interferir no funcionamento original, foi necessário criar uma interface física onde os sinais provenientes do sistema de controlo de processos pudessem ser reproduzidos e, simultaneamente, fosse possível injetar sinal de comando de forma a permitir que o sistema de controlo de processos pudesse funcionar submetido às condições programadas no LabVIEW.

Os sensores e atuadores que sofreram adaptações, de forma a ser possível adquirir os seus sinais, foram o sensor *on/off* de nível água no reservatório cilíndrico, a bomba de recirculação de água, a eletroválvula de escoamento do reservatório cilíndrico e o sensor LVDT (Transdutor Diferencial Linear Variável).

A interface física é fundamentalmente uma caixa de barramentos que teve de ser criada e ligada ao sistema de controlo de processos para se poder obter e injetar sinais no sistema sem perturbar o seu normal funcionamento. Assim os sinais podem ser facilmente duplicados, podendo ser usados pelo próprio sistema e, simultaneamente, pelo LabVIEW.

A caixa de barramentos, visível na Figura 12, foi ligada à parte traseira do módulo didático do sistema de controlo de processos, onde está toda a cablagem. Esta serve para criar derivações de forma a ser possível adquirir os sinais provenientes do sistema.

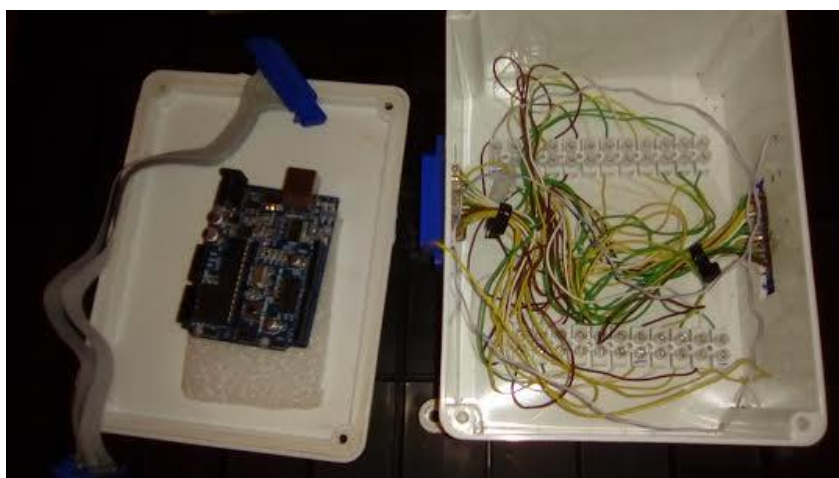


Figura 12- Placa Arduino e caixa de barramentos

Originalmente, os sinais provenientes dos sensores do sistema de controlo de processos são enviados para a placa de circuito integrado original que processa esses dados através de um cabo, permitindo realizar os exercícios didáticos do sistema. Assim, criaram-se derivações em cada um dos fios do cabo de forma a fazer ligações à placa Arduino. Desta forma tem-se acesso à leitura dos sinais. Esta adaptação não compromete a instalação original do sistema de controlo de processos.

O comando é efetuado por outras derivações exteriores à caixa de barramentos, de forma a não existir o perigo de danificar o sistema original.

A injeção de sinal é feita de modo semelhante. O sinal sai da placa Arduino, é transformado e injetado nas derivações dos atuadores.

No caso da eletroválvula de escoamento do tanque, do sensor de nível LVDT e da bomba recirculação de água foi necessário inserir *hardware* complementar de forma a retificar o sinal para que possa ser interpretado pela placa Arduino e conseqüentemente pelo LabVIEW.

6.2 Sensor de nível de água

Este sensor, *on/off*, de nível, visível na Figura 13, é acionado sempre que o nível de água do reservatório cilíndrico se aproxima da sua capacidade máxima. Trata-se de um flutuador colocado no topo do reservatório que ao subir, acompanhando a subida de nível da água, faz acionar um interruptor. Este sensor, do tipo interruptor normalmente aberto, fecha quando o nível de água atinge o máximo.



Figura 13- Sensor de nível de água

De forma a ser integrado no LabVIEW, a adaptação deste sensor exigiu algum tratamento de *hardware*. Como este sensor é um interruptor *on/off*, optou-se por adicionar um relé accionado com uma tensão de 5 V. Este accionamento só é feito quando o sensor é accionado porque vai permitir a passagem dos 5 V para activar o relé. O relé em estado *off* liga a entrada na placa Arduino responsável pela leitura do sinal à terra, de forma a garantir que na interface criada em LabVIEW não haja interferências e conseqüentemente uma possível leitura errónea. Quando o sensor de nível é activado, os 5 V são conduzidos ao relé,

este deixa de estar ligado à terra e faz com que a entrada da placa Arduino fique ligada aos 5 V. Sendo assim, a placa Arduino tem agora a entrar os 5 V provenientes do relé que foi activado pelo sensor de nível. Isso faz com que a leitura do sinal seja possível no LabVIEW com um alto grau de fiabilidade.

A Figura 14 esquematiza a montagem do circuito.

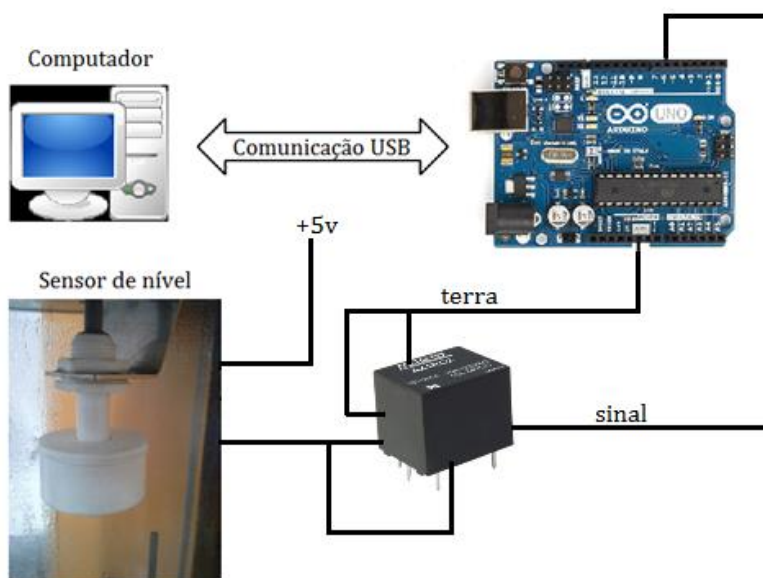


Figura 14- Circuito do sensor de nível de água

Interpretando e analisando o esquema observa-se que, se o sinal que vai para placa Arduino for de 5 V, aparece no LabVIEW a indicação que o nível de água está no máximo, através da comunicação USB entre placa Arduino e o computador com o *software* LabVIEW. Estando o relé sem energia no induzido, o sinal que vai para placa Arduino está ligado a terra e aparece *off* porque o *input* para a placa Arduino é 0 V. Quando o sensor passa a *on* devido à subida da água, passa a conduzir os 5 V que vão energizar o relé e fazê-lo comutar, deixando de estar ligado à terra e passando a conduzir os 5 V para o sinal, fazendo o *input* deste sinal na placa Arduino e posteriormente no *software*, podendo ter uma leitura eficaz do sensor.

Ao nível da programação em LabVIEW, esquematizada na Figura 15, esta exigiu a implementação do bloco de saída de comando digital e ainda a definição do sinal com sendo uma saída.

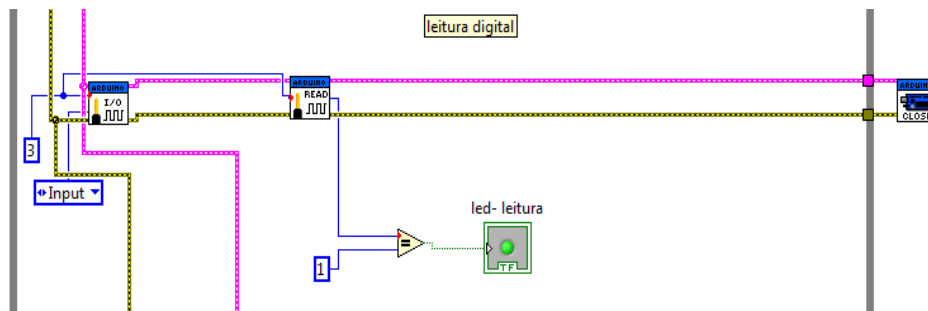


Figura 15- Excerto do diagrama de blocos em LabVIEW

Definiu-se primeiro um bloco de entradas/saídas (bloco I/O), sendo este o bloco que lê ou envia sinais digitais. Neste caso, como faz a leitura definiu-se como uma entrada. Em seguida definiu-se a porta de entrada do sinal na placa Arduino (no circuito é a porta digital 3). Depois adicionou-se um bloco de leitura (bloco READ) que serve para o LabVIEW saber que vai ler o sinal.

A informação é enviada à placa Arduino através de um bloco de finalização (bloco CLOSE). Foi ainda adicionado um bloco lógico de sinalização que ativa um sinal luminoso quando o sinal de 5 V existe.

6.3 Bomba de recirculação

A bomba de recirculação, visível na Figura 16, requereu uma adaptação mais exaustiva para poder ser controlada pelo LabVIEW. Trata-se de uma bomba que funciona em corrente contínua, com uma intensidade de corrente eléctrica de 2 A.

Para controlar atuadores com estas características é necessário recorrer ao comando por modulação da largura do impulso, *Pulse Width Modulation* (PWM), da placa Arduino. O comando por PWM permite que sistemas digitais controlem a quantidade de potência que uma determinada carga, neste caso a bomba de recirculação de água do sistema, recebe. Os impulsos são normalmente gerados por um microcontrolador, neste caso pela placa Arduino.



Figura 16- Bomba de recirculação

Para explicar melhor o funcionamento do PWM pode-se fazer uma analogia com um simples botão *on/off*. Quando o botão está ligado a potência transmitida é de 100% sendo nula quando está desligado. A placa Arduino controla o período de tempo em que o botão permanece ligado, tendo como tensão de saída 5 V enviados à placa de comando da bomba de recirculação de água e que funciona como interruptor, definindo assim o tempo em que está ativo. A placa Arduino conta ciclos de tempo constantes, isto é, por um determinado tempo ocorre uma variação liga/desliga um dado número de vezes. Tal situação designando-se por variação de pulso. Mais velocidade do sinal, significa mais impulso (mais liga/desliga) por um determinado tempo. Menos velocidade significa o oposto. Durante parte desse tempo, a totalidade ou nenhuma, a saída PWM da placa Arduino pode estar com a saída ativa, consoante a solicitação, definindo-se assim o valor do parâmetro *Duty-Cycle*. Este parâmetro define, para um determinado período, o número de pulsos enviados pela placa Arduino. A Figura 17 esquematiza o *Duty-Cycle*.

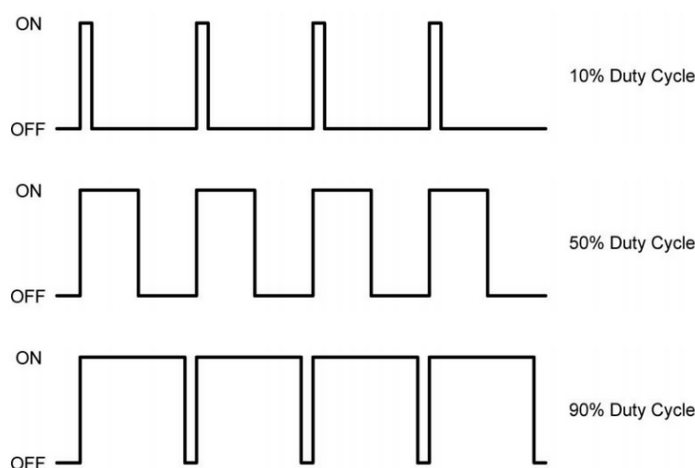


Figura 17- Duty-Cycle

O *Duty-Cycle* pode ser calculado partindo do facto que o impulso é uma onda quadrada, o que permite um funcionamento ótimo do PWM. Deste modo consegue-se prever a forma como varia a largura de impulso da onda. Para tal, é necessário considerar dois parâmetros – largura do impulso e período total da onda.

$$Duty - Cycle = 100 \times \frac{\text{largura de impulso}}{\text{período total da onda}} [\%] \quad (1)$$

Em que a largura de impulso corresponde ao tempo em que o sinal está ativado (e a bomba de recirculação de água recebe energia) e o período corresponde ao tempo de ciclo da onda.

No entanto, devido à sua fraca potência, o impulso emitido pela placa Arduino é insuficiente para colocar a bomba de recirculação de água em funcionamento. É necessária uma placa de comando adicional ativada a cada impulso recebido pelo sinal PWM. Essa placa

é designada por ponte H, visível na Figura 18, e permite converter uma corrente fixa numa corrente variável que pode comutar entre *on/off* várias vezes de forma muito rápida e segura, fazendo a variação de impulso que recebe da placa Arduino, mas de forma mais amplificada.

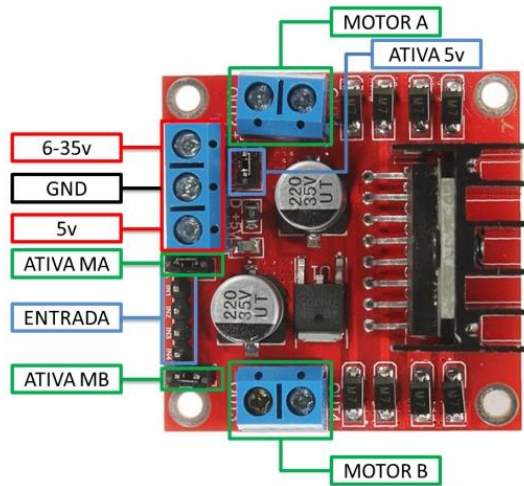


Figura 18- Ponte H

O funcionamento de uma ponte H, esquematizada na Figura 19, faz-se pelo uso de componentes eletrónicos que permitem o chaveamento (comutação *on/off*), usado para determinar o sentido e a corrente de um motor DC. A ponte H possui um tiristor com um acesso que recebe o sinal PWM proveniente da placa Arduino. Quando o sinal PWM está ativo o tiristor deixa passar a corrente de alimentação, colocando o motor em funcionamento.

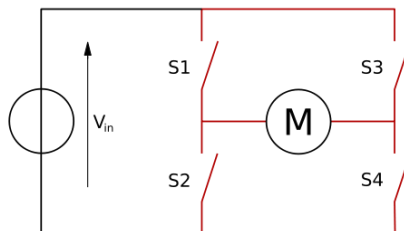


Figura 19- Esquema do circuito da ponte H

Para adaptar este circuito de comando foi necessário ir buscar a alimentação da bomba de recirculação de água ao próprio sistema. Seguidamente instalou-se a ponte H com esta mesma alimentação e ligou-se a esta a placa Arduino que permite o impulso PWM. De referir que a ponte H tem uma alimentação individual de 12 V.

Em termos de ligação de todo o conjunto, analisa-se a Figura 20 para ter uma noção mais abrangente.

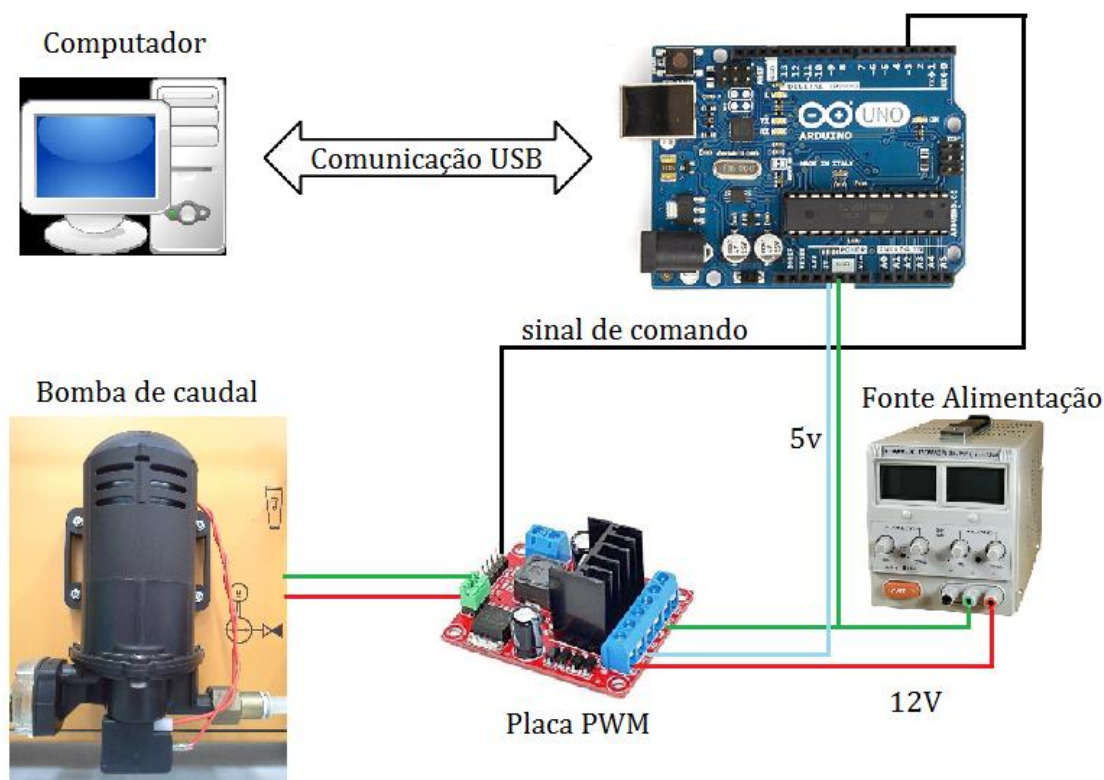


Figura 20- Esquematização do circuito de accionamento da bomba de recirculação

O LabVIEW executado pelo computador estabelece comunicação com a placa Arduino através da comunicação USB. Através do LabVIEW seleciona-se qual o pulso desejado que irá determinar a velocidade de funcionamento da bomba. Essa informação recebida pela placa Arduino é convertida em pulsos (uma saída com sinais pulsados de 5 V) e enviada para a placa PWM, que ao receber esses pulsos acciona a alimentação (fonte de alimentação de 12 V) e envia também de forma pulsada os 12 V à bomba de recirculação. A quantidade de pulsos recebidos pela placa PWM irá definir quanta quantidade se fornece à bomba de recirculação, controlando assim a sua velocidade.

Ao nível da programação em LabVIEW, esquematizada na Figura 21, também foi relativamente simples programar os blocos de forma a fornecer o sinal PWM para o accionamento.

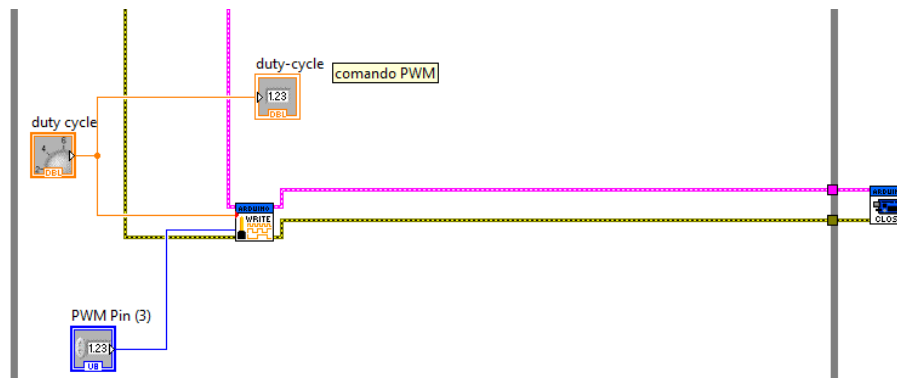


Figura 21- Excerto do diagrama de blocos em LabVIEW

De uma forma mais detalhada observa-se que a programação tem três blocos importantes: o bloco PWM, um bloco de leitura (bloco WRITE) e o CLOSE.

O bloco PWM é responsável pelo reconhecimento da porta de comando do sinal do PWM na placa Arduino. É por aqui que sai o sinal que aciona a placa de comando (ponte H), que vai depois controlar o funcionamento da bomba.

O bloco WRITE é o bloco seleccionado no LabVIEW que serve para informar o programa que se está a comandar a placa Arduino, ou seja, que se está a enviar sinal. Para este bloco trabalhar corretamente é necessário introduzir parâmetros, sob a forma de outros blocos, para que possam ser processados e definir assim o funcionamento da bomba. Esses blocos são os *Duty-Cycle* que definem os impulsos a injetar no circuito de comando e que é comandado a partir da interface, e o PWM já descrito atrás.

Todas essas informações são conduzidas para o bloco CLOSE, que finaliza a informação obtida e a envia para a placa Arduino para operar o circuito de comando.

A montagem do circuito de comando pode ser vista na Figura 22.

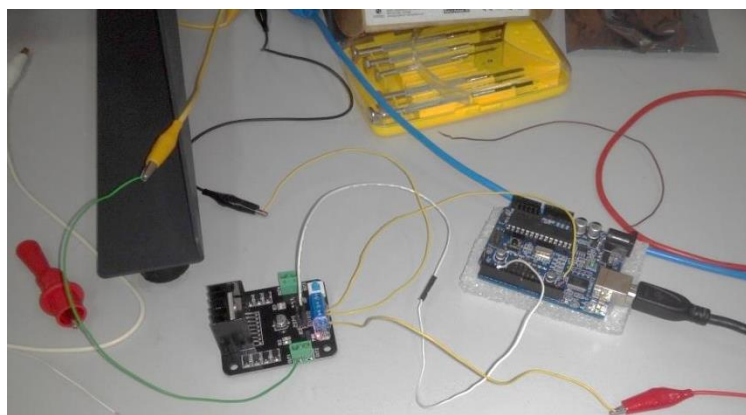


Figura 22- Circuito de comando em fase de testes

6.4 Sensor LVDT

O sensor LVDT que equipa o sistema de controlo de processos é usado para medir o nível de água presente no reservatório cilíndrico. Conforme a Figura 23 mostra, está situado no exterior do reservatório e ligado a uma haste cilíndrica de pequeno diâmetro que tem na sua extremidade uma bóia. O conjunto haste-bóia está colocado no interior do tanque.



Figura 23- Sensor LVDT

Para perceber como este sensor foi adaptado ao sistema híbrido Arduino/LabVIEW que foi desenvolvido, descreve-se sumariamente o princípio de funcionamento do sensor LVDT montado no sistema de controlo de processos.

O sensor LVDT pode ser considerado um transdutor de deslocamento que pertence à categoria dos sensores passivos, sendo caracterizado pela necessidade alimentação externa para garantir para o seu funcionamento. Neste caso específico, a alimentação é efetuada por corrente AC, a qual vai excitar a bobina primária do sensor. Essa excitação vai induzir uma tensão de saída nas bobinas secundárias.

A Figura 24 esquematiza esta situação.

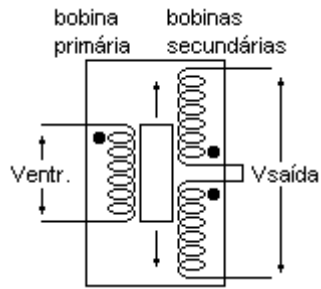


Figura 24- Tensão nas bobinas de um LVDT

Essa indução que a bobina primária faz nas duas bobinas secundárias sofre uma variação proporcional e linear à medida que o núcleo móvel, de material ferromagnético, se movimenta (mecanicamente solidário com a bóia, mediante uma haste), à medida que o nível da água no reservatório varia. Se a tensão alternada na bobina primária é a alimentação do LVDT, a tensão das duas bobinas secundárias (que sofrem uma variação de tensão provocada pelo deslocamento do núcleo móvel que vai influenciar o fluxo magnético nas bobinas secundárias) corresponde à saída, que será processada para determinar qual o nível atual de líquido que o reservatório tem.

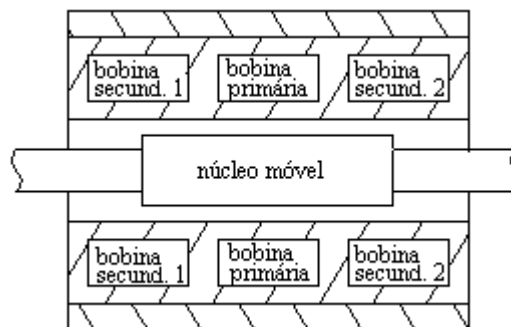


Figura 25- Esquema de um LVDT

A forma como varia a tensão à saída das duas bobinas secundárias, provocada pelo deslocamento do núcleo móvel, à medida que o nível da água no reservatório cilíndrico varia, vai ser em seguida analisada. Pretende-se explicar como foi feito o processamento deste sinal, de forma a poder ser lido pelo sistema híbrido Arduino/LabVIEW.

Quando o núcleo móvel está ao centro e equidistante das duas bobinas secundárias, a tensão de saída é zero porque as bobinas em posições contrárias anulam-se mutuamente, uma vez que as tensões induzidas pelo núcleo ferromagnético são iguais para as duas bobinas secundárias ligadas entre si. Quando o núcleo sai da posição central e se desloca fica mais próximo de uma bobina secundária e mais afastado da outra. Tal significa que a indução magnética exercida pelo núcleo é maior na bobina secundária mais próxima, resultando numa maior tensão, o que provoca uma diferença de potencial e, conseqüentemente resulta numa tensão de saída. Se o núcleo se deslocar no sentido contrário e induzir a outra bobina

secundária, a tensão resultante será a oposta relativamente à anterior. A Figura 26 ilustra o que se acabou de se descrever.

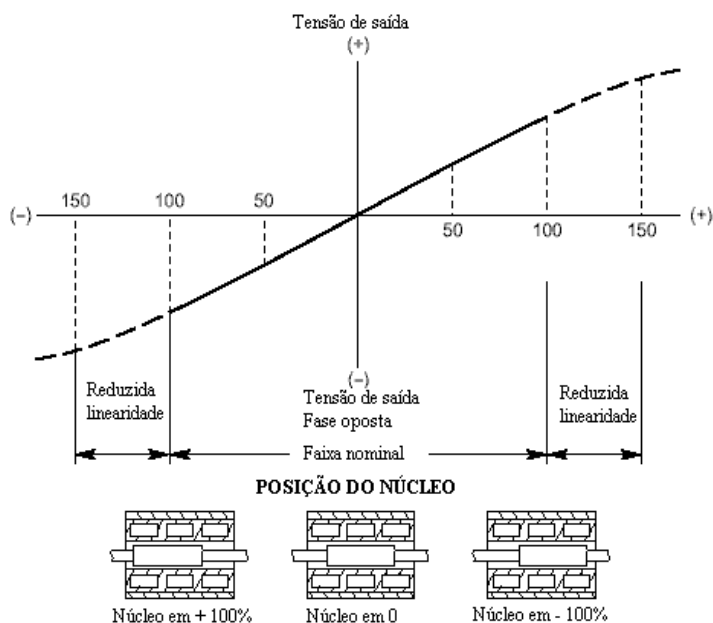


Figura 26- Variação da tensão e linearidade função da posição do núcleo

A adaptação do LVDT ao sistema desenvolvido não foi simples e nem directa, uma vez que o sinal emitido por este sensor é AC e a placa Arduino não está preparada para lidar com este tipo de sinal. Para funcionar, o *hardware* de conversão tem de transformar o sinal alternado num sinal analógico DC para poder então ser lido pela placa Arduino e, conseqüentemente permitir uma leitura no programa elaborado em LabVIEW.

A solução consistiu na criação e posterior construção de uma placa electrónica capaz de retificar o sinal AC da saída do LVDT e o converteu num sinal analógico DC. Houve, contudo, alguma dificuldade inicial em filtrar o sinal e eliminar o ruído, de modo a extrair apenas a parte do sinal do sensor que permitisse ler o valor de tensão que varia de forma directamente proporcional à variação do nível da água no reservatório cilíndrico.

O professor Frederico Santos, do Departamento de Engenharia Electrotécnica, foi uma preciosa ajuda nesta fase. Dado o constante ruído obtido na leitura do sinal de tensão desejado, foi necessário identificá-lo. Para isso recorreu-se a um osciloscópio, tal como a Figura 27 mostra. Desta forma conseguiu-se desenhar um circuito retificador que permitiu eliminar o ruído. Assim o sinal de tensão desejado já podia ser lido pela placa Arduino.

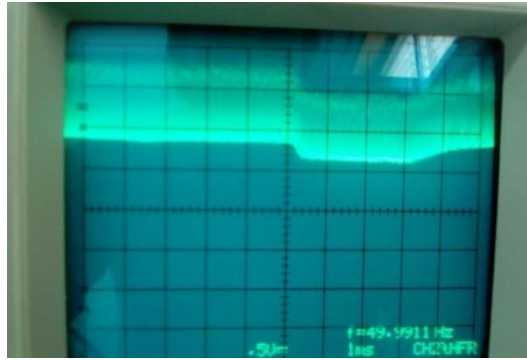


Figura 27- Leitura do sinal com ruído

A Figura 28 esquematiza o circuito retificador. Este é constituído por dois condensadores, duas resistências e um díodo.

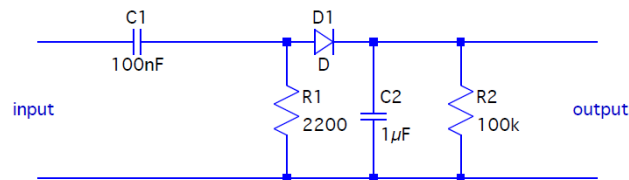


Figura 28- Circuito retificador

As resistências têm a função de fazer dissipar a tensão eléctrica de forma a baixar a tensão do circuito à saída. O díodo estabiliza o sinal e impede que haja correntes contrárias que possam danificar o mesmo. Dado que se trata de corrente AC, os condensadores irão compensar o sinal sinusoidal e manter o sinal constante, criando assim uma corrente DC. Quando a tensão sinusoidal sobe, o condensador carrega. Quando desce o condensador descarrega a energia acumulada e mantém a tensão de forma a tornar o sinal constante.

A Figura 29 ilustra esta última parte.

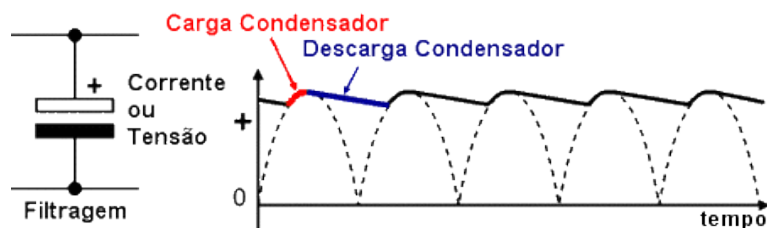


Figura 29- Compensação da corrente AC feita pelo condensador e conversão em DC

A construção física da placa electrónica correspondente ao circuito retificador pode ser vista na Figura 30.

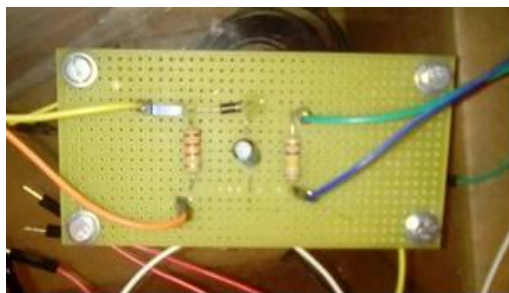


Figura 30- Placa electrónica com circuito rectificador

O sinal, agora retificado antes de entrar na placa Arduino, é visível na Figura 31. O sinal varia, sensivelmente, entre 1 V e 3 V.



Figura 31- Leitura do sinal após retificação

Para uma melhor compreensão do circuito montado analisa-se a Figura 32, que esquematiza a montagem.

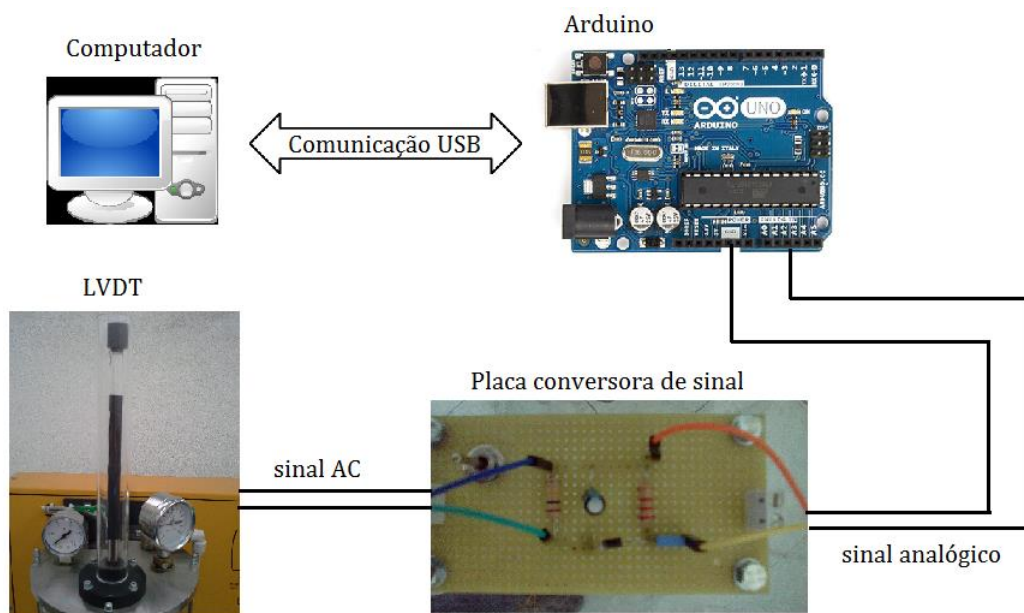


Figura 32- Esquematização da montagem do circuito

Analisando em detalhe verifica-se que o computador, através do LabVIEW, comunica com a placa Arduino através da porta USB. A placa Arduino está programada para captar o sinal proveniente da placa conversora (e enviá-lo posteriormente para o computador) que debita uma tensão proporcional ao nível de água no depósito. A placa conversora resulta da necessidade de converter o sinal AC proveniente do LVDT sensor que detecta o nível de água pelo seu deslocamento para o sinal analógico da placa Arduino.

Em termos de programação de LabVIEW, recorreu-se aos blocos que permitem a leitura de sinal analógico.

O diagrama de blocos está esquematizado na Figura 33.

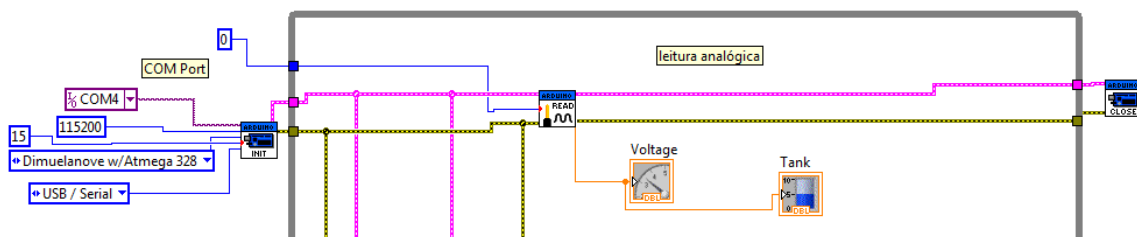


Figura 33- Excerto do diagrama de blocos em LabVIEW

Analisando a programação, para comunicar com a placa Arduino foram necessários três blocos fundamentais: o INT, o READ e o CLOSE.

O bloco INT é responsável pela recolha de informação por parte da placa Arduino, nomeadamente o tipo de comunicação (feito por USB), o tipo de placa Arduino (neste caso a UNO), os *bytes* de resolução (15), o *baud rate*, ou seja, o fluxo de informação (115200), e a porta de comunicação usada pelo computador (CM4).

O bloco READ é o bloco de leitura, responsável pela captação do sinal analógico processado na placa, e permitindo a sua leitura no programa feito em LabVIEW.

O bloco CLOSE é o bloco que finaliza a programação estabelecida, ficando o sistema apto a receber o sinal analógico na porta especificada da placa Arduino.

De referir ainda que para ligar os blocos entre si de forma a estabelecer um fluxo de informação, existem duas linhas, uma rosa e outra amarela. A rosa é responsável pelos dados enquanto a amarela lida com o erro inerente a este tipo de sensores.

Os restantes blocos presentes (*Voltage* e *Tank*) pertencem ao LabVIEW original e servem para exibir os mostradores no painel frontal, de forma a ser possível ao utilizador acompanhar os valores do LVDT em tempo real.

6.5 Válvula de drenagem

Uma vez bombeada a água do tanque para o reservatório cilíndrico através da bomba de recirculação, torna-se necessário drenar a água novamente para o tanque, a fim de repetir os ensaios. O sistema de controlo de processos usa uma electroválvula, visível na Figura 34, para este efeito.



Figura 34- Eletroválvula de drenagem

Trata-se de um componente bastante usado na indústria. Quando a bobina não está electrizada, a válvula está fechada e não há passagem de água. Isto é conseguido porque existe um êmbolo que é empurrado por uma mola contra a sede da válvula impedindo a saída de água. Quando a bobina da electroválvula é electrizada, existe passagem de corrente contínua, neste caso com uma tensão de 12 V, na bobina, a qual vai gerar um campo magnético que irá atrair o êmbolo na sua direcção. Como a bobina está por cima do êmbolo, este irá subir e comprimir a mola, abrindo assim a válvula e permitindo a passagem de água. Assim que a bobina deixa de ser eletrizada, a mola distende e empurra o êmbolo para baixo, voltando a válvula à sua posição original, bloqueando novamente a passagem da água.

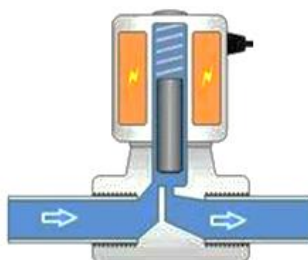


Figura 35- Esquema da eletroválvula de drenagem

Para a placa Arduíno controlar a electroválvula de drenagem, houve a necessidade de converter a sua tensão de funcionamento, pois a saída de comando da placa Arduíno usa 5V e a eletroválvula funciona com 12 V. Recorreu-se então a um relé, ligado à electroválvula, de modo a harmonizar a sua tensão com a da placa Arduíno.

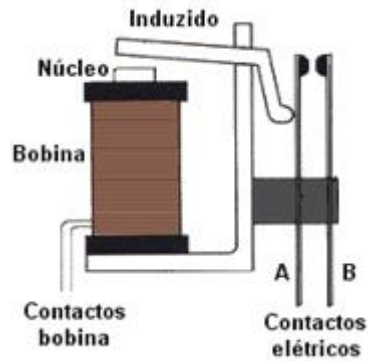


Figura 36- Esquema de um relé

Trata-se de um componente electrónico com um princípio de funcionamento bastante simples, mas muito útil. O circuito cuja tensão se pretende converter é ligado à bobina (induzido) do relé. Quando a bobina recebe tensão gera um campo magnético que vai atrair uma alavanca (induzido) que, por sua vez, irá juntar os contactos eléctricos. Uma vez juntos, o circuito fica fechado com a tensão pretendida. É como se interruptor fosse ligado estando agora todo o circuito pronto a conduzir electricidade.



Figura 37- Relé usado

Este componente permite que a tensão de 5 V que sai da placa Arduino, alimente a bobina do relé, funcionando como interruptor de um circuito submetido a uma tensão de 12 V. Este circuito irá então acionar a eletroválvula sempre que solicitado pela placa Arduino.

Na Figura 38 esquematiza-se pormenorizadamente toda o circuito de acionamento da electroválvula de drenagem.

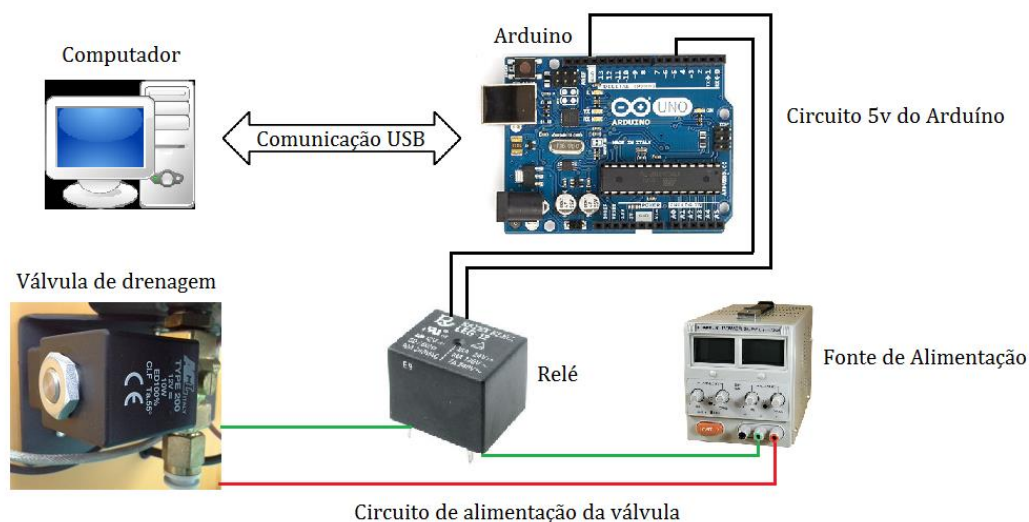


Figura 38- Esquema do circuito de acionamento da eletroválvula de drenagem

Desta forma está garantido o funcionamento da eletroválvula de drenagem (12 V) proveniente da fonte de alimentação, accionado pelo relé que se activa com uma tensão de 5V proveniente da placa Arduino. Para que essa tensão de saída da Arduino seja possível, ela recebe a informação proveniente do LabVIEW, onde está a interface, e que está a ser processada pelo computador.

Em termos de programação, para enviar o sinal para a placa Arduino de modo que esta controle a electroválvula de drenagem, teve de se recorrer a alguns blocos e interligá-los de forma a criar uma sequência de informação, conforme se pode observar na Figura 39.

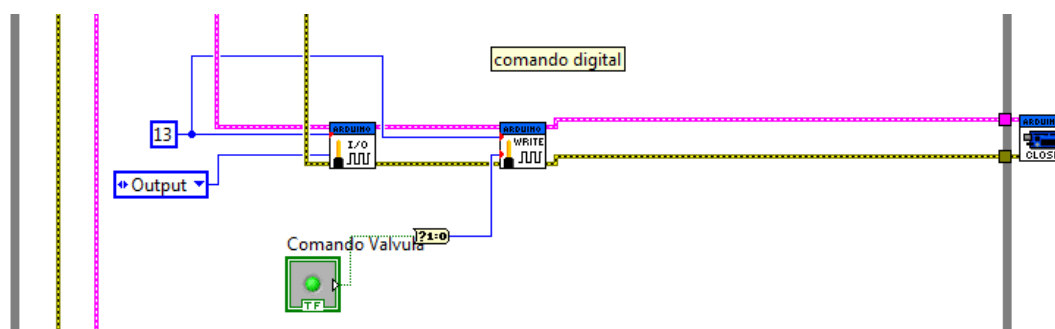


Figura 39- Excerto do diagrama de blocos em LabVIEW

A programação é constituída por três blocos basilares: I/O, WRITE e CLOSE. O bloco I/O tem a finalidade de recolher dados para posteriormente proceder ao comando. As linhas rosa e castanhas que ligam a este bloco fornecem informações sobre os recursos (nomeadamente, tipo de placa Arduino, características técnicas) e o erro respectivamente. Neste mesmo bloco há ainda recolha de mais informações, como a porta digital de saída pela qual a placa Arduino vai efectuar o comando à tensão de 5 V e a definição do bloco, se deve ser entrada (para leituras de sinal) ou saída (que é o caso).

O bloco WRITE tem como função preparar e comandar a placa Arduíno para a transmissão do sinal digital de saída e definir assim o sentido desse mesmo sinal (do LabVIEW para a placa Arduíno). Para que este bloco funcione necessita também de recolher informações como o erro e as informações genéricas anteriormente descritas e que são representadas pelas linhas castanha e rosa, respectivamente. Neste caso também é necessária a identificação da porta da placa Arduíno por onde se processa a saída de sinal, assim como o comando usado no LabVIEW para dar início a este processo, representado por um botão na interface, como mostra a Figura 40.



Figura 40- Ilustração do botão de comando no LabVIEW

O último bloco é o CLOSE que tem o propósito de enviar todas as informações reunidas e descarregá-las na placa Arduíno, executando o código que foi previamente programado, neste caso de comando do sinal digital de 5 V.

7 Exercício didático

7.1 Introdução

Depois de os sensores e actuadores estarem perfeitamente coligados está-se em condição de realizar exercícios didáticos à semelhança dos que se fazem com o sistema de controlo de processos. Este exercício é experimental e visa utilizar toda a gama de sensores para testar a sua funcionalidade e mostrar a viabilidade do sistema híbrido Arduíno/LabVIEW desenvolvido.

O exercício vai consistir em encher o depósito de água de forma gradual. Com o aumentar do nível vai-se observar e analisar a variação do sensor LVDT até activar o sensor de nível máximo. Após se desligar a bomba e o nível de água estar no máximo, drenar-se-á a água do reservatório para o taque. Para tal recorre-se à electroválvula de drenagem accionada pelo sistema híbrido Arduíno/LabVIEW. Depois também se pode analisar o comportamento do sensor de nível LVDT com a descida do nível de água, fazendo o devido estudo e comparação de valores.

7.2 Procedimento

Na montagem técnica deste exercício convém fazer uma ressalva de extrema importância.

A placa de sinal modulado PWM que comanda a bomba de recirculação, embora testada com sucesso em laboratório e mostrada a viabilidade, danificou-se acusando a sobrecarga. Tal ficou a dever-se à fragilidade desta placa em particular e ao excesso de solicitação que lhe foi imposta. Para contornar esta situação e devido ao escasso tempo disponível para obter uma nova placa electrónica PWM mais viável, teve de se instalar um sistema *on/off* com relé em que se comanda o ligar e o desligar da bomba de recirculação. Quando ligada esta funciona com velocidade constante anulando-se, com esta opção mais simples, a funcionalidade de comandar a velocidade e a quantidade de caudal que a bomba de recirculação debita. Ainda assim não inviabiliza em nada a validade das acções experimentais abordadas de seguida.

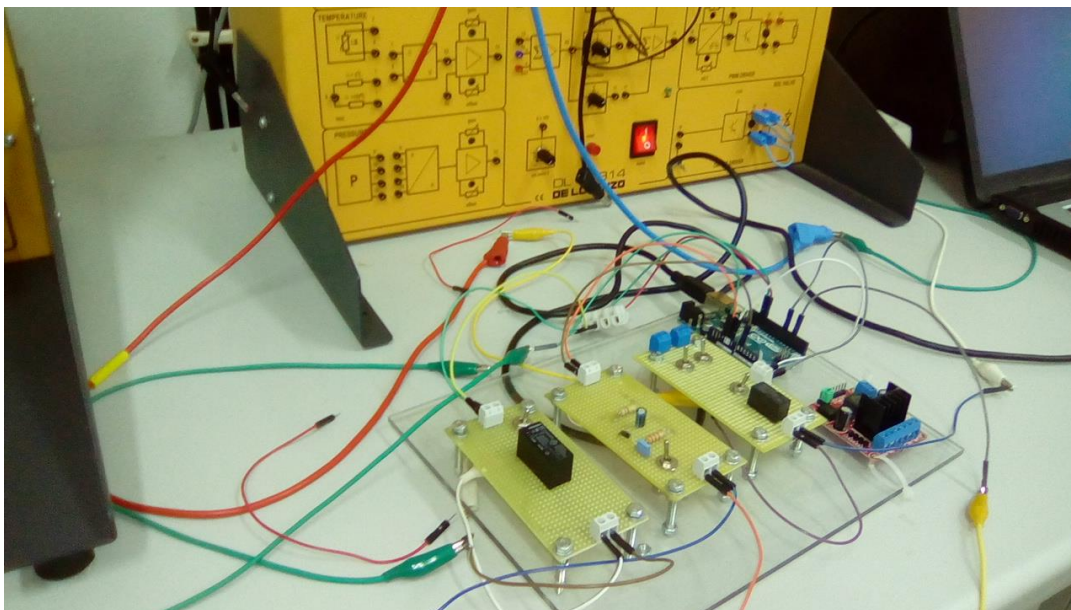


Figura 41- Montagem final do sistema híbrido Arduino/LabVIEW

Com o sistema híbrido Arduino/LabVIEW completamente instalado, como se pode observar na Figura 41, foi então ligada a bomba de recirculação mediante o acionamento do relé. Rapidamente se observa o nível de água sobe no reservatório e o sinal eléctrico a variar (aumentando proporcionalmente). Esta leitura é fornecida pelo sensor LVDT. É importante salientar que quando o reservatório atinge uma capacidade muito próxima do seu máximo, a tensão medida no sensor LVDT sofre um decréscimo. Isto acontece porque o sensor sofre o efeito de histerese electromagnética. Este fenómeno característico ocorre devida à elevada saturação do material ferromagnético provocada pela elevada solicitação provocando uma pequenina desmagnetização. Quando a água atinge o nível máximo, o sensor de nível máximo acende no painel virtual e informa sobre o ocorrido. Desliga-se a bomba e acciona-se posteriormente a válvula de drenagem comandada electricamente. Cria-se assim um sistema de comando computadorizado.

Para ter uma noção mais abrangente da forma como decorreu este ensaio, pode-se observar em imagens sequenciais a forma de como este se processa. Conforme mostra a Figura 42, inicia-se o *software* LabVIEW, já depois de o Arduino estar a correr. Liga-se a bomba de recirculação no botão de comando por relé, no qual se põe a bomba a correr através do sistema do relé uma vez que o sistema por modulação PWM não se encontra activo. A bomba deverá ligar e o nível de água deve começar a subir.

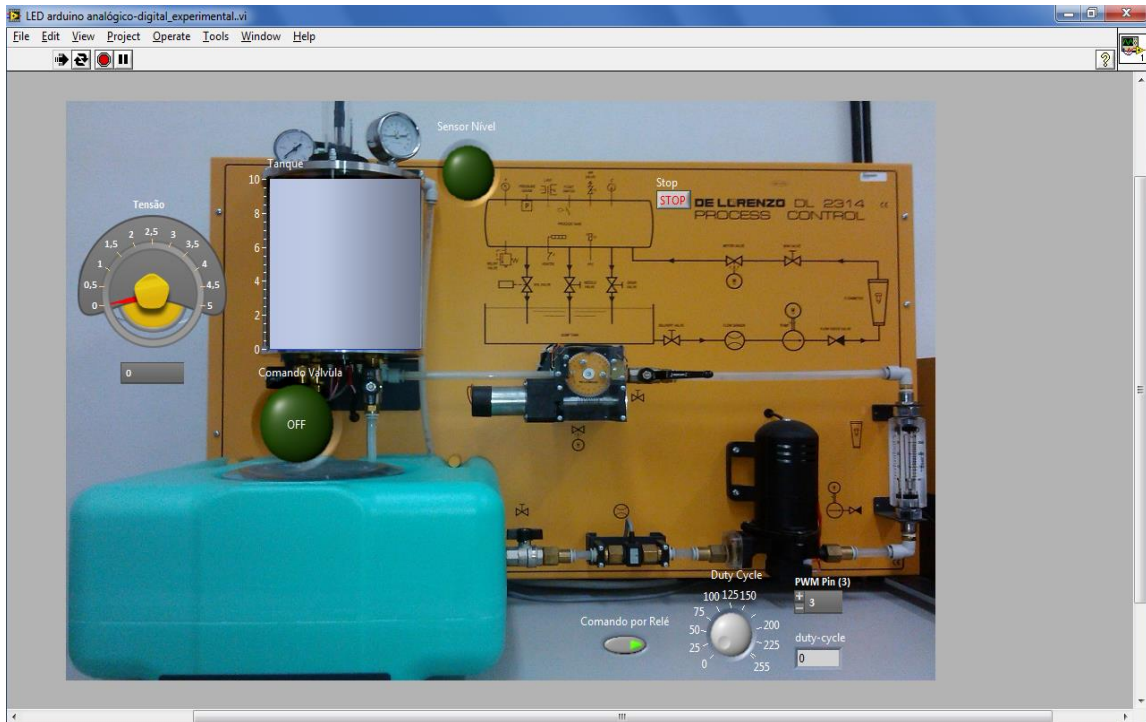


Figura 42- Iniciação da interface e activação da bomba de recirculação

Em seguida pode-se observar na Figura 43 ao aumento de tensão eléctrica de forma gradual e proporcional (no indicador tensão) à medida que o nível de água no tanque aumenta. Esta leitura de tensão provém do sensor de nível LVDT.

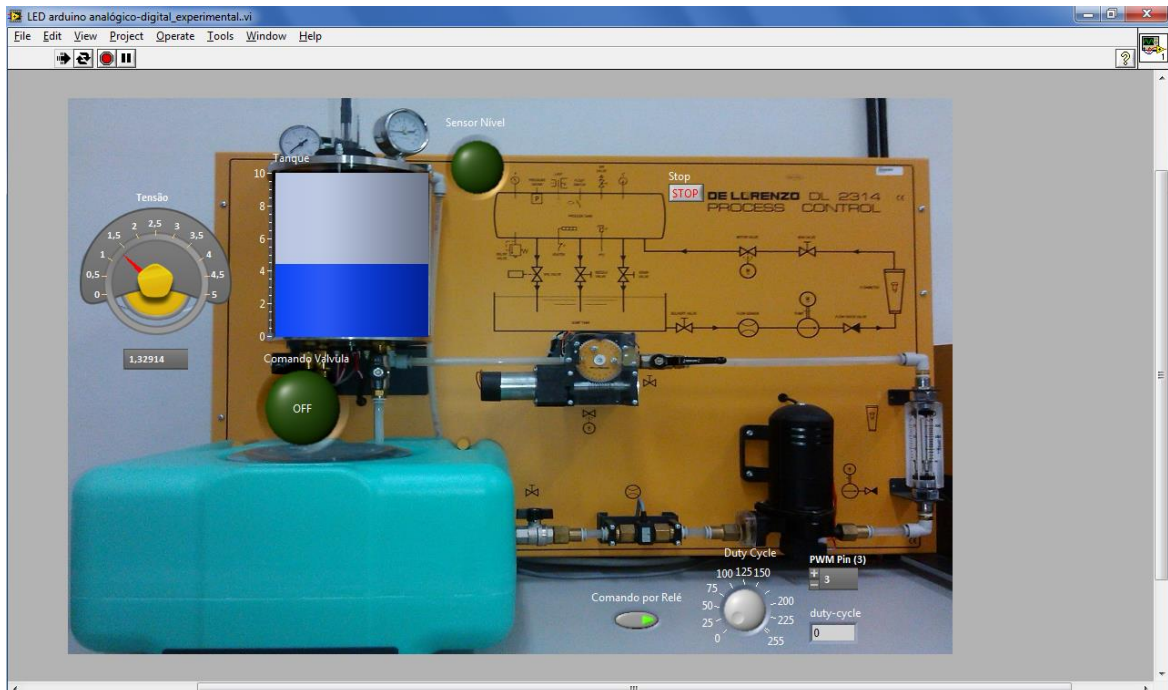


Figura 43- Subida do nível de água e do nível de tensão do sensor LVDT

De salientar a particularidade seguinte: quando a água atinge um nível perto do máximo, acontece uma pequena queda do valor da tensão enquanto o nível de água continua a subir. Esta situação ocorre devido ao fenómeno ferromagnético denominado histerese já explicado anteriormente.

Quando a água atinge o topo é activado o sensor de nível fim de curso que envia um sinal. Uma vez alcançado este patamar torna-se necessário drenar a água do reservatório para o tanque. Para isso activa-se a válvula de drenagem até esvaziar o reservatório, como se pode ver na Figura 44. Durante este processo pode-se ver o valor da tensão do sensor LVDT a descer proporcionalmente em consonância com o nível da água.

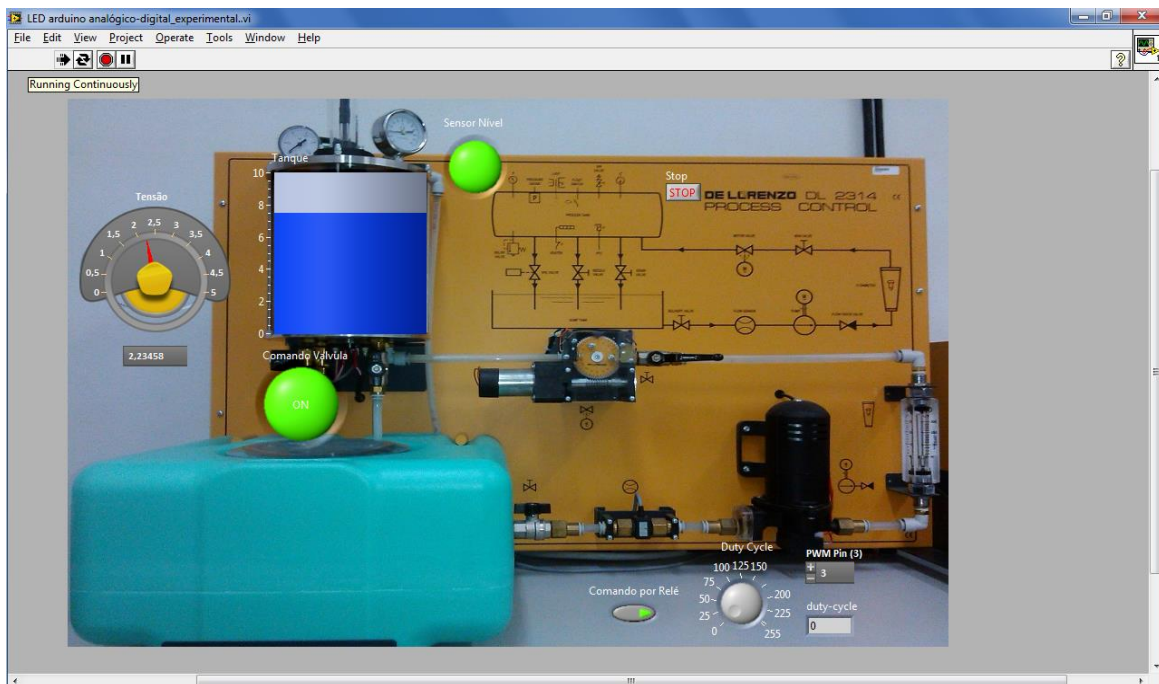


Figura 44- Sensor de nível activado e activação da válvula de drenagem

8 Conclusão

Com este trabalho ficou provado que é possível utilizar o sistema de controlo de processos DeLorenzo DL2314 e adaptá-lo a tecnologias mais actuais.

Apesar de algumas limitações inerentes típicas de um equipamento com tecnologia menos actual, foi mais complicado do ponto técnico dotá-la com monitorização por computador. Mesmo com essas barreiras, foi possível alcançar o objectivo primordial, que é adaptar tecnologia mais actual mantendo a integridade de funcionamento do equipamento com todos os componentes de origem.

Depois dessa fase alcançada, o objectivo seria realizar exercícios didácticos usando os sensores e actuadores disponíveis no sistema de controlo de processo, previamente ligados ao sistema híbrido Arduíno/LabVIEW. Esse objectivo foi alcançado e provado ser possível, embora haja ainda um enorme potencial de desenvolvimento para a realização de outros exercícios didácticos, bem como a adaptação de outros sensores, existentes no sistema de controlo de processos, para a execução de diferentes tipos de sensores.

9 Trabalho futuro

Em termos de trabalho futuro existe, neste projecto, um enorme potencial de desenvolvimento e continuação.

Há a possibilidade de introduzir de forma permanente a placa com o integrado L298N (ponte H) que deu problemas de fiabilidade. Permitirá o controlo do caudal de água fornecida pela bomba de recirculação dando mais liberdade para a criação e resolução de futuros exercícios.

Outra sugestão futura para o desenvolvimento deste projecto passa pela adaptação de mais sensores existentes no sistema de controlo de processos, como por exemplo, o da resistência de aquecimento, permitindo alterar a temperatura da água durante os exercícios.

10 Bibliografia

Direção Geral do Ensino Superior, (2010). *Dez anos de concurso nacional: 2000 – 2009*. Ministério da Ciência, Tecnologia e Ensino Superior

Disponível em: http://www.dges.mctes.pt/NR/rdonlyres/039A454A-C7D8-46AB-BB43-636C2E4AF30C/4576/DGESacesso_DezAnosCN_0009.pdf

Olteanu, R.; Dumitrescu, C.; Gorghiu, G.; Gorghiu, L. (2009). *Related aspects to the impact of virtual instruments implementation in the teaching process*. *Procedia-Social and Behavioral Sciences*, 1 (1): 780-784

Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877042809001426>

Gillich, G.-R.; Frunzaverde, D.; Gillich, N.; Amariei, D. (2010). *The use of virtual instruments in engineering education*. *Procedia-Social and Behavioral Sciences*, 2 (2): 3806-3810

Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877042810006348>

Santori, M. (1990). *An instrument that isn't really (Laboratory Virtual Instrument Engineering Workbench)*. *IEEE Spectrum*, 27 (8): 36-39.

Disponível em: <http://ieeexplore.ieee.org/document/58432/>

Sumathi, S.; Surekha, P. (2007). *LabVIEW based Advanced Instrumentation Systems*. Springer

Goldberg, H. (2000). *What is Virtual Instrumentation?* *Instrumentation & Measurement Magazine*, 3 (4): 10-13

Disponível em: <http://ieeexplore.ieee.org/document/887453/>

Banzi, M. (2011). *Getting Started with Arduino 2nd Edition*. Make: Books

<http://www.ni.com/labview/pt/>

