



Universidade de Aveiro
2011

Departamento de Engenharia Mecânica

**DIOGO MAIA SOARES UTILIZAÇÃO DE SERVIÇOS WEB PARA O
CONTROLO INTEGRADO DE PRODUÇÃO**

Diogo Maia Soares **UTILIZAÇÃO DE SERVIÇOS WEB PARA O
CONTROLO INTEGRADO DE PRODUÇÃO**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação científica do Professor Doutor José Paulo Oliveira Santos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Agradeço este trabalho a quem mo proporcionou e a quem me motivou para nunca desistir. A eles, nunca os esquecerei!

O Júri

Presidente	Professor Doutor Rui Pedro Ramos Cardoso Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro
Arguente	Professor Doutor Francisco José de Oliveira Restivo Professor Associado da Faculdade de Engenharia da Universidade do Porto
Orientador	Professor Doutor José Paulo Oliveira Santos Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

Agradecimentos Este projecto de investigação apenas foi exequível devido à colaboração de muitas pessoas que, graças aos seus incentivos e instruções, me encorajaram sempre a seguir em frente e a nunca desistir.

Tenho a perfeita noção que houve momentos em que desmotivei, pois não conseguia ultrapassar algumas dificuldades em estudo. No entanto, com o apoio científico levado a cabo por parte do Professor Doutor José Paulo Oliveira Santos e o apoio emocional dos meus familiares e amigos, consegui ultrapassar todas as barreiras encontradas. Ao docente em questão agradeço toda a disponibilidade, empenho e paciência que teve para me ouvir, para me compreender, para activar e desenvolver o meu pensamento, para me auxiliar a encontrar soluções sobre algumas problemáticas relativas ao presente estudo.

Agradeço também aos meus familiares e em especial aos meus pais, avós e irmão, pois sempre depositaram em mim uma enorme força e confiança. Estas pessoas foram fundamentais para chegar até esta fase.

À minha namorada por todo o carinho, motivação e incentivo transmitidos no decorrer desta dissertação.

Aos meus colegas e amigos por me acompanharem nesta jornada.

A eles dedico esta dissertação e a todos, o meu muito obrigado!

Palavras-chave	Controlo Integrado de Produção, SCADA, Microcontroladores, ODBC, Web Services, Supervisão, Controlo, TCP/IP, RS232, SPI, PLC.
Resumo	<p>O presente estudo tem como principal finalidade aferir a hipótese de utilização de microcontroladores para substituir máquinas industriais de controlo e monitorização, utilizando uma plataforma SCADA como interface de suporte para o utilizador.</p> <p>O aumento da competitividade na indústria tem aberto portas a sistemas que procuram integrar os vários módulos de produção das empresas, de modo a conseguir que a informação chegue de uma forma perceptível aos diversos níveis existentes numa empresa. Para tal, as plataformas SCADA têm tido um contributo elevado, devido às suas características cada vez mais abrangentes de integração, supervisão, monitorização e controlo.</p> <p>Por outro lado, a tecnologia que suporta os microcontroladores tem evoluído cada vez mais, sendo possível implementar programas de controlo e de comunicação cada vez mais complexos, alargando ainda mais a abrangência das potenciais áreas de utilização destes aparelhos. Além disso, o custo deste tipo de aparelhos e o custo do desenvolvimento de tecnologias associadas aos microcontroladores é bastante inferior quando comparadas com o custo de outras tecnologias de controlo e monitorização industriais.</p> <p>Através da combinação destas duas tecnologias, aliadas à necessidade de redução de custos para uma maior competitividade das empresas, o aluno propõe o controlo e monitorização de uma linha de transferência circular, existente no Laboratório de Sistemas Flexíveis de Produção, do Departamento de Engenharia Mecânica da Universidade de Aveiro.</p>

Keywords Computer Integrated Manufacturing, SCADA, Microcontrollers, ODBC, Web Services, Supervision, Control, TCP/IP, SPI, PLC.

Abstract This study has as its main purpose to assess the possibility of using microcontrollers to replace industrial control and monitoring machinery, using SCADA platform as user support interface.

This alternative arises due to increased competitiveness in industry that has opened doors to systems searching to integrate all the companies' production modules, in a way to achieve that the information reaches, in a noticeable way, to all the existing levels in a company. To this end, the SCADA platforms have had a great contribution, due to their characteristics, even more embrative to integration, supervision, monitoring and control.

On the other hand, the technology that supports microcontrollers has evolved increasingly, being possible to implement control and communication programs even more complex, extending even further the range of potential use of this devices. In addition, the cost of this microcontrollers and the cost of this developing technologies associated with microcontrollers is much lower when compared with the cost of other industrial control and monitoring technologies.

Through de combination of these two technologies, coupled to the need to reduce costs to a bigger company's competitiveness, it is proposed to control and monitor a circular transfer line, from the Flexible Production Systems Laboratory, in the Mechanical Engineering Department, at Aveiro University.

Índice

Capítulo 1.....	1
Introdução	1
1.1 Contexto	1
1.2 Motivação.....	2
1.3 Objectivos.....	4
1.4 Organização da dissertação	4
Capítulo 2.....	6
Estado de Arte	6
2.1 Definição de SCADA	6
2.2 Revisão aos softwares SCADA.....	9
2.3 Projectos desenvolvidos na Área	15
Capítulo 3.....	23
Tecnologias de suporte.....	23
3.1 Protocolos e tecnologias de comunicação	23
3.2 Microcontroladores.....	33
3.3 Programmable Logic Controller	36
Capítulo 4.....	39
Solução Proposta	39
4.1 Arquitectura da solução proposta	40
4.2 Hardware Seleccionado.....	46
4.3 Código desenvolvido	51
4.3.1 Funcionamento do Controlador de Linha.....	52
4.3.2 Funcionamento do Servidor TCP/IP de Linha	60
4.4 Software	66
4.4.1 Cliente TCP/IP	67
4.4.2 Interface Movicon.....	70
Capítulo 5.....	72
Implementação e Análise de Desempenho.....	72
5.1 Implementação	72
5.2 Análise de desempenho	78
Capítulo 6.....	81

Conclusão.....	81
Anexos	85
A.1 Diagramas	87
Referências	94

Lista de Figuras

Figura 2. 1 - Pirâmide de interesses no desenvolvimento de sistemas SCADA-----	6
Figura 2. 2 - Arquitectura de um sistema SCADA genérico [Bentley Systems Inc, 2004]-	8
Figura 2. 3 - Interface de uma plataforma SCADA utilizando Movicon 11 -----	10
Figura 2.4 - Sistema SCADA utilizando plataforma Lookout [National Instruments, 2011] -----	12
Figura 2. 5 - Sistema SCADA desenvolvido utilizando CitectSCADA [Citect, 2011] -----	13
Figura 2. 6 - Monitorização e controlo de processo fabril utilizando a plataforma ZenOn [SPS Magazin, 2011]. -----	14
Figura 2. 7 – Arquitectura SCADA RESLab [Patel et al., 2004]. -----	15
Figura 2. 8 - HMI do sistema SCADA RESLab [Patel et al., 2004]. -----	16
Figura 2. 9 - Arquitectura dos quatro diferentes ensaios efectuados (Barana et al., 2010). -----	17
Figura 2. 10 - Interface PVSS para controlo do armazenamento de gás (EGYPTOIL-GAS, 2011). -----	18
Figura 2. 11 - Vista geral do software EPICS, implantado no projecto KEKB (KEKB, 2000). -----	19
Figura 2. 12 - Sistema SCADA Cliente DEM [Silva e Costa, 2009]. -----	21
Figura 2. 13 - Arquitectura do sistema desenvolvido para automação baseada em serviços Web (Phaithoonbuathong, 2009).-----	22
Figura 3. 1 - Diagrama de funcionamento do módulo de transmissão USART (Microchip, 2001). -----	26
Figura 3. 2 - Bloco de recepção USART (Microchip, 2001). -----	27
Figura 3. 3 - Modelo de Referência OSI, comparativamente ao modelo TCP/IP (Learn Networking, 2008). -----	29
Figura 3. 4 - Implementação SPI entre Mestre e Servo -----	30
Figura 3. 5 - Diagrama de Transferência de Dados SPI-----	31
Figura 3. 6- Componentes de uma ligação ODBC e sua relação (Simba, 2011).-----	32
Figura 3. 7 - Sistema com microcontrolador integrado-----	34
Figura 3. 8 - Diagrama de blocos dos microcontroladores PIC18F4682 e PIC18F4685 (Microchip, 2009”. -----	35
Figura 3. 9 - Arquitectura geral de um PLC (MME, 2006).-----	36
Figura 3. 10 - Componentes físicos de um sistema controlado por PLC (Albertjonas, 2010). -----	37
Figura 5. 1 - Excerto da linha de transporte utilizada no Laboratório -----	72
Figura 5. 2 - Módulo de hardware implementado para o controlo da Linha de Produção -----	73
Figura 5. 3 - Hardware do módulo de comunicação do sistema-----	74
Figura 5. 4 – Interface do objecto de ligação TCP/IP -----	75
Figura 5. 5 - Interface SCADA utilizando Movicon, ecrã principal -----	76
Figura 5. 6 - Ecrã de comando do Sistema, usando Movicon -----	77

Figura A. 1 - Diagrama de Funcionamento da interface SPI (Microchip, 1997) -----	89
Figura A. 2 - Diagrama de operação do bloco TCP/IP da Microchip -----	90
Figura A. 3- Diagrama Eléctrico 1 da Placa Ethernet PICTail Daughter Board da Microchip (Microchip, 2006)-----	91
Figura A. 4 - Diagrama Eléctrico 2 da Placa Ethernet PICTail Daughter Board da Microchip (Microchip, 2006)-----	92

Lista de Tabelas

Tabela 3.1 – Estrutura de uma mensagem Ethernet -----	24
Tabela 3.2 - Estrutura de uma mensagem RS-232 -----	25
Tabela 3.3 - Relação Velocidade versus Distância numa comunicação RS-232 -----	26
Tabela 4.1 – Excerto de variáveis e seus valores utilizados neste sistema -----	41
Tabela 4.2 - Mensagem de resposta com actualização da posição das paletes -----	43
Tabela 4.3 - Mensagem de resposta com actualização sobre ocupação das barragens -----	44
Tabela 4.4 – Estrutura de um pedido de actualização -----	45
Tabela 4.5 - Estrutura da mensagem de comando -----	45
Tabela 4.6 - Mensagem de finalização de comando -----	46
Tabela 4.7 - Entradas e saídas das respectivas barragens -----	56
Tabela 5.1 - Lista dos componentes utilizados e respectivo custo -----	79

Lista de Acrónimos

DMBS	DataBase Mangement System
DSN	Data Source Name
E/S	Entradas e Saídas
HMI	Human Machine Interface
IP	Internet Protocol
MAC	Machine Address
ODBC	Open DataBase Connectivity
PLC	Programmable Logic Controller
RS232	Recommended Standard
RTU	Remote Terminal Unit
SPI	Serial Port Interface
SCADA	Supervisory Control and Data Acquisition
SQL	Standard Query Language
TCP	Transmssion Control Protocol
USART	Universal Synchronous Assynchronous Receiver Transmitter
VB	Visual Basic

Capítulo 1

Introdução

1.1 Contexto

É sabido que “A tecnologia de automação industrial e controlo de processos evoluem rapidamente, a necessidade de máquinas e sistemas cada vez mais flexíveis e reajustáveis que acompanhem as alterações no ciclo de produção têm trazido cada vez mais importância às ferramentas informáticas e de comunicação” (Quintã, 2008).

A crescente competitividade no sector industrial tem conduzido as empresas a procurarem flexibilizar, distribuir e rentabilizar os seus sistemas de produção. Para tal, têm surgido sistemas cada vez mais versáteis, com grande capacidade de integração e com custos inferiores, tendo em conta a sua versatilidade.

As empresas procuram distribuir os seus centros de produção, criando módulos de produção intra e inter-localização fabril, isto é, procuram produzir dentro de uma localização um determinado produto, mas existe uma distribuição da empresa e dos seus produtos por várias localizações, que se estende desde as centenas de metros até aos milhares de quilómetros.

É defendido que “O recurso a processos automatizados possibilita um maior controlo e conhecimento dos mesmos, abrindo portas para os sistemas de supervisão e controlo (SCADA), que possibilitam a supervisão e controlo em tempo real ou com apenas algum atraso. Nas grandes unidades fabris, ETAR’s, instalações eléctricas e mesmo em edifícios é já um dado adquirido o recurso a sistemas SCADA, permitindo, em alguns casos, o acesso a dados via Internet e a supervisão e controlo local ou remoto” (Silva e Costa, 2009).

Para assegurar toda a informação que é processada, para garantir uma aquisição de dados e um controlo de supervisão foram criados sistemas *SCADA*, *Supervisory Control and Data Acquisition*. Este tipo de sistemas engloba toda a obtenção de

informação, a sua transmissão pelos meios físicos, tais como Ethernet TCP/IP, RS232, CAN, Profibus e outros, e o processamento e visualização de dos dados nas várias estações e na estação-mestre, do inglês “*master-station*”.

Relativamente à automação industrial, há vários tipos de equipamentos utilizados para o controlo e comando de máquinas, tais como PLC, *Programmable Logic Controller*, DCS, *Distributed Control System*, robótica, microcontroladores, instrumentação, entre outros.

Na área específica dos microcontroladores, tem havido um desenvolvimento muito acentuado e, apesar de só existirem desde 1978, têm cada vez mais uma influência forte no quotidiano, como por exemplo, em sistemas de gestão de motores automóveis, instrumentos de medição electrónica, sistemas de segurança e automação industrial, entre muitas mais. Esta propagação nos elementos do dia-a-dia deve-se à sua capacidade de memória, à possibilidade de utilizar diversos tipos de comunicação com outros microcontroladores ou outros equipamentos de controlo, leitura e escrita de saídas e entradas digitais e analógicas e ao baixo custo, entre outras características.

Neste contexto, procura-se desenvolver o controlo de uma máquina de um ambiente industrial, nomeadamente uma linha de transporte, através de microcontroladores, criando uma interface SCADA para supervisão, controlo e aquisição de dados de todo o processo a controlar.

1.2 Motivação

Devido à sua interface de fácil interacção com o utilizador, os sistemas SCADA mostram ser uma boa alternativa para o controlo, supervisão e aquisição de dados numa plataforma de produção industrial. É public que “*SCADA is an acronym for Supervisory Control and Data Acquisition. SCADA systems are used to monitor and control a plant or equipment in industries such as telecommunications, water and waste control, energy, oil and gas refining and transportation*” em (NCS, 2004).

Estes sistemas englobam a transferência de dados entre um servidor central SCADA, vários RTU, *Remote Terminal Units*, ou PLC, o servidor central e os terminais do operador. Também se sabe que *“These systems can be relatively simple, such as one that monitors environmental conditions of a small office building, or very complex, such as a system that monitors all the activity in a nuclear power plant or the activity of a municipal water system”* (NCS, 2004).

Em termos de dispositivos de comando, procura-se criar uma alternativa aos sistemas convencionais, que utilizam PLC e que anteriormente controlavam a linha de distribuição do laboratório utilizada nesta investigação, por microcontroladores capazes de controlar máquinas industriais de igual modo que os PLC, mas a um custo bastante inferior, dimensões reduzidas e numa linguagem de programação bastante difundida, quer seja em *“C”*, *Assembly* ou *Visual Basic*.

Devido à sua evolução, hoje em dia os microcontroladores conseguem ter uma boa memória para programação, permitindo a execução de programas bastante complexos, possibilita estabelecer vários tipos de comunicação, tais como comunicação série, Ethernet, wireless, rádio, Bluetooth, com diversos aparelhos, desde sensores, actuadores, motores, outros microcontroladores, servidores, e muito mais.

Pelos motivos aqui referidos, procura-se desenvolver uma solução que integre as potencialidades dos microcontroladores em termos de comunicação, controlo de periféricos e programação, aliada à versatilidade dos sistemas SCADA para monitorizar, integrar dispositivos e domínio desses dispositivos para controlar uma linha de produção que se encontra no Laboratório de Sistemas Flexíveis de Produção.

1.3 Objectivos

O principal objectivo deste trabalho é implementar um protótipo que controle e monitorize linhas de produção fabril, utilizando como plataforma de ensaio uma linha de transferência circular que se encontra no laboratório de Sistema Flexível de Produção.

Este protótipo deve incluir uma aplicação SCADA que gere um sistema de hardware responsável pelo controlo da linha de produção e pela comunicação com a aplicação SCADA.

O sistema SCADA terá, como é fundamental num sistema de monitorização de uma linha de transporte, de monitorizar a ocupação das barragens e a posição actual das paletes da linha de produção. Além desses objectivos, terá de ser capaz de distribuir as paletes pelas barragens da linha numa situação inicial, dar instruções de comando para movimentos de paletes pela linha e identificar o estado actual de operação da linha de produção.

A parte física será responsável pelo processamento de instruções provenientes da plataforma SCADA para movimentos de paletes, durante a fase de comando, deve ter autonomia para distribuir as paletes uniformemente pelas barragens no início do programa e deve ser capaz de enviar mensagens de actualização das suas entradas e saídas, que comandam e monitorizam as barragens. Deve também possuir um algoritmo capaz de fazer o cálculo do posicionamento das paletes em cada instante, e transmitir esse posicionamento à plataforma SCADA.

1.4 Organização da dissertação

Esta dissertação está organizada em 6 capítulos e Y apêndices.

O capítulo dois é caracterizado pela análise à tecnologia principal deste projecto, o sistema SCADA, onde se explica o funcionamento deste tipo de sistema e onde se efectua uma revisão global aos diferentes sistemas SCADA comerciais existentes e qual o mais indicado para este sistema. Além disso, é efectuado um estudo sobre trabalhos

de investigação, nas áreas de sistemas SCADA e implementação de microcontroladores para controlo e monitorização industrial, e as suas características, com o objectivo de enriquecer o trabalho a desenvolver ou criar alternativas ao que já está implementado.

No terceiro capítulo, abordam-se tecnologias que serviram de base para este estudo, abrangendo os diversos protocolos e métodos de comunicação utilizados, além de se analisar as tecnologias de microcontroladores e PLC.

No capítulo quatro apresenta-se o estudo conceptual da solução proposta. Neste expõe-se a interface física do protótipo, o módulo de comunicações e ainda se descreve as aplicações desenvolvidas.

O quinto capítulo é reservado para a exposição da implementação em laboratório do protótipo desenvolvido. Apresenta-se o módulo de hardware e o aspecto das interfaces implementadas para comunicação, monitorização e controlo do sistema. Por fim, neste capítulo é efectuada uma análise às escolhas efectuadas neste estudo.

No sexto capítulo são expostas as conclusões deste estudo e sugestões a considerar para trabalhos futuros.

No apêndice A é descrito o diagrama de funcionamento da interface SPI, implementada pela Microchip, de seguida demonstra-se o modo de operação do bloco TCP/IP, desenvolvido pela Microchip, e por último encontram-se duas figuras com o esquema eléctrico da Placa Ethernet desenvolvida pela mesma empresa referida anteriormente.

Capítulo 2

Estado de Arte

2.1 Definição de SCADA

SCADA é a abreviação para *Supervisory Control and Data Acquisition*, isto é, Supervisão, Controlo e Aquisição de Dados. Estas são as áreas principais de um sistema SCADA. Este necessita de gerir e controlar remotamente todos os dados e processos de uma determinada instalação fabril, que pode ser localizada numa única estação ou distribuída por várias estações.

Na década de 60, surgiram computadores que continham processadores capazes de realizarem funções em tempo real, criados pela Westinghouse e GE, denominados PRODAC e GETAC, respectivamente. A lógica requerida para construir uma estação mestre, capaz de fornecer todas as informações para um sistema SCADA era de tal forma complexa que a vantagem de utilizar computadores tornou-se óbvia. À medida que os equipamentos evoluíam, as funções do sistema de SCADA também progrediam, começando a serem capazes de examinar, monitorizar os dados ou o seu estado e mais tarde, fazer o registo periódico de dados.



Figura 2. 1 - Pirâmide de interesses no desenvolvimento de sistemas SCADA

O desenvolvimento deste tipo de sistemas tornou-se rápido devido ao grande interesse da indústria, figura 2.1, que via nos sistemas SCADA uma grande oportunidade para aumentar a produção, além de reduzir custos, consumo de energia e mão-de-obra, melhorando as condições de segurança e a qualidade no trabalho.

É sabido que “actualmente devido ao progresso e optimização dos meios, os sistemas SCADA permitem a utilização de tecnologias de computação e comunicação para automatizar a monitorização e controlo dos processos industriais, efectuando recolha de dados em ambientes industriais e a respectiva apresentação ao utilizador, com recurso a Interfaces Homem-Máquina, HMI” (Quintas, 2004).

É do conhecimento que “Estas plataformas de integração industrial abrangem aplicações múltiplas, estando presentes desde a indústria de celulose, petrolífera, têxtil, passando pela indústria metalúrgica, automóvel” (Zamprona, 2008). Além das aplicações na indústria, cada vez mais se encontra nas áreas de gestão de sistemas complexos, tais como centrais energéticas, redes municipais de água ou redes de transporte ferroviário.

Em relação a disponibilidade e segurança da informação, estes sistemas têm grande preponderância, garantindo que a informação se encontra disponível e segura, caso seja necessário utilizá-la.

Quanto à eficiência do processo de monitorização e controlo, estas plataformas garantem, em tempo real, a visualização dos dados e dos processos fabris que estão a ocorrer, através de interfaces de fácil interacção e compreensão. Os softwares SCADA efectuam previsões, relatórios e gráficos, facilitando a tomada de decisões para o sistema ou para o operador permitindo, desta forma, um aumento da segurança nos processos e redução de custos. Para António Rocha Quintas, “a velocidade elevada de actualização dos dados é possível devido à utilização de hardware e software normalizado nos sistemas SCADA, juntamente com protocolos de comunicação eficientes, melhorando a conectividade à rede local e à Internet” (2004).

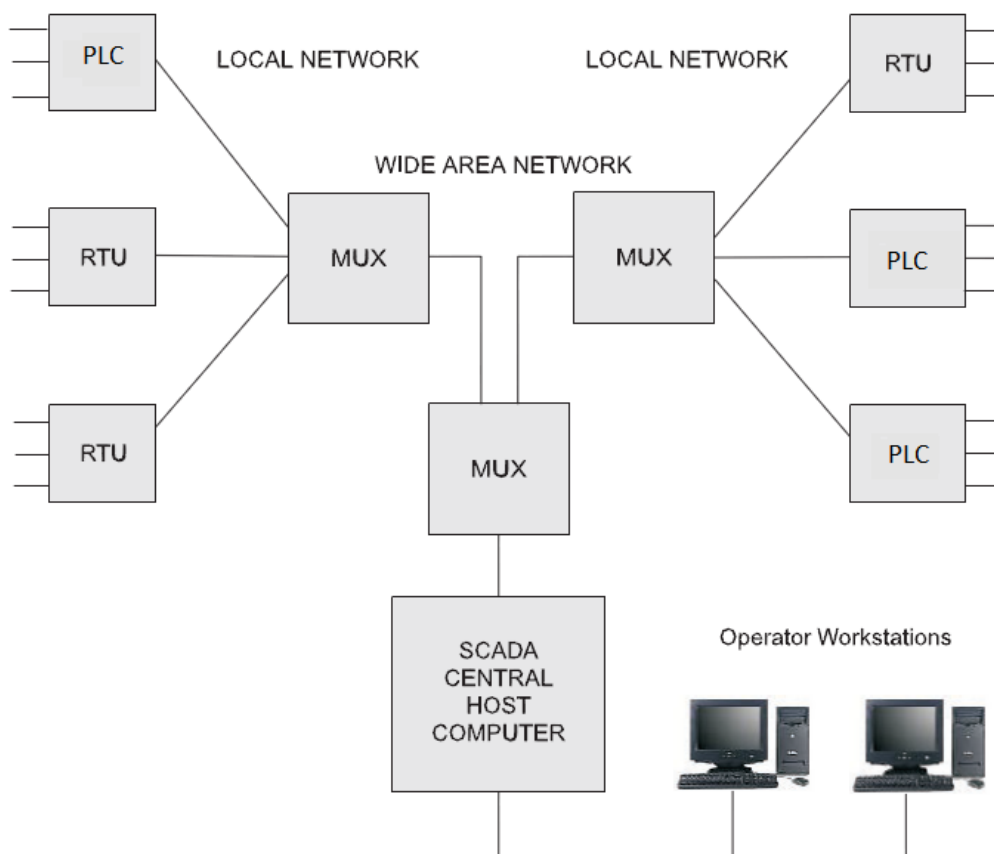


Figura 2. 2 - Arquitectura de um sistema SCADA genérico [Bentley Systems Inc, 2004]

Na figura 2.2 demonstra-se a arquitectura geral de um sistema SCADA. Este é “composto por PLC ou RTU, por um ou diversos sistema de comunicação, além de um ou vários computadores/servidores anfitrião central e uma ou várias Interfaces Homem-Máquina, HMI”, (NCS, 2004). Nesta arquitectura, pode haver mais que uma RTU ou PLC para interagirem com os dispositivos de detecção de campo, caixas de comutação para controlo locais ou actuadores de válvula.

É do conhecimento que “o sistema de comunicações é utilizado para transferir informação entre os RTU ou PLC e os computadores no servidor central SCADA e entre este e a HMI, que pode encontrar-se geograficamente remota. O tipo de comunicação pode ser de vários tipos, desde rádio, satélite, cabo, entre outros ou uma combinação entre estes” (Bentley Systems Inc, 2004).

A interface homem máquina é utilizada para que o servidor central SCADA e a aplicação do terminal de operação suporte o sistema de comunicações, monitorize e controle remotamente os dispositivos de campo. É publico que “pode ainda encontrar-

se ligada a uma base de dados de modo a guardar todos os registos de sensores, actuadores, emergências e até criar históricos de operação. Ao guardar toda esta informação, é capaz de proporcionar gráficos de utilização para análise e gestão dos processos” (Silva e Costa, 2009).

2.2 Revisão aos softwares SCADA

Neste subcapítulo pretende-se fazer uma revisão aos sistemas SCADA existentes no mercado ou desenvolvidos no âmbito de algum tema de dissertação de mestrado ou de doutoramento.

Hoje em dia, os softwares SCADA são tão abrangentes que já não se limitam a monitorizar e controlar entradas e saídas, E/S, ou a processar simples programas automáticos, com interfaces simples, sem acesso a base de dados.

Em países mais desenvolvidos, as companhias são forçadas a compensar o preço mais caro da mão-de-obra, e para tal, tem sido necessário aperfeiçoar os processos de produção tornando-os mais eficientes e transparentes.

De modo a contribuir para resolver esta situação, os sistemas SCADA actualmente já têm a capacidade de recolher e arquivar dados de produção que se traduzem em importantes parâmetros para tomar decisões e que se disponibilizam de forma acessível para os vários utilizadores. Para tal, “é disponibilizada a informação correctamente pelos vários sectores da empresa, tais como o sector de contabilidade, ou logística, ou o departamento de controlo de qualidade, de gestão” (Costa et al., 2007), a empresa só tem a beneficiar com estas informações”.

De seguida apresentam-se os principais softwares SCADA comerciais disponíveis, além de projectos SCADA desenvolvidos no âmbito de mestrados e doutoramentos.

Movicon

É um software SCADA desenvolvido pela Progea, que se encontra na versão 11, sinónimo de vários desenvolvimentos neste sistema. É uma plataforma de SCADA/HMI que “garante flexibilidade de desenvolvimento, versatilidade e independência” (Progea, 2011).

Este software garante fluxo de informação em tempo real entre os vários sectores de uma companhia, mesmo numa empresa de grandes dimensões. A imagem 2.3 ilustra uma interface desenvolvida em Movicon.



Figura 2. 3 - Interface de uma plataforma SCADA utilizando Movicon 11

Esta plataforma é muito versátil e com uma grande abertura no seu modo de operação, dado que a tecnologia é toda baseada em *eXtensive Markup Language*, XML, uma metalinguagem definida pelo padrão internacional do consórcio W3C. Este definiu que “XML é um formato de texto simples e muito flexível, derivado da *Standard Generalized Markup Language*, SGML, ISO 8879” (W3C, 2011). A linguagem

XML caracteriza-se pela sua capacidade de descrever diversos tipos de dados, facilitando a partilha de informações num determinado sistema.

Para além das características mencionadas, “também possibilita a escrita e desenvolvimento de scripts em várias linguagens dado que possui linguagens de programação incorporadas, tais como Visual Basic para Aplicações, VBA, Visual Basic.Net, lista de instruções como num PLC e linguagem de sinapses” (Movicon, 2011).

Além das particularidades descritas anteriormente, esta versão ainda suporta qualquer sistema operativo Windows, a partir do Windows™ CE, possui alarmes e um gestor para estes, possui uma biblioteca de drivers de E/S, análise estatística de produção e protecção de dados.

Lookout

A plataforma Lookout, desenvolvida pela National Instruments é um “software de automação baseados em objectos ActiveX, garantindo facilidade de uso e controlo nas aplicações de automação e de aquisição industriais, sendo o seu processamento conduzido por eventos” (National Instruments, 2011).

Para o desenvolvimento de aplicações de forma simples, apenas é necessário criar objecto e ligá-los, sem necessidade de programação, codificação ou compilação manual.

Esta plataforma possibilita a utilização de uma rede multi-cliente e multi-servidor, garantindo o controlo e a monitorização Web, possuindo histórico distribuído, utilizando cliente e servidor OPC, *Object linking extension for Process Control*, e a deslocalização de alarmes. A figura 2.4 descreve um sistema SCADA utilizado para o controlo de uma instalação fabril.

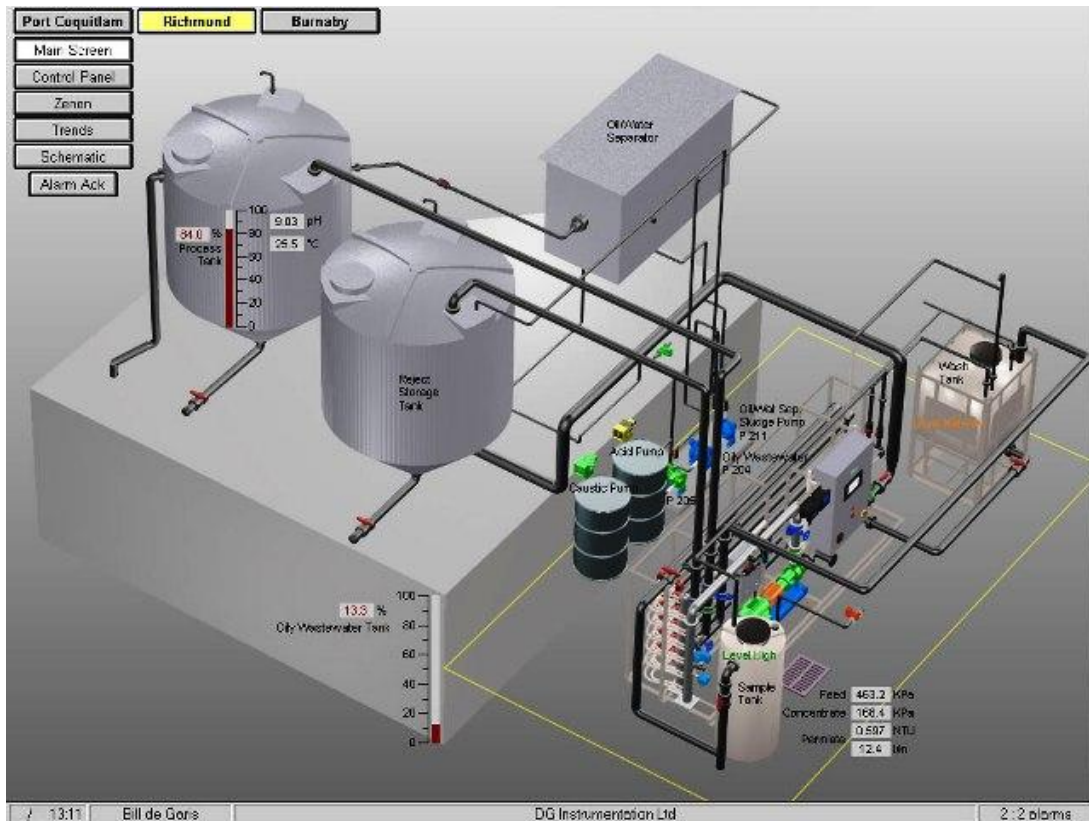


Figura 2.4 - Sistema SCADA utilizando plataforma Lookout [National Instruments, 2011]

CitectSCADA

A companhia Citect, pertencente ao grupo Schneider Electric, é uma empresa de desenvolvimento de software especializada na indústria de automação e controlo, que desenvolveu a plataforma CitectSCADA.

Este software foi criado com o objectivo de fornecer um controlo versátil para operações em tempo real e operações de projecto.

Em termos visuais, é bastante apelativo, como se pode visualizar na figura 2.5, dado que permite a visualização gráfica avançada de processos. Relativamente à análise de produto, esta plataforma permite a análise e o controlo dos produtos através do controlo estatístico de processo, uma ferramenta desenvolvida pela Citect.

Além do referido, também “possibilita a criação de gráficos em tempo real e do histórico de operação, permite a incorporação de relatórios de processo, uma gestão dedicada de alarmes e faculta a criação de opções de agrupamentos avançadas para

controlo, utiliza linguagens de programação Cicode e CitectVBA” (Citect, 2011). No que se refere às ligações às E/S do sistema SCADA, estas são efectuadas através de drivers específicos.



Figura 2. 5 - Sistema SCADA desenvolvido utilizando CitectSCADA [Citect, 2011]

Zenon

A plataforma Zenon é uma ferramenta SCADA desenvolvida pela CopaData, orientada para a automação industrial orientada a objectos, com um design livre. Pode ser utilizada apenas como HMI ou como SCADA.

A interoperabilidade deste sistema cria “interfaces eficientes com qualquer software ou hardware possíveis, sendo capaz de ligar todos os padrões, desde COM, XML e ActiveX, além de um gestor de bases de dados SQL Server 2005” (Copadata, 2011).

Este software é capaz de adquirir dados em tempo real, gerir funcionamentos incorrectos, através de alarmes e ainda possui uma ferramenta responsável pela análise da performance industrial, como características principais.

Além destas potencialidades, permite aos utilizadores do software EPLAN visualizar e controlar partes do sistema sem que sejam precisos conhecimentos adicionais, utilizando os diagramas eléctricos originais, e é possível, durante a fase de desenvolvimento, aceder ao ZenOn através do EPLAN e vice-versa.

Uma das grandes vantagens deste sistema é que “devido à existência de troca de dados entre o desenho da instalação e a manutenção torna possível a engenharia automática” (CopaData, 2011).

A figura 2.6 mostra uma interface da plataforma referida num processo da indústria automóvel.



Figura 2. 6 - Monitorização e controlo de processo fabril utilizando a plataforma ZenOn [SPS Magazin, 2011].

2.3 Projectos desenvolvidos na Área

RESLab

A plataforma RESLab é um sistema SCADA desenvolvido por investigadores da Universidade de Murdoch, na Austrália, com o objectivo de comunicar com o laboratório RESLab e controlar os seus dispositivos de E/S.

Este laboratório consiste numa instalação dedicada aos sistemas de energias renováveis e a testes de componentes para esses sistemas.

Este sistema SCADA difere dos sistemas tradicionais dado que “suporta um ambiente de operação com contínuas alterações dependendo do teste a ser realizado” (Patel et al., 2004).

A figura 2.7 descreve o módulo geral de funcionamento desta implementação SCADA.

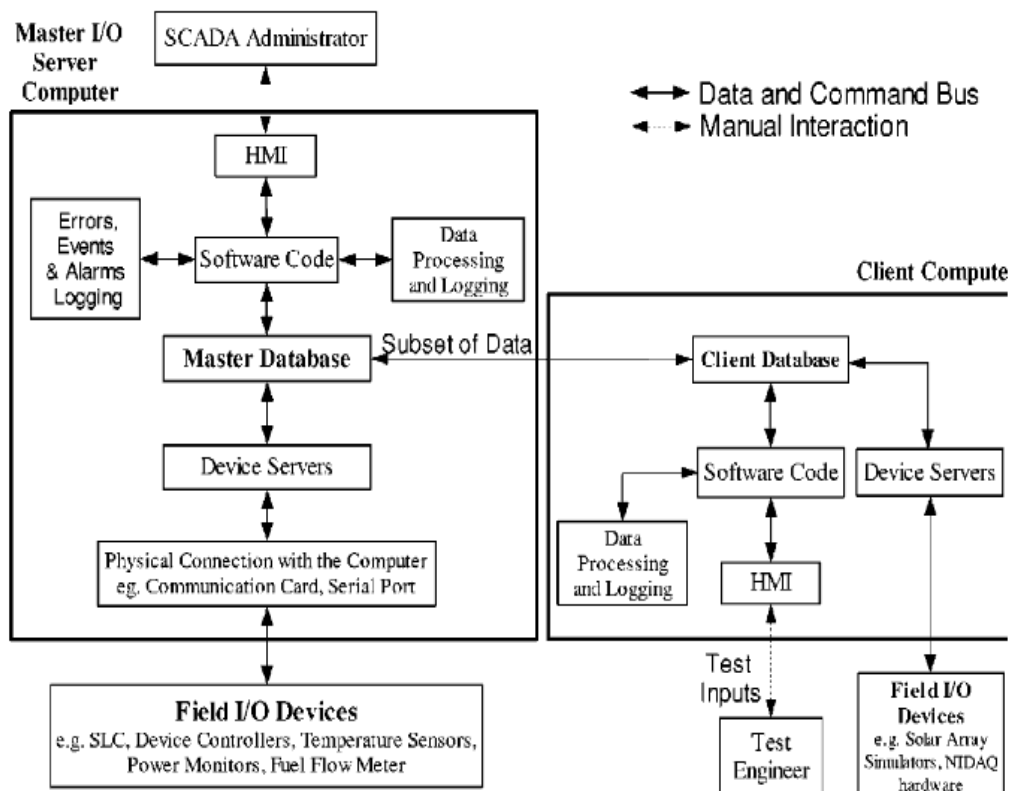


Figura 2. 7 – Arquitectura SCADA RESLab [Patel et al., 2004].

A plataforma RESLab é baseada na linguagem de programação visual LabVIEW, da National Instruments, e no seu módulo *Datalogging and Supervisory Control*.

Nesta plataforma, o servidor mestre de E/S é responsável pela gestão da base de dados, que é partilhada com os clientes, pelo controlo de erros, eventos, alarmes, e também pelo processamento de dados e histórico de operações, pelas funções como servidor e, além disso, pela comunicação com as suas E/S de baixo nível.

Relativamente ao cliente incorporado no sistema, este gere a sua base de dados, que está ligada à base de dados central. Além desta gestão, o cliente executa o seu código, processa os seus dados e o seu histórico enquanto que a sua HMI recebe as entradas dos testes de engenharia efectuados.

Os servidores do dispositivo comunicam com a base de dados, escrevendo e lendo na mesma. Também controlam os aparelhos de baixo nível, como painéis solares, a placa de aquisição de dados na National Instrument, NDAQ, entre outros.

A figura 2.8 demonstra a interface criada deste sistema para o controlo e monitorização do laboratório.

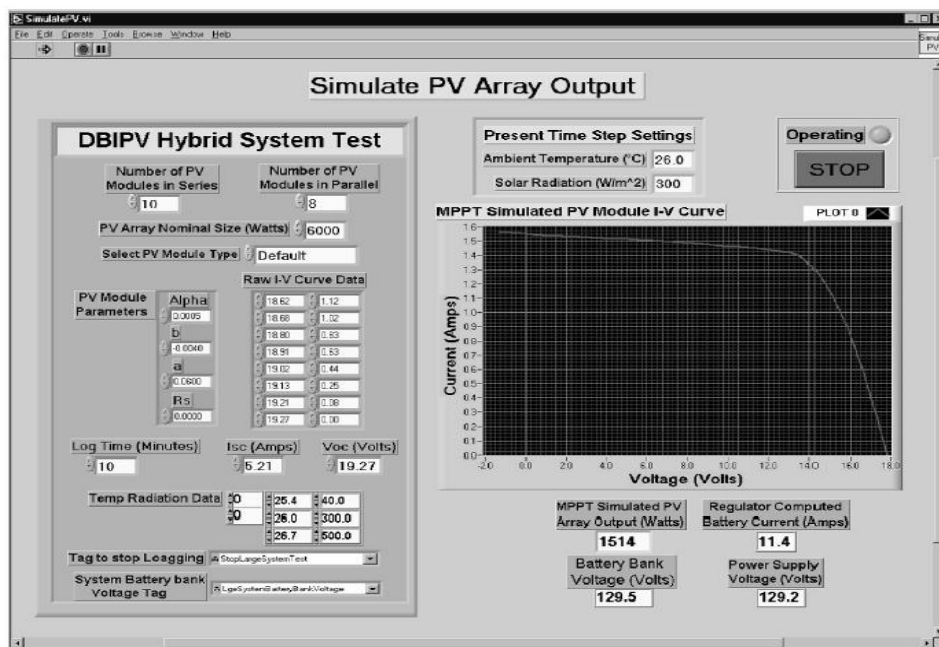


Figura 2. 8 - HMI do sistema SCADA RESLab [Patel et al., 2004].

EPICS

Em 2010, um grupo de investigadores em Itália efectuou um estudo sobre qual o software SCADA mais indicado para o controlo de dois aparelhos experimentais, nomeadamente um denominado SPIDER, aparelho protótipo para a produção de iões de Deutério extraídos de plasma por radiofrequência, e outro denominado MITICA, um aparelho injectador de MegaVolts da ITER.

Para os autores deste ensaio, o objectivo era “garantir a operação correcta de SPIDER e MITICA através da automatização dos processos e a monitorização da instalação, implementado através de controladores adequados juntamente com o software SCADA apropriado” (Barana et al., 2010).

Para tal, foram testados quatro softwares SCADA, duas versões comerciais, e os restantes dois softwares são Open-Source, isto é, software desenvolvido em código livre para poderem ser alterado pelos utilizadores.

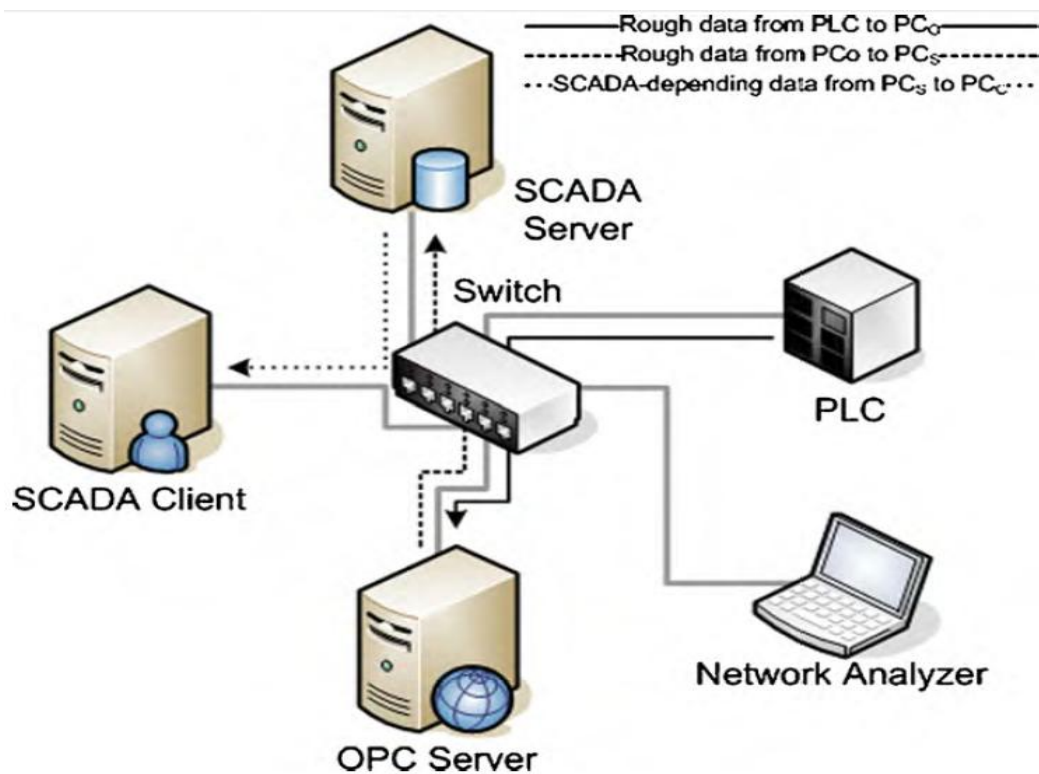


Figura 2. 9 - Arquitectura dos quatro diferentes ensaios efectuados (Barana et al., 2010).

O teste efectuado consistiu na troca de informações entre o sistema SCADA e um PLC, em que se estudou a facilidade de desenvolvimento de uma interface composta por um gráfico que registasse as variações nas leituras e um painel onde se pudesse definir os parâmetros a utilizar, o modo de preparação de dois tipos de comunicação TCP/IP. Uma das ligações era directa entre o SCADA e o PLC e a restante era uma ligação indirecta através de um servidor OPC, ilustrado na figura 2.9 .

Neste estudo concluiu-se que os SCADAs EPICS e PVSS são basicamente equivalentes. Enquanto que o primeiro teve uma performance de rede melhor, o segundo provou ser melhor na configuração da comunicação, graças à interface de utilizador que facilitava a escolha do driver e da relação do grupo a ser associado com cada variável.

Ambas as plataformas SCADA são apoiadas por instituições de renome, sendo que o sistema central de controlo CODAC apoia o software EPICS, PVSS é “apadrinhado” pelo CERN. As figuras 2.10 e 2.11 demonstram as interfaces das plataformas referidas.

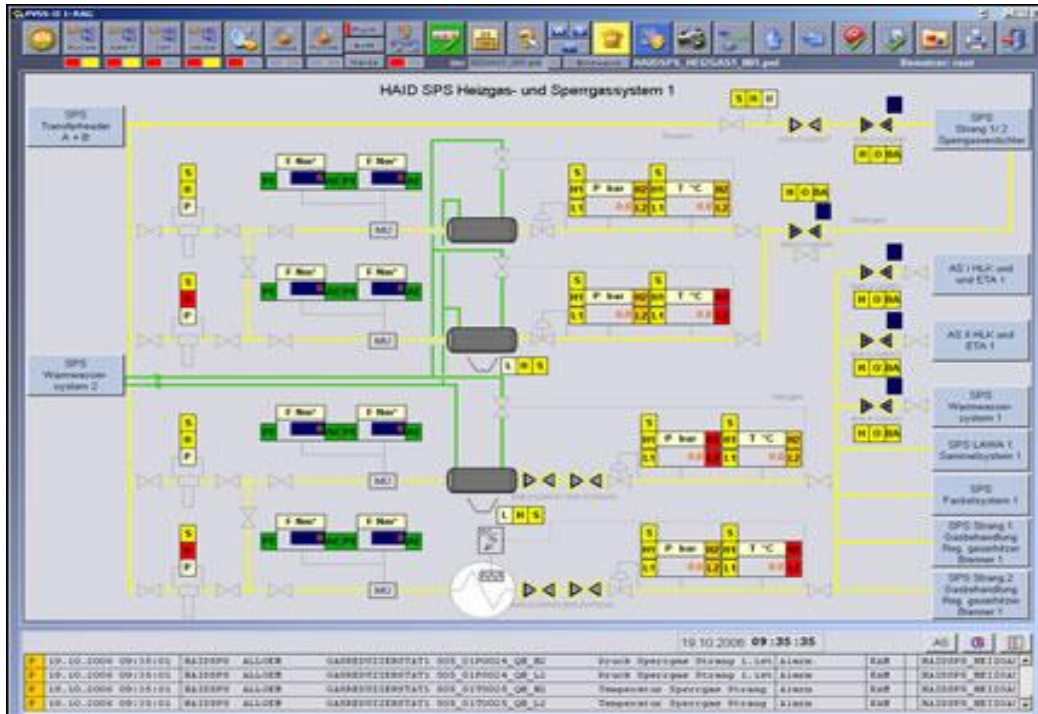


Figura 2. 10 - Interface PVSS para controlo do armazenamento de gás (EGYPTOIL-GAS, 2011).

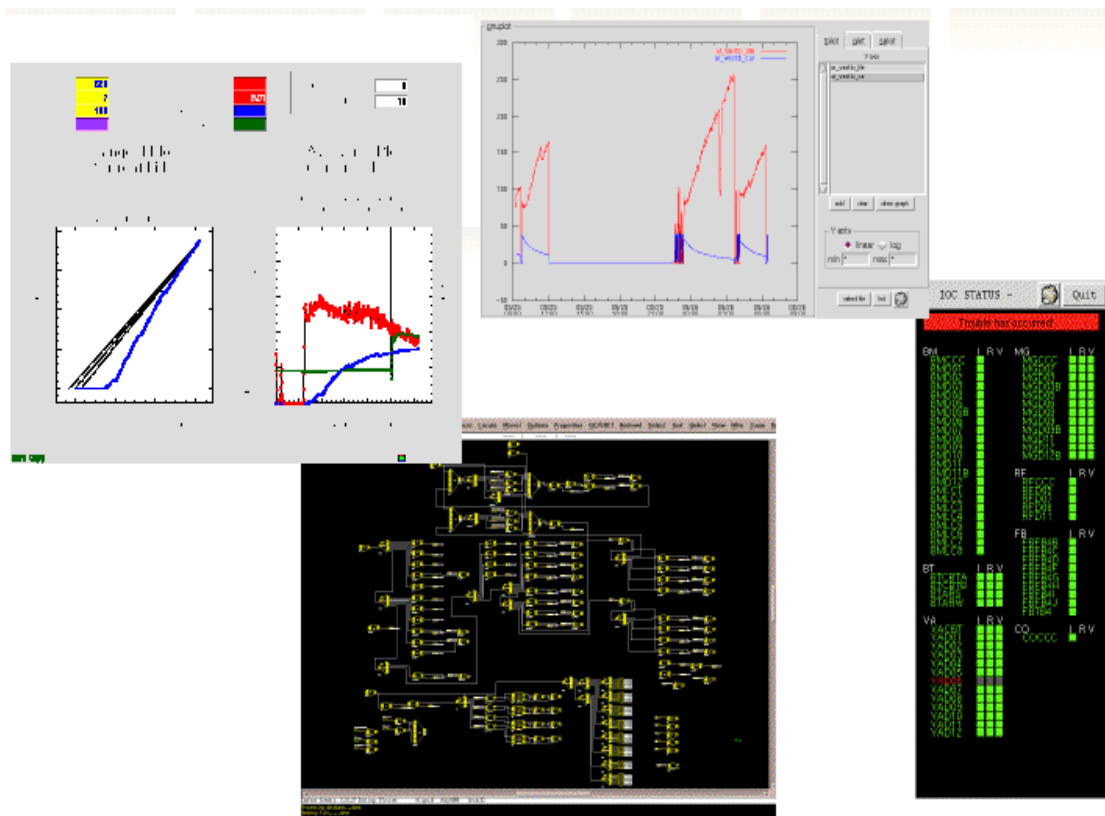


Figura 2. 11 - Vista geral do software EPICS, implantado no projecto KEKB (KEKB, 2000).

Sistema Flexível de Produção

No seu estudo, André Quintã “analisou arquitecturas orientadas a serviços, SOA, implementações de serviços Web e os seus padrões SOAP, WSDL e UDDI. Este estudo teve como objectivo a utilização dessas arquitecturas como padrão para sistemas de produção flexíveis, modulares, cooperativos e descentralizados” (Quintã, 2008).

No seu trabalho efectuado, desenvolveu infra-estruturas informáticas para três casos de estudo, utilizando SOA, que envolveu vários recursos industriais de produção e transporte.

Um dos recursos utilizados foi a linha de transferência circular utilizada pelo autor da actual dissertação. Nesse projecto, “o controlo foi efectuado através de PLCs industriais, nomeadamente o modelo A1S da Mitsubishi, que possui cartas de

comunicação RS232 e TCP/IP, comunicando com o PC através do driver MX Components” [Quintã, 2008].

No caso de estudo referido, o autor utilizou a linha de transferência, o terceiro caso, no qual foi implementada uma infra-estrutura baseada em SOA para o controlo integrado do sistema flexível de produção do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Para o controlo integrado do referido caso, desenvolveram-se várias aplicações informáticas, de acordo com a arquitectura proposta, enquanto que para o controlo individual de cada recurso foram desenvolvidas aplicações locais, serviços Web e uma aplicação central, que utiliza os vários recursos implementados.

ClienteDEM

No projecto desenvolvido por Diogo Silva e Costa desenvolveu-se uma “solução que permitiu a supervisão e controlo de PLCs, de uma forma local e remota, de fácil integração com os sistemas existentes e que possibilita que o utilizador aplique as funções criadas numa aplicação remota, de forma transparente e simples, mediante a sua necessidade” (Silva e Costa, 2009).

O autor desenvolveu serviços Web que permitiram aceder a recursos remotos, nomeadamente PLCs. Os serviços em causa permitiram a supervisão e o controlo dos equipamentos, sendo o sistema implementado do tipo cliente-servidor, com uma arquitectura descentralizada e orientada a serviços. Isto possibilita a integração de serviços em aplicações remotas e a monitorização e controlo remoto de equipamentos.

A figura 2.12 ilustra o sistema desenvolvido pelo autor.



Figura 2. 12 - Sistema SCADA Cliente DEM [Silva e Costa, 2009].

Na opinião do autor, e fazendo uma análise qualitativa e comparativa aos servidores OPC, “o sistema desenvolvido revela características e valências idênticas às de um servidor OPC, apresentando, no entanto, a vantagem de permitir a supervisão e controlo remoto de equipamentos, de forma completa, e pelo facto de se tratar de uma solução, em parte, *freeware*” (Silva e Costa, 2009).

Controlo e supervisão de produção, utilizando serviços Web

Neste projecto, os autores desenvolveram um estudo no qual demonstram a utilização de serviços Web, tanto na construção de funcionalidades de controlo de componentes de dispositivos, como monitorização de processos e sistemas de planeamento. Também no que se refere à habilidade para reconfigurar e gerir

dispositivos de baixo nível a partir de níveis de controlo de produção e gestão de topo, os autores desenvolveram uma interface de ligação por serviços Web e comunicação por mensagens SOAP, *Simple Object Access Protocol*, entre sistemas de controlo e aplicações de gestão.

Os autores implementam o protocolo de controlo de baixo nível em microcontroladores incorporados, baseados na arquitectura ARM9, comunicando por TCP/IP com o restante sistema.

A figura 2.13 procura esclarecer o sistema implementado nesta investigação.

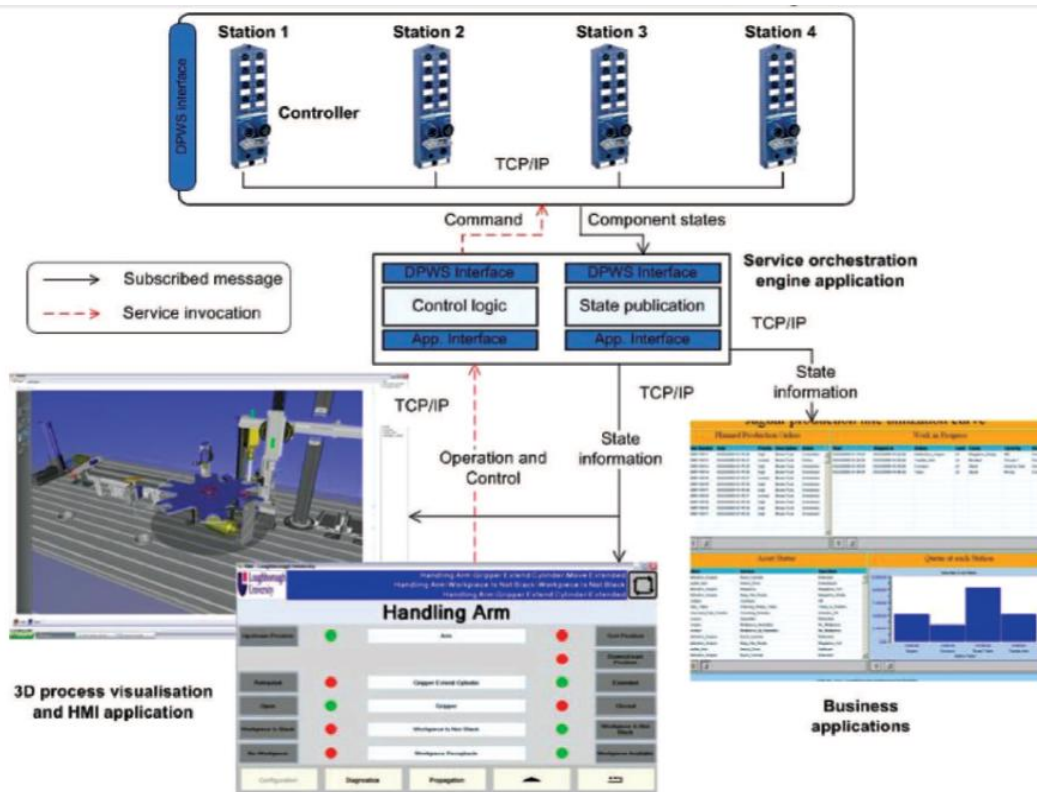


Figura 2. 13 - Arquitectura do sistema desenvolvido para automação baseada em serviços Web (Phaithoonbuathong, 2009).

Capítulo 3

Tecnologias de suporte

3.1 Protocolos e tecnologias de comunicação

Ethernet

É do conhecimento que “No final da década de sessenta, Norman Abramson e os seus colegas desenvolveram uma rede por rádio transmissão, para comunicar entre as diversas ilhas havaianas, e deram à rede o nome de Aloha. Metcalfe apercebeu-se que poderia implementar melhorias nesta rede, nomeadamente na arbitrariedade do acesso a um canal de comunicações partilhado e daí iniciou o desenvolvimento da Ethernet” (Spurgeon, 2000).

Ainda acerca da história da Ethernet, “Metcalfe desenvolveu um sistema que incluía um mecanismo de detecção quando ocorresse uma colisão (*Collision Detect*), um outro mecanismo de escuta antes de transmitir (*Carrier Sense*), no qual as estações esperavam por actividade antes de transmitirem e outro para suportar o acesso a um canal partilhado por múltiplas estações (*Multiple Access*). Juntando todos estes mecanismos, percebe-se porque o protocolo de acesso a canais Ethernet se chama *Carrier Sense Multiple Access with Collision Detect (CSMA/CD)*” (Spurgeon, 2000).

Na actualidade, existem diversos tipos de Ethernet, que diferem umas das outras na sua velocidade e no meio físico de transmissão. Relativamente à primeira característica diferenciadora, podem ser distinguidos em:

- Standard Ethernet (10Mbit/s).
- Fast Ethernet (100Mbit/s).
- Gigabit Ethernet (1000Mbits/s).
- 10-Gigabit Ethernet (10000Mbit/s).

Comparativamente ao meio físico de transmissão, pode-se implementar uma rede Ethernet pelos seguintes meios:

- Cabo coaxial.
- Par entrançado.
- Fibra óptica.
- Sem fios (Wireless).

É sabido que “a estrutura de uma mensagem Ethernet é definida pelo padrão IEE 802.3., introduzido em 1997, que não inclui o preâmbulo e o início de mensagem. A dimensão mínima de uma mensagem é de 64bytes e um máximo de 1518bytes. A tabela 3.1 demonstra a constituição de uma mensagem Ethernet” (Kasim, 2008).

Tabela 3.1 - Estrutura de uma mensagem Ethernet

Campo	Bytes
Preâmbulo (Indica a transmissão de uma mensagem)	7
Delimitador do Start-Of-Frame (SOF)	1
Endereço de Destino (DA)	6
Endereço do Transmissor (SA)	6
Comprimento/ Tipo	2
Dados	46 – 1500
Sequência de Verificação de Mensagem (FCS)	4

RS232

Em 1962, criou-se um padrão para estabelecer a comunicação entre uma máquina de escrever e um modem. Este foi o primeiro de inúmeras aplicações onde a comunicação por RS232 esteve envolvida.

A *Electronics Industries Association* definiu em 1969 o padrão RS-232-C, definindo os níveis de tensão, características mecânicas da interface, tais como as fichas de conexão macho-fêmea e a identificação dos pinos, entre outras características. No entanto, não definiu a codificação dos caracteres a utilizar, do tipo ASCII, código Baudot ou EBCDIC, não definiu o enquadramento dos caracteres no fluxo de dados, isto é, a sua posição na mensagem a transmitir e muito mais.

É sabido que “actualmente, no protocolo de comunicação RS-232, os caracteres são enviados um a um, como um conjunto de bits, utilizando comumente uma codificação “start-stop assíncrono”, sendo que cada mensagem necessita de enviar pelo menos dez bits para enviar um caracter” (Wikipedia, 2011). A tabela 3.2 demonstra a estrutura típica de uma mensagem RS-232.

Tabela 3.2 - Estrutura de uma mensagem RS-232

Campo	Start bit	Dados	Bit Paridade	Stop Bit
Bits	1	7/8	1	1/1.5/2

Relativamente aos conectores, estes são em forma de D, e podem ter desde nove a vinte e cinco pinos. Cada pino é dedicado a uma função específica, dos quais convém salientar os mais utilizados numa comunicação série, que são os pinos dois e três. O pino dois é responsável pela recepção de dados, enquanto que o pino três é responsável pela sua transmissão.

Em termos de distância de conexão, é recomendado que os dispositivos não estejam muito distantes, máximo de quinze metros para poder usufruir do máximo da velocidade de transmissão, dado que esta varia com a distância entre dispositivos. A tabela 3.3 demonstra a variação da taxa de transmissão com o comprimento de cabo.

Tabela 3.3 - Relação Velocidade versus Distância numa comunicação RS-232

Baud rate	Comprimento máximo cabo (m)
19200	15
9600	150
4800	304
2400	914

A tensão máxima e mínima que se pode aplicar, utilizando este tipo de comunicação é 3V a 15V e -3V a -15V, respectivamente, em que a tensão positiva equivale ao zero e a negativa equivale ao um.

A Microchip possui uma interface dedicada para este tipo de comunicação denominada USART. Este termo USART provém da expressão *Universal Synchronous Asynchronous Receiver Transmitter*, sendo um padrão existente para a comunicação série síncrona e assíncrona.

As operações síncronas utilizam um relógio e bus de dados enquanto que numa transmissão assíncrona não existe relógio dedicado acompanhando os dados no bus, o que faz com que o modo mais comum de comunicação seja no modo assíncrono.

O módulo de transmissão USART é descrito na figura 3.1

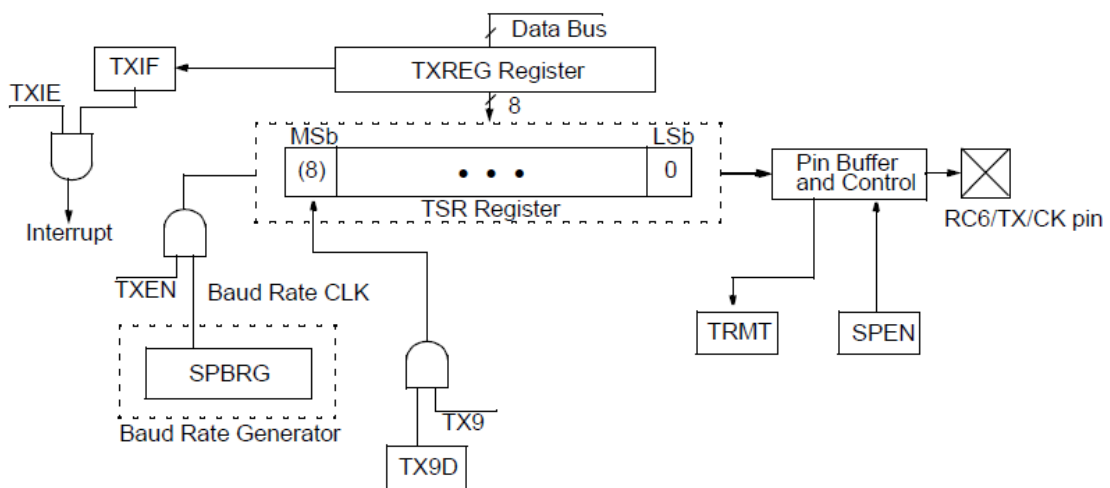


Figura 3. 1 - Diagrama de funcionamento do módulo de transmissão USART (Microchip, 2001).

Este bloco é responsável pela transmissão de dados por RS232 e fica activo quando o bit TXEN assume valor um. Assim que o stop bit da mensagem anterior é enviado e o gerador de BaudRate, BRG, produz uma mudança no relógio, o registo TSR é carregado com os dados a enviar. Ao fazê-lo, o registo TXREG fica vazio e a flag de interrupção à transmissão, TXIF, é despoletada. Assim que o registo TSR fica vazio, é activa a variável TRMT.

Relativamente ao módulo de recepção assíncrona, este encontra-se demonstrado na figura 3.2.

Os dados são recebidos no pino RC7, e são reencaminhados para o bloco de recuperação de dados, onde, se a recepção contínua estiver activa, CREN, são enviados, a uma velocidade 16 vezes superior ao baudrate, para o registo RSR. É do conhecimento que “assim que se detecta o stop bit, este registo conduz os dados para o registo RCREG, caso este esteja vazio. Ao completar a transferência, a variável RCIF é activada” (Microchip, 2001).

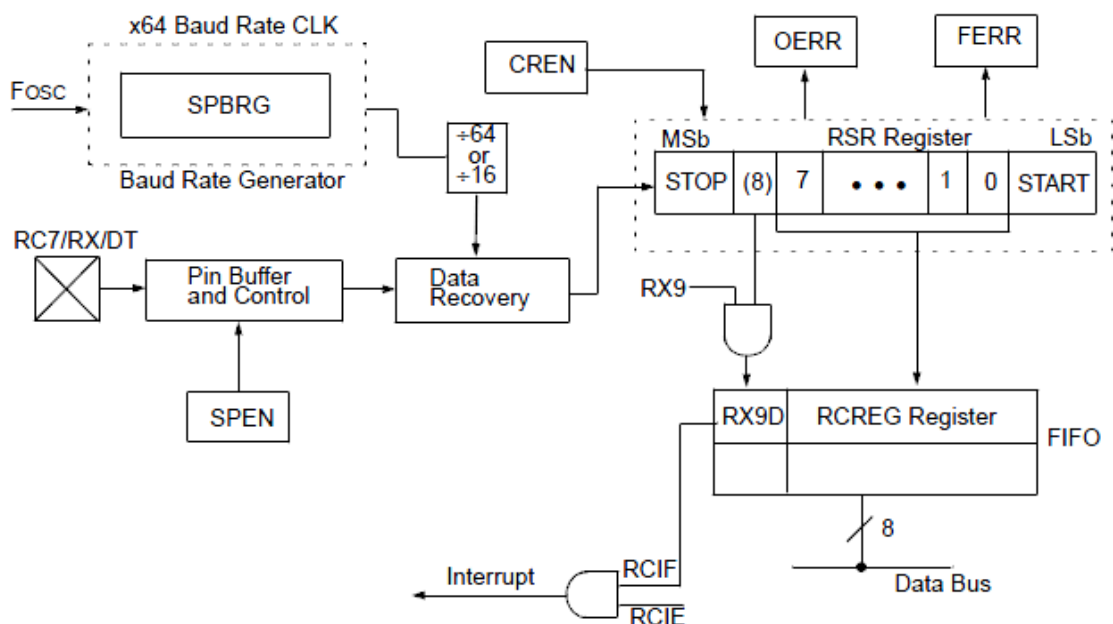


Figura 3. 2 - Bloco de recepção USART (Microchip, 2001).

TCP/IP

Em termos de história do protocolo TCP/IP, “por alturas entre a década de sessenta e setenta houve um grande desenvolvimento de redes e protocolos, mas quando se começou a tentar ligar os diversos protocolos, houve uma grande confusão sobre qual das redes se haveria de implementar, qual o protocolo a utilizar para uma determinada aplicação, e isso causou um grande problema” (Parziale et al., 2006).

Ainda nessa época, “a problemática derivada pela diversidade de protocolos foi reconhecida nos primórdios dos anos setenta, por um grupo de investigadores financiados pelo *Defense Advanced Research Projects Agency* (DARPA). O seu trabalho focou-se na interligação de redes, através da definição de um conjunto de protocolos de modo a que as aplicações pudessem comunicar entre elas, independentemente da tecnologia de rede que utilizassem” (Parziale et al., 2006).

Em 1974 implementou-se a rede ARPANET, constituída por sistemas chamados de 1822. Mas rapidamente o tráfego de dados excedeu a capacidade das linhas de 56Kbps, e à ARPANET sucedeu-se a NSFNET e a MILNET, para efeitos militares.

A NSFNET foi evoluindo, foi-se tornando mais comercial e hoje em dia é um meio central para a partilha de informação e comunicação do mundo, sendo que evoluiu para a Internet que conhecemos de hoje em dia.

O modelo de referência *Open Systems Interconnection* (OSI), ISSO 7498, define um modelo de sete camadas de comunicação de dados com transporte físico na camada inferior e protocolos das aplicações nos níveis cimeiros.

Este modelo é “aceite como base para o entendimento de como uma pilha de protocolo de rede deve operar e serve como referência para comparação da implementação de pilhas de rede” (Parziale et al., 2006).

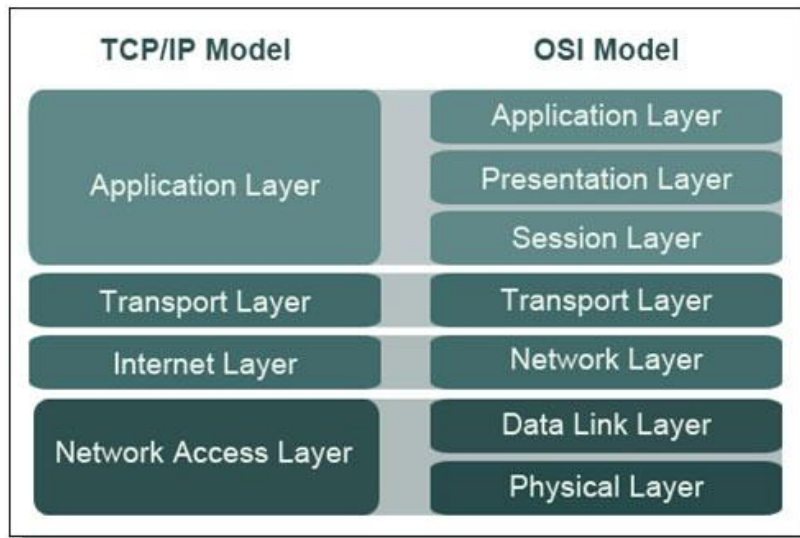


Figura 3. 3 - Modelo de Referência OSI, comparativamente ao modelo TCP/IP (Learn Networking, 2008).

Na figura 3.3 compara-se o modelo OSI com o modelo TCP/IP de modo a que se consiga perceber onde se situa e como funciona, de uma maneira muito geral, este protocolo.

O TCP, *Transmission Control Protocol*, é um protocolo versátil e robusto, ideal para redes globais de grande fluxo dado que assegura a entrega de pacotes de dados e é a base para o protocolo *HyperText Transfer Protocol*, HTTP. Este protocolo situa-se na camada de transporte do modelo OSI e assegura o envio e recepção dos dados, na sequência devida e sem erros, pela rede, além de conseguir recuperar pacotes perdidos, eliminar pacotes duplicados e é capaz de restabelecer a ligação caso haja problema no sistema ou na rede, devido à sua ligação ponto a ponto entre as duas aplicações.

O protocolo *Internet Protocol*, IP, é utilizado pelo TCP para entrega dos pacotes na rede, sendo cada protocolo identificado pelo seu IP, independentemente do sistema ou da rede que exista entre dois utilizadores. A comunicação, no entanto, é efectuada num modo não confirmado, não havendo a garantia da entrega dos pacotes de dados. Estes dois protocolos em conjunto suportam a rede de Internet, sendo o suporte para transporte e endereçamento das especificações aqui referidas.

SPI

SPI é o acrónimo de *Serial Port Interface* e é um protocolo síncrono que permite a comunicação entre um dispositivo *Master* e um dispositivo *Slave*. Foi estabelecido pela Motorola, e implementado, numa fase inicial, para os seus processadores de comunicação, nomeadamente MPC8260 e para o microcontrolador M68HC11.

Os dispositivos conectados à rede estão sincronizados devido ao sinal de *clock* fornecido pelo mestre, notificando quando os dados são alterados e quando estão válidos para leitura. O dispositivo mestre controla o *clock* da comunicação e este controla toda a comunicação na rede, limitando-a aos momentos em que existe um sinal de relógio e por isso, todos os servos ficam dependentes do *clock*.

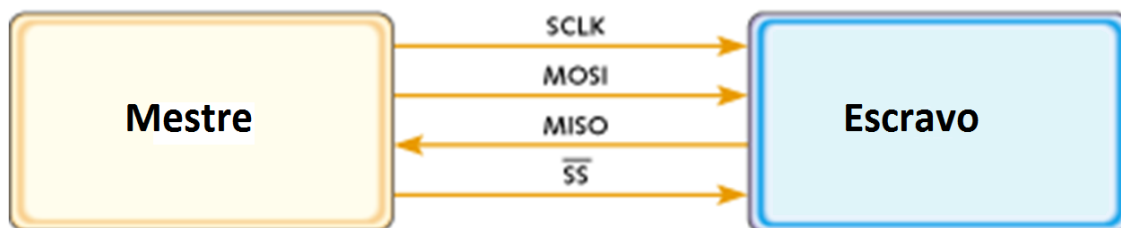


Figura 3. 4 - Implementação SPI entre Mestre e Servo

O protocolo SPI especifica quatro sinais, numa ligação Mestre Escravo, como ilustra a figura 3.4. O primeiro, SCLK, corresponde ao sinal de clock, gerado pelo dispositivo mestre e enviado para o servo. O segundo e terceiro são referentes à transmissão de dados, respectivamente *Master Data Output Slave Data Input* e *Master Data Input Slave Data Output*. Esta característica deve-se ao facto do protocolo SPI implementar a troca de informação, isto é, a comunicação é *full-duplex* entre os seus dispositivos, embora seja o mestre que regule toda a transmissão através do *clock*. O último sinal é responsável pela selecção do servo que deve interpretar a mensagem, *Slave Select* ou *Chip Select*.

O diagrama 3.5 ilustra a transferência de dados por SPI, de um modo geral, implementada pela Microchip.

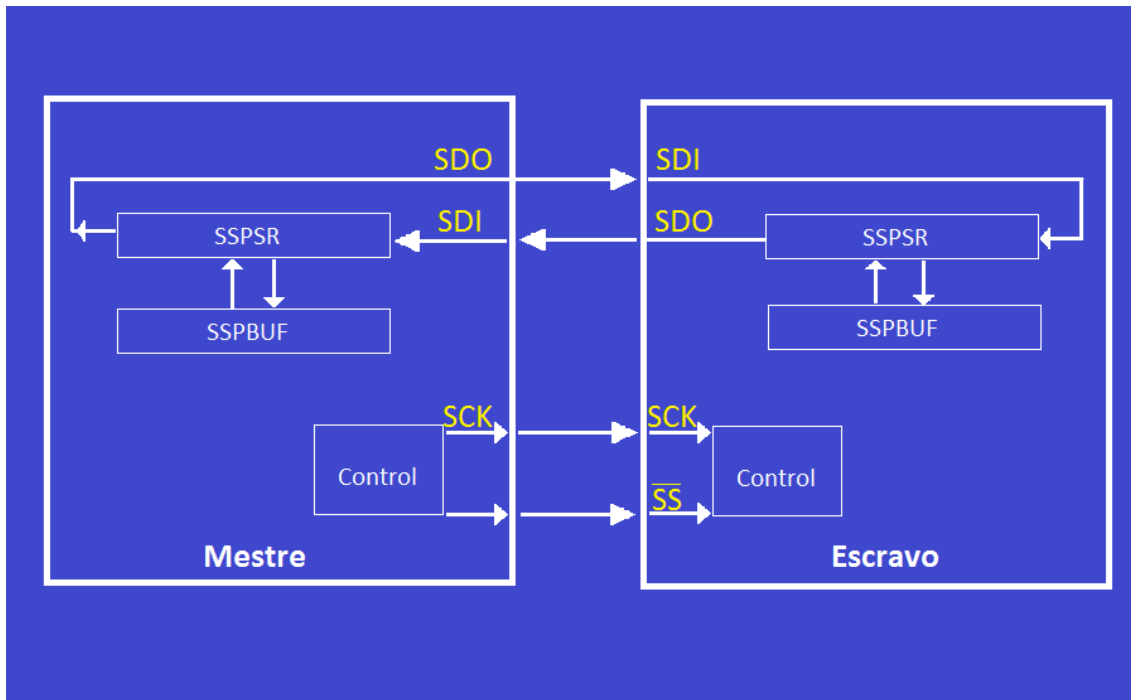


Figura 3. 5 - Diagrama de Transferência de Dados SPI

Pode-se constatar na figura 3.5 que “o SPI cria um loop de dados entre dois dispositivos. Neste loop, os dados são transmitidos pelo mestre pela linha SDO, *Serial Data Output*, e os dados são recebidos pela linha série de entrada de dados, SDI. O clock, SCK, controla quando e quão rápido a informação é trocada pelos dois dispositivos, e o sinal SS permite que o dispositivo mestre controle o endereço particular de um servo, permitindo que hajam vários servos e simplifica a comunicação” (Microchip, 1997).

Os registos SSPSR são registos de carregamento de dados de transmissão e recepção, que podem ser carregados pelos dados recebidos ou pelos dados a enviar pelo buffer SSPBUF. Este buffer, além de armazenar a informação para transmitir ao registo SSPSR, também recebe a informação que provém deste para ser processada pelo microcontrolador.

ODBC

ODBC é a sigla para *Open DataBase Connectivity*, um padrão que é uma interface de aplicações de programação para aceder a ficheiros com dados. Noutras palavras, ODBC permite que haja a troca simples de informação entre aplicações que suportam ODBC.

Este padrão foi desenvolvido em 1992 pelo *SQL Access Group* para generalizar a utilização de um sistema de gestão de base de dados, *DBMS*, por uma aplicação.

É definido como “uma aplicação que utilize um driver ODBC consegue ligar-se a uma *DMBS*, de modo a usar uma *data source*, isto é, um ficheiro que permita a ligação para leitura e escrita nos campos da base de dados, utilizando linguagem *SQL*” (Simba, 2011). A figura 3.6 descreve os componentes de uma ligação ODBC e a sua relação.

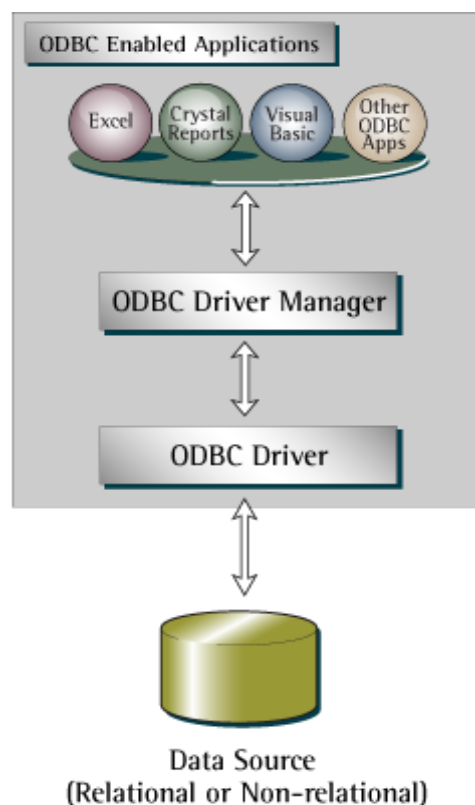


Figura 3. 6- Componentes de uma ligação ODBC e sua relação (Simba, 2011).

Em termos de gestor de drivers ODBC, o Windows possui uma ferramenta denominada *ODBC Data Source Administrator*, que é responsável por controlar todos

os drivers e é através do qual se criam novos ficheiros DSN, *Data Source Name*. Os sistemas operativos diferentes do Windows podem escolher versões *open-source* do gestor de drivers ODBC, como por exemplo unixODBC ou iODBC, ou então optar por versões comerciais.

O gestor dos drivers ODBC processa chamadas de funções ODBC ou passa-as para um driver ODBC e resolve conflitos de versões ODBC. O driver ODBC processa as chamadas de função ODBC, submete pedidos SQL para uma data source e retorna os resultados para a aplicação. Ainda é capaz de modificar um comando da aplicação para que este esteja conforme a sintaxe suportada pela base de dados associada.

A data source consiste nos dados que o utilizador pretende aceder e o seu sistema de gestão da base de dados.

3.2 Microcontroladores

Um microcontrolador, também referido como MCU ou μC , é definido como “um circuito integrado composto por, no mínimo, um microprocessador, memórias de dados e programa e portas de entrada e saída” (Ibrahim, 2010).

No início da década de 70, a Intel produz o primeiro microprocessador, de quatro bits, chamado 4004, utilizado em calculadoras Basicom. Em meados dessa década, é fabricado o processador 8080, vanguardista na sua memória de 8bits. Em 1978, a Intel introduz o primeiro processador de 16bits, o 8086.

Nesta fase, os microprocessadores começavam a ser utilizados em várias aplicações, tais como carros, calculadoras, relógios digitais, computadores, entre outros. Em 1985, a Intel introduz o primeiro microprocessador de 32bits, o 80286, sendo cada vez mais evoluídos e potenciando cada vez a sua aplicabilidade.

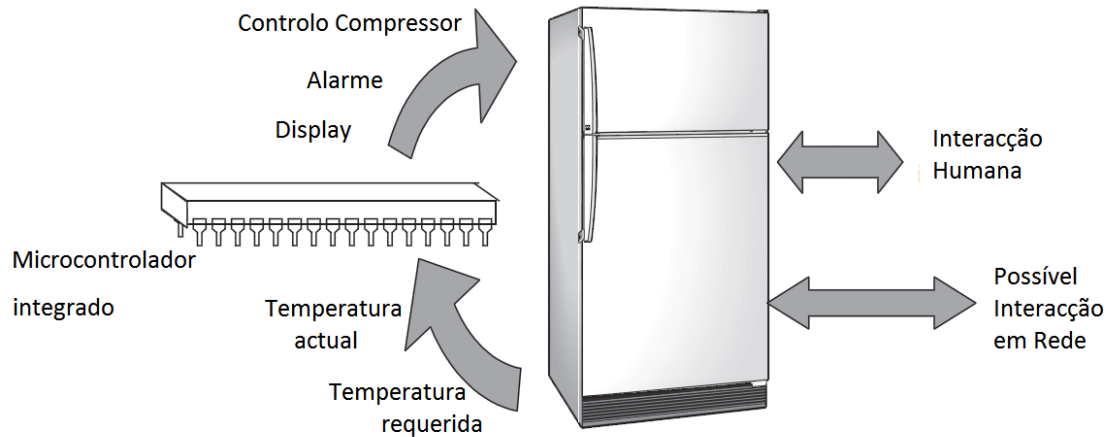


Figura 3. 7 - Sistema com microcontrolador integrado

À medida que as pessoas rapidamente reconheciam e exploravam o poder computacional dos microprocessadores, aperceberam-se que eles podiam ser utilizados com outro fim, o controlo. Os projectistas começaram a colocar microprocessadores em diversos produtos, figura 3.7, que nada tinham relacionado com computação, como o frigorífico, o fogão, brinquedos, e muito mais.

É sabido que “a necessidade destas aplicações não era o alto poder computacional, nem quantidade elevada de memória ou rapidez. Daqui começaram a surgir uma família de microprocessadores dedicada ao controlo, chamada microcontroladores” (Wilmsrst2007).

Actualmente, “existe uma grande variedade de tipos de microcontroladores e de arquitecturas, funcionalidades e capacidade de processamento” (Vaz, 2010).

Os mais difundidos são os da família PIC, fabricados pela Microchip Technology, que possuem arquitectura Harvard, destacando-se da arquitectura Princeton por ter duas memórias independentes, uma para dados e outra para o programa.

A figura 3.8 demonstra o diagrama de blocos de um dos microcontroladores utilizado, o PIC18F4685.

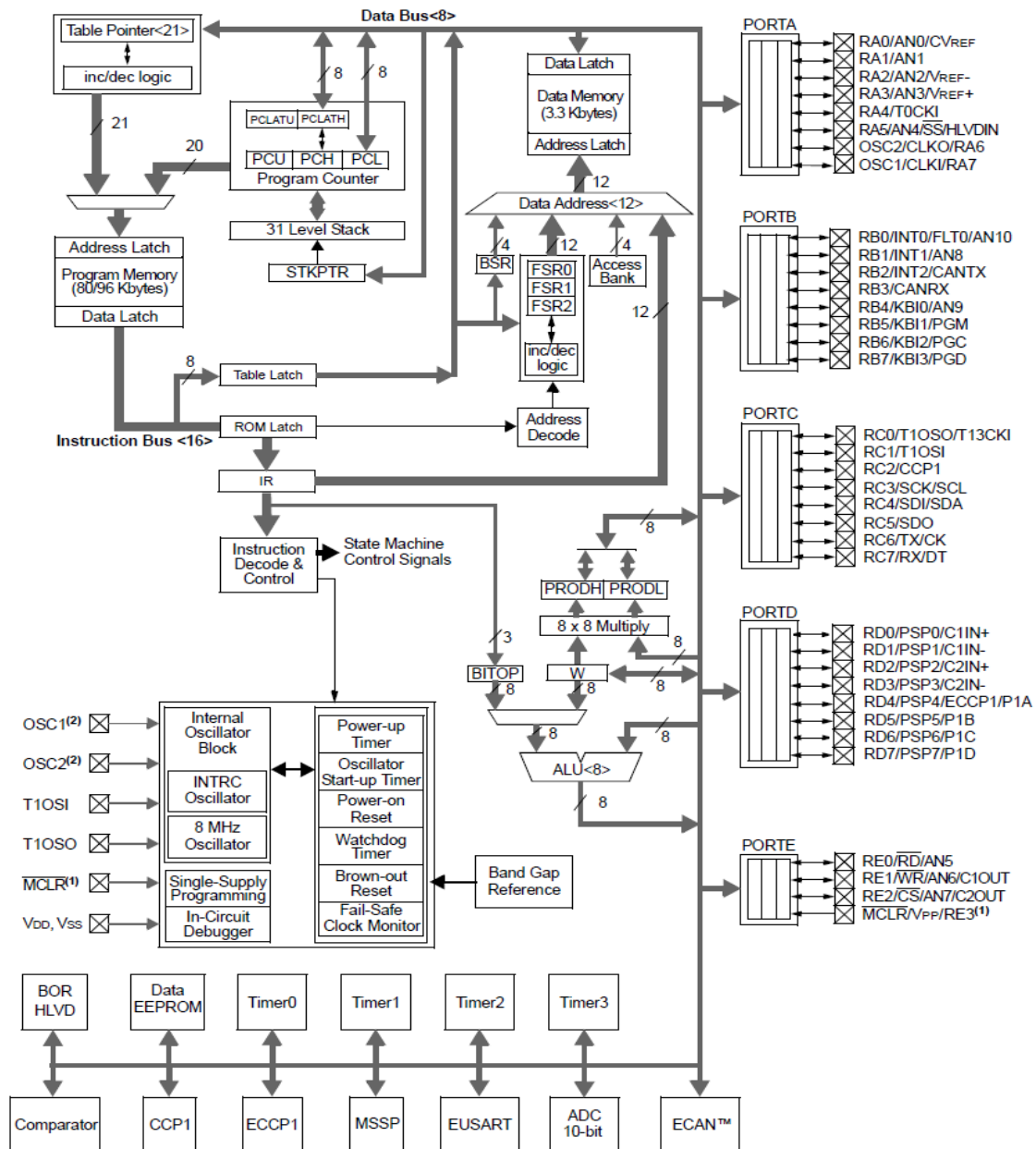


Figura 3. 8 - Diagrama de blocos dos microcontroladores PIC18F4682 e PIC18F4685 (Microchip, 2009”.

Os PIC possuem núcleos de processamento de 8, 16 ou 32bits, e possuem encapsulamentos variados, que variam desde 6 a 100 pinos. Pode-se apreender que “a programação dos PIC pode ser feita com recurso a linguagens de baixo nível como é o caso do Assembly ou em alternativa linguagens de alto nível como por exemplo C, Basic ou Pascal” (Ibrahim, 2010).

Em termos de implementação do código desenvolvido no microcontrolador, é fornecido “um *software* de desenvolvimento com compilador de C integrado, o MPLAB, com o qual se pode criar o programa do PIC. Para transferir o programa para o PIC é necessário um módulo de *hardware* adicional, por exemplo o ICD2 ou PICKit3 comercializados pela Microchip” (Microchip, 2010).

3.3 Programmable Logic Controller

Um controlador lógico programável, PLC, é um tipo especial de controlador baseado em microprocessadores que utiliza uma memória programável para armazenar instruções e implementar funções como lógica, sequencia, tempo, contagem e aritmética, de modo a controlar máquinas e processos.

É sabido que “estes são desenvolvidos para serem operados por técnicos com um conhecimento algo limitado de computadores e linguagem de computação” (Bolton, 2006).

A imagem 3.9 descreve a arquitectura de um PLC.

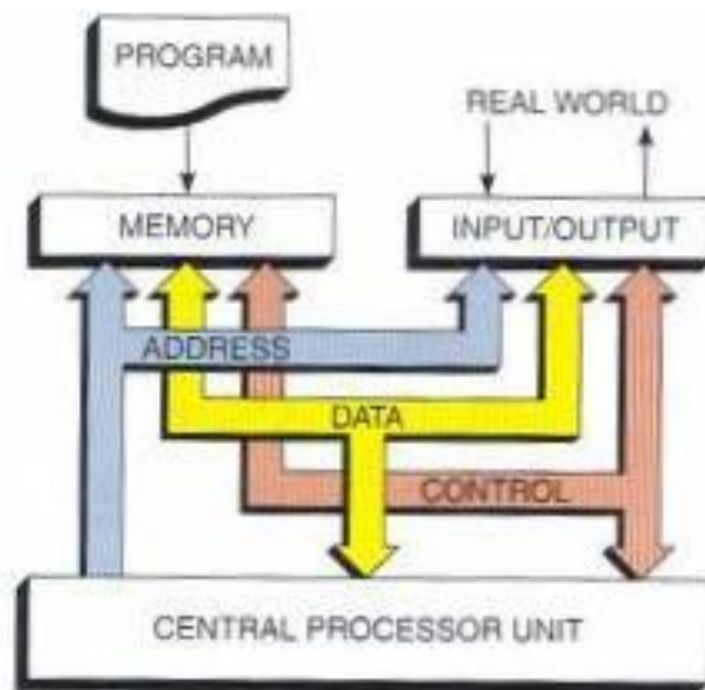


Figura 3. 9 - Arquitectura geral de um PLC (MME, 2006).

A engenharia de controlo tem evoluído ao longo do tempo. No passado, os humanos eram o principal modo para controlar um sistema. Mais recentemente, a electricidade tem sido usada para controlo e desde cedo, o controlo eléctrico foi baseado em relés. É conhecido que “o desenvolvimento de computadores de baixo custo trouxe a revolução, os PLC, a partir da década de setenta, e tornou-se a escolha mais comum para o controlo de produção (Jack, 2005).

Tipicamente “um sistema PLC tem os componentes funcionais básicos de uma unidade de processamento, nomeadamente memória, fonte de alimentação, interface de entradas e saídas, E/S, módulo de comunicações e o dispositivo de programação do PLC” [Bolton, 2006], além de poder ter ranhuras para módulos de expansão. A figura 3.10 demonstra esta organização num sistema controlado por PLC.

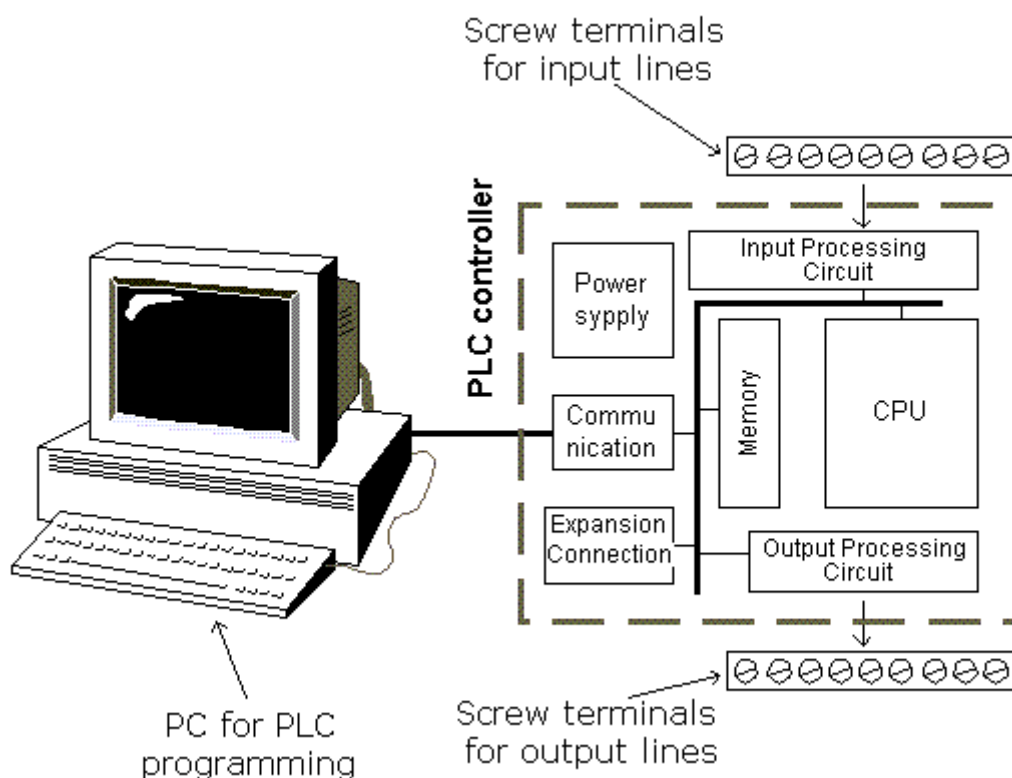


Figura 3. 10 - Componentes físicos de um sistema controlado por PLC (Albertjonas, 2010).

A unidade de processamento ou unidade central de processamento, CPU, contém o microprocessador, sendo que este é responsável por interpretar os sinais de entrada, activar as saídas de acordo com o programa armazenado na sua memória.

A unidade de fornecimento energético é necessária para converter a alimentação alternada para correntes contínuas de baixa voltagem, necessária para o processador e para as entradas e saídas controladas.

O dispositivo de programação é utilizado para introduzir o programa desenvolvido neste dispositivo para a unidade de memória do PLC.

A unidade de memória, tal como o nome indica, é onde se armazena o programa para que possa ser utilizado pelo microprocessador para o controlo e onde salva a informação recebida das entradas para processamento.

As interfaces E/S são responsáveis pela recepção de informação proveniente de aparelhos externos, tais como sensores de temperatura ou células foto-eléctricas, e envio de sinais para dispositivos externos, que podem ser electroválvulas ou bobines de arranque de motores, entre outros.

A interface de comunicações é utilizada para receber e enviar dados pela rede de comunicações de ou para PLCs remotos.

Capítulo 4

Solução Proposta

Neste capítulo faz-se a descrição do protótipo desenvolvido. Inicialmente é apresentada a arquitectura da solução implementada, na qual se explica a interacção entre os diversos elementos constituintes e as estrutura das mensagens utilizadas para comunicação.

De seguida, expõe-se a interface física do protótipo, discutindo-se as escolhas do dispositivo de controlo e o seu modo de operação. Depois de se efectuar a análise ao controlador, explica-se o funcionamento do módulo de comunicação do sistema, responsável por ligar por comunicação série e por TCP/IP todo o sistema.

Após este subcapítulo, são descritas as aplicações desenvolvidas, quer na plataforma SCADA, quer no programa Visual Basic.

4.1 Arquitectura da solução proposta

Para este projecto desenvolveu-se uma arquitectura de um sistema, demonstrada na figura 4.1. Esta procura descrever numa forma bastante abreviada o funcionamento da arquitectura em questão, bem como a interacção entre os seus módulos.



Figura 4. 1- Arquitectura geral do sistema

Este sistema é composto pela plataforma Movicon, que garante a monitorização, a supervisão e a aquisição de dados para o utilizador, por uma base de dados em Microsoft™ Access, não visível na figura 4.1. Esta base de dados é gerida simultaneamente pela plataforma SCADA e por uma interface de comunicação desenvolvida utilizando o software Microsoft™ Visual Basic, que além de comunicar com o sistema SCADA, comunica por TCP-IP com um servidor. Este interage por comunicação série com o controlador da linha de transporte circular existente no Laboratório de Sistemas Flexíveis de Produção, que pretende simular uma linha de produção de uma fábrica.

Em termos de comunicação entre a interface gráfica e a plataforma de comunicação em Visual Basic, pode-se afirmar que estas interagem indirectamente através da base de dados criada através do software Access da Microsoft. Assim, estabelecem uma ligação a esta através da conectividade ODBC por um ficheiro DSN de utilizador, User DSN. As duas aplicações aqui referidas estão permanentemente ligadas à base de dados mencionada anteriormente, a qual consultam periodicamente ou escrevem assim que chega uma instrução para tal.

A tabela 4.1 demonstra a base de dados utilizada para o presente estudo. Esta foi criada automaticamente pela aplicação Movicon, onde se encontram as variáveis definidas para partilha em tempo real.

Tabela 4.4 – Excerto de variáveis e seus valores utilizados neste sistema

Name	Val	MinVal	MaxVal	AveVal
barr_dest	3			
barr1	false			
barr2	false			
barr3	false			
barr4	false			
barr5	false			
barr6	false			
cexe_	false			
pal_mov	1			
Palete1	0			
Palete2	0			
Palete3	0			
Palete4	0			
Palete5	0			
Palete6	0			
leit_barr	b000000n			
leit paletes	p000000n			

A ligação remota da aplicação em Visual Basic com o servidor da linha é efectuada através do protocolo TCP/IP. Este possibilita o envio de mensagens com vista a actualizar as posições das paletes e a ocupação de barragens para a interface Movicon, de comando para o controlador e de estado de execução da linha.

Escolheu-se este tipo de ligação devido ao seu baixo custo, alta velocidade e fiabilidade. A fiabilidade da comunicação por Ethernet usando TCP-IP deve-se à característica *three-way handshake*, à confirmação do envio e recepção dos dados e à capacidade de reenvio, em caso de falha. O endereço IP do servidor foi configurado como 169.254.1.1 e o porto utilizado é 9760.

Em relação à comunicação do servidor para o controlador de linha, deu-se preferência à comunicação série devido ao seu baixo custo, fiabilidade e sobretudo porque o microcontrolador seleccionado, assim como a maioria dos microcontroladores, incorpora uma interface de comunicação série e tem dois pinos dedicados. Portanto, a ligação física do servidor foi configurada com velocidade de 9600bit/s, oito bits de dados, paridade nula, sem stop bit e nem controlo de fluxo.

A figura 4.2 descreve a interacção entre os vários componentes desta arquitectura, os protocolos utilizados e alguns exemplos de mensagens trocadas entre os componentes.

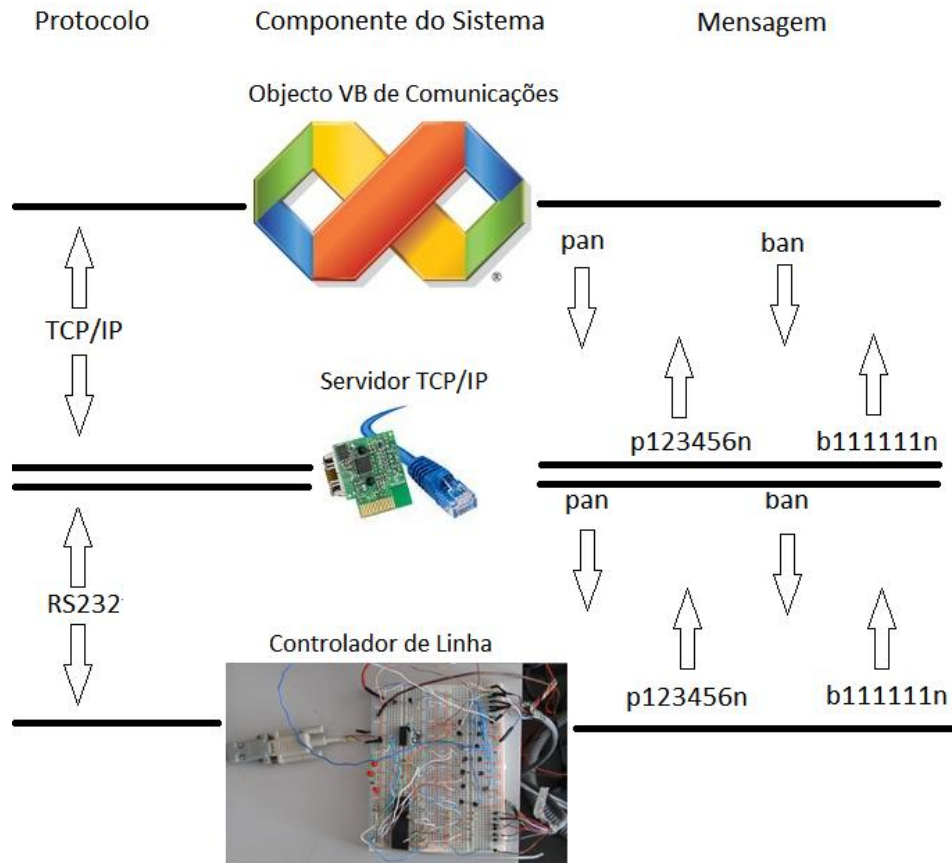


Figura 4. 2 - Mensagens e protocolos utilizados nesta arquitectura

As mensagens enviadas entre os vários módulos desta plataforma são iguais entre os diferentes módulos, apenas variando no tipo de ligação física sob a qual são enviadas, como está ilustrado na figura 4.2.

De seguida, descrevem-se os vários tipos de mensagens transmitidas. Estas são de dimensão variável, dependendo do tipo de mensagem e do transmissor. O quadro que se segue ilustra uma mensagem típica de actualização da posição das paletes na linha, enviada pelo servidor.

Tabela 4.5 - Mensagem de resposta com actualização da posição das paletes

Tipo de mensagem	Palete 1	Palete 2	Palete 3	Palete 4	Palete 5	Palete 6	Fim de mensagem
p	1-6	1-6	1-6	1-6	1-6	1-6	n

O primeiro caracter indica o tipo da mensagem, neste caso uma mensagem sobre paletes dado que começa com o caracter “p”. Os seguintes seis caracteres referem a posição actual da respectiva paleta, isto é, a barragem em que se encontra a paleta daquele índice, sendo que a mensagem é terminada com o caracter “n”.

A posição actual da paleta é calculada por cada barragem, isto é, cada paleta, no fim da distribuição inicial, é numerada com o mesmo valor da barragem em que se encontra e a cada barragem é atribuído um valor, de um a seis, com a indicação de qual é a paleta que se encontra nessa barragem. Este é o ponto de partido para o inicio do cálculo da posição actual. A partir deste momento, entramos no modo de execução, e cada vez que uma paleta passa por uma barragem, a barragem diminui o valor da paleta que lá se encontra, tendo em conta o sentido de rotação e a numeração das barragens estipuladas.

Por exemplo, a mensagem “p345612n” indica que se está a enviar uma mensagem de actualização da posição das paletes, em que a paleta um encontra-se na barragem cinco, a paleta quatro encontra-se na barragem dois, e assim por diante. No instante em que a barragem dois detecta a passagem de mais uma paleta, decrementa o número da paleta que lá se encontra, sendo neste momento a paleta um, e assim por diante.

No que concerne acerca da mensagem de actualização de barragens, o formato é semelhante à mensagem de actualização das paletes, diferenciando-se no primeiro caracter. Neste caso foi escolhida a letra “b”, e no conteúdo dos dados, que varia entre zero e um, consoante o estado de ocupação da barragem, zero se está livre, ou um caso esteja presente uma paleta. O índice da mensagem traduz o número da barragem. A tabela que se segue descreve a mensagem referida.

Tabela 4.6 - Mensagem de resposta com actualização sobre ocupação das barragens

Tipo de mensagem	Barragem 1	Barragem 2	Barragem 3	Barragem 4	Barragem 5	Barragem 6	Fim de mensagem
B	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	n

As mensagens acima indicadas são enviadas em resposta a mensagens de actualização vindas do módulo que se encontra hierarquicamente no nível acima, o servidor TCP/IP. O receptor interpreta o primeiro carácter e responde com a mensagem correspondente. Caso o carácter em questão seja “p”, o módulo receptor responde com a posição actual das paletes. Caso seja “b”, é enviado o array de ocupação das barragens.

O quadro seguinte descreve a estrutura de um pedido de actualização.

Tabela 4.7 – Estrutura de um pedido de actualização

Tipo de mensagem	Actualização	Fim de mensagem
p/b	a	n

Em termos de comando para a linha, a tabela seguinte demonstra uma mensagem de comando. O primeiro campo indica o tipo de mensagem enviada, neste caso, é ocupado pelo carácter “c”, o segundo índice refere qual a barragem de destino para o qual se pretende enviar a paleta escolhida. Esta escolha representa-se no terceiro campo desta mensagem. Qualquer um destes dois campos tem de ser preenchido com caracteres com o valor entre um e seis. O último carácter diz respeito à ordenação e execução da mensagem. Se este não tiver o valor um, o comando não é executado. De seguida, ilustra-se a estrutura da mensagem de comando.

Tabela 4.8 - Estrutura da mensagem de comando

Tipo de mensagem	Barragem de destino	Paleta a mover	Execução de comando	Fim de mensagem
c	1 a 6	1 a 6	0 / 1	n

Relativamente às mensagens de estado de operação da linha, o controlador é responsável por enviar três tipos de mensagens, sendo elas de início de sistema, de fim de distribuição inicial das paletes e de fim de desempenho de um comando. O primeiro estado acontece quando se inicializa o controlador de linha. O estado de fim de distribuição inicial ocorre quando se dá por terminada a distribuição de cada paleta por cada barragem, depois de se ligar o microcontrolador. O último estado sucede assim que termina a execução de um comando. Nesta altura, o controlador da linha envia uma mensagem que notifica os restantes módulos terminou o processo a

executar. A mensagem referida é composta por três campos, descritos na tabela a seguir.

Tabela 4.9 - Mensagem de finalização de comando

Tipo de mensagem	Execução	Fim de mensagem
c	e	n

No que se refere à mensagem de início de sistema, a palavra “start” é enviada entre os módulos, e quando termina o processo de distribuição inicial das paletes, é enviada a palavra “dif”, sempre com o carácter “n” como terminador.

4.2 Hardware Seleccionado

Depois da escolha da arquitectura do sistema, descrita na secção prévia, realizou-se a selecção e desenvolvimento de cada módulo físico da plataforma, tendo em conta o objectivo de criar uma solução robusta e de baixo custo.

Em relação ao microcontrolador para controlo da linha escolheu-se o PIC16F877 com um encapsulamento de 40 pinos, memória de dados EEPROM de duzentos e cinquenta e seis bytes e memória de dados de 368 bytes, três temporizadores, um de 16 bits e os restantes de 8 bits. Dos 40 pinos, alguns estão preparados para outras funcionalidades além de entrada ou saída digital. Existem oito canais de dez bits com conversor Analógico para Digital, ADC, duas funções de captura, comparação ou *Pulse Wide Modulation*, PWM, portas de comunicação série *Universal Synchronous Asynchronous Receiver Transmitter*, USART, *Serial Peripheral Interface*, SPI, ou *Inter-Integrated Circuit*, I²C e utilizou-se um cristal de frequência 4MHz.

Os sensores utilizados na linha são cilíndricos indutivos de proximidade da Omron, modelo E2E-X3D1-N. Estes sensores detectam objectos ferrosos até 2.4mm e funcionam com tensões a partir dos 12V até 24V. Esta tensão é demasiado alta e não é suportada pelo microcontrolador, que suporta no máximo tensões de 5.5V, aproximadamente, nos seus pinos. Por isso, para se poder adquirir o sinal, utilizou-se um divisor resistivo, com duas resistências de igual valor, 1KΩ, conseguindo fazer

decair a tensão para metade em cada resistência e adquirindo esse sinal. Este sistema está demonstrado na figura 4.3.

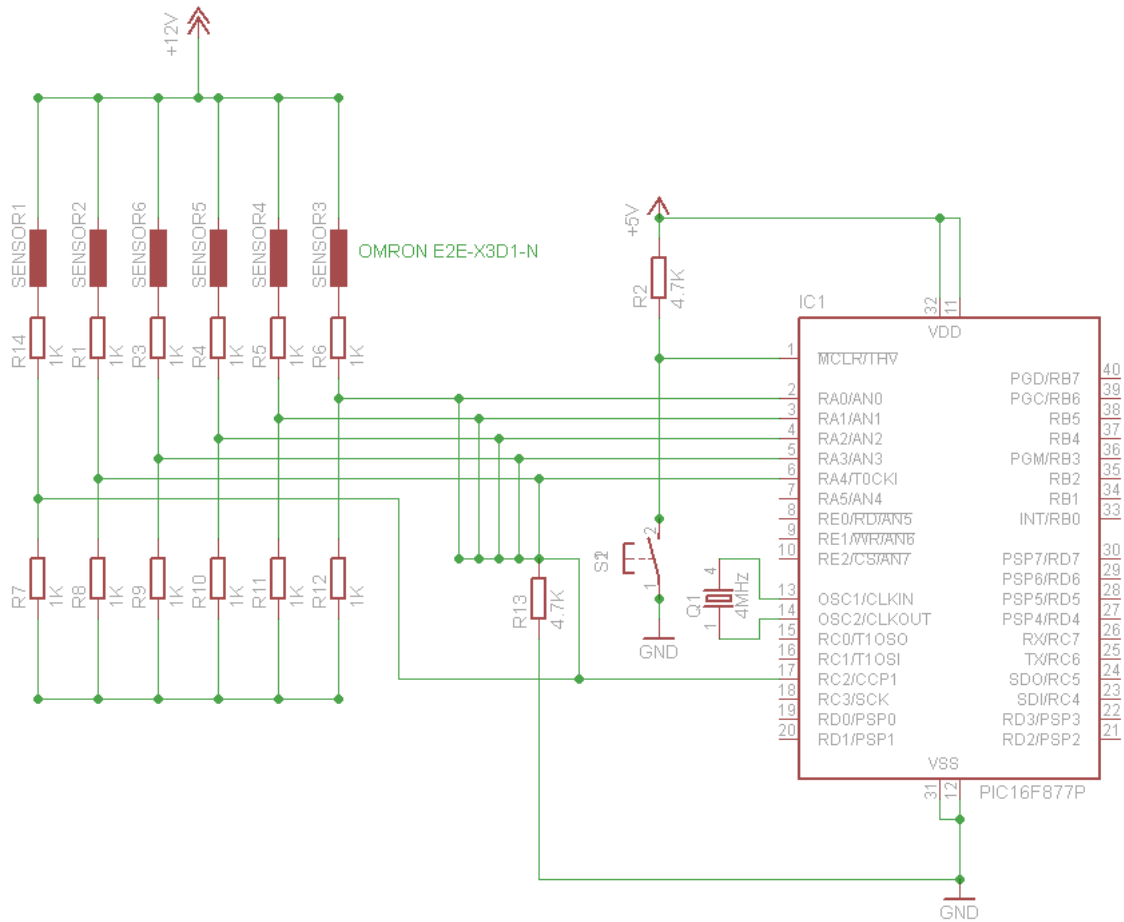


Figura 4. 3 - Esquema eléctrico das entradas do controlador

No que concerne aos actuadores, foram utilizadas electroválvulas, que funcionam com tensões a partir de 24V. Como as saídas do microcontrolador só permitem emitir tensões até 5.5V, projectou-se um sistema de interruptores com transístores, que activam as electroválvulas quando o sinal do microcontrolador está a 5V. Assim, foram utilizados transístores NPN e PNP, nomeadamente C547B e C557C. Foram ainda utilizadas resistências de 10kΩ para fazer decair a tensão na base dos transístores para estes não entrarem em saturação. Em anexo encontram-se mais informações técnicas sobre os transístores utilizados.

As electroválvulas utilizadas são do tipo 2/2, isto é, de duas vias e duas posições, com retorno por mola, à excepção da primeira barragem, em que a sua electroválvula não possui retorno por mola e tem um mecanismo de elevação.

No que refere ao movimento do tapete, existem quatro motores trifásicos assíncronos responsáveis pelo movimento, da marca Bonfiglioli, modelo T56B, com 0.09kW de potência. Em anexo inclui-se mais detalhes específicos acerca do motor referido.

O esquema eléctrico ilustrado na figura 4.4 demonstra o que foi implementado para controlar as electroválvulas da linha.

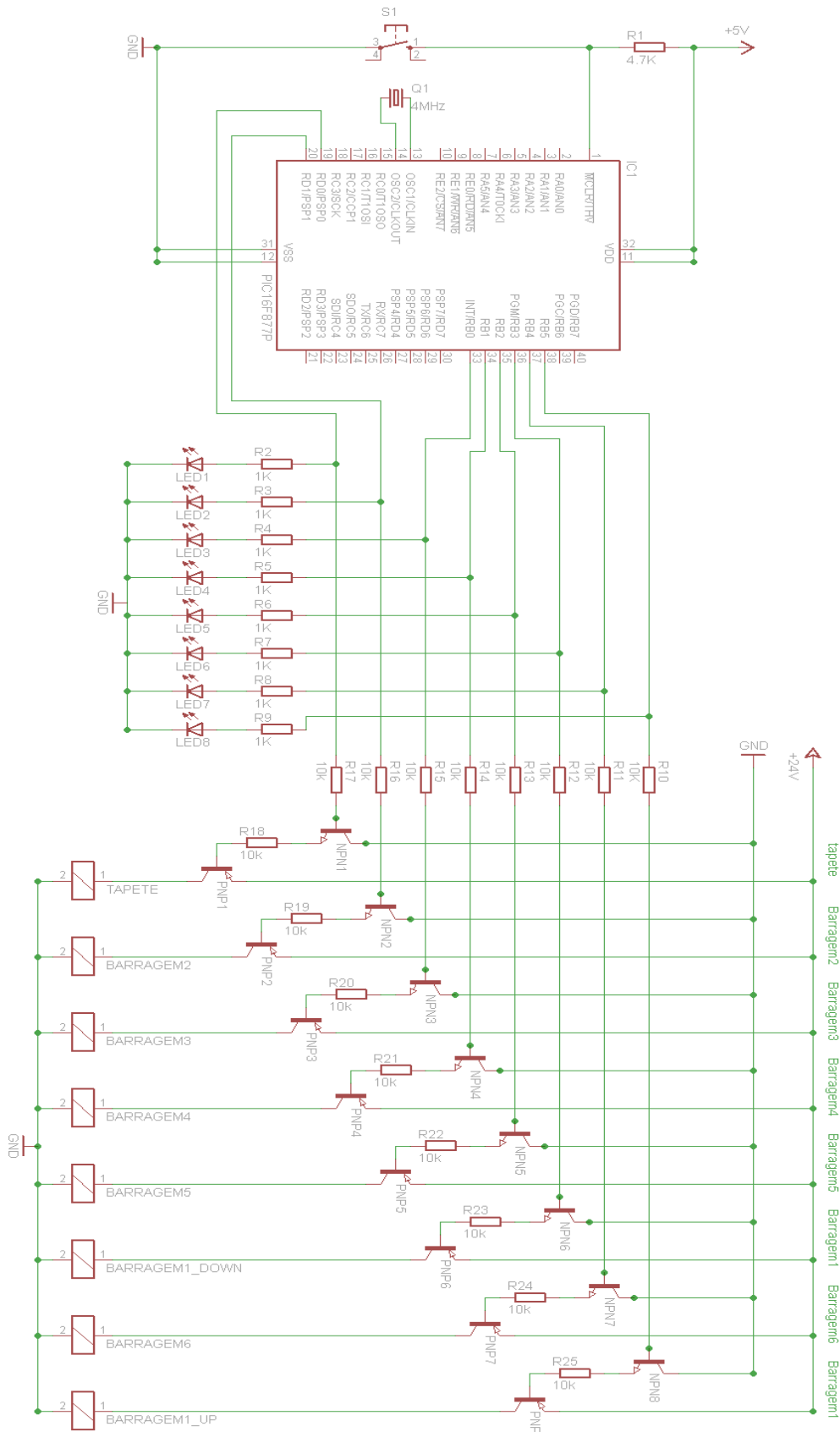


Figura 4. 4 - Esquema eléctrico das saídas utilizadas no controlo da linha

Relativamente ao microcontrolador utilizado como servidor, seleccionou-se o PIC18F4685, também com um encapsulamento de 40 pinos. Deu-se preferência a este microcontrolador devido à sua maior capacidade de memória, com 96KBytes de memória de programa, onde se pôde implementar os protocolos de comunicação por TCP-IP e comunicação série. Em termos de memória de dados, armazena até 3328 bytes de memória de dados e 1024 de bytes na EEPROM. Possui onze canais de ADC de dez bits, dois CCP, possibilita SPI, Master I²C, EUSART, um temporizador de oito bits e três de dezasseis bits. A figura 4.5 ilustra o microcontrolador utilizado, incorporado na placa PICDEM 2 Plus 2006 da Microchip.

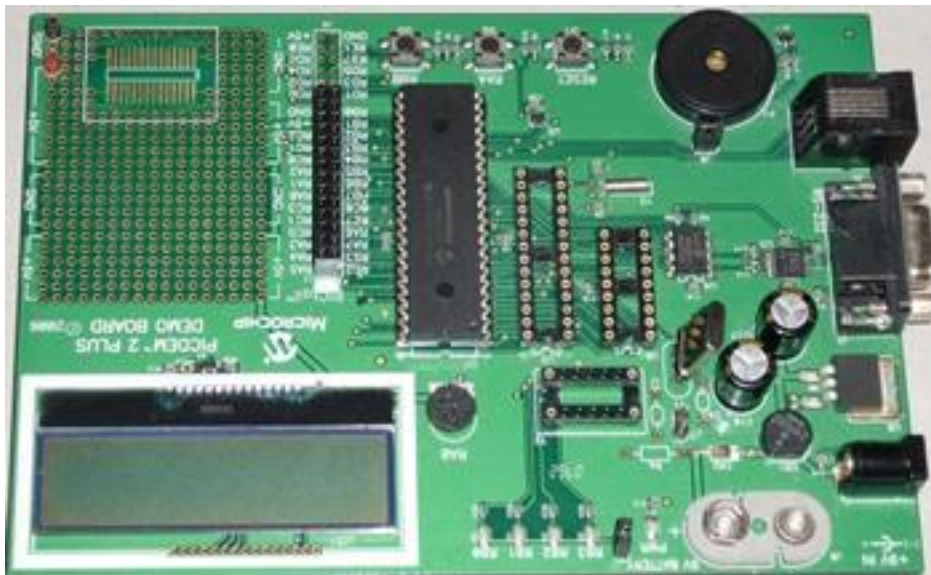


Figura 4. 5 - Placa Microchip PICDEM 2 Plus com PIC18F4685

Em termos de hardware para suportar a comunicação Ethernet usando o PIC18f4685, preferiu-se a placa da Microchip™ PICtail Ethernet. Esta possui um controlador Ethernet ENC28J60 que liga ao conector fêmea RJ-45, assegurando uma ligação Ethernet de 10 Base-T.

Nesta configuração, as aplicações a que se destina apenas necessitam de enviar alguns bytes e, como tal, não requerem largura de banda elevada, sendo esta placa o componente físico indicado. O controlador é compatível com o standard IEE 802.3 que especifica a camada física e a sub-camada MAC da camada de ligação de dados do modelo OSI.

Este controlador tem MAC integrado, possui um buffer de transmissão e recepção de 8KByte e utiliza a interface SPI do microcontrolador onde se conecta. Em anexo inclui-se o esquema eléctrico desta placa Ethernet da Microchip. A figura 4.6 representa a placa Ethernet utilizada.



Figura 4.6 - Placa Microchip Ethernet PICtail Daughter Board

4.3 Código desenvolvido

Depois de estudado o funcionamento dos diversos componentes que envolvem a parte física do sistema de controlo da linha de transferência circular e verificado todas as conexões, iniciou-se o desenvolvimento do software a implementar em ambos os microcontroladores.

Os códigos foram desenvolvidos em linguagem C recorrendo ao software MPLAB IDE versão 8.76, sendo este fornecido pela Microchip. O código do controlador foi compilado através do compilador de C Hi-Tech C *Compiler*, versão 9.82, enquanto que a compilação do código do servidor se fez utilizando o compilador MPLAB C *Compiler for PIC18*, versão 3.40.

Os subcapítulos que se seguem descrevem minuciosamente o funcionamento do hardware utilizado.

4.3.1 Funcionamento do Controlador de Linha

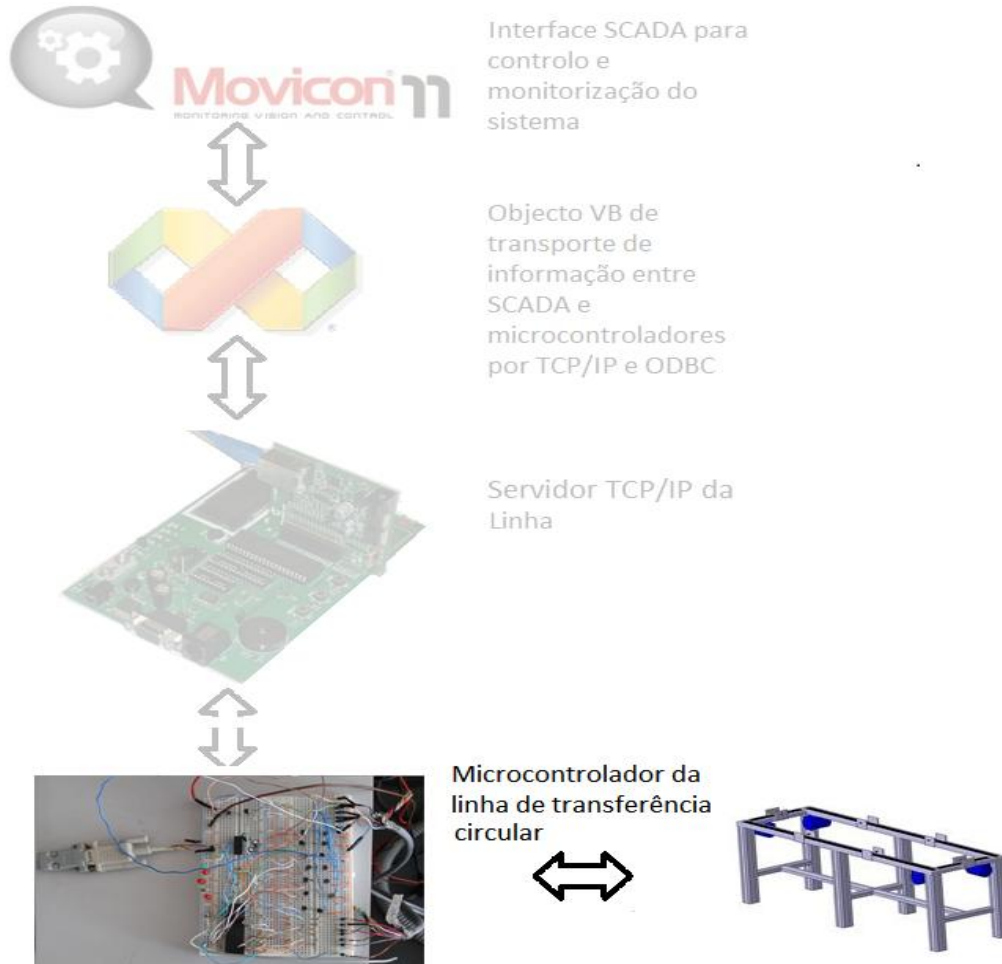


Figura 4. 7 - Esquema geral, destaque ao controlador de linha

O controlador de linha encontra-se na base da hierarquia desta plataforma implementada, como ilustra a figura 4.7. Este inicialmente começa por ler alguns parâmetros de configuração, seguidamente cria as novas variáveis e por fim estrutura as mensagens pelos seus devidos campos.

Além disso, define os pinos dedicados ao controlo e monitorização da linha, colocando os valores $0x1F$, $0x00$, $0x00$ e $0x00$ nos Portos A, B, D e E, respectivamente, e o valor um no bit TRISC2. Tendo isto em conta, definiu-se assim as entradas e saídas do microcontrolador que foram utilizados neste programa, sendo que os 5 primeiros bits

do Porto A e o bit dois do Porto C serão utilizados como entradas e os restantes referidos acima serão configurados como saídas.

Relativamente a operações que envolvam tempo, foram utilizados os temporizadores um e dois. O primeiro é um temporizador de 16bits, consistindo em dois registos de oito bits, e nele foi definido que utilizaria o relógio do microcontrolador interno, isto é, dividindo-se por quatro a frequência do oscilador, com um *prescaler* de 1:8 e activando-se a sua interrupção. O temporizador dois possui um registo de oito bits, com *prescaler* e *postscaler*, definidos de 1:16 e 1:8, respectivamente.

Na figura 4.8, demonstra-se uma vista de pormenor da parte física do controlador da linha de transferência circular.

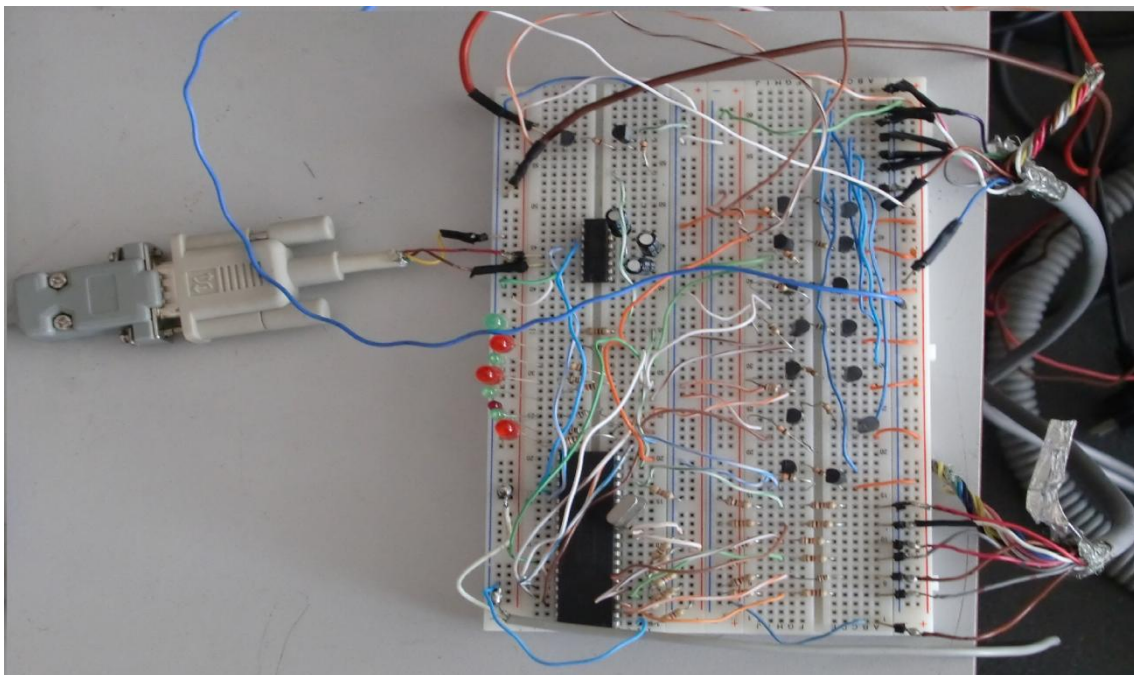


Figura 4. 8 - Vista de pormenor do controlador de linha desenvolvido

De seguida, definiu-se todos os parâmetros da comunicação série RS232, permissão de interrupções, portas de entradas e saída, temporizadores e desactivou-se algumas funcionalidades de pinos que vamos utilizar apenas como saídas ou entradas digitais.

O fluxograma 4.9 descreve o funcionamento geral do controlador de linha, dando ênfase especial à parte de processamento de interrupções, nomeadamente dos temporizadores e da recepção série.

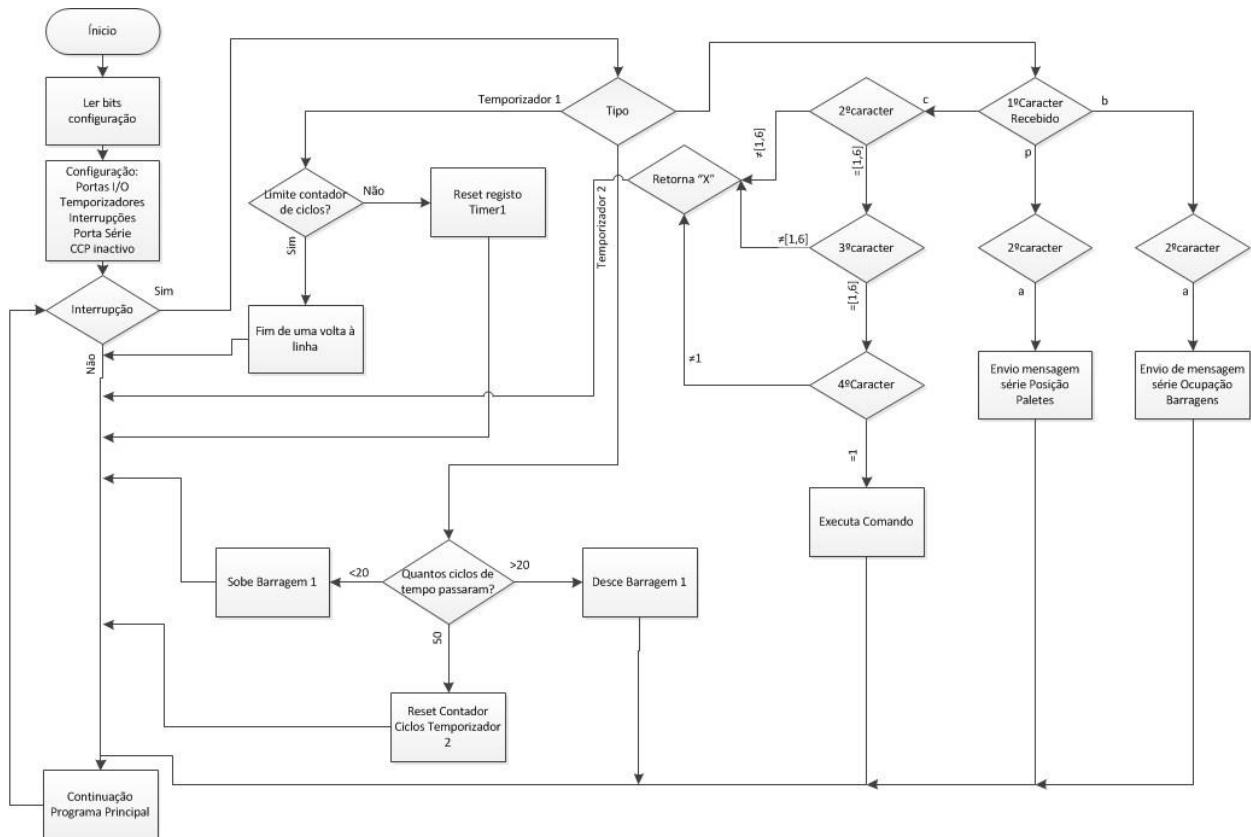


Figura 4.9 - Fluxograma do funcionamento geral do controlador de linha

Depois do passo anterior, o programa inicia um ciclo infinito onde o programa vai gerir o comportamento da linha passo a passo e também reagir às interrupções, quer de mensagem série, quer de temporizador. Inicialmente é enviada a mensagem série “start” notificando o servidor que o controlador está a inicializar e passa-se para o passo seguinte. Caso o servidor não receba a mensagem, o controlador é notificado e reenvia a mensagem.

Durante a etapa seguinte começamos a fase de distribuição inicial das paletes pelas barragens. Inicia-se um temporizador de 16bits e abrem-se todas as barragens, excepto a primeira e acciona-se o tapete, activando assim a saída RD0 do microcontrolador. O objectivo passa por acumular todas as paletes atrás da primeira

barragem. Após trinta e cinco segundos, as paletes encontram-se atrás da primeira barragem e procede-se ao próximo passo. De seguida, abre-se a primeira barragem e deixam-se passar cinco paletes, estando as restantes barragens fechadas. À sexta palete, a barragem um fecha-se e passamos à próxima fase.

A entrada do microcontrolador que recebe o sinal do sensor da barragem um é o pino RC2 e as saídas que controlam a subida e descida desta barragem são os pinos RB3 e RB5. Esta barragem é diferente das restantes porque possui um sistema de elevação, accionado pela sua electroválvula.

Com o sistema descrito no parágrafo anterior, para se ordenar a subida da barragem tem de se activar o pino 5 do porto B e desactivar a saída RB3. Para se para efectuar o movimento contrário, tem de se colocar a saída RB5 a zero e activar o pino RB3.

Na fase seguinte, a segunda barragem deixa passar quatro paletes e fecha-se quando chega a quinta palete. No final deste percurso passamos ao passo seguinte. Neste, a quarta barragem abre e deixa passar três paletes e à quarta fecha.

A entrada do microcontrolador responsável por verificar a presença de paletes na segunda barragem é a entrada RA4 e na terceira barragem é RA0, enquanto que a saída encarregada de ordenar a abertura e fecho da barragem é RD1 e na terceira barragem é o pino RB0. Assim, restam distribuir três paletes por três barragens.

Os três passos seguintes são destinados a distribuir as restantes paletes pelas barragens, utilizando o processo anterior.

Os pinos do microcontrolador responsáveis pelos *inputs* das barragens quatro, cinco e seis são, respectivamente, RA1, RA2, e RA3. As saídas do microcontrolador responsáveis pelo controlo das barragens são, por ordem, RB1, RB4 e RB2.

A tabela 4.7 descreve, de uma forma mais explícita a relação barragens, entradas e saídas do microcontrolador.

Tabela 4.10 - Entradas e saídas das respectivas barragens

Barragem	Entrada	Saída
1	RC2	RB5/RB3
2	RA4	RD1
3	RA0	RB0
4	RA1	RB1
5	RA2	RB4
6	RA3	RB2

Alguns dos pinos apresentados na tabela requereram alterações nas suas propriedades. Estas alterações desactivaram as funcionalidades que estão activas por defeito. Para tal, os pinos do Porto A, aqui utilizados como entradas digitais, necessitaram que se desactivasse a sua funcionalidade analógica, colocando o valor 0110 nos registos PCFG3:PCFG0. Neste sentido, é necessário desactivar a interrupção externa do pino RB0, colocando a zero o bit INTE. Outros pinos que necessitaram de configuração prévia são RB3, cuja funcionalidade inicial é de programação em baixa voltagem, e que se tem de colocar a zero, RC2 que possibilita a função *Capture 2/ Compare 2/ PWM2* como entrada digital.

A figura 4.10 descreve resumidamente o modo de funcionamento da secção de distribuição inicial das paletes pela linha, dentro do ciclo do programa principal.

À medida que esta fase de distribuição inicial decorre, faz-se a actualização da ocupação das barragens. A actualização das paletes ainda não é efectuada nesta altura dado que as paletes não estão distribuídas.

Quando se finaliza a distribuição inicial, o controlador notifica o servidor desse facto, por comunicação série, com a mensagem “difn”, enviando além desta, duas mensagens, uma com a ocupação das barragens, outra com a posição destas, que neste instante, encontram-se distribuídas pela equação 1.

$$barragem[n] = palete[n], \text{ com } n \in [1, 6] \quad (1)$$

A partir deste momento, entramos na fase de controlo, em que o operador coordena as deslocações da paleta pretendida para a barragem a destinar através da interface Movicon.

Convém salientar que o comando de uma paleta provoca a movimentação das restantes de modo a garantir sempre a presença de uma paleta por barragem.

Durante esta fase, o controlador de linha aguarda por uma mensagem série de comando do servidor. Assim que esta é recebida, verifica qual a fase em que o controlador se encontra, isto é, se se encontra ocupado a processar algum comando ou se ainda está na fase de distribuição inicial.

Após esta verificação, o microcontrolador copia a mensagem para um buffer dedicado à mensagem de comando e interpreta os caracteres recebidos, confirmando se o valor recebido nos segundo e terceiro campos da mensagem são inteiros, entre um e seis, e se o quarto índice tem valor um e, em caso positivo, processa a mensagem para as variáveis dedicadas. De seguida, efectua o cálculo do número de passagens que as paletes terão de fazer por uma determinada barragem até estar terminado o comando, utilizando a equação 2.

$$conta = barr_dest - paleta_mover + 1 \quad (2)$$

Na equação 2, é necessário somar mais um devido ao facto do microcontrolador contar logo como uma passagem quando inicia o ciclo de comando. Após o cálculo,

compara-se o valor do resultado, pois se este for menor que 0 tem que se adicionar seis. Por exemplo, quando se movimenta a paleta seis para a barragem de destino dois, tem de se adicionar seis ao resultado. Depois desta operação e se o resultado for maior que zero passa-se à fase seguinte.

Nesta fase, todas as barragens deixam passar paletes e incrementam a variável *pass_barrN*, em que N é o número da barragem incrementada, à medida que as paletes vão passando. Cada barragem compara o valor dessa variável com a variável conta, e quando estes se igualam, a barragem não permite a passagem de mais nenhuma paleta e incrementa outra variável, *contagem_total*. Quando esta é igual a seis, significa que as seis barragens executaram o comando.

Em seguida, passa-se para o passo seguinte, na qual se desliga o tapete e coloca-se a zero as variáveis de comando. Posto isto, enviam-se três mensagens série para o servidor, uma a notificar que o comando foi executado, outra a posição das paletes e, por fim, uma com a ocupação das barragens. Posteriormente, colocam-se a zero os contadores utilizados no passo anterior e passa-se à fase seguinte, onde o microcontrolador fica à espera de outra mensagem de comando.

4.3.2 Funcionamento do Servidor TCP/IP de Linha



Figura 4. 11 - Esquema geral, destacando servidor de linha

O servidor de linha utilizado, que se encontra na penúltima posição da hierarquia do sistema desenvolvido, como se pode verificar pela imagem 4.11, incorpora algumas funcionalidades que existem num pacote de soluções que a Microchip fornece, da versão 2010-10-19. Deste pacote, utiliza-se um servidor Web, adaptado à exigência deste projecto, para comunicação por TCP/IP com a interface de comunicação remota em Visual Basic, e adapta-se o código fonte deste pacote para comunicar serialmente com o controlador de linha. Na figura 4.11 ilustra-se o servidor incorporado neste sistema e a sua posição na plataforma implementada.

Neste subcapítulo vai-se procurar explicar o funcionamento base do pacote de soluções da Microchip, além de todas as alterações e adaptações efectuadas a este. Este pacote inclui uma aplicação que contribui para alterar o código automaticamente, nomeadamente modificando macros, consoante o objectivo que o programador pretende dar ao servidor.

Este programa chama-se *TCPIP Configuration Wizard*, e nele definiu-se a construção de um servidor Web, incluiu-se um exemplo de um servidor TCP genérico, suporte para DHCP para um cliente e servidor, *IP Gleaning*, servidor ICMP e suporte UART. Define-se o nome anfitrião, neste caso é MCHPBOARD, o endereço MAC por defeito, em hexadecimal, é 00:04:A3:00:25:40, em que os três últimos bytes hexadecimais são únicos da placa Ethernet PICtail utilizada. O endereço IP deste servidor é o 169.254.1.1 e a máscara SubNet é 255.255.0.0 e a dimensão do buffer de transmissão e recepção é de 20bytes.

Após estas definições, o microcontrolador carrega-as no seu programa, como macros. No início do programa, o servidor activa o temporizador zero e a sua interrupção, coloca um prescaler de 1:256. De seguida armazena as variáveis relacionadas com a pilha TCP/IP e porta série na estrutura APP_CONFIG, que inclui desde endereços, *flags*, e *strings* de nome *NetBIOS Server Name* ou NBNS, que foram configurados na aplicação *TCPIP Configuration Wizard*. Seguidamente, inicia-se a pilha TCP/IP da Microchip. Nesta fase, o servidor inicializa a sua configuração, iniciando o módulo *Serial Peripheral Interface*, SPI, bem como e todos os registos no controlador ENC28J60. Após este passo, inicializa-se o módulo TCP, em que se define os respectivos buffers de recepção e transmissão desse protocolo na memória e inicializa cada socket no estado fechado. Se a memória alocada for insuficiente para os sockets TCP, a função prende naquele local de modo a ser capturada pelo debugger. Na fase seguinte, o servidor cria as mensagens que vai enviar por comunicação série e por TCP/IP. A partir deste momento, todos os itens foram inicializados, dando início ao ciclo multitarefas cooperante. Este ciclo vai executar continuamente todas as tarefas relacionadas com a pilha, e é aqui que estão integradas parte das adaptações efectuadas.

De seguida, entramos num ciclo infinito, onde primeiro se utiliza uma função de estados finitos que efectua tarefas da pilha, que incluem processamento da chegada

de pacotes, tipo de pacotes, além da chamada da entidade apropriada na pilha para processar esse pacote e também executa operações controladas por tempo. Nesta função, realizam-se periodicamente todas as tarefas por TCP requeridas, tais como recepção, envio e retransmissão, envio de reconhecimento, encerramento da conexão, sincronismo, ligação a *sockets* e processamento de dados recebidos. Após esta fase, realiza-se a tarefa de servidor DHCP, na qual se processam pedidos DHCP e distribuição de endereços IP.

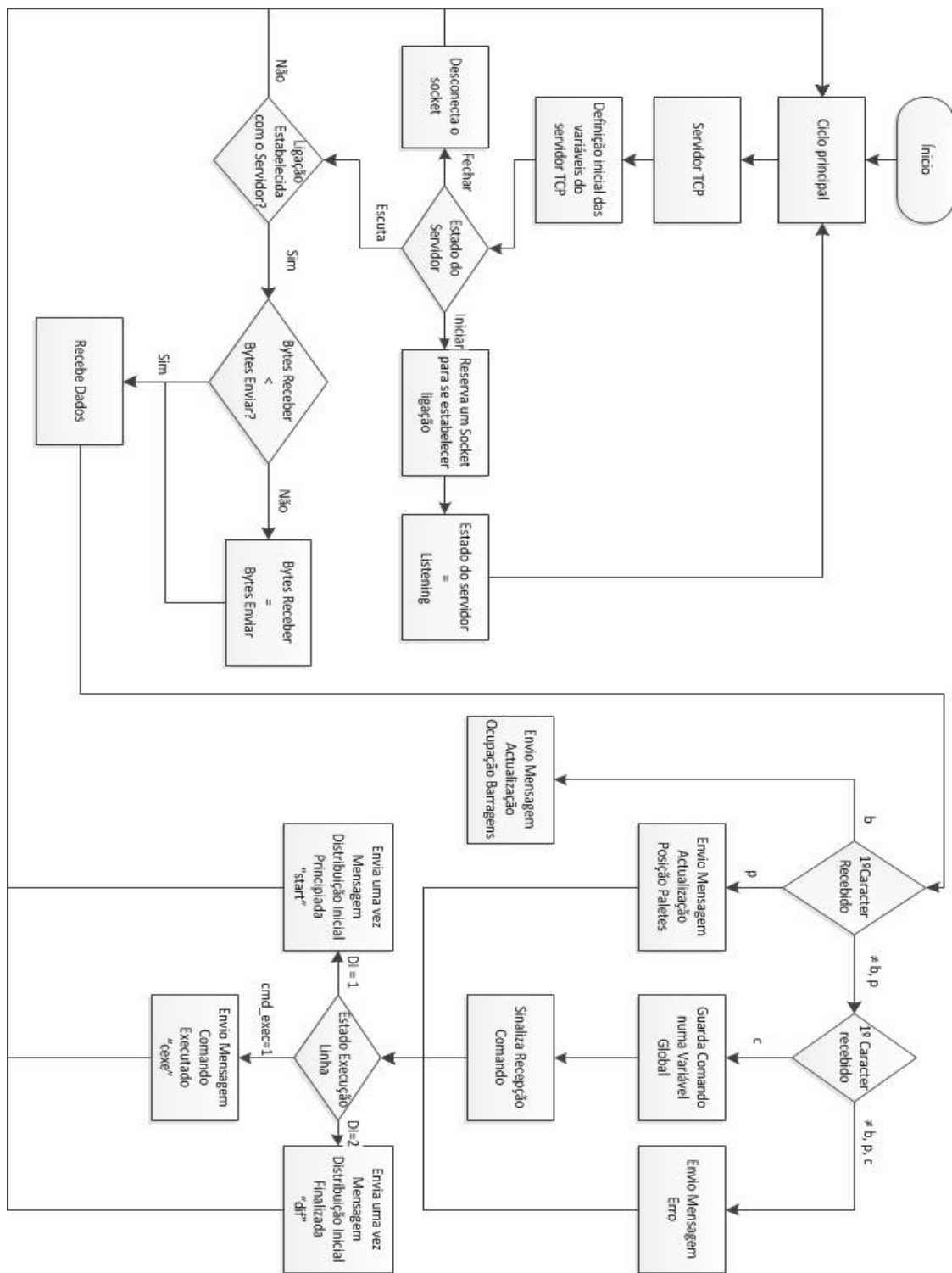


Figura 4. 12- Diagrama de fluxos do servidor TCP.

Após o passo anterior, implementa-se o servidor TCP/IP na porta 9760. A figura 4.12 descreve o modo de funcionamento do servidor.

Este é invocado periodicamente pelo ciclo infinito ficando à espera de ligações nesse momento, alocando um socket para esse propósito. Quando uma ligação é efectuada, o servidor lê todos os dados a receber e transfere-os do buffer TCP RX FIFO, para um buffer local.

A partir daqui executa-se o servidor, onde se interpreta o conteúdo da mensagem recebida. Caso a primeira letra recebida seja um “*b*” ou um “*p*”, reenvia-se uma mensagem com a actualização da ocupação das barragens ou da posição das paletes no controlador, respectivamente.

Se o primeiro character recebido seja um “*c*”, verifica-se se o quarto character é “*1*”, e em caso positivo, copia-se a mensagem recebida para um buffer global. Posteriormente, responde-se para o cliente TCP/IP que a mensagem foi correctamente recebida e activa-se um flag geral que é interpretado no ciclo infinito, onde reencaminha a mensagem recebida para o controlador de linha, passando-lhe a ordem de comando. Caso a mensagem recebida não se inicialize por “*b*”, “*p*” ou “*c*”, reenvia-se uma mensagem de erro para o cliente.

Ainda acerca do servidor TCP/IP, este é responsável por interpretar o estado do controlador, declarado numa variável global, e dependendo do valor desta. Após este passo, é reenviado uma mensagem para o cliente, nomeadamente “*start*”, que descreve o estado de inicio do controlador, ou “*dif*” que notifica o fim da distribuição inicial, e ainda se actualiza o estado da posição das paletes e a ocupação das barragens no controlador.

Se o controlador tiver enviado uma mensagem série que sinalize o fim da execução de um comando, envia-se para o cliente a mensagem “*cexe*” além de se actualizar o estado da posição das paletes e a ocupação das barragens no controlador. No final desta operação, o servidor desconecta-se do cliente.

Voltando ao ciclo principal, se o servidor receber uma mensagem série, activa a flag de recepção pela porta série e copia o que foi recebido e se encontra no registo RCREG para um buffer global. De seguida, é interpretado o primeiro character da mensagem, se for “*b*” ou “*p*” significa que se recebeu uma mensagem de actualização da ocupação das barragens ou da posição actual das paletes e armazena-se esses

valores numa variável específica para o efeito. No entanto, se o primeiro carácter recebido for um “s”, significa que o controlador iniciou o processo de distribuição inicial. Nesse caso, altera-se o valor de uma variável global que vai ser lida na execução do servidor TCP/IP e que envia esse estado para o cliente TCP/IP. No entanto, se o primeiro carácter for “d”, significa que o controlador terminou o processo de distribuição inicial, então essa variável global toma outro valor e é enviada uma mensagem a notificar o cliente TCP/IP dessa alteração.

Ainda interpretando a mensagem série recebida, caso esta inicie por “c”, significa que o comando enviado para execução foi terminado, alterando-se uma variável global que quando for chamada a função do servidor TCP/IP, fará com que seja enviada uma mensagem remota com a notificação de comando executado. Caso se receba o carácter “X” pela porta série, significa que a mensagem de comando não foi bem recebida pelo servidor e é necessário reenviá-la. Se o carácter recebido não for nenhum dos referidos, é enviado o carácter “X” pela porta série para o controlador reenviar a mensagem.

O servidor, no seu ciclo infinito, também envia uma mensagem de actualização para o controlador, na qual alterna o pedido entre actualização da posição das paletes e da ocupação das barragens, de modo a que haja um registo permanente do estado do controlador de linha.

4.4 Software

Para a implementação do módulo de software do sistema desenvolvido, optou-se por um objecto de comunicação cliente por TCP/IP desenvolvido em Visual Basic, utilizando o programa Microsoft Visual Studio 2010, e uma interface de controlo e supervisão utilizando o software Movicon.

A escolha desta arquitectura teve como base o facto de terem sido testados e validados os seus desempenhos em trabalhos anteriores, no que diz respeito ao Visual Basic e aos seus objectos de comunicação por TCP/IP e acesso a base de dados utilizando ligações ODBC. A figura 4.13 ilustra a interacção existente entre os módulos referidos e o restante sistema.

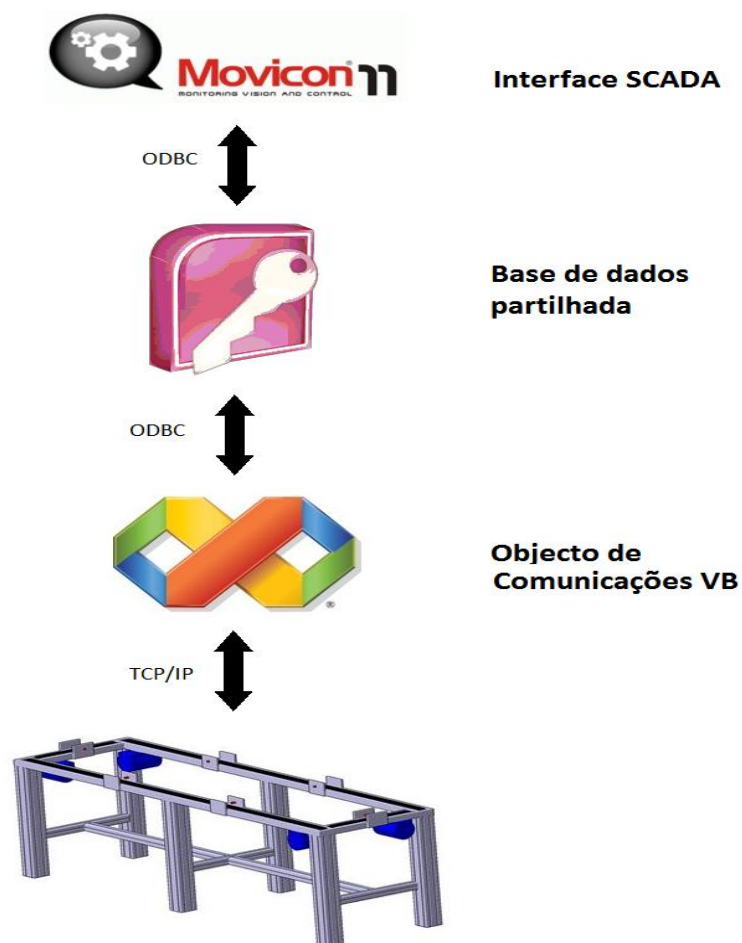


Figura 4. 13 - Relação entre Módulos de Software e sistema físico

É conhecido que “O Visual Basic é uma linguagem de programação dirigida por eventos que evoluiu a partir da linguagem de programação Basic, possui um ambiente de desenvolvimento integrado com uma componente gráfica, o que facilita a criação de interfaces das aplicações” (MSDN, 2010).

É público que “O Microsoft Access é uma ferramenta de gestão de informação que ajuda o utilizador a armazenar informação para referência, consulta e análise. O Microsoft Access pode ajudar a superar as limitações encontradas quando se tenta gerir grandes quantidades de informação” (OpenGate Software, 2011).

4.4.1 Cliente TCP/IP



Figura 4. 14 - Arquitectura geral do sistema, destaque do cliente TCP/IP

Na imagem 4.14 localiza-se o objecto de comunicação VB dentro da estrutura implementada neste estudo.

A figura 4.15 demonstra o funcionamento do objecto de comunicação desenvolvido em Visual Basic.

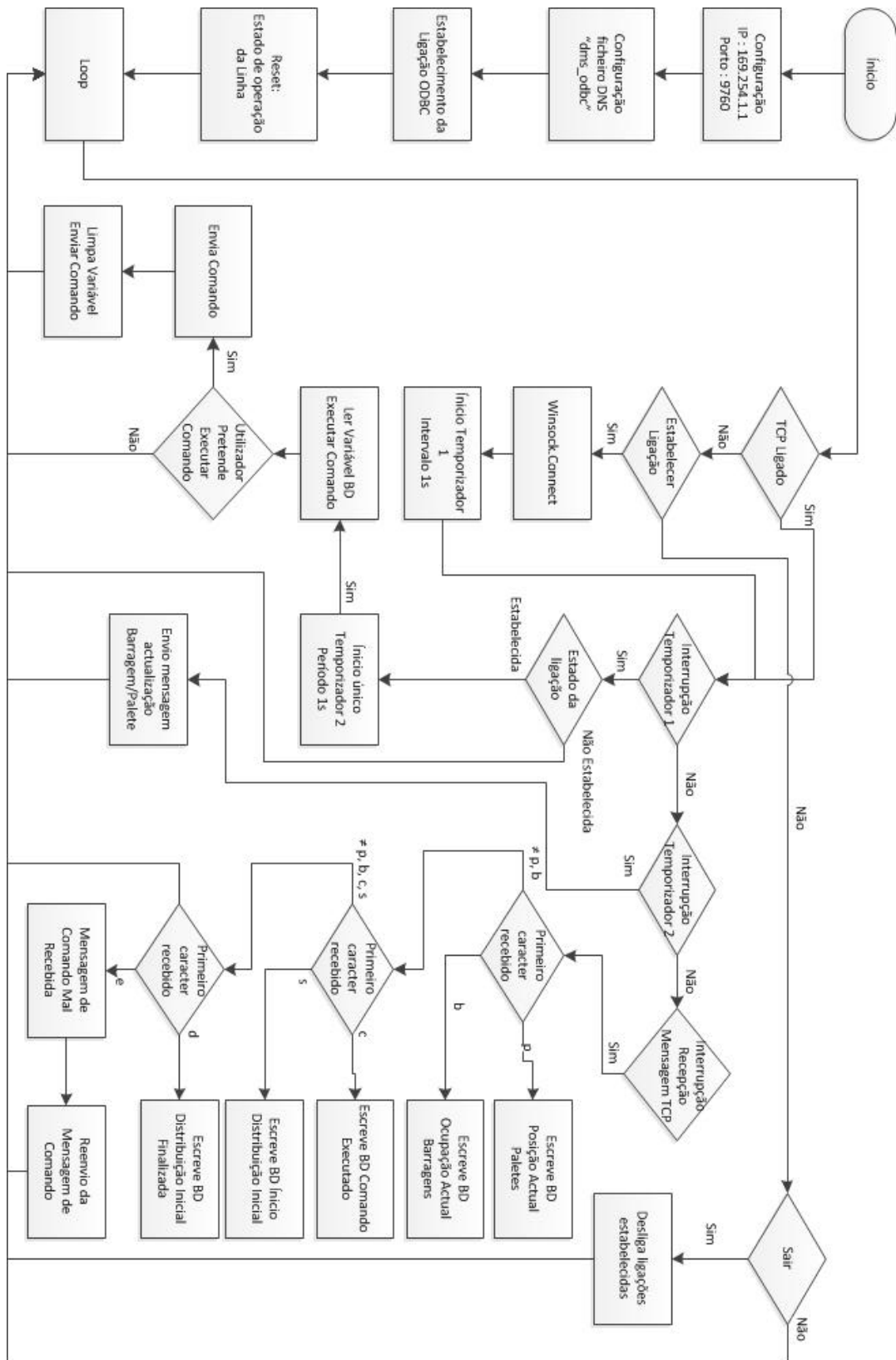


Figura 4. 15 - Diagrama de funcionamento do Cliente TCP/IP

Neste programa, inicialmente é configurado o endereço IP, o porto de acesso ao servidor de linha e a ligação ODBC ao ficheiro DNS “dms_odbc”.

Após este passo, faz-se o reset das variáveis de estado actual do controlador, caso da distribuição inicial, comando executado. De seguida, estabelece-se a ligação ao servidor, no IP 169.254.1.1, porto 9760.

A partir deste momento, a ligação mantém-se e a comunicação com o servidor é efectuada numa base temporal de um segundo. Neste intervalo, ocorre a troca de mensagens de actualização da posição de paletes e da ocupação de barragens.

Ao mesmo tempo, o cliente TCP/IP inicia um outro temporizador que vai ler, de meio em meio segundo, o valor da variável na base de dados que determina o envio de um comando para a linha, mas que só é exequível quando se está no estado de distribuição inicial finalizada. A leitura e escrita das variáveis na base de dados são realizadas através de comandos SQL.

Cada vez que são recebidos dados do servidor, gera um evento neste software, e procede-se à interpretação da mensagem recebida.

A interpretação da mensagem baseia-se no primeiro carácter, e consoante este, grava-se o valor para a variável respectiva da base de dados, excepto quando se recebe uma mensagem de erro do servidor e é necessário reler da base de dados o último comando dado pelo utilizador e reenviá-lo para o servidor.

4.4.2 Interface Movicon

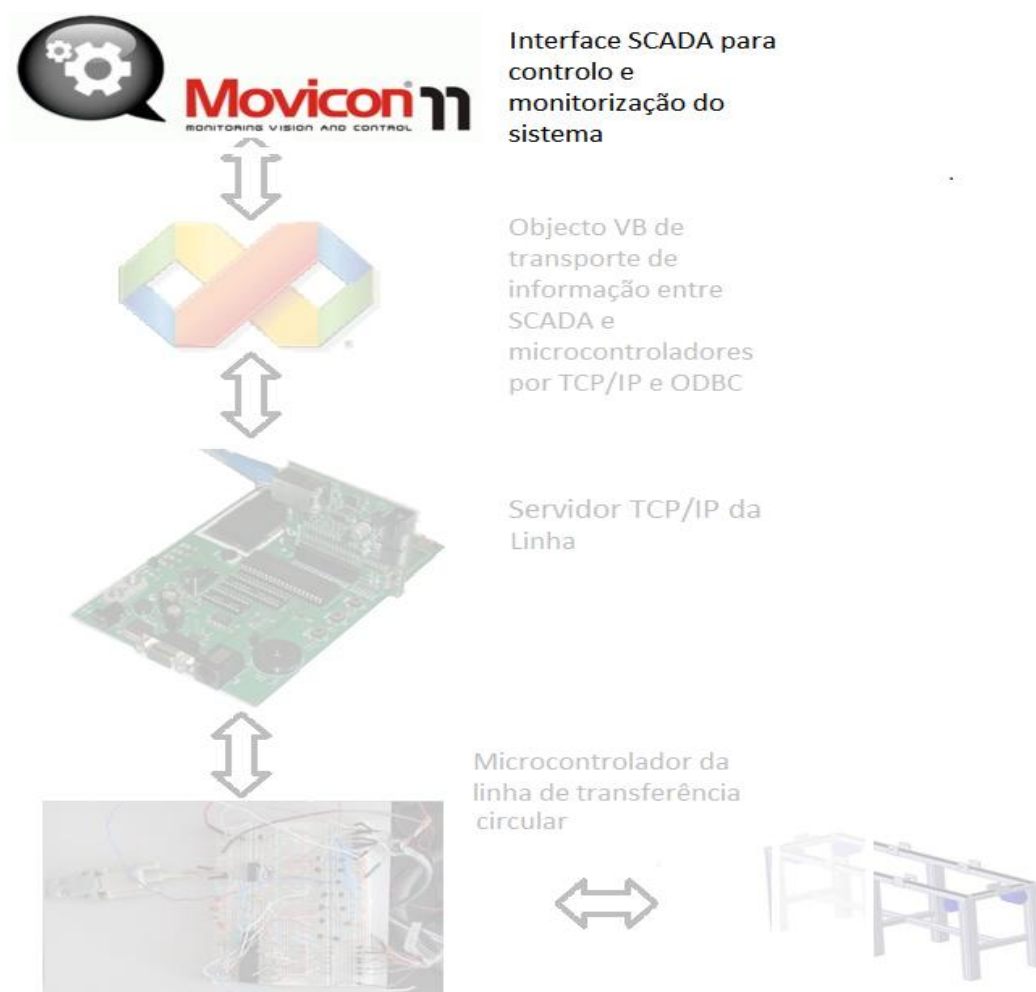


Figura 4. 16 - Diagrama geral de funcionamento, destaque SCADA

De modo a criar uma interface de fácil utilização e compreensão, optou-se por empregar o software SCADA da Movicon devido às já referenciadas características. Esta interface encontra-se na posição cimeira, em termos de acesso do utilizador, da hierarquia implementada. Neste software foram criadas variáveis dentro do programa, do tipo bit, *double* ou *string*.

Para se partilhar as variáveis dentro dos vários ecrãs e dos scripts do Movicon, teve-se de indicar o tipo de área de memória a que pertencem e o seu endereço, tendo em conta que variáveis do tipo bit se endereçam da forma "*inteiro.inteiro*" e *arrays* da forma "*(inteiro(inteiro))*".

No entanto, de modo a que as variáveis fossem partilhadas com outras aplicações, neste caso com a base de dados Access da Microsoft, teve de se configurar vários parâmetros, nomeadamente a sua partilha por ODBC, o tempo em que era necessário ler ou reescrever a variável e o modo de utilização da variável. Isto é, se é de entrada, saída ou ambos.

Na base de dados em tempo real do Movicon, *RTDB*, teve de se definir o ficheiro DSN que permitisse a ligação ODBC, neste caso utilizou-se o ficheiro "*dms_odbc*", e indicar a tabela "*table1*" onde se armazenam as variáveis e os seus valores.

A base de dados, para que pudesse ser acessível pelo Movicon, necessita de se encontrar na directoria raiz do disco rígido, caso contrário, não seria possível editar os campos nem ler as suas variáveis, dado que o Movicon não encontraria a base de dados definida e criaria uma nova, na pasta onde se encontra o projecto.

Em termos de ecrãs da interface, foram criados dois ecrãs com diferentes funcionalidades. O primeiro, e principal, é responsável pela monitorização da posição das paletes, da ocupação das barragens, da identificação do estado de execução da linha e do fim da execução do comando. O segundo apenas é visível quando se pretende dar um comando à linha e, por isso, é bastante simplista e redutor, sendo uma espécie de janela de *pop-up*.

No que diz respeito a scripts, foram criados três scripts gerados por eventos, nomeadamente o fim da execução de um comando, o início da distribuição inicial e o seu fim. Estes servem para alterar o estado das respectivas variáveis dentro do Movicon, evitando que seja o cliente TCP/IP a processar todas essas tarefas.

Capítulo 5

Implementação e Análise de Desempenho

Neste capítulo é descrita a implementação em laboratório do protótipo desenvolvido. Inicialmente é apresentado o módulo de hardware e de seguida é mostrado o aspecto das interfaces desenvolvidas para a monitorização e controlo do sistema e o objecto cliente TCP/IP, desenvolvido em Visual Basic. Por fim, é realizada uma análise do desempenho do protótipo.

5.1 Implementação

A figura 5.1 demonstra uma parte da linha de transporte utilizada neste projecto, existente no Laboratório de Sistemas Flexíveis de Produção.

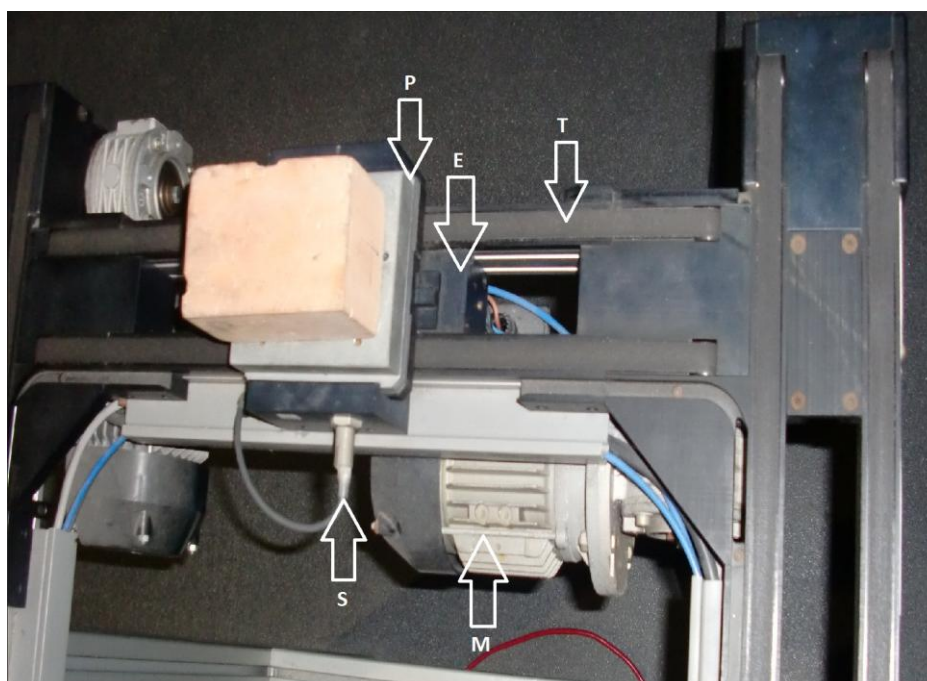


Figura 5. 1 - Excerto da linha de transporte utilizada no Laboratório

Esta linha é composta por um tapete (T), movido por quatro motores trifásicos assíncronos (M), seis electroválvulas (E), seis sensores indutivos (S) e igual número de paletes de transporte (P). De forma a controlar este sistema, foi testado e implementado com sucesso um programa desenvolvido em linguagem C aplicado no microcontrolador 16F877 da Microchip.

Este software tinha como objectivo dirigir a linha de transporte e assegurar toda a monitorização dos seus elementos funcionais, nomeadamente as suas barragens e as suas paletes, utilizando elementos básicos de electrónica tais como resistências, transístores PNP e NPN e LEDs.

A figura 5.2 mostra o módulo de hardware implementado para controlo e aquisição de dados de cada sensor e actuador do sistema, com o adaptador série para comunicação com outros equipamentos.

Este módulo foi implementado em duas placas de teste, porque, para cumprimento dos prazos limite para a entrega desta tese, não foi possível desenvolver placas de circuito impresso (PCB).

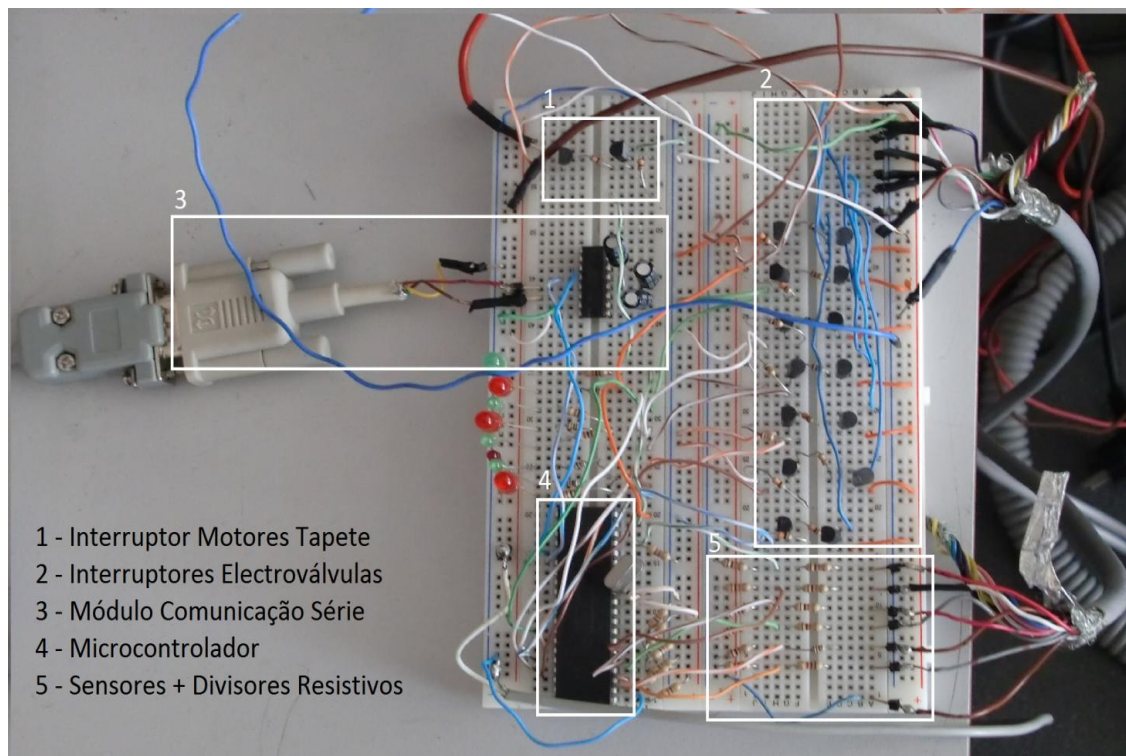


Figura 5. 2 - Módulo de hardware implementado para o controlo da Linha de Produção

Em termos de comunicação, foi utilizado um módulo específico para esse efeito, escolhendo-se a placa PICDEM2 PLUS da Microchip, com o microcontrolador 18F4685 incorporado, e a placa Ethernet PICTail ENC28J60. Desta forma, assegura-se dois tipos de comunicação, a comunicação série com o controlador de linha e, por TCP/IP, com a interface de monitorização e controlo remotos, garantindo uma ligação TCP/IP com velocidade 10 Mbits/s.

Optou-se por esta solução já que o autor desenvolveu projectos por TCP/IP e testou-os com estes módulos, tendo maior à vontade com estes elementos. Este módulo tem a peculiaridade de se ter desenvolvido um adaptador série, de modo a substituir o que está integrado na placa que se danificou durante o desenvolvimento deste projecto.

De seguida, na figura 5.3 apresenta-se o módulo de comunicação com microcontrolador implementado.

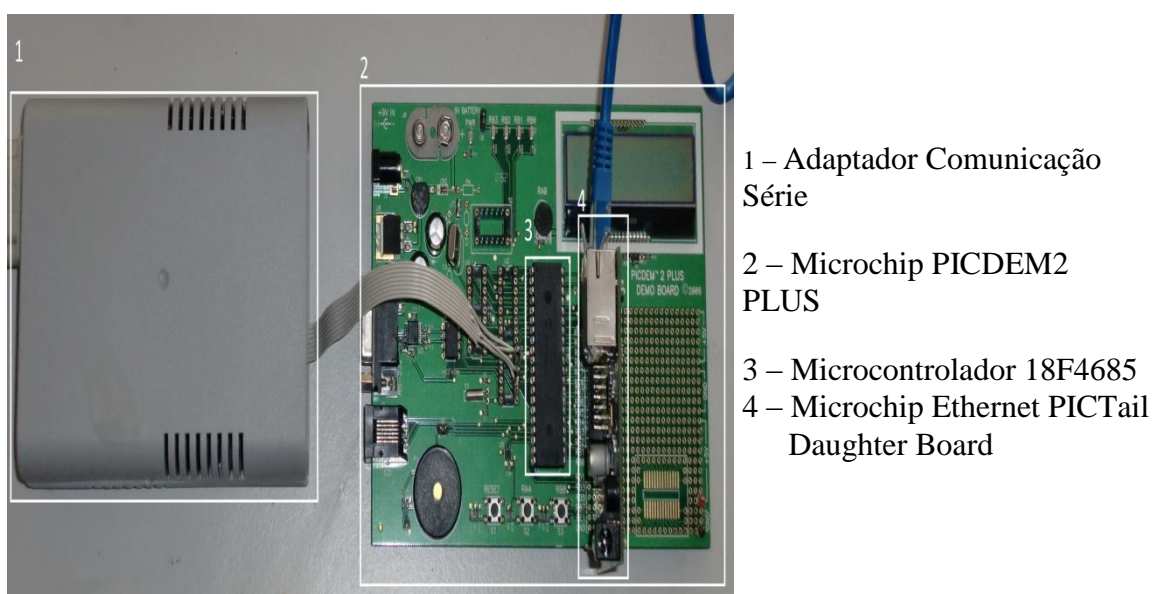


Figura 5. 3 - Hardware do módulo de comunicação do sistema

Na figura 5.4 é possível observar o objecto de comunicação em Visual Basic implementado. A sua interface é extremamente simples, dado que apenas é utilizado como ponte entre a interface gráfica criada utilizando o Movicon e toda a parte física do sistema. Este objecto não é configurável, dado que foi projectado para ser um comunicador entre os dois meios, software e hardware.

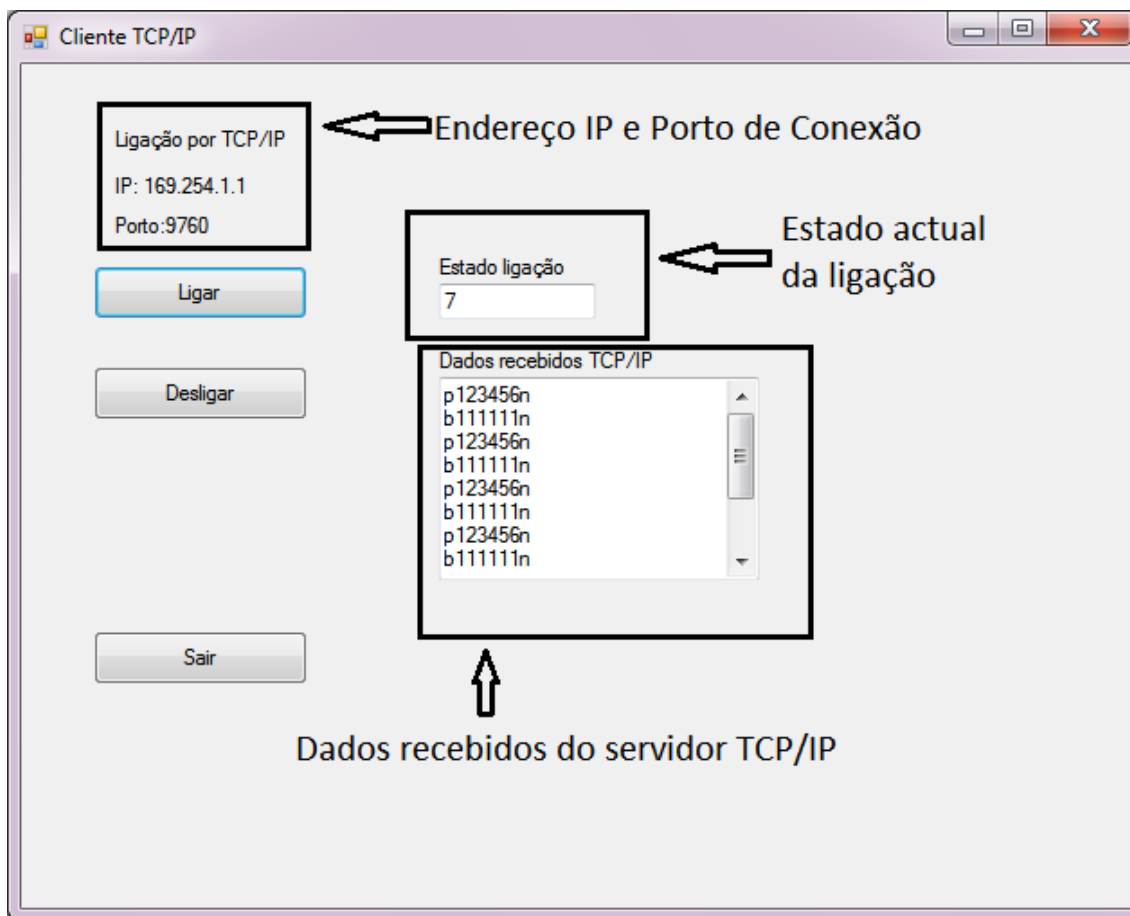


Figura 5. 4 – Interface do objecto de ligação TCP/IP

A figura 5.5 mostra o ecrã principal desenvolvido utilizando o software SCADA Movicon. Este é constituído por uma caixa de texto onde se anuncia o estado de operação do sistema (1), não editável, por seis luzes de controlo de ocupação das barragens (3). Além destes componentes, também contém seis caixas de texto onde se informa a posição de cada paleta (4), igualmente não editáveis, possui um botão que abre uma janela *pop-up* que possibilita o envio de um comando para a linha (5) e uma luz sinalizadora da execução do último comando (6).

Além destas características anteriormente mencionadas, possui um diagrama (2) composto por várias estações de operação de uma fábrica, desde secagem, foto-radiação, compressão, pressurização, cozimento e garra de mudança de posto. Adicionalmente, também representa a ocupação das barragens, através de luzes indicadoras e de animação dinâmica dos ícones representativos de cada estação.

Este diagrama procura transmitir algum potencial da personalização e dinâmica que este software possibilita.

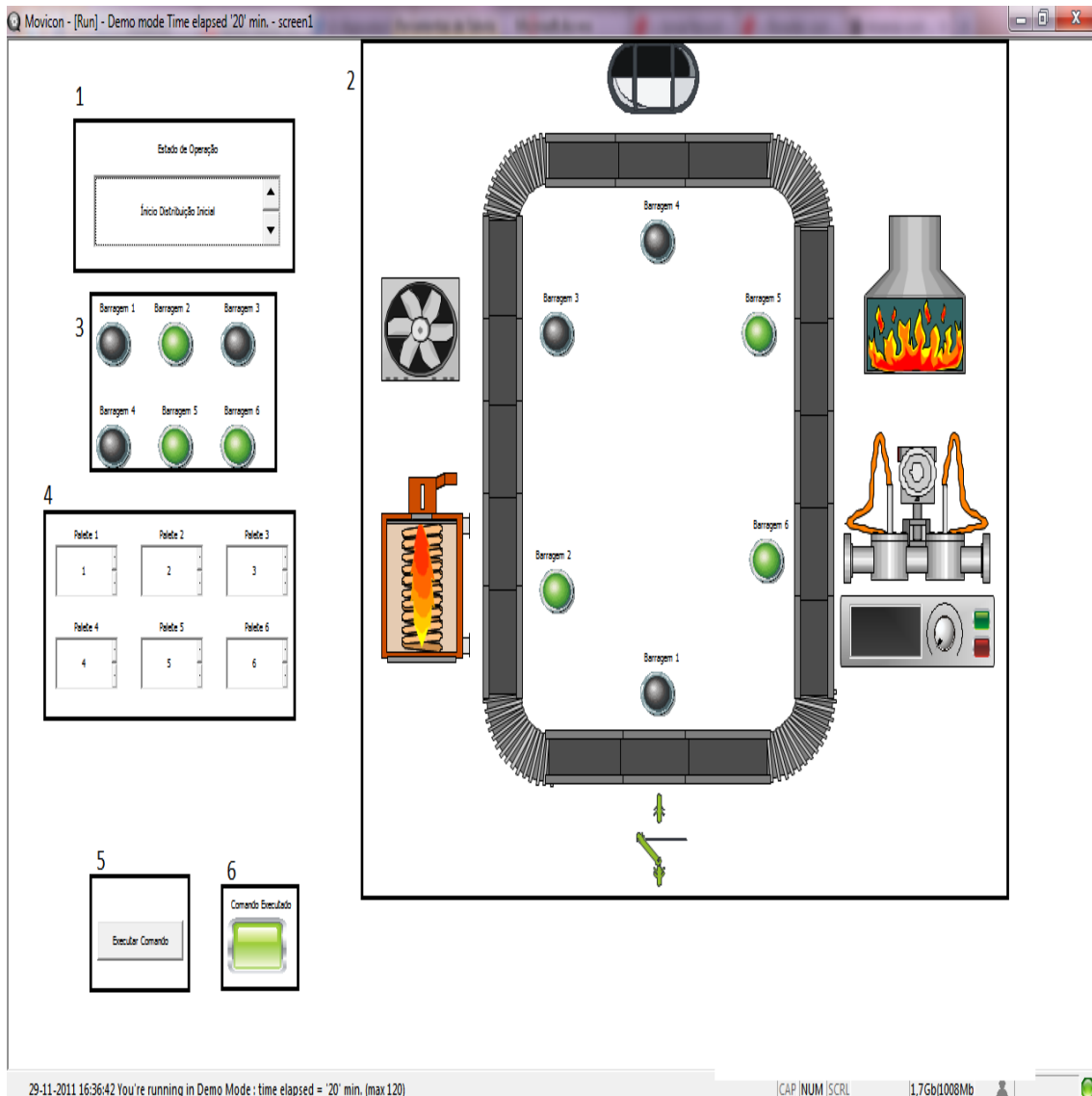


Figura 5. 5 - Interface SCADA utilizando Movicon, ecrã principal

O ecrã de comando da interface SCADA é mostrado na figura 5.6, sendo constituído por duas caixas de texto, editáveis, onde se coloca o valor, entre um e seis, da barragem de destino e da paleta a mover, respectivamente. Adicionalmente, este ecrã contém um botão que gera um evento onde escreve a mensagem de comando, composta pelas variáveis referidas, e fecha automaticamente a janela *pop-up*, regressando ao ecrã principal para se proceder à monitorização do sistema.

Optou-se por colocar o módulo de comando separado da janela principal, separando assim as vertentes de monitorização e controlo, para mais fácil compreensão e melhor organização do sistema.

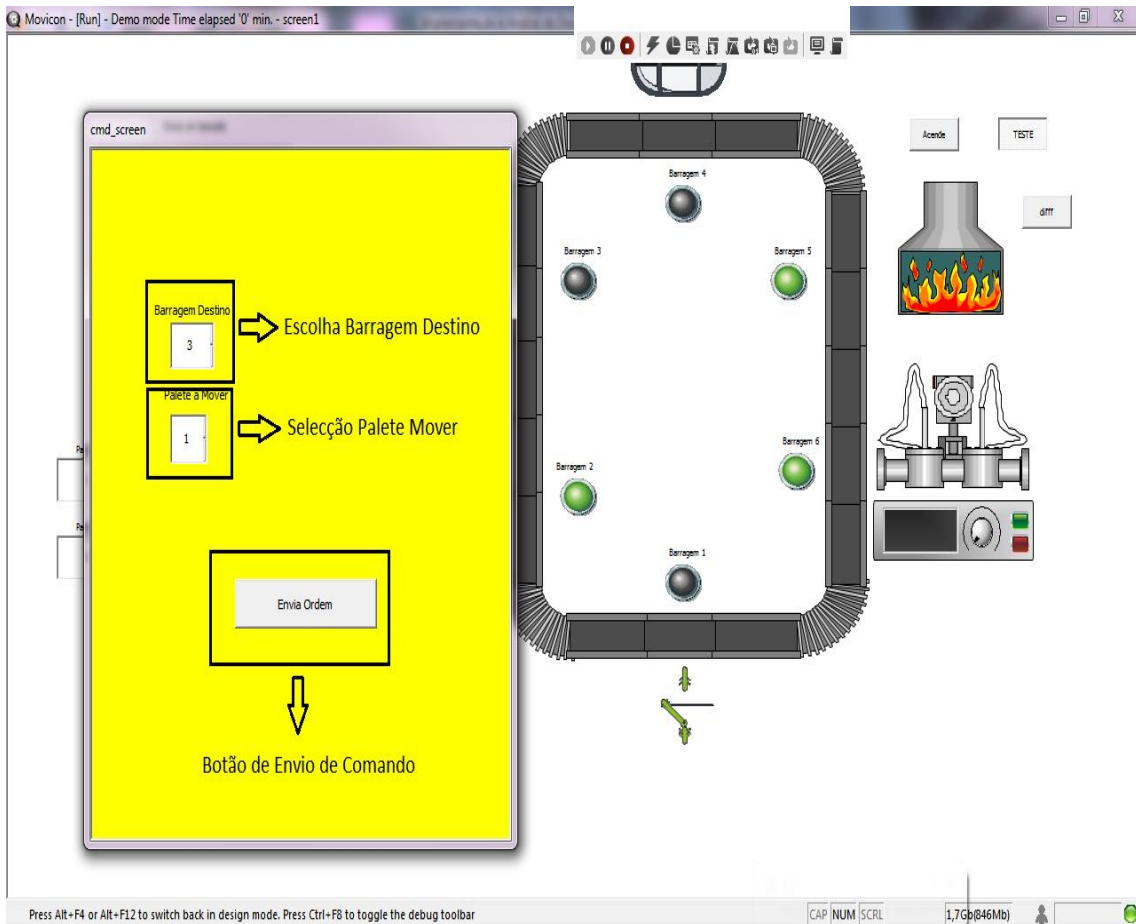


Figura 5. 6 - Ecrã de comando do Sistema, usando Movicon

5.2 Análise de desempenho

Nesta fase de desenvolvimento do protótipo não foram implementados ciclos automáticos de movimento, por selecção do ciclo, através do controlador de linha, o que aumentaria a complexidade dos vários programas projectados. Portanto o controlo da linha é apenas feito manualmente. Também não foi desenvolvida a capacidade de controlo e monitorização através de serviços Web.

No que diz respeito ao microcontrolador da linha, é necessário avaliar o desempenho do seu hardware e software. Em termos gerais, ambas as partes cumpriram com o requerido.

A parte física comprovou que suporta eficientemente as tensões e correntes requeridas para os sensores, transístores, para as electroválvulas e motores, sem haver sobreaquecimento do sistema, encontrando-se dentro da tolerância do microcontrolador.

Em termos de software, este provou ser eficaz e robusto para a aplicação desenvolvida, garantindo a gestão de todas as entradas e saídas do sistema, gerindo convenientemente a recepção e envio de mensagens série e controlando devidamente todas as variáveis e estados utilizados permitindo o funcionamento devido do sistema. No entanto, o software pode ser optimizado dado que, quando se encontra a processar um passo do programa principal, como por exemplo, na presença de uma palete numa barragem, e, nesse momento recebe um pedido de actualização, interrompe o processo no programa principal para enviar a mensagem. Neste instante, há um bloqueio instantâneo, a na ordem dos milissegundos, na barragem, mas que não afecta a funcionalidade do sistema. No entanto, esta imperfeição pode ser corrigida colocando uma condição de interrupção, que garanta que não interrompe nenhum processo.

O software implementado no servidor TCP/IP garante a execução de comandos e a notificação do estado de execução da linha com um atraso mínimo temporal, na ordem dos milissegundos, devido às interrupções geradas no programa desenvolvido. É autónomo no pedido de actualizações ao controlador de linha, guardando as variáveis, e só informando o cliente TCP/IP das actualizações quando este as pede.

No que concerne ao hardware implementado no servidor TCP/IP, o adaptador de comunicação série utilizado é fiável e substitui o adaptador original da placa PICDEM2 Plus.

Relativamente ao programa desenvolvido em Visual Basic, este assegura o transporte de dados com fiabilidade entre o servidor TCP/IP e a interface gráfica. Esta proporciona um ambiente de fácil utilização, veloz e intuitivo, no qual todas as variáveis físicas da linha, designadamente paletes e barragens, e as variáveis de controlo, se encontram representadas e são passíveis de interacção.

Um dos objectivos deste projecto era arranjar uma alternativa aos PLC no controlo e comunicação com máquinas industriais. A tabela seguinte descreve o custo por unidade de cada constituinte do hardware utilizado.

Tabela 5.11 - Lista dos componentes utilizados e respectivo custo

Componente	Quantidade	Custo/Unidade(€/un)
Microchip PICDEM2 Plus 2006	1	99.12
Microchip Ethernet PICtail Daughter Board	1	34.57
PIC 18F4685	1	9.49
PIC 16F877	1	6.29
Transistor BC557C	8	0.182
Transistor BC547B	8	0.24
Resistência	43	0.078
Transformador 24V	1	19.90
Transformador 12V	1	15.90
Cristal 4Mhz	2	2.48
Cabo RS232 Macho-Macho Cruzado	1	3.37
Adaptador RS232 Fêmea	1	0.50
MAX232	1	0.95
Condensadores 1µF	3	0.116
Total		201.178€

O custo estimado para o módulo de hardware desenvolvido é de 201.178€. Os preços que constam da tabela 5.1, à excepção da fonte de alimentação, foram retirados do site do distribuidor electrónico Farnell, Mainland Mart.com e de uma empresa de artigos de electrónica TvLar, para encomendas unitárias. Deste orçamento

é importante salientar que o custo associado à placa de desenvolvimento PICDEM2 PLUS, que representa quase 50% do custo total, pode ser facilmente reduzido, se for substituída por uma placa desenvolvida com características semelhantes, mas sem componentes desnecessários ao necessário, num projecto futuro, especificamente para esta implementação.

Capítulo 6

Conclusão

Na presente dissertação foi abordado um sistema que permite o controlo e monitorização de uma linha de transferência circular existente no laboratório de Sistemas Flexíveis de Produção, através do uso de microcontroladores para controlo e comunicação, além de um objecto de comunicação VB e uma plataforma SCADA.

Em termos de software SCADA seleccionado, escolheu-se a plataforma Movicon, do fabricante Progea, porque possui uma versão de demonstração com poucas limitações e inúmeros textos de apoio. Este software revelou-se uma escolha interessante, dado que, apesar das suas limitações por ser uma versão de demonstração, possui praticamente todos os recursos disponíveis e, no geral, a adaptação por parte do autor a um software desconhecido revelou-se fácil e intuitiva. No entanto, não se conseguiu implementar a plataforma de comunicação TCP/IP integrada com o Movicon devido a problemas na licença do objecto WinSock, mesmo depois de se seguir os tutoriais da marca, que referiam como conseguir resolver esta situação. O autor presume que seja uma limitação da versão de demonstração e, por isso, implementou o objecto de comunicação em VB para conseguir comunicar com o servidor por TCP/IP, lendo e escrevendo na base de dados do Movicon.

Em termos de comunicação, a utilização de uma solução por comunicação série garantiu a fidelidade da passagem de dados do controlador para o servidor, no entanto, seria interessante desenvolver um circuito que integrasse apenas o microcontrolador 18F4685 e comunicação por TCP/IP, reduzindo o número de intermediários na comunicação, o que aumentaria a fiabilidade da comunicação e também reduziria o custo do projecto.

As mensagens trocadas entre cada componente foram implementadas de modo a garantir a unicidade da mensagem, quando comparadas entre si, no entanto, como sugestão futura, pode-se procurar implementar um algoritmo codificador da mensagem, que reduza o tamanho desta, e aumentando a eficiência em termos de mensagens recebidas válidas.

Relativamente aos microcontroladores seleccionados, estes revelaram-se uma escolha acertada em ambos, dado que se conseguiu que o controlador de linha processasse todas as instruções de comando, de memória e de comunicação com o servidor TCP/IP. O servidor TCP/IP revelou fiabilidade na comunicação por RS232 e por TCP/IP com o controlador de linha e com o objecto de interface de comunicações VB, respectivamente, mesmo depois de se ter inutilizado o adaptador de comunicação série.

O restante hardware do controlador de linha, divisores resistivos e transístores como interruptores, foram o complemento ideal do controlador para a aquisição e envio de sinais para a linha de transferência, dado que cumpriram integralmente as funções para as quais foram implementados, sem aquecimento excessivo ou outros danos colaterais para com o restante equipamento. Esta solução revelou-se de baixo custo, quando comparado com os preços que os actuais fabricantes de PLCs industriais implementam.

Os módulos desenvolvidos foram eficazes na comunicação, tanto para o servidor, por parte do objecto de comunicação VB, como na comunicação com a base de dados, também por parte do objecto de comunicação VB assim como por parte da interface SCADA. Esta foi criada com o objectivo de ser de fácil compreensão e com o máximo de informação possível relativa à estrutura que se está a controlar, separando as funções de controlo das funções de monitorização, e descrevendo todos os valores das variáveis da linha de transferência.

Em termos económicos, pode-se concluir que esta alternativa aos PLCs industriais é bastante interessante, dado o seu baixo custo. A sua desvantagem é que o código implementado no microcontrolador não é configurável, isto é, é específico para a linha de transferência em questão. No entanto, as funções implementadas em código podem facilmente ser adaptadas para outras linhas, nas quais varia apenas o número de paletes e barragens. Em termos futuros seria interessante evoluir este código desenvolvido para que permitisse a adaptabilidade a diversas linhas de transferência.

Em termos globais, o sistema é robusto e comporta-se de uma forma bastante convincente, apesar de haver espaço de manobra para melhorias em trabalhos

futuros, nomeadamente na velocidade de troca de mensagens de actualização entre os diversos componentes

Além disso, seria interessante desenvolver uma placa de circuito impresso que esteja preparada para actuar como controlador da linha de transferência e como servidor TCP/IP, implementando todo o código de comunicação, controlo e monitorização num só microcontrolador.

Um outro objectivo passava por apresentar uma solução economicamente competitiva com sistemas de controlo equivalentes, nomeadamente PLCs, placas de aquisição de dados, RTUs, entre outros, e esse objectivo foi cumprido dado que o orçamento para este sistema é bastante inferior ao custo de uma solução que passasse por um PLC a controlar e monitorizar o sistema.

Outra solução poderia passar por integrar esta solução num ambiente flexível de produção, todos controlados através de microcontroladores.

Também seria interessante substituir a HMI SCADA utilizada pelo sistema implementado por Diogo Silva e Costa (2009), tornando a solução global ainda mais competitiva.

Anexos

A.1 Diagramas

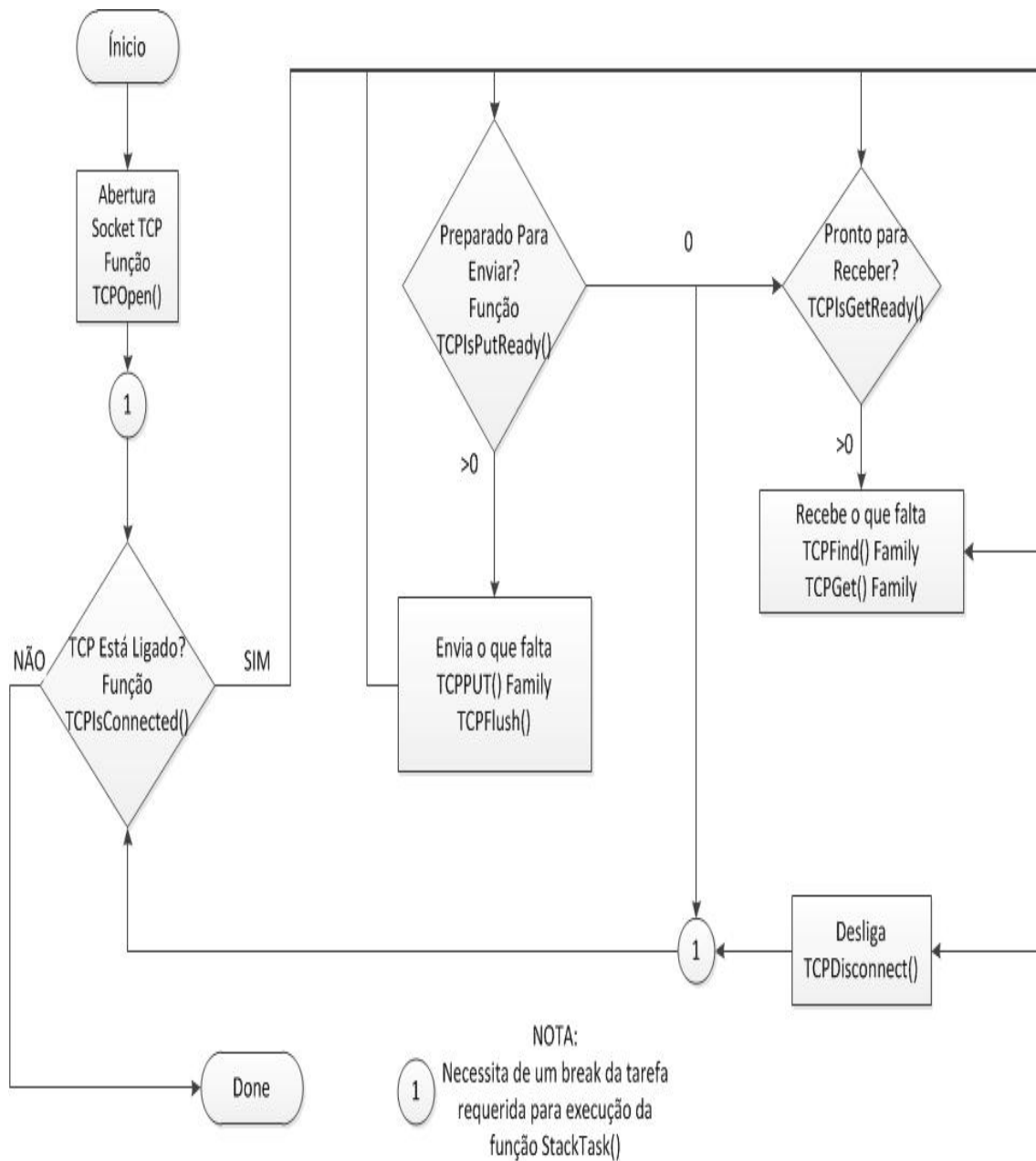


Figura A. 2 - Diagrama de operação do bloco TCP/IP da Microchip

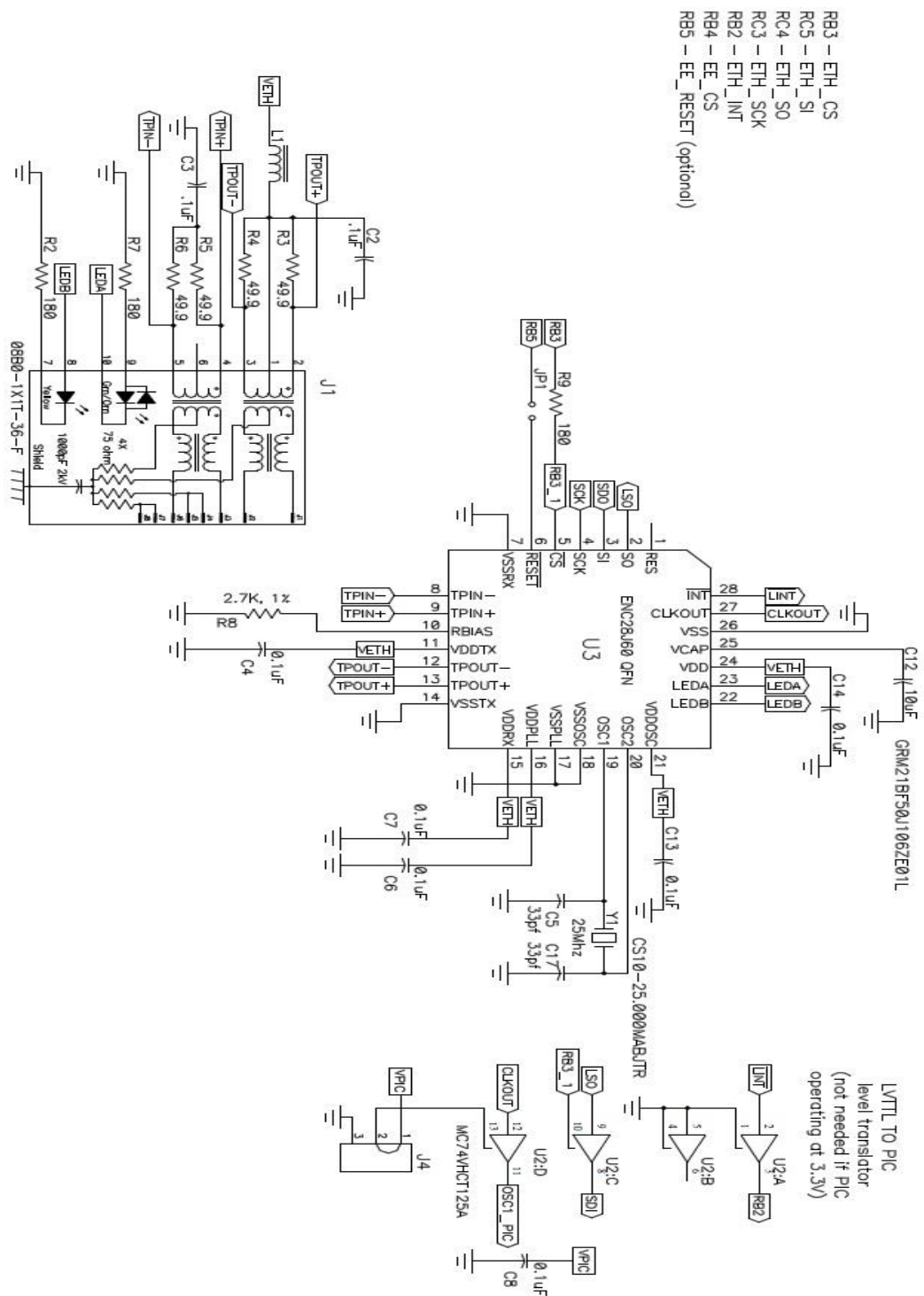


Figura A. 3- Diagrama Eléctrico 1 da Placa Ethernet PICTail Daughter Board da Microchip (Microchip, 2006)

Referências

[Albertjonas, 2010]

ALBERTJONAS – **Architecture of PLC**. [Em linha]. 2010. Disponível na WWW:
<URL: <http://albertjonas.blogspot.com/2010/09/architecture-of-plc.html>>

[Barana, O., et al (2010)]

BARANA, O. [et al.] - **Comparison between commercial and open-source SCADA packages - A case study**. 2010. Padova: Elsevier.

[Bentley Systems Inc, 2004]

BENTLEY SYSTEMS INC - **Fundamentals of SCADA** [em linha]. 2004. Disponível em
WWW: <URL:<http://www.wateronline.com/download.mvc/The-Fundamentals-Of-SCADA-0002>>.

[Bolton, 2006]

BOLTON, W. - **Programmable Logic Controllers**. Oxford: Newnes Fourth Edition. 2006.
ISBN 0-7506-8112-8.

[Citect, 2011]

CITECT – **CitectSCADA**. [Em linha]. 2011. Disponível na WWW:
<URL: <http://www.citect.com/>>

[CopaData, 2011]

COPADATA – **Zenon** [Em linha]. 2011. Disponível na WWW:
<URL: <http://www.copadata.com/en/home.html>>

[Costa, 2007]

COSTA, Diogo ; RITO, Rui - **SOFTWARE DE SUPERVISÃO E CONTROLO DE SISTEMAS DE PRODUÇÃO**. Aveiro: Universidade de Aveiro. 2007. Projecto.

[EGYPTOIL-GAS, 2011]

EGYPTOIL-GAS. **PVSS II - Best fit SCADA system for the Oil and Gas industry**. [em linha]. 2011. Disponível na WWW:

<URL:http://egyptoil-gas.com/read_article_issues.php?AID=64>

[Ibrahim, 2010]

IBRAHIM, Dogan – **SD Card Projects Using The PIC Microcontroler**. 1ª ed. Burlington: Newnes. 2010. ISBN: 978-1-85617-719-1.

[Jack, 2005]

JACK, Hugh - **Automating Manufacturing Systems with PLCs**. [em linha]. 2005.

Disponível na WWW: <URL <http://webbooks.net/freestuff/plc.pdf>> .

[Kasim, 2008]

KASIM, Abdul [et. all] – **Delivering Carrier Ethernet: Extending Ethernet Beyond the LAN**. 1st Ed.. New York: McGraw-Hill, 2008. ISBN: 978-0-07-148747-4

[KEKB, 2000]

KEKB - **KEKB Accelerator Control System**. [em linha]. 2000. Disponível na WWW:

<URL: <http://www-acc.kek.jp/WWW-ACC-exp/KEKB/control/KEKB-Control-home.html#KEKB%20Accelerator%20Control%20System>>

[Learn Networking, 2008]

LEARN NETWORKING – **The TCP/IP Stack and the OSI Model** [Em linha]. 2008.

Disponível na WWW:

<URL: <http://learn-networking.com/tcp-ip/the-tcpip-stack-and-the-osi-model>>

[Microchip, 1997]

MICROCHIP (1997) – Synchronous Serial Port (SSP). In MICROCHIP - **In PICmicro MID-RANGE MCU FAMILY**. 1997.

[Microchip, 2001]

MICROCHIP – **PIC16F87X Data Sheet** [Em linha]. Actual.2001. Disponível na WWW:

<URL:

<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010242>>

[Microchip, 2006]

MICROCHIP – [Em linha]. Actual.2006. Disponível na WWW:

<URL:

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en025831>

[Microchip, 2009]

MICROCHIP – **PIC18F2682/2685/4682/4685 Data Sheet** [Em linha]. Actual.2009.

Disponível na WWW:

<URL:<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010242>>

[Microchip, 2010]

MICROCHIP – **MPLAB ICD2** [Em linha]. 2010. Disponível na WWW:

<URL:

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en010046 >

[MME, 2006]

MME – Basics of PLC Programming. **Industry**. 2006.

[MSDN, 2010]

MSDN - **Getting Started with Visual Basic** [Em linha]. Microsoft Corporation. actual.

2010. [Consult. Out. 2010]. Disponível na WWW:

<URL: <http://msdn.microsoft.com/en-us/library/8hb2a397.aspx>>.

[National Instruments, 2011]

NATIONAL INSTRUMENTS. **NI LookOut** [em linha]. 2011. National Instruments.

Disponível na WWW: <URL:<http://sine.ni.com/nips/cds/view/p/lang/en/nid/12511>>.

[NCS, 2004]

NCS – **Supervisory Control and Data Acquisition (SCADA) Systems**. Boletim. 2004.

Arlington, Virginia. (2004).

[OpenGate Software, 2011]

OPENGATE SOFTWARE - **What is Microsoft Access Used For?** [em linha]. 2011.

Disponível em: <URL:

<http://www.opengatesw.net/ms-access-tutorials/What-Is-Microsoft-Access-Used-For.htm>>

[Parziale, Lydia [et al], 2006]

PARZIALE, LYDIA [et al] (2006) – **TCP/IP Tutorial and Technical Overview**. 2006.

[Phaithoonbuathong, 2009]

PHAITHOONBUATHONG, Punnuluk [et al.]. - **Web services-based automation for the control and monitoring of production systems**. Leicestershire. 2009.

[Patel et al., 2004]

PATEL, M. [et al.] - *Development of a novel SCADA system for laboratory testing*.

SCADA Perth, 2004. Documento

[Progea, 2011]

PROGEA – **Movicon 11** [Em linha]. 2011. Disponível na WWW:

<URL:<http://www.progea.com/en-us/products/movicon11scadahmi.aspx>>

[Quintã, 2008]

QUINTÃ, André Figueiredo (2008). Integração de sistemas de produção. Aveiro: Departamento de Engenharia Mecânica. Universidade de Aveiro, 2008. Dissertação de mestrado.

[Quintas, 2004]

QUINTAS, António Rocha – **SCADA**. 2004. Porto: Faculdade de Engenharia da Universidade do Porto, 2004. Documento

[Silva e Costa, 2009]

SILVA E COSTA, Diogo (2009). Sistemas de visão e controlo integrados. Aveiro: Departamento de Engenharia Mecânica. Universidade de Aveiro, 2009. Dissertação de mestrado.

[Simba, 2011]

SIMBA - **ODBC** [Em linha]. 2011. Disponível na WWW:
<URL: <http://www.simba.com/odbc.htm>>

[SPS Magazin, 2011]

SPS MAGAZIN - **COPA-DATA GmbH** [Em linha]. 2011. Disponível na WWW:
<URL: <http://sps-magazin.de/?inc=hersteller/firmen/index&prefix=49&firma=copadata&op=nh>>

[Spurgeon, 2000]

SPURGEON, Charles E. – **Ethernet: The Definitive Guide** 1ª ed. Sebastopol: O'Reilly Media Inc. 2000. ISBN: 1-56592-660-9

[Strauss, 2003]

STRAUSS, Cobus - **Practical Electrical Network Automation and Communication Systems**. Oxford: Newnes. 2003. ISBN 0750658010.

[Vaz, 2010]

VAZ, Luis Carlos Teixeira - Supervisão e controlo remoto da iluminação na via pública.

Aveiro: Departamento de Engenharia Mecânica. Universidade de Aveiro, 2010.

Dissertação de mestrado.

[Wikipedia, 2011]

WIKIPEDIA - **RS232**. [em linha]. 2009. Disponível em:

<URL:<http://en.wikipedia.org/wiki/RS-232>>

[Wilmshurst, 2007]

WILMSHURST, Tim – **Designing Embedded Systems with PIC Microcontrollers**. Oxford:

Newnes. 2007. ISBN 0-7506-6755-9.

[W3C, 2011]

W3C - **Extensible Markup Language (XML)**. [em linha]. 2011. Disponível na WWW:

<URL: <http://www.w3.org/XML/>>