

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Unidade de Controlo de Traçabilidade e Interface com Smartcamera

Miguel Tiago de Carvalho Cardoso de Meneses

Dissertação realizada no âmbito do Mestrado Integrado em Engenharia Electrotécnica e
de Computadores Major Automação

Orientador: Miguel Velhote Correia (Prof. Dr.)

Co-orientador: Fernando Lobato Dias Leite (Eng.)

Junho de 2010

Resumo

As exigências impostas pela indústria automóvel no controlo e etiquetagem de todos os componentes de um automóvel, fez com que o controlo de traçabilidade fosse cada vez mais uma necessidade a implementar em todos os fornecedores. A estes cabe a responsabilidade de gerir e organizar a sua produção, obrigando a um controlo de traçabilidade em todos os postos de trabalho para assim garantir a correcta sequência na montagem, verificação e teste dos seus componentes, minimizando o risco de defeitos. Praticamente todas estas empresas utilizam sistemas desenvolvidos internamente ou pelo próprio cliente, não havendo claramente um *standard*. Aliado a este controlo, a indústria automóvel impõe ainda uma verificação a cem por cento de todos os produtos e componentes, sendo muitas vezes apenas possível através de sistemas de visão artificial.

De forma a colmatar a necessidade de controlo de traçabilidade em muitas indústrias, foi desenvolvido neste trabalho uma unidade para o controlo da traçabilidade de produtos, testado com um sistema de visão artificial com processamento interno, designada por *smartcamera*. Esta unidade de controlo consiste numa aplicação instalada num computador servidor que concentra e armazena toda a informação da produção, gerindo todos os produtos produzidos em vários postos de trabalho ligados numa rede *Ethernet*. Os diversos postos de trabalho são simulados por uma *smartcamera*, que controla um produto em diferentes fases de montagem, com diferentes inspecções. O produto utilizado no teste foi um PCB de dimensões pequenas, identificado por um código *Datamatrix* único, com diversos componentes electrónicos, sendo verificado a presença de dois relés consoante o posto de trabalho em execução.

No registo de traçabilidade é possível saber não só a sequência de montagem do produto nos vários postos de trabalho, como o resultado do teste em cada um deles e o estado de conclusão do produto. A *smartcamera* necessitou do desenvolvimento de funções, em linguagem *Perl*, para ser autónoma no envio, recepção e tratamento das mensagens, uma vez que as ferramentas do seu software estão focadas para aplicações de inspecção de imagens, sendo mais limitadas a nível de comunicações. A comunicação entre esta e o servidor utilizou um protocolo próprio onde é definido o tipo de mensagens e formatação a usar no início de cada ciclo de inspecção. A aplicação do servidor de traçabilidade foi desenvolvida na linguagem *C#*, permitindo fazer o interface visual da comunicação com os vários postos de trabalho activos, consultar as tabelas de traçabilidade e de parametrização dos modelos, assim como configurar o interface utilizado com os postos de trabalho.

Utilizando vários PCB diferentes e alternando o posto de trabalho na *smartcamera*, foi possível simular a montagem real de um produto. A unidade de controlo de traçabilidade mostrou ser capaz de garantir a sequência correcta de montagem de um produto e impedir que um produto concluído seja re-introduzido novamente na montagem. A estrutura deste software deixa em aberto a possibilidade do desenvolvimento de novas versões com outras funcionalidades em comunicações e armazenamento de dados, contendo mais informações sobre cada produto que os possam caracterizar de forma mais eficaz na detecção de um defeito.

Abstract

The requirements of the automotive industry in monitoring and labeling of all components of a car has made the tracking control a need to implement in all suppliers. These have the responsibility for managing and organizing the entire production, requiring a tracking control in all production positions, to ensure the correct sequence in the assembly and testing, minimizing the risk of defects. Almost all the companies use systems developed internally or by the client, without a clear standard. Allied to this control, the automotive industry also requires a check to one hundred percent of all products and components, being often only possible by industrial vision systems.

In order to fulfill this need of control in many industries, it was developed, in this work, a unit to monitor the traceability of products, tested with an industrial vision system with internal processing, called smartcamera. This control unit was installed on a server computer that gathers and stores all the production information, managing all items produced at working stations, connected by an Ethernet network. These working stations are simulated by the smartcamera which controls a different product assembly phase, with different inspections. The product used in the test was a small PCB, identified by a unique Datamatrix code with various electronic components. In this PCB the presence of two relays is confirmed by the smartcamera, depending of which working station is running.

In the register of traceability, it is possible to follow the assembly of the product in various work stations, and also the test results in each one and the completion status of the product. The smartcamera needed the development of some communication functions, in Perl language, since the tools of the application software are targeted for images inspection, being more limited in communications. The communication between the smartcamera and the server used a specific protocol where it is defined the type of messages and formatting to use at the beginning of each inspection cycle. The application of the traceability server has been developed in C # language, doing the visual interface and allowing the communication with various working stations. It also allows seeing the traceability tables, the configuration of the models, and configuring the interface used with the working stations.

Using several PCBs with different IDs and switching the working station in the smartcamera, it was possible to simulate a real assembly production. The unit tracking control ensures the correct assembly sequence and prevents a finished product to be re-introduced again in the assembly line, demonstrating the main characteristics of a system of this kind. The structure of this software allows the development of new versions, with other features in communications and data storage, containing information about each product that may characterize more effectively in the detection of a defect.

Agradecimentos

Este trabalho e todo o material utilizado foi gentilmente cedido pela empresa Controlar - Electrónica Industrial e Sistemas, Lda, a quem desde já agradeço, em especial ao Eng. Fernando Leite que me incentivou e apoiou na realização do mesmo.

Ao Professor Dr. Miguel Velhote Correia, pela orientação dada ao longo do trabalho e pela atitude incansável que teve sempre para comigo.

Ao Nico Hooiveld, pelo apoio permanente na *Microscan*, e ajuda no desenvolvimento do software da *smartcamera*.

Por fim, mas não menos importantes, às duas mulheres da minha vida, Cristina e Sofia, sem as quais este trabalho não teria sido possível, a quem agradeço e dedico este trabalho.

O Autor

“One machine can do the work of fifty ordinary men.
No machine can do the work of one extraordinary man”

Elbert Hubbard

Conteúdo

| | | |
|----------|---|-----------|
| 1 | Introdução | 1 |
| 1.1 | Motivação e Objectivos a Alcançar | 1 |
| 1.2 | Traçabilidade | 2 |
| 1.3 | Equipamentos Utilizados | 2 |
| 1.4 | Arquitectura do Sistema | 3 |
| 1.5 | Descrição do Documento | 4 |
| 2 | Protocolo de Comunicação | 7 |
| 2.1 | Formatação de Mensagens | 7 |
| 2.1.1 | Mensagem de Verificação de Servidor – TUC Find | 8 |
| 2.1.2 | Mensagem para Ligação ao Servidor – TUC LogIn | 8 |
| 2.1.3 | Mensagem para Registo de Modelo – TUC Model | 9 |
| 2.1.4 | Mensagem para Registo de Produto – TUC Product | 9 |
| 2.1.5 | Mensagem de Envio de Resultado ao Servidor TUC – TUC Result | 10 |
| 2.2 | Sequência de Mensagens | 10 |
| 2.3 | Conclusões | 11 |
| 3 | Software da Smartcamera | 13 |
| 3.1 | Características da <i>Smartcamera</i> | 13 |
| 3.2 | Software de Desenvolvimento | 14 |
| 3.3 | Descrição do Software da <i>Smartcamera</i> | 16 |
| 3.3.1 | Sequência de Teste da <i>Smartcamera</i> | 19 |
| 3.4 | Blocos de Função | 24 |
| 3.5 | Conclusões | 25 |
| 4 | Servidor TUC | 27 |
| 4.1 | Características do Servidor TUC | 27 |
| 4.2 | Interface Série TUC | 28 |
| 4.3 | Tabela de Configuração TUC Modelos | 30 |
| 4.4 | Tabela TUC Traçabilidade | 31 |
| 4.5 | Conclusões | 33 |
| 5 | Ensaio Práticos | 35 |
| 5.1 | Posto de Trabalho e Equipamento Utilizado | 35 |
| 5.2 | Ensaio de Controlo de Traçabilidade | 37 |
| 5.3 | Conclusões | 41 |
| 6 | Conclusões e Trabalho Futuro | 43 |

Lista de Figuras

| | | |
|-----|--|----|
| 1.1 | Arquitectura do sistema de gestão de traçabilidade | 3 |
| 1.2 | Ciclo de posto de trabalho com e sem sistema de traçabilidade | 4 |
| 1.3 | Arquitectura do sistema de gestão de traçabilidade usado com a <i>smartcamera</i> | 4 |
| 3.1 | Software de desenvolvimento <i>VisionScape FrontRunner</i> | 15 |
| 3.2 | Janela de configuração de sequência e parâmetros de ferramentas | 16 |
| 3.3 | Seleção de ferramentas para introduzir na inspecção | 17 |
| 3.4 | Zona de teste e identificação de posto de trabalho na <i>smartcamera</i> | 19 |
| 3.5 | Sequência de teste da <i>smartcamera</i> com todos os ciclos de execução. O ciclo “Inspection” está expandido para uma melhor compreensão do seu conteúdo. | 20 |
| 3.6 | Detalhe dos ciclos "TUC Manager", “TUC Command” e "Timeout TUC Find" | 22 |
| 3.7 | Detalhe da identificação da posição da peça | 23 |
| 4.1 | Software servidor TUC | 28 |
| 4.2 | Software servidor TUC - Configurações de interface série | 29 |
| 4.3 | Software servidor TUC - Interface série servidor-cliente desligada | 30 |
| 4.4 | Software servidor TUC - Interface série servidor-cliente ligada | 30 |
| 4.5 | Software servidor TUC - Tabela TUC Modelos | 31 |
| 4.6 | Software servidor TUC - Edição de tabela TUC Modelos | 32 |
| 4.7 | Software servidor TUC - Tabela TUC Traçabilidade | 33 |
| 4.8 | Software servidor TUC - Edição de tabela TUC Traçabilidade | 33 |
| 5.1 | Equipamento utilizado no trabalho | 36 |
| 5.2 | Produto utilizado para o teste | 36 |

Lista de Tabelas

| | | |
|-----|--|----|
| 5.1 | Testes efectuados nos postos de trabalho | 36 |
|-----|--|----|

Abreviaturas e Símbolos

| | |
|--------|---|
| PCB | <i>Printed Circuit Board</i> |
| XML | <i>eXtensible Markup Language</i> |
| TUC | <i>Traceability Unit Control</i> |
| TCP/IP | <i>Transmission Control Protocol over Internet Protocol</i> |

Capítulo 1

Introdução

A crescente evolução da indústria automóvel levou a que as exigências na fabricação de cada componente fosse levada a níveis de qualidade muito elevados, tornando a sua identificação imprescindível quando se pretende localizar um defeito numa determinada série de produção. Isto fez com que fosse necessário não só ter o registo de todos os produtos produzidos, com o maior detalhe possível, como garantir que estes fossem montados pela sequência correcta nos vários postos de trabalho, impedindo desde logo os defeitos originados na produção. Este nível de excelência faz com que a traçabilidade seja um tema cada vez mais falado nos nossos dias, deixando um grande vazio sobre a forma como é tratado e abrindo a possibilidade da criação de um *standard*.

1.1 Motivação e Objectivos a Alcançar

A criação de um sistema capaz de integrar o controlo de traçabilidade, que fosse universal ao ponto de poder comunicar com múltiplos equipamentos, proporcionou o desenvolvimento deste trabalho. Sendo comum que os equipamentos utilizados para o interface com um controlo deste género necessitem de um computador, a utilização de uma *smartcamera* para este fim, que dispensava qualquer outro equipamento, tornou ainda mais aliciante este projecto. Pretendeu-se criar não só a plataforma de um controlo de traçabilidade, onde estaria concentrada toda a informação da produção, como desenvolver funções próprias na *smartcamera* que interagissem com este sistema, podendo ser utilizado noutras aplicações diferentes. A comunicação entre os dois sistemas teria de ser preferencialmente *Ethernet*, por ser cada vez mais a rede utilizada na indústria, e por permitir velocidades de transmissão elevadas. Esta comunicação teria de se basear num protocolo próprio, definido de acordo com os dados necessários no registo da traçabilidade, que possibilitasse futuras expansões. Todos os produtos teriam de ser identificados e registados de forma a poderem ser consultados no sistema de gestão de traçabilidade, durante e após a sua produção. Com estes registos o sistema deveria garantir que um produto utilize, de forma sequencial, os diferentes postos de trabalho configurados para a sua montagem, sendo impedida a montagem de um qualquer produto que não obedeça à sua sequência de montagem ou que já se encontre concluído.

1.2 Traçabilidade

A palavra traçabilidade deriva da palavra traço, com origem do latim *tractus*, sendo empregue no sentido de seguir passo a passo a vida de um objecto ou produto desde da sua origem até à utilização final. Esta definição aplica-se ao tema deste trabalho, onde se pretende seguir numa cadeia de produção, todos os produtos em todos os postos de trabalho envolvidos. A traçabilidade irá por isso garantir que todos os produtos tenham uma identificação própria, capaz de os distinguir ao longo de um processo de montagem. Esta série de requisitos faz com que cada posto de trabalho tenha de ter disponível a informação de cada produto no início de cada ciclo de trabalho, certificando-se de que este pode iniciar a sua montagem. Tal só é possível se esta informação estiver centralizada num único local, sendo partilhada de forma regrada com cada posto de trabalho. Desta forma, cada posto de trabalho terá apenas de identificar o produto e solicitar a informação desse ao sistema de gestão de traçabilidade para que possa prosseguir a sua montagem, não tendo a necessidade de armazenar nenhuma informação sobre os produtos produzidos. O sistema de gestão de traçabilidade irá verificar se o produto pode ser montado no posto de trabalho, garantindo que obedece à sequência de montagem definida, impedindo a entrada a produtos que tenham falhado a montagem num dos postos de trabalho ou que tentem repetir a operação no mesmo posto de trabalho. Com base neste conceito, pretende-se que sejam evitados erros na montagem dos produtos e registar, da forma mais completa possível, a informação de cada produto na fase de montagem. Depois da conclusão da montagem, a traçabilidade irá permitir encontrar os dados de produção de um produto após a detecção de um defeito reclamado pelo cliente. Numa situação extrema, os produtos poderão mesmo necessitar de serem substituídos no cliente final sobre o risco de provocarem algum tipo de acidente. A identificação dos produtos com defeito, pertencentes a um determinado lote de produção, só será possível localizar se todos estes estiverem identificados e registados no sistema de gestão de traçabilidade, sendo esta a maior vantagem na utilização de um sistema de traçabilidade.

1.3 Equipamentos Utilizados

Para esta aplicação foi proposta a utilização da *smartcamera* VS1 da marca *Microscan* [6], devido às vastas possibilidades de interface, contendo entradas e saídas digitais, uma porta de comunicação *Ethernet* e uma outra série RS-232. Esta *smartcamera* permite ainda o desenvolvimento de funções e ferramentas em linguagem de baixo nível, podendo ser utilizada sem o recurso a mais nenhum outro equipamento. Com todas estas funcionalidades, a *smartcamera* mostrou ter um enorme potencial ao permitir conjugar num único equipamento, um posto de trabalho de verificação visual completo.

Para simular a montagem de um produto real que pudesse ser verificado pela *smartcamera*, utilizou-se um pequeno PCB com alguns componentes, de forma a reconhecer a presença de dois relés. Este PCB é identificado com um código *Datamatrix* (código de duas dimensões), impresso numa etiqueta, descodificável pela *smartcamera*.

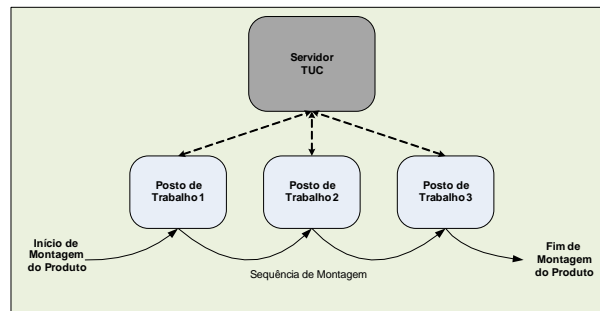


Figura 1.1: Arquitectura do sistema de gestão de rastreabilidade

Todos os equipamentos utilizados, incluindo a iluminação para a inspeção da peça e a estrutura de fixação da *smartcamera*, foram disponibilizados pela empresa Controlar - Electrónica Industrial e Sistemas, Lda, onde este trabalho foi elaborado.

1.4 Arquitectura do Sistema

Durante o projecto do sistema de gestão de rastreabilidade, intitulado por TUC (*Traceability Unit Control*), foram definidas as suas características e todas as partes envolventes. Este sistema é constituído por um servidor onde é gerido o fluxo dos produtos, intitulado por servidor TUC, e por vários postos de trabalho onde é feita a montagem dos produtos (figura 1.1). O servidor TUC e cada um dos postos de trabalho terão um canal de comunicação independente para que sejam trocadas informações entre ambos. Com esta topologia, cada posto de trabalho terá de cumprir uma seqüência de comandos, que lhe permitam validar cada produto antes de iniciar a sua montagem. Desta forma, depois de cada posto de trabalho e respectivos produtos serem configurados no servidor TUC, o início da montagem de um produto só será feito depois do servidor TUC verificar que o produto pode iniciar a montagem nesse local.

Com esta estrutura, cada posto de trabalho necessitará apenas de executar uma série de comandos antes do ciclo de montagem, para a validação do produto, e outros após o ciclo de montagem, para reportar o resultado da montagem ao servidor (figura 1.2). Se o resultado da montagem for mal sucedido ou se o produto finalizar o seu processo de montagem, o seu código não será aceite em mais nenhum posto de trabalho. Os comandos necessários para que se possa garantir a rastreabilidade, contêm funções próprias de um protocolo de comunicação desenvolvido, usado na formatação das mensagens enviadas e recebidas de todos os equipamentos. Com base nos requisitos descritos, adaptou-se esta topologia de comunicações à situação proposta, usando a *smartcamera* com a função de simular os vários postos de trabalho, comunicando com o servidor TUC através de uma rede *Ethernet*. Este equipamento permite comutar facilmente as funções de vários postos de trabalho, mas não em simultâneo, tendo definido internamente o ciclo de cada um deles. Para demonstrar uma aplicação real da funcionalidade deste sistema, utilizou-se um PCB com alguns componentes, cuja montagem necessita de três postos de trabalho. Embora estes postos de trabalho sejam feitos na mesma *smartcamera*, o funcionamento de cada posto de trabalho é feito de

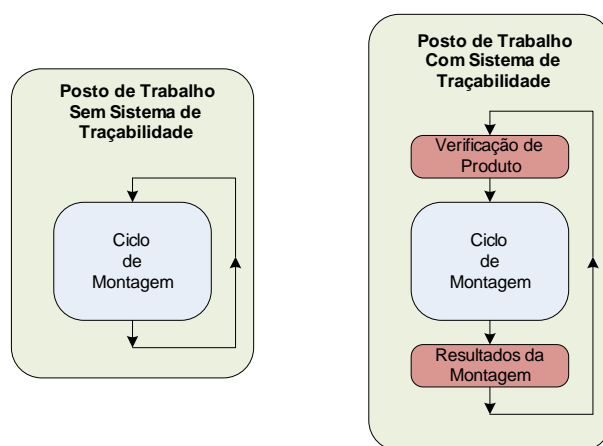


Figura 1.2: Ciclo de posto de trabalho com e sem sistema de traçabilidade

forma completamente independente (figura 1.3).

1.5 Descrição do Documento

Este documento descreve o trabalho focando as várias fases de desenvolvimento pela mesma ordem cronológica com que foram executadas. Partindo da arquitectura do sistema de gestão de traçabilidade, descrito neste capítulo, evoluiu-se no sentido de se criar um protocolo capaz de integrar todas estas necessidades e que fosse utilizável em múltiplas formas de comunicação (capítulo 2). A implementação do protocolo de comunicação na *smartcamera*, a que foi dedicada grande parte deste trabalho, implicou o desenvolvimento de blocos de função programados em linguagem baixo nível e uma estrutura de programa que conjuga o controlo de traçabilidade com os diferentes ciclos de montagem de cada posto de trabalho. Todos estes pormenores poderão ser consultados no capítulo 3. O sistema de gestão de traçabilidade que agrega as configurações de postos de trabalho, produtos e arquivo de resultados, foi desenvolvido posteriormente ao software da *smartcamera* e encontra-se descrito no capítulo 4. Para demonstrar o funcionamento do sistema de gestão de traçabilidade, foram criadas várias situações de montagem com os PCB nos postos de trabalho, descritas no capítulo 5. As conclusões e ideias que surgiram ao longo deste trabalho,

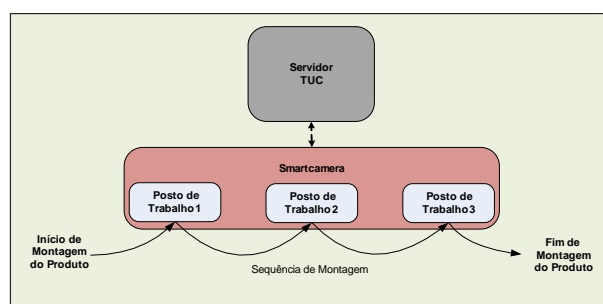


Figura 1.3: Arquitectura do sistema de gestão de traçabilidade usado com a *smartcamera*

foram expostas no último capítulo deste documento, capítulo 6, deixando em aberto um longo caminho na evolução de um sistema de controlo de traçabilidade.

Capítulo 2

Protocolo de Comunicação

A comunicação entre os vários equipamentos necessitou do desenvolvimento de um protocolo de comunicação que identificasse a origem, o destino e o conteúdo de cada mensagem. Esta comunicação é sempre gerida pelo próprio posto de trabalho, enviando mensagens ao servidor TUC a solicitar permissão para se ligar, verificar a entrada de um novo produto e comunicar o resultado do teste. Por cada mensagem enviada, o servidor TUC envia uma mensagem de resposta, permitindo ao posto de trabalho continuar o teste. Com esta estrutura de comunicações, o servidor TUC apenas envia repostas a mensagens recebidas dos postos de trabalho, fazendo o registo de toda a informação que lhes é enviada. Quando uma das mensagens não retorna uma resposta, o ciclo de teste do produto termina com um resultado mau. Embora este protocolo fosse definido para a comunicação entre postos de trabalho e servidor, o mesmo pode ser futuramente expandido para comunicações entre postos de trabalho ou mesmo diferentes servidores, porque o formato da mensagem não restringe a esta única utilização.

2.1 Formatação de Mensagens

As mensagens foram definidas de forma tornar o protocolo expansível e configurável, para as situações de uma produção real. Para tal, separou-se o conteúdo da mensagem em vários campos:

- Caractere de iniciação.
- Identificação do posto de origem.
- Identificação do posto de destino.
- Identificação do comando.
- Dados a enviar.
- Caractere de terminação.

As mensagens são facilmente identificadas através dos caracteres de início “<” e fim ”>”, sendo as várias partes da mensagem separadas pelo caractere “;”. O tamanho dos campos da mensagem não

estão limitados em tamanho, tendo sido formatados com um tamanho exacto de três caracteres para a identificação dos postos de origem e destino, e dois caracteres para o comando neste trabalho. Os dados a enviar não têm um tamanho definido, tendo no mínimo um caractere. Este campo é apenas necessário para as mensagens enviadas pelo servidor, como resposta a alguma pergunta solicitada por um posto de trabalho. No entanto, para não criar um novo formato no envio de mensagens ao servidor, este campo existe neste tipo de mensagens, podendo ter um valor qualquer. Na sequência da explicação anterior, as mensagens ficam formatadas da seguinte forma: “<ooo,ddd,cc,x>”, onde os caracteres “ooo” representam a identificação da origem, os caracteres “ddd” representam a identificação do destino, os caracteres “cc” representam o código da mensagem e o caractere “x” os dados a enviar.

Os códigos usados para a identificação dos postos de trabalho um, dois e três, executados pela *smartcamera*, é representada por “p01”, “p02” e “p03” respectivamente. Para o servidor utilizou-se a identificação “tuc”.

2.1.1 Mensagem de Verificação de Servidor – TUC Find

Esta mensagem permite ao posto verificar a presença do servidor TUC antes de iniciar a verificação do produto. No caso da comunicação não obter resposta, o posto irá terminar o ciclo com um erro de comunicação. Esta mensagem pode ser simplesmente usada para a confirmação do estado activo do servidor.

- Código da mensagem: 01
- Exemplo para o envio de mensagem do posto 1 para o servidor TUC:
 - Formato de envio da mensagem: <p01,tuc,01,0>
 - Formato da resposta da mensagem: <tuc,p01,01,x>
 - O caractere “x” permite apenas a resposta:
 - * 1: Servidor TUC Ligado

2.1.2 Mensagem para Ligação ao Servidor – TUC LogIn

Após a verificação da ligação ao servidor TUC com a mensagem TUC Find, o posto de trabalho tem ainda de se ligar ao servidor para iniciar uma sessão, através da mensagem TUC LogIn. Esta mensagem irá apenas enviar a identificação do posto de trabalho que se pretende ligar. No servidor TUC são registados todos os postos de trabalho que podem ser ligados, permitindo a ligação apenas desses. No instante depois de ter sido feita a ligação ao servidor, o servidor irá guardar os dados do posto de trabalho, ficando a aguardar o envio das mensagens referentes ao produto e modelo em teste.

- Código da mensagem: 02
- Exemplo para o envio de mensagem do posto 1 para o servidor TUC:

- Formato de envio da mensagem: <p01,tuc,02,0>
- Formato da resposta da mensagem: <tuc,p01,02,x>
- O caractere “x” permite as seguintes respostas do servidor:
 - * 0: Servidor TUC não permite ligação de posto de trabalho.
 - * 1: Servidor TUC registou o posto de trabalho.

2.1.3 Mensagem para Registo de Modelo – TUC Model

O início de um teste começa com a leitura do código de identificação do equipamento que se pretende testar. Nesse código encontra-se o número do modelo e o número do produto. O número do modelo identifica uma família de produtos, todos eles identificados com números de série diferentes. Esta mensagem permite autorizar a produção de um determinado modelo num posto de trabalho, e assim garantir que apenas os modelos configurados podem ser produzidos. Desta forma, sendo apenas autorizada a produção de determinados modelos, os postos de trabalho só poderão aceitar produtos desse modelo, evitando assim ocupar desnecessariamente um posto de trabalho com modelos não autorizados.

- Código da mensagem: 03
- Exemplo para o envio de mensagem do posto 1 para o servidor TUC:
 - Formato de envio da mensagem: <p01,tuc,03,0>
 - Formato da resposta da mensagem: <tuc,p01,03,x>
 - O caractere “x” permite as seguintes respostas do servidor:
 - * 0: Servidor TUC não autoriza a produção deste modelo no posto de trabalho.
 - * 1: Servidor TUC autoriza a produção deste modelo no posto de trabalho.

2.1.4 Mensagem para Registo de Produto – TUC Product

Após o registo de um modelo, o posto de trabalho envia ao servidor TUC o código do produto que se pretende testar ou montar. Esta verificação permite ao posto de trabalho saber se o produto específico já foi testado no posto de trabalho anterior ou mesmo se falhou algum teste que o impeça de iniciar a produção nesse posto de trabalho. Para o posto de trabalho interessa apenas saber se pode, ou não, iniciar o teste do produto, não sendo por isso necessário uma resposta detalhada.

- Código da mensagem: 04
- Exemplo para o envio de mensagem do posto 1 para o servidor TUC:
 - Formato de envio da mensagem: <p01,tuc,04,0>
 - Formato da resposta da mensagem: <tuc,p01,04,x>
 - O caractere “x” permite as seguintes respostas do servidor:

- * 0: Servidor TUC não autoriza a produção deste produto no posto de trabalho.
- * 1: Servidor TUC autoriza a produção deste produto no posto de trabalho.

2.1.5 Mensagem de Envio de Resultado ao Servidor TUC – TUC Result

Assim que o teste é concluído, o posto de trabalho envia o resultado do teste ao servidor TUC. Este resultado é a confirmação de que o produto foi testado com ou sem sucesso, ficando registrado no detalhe do produto.

- Código da mensagem: 05
- Exemplo para o envio de mensagem do posto 1 para o servidor TUC, com a informação do resultado do teste:
 - Formato de envio da mensagem: <p01,tuc,05,y>
 - Formato da resposta da mensagem: <tuc,p01,05,x>
 - O caractere “y” permite as seguintes opções:
 - * 0: Posto de trabalho falhou a montagem ou teste do produto.
 - * 1: Posto de trabalho concluiu o produto com sucesso.
 - O caractere “x” permite as seguintes respostas do servidor:
 - * 0: Servidor TUC não conseguiu fazer o registo do teste.
 - * 1: Servidor TUC registou o teste com sucesso.

2.2 Sequência de Mensagens

A comunicação entre um posto de trabalho e o servidor TUC obedece a uma sequência própria de mensagens. As mensagens têm como ponto de partida os postos de trabalho, e destino o servidor TUC, cabendo a este dar resposta a cada uma delas. No início de uma transmissão, assim que um posto de trabalho é aceite pelo servidor TUC, é aberta uma sessão com este. Nessa altura o servidor aguarda a recepção dos dados do modelo do produto a dar entrada no posto de trabalho, e de um resultado para a finalização do ciclo de montagem ou teste. Assim que o servidor envia a confirmação desse resultado, a sessão com esse posto de trabalho é fechada, permitindo ao posto de trabalho iniciar uma nova sessão posteriormente. A sequência de mensagens entre um posto de trabalho (posto 1, representado por “p01”) e o servidor TUC (representado por “tuc”) é descrita nos seguintes passos:

1. Posto de trabalho envia mensagem TUC Find para confirmar a presença do servidor TUC. O código enviado será: “<p01,tuc,01,0>”.
2. Servidor TUC envia mensagem ao posto de trabalho, confirmando a mensagem TUC Find. O código enviado será: “<tuc,p01,01,1>”.

3. Posto de trabalho envia mensagem TUC LogIn para abrir uma sessão com o servidor TUC. O código enviado será: “<p01,tuc,02,0>”;
4. Servidor TUC envia mensagem ao posto de trabalho a confirmar o início da sessão. O código enviado será: “<tuc,p01,02,1>”.
5. Posto de trabalho envia código de modelo do equipamento (por exemplo: “123”), com mensagem TUC Model, pedindo autorização para o teste ou montagem desse modelo. O código enviado será: “<p01,tuc,03,123>”.
6. Servidor TUC envia mensagem de confirmação ao posto de trabalho. O código enviado será: “<tuc,p01,03,1>”;
7. Posto de trabalho envia código de produto do equipamento (por exemplo: “456”), com mensagem TUC Product, para a permissão da entrada deste no mesmo. O código a enviar será: “<p01,tuc,04,456>”.
8. Servidor TUC envia uma mensagem ao posto de trabalho, confirmando o produto. O código a enviar será: ”<tuc,p01,04,1>”.
9. Posto de trabalho envia resultado do teste no final da montagem ou teste, com a mensagem TUC Result. O código a enviar será: “<p01,tuc,05,1>”, para um resultado de teste bem sucedido.
10. Servidor TUC retorna a mensagem TUC Result com a confirmação da sua recepção e arquivo, terminado a sessão com o posto de trabalho. O código a enviar será: “<tuc,p01,05,1>”

A sequência das mensagens tem de ser respeitada, sendo a sessão terminada pelo servidor TUC se esta não se verificar. O servidor pode no entanto terminar a sessão com um posto de trabalho caso as mensagens ultrapassem o número de caracteres definido.

2.3 Conclusões

O protocolo TUC foi desenvolvido para a simples de troca de dados entre um servidor e diversos postos de trabalho, sendo as mensagens definidas de acordo com a realidade de uma linha de produção em ambiente industrial. Estas mensagens permitem ao servidor dar permissão aos testes de cada produto em cada posto de trabalho, regulando a sua passagem em cada um deles. As mensagens foram pensadas de forma tornar a sua compreensão simples e intuitiva, facilitando o teste em modo manual através de programas que permitam o envio de mensagens. Este protocolo permite acrescentar novas mensagens com outras funcionalidades, tornando-o versátil a aplicações onde seja necessária a comunicação entre diferentes postos de trabalho ou mesmo diferentes servidores.

Capítulo 3

Software da *Smartcamera*

Para o ensaio das funcionalidades do sistema de gestão de traçabilidade foi utilizado uma *smartcamera*, uma vez que permite agregar num único equipamento todas as funções de um posto de trabalho. Uma *smartcamera* é um equipamento para o controlo de qualidade que permite, a partir da aquisição de imagens, inspeccionar um determinado produto, tendo como principal vantagem juntar num único equipamento todas as funcionalidades de comunicação e processamento, dispensando a utilização de um computador. As ferramentas de desenvolvimento que a *smartcamera* disponibiliza para a inspeção gráfica e o controlo de qualidade são extremamente completas, sendo contudo limitada no que diz respeito à utilização de comunicações. Esta lacuna na *smartcamera* pode contudo ser ultrapassada com o recurso à utilização de blocos de funções desenvolvidos em programação de baixo nível, onde se consegue aceder a quase todos os recursos físicos do equipamento. Desta forma é possível a implementação de um protocolo de comunicação que permita fazer o controlo de traçabilidade de produtos.

3.1 Características da *Smartcamera*

A *smartcamera* utilizada é o modelo VS1 da marca *Microscan*, o único modelo com estas características desta marca [6]. Este sistema permite fazer a identificação de padrões previamente definidos, reconhecer códigos de barras ou *Datamatrix*, fazer o controlo dimensional e geométrico, reconhecimento de caracteres, entre muitas outras funcionalidades no tratamento de imagens. A nível de comunicações têm uma porta de comunicação *Ethernet*, uma porta de comunicação série RS-232, entradas e saídas digitais. Sendo uma câmara com um processamento interno, que pode ser instalada sem a necessidade de equipamentos adicionais, com capacidade de processar até 3600 inspeções por minuto, é uma solução muito interessante quando comparada com sistemas de visão com processamento em computador, cuja probabilidade de avaria é muito superior. Esta câmara tem ainda de uma extensa área de memória interna para ser utilizada como variáveis binárias, identificada como “virtual I/O”.

A *smartcamera* está preparada para utilizar exclusivamente a porta de comunicações *Ethernet* para a programação e o teste de software, no entanto, a porta de comunicações série permite ter as

mesmas funcionalidades. O interface série é extremamente importante para tornar a *smartcamera* compatível com sistemas mais limitados em comunicações. Após instalada a *smartcamera* num posto de trabalho, a visualização das suas inspecções e estatísticas podem ser facilmente adquiridas com recurso a um browser ligado na mesma rede *Ethernet* pois esta também disponibiliza uma página *Web* com um ambiente gráfico configurável.

3.2 Software de Desenvolvimento

Para o desenvolvimento da aplicação da *smartcamera* utilizou-se o software de desenvolvimento *VisionScape FrontRunner V4.1.1* da *Microscan* [2]. Este software permite compor aplicações de visão artificial de *smartcameras* e de câmaras sem processamento. Neste último grupo de câmaras podem-se separar dois tipos, as câmaras com saída de vídeo analógicas, que necessitam de uma placa gráfica para a digitalização das imagens, e as câmaras com comunicação *Ethernet Giga bit* designadas por *GigE*. As câmaras analógicas foram os primeiros sistemas de visão a ser utilizados, que obriga ao processamento da imagem numa placa gráfica independente, designada por *Frame Grabber*. Hoje em dia, a conversão das imagens em formato digital é feito directamente nas câmaras e enviado em redes de comunicação para o computador onde é feita a aquisição e tratamento das imagens. As câmaras *GigE* pertencem a este último grupo de câmaras que, aliado a elevadas performances de comunicação, permitem ser ligadas até 8 unidades numa única aplicação de software.

A visualização da aplicação desenvolvida no software *VisionScape FrontRunner*, necessita de outro programa, designado *VisionScape ApplicationRunner* [1]. Este software permite apenas fazer a visualização da inspecção, aceder a dados de estatística e resultados, sendo utilizado apenas para a executar a aplicação desenvolvida no outro software.

As ferramentas disponíveis no software *VisionScape FrontRunner* para o desenvolvimento de uma aplicação de visão são muito vastas, sendo completadas por dois tipos de blocos de funções de programação de baixo nível, para a construção de funções totalmente personalizadas, designados por “Custom Step” e “CustomVision Tool” [4, 5]. O primeiro tipo permite construir funções para trabalhar com variáveis, comunicações e todos os recursos físicos da câmara. O outro tipo permite construir ferramentas que utilizem a imagem adquirida pela câmara. Ambos os blocos de função são programados em linguagem *Perl*, sendo editados com um simples programa de processamento de texto.

As ferramentas que o *VisionScape FrontRunner* dispõe para comunicação são suficientes para o envio e transmissão de dados simples, não sendo totalmente flexíveis para a implementação de um protocolo de comunicação com o tratamento de dados. Esta limitação pode no entanto ser colmatada com o recurso a blocos de funções baseadas em “Custom Step”. No conjunto das ferramentas disponíveis existem ainda outras funções para o controlo da sequência do programa, escrita e leitura de sinais, que alteram a ordem de execução do programa de teste. Um programa simples é composto apenas por uma sequência de comandos e funções, sendo todos eles percorridos de forma encadeada e executados por uma única *thread*. Em programas mais complexos pode existir

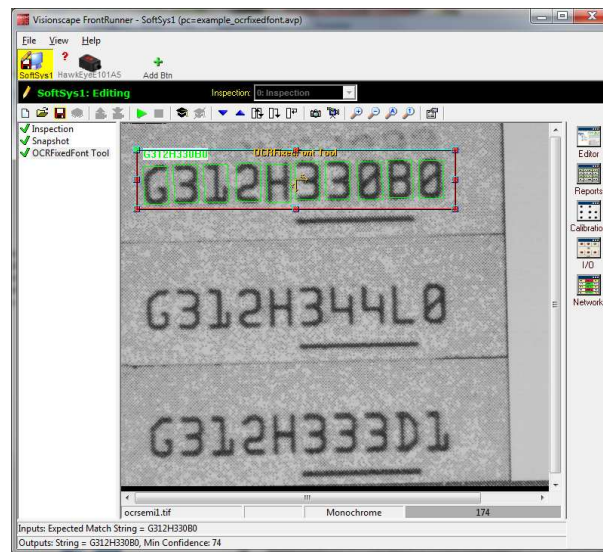


Figura 3.1: Software de desenvolvimento *VisionScape FrontRunner*

mais do que uma sequência a ser executada ao mesmo tempo, sendo cada uma delas gerida por uma *thread*. Esta gestão de recursos possibilita, por exemplo, separar uma sequência para o teste e outra para a gestão de um protocolo de comunicações.

O ambiente de trabalho do VisionScape FrontRunner é composto por uma área central onde é apresentada a imagem e onde são aplicadas as ferramentas de inspeção, uma coluna à esquerda com os passos e resultados da inspeção, uma coluna à direita com as principais funções de parametrização, e uma barra de estado que permite fazer o controlo de funções da câmara e da inspeção. Por cima da barra de estado é ainda possível escolher qual o equipamento que se pretende controlar, sendo sempre possível simular um equipamento com o computador, opção “SoftSys” ou ligar directamente a um equipamento previamente configurado, testando num ambiente real. Na barra da direita existem botões para a edição de diversos parâmetros do sistema: “Editor” para a edição do programa de teste, “Reports” para a configuração dos resultados que se pretendem exportar em cada inspeção, “Calibration” para a calibração das imagens com dimensões reais, “I/O” para a visualização e teste de sinais digitais de interface da câmara e “Network” para a visualização da rede abrangida pela câmara e computador.

A opção “Editor” permite visualizar a sequência do teste e ajustar todos passos e parâmetros de cada ferramenta de inspeção. A coluna da esquerda apresenta o esquema da sequência do teste, com todos os detalhes, sobre a forma de tabela na coluna da direita (figura 3.2).

As figuras 3.1 e 3.2 apresentam um exemplo de um programa para o reconhecimento de um conjunto de caracteres. A página principal do software (figura 3.1) mostra a imagem que se pretende analisar, tendo sobre ela um rectângulo a vermelho que corresponde à inspeção escolhida. Nesta situação é pretendido reconhecer um grupo fixo de caracteres “G312H330B0” numa determinada área da imagem, representada pelo rectângulo a vermelho. A ferramenta “OCR Fixed Font Tool” utilizada nesta inspeção, serve para a identificação de um grupo de caracteres fixo, reprovando o teste se algum dos caracteres não existir ou for diferente. Para o ajuste dessa ferra-

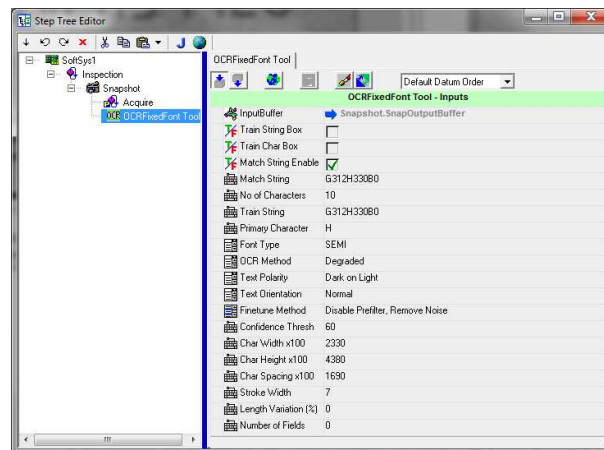


Figura 3.2: Janela de configuração de sequência e parâmetros de ferramentas

menta usa-se o menu “Editor” a partir da página principal, que abre uma janela de configuração com o detalhe de cada passo da inspeção, apresentado na figura 3.2. Nesta página, a coluna da direita mostra todos os detalhes da ferramenta “OCR Fixed Font Tool”, seleccionada na coluna da esquerda. Para uma inspeção simples como esta, são sempre necessários ter uma série de passos na sequência do teste. Esses passos correspondem a uma hierarquia entre pais e filhos, representados pela forma de escada na coluna da esquerda. O primeiro passo é sempre o equipamento onde é feito o teste, sendo este o topo da pirâmide do sistema de visão. De seguida é introduzido um ciclo de inspeção e sobre este a aquisição da imagem e as ferramentas necessárias para o teste. No exemplo anterior, o equipamento de teste é designado por “SoftSys1” pois corresponde a uma simulação em computador. Sobre este existe um ciclo de inspeção, designado “Inspection”, sendo este composto por uma aquisição de imagem, “Snapshot” e uma ferramenta de inspeção de caracteres, “OCR Fixed Font Tool”. A hierarquia entre os vários patamares do teste permite a herança de alguns parâmetros, como é visível na ferramenta “OCR Fixed Font Tool”. Este passo depende do passo “Snapshot”, usando a imagem por ele adquirida para a sua execução.

Entre as ferramentas disponibilizadas pelo software, podem encontrar-se diversos grupos: ferramentas de análise, designadas por “Analysis Tools”, ferramentas para o tratamento e pré-processamento de imagens, designadas por “Image/PreProcessing”, ferramentas para o controlo dimensional, designadas por “Measurement”, ferramentas de controlo e sequência do programa, designadas por “Program Control” e ferramentas personalizáveis pelo utilizador, designadas por “Script” (figura 3.3).

3.3 Descrição do Software da *Smartcamera*

Após o desenvolvimento de um protocolo para o controlo da traçabilidade e de um conceito de comunicações entre postos de trabalho e servidor, foram exploradas todas as características da *smartcamera* para por em prática o protocolo TUC. De uma forma geral, as *smartcameras* estão preparadas para um conceito de processamento baseado numa sequência de teste, percorrendo

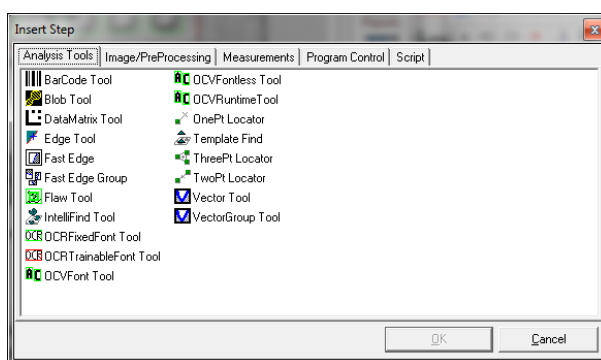


Figura 3.3: Selecção de ferramentas para introduzir na inspecção

passo a passo todas as ferramentas e funções inseridas no programa. Nesses passos podem existir ferramentas para a escrita e leitura nas diferentes portas de comunicação da câmara, não sendo contudo flexíveis ao ponto de implementar um protocolo com perguntas e respostas aleatórias. Surgiu então a necessidade de desenvolver uma ferramenta que se ajustasse a esta necessidade, totalmente flexível para expansões no protocolo e que estivesse permanentemente a monitorizar um porta de comunicação. Este tipo de requisitos corresponde a dizer que a *smartcamera* terá de fazer simultaneamente o ciclo de inspecção de um determinado componente, comunicar com o servidor TUC e gerir as mensagens enviadas pelo servidor TUC. Embora o ciclo de inspecção seja parte integrante de um ciclo de teste simples, este não consegue abranger ao mesmo tempo os ciclos de comunicação e o tratamento de mensagens com o servidor TUC. Por esta razão foi necessário separar estas funcionalidades em instâncias do programa diferentes, correspondendo a três ciclos de inspecção. No primeiro é executado o ciclo de teste, no segundo são recebidas as mensagens do servidor TUC e no terceiro são interpretadas as mensagens recebidas. Desta forma, cada ciclo de inspecção é executado por uma thread no processamento da câmara, não entrando em conflito.

O ciclo que executa o teste necessita de enviar diversos comandos ao servidor TUC durante o arranque e a conclusão do teste do produto, tal como está especificado pelo protocolo TUC. O envio de comandos ao servidor TUC utiliza mensagens com formatos fixos, em passos bem definidos do programa. Para tal utilizaram-se funções de escrita desenvolvidas em ferramenta “Custom Step” que permitissem ter o corpo da mensagem configurável e assim enviar as mensagens de TUC Find, TUC LogIn, TUC Model, TUC Product e TUC Result. A recepção das respostas do servidor TUC obrigou a ter um bloco de função separado do ciclo da inspecção, tal como já foi explicado, que irá actualizar diferentes variáveis do programa, uma por cada tipo de mensagem. Estas variáveis irão ter o valor correspondente ao caractere de dados da mensagem, simbolicamente representado por “x” na formatação das mensagens, explicado no capítulo anterior. Este valor irá permitir que o ciclo de teste continue ou seja interrompido com um determinado erro. Dado que a transmissão de dados entre a *smartcamera* e o servidor TUC pode ter problemas de comunicação ou mesmo estar desligada, tornou-se indispensável definir um tempo limite para obter uma resposta do servidor TUC. Para tal, foram criadas novos ciclos, executados paralela-

mente aos outros dois, um para cada tipo de mensagem enviada. Estes ciclos irão contabilizar o tempo entre o envio e a recepção dos dados de cada mensagem, terminando o ciclo de teste caso esse tempo seja ultrapassado. Estes ciclos são de uma enorme importância uma vez que impedem o ciclo de teste de bloquear num determinado passo da sequência.

O desenvolvimento das funções de comunicação da *smartcamera* foi a parte onde se despendeu mais tempo pois envolveu a programação de blocos de função de baixo nível em ferramentas “Custom Step”. Estando inicialmente previsto que a *smartcamera* iria comunicar através de uma rede *Ethernet*, a forma inicialmente encontrada era a utilização de uma porta de comunicação dentro do endereço IP do equipamento, para a leitura e escrita de mensagens. Muito embora as ferramentas de escrita de mensagens sejam configuráveis, através da porta de comunicação TCP/IP, o mesmo já não acontece com a leitura de mensagens. Este desenvolvimento envolveu os representantes da marca na Europa e a equipa de desenvolvimento nos Estados Unidos da América, nunca tendo sido necessária a implementação de algo semelhante. Após diversas experiências, concluiu-se que as funções de baixo nível disponíveis na *smartcamera* não permitem fazer a recepção de mensagens nas portas TCP/IP, sendo apenas possível a leitura de mensagens fixas, utilizando as ferramentas disponíveis no próprio software de desenvolvimento. Desta forma, a única possibilidade encontrada para o envio e recepção de mensagens *smartcamera* foi a utilização da porta série da mesma. Nesta porta de comunicação era possível utilizar todas as funcionalidades em programação de baixo nível e desenvolver as funções de leitura e escrita. O tipo de comunicações escolhido impõe contudo restrições no desenvolvimento do servidor TUC, pois implica que se utilize uma porta de comunicação série por cada posto de trabalho, em vez de uma ligação TCP/IP, interferindo igualmente com toda a cablagem necessária para se aplica uma solução destas num ambiente industrial. De forma a minimizar todos estes inconvenientes, inoportáveis numa linha de produção, usou-se um conversor de comunicação série sobre uma rede *Ethernet* que simula uma porta de comunicação série no computador através de uma pequena aplicação instalada, que reconhece os conversores de comunicação ligados nessa rede. Desta forma, com o recursos a estes conversores, a ligação do servidor TUC à linha de produção é através de uma porta *Ethernet*, simulando internamente no servidor diferentes portas de comunicação série referentes a cada *smartcamera* utilizada.

A *smartcamera* tem a capacidade de simular um posto de trabalho real, uma vez que tem todas as características necessárias para esse fim. Dispõe de entradas digitais para a ligação de sensores, saídas digitais para indicação do resultado do teste com, por exemplo, sinalizadores, e portas de comunicação para a implementação de um protocolo com o servidor de gestão de traçabilidade. Dado que a simulação de uma produção real necessita de vários postos de trabalho a fim de testar correctamente um controlo de traçabilidade de produtos, o programa da *smartcamera* foi desenvolvido de forma a tornar o número do posto de trabalho configurável. Desta forma, o número do posto de trabalho é alterado por intermédio de variáveis, sendo a informação do posto de trabalho utilizada em todos os passos e funções do programa, não necessitando de ter sequências de teste diferentes para cada posto de trabalho. A distinção entre os vários postos de trabalho é feita apenas no ciclo de teste e verificação do equipamento, tendo cada posto de trabalho

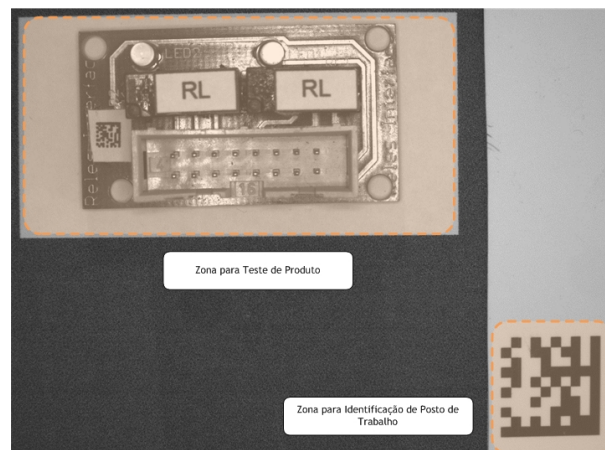


Figura 3.4: Zona de teste e identificação de posto de trabalho na *smartcamera*

uma teste específico no ciclo de inspeção.

3.3.1 Sequência de Teste da *Smartcamera*

Para implementar as funcionalidades de um posto de trabalho na *smartcamera*, foram utilizadas, sempre que possível, ferramentas disponíveis no software *VisionScape FontRunner*, e desenvolvidas outras para situações mais específicas, em blocos de função de programação de baixo nível. O programa é composto por oito ciclos paralelos, o primeiro para a execução do teste, o segundo para a gestão de comunicações com o servidor TUC, o terceiro para a interpretação dos comando enviados pelo servidor TUC, e os restantes cinco para o controlo dos tempos entre o envio e a recepção das mensagens. A zona de inspeção da peça e a respectiva identificação do posto de trabalho, estão limitadas em áreas bem definidas no campo de visão da câmara, conforme pode ser visto na figura 3.4.

O ciclo de inspeção, denominado “Inspection” no programa, concentra toda a informação gerida pelos outros ciclos paralelos, sendo por isso explicado com maior detalhe. Este ciclo foi dividido na figura 3.5 em sete partes distintas, para ser interpretado mais facilmente. A estrutura desde ciclo é formada por vários passos e condições que dão a forma em escada à medida que o ciclo se desenrola. Esta forma caracteriza este ciclo numa cadeia de dependências pois cada condição do ciclo pode fazê-lo terminar em situações de erro. O ciclo “Inspection” é iniciado por um sinal digital, designado frequentemente por *trigger*, no passo “Trigger” do programa. Neste passo é aguardada a mudança de estado numa entrada digital da *smartcamera*. Após a leitura desse sinal, no passo “Limpa Virtual I/O”, são iniciadas as variáveis locais usadas para o sincronismo entre o ciclo “Inspection” e os restantes ciclos.

Após ser dado o arranque do ciclo, e feita a iniciação das variáveis, o ciclo encontra-se pronto para iniciar um teste, fazendo a aquisição da imagem com um passo denominado “Snapshot”. Esta ferramenta cria por defeito o dependente “Acquire” onde é digitalizada a imagem e tem todas as restantes ferramentas do ciclo no mesmo alinhamento. Esta forma significa que as ferramentas utilizadas no decorrer do ciclo “Inspection” podem utilizar a imagem adquirida no passo “Acquire”

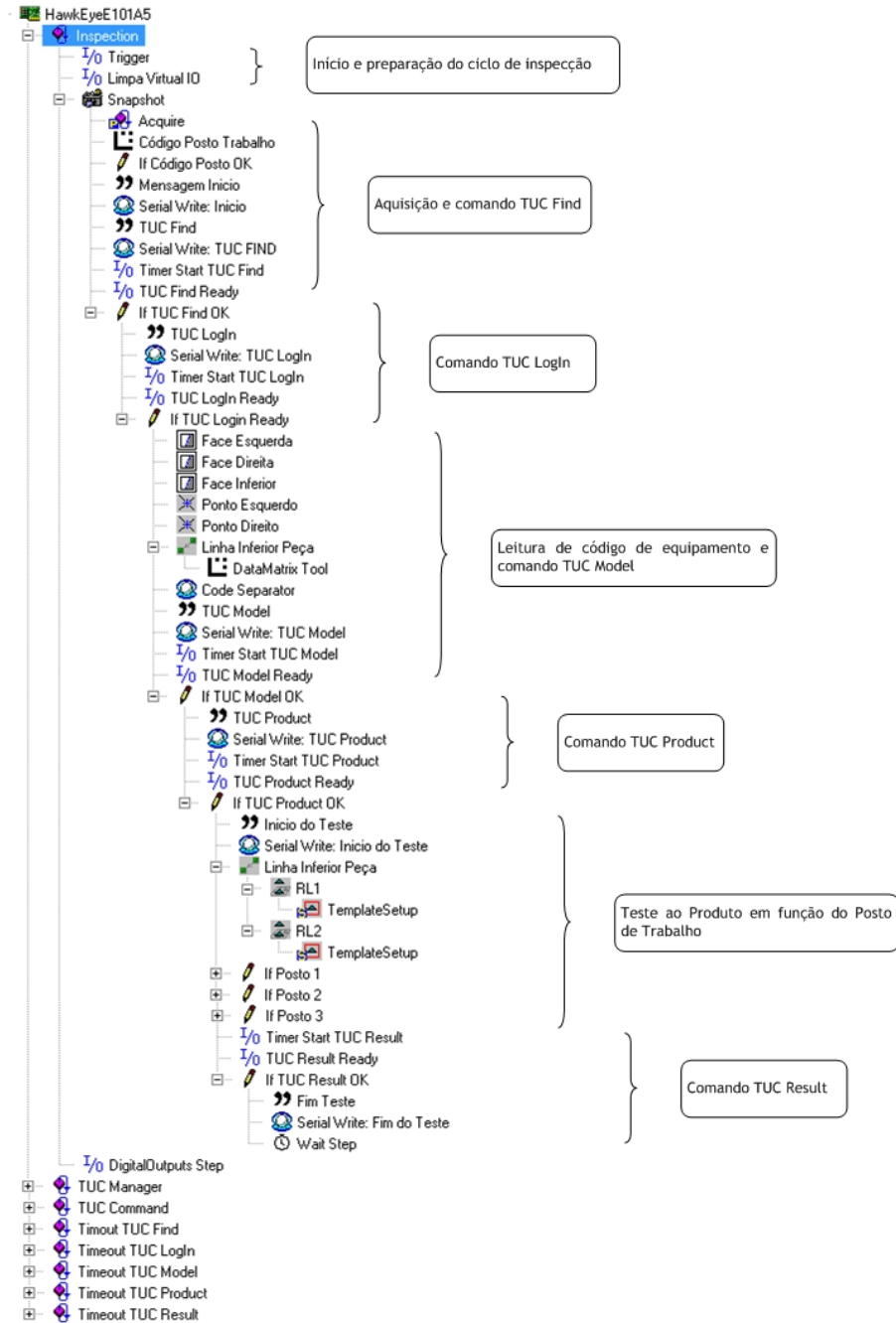


Figura 3.5: Sequência de teste da *smartcamera* com todos os ciclos de execução. O ciclo “Inspection” está expandido para uma melhor compreensão do seu conteúdo.

para inspeção. A configuração do posto de trabalho é feita no passo seguinte, “Código Posto Trabalho”, com a leitura de um código *Datamatrix* que identifica o posto de trabalho e é utilizado ao longo do ciclo de inspeção, por outras ferramentas, para a formatação das mensagens enviadas ao servidor TUC. Assim que o posto de trabalho é reconhecido, é iniciada a comunicação entre a *smartcamera* e o servidor, com a mensagem “Serial Write: TUC Find”. Esta mensagem é enviada com recurso à ferramenta de envio desenvolvida na ferramenta “Custom Step”, que permite o envio de uma mensagem configurável na porta série da *smartcamera*. Dado que a *smartcamera* simula os vários postos de trabalho, esta mensagem incorpora a variável resultante do passo “Código Posto Trabalho”, a fim de poder enviar a mensagem relativa ao posto de trabalho em execução. Esta mensagem tem o formato “<ooo,tuc,01,0>”, onde os caracteres “ooo” são, por exemplo, “p01” para uma mensagem enviada pelo posto de trabalho um. Após o envio da mensagem, o servidor TUC terá de dar uma resposta à mensagem enviada num período de tempo definido por um temporizador no ciclo “Timeout TUC Find”. Este temporizador é iniciado com a escrita de uma variável binária no passo “Timer Start TUC Find”, imediatamente a seguir ao envio da mensagem. Nesta altura, o ciclo de inspeção fica a aguardar por uma condição verdadeira no passo “TUC Find Ready”, controlada pelas variáveis dos ciclos de comunicação com o servidor “TUC Command”, e pelo ciclo de temporização “Timeout TUC Find” (figura 3.6). O primeiro destes dois ciclos a ser executado irá fazer avançar o ciclo de inspeção para o passo “If TUC Find OK”, onde será lido o resultado do envio da mensagem. Neste passo é verificada a resposta do servidor TUC à mensagem enviada. Se a mensagem tiver o valor “1” o ciclo prossegue para o próximo passo, caso tenha um valor diferente, o ciclo termina com um erro. Quando a mensagem é recebida e interpretada pelo ciclo “TUC Command”, este irá activar uma variável binária para concluir o passo “TUC Find Ready” e escrever o resultado da mensagem recebida na variável lida no passo “If TUC Find OK”. Esta última variável é iniciada com o valor “0” na execução do passo “Reset Virtual I/O”, no início do ciclo de inspeção. Se o servidor TUC enviar uma mensagem de retorno, esta variável irá ser alterada para o valor 1 e assim validar a condição do passo “If TUC Find OK”. No caso em que o servidor TUC não enviar nenhuma resposta, o ciclo de temporização “Timeout TUC Find” irá terminar a sua execução no final do tempo definido, activando a variável do passo “TUC Find Ready”. Nesta situação, como não foi lida nenhuma mensagem, a condição do passo “If TUC Find OK” irá falhar pois a variável continua com o valor inicial de “0”, impedindo o ciclo de prosseguir. Desta forma, com o recurso a duas variáveis, é possível fazer o controlo de cada mensagem enviada ao servidor TUC no ciclo de inspeção.

Após a verificação da presença do servidor TUC com a mensagem TUC Find, o posto de trabalho tem de iniciar a sessão com o servidor TUC com a mensagem TUC LogIn. O processamento desta mensagem é feito de forma semelhante à anterior, sendo o corpo da mensagem, por exemplo, “<p01,tuc,02,0>” para o posto de trabalho um. O processamento da resposta e o controlo do tempo de resposta são feitos pelos ciclos “TUC Command” e “Timeout TUC LogIn”, à imagem do anterior. Para que o ciclo de inspeção continue, no passo “If TUC LogIn OK”, a mensagem recebida pelo servidor terá de ser “1”.

A identificação do modelo e produto a testar é feita com a leitura de um código *datamatrix*.

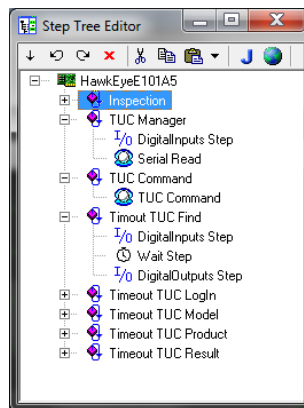


Figura 3.6: Detalhe dos ciclos "TUC Manager", "TUC Command" e "Timeout TUC Find"

Este formato de código tem a vantagem de permitir a introdução de caracteres alfanuméricos e de ter uma capacidade de armazenamento elevada, face ao espaço reduzido que ocupa. Uma vez que a peça se encontra sempre na mesma posição, foi necessário adicionar algumas ferramentas de forma a posicionar a ferramenta de leitura do código sempre na mesma posição relativa da peça. Esta posição é facilmente encontrada após a identificação de dois pontos de referência na peça, previamente calibrados, que deslocam a área de leitura do código *Datamatrix* em função da posição da peça (figura 3.7). Dado que a peça é rectangular, utilizaram-se os dois pontos extremos da face inferior da peça encontrados através da intercepção das linhas das faces esquerda, direita e inferior, pelas ferramentas "Face Esquerda", "Face Direita" e "Face Inferior". Assim que encontrados os dois pontos de referência da peça, com a união das linhas da face esquerda e inferior e das linhas da face direita e inferior, com as ferramentas "Ponto Esquerdo" e "Ponto Direito", é traçada a linha inferior da peça com a ferramenta "Linha Inferior Peça". Essa ferramenta tem como dependente a ferramenta de leitura do código da peça, denominado por "DataMatrix Tool", uma vez que a zona de leitura do código depende directamente da posição e orientação dessa linha. Após lido o código da peça, a função "Code Separator" irá separar o código do modelo e produto, que será usado posteriormente nas mensagens de envio TUC Model e TUC Product. Esta função, desenvolvida em "Custom Step", faz a separação do código do modelo e do código do produto, separados pelo caractere "-", como pode ser visto na figura 3.7, um exemplo de uma peça com o código de modelo "001" e código de produto "100". Assim que identificada a peça, a mensagem "TUC Model" envia a mensagem "<p01,tuc,03,001>" ao servidor TUC, identificando o posto de trabalho um por "p01" e o modelo "001", para o exemplo da figura 3.7.

A mensagem TUC Product tem um formato semelhante ao anterior, sendo composta pelos caracteres "<p01,tuc,04,100>", para o posto de trabalho um, e um código de produto "100". A recepção de mensagens e a monitorização do tempo decorrido para as mensagens TUC Model e TUC Product são feitos nos ciclos "TUC Command", "Timeout TUC Model" e "Timeout TUC Product", em tudo semelhante aos anteriores envios de mensagens. As condições impostas nos passos "If TUC Model OK" e "If TUC Product OK" serão validadas com uma mensagem com o valor "1" para assim se iniciar o teste ao produto.

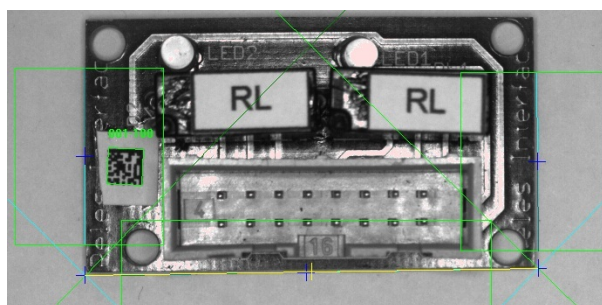


Figura 3.7: Detalhe da identificação da posição da peça

O controlo de traçabilidade impede um equipamento de ser testado num posto de trabalho sem que seja feita a verificação da sequência da sua evolução de montagem num servidor externo. Após o início da sessão com o servidor TUC e de ser verificada a permissão para um determinado equipamento, o teste é iniciado e, dependendo do posto de trabalho seleccionado, será feito um controle específico. Nesta aplicação utilizou-se um equipamento que é constituído por dois relés, identificados por “RL” na figura 3.7. O teste do equipamento corresponde à identificação da presença destes relés. Para que se simulasse a montagem de um equipamento em três postos de trabalho diferentes, considerou-se que no primeiro posto não haveriam relés no equipamento, no posto de trabalho dois seria verificada a presença apenas do relé direito e no posto de trabalho três verificada a presença dos dois relés. Desta maneira, na sequência de teste, os passos “RL1” e “RL2” irão identificar um padrão previamente configurado nas posições dos relés direito e esquerdo, respectivamente, de forma a identificar a sua presença. Estas duas ferramentas estão sobre a dependência da anterior “Linha Inferior Peça”, que corresponde à mesma linha de referência da leitura do código da peça, para que fossem posicionadas as ferramentas de identificação dos relés na zona correcta da peça. Após a verificação da presença dos relés, a sequência continua no ciclo do posto lido, podendo ser entre o posto de trabalho um, dois e três, entrando num dos passos: “If Posto 1”, “If Posto 2” ou “If Posto 3”. O teste feito para cada posto de trabalho corresponde à verificação da presença dos relés certos e posterior envio do resultado ao servidor TUC, com a ferramenta “Serial Write: TUC Result”. A realização do teste é traduzida num valor binário com um simples significado de passou ou falhou o teste. Na situação de um teste correcto, a mensagem enviada ao servidor TUC pelo posto de trabalho um, seria constituída pelos caracteres “<p01,tuc,05,1>”. O envio do resultado do teste ao servidor TUC, é confirmado com uma resposta que valida a condição “If TUC Result OK” com o valor “1”. No seguimento deste passo, o teste termina, sendo de novo iniciado o ciclo de inspecção “Inspection”. Após a conclusão do teste, o ciclo “Inspection” termina com a escrita do valor “0” na variável que faz o *trigger* do teste, no passo “DigitalOutputs Step”.

O ciclo “TUC Manager” é o responsável pela recepção das mensagens do servidor TUC, em resposta às mensagens enviadas no ciclo de inspecção “Inspection”. Este ciclo é constituído por uma ferramenta desenvolvida em “Custom Step”, para a leitura da mensagem recebida na porta série, “Serial Read”, que é precedido pelo passo “DigitalInputs Step” que restringe a execução

deste ciclo com o estado a “1” da variável que faz o trigger do ciclo de inspecção.

A interpretação da mensagem recebida pela ferramenta “Serial Read” no ciclo “TUC Manager”, é permanentemente monitorizada pela ferramenta “TUC Command”, dentro do ciclo com o mesmo nome. Esta ferramenta utiliza a base “Custom Step” e separa todos os campos da mensagem recebida, separando os vários campos da mensagem apenas quando a mensagem se destina a esse posto de trabalho. As mensagens recebidas são depois convertidas em variáveis acessíveis em qualquer parte do programa, validando os passos que aguardam a resposta do servidor TUC, como por exemplo, o passo “TUC Model Ready”. Após a validação desse passo, é verificada a resposta do servidor TUC à pergunta efectuada e, dependendo da resposta, o ciclo continua ou termina.

Os ciclos de temporização usam todos a mesma estrutura de passos. A temporização é iniciada pela mudança de estado de uma variável binária no passo “DigitalInput Step” que, no caso do ciclo “Timeout TUC Find”, é activa no passo “Timer Start TUC Find”. Segue-se um passo onde é contabilizado um tempo específico e por fim alterado o estado de uma variável no passo seguinte, “DigitalOutput Step”. Quando, por qualquer razão, a *smartcamera* não recebe nenhuma resposta do servidor no tempo previsto, a execução deste último passo irá desbloquear o ciclo de inspecção que se encontra no passo “TUC Find Ready”, falhando na verificação da mensagem recebida no passo “If TUC Find OK”. Os restantes ciclos de temporização das mensagens TUC LogIn, TUC Model, TUC Product e TUC Result são processados da mesma forma, todos eles independentes do ciclo de inspecção.

A estrutura do programa utilizado na *smartcamera* tornou-se um pouco extenso devido à compatibilidade que se pretendia ter entre os diferentes postos de trabalho com o recurso a um único programa. Este pode ser simplificado se for feito para um único posto de trabalho com um único teste. O teste efectuado ao equipamento serve apenas de exemplo pois está longe da complexidade que se consegue com um equipamento deste género.

3.4 Blocos de Função

O uso de blocos de função surgiu com a necessidade de se criar funções específicas, não disponíveis nas ferramentas do software de desenvolvimento, a partir de ferramenta “Custom Step”. Esta linguagem de baixo nível, programada em linguagem *Perl*, permite aceder a quase todas as funcionalidades de hardware da *smartcamera*, expandindo as suas capacidades ao criar funções personalizadas. Os blocos de funções são constituídos por um ficheiro de extensão “.pm” editáveis com um simples programa de processamento de texto. Após a utilização de um destes ficheiros na ferramenta “Custom Step”, o mapa de entradas e saídas da ferramenta é alterada de acordo com o definido no programa, podendo ter variáveis de entrada, variáveis de saída e constantes. O mesmo programa pode ser utilizado em várias ferramentas “Custom Step” em simultâneo, cada uma delas com identidade própria e independente.

3.5 Conclusões

A aplicação desenvolvida na *smartcamera* exigiu um estudo profundo de todas as suas capacidades e limitações, atribuindo-lhe características para as quais o próprio software de desenvolvimento não estava preparado. O uso de ciclos de processamento próprios para a gestão de comunicações expandiu os limites de um equipamento que se pensava estar unicamente direccionado para o controlo de qualidade visual. Estas novas funcionalidades fazem com que um posto de trabalho, ligado a um servidor de traçabilidade, possa ser feito directamente por uma *smartcamera*, dispensando a utilização de um autómato programável ou de um computador.

Capítulo 4

Servidor TUC

O controlo de traçabilidade só é possível quando existe na estrutura da rede de comunicação dos diversos postos de trabalho, uma unidade que faça a gestão e o controlo de todos os produtos em fabrico. Esses produtos terão de ser produzidos em postos de trabalho específicos, com uma determinada sequência, ficando o seu registo guardado no servidor para posteriormente poderem ser consultados. Com base nestes requisitos, foi desenvolvido um servidor de traçabilidade que comunicasse com os diversos postos de trabalho, simulados pela *smartcamera* utilizada, e gerisse o fluxo dos produtos por ela inspeccionados.

4.1 Características do Servidor TUC

A aplicação de servidor foi desenvolvida na plataforma *Visual Studio* na linguagem *C#*, que permite a integração de ferramentas de comunicação, armazenamento de dados e interface com o utilizador de forma muito flexível (figura 4.1). O servidor foi focado especificamente para o interface com a *smartcamera*, tendo no entanto características que o permitem evoluir a nível de comunicações, armazenamento e interface.

Dado que a *smartcamera* utiliza uma comunicação série para transmissão e recepção de comandos, utilizou-se uma porta série do computador que executa a aplicação do servidor TUC para a sua monitorização. Tal como explicado no capítulo anterior, esta porta série pode ser uma porta física ou uma porta virtual, simulada com a aplicação de um conversor de comunicação série na rede *Ethernet*. Esta foi a forma encontrada para integrar a *smartcamera* num ambiente industrial onde a rede de comunicações entre os vários postos de trabalho é *Ethernet*. O servidor TUC permite configurar a porta de comunicação, enviar comandos manualmente, podendo inclusive desligar o controlo de traçabilidade. Desta forma é possível comunicar com a *smartcamera* manualmente e dar resposta às mensagens recebidas com comandos editáveis.

As mensagens recebidas e enviadas pela porta série podem ser visualizadas directamente num terminal, não sendo a forma mais indicada para a interpretação das mensagens. Por esta razão, o

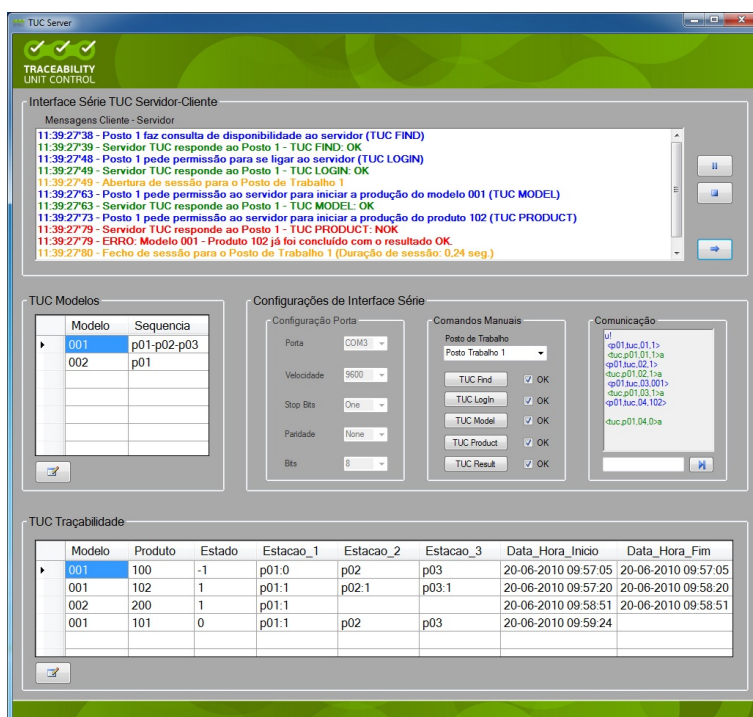


Figura 4.1: Software servidor TUC

servidor TUC disponibiliza uma janela com a informação, em forma de texto dos comandos enviados e recebidos, onde é possível visualizar directamente a inserção de produtos, erros relacionados com a traçabilidade e tempos de execução do teste e de sessão com o servidor TUC.

A configuração dos modelos e o registo da traçabilidade, é feito em duas tabelas visíveis no servidor TUC, a tabela TUC Modelos e a tabela TUC Traçabilidade. A tabela TUC Modelos permite configurar todos os modelos e a sua sequência de montagem dos postos de trabalho. Esta tabela é utilizada na verificação dos modelos e postos de trabalho sempre que solicitada numa mensagem recebida. A tabela TUC Traçabilidade guarda a informação dos produtos que já foram submetidos a pelo menos um teste num posto de trabalho. Esta tabela regista todos os produtos com a sequência de montagem dos postos de trabalho, o estado de produção do produto e os instantes de introdução e conclusão da sua fabricação. Ambas as tabelas guardam os dados em ficheiros do formato XML independentes, que além de terem uma elevada capacidade de compressão de dados, permitem uma leitura simples num editor de texto. Desta forma, os dados são facilmente copiados e armazenados com a manipulação de dois ficheiros.

4.2 Interface Série TUC

O interface com a *smartcamera* é configurável na secção “Configurações de Interface Série” da aplicação. Esta secção é dividida em três zonas distintas, a sub-secção “Configuração Porta”, a sub-secção “Comandos Manuais” e a sub-secção “Comunicação”. A configuração da porta série utilizada, velocidades e formatação da comunicação, são feitas na sub-secção “Configuração

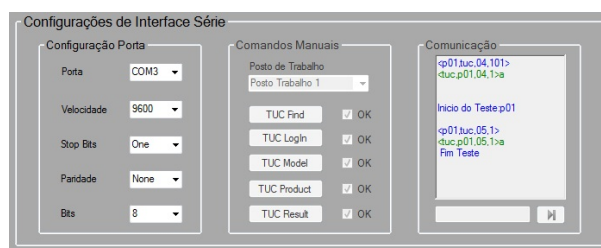


Figura 4.2: Software servidor TUC - Configurações de interface série

Porta”. Esta sub-secção só é editável quando o interface série está desligado, utilizando os botões de comando na secção “Interface Série TUC Servidor-Cliente”, abortados mais à frente.

A utilização das sub-secções “Comando Manuais” e “Comunicação” só são possíveis quando o interface série está ligado, ao contrário da sub-secção “Configuração Porta”. É possível ver na figura 4.2 o estado da secção “Configurações de Interface Série” com o interface desligado, deixando apenas acessível para edição a sub-secção “Configuração Porta”. A sub-secção “Comandos Manuais” serve, como o nome indica, para o envio de comandos manuais aos postos de trabalho. Nesta zona é possível escolher o posto de trabalho e enviar as mensagens TUC Find, TUC LogIn, TUC Model, TUC Product e TUC Result, através dos botões com os mesmos nomes. Para todas estas mensagens, é possível escolher se a resposta ao posto de trabalho é afirmativa ou não, através da selecção da caixa com a identificação OK, à direita de cada botão. As mensagens enviadas e recebidas são visualizadas sobre a forma de código na sub-secção “Comunicação”, identificando-as a azul as mensagens recebidas, e a verde as mensagens enviadas. Nesta janela é ainda possível enviar um comando completamente manual, utilizando a janela de escrita e o botão de envio que se encontram na parte inferior da sub-secção.

A secção “Interface Série TUC Servidor-Cliente” apresenta a informação relativa às mensagens enviadas e recebidas, sobre a forma de texto, facilmente interpretável por um utilizador. Nesta secção encontra-se, à direita da janela das mensagens, a zona onde se liga e desliga o interface série. Nas figuras 4.3 e 4.4 é possível ver os botões que são visíveis, dependendo do estado de ligação do interface, desligado na figura 4.3 e ligado na figura 4.4. Quando o servidor tem o interface desligado, o único botão visível é o botão para a sua ligação, que tem no seu interior um triângulo azul. Assim que o interface é ligado, esta secção apresenta outros três botões, o botão para desligar o controlo de traçabilidade, o botão para desligar o interface e o botão para o envio de um comando de arranque de inspecção da *smartcamera*, descritos pela ordem de cima para baixo.

Assim que o interface é ligado, o sistema de traçabilidade fica activo, aguardando a comunicação com um posto de trabalho. Para esta aplicação específica, e para facilitar a utilização, optou-se por colocar um botão nesta secção para o arranque de inspecção na *smartcamera*, dado que apenas se utiliza um equipamento que simula os vários postos de trabalho. Numa aplicação final, o comando para o arranque da inspecção seria no próprio posto de trabalho. O botão para desligar o sistema de traçabilidade serve para que se utilizem os comandos manuais da secção “Configuração de Interface Série”, e ensaiar manualmente as respostas aos postos de trabalho. Assim que pressionado, este botão fica intermitente, aparecendo novamente o botão de ligação da

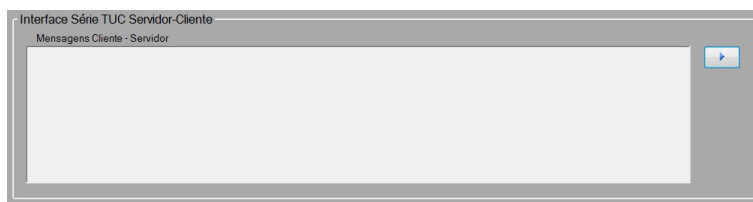


Figura 4.3: Software servidor TUC - Interface série servidor-cliente desligada

interface visível. Para se ligar novamente o controlo de traçabilidade, é necessário pressionar o botão de ligação do interface série, deixando os botões no mesmo estado em que se encontravam antes de ser interrompida.

As mensagens apresentadas na janela da secção “Interface Série TUC Servidor-Cliente” identificam sobre a forma de diferentes cores, as mensagens recebidas e enviadas, a abertura e fecho de sessão com um posto de trabalho, e as situações de erro. As mensagens recebidas são identificadas pela cor azul, começando por identificar o posto de trabalho que enviou a mensagem e qual o tipo de mensagem segundo o protocolo TUC. Após a recepção de uma mensagem, o servidor envia uma resposta ao posto de trabalho com a resposta à mensagem recebida, identificando o tipo de mensagem enviada e o resultado da resposta. Esta mensagem pode ter duas cores diferentes, dependendo do resultado da resposta, verde quando a resposta é afirmativa ou vermelha quando a resposta é negativa. Assim que um posto de trabalho é aceite pelo servidor, após o envio da mensagem TUC LogIn, é aberta uma sessão de trabalho no servidor. Esta sessão é identificada por uma mensagem de cor amarela na janela sendo finalizada assim que termina a comunicação com o posto de trabalho, após o envio da mensagem TUC Result, ou quando o servidor não permite um determinado produto num posto de trabalho.

Todas as mensagens identificam a hora, minuto, segundo e décimo de segundo a que foram executados, no início de cada mensagem. O tempo de execução do teste no posto de trabalho e da duração da sessão, são igualmente apresentados no corpo das respectivas mensagens, servindo apenas para efeito de controle por parte de um utilizador.

4.3 Tabela de Configuração TUC Modelos

Todos modelos e os postos de trabalho permitidos são introduzidos na tabela TUC Modelos, descrevendo igualmente a sequência de operação em cada posto de trabalho. Nesta tabela simplificou-se a introdução dos dados para que apenas fossem necessárias duas colunas, a coluna

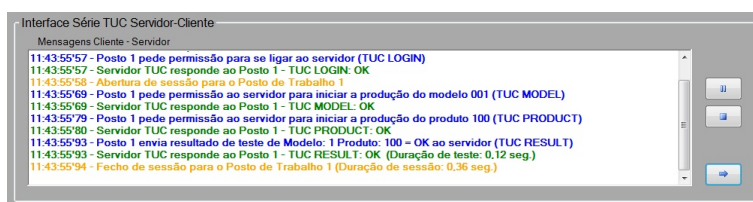


Figura 4.4: Software servidor TUC - Interface série servidor-cliente ligada

| Modelo | Sequencia |
|--------|-------------|
| 001 | p01-p02-p03 |
| 002 | p01 |
| | |
| | |
| | |

Figura 4.5: Software servidor TUC - Tabela TUC Modelos

“Modelo” e a coluna “Sequência”. Na coluna “Modelo” são introduzidos os diversos modelos de produtos aceites pelo sistema de traçabilidade, seguidos dos respectivos postos na coluna “Sequência” onde estes são produzidos. A coluna “Sequência” apresenta os postos de trabalho de forma sequencial, separados pelo caractere “-”. Tal como explicado anteriormente, os postos de trabalho são identificados por “p01”, “p02” e “p03” para os postos de trabalho um, dois e três respectivamente. Nesta aplicação específica, limitou-se a sequência de postos de trabalho até um máximo de três, sendo o mesmo número de postos de trabalho que a aplicação da *smartcamera* está preparada.

Na figura 4.5 é possível ver que na tabela TUC Modelos estão configurados dois modelos: 001 e 002. No caso do modelo 001, a sua montagem obriga à passagem por três postos de trabalho diferentes, primeiro pelo posto de trabalho um, seguido pelo número dois, e finalizando no posto de trabalho três. Para os produtos do modelo 002, a montagem é mais simples, sendo apenas necessária a passagem pelo posto de trabalho um. Os produtos de cada modelo são concluídos assim que a execução no último posto de trabalho for feita com sucesso, no caso do modelo 001 no posto de trabalho três, enquanto que no modelo 002 no posto de trabalho um. Desta forma, a tabela permite agrupar, apenas em duas colunas, todos os modelos e sequências de montagem nos postos de trabalho.

A edição da tabela é feita com o recurso ao botão no canto inferior esquerdo da secção “TUC Modelos”, deixando-a acessível para adicionar novos modelos ou fazer alterações. Assim que pressionado o botão de edição da tabela, aparecem dois novos botões no canto inferior direito, ambos para fechar a edição da tabela (figura 4.6). O botão da esquerda permite fechar a edição da tabela sem guardar as alterações efectuadas, enquanto que o da direita fecha a edição guardando as alterações.

4.4 Tabela TUC Traçabilidade

O controlo da traçabilidade de todos os produtos é acompanhada na tabela “TUC Traçabilidade”, onde é possível verificar o estado de cada produto, a sequência dos postos de trabalho, o resultado do teste em cada posto de trabalho e a hora de início e conclusão. Cada linha da tabela corresponde a um produto específico, sendo identificado pelo seu número de modelo e produto, nas colunas “Modelo” e “Produto”. À direita da coluna “Produto” encontra-se a coluna “Estado”,

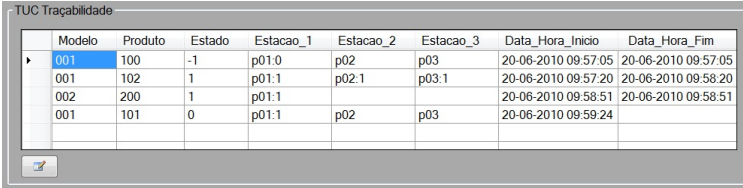
| Modelo | Sequencia |
|--------|-------------|
| 001 | p01-p02-p03 |
| 002 | p01 |
| | |

Figura 4.6: Software servidor TUC - Edição de tabela TUC Modelos

que identifica a fase de montagem do produto com três valores diferentes: “0” para um produto ainda em execução, “-1” para um produto concluído com um resultado mau, e “1” para um produto concluído com sucesso. Entende-se por um produto ainda em execução todos os produtos que passaram por pelo menos um posto de trabalho com resultado bom, que não tiveram nenhum resultado mau, e que ainda não terminaram o seu processo de fabrico. A sequência da montagem nos postos de trabalho é apresentada nas três colunas seguintes, “Estação 1”, “Estação 2” e “Estação 3”. Em cada uma destas células é apresentado o posto de trabalho e, caso já tenha sido executado, o resultado do teste. O posto de trabalho é apresentado pelos mesmos caracteres da tabela TUC Modelos seguido do caractere “:” e do resultado do teste, “1” para um teste bem sucedido e “0” para um teste mau, apenas quando o produto já foi submetido ao teste dessa estação. As duas últimas tabelas apresentam a data e hora de início e fim de produção de cada produto, nas colunas “Data Hora Início” e “Data Hora Fim”.

A tabela apresentada na figura 4.7 demonstra várias situações de um processo de fabrico, usando produtos de dois modelos diferentes, 001 e 002, definidos na tabela “TUC Modelos” da figura 4.5. Os produtos do modelo 001 são produzidos sequencialmente nos postos de trabalho um, dois e três, enquanto que os produtos do modelo 002 são produzidos unicamente no posto de trabalho um. Esta definição da sequência de postos de trabalho é convertida nas colunas “Estação 1”, “Estação 2” e “Estação 3”. Começando pela primeira linha, o produto 100 do modelo 001, reprovou no teste da estação um, ficando por isso concluído com insucesso. Este resultado é traduzido no valor “-1” na coluna estado, e pelo valor “:0” na coluna “Estação 1” após o nome do posto de trabalho. Após ter reprovado num teste, o produto fica etiquetado como mau e não pode ser testado novamente. Na segunda linha temos o exemplo do produto 102 do modelo 001, que terminou a sua execução em todos os postos de trabalho com o resultado bom, estando o seu ciclo de montagem terminado, identificado com o valor “1” na coluna “Estado”. A terceira linha mostra um produto modelo 002 concluído com sucesso, que apenas utiliza um posto de trabalho, deixando as colunas “Estação 2” e “Estação 3” vazias. A última linha mostra um produto que ainda está em execução, identificado com o valor “0” na coluna “Estado”, e que passou no teste da estação um.

Esta tabela, à imagem da tabela TUC Modelos, também pode ser editada com o recurso ao botão no canto inferior esquerdo da secção “TUC Traçabilidade”. Após a edição da tabela, os dois botões que aparecem no canto inferior direito, permitem guardar os novos dados ou ignorar



| Modelo | Produto | Estado | Estacao_1 | Estacao_2 | Estacao_3 | Data_Hora_Inicio | Data_Hora_Fim |
|--------|---------|--------|-----------|-----------|-----------|---------------------|---------------------|
| 001 | 100 | -1 | p01:0 | p02 | p03 | 20-06-2010 09:57:05 | 20-06-2010 09:57:05 |
| 001 | 102 | 1 | p01:1 | p02:1 | p03:1 | 20-06-2010 09:57:20 | 20-06-2010 09:58:20 |
| 002 | 200 | 1 | p01:1 | | | 20-06-2010 09:58:51 | 20-06-2010 09:58:51 |
| 001 | 101 | 0 | p01:1 | p02 | p03 | 20-06-2010 09:59:24 | |

Figura 4.7: Software servidor TUC - Tabela TUC Traçabilidade

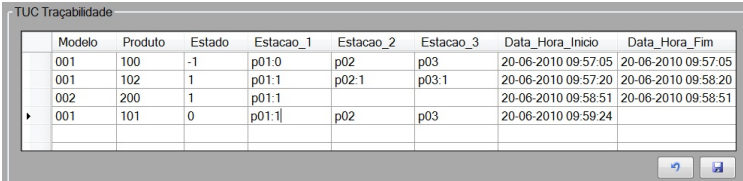
as alterações e recarregar os valores antigos (figura 4.8).

4.5 Conclusões

No decorrer do desenvolvimento desta aplicação, foram identificadas muitas possibilidades de melhoria com vista a tornar o servidor de traçabilidade o mais completo e abrangente possível. A *smartcamera* impôs, devido a limitações do próprio equipamento, que fosse usada uma comunicação série, colocando de parte a comunicação por mensagens sobre o protocolo TCP para esta aplicação. Contudo, o servidor não é limitado ao ponto de só ter uma única interface de comunicação, podendo-se adicionar outras ferramentas que o integrem numa topologia mais industrial, como são as comunicações *Ethernet*, *Profibus* e *Devicenet*.

A configuração dos modelos e dos postos de trabalho foi abordada com alguma simplicidade nesta aplicação, sendo suficiente em muitas situações reais, mas tornando-se limitada em alguns casos mais específicos. A possibilidade de existirem grupos de produtos dentro de um determinado modelo que tenham um tratamento diferente dos restantes, para situações de pequenas séries com características especiais, ou a possibilidade de existirem mais do que um posto de trabalho com as mesmas funções, não poderá ser feita numa tabela como esta, implicando a redefinição dos requisitos deste sistema.

O registo dos produtos na tabela de traçabilidade utilizou um número fixo de três estações de trabalho, um campo com o estado do produto e a hora de início e fim de produção, escritos num ficheiro de extensão XML. O formato do ficheiro utilizado tornou-se muito simples de trabalhar, tendo a facilidade de ser aberto num editor de texto ou mesmo num *browser*. No entanto, para a sua utilização num ambiente fabril e o registo de muita informação, deverá evoluir para um formato de base de dados com mais colunas, dando a possibilidade de guardar mais informação sobre o sistema.



| Modelo | Produto | Estado | Estacao_1 | Estacao_2 | Estacao_3 | Data_Hora_Inicio | Data_Hora_Fim |
|--------|---------|--------|-----------|-----------|-----------|---------------------|---------------------|
| 001 | 100 | -1 | p01:0 | p02 | p03 | 20-06-2010 09:57:05 | 20-06-2010 09:57:05 |
| 001 | 102 | 1 | p01:1 | p02:1 | p03:1 | 20-06-2010 09:57:20 | 20-06-2010 09:58:20 |
| 002 | 200 | 1 | p01:1 | | | 20-06-2010 09:58:51 | 20-06-2010 09:58:51 |
| 001 | 101 | 0 | p01:1 | p02 | p03 | 20-06-2010 09:59:24 | |

Figura 4.8: Software servidor TUC - Edição de tabela TUC Traçabilidade

O registo dos resultados dos produtos testados foi simplificado ao posto de existirem apenas produtos que tenham passado ou falhado no teste, ficando registados com o valor respectivo “1” e “0”. Há no entanto a necessidade de se registar valores específicos, nomeadamente em postos onde são efectuados testes, devendo ser considerada a configuração dos resultados dos testes para cada produto e posto de trabalho numa nova versão do servidor TUC. Desta forma, ao ser consultado um determinado produto produzido, será possível encontrar uma informação mais completa do que apenas o resultado final do teste, como por exemplo os valores com que este fez cada teste, os seus limites de aceitação e até o desvio para a média estatística.

O tempo de execução das mensagens entre o cliente e o servidor não era um requisito para o funcionamento do sistema, correspondendo neste exemplo a cerca do dobro do tempo do teste realizado pela *smartcamera*. Numa futura versão do software, poderá no entanto ser optimizado de forma a ter o menos impacto possível no tempo de ciclo do posto de trabalho. A evolução poderá acontecer igualmente no conteúdo das mensagens enviadas, havendo a possibilidade de serem expandidas a nível de parâmetros e conter outras informações relevantes que tornem o sistema mais completo.

Capítulo 5

Ensaio Práticos

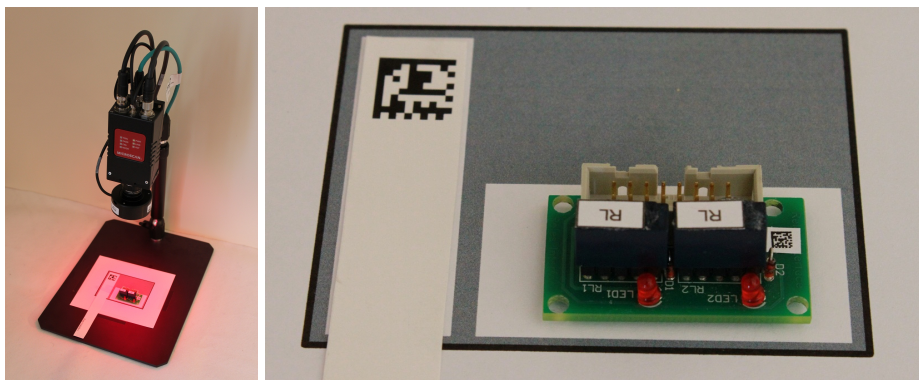
Os testes de simulação dos postos de trabalho foram feitos com uma *smartcamera*, algumas placas PCB com componentes electrónicos, e etiquetas para a identificação do posto de trabalho. Com este conjunto simulou-se o funcionamento de três postos de trabalho distintos com uma única *smartcamera*, fazendo simplesmente a troca da etiqueta com o número do posto de trabalho.

5.1 Posto de Trabalho e Equipamento Utilizado

Uma linha de produção pressupõe que a montagem de um determinado produto evolui ao longo dum processo de fabrico, tornando-se mais completo e, normalmente com mais componentes. Para a simulação de uma situação idêntica, neste trabalho utilizou-se um PCB como o produto que iria evoluir ao longo de três postos de trabalho, todos eles formados pela mesma *smartcamera* (ver figura 5.1a). Este equipamento pretende fazer operações distintas para cada um dos postos de trabalho escolhidos, sendo utilizada a mesma *smartcamera* como forma de simplificar o teste.

A área inspeccionada pela *smartcamera* é composta por duas zonas distintas, a zona de teste de produto e a zona de identificação do posto de trabalho, tal como falado anteriormente no capítulo 3 (figura 3.4). Nestas duas áreas deverá ser colocado o PCB na zona de inspeção e uma etiqueta rectangular sobre a zona de identificação do posto de trabalho (figura 5.1b). A etiqueta rectangular tem um código *Datamatrix* na extremidade, que permite à *smartcamera* identificar o número do posto de trabalho que será processado. Em função desta etiqueta a *smartcamera* irá assumir a execução de um posto de trabalho, alterando o formato das mensagens enviadas ao servidor TUC, e o teste do produto.

O PCB utilizado é composto por diversos componentes electrónicos, entre eles dois relés, identificados por relé 1 e por relé 2, e um código *Datamatrix* que identifica a peça com um número de modelo e número de produto (figura 5.2). Ao ser iniciada a inspeção da peça, a leitura do código de identificação e a presença dos relés é feita assim que é identifica a posição da peça dentro da zona de teste do produto. A resultado da inspeção irá depender do posto de trabalho em execução e da presença dos relés, conforme se pode verificar na tabela 5.1.



(a) Smartcamera e posto de trabalho utilizado

(b) Zona de teste

Figura 5.1: Equipamento utilizado no trabalho

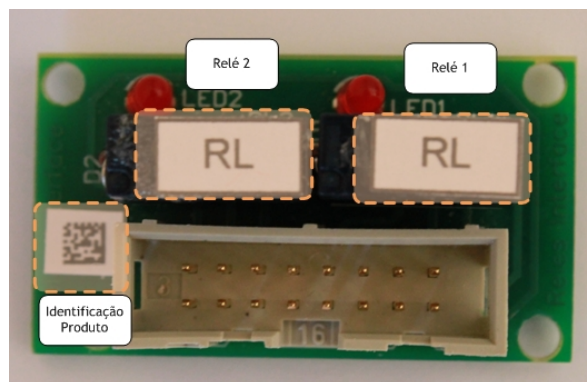


Figura 5.2: Produto utilizado para o teste

| Número do Posto de Trabalho | Detecção Relé 2 | Detecção Relé 1 |
|-----------------------------|-----------------|-----------------|
| Posto de Trabalho 1 | não | não |
| Posto de Trabalho 2 | não | sim |
| Posto de Trabalho 3 | sim | sim |

Tabela 5.1: Testes efectuados nos postos de trabalho

5.2 Ensaio de Controlo de Traçabilidade

Para que fosse possível ensaiar diversos cenários no controlo da traçabilidade, foram usados alguns PCB com identificações diferentes, onde eram montados e desmontados os relés de acordo com o teste que era pretendido fazer. Desta forma, foi possível criar várias situações para os quais o servidor TUC teria de estar preparado. O diálogo entre o cliente e o servidor terá sempre uma pergunta e uma resposta, cabendo sempre ao cliente iniciar o diálogo com a formatação da pergunta, neste caso, a *smartcamera*. De acordo com a pergunta formulada e com os dados disponíveis sobre o cliente, o servidor irá permitir a continuação do diálogo ou a sua interrupção. Este diálogo é facilmente interpretado nas mensagens que aparecem na secção “Interface Série TUC Servidor-Cliente” na aplicação do servidor TUC (figuras 4.3 e 4.4), que será utilizado nos pontos seguintes como exposição de alguns exemplos.

Verificação de Posto de Trabalho - TUC LogIn

O comando TUC LogIn pode terminar uma comunicação entre o servidor e o cliente quando o posto de trabalho do cliente não está configurado na tabela de postos de trabalho disponíveis, tabela TUC Modelos. Nesta situação o servidor não chega a abrir a sessão com o posto de trabalho.

Como exemplo são apresentadas as diversas mensagens entre o posto de trabalho 4 e o servidor TUC:

```
11:49:44'03 – Posto 4 faz consulta de disponibilidade ao servidor (TUC FIND)
11:49:44'04 – Servidor TUC responde ao Posto 4 – TUC FIND: OK
11:49:44'17 – Posto 4 pede permissão para se ligar ao servidor (TUC LOGIN)
11:49:44'18 – Servidor TUC responde ao Posto 4 – TUC LOGIN: NOK
11:49:44'18 – ERRO: Posto de trabalho não disponível no servidor TUC
```

Verificação do Modelo - TUC Model

Após a validação com o comando TUC LogIn, o servidor abre uma sessão com o posto de trabalho e verifica o código do modelo assim que é enviada a mensagem TUC Model. O código do modelo que se pretende testar pode no entanto não estar relacionado com o posto de trabalho, terminando a sessão entre o cliente e o servidor e impedindo o cliente de prosseguir o teste desse equipamento.

Para exemplificar esta situação, são apresentadas as mensagens entre o posto de trabalho 4 e o servidor, durante a execução de um teste a um produto do modelo 001.

```
11:53:24'39 – Posto 4 faz consulta de disponibilidade ao servidor (TUC FIND)
11:53:24'40 – Servidor TUC responde ao Posto 4 – TUC FIND: OK
11:53:24'58 – Posto 4 pede permissão para se ligar ao servidor (TUC LOGIN)
11:53:24'59 – Servidor TUC responde ao Posto 4 – TUC LOGIN: OK
11:53:24'59 – Abertura de sessão para o Posto de Trabalho 4
11:53:24'70 – Posto 4 pede permissão ao servidor para iniciar a produção do modelo
001 (TUC MODEL)
11:53:24'71 – Servidor TUC responde ao Posto 4 – TUC MODEL: NOK
```

11:53:24'71 – ERRO: Modelo 001 não disponível para o Posto de Trabalho 4
11:53:24'71 – Fecho de sessão para o Posto de Trabalho 4 (Duração de sessão: 0,11 seg.)

Verificação do Produto - TUC Product e TUC Result

Assim que o posto de trabalho e o código do modelo são validados, é aguardada a recepção do código do produto para que se possa verificar o seu estado, com o comando TUC Product. Esta verificação é mais complexa pois são possíveis diversas situações de erro, tais como a verificação do último posto de trabalho, a correcta sequência de montagem e o estado de conclusão do produto.

Quando é terminada a execução do teste de forma normal, o produto pode ou não ter passado no teste desse posto de trabalho e mediante isso, o servidor irá guardar os dados desse produto para que fique disponível para posteriores consultas.

O exemplo seguinte, mostra a execução do modelo 001 e do produto 100 no posto de trabalho 1. O produto passou no teste do posto de trabalho.

11:57:34'82 – Posto 1 faz consulta de disponibilidade ao servidor (TUC FIND)
11:57:34'82 – Servidor TUC responde ao Posto 1 – TUC FIND: OK
11:57:34'96 – Posto 1 pede permissão para se ligar ao servidor (TUC LOGIN)
11:57:34'96 – Servidor TUC responde ao Posto 1 – TUC LOGIN: OK
11:57:34'96 – Abertura de sessão para o Posto de Trabalho 1
11:57:35'08 – Posto 1 pede permissão ao servidor para iniciar a produção do modelo 001 (TUC MODEL)
11:57:35'08 – Servidor TUC responde ao Posto 1 – TUC MODEL: OK
11:57:35'18 – Posto 1 pede permissão ao servidor para iniciar a produção do produto 100 (TUC PRODUCT)
11:57:35'18 – Servidor TUC responde ao Posto 1 – TUC PRODUCT: OK
11:57:35'31 – Posto 1 envia resultado de teste de Modelo: 001 Produto: 100 = OK ao servidor (TUC RESULT)
11:57:35'31 – Servidor TUC responde ao Posto 1 – TUC RESULT: OK (Duração de teste: 0,12 seg.)
11:57:35'32 – Fecho de sessão para o Posto de Trabalho 1 (Duração de sessão: 0,35 seg.)

No exemplo seguinte, a situação é idêntica à anterior, com a diferença do produto ter reprovado no teste do posto de trabalho. Além da diferença da mensagem com o resultado do teste, nesta situação é apresentada uma nova linha, indicando que o produto foi concluído com um resultado mau ou NOK. Isto significa que este o produto não poderá entrar novamente em nenhum posto de trabalho.

12:36:42'97 – Posto 1 faz consulta de disponibilidade ao servidor (TUC FIND)
12:36:42'98 – Servidor TUC responde ao Posto 1 – TUC FIND: OK
12:36:43'11 – Posto 1 pede permissão para se ligar ao servidor (TUC LOGIN)
12:36:43'12 – Servidor TUC responde ao Posto 1 – TUC LOGIN: OK
12:36:43'12 – Abertura de sessão para o Posto de Trabalho 1
12:36:43'24 – Posto 1 pede permissão ao servidor para iniciar a produção do modelo 001 (TUC MODEL)

12:36:43'24 – Servidor TUC responde ao Posto 1 – TUC MODEL: OK
12:36:43'35 – Posto 1 pede permissão ao servidor para iniciar a produção do produto 100 (TUC PRODUCT)
12:36:43'35 – Servidor TUC responde ao Posto 1 – TUC PRODUCT: OK
12:36:43'48 – Posto 1 envia resultado de teste de Modelo: 001 Produto: 100 = NOK ao servidor (TUC RESULT)
12:36:43'48 – Servidor TUC responde ao Posto 1 – TUC RESULT: OK (Duração de teste: 0,12 seg.)
12:36:43'49 – Modelo: 001 – Produto: 100 concluído com resultado NOK
12:36:43'49 – Fecho de sessão para o Posto de Trabalho 1 (Duração de sessão: 0,37 seg.)

Quando um produto passa sem problemas pelos diversos postos de trabalho até atingir o último, a sequência de mensagens apresenta, tal como no exemplo anterior, uma linha com a indicação de que o produto foi concluído. No exemplo seguinte, o produto 100 do modelo 001 terminou a sua montagem no posto de trabalho 3 com sucesso, ou OK.

12:40:04'96 – Posto 3 faz consulta de disponibilidade ao servidor (TUC FIND)
12:40:04'97 – Servidor TUC responde ao Posto 3 – TUC FIND: OK
12:40:05'09 – Posto 3 pede permissão para se ligar ao servidor (TUC LOGIN)
12:40:05'09 – Servidor TUC responde ao Posto 3 – TUC LOGIN: OK
12:40:05'10 – Abertura de sessão para o Posto de Trabalho 3
12:40:05'21 – Posto 3 pede permissão ao servidor para iniciar a produção do modelo 001 (TUC MODEL)
12:40:05'21 – Servidor TUC responde ao Posto 3 – TUC MODEL: OK
12:40:05'31 – Posto 3 pede permissão ao servidor para iniciar a produção do produto 100 (TUC PRODUCT)
12:40:05'32 – Servidor TUC responde ao Posto 3 – TUC PRODUCT: OK
12:40:05'45 – Posto 3 envia resultado de teste de Modelo: 001 Produto: 100 = OK ao servidor (TUC RESULT)
12:40:05'45 – Servidor TUC responde ao Posto 3 – TUC RESULT: OK (Duração de teste: 0,13 seg.)
12:40:05'45 – Modelo: 001 – Produto: 100 concluído com resultado OK
12:40:05'45 – Fecho de sessão para o Posto de Trabalho 3 (Duração de sessão: 0,35 seg.)

O sistema de traçabilidade garante não só que é respeitada a sequência de montagem como a verificação do estado do produto. O pedido de entrada de um produto pelo comando TUC Product pode identificar que o produto já se encontra concluído, ou que o produto pretende repetir um teste no mesmo posto de trabalho ou que não está a ser respeitada a sequência de montagem definida na tabela TUC Modelos. Todas estas situações fazem terminar a sessão com o posto de trabalho com a falha no comando TUC Product.

As mensagens seguintes, apresentam a situação de um produto que repete um posto de trabalho ou que tenta desrespeitar a sequência de montagem definida.

12:08:19'14 – Posto 1 faz consulta de disponibilidade ao servidor (TUC FIND)
12:08:19'15 – Servidor TUC responde ao Posto 1 – TUC FIND: OK
12:08:19'27 – Posto 1 pede permissão para se ligar ao servidor (TUC LOGIN)
12:08:19'28 – Servidor TUC responde ao Posto 1 – TUC LOGIN: OK

12:08:19'28 – Abertura de sessão para o Posto de Trabalho 1
12:08:19'39 – Posto 1 pede permissão ao servidor para iniciar a produção do modelo 001 (TUC MODEL)
12:08:19'39 – Servidor TUC responde ao Posto 1 – TUC MODEL: OK
12:08:19'50 – Posto 1 pede permissão ao servidor para iniciar a produção do produto 100 (TUC PRODUCT)
12:08:19'50 – Servidor TUC responde ao Posto 1 – TUC PRODUCT: NOK
12:08:19'51 – ERRO: Modelo 001 – Produto 100 ainda não pode ser testado ou já foi testado neste Posto de Trabalho.
12:08:19'51 – Fecho de sessão para o Posto de Trabalho 1 (Duração de sessão: 0,22 seg.)

A verificação do estado do produto é fundamental para que se saiba que o produto ainda está em processo de montagem. No exemplo seguinte, o produto 100 do modelo 001 tentava entrar no posto de trabalho 3 mesmo após ter sido concluído com sucesso, ou OK.

12:32:36'64 – Posto 3 faz consulta de disponibilidade ao servidor (TUC FIND)
12:32:36'65 – Servidor TUC responde ao Posto 3 – TUC FIND: OK
12:32:36'78 – Posto 3 pede permissão para se ligar ao servidor (TUC LOGIN)
12:32:36'78 – Servidor TUC responde ao Posto 3 – TUC LOGIN: OK
12:32:36'79 – Abertura de sessão para o Posto de Trabalho 3
12:32:36'89 – Posto 3 pede permissão ao servidor para iniciar a produção do modelo 001 (TUC MODEL)
12:32:36'90 – Servidor TUC responde ao Posto 3 – TUC MODEL: OK
12:32:37'00 – Posto 3 pede permissão ao servidor para iniciar a produção do produto 100 (TUC PRODUCT)
12:32:37'01 – Servidor TUC responde ao Posto 3 – TUC PRODUCT: NOK
12:32:37'01 – ERRO: Modelo 001 – Produto 100 já foi concluído com o resultado OK.
12:32:37'01 – Fecho de sessão para o Posto de Trabalho 3 (Duração de sessão: 0,22 seg.)

Uma situação idêntica à anterior, acontece com um produto concluído com um resultado mau, ou NOK. No exemplo seguinte, o produto 100 do modelo 001 faz a tentativa para realizar o teste no posto de trabalho 3 após ter sido concluído com o resultado NOK.

12:34:15'31 – Posto 3 faz consulta de disponibilidade ao servidor (TUC FIND)
12:34:15'31 – Servidor TUC responde ao Posto 3 – TUC FIND: OK
12:34:15'44 – Posto 3 pede permissão para se ligar ao servidor (TUC LOGIN)
12:34:15'44 – Servidor TUC responde ao Posto 3 – TUC LOGIN: OK
12:34:15'44 – Abertura de sessão para o Posto de Trabalho 3
12:34:15'55 – Posto 3 pede permissão ao servidor para iniciar a produção do modelo 001 (TUC MODEL)
12:34:15'56 – Servidor TUC responde ao Posto 3 – TUC MODEL: OK
12:34:15'65 – Posto 3 pede permissão ao servidor para iniciar a produção do produto 100 (TUC PRODUCT)
12:34:15'66 – Servidor TUC responde ao Posto 3 – TUC PRODUCT: NOK
12:34:15'66 – ERRO: Modelo 001 – Produto 100 já foi concluído com o resultado NOK.
12:34:15'67 – Fecho de sessão para o Posto de Trabalho 3 (Duração de sessão: 0,21 seg.)

5.3 Conclusões

A utilização dos PCB no teste da *smartcamera* teve o intuito de criar diversas situações reais e verificar a eficácia do sistema de controlo de traçabilidade. A *smartcamera* foi convertida em três postos de trabalho utilizando simples etiquetas, inspeccionando os relés do PCB de acordo com o posto de trabalho configurado. A possibilidade de simular falhas na montagem do produto, postos de trabalho não configurados no servidor e produtos não permitidos foi muito simples, consistindo na colocação e remoção dos relés nos PCB, alterando as etiquetas dos postos de trabalho e usando vários PCB com identificações diferentes. Com os ensaios efectuados foi possível concluir que as características fundamentais de um sistema de gestão de traçabilidade foram verificadas, deixando aberta a possibilidade de se evoluir para um controlo mais completo e eficaz, numa nova versão de software.

Capítulo 6

Conclusões e Trabalho Futuro

O estudo e desenvolvimento de uma aplicação de controlo de traçabilidade foi um projecto aliciante e até um pouco ambicioso, tendo em conta o tempo disponível para a sua realização. O uso de uma *smartcamera* como equipamento com a finalidade de testar o funcionamento do sistema foi um desafio pois, as funcionalidades desta a nível de comunicações e gestão de mensagens são um pouco limitadas quando comparadas com as ferramentas que esta dispõe para o controlo de qualidade visual de objectos. Estes problemas foram ultrapassados com o recurso a funções programadas em linguagem de baixo nível e diversos ciclos de execução paralelos, provando ser muito flexível na programação. Após a identificação das limitações no controlo da porta *Ethernet* da *smartcamera* com programação de baixo nível, impostas pelo seu hardware e software, contornou-se este problema recorrendo à sua porta série para a comunicação com o sistema de gestão de traçabilidade. Desta forma, embora se tenha utilizado uma rede *Ethernet* para a comunicação com a *smartcamera*, as mensagens são enviadas por um conversor que simula uma porta série no servidor, o que não corresponde directamente a uma comunicação com protocolo de mensagens TCP/IP como era pretendido. A utilização deste equipamento provou ter muito mais funcionalidades do que apenas a inspecção visual de componentes, dispensando em muitas situações a interligação a um autómato programável ou um computador. O protocolo de comunicação utilizado no sistema foi alvo de diversas evoluções ao longo do trabalho, tendo sempre por objectivo a construção de um código de simples interpretação, que pudesse facilitar o envio em modo manual durante os testes com a *smartcamera*. Os cinco comandos disponíveis no protocolo foram suficientes para a construção deste sistema de controlo de traçabilidade, deixando-o expansível ou mesmo utilizável noutras aplicações diferentes. O software desenvolvido e instalado no servidor TUC, demonstrou ser possível resolver os principais problemas de traçabilidade num processo fabril com o recurso a apenas duas tabelas simples. Esta abordagem foi no entanto simplificada ao ponto de limitar a montagem de um produto no máximo a três postos de trabalho, e do resultado do teste ter apenas dois estados, bom e mau. Os dados expostos nas tabelas utilizaram ficheiros XML com os campos de informação indispensável, como o estado de conclusão do produto, a hora de início e fim de produção, e o número do modelo e produto que o caracterizam. O formato deste ficheiro foi escolhido devido à simplicidade da sua edição e manipulação de dados, estando

a aplicação preparada para a utilização de um ficheiro de base de dados sem ter de fazer alterações significativas no programa. Os testes efectuados ao sistema com os PCB, permitiram por à prova as capacidades do controlo de traçabilidade, ao criar diversas situações reais de montagem e erros.

Foram várias as ideias que surgiram durante a realização do trabalho, devendo ser amadurecidas e debatida antes de serem postas em prática, numa futura versão do software. Começando pela flexibilidade deixada pelo protocolo de comunicação para outras utilizações não relacionadas com a traçabilidade. Surge muitas vezes a necessidade de sincronizar postos de trabalho, sendo necessária uma ligação entre eles para a troca de sinais. Se ambos os postos de trabalho estiverem ligados à rede de comunicação de traçabilidade, é possível desenvolver novas funções para esse fim, onde no corpo da mensagem enviada seria utilizado o número do posto de trabalho com que se quer comunicar, em vez do servidor TUC. O mesmo acontece com o servidor, podendo ser utilizado mais do que um servidor para descentralizar o armazenamento de dados ou fazer redundância de informação, sendo neste caso usado o nome do segundo servidor no corpo da mensagem, em vez do número do posto de trabalho. O servidor TUC poderá evoluir em muitas direcções diferentes, desde a forma como os dados são armazenados, à informação recolhida nos postos de trabalho, passando pelos múltiplos interfaces possíveis, até à utilização de diferentes ambientes gráficos para vários tipos de utilizadores. Existe por isso muitas áreas de desenvolvimento que foram identificadas mas muito pouco exploradas, que possibilitam que uma aplicação deste género tenha um conteúdo ainda mais abrangente e completo sobre o sistema envolvente.

As oportunidades encontradas deixam uma perspectiva de se criar uma plataforma de controlo de traçabilidade *standard* comercializável. A possibilidade de utilizar mais do que um único interface, alarga o seu campo de aplicação num ambiente industrial, onde se podem encontrar nos postos de trabalho múltiplas marcas de autómatos programáveis e distintas plataformas de desenvolvimento em computadores. A expansão para redes de comunicação como *Ethernet*, *Profibus* e *Devicenet*, suportado por blocos de software que permitam a sua comunicação em diferentes marcas de produtos e plataformas de software de desenvolvimento, serviria como um forte argumento para a sua introdução no mercado.

Bibliografia

- [1] Microscan. *Visionscape AppRunner User Manual*, v4.1.0 edition, February 2010.
- [2] Microscan. *Visionscape FrontRunner User Manual*, v4.1.0 edition, February 2010.
- [3] Microscan. *Visionscape .NET Programmers Manual*, v4.1.0 edition, February 2010.
- [4] Microscan. *Visionscape Perl Script Custom Tool Programmers Manual*, v4.1.0 edition, February 2010.
- [5] Microscan. *Visionscape Tools Reference Manual*, v4.1.0 edition, February 2010.
- [6] Microscan. *VS-1 Smart Camera Guide*, v4.1.0 edition, February 2010.
- [7] Quality and Test Systems. *Based (Winsock) Device Communication*. FORD Plant Floor Systems, v1.1 edition, June 2007.
- [8] Information Systems. *Socket Interface Specification*. FORD Electronics Division, v0.1 edition, August 1993.
- [9] Information Systems. *Standard Message Format CIM*. FORD Electronics Division, v1.2 edition, June 2007.