

RECYT

Año 16 / Nº 22 / 2014 / 32–39#

Integración entre aplicaciones comerciales e industriales a través del estándar COM en un prototipo

Integration among commercial and industrial applications through the standard COM on a prototype

Integração entre aplicativos comerciais e industriais através do padrão COM em um protótipo

Júlio César Calegari¹, José de Souza^{2,*}, Lirio Schaeffer²

1 - Facultades Integradas de Taquara, FACCAT, Taquara, RS, Brasil.

2 - Programa de Posgrado en Ingeniería de Minas, Metalurgia y Materiales PPGE3M, Escuela de Ingeniería de la Universidad Federal de Rio Grande del Sur, UFRGS, Brasil.

* E-mail: josesouza@liberato.com.br

Resumen

Las aplicaciones de *software* usadas en el área comercial no son desarrolladas, en su mayoría, con el objetivo de cambiar informaciones con dispositivos industriales. Sus recursos nativos no permiten el intercambio de datos en la mayor parte de los casos. La comunicación entre diferentes aplicaciones puede ayudar en la automatización y en el control de determinados procesos. Este trabajo se relaciona con el área de automatización industrial y se ejecutó con el objetivo de desarrollar una aplicación para añadir recursos al Office Excel, integrando al mismo los protocolos de comunicación usados en redes industriales. Se desarrolló la aplicación con el fin de realizar la conversión entre estándares de comunicación, administrando las informaciones entre dispositivos industriales y plantillas del Office Excel, creando una nueva opción para monitorear procesos en la industria.

Palabras clave: Automatización; Supervisorio; Aplicación; Integración; Controladores lógicos programables.

Abstract

Software applications used in the commercial area are mostly not developed with the aim of exchanging information with industrial devices. Native resources do not allow the exchange of data in most cases. The communication between different applications can help control and automation of certain processes. This work is related to the industrial automation area. It was executed with the goal of developing an application to add features to Office Excel, integrating those communication protocols used in industrial networks. The application was developed in order to perform the conversion between standard communications, managing information between industrial devices and Office Excel spreadsheets, creating a new option to monitor processes in the industry.

Keywords: Automation; Supervisory; Application; Integration; Programmable logic controllers.

Resumo

Os aplicativos de software usados na área comercial, em sua grande maioria, não são desenvolvidos com o objetivo de trocar informações com dispositivos industriais. Seus recursos nativos não permitem o intercâmbio de dados na maioria dos casos. A comunicação entre diferentes aplicativos pode auxiliar a automação e controle de determinados processos. Este trabalho está relacionado à área de automação industrial, foi executado com o objetivo de desenvolver um aplicativo para agregar recursos ao Office Excel, integrando a este os protocolos de comunicação usados em redes industriais. O aplicativo foi desenvolvido com a finalidade de realizar a conversão entre padrões de comunicação, gerenciando as informações entre dispositivos industriais e planilhas do Office Excel, criando uma nova opção para monitorar processos na indústria.

Palavras-chave: Automação; Supervisorio; Aplicativo; Integração; Controladores lógicos programáveis.

Introdução

Nos sistemas de automação a comunicação com os mais variados tipos de dispositivos para aquisição de dados e controle de processos é muito comum. O mercado está repleto de diversas tecnologias e seus respectivos fabricantes, com diferentes protocolos de rede. Atualmente existem vários protocolos de rede usados para a comunicação entre controladores e elementos de campo, sendo que cada fabricante possui seus próprios padrões na implementação dos protocolos usados em seus equipamentos [1].

As ferramentas existentes e comercializadas no mercado exigem do desenvolvedor uma afinidade em áreas complexas como a comunicação de dados, integrando-o com uma infinidade de protocolos usados em redes industriais. Amenizar o grau de dificuldade, desenvolvendo um aplicativo amigável que disponibilize as informações sem grande esforço por parte do usuário, irá tornar esta tarefa menos árdua para o desenvolvedor [1, 2].

Este trabalho foi realizado com o intuito de aproveitar os recursos do programa Office Excel, integrando ao software protocolos de comunicação usados em redes industriais. Cabe esclarecer que o sistema proposto não tem garantias de robustez e confiabilidade que requer um sistema de controle, uma vez que utiliza tanto um sistema operacional e aplicações e desenvolvimentos que não são certificados ou testados. Os testes realizados deram-se a partir de um protótipo em laboratório. Isto foi executado por meio de outro programa capaz de fazer a conversão entre padrões de comunicação diferentes, gerenciando as informações entre dispositivos industriais e planilhas do Office Excel. Esta ação visa construir uma alternativa para monitorar processos na indústria. O objetivo é criar um aplicativo de configuração fácil, amigável e intuitivo ao usuário, uma ferramenta de interface simples, permitindo que este possa se dedicar mais a parte visual do aplicativo SCADA, sem se incomodar com o gerenciamento de protocolos de comunicação e aquisição de dados em dispositivos de *hardware*. Além disso, buscou-se possibilitar que recursos visuais e funcionais de alto nível, existentes em *softwares* como o Office Excel pudessem ser aproveitados para fins de automação na área da indústria, sem a necessidade de grandes configurações por parte do usuário.

Para realizar os testes e a implementação da troca de dados entre o CLP (*Programmable Logic Controller*) e a planilha do Office Excel, utilizou-se o protocolo de comunicação Modbus TCP. Este será disponibilizado como DLL (*Dynamic-link library*), parte de *software*, neste caso, um módulo externo com funções de acesso aos dados no equipamento monitorado, esta por sua vez, irá prover o suporte ao protocolo de comunicação implementado. Este artigo apresenta os resultados obtidos no desenvolvimento de um aplicativo capaz de interagir com protocolos usados em redes industriais, repassando a informação para *softwares* comerciais que possuem suporte ao padrão COM (*Compo-*

nent Object Model) em um protótipo laboratorial construído para testes com o Office Excel e utilização de interface de visualização e operação de sistemas supervisórios.

Controladores Programáveis

Na indústria, uma infinidade de equipamentos utilizados em plantas de automação são controlados através de CLP's (Controladores Lógico Programáveis), trata-se de um dispositivo eletrônico baseado em microprocessador programável pelo usuário [1]. Estes dispositivos são providos de memória programável para guardar instruções lógicas de programação feitas e compiladas em linguagens computacionais desenvolvidas de acordo com as características do seu microprocessador [2]. Os dados de entrada são analisados de forma lógica por uma linguagem específica do microprocessador, este por sua vez, gera uma saída atuando nos equipamentos periféricos ao equipamento de CLP instalado [1-3].

Software Unity

Uma ferramenta utilizada no desenvolvimento de aplicativos de *software* para CLP's está disponível como aplicativo de configuração e programação que integra múltiplos modelos de CLP's. O Unity é um *software* de programação comum para as linhas de CLP Modicon Premium, Atrium e Quantum. O mesmo disponibiliza os cinco idiomas de programação exigidos pela norma IEC61131-3, que é a terceira de cinco partes que se divide a norma IEC61131, que se destina a padronizar as linguagens de programação para CLP's [4].

Redes Industriais

As arquiteturas de padrões em redes industriais destinam-se a integrar os componentes de um sistema de automação e são caracterizadas em função do protocolo utilizado [5]. Seguindo as teorias de redes industriais [5], são os diferentes protocolos de comunicação utilizados na troca de informação entre os equipamentos que envolvem uma planta automatizada.

São muitas as tecnologias para transmissão de dados em redes industriais, entre as mais conhecidas estão o Modbus RTU, Modbus TCP, Profibus, CANopen e DeviceNet. A grande vantagem da utilização de redes está em ligar muitos instrumentos no campo (chão de fábrica) com uma quantidade mínima de cabos. Entre todas as tecnologias associadas ao controle industrial, as redes de comunicação industriais são as que mais sofreram evoluções na última década [6-7].

Protocolo Modbus

O protocolo Modbus [8] é baseado no paradigma de comunicação mestre-escravo, onde o mestre é o responsável por coordenar a comunicação entre os escravos que compõem a rede. Possui uma série de variações para diferentes tipos de barramento industrial. Durante a comunicação e posterior troca de dados [9], a checagem de erro é feita pela conferência CRC (*Cyclic redundancy check*) em modo RTU e LCR (*Longitudinal Redundancy Check*) em modo ASCII. Por segurança o mestre envia a mensagem ao escravo e aguarda um determinado tempo para gerar uma exceção de erro. Caso o escravo não responda dentro da faixa de tempo estipulada, a exceção é gerada e o erro pode ser tratado conforme o nível de sua gravidade na aplicação. Um exemplo está disponível em www.infopl.net/files/descargas/wonderware/infopl_net_controldeprocesos.pdf.

O protocolo Modbus é encapsulado no protocolo TCP/IP e transmitido através de redes padrão ethernet¹ [6]. Trata-se de um protocolo mestre escravo, porém ao ser encapsulado no protocolo TCP/IP, esta nova variante do Modbus, acaba viabilizando a implantação de vários mestres e escravos em um mesmo barramento. Em Modbus TCP não é utilizada a verificação de erros convencional do protocolo Modbus. Este deixa a verificação de erros a cargo do protocolo TCP/IP [10].

O Padrão COM

É um padrão orientado a objetos, estabelecido pela Microsoft, define o modo como um objeto pode ser usado por outros, assim como estes podem interagir entre processos diferentes na troca de informações, ou seja, é um padrão usado para que uma aplicação cliente possa acessar os serviços de uma aplicação servidora [3].

Através deste padrão é possível integrar aplicações distintas, desenvolvidas em diferentes plataformas, que integram à tecnologia COM. O padrão permite a comunicação entre processos paralelos, onde está baseada a tecnologia OLE (*Object Linking and Embedding*).

O termo COM usado no desenvolvimento de ferramentas de *software* e se refere a um grupo de tecnologias que incluem OLE, OLE Automation, COM e DCOM (*Decentralized Component Object Model*) [2]. O DCOM é uma extensão do COM, ideal para ser usado em sistemas distribuídos, permite a comunicação entre aplicações rodando em máquinas distintas e provê os mesmos serviços do COM, porém utiliza como meio de acesso uma rede de comunicação de dados [3]. Um objeto COM é um código

1- O padrão IEEE 802.3, mais conhecido como Ethernet, é uma rede de difusão de barramento com controle, descentralizado, em geral operando em velocidades de 10Mbps a 10Gbps. Os computadores em uma rede Ethernet podem transmitir sempre que necessário, se dois ou mais pacotes colidirem, cada computador aguardará um tempo aleatório e fará uma nova tentativa mais tarde.

binário, para ser utilizado por outros programas, independente da linguagem em que foram desenvolvidos. O COM possui as seguintes funcionalidades [2]:

- Criar objetos que podem ser usados por várias linguagens de programação;
- Criar controles ActiveX;
- Controlar programas através da automação OLE, como Office Word e o Office Excel;
- Trabalhar com objetos ou programas em outras máquinas (DCOM).

Aplicativos SCADA

Os sistemas SCADA (*Supervisory Control and Data Acquisition*) são desenvolvidos para atuarem como interface entre o homem e a máquina (IHM). Compreende um meio onde se traduz o que está acontecendo na máquina ou no sistema automatizado, ou melhor, um dispositivo que está entre o homem e a máquina [3]. Supervisórios são encontrados em estações locais ou remotas de processos industriais. Por meio do sistema de supervisão o operador pode interagir com o processo a ser controlado, observando, analisando e intervindo quando necessário nas variáveis de processo. O aplicativo SCADA pode atuar na coleta dados, valores dos equipamentos em campo, calculando, modificando e elaborando relatórios para futuras tomadas de decisão do operador [12]. Estes aplicativos são desenvolvidos para obter as informações de um processo, tratá-las quando necessário e disponibilizá-las ao usuário, em muitos processos o sistema de supervisão pode interagir de forma autônoma e tomar decisões pré-programadas em andamento. Disponível em www.infopl.net/files/descargas/wonderware/infopl_net_controldeprocesos.pdf. Algumas ferramentas de supervisão e controle, devem permitir que sejam monitoradas e rastreadas as informações de um processo. A interface disponibiliza um *menu* para o envio de comandos aos sistemas de controle [13]. A figura 1 demonstra a interface para usuário. Disponível em www.mes-sigma.net/Cursos/images/Automatizacion%20con%20Excel.pdf.

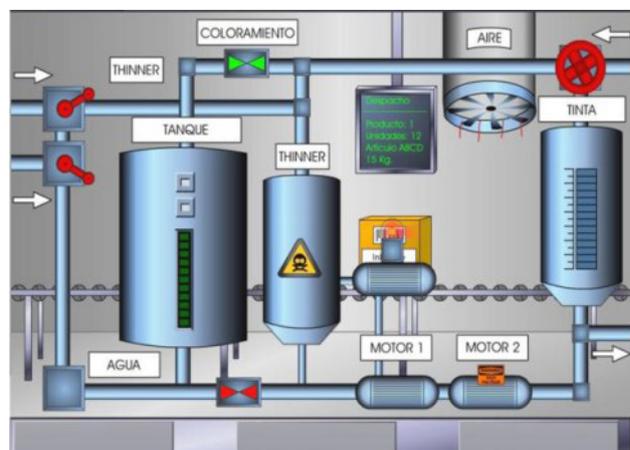


Figura 1: Exemplo de interface gráfica.

Os sistemas supervisórios são geralmente utilizados com a finalidade de fazer o controle de processos e o posterior reconhecimento de eventuais falhas que ocorrem em um sistema automatizado por dispositivos eletrônicos, CLP's que controlam os equipamentos através de sua interface de *hardware* [1]. Os aplicativos SCADA estabelecem a relação entre o homem e a máquina, para que ambos possam se comunicar através da manipulação de variáveis de processo [11]. Essas variáveis são passadas entre dispositivos via rede de dados e estes interpretados de acordo com o protocolo estabelecido para comunicação entre os integrantes da rede.

Planilhas Office Excel e VBA

O sistema VBA (*Visual Basic for Applications*), recurso que permite criar funções personalizadas em planilhas do Office Excel, pode fazer o mesmo em outras ferramentas do pacote Microsoft Office. Um complemento que permite a criação de macros, sub-rotinas para executar funções pré-programadas. Dados sobre integração do Excel com CLP's estão em www.stackoverflow.com/questions/17493359/how-can-i-get-data-from-multiple-siemens-plcs-into-excel.

METODOLOGIA

Para o desenvolvimento do sistema, o modelo adaptativo de desenvolvimento ágil DAS (*Adaptive Software Development*) foi o modelo eleito para implementar o paradigma de desenvolvimento de *software* orientado a objetos, pois este modelo aborda os sistemas computacionais de forma simples e clara [14]. Este modelo permite que etapas diferentes do projeto sejam desenvolvidas juntas em paralelo, isto ajuda na divisão das tarefas em problemas menores [14]. Esse método é muito semelhante ao conceito de orientação a objetos, onde a técnica é dividir para conquistar, além de permitir e facilitar mudanças em passos anteriores, durante o desenvolvimento do projeto. Em relação aos sistemas OPC (*OLE for Process Control*)², que se assemelham a este projeto, o grande benefício é a padronização no acesso aos dados, de diferentes fabricantes [3].

Além da padronização do acesso aos dados nos CLP's, ocorrerá a troca de informação entre aplicativos através do padrão COM. Este possibilita a comunicação entre processos e a criação dinâmica de objetos em qualquer linguagem de programação que suporte a tecnologia.

Na figura 2 se exhibe o diagrama de caso de uso geral da aplicação desenvolvida.

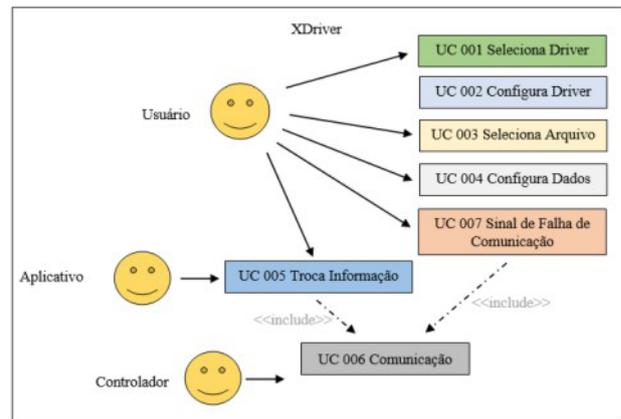


Figura 2: Diagrama de Caso de Uso Geral.

O projeto divide-se em objetos distintos, com funções diferentes entre si, mas que juntos, irão interagir na solução do problema proposto. As etapas ou módulos de desenvolvimento em paralelo foram os seguintes: i) Desenvolvimento do driver de comunicação Modbus TCP, DLL com protocolo implementado dentro de padrões que devem ser seguidos pelos futuros protocolos a serem desenvolvidos, de maneira a possibilitar o intercâmbio de *drivers* na interface de configuração da ferramenta XDriver; ii) Desenvolvimento das classes do aplicativo XDriver, aplicativo utilizado para troca de dados, o objetivo principal do projeto, estas por sua vez foram desenvolvidas de forma paralela e posteriormente foram integradas e testadas entre si; iii) Desenvolvimento de rotinas VBA em planilha do Office Excel, para demonstrar e exemplificar as funcionalidades do aplicativo desenvolvido; iv) Elaboração, desenvolvimento e configuração de rotina de instruções em um CLP com suporte ao protocolo Modbus TCP, usado nos testes do aplicativo desenvolvido.

Para o desenvolvimento do XDriver, foram definidos os seguintes requisitos funcionais (RF) a serem implementados:

- RF001: O sistema deve prever a seleção e troca do *driver* de comunicação, mantendo as demais configurações de arquivo e dados inalteradas quando desejado;
- RF002: O driver deve ter uma interface de configuração acessível ao usuário, esta deve ser acessada pela chamada padrão de interface de *driver* do aplicativo;
- RF003: O sistema deve prever uma seção para o tratamento e configuração de arquivos, com a finalidade de receber as informações dos controladores, CLP's;
- RF004: O sistema deve prever uma seção para o tratamento e configuração dos dados que serão trocados entre aplicativo e controlador;
- RF005: O aplicativo deve enviar dados do controlador para o arquivo selecionado e deste para o controlador conforme configurado na seção de dados;
- RF006: O aplicativo deve prever a comunicação com controladores lógicos programáveis diferentes de maneira padronizada, através do uso de *driver* com protocolo compatível ao controlador utilizado;

2- A OPC Foundation surgiu em 1995, sendo formada pelas empresas Fisher-Rosemount, Rockwell Software, Opto 22, Intellution e Intuitive Technology, com o objetivo de criar um mecanismo padronizado de troca de informações entre dispositivos dos mais diversos fabricantes.

- RF007: O sistema deve detectar uma interrupção no processo de comunicação com o CLP, dentro das diretrizes de seu protocolo e gerar aviso ao usuário, alertando-o sobre o fato.

Na finalização o produto encontra-se na sua concepção final, a partir dos resultados obtidos através dos testes, as etapas anteriores serão analisadas e os arquivos de documentação serão alterados conforme as correções ou atualizações de *software* que venham a ocorrer. O usuário terá a opção de seleção e configuração dos *drivers* implementados com os protocolos de rede conforme indicado na figura 3.



Figura 3: Tela utilizada na seleção e configuração de Drive.

Neste caso o Modbus TCP será o *driver* disponível para as aplicações iniciais do projeto, demais protocolos podem ser implementados em forma de DLL, seguindo um padrão definido de assinaturas de método para as funções de manipulação de dados. Desta forma o ato de implementar protocolos diferentes não implicará na estrutura das demais funcionalidades e configurações já determinadas na área de troca de dados pelo usuário. A figura 4 mostra a tela de configuração do *Driver* Modbus TCP utilizado.

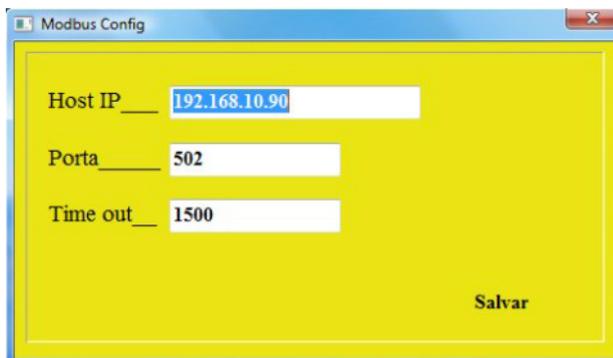


Figura 4: Tela utilizada na configuração personalizada do Driver.

Outra função importante disponível é a opção de seleção da planilha utilizada na troca de informação. A mesma será usada como interface para elaboração de aplicativos SCADA, que compreende um meio onde se traduz o que está acontecendo na máquina ou no sistema, um dispositivo que está entre o homem e a máquina [3]. A

interface para a seleção de arquivos é mostrada na figura 5.

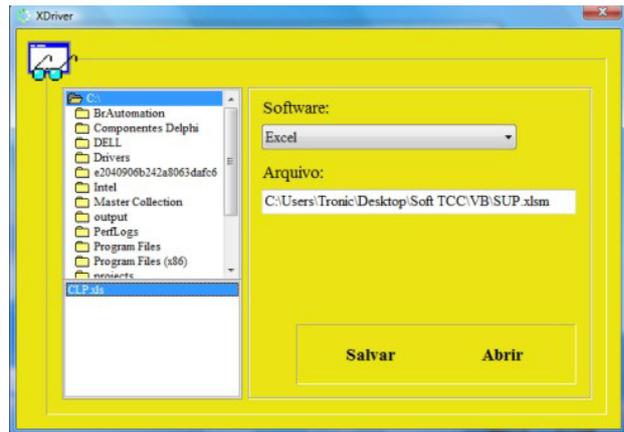


Figura 5: Tela utilizada na seleção de arquivos.

Através da seleção de planilhas o usuário irá definir o arquivo destino e a configuração das células para o recebimento e envio dos dados, incluindo a formatação e as variantes dos dados a serem trocados entre aplicativos, como mostra a figura 6.



Figura 6: Tela utilizada na configuração de dados.

O campo “Tag” é o identificador da tarefa, um simples marcador para referência do usuário. O “Endereço” indica o início da área de memória a ser lida no CLP, “Var” se refere ao tipo da variável a ser acessada na área de memória do controlador, pode ser do tipo word ou bit. O campo “Função” indica a ação de escrita ou leitura dos dados endereçados, “Cel=(L,C,P) representa respectivamente linha, coluna e página na planilha do Office Excel, onde os dados devem ser plotados ou lidos. O campo “Bloco” indica o tamanho do vetor a ser lido no controlador a partir do endereço inicial.

Exemplo: Endereço = 100 e Bloco = 10 teremos o seguinte vetor data [0..9] *of word*, neste caso teremos uma leitura ou escrita das seguintes áreas de memória no CLP, 100, 101 até 109. Na planilha do Office Excel os valores são tratados de forma semelhante, ou seja, após a primeira célula selecionada as demais são utilizadas na sequência, de cima para baixo, a partir da primeira célula indicada na coluna da tabela, conforme figura 7.

	A	B	C	D	E
1					
2	10000				
3	10001				
4	10002				
5	10003				
6	10004				
7	10005				
8	10006				
9	10007				
10	10008				
11	10009				
12					

Figura 7: Exemplo de dados plotados na planilha.

No caso da configuração de variável do tipo bit, a opção bloco não é levada em consideração, pois cada bit corresponde a uma posição de memória a ser manipulada, portanto não se pode manipular mais de um bit de forma simultânea, na mesma chamada de função. Para cada linha na tabela de dados configurada, foi adicionado ao lado direito da tela um campo auxiliar, este serve para adicionar um comentário mais elaborado sobre a respectiva configuração selecionada pelo usuário.

Resultados

Como resultado foi obtido o aplicativo denominado XDriver, que de acordo com o propósito do projeto estabelece a comunicação com dispositivos de controle usados em automação industrial e *softwares* comerciais que possuem o suporte ao padrão COM, neste caso uma planilha eletrônica do Office Excel. A figura 8 mostra a tela inicial do aplicativo, esta disponibiliza os ícones de acesso às ferramentas de configuração. O número na parte superior a esquerda faz referência à linha acessada na tabela de configuração de dados do aplicativo XDriver.



Figura 8: Interface de serviço do aplicativo XDriver.

Basicamente o *software* desenvolvido faz a troca de informação entre aplicativos independentes, com funções e recursos, que uma vez integrados pela ferramenta, irão atuar em conjunto na solução do problema a ser resolvido.

De um lado está o controlador lógico CLP gerenciando

uma máquina ou um dispositivo qualquer na indústria ou comércio, na outra extremidade uma planilha eletrônica do Office Excel atuando como uma interface de supervisão e controle, permitindo a visualização e interação do usuário com as variáveis de processo do sistema.

Na tela principal do aplicativo está disponível o acesso as funcionalidades do *software*, assim como a visualização do status de comunicação com o dispositivo a ser monitorado, em caso de falhas, devido ao limite de *time out* estabelecido durante a transferência de dados entre o XDriver e o CLP, ocorrerá uma que será sinalizada na tela do aplicativo.

A figura 9 mostra a implementação em uma planilha eletrônica do Office Excel utilizada como protótipo de teste da ferramenta desenvolvida.

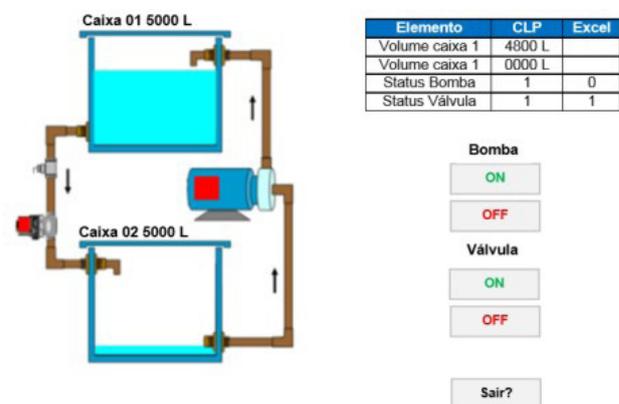


Figura 9: Aplicativo Office Excel.

O objetivo é representar um circuito hidráulico com o status de uma válvula solenóide para dreno do reservatório superior (caixa 01) e uma bomba elétrica que retira o fluido do reservatório inferior (caixa 02). No lado direito superior da tela fica localizada uma tabela que mostra os valores das variáveis de processo, assim como, o status dos dispositivos e nível dos reservatórios. Esta aplicação utiliza a planilha do Office Excel como base para o desenvolvimento da interface visual e através do padrão COM é realizado a troca de informação com o XDriver.

De forma semelhante, no módulo responsável pela comunicação com o controlador, através do protocolo Modbus TCP, a comunicação foi implementada gerenciando a troca de dados e sinalizando erros advindos de falhas no meio físico da rede, como falta de energia no controlador e rompimento do cabo de comunicação.

Conforme visto, a integração entre os aplicativos se realiza por dois módulos distintos, de um lado o padrão COM, responsável pela integração entre Office Excel e o aplicativo XDriver e na outra extremidade o protocolo Modbus TCP, este utilizado na comunicação entre o aplicativo XDriver e o CLP.

A figura 10 ilustra de forma mais clara os módulos de comunicação do aplicativo desenvolvido.

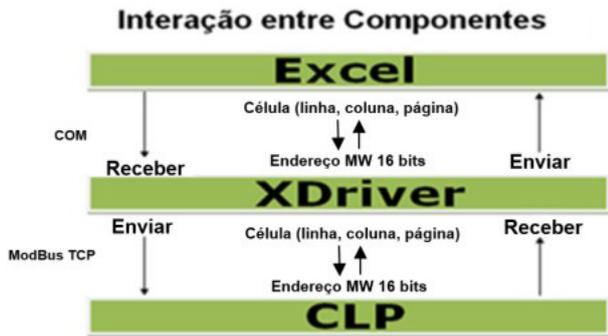


Figura 10: Módulos de comunicação utilizados pelo aplicativo XDriver.

A figura 11 mostra a estrutura que representa os componentes do sistema envolvidos no desenvolvimento do projeto. Os componentes estão divididos e representados em objetos diferentes de acordo com suas funcionalidades.

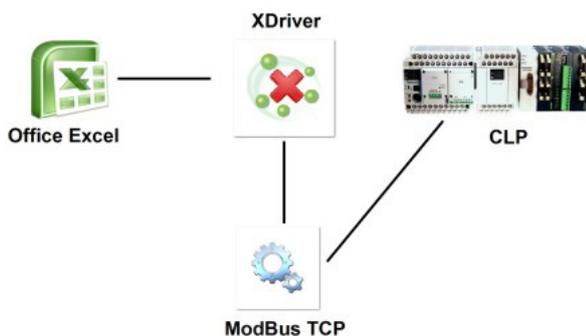


Figura 11: Componentes que integram o sistema.

Conclusão

O trabalho resultou na criação de um aplicativo que administra a troca de dados entre a aplicação comercial, neste caso o Office Excel e o controlador programável utilizado durante os testes. O mesmo possui suporte ao protocolo Modbus TCP, o que não impede que o aplicativo possa interagir com outros protocolos de comunicação. O algoritmo responsável pelo protocolo de comunicação com o controlador fica encapsulado na DLL e as chamadas de função são padronizadas entre o aplicativo e a mesma.

O rendimento e a estabilidade da comunicação do aplicativo desenvolvido com o CLP está diretamente relacionado com a implementação desta DLL, que neste caso, representa as partes de *software* com funções de acesso a dados no CLP.

Utilizando o *driver* desenvolvido, que foi disponibilizado na forma de DLL, encarregado de prover o suporte ao protocolo Modbus TCP, a comunicação manteve-se estável entre o controlador e o aplicativo XDriver, respeitando o tempo máximo estipulado para a resposta do dispositivo escravo.

O teste de *time out* foi realizado entre faixas de 500 a 1500 ms, dentro destes valores de monitoramento, pode

ocorrer uma série de erros ou atraso na comunicação após o envio de uma requisição do aplicativo ao *driver*, que por sua vez, encaminha a requisição ao CLP e retorna o resultado ao aplicativo.

Tornou-se possível através do aplicativo XDriver, que recursos visuais e funcionais de alto nível, existentes no Office Excel pudessem ser direcionados para fins de automação na área da indústria, sem a necessidade de grandes configurações por parte do usuário.

Ao final foi obtido um sistema capaz de atuar em segundo plano no sistema operacional, despercebido do usuário final, sendo um *software* executável local que interage com controladores industriais através de *drivers* específicos para esta função, responsáveis pelo acesso aos dados na área de memória destes equipamentos.

Referências

1. Bolzani, C. A. M. Análise de Arquiteturas e Desenvolvimento de uma Plataforma para Residências Inteligentes. Tese de Doutorado em Engenharia Elétrica, Escola Politécnica São Paulo, 2010.
2. Silva, J. Microcontrolador 8051 (Hardware e Software). 3º ed. São Paulo: Editora Érica, 1994.
3. Cândido, R. V. B. Padrão OPC: Uma Alternativa de Substituição dos Drivers Proprietários para Acessar Dados de CLP's. Monografia Bacharelado em Ciência da Computação, Universidade FUMEC, Belo Horizonte Minas Gerais, 2004.
4. Guimarães, H. C. F. Norma IEC 61131-3 Para Programação de Controladores Programáveis: Estudo e Aplicação. Projeto de Graduação, Universidade Federal do Espírito Santo, 2005.
5. Martins, M. R. A. Integração de Sistemas em Middleware. Dissertação de Mestrado em Engenharia Elétrica, Escola Politécnica São Paulo, 2010.
6. Nogueira, T. A. Redes de Comunicação para Sistemas de Automação Industrial. Monografia Engenharia e Controle de Automação, Universidade Federal de Ouro Preto, Minas Gerais, 2009.
7. Djiev, S. Industrial Networks for Communication and Control. Technical University of Sofia, 2003. Disponível em: <<http://anp.tu-sofia.bg/djiev/Networks.htm>>. Acesso em: 14 de março de 2012.
8. Kobayashi, T. H. Uma Ferramenta de Manipulação de Pacotes para Análise de Protocolos de Redes Industriais Baseados em TCP/IP. Dissertação de Mestrado em Engenharia da Computação, Universidade Federal do Rio Grande do Norte, 2009.
9. Oliveira, D. B. Programa de Computador para Análise de Relações entre Estrutura Química e Atividade Biológica. Dissertação de Mestrado em Física, Universidade Federal do Espírito Santo, 2001.
10. Gonçalves, T.A. V. Projeto e Implementação de uma

- aplicação de pequena rede industrial para controle de ETAR. Dissertação de Mestrado Integrado em Engenharia Eletrônica Industrial e Computadores, Universidade do Minho, Braga, Portugal, 2009.
1. 11. **Tibola, L. R.** Geração de Sistemas Supervisórios a partir de Modelos Orientados a Objetos. Dissertação de Mestrado em Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.
 1. 12. **Silva, A. P. G. y Salvador, M.** O que são sistemas supervisórios? - 2004. Disponível em: http://www.wectrus.com.br/artigos/sist_superv.pdf Acessado em: 10 de março de 2012.
 1. 13. **Ballock, I. R.** Protótipo de um Sistema Para Supervisão de Equipamentos Industriais. Trabalho de conclusão de curso Ciência da Computação, Universidade Regional de Blumenau, Santa Catarina, 2003.
 1. 14. **Pressman, R.** Engenharia de Software. 6ª ed. Rio de Janeiro: McGrawHill, 2006.

Recibido: 08/03/2013

Aprobado: 11/03/2014