

MATHEUS MUNHOZ LAGES MARCHENTA

CONTROLE DE MANUFATURA VIA INTERNET

Monografia apresentada a Escola
Politécnica da Universidade de São
Paulo referente à disciplina PMR
2550 - Projeto de Conclusão do
Curso II

São Paulo

2009

MATHEUS MUNHOZ LAGES MARCHENTA

CONTROLE DE MANUFATURA VIA INTERNET

Monografia apresentada a Escola
Politécnica da Universidade de São
Paulo referente à disciplina PMR
2550 - Projeto de Conclusão do
Curso II

Curso de Graduação: Engenharia
Mecatrônica

Orientador: Prof. Dr. Fabrício
Junqueira

São Paulo

2009

Aos meus pais Ovanir Marchenta
Filho e Renata Munhoz Lages
Marchenta e às minhas irmãs
Maria Paula Munhoz Lages
Marchenta e Maria Fernanda
Munhoz Lages Marchenta.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DA
ENGENHARIA MECÂNICA/NAVAL DA ESCOLA POLITÉCNICA (EPMN) –
USP.

Marchenta, Matheus Munhoz Lages
Controle de Manufatura via Internet / M.M.L.
Marchenta. -- São Paulo, 2009.
55 p. e 3 apêndices

Trabalho de Formatura - Escola Politécnica da
Universidade de São Paulo. Departamento de
Engenharia Mecatrônica e de Sistemas Mecânicos.

1. Manufatura 2. Internet 3. Controladores Programáveis
I. Universidade de São Paulo. Escola Politécnica.
Departamento de Engenharia Mecatrônica e de Sistemas
Mecânicos II. t.

Agradecimentos

Aos meus pais, Ovanir Marchenta Filho e Renata Munhoz Lages Marchenta, e minhas irmãs, Maria Fernanda Munhoz Lages Marchenta e Maria Paula Munhoz Lages Marchenta, por me dar todo o suporte necessário durante os anos de faculdade.

Ao meu orientador, Prof. Dr. Fabrício Junqueira pela sua constante orientação e incentivo para o desenvolvimento deste trabalho. Aos colegas José Maurício e José Garcia, pelo apoio durante o desenvolvimento deste projeto.

À Escola Politécnica da USP, em especial ao Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, que institucionalmente viabilizaram este trabalho.

Sumário

LISTA DE FIGURAS	IV
LISTA DE TABELAS.....	V
LISTA DE ABREVIATURAS	VI
RESUMO	VII
ABSTRACT	VIII
1. INTRODUÇÃO.....	1
1.1. OBJETIVO	2
1.2. ESTRUTURA.....	3
2. REVISÃO BIBLIOGRÁFICA	4
2.1. REDE DE PETRI E SFC	5
2.1.1. <i>Conceitos gerais de rede de Petri</i>	6
2.2. SISTEMAS DISTRIBUÍDOS.....	8
2.2.1. <i>Manufatura dispersa</i>	8
2.3. WEB SERVICE	9
2.4. UNIFIED MODELING LANGUAGE (UML).....	10
2.5. CONTROLE SUPERVISÓRIO DE SED	10
2.6. CONTROLADOR LÓGICO PROGRAMÁVEL.....	11
2.7. PROFIBUS	14
2.8 OPC.....	15
3. CÉLULA DE MANUFATURA DIDÁTICA	17
3.1. ESTAÇÃO DE DISTRIBUIÇÃO	17
3.2. ESTAÇÃO DE INSPEÇÃO OU DE TESTES.....	18
3.3. ESTAÇÃO DE TRANSPORTE.....	19
3.4. ESTAÇÃO DE MONTAGEM	20
4. PROJETO.....	26
4.1. FUNCIONAMENTO DA ESTAÇÃO DE MONTAGEM	26
4.2. PROJETO DO SOFTWARE DE CONTROLE DO CLP	29
4.2.1. <i>Mapeamento das entradas e saídas do CLP</i>	29
4.2.2. <i>Especificação do software de controle do CLP</i>	32
4.3. MAPEAMENTO DAS VARIÁVEIS NO SERVIDOR OPC	32
4.4. WEBSERVICES.....	33

4.5. INTERFACE WEB	37
4.5.1. Acesso à interface	40
4.5.2. Utilização da interface (Supervisório, Operação e Monitoração)	41
5. COMENTÁRIOS FINAIS E CONCLUSÕES.....	45
REFERÊNCIAS BIBLIOGRÁFICAS	47
APÊNDICE A – WEB SERVICE.....	I
APÊNDICE B – APLICAÇÃO WEB	I
APÊNDICE C – PROGRAMAÇÃO DO CLP	I

Lista de Figuras

FIGURA 1. ESQUEMA ILUSTRATIVO DA CÉLULA DE MANUFATURA DIDÁTICA (MINI CIM)	2
FIGURA 2. LINGUAGENS GRÁFICAS DE DOCUMENTAÇÃO (MIYAGI, 1996)	6
FIGURA 3. ELEMENTOS DE REDE DE PETRI.....	7
FIGURA 4. INTERAÇÃO ENTRE ELEMENTOS ATIVOS E PASSIVOS	8
FIGURA 5. ESTRUTURA BÁSICA DE UM <i>WebService</i>	9
FIGURA 6. SED EM MALHA FECHADA (RIBEIRO CURY, 2001)	11
FIGURA 7. CICLO DE PROCESSAMENTO DE UM CLP (PINTO, 2008).....	12
FIGURA 8. SIMATIC S7 300 DESIGN (PINHEIRO, 2006)	12
FIGURA 9. ESTRUTURA DE AUTOMAÇÃO VIA PROFIBUS (PROFIBUS, 2006).....	15
FIGURA 10. MONTAGEM DAS DIFERENTES PEÇAS PELO MINI CIM.....	17
FIGURA 11. ESQUEMA ILUSTRATIVO DA MINI CÉLULA DE MANUFATURA DIDÁTICA (MINI CIM)	18
FIGURA 12. ESTAÇÕES DE DISTRIBUIÇÃO E TESTES	19
FIGURA 13. DETALHAMENTO DOS COMPONENTES DA ESTAÇÃO DE TRANSPORTE	20
FIGURA 14. DETALHAMENTO DOS COMPONENTES DA ESTAÇÃO DE MONTAGEM	20
FIGURA 15. MAGAZINE DE MOLAS	21
FIGURA 16. MAGAZINE DE PINOS DE SUPORTE	21
FIGURA 17. MAGAZINE DE TAMPAS	22
FIGURA 18. MÓDULO DE TRAVAMENTO	22
FIGURA 19. (A) - ACIONAMENTO DO EIXO Y; (B) - ACIONAMENTO DO EIXO X.....	23
FIGURA 20. CLP E MÓDULO DE CONTROLE DOS MOTORES (DESTAQUE).....	23
FIGURA 21. EIXO Z, SENSORES DE POSIÇÃO E GARRA DEDICADA.....	24
FIGURA 22. DETALHES DO CARRO TRANSPORTADOR	24
FIGURA 23. MAGAZINES, APOIO DE PALETE E MÓDULO DE TRAVAMENTO	25
FIGURA 24. ESTRUTURA DE PROJETO	27
FIGURA 25. DIAGRAMA SFC DA SEQUÊNCIA DE MONTAGEM (GARRA)	28
FIGURA 26. BLOCO DE INICIALIZAÇÃO DO EIXO Z (<i>LADDER</i>)	32
FIGURA 27. LÓGICA INTERNA DO BLOCO DE INICIALIZAÇÃO DO EIXO Z (<i>LADDER</i>).....	33
FIGURA 28. LISTA DE FUNÇÕES DO <i>WebService</i> (ACESSADO VIA NAVEGADOR).....	36
FIGURA 29. ACESSO ÀS FUNÇÕES DO <i>WebService</i> (ACESSADO VIA NAVEGADOR).....	36
FIGURA 30. ADICIONANDO UMA "WEB REFERENCE" (VISUAL STUDIO)	37
FIGURA 31. DIAGRAMA DE CASOS DE USO DA INTERFACE (OPERAÇÃO/MONITORAÇÃO)	38
FIGURA 32. DIAGRAMA DE CASOS DE USO DA INTERFACE (SUPERVISÃO)	39
FIGURA 33. ACESSO DE USUÁRIOS REGISTRADO (OPÇÕES DE ACESSO).....	40
FIGURA 34. HIERARQUIA DE ACESSO	41
FIGURA 35. INTERFACE DO SUPERVISOR	42
FIGURA 36. ATIVANDO OPERAÇÃO.....	42
FIGURA 37. CONTROLES DO SUPERVISOR	43
FIGURA 38. INTERFACE DO OPERADOR.....	43
FIGURA 39. INTERFACE DO MONITOR.....	44

Lista de Tabelas

TABELA 1. ESPECIFICAÇÃO DO CPU 315-2 DP (KANEKO, 2008).....	13
TABELA 2. TIPOS DE MÓDULOS DE EXPANSÃO PARA O SIMATIC S7-300 (KANEKO, 2008)	14
TABELA 3. ESTADOS DO DIAGRAMA DE CONTROLE.....	29
TABELA 4. TRANSIÇÕES DO DIAGRAMA NA ORDEM DE OCORRÊNCIA.....	30
TABELA 5. VARIÁVEIS DO CLP	31
TABELA 6. POSSÍVEIS POSIÇÕES DOS EIXOS X E Y (KANEKO, 2008)	35
TABELA 7. SEQUÊNCIA DE OPERAÇÃO.....	42

Lista de Abreviaturas

CLP - Controlador Lógico Programável

SED - Sistema a Eventos Discretos

SFC - *Sequential Flow Chart*

TI - Tecnologia da Informação

Resumo

Os mercados estão se tornando mais globais e independentes de barreiras geográficas e um maior número de indústrias de manufatura tem se estabelecido de forma distribuída e dispersa, aproveitando o crescimento das redes de comunicação e da Tecnologia da Informação e Comunicação (TIC), permitindo uma maior cooperação transnacional. A utilização da TI nos processos produtivos também vem permitindo a integração dos diversos sistemas da empresa, em seus diferentes níveis. Como exemplo, pode-se citar os sistemas de supervisão e controle, responsáveis pelas operações do chão de fábrica, que são compostos por uma coleção heterogênea de *hardwares* e *softwares* distribuídos em um conjunto de plataformas (estações de operação, unidades remotas, controladores programáveis, etc.). Para permitir a integração dos diversos sistemas da empresa e de várias empresas, que cooperam no desenvolvimento de produtos globais, pode-se, por exemplo, utilizar *WebServices*. Porém, apenas recentemente vêm-se observado estudos visando a disponibilização de equipamentos de manufatura como serviços. Neste sentido, foi projetada e implementada uma solução, baseada em *WebService*, para disponibilizar a estação de montagem da célula de manufatura didática (Mini CIM), presente no Laboratório de Sistemas de Automação, para monitoração e/ou teleoperação.

Abstract

The markets are becoming more global and independent of geographical barriers and a bigger number of industries have been established in a distributed and scattered way, taking advantage of communication networks and Information and Communication Technology (ICT) growth, permitting the increase of transnational cooperation. The utilization of the TI in the productive process also has permitted the integration of the diverse company systems, in different levels. As example, it can be cited the supervision and control systems, responsible for the manufacturing operations, that are composed by a heterogeneous collection of hardware and software distributed in platforms (stations of operation, remote units, programmable controllers, etc.). To permit the integration of the diverse company systems and also several companies, which cooperate in the development of global products, it can be used, for example, WebServices. However, only recently it has been observed studies aiming at the manufacture equipment availability as a service. Regarding that, it was projected and implemented a solution, using a WebService, to make the assembly station of the didactic manufacture cell (Mini CIM), present in the Automation Systems Laboratory, available for monitoring and/or teleoperation.

1. Introdução

A revolução industrial implicou em mudanças profundas e permanentes para a sociedade como um todo. Ela marcou o princípio da grande transição da sociedade agrícola para a sociedade urbana, bem como a superação do trabalho manual humano pelo uso das máquinas (SILVA FILHO, 2001).

O momento do final do século XX ao começo do século XXI é caracterizado como o período posterior à revolução tecnológica, ou a revolução digital. A era da informação, como é contextualizada, tem como ponto central a otimização do tempo despendido por processos sistêmicos padronizados. O processamento e a troca de informações progrediram numa taxa jamais antes vivenciada (SILVA FILHO, 2001).

Nos processos produtivos, o avanço se mostra presente em vários aspectos como melhores materiais, melhores processos produtivos, processos menos suscetíveis a falhas, com produtividade aprimorada e maior flexibilidade. Como exemplo deste avanço tem-se a evolução da tecnologia da informação e sua contribuição para o constante aprimoramento dos processos produtivos no final do século XX, que vem contribuindo de maneira permanente e cada vez mais efetiva nos processos de produção, estreitando fronteiras, conferindo flexibilidade e produtividade a tais processos (JUNQUEIRA, 2006).

A introdução da TI nos processos produtivos também vem permitindo a integração dos diversos sistemas da empresa, em seus diferentes níveis. Como exemplo, pode-se citar os sistemas de supervisão e controle, responsáveis pelas operações do chão de fábrica, típicos em uma instalação de médio e grande porte que são compostos por uma coleção heterogênea de *hardwares* e *softwares* distribuídos em um conjunto de plataformas (estações de operação, unidades remotas, controladores programáveis, etc.) A necessidade de integração de sistemas heterogêneos implica um aumento da complexidade e diversidade dos sistemas (JUNQUEIRA, 2006).

Uma telemanufatura é definida como uma atividade onde um cliente utiliza algum serviço oferecido por um servidor disponibilizado via Rede de Comunicação através de vias de informação (internet) para executar em tempo real operações e processos necessários para o projeto e a produção de bens

(MALEK, 1998). Dessa forma, sistemas de teleoperação de equipamentos industriais, que é o conceito implementado neste trabalho, enquadram-se no contexto de telemanufatura nos aspectos referentes ao controle da manufatura em operações de chão de fábrica (ÁLVARES *et al.*, 2001).

Para permitir a teleoperação e a integração dos diversos sistemas da empresa e de várias empresas, que cooperam no desenvolvimento de produtos globais, pode-se, por exemplo, utilizar *WebServices*. Porém, apenas recentemente, tem-se observado estudos visando à disponibilização de equipamentos de manufatura como serviços (BELL, 2006).

1.1. Objetivo

Neste sentido, pretende-se, neste trabalho, estudar, projetar e implementar uma solução para disponibilizar equipamentos de manufatura para monitoração e/ou teleoperação via internet. Ou seja, uma solução para sistemas de teleoperação de equipamentos industriais.

Como unidade industrial será utilizada a estação de montagem da célula de manufatura didática (Mini CIM) presente no Laboratório de Sistemas de automação (Figura 1), mais especificamente a estação de montagem desta unidade.

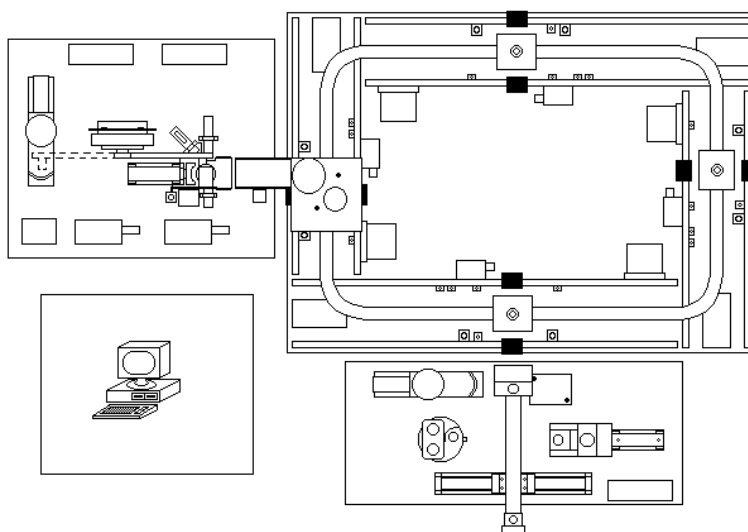


Figura 1. Esquema ilustrativo da célula de manufatura didática (Mini CIM)

1.2. Estrutura

No capítulo 2 é feita uma revisão bibliográfica, contendo a fundamentação teórica para o presente trabalho, como linguagens para programação de CPs, sistemas distribuídos, *WebService*, UML, controle supervisorio, Controlador Lógico Programável, PROFIBUS. No capítulo 3 é apresentada uma descrição do sistema a partir do qual o projeto foi desenvolvido e no capítulo 4 é apresentado o projeto, com o mapeamento de requisitos e a descrição das soluções desenvolvidas. No capítulo 5 são apresentados os comentários finais e conclusão do trabalho.

2. Revisão bibliográfica

Com o desenvolvimento dos Controladores Programáveis (CPs) surgiu a necessidade do desenvolvimento das linguagens para documentar os projetos. Inicialmente utilizava-se o próprio diagrama de relés como documentação, o que acabava por camuflar a dinâmica do processo. Como este diagrama apresentava os procedimentos e intertravamentos na forma de contatos e bobinas, era difícil sua compreensão por outros que não o próprio projetista, dificultando bastante sua manutenção. Além da dificuldade de se documentar cálculos algébricos mais complexos no diagrama (MIYAGI, 1996).

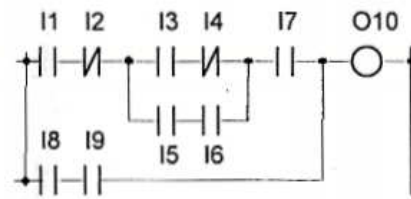
Assim, no IEC (*International Electrotechnical Committee*), foram desenvolvidos padrões de forma geral que possibilitam acompanhar o desenvolvimento dos CPs e dos próprios *softwares* de programação. Estas formas de documentação foram classificadas em textuais, gráficas e tabulares (MIYAGI, 1996).

- **Textuais:** Utiliza símbolos e expressões matemáticas para descrever o sistema de controle:
 - o **Álgebra booleana:** Utiliza expressões booleanas para representar a lógica do processo, mas por não possibilitar temporizações nem sequenciamentos, é utilizada apenas na concepção do projeto;
 - o **- IL (*Instruction List*):** É uma lista continua de informações essenciais, com load (LD), AND, OR, *store* (ST), que representam as entradas e saídas do CP, na sua ordem de execução;
 - o **ST (*Structured Text*):** É a representação em linguagem de alto nível do processo, não representando diretamente a ordem de execução. Possibilita a estruturação de programas que utilizam operações de comparação como IF, CASE.
- **Gráficas:** Este tipo de linguagem facilita a visualização do fluxo do procedimento de controle, facilitando a manutenção e depuração de erros. Existem linguagens adequadas para representação das operações lógicas (*Ladder Diagram* e *Function Block Diagram*) e para representação dos procedimentos de controle (fluxograma e *Sequential Flow Chart*):

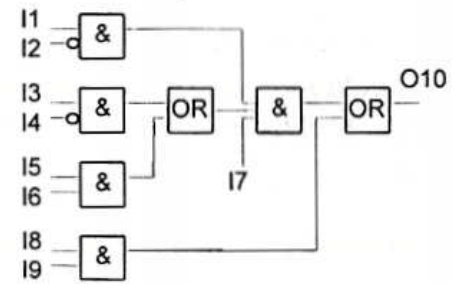
- o **LD (*Ladder Diagram*)**: É o próprio digrama de relés, citado anteriormente. O LD é definido de forma ampla no IEC, permitindo, assim, a inclusão de elementos SFC, blocos funcionais e outras características que permitam seu uso em sistemas mais complexos, o que faz com este diagrama ainda seja muito utilizado (Figura 2a);
- o **FDB (*Function Block Diagram*)**: Representa as combinações de blocos lógicos AND, OR, etc., além de blocos correspondentes a processamentos numéricos como soma, multiplicação, comparação, que permitem a representação de funções de alto nível (Figura 2b);
- o **Fluxograma**: Adequado para controles puramente sequenciais, utiliza a mesma idéia do fluxograma de programação de computadores, acrescido de funções para controle de sistemas a eventos discretos (SEDs) (Figura 2c);
- o **SFC (*Sequential Flow Chart*)**: Adequada para controle de SEDs que envolvam *steps* (passos, condições) e *transitions* (transições). Foi desenvolvido baseado na rede de Petri, que são técnicas de representação efetiva do funcionamento dos sistemas, não importando sua complexidade (Figura 2d).
- **Tabulares**: É a descrição do SED em forma de tabelas:
 - o **Decision table**: São tabelas que contém, para cada passo, sua ação, a identificação do próximo passo e a condição para transição para o próximo passo. Os digramas de tempos também são considerados um tipo de *decision table*.

2.1. Rede de Petri e SFC

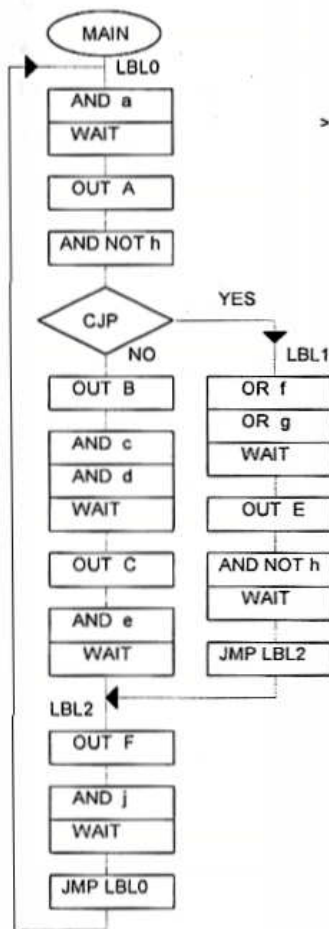
Como visto, a linguagem SFC representa de forma efetiva e clara as especificações e o funcionamento de um SED, possibilitando o fácil entendimento de sua lógica quando necessário, seja para manutenção ou depuração de erros, e por isto é considerada superior às outras técnicas de representação. Esta linguagem, como dito anteriormente, foi desenvolvida com base em rede de Petri e para seu maior entendimento é necessário ter conhecimento do funcionamento deste tipo de rede (MIYAGI, 1996).



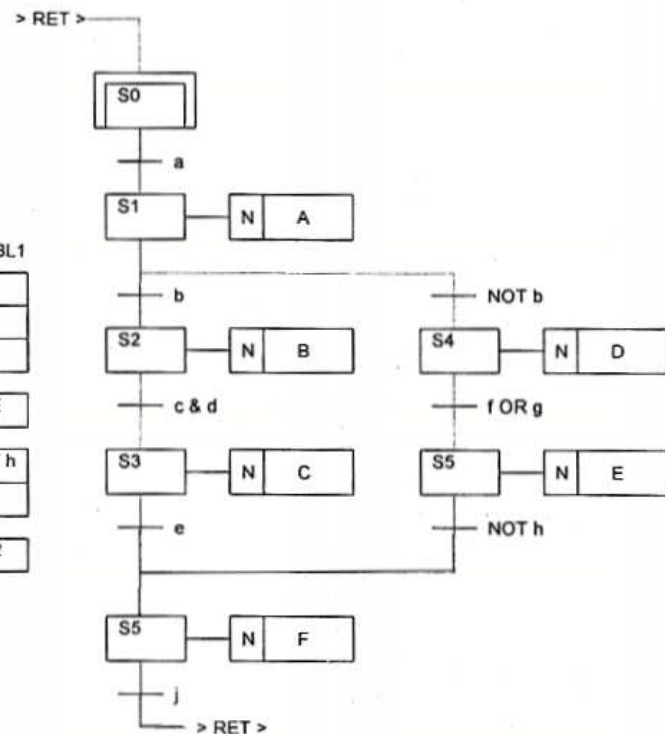
(a) LD



(b) FBD



(c) Fluxograma



(d) SFC

Figura 2. Linguagens gráficas de documentação (MIYAGI, 1996)

2.1.1. Conceitos gerais de rede de Petri

Muito utilizada para descrever SEDs, a rede de Petri possui componentes ativos e passivos. Os componentes passivos são representados por círculos e indicam **lugares** ou condições fixas que não estão ligadas

diretamente a ações. Já os elementos ativos são representados por retângulos e indicam **transições**, eventos ou, simplesmente, os atores de determinada ação. **Arcos orientados** indicam o fluxo de itens (informações ou objetos). **Marcas** nos elementos passivos indicam que estão satisfeitas condições ou que há itens naquela posição, possibilitando a ação do elemento ativo (MIYAGI, 1996).

Como pode ser visto na Figura 3, pode haver mais de um **arco** chegando a um elemento ativo e, para que haja a mudança de estado, é necessário que todas suas condições estejam satisfeitas, ocasionando o preenchimento das posições de saída deste elemento ativo (MIYAGI, 1996).

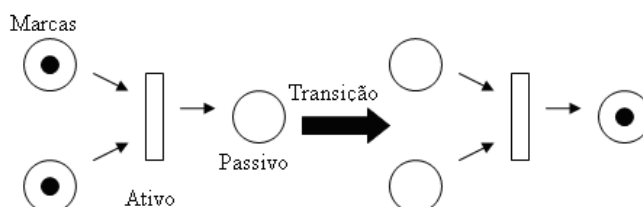


Figura 3. Elementos de rede de Petri

Vale observar que nunca existirão **arcos** ligando dois elementos ativos ou dois passivos. Se isto ocorrer, a separação entre os componentes não foi corretamente avaliada durante a modelagem. Por exemplo, supondo o processo de usinagem utilizando-se duas máquinas, M1 e M2, alimentadas por esteiras, E1 e E2, sendo que E2 interliga M1 e M2. Ao modelar o processo, a máquina M1 será o componente ativo, indicando a **transição** (no caso a própria usinagem em M1) entre uma esteira (E1) e a outra (E2), ambos elementos passivos. Ocorrida a **transição**, M2 terá sua condição de funcionamento satisfeita. Pode-se ver, portanto, que é necessária a presença da esteira (passiva) para ligar um elemento ativo a outro (Figura 4), um erro comum é considerar a passagens entre as máquinas de forma direta, impossibilitando a análise por rede de Petri (MIYAGI, 1996).

Considerando estes construtores básicos de uma rede de Petri, podem ser criados diversos tipos de modelos, que levam em consideração pontos de

vistas diferentes para determinar quais serão os elementos ativos e passivos. Estes conceitos rede de Petri, como a transição entre estados e determinação do tipo dos elementos, foram mantidos durante o desenvolvimento do SFC (MIYAGI, 1996).

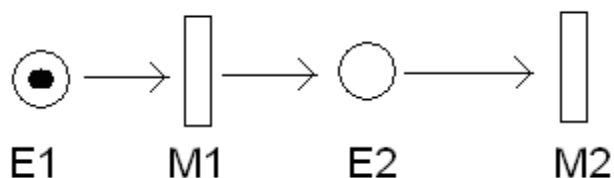


Figura 4. Interação entre elementos ativos e passivos

2.2. Sistemas distribuídos

Um sistema distribuído é uma coleção de computadores independentes que se apresenta ao usuário como um sistema único e consistente (TANENBAUM, 2007).

Os sistemas distribuídos englobam dois aspectos. O primeiro diz respeito ao *hardware*: as máquinas são autônomas. O segundo diz respeito ao *software*: os usuários pensam no sistema como um único computador (TANENBAUM, 2007).

2.2.1. Manufatura dispersa

Segundo Junqueira (2006), uma rede de manufatura dispersa consiste de instalações físicas que estão geograficamente dispersas, mas precisam se comunicar e trabalhar cooperativamente trocando uma grande quantidade de informações e dados entre suas próprias fábricas, empresas terceirizadas e fornecedores. O projeto e operação deste tipo de sistema produtivo requerem uma abordagem de projeto distribuído, em que equipes geograficamente distribuídas em diferentes locais físicos colaboraram para a especificação do sistema e avaliação dos processos produtivos.

A presente evolução da TI e sua aplicação nos diversos processos produtivos permitiram a integração de sistemas heterogêneos. Como exemplo do presente trabalho, o sistema de controle e monitoração da célula de

manufatura didática interage com um conjunto heterogêneo de *hardware* ((sensores, atuadores, controlador programável) e *software* (programa do CLP, *WebService*, e interface com o usuário, entre outros), a fim de monitorar e controlar o processo de montagem (JUNQUEIRA, 2006).

2.3. Web service

Muito utilizado quando é necessário o uso de uma tecnologia que garanta a troca de dados com rapidez e de forma simples, o *WebService* realiza a transferência de dados no formato XML (*eXtensible Markup Language*), encapsulados pelo protocolo SOAP (*Simple Object Access Protocol*). De maneira resumida, *WebService* é qualquer serviço disponibilizado via internet, que usa o sistema de envio no padrão XML e não é ligado a nenhum sistema operacional específico ou linguagem de programação (CERAMI, 2002), como pode ser observado na Figura 5.

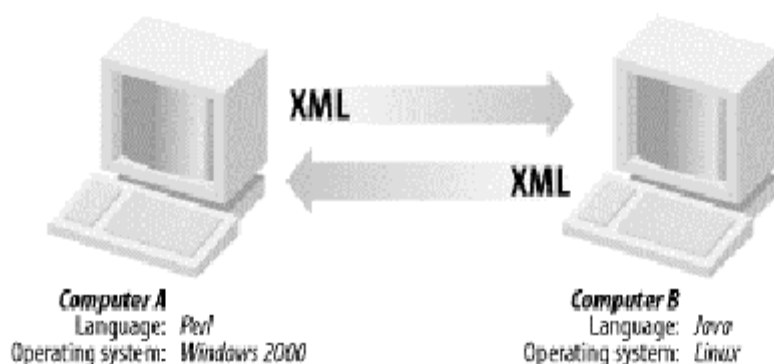


Figura 5. Estrutura básica de um *WebService*

WebService está se tornando a técnica dominante para representação e distribuição de informações por meio de sistemas múltiplos, inclusive para sistemas que não foram inicialmente planejados ou desenvolvidos para interagirem (BELL, 2006).

2.4. Unified Modeling Language (UML)

A UML surgiu como um esforço para criar uma linguagem de modelagem, advinda da unificação dos métodos pré-existentes de modelagem: Booch e OMT. A primeira versão apresentada (UML 0.9) surgiu em 1996, contudo a versão 1.3, que é utilizada neste trabalho, surgiu apenas no final de 1998 (BOOCH, 1999).

A UML é adequada para modelagem de sistemas, cuja abrangência poderá incluir sistemas de informação corporativos a serem distribuídos a aplicações baseadas em Web e até sistemas embarcados complexos de tempo real. É uma linguagem muito expressiva, abrangendo todas as visões necessárias ao desenvolvimento e implantação desses sistemas (BOOCH, 1999).

Essa linguagem se destina principalmente a sistemas complexos de *software*. Tem sido empregada de maneira efetiva em domínios como os seguintes (BOOCH, 1999):

- Sistemas de informações corporativos;
- Serviços bancários e financeiros;
- Telecomunicações;
- Transportes;
- Defesa/espço aéreo;
- Vendas de varejo;
- Eletrônica médica;
- Científicos;
- Serviços distribuídos baseados na Web.

2.5. Controle supervisorio de SED

Um sistema a ser controlado corresponde geralmente a um conjunto de subsistemas ou equipamentos arrançados segundo uma distribuição especial. Estes subsistemas têm isoladamente um comportamento básico original, que quando atuando em conjunto com os demais subsistemas, deve ser restringido

de forma a cumprir a função coordenada a ser executada pelo sistema global (RIBEIRO CURY, 2001).

Para que os subsistemas (no caso os equipamentos da estação de montagem do Mini CIM) atuem de forma coordenada, introduz-se um agente de controle, ou supervisor. Considerando-se o supervisor interagindo com a planta em malha fechada (Figura 6), os eventos ocorridos na planta são observados pelo supervisor, que define quais eventos, dentre os fisicamente possíveis de ocorrerem no estado atual, devem se seguir, ou seja, o supervisor tem ação habilitadora e desabilitadora de eventos. O conjunto de eventos habilitados num dado instante pelo supervisor define uma entrada de controle, que é atualizada a cada nova ocorrência de evento observada na planta (RIBEIRO CURY, 2001).

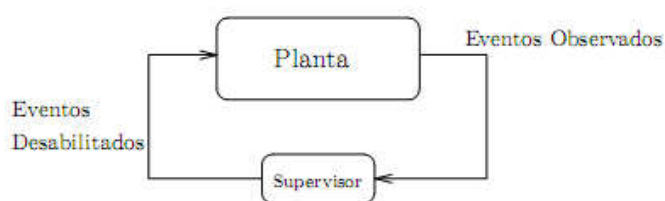


Figura 6. SED em malha fechada (RIBEIRO CURY, 2001)

2.6. Controlador lógico programável

O controlador lógico programável (CLP) é um dispositivo definido como um computador industrial capaz de armazenar instruções para realização de funções de controle, além de realizar operações lógicas e aritméticas, manipulação de dados e comunicação em rede, amplamente utilizado no controle de Sistemas Automatizados (PINTO, 2008). A Figura 7 exemplifica um ciclo de processamento de um CLP.

Antes do início do ciclo, é feita a “inicialização”, na qual todas as condições iniciais, tais como a definição dos estados iniciais das memórias e configuração dos dispositivos, são estabelecidas. A partir deste ponto inicia-se o ciclo com a “Leitura das entradas”, estágio na qual os endereços definidos como entradas são lidos. Em seguida é feita a varredura do “Programa”, que realiza as alterações lógicas de acordo com as entradas obtidas, e, por fim,

realiza-se a “Atualização das saídas”, na qual as alterações lógicas necessárias são transmitidas para os endereços de saída. Este ciclo é repetido indefinidamente, ou até a chegada ao estado final requerido.

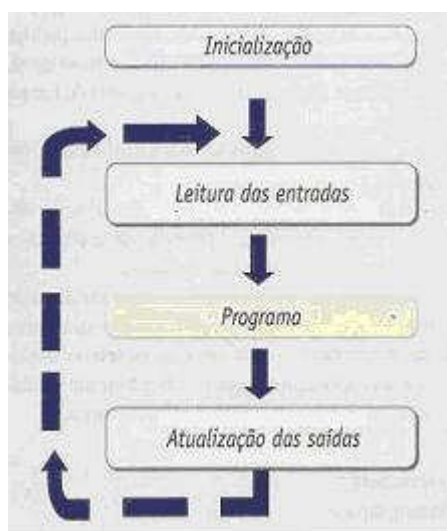


Figura 7. Ciclo de processamento de um CLP (PINTO, 2008)

Como exemplo de controlador lógico programável, será descrito a seguir o Siemens SIMATIC S7-300 com a CPU 315-2 DP (Figura 8), explicitando-se as suas partes principais.

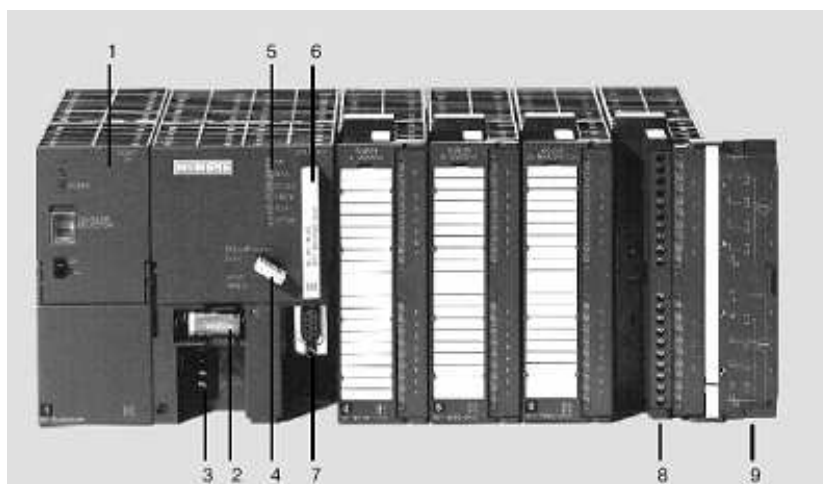


Figura 8. SIMATIC S7 300 design (PINHEIRO, 2006)

Na Figura 8 são ilustrados os seguintes componentes:

- 1-Fonte de energia
- 2-Bateria de reserva (CPU 313 e acima)
- 3-Conector 24V DC
- 4-Chave de operação (On/Off e Reset do CLP)
- 5-LEDs (Indicam status do sistema e falhas)
- 6-Cartão de memória (CPU 313 e acima)
- 7-MPI (Interface Multiponto)
- 8-Conector Frontal
- 9-Porta da frente (proteção dos conectores e identificação das ligações elétricas do módulo)

Este modelo tem integrado a si uma porta MPI e uma segunda interface de comunicação integrada PROFIBUS (PINHEIRO, 2006). Ele possui uma CPU que possibilita tempos de ciclo de até $1\mu\text{s}$ por instrução binária (Tabela 1) (KANEKO, 2008).

Tabela 1. Especificação do CPU 315-2 DP (KANEKO, 2008)

	CPU 315-2 DP
Memória principal	64 Kbyte
Tempos de processamento • Operações binárias • Operações com palavras	0,3 μs a 0,62 μs 1 μs
Faixas de endereçamento • Área total de endereçamento de E/S • Área de Imagem de processo E/S • Canais digitais totais • Canais analógicos totais	1/1 kbyte 128/128 byte máx. 8192 (1024 centrais) máx. 512(máx. 256 entradas ou 128 saídas centrais)
Interfaces Profibus DP • Número de segmentos DP integrado/CP342-5 • No. estações escravas DP por CPU (interfaces integradas / CP 342-5) • Velocidade de transmissão	1 / 1 64/64 12 Mbit/s

Junto ao CLP podem ser adicionados módulos de expansão que conferem uma versatilidade do Controlador frente à diversos tipos de aplicações (Tabela 2). Numa configuração centralizada podem ser utilizados até 32 módulos de expansão (KANEKO, 2008).

Tabela 2. Tipos de módulos de expansão para o SIMATIC S7-300 (KANEKO, 2008)

Módulos de I/O	- Digitais
	- Analógicos
Módulos de Comunicação	- Profibus DP
	- Ethernet
	- AS-interface
	- Serial Ponto-a-Ponto
	- Modbus
Módulos de Função	- Contadores rápidos
	- Controle de posição
	- Controle de motor de passo
	- Controle em malha fechada
	- Saídas de pulso rápida

2.7. PROFIBUS

O PROFIBUS ou *Process Field Bus* é o tipo de *fieldbus* mais bem sucedido do mundo, tendo mais de 28 milhões de equipamentos instalados no final de 2008. Por utilizar uma estrutura de “bus” de cabo único, essa tecnologia elimina excesso de cabos de transmissão de dados em linhas de produção e sistemas de automação em geral, reduzindo drasticamente custos com instalação (PROFIBUS, 2006). Seu projeto se iniciou em 1987 na Alemanha, como um consórcio entre diversas empresas e institutos para criação de um projeto estratégico *fieldbus*. O objetivo era a criação de um barramento de campo *bitserial*, com uma padronização de interfaces de dispositivos de campo. Inicialmente presente na automação de manufaturas, a partir de 1995 o PROFIBUS começou a ser implementado na automação de processos em geral (KANEKO, 2008).

No caso da presente aplicação, o padrão de meio de transmissão do PROFIBUS no Laboratório de Sistemas de Automação é o RS-485 de periferia

descentralizada, ou “PROFIBUS DP”. Esse sistema permite a operação conjunta de diversos subsistemas, diferenciados entre mestres e escravos. A Figura 9 ilustra uma instalação básica com comunicação de campo PROFIBUS.

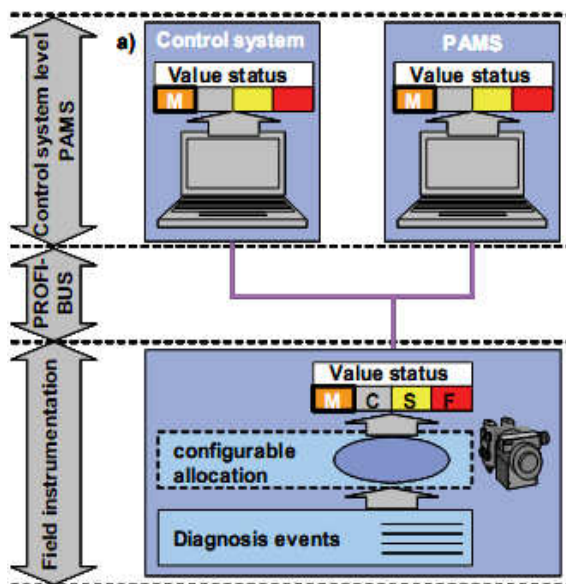


Figura 9. Estrutura de automação via PROFIBUS (PROFIBUS, 2006)

2.8 OPC

O OPC (*OLE for Process Control*) é composto por uma série de padrões de interface aberta para comunicação entre aplicações de Automação/Controle, Sistemas/Dispositivos de campo e aplicações de Negócios/Escritórios na indústria de controle de processo, e suas especificações são estabelecidas pela *OPC Foundation* (organização sem fins lucrativos, patrocinada pela Intellution, Microsoft e outras empresas).

O OPC é baseado nas tecnologias Microsoft de OLE (*Object Linking and Embedding*), COM (*Component Object Model*) e DCOM (*Distributed Component Object Model*) e é caracterizado por uma estrutura cliente/servidor, que funciona independente de linguagem de programação e suporta a integração e a operação em um sistema heterogêneo ou distribuído.

Quanto à transferência de dados, o servidor OPC atua como um “Portal de Software”, lendo informação dos dispositivos de campo e transformando-as

em dados OPC, com o propósito de integração dos diversos sistemas e dispositivos (LING,CHEN,YU, 2004).

Neste projeto, a comunicação entre uma aplicação e os dispositivos pode, portanto, ser feita utilizando um servidor OPC como um intermediador, que obtém as informações dos dispositivos via PROFIBUS e as disponibiliza como dados OPC.

3. Célula de manufatura didática

Para desenvolvimento do projeto, como citado na introdução, será utilizada a célula de manufatura didática (Mini CIM) presente no laboratório de Automação e Sistemas da Escola Politécnica.

O Mini CIM é usada para manufatura de determinadas peças: Cilindro de corpo rosa e pino preto, cilindro de corpo prateado (alumínio) e pino preto, e cilindro de corpo preto e pino cinza. A Figura 10 ilustra a composição de cada tipo diferente das peças:

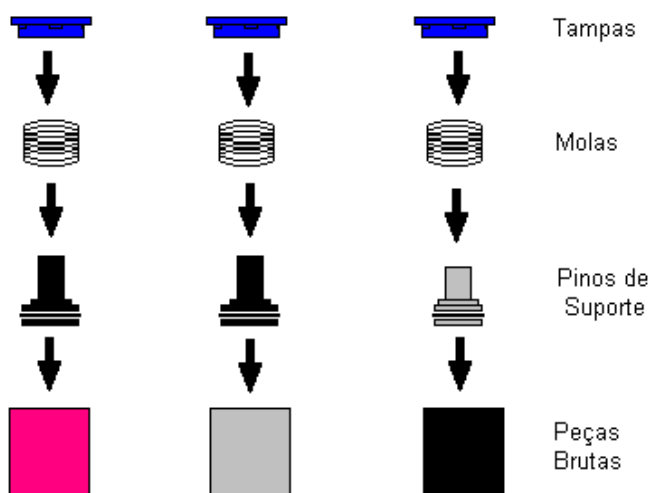


Figura 10. Montagem das diferentes peças pelo Mini CIM

A mini célula de manufatura é composta por 4 subestações, como ilustrado na Figura 11.

3.1. Estação de distribuição

É uma estação composta por um magazine com ejedor, alimentadores de cavidades vibratórios, transportadores inclinados e silos com dispositivos limitadores de alimentação.

Essa estação tem como funções:

- Separar uma peça do magazine de distribuição;
- Disponibilizar a peça para o processo subsequente (inspeção).

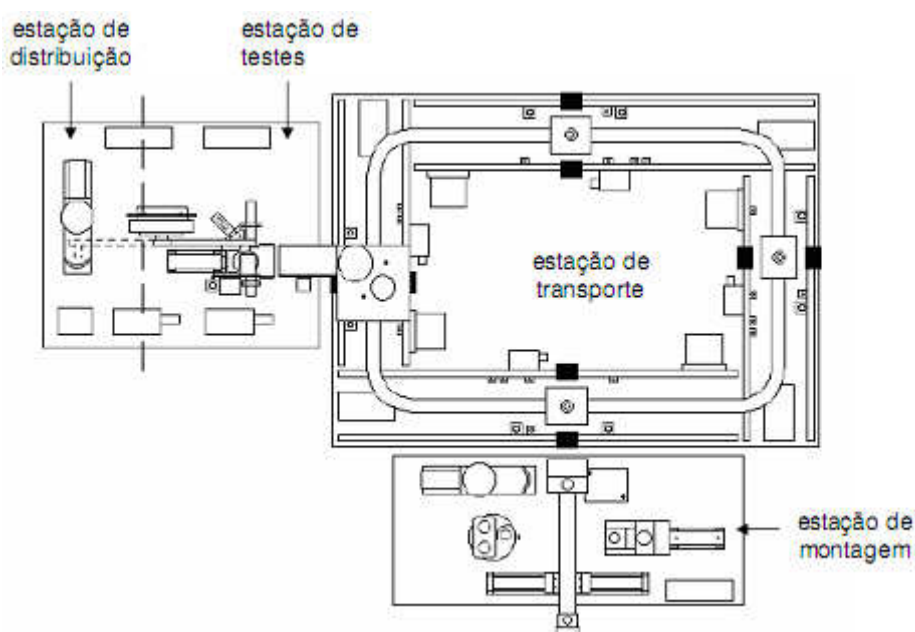


Figura 11. Esquema ilustrativo da mini célula de manufatura didática (Mini CIM)

3.2. Estação de inspeção ou de testes

A estação de inspeção adquire informação quanto a características físicas das peças, e faz uma distinção entre uma peça aceita para continuidade do processo ou uma peça rejeitada para tanto.

As funções da estação de inspeção são:

- Estabelecer as características do material da peça;
- Checar as dimensões da peça;
- Descartar ou disponibilizar uma peça para a estação subsequente.

Essa unidade é caracterizada por dois níveis. No nível inferior estão dispostos três sensores: indutivo, óptico e capacitivo. O sensor óptico é o único que permanece acoplado à plataforma elevatória, sendo que os demais são fixos na base da unidade. O sensor indutivo identifica peças metálicas, o sensor óptico identifica peças metálicas e rosas e o sensor capacitivo identifica a presença de peça, sendo que os sinais emitidos pelos sensores devem ser analisados em conjunto, para uma identificação correta da peça.

O nível superior realiza o teste de altura das peças por meio de uma haste com um atuador/cilindro pneumático e um sensor piezelétrico.

Após a realização dos testes, as peças aprovadas seguem para o carro transportador por meio de uma rampa (superior), ou são descartadas por uma rampa (inferior). A Figura 12 apresenta um esquema ilustrativo das estações de teste e distribuição.

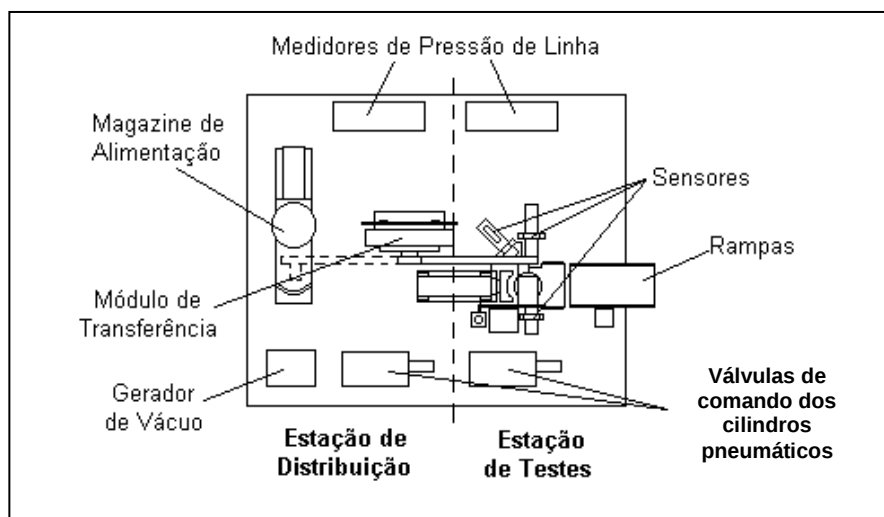


Figura 12. Estações de distribuição e testes

3.3. Estação de transporte

A estação de transporte ou sistema inteligente de transporte é um conjunto de esteiras que forma um circuito, sendo que, uma vez acionados os motores, as esteiras permanecem todas em movimento. No circuito há quatro posições de parada dos carros pré-determinadas, onde ocorrem carregamentos e descarregamentos. Em cada uma dessas posições há um conjunto de sensores de presença e travas de movimento para realização das tarefas previstas. Na presente montagem do Mini CIM, são utilizadas somente as duas paradas específicas (Figura 13), sendo que a primeira é destinada à estação de distribuição e testes, e a segunda destinada à estação de montagem.

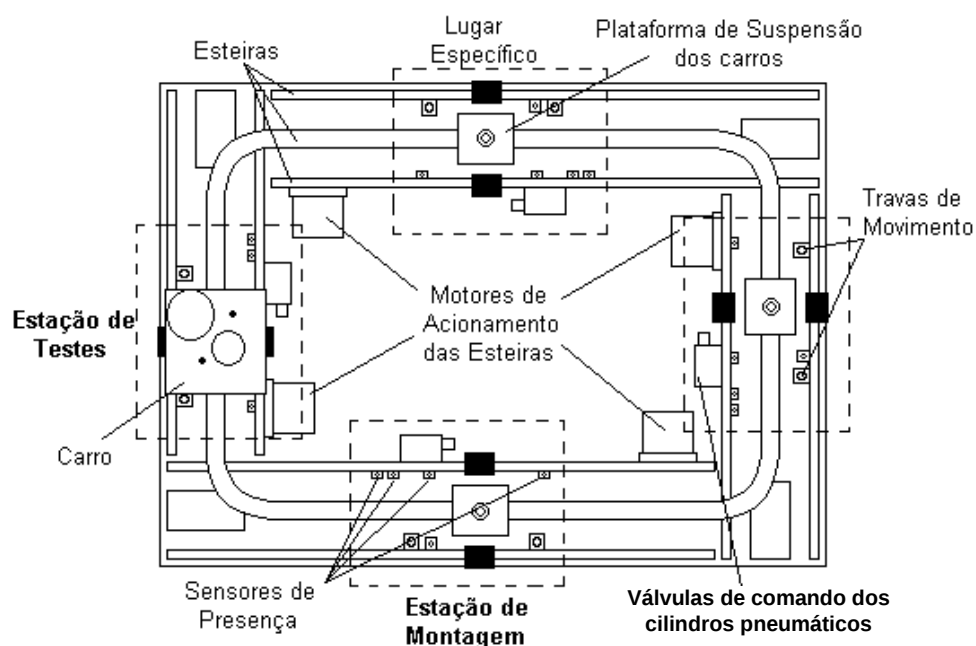


Figura 13. Detalhamento dos componentes da estação de transporte

3.4. Estação de montagem

A Figura 14 fornece uma ilustração geral dos componentes da unidade de montagem.

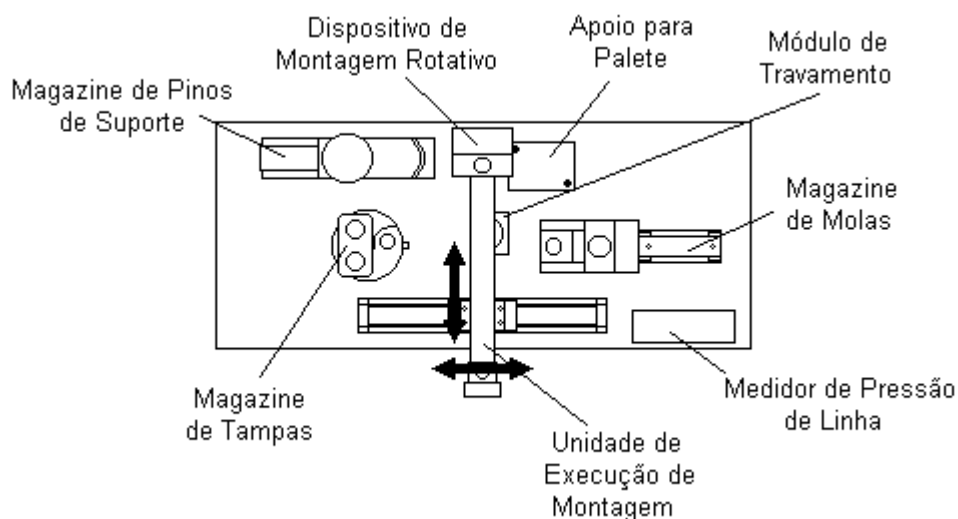


Figura 14. Detalhamento dos componentes da Estação de Montagem

Para realização do projeto de controle e supervisão desta estação são apresentados a seguir seus componentes constituintes e suas funcionalidades.

Magazine de molas - responsável pela disponibilização das mesmas (Figura 15).

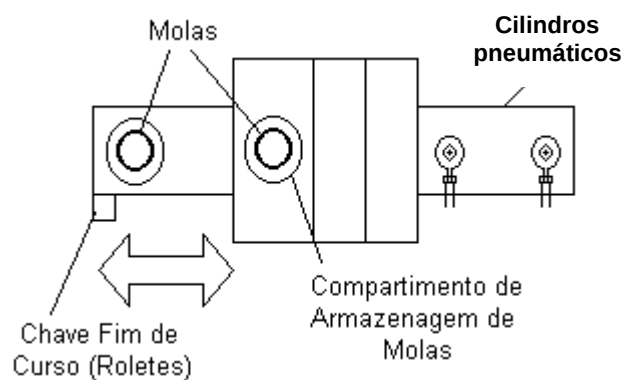


Figura 15. Magazine de molas

Magazine de pinos de suporte - responsável pela disponibilização de um dos dois tipos de pinos, preto e prateado (Figura 16).

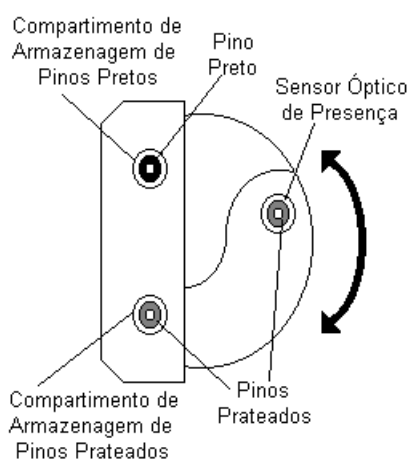


Figura 16. Magazine de pinos de suporte

Magazine de tampas - disponibiliza as tampas dos cilindros (Figura 17).

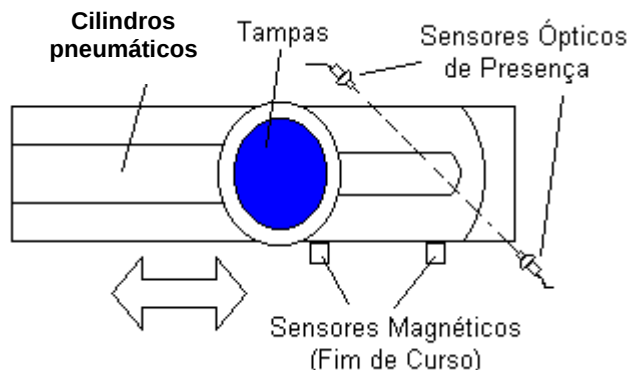


Figura 17. Magazine de tampas

Módulo de travamento - fixa o corpo da peça bruta à plataforma da estação (Figura 18).

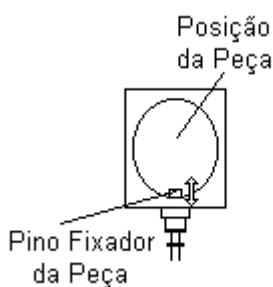


Figura 18. Módulo de travamento

Além dos módulos apresentados, há também uma unidade de montagem cartesiana, que constitui a estrutura de sustentação da garra dedicada. Para que a garra possa executar a montagem, esta possui 4 graus de liberdade: translação nos eixos x,y e z e uma rotação de sua garra em torno de z. A seguir, os atuadores e sensores envolvidos no acionamento das translações e rotações citadas são descritos:

Eixo x e y:

Nestes dois eixos a movimentação é feita por motores CC ligados a eixos “sem-fim”, que têm suas rotações detectadas por *encoders*, com o intuito de obter seu posicionamento em relação à posição inicial (Figuras 19 (a) e (b)). Para que estes *encoders* sejam interpretados é utilizado um módulo ligado

diretamente no CLP (Figura 20). Funcionando como dispositivos de segurança há chaves de fim de curso, que também possibilitam o posicionamento absoluto dos eixos.



Figura 19. (a) - Acionamento do eixo y; (b) - Acionamento do eixo x

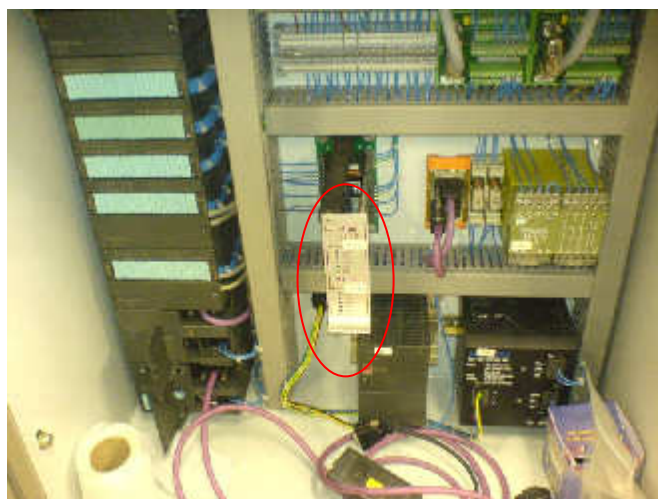


Figura 20. CLP e módulo de controle dos motores (destaque)

Eixo z:

Este eixo também é acionado por um motor CC e um “sem-fim”, mas não utiliza *encoder*. Para determinar a posição que este eixo se encontra durante o movimento existem 4 sensores de posição (Figura 21), além de chaves de fim de curso.



Figura 21. Eixo Z, sensores de posição e garra dedicada

A fim de realizar algumas ações, como por exemplo a montagem da tampa, a garra (Figura 22) possui uma rotação em z de 270° , proporcionada por um cilindro pneumático rotativo.



Figura 22. Detalhes do carro transportador

Esta estação possui, ainda, um apoio para palete, onde o carrinho é posicionado durante a montagem. Este apoio pode ser visto juntamente com os outros magazines e o módulo de travamento na Figura 23.

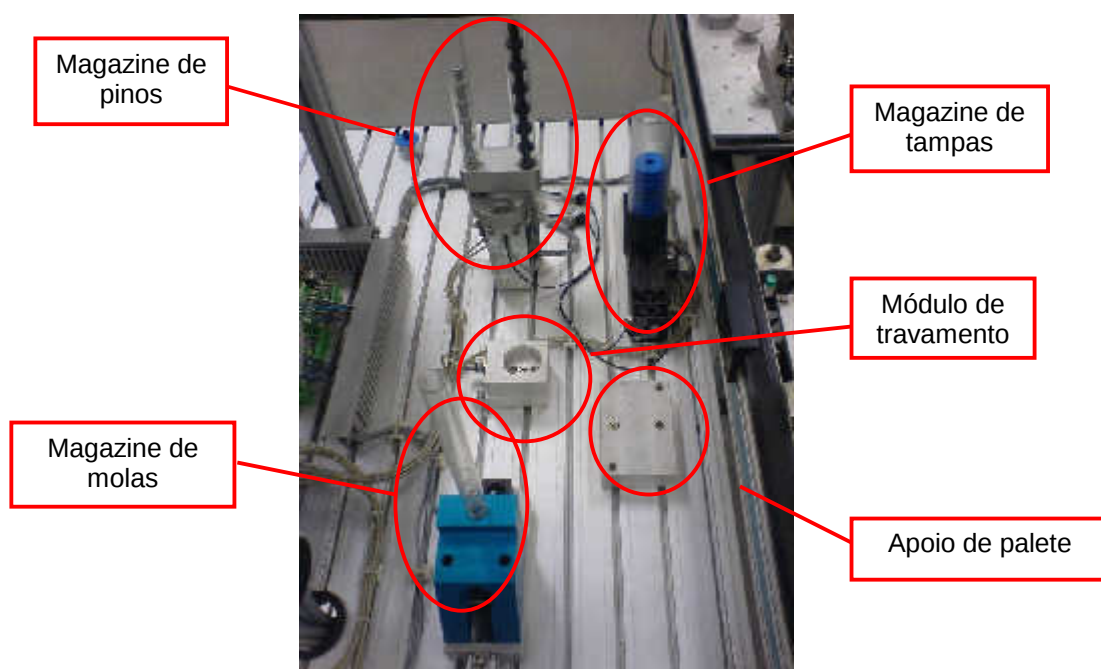


Figura 23. Magazines, apoio de paleta e módulo de travamento

4. Projeto

O projeto visa a implementação de uma solução para controle e supervisão de uma unidade de manufatura ou sistema produtivo via teleoperação. Para tanto, antes de implementar o sistema de teleoperação propriamente dito, foi necessário o mapeamento do processo produtivo referente à estação de montagem, foco deste trabalho, detalhando suas entradas (sensores) e saídas (atuadores) (item 4.1).

Posteriormente, foi feita a programação do CLP para controle de todos os atuadores do processo, utilizando a comunicação via MPI para transferência e testes desta programação. Para a comunicação entre o controlador local (PC munido de uma placa CP 5613 A2) e o CLP (a ele conectado um módulo CP 5613), foi utilizado o padrão de comunicação PROFIBUS. Para permitir o acesso remoto às funções programadas no CLP, fez-se necessário utilizar um servidor OPC (responsável por mapear variáveis de memória do CLP para aplicações Windows) e desenvolver um *WebService*, que se conecta a este servidor. Por sua vez, no *WebService* foram implementadas as funções que permitem, de fato, o acesso remoto à estação de montagem do Mini CIM. Houve também a necessidade do desenvolvimento de uma aplicação Web que disponibiliza uma interface para acesso ao Mini CIM via Internet. A interface se comunica com o *WebService*, via internet ou rede local, para obter informações do CLP ou modificar o *status* de suas memórias. Com esta arquitetura, pretende-se monitorar e/ou teleoperar a célula de manufatura do Laboratório de Sistemas de Automação, através da internet. A Figura 24 mostra uma visão geral da estrutura do projeto.

4.1. Funcionamento da estação de montagem

Quando o corpo da peça chega no carro de transporte (na posição determinada e acusada pelos sensores do sistema de transporte). O carro é então, suspenso pela garra dedicada e colocado sobre o apoio de palete. Neste ponto, a estação de montagem inicia sua rotina de funcionamento.

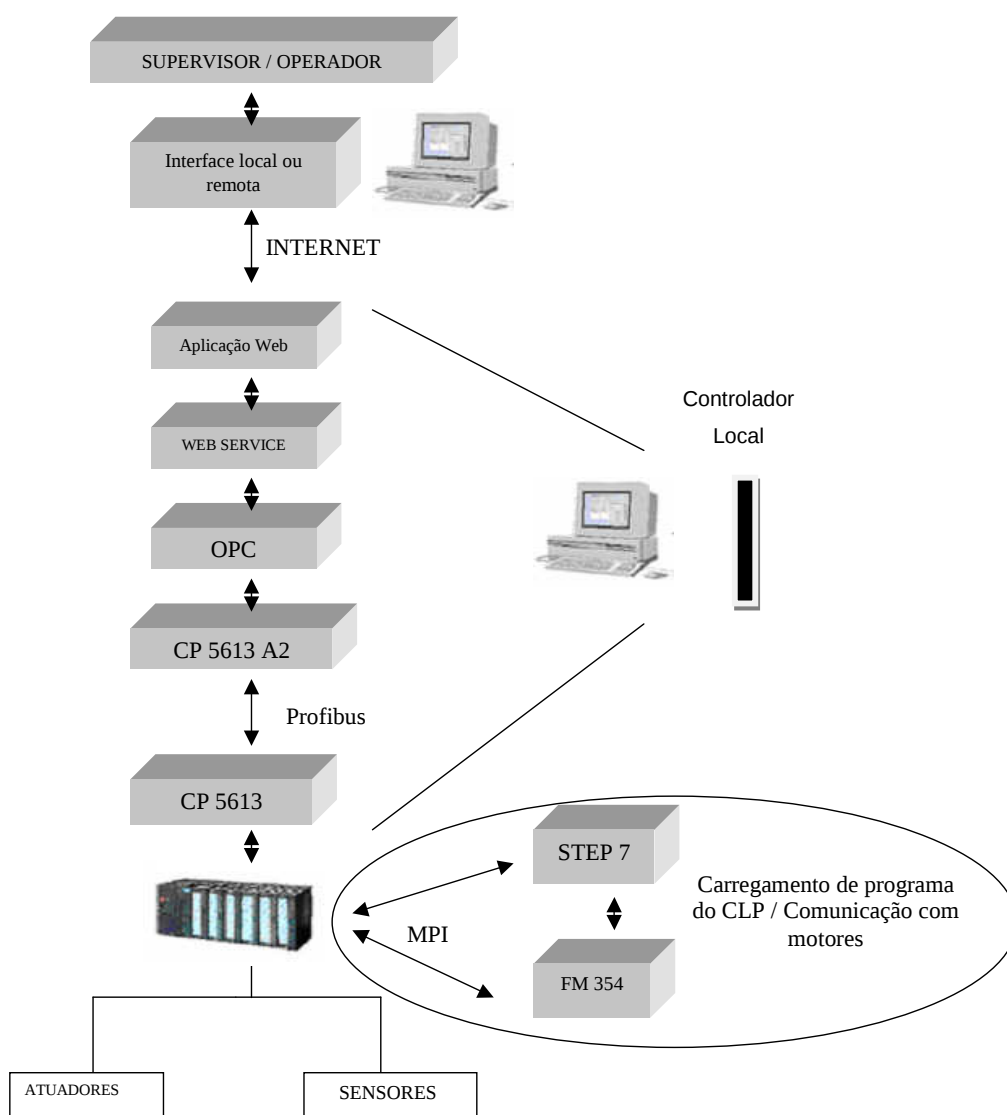


Figura 24. Estrutura de projeto

É então iniciada a montagem do sistema cilíndrico, com a possibilidade de haver duas variantes. Inicialmente, a unidade de execução de montagem cartesiana se move para a posição onde o corpo da peça bruta está pronto para ser pego.

O corpo cilíndrico é pego pela garra dedicada e então transferido para o módulo de travamento da estação de montagem (Figura 23). O corpo permanece fixado neste módulo pelo restante do processo de montagem.

A garra então retira um pino de suporte do magazine de pinos de suporte da estação de montagem, sendo ele de alumínio para peças pretas ou de plástico preto para peças prateadas ou rosas. O pino é então posicionado

sobre o módulo de travamento e em seguida pressionado sobre o corpo da peça.

A seguir, uma mola é retirada do magazine de molas e posicionada sobre o módulo de travamento, onde é montada junto ao corpo da peça. Uma tampa é extraída do magazine de tampas e encaminhada ao módulo de travamento, onde é rotacionada em 270° na peça, pelo dispositivo de montagem rotativo para que seja fixada no corpo da peça.

Completada a montagem da peça, ela é liberada pelo módulo de travamento e é posicionada mais uma vez, pela garra, sobre o carro de transporte, encerrando-se assim a rotina da estação de montagem, quando o carro deixa o local específico correspondente a esta estação.

4.1.1. Diagrama SFC da sequência de montagem para a garra

Com a descrição do item 4.1 foi possível desenvolver um diagrama em linguagem SFC para descrever os eventos de montagem da peça. Este pode acompanhada passo a passo no digrama da Figura 25, juntamente com as Tabelas 3 e 4.

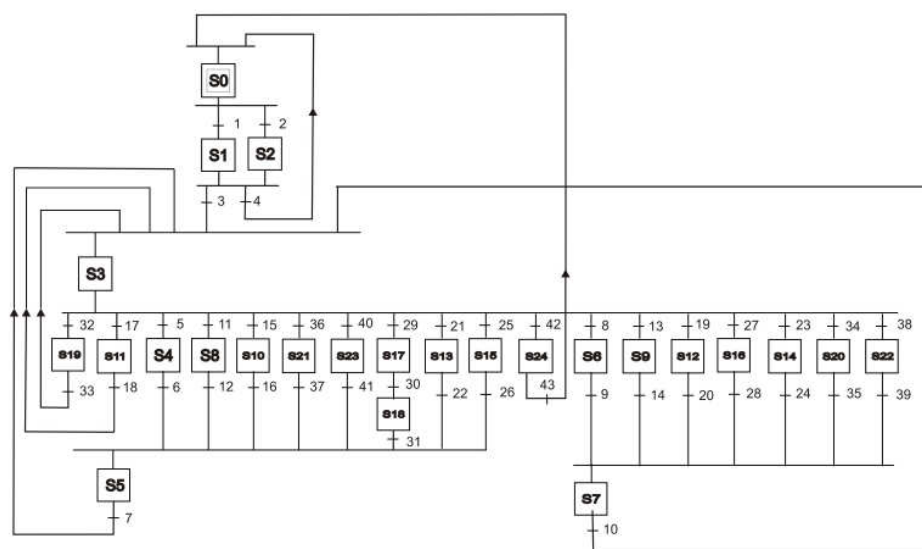


Figura 25. Diagrama SFC da sequência de montagem (garra)

Tabela 3. Estados do diagrama de controle

Estado	Descrição
s0	Estação em repouso
s1	Montagem com pino preto (aciona pistões dos magazines)
s2	Montagem com pino cinza (aciona pistões dos magazines)
s3	Escolha da etapa de montagem
s4	Move a garra até estação de transporte (apenas em x e y)
s5	Abre garra
s6	Move a garra até estação de transporte (em z)
s7	Fecha garra
s8	Move o palete até apoio do palete
s9	Move a garra até o corpo
s10	Move o corpo até estação de travamento (alinhamento corpo)
s11	Trava o corpo
s12	Move a garra até magazine de pinos
s13	Move o pino até estação de travamento (alinhamento pino)
s14	Move a garra até magazine de molas
s15	Move a mola até estação de travamento (alinhamento mola)
s16	Move a garra até magazine de tampas
s17	Move a tampa até estação de travamento (sem abrir a garra)
s18	Rotaciona a garra
s19	Destrava peça montada
s20	Move a garra até a peça montada (em z)
s21	Move a peça até local de armazenamento da peça
s22	Move a garra até o pino do palete
s23	Move o palete até a estação de montagem
s24	Retorna a garra para a posição inicial (translação e rotação)

4.2. Projeto do software de controle do CLP

Para a programação das funções básicas do CLP, procurou-se manter as funções bem simples, realizando apenas a ativação de cada saída e deixando os inter-travamentos e lógicas de movimentação para serem programados em camadas superiores do projeto, mais especificamente no *WebService*.

4.2.1. Mapeamento das entradas e saídas do CLP

Os eventos que ocorrem durante a montagem da peça são monitorados e tratados pelo CLP. Com esta finalidade este recebeu uma programação e teve suas saídas e atuadores mapeados. A estratégia utilizada para tratamento dos eventos foi a disponibilização de endereços de memória, iniciados por “M”, que tem como objetivo ativar funções pré-programadas no CLP. A seguir (Tabela 5), pode-se encontrar a listagem com as entradas (iniciadas por “I”), as saídas (iniciadas por “Q”) e as memórias para controle (iniciadas por “M”).

Tabela 4. Transições do diagrama na ordem de ocorrência

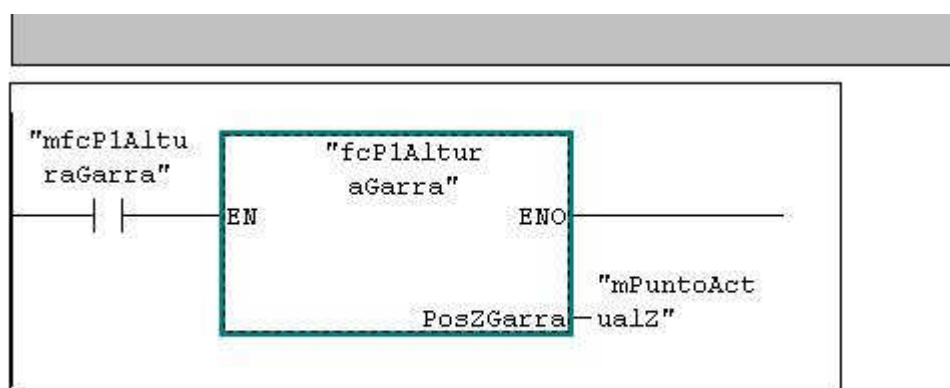
Transição	Descrição
1	Pronto para montar (pino preto)
2	Pronto para montar (pino cinza)
3	Todos os sensores detectaram as partes necessárias
4	Falta parte (retorna ao início)
5	Ir para estação de montagem
6	Estação de montagem atingida
7	Garra aberta
8	Descer até o corpo (estação de montagem)
9	Corpo atingido
10	Garra fechada
11	Ir até o apoio de palete
12	Apoio do palete atingido
7	Garra aberta
13	Ir até o local de armazenamento do corpo
14	Local de armazenamento atingido
10	Garra fechada
15	Ir até estação de travamento (alinhamento corpo)
16	Estação de travamento atingida
7	Garra aberta
17	Acionar pistão de travamento do corpo
18	Pistão de travamento acionado
19	Ir até magazine de pinos
20	Magazine de pinos atingido
10	Garra fechada
21	Ir até estação de travamento (alinhamento pino)
22	Estação de travamento atingida
10	Garra aberta
23	Ir até magazine de molas
24	Magazine de molas atingido
10	Garra fechada
25	Ir até estação de travamento (alinhamento mola)
26	Estação de travamento atingida
7	Garra aberta
27	Ir até magazine de tampas
28	Magazine de tampas atingido
10	Garra fechada
29	Ir até estação de travamento (alinhamento tampa)
30	Estação de travamento atingida
31	Garra rotacionada para 0°
7	Garra aberta
32	Destruar pistão de travamento do corpo
33	Pistão de travamento destravado
34	Mover em z até a peça montada
35	Peça atingida
10	Garra fechada
36	Ir até o local de armazenamento do corpo
37	Local de armazenamento atingido
7	Garra aberta
38	Ir até o pino de transporte do palete
39	Pino do palete atingido
10	Garra fechada
40	Ir para estação de montagem
41	Estação de montagem atingida
7	Garra aberta
42	Voltar garra para posição inicial
43	Garra na posição inicial

Tabela 5. Variáveis do CLP

Nome	Acesso	Endereço	Tipo	Comentários
EmUso	M	255.2	BOOL	Indica uso do CLP
iAbrirGarra	I	16.0	BOOL	Sensor de garra aberta
iCilindroMolaRecuado	I	20.2	BOOL	Sensor de cilindro de mola recuado
iCilindroTampaEstendido	I	20.5	BOOL	Sensor de cilindro de tampa estendido
iCilindrotampaRecuado	I	20.4	BOOL	Sensor de cilindro de tampa recuado
iFechadaGarra	I	16.1	BOOL	Sensor de garra fechada
iforneceMola	I	20.3	BOOL	Informa que a mola foi fornecida
iFornecePino	I	20.7	BOOL	Informa que o pino foi fornecido
iForneceTampa	I	20.6	BOOL	Informa que a tampa foi fornecido
iGarra0	I	16.2	BOOL	Sensor de garra em 0 grau
iGarra270	I	16.3	BOOL	Sensor de garra em 270 graus
iGarraZ1	I	16.4	BOOL	Detecta a garra no ponto superior
iGarraZ2	I	16.5	BOOL	Detecta a garra no ponto intermediário superior
iGarraZ3	I	16.6	BOOL	Detecta a garra no ponto intermediário inferior
iGarraZ4	I	16.7	BOOL	Detecta a garra no ponto inferior
InitXY	M	255.0	BOOL	Habilita movimentação inicial em XY
iPinoPrata	I	20.1	BOOL	Sensor de pino prata
iPinoPreto	I	20.0	BOOL	Sensor de pino preto
mAbrirGarra	M	30.2	BOOL	Comando de abrir garra
mDesligarTrava	M	30.1	BOOL	Comando de Desligar Trava
mEmergencia	M	255.4	BOOL	Ativa emergência
mError	M	25.0	BOOL	Desliga indicativo de erro
mEstenderCilindroMola	M	31.1	BOOL	Comando de tirar mola do buffer
mEstenderCilindroTampa	M	30.6	BOOL	Comando de tirar tampa do buffer
mFcP0XYConfiguracao	M	20.2	BOOL	Configuração inicial de XY
mfcP1AlturaGarra	M	21.0	BOOL	Ativa a função de posição inicial no plano XY
mFcPXYConfiguracao	M	25.4	BOOL	Configuração de XY
mFechargarra	M	30.3	BOOL	Comando de fechar a garra
mGarra0	M	30.4	BOOL	Comando de girar garra para 0 graus
mGarra270	M	30.5	BOOL	Comando de girar garra para 270 graus
mLigarTrava	M	30.0	BOOL	Comando ligar trava
mMotoresXYP0Tmp	M	25.1	BOOL	Variável auxiliar
mMotoresXYP1Tmp	M	25.2	BOOL	Variável auxiliar
mMovimentarXY	M	20.0	BOOL	Comando de movimentar XY
mMovimentarZ	M	21.1	BOOL	Comando de movimentar Z
MovXY	M	255.1	BOOL	Habilita mov. De XY
mPinoPrata	M	31.3	BOOL	Comando de tirar pino prata do buffer
mPinoPreto	M	31.2	BOOL	Comando de tirar pino preto do buffer
mPontoAtualZ	MW	40	WORD	Ponto atual da garra em Z
mPontoAtivoZ	MW	48	INT	Ponto ativo em Z
mPontoAtualXY	MW	46	WORD	Ponto Atual em XY
mPontoDesejadoXY	MW	44	WORD	Ponto desejado em XY
mPontoDesejadoZ	MW	42	WORD	Ponto desejado da garra em Z
mPontoXYAlcancado	M	20.1	BOOL	Indica que a garra esta no ponto desejado no plano XY
mPontoZAlcancado	M	21.2	BOOL	
mRecuarCilindroMola	M	31.0	BOOL	Comando de recuar cilindro de mola
mRecuarCilindroTampa	M	30.7	BOOL	Comando para recuar cilindro de tampa
qAbrirgarra	Q	16.0	BOOL	Atuador para abrir garra
qBaixarZ	Q	16.5	BOOL	Atuador para baixar a garra
qEstenderCilindroTampa	Q	20.5	BOOL	Atuador para estender cilindro de tampa
qFecharGarra	Q	16.1	BOOL	Atuador para fechar garra
qGarra0	Q	16.2	BOOL	Atuador para colocar a garra em 0 grau
qGarra270	Q	16.3	BOOL	Atuador para colocar a garra em 270 graus
qPinoPrata	Q	20.1	BOOL	Atuador para tirar o pino prata
qPinoPreto	Q	20.0	BOOL	Atuador para tirar o pino preto
qRecuarCilindroMola	Q	20.2	BOOL	Atuador para recuar cilindro mola
qRecuarCilindroTampa	Q	20.4	BOOL	Atuador para recuar cilindro tampa
qTravaCorpo	Q	20.7	BOOL	Atuador para estender seguro
qSubirZ	Q	16.4	BOOL	Atuador para subir a garra

4.2.2. Especificação do software de controle do CLP

O *software* de controle do CLP é constituído de blocos de controle (vide Apêndice C) que contem entradas e saídas. Estes blocos são programados em *Ladder* e possuem a lógica necessária para tratamento das entradas. Os principais blocos utilizados são aqueles que se referem a ativação das funções dos atuadores do sistema, entre eles esta, por exemplo, o bloco de inicialização do motor do eixo Z, que leva a garra para a posição mais alta, este bloco pode ser visto na Figura 26.



Network 8 : Title:

Figura 26. Bloco de inicialização do eixo Z (*Ladder*)

Cada bloco contém sua própria lógica interna, por exemplo, a lógica de acionamento do motor do eixo Z pode ser vista na Figura 27. Esta lógica se baseia na leitura do sensor da posição Z1 (mais alta), mantendo o motor ligado no sentido de subida até que este sensor seja ativado. Ao terminar a saída que indica o ponto atual de Z é atualizada para 1. Observe, também, que existe uma chave de emergência, para que o motor possa ser desativado quando necessário.

O restante da programação do CLP pode ser encontrado no Apêndice C.

4.3. Mapeamento das variáveis no servidor OPC

Todas as entradas e memórias (I e M respectivamente) foram mapeadas para o servidor OPC por meio do *software* OPCScout, que utiliza o nome e

endereço das variáveis (Tabela 5) para mapeá-las. As saídas (Q) não foram mapeadas para o servidor OPC porque o acesso aos atuadores é feito de forma indireta, por meio das memórias (M) definidas no CLP. Assim, o acesso a estas variáveis se torna fácil e dinâmico, uma vez que basta a aplicação utilizar um método de leitura/escrita do servidor para obter/modificar o status da variável.

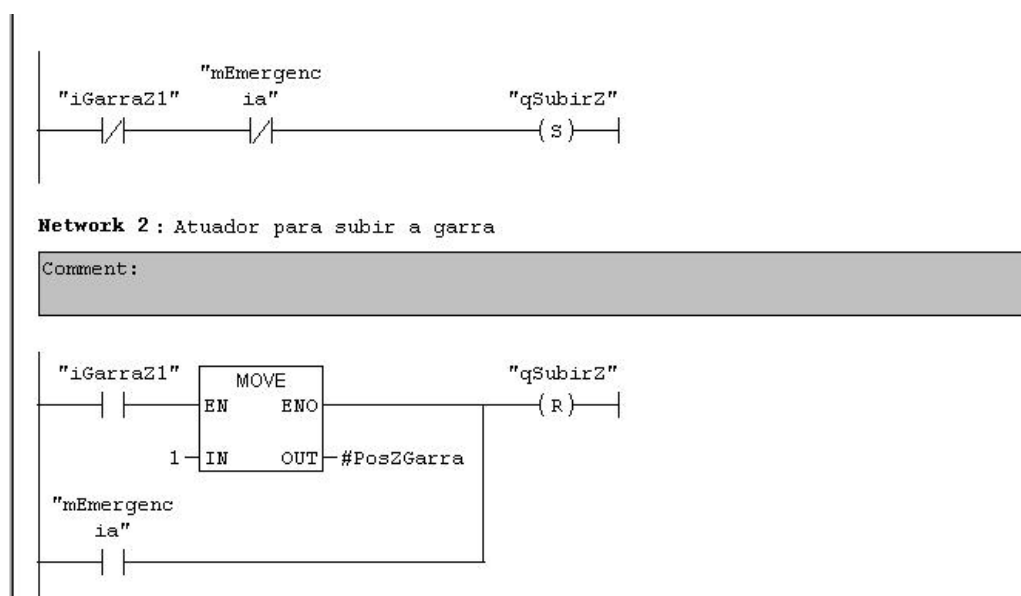


Figura 27. Lógica interna do bloco de inicialização do eixo Z (Ladder)

4.4. WebServices

O *WebService* é um dos componentes principais deste projeto, já que é por meio dele que as funções do Mini CIM serão disponibilizadas na internet, de forma padronizada e simples. O acesso às informações (*status* dos sensores e atuadores) e o acionamento dos atuadores são feitos por meio do servidor OPC, ou seja, é este *WebService* que fará o papel de cliente do servidor OPC e, portanto, utilizará as classes e métodos de comunicação OPC.

Como informado anteriormente é nesta camada que estarão as funções que encapsulam os comandos do CLP e possibilitam o controle das funções deste modelo de manufatura.

Adotou-se o C# (CSharp) como linguagem de programação, e a ferramenta Visual Studio 2008. Os protótipos das funções desenvolvidas (*WebMethods*), seguidos de suas descrições, estão listados abaixo:

- `public string CorPino(bool prata)`: Escolhe o pino que será utilizado na montagem. Se “prata” é verdadeiro, disponibiliza o pino prata, caso contrário o preto;
- `public string EmUsoread()`: Esta função faz a leitura da variável que indica se o CLP está em uso. Retorna o valor desta variável.
- `public string EmUsowrite (bool uso)`: Escreve nesta variável “true” ou “false”, indicando em uso ou não, respectivamente.
- `public string Emergência (bool parar)`: Ativa a emergência, desativando todos os motores.
- `public string Init()`: Leva a garra a sua posição inicial e verifica se a estação de montagem está ocupada;
- `public string MoveXY(string pos)`: Movimenta a garra no eixo Z, até a posição indicada pela entrada “pos”. As possíveis posições estão na Tabela 6;
- `public string MoveZ(string pos)`: Movimenta a garra no eixo Z, até a posição indicada pela entrada “pos” (de 1 a 4, sendo 1 a posição mais acima e 4 a mais abaixo);
- `public string PosCilindroMola(bool avançar)`: Realiza o avanço do cilindro quando o entrada é verdadeira e recua quando falsa;
- `public string PosCilindroTampa(bool avançar)`: Realiza o avanço do cilindro quando o entrada é verdadeira e recua quando falsa;
- `public string[] Read()`: Esta função faz a leitura de todas as variáveis de entrada do CLP. Armazenando o status dos sensores em um vetor para a sua utilização no controle do processo.
- `public string SentidoGarra(bool p270)`: Movimento giratório da garra. Se a entrada é verdadeira a garra vai para posição de 270°, caso contrário vai para 0°;
- `public string StatusGarra(bool abrir)`: Abertura da garra. Se “abrir” é verdadeiro, a garra abre, caso contrário fecha;

- public string StatusTrava(bool travar): Trava a peça corpo no módulo de travamento quando a entrada é verdadeira e libera quando falsa;
- public string Zera(): Realiza as atividades iniciais, necessárias para início do processo, tais como: recuar todos os cilindros e iniciar a garra aberta e com rotação de 270°.

Tabela 6. Possíveis posições dos eixos x e y (KANEKO, 2008)

Posição	motor 1(mm)	motor 2 (mm)	Descrição
1	70	80	Configuração inicial
2	238.395	55.49	Garra sobre estação de transporte
3	106.64	268.16	Coloca carro na estação de montagem
4	213.95	308.9	Retira peça do carro
5	252.59	407.425	Coloca peça na base de montagem
6	485.25	437.355	Pega pino
7	177.92	409.385	Coloca pino na base de montagem
8	13.145	429.385	Pega mola
9	321.18	406.14	Coloca mola na base de montagem
10	502.635	243.815	Pega tampa
11	251.32	408.88	Coloca tampa na base de montagem
12 = 4	213.95	308.9	Coloca peça no carro
13 = 3	106.64	268.16	Retira carro da estação de montagem
14 = 2	238.395	55.49	Garra sobre estação de transporte

Durante o desenvolvimento deste *WebService* foi utilizado um arquivo XML, para tabelar os endereços de cada variável no servidor OPC, utilizado como canal de comunicação entre o CLP e o *WebService*. Este arquivo XML pode ser encontrado em anexo, denominado de “opcconfig.xml” (Apêndice A).

4.4.1. Testes e utilização do *WebService*

Para teste da utilização do *WebService* criado foi utilizado, inicialmente, o acesso direto por meio de um navegador de internet, desta forma é possível utilizá-lo de forma simples, pois a ferramenta de desenvolvimento (Visual Studio) cria automaticamente uma interface Web. As Figuras 28, 29 e 30 apresentam exemplos de validação via acesso direto.

Na Figura 28 é possível observar que todas as funções são disponibilizadas de forma direta e podem ser acessadas como exemplificado

na Figura 29. Note que o campo para a entrada “pos” é gerado automaticamente.

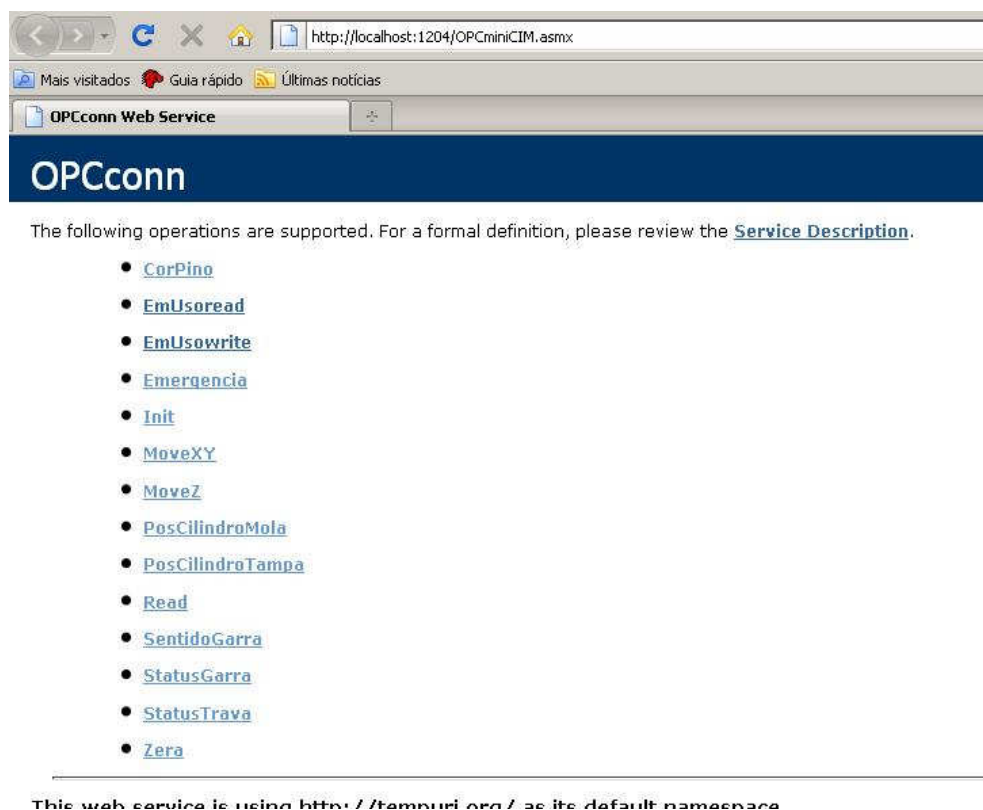


Figura 28. Lista de funções do WebService (acessado via navegador)

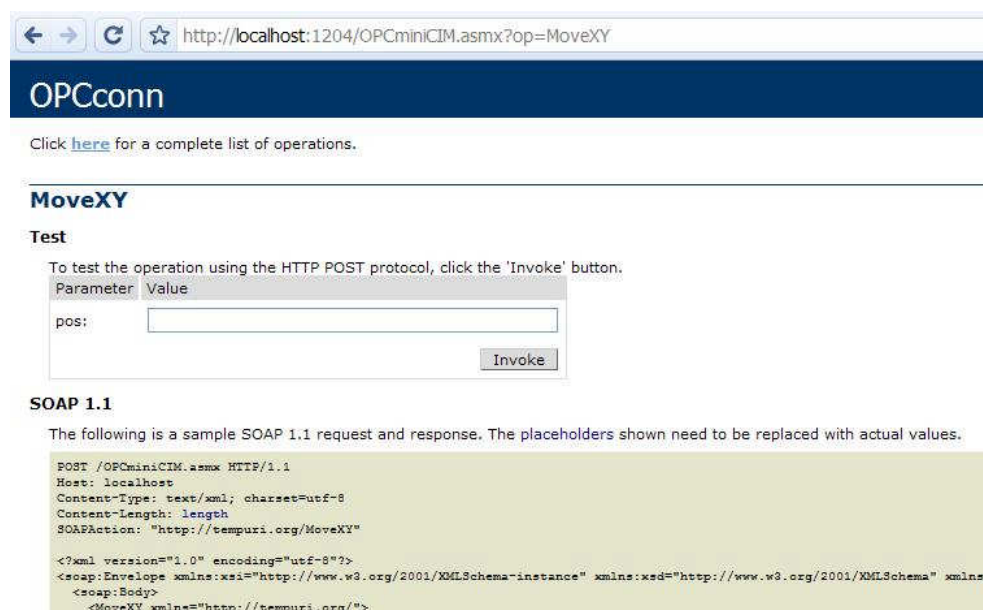


Figura 29. Acesso às funções do WebService (acessado via navegador)

Na interface Web elaborada deste projeto (ver item 4.5) o acesso a este *WebService* é feito de forma direta, adicionando o projeto do *WebService* como referência Web do projeto do Visual Studio (Figura 30) e instanciar um objeto pertencente ao grupo determinado da construção deste *WebService* para acessar funções.



Figura 30. Adicionando uma “Web Reference” (Visual Studio)

4.5. Interface web

Para desenvolvimento da interface gráfica foi necessário realizar previamente um estudo de caso. Para tanto foi desenvolvido um diagrama de casos de uso para Operação/Monitoração, apresentado na Figura 31.

No diagrama de casos de uso da Figura 31 estão ilustradas as funções que serão exercidas pelos atores da interface a ser criada via *WebService*.

Neste projeto, são considerados atores os usuários e o Servidor da aplicação Web e pelo CLP, atuando de forma conjunta. Os usuários são representados pelos operadores e monitores do Mini CIM, ou seja, pessoas que terão, respectivamente, acesso as funções de operação, como acionamento dos motores para realização das atividades, e de monitoração, como a verificação do *status* dos sensores e monitoração por meio de uma

câmera. O Servidor/CLP é representado pelo *hardware* responsável por obter dados do sistema e atuar sobre este.

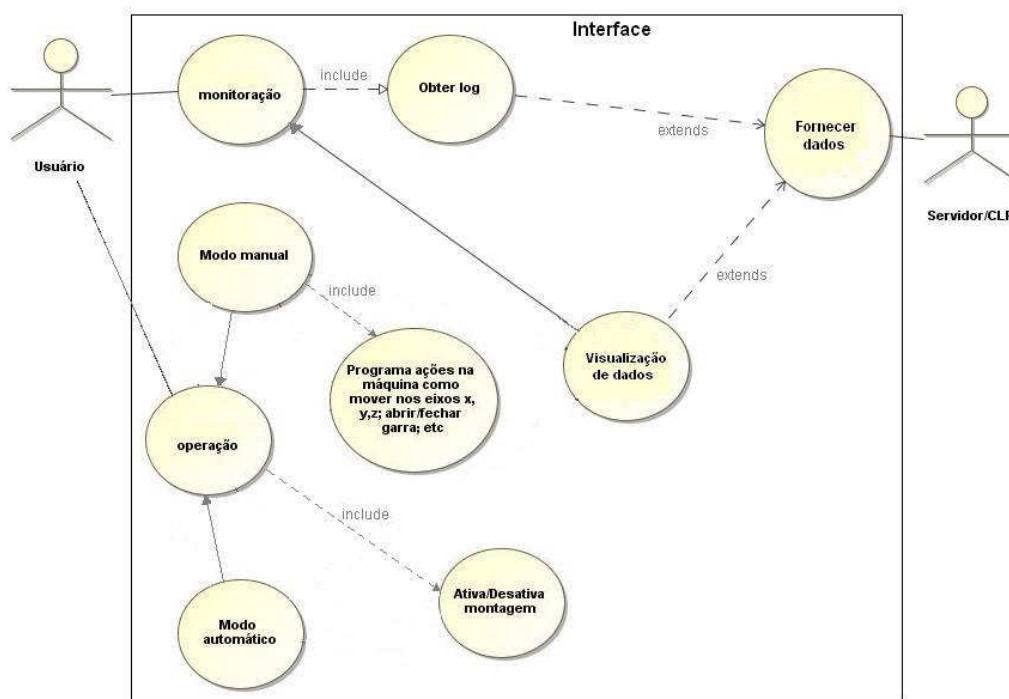


Figura 31. Diagrama de casos de uso da interface (Operação/Monitoração)

A seguir, a descrição dos casos de uso:

- **Usuários:** os casos de uso que estes dois (operadores e monitores) têm em comum é a monitoração, que tem a funcionalidade de poder gerar *logs* e é a generalização da visualização da interface gráfica. No entanto somente o operador pode realizar o próximo caso de uso: a operação, que inclui a opção de liga e desliga e representa a generalização de dois modos: o manual, que inclui a funcionalidade de programar ações para a célula de manufatura, e o automático. Vale observar que os usuários podem acessar a interface via internet ou via endereço local, direto no servidor, mas os casos de uso serão os mesmos;
- **Servidor/CLP:** este ator terá como caso de uso o fornecimento de dados. Este caso de uso é estendido pela geração de *logs* e visualização dos dados na interface, uma vez que não depende destes últimos dois casos para funcionar.

Já para o desenvolvimento da interface para o Supervisor, foram considerados os seguintes casos de uso (Figura 32):

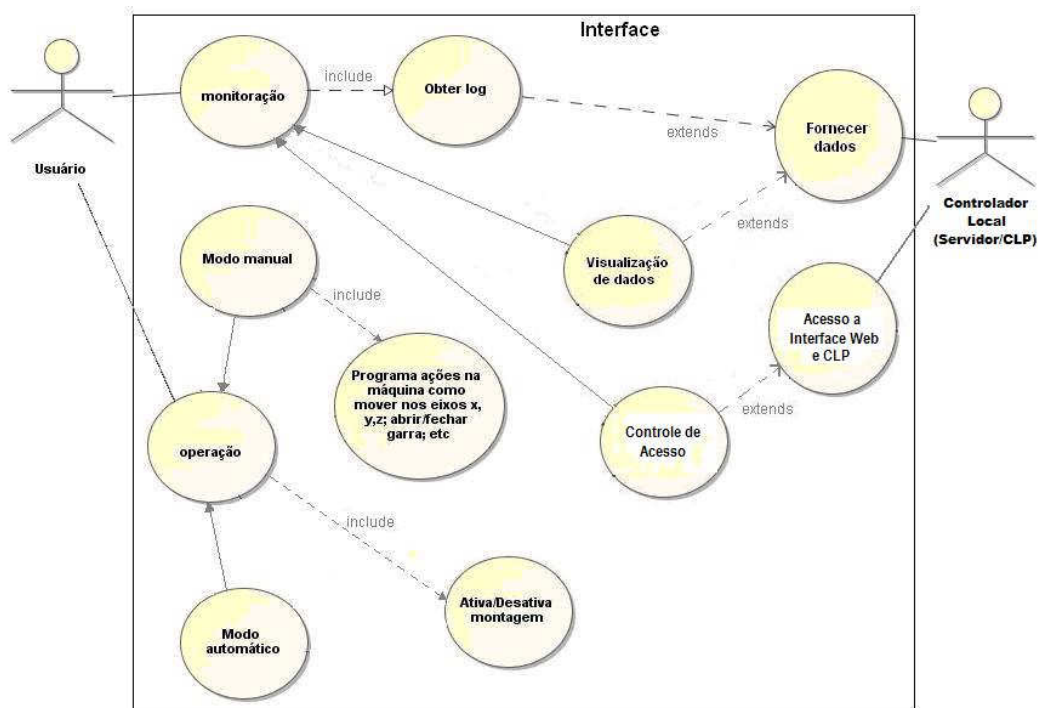


Figura 32. Diagrama de casos de uso da interface (Supervisão)

Observe que, para este caso, foram incluídas atividades tanto para o usuário quanto para o Servidor/CLP. Neste diagrama o usuário é representado pelo supervisor do Mini CIM, que terá acesso a todas as funções deste sistema de teleoperação do Mini CIM, ou seja, operação, monitoração e controle de acessos de usuários. O servidor/CLP, ainda é representado pelo *hardware* do sistema.

A seguir, os casos de uso adicionados neste diagrama:

- Para os usuários foi adicionado o "Controle de Acesso", que é generalizado, também, pela monitoração e tem a função de criar/excluir contas de usuário, além de acompanhar qual usuário está *on-line* e permitir/restringir acesso ao Mini CIM;
- Para o Servidor/CLP foi adicionado o "Acesso a Interface Web e ao CLP", que é o mecanismo de acesso (*login/logout*) às funcionalidades do Mini CIM. Este caso de uso é estendido pelo Controle de Acesso, uma vez que não depende deste último caso para funcionar.

4.5.1. Acesso à interface

A interface deve ser acessada por meio de um navegador de internet, utilizando o endereço IP do Controlador local. Após entrar na página inicial será requisitado ao usuário que ele faça o *login*, para que possa acessar as próximas páginas. A Figura 33 mostra as opções de acesso de um usuário registrado.



Figura 33. Acesso de usuários registrado (opções de acesso)

Dependendo do usuário que está acessando o sistema, as páginas que terão seu acesso liberado serão diferentes, isto ocorre devido ao uso de um sistema de grupos de acesso (*roles*).

Os grupos de acesso (*roles*) estão definidos da seguinte forma:

- Monitor: o usuário pertencente a este grupo possui acesso apenas à página de Monitoração, na qual ele pode acompanhar o processo e o *status* dos sensores.
- Operador: o usuário pertencente a este grupo possui acesso às páginas do grupo “monitor”, mais o acesso as funções de operação da estação de montagem do Mini CIM. Este usuário acessa a página de operação para acesso as funcionalidades.

- Supervisor: o usuário pertencente a este grupo tem acesso a todas as paginas do grupo “operador” mais o controle de usuários, e habilitação/desabilitação do CLP. Este usuário acessa a página do Supervisório para acesso às funcionalidades.

Em resumo, o acesso às funcionalidades é definido seguindo a ordem do diagrama da Figura 34.

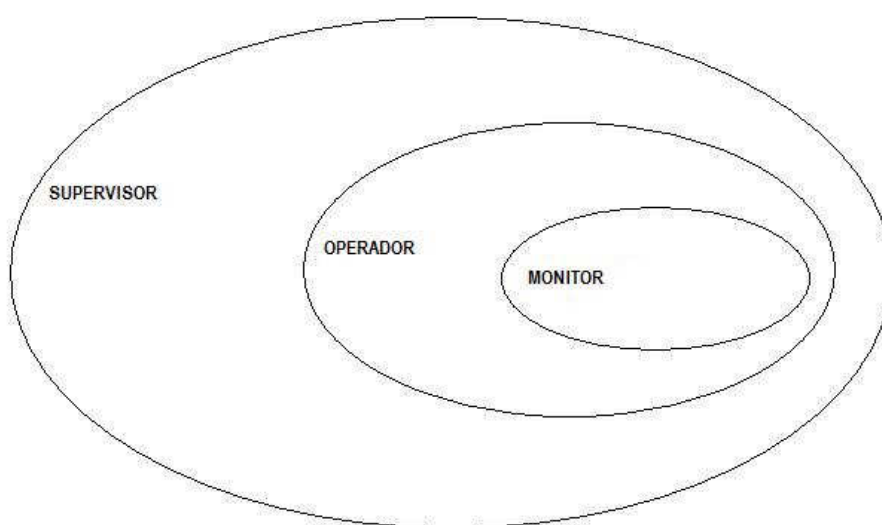


Figura 34. Hierarquia de acesso

4.5.2. Utilização da interface (Supervisório, Operação e Monitoração)

Após o *login* e a escolha da página a ser acessada, a interface será exibida, disponibilizando as funções que podem ser acessadas pelo usuário. Na Figura 35 pode ser vista a interface de um supervisor.

Nesta página (Supervisório), para operação da estação de montagem do Mini CIM, é necessário clicar sobre o campo “Ativar Operação” para habilitar a operação (Figura 36). Neste momento será realizada a rotina “Zera” do *WebService* (item 4.4), que leva os cilindros e a garra para a posição inicial. Após isto será liberado o botão iniciar, que, ao ser pressionado, realiza a sequência de sincronização dos motores de X e Y, além de enviar a garra para a posição mais alta.

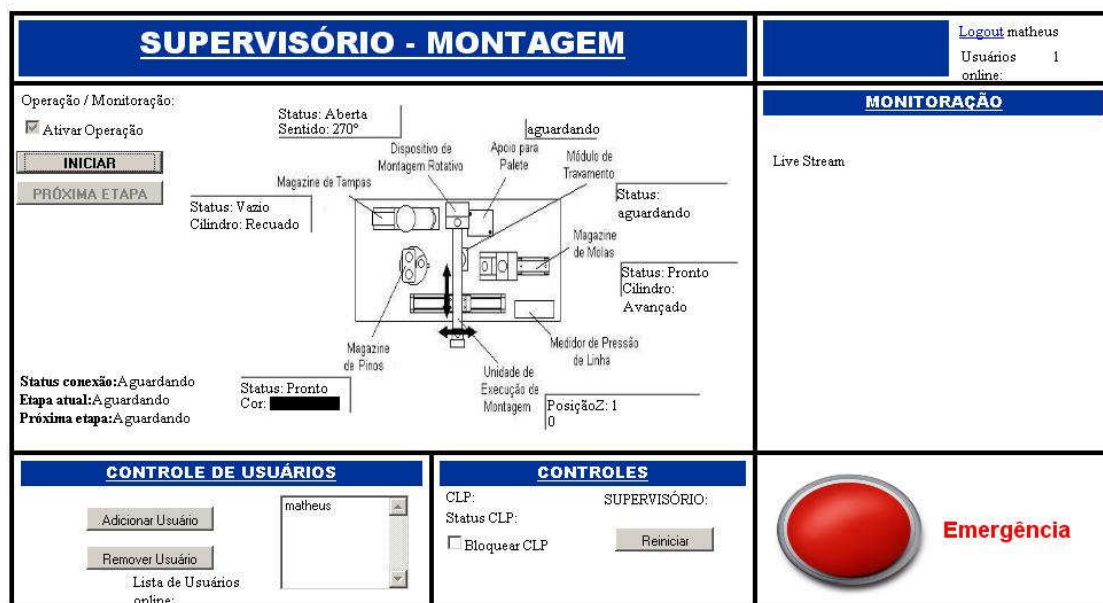


Figura 35. Interface do Supervisor



Figura 36. Ativando operação

Desta forma, quando os motores estiverem prontos para movimentar, aperta-se sequencialmente o botão “INICIAR” para realizar a montagem. A seguir, na Tabela 7, a sequência de montagem:

Tabela 7. Sequência de operação

Etapa	Descrição
1	Move paleta
2	Retira peça
3	Prende peça
4	Escolhe pino
5	Posiciona pino
7	Posiciona mola
8	Posiciona tampa
9	Retorna ao paleta
10	Devolve o paleta
11	Finaliza

O supervisor possui duas atividades realizadas somente por este grupo de acesso: O controle de usuários e o Controle de uso do CLP (Figura 37).

CONTROLE DE USUÁRIOS	CONTROLES
Adicionar Usuário Remover Usuário Lista de Usuários online: matheus	CLP: SUPERVISÓRIO: Status CLP: ☐ Bloquear CLP Reiniciar

Figura 37. Controles do Supervisor

O Controle de usuários permite que o supervisor adicione e remova usuários, acessados diretamente pelos botões na Figura 37, além de visualizar quantos e quais usuários estão on-line. Já o Controle do CLP bloqueia ou libera o uso do CLP, por meio da função “EmUsowrite” disponibilizada pelo *WebService*. Um fato importante sobre o controle do CLP é que se um usuário já estiver utilizando o CLP não é possível que outro usuário inicie a montagem.

O operador acessa a interface apresentada na Figura 38, que disponibiliza apenas as funções de operação e monitoração, sem os controles citados anteriormente.

OPERAÇÃO - MONTAGEM		Logout matheus Usuários online:
Operação / Monitoração: ☐ Ativar Operação INICIAR PRÓXIMA ETAPA	Status: aguardando Sentido: aguardando Magazine de Tampas Status: aguardando Magazine de Molas Status: aguardando Magazine de Pinos Status: aguardando Unidade de Execução de Montagem Status: aguardando Medidor de Pressão de Linha posiçãoZ: aguardando posiçãoXY: aguardando	MONITORAÇÃO Live Stream
		Emergência

Figura 38. Interface do Operador

Por fim, o monitor pode acessar a interface da Figura 39, que apresenta apenas as funcionalidades de monitoração, comuns a todas as interfaces. Estas funcionalidades incluem o acompanhamento dos sensores, evidenciados ao redor do desenho esquemático da estação de montagem, e a câmera de acompanhamento em tempo real, disponibilizada no espaço “monitoração”, à direita.

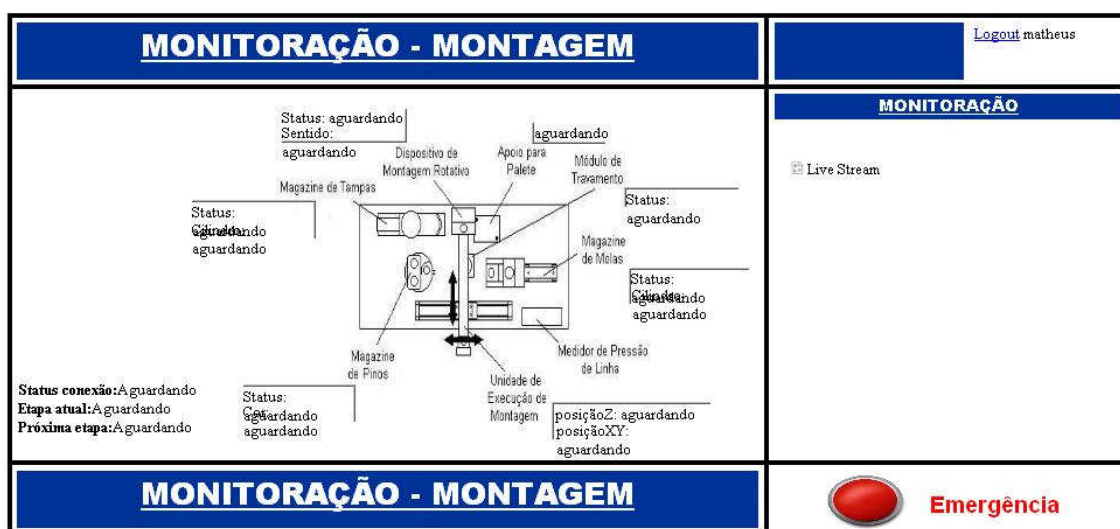


Figura 39. Interface do Monitor

5. Comentários finais e conclusões

Observando a tendência atual de descentralização dos processos produtivos e a necessidade de eliminar as barreiras geográficas para o controle de processos distribuídos por diversas unidades. Neste sentido, foi proposta a utilização de *WebService* para disponibilizar um processo de manufatura como um serviço que pode ser acessado e operado remotamente.

Este projeto utilizou a estação de montagem da célula de manufatura didática (Mini CIM), presente no Laboratório de Sistemas de Automação, para desenvolver uma interface para teleoperação deste processo. Para tanto, o projeto foi dividido em camadas, que se comunicam para que o objetivo final seja alcançado.

A primeira camada se refere ao CLP e seu programa, que atua diretamente nas funções da célula de manufatura. Esta camada é responsável por organizar o acesso aos atuadores e permitir sua utilização pelas outras camadas.

A segunda camada é composta por um *WebService*, que utiliza um servidor OPC e as placas/módulos da rede PROFIBUS, para se comunicar com a primeira camada. Por meio de funções simples, que podem ser acessadas via Internet, esta camada possibilita que as memórias do CLP sejam acessadas, para obter informações sobre os sensores e atuadores ou realizar uma atividade pertencente ao processo.

Dada a necessidade de disponibilizar as funcionalidades da estação de montagem do Mini CIM de forma organizada e controlada, foi criada uma terceira camada que consiste de uma aplicação web. Esta camada é responsável por disponibilizar uma série de páginas de internet, que obtém as informações sobre um usuário e permite o acesso as funções do processo de manufatura que este usuário pode utilizar. O acesso diferenciado é uma característica importante deste projeto, já que ao analisar um processo pode se observar que as atividades dos diferentes usuários não são todas iguais e que nem todos devem ter acesso a todas as particularidades do controle do processo.

É importante salientar que as camadas são independentes entre si, uma vez que outro programa ou aplicação poderia ser desenvolvido para utilizar a primeira e segunda camada deste projeto. Um exemplo prático disto é que o *WebService* poderia ser consumido por qualquer outra aplicação que quisesse utilizar os métodos disponibilizados por este. A interface com o usuário deste projeto poderia, por exemplo, ser substituída por uma aplicação do Windows ou uma para o celular.

A aplicação Web pode ser vista como a camada que reúne todos os requisitos deste projeto e contém as informações essenciais, como a sequência de montagem da peça e as informações dos usuários. Outro papel importante da interface disponibilizada pela aplicação Web é o fornecimento das informações da célula de manufatura e o acompanhamento do processo por meio da câmera de monitoração, que permite que o usuário tenha a exata percepção do andamento da montagem da peça.

Por fim, este projeto possibilitou a utilização de um *WebService* para controle e teleoperação de uma manufatura via internet, por meio de uma interface que pode ser visualizada por um usuário utilizando um computador com baixos recursos de processamento e uma rede de comunicação de 100Mbps. Apesar de ser uma rede de baixa velocidade, deve-se salientar que esta era dedicada à aplicação, e não de uso comercial. Além disso, este projeto poderia ser utilizado para o controle das outras estações do Mini CIM, ou qualquer outra manufatura, se forem realizadas as devidas adaptações.

Referências Bibliográficas

- ÁLVARES, A. J., SOUSA, A. I., Telemanufatura: Teleoperação via web de uma máquina de oxi-corte CNC, Disponível em: <[http://moscoso.org/pub/info/books /graco.unb.br/jatlite/papers_graco/cidim2001_weboxicorte.pdf2001](http://moscoso.org/pub/info/books/graco.unb.br/jatlite/papers_graco/cidim2001_weboxicorte.pdf2001)>. Acesso em: 17 de maio de 2009.
- BELL, D., CESARE, S., IACOVELLI, N., LYCETT, M., MERICO, A., *A framework for deriving semantic web services*. In: **Springer Science and Business Media online**, LLC 2006.
- CERAMI, E., ***Web Services Essentials***. Publisher: O`Reilly, First Edition, 2002.
- JUNQUEIRA, F., **Modelagem de sistemas flexíveis de movimentação de Materiais através de Redes de Petri**, Dissertação de Mestrado apresentada à Escola Politécnica, São Paulo, 2001.
- JUNQUEIRA, F., **Modelagem e Simulação distribuída de sistemas produtivos**, Tese de Doutorado apresentada à Escola Politécnica, São Paulo, 2006.
- JUNQUEIRA, F., MIYAGI, P. E., Modelagem e Simulação Distribuída de Sistema Produtivo Baseados em Rede de Petri, **Revista Controle & Automação**, Vol. 20, No 01, Janeiro, Fevereiro e Março 2009.
- KANEKO, A. L., **Desenvolvimento de uma interface gráfica para supervisão de um sistema produtivo teleoperado**, Monografia apresentada à Escola Politécnica, São Paulo, 2008.
- LING, Z., CHEN, W., YU, J., Research and Implementation of OPC Server Based on Data Access Specification, **Apresentado no 5º Congresso Mundial de Controle e Automação Inteligente**, Hangzhou, P.R. China, Junho de 2004. Disponível em <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1340887>
- MALEK, L. A., WOLF, C., GUYOT, P. D., *Telemanufacturing: A Flexible Manufacturing Solution*, In **International Journal of Production Economics**, 1998, ISSN: 0925-5273.
- MIYAGI, P. E., **Controle Programável**. São Paulo: Editora Edgard Blücher, 1996.

PINTO, P. H., Funcionamento de um Controlador Lógico Programável.

Controle de Contaminação, No 110, Junho 2008.

RIBEIRO CURY, J. E., Teoria de Controle Supervisório de Sistemas a Eventos Discretos, **V Simpósio Brasileiro de Automação Inteligente**, Canela RS, Novembro de 2001. Disponível em <<http://www.das.ufsc.br/~cury/cursos/apostila.pdf>>.

SILVA FILHO, A. M., A Era da Informação. **Revista Espaço Acadêmico**, Ano 1, No 02, Julho 2001 ISSN 1519.6186.

Apêndice A – Web Service

OPCminiCIM.asmx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Xml;
using opcconn;
using System.Threading;

namespace OPCcom
{
    /// <summary>
    /// Summary description for Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
    // [System.Web.Script.Services.ScriptService]
    public class OPCconn : System.Web.Services.WebService
    {

        [WebMethod]
        public string EmUsowrite(bool uso)
        {
            XmlDocument xmlDoc;
            opcClient opc;
            xmlDoc = new XmlDocument();
            xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
            XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
            opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
            if (opc.ErrorMessage.Length > 0)
            {
                opc.Dispose();
                return "OPC não disponível";
            }
            else
            {
                foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
                {
                    opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
                    if (opc.ErrorMessage.Length == 0)
                    {
                        foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                        {
                            opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
                                xmlItem.Attributes.GetNamedItem("memory").Value);
                            if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                            {
                                return opc.GetGroupByPosition(0).ErrorMessage;
                            }
                        }
                    }
                }
                else
                {
                    return opc.ErrorMessage;
                }
            }
            opcClientGroup group = opc.GetGroupByPosition(0);
            if (uso)
            {
                group.GetItemById("44").Write("0");//acertar numero
                group.GetItemById("44").Write("1");
            }
        }
    }
}

```

```

        else
        {
            group.GetItemById("44").Write("0");
            group.GetItemById("44").Write("1");
        }
        return "EmUso";
    }
}

[WebMethod]
public string Zera()
{
    XmlDocument xmlDoc;
    opcClient opc;
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        opc.Dispose();
        return "OPC não disponível";
    }
    else
    {
        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {
            opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
            if (opc.ErrorMessage.Length == 0)
            {
                foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                {
                    opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
                    xmlItem.Attributes.GetNamedItem("memory").Value);
                    if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                    {
                        return opc.GetGroupByPosition(0).ErrorMessage;
                    }
                }
            }
            else
            {
                return opc.ErrorMessage;
            }
        }
        opcClientGroup group = opc.GetGroupByPosition(0);
        group.GetItemById("24").Write("0");
        group.GetItemById("25").Write("1");
        while (group.GetItemById("10").Read() == "0") ;
        group.GetItemById("23").Write("0");
        group.GetItemById("17").Write("1");
        group.GetItemById("19").Write("0");
        group.GetItemById("33").Write("1");
        group.GetItemById("20").Write("0");
        group.GetItemById("34").Write("1");
        group.GetItemById("26").Write("0");
        group.GetItemById("18").Write("1");
        return "Zera";
    }
}

[WebMethod]
public string EmUsoread()
{
    XmlDocument xmlDoc;
    opcClient opc;
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        opc.Dispose();
        return "OPC não disponível";
    }
    else

```

```

    {
        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {
            opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
            if (opc.ErrorMessage.Length == 0)
            {
                foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                {
                    opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
xmlItem.Attributes.GetNamedItem("memory").Value);
                    if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                    {
                        return opc.GetGroupByPosition(0).ErrorMessage;
                    }
                }
            }
            else
            {
                return opc.ErrorMessage;
            }
        }
        opcClientGroup group = opc.GetGroupByPosition(0);
        return group.GetItemById("44").Read(); ;
    }
}

[WebMethod]
public string MoveZ(string pos)
{
    XmlDocument xmlDoc;
    opcClient opc;
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        opc.Dispose();
        return "OPC não disponível";
    }
    else
    {
        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {
            opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
            if (opc.ErrorMessage.Length == 0)
            {
                foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                {
                    opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
xmlItem.Attributes.GetNamedItem("memory").Value);
                    if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                    {
                        return opc.GetGroupByPosition(0).ErrorMessage;
                    }
                }
            }
            else
            {
                return opc.ErrorMessage;
            }
        }
        opcClientGroup group = opc.GetGroupByPosition(0);
        group.GetItemById("35").Write(pos);
        System.Threading.Thread.Sleep(1000);
        group.GetItemById("28").Write("1");
        while (group.GetItemById("36").Read() != pos) ;
        group.GetItemById("28").Write("0");
        return "Move Z";
    }
}

[WebMethod]
public string MoveXY(string pos)
{
    XmlDocument xmlDoc;

```

```

opcClient opc;
xmlDoc = new XmlDocument();
xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
if (opc.ErrorMessage.Length > 0)
{
    opc.Dispose();
    return "OPC não disponível";
}
else
{
    foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
    {
        opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
        if (opc.ErrorMessage.Length == 0)
        {
            foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
            {
                opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
                xmlItem.Attributes.GetNamedItem("memory").Value);
                if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                {
                    return opc.GetGroupByPosition(0).ErrorMessage;
                }
            }
        }
        else
        {
            return opc.ErrorMessage;
        }
    }
    opcClientGroup group = opc.GetGroupByPosition(0);
    group.GetItemById("39").Write("1");
    group.GetItemById("39").Write("0");
    group.GetItemById("38").Write(pos);
    group.GetItemById("43").Write("1");
    group.GetItemById("41").Write("1");
    System.Threading.Thread.Sleep(1000);
    group.GetItemById("27").Write("1");
    while (group.GetItemById("31").Read() == "0");
    group.GetItemById("27").Write("0");
    return "Move XY";
}
}
[WebMethod]
public string Init()
{
    XmlDocument xmlDoc;
    opcClient opc;
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        opc.Dispose();
        return "OPC não disponível";
    }
    else
    {
        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {
            opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
            if (opc.ErrorMessage.Length == 0)
            {
                foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                {
                    opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
                    xmlItem.Attributes.GetNamedItem("memory").Value);
                    if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                    {
                        return opc.GetGroupByPosition(0).ErrorMessage;
                    }
                }
            }
        }
    }
}

```

```

    }
    }
    else
    {
        return opc.ErrorMessage;
    }
}
opcClientGroup group = opc.GetGroupByPosition(0);
group.GetItemById("22").Write("1");
while (group.GetItemById("36").Read() != "1" );
group.GetItemById("22").Write("0");
group.GetItemById("42").Write("1");

group.GetItemById("40").Write("1");
System.Threading.Thread.Sleep(1000);

group.GetItemById("27").Write("1");
while (group.GetItemById("31").Read() == "0" );
group.GetItemById("40").Write("0");
group.GetItemById("42").Write("0");
group.GetItemById("27").Write("0");
group.GetItemById("39").Write("1");

group.GetItemById("39").Write("0");

return "Init";
}
}
[WebMethod]
public string SentidoGarra(bool p270)
{
    XmlDocument xmlDoc;
    opcClient opc;
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        opc.Dispose();
        return "OPC não disponível";
    }
    else
    {
        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {
            opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
            if (opc.ErrorMessage.Length == 0)
            {
                foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                {
                    opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
xmlItem.Attributes.GetNamedItem("memory").Value);
                    if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                    {
                        return opc.GetGroupByPosition(0).ErrorMessage;
                    }
                }
            }
            else
            {
                return opc.ErrorMessage;
            }
        }
        opcClientGroup group = opc.GetGroupByPosition(0);
        if (p270)
        {
            group.GetItemById("24").Write("0");
            group.GetItemById("25").Write("1");
            while (group.GetItemById("10").Read() == "0" );
        }
        else
        {
            group.GetItemById("25").Write("0");

```

```

        group.GetItemById("24").Write("1");
        while (group.GetItemById("10").Read()!="0");
    }
    return "Sentido Garra";
}
}
[WebMethod]
public string StatusGarra(bool abrir)
{
    XmlDocument xmlDoc;
    opcClient opc;
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        opc.Dispose();
        return "OPC não disponível";
    }
    else
    {
        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {
            opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
            if (opc.ErrorMessage.Length == 0)
            {
                foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                {
                    opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
xmlItem.Attributes.GetNamedItem("memory").Value);
                    if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                    {
                        return opc.GetGroupByPosition(0).ErrorMessage;
                    }
                }
            }
            else
            {
                return opc.ErrorMessage;
            }
        }
        opcClientGroup group = opc.GetGroupByPosition(0);
        if (abrir)
        {
            group.GetItemById("23").Write("0");
            group.GetItemById("17").Write("1");
        }
        else
        {
            group.GetItemById("23").Write("1");
            group.GetItemById("17").Write("0");
        }
        return "Status Garra";
    }
}
[WebMethod]
public string CorPino(bool prata)
{
    XmlDocument xmlDoc;
    opcClient opc;
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        opc.Dispose();
        return "OPC não disponível";
    }
    else
    {
        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {

```

```

        opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
        if (opc.ErrorMessage.Length == 0)
        {
            foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
            {
                opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
                xmlItem.Attributes.GetNamedItem("memory").Value);
                if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                {
                    return opc.GetGroupByPosition(0).ErrorMessage;
                }
            }
        }
        else
        {
            return opc.ErrorMessage;
        }
    }
    opcClientGroup group = opc.GetGroupByPosition(0);
    if (prata)
    {
        group.GetItemById("30").Write("0");
        group.GetItemById("29").Write("1");
    }
    else
    {
        group.GetItemById("29").Write("0");
        group.GetItemById("30").Write("1");
    }
    return "CorPino";
}
}
[WebMethod]
public string PosCilindroMola(bool avancar)
{
    XmlDocument xmlDoc;
    opcClient opc;
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        opc.Dispose();
        return "OPC não disponível";
    }
    else
    {
        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {
            opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
            if (opc.ErrorMessage.Length == 0)
            {
                foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                {
                    opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
                    xmlItem.Attributes.GetNamedItem("memory").Value);
                    if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                    {
                        return opc.GetGroupByPosition(0).ErrorMessage;
                    }
                }
            }
        }
        else
        {
            return opc.ErrorMessage;
        }
    }
    opcClientGroup group = opc.GetGroupByPosition(0);
    if (avancar)
    {
        group.GetItemById("33").Write("0");
        group.GetItemById("19").Write("1");
    }
}

```

```

        else
        {
            group.GetItemById("19").Write("0");
            group.GetItemById("33").Write("1");
        }
        return "Cilindro Mola";
    }
}

[WebMethod]
public string PosCilindroTampa(bool avancar)
{
    XmlDocument xmlDoc;
    opcClient opc;
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        opc.Dispose();
        return "OPC não disponível";
    }
    else
    {
        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {
            opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
            if (opc.ErrorMessage.Length == 0)
            {
                foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                {
                    opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
xmlItem.Attributes.GetNamedItem("memory").Value);
                    if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                    {
                        return opc.GetGroupByPosition(0).ErrorMessage;
                    }
                }
            }
            else
            {
                return opc.ErrorMessage;
            }
        }
        opcClientGroup group = opc.GetGroupByPosition(0);
        if (avancar)
        {
            group.GetItemById("34").Write("0");
            group.GetItemById("20").Write("1");
        }
        else
        {
            group.GetItemById("20").Write("0");
            group.GetItemById("34").Write("1");
        }
        return "Cilindro Tampa";
    }
}

[WebMethod]
public string StatusTrava(bool travar)
{
    XmlDocument xmlDoc;
    opcClient opc;
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        opc.Dispose();
        return "OPC não disponível";
    }
    else
    {

```

```

        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {
            opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
            if (opc.ErrorMessage.Length == 0)
            {
                foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                {
                    opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
xmlItem.Attributes.GetNamedItem("memory").Value);
                    if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                    {
                        return opc.GetGroupByPosition(0).ErrorMessage;
                    }
                }
            }
            else
            {
                return opc.ErrorMessage;
            }
        }
        opcClientGroup group = opc.GetGroupByPosition(0);
        if (travar)
        {
            group.GetItemById("18").Write("0");
            group.GetItemById("26").Write("1");
        }
        else
        {
            group.GetItemById("26").Write("0");
            group.GetItemById("18").Write("1");
        }
        return "Status Trava";
    }
}

[WebMethod]
public string Emergencia(bool parar)
{
    XmlDocument xmlDoc;
    opcClient opc;
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        {
            opc.Dispose();
            return "OPC não disponível";
        }
    }
    else
    {
        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {
            {
                opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
                if (opc.ErrorMessage.Length == 0)
                {
                    foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                    {
                        opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
xmlItem.Attributes.GetNamedItem("memory").Value);
                        if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                        {
                            return opc.GetGroupByPosition(0).ErrorMessage;
                        }
                    }
                }
            }
            else
            {
                return opc.ErrorMessage;
            }
        }
    }
    opcClientGroup group = opc.GetGroupByPosition(0);
    if (parar)
    {
        group.GetItemById("45").Write("1");
    }
}

```

```

    }
    else
    {
        group.GetItemById("45").Write("0");
    }
    return "emergencia";
}
}

[WebMethod]
public string[] Read()
{
    XmlDocument xmlDoc;
    opcClient opc;
    string[] leitura;
    int i;
    string aux;
    leitura = new string[50];
    xmlDoc = new XmlDocument();
    xmlDoc.Load(Server.MapPath(@"XML\opcconfig.xml"));
    XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];
    opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);
    if (opc.ErrorMessage.Length > 0)
    {
        opc.Dispose();
        leitura[0] = "OPC não disponível";
        return leitura;
    }
    else
    {
        foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group"))
        {
            opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
            if (opc.ErrorMessage.Length == 0)
            {
                foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item"))
                {
                    opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
                    xmlItem.Attributes.GetNamedItem("memory").Value);
                    if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0)
                    {
                        leitura[0] = opc.GetGroupByPosition(0).ErrorMessage;
                        return leitura;
                    }
                }
            }
            else
            {
                leitura[0] = opc.ErrorMessage;
                return leitura;
            }
        }
        opcClientGroup group = opc.GetGroupByPosition(0);

        for (i = 1; i < 45; i++)
        {
            aux = i.ToString();
            leitura[i] = group.GetItemById(aux).Read();
        }
        return leitura;
    }
}
}
}

```

opcconfig.xml

```

<?xml version="1.0" encoding="iso-8859-1" ?>
<root>
  <server name="OPC.SimaticNET">
    <group name="Montagem">
      <item id="1" memory="S7:[S7 montagem]iAbrirGarra" desc=""/>
      <item id="2" memory="S7:[S7 montagem]iCilindroMolaRecuado" desc=""/>
      <item id="3" memory="S7:[S7 montagem]iCilindroTampaEstendido" desc=""/>
    
```

```

<item id="4" memory="S7:[S7 montagem]iCilindrotampaRecuado" desc=""/>
<item id="5" memory="S7:[S7 montagem]iFechadaGarra" desc=""/>
<item id="6" memory="S7:[S7 montagem]iForneceMola" desc=""/>
<item id="7" memory="S7:[S7 montagem]iFornecePino" desc=""/>
<item id="8" memory="S7:[S7 montagem]iForneceTampa" desc=""/>
<item id="9" memory="S7:[S7 montagem]iGarra0" desc=""/>
<item id="10" memory="S7:[S7 montagem]iGarra270" desc=""/>
<item id="11" memory="S7:[S7 montagem]iGarraZ1" desc=""/>
<item id="12" memory="S7:[S7 montagem]iGarraZ2" desc=""/>
<item id="13" memory="S7:[S7 montagem]iGarraZ3" desc=""/>
<item id="14" memory="S7:[S7 montagem]iGarraZ4" desc=""/>
<item id="15" memory="S7:[S7 montagem]iPinoPrata" desc=""/>
<item id="16" memory="S7:[S7 montagem]iPinoPreto" desc=""/>
<item id="17" memory="S7:[S7 montagem]mAbrirGarra" desc=""/>
<item id="18" memory="S7:[S7 montagem]mDesligarTrava" desc=""/>
<item id="19" memory="S7:[S7 montagem]mEstenderCilindroMola" desc=""/>
<item id="20" memory="S7:[S7 montagem]mEstenderCilindroTampa" desc=""/>
<item id="21" memory="S7:[S7 montagem]mFcP0XY" desc=""/>
<item id="22" memory="S7:[S7 montagem]mfcP1AlturaGarra" desc=""/>
<item id="23" memory="S7:[S7 montagem]mFechargarra" desc=""/>
<item id="24" memory="S7:[S7 montagem]mGarra0" desc=""/>
<item id="25" memory="S7:[S7 montagem]mGarra270" desc=""/>
<item id="26" memory="S7:[S7 montagem]mLigarTrava" desc=""/>
<item id="27" memory="S7:[S7 montagem]mMovimentarXY" desc=""/>
<item id="28" memory="S7:[S7 montagem]mMovimentarZ" desc=""/>
<item id="29" memory="S7:[S7 montagem]mPinoPrata" desc=""/>
<item id="30" memory="S7:[S7 montagem]mPinoPreto" desc=""/>
<item id="31" memory="S7:[S7 montagem]mPontoXYAlcancado" desc=""/>
<item id="32" memory="S7:[S7 montagem]mPontoZAlcancado" desc=""/>
<item id="33" memory="S7:[S7 montagem]mRecuarCilindroMola" desc=""/>
<item id="34" memory="S7:[S7 montagem]mRecuarCilindroTampa" desc=""/>
<item id="35" memory="S7:[S7 montagem]mPontoDesejadoZ" desc=""/>
<item id="36" memory="S7:[S7 montagem]mPontoAtualZ" desc=""/>
<item id="37" memory="S7:[S7 montagem]mPontoAtualXY" desc=""/>
<item id="38" memory="S7:[S7 montagem]mPontoDesejadoXY" desc=""/>
<item id="39" memory="S7:[S7 montagem]mError" desc=""/>
<item id="40" memory="S7:[S7 montagem]mFcP0XYConfiguracao" desc=""/>
<item id="41" memory="S7:[S7 montagem]mFcPXYConfiguracao" desc=""/>
<item id="42" memory="S7:[S7 montagem]InitXY" desc=""/>
<item id="43" memory="S7:[S7 montagem]MovXY" desc=""/>
<item id="44" memory="S7:[S7 montagem]EmUso" desc=""/>
<item id="45" memory="S7:[S7 montagem]mEmergencia" desc=""/>
</group>
</server>
</root>

```

Apêndice B – Aplicação Web

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void LoginStatus1_LoggingOut(object sender, LoginCancelEventArgs e)
    {

    }

    protected void LoginView1_ViewChanged(object sender, EventArgs e)
    {

    }
}
```

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
p.clsRef,span.selflink,span.ui,span.input,span.label,span.keyword{font-weight:700}
        #form1
        {
            height: 160px;
        }
        .style1
        {
            text-align: center;
            margin-left: 69px;
        }
        .style3
        {
            font-size: x-large;
            font-weight: bold;
        }
        .style5
        {
            font-size: large;
        }
    </style>
</head>
<body style="background-color: #FFFFFF">
    <form id="form1" runat="server">
        <div style="text-align: center; width: 801px; margin-left: 125px">

            <div class="style1">
                <asp:LoginStatus ID="LoginStatus1" runat="server" style="text-align: right; top: 18px; left: 811px; position: absolute; height: 19px; width: 34px;"
                    onloggingout="LoginStatus1_LoggingOut" />
                <asp:LoginName ID="LoginName1" runat="server"

                    style="top: 18px; left: 864px; position: absolute; height: 19px; width: 76px" />
                <asp:LoginView ID="LoginView1" runat="server"
                    onviewchanged="LoginView1_ViewChanged">
                    <LoggedInTemplate>
                        Aperte<span class="style5">&nbsp;<asp:HyperLink ID="Member" runat="server" CssClass="style5"
```

[illegible]


```

        Response.Redirect("/MiniCIM/Member/Operacao/Operador.aspx");
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        Response.Redirect("/MiniCIM/Member/Monitoracao/Monitor.aspx");
    }
}

UserAccess.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="UserAccess.aspx.cs"
Inherits="Member_MiniCIM_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<style type="text/css">
    .style1
    {
        text-align: center;
        width: 824px;
    }
    .style2
    {
        font-family: Arial, Helvetica, sans-serif;
    }
    #form1
    {
        height: 541px;
    }
</style>
</head>
<body>
<form id="form1" runat="server">
<div class="style1">

    <b><span class="style2">MiniCIM
    <asp:LoginName ID="LoginName1" runat="server"
        style="top: 19px; left: 742px; position: absolute; height: 19px; width: 90px" />
    </span>
    <br class="style2" />
    <span class="style2">Estação de Montagem<asp:LoginStatus ID="LoginStatus1"
        runat="server"
        style="top: 20px; left: 692px; position: absolute; height: 19px; width: 44px" />
    </span></b>

</div>
<asp:Button ID="Button1" runat="server" onclick="Button1_Click"
    style="top: 178px; left: 29px; position: absolute; height: 26px; width: 127px"
    Text="Supervísório" />
<p>
    <asp:Image ID="Image1" runat="server" ImageUrl="~/pics/MONT.JPG"

        style="top: 82px; left: 287px; position: absolute; height: 409px; text-align: center; width: 369px" />
</p>
<p>
    <asp:Button ID="Button3" runat="server" onclick="Button3_Click"
        style="top: 248px; left: 29px; position: absolute; height: 26px; width: 127px"
        Text="Operação" />
</p>
<asp:Button ID="Button2" runat="server" onclick="Button2_Click"
    style="top: 318px; left: 29px; position: absolute; height: 26px; width: 127px"
    Text="Monitorar" />
</form>
</body>
</html>

Supervisor.aspx.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;

```

```

using System.Web.UI.WebControls;

public partial class Member_Supervisorio_Default : System.Web.UI.Page
{
    private WebReference.OPCconn objWSsend;
    private WebReference.OPCconn objWSread;
    private static int etapa;
    private string response;
    private string[] leitura;
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        objWSread = new WebReference.OPCconn();
        leitura = new string[50];
        leitura = objWSread.Read();
        if (leitura[44] != "0")
        {
            Label3.Visible = true;
        }
        else
        {
            objWSsend = new WebReference.OPCconn();
            System.Threading.Thread.Sleep(1000);
            response = objWSsend.Init();
            System.Threading.Thread.Sleep(2000);
            response = objWSsend.Init();
            Label1.Text = response;
            Button2.Enabled = true;
            Button1.Enabled = false;
            etapa = 1;
        }
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        string aux;
        objWSsend = new WebReference.OPCconn();
        Button2.Enabled = false;
        switch (etapa)
        {
            case 1:

                Label2.Text = "Move pallet";
                Label3.Text = "Retira corpo";
                response = objWSsend.MoveXY("13");
                response = objWSsend.MoveZ("2");
                response = objWSsend.StatusGarra(false);
                response = objWSsend.MoveZ("1");
                response = objWSsend.MoveXY("2");
                Label1.Text = response;
                etapa = 2;

                break;
            case 2:

                Label2.Text = "Retira corpo";
                Label3.Text = "Prende corpo";
                response = objWSsend.MoveZ("4");
                response = objWSsend.StatusGarra(true);
                response = objWSsend.MoveZ("3");
                response = objWSsend.MoveXY("3");
                response = objWSsend.MoveZ("4");
                response = objWSsend.StatusGarra(false);
                response = objWSsend.MoveZ("3");
                response = objWSsend.MoveXY("4");
                Label1.Text = response;
                etapa = 3;
            }
        }
    }
}

```

```

break;
case 3:

    Label2.Text = "Prende corpo";
    Label3.Text = "Escolhe pino";
    response = objWSsend.MoveZ("4");
    System.Threading.Thread.Sleep(500);
    response = objWSsend.StatusGarra(true);
    response = objWSsend.StatusTrava(true);
    Label1.Text = response;
    etapa = 4;

break;
case 4:
    Label2.Text = "Escolhe pino";
    Label3.Text = "Posiciona pino";
    Button2.Enabled = false;
    RadioButtonList1.Visible=true;
    Label4.Visible = true;

break;
case 5:

    Label2.Text = "Posiciona pino";
    Label3.Text = "Posiciona mola";
    response = objWSsend.MoveZ("1");
    response = objWSsend.MoveXY("5");
    response = objWSsend.MoveZ("3");
    response = objWSsend.StatusGarra(false);
    response = objWSsend.MoveZ("1");
    response = objWSsend.MoveXY("6");
    response = objWSsend.MoveZ("4");
    response = objWSsend.StatusGarra(true);
    Label1.Text = response;
    etapa = 6;

break;
case 6:

    Label2.Text = "Posiciona mola";
    Label3.Text = "Posiciona tampa";
    response = objWSsend.PosCilindroMola(true);
    response = objWSsend.MoveZ("1");
    response = objWSsend.MoveXY("7");
    response = objWSsend.MoveZ("4");
    response = objWSsend.StatusGarra(false);
    response = objWSsend.MoveZ("1");
    response = objWSsend.MoveXY("8");
    response = objWSsend.MoveZ("4");
    response = objWSsend.StatusGarra(true);
    response = objWSsend.PosCilindroMola(false);
    Label1.Text = response;
    etapa = 7;

break;
case 7:
    response = objWSsend.PosCilindroTampa(true);
    System.Threading.Thread.Sleep(500);
    response = objWSsend.PosCilindroTampa(false);
    Label2.Text = "Posiciona tampa";
    Label3.Text = "Retorna ao pallet";
    response = objWSsend.MoveZ("1");
    response = objWSsend.SentidoGarra(false);
    response = objWSsend.MoveXY("9");
    response = objWSsend.MoveZ("4");
    response = objWSsend.StatusGarra(false);
    response = objWSsend.MoveZ("1");
    response = objWSsend.SentidoGarra(true);
    response = objWSsend.MoveXY("4");
    response = objWSsend.MoveZ("4");
    response = objWSsend.SentidoGarra(false);
    System.Threading.Thread.Sleep(1000);
    response = objWSsend.StatusGarra(true);

```

```

Label1.Text = response;
etapa = 8;

break;
case 8:

    Label2.Text = "Retorna ao pallet";
    Label3.Text = "Devolve pallet";
    response = objWSsend.StatusTrava(false);
    response = objWSsend.StatusGarra(false);
    response = objWSsend.MoveZ("1");
    response = objWSsend.MoveXY("11");
    response = objWSsend.MoveZ("4");
    response = objWSsend.StatusGarra(true);
    response = objWSsend.MoveZ("1");
    response = objWSsend.MoveXY("12");
    response = objWSsend.MoveZ("4");
    response = objWSsend.StatusGarra(false);
    Label1.Text = response;
    etapa = 9;

    break;
case 9:

    Label2.Text = "Devolve pallet";
    Label3.Text = "Finaliza";
    response = objWSsend.MoveZ("1");
    response = objWSsend.MoveXY("13");
    response = objWSsend.MoveZ("2");
    response = objWSsend.StatusGarra(true);
    response = objWSsend.MoveZ("1");
    Label1.Text = response;
    etapa = 10;

    break;
case 10:

    Label2.Text = "Finaliza";
    Label3.Text = " --- ";
    Label1.Text = response;
    etapa = 2;
    RadioButtonList1.Enabled = true;
    RadioButtonList1.Visible = false;
    response = objWSsend.Zera();
    break;
}
Button2.Enabled = true;
}

protected void Timer1_Tick(object sender, EventArgs e)
{
    leitura = new string[40];
    objWSread = new WebReference.OPCconn();
    leitura = objWSread.Read();
    if (leitura[1] == "0")
    {
        rot.Text = "Status: Aberta";
    }
    if (leitura[2] == "0")
    {
        molas2.Text = "Cilindro: Avançado";
    }
    else{
        molas2.Text = "Cilindro: Recuado";
    }
    if (leitura[3] == "0")
    {
        Tampas2.Text = "Cilindro: Recuado";
    }
    if (leitura[4] == "0")
    {
        Tampas2.Text = "Cilindro: Avançado";
    }
    if (leitura[5] == "0")

```

```

{
    rot.Text = "Status: Fechada";
}
if (leitura[6] == "0")
{
    molas.Text = "Status: Pronto";
}
else{
    molas.Text = "Status: Vazio";
}
if (leitura[7] == "0")
{
    Pinos1.Text = "Status: Pronto";
}
else{
    Pinos1.Text = "Status: Vazio";
    Panel1.BackColor = System.Drawing.Color.Empty;
}
if (leitura[8] == "0")
{
    Tampas1.Text = "Status: Vazio";
}
else{
    Tampas1.Text = "Status: Pronto";
}
if (leitura[9] == "0")
{
    rot2.Text = "Sentido: 270°";
}
if (leitura[10] == "0")
{
    rot2.Text = "Sentido: 0°";
}
if (leitura[11] != "0")
{
    exec.Text = "PosiçãoZ: 1";
}
if (leitura[12] != "0")
{
    exec.Text = "PosiçãoZ: 2";
}
if (leitura[13] != "0")
{
    exec.Text = "PosiçãoZ: 3";
}
if (leitura[14] != "0")
{
    exec.Text = "PosiçãoZ: 4";
}
if (leitura[15] == "0")
{
    pinos2.Text = "Cor:";
    Panel1.BackColor = System.Drawing.Color.Black;
}
if (leitura[16] == "0")
{
    pinos2.Text = "Cor:";
    Panel1.BackColor = System.Drawing.Color.Silver;
}
exec2.Text = leitura[37];
Label6.Text = System.Web.Security.Membership.GetNumberOfUsersOnline().ToString();
ListBox1.Items.Clear();
foreach (System.Web.Security.MembershipUser u in System.Web.Security.Membership.GetAllUsers())

// Membership.GetAllUsers will return MembershipCollctions which holds individual Membership User objects
{
    if (u.IsOnline)
    {
        ListBox1.Items.Add(u.UserName);
    }
}
}

protected void Button3_Click(object sender, EventArgs e)

```

```

{
    objWSsend = new WebReference.OPCconn();
    etapa = 5;
    if (RadioButtonList1.SelectedValue == "1")
    {
        response = objWSsend.CorPino(false);
    }
    else
    {
        response = objWSsend.CorPino(true);
    }
    Button3.Visible = false;
    RadioButtonList1.Enabled = false;
    Label4.Visible = false;
    Button2.Enabled = true;
}

protected void RadioButtonList1_SelectedIndexChanged(object sender, EventArgs e)
{
    Button3.Visible = true;
}

protected void CLP_CheckedChanged(object sender, EventArgs e)
{
    objWSread = new WebReference.OPCconn();
    objWSsend = new WebReference.OPCconn();
    leitura = new string [50];
    leitura = objWSread.Read();
    if (leitura[44] != "0")//acertar numero
    {
        Label9.Text = "CLP em uso. Não foi possível bloquear";
    }
    else
    {
        objWSsend.EmUsowrite(true);
        Label9.Text = "";
    }
}

protected void operacao_CheckedChanged(object sender, EventArgs e)
{
    objWSsend = new WebReference.OPCconn();
    response = objWSsend.Zera();
    Button1.Enabled = true;
    operacao.Enabled = false;
}

protected void Button4_Click(object sender, EventArgs e)
{
    Response.Redirect("/Member/Supervísório/AddUser.aspx");
}

protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    objWSsend = new WebReference.OPCconn();
    response = objWSsend.Emergencia(true);
}

protected void Button6_Click(object sender, EventArgs e)
{
    objWSsend = new WebReference.OPCconn();
    response = objWSsend.Emergencia(false);
    etapa = 1;
    Button1.Enabled = false;
    RadioButtonList1.Enabled = true;
    RadioButtonList1.Visible = false;
    operacao.Enabled = true;
    Button2.Enabled = false;
}

protected void Button5_Click(object sender, EventArgs e)
{
    Response.Redirect("/Member/Supervísório/RemoveUser.aspx");
}

```

```

}
Supervisor.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Supervisor.aspx.cs"
Inherits="Member_Supervisorio_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<style type="text/css">
.style6
{
text-align: left;
font-weight: bold;
font-size: x-large;
font-family: "Comic Sans MS";
height: 28px;
width: 709px;
}
.style7
{
height: 225px;
width: 709px;
}
.style8
{
height: 19px;
width: 709px;
}
.style11
{
height: 28px;
width: 288px;
}
.style12
{
height: 225px;
width: 288px;
}
.style13
{
height: 19px;
width: 288px;
}
.style14
{
height: 28px;
width: 312px;
}
.style15
{
height: 225px;
width: 312px;
}
.style16
{
height: 19px;
width: 312px;
}
#webcam1
{
top: 163px;
left: 843px;
position: absolute;
}
.style18
{
text-align: center;
font-family: "Arial Black";
font-weight: bold;
font-size: large;
color: #FFFFFF;

```

```

        text-decoration: underline;
    }
    .style19
    {
        text-align: center;
        font-family: "Arial Black";
        font-weight: bold;
        font-size: large;
        color: #FFFFFF;
        text-decoration: underline;
    }
    .style20
    {
        text-align: center;
        font-family: "Arial Black";
        font-weight: bold;
        font-size: xx-large;
        color: #FFFFFF;
        text-decoration: underline;
    }
</style>
</head>
<body>
    <form id="form1" runat="server">
    <table style="width: 124%; height: 476px; margin-left: 10px; margin-right: 0px; margin-bottom: 0px;">
        <tr>
            <td class="style11">
                <asp:ScriptManager ID="ScriptManager1" runat="server">
                </asp:ScriptManager>
                <asp:Panel ID="Panel12" runat="server" BackColor="#003399"
                    style="top: 500px; left: 22px; position: absolute; height: 28px; width: 438px">
                    <div class="style18">
                        CONTROLE DE USUÁRIOS</div>
                    </asp:Panel>
                </td>
            <td class="style6">
                SUPERVISÓRIO<br />
            </td>
            <td class="style14">
                <asp:LoginStatus ID="LoginStatus1" runat="server"

                    style="top: 22px; left: 1045px; position: absolute; height: 19px; width: 34px" />
                <asp:LoginName ID="LoginName1" runat="server"

                    style="top: 22px; left: 1097px; position: absolute; height: 19px; width: 76px" />
            </td>
        </tr>
    </table>
    <tr>
        <td class="style12">

            <asp:Panel ID="Panel5" runat="server"
                style="top: 10px; left: 820px; position: absolute; height: 652px; width: 5px; background-color: #000000">
            </asp:Panel>
            <asp:Panel ID="Panel7" runat="server"
                style="top: 10px; left: 10px; position: absolute; height: 652px; width: 5px; background-color: #000000">
            </asp:Panel>
            <asp:Panel ID="Panel8" runat="server"
                style="top: 10px; left: 1210px; position: absolute; height: 652px; width: 5px; background-color: #000000">
            </asp:Panel>
            <asp:Panel ID="Panel4" runat="server"
                style="top: 493px; left: 467px; position: absolute; height: 169px; width: 5px; background-color: #000000">
            </asp:Panel>
            <asp:Panel ID="Panel6" runat="server"
                style="top: 659px; left: 10px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
            </asp:Panel>
            <asp:Panel ID="Panel9" runat="server"

                style="top: 86px; left: 10px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
            </asp:Panel>
            <asp:Panel ID="Panel2" runat="server"
                style="top: 489px; left: 10px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
            </asp:Panel>
            <asp:Panel ID="Panel3" runat="server"

```

```

        style="top: 10px; left: 10px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
</asp:Panel>
<asp:Label ID="Label15" runat="server"
    style="top: 532px; left: 659px; position: absolute; height: 19px; width: 148px"
    Text="SUPERVISÓRIO:"></asp:Label>
<asp:Label ID="Label10" runat="server"
    style="top: 530px; left: 486px; position: absolute; height: 19px; width: 148px"
    Text="CLP:"></asp:Label>

<asp:Label ID="Label9" runat="server"
    style="top: 97px; left: 24px; position: absolute; height: 19px; width: 168px"
    Text="Operação / Monitoração:"></asp:Label>

<asp:Label ID="Label13" runat="server"
    style="top: 126px; left: 166px; position: absolute; height: 19px; width: 110px"
    Text="CLP em uso" ForeColor="Red" Visible="False"></asp:Label>
</td>
<td class="style7">
    <asp:Image ID="Image1" runat="server"
        style="width: 410px; top: 148px; left: 297px; position: absolute; height: 314px; z-index: -1;"
        ImageUrl="~/pics/montagem.JPG" />

    <asp:ImageButton ID="ImageButton1" runat="server" ImageUrl="~/pics/emerg.JPG"
        onclick="ImageButton1_Click"
        style="top: 497px; left: 825px; position: absolute; height: 160px; width: 203px" />
    <asp:Panel ID="Panel13" runat="server" BackColor="#003399"
        style="top: 20px; left: 831px; position: absolute; height: 61px; width: 203px">
        <div class="style20">
        </div>
    </asp:Panel>
    <asp:Panel ID="Panel11" runat="server" BackColor="#003399"
        style="top: 20px; left: 20px; position: absolute; height: 61px; width: 794px">
        <div class="style20">
            SUPERVISÓRIO - MONTAGEM</div>
    </asp:Panel>
    <asp:Panel ID="Panel14" runat="server" BackColor="#003399"
        style="top: 96px; left: 831px; position: absolute; height: 28px; width: 372px">
        <div class="style19">
            MONITORAÇÃO</div>
    </asp:Panel>
    <asp:Panel ID="Panel10" runat="server" BackColor="#003399"
        style="top: 500px; left: 479px; position: absolute; height: 28px; width: 334px">
        <div class="style19">
            CONTROLES</div>
    </asp:Panel>
</td>
<td class="style15">
    <script type="text/javascript">
        var height_array = new Array();
        var width_array = new Array();
        width_array[1] = 352;
        height_array[1] = 288;
    </script>
    
    <script type="text/javascript">
    <!--
        currentCamera1 = 1;
        errorimg1 = 0;
        document.images.webcam1.onload = Dolt1;
        document.images.webcam1.onerror = ErrorImage1;
        function LoadImage1() {
            uniq1 = Math.random();
            document.images.webcam1.src = "http://192.168.0.194:8080/cam_" + currentCamera1 + ".jpg?uniq=" + uniq1;
            document.images.webcam1.onload = Dolt1;
        }
        function PTZMouseDown1(e) {
            var IE = document.all ? true : false;
            var x, y;
            var myx, myy;
            var myifr = document.getElementById("_iframe-ptz");
            tp = getElPos1();

```

```

        myx = tp[0];
        myy = tp[1];
        if (IE) {
            var scrollX = document.documentElement.scrollLeft ? document.documentElement.scrollLeft :
document.body.scrollLeft;
            var scrollY = document.documentElement.scrollTop ? document.documentElement.scrollTop :
document.body.scrollTop;
            x = event.clientX - myx + scrollX;
            y = event.clientY - myy + scrollY;
        } else {
            x = e.pageX - myx;
            y = e.pageY - myy;
        }
        if ((width_array[currentCamera1] != null) && (width_array[currentCamera1] > 0)) x = Math.round((x * 400) /
width_array[currentCamera1]);
        if ((height_array[currentCamera1] != null) && (height_array[currentCamera1] > 0)) y = Math.round((y * 300) /
height_array[currentCamera1]);
        if (x > 400) x = 400;
        if (y > 300) y = 300;
        if (myifr != null) myifr.src = "http://192.168.0.194:8080/ptz?src=" + currentCamera1 + "&moveto_x=" + x +
"&moveto_y=" + y + "";
        return true;
    }
    function getElPos1() {
        el = document.images.webcam1;
        x = el.offsetLeft;
        y = el.offsetTop;
        elp = el.offsetParent;
        while (elp != null) {
            x += elp.offsetLeft;
            y += elp.offsetTop;
            elp = elp.offsetParent;
        }
        return new Array(x, y);
    }
    function ErrorImage1() {
        errorimg1++;
        if (errorimg1 > 3) {
            document.images.webcam1.onload = "";
            document.images.webcam1.onerror = "";
            document.images.webcam1.src = "offline.jpg";
        } else {
            uniq1 = Math.random();
            document.images.webcam1.src = "http://192.168.0.194:8080/cam_" + currentCamera1 + ".jpg?uniq=" + uniq1;
        }
    }
    function Dolt1() {
        errorimg1 = 0;
        window.setTimeout("LoadImage1()", 40);
    }
    //-->
</script>
    &nbsp;  </td>
</tr>
<tr>
    <td class="style13">
        <br />
        <br />
        <asp:UpdatePanel ID="UpdatePanel3" runat="server">
            <ContentTemplate>
                <b>
                    <asp:Timer ID="Timer1" runat="server" Interval="1000" ontick="Timer1_Tick">
                    </asp:Timer>
                    <asp:Button ID="Button1" runat="server" Font-Bold="True"
                        onclick="Button1_Click"
                        style="text-align: center; top: 162px; left: 18px; position: absolute; height: 26px; width: 159px"
                        Text="INICIAR" Enabled="False" />
                    <asp:Button ID="Button2" runat="server" Enabled="False" Font-Bold="True"
                        onclick="Button2_Click"
                        style="top: 195px; left: 18px; position: absolute; height: 26px; width: 159px"
                        Text="PRÓXIMA ETAPA" />
                    Status conexão:</b><asp:Label ID="Label1" runat="server" Text="Aguardando"></asp:Label>
                    <asp:Label ID="Tampas2" runat="server"
                        style="top: 232px; left: 207px; position: absolute; height: 19px; width: 126px"

```

```

        Text="Cilindro: aguardando"></asp:Label>
<asp:Label ID="rot2" runat="server"
    style="top: 128px; left: 304px; position: absolute; height: 19px; width: 126px"
    Text="Sentido: aguardando"></asp:Label>
<asp:Label ID="molas2" runat="server"
    style="top: 302px; left: 679px; position: absolute; height: 19px; width: 127px"
    Text="Cilindro: aguardando"></asp:Label>
<asp:Button ID="Button3" runat="server" onclick="Button3_Click"
    style="top: 291px; left: 26px; position: absolute; height: 26px; width: 56px"
    Text="OK" Visible="False" />
<asp:Label ID="Label4" runat="server"
    style="top: 295px; left: 95px; position: absolute; height: 19px; width: 194px"
    Text="Escolha uma cor para continuar" Visible="False"></asp:Label>
<asp:Button ID="Button6" runat="server" onclick="Button6_Click"
    style="top: 573px; left: 670px; position: absolute; height: 26px; width: 108px"
    Text="Reiniciar" />
<br />
<b> Etapa atual:</b><asp:Label ID="Label2" runat="server" Text="Aguardando"></asp:Label>
<asp:Label ID="pinos2" runat="server"
    style="top: 427px; left: 262px; position: absolute; height: 19px; width: 107px"
    Text="Cor: aguardando"></asp:Label>
<asp:RadioButtonList ID="RadioButtonList1" runat="server"
    style="top: 233px; left: 18px; position: absolute; height: 27px; width: 82px"
    onselectedindexchanged="RadioButtonList1_SelectedIndexChanged"
    Visible="False" CausesValidation="True" AutoPostBack="True">
    <asp:ListItem Value="1">Preto</asp:ListItem>
    <asp:ListItem Value="0">Prata</asp:ListItem>
</asp:RadioButtonList>
<asp:Panel ID="Panel1" runat="server"
    style="top: 430px; left: 294px; position: absolute; height: 15px; width: 76px">
</asp:Panel>
<asp:Label ID="Label5" runat="server"
    style="top: 49px; left: 1047px; position: absolute; height: 19px; width: 97px"
    Text="Usuários online:"></asp:Label>
<asp:Label ID="Label6" runat="server"
    style="top: 49px; left: 1147px; position: absolute; height: 19px; width: 34px"></asp:Label>
<asp:Label ID="Label7" runat="server"
    style="top: 622px; left: 145px; position: absolute; height: 19px; width: 160px"
    Text="Lista de Usuários online:"></asp:Label>
<asp:CheckBox ID="operacao" runat="server"
    style="top: 126px; left: 23px; position: absolute; height: 21px; width: 136px"
    Text="Ativar Operação" oncheckedchanged="operacao_CheckedChanged"
    CausesValidation="True" AutoPostBack="True" />
<asp:CheckBox ID="CLP" runat="server" AutoPostBack="True"
    oncheckedchanged="CLP_CheckedChanged"
    style="top: 578px; left: 483px; position: absolute; height: 21px; width: 116px"
    Text="Bloquear CLP" />
<asp:Label ID="Label11" runat="server"

    style="top: 599px; left: 499px; position: absolute; height: 19px; width: 155px"
    ForeColor="#CC0000"></asp:Label>
<asp:Label ID="Label8" runat="server"
    style="top: 552px; left: 486px; position: absolute; height: 19px; width: 155px"
    Text="Status CLP:"></asp:Label>
<asp:ListBox ID="ListBox1" runat="server"

    style="top: 537px; left: 305px; position: absolute; height: 104px; width: 141px">
</asp:ListBox>
<asp:Button ID="Button5" runat="server"
    style="top: 593px; left: 94px; position: absolute; height: 26px; width: 138px"
    Text="Remover Usuário" PostBackUrl="Member/Supervisão/Removeuser.aspx"
    onclick="Button5_Click" />
<asp:Button ID="Button4" runat="server"
    style="top: 550px; left: 94px; position: absolute; height: 26px; width: 138px"
    Text="Adicionar Usuário" PostBackUrl="Member/Supervisão/Adduser.aspx" />
<br />
<b> Próxima etapa:</b><asp:Label ID="Label3" runat="server" Text="Aguardando"></asp:Label>
<asp:Label ID="apoio" runat="server"
    style="top: 128px; left: 571px; position: absolute; height: 19px; width: 76px; border-left-style: ridge;
border-left-color: #808080; border-bottom-style: ridge; border-bottom-color: #808080"
    Text="aguardando"></asp:Label>
<asp:Label ID="Tampas1" runat="server"

```

```

        style="top: 210px; left: 208px; position: absolute; height: 38px; width: 129px; border-right-style: ridge;
border-right-color: #808080; border-top-style: ridge; border-top-color: #808080"
        Text="Status: aguardando"></asp:Label>
        <asp:Label ID="exec" runat="server"
        style="top: 426px; position: absolute; height: 37px; width: 166px; left: 593px; border-top-style: ridge;
border-top-color: #808080; border-left-style: ridge; border-left-color: #808080;"
        Text="posiçãoZ: aguardando"></asp:Label>
        <asp:Label ID="Pinos1" runat="server"
        style="top: 408px; position: absolute; height: 37px; width: 118px; left: 262px; border-right-style: ridge;
border-right-color: #808080; border-top-style: ridge; border-top-color: #808080; bottom: 128px;"
        Text="Status: aguardando"></asp:Label>
        <asp:Label ID="molas" runat="server"
        style="top: 282px; left: 675px; position: absolute; height: 36px; width: 126px; border-left-style: ridge;
border-left-color: #808080; border-top-style: ridge; border-top-color: #808080; bottom: 229px;"
        Text="Status: aguardando"></asp:Label>
        <asp:Label ID="rot" runat="server"
        style="top: 112px; left: 304px; position: absolute; height: 34px; width: 130px; border-right-style: ridge;
border-right-color: #808080; border-bottom-style: ridge; border-bottom-color: #808080;"
        Text="Status: aguardando"></asp:Label>
        <asp:Label ID="travamento" runat="server"
        style="top: 196px; left: 670px; position: absolute; height: 19px; width: 120px; border-left-style: ridge;
border-left-color: #808080; border-top-style: ridge; border-top-color: #808080"
        Text="Status: aguardando"></asp:Label>
        <asp:Label ID="exec2" runat="server"
        style="top: 447px; left: 597px; position: absolute; height: 19px; width: 153px"
        Text="posiçãoXY: aguardando"></asp:Label>
    </ContentTemplate>
</asp:UpdatePanel>
</td>
<td class="style8">
    <asp:Label ID="Label14" runat="server"
    style="top: 561px; left: 1028px; position: absolute; height: 19px; width: 34px; color: #FF0000; font-size: x-
large; font-weight: 700; font-family: Arial, Helvetica, sans-serif"
    Text="Emergência"></asp:Label>
</td>
<td class="style16">
</td>
</tr>
</table>
</form>
</body>
</html>

```

RemoveUser.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Member_Supervisorio_Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        System.Web.Security.Membership.DeleteUser(TextBox1.Text);
    }
}

```

RemoveUser.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="RemoveUser.aspx.cs"
Inherits="Member_Supervisorio_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        #form1
    {

```

[illegible]

AddUser.aspx.cs

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;
```

```
public partial class Member_Supervisorio_Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```

AddUser.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="AddUser.aspx.cs"
Inherits="Member Supervisorio Default" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

[illegible]

Operator.aspx.cs

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;
```

```

public partial class Member_Operacao_Default : System.Web.UI.Page
{
    private WebReference.OPCconn objWSSend;
    private WebReference.OPCconn objWSread;
    private static int etapa;
    private string response;
    private string[] leitura;
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        objWSread = new WebReference.OPCconn();
        leitura = new string[50];
        leitura = objWSread.Read();
        if (leitura[44] != "0")
        {
            Label3.Visible = true;
        }
        else
        {
            objWSSend = new WebReference.OPCconn();
            System.Threading.Thread.Sleep(1000);
            response = objWSSend.Init();
            System.Threading.Thread.Sleep(2000);
            response = objWSSend.Init();
            Label1.Text = response;
            Button2.Enabled = true;
            Button1.Enabled = false;
            etapa = 1;
        }
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        string aux;
        objWSSend = new WebReference.OPCconn();
        Button2.Enabled = false;
        switch (etapa)
        {
            case 1:

                Label2.Text = "Move pallet";
                Label3.Text = "Retira corpo";
                response = objWSSend.MoveXY("13");
                response = objWSSend.MoveZ("2");
                response = objWSSend.StatusGarra(false);
                response = objWSSend.MoveZ("1");
                response = objWSSend.MoveXY("2");
                Label1.Text = response;
                etapa = 2;

                break;
            case 2:

                Label2.Text = "Retira corpo";
                Label3.Text = "Prende corpo";
                response = objWSSend.MoveZ("4");
                response = objWSSend.StatusGarra(true);
                response = objWSSend.MoveZ("3");
                response = objWSSend.MoveXY("3");
                response = objWSSend.MoveZ("4");
                response = objWSSend.StatusGarra(false);
                response = objWSSend.MoveZ("3");
                response = objWSSend.MoveXY("4");
                Label1.Text = response;
                etapa = 3;

                break;
        }
    }
}

```

case 3:

```
Label2.Text = "Prende corpo";
Label3.Text = "Escolhe pino";
response = objWSsend.MoveZ("4");
System.Threading.Thread.Sleep(500);
response = objWSsend.StatusGarra(true);
response = objWSsend.StatusTrava(true);
Label1.Text = response;
etapa = 4;
```

break;

case 4:

```
Label2.Text = "Escolhe pino";
Label3.Text = "Posiciona pino";
Button2.Enabled = false;
RadioButtonList1.Visible = true;
Label4.Visible = true;
```

break;

case 5:

```
Label2.Text = "Posiciona pino";
Label3.Text = "Posiciona mola";
response = objWSsend.MoveZ("1");
response = objWSsend.MoveXY("5");
response = objWSsend.MoveZ("3");
response = objWSsend.StatusGarra(false);
response = objWSsend.MoveZ("1");
response = objWSsend.MoveXY("6");
response = objWSsend.MoveZ("4");
response = objWSsend.StatusGarra(true);
Label1.Text = response;
etapa = 6;
```

break;

case 6:

```
Label2.Text = "Posiciona mola";
Label3.Text = "Posiciona tampa";
response = objWSsend.PosCilindroMola(true);
response = objWSsend.MoveZ("1");
response = objWSsend.MoveXY("7");
response = objWSsend.MoveZ("4");
response = objWSsend.StatusGarra(false);
response = objWSsend.MoveZ("1");
response = objWSsend.MoveXY("8");
response = objWSsend.MoveZ("4");
response = objWSsend.StatusGarra(true);
response = objWSsend.PosCilindroMola(false);
Label1.Text = response;
etapa = 7;
```

break;

case 7:

```
response = objWSsend.PosCilindroTampa(true);
System.Threading.Thread.Sleep(500);
response = objWSsend.PosCilindroTampa(false);
Label2.Text = "Posiciona tampa";
Label3.Text = "Retorna ao pallet";
response = objWSsend.MoveZ("1");
response = objWSsend.SentidoGarra(false);
response = objWSsend.MoveXY("9");
response = objWSsend.MoveZ("4");
response = objWSsend.StatusGarra(false);
response = objWSsend.MoveZ("1");
response = objWSsend.SentidoGarra(true);
response = objWSsend.MoveXY("10");
response = objWSsend.MoveZ("4");
response = objWSsend.SentidoGarra(false);
System.Threading.Thread.Sleep(1000);
response = objWSsend.StatusGarra(true);
Label1.Text = response;
etapa = 8;
```

```

        break;
    case 8:

        Label2.Text = "Retorna ao pallet";
        Label3.Text = "Devolve pallet";
        response = objWSsend.StatusTrava(false);
        response = objWSsend.StatusGarra(false);
        response = objWSsend.MoveZ("1");
        response = objWSsend.MoveXY("11");
        response = objWSsend.MoveZ("4");
        response = objWSsend.StatusGarra(true);
        response = objWSsend.MoveZ("1");
        response = objWSsend.MoveXY("12");
        response = objWSsend.MoveZ("4");
        response = objWSsend.StatusGarra(false);
        Label1.Text = response;
        etapa = 9;

        break;
    case 9:

        Label2.Text = "Devolve pallet";
        Label3.Text = "Finaliza";
        response = objWSsend.MoveZ("1");
        response = objWSsend.MoveXY("13");
        response = objWSsend.MoveZ("2");
        response = objWSsend.StatusGarra(true);
        response = objWSsend.MoveZ("1");
        Label1.Text = response;
        etapa = 10;

        break;
    case 10:

        Label2.Text = "Finaliza";
        Label3.Text = " --- ";
        Label1.Text = response;
        etapa = 1;
        RadioButtonList1.Enabled = true;
        RadioButtonList1.Visible = false;
        operacao.Enabled = true;
        Button2.Enabled = false;
        break;
    }
    Button2.Enabled = true;
}

protected void Timer1_Tick(object sender, EventArgs e)
{
    leitura = new string[40];
    objWSread = new WebReference.OPCconn();
    leitura = objWSread.Read();
    if (leitura[1] == "0")
    {
        rot.Text = "Status: Aberta";
    }
    if (leitura[2] == "0")
    {
        molas2.Text = "Cilindro: Avançado";
    }
    else
    {
        molas2.Text = "Cilindro: Recuado";
    }
    if (leitura[3] == "0")
    {
        Tampas2.Text = "Cilindro: Recuado";
    }
    if (leitura[4] == "0")
    {
        Tampas2.Text = "Cilindro: Avançado";
    }
    if (leitura[5] == "0")

```

```

    {
        rot.Text = "Status: Fechada";
    }
    if (leitura[6] == "0")
    {
        molas.Text = "Status: Pronto";
    }
    else
    {
        molas.Text = "Status: Vazio";
    }
    if (leitura[7] == "0")
    {
        Pinos1.Text = "Status: Pronto";
    }
    else
    {
        Pinos1.Text = "Status: Vazio";
        Panel1.BackColor = System.Drawing.Color.Empty;
    }
    if (leitura[8] == "0")
    {
        Tampas1.Text = "Status: Vazio";
    }
    else
    {
        Tampas1.Text = "Status: Pronto";
    }
    if (leitura[9] == "0")
    {
        rot2.Text = "Sentido: 270°";
    }
    if (leitura[10] == "0")
    {
        rot2.Text = "Sentido: 0°";
    }
    if (leitura[11] != "0")
    {
        exec.Text = "PosiçãoZ: 1";
    }
    if (leitura[12] != "0")
    {
        exec.Text = "PosiçãoZ: 2";
    }
    if (leitura[13] != "0")
    {
        exec.Text = "PosiçãoZ: 3";
    }
    if (leitura[14] != "0")
    {
        exec.Text = "PosiçãoZ: 4";
    }
    if (leitura[15] == "0")
    {
        pinos2.Text = "Cor:";
        Panel1.BackColor = System.Drawing.Color.Black;
    }
    if (leitura[16] == "0")
    {
        pinos2.Text = "Cor:";
        Panel1.BackColor = System.Drawing.Color.Silver;
    }
    exec2.Text = leitura[37];
    Label6.Text = System.Web.Security.Membership.GetNumberOfUsersOnline().ToString();
}

protected void Button3_Click(object sender, EventArgs e)
{
    objWSsend = new WebReference.OPCconn();
    etapa = 5;
    if (RadioButtonList1.SelectedValue == "1")
    {
        response = objWSsend.CorPino(false);
    }
}

```

```

else
{
    response = objWSsend.CorPino(true);
}
Button3.Visible = false;
RadioButtonList1.Enabled = false;
Label4.Visible = false;
Button2.Enabled = true;
}

protected void RadioButtonList1_SelectedIndexChanged(object sender, EventArgs e)
{
    Button3.Visible = true;
}

protected void operacao_CheckedChanged(object sender, EventArgs e)
{
    objWSsend = new WebReference.OPCconn();
    response = objWSsend.Zera();
    Button1.Enabled = true;
    operacao.Enabled = false;
}

protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    objWSsend = new WebReference.OPCconn();
    response = objWSsend.Emergencia(true);
}
}
Operador.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Operador.aspx.cs"
Inherits="Member_Operacao_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<style type="text/css">
    .style6
    {
        text-align: left;
        font-weight: bold;
        font-size: x-large;
        font-family: "Comic Sans MS";
        height: 28px;
        width: 709px;
    }
    .style7
    {
        height: 225px;
        width: 709px;
    }
    .style8
    {
        height: 19px;
        width: 709px;
    }
    .style11
    {
        height: 28px;
        width: 288px;
    }
    .style12
    {
        height: 225px;
        width: 288px;
    }
    .style13
    {
        height: 19px;
        width: 288px;
    }

```

```

}
.style14
{
    height: 28px;
    width: 312px;
}
.style15
{
    height: 225px;
    width: 312px;
}
.style16
{
    height: 19px;
    width: 312px;
}
#webcam1
{
    top: 163px;
    left: 843px;
    position: absolute;
}
.style19
{
    text-align: center;
    font-family: "Arial Black";
    font-weight: bold;
    font-size: large;
    color: #FFFFFF;
    text-decoration: underline;
}
.style20
{
    text-align: center;
    font-family: "Arial Black";
    font-weight: bold;
    font-size: xx-large;
    color: #FFFFFF;
    text-decoration: underline;
}
</style>
</head>
<body>
<form id="form1" runat="server">
<table style="width: 124%; height: 476px; margin-left: 10px; margin-right: 0px; margin-bottom: 0px;">
<tr>
<td class="style11">
<asp:ScriptManager ID="ScriptManager1" runat="server">
</asp:ScriptManager>
</td>
<td class="style6">
SUPERVISÓRIO<br />
</td>
<td class="style14">
<asp:LoginStatus ID="LoginStatus1" runat="server"

        style="top: 22px; left: 1045px; position: absolute; height: 19px; width: 34px" />
<asp:LoginName ID="LoginName1" runat="server"

        style="top: 22px; left: 1097px; position: absolute; height: 19px; width: 76px" />
</td>
</tr>
<tr>
<td class="style12">

<asp:Panel ID="Panel5" runat="server"
        style="top: 10px; left: 820px; position: absolute; height: 652px; width: 5px; background-color: #000000">
</asp:Panel>
<asp:Panel ID="Panel7" runat="server"
        style="top: 10px; left: 10px; position: absolute; height: 652px; width: 5px; background-color: #000000">
</asp:Panel>
<asp:Panel ID="Panel8" runat="server"
        style="top: 10px; left: 1210px; position: absolute; height: 652px; width: 5px; background-color: #000000">

```

```

</asp:Panel>
<asp:Panel ID="Panel6" runat="server"
    style="top: 659px; left: 10px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
</asp:Panel>
<asp:Panel ID="Panel9" runat="server"

    style="top: 86px; left: 10px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
</asp:Panel>
<asp:Panel ID="Panel2" runat="server"
    style="top: 489px; left: 10px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
</asp:Panel>
<asp:Panel ID="Panel3" runat="server"
    style="top: 10px; left: 10px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
</asp:Panel>

<asp:Label ID="Label9" runat="server"
    style="top: 97px; left: 24px; position: absolute; height: 19px; width: 168px"
    Text="Operação / Monitoração:"></asp:Label>

<asp:Label ID="Label13" runat="server"
    style="top: 126px; left: 166px; position: absolute; height: 19px; width: 110px"
    Text="CLP em uso" ForeColor="Red" Visible="False"></asp:Label>

</td>
<td class="style7">
    <asp:Image ID="Image1" runat="server"
        style="width: 410px; top: 148px; left: 297px; position: absolute; height: 314px; z-index: -1;"
        ImageUrl="~/pics/montagem.JPG" />

    <asp:ImageButton ID="ImageButton1" runat="server" ImageUrl="~/pics/emerg.JPG"
        onclick="ImageButton1_Click"
        style="top: 497px; left: 825px; position: absolute; height: 160px; width: 203px" />
    <asp:Panel ID="Panel13" runat="server" BackColor="#003399"
        style="top: 20px; left: 831px; position: absolute; height: 61px; width: 203px">
        <div class="style20">
        </div>
    </asp:Panel>
    <asp:Panel ID="Panel11" runat="server" BackColor="#003399"
        style="top: 20px; left: 20px; position: absolute; height: 61px; width: 794px">
        <div class="style20">
            OPERAÇÃO - MONTAGEM</div>
    </asp:Panel>
    <asp:Panel ID="Panel14" runat="server" BackColor="#003399"
        style="top: 96px; left: 831px; position: absolute; height: 28px; width: 372px">
        <div class="style19">
            MONITORAÇÃO</div>
    </asp:Panel>
    <asp:Panel ID="Panel10" runat="server" BackColor="#003399"

        style="top: 500px; left: 21px; position: absolute; height: 153px; width: 792px">
        <div class="style19">
        </div>
    </asp:Panel>

</td>
<td class="style15">
    <script type="text/javascript">
        var height_array = new Array();
        var width_array = new Array();
        width_array[1] = 352;
        height_array[1] = 288;

</script>

<script type="text/javascript">
<!--
    currentCamera1 = 1;
    errorimg1 = 0;
    document.images.webcam1.onload = Dolt1;
    document.images.webcam1.onerror = ErrorImage1;
    function LoadImage1() {
        uniq1 = Math.random();
        document.images.webcam1.src = "http://192.168.0.194:8080/cam_" + currentCamera1 + ".jpg?uniq=" + uniq1;
        document.images.webcam1.onload = Dolt1;

```

```

    }
    function PTZMouseDown1(e) {
        var IE = document.all ? true : false;
        var x, y;
        var myx, myy;
        var myifr = document.getElementById("_iframe-ptz");
        tp = getElPos1();
        myx = tp[0];
        myy = tp[1];
        if (IE) {
            var scrollX = document.documentElement.scrollLeft ? document.documentElement.scrollLeft :
document.body.scrollLeft;
            var scrollY = document.documentElement.scrollTop ? document.documentElement.scrollTop :
document.body.scrollTop;
            x = event.clientX - myx + scrollX;
            y = event.clientY - myy + scrollY;
        } else {
            x = e.pageX - myx;
            y = e.pageY - myy;
        }
        if ((width_array[currentCamera1] != null) && (width_array[currentCamera1] > 0)) x = Math.round((x * 400) /
width_array[currentCamera1]);
        if ((height_array[currentCamera1] != null) && (height_array[currentCamera1] > 0)) y = Math.round((y * 300) /
height_array[currentCamera1]);
        if (x > 400) x = 400;
        if (y > 300) y = 300;
        if (myifr != null) myifr.src = "http://192.168.0.194:8080/ptz?src=" + currentCamera1 + "&moveto_x=" + x +
"&moveto_y=" + y + "";
        return true;
    }
    function getElPos1() {
        el = document.images.webcam1;
        x = el.offsetLeft;
        y = el.offsetTop;
        elp = el.offsetParent;
        while (elp != null) {
            x += elp.offsetLeft;
            y += elp.offsetTop;
            elp = elp.offsetParent;
        }
        return new Array(x, y);
    }
    function ErrorImage1() {
        errorimg1++;
        if (errorimg1 > 3) {
            document.images.webcam1.onload = "";
            document.images.webcam1.onerror = "";
            document.images.webcam1.src = "offline.jpg";
        } else {
            uniq1 = Math.random();
            document.images.webcam1.src = "http://192.168.0.194:8080/cam_" + currentCamera1 + ".jpg?uniq=" + uniq1;
        }
    }
    function Dolt1() {
        errorimg1 = 0;
        window.setTimeout("LoadImage1();", 40);
    }
}
//-->
</script>
        &nbsp;  </td>
    </tr>
    <tr>
        <td class="style13">
            <br />
            <br />
            <asp:UpdatePanel ID="UpdatePanel3" runat="server">
                <ContentTemplate>
                    <b>
                        <asp:Timer ID="Timer1" runat="server" Interval="1000" ontick="Timer1_Tick">
                        </asp:Timer>
                        <asp:Button ID="Button1" runat="server" Font-Bold="True"
                            onclick="Button1_Click"
                            style="text-align: center; top: 162px; left: 18px; position: absolute; height: 26px; width: 159px"
                            Text="INICIAR" Enabled="False" />
                    </b>
                </ContentTemplate>
            </asp:UpdatePanel>
        </td>
    </tr>

```

```

<asp:Button ID="Button2" runat="server" Enabled="False" Font-Bold="True"
onclick="Button2_Click"
style="top: 195px; left: 18px; position: absolute; height: 26px; width: 159px"
Text="PRÓXIMA ETAPA" />
Status conexão:</b><asp:Label ID="Label1" runat="server" Text="Aguardando"></asp:Label>
<asp:Label ID="Tampas2" runat="server"
style="top: 232px; left: 207px; position: absolute; height: 19px; width: 126px"
Text="Cilindro: aguardando"></asp:Label>
<asp:Label ID="rot2" runat="server"
style="top: 128px; left: 304px; position: absolute; height: 19px; width: 126px"
Text="Sentido: aguardando"></asp:Label>
<asp:Label ID="molas2" runat="server"
style="top: 302px; left: 679px; position: absolute; height: 19px; width: 127px"
Text="Cilindro: aguardando"></asp:Label>
<asp:Button ID="Button3" runat="server" onclick="Button3_Click"
style="top: 291px; left: 26px; position: absolute; height: 26px; width: 56px"
Text="OK" Visible="False" />
<asp:Label ID="Label4" runat="server"
style="top: 295px; left: 95px; position: absolute; height: 19px; width: 194px"
Text="Escolha uma cor para continuar" Visible="False"></asp:Label>
<br />
<b> Etapa atual:</b><asp:Label ID="Label2" runat="server" Text="Aguardando"></asp:Label>
<asp:Label ID="pinos2" runat="server"
style="top: 427px; left: 262px; position: absolute; height: 19px; width: 107px"
Text="Cor: aguardando"></asp:Label>
<asp:RadioButtonList ID="RadioButtonList1" runat="server"
style="top: 233px; left: 18px; position: absolute; height: 27px; width: 82px"
onselectedindexchanged="RadioButtonList1_SelectedIndexChanged"
Visible="False" CausesValidation="True" AutoPostBack="True">
<asp:ListItem Value="1">Preto</asp:ListItem>
<asp:ListItem Value="0">Prata</asp:ListItem>
</asp:RadioButtonList>
<asp:Panel ID="Panel1" runat="server"
style="top: 430px; left: 294px; position: absolute; height: 15px; width: 76px">
</asp:Panel>
<asp:Label ID="Label5" runat="server"
style="top: 49px; left: 1047px; position: absolute; height: 19px; width: 97px"
Text="Usuários online:"></asp:Label>
<asp:Label ID="Label6" runat="server"
style="top: 49px; left: 1147px; position: absolute; height: 19px; width: 34px"></asp:Label>
<asp:CheckBox ID="operacao" runat="server"
style="top: 126px; left: 23px; position: absolute; height: 21px; width: 136px"
Text="Ativar Operação" onclick="operacao_CheckedChanged"
CausesValidation="True" AutoPostBack="True" />
<br />
<b> Próxima etapa:</b><asp:Label ID="Label3" runat="server" Text="Aguardando"></asp:Label>
<asp:Label ID="apoio" runat="server"
style="top: 128px; left: 571px; position: absolute; height: 19px; width: 76px; border-left-style: ridge;
border-left-color: #808080; border-bottom-style: ridge; border-bottom-color: #808080"
Text="aguardando"></asp:Label>
<asp:Label ID="Tampas1" runat="server"
style="top: 210px; left: 208px; position: absolute; height: 38px; width: 129px; border-right-style: ridge;
border-right-color: #808080; border-top-style: ridge; border-top-color: #808080"
Text="Status: aguardando"></asp:Label>
<asp:Label ID="exec" runat="server"
style="top: 426px; position: absolute; height: 37px; width: 166px; left: 593px; border-top-style: ridge;
border-top-color: #808080; border-left-style: ridge; border-left-color: #808080;"
Text="posiçãoZ: aguardando"></asp:Label>
<asp:Label ID="Pinos1" runat="server"
style="top: 408px; position: absolute; height: 37px; width: 118px; left: 262px; border-right-style: ridge;
border-right-color: #808080; border-top-style: ridge; border-top-color: #808080; bottom: 128px;"
Text="Status: aguardando"></asp:Label>
<asp:Label ID="molas" runat="server"
style="top: 282px; left: 675px; position: absolute; height: 36px; width: 126px; border-left-style: ridge;
border-left-color: #808080; border-top-style: ridge; border-top-color: #808080; bottom: 229px;"
Text="Status: aguardando"></asp:Label>
<asp:Label ID="rot" runat="server"
style="top: 112px; left: 304px; position: absolute; height: 34px; width: 130px; border-right-style: ridge;
border-right-color: #808080; border-bottom-style: ridge; border-bottom-color: #808080;"
Text="Status: aguardando"></asp:Label>
<asp:Label ID="travamento" runat="server"
style="top: 196px; left: 670px; position: absolute; height: 19px; width: 120px; border-left-style: ridge;
border-left-color: #808080; border-top-style: ridge; border-top-color: #808080"
Text="Status: aguardando"></asp:Label>

```

```

        <asp:Label ID="exec2" runat="server"
            style="top: 447px; left: 597px; position: absolute; height: 19px; width: 153px"
            Text="posiçãoXY: aguardando"></asp:Label>
    </ContentTemplate>
</asp:UpdatePanel>
</td>
<td class="style8">
    <asp:Label ID="Label14" runat="server"
        style="top: 561px; left: 1028px; position: absolute; height: 19px; width: 34px; color: #FF0000; font-size: x-
        large; font-weight: 700; font-family: Arial, Helvetica, sans-serif"
        Text="Emergência"></asp:Label>
    </td>
<td class="style16">
</td>
</tr>
</table>
</form>
</body>
</html>

```

Monitor.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Member_Monitoracao_Default : System.Web.UI.Page
{
    private WebReference.OPCconn objWSsend;
    private WebReference.OPCconn objWSread;
    private static int etapa;
    private string response;
    private string[] leitura;
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Timer1_Tick(object sender, EventArgs e)
    {
        leitura = new string[40];
        objWSread = new WebReference.OPCconn();
        leitura = objWSread.Read();
        if (leitura[1] == "0")
        {
            rot.Text = "Status: Aberta";
        }
        if (leitura[2] == "0")
        {
            molas2.Text = "Cilindro: Avançado";
        }
        else
        {
            molas2.Text = "Cilindro: Recuado";
        }
        if (leitura[3] == "0")
        {
            Tampas2.Text = "Cilindro: Recuado";
        }
        if (leitura[4] == "0")
        {
            Tampas2.Text = "Cilindro: Avançado";
        }
        if (leitura[5] == "0")
        {
            rot.Text = "Status: Fechada";
        }
        if (leitura[6] == "0")
        {
            molas.Text = "Status: Pronto";
        }
        else
    }
}

```

```

        {
            molas.Text = "Status: Vazio";
        }
        if (leitura[7] == "0")
        {
            Pinos1.Text = "Status: Pronto";
        }
        else
        {
            Pinos1.Text = "Status: Vazio";
            Panel1.BackColor = System.Drawing.Color.Empty;
        }
        if (leitura[8] == "0")
        {
            Tampas1.Text = "Status: Vazio";
        }
        else
        {
            Tampas1.Text = "Status: Pronto";
        }
        if (leitura[9] == "0")
        {
            rot2.Text = "Sentido: 270°";
        }
        if (leitura[10] == "0")
        {
            rot2.Text = "Sentido: 0°";
        }
        if (leitura[11] != "0")
        {
            exec.Text = "PosiçãoZ: 1";
        }
        if (leitura[12] != "0")
        {
            exec.Text = "PosiçãoZ: 2";
        }
        if (leitura[13] != "0")
        {
            exec.Text = "PosiçãoZ: 3";
        }
        if (leitura[14] != "0")
        {
            exec.Text = "PosiçãoZ: 4";
        }
        if (leitura[15] == "0")
        {
            pinos2.Text = "Cor:";
            Panel1.BackColor = System.Drawing.Color.Black;
        }
        if (leitura[16] == "0")
        {
            pinos2.Text = "Cor:";
            Panel1.BackColor = System.Drawing.Color.Silver;
        }
        exec2.Text = leitura[37];
    }

    protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
    {
        objWSread = new WebReference.OPCconn();
        response = objWSread.Emergencia(true);
    }
}

```

Monitor.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Monitor.aspx.cs"
Inherits="Member_Monitoracao_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">

```

```

<title></title>
<style type="text/css">
  .style6
  {
    text-align: left;
    font-weight: bold;
    font-size: x-large;
    font-family: "Comic Sans MS";
    height: 28px;
    width: 709px;
  }
  .style7
  {
    height: 225px;
    width: 709px;
  }
  .style8
  {
    height: 19px;
    width: 709px;
  }
  .style11
  {
    height: 28px;
    width: 288px;
  }
  .style12
  {
    height: 225px;
    width: 288px;
  }
  .style13
  {
    height: 19px;
    width: 288px;
  }
  .style14
  {
    height: 28px;
    width: 312px;
  }
  .style15
  {
    height: 225px;
    width: 312px;
  }
  #webcam1
  {
    top: 163px;
    left: 843px;
    position: absolute;
  }
  .style19
  {
    text-align: center;
    font-family: "Arial Black";
    font-weight: bold;
    font-size: large;
    color: #FFFFFF;
    text-decoration: underline;
  }
  .style20
  {
    text-align: center;
    font-family: "Arial Black";
    font-weight: bold;
    font-size: xx-large;
    color: #FFFFFF;
    text-decoration: underline;
  }
</style>
</head>
<body>
  <form id="form1" runat="server">

```

```

<table style="width: 124%; height: 476px; margin-left: 10px; margin-right: 0px; margin-bottom: 0px;">
  <tr>
    <td class="style11">
      <asp:ScriptManager ID="ScriptManager1" runat="server">
      </asp:ScriptManager>
    </td>
    <td class="style6">
      SUPERVISÓRIO<br />
    </td>
    <td class="style14">
      <asp:LoginStatus ID="LoginStatus1" runat="server"

      style="top: 22px; left: 1045px; position: absolute; height: 19px; width: 34px" />
      <asp:LoginName ID="LoginName1" runat="server"

      style="top: 22px; left: 1097px; position: absolute; height: 19px; width: 76px" />
    </td>
  </tr>
  <tr>
    <td class="style12">
      <asp:Panel ID="Panel5" runat="server"

      style="top: 10px; left: 820px; position: absolute; height: 560px; width: 5px; background-color: #000000">
      </asp:Panel>
      <asp:Panel ID="Panel7" runat="server"

      style="top: 10px; left: 10px; position: absolute; height: 559px; width: 5px; background-color: #000000">
      </asp:Panel>
      <asp:Panel ID="Panel8" runat="server"

      style="top: 10px; left: 1210px; position: absolute; height: 559px; width: 5px; background-color: #000000">
      </asp:Panel>
      <asp:Panel ID="Panel6" runat="server"

      style="top: 567px; left: 12px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
      </asp:Panel>
      <asp:Panel ID="Panel9" runat="server"

      style="top: 86px; left: 10px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
      </asp:Panel>
      <asp:Panel ID="Panel2" runat="server"
      style="top: 489px; left: 10px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
      </asp:Panel>
      <asp:Panel ID="Panel3" runat="server"
      style="top: 10px; left: 10px; position: absolute; height: 5px; width: 1200px; background-color: #000000">
      </asp:Panel>
    </td>
    <td class="style7">
      <asp:Image ID="Image1" runat="server"
      style="width: 410px; top: 148px; left: 297px; position: absolute; height: 314px; z-index: -1;"
      ImageUrl="~/pics/montagem.JPG" />

      <asp:Panel ID="Panel13" runat="server" BackColor="#003399"
      style="top: 20px; left: 831px; position: absolute; height: 61px; width: 203px">
        <div class="style20">
        </div>
      </asp:Panel>
      <asp:Panel ID="Panel15" runat="server" BackColor="#003399"

      style="top: 20px; left: 20px; position: absolute; height: 61px; width: 794px">
      <div class="style20">
        MONITORAÇÃO - MONTAGEM</div>
      </asp:Panel>
      <asp:Panel ID="Panel11" runat="server" BackColor="#003399"

      style="top: 500px; left: 20px; position: absolute; height: 61px; width: 794px">
      <div class="style20">
        MONITORAÇÃO - MONTAGEM</div>
      </asp:Panel>
      <asp:Panel ID="Panel14" runat="server" BackColor="#003399"
      style="top: 96px; left: 831px; position: absolute; height: 28px; width: 372px">

```

```

<div class="style19">
    MONITORAÇÃO</div>
</asp:Panel>

<asp:ImageButton ID="ImageButton1" runat="server" ImageUrl="-/pics/emerg.JPG"
    onclick="ImageButton1_Click"

    style="top: 497px; left: 878px; position: absolute; height: 70px; width: 103px; bottom: 5px;" />

</td>
<td class="style15">
    <script type="text/javascript">
        var height_array = new Array();
        var width_array = new Array();
        width_array[1] = 352;
        height_array[1] = 288;
    </script>
    
    <script type="text/javascript">
    <!--
        currentCamera1 = 1;
        errorimg1 = 0;
        document.images.webcam1.onload = Dolt1;
        document.images.webcam1.onerror = ErrorImage1;
        function LoadImage1() {
            uniq1 = Math.random();
            document.images.webcam1.src = "http://192.168.0.194:8080/cam_" + currentCamera1 + ".jpg?uniq=" + uniq1;
            document.images.webcam1.onload = Dolt1;
        }
        function PTZMouseDown1(e) {
            var IE = document.all ? true : false;
            var x, y;
            var myx, myy;
            var myifr = document.getElementById("_iframe-ptz");
            tp = getElPos1();
            myx = tp[0];
            myy = tp[1];
            if (IE) {
                var scrollX = document.documentElement.scrollLeft ? document.documentElement.scrollLeft :
document.body.scrollLeft;
                var scrollY = document.documentElement.scrollTop ? document.documentElement.scrollTop :
document.body.scrollTop;
                x = event.clientX - myx + scrollX;
                y = event.clientY - myy + scrollY;
            } else {
                x = e.pageX - myx;
                y = e.pageY - myy;
            }
            if ((width_array[currentCamera1] != null) && (width_array[currentCamera1] > 0)) x = Math.round((x * 400) /
width_array[currentCamera1]);
            if ((height_array[currentCamera1] != null) && (height_array[currentCamera1] > 0)) y = Math.round((y * 300) /
height_array[currentCamera1]);
            if (x > 400) x = 400;
            if (y > 300) y = 300;
            if (myifr != null) myifr.src = "http://192.168.0.194:8080/ptz?src=" + currentCamera1 + "&moveto_x=" + x +
"&moveto_y=" + y + "";
            return true;
        }
        function getElPos1() {
            el = document.images.webcam1;
            x = el.offsetLeft;
            y = el.offsetTop;
            elp = el.offsetParent;
            while (elp != null) {
                x += elp.offsetLeft;
                y += elp.offsetTop;
                elp = elp.offsetParent;
            }
            return new Array(x, y);
        }
    </script>
    <script type="text/javascript">
        function ErrorImage1() {
            errorimg1++;
            if (errorimg1 > 3) {

```

```

document.images.webcam1.onload = "";
document.images.webcam1.onerror = "";
document.images.webcam1.src = "offline.jpg";
} else {
    uniq1 = Math.random();
    document.images.webcam1.src = "http://192.168.0.194:8080/cam_" + currentCamera1 + ".jpg?uniq=" + uniq1;
}
}
function Dolt1() {
    errorimg1 = 0;
    window.setTimeout("LoadImage1()", 40);
}
//-->
</script>

<tr>
    <td class="style13">
        <br />
        <br />
        <asp:UpdatePanel ID="UpdatePanel3" runat="server">
            <ContentTemplate>
                <b>
                    <asp:Timer ID="Timer1" runat="server" Interval="1000" ontick="Timer1_Tick">
                    </asp:Timer>
                    Status conexão:</b><asp:Label ID="Label1" runat="server" Text="Aguardando"></asp:Label>
                    <asp:Label ID="Tampas2" runat="server"
                        style="top: 232px; left: 207px; position: absolute; height: 19px; width: 126px"
                        Text="Cilindro: aguardando"></asp:Label>
                    <asp:Label ID="rot2" runat="server"
                        style="top: 128px; left: 304px; position: absolute; height: 19px; width: 126px"
                        Text="Sentido: aguardando"></asp:Label>
                    <asp:Label ID="molas2" runat="server"
                        style="top: 302px; left: 679px; position: absolute; height: 19px; width: 127px"
                        Text="Cilindro: aguardando"></asp:Label>
                    <br />
                    <b> Etapa atual:</b><asp:Label ID="Label2" runat="server" Text="Aguardando"></asp:Label>
                    <asp:Label ID="pinos2" runat="server"
                        style="top: 427px; left: 262px; position: absolute; height: 19px; width: 107px"
                        Text="Cor: aguardando"></asp:Label>
                    <asp:Panel ID="Panel1" runat="server"
                        style="top: 430px; left: 294px; position: absolute; height: 15px; width: 76px">
                    </asp:Panel>
                    <br />
                    <b> Próxima etapa:</b><asp:Label ID="Label3" runat="server" Text="Aguardando"></asp:Label>
                    <asp:Label ID="apoio" runat="server"
                        style="top: 128px; left: 571px; position: absolute; height: 19px; width: 76px; border-left-style: ridge;
border-left-color: #808080; border-bottom-style: ridge; border-bottom-color: #808080"
                        Text="aguardando"></asp:Label>
                    <asp:Label ID="Tampas1" runat="server"
                        style="top: 210px; left: 208px; position: absolute; height: 38px; width: 129px; border-right-style: ridge;
border-right-color: #808080; border-top-style: ridge; border-top-color: #808080"
                        Text="Status: aguardando"></asp:Label>
                    <asp:Label ID="exec" runat="server"
                        style="top: 426px; position: absolute; height: 37px; width: 166px; left: 593px; border-top-style: ridge;
border-top-color: #808080; border-left-style: ridge; border-left-color: #808080;"
                        Text="posiçãoZ: aguardando"></asp:Label>
                    <asp:Label ID="Pinos1" runat="server"
                        style="top: 408px; position: absolute; height: 37px; width: 118px; left: 262px; border-right-style: ridge;
border-right-color: #808080; border-top-style: ridge; border-top-color: #808080; bottom: 128px;"
                        Text="Status: aguardando"></asp:Label>
                    <asp:Label ID="molas" runat="server"
                        style="top: 282px; left: 675px; position: absolute; height: 36px; width: 126px; border-left-style: ridge;
border-left-color: #808080; border-top-style: ridge; border-top-color: #808080; bottom: 229px;"
                        Text="Status: aguardando"></asp:Label>
                    <asp:Label ID="rot" runat="server"
                        style="top: 112px; left: 304px; position: absolute; height: 34px; width: 130px; border-right-style: ridge;
border-right-color: #808080; border-bottom-style: ridge; border-bottom-color: #808080;"
                        Text="Status: aguardando"></asp:Label>
                    <asp:Label ID="travamento" runat="server"
                        style="top: 196px; left: 670px; position: absolute; height: 19px; width: 120px; border-left-style: ridge;
border-left-color: #808080; border-top-style: ridge; border-top-color: #808080"
                        Text="Status: aguardando"></asp:Label>
                    <asp:Label ID="exec2" runat="server"

```

```

        style="top: 447px; left: 597px; position: absolute; height: 19px; width: 153px"
        Text="posiçãoXY: aguardando"></asp:Label>
    </ContentTemplate>
</asp:UpdatePanel>
</td>
<td class="style8">
    <asp:Label ID="Label14" runat="server"
        style="top: 522px; left: 997px; position: absolute; height: 19px; width: 34px; color: #FF0000; font-size: x-
large; font-weight: 700; font-family: Arial, Helvetica, sans-serif"
        Text="Emergência"></asp:Label>
    </td>
</tr>
</table>
</form>
</body>
</html>

```

Apêndice C – Programação do CLP

OB1 - <offline>

"CYCL_EXC" Cycle Execution

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 11/29/2009 11:56:57 PM

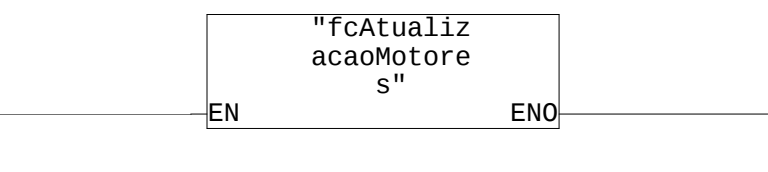
Interface: 02/15/1996 04:51:12 PM

Lengths (block/logic/data): 00666 00514 00022

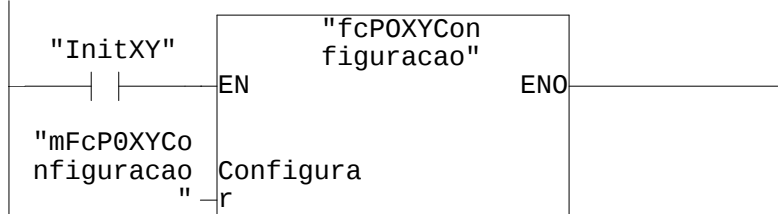
Name	Data Type	Address	Comment
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Block: OB1 "Main Program Sweep (Cycle)"

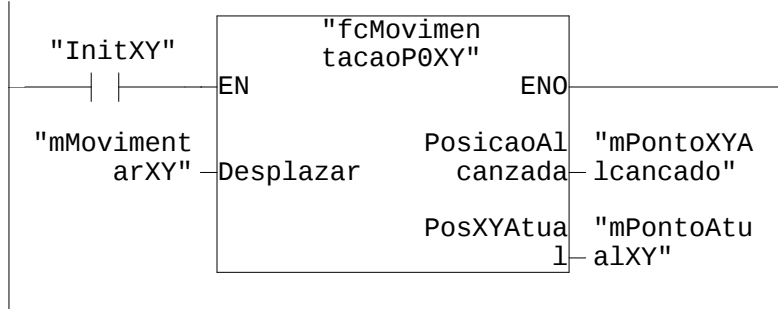
Network: 1



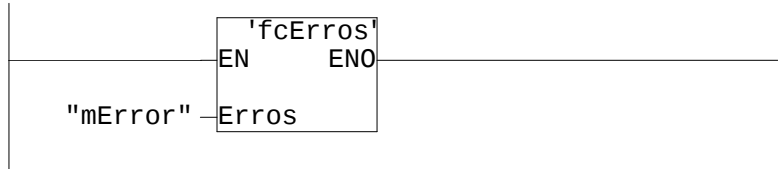
Network: 2



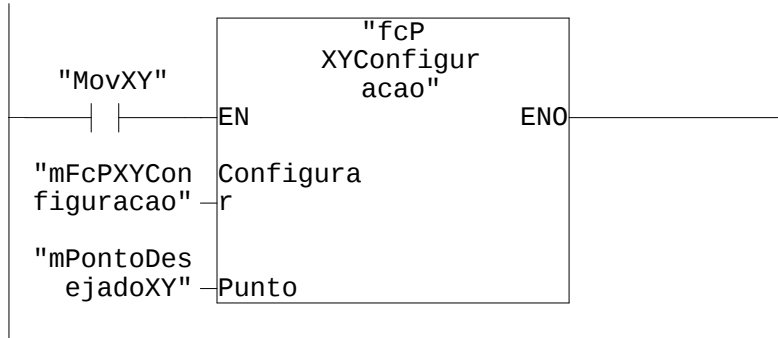
Network: 3



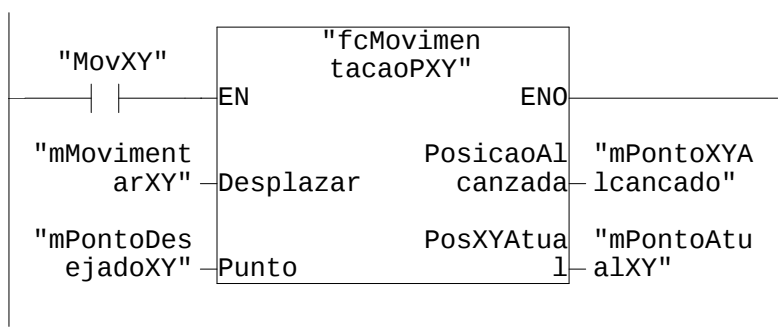
Network: 4



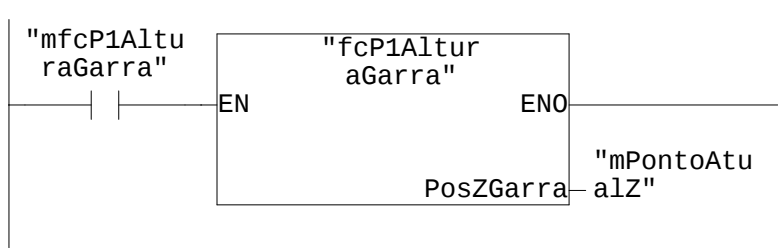
Network: 5



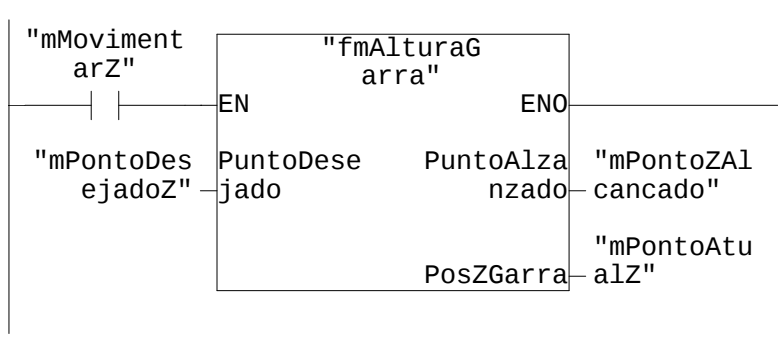
Network: 6



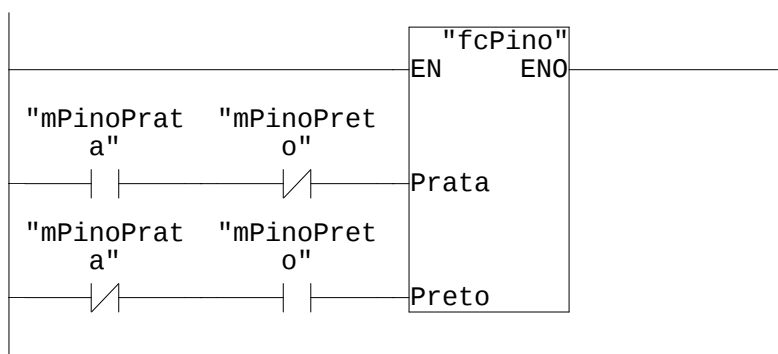
Network: 7



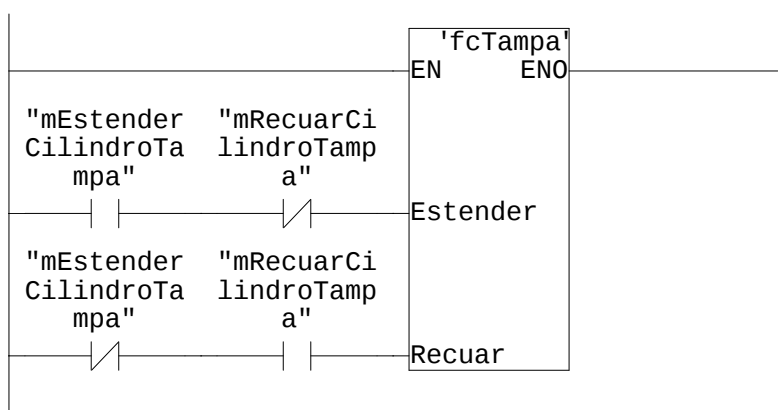
Network: 8



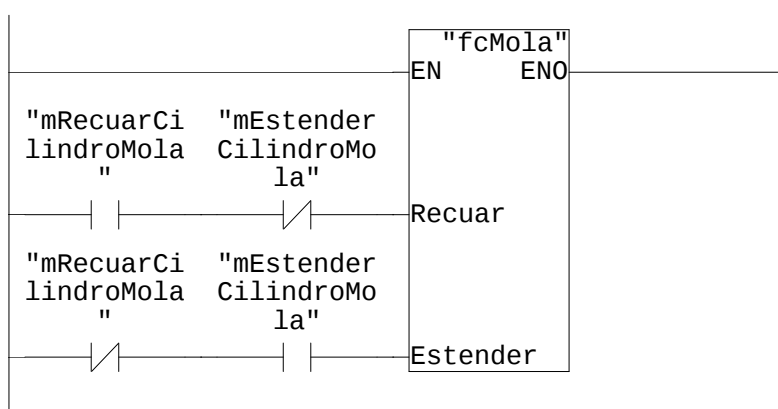
Network: 9



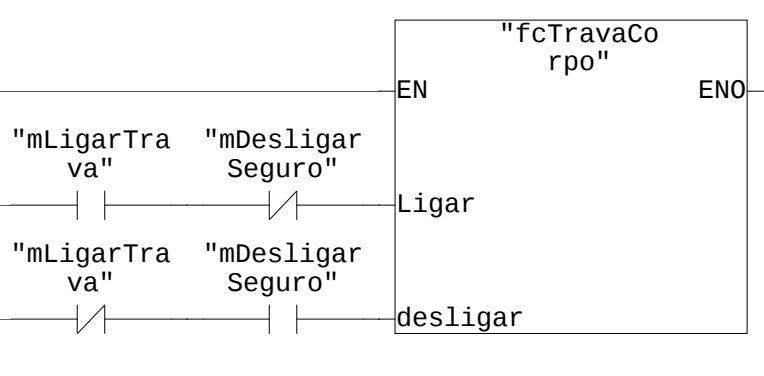
Network: 10



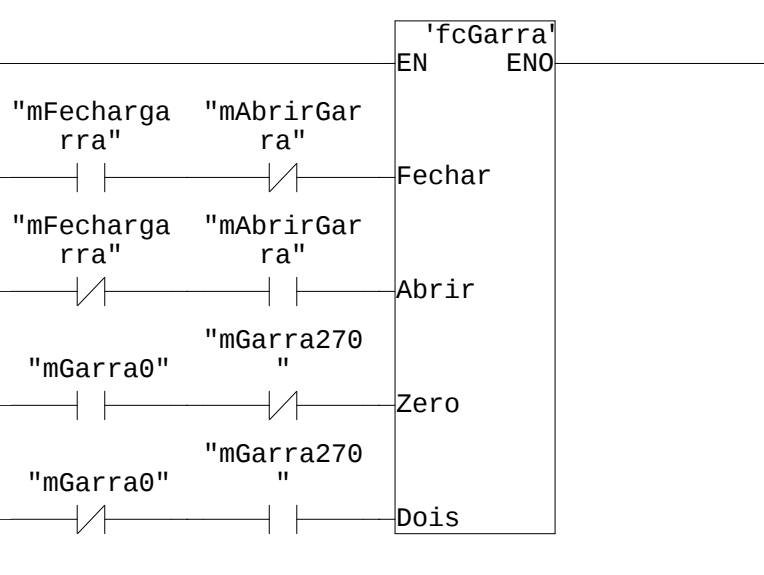
Network: 11



Network: 12



Network: 13



OB100 - <offline>

"COMPLETE RESTART" Complete Restart

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 11/30/2009 01:09:14 AM

Interface: 11/14/2009 01:46:42 PM

Lengths (block/logic/data): 00240 00110 00028

Name	Data Type	Address	Comment
TEMP		0.0	
OB100_EV_CLASS	Byte	0.0	16#13, Event class 1, Entering event state, Event logged in diagnostic buffer
OB100_STRTUP	Byte	1.0	16#81/82/83/84 Method of startup
OB100_PRIORITY	Byte	2.0	27 (Priority of 1 is lowest)
OB100_OB_NUMBR	Byte	3.0	100 (Organization block 100, OB100)
OB100_RESERVED_1	Byte	4.0	Reserved for system
OB100_RESERVED_2	Byte	5.0	Reserved for system
OB100_STOP	Word	6.0	Event that caused CPU to stop (16#4xxx)
OB100_STRT_INFO	DWord	8.0	Information on how system started
OB100_DATE_TIME	Date_And_Time	12.0	Date and time OB100 started
tmp	Bool	20.0	
tmp1	Bool	20.1	

Block: OB100 "Complete Restart"

Network: 1

```
CALL "POS_INIT"  
DB_NO    :=15  
CH_NO    :=1  
LADDR    :=272  
RET_VAL:=MW2
```

Network: 2

```
CALL "POS_INIT"  
DB_NO    :=25  
CH_NO    :=1  
LADDR    :=288  
RET_VAL:=MW4
```

FC5 - <offline>

"fcAtualizacaoMotores" Atuliza os dados do proyecto

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 10/09/2009 08:45:45 PM

Interface: 10/09/2009 08:41:45 PM

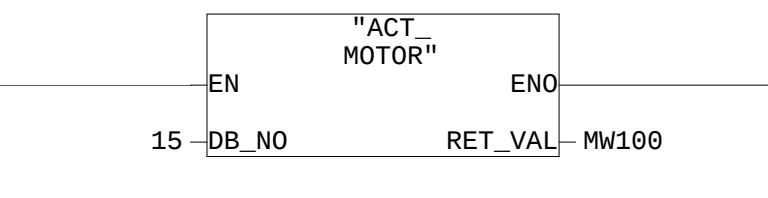
Lengths (block/logic/data): 00160 00066 00004

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC5 Atuliza motores

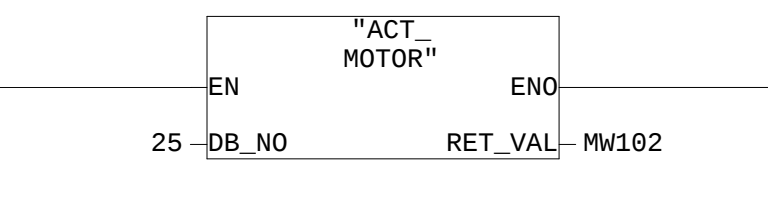
Network: 1

Atuliza dados do motor x



Network: 2

Atuliza dados do motor y



OB1 - <offline>

"CYCL_EXC" Cycle Execution

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 11/29/2009 11:56:57 PM

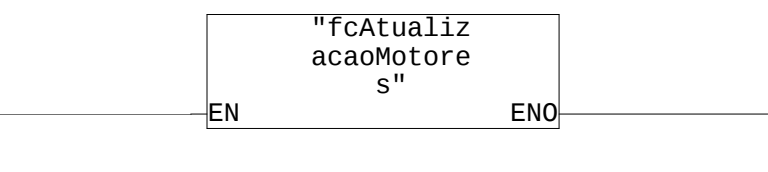
Interface: 02/15/1996 04:51:12 PM

Lengths (block/logic/data): 00666 00514 00022

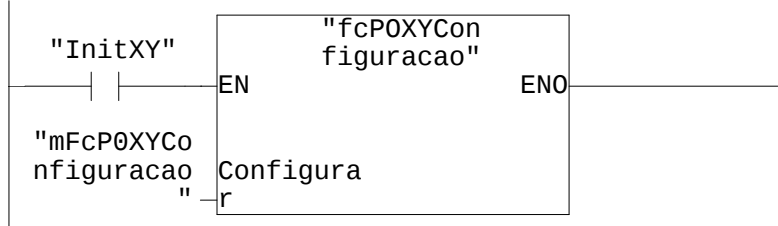
Name	Data Type	Address	Comment
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Block: OB1 "Main Program Sweep (Cycle)"

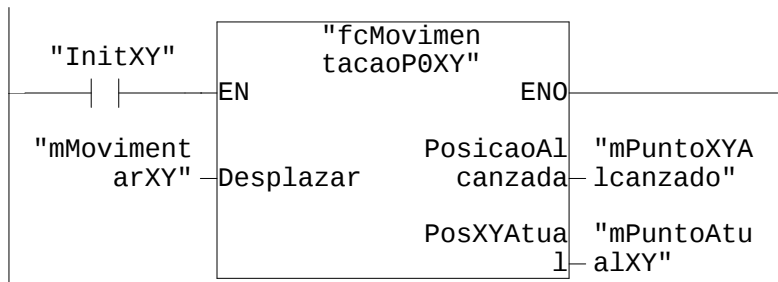
Network: 1



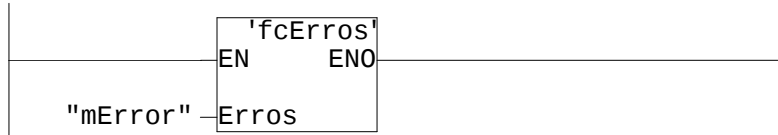
Network: 2



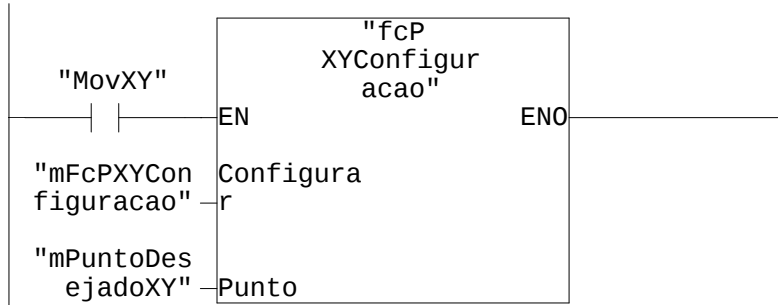
Network: 3



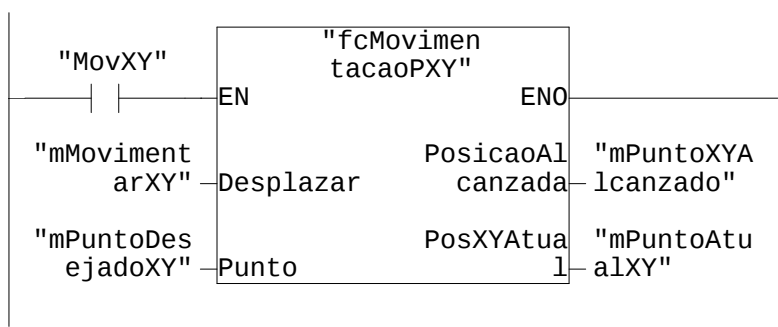
Network: 4



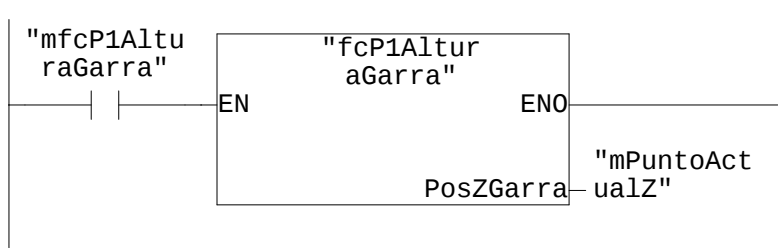
Network: 5



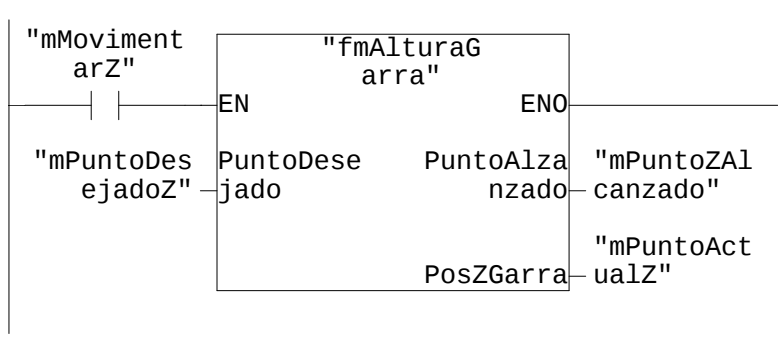
Network: 6



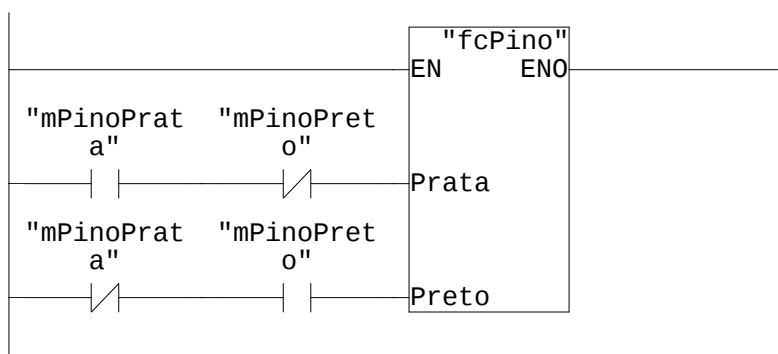
Network: 7



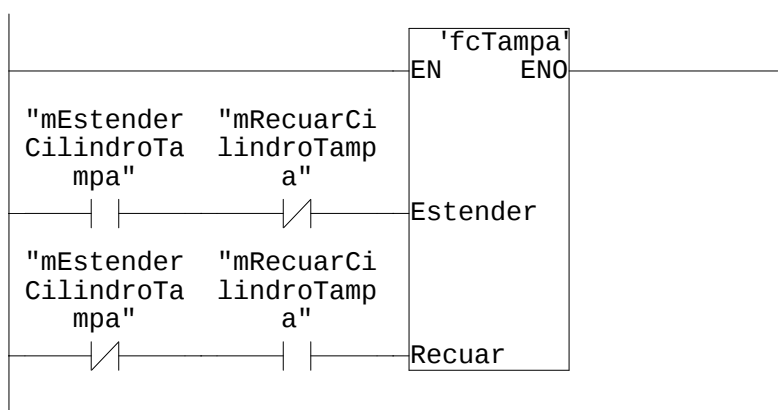
Network: 8



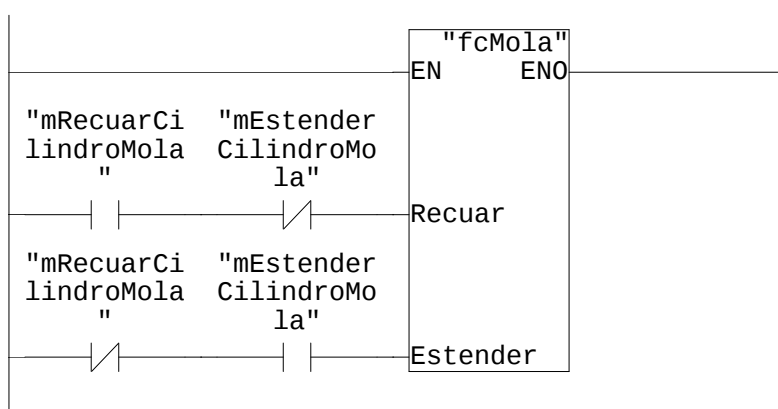
Network: 9



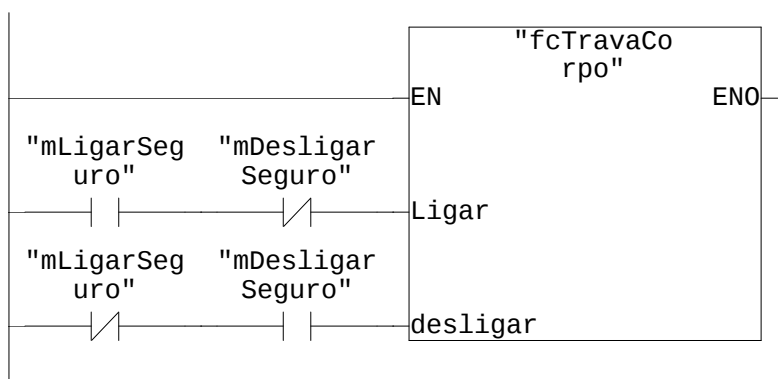
Network: 10



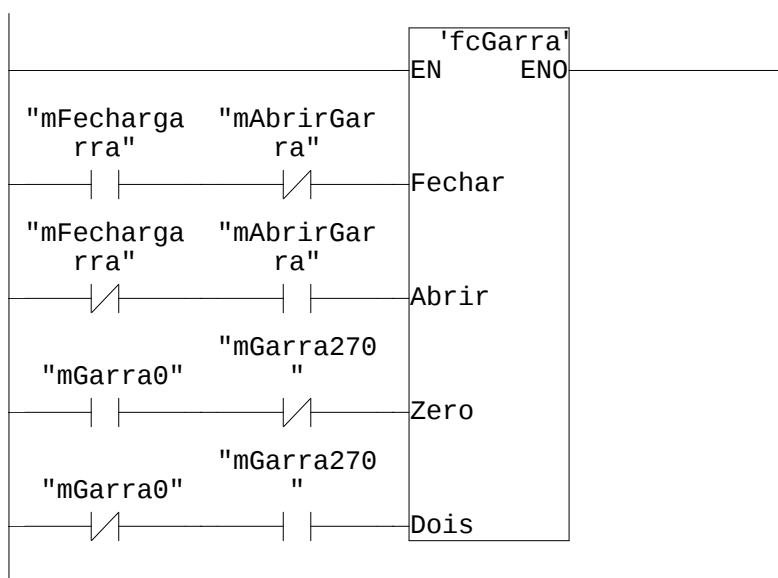
Network: 11



Network: 12



Network: 13



FC5 - <offline>

"fcAtualizacaoMotores" Atuliza os dados do proyecto

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 10/09/2009 08:45:45 PM

Interface: 10/09/2009 08:41:45 PM

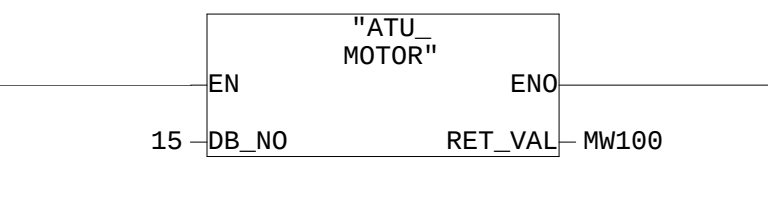
Lengths (block/logic/data): 00160 00066 00004

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC5 Atualiza motores

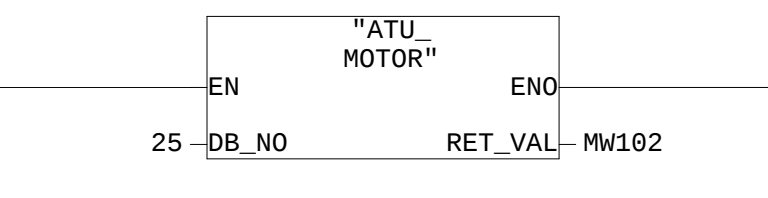
Network: 1

Atualiza dados do motor x



Network: 2

Atualiza dados do motor y



FC16 - <offline>

"fcPOXYConfiguracao" Movimenta ao P1

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 11/30/2009 10:52:13 PM

Interface: 11/24/2009 08:36:14 PM

Lengths (block/logic/data): 00348 00218 00002

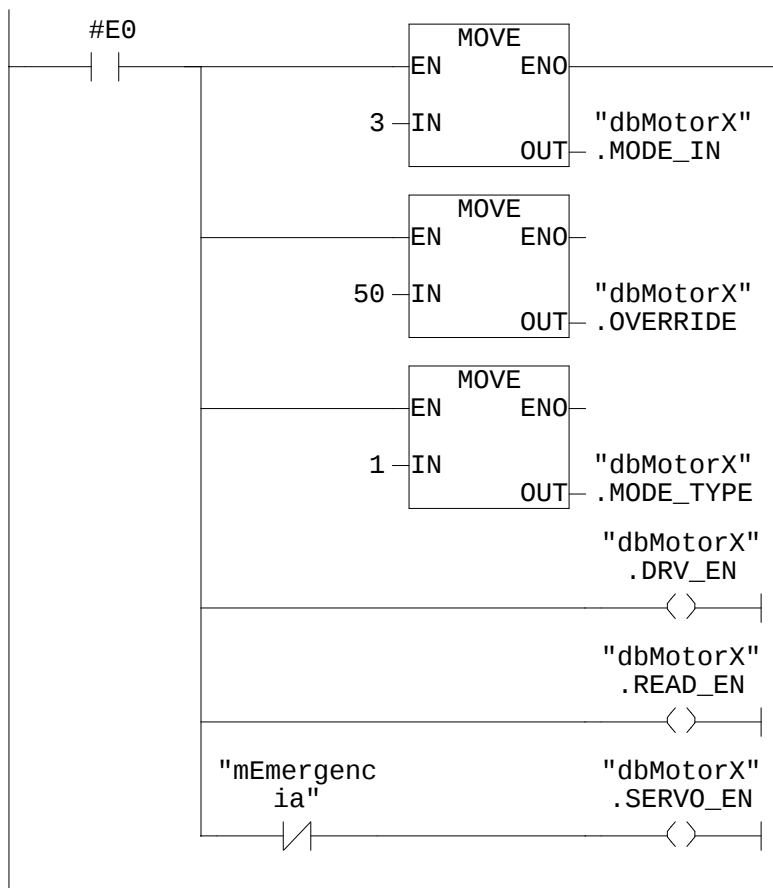
Name	Data Type	Address	Comment
IN		0.0	
Configurar	Bool	0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
E0	Bool	0.0	
Tmp	Bool	0.1	
RETURN		0.0	
RET_VAL		0.0	

Block: FC16

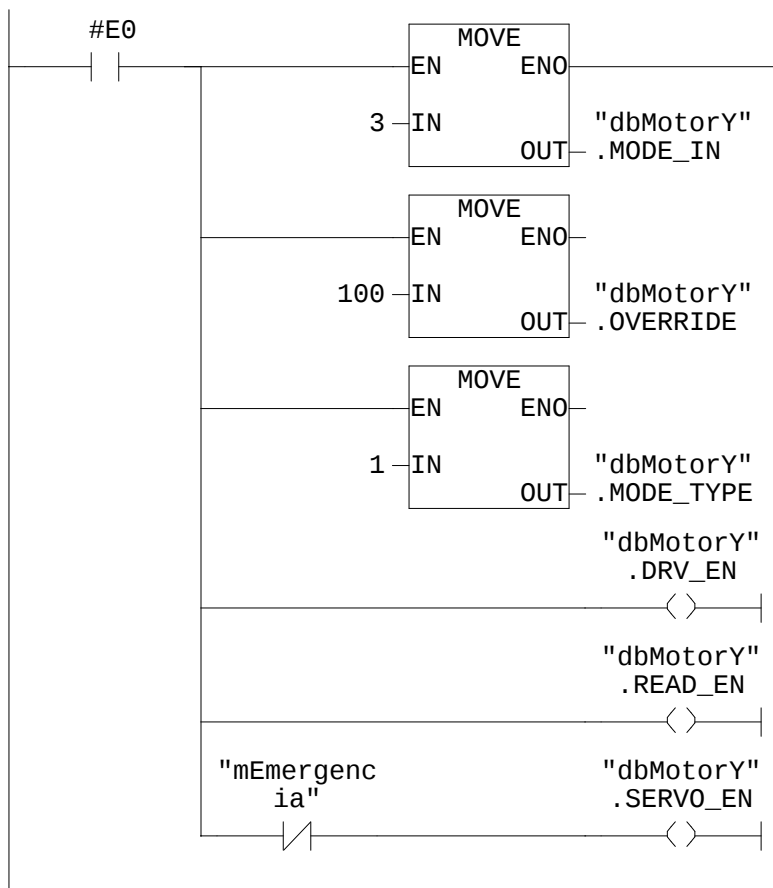
Network: 1 servo enable



Network: 2	drive enable
------------	--------------



Network: 3	drive enable
------------	--------------



FC17 - <offline>

"fcP XYConfiguracao" Movimentacao nos pontos P1 ate P14

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 11/30/2009 10:53:03 PM

Interface: 11/28/2009 05:02:16 PM

Lengths (block/logic/data): 00478 00332 00002

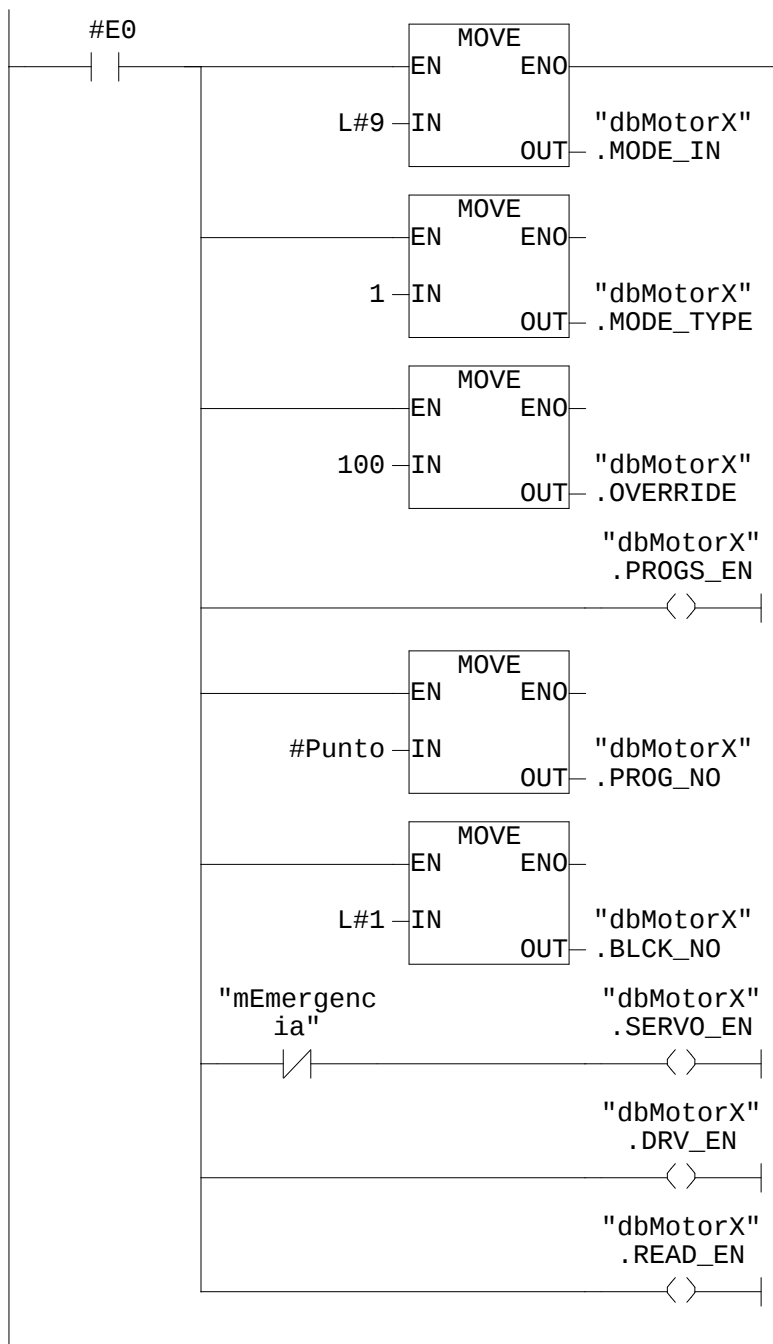
Name	Data Type	Address	Comment
IN		0.0	
Configurar	Bool	0.0	
Punto	Word	2.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
tStar	Bool	0.0	
tDesplazar	Bool	0.1	
E0	Bool	0.2	
E1	Bool	0.3	
E2	Bool	0.4	
okConfiguracao	Bool	0.5	
okMovimentacao	Bool	0.6	
okAtualizacao	Bool	0.7	
RETURN		0.0	
RET_VAL		0.0	

Block: FC17

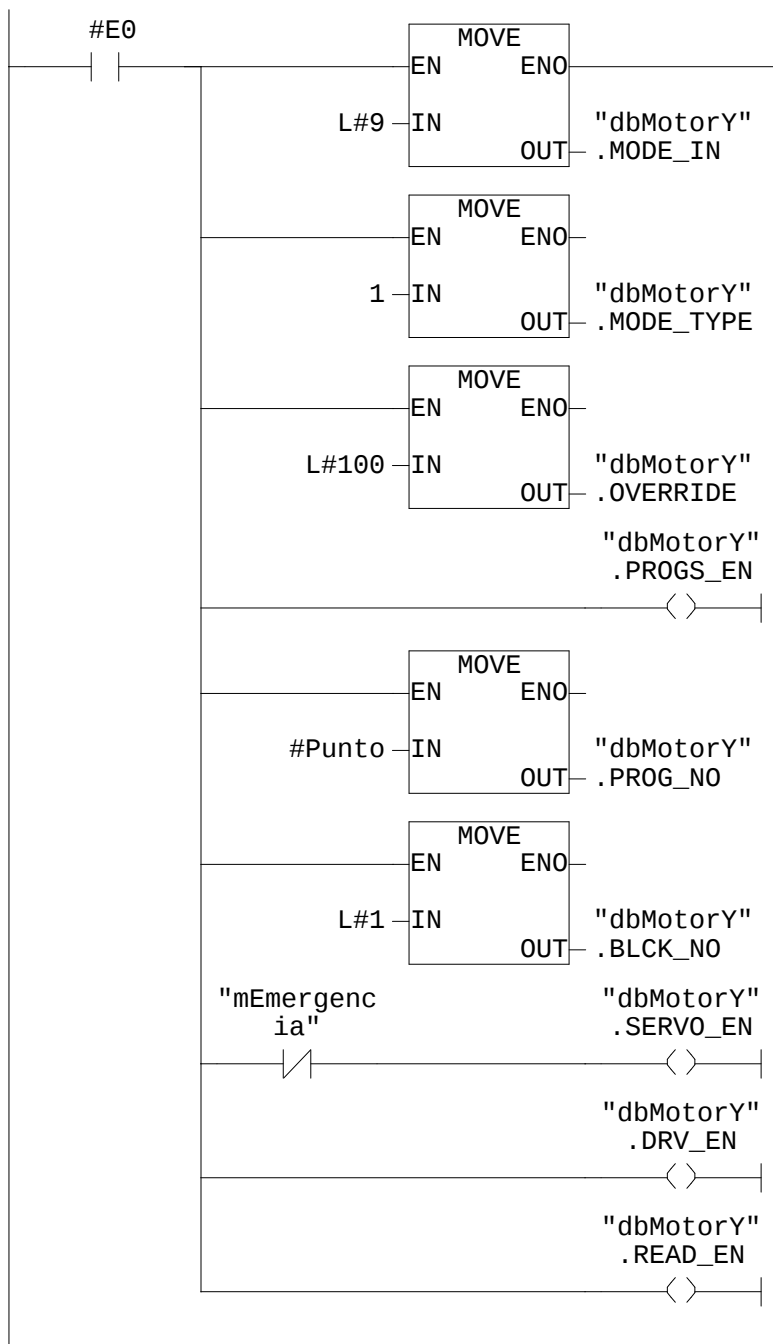
Network: 1 read enable



Network: 2	read enable
------------	-------------



Network: 3	read enable
------------	-------------



FC18 - <offline>

"fcErros" Acknowledge all Garante que todos os erros sejam ignorados

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 11/28/2009 03:20:17 PM

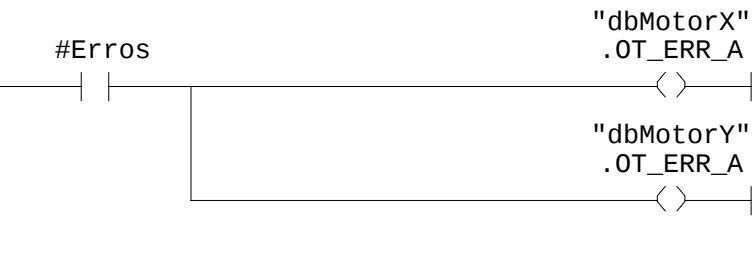
Interface: 11/28/2009 03:20:17 PM

Lengths (block/logic/data): 00112 00018 00000

Name	Data Type	Address	Comment
IN		0.0	
Erros	Bool	0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC18

Network: 1 quit error



OB1 - <offline>

"CYCL_EXC" Cycle Execution

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 11/29/2009 11:56:57 PM

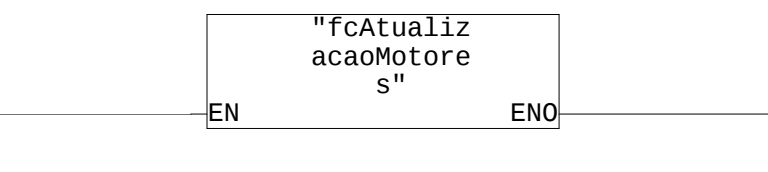
Interface: 02/15/1996 04:51:12 PM

Lengths (block/logic/data): 00666 00514 00022

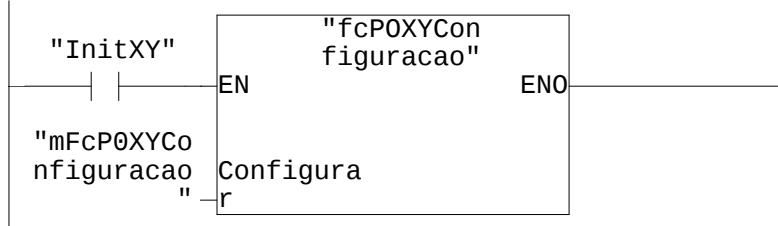
Name	Data Type	Address	Comment
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Block: OB1 "Main Program Sweep (Cycle)"

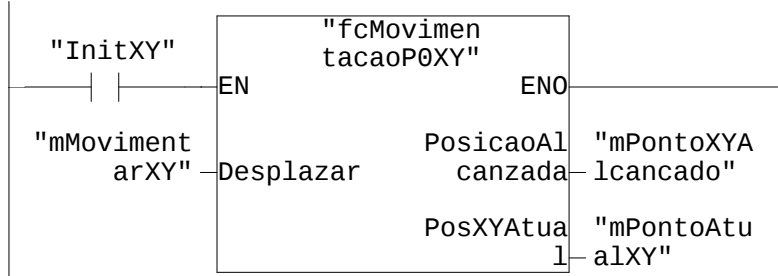
Network: 1



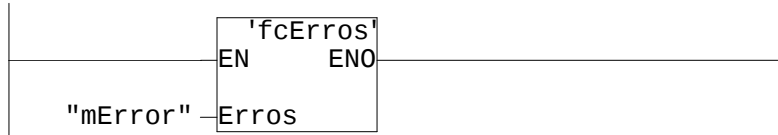
Network: 2



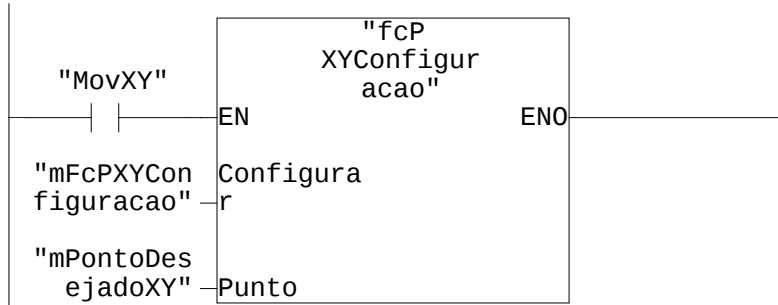
Network: 3



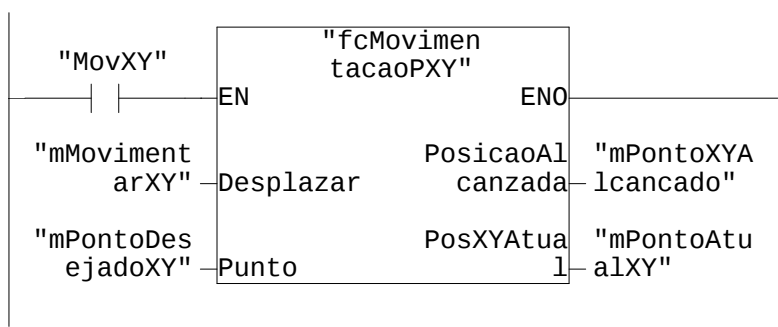
Network: 4



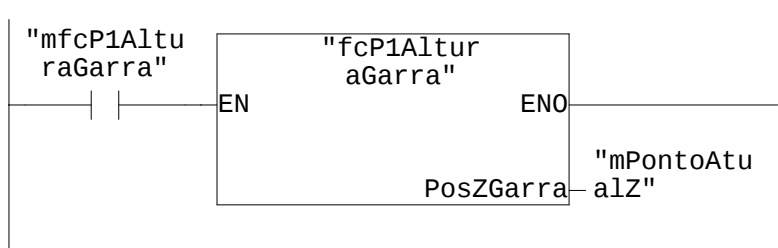
Network: 5



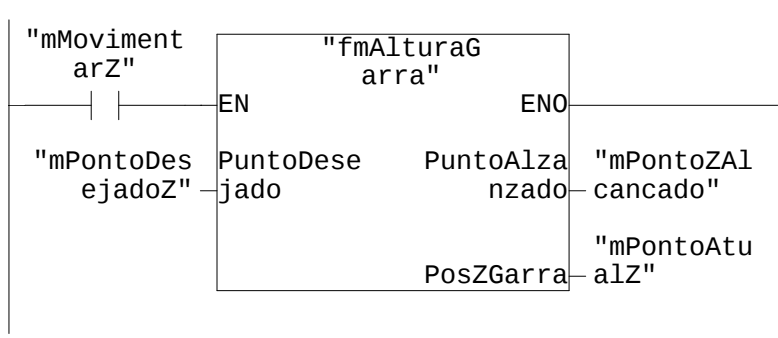
Network: 6



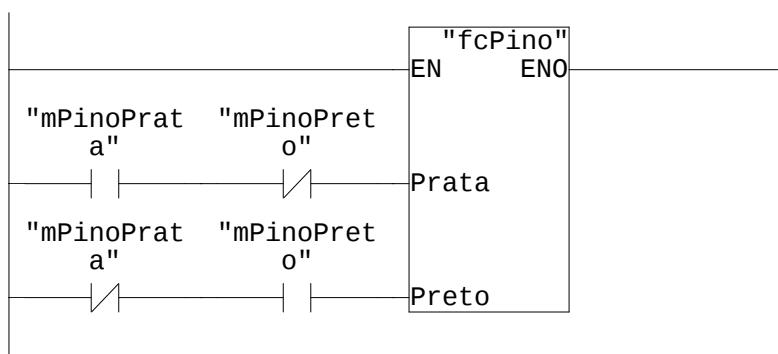
Network: 7



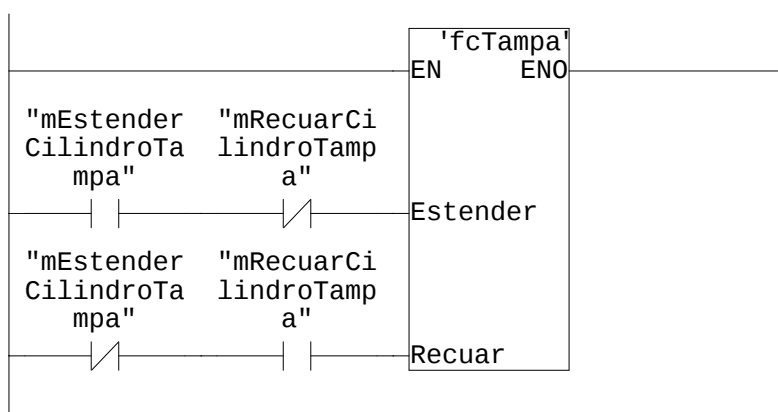
Network: 8



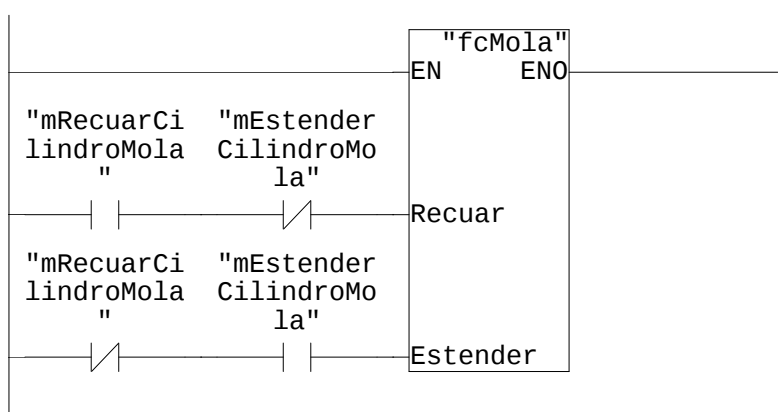
Network: 9



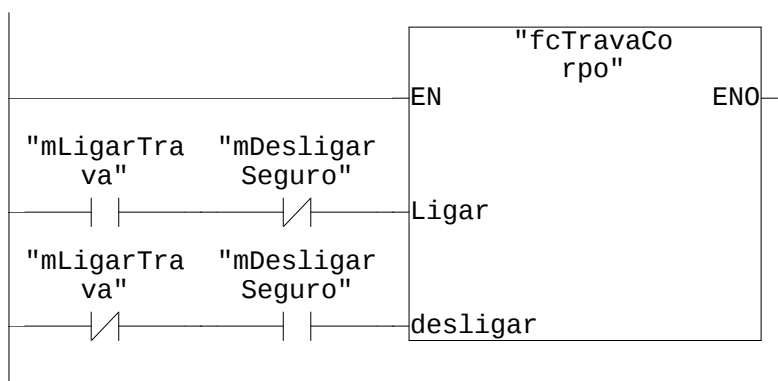
Network: 10



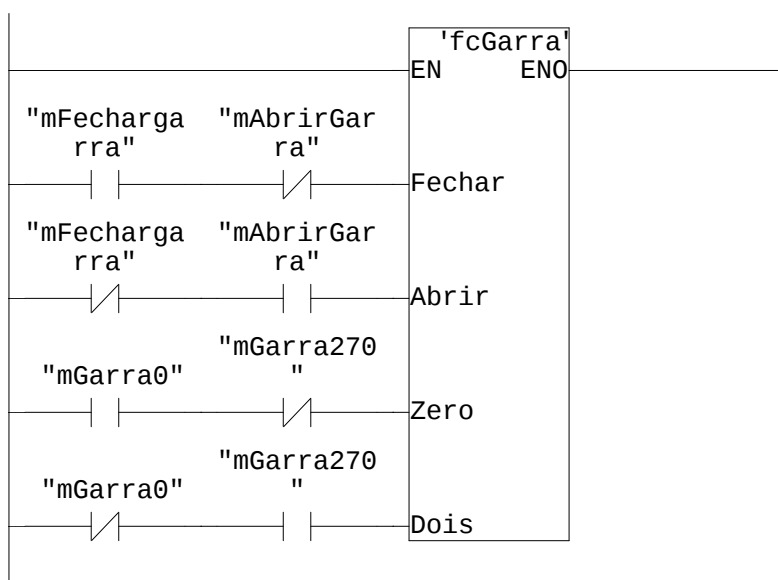
Network: 11



Network: 12



Network: 13



FC5 - <offline>

"fcAtualizacaoMotores" Atuliza os dados do proyecto

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 10/09/2009 08:45:45 PM

Interface: 10/09/2009 08:41:45 PM

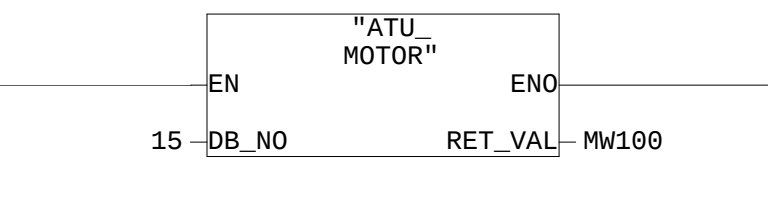
Lengths (block/logic/data): 00160 00066 00004

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC5 Atualiza motores

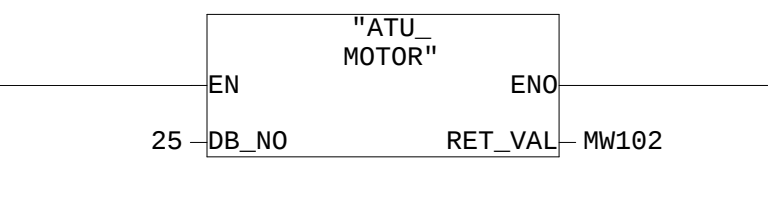
Network: 1

Atualiza dados do motor x



Network: 2

Atualiza dados do motor y



FC16 - <offline>

"fcPOXYConfiguracao" Movimenta ao P1

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 11/30/2009 10:52:13 PM

Interface: 11/24/2009 08:36:14 PM

Lengths (block/logic/data): 00348 00218 00002

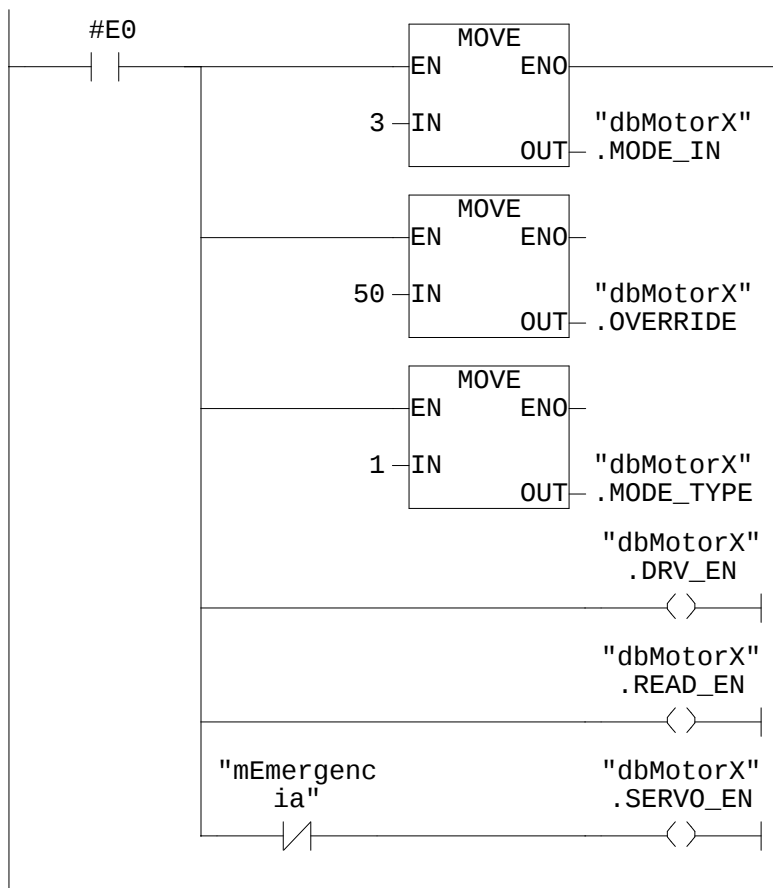
Name	Data Type	Address	Comment
IN		0.0	
Configurar	Bool	0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
E0	Bool	0.0	
Tmp	Bool	0.1	
RETURN		0.0	
RET_VAL		0.0	

Block: FC16

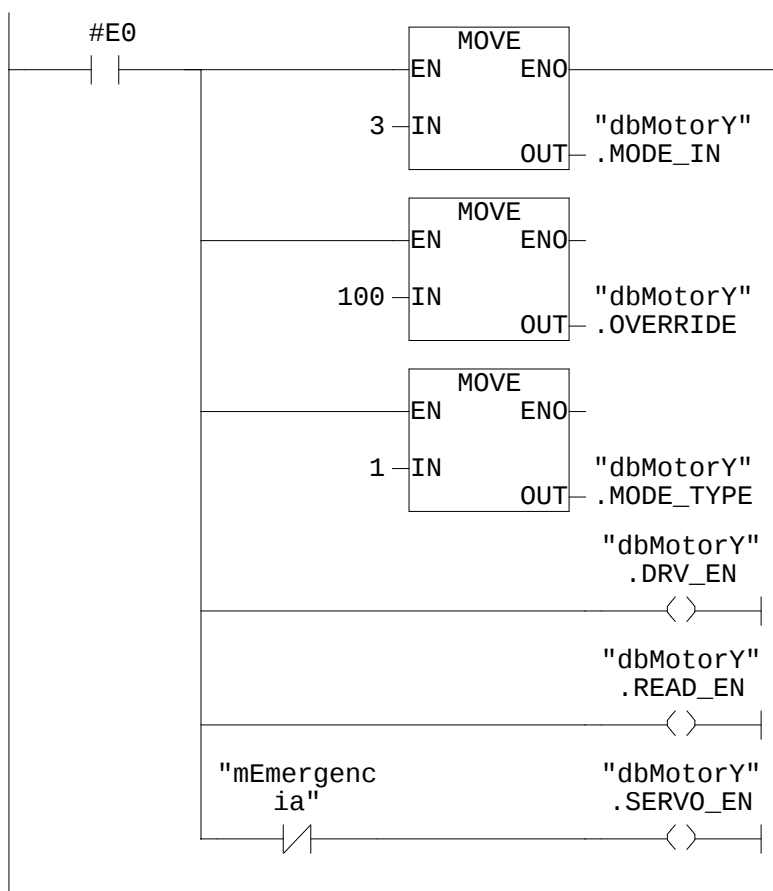
Network: 1 servo enable



Network: 2	drive enable
------------	--------------



Network: 3	drive enable
------------	--------------



FC17 - <offline>

"fcP XYConfiguracao" Movimentacao nos pontos P1 ate P14

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 11/30/2009 10:53:03 PM

Interface: 11/28/2009 05:02:16 PM

Lengths (block/logic/data): 00478 00332 00002

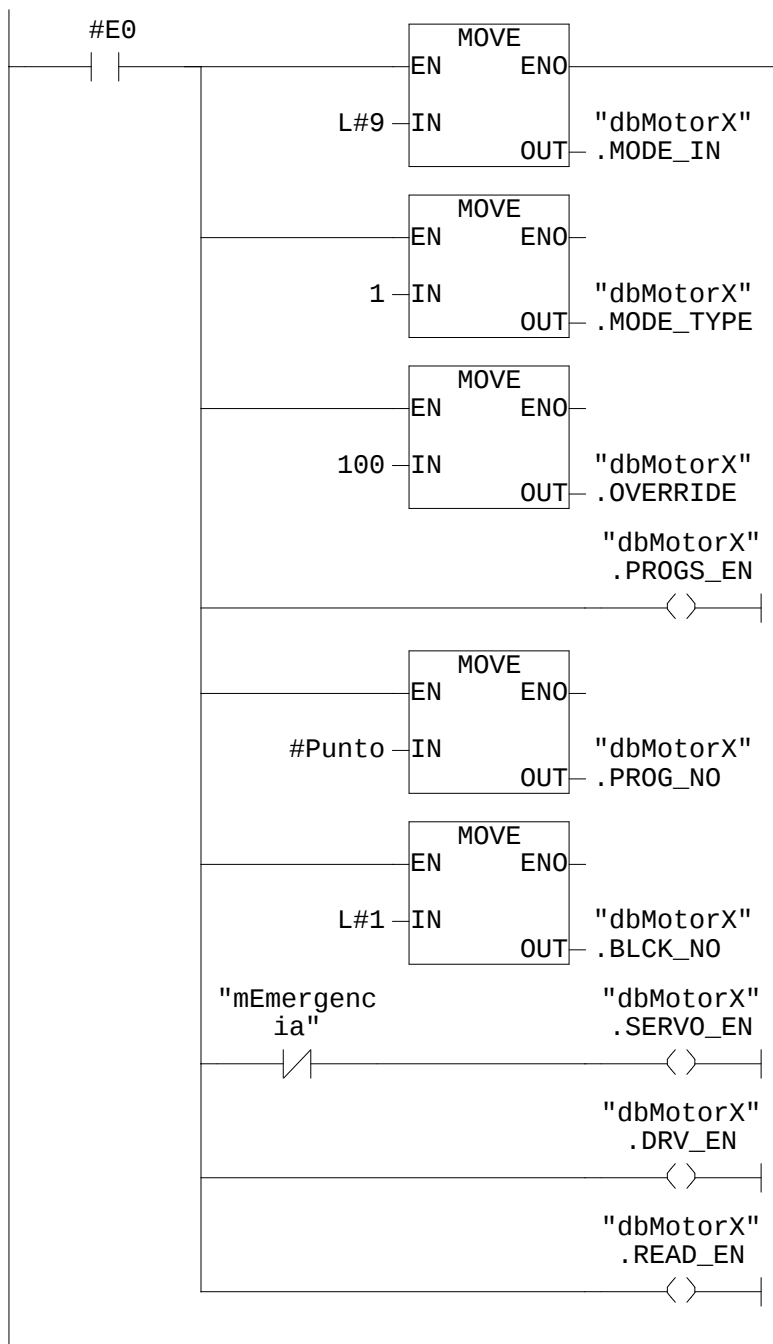
Name	Data Type	Address	Comment
IN		0.0	
Configurar	Bool	0.0	
Punto	Word	2.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
tStar	Bool	0.0	
tDesplazar	Bool	0.1	
E0	Bool	0.2	
E1	Bool	0.3	
E2	Bool	0.4	
okConfiguracao	Bool	0.5	
okMovimentacao	Bool	0.6	
okAtualizacao	Bool	0.7	
RETURN		0.0	
RET_VAL		0.0	

Block: FC17

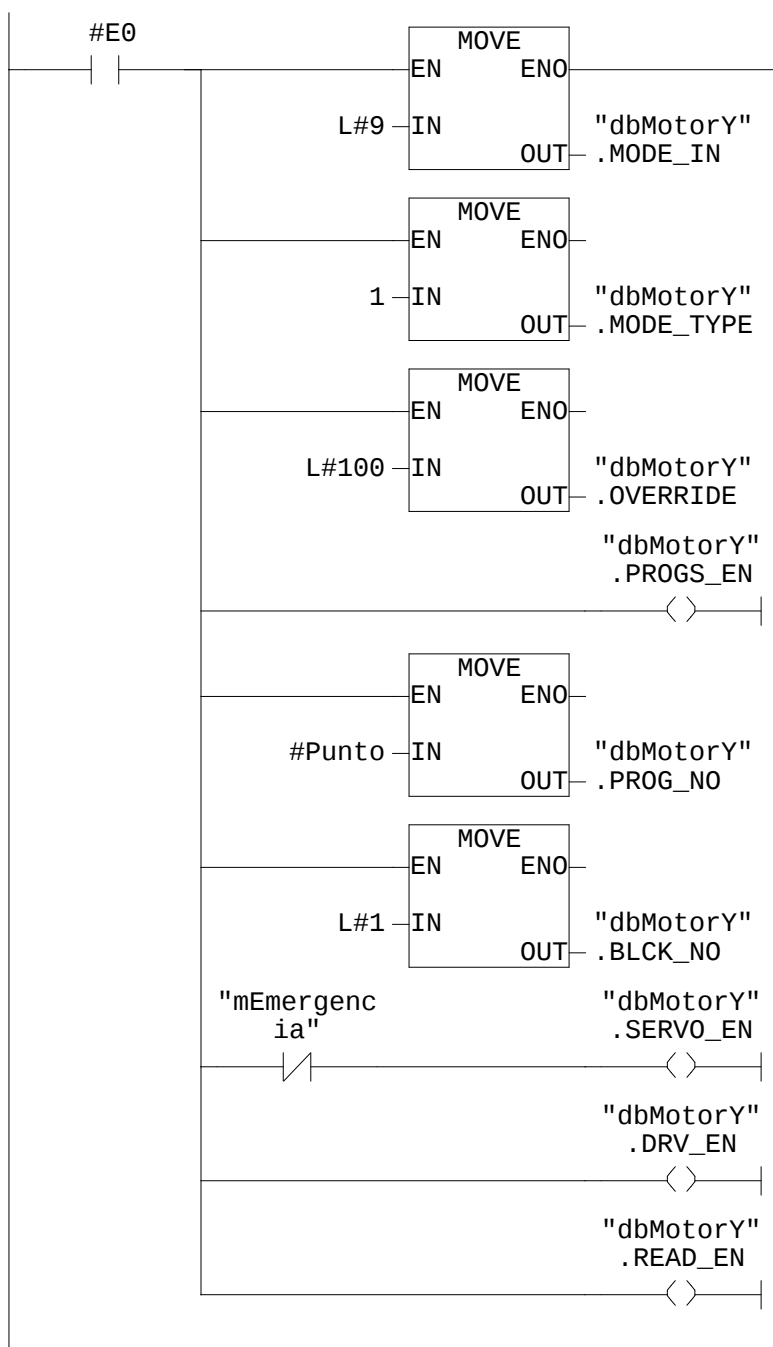
Network: 1 read enable



Network: 2	read enable
------------	-------------



Network: 3	read enable
------------	-------------



FC18 - <offline>

"fcErros" Acknowledge all Garante que todos os erros sejam ignorados

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 11/28/2009 03:20:17 PM

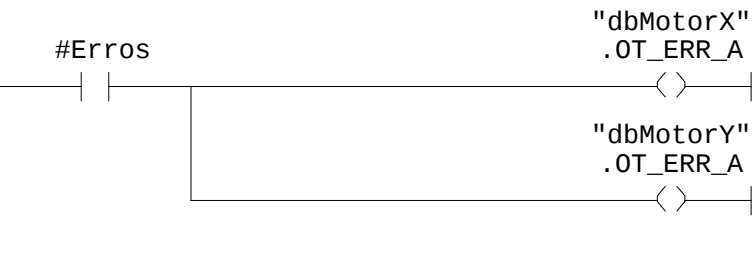
Interface: 11/28/2009 03:20:17 PM

Lengths (block/logic/data): 00112 00018 00000

Name	Data Type	Address	Comment
IN		0.0	
Erros	Bool	0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC18

Network: 1 quit error



FC24 - <offline>

"fcTampa" Suministra a peca Tampa

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 10/11/2009 03:49:26 AM

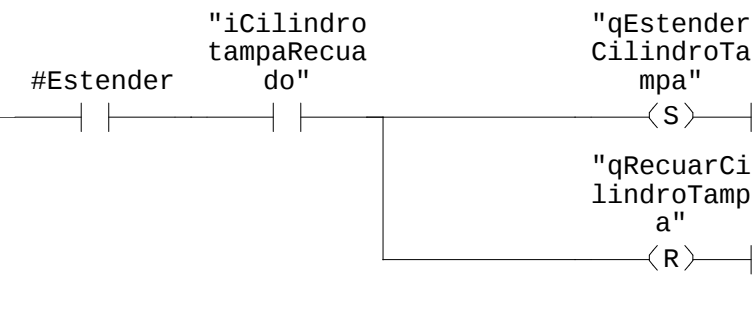
Interface: 10/11/2009 03:48:20 AM

Lengths (block/logic/data): 00120 00022 00000

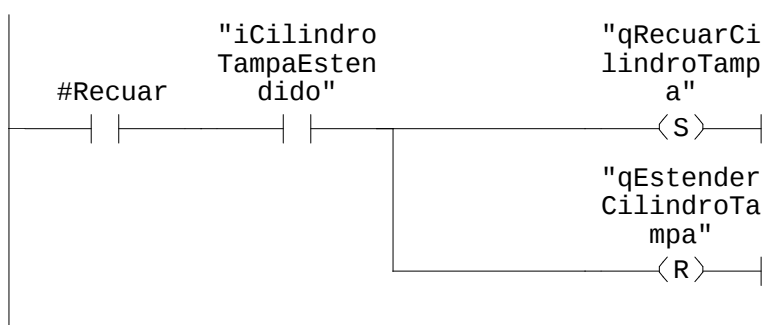
Name	Data Type	Address	Comment
IN		0.0	
Estender	Bool	0.0	
Recuar	Bool	0.1	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC24

Network: 1 Atuador para estender cilindro de tampa



Network: 2



FC25 - <offline>

"fcMola" Fornece a peca mola

Name:

Family:

Author:

Version: 0.1

Block version: 2

Time stamp

Code:

Interface:

10/14/2009 01:26:33 PM

10/11/2009 03:59:26 AM

Lengths (block/logic/data): 00112 00014 00000

Name	Data Type	Address	Comment
IN		0.0	
Recuar	Bool	0.0	
Estender	Bool	0.1	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC25

Network: 1 Atuador para recuar cilindro mola



Network: 2 Atuador para recuar cilindro mola



FC26 - <offline>

"fcPino" Fornece a peca pino, preto e prata

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 10/11/2009 04:15:17 AM

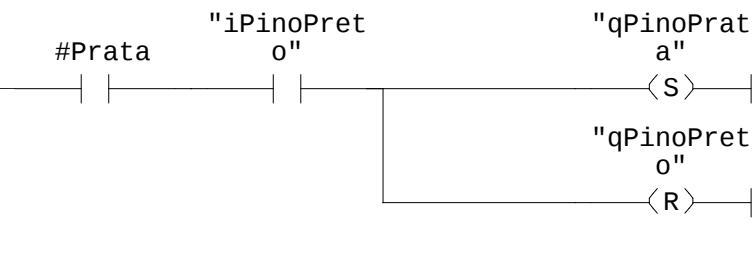
Interface: 10/11/2009 04:15:17 AM

Lengths (block/logic/data): 00120 00022 00000

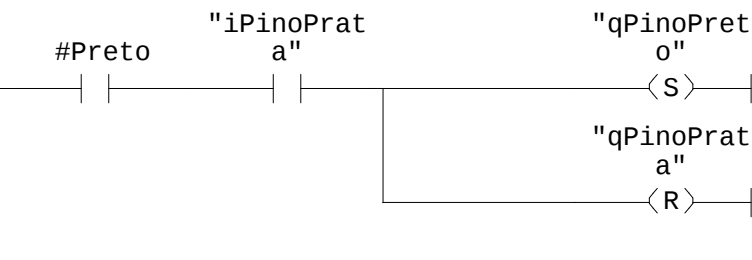
Name	Data Type	Address	Comment
IN		0.0	
Prata	Bool	0.0	
Preto	Bool	0.1	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC26

Network: 1



Network: 2



FC27 - <offline>

"fmAlturaGarra" Posiciona a garra na orientacao Z

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 11/30/2009 10:56:16 PM

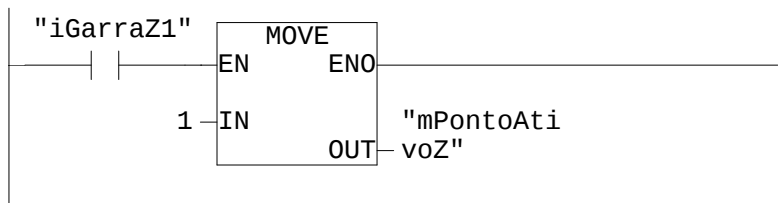
Interface: 11/28/2009 02:45:03 PM

Lengths (block/logic/data): 00450 00268 00008

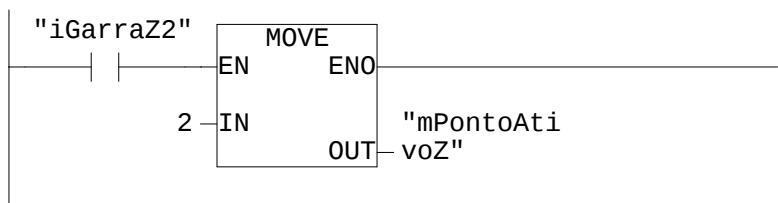
Name	Data Type	Address	Comment
IN		0.0	
PuntoDesejado	Word	0.0	
OUT		0.0	
PuntoAlzanzado	Bool	2.0	
PosZGarra	Word	4.0	
IN_OUT		0.0	
TEMP		0.0	
E0	Bool	0.0	
E1	Bool	0.1	
E2	Bool	0.2	
E3	Bool	0.3	
E4	Bool	0.4	
E5	Bool	0.5	
okConversao	Bool	0.6	
Subir	Bool	0.7	
Baixar	Bool	1.0	
Quieto	Bool	1.1	
okSubir	Bool	1.2	
okBaixar	Bool	1.3	
okQuieto	Bool	1.4	
okAtualizar	Bool	1.5	
tdesplazar	Bool	1.6	
pActualInt	Int	2.0	
pDeseadoInt	Int	4.0	
okPuntoActivo	Bool	6.0	
okChegou	Bool	6.1	
RETURN		0.0	
RET_VAL		0.0	

Block: FC27

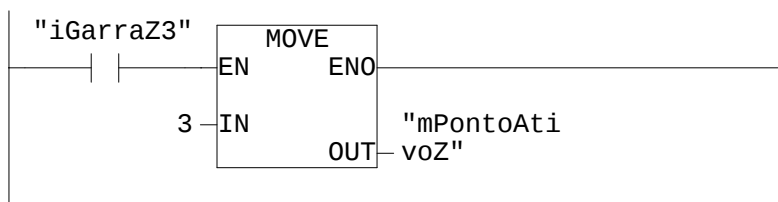
Network: 1 Atuador para baixar a garra



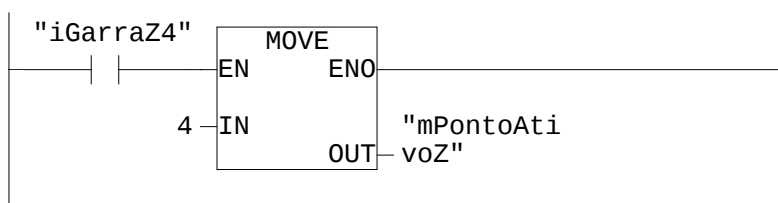
Network: 2



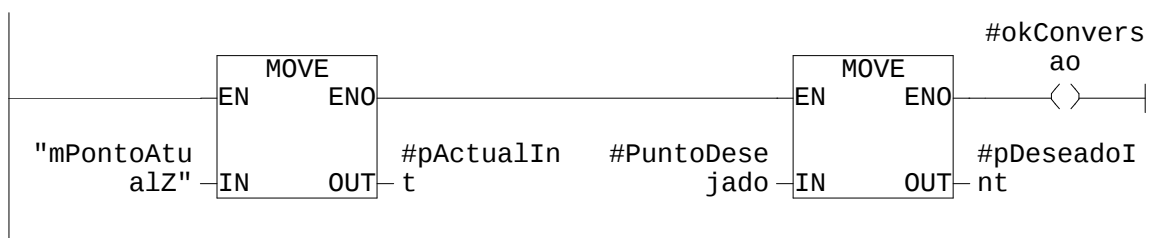
Network: 3



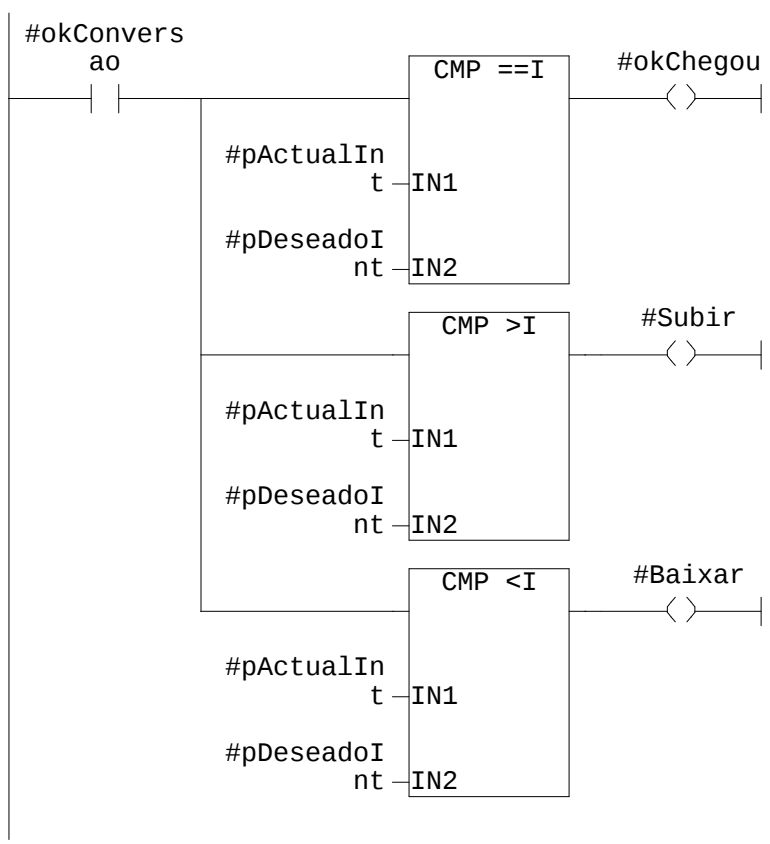
Network: 4



Network: 5

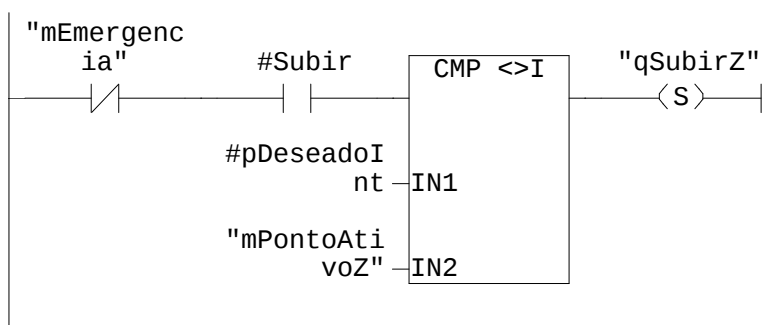


Network: 6 Atuador para subir a garra



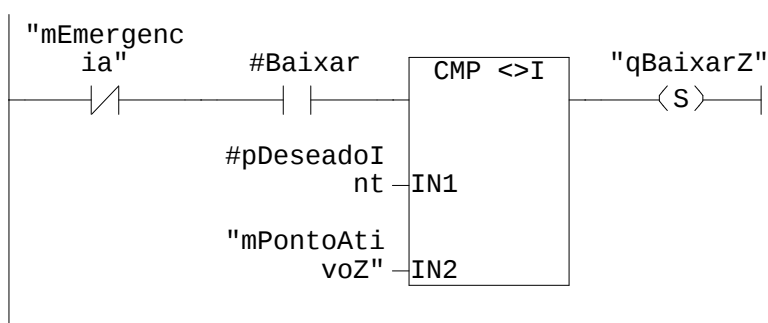
Network: 7

Atuador para subir a garra



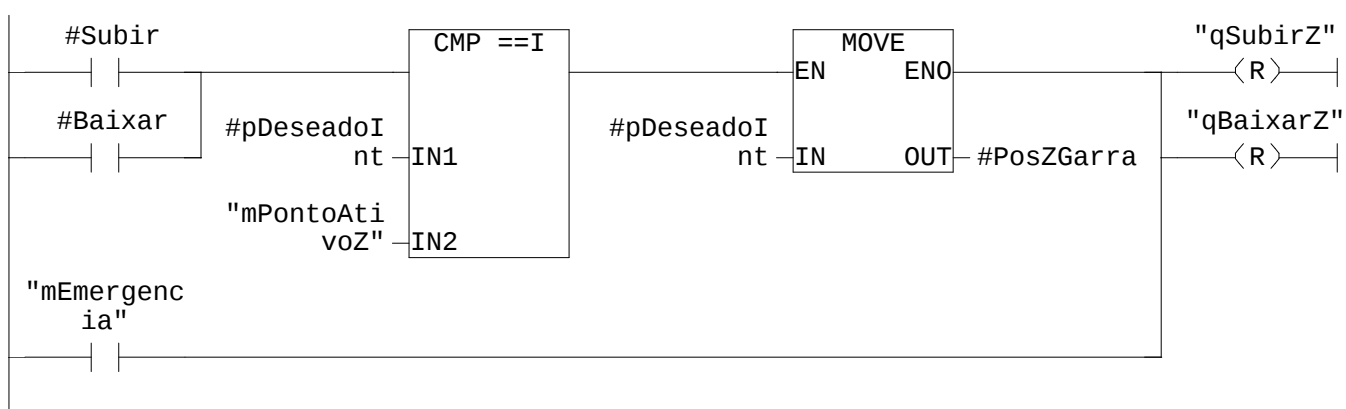
Network: 8

Atuador para subir a garra



Network: 9

Atuador para subir a garra



FC28 - <offline>

"fcP1AlturaGarra"

Name:

Author:

Time stamp

Code:

Interface:

Lengths (block/logic/data):

Family:

Version: 0.1

Block version: 2

11/30/2009 10:54:48 PM

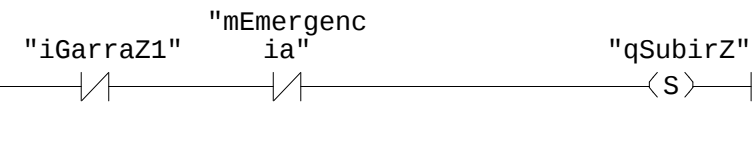
11/28/2009 02:42:35 PM

00136 00038 00000

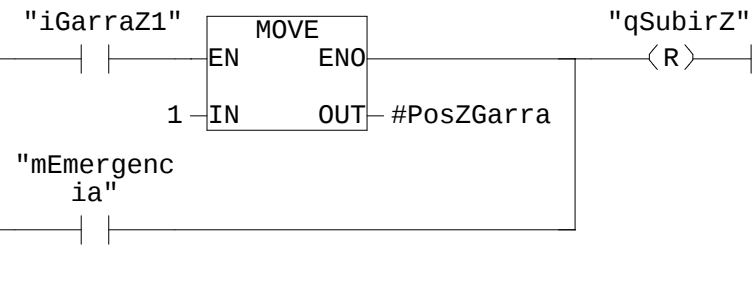
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
PosZGarra	Word	0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC28

Network: 1 Atuador para subir a garra



Network: 2 Atuador para subir a garra



FC30 - <offline>

"fcMovimentacaoP0XY"movimento ao ponto zero

Name:Family:

Author:Version: 0.1

Block version: 2

Time stamp Code:11/30/2009 01:06:20 AM

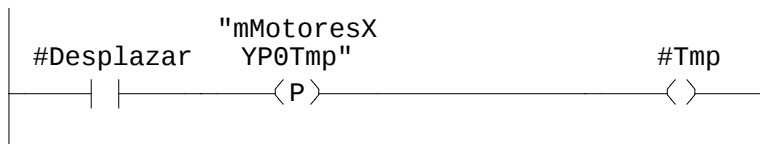
Interface:11/28/2009 02:29:35 PM

Lengths (block/logic/data): 00186 00068 00002

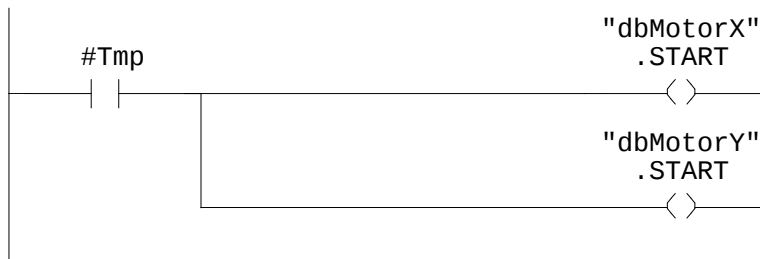
Name	Data Type	Address	Comment
IN		0.0	
Desplazar	Bool	0.0	
OUT		0.0	
PosicaoAlcanzada	Bool	2.0	
PosXYAtual	Word	4.0	
IN_OUT		0.0	
TEMP		0.0	
Tmp	Bool	0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC30

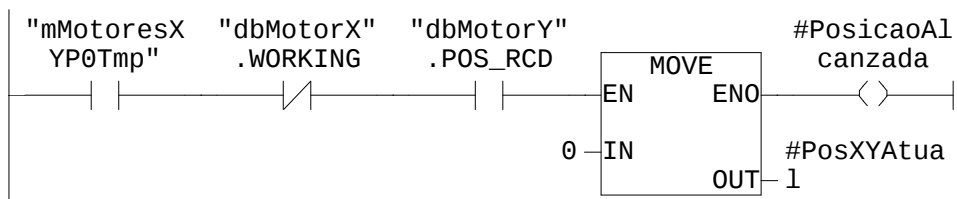
Network: 1start



Network: 2start



Network: 3	Dados do motorX
------------	-----------------



FC31 - <offline>

"fcMovimentacaoPXY"

Name:

Author:

Time stamp Code:

Interface:

Lengths (block/logic/data):

Family:

Version: 0.1

Block version: 2

11/30/2009 01:05:55 AM

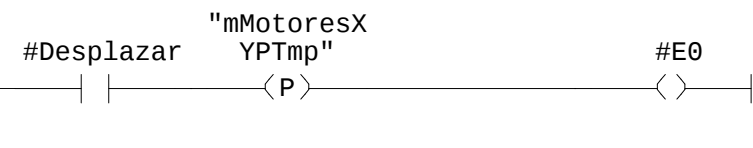
11/28/2009 04:49:21 PM

00178 00068 00002

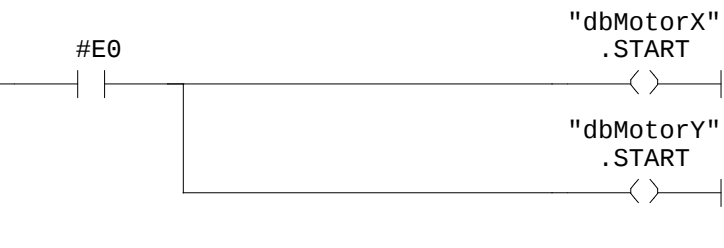
Name	Data Type	Address	Comment
IN		0.0	
Desplazar	Bool	0.0	
Punto	Word	2.0	
OUT		0.0	
PosicaoAlcanzada	Bool	4.0	
PosXYAtual	Word	6.0	
IN_OUT		0.0	
TEMP		0.0	
E0	Bool	0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC31

Network: 1start



Network: 2start



Network: 3

Dados do motorX

