UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE ENGENHARIA ELÉTRICA

ROBERTO DOUGLAS GUIMARÃES DE AQUINO

DESENVOLVIMENTO DE UMA FERRAMENTA PARA AUTOMAÇÃO E SUPERVISÃO, BASEADA NA PLATAFORMA ARDUINO E CONTROLADORES LÓGICOS PROGRAMÁVEIS, PARA APLICAÇÃO EM AMBIENTES INDUSTRIAIS

SOBRAL - CE

2015.2

ROBERTO DOUGLAS GUIMARÃES DE AQUINO

DESENVOLVIMENTO DE UMA FERRAMENTA PARA AUTOMAÇÃO E SUPERVISÃO, BASEADA NA PLATAFORMA ARDUINO E CONTROLADORES LÓGICOS PROGRAMÁVEIS, PARA APLICAÇÃO EM AMBIENTES INDUSTRIAIS

Trabalho de Conclusão de Curso apresentado ao Corpo Docente do Departamento de Engenharia Elétrica da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Me. Rômulo Nunes de C. Almeida.

SOBRAL - CE

2015.2

ROBERTO DOUGLAS GUIMARÃES DE AQUINO

DESENVOLVIMENTO DE UMA FERRAMENTA PARA AUTOMAÇÃO E SUPERVISÃO, BASEADA NA PLATAFORMA ARDUINO E CONTROLADORES LÓGICOS PROGRAMÁVEIS, PARA APLICAÇÃO EM AMBIENTES INDUSTRIAIS

Trabalho de Conclusão de Curso apresentado ao Corpo Docente do Departamento de Engenharia Elétrica da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Engenheiro Eletricista.

Aprovada em: 04/02/2016.

BANCA EXAMINADORA

Prof. Me. Rômulo Nunes de C. Almeida
Universidade Federal do Ceará (UFC)

Prof. Dr. Vandilberto Pereira Pinto
Universidade Federal do Ceará (UFC)

Prof. Killdary Aguiar de Santana
Universidade Federal do Ceará (UFC)

Prof. Me. Wilkley Bezerra Correia
Universidade Federal do Ceará (UFC)

Eng. Washington Luis A. Siqueira
Engenheiro Eletricista

AGRADECIMENTOS

À minha família, pelo incentivo à conquista desse sonho e pelo apoio financeiro.

Ao Programa de Educação Tutorial do curso de Engenharia Elétrica, pelo apoio financeiro e pelas atividades engrandecedoras que me proporcionaram.

Ao Prof. Me. Rômulo Nunes de C. Almeida, pela paciência e excelente orientação.

Aos participantes da banca examinadora Prof. Dr. Vandilberto Pereira Pinto e o Prof. Killdary Aguiar da Santana pelo tempo, pelas valiosas colaborações e sugestões.

Aos professores do campus, que me apoiaram bastante nesse percurso.

E, finalmente, aos amigos e colegas que fiz no decorrer dessa jornada.

Meus sinceros agradecimentos.

RESUMO

O projeto tem como objetivo desenvolver uma ferramenta simples e de baixo custo que auxilie e dê suporte para sistemas de controle e automação em ambientes industriais. A ferramenta utiliza um sistema supervisório ou SCADA (*Supervisory Control and Data Acquisition*), que consiste em um sistema que utiliza um *software* para a coleta, monitoramento e supervisão de dados. Para o projeto do sistema supervisório é utilizado o software Elipse SCADA em sua versão 2.29. Como controladores são utilizados: a plataforma Arduino em sua versão Uno e o controlador lógico programável (CLP) Siemens S7 1214c AC/DC/RLY, amplamente utilizado no setor industrial. Os dispositivos são interligados utilizando drivers específicos, disponibilizados pelos próprios fabricantes. Os testes de comunicação são realizados em vários estágios de acionamento e os resultados obtidos após os ensaios são satisfatórios.

Palavras-chave: Controladores Lógicos Programáveis, Arduino, Sistema Supervisório, Automação Industrial.

ABSTRACT

The project aims to develop a simple and inexpensive tool that helps and supports for control and automation systems in industrial environments. The tool uses a supervisory system or SCADA (Supervisory Control and Data Acquisition), consisting of a system that uses a software for collecting, monitoring and supervision of data. To the supervisory system design is used Elipse SCADA software in version 2.29. As controllers are used: the Arduino platform in version Uno and the programmable logic controller (PLC) Siemens S7 1214c AC/DC/RLY, widely used in industry. The devices are interconnected using specific drivers, made available by manufacturers. The communication tests are conducted at various stages of activation and results obtained after the tests are satisfactory.

Keywords: Programmable Logic Controllers, Arduino, Supervisory System, Industrial Automation.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABINEE	Associação Brasileira de Normas Técnicas
AC ou CA	Alternating Current
ATM	Asynchronous Transfer Mode
CI	Circuito Integrado
CLP ou PLC	Controlador Lógico Programável
DAC	Cartões de Aquisição de Dados
DAO	Data Access Objects
DB	Diagram Block
DC ou CC	Direct Current
DTR	Terminal de Dados Prontos
EEPROM	Electrically-Erasable Programmable Read-Only Memory
EPROM	Erasable Programmable Read-Only Memory
ERP	Enterprise Resource Planning
FDDI	Fiber Distributed Data Interface
FTDI	Future Technology Devices International
GND	Ground
GPRS	General Packet Radio Services
HSE	High Speed Ethernet
I2C	Inter-Integrated Circuit
ICSP	Serial Programming In-Circuit
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IHM ou HMI	Interface Homem Máquina
IP	Internet Protocol
ISA	International Society of Automation
ISO	International Organization for Standardization
LAN	Local Area Network
LDR	Light Dependent Resistor
LED	Light Emitting Diode
Micro SD Card	Micro Secure Digital Card
MPI	Message Passing Interface

ODBC	Open Data Base Connectivity
OLE	Object Linking and Embedding
OPC	OLE for Process Control
PC	Particular Computer
PoE	Power over Ethernet
PPI	Point to Point Interface
PROFIBUS	Process Field Bus
PWM	Pulse-Width Modulation
RFID	Radio-Frequency Identification
RLY	Relé
RTD	Resistance Temperature Detector
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SMS	Short Message Service
SNMP	Simple Network Management Protocol
SPI	Serial Peripheral Interface
SQL	Structured Query Language
SRAM	Static Random Access Memory
TCP	Transmission Control Protocol
TTL	Transistor-Transistor Lógico
UART	Universal Asynchronous Receiver/Transmitter
UCP ou CPU	Unidade Central de Processamento
UDP	User Datagram Protocol
USB	Universal Serial Bus
WAN	Wide Area Network

LISTA DE SÍMBOLOS

%	Porcento
®	Marca Registrada
A	Ampere
g	Gramas
Hz	Hertz
KB	Quilobit
kHz	Quilohertz
mA	Miliampere
MB	Megabit
Mbits/s	Megabits por segundo
MHz	Megahertz
mm	Milímetros
ns	Nanossegundos
psi	Libra-força por polegada quadrada
V	Volts

SUMÁRIO

1. INTRODUÇÃO

Verifica-se, atualmente, nos países mais industrializados a ocorrência de um processo de automação das plantas industriais que embora gradual, configura-se progressivo e inelutável. Os equipamentos utilizados registram aumentos significativos das vendas e constantes aperfeiçoamentos, possibilitando relevantes ganhos de produtividade e qualidade.

No segmento de automação industrial, podem-se distinguir dois nichos: o de equipamentos seriados, que são produzidos naturalmente pelas indústrias, e os equipamentos customizados, feitos sob encomenda. Segundo pesquisa realizada pela Associação Brasileira da Indústria Elétrica e Eletrônica (2013), os equipamentos seriados, geralmente vinculados à modernização do parque industrial em si, obtiveram um crescimento bastante expressivo. Já os equipamentos customizados, normalmente destinados a novos projetos e ampliações de plantas industriais, tiveram uma performance abaixo do que a do ano de 2012. Dados que corroboram este cenário foram apresentados pelo Instituto Brasileiro de Geografia e Estatística (IBGE). Conforme estudo realizado em 2013, os bens de capital para fins industriais cresceram 14% no que abrange os equipamentos seriados e caíram 19% no que diz respeito aos equipamentos não seriados. Isto reflete o momento da indústria no país, que não é de grandes investimentos. Dessa forma, é clara a necessidade de novas pesquisas destinadas à descoberta de novas tecnologias ou a inovação do setor. (O SETOR ELÉTRICO, 2015)

Para Rosário (2009), inovar é fazer coisas diferentes ou de outra maneira. É sair da rotina e experimentar novas soluções ou formular novos problemas, utilizando a criatividade para satisfazer necessidades não ou insuficientes satisfeitas. E nesse ambiente competitivo, a capacidade de inovar tornou-se essencial.

Naturalmente, os sistemas de controle e automação são utilizados para otimizar processos produtivos, assegurar a integridade dos operários ou evitar interferência humana, garantindo assim qualidade padrão e singular aos produtos. Assim, esses sistemas podem reduzir a quantidade de mão de obra necessária em um processo industrial. Outra forma de lidar com a eficiência do processo é configuração de um sistema supervisório que monitore o sistema de automação industrial de forma a notificar o operador de possíveis falhas e possa, inclusive, solucionar problemas.

1.2. Delimitação do Tema

A busca por novas ferramentas na área de automação industrial, contribui diretamente para o processo de inovação da área e a redução dos preços de equipamentos da área. Contribuindo para que pequenas empresas, possam contar com sistemas de soluções especificas, características de sistemas customizados, que possuam a praticidade dos equipamentos seriados, fácil aquisição e custo reduzido. Com base nestas premissas o presente trabalho busca a integração de sistemas de automação industrial baseados em Controladores Lógicos Programáveis (CLP´s) junto a sistemas microcontrolados de baixo custo baseados em plataforma *"open source"* (Arduino).

A comunicação entre os dispositivos será realizada através de um software supervisório utilizando a rede local (LAN), devido à simplicidade quanto à configuração desse tipo de rede, baixo custo e o amplo uso em ambientes industriais.

Dessa forma, o processo de gerenciamento de dados, supervisão e automação poderá ser realizado pelo software supervisório e a comunicação entre as ferramentas será direta e sem os problemas intrínsecos à comunicação interdispositivos.

O software escolhido como ferramenta para a criação do supervisório foi o Elipse SCADA versão 2.29, da Elipse Software, devido a ampla utilização do software no mercado, por ser uma empresa nacional e disponibilizar uma versão teste gratuita. Em sua versão demonstrativa são disponibilizadas uma quantidade limitada de marcadores (tags) que serão suficientes para o estudo.

O controlador lógico programável (CLP) utilizado deverá possuir ampla disponibilidade de uso nos laboratórios do campus, ter considerável aceitação e utilização nas indústrias locais. Um bom suporte e assistência técnica também deverão ser considerados, visto que serão utilizados protocolos e métodos de comunicação pouco usuais. Dessa forma, foi escolhido o controlador Siemens S7 1214c AC/AD/RLY.

O Arduino como uma ferramenta robusta de baixo custo quando comparado ao CLP e segundo Monk (2015) possui ambiente de desenvolvimento integrado (IDE) próprio, simples e de fácil programação. Mostrou-se bastante conveniente para auxiliar nas tarefas até então desempenhadas exclusivamente pelos CLPs. Dessa forma, essa plataforma foi escolhida como controlador de suporte e deverá se conectar ao supervisório, assim como o CLP, e trocar informações com todas as demais ferramentas.

Através de pesquisas e levantamentos de preços e custos, observou-se que ao utilizar a plataforma Arduino em processos menos complexos e que não exigem o uso

específico de um CLP, seja na função de substituição ou suporte, pode-se encontrar uma redução em torno de 90% no valor final do investimento.

A escolha do tipo de protocolo de comunicação TCP/IP está baseada na confiabilidade na transmissão dos dados, na realização de transferência simultânea entre cliente-servidor e a presença de controles avançados para a manutenção do desempenho da conexão (FOROUZAN; FEGAN, 2009).

Os estudos realizados na área geralmente contemplam exclusivamente a interligação entre a ferramenta Arduino e o supervisório, como é o caso de Annadate e Raj (2013) e Tung *et al.* (2013). Entretanto, a pesquisa que segue visa a análise de viabilidade da comunicação entre CLP, Arduino e supervisório através do mesmo protocolo de comunicação TCP/IP.

1.4. Objetivos

1.4.1. Objetivos Gerais

- Configurar a comunicação entre controladores lógico programáveis e outras plataformas através de software supervisório para automação e supervisão de dados.

1.4.2. Objetivos Específicos

- Conhecer os princípios de funcionamento dos controladores e softwares supervisórios;
- Programar e configurar o sistema supervisório;
- Programas e configurar os controladores;
- Realizar a comunicação entre os dispositivos e o supervisório;
- Observar os problemas de comunicação interdispositivos;
- Simular testes de comunicação das ferramentas em rede local;

1.5. Procedimentos Metodológicos

Segundo Silva e Menezes (2005), pesquisa aplicada possui como principal objetivo gerar conhecimentos para aplicação prática dirigida à solução de problemas específicos. Envolvendo sempre verdades e interesses locais.

Dessa forma, pode-se classificar o projeto que segue como uma pesquisa aplicada, pois se tem a evidência do problema, dificuldade de interligação entre dos dispositivos de automação e altos preços de mercados das ferramentas. E uma proposta de solução, através do uso de um sistema de supervisão e controle.

Como a pesquisa está sendo realizada em parceria com o curso de engenharia da computação da mesma instituição e campus, através de outro pesquisador, pode-se classifica-la também como uma pesquisa participante. Onde, segundo Gil (1991), a pesquisa participante desenvolve-se a partir da interação entre pesquisadores e membros das situações investigadas.

1.6. Estrutura do Trabalho

Nesse capítulo foi apresentado e delimitado o tema do estudo, foram apresentados os objetivos para o desenvolvimento do projeto e os procedimentos metodológicos que serão utilizados. Para o capítulo 2 foi reservada a revisão bibliográfica do tema, tratando desde a definição de automação industrial, redes industriais, controladores lógicos programáveis, a plataforma Arduino e o que são sistemas de supervisão e aquisição de dados. O capítulo 3 trata principalmente do método utilizado para a resolução da problemática proposta, com a configuração dos controladores, do sistema supervisório e a comunicação entre os dispositivos. O capítulo 4 é destinado à simulação do método em laboratório e apresentação dos resultados obtidos, nele é tratado, essencialmente, a comunicação entre os controladores. O estudo então é finalizado com o capítulo 5, que consolida os resultados obtidos e sugere pontos a serem trabalhos em pesquisas futuras.

2. REVISÃO BIBLIOGRÁFICA

2.1. Automação

A palavra *automation* foi inventada pelo marketing da indústria de equipamentos na década de 1960. Entende-se por automação como sendo qualquer sistema baseado em computadores, que substitua o trabalho humano em favor de segurança das pessoas, da qualidade dos produtos, da rapidez da produção ou da redução de custos, assim aperfeiçoando os complexos objetivos das industrias e dos serviços. (MORAES; CASTRUCCI, 2013)

A automação envolve a implantação de sistemas interligados e assistidos por redes de comunicação, compreendendo sistemas supervisórios e interfaces homem-máquina que possam auxiliar os operadores no exercício da supervisão e da análise dos problemas que porventura possam ocorrer.

Uma das vantagens em se utilizar sistemas que envolvam diretamente a informatização é a possibilidade da expansão utilizando recursos de fácil acesso, nesse contexto, são de extraordinária importância os controladores lógicos programáveis (CLPs), que tornam a automação industrial uma realidade onipresente.

O processo de controle e automação industrial pode ser dividido em dois ramos, como o próprio nome sugere. A engenharia de controla dinâmico pesquisa modelos matemáticos do processo e projeta as malhas de realimentação capazes de manter as condições operacionais nos valores eficientes a despeito das perturbações previsíveis no processo e na qualidade dos insumos. O outro ramo complementar a esse, seria a engenharia de automação, que prioriza o controle lógico, através de implementação das regras desejadas para dos eventos discretos do processo, ou seja, das manobras capazes de levá-los aos níveis da operação eficiente. Deve-se também considerar os níveis de segurança para os componentes e para as pessoas, assim como os requisitos de monitoração, alarme e intervenção por parte dos operadores e os relatórios gerenciais. A engenharia de automação deve selecionar os equipamentos de automação e de rede, programá-los, testá-los e acompanhar o desempenho no *start-up*. (MORAES; CASTRUCCI, 2013)

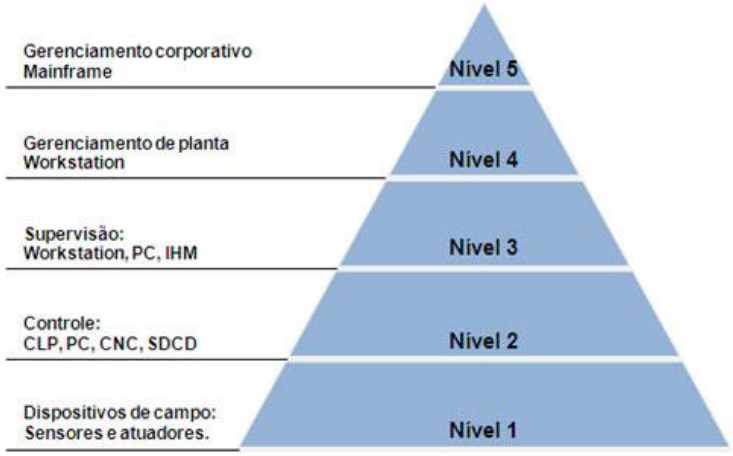
No processo de automação há uma combinação de dois tipos de controle, numa proporção infinitamente variável. O desafio maior da engenharia, no entanto, parece ser implementar com segurança todas as necessidades de controle lógico, dinâmico e

comunicação digital.

2.1.1. A Pirâmide de Automação

A automação industrial exige a realização de muitas funções. A Figura 1, chamada de Pirâmide de Automação, representa os níveis de automação encontrados em uma planta industrial.

Figura 1 - Representação da Pirâmide de Automação



Fonte: Reymaster.

Moraes e Castrucci (2013) definem cada um dos níveis da seguinte forma:

- **Nível 1 (Dispositivos de Campo):** Esse é o nível das máquinas, dispositivos e componentes (chão-de-fábrica)
- **Nível 2 (Controle):** Esse é o nível dos controladores digitais, dinâmicos, lógicos e de algum tipo de supervisão associada ao processo. Nesse nível se concentram as informações adquiridas no Nível 1 e as Interfaces Homem-Máquina (IHM).
- **Nível 3 (Supervisão):** Permite o controle do processo produtivo da planta; normalmente é constituído por bancos de dados com informações dos índices de qualidade da produção, relatórios e estatísticas do processo, índices de produtividade, algoritmos de otimização da operação produtiva
- **Nível 4 (Gerenciamento de Planta):** É o nível responsável pela programação e pelo planejamento da produção, realizando o controle e logística dos suprimentos.
- **Nível 5 (Gerenciamento Corporativo):** É o nível responsável pela administração dos recursos da empresa, em que se encontram os softwares para gestão de vendas e gestão financeira. É nesse nível onde se realizam a tomada de decisões e o gerenciamento de todo o sistema.

2.2. Redes Industriais

2.2.1. Histórico

Na década de 40, a instrumentação de processo nas fábricas era realizada através da monitoração de sinais físicos de pressão de 3 a 15psi. Até a década de 60, quando os sinais analógicos de 4 a 20mA foram introduzidos na indústria com a finalidade de monitorar dispositivos de campo. Com o desenvolvimento dos processadores nos anos 70, surgiu a ideia de se utilizar computadores para o controle e monitoramento dos processos produtivos. Através dos computadores, várias etapas do controle e automação poderiam ser feitas de forma diferente de modo a se adaptar mais precisamente a cada processo. (LUGLI; SANTOS, 2009)

Na década de 80, os primeiros sensores inteligentes começaram a ser desenvolvidos, assim como os controles digitais associados a esses equipamentos. Dessa forma, surgiu a necessidade de algo que pudesse interliga-los. Assim, nasce a ideia de criação de uma rede que interligasse todos os dispositivos e disponibilizasse os dados adquiridos e trabalhados por eles em um meio físico. Dentro desse contexto, 21 companhias e institutos uniram forças e criaram um projeto estratégico *fieldbus*. O objetivo era a realização e estabilização de um barramento de campo bitserial, sendo o requisito básico a padronização da interface de dispositivos de campo. Entre os grupos reunidos, a International Society of Automation (ISA), a International Electrotechnical Commission (IEC), o comitê de padronização do PROFIBUS (norma alemã) e o comitê de padronização do FIP (norma francesa), formaram o comitê internacional IEC/ISA SP50 Fieldbus. (LUGLI; SANTOS, 2009)

O desenvolvimento desse padrão internacional perdurou até o ano de 2000, onde todas as organizações interessadas convergiram para criar o fieldbus padrão IEC, denominado IEC 61158, com oito protocolos, listados a seguir:

- Tipo 1 – Foundation Fieldbus H1
- Tipo 2 – ControlNet
- Tipo 3 – PROFIBUS
- Tipo 4 – P-Net
- Tipo 5 – FOUNDATION Fieldbus HSE (*High Speed Ethernet*)
- Tipo 6 – Interbus
- Tipo 7 – SwiftNet

- Tipo 8 – WorldFIP

Mesmo com todos esses protocolos, não foi possível abranger todas as aplicações na indústria. Mais adiante, então, foi criada a IEC 61784, responsável por definir os chamados *profiles*, e foram corrigidas algumas especificações da IEC 61158. As correções realizadas estão consolidadas na tabela 2.1. (LUGLI; SANTOS, 2010)

Tabela 1 – Padrões e protocolos de acordo com a IEC 61784 e IEC 61158.

IEC 61784	IEC 61158		Nome de Mercado
	Meio Físico	Data Link Layer	
CPF - 1/1	TIPO 1	TIPO 1	FOUNDATION FIELDBUS (H1)
CPF - 1/2	ETHERNET	TCP/UDP/IP	FOUNDATION FIELDBUS (HSE)
CPF - 1/3	TIPO 1	TIPO 1	FOUNDATION FIELDBUS (H2)
CPF - 2/1	TIPO 2	TIPO 2	CONTROLNET
CPF - 2/2	ETHERNET	TCP/UDP/IP	ETHERNET/IP
CPF - 3/1	TIPO 3	TIPO 3	PROFIBUS-DP
CPF - 3/2	TIPO 1	TIPO 3	PROFIBUS-PA
CPF - 3/3	ETHERNET	TCP/UDP/IP	PROFINET
CPF - 4/1	TIPO 4	TIPO 4	P-NET RS-485
CPF - 4/1	TIPO 4	TIPO 4	P-NET RS-232
CPF - 5/1	TIPO 1	TIPO 7	WORLDFIP (MPS, MCS)
CPF - 5/2	TIPO 1	TIPO 7	WORLDFIP (MPS, MCS, SubMMS)
CPF - 5/3	TIPO 1	TIPO 7	WORLDFIP (MPS)
CPF - 6/1	TIPO 8	TIPO 8	INTERBUS
CPF - 6/2	TIPO 8	TIPO 8	INTERBUS TCP/IP
CPF - 6/3	TIPO 8	TIPO 8	INTERBUS SUBNET
CPF - 7/1	TIPO 6	TIPO 6	SWIFTNET TRANSPORT
CPF - 7/2	TIPO 6	TIPO 6	SWIFTNET FULL STACK

Fonte: Lugli e Santos (2010).

Como pode ser observado na tabela, vários padrões de protocolos de ethernet foram incluídos, bem como IP, TCP e UDP. Dentre os vários padrões de fieldbus utilizados no ambiente industrial, as redes DeviceNet, PROFIBUS, Interbus e Fieldbus Foundation são os mais utilizados, de acordo com a preferência e aplicação.

Atualmente, cada fabricante já tem sua solução para o ambiente industrial em Ethernet: o PROFINET, da associação PROFIBUS; o Ethernet/IP, da associação ODVA, onde IP significa *Industrial Protocol* e cuja proposta está baseada em uma evolução do Devicenet e Controlnet; e o HSE (High Speed Ethernet) da associação Fieldbus Foundation, que interconecta as redes H1 – Foundation Fieldbus). Com a existência de uma grande quantidade de soluções para a Industrial Ethernet, a interoperabilidade

buscada acabou ficando comprometida. (LUGLI; SANTOS, 2010)

Dessa forma, a Ethernet foi inserida no ambiente industrial, entretanto, alguns problemas começaram a surgir em sua fase inicial. Inicialmente, a Ethernet não foi considerada ideal para a indústria por ser probabilística e, dessa forma, não garantir a confiabilidade de transmissão dos dados. Entretanto, com o advento do switch industrial, o controle de colisões garantiu a transmissão efetiva das informações e assim a Ethernet teve uma real chance de agregar-se ao chão de fábrica. Hoje, a presença da Ethernet no ambiente industrial é um fato bastante concreto e facilmente observável. (DECOTIGNE, 2001)

2.2.2. Rede Ethernet

Uma rede existe quando é feita a interligação de computadores de forma local ou remota. Para fazer essa interligação, são necessários os componentes que formam a rede, tais como placas, cabos, conectores e outros aparelhos. Quando a interconexão é local, dizemos que nossa rede é uma LAN (Local Area Network). Quando é remota, nossa rede é conhecida como WAN (Wide Area Network). A rede LAN é formada por computadores interligados por meio de cabos, ondas de rádio ou infravermelho, em um mesmo local físico, dispensando a necessidade de modems. O conjunto de elementos que permitem a comunicação entre os computadores define o meio, o qual pode utilizar diversas tecnologias, tais como: Ethernet, Token Ring, Token Bus, FDDI (Fiber Distributed Data Interface) ou ATM. A tecnologia mais utilizada é conhecida por Ethernet. Isto acontece devido à sua simplicidade de instalação, seu baixo custo e, principalmente, em virtude dos investimentos realizados pela indústria nesta tecnologia, que a levou ao topo entre as concorrentes. O Ethernet é um canal físico por onde os dados podem fluir de um computador para outro. (MENDES, 2007)

2.3. Controlador Lógico Programável (CLP)

2.3.1. Histórico

Os primeiros sistemas de controle e automação eram mecânicos e foram desenvolvidos no final do século XIX, com o intuito de otimizar alguns processos de fabricação na época. Durante as primeiras décadas do século XX, com o avanço da

engenharia, os dispositivos mecânicos foram substituídos por relés, favorecendo o desenvolvimento de sistemas de controle mais complexos, sofisticados e de maior vida útil.

Logo após, com o surgimento dos circuitos integrados (CIs), vislumbrou-se uma nova geração de sistemas de controle através dos controladores lógico programáveis (CLPs). Através da tecnologia Transistor-Transistor Lógico (TTL) intrínseca dos CIs, obteve-se um tempo de processamento mais rápido, um ganho espacial e vida útil prolongada, quando comparado com os relés. (ROQUE, 2014)

Ainda para Roque (2014), o CLP pode ser definido como um equipamento eletrônico que utiliza uma memória programável para o armazenamento de instruções capazes de realizar funções específicas, como lógica, sequenciamento, registro, temporização, contadores e operações aritméticas para controlar, através de módulos de entrada/saída digitais ou analógicas, vários tipos de máquinas ou processos. Por ser um equipamento robusto, o CLP é perfeitamente adequado ao ambiente industrial. Possuindo resistência a ruídos, poeira, umidade e perturbações eletromagnéticas, além de apresentar pequenas dimensões e interface amigável com o usuário.

2.3.2. Arquitetura

Segundo Moraes e Castrucci (2013), um CLP é constituído basicamente de: fonte de alimentação, Unidade Central de Processamento (UCP), memórias dos tipos fixo e volátil, dispositivos de entrada e saída e terminal de programação.

A fonte de alimentação tem como função converter corrente alternada em contínua para alimentar o controlador. Em caso de suspensão de energia, há uma bateria que impede a perda do programa do usuário. Os tipos de fontes são:

- **Source:** Fonte de energia interna ao controlador.
- **Sink:** Fonte de energia externa ao controlador.

Seguindo, temos a UCP que é responsável pela execução do programa e pela atualização da memória de dados e da memória-imagem das entradas e saídas. A memória EPROM contém o programa monitor, desenvolvido pelo fabricante, responsável pela inicialização do controlador, armazenamento de alguns dados e gerenciamento de algumas sequências de operações. Esses dados não são acessíveis ao usuário. Dessa forma, a memória do usuário armazena o programa desenvolvido pelo próprio usuário. A UCP processa esse programa e atualiza a memória de dados internos e de imagem das entradas

e saídas. A memória possui dois estados distintos:

- **RUN:** Controlador em operação, realizando sempre varredura cíclica;

- **PROG:** Controlador parado, geralmente ocorre durante o carregamento do programa do usuário no dispositivo.

A memória de dados retém os dados referentes ao processamento do programa do usuário. Dessa forma, esses dados podem ser facilmente acessados.

A memória-Imagem de entradas e saídas reproduz o estado dos periféricos de entrada e saída. Onde, geralmente, os dispositivos de entrada são chaves, seletoras e limitadoras, por exemplo. E os dispositivos de saída são motores, solenoides, etc. E os estados de 0 ou 1 são configurados conforme o dispositivo.

2.3.3. Módulos de entrada e saída (E/S ou I/O)

Esses módulos diferem conforme a função (entrada ou saída), o método de acionamento e a forma da corrente trabalhada (alternada ou contínua). Existem ainda os módulos analógicos de entrada e saída, onde geralmente são conectados os dispositivos de sensoriamento. Dessa forma, conversores analógico-digitais (A/D) e digitais-analógicos (D/A) intercomunicam esses módulos. (MORAES; CASTRUCCI, 2013)

2.3.3.1. Módulos de Saída (S ou O) do Controlador

Para Moraes e Castrucci (2013), existem basicamente três métodos de acionamento desses módulos:

- **Saída a Relé**

Quando ativado o endereço da saída, um solenoide correspondente é atiçado, fechando o contato da saída do controlador.

A vantagem desse método está na sua imunidade a transientes de rede. Entretanto, devido ao mecanicismo do processo, nota-se uma vida útil do dispositivo em relação aos demais, permitindo um número total de acionamento entre 150.000 e 300.000, com capacidade de até 0,5A.

- **Saída a Triac**

Como o tipo sugere, o principal componente de acionamento é um triac e é geralmente utilizado quando a fonte de corrente é alternada. Possui vida útil bastante longa, possibilitando até 10 milhões de acionamentos de até 1A.

- **Saída a Transistor**

Nesse caso, o componente de acionamento pode ser um transistor comum ou um do tipo efeito de campo. Geralmente esse método é o mais utilizado e é utilizado quando a fonte de corrente é contínua. Possui vida útil bastante longa assim como o com saída a triac, discutido anteriormente.

2.3.3.2. Módulos de Entrada (E ou I) do Controlador

Os módulos de entrada dos controladores geralmente contêm optoisoladores em cada um dos acessos. Quando um circuito externo é acionado, um diodo emissor de luz sensibiliza o componente de base e o circuito interno da entrada correspondente é energizado. A vida útil desse módulo é bastante longa, com cerca de até 10 milhões de acionamentos de 100mA. (MORAES; CASTRUCCI, 2013)

2.3.4. Endereçamento

Os métodos de endereçamento de entrada e saída são realizados de formas bastante semelhantes. A seguir temos um padrão de estrutura comentado com um exemplo.

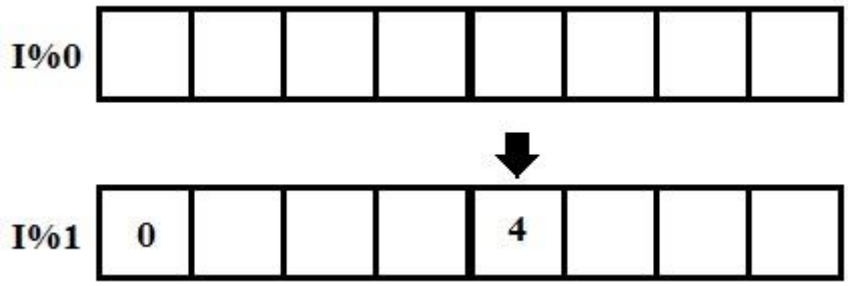
A%B.C

A: Tipo de acesso. Geralmente usa-se I para entrada e O para saída.
B: Localização do respectivo bloco de entrada ou saída na memória do controlador;
C: Endereço do bit da imagem da palavra de entrada ou saída;

Ex:.

Para o endereço **I%1.4**, temos:

Figura 2 – Representação do endereço de memória



Fonte: Elaborado pelo autor.

2.3.5. Terminal de Programação

Consiste em um periférico conectado ao controlador cujas principais funções são: programação e edição de instruções de forma online ou off-line, monitoração dos processos desenvolvidos, diagnóstico de falhas e problemas, transferência do software aplicativo e gerenciamento de memória. Esse dispositivo pode ser um computador (PC), um dispositivo portátil ou uma alguma interface homem máquina (IHM).

2.3.6. Ciclo de Execução (Scan) em Operação Normal (Modo RUN)

Em um ciclo normal de execução, o controlador realiza as seguintes ações de forma cíclica e recorrente:

- Atualiza as entradas (*scan* das entradas)
- Processa as instruções do software aplicativo (*scan* do programa)
- Atualiza as saídas (*scan* das saídas)

2.3.7. Terminais Remotos de Entrada e de Saída

Devido a limitação da quantidade de dispositivos que podem ser ligados diretamente aos blocos de entrada e saída ou devido alguma impossibilidade espacial, torna-se imperativo a criação de uma rede remota para a intercomunicação dos dispositivos periféricos ao controlador.

Dessa forma, o controlador atualiza os dados de entrada e saída locais em

sincronismo com a varredura do software. Entretanto, a atualização dos dados de entrada e saída remotos e o software é realizada de forma assíncrona. (MORAES; CASTRUCCI, 2013)

2.3.8. Especificações de controladores lógico programáveis

Para a automação de um controlador lógico programável, deve-se observar alguns pontos:

- Compatibilidade entre a instalação elétrica externa e os pontos de entrada e saída do controlador;
- Compatibilidade dos equipamentos eletromecânicos;
- Existência de chaves de proteção para os dispositivos;
- Tipo e forma de endereçamento e sinais;

2.3.9. CLP Simatic S7 – 1200

Segundo Siemens (2012) as principais características e vantagens do SIMATIC S7-1200 são observadas na versatilidade que o produto oferece, na possibilidade de modularização e no tamanho reduzido de seus componentes. Oferece interfaces de comunicação que atendem o mais alto padrão de exigência na indústria e possuem poderosas funções tecnológicas que os tornam parte integrante da solução mais completa para o processo de automação.

A linha SIMATIC S7-1200 possui quatro diferentes modelos de CLP: 1211C, 1212C, 1214C e 1215C, podendo ser expandido de forma a atender aos requisitos de sua aplicação. Apresenta interface PROFINET e poderosas funções tecnológicas integradas em um projeto flexível. Isso permite uma comunicação simples e soluções eficientes para exigências tecnológicas, que são perfeitamente adequadas às necessidades de automação em uma ampla variedade de aplicações.

Por se tratar de um controlador IO, o SIMATIC S7-1200 permite uma conexão completa com dispositivos PROFINET IO. Além disso, a interface integrada PROFINET possibilita conexão com SIMATIC STEP 7 Basic para planejamento de projeto, programação e visualização dos SIMATIC IHM Basic Panels, comunicação entre CLPs e dispositivos de terceiros como opção de integração avançada. A conexão standard fieldbus PROFIBUS para reações rápidas, por exemplo, é possível graças aos módulos

de comunicação PROFIBUS. Para comunicação remota via internet wireless o S7-1200 pode ser equipado com interface GSM/GPRS. Além disso, atuadores e sensores AS-i do nível mais baixo, de campo, podem ser conectados por meio do módulo de comunicação mestre AS-i.

2.3.9.1. Características gerais da CPU

O controlador escolhido para a pesquisa foi o 1214C modelo AC/DC/Relé, devido maior disponibilidade nos laboratórios. Seguem as características do controlador supracitado. A Figura 3 ilustra perfeitamente o controlador em questão.

- Memória de Trabalho Integrada de 75KB
- Memória de Carga Integrada de 4MB
- Memória Retentiva Integrada de 10KB
- Cartão de Memória Adicional SIMATIC (Não utilizado)
- 14 Entradas e 10 Saídas Digitais Integradas
- 2 Entradas Analógicas Integradas
- Imagem de Processo de 1024 Bytes por Entrada e Saída
- Real Time Clock (RTC) de 20 dias
- Tempo de Instrução de 85ns

Figura 3 – CLP Siemens S7 1214c AC/DC/RLY

Fonte: SIEMENS. Controlador Simatic. Simatic s7 - 1200.

2.3.9.2. Expansões de I/O

O controlador também conta com a possibilidade de expansão de entrada/saída (I/O) de 4 blocos de 200kHz ou 1 analógico, diretamente na CPU. A quantidade máxima de blocos de entrada/saída digitais são 284 e analógicos são de 67. A é realizada através de sensores PT100 e RTD.

- HSP Single Phase: 3 @ 100kHz e 3 @ 30kHz
- HSP Quadrature Phase: 3 @ 80kHz e 3 @ 30kHz
- Saída em Pulso: 4 @ 100kHz (Saída DC) e 4 @ 1Hz (Saída Relé)
- 14 Entradas em Pulso
- Leitura de Temperatura pela Entrada Analógica: PT100 e RTD

2.3.9.3. Comunicação e conectividade

O controlador possui Módulo de Expansão de Comunicação de até 3 módulos (RS485 e/ou RS232), comunicação ethernet através de 1 interface RJ45 10/100 Mbits/s com até 16 conexões (11 entre CPUs, 4 com HMIs e 1 Field PG), protocolos USS, Modbus TCP/IP Mestre, Modbus RTU Mestre/Escravo e ASC II.

2.3.9.4. Aplicações

No quesito aplicações, o modelo conta com controle de até 16 malhas PID (com Autotunning), Comunicação GPRS/SMS e OPC, Leitura de Tags RFID e Data Matrix.

2.4. Arduino

Segundo os desenvolvedores, o Arduino é uma plataforma de prototipagem de código aberto, criado pelo *Ivrea Interaction Design Institute* com o objetivo de oferecer uma ferramenta simples para rápida prototipagem, visando estudantes que não possuem experiência com eletrônica e programação. Logo que as placas ganharam reconhecimento pela comunidade em geral, iniciou-se um processo de mudança com o intuito de que as novas placas se adaptassem às novas necessidade e desafios.

As placas Arduino são aptas a lerem entradas diversas, e transformá-la em uma determinada saída. Pode-se configurar também a placa para realizar determinada ação

através do envio de um conjunto de informações ao microcontrolador. A plataforma utiliza uma linguagem de programação própria, baseada em *Wiring*. Conta também com um ambiente de desenvolvimento integrado (IDE), baseado em *Processing*.

Todas as placas Arduino e sua IDE possuem código aberto, possibilitando que os usuários possam realizar seus projetos de forma independente e eventualmente adaptá-los as suas necessidades específicas.

Para McRoberts (2010), o Arduino é um sistema embarcado, ou seja, que pode interagir com seu ambiente por hardware e software incorporados a um dispositivo com um objetivo pré-definido. A plataforma também pode ter suas aplicações estendidas utilizando placas que desempenham funções complementares, as quais são facilmente conectadas. Estas placas são chamadas de módulos ou *Shields* (escudos, em inglês). Dentre as funções desempenhadas, as principais são: receptores GPS, módulos de rede ethernet ou wireless, dentre outros.

Entre os diversos modelos de placa da plataforma Arduino, cada uma com dimensões e especificações únicas, uma das mais comuns e acessíveis é a versão UNO. A Figura 4 apresenta o dispositivo Arduino UNO.

Figura 4 – Arduino UNO

Fonte: ARDUINO. Products. Arduino Uno.

2.4.1. Arduino Uno

Segundo os fabricantes, o Arduino Uno é uma placa de microcontrolador baseada no ATmega328P. A placa dispõe de 14 pinos digitais de entrada/saída (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal de quartzo 16 MHz, uma conexão USB, uma fonte de alimentação, um ICSP header e um botão de reset. Ainda segundo os criadores, a placa possui todos os elementos necessários para o perfeito funcionamento do microcontrolador. Basta conectá-lo a um computador com um cabo USB, ligá-lo com um adaptador AC-DC ou utilizar uma bateria.

Em italiano, país do principal idealizador, "Uno" significa "Um" e esse nome foi escolhido para o lançamento do software (IDE) e do hardware em sua versão 1.0. Essas versões serviram de referência para as posteriores placas. A versão Uno também foi a primeira de uma série de placas que utilizavam o USB (Universal Serial Bus) como principal forma de conexão.

2.4.1.1. Especificações Técnicas

Segundo os fabricantes, a versão Uno do Arduino possui as seguintes especificações técnicas.

- Microcontrolador ATmega328P
- Tensão operacional de 5V
- Tensão de entrada (recomendado) de 7-12V
- Tensão de entrada (limite) de 6-20V
- 14 entradas/saídas digitais
- 6 entradas/saídas com PWM (Pulse Width Modulation)
- 6 entradas analógicas
- Corrente DC por entrada de 20 mA
- Corrente da entrada de 3.3V de 50 mA
- Memória Flash de 32 KB
- SRAM de 2 KB
- EEPROM de 1 KB
- Velocidade de clock de 16 MHz
- Comprimento de 68,6 milímetros
- Largura de 53,4 mm

- Peso de 25 g

2.4.1.2. Programação

O ATmega328 nessa versão vem pré-programado com um bootloader que permite o envio de novos códigos sem o uso de um programador de hardware externo. O protocolo de comunicação utilizado é o STK500. Há também a possibilidade de programar o microcontrolador através do ICSP header (Serial Programming In-Circuit) usando Arduino ISP ou similar.

A versão Uno possui um polifusível reajustável responsável por proteger as portas USB do computador conectado de curtos e sobrecorrente. Embora a maioria dos computadores possuírem sua própria proteção interna, o fusível fornece uma camada extra de proteção. Caso a corrente fornecida à porta supere 500 mA, o fusível encerra automaticamente a ligação até que a sobrecarga ou o curto seja removido.

2.4.1.3. Diferenças das outras

Essa versão difere de todas as placas anteriores que não utilizam o *driver* FTDI USB para serial. Em vez disso, ele apresenta o Atmega16U2 (Atmega8U2 até a versão R2) programado como um conversor USB para serial.

2.4.1.4. Alimentação

A alimentação pode ser realizada através da ligação USB ou com uma fonte de alimentação externa. Quando utilizada, a fonte de alimentação é selecionada automaticamente.

A fonte externa de energia pode vir com um adaptador AC-DC ou bateria. O adaptador pode ser conectado através de um plug de 2.1 milímetros de centro-positivo na entrada de alimentação da placa. Os conectores de uma bateria podem ser inseridos nas entradas e GND e Vin para alimentação.

A placa pode operar com um fornecimento externo de 6 a 20 volts. Se a tensão fornecida for inferior a 7V, o pino de 5V pode fornecer uma tensão inferior 5V e a placa pode ficar instável. Caso a alimentação exceda 12V, o regulador de tensão pode superaquecer e danificar a placa. O intervalo recomendado é de 7 a 12 volts.

Seguem os pinos de alimentação e suas respectivas funções:

- Vin: Entrada de tensão quando é utilizada uma fonte de alimentação externa. Pode-se fornecer tensão por este pino ou realizar o fornecimento de tensão através da tomada de energia.
- 5V: Esse pino fornece 5V em corrente contínua através do regulador da placa. As formas de alimentação podem ser: a partir da tomada de energia DC (7-12V), do conector USB (5V) ou através do pino Vin da placa (7-12V). Fornecimento de tensão através dos pinos 5V ou 3.3V ignoram o regulador e pode danificar a placa.
- 3V3: Fornece 3.3V através do regulador da placa. Com corrente máxima de 50 mA.
- GND: Pinos de terra.
- IOREF: A esse pino deve ser fornecida a tensão de referência com a qual o microcontrolador deverá operar. Um shield configurado corretamente pode ler a tensão do pino IOREF e selecionar a fonte de alimentação adequada ou habilitar os conversores nas saídas para operar com a 5V ou 3.3V.

2.4.1.5. Memória

O ATmega328 tem 32 KB (com 0,5 KB ocupado pelo inicializador). Ele também tem 2 KB de SRAM e 1 KB de EEPROM.

2.4.1.6. Entradas e Saídas

Cada um dos 14 pinos digitais do Uno pode ser usado como uma entrada ou saída, usando as funções pinMode(), digitalWrite() ou digitalRead(). Cada um dos pinos opera com 5 volts e podem fornecer ou receber 20mA. Possuem também um resistor pull-up interno (desconectado por padrão) de 20-50k ohm. Deve-se evitar ultrapassar 40 mA em qualquer um dos pinos de entrada ou saída sob condição de causar danos permanentes ao microcontrolador.

Além disso, alguns pinos possuem funções especializadas, que seguem:

- Serial: Pinos 0 (RX) e 1 (TX). Utilizado para receber (RX) e transmitir dados seriais (TX) TTL.
- Interrupções externas: Pinos 2 e 3. Estes pinos podem ser configurados para disparar uma interrupção por um valor baixo, uma borda de subida, descida ou uma mudança de valor. Geralmente a função attachInterrupt() é a mais utilizada.

- PWM: Pinos 3, 5, 6, 9, 10, e 11. Fornecem saída PWM de 8 bits através da função analogWrite().
- SPI: Pinos 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estes pinos suportam comunicação SPI utilizando a biblioteca apropriada.
- LED: Pino 13. Enquanto o pino é alimentado com nível alto de tensão, o LED permanecerá ligado, quando o nível de tensão retorna para baixo, o LED desligará.
- TWI: Pino A4 ou SDA e A5 ou pino SCL. Suporta comunicação TWI utilizando a biblioteca Wire.

Além disso, a versão conta com 6 entradas analógicas, rotuladas de A0 a A5, cada uma com 10 bits de resolução (i.e. 1024 valores diferentes). Por padrão elas dividem os valores entre 0 e 5V, embora seja possível mudar a extremidade superior da área de distribuição utilizando o pino AREF e a função analogReference().

Há um par de outros pinos na placa:

- AREF. Tensão de referência para as entradas analógicas. Usado com a função analogReference().
- Reset. Esse pino é utilizado para resetar o microcontrolador, através de um sinal de nível baixo.

2.4.1.7. Comunicação

O Arduino Uno possui uma série de facilidades para se comunicar com um computador, uma outra placa Uno ou outros microcontroladores. O ATmega328 fornece UART TTL (5V) de comunicação serial, que está disponível nos pinos digitais 0 (RX) e 1 (TX). No ATmega16U2 esta comunicação serial é realizada através de USB e aparece como uma porta COM virtual para o software no computador. O firmware 16U2 usa os drivers USB COM como padrão e nenhum driver externo é necessário. No entanto, no Windows, um arquivo .inf é necessário. O Software Arduino (IDE) inclui um monitor serial que permite que dados simples de texto sejam enviados para a placa. Os LEDs RX e TX na placa piscam quando os dados estão sendo transmitidos via USB serial e quando a entrada USB está conectada ao computador (mas não para comunicação serial nos pinos 0 e 1).

O ATmega328 também suporta I2C (TWI) e comunicação SPI. O Software Arduino (IDE) inclui uma biblioteca Wire para simplificar o uso do barramento I2C.

2.4.1.8. Reset Automático

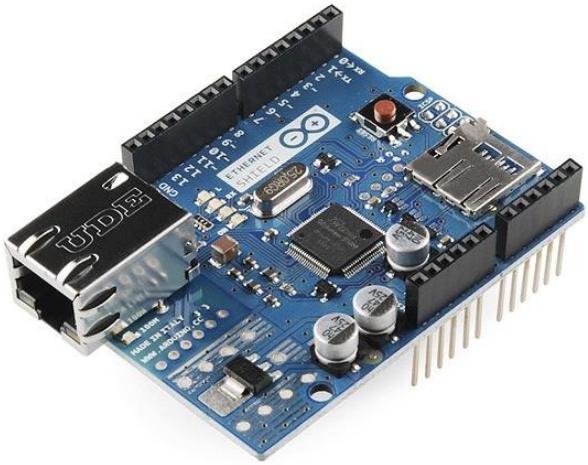
A versão Uno também permite o reset através do software, enquanto a placa estiver ligada ao computador. Uma das linhas de controle de fluxo do hardware (DTR) está ligada ao reset através de um capacitor 100nF. Quando esta linha é acionada, a placa reinicia por um curto espaço de tempo. O Software Arduino (IDE) usa esse recurso para permitir o upload do código, simplesmente pressionando o botão de upload na barra de ferramentas de interface.

Outra característica do Uno é que quando ele é conectado a um computador rodando Mac OS X ou Linux, a placa é reiniciada cada vez que uma nova conexão é feita com o software (via USB). Isso sem que haja perda no desempenho. As funções listadas também podem ser desabilitadas através de algumas mudanças na própria placa.

2.4.2. Arduino Ethernet Shield

O *Arduino Ethernet Shield* é basicamente um módulo que permite que uma placa Arduino se conecte à internet. Baseia-se no chip Ethernet Wiznet W5100, que fornece uma rede (IP) capaz de se conectar utilizando tanto o protocolo TCP como o UDP. Ele suporta até quatro conexões de soquete simultâneas. O *shield* se conecta a uma placa Arduino utilizando conectores que se se encaixam perfeitamente à "placa mãe". Dessa forma, o *shield* permanece estável e permite a ligação de outros módulos na forma de pilha. A Figura 5 apresenta perfeitamente o dispositivo.

Figura 5 – Arduino Ethernet Shield



Fonte: ARDUINO. Products. Arduino Ethernet Shield.

O *shield* possui um controlador de *reset*, que garante que o módulo Ethernet W5100 seja devidamente reiniciado ao ligar. O botão de *reset* na placa reinicia tanto o W5100 como a placa Arduino.

O módulo possui uma conexão RJ-45 padrão, com um transformador de linha integrada e possibilidade de uso de PoE (Power over Ethernet). O módulo PoE não acompanha o *shield ethernet* e deve ser adquirido separadamente.

Há também um slot para um cartão micro-SD integrado, que pode ser utilizado para armazenar arquivos de rede. É compatível com todas as placas Arduino e Genuino. O leitor de cartão micro SD é acessível através da Biblioteca SD. O Arduino comunica-se tanto com o W5100 como com o cartão SD utilizando o barramento SPI (através do ICSP header). O pino 10 é usado para selecionar o W5100 e o pino 4 para o cartão SD. Estes pinos não podem ser usados como entradas ou saídas. Como o W5100 e o cartão SD utilizam o mesmo barramento SPI, apenas um deles poderá estar ativo.

O *shield* conta também com uma série de LEDs informativos que indicam, respectivamente:

- **PWR:** indica que a placa e o *shield* estão energizados.
- **LINK:** indica a presença de uma conexão de rede e pisca quando o *shield* transmite ou recebe dados.
- **FULLD:** indica que a conexão de rede é de duplo sentido.
- **100M:** indica a presença de uma ligação de rede 100 Mbits/s
- **RX:** pisca quando o *shield* recebe dados.
- **TX:** pisca quando o *shield* envia dados.
- **COLL:** pisca quando são detectadas colisões na rede.

2.5. Sistemas Supervisórios

Para Seixas Filho (1999), sistemas supervisórios são sistemas desenvolvidos com o intuito de estabelecer maior interatividade entre os operários e os processos produtivos, sejam eles em estações de supervisão local ou estações concentradoras de dados em processos distribuídos. Estes sistemas são baseados em microcomputadores interligados a controladores programáveis, estações remotas ou outros equipamentos de aquisição de dados.

Os sistemas supervisórios também são conhecidos como SCADA (Supervisory Control and Data Aquisition). Daneels e Salter (2000) explicam que os softwares

supervisórios permitem que sejam monitoradas e gerenciadas as informações de um processo produtivo ou uma instalação física. Tais informações são coletadas através de equipamentos de aquisição de dados, e são manipuladas, analisadas, decompostas, armazenadas e exibidas ao usuário através de uma interface.

Pires et al. (2005) defende que os sistemas SCADA possuem grande importância estratégica já que estão presentes na maioria das indústrias responsáveis pelo desenvolvimento tecnológico e econômico de um país. As aplicações da tecnologia SCADA alcançam praticamente todo o espectro do setor produtivo e são utilizadas principalmente: na indústria química, petroquímica, de cimentos, na indústria alimentícia, na produção e distribuição de energia elétrica, na distribuição de água, no controle de oleodutos, gasodutos, centrais nucleares, edifícios inteligentes e tráfego.

No início os sistemas supervisórios possuíam arquiteturas centralizadas, fechadas, com conectividade externa limitada e utilizavam hardware e software próprios. Os dados eram visualizados através de um painel de lâmpadas e indicadores, sem qualquer interface operacional. (SILVA e SALVADOR, 2005).

Existem atualmente no mercado vários softwares supervisórios disponíveis, dentre os quais, observamos alguns: Axeda Supervisor e Wizcon, da Axeda Systems Inc; iFIX, da Intellution; Elipse SCADA e Elipse E3, da Elipse; Indusoft, da Indusoft e InTouch, da Wonderware.

2.5.1. Elipse SCADA

Segundo o fabricante, o software Elipse em sua versão SCADA permite, através da coleta de informações de qualquer tipo de equipamento, monitoramento e controle de máquinas e processos no chão de fábrica e gerenciamento da produção. Os dados são apresentados de forma gráfica e em tempo real, permitindo o tratamento dessas informações de diversas maneiras, como: armazenamento histórico, geração de relatórios, conexão remota, entre outras possibilidades.

2.5.1.1. Benefícios

- **Fácil configuração**

O Elipse SCADA conta com o exclusivo Organizer (Árvore do Aplicativo), uma maneira muito simples e fácil para criar, organizar e documentar aplicativos. O usuário

acessa todos os elementos do sistema e suas propriedades, navegando em uma árvore hierárquica que fornece uma visão geral do aplicativo de modo a "organizar" naturalmente o trabalho de configuração e documentação. O Elipse SCADA também permite a edição, através da ferramenta de configuração on-line onde se pode alterar o aplicativo sem a necessidade de interromper a execução.

- **Interface gráfica mais clara, lógica e intuitiva**

A criação da interface com o usuário é feita de maneira simples e rápida. Vários recursos visuais estão disponíveis como animações, displays, botões, gráficos e outros ligados diretamente com as variáveis de campo (tags). Além disso, o Elipse SCADA conta com uma extensa biblioteca gráfica de desenhos e possibilita importar imagens de vários formatos de modo a facilitar a criação de telas. A operação do sistema pode ser feita via mouse, teclado ou touchscreen.

- **Conectividade com qualquer equipamento**

Existem várias maneiras para trocar informações com equipamentos de aquisição de dados como PLCs (Controladores Lógico Programáveis), DACs (Cartões de Aquisição de Dados), RTUs (Unidades Remotas), controles e outros tipos de equipamentos.

2.5.1.2. Recursos

- **Aplicações Remotas**

O Elipse SCADA permite o acesso remoto a outras aplicações via rede. O método utilizado baseia-se no conceito de Aplicações Remotas, onde os dados de uma aplicação qualquer (Servidor) são acessados por um cliente que poderá realizar a leitura e escrita de qualquer parâmetro.

- **Cross-Reference**

A ferramenta de referência cruzada permite que, em qualquer momento da configuração, o usuário visualize todos os pontos onde um determinado Tag ou objeto está sendo referenciado.

- **Históricos**

Os históricos são estruturas responsáveis pelo registro dos dados do processo monitorado para sua posterior análise. Podem ser utilizados para processos contínuos ou bateladas, armazenando dados em intervalos de tempo fixos ou por eventos. A ferramenta de análise histórica pode ser utilizada para uma visualização mais sofisticada dos dados, permitindo zoom e filtros de dados.

- **Alarmes**

O Elipse SCADA permite associar diferentes tarefas a alarmes. O usuário pode, em tempo de execução, alterar limites, habilitar, desabilitar e modificar mensagens de alarmes. O objeto de visualização permite monitorar em tempo real. Há também a possibilidade de classificar os alarmes por grupos, gravando os dados em arquivos separados. Adicionalmente, é possível enviar traps a um cliente SNMP (Simple Network Management Protocol) na ocorrência de um alarme.

- **Bancos de Dados**

A integração com qualquer base de dados é muito simples através de recursos ODBC (Open DataBase Connectivity) e DAO (Data Access Objects). Wizards auxiliam no processo de conexão ou criação de uma base de dados qualquer, entre elas SQL Server®, Access®, Oracle® e DBase®. Ainda é possível realizar a integração com sistemas corporativos (ERP) como o SAP.

- **Relatórios**

No mesmo ambiente de desenvolvimento, é possível criar qualquer tipo de relatório seja em modo gráfico, texto ou formatado pelo usuário aplicando-se consultas ou filtros específicos. Os relatórios podem ser impressos automaticamente, utilizando dados históricos e/ou de tempo real. O Elipse SCADA possui um exclusivo sistema de alias de impressoras, permitindo a impressão dos dados em dispositivos diferentes.

- **Scripts**

O Elipse SCADA possui uma linguagem de programação própria - Elipse Basic - que permite definir lógicas ou criar sequências de procedimentos semelhantes ao Visual Basic. Os scripts são orientados por eventos, como o pressionar de uma tecla, pela mudança de uma variável ou ainda por um intervalo regular de tempo.

- **Ferramentas de depuração**

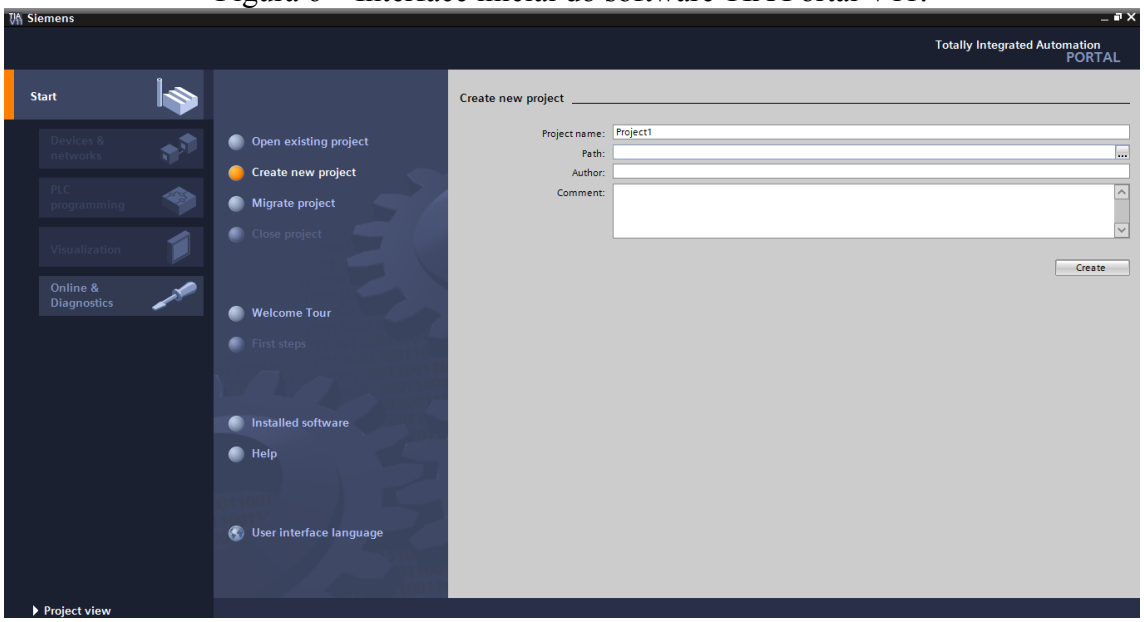
Para facilitar o processo de depuração durante a criação de aplicativos, existem ferramentas como o WatchWindow e ScriptWindow que informam sobre o estado dos processos ou de qualquer componente do sistema.

3. CONFIGURAÇÃO DAS FERRAMENTAS E DISPOSITIVOS

3.1. Configuração do CLP

Como proposto, será utilizado o CLP da Siemens S7 1214C AC/DC/RLY como controlador. A configuração e programação desse dispositivo é realizada através do software TIA PORTAL V11, disponibilizado pela própria Siemens. A interface inicial do programa é apresentada pela Figura 6.

Figura 6 – Interface inicial do software TIA Portal V11.

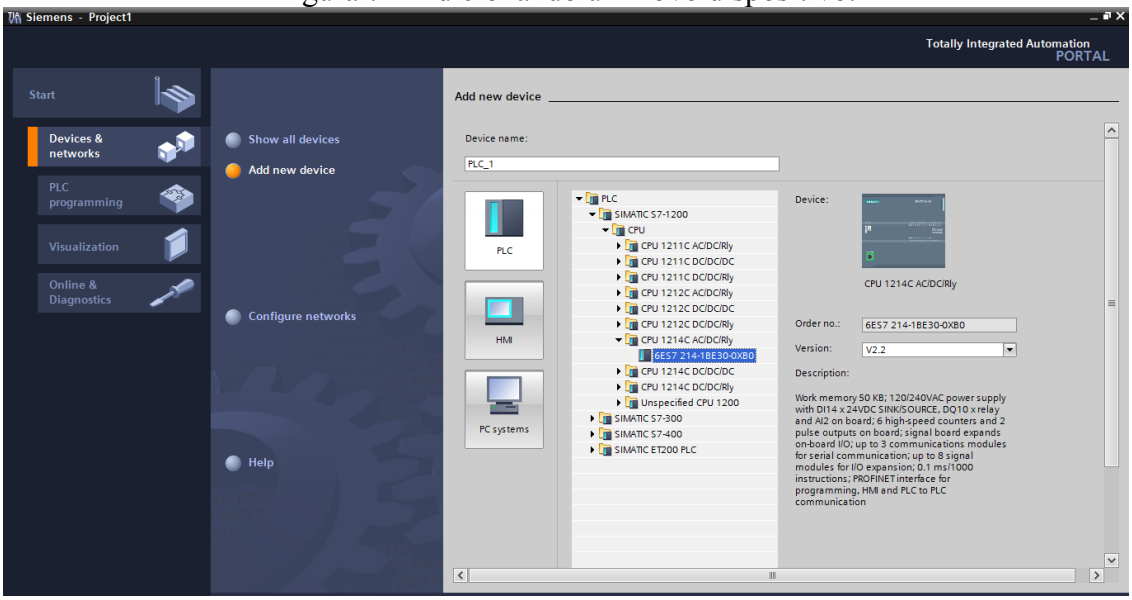


Fonte: Capturado pelo autor.

Para a criação de um novo projeto, podemos nomeá-lo em **Project name** e definir o diretório onde o projeto estará hospedado através da informação **Path**. Os dados **Author** e **Comments** são opcionais. Após definir os dados, clica-se no botão **Create**.

Logo após, deve-se configurar o dispositivo controlador utilizado. Acessa-se então a opção **Configure a device**.

Figura 7 – Adicionando um novo dispositivo.

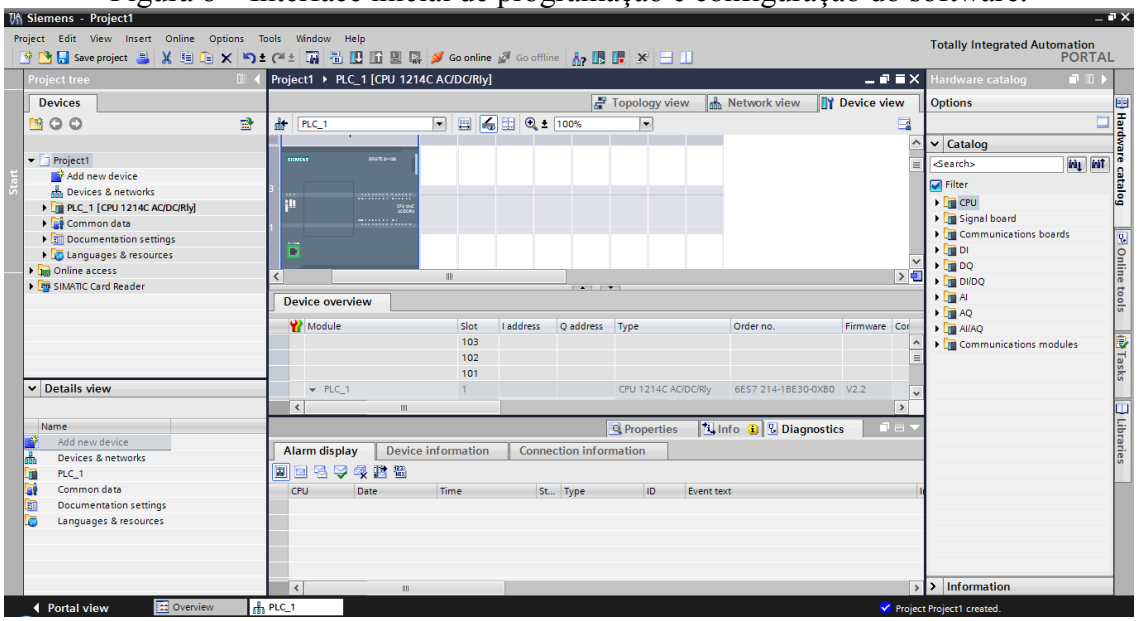


Fonte: Capturado pelo autor.

Em **Add new device**, seleciona-se a opção **PLC**, o modelo, a versão do firmware em **Version** e clica-se no botão **Add**. O nome do dispositivo pode ser alterado através da informação **Device name**, entretanto, o software irá gerar um nome caso não seja informado. Nesse estudo, utilizou-se a CPU 1214C AC/DC/Rly, como mostrado na Figura 7.

Já na área de programação do dispositivo, apresentada pela Figura 8, deverá ser configurado o endereço de IP do controlador.

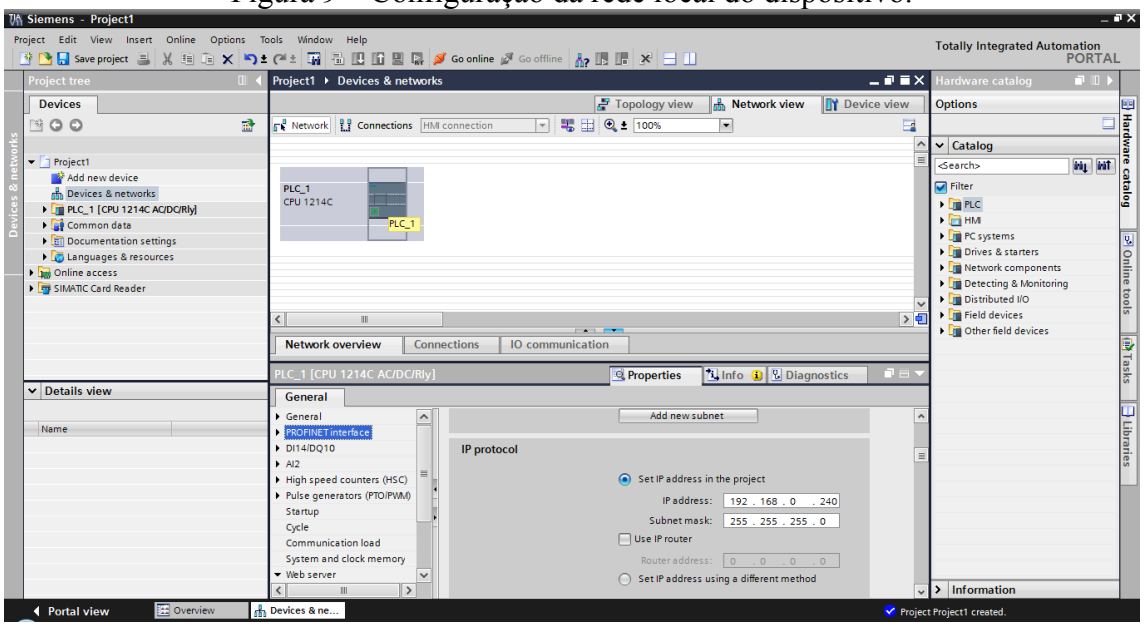
Figura 8 – Interface inicial de programação e configuração do software.



Fonte: Capturado pelo autor.

Deve-se então acessar a opção **Devices & networks** no menu do lado esquerdo. A tela de configuração de rede será apresentada pela figura a seguir.

Figura 9 – Configuração da rede local do dispositivo.

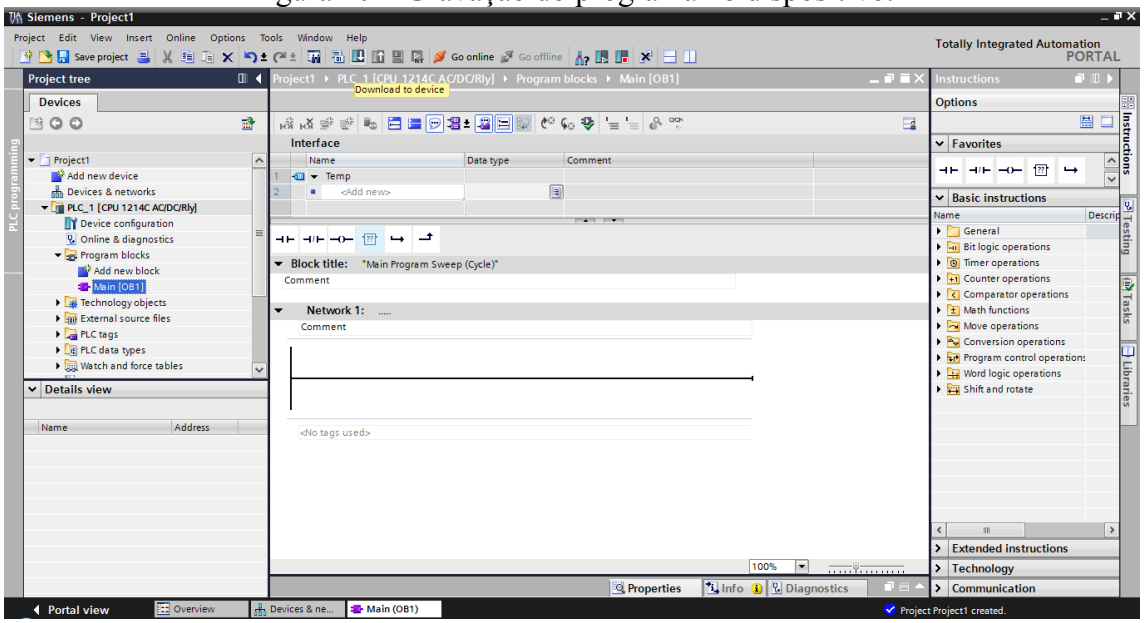


Fonte: Capturado pelo autor.

Após acessar **Devices & networks**, seleciona-se o controlador e **Properties** em **Network overview**. Na opção **PROFINET interface**, deve-se buscar a seção **Ethernet addresses** e **IP protocol**. Em **IP address**, deve-se informar o endereço de IP específico do controlador. No caso desse estudo, o endereço de IP utilizado foi 192.168.0.240 para o controlador PLC_1.

Assim, no menu à esquerda, expande-se o item **PLC_1**, em seguida **Program blocks** e seleciona-se **Main [OB1]**. A figura a seguir mostra onde o item **Main** pode ser encontrado.

Figura 10 – Gravação do programa no dispositivo.

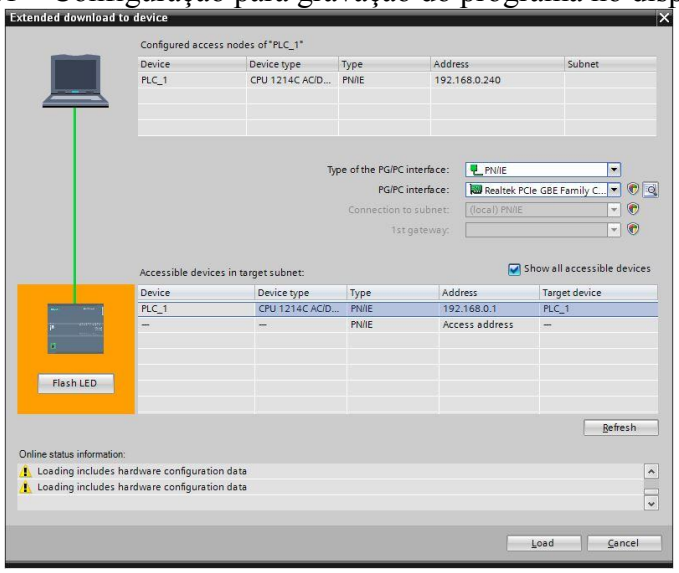


Fonte: Capturado pelo autor.

Na barra de ferramentas superior deverá existir o botão **Download to device**. Essa função grava o programa no controlador. Para isso, o computador e o controlador devem estar comunicados através de um cabo de rede RJ45 e o endereço de IP do computador deverá estar estático.

A tela de configuração de gravação do programa no controlador, aberta através da função **Download do device**, é apresentada pela Figura 11.

Figura 11 – Configuração para gravação do programa no dispositivo.



Fonte: Capturado pelo autor.

Na seção **Configure access nodes of PLC_1**, deverão ser mostrados os dispositivos configurados que serão gravados. Para a configuração da comunicação, deverá ser selecionado **PN/IE** em **Type of the PG/PC interface** e a placa de rede do computador deverá ser selecionada em **PG/PC interface**. Assim, os dispositivos elegíveis para gravação deverão ser mostrados na seção **Acessible devices in target subnet**, logo abaixo. Em alguns casos, o dispositivo conectado não aparece. Isso pode ser resolvido marcando a opção **Show all acessible devices**. Após selecionar o controlador, deve-se clicar em **Load**. Após a verificação do código e das configurações estabelecidas, uma janela de confirmação (**Load preview**) irá surgir e o botão **Load** deverá ser novamente acionado. O processo de gravação irá iniciar e ao final, uma janela apresentando o resultado da gravação será exibida (**Load results**) e finalmente o botão **Finish** irá finalizar o processo e iniciar os módulos.

Para a configuração de um novo CLP, o dispositivo deverá ser adicionado, configurado e gravado assim como descrito anteriormente para o PLC_1.

3.2. Configuração do Sistema Supervisório

Como proposto, o estudo utilizou a ferramenta Elipse SCADA como supervisório dos processos controlados pelo Arduino e pelo CLP. Além disso, como forma de comunicação entre ambos através de scripts programados dentro do próprio software supervisório. Dessa forma, facilita-se bastante a comunicação entre os dispositivos.

Inicialmente foi realizada a comunicação entre dois CLPs Siemens S7-1200 CPU 1214C AC/DC/RLY através de uma única tela de supervisório, apresentada a seguir.

Figura 12 – Programa supervisório para a comunicação entre CLPs.

Fonte: Capturado pelo autor.

Os dois controladores foram ilustrados e nomeados respectivamente como PLC_1 e PLC_2. Para ilustrar as entradas e saídas de cada um dos supervisórios foram utilizados 9 botões com animações do tipo liga/desliga. Três botões foram reservados para as entradas do PLC_1 (DI a.0, a.1 e a.2) e outros três botões ficaram reservados para o PLC_2, com o mesmo nome de referência. Para as saídas, também foram reservados três botões (DQ a.0, a.1 e a.2) para cada um dos controladores. Foi utilizada essa quantidade de botões devido à limitação de *tags* característica da versão de demonstração do software. Como animação, utilizou-se um artifício que ilustra bem o sinal aplicado ou recebido em cada um dos botões. Se o botão estiver desligado, ele permanecerá com a cor preta. Caso o botão seja ligado, sua cor mudará para verde, informando assim que tipo de sinal está sendo aplicado na entrada/saída.

A comunicação entre os controladores e o supervisório foi realizada através do driver *"Driver SIEMENS MProt (MPI/PPI/ISO-TCP) v3.1.1 (IOKitLib v2.0.37)"*, disponível para download no site do Elipse SCADA.

3.2.1. Configuração do driver Mprot

A configuração do driver é realizada pelo usuário no próprio software supervisório com o auxílio de um manual que acompanha o driver (SIEMENS e VIPA, 2014). O manual possui as versões inglês e português-brasileiro.

Para a configuração do driver, deve-se acessar primeiro o "Organizer" (Alt+O) e

adicionar o driver em “Drivers”. Após isso, iniciam-se os ajustes clicando no botão “Extras”. Durante os estudos, especificamente nesse momento, houveram algumas dificuldades devido a informações importantes apresentadas de forma pouco clara pelo manual. Dessa forma, será descrito um pequeno guia para a configuração do driver.

Na aba ***MProt***, na seção ***General***:

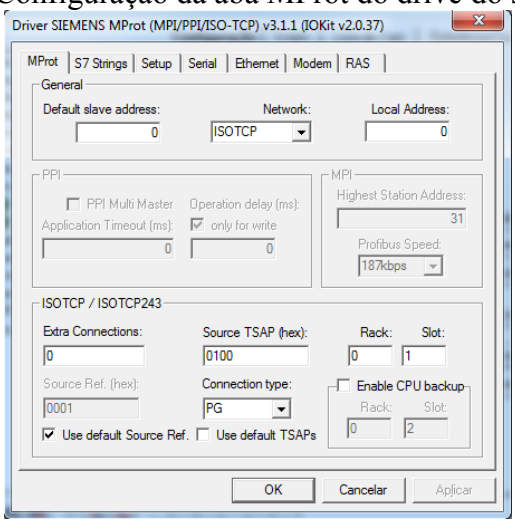
- Em ***Default slave address*** deverá permanecer como zero.
- Em ***Network***, seleciona-se o protocolo ISOTCP.
- Em ***Local Address*** mantem o valor zero de default.

Na seção ***ISOTCP/ISOTCP243***:

- Em ***Extra Connections*** deve-se informar o valor zero
- Deve-se marcar a caixa ***Use default Source Ref***. e deixar desmarcada a caixa ***Use default TSAPs***.
- Em ***Source TSAP (hex)*** deve-se informar o valor 0100 e os dados 0 e 1 em ***Rack*** e ***Slot***, respectivamente.
- ***Connection type*** deverá permanecer em PG.

Ao final, a janela deverá encontrar-se dessa forma, apresentada a seguir.

Figura 13 – Configuração da aba MProt do drive do supervisório.



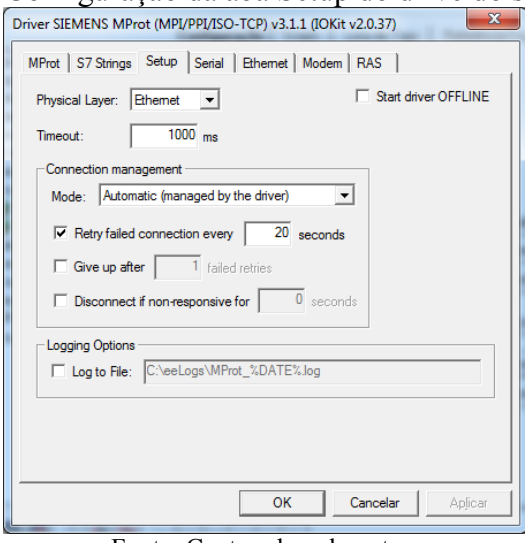
Fonte: Capturado pelo autor.

Na aba ***Setup***:

- Em ***Physical Layer*** deverá se mudado para Ethernet.
- As demais configurações deverão permanecer intactas.

Ao final, a janela deverá estar configurada dessa forma, apresentada a seguir.

Figura 14 – Configuração da aba Setup do drive do supervisório.



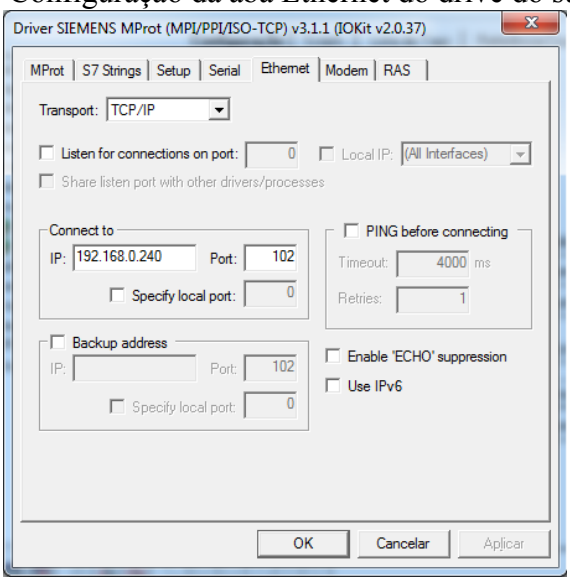
Fonte: Capturado pelo autor.

Na aba ***Ethernet***:

- O protocolo em ***Transport*** deverá ser o TCP/IP.
- No espaço ***Connect to*** deverá ser informado o IP configurado no CLP através do software de programação do próprio controlador. Para a pesquisa foi utilizado o endereço de IP 192.168.0.240, disponibilizado especificamente para fim dessa comunicação. Em ***Port***, por default, deverá ser utilizada a porta 102 por padrão.
- As demais configurações deverão permanecer intactas.

Ao final, a janela deverá estar configurada dessa forma, apresentada a seguir.

Figura 15 – Configuração da aba Ethernet do drive do supervisório.

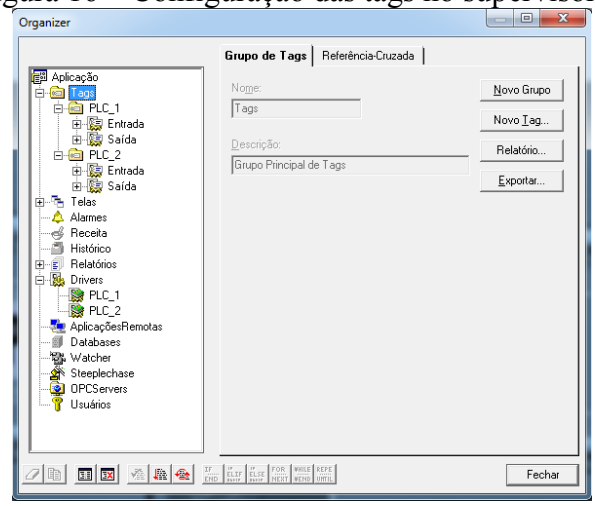


Fonte: Capturado pelo autor.

3.2.2. Configuração das tags

Finalizada essa etapa, agora serão configuradas as tags que serão utilizadas nos botões e scripts do supervisório. Foram criados dois grupos referentes aos respectivos controladores, nomeados de PLC_1 e PLC_2, como mostra a Figura 16. Para cada driver de controlador configurado, deve-se utilizar um arquivo. No caso dessa pesquisa, utilizou-se dois arquivos diferentes do mesmo driver para realizar a configuração dos dois controladores.

Figura 16 – Configuração das tags no supervisório.



Fonte: Capturado pelo autor.

Segundo Siemens e Vipa (2014), para a configuração das tags, deve-se informar os valores de N1, N2, N3 e N4. O manual disponibilizado junto ao driver desempenha papel fundamental nessa etapa, onde serão especificados dados como: tipo de dado, área e endereçamentos. Para a configuração dos valores de N, temos:

- **N1:** Endereço do PLC. Se for igual a 0 (zero) e protocolo diferente de **ISOTCP** ou **ISOTCP243**, é substituído pelo **Default Slave Address**. Se for protocolo **ISOTCP** ou **ISOTCP243**, este valor deve ser deixado em 0 (zero).
- **N2:** Tipo de dado e Área. O valor deve ser composto pelo tipo de dado multiplicado por 100 mais a área (a fórmula é **N2/B2 = TIPO DE DADO × 100 + ÁREA**). Para acessar tais informações, consultar Tabelas 01 e 02.
- **N3:** Se a área selecionada for **V (DB)**, preencha com o número do bloco DB. Caso contrário, deixe em 0 (zero). Caso a memória contenha um bloco DB único ou não especificado, preencha com o valor 1 (um).

- **N4:** Endereço na área ou *offset* do bloco DB. Para usar tipos de dados que ocupam mais de um byte, devem ser colocados endereços múltiplos de dois para tipos de dois bytes (16 bits com e sem sinal) e múltiplos de quatro para tipos de quatro bytes (32 bits com e sem sinal e ponto flutuante de 32 bits).

Tabela 2 – Tipos de dados para configuração de N2.

TIPO DE DADO	
TIPO	SIGNIFICADO
0	Padrão da Área
1	BOOL (Booleano)
2	BYTE (oito bits sem sinal)
3	WORD (16 bits sem sinal)
4	INT (16 bits com sinal)
5	DWORD (32 bits com sinal)
6	DINT (32 bits com sinal)
7	REAL (32 bits de ponto flutuante - IEEE 754)
8	STRING
12	S5TIME (tempo em segundos, 32 bits de ponto flutuante - IEEE 754)

Fonte: Siemens e Vipa (2014).

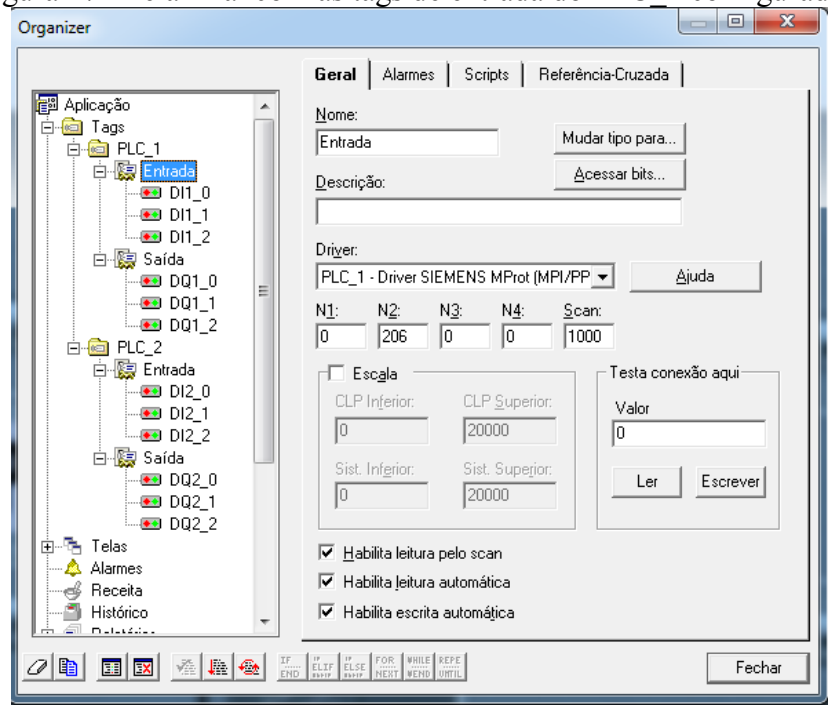
Tabela 3 – Área de dados para configuração de N2.

ÁREA	
TIPO	SIGNIFICADO
0	S
1	SM
2	AI (Analog Input)
3	AQ (Analog Output)
4	C (Counter)
5	T (Timer)
6	I (Digital Input)
7	Q (Digital Output)
8	M (Memory)
9	V (DB)
10	HC (High Speed Counter)

Fonte: Siemens e Vipa (2014).

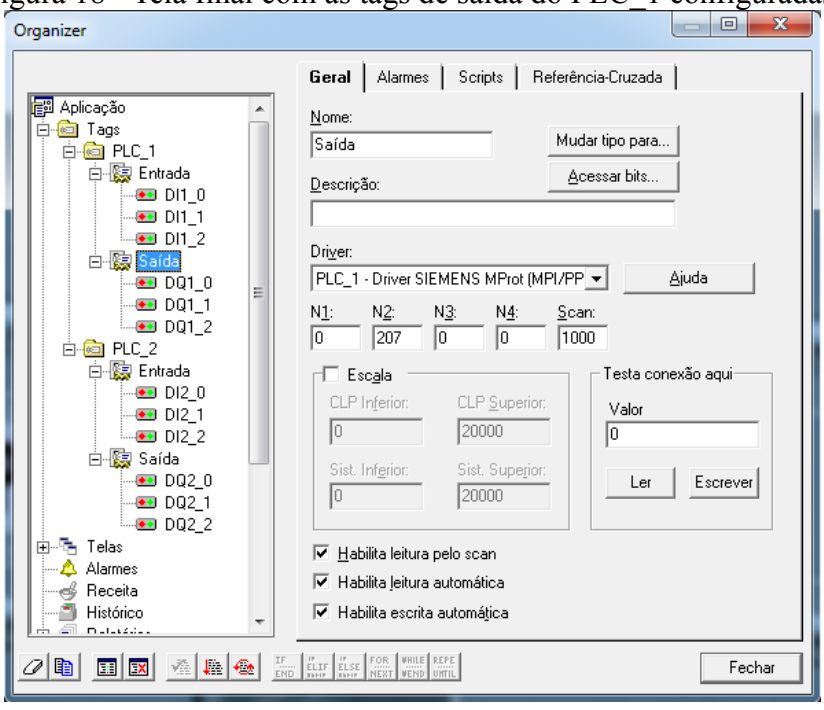
Finalmente, os bits de cada uma das tags foram divididos e devidamente endereçados. Cada uma das tags e seus respectivos bits acessados são apresentados através das figuras a seguir.

Figura 17 - Tela final com as tags de entrada do PLC_1 configuradas.



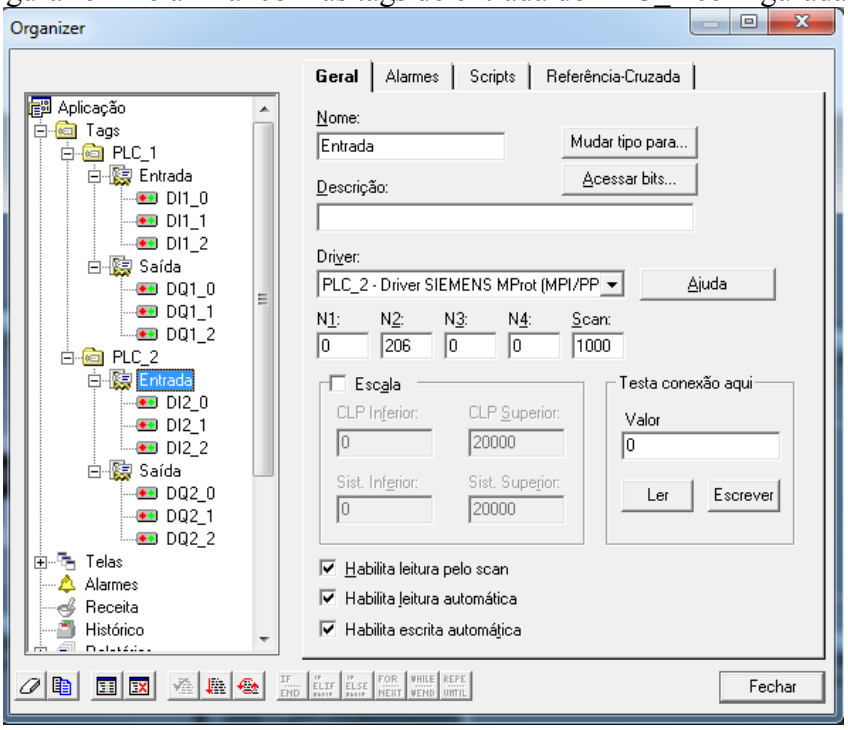
Fonte: Capturado pelo autor.

Figura 18 - Tela final com as tags de saída do PLC_1 configuradas.



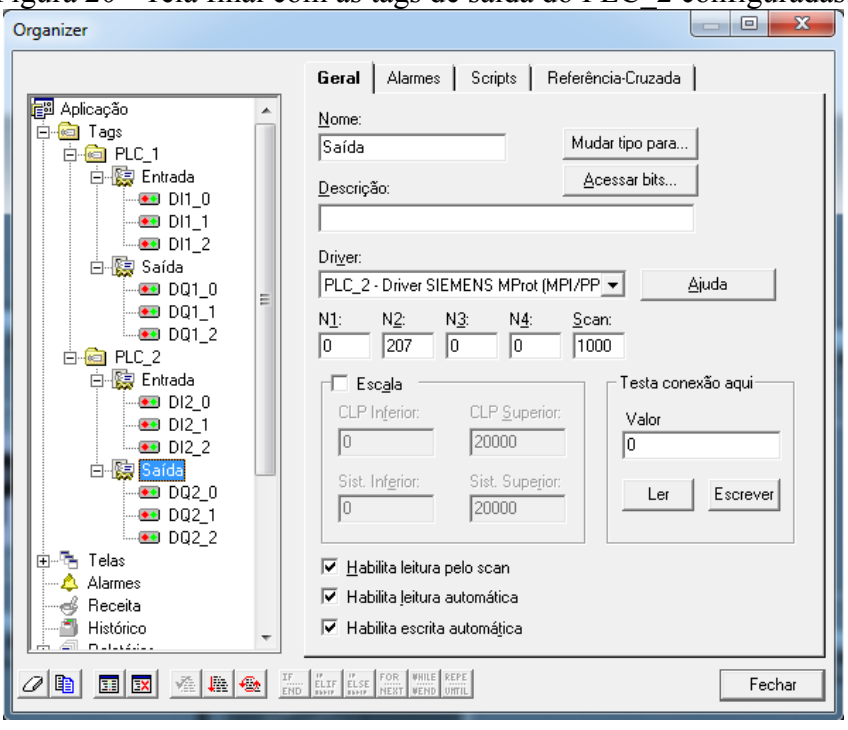
Fonte: Capturado pelo autor.

Figura 19 - Tela final com as tags de entrada do PLC_2 configuradas.



Fonte: Capturado pelo autor.

Figura 20 - Tela final com as tags de saída do PLC_2 configuradas.

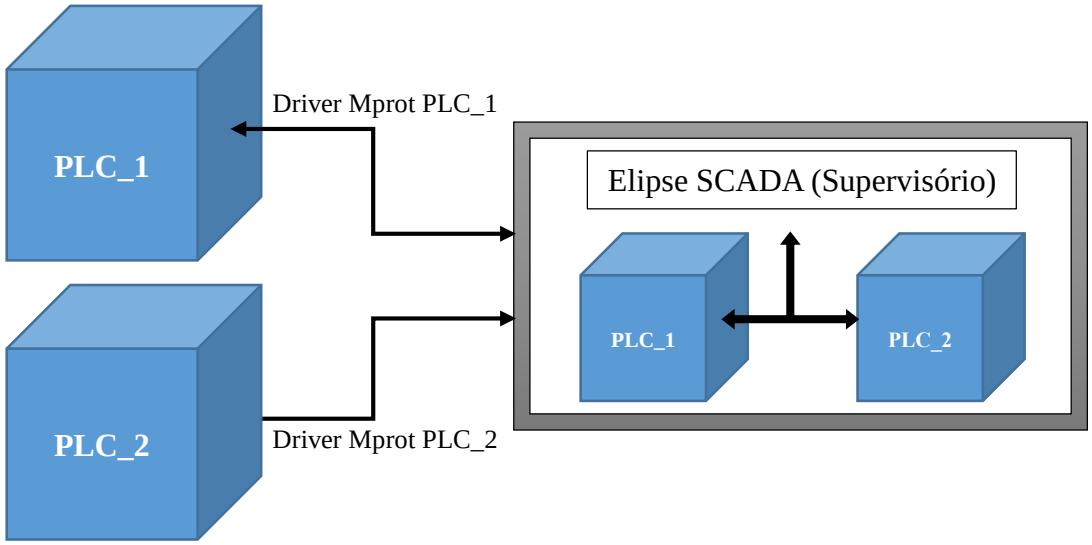


Fonte: Capturado pelo autor.

Nesse momento, a comunicação entre os dispositivos deverá estar devidamente estabelecida. Na seção **Testa conexão aqui**, localizada na parte inferior direita, a comunicação poderá ser verificada através dos botões **Ler** e **Escrever**.

Nesse ponto, está configurada a rede de comunicação entre os CLPs e o supervisório. Foi desenvolvida também uma interface no supervisório para que a comunicação entre os dispositivos fosse testada e observada. Uma figura representativa é apresentada a seguir.

Figura 21 - Representação do progresso de configuração das ferramentas



Fonte: Criado pelo autor.

A configuração da comunicação entre o Arduino e o supervisório é melhor abordada no trabalho de conclusão de curso: "**Solução de Automação Industrial de Baixo Custo Baseada em Plataforma Microcontrolada Arduino"**, desenvolvido pelo aluno Rafael Moreira Albuquerque, em parceria com o curso de Engenharia da Computação da Universidade Federal do Ceará – Campus Sobral. Para o próximo capítulo, realizaremos os devidos testes de comunicação entre as ferramentas e posterior análise dos resultados.

4. RESULTADOS EXPERIMENTAIS

4.1. Comunicação em duplo sentido CLP – CLP

Para os testes de comunicação entre os dois controladores Siemens S7 1214C AC/DC/RLY e o supervisório Elipse SCADA foi utilizado o método descrito pelo capítulo 3, desse mesmo estudo. Inicialmente foram realizadas as configurações dos endereços de IP da rede local de cada um dos CLPs, como mostrado na seção 3.1 do capítulo 3. Em seguida, foi criada uma interface no software para os devidos testes, como descrito na seção 3.2 do capítulo 3.

Todos os equipamentos (controladores e computador) permaneceram ligados à rede local do campus. O endereço de IP estático utilizado para o primeiro controlador foi 192.168.0.240. E para o segundo controlador foi utilizado o endereço de IP estático 192.168.0.241. Ambos disponibilizados pelo Departamento de Tecnologia da Informação (DTI) do próprio campus. Para o computador, foi utilizado o endereço de IP estático 192.168.0.99, escolhido pelo pesquisador com o intuito de evitar conflito de endereços. Todos os endereços utilizaram máscara de sub-rede 255.255.255.0.

Com o intuito de testar a comunicação direta entre os controladores, foi criado um código simples em forma de script no software supervisório. Caso a entrada digital DI a.2 do primeiro controlador (PLC_1) recebesse nível alto de tensão (24Vdc), a saída digital DQ a.2 do segundo controlador (PLC_2) forneceria nível alto de tensão, e vice-versa. Todos os estados deveriam ser adquiridos e exibidos no software supervisório em tempo real. Os testes foram realizados em laboratório. Os equipamentos foram devidamente ligados e os testes foram iniciados.

Os resultados serão apresentados em forma de figuras, adquiridas durante os testes, e exibidos na seguinte disposição.

Figura 22 – Apresentação dos resultados da comunicação CLP - CLP.

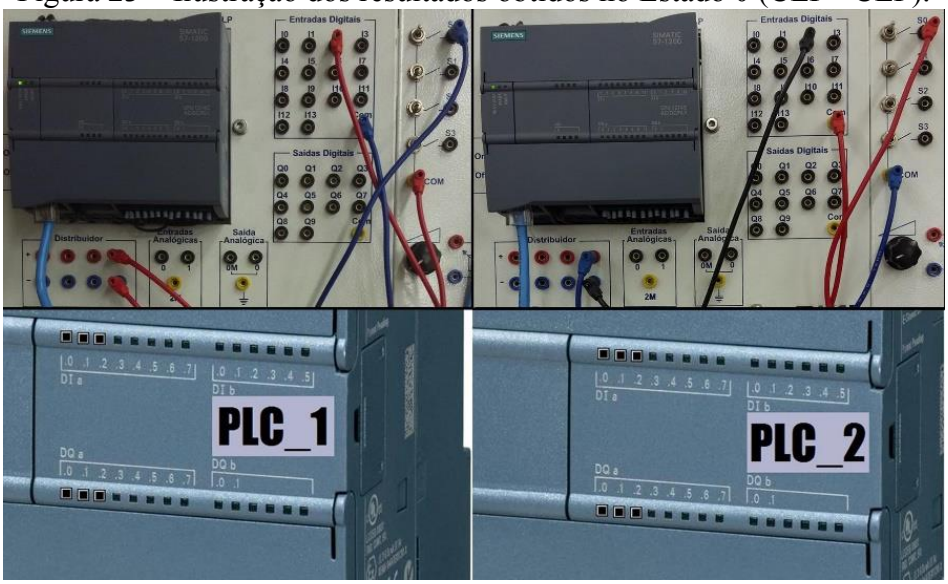
Estado do Primeiro Controlador (PLC_1)	Estado do Segundo Controlador (PLC_2)
Tela da Aplicação no Software Supervisório (Elipse SCADA)	

Fonte: Elaborado pelo autor.

- **Estado 0**

Como estado inicial, os controladores e o computador com o software supervisório são ligados e estão em perfeita comunicação. Entretanto, não há acionamento dos comandos. A Figura 19 demonstra cada um dos dispositivos ligados e seus respectivos estados iniciais.

Figura 23 – Ilustração dos resultados obtidos no Estado 0 (CLP - CLP).

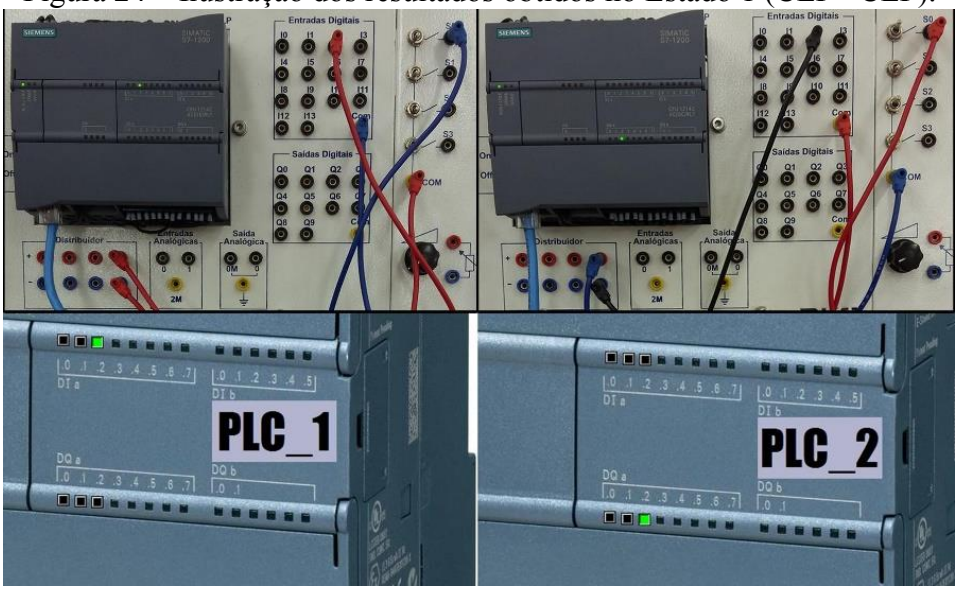


Fonte: Capturado pelo autor.

- **Estado 1**

No próximo estado, a entrada digital DI a.2 do primeiro controlador é acionada através de uma chave, e é observada a eficácia da comunicação através do sinal luminoso exibido pela saída digital DQ a.2 do segundo controlador. A figura a seguir demonstra o resultado obtido nessa etapa.

Figura 24 – Ilustração dos resultados obtidos no Estado 1 (CLP - CLP).



Fonte: Capturado pelo autor.

- **Estado 2**

No estado seguinte, a entrada digital DI a.2 do primeiro controlador permaneceu acionada, para que fosse verificada a estabilidade da interação entre as duas respostas, e a entrada digital DI a.2 do segundo controlador foi ligada. Como descrito inicialmente pelo código, um sinal alto aplicado na entrada digital DI a.2 de um dos controladores, deveria refletir na saída digital DQ a.2 do outro controlador. Dessa forma, após a entrada digital DI a.2 do segundo controlador receber sinal alto, foi observado o acionamento da saída digital DQ a.2 do primeiro controlador. A figura a seguir demonstra os resultados obtidos nesse estado.

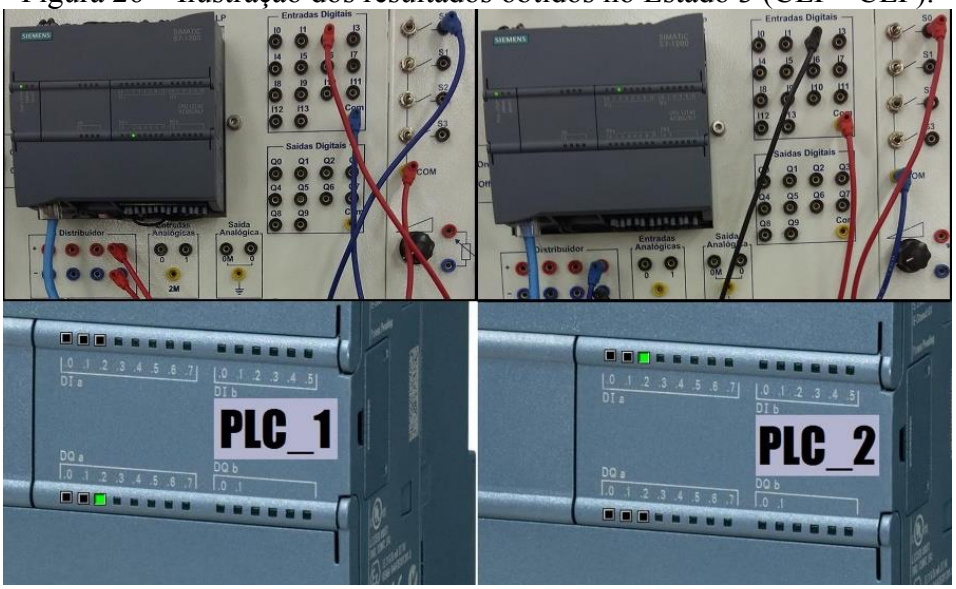
Figura 25 – Ilustração dos resultados obtidos no Estado 2 (CLP - CLP).

Fonte: Capturado pelo autor.

- **Estado 3**

Como último estado, a entrada digital DI a.2 do segundo controlador permanecerá acionada. Entretanto, a entrada digital DI a.2 do primeiro controlador, anteriormente ligada, agora deverá ser desligada. Como esperado, a saída digital DQ a.2 do primeiro controlador permaneceu acionada, e a saída digital DQ a.2 do segundo controlador foi desligada. A figura que segue apresenta os resultados obtidos nessa etapa final.

Figura 26 – Ilustração dos resultados obtidos no Estado 3 (CLP - CLP).



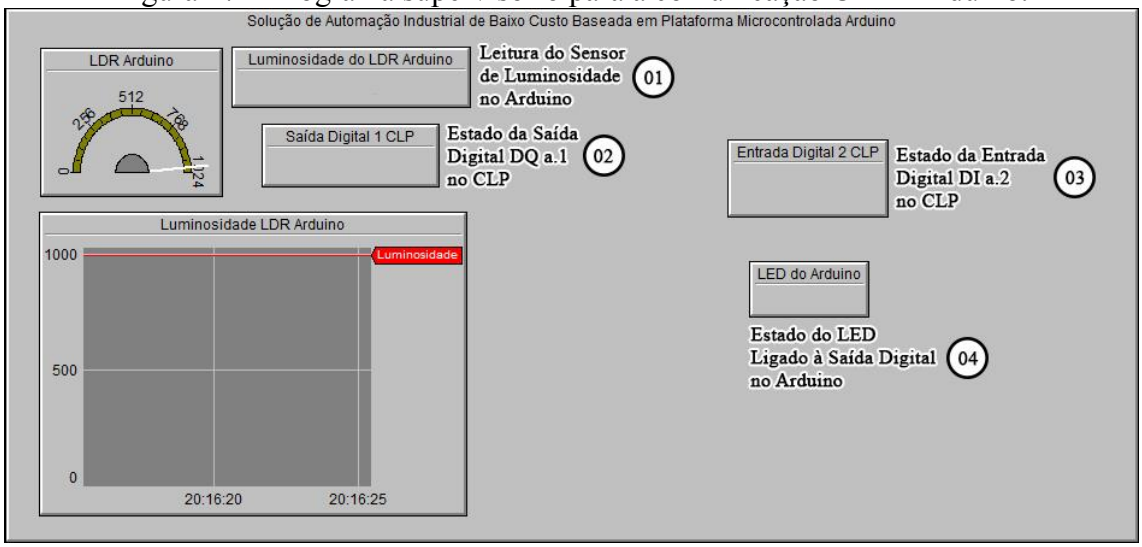
Fonte: Capturado pelo autor.

4.2. Comunicação de duplo sentido CLP – Arduino

Para os testes de comunicação entre o controlador Siemens S7 1214C AC/DC/RLY e a plataforma Arduino em sua versão Uno através do supervisório Elipse SCADA foram utilizados métodos bastantes similares aos descritos na seção anterior. Para a configuração do controlador, foi utilizada a mesma configuração descrita pelo capítulo 3. A plataforma Arduino foi configurada para comunicar-se com a rede local através de um módulo (shield) ethernet, adquirido separadamente da versão Uno. Para a comunicação dos dispositivos (CLP e Arduino) com o supervisório, foram utilizados drivers distintos, ambos disponibilizados pelo desenvolvedor do software supervisório. Para a comunicação entre o Arduino e o supervisório, foi utilizado o Driver Modicon Modbus Master (ASC/RTU/TCP) em sua versão v3.0.11, revisado em 29/05/2015. Para a comunicação entre o CLP e o supervisório, foi utilizado o Driver Siemens M-Prot (PPI, MPI, ISOTCP) em sua versão v3.1.2, com última revisão em 12/05/2015.

Inicialmente foram realizadas as configurações dos endereços de IP para a comunicação com a rede local de cada um dos dispositivos. Em seguida, foi criada uma interface no software supervisório para a realização dos devidos testes, que será apresentada a seguir através da Figura 23.

Figura 27 – Programa supervisório para a comunicação CLP - Arduino.



Fonte: Albuquerque (2015).

Na seção 01 da figura, temos a leitura em tempo real do sensor de luminosidade (LDR) através do Arduino. Na seção 02 é apresentado o estado da saída digital DQ a.1 no CLP. No mostrador 03 é apresentado o estado da entrada digital DI a.2 no CLP. E na

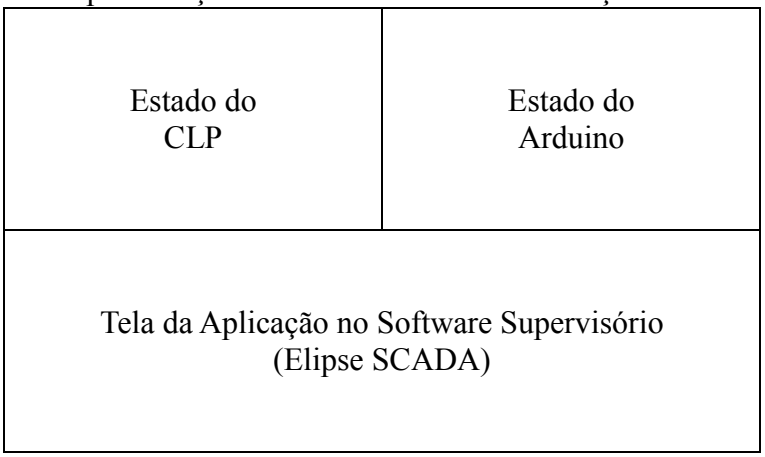
seção 04 é demonstrado o estado do LED, acoplado na saída digital do Arduino. A área 01 apresenta valores analógicos em um intervalo entre 0 a 1000. As áreas 02, 03 e 04 exibem valores digitais, exibindo valor lógicos 1 para sinais altos e 0 para sinais baixos.

Todos os equipamentos (CLP, Arduino e computador) permaneceram ligados à rede local do campus. O endereço de IP estático utilizado para o controlador foi 192.168.0.240. Para o Arduino foi utilizado o endereço de IP estático 192.168.0.241. Ambos disponibilizados pelo Departamento de Tecnologia da Informação (DTI) do próprio campus. Para o computador, foi utilizado o endereço de IP estático 192.168.0.99, escolhido pelo pesquisador com o intuito de evitar conflito de endereços. Todos os endereços utilizaram máscara de sub-rede 255.255.255.0.

Com o intuito de testar a comunicação direta entre os dispositivos, foram criados alguns códigos em forma de script no software supervisório. Para a comunicação Arduino – CLP foi idealizada a seguinte aplicação: caso identificada luminosidade máxima pelo sensor LDR (1000), a saída digital DQ a.1 deveria ser acionada diretamente. E caso a luminosidade seja inferior à máxima (<990), a mesma saída deveria permanecer desligada. Para a comunicação CLP – Arduino foi idealizada a seguinte aplicação: caso seja acionada a entrada digital DI a.2, o LED indicativo ligado ao Arduino deveria acender, simbolizando sinal lógico alto. Todos os estados deveriam ser adquiridos e exibidos no software supervisório em tempo real. Os testes foram realizados em laboratório. Os equipamentos foram devidamente ligados e os testes foram iniciados.

Os resultados serão apresentados em forma de figuras, adquiridas durante os testes, e exibidos na seguinte disposição.

Figura 28 - Apresentação dos resultados da comunicação CLP - Arduino.

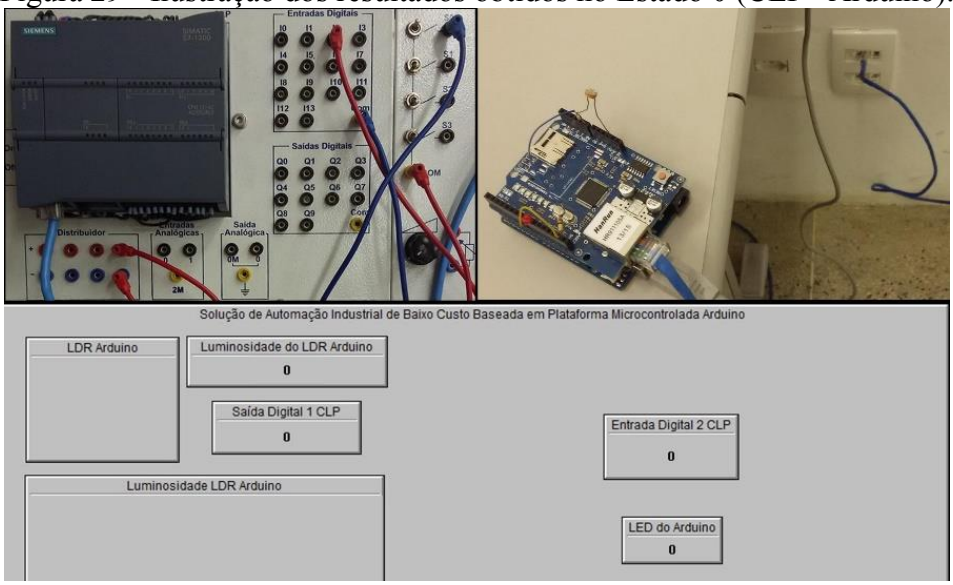


Fonte: Elaborado pelo autor.

- **Estado 0**

Como estado inicial, o Arduino, o CLP e o computador com o software supervisório estão interligados e estão em perfeita comunicação. Entretanto, não há fornecimento de energia aos componentes e, consequentemente, não há acionamento dos comandos. A Figura 25 demonstra cada um dos dispositivos e seus respectivos estados iniciais.

Figura 29 - Ilustração dos resultados obtidos no Estado 0 (CLP - Arduino).

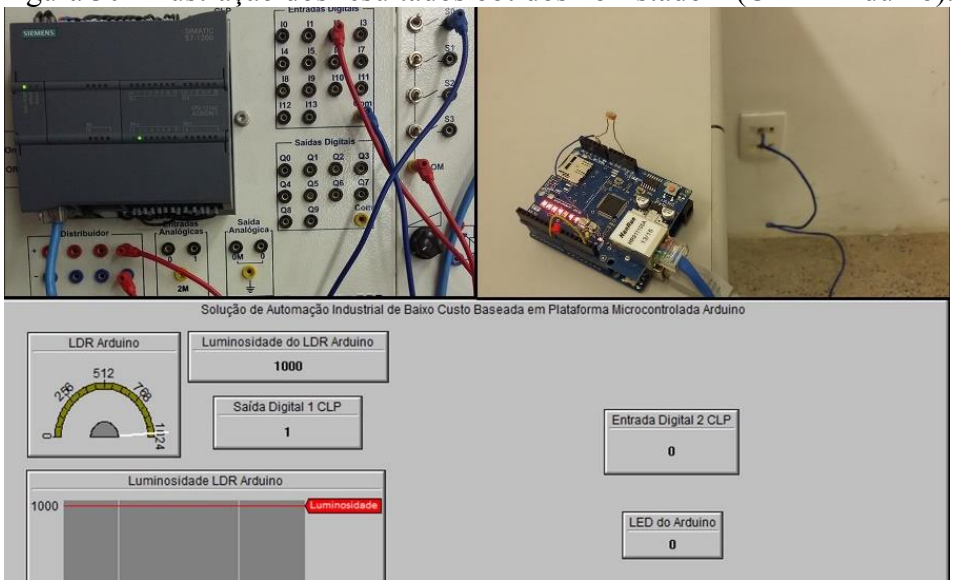


Fonte: Capturado pelo autor.

- **Estado 1**

No próximo estado, é fornecida energia aos dispositivos e observa-se logo a ação do sensor de luminosidade (LDR). Como esperado, em caso de luminosidade máxima, a saída digital DQ a.1 foi prontamente acionada. Os estados dos respectivos dispositivos são exibidos no software supervisório, como previsto. A figura a seguir demonstra o resultado obtido nessa etapa.

Figura 30 - Ilustração dos resultados obtidos no Estado 1 (CLP - Arduino).

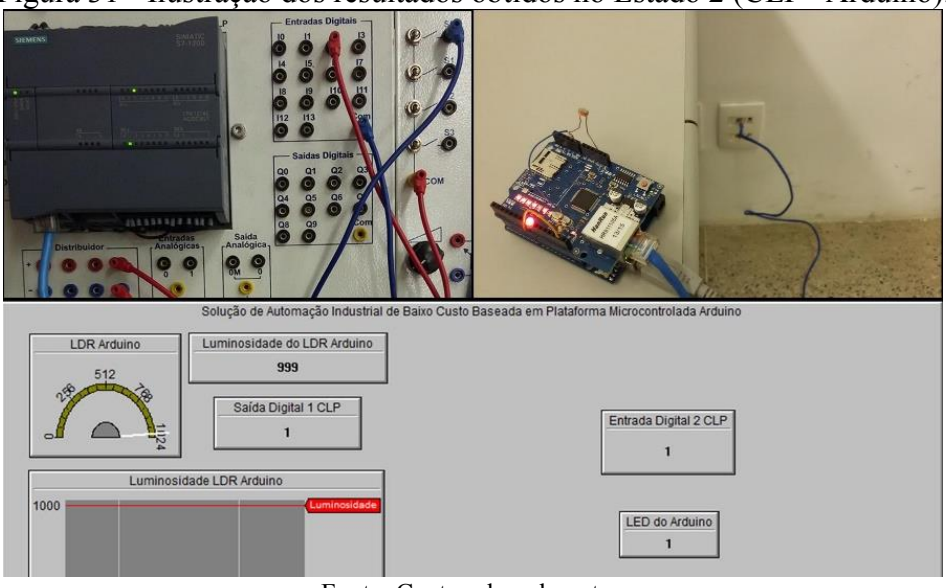


Fonte: Capturado pelo autor.

- **Estado 2**

No estado seguinte, a entrada digital DI a.2 do CLP foi acionada através de uma chave. Como descrito inicialmente pelo código, o LED presente no Arduino exibiu sinal alto, indicando perfeita comunicação entre os dispositivos. A resposta da saída digital DQ a.1 do CLP ao grau de luminosidade verificado pelo Arduino foi mantida, para que fosse verificada a estabilidade da interação entre as duas respostas. Os respectivos estados dos dispositivos são perfeitamente exibidos no software supervisório. A figura a seguir demonstra os resultados obtidos nesse estado.

Figura 31 - Ilustração dos resultados obtidos no Estado 2 (CLP - Arduino).

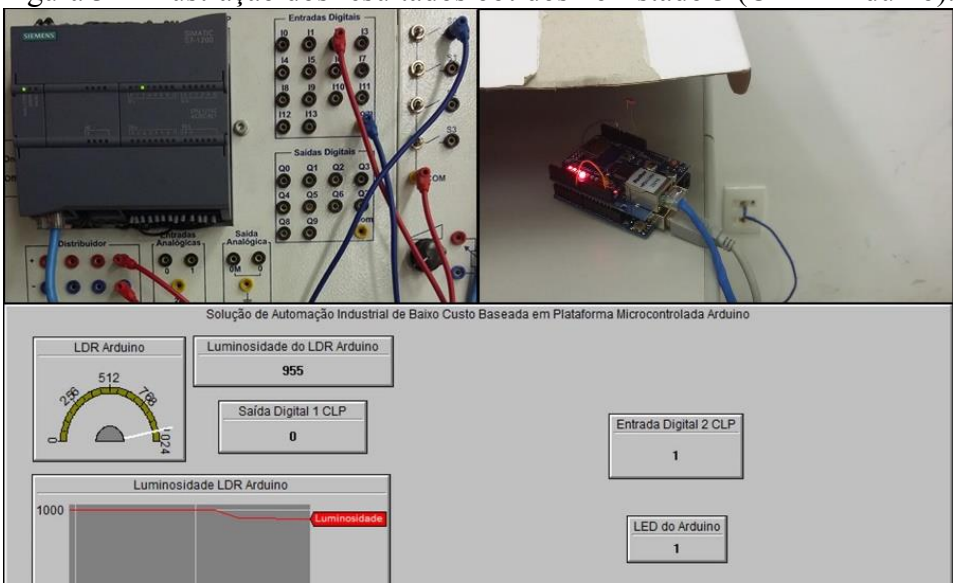


Fonte: Capturado pelo autor.

- **Estado 3**

Como último estado, a entrada digital DI a.2 do CLP continuou sendo acionada através da chave e a resposta observada no Arduino através do LED permaneceu inalterada. Assim, a resposta à variação de luminosidade obtida pelo sensor LDR do Arduino foi testada. Após a diminuição da luminosidade, através da utilização de um anteparo, a saída digital DQ a.1 do CLP demonstrou sinal lógico baixo, como previsto. Os respectivos estados dos dispositivos são exibidos em tempo real através do software supervisório. A figura que segue apresenta os resultados obtidos nessa etapa final.

Figura 32 - Ilustração dos resultados obtidos no Estado 3 (CLP - Arduino).



Fonte: Capturado pelo autor.

Após a realização dos testes de comunicação entre as ferramentas em laboratório, aquisição dos dados e análise dos resultados. Tanto entre CLPs e supervisório, quanto entre CLP, Arduino e supervisório, notou-se facilmente que os resultados obtidos foram satisfatórios e leva-se a crer que os objetivos da pesquisa foram alcançados.

O próximo, e último capítulo, foi reservado para as conclusões adquiridas com a análise dos resultados obtidos. Ao final são sugeridos temas que podem ser utilizados para dar continuidade a futuros estudos referentes a esse trabalho.

5. CONCLUSÕES E TRABALHOS FUTUROS

Como proposto inicialmente, o trabalho se voltou para a efetiva comunicação entre os controladores lógicos programáveis, amplamente utilizados como ferramenta de automação nas indústrias, e outras plataformas, no caso, o Arduino. Para isso, foi utilizado um software supervisório como forma de comunicação entre os dispositivos e aquisição de dados. A justificativa da pesquisa se pautou principalmente na busca de novas ferramentas para a automação industrial, redução de custos e busca por métodos mais simples para a comunicação interdispositivos.

Inicialmente foram definidos conceitos importantes para a compreensão do trabalho e da proposta de pesquisa, como, o conceito de automação, redes industriais, os dispositivos utilizados, como os controladores lógicos programáveis e o Arduino, e finalmente, o que seriam e como funcionam os sistemas supervisórios.

Após a revisão e explanação dos principais conceitos utilizados, foi apresentada a forma de configuração das ferramentas utilizadas, como: A configuração dos controladores lógicos programáveis e a configuração do sistema supervisório, utilizado na comunicação dos dispositivos.

Ao final, foram realizados os devidos testes em laboratório e os resultados obtidos foram bastantes satisfatórios. Foi observada a efetiva comunicação entre os controladores lógicos programáveis e o supervisório em vários estados distintos de acionamento. Em seguida, foi identificada a comunicação direta entre as ferramentas Arduino, controlador lógico programável e supervisório, também em diferentes estados de acionamento. Durante a realização dos testes em laboratório, foram identificadas algumas dificuldades com a comunicação entre os CLPs utilizando a interface TIA Portal V11, oferecida pelo fabricante do controlador. Como esperado, essas dificuldades foram facilmente resolvidas com a substituição do método de comunicação usual pelo supervisório. Durante a realização dos ensaios, foi observado que os resultados obtidos durante os testes corroboraram completamente com os resultados esperados.

Como sugestão para trabalhos futuros, pode-se estudar a possibilidade de adição de uma ferramenta de supervisão via web, também disponibilizada pelo Elipse Software, que realizaria a comunicação entre os dispositivos, aquisição e apresentação de dados via web. Pode-se também estudar a possibilidade de adição de módulos de comunicação wireless para os dispositivos utilizados, a fim de reduzir custos com cabos e resolver problemas com barreiras físicas.

REFERÊNCIAS

ALBUQUERQUE, Rafael Moreira. **Solução de Automação Industrial de Baixo Custo Baseada em Plataforma Microcontrolada Arduino**. 2015. Monografia (Graduação em Engenharia da Computação) – Departamento de Engenharia da Computação, Universidade Federal do Ceará, Sobral, 2015.

ARDUINO. Products. **Arduino Ethernet Shield**.
Disponível em: < https://www.arduino.cc/en/Main/ArduinoEthernetShield>
Acesso em: 09 de dezembro de 2015.

ARDUINO. Products. **Arduino Uno**.
Disponível em: <https://www.arduino.cc/en/Main/ArduinoBoardUno>
Acesso em: 09 de dezembro de 2015.

DANEELS, Alex; SALTER, Wayne. **What is SCADA?.** CERN - European Organization for Nuclear Research, CNL-2000-003, Vol. XXXV, issue no 3.

DECOTIGNIE, Jean Dominique. **A perspective on Ethernet-TCP/IP as a fieldbus**. IFAC International Conference on Fieldbus Systems and theirs Applications, 2001.

ELIPSE SOFTWARE. Produtos. **Elipse SCADA**.
Disponível em: <http://www.elipse.com.br/port/scada.aspx>
Acesso em: 09 de dezembro de 2015.

FOROUZAN, Behrouz A.; FEGAN, Sophia Chung**. Protocolo TCP/IP.** 3ed – Porto Alegre: AMGH Editora, 2009.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 1991.

LUGLI, Alexandre B.; SANTOS, Max M. D. **Rede Industriais para Automação Industrial: AS-I, Profibus e Profinet**. - São Paulo: Editora Érica, 2010.

LUGLI, Alexandre B.; SANTOS, Max M. D. **Sistemas de Fieldbus para Automação Industrial: Devicenet, CANOpen, SDS e Ethernet**. - São Paulo: Editora Érica, 2009.

MCROBERTS, Michael. **Beggining Arduino**. Apress. Nova Iorque: 2010.

MENDES, Douglas Rocha. **Redes de Computadores: Teoria e Prática.** 2ed – São Paulo: Novatec, 2015.

MONK, Simon. **Programação com Arduino II: Passos avançados com sketches.** 1ed - Porto Alegre: Bookman Editora, 2015.

MORAES, Cícero Couto de; CASTRUCCI, Plínio de Lauro. **Engenharia de Automação Industrial**. 2ed – Rio de Janeiro: LTC, 2013.

O SETOR ELÉTRICO. Pesquisas de Mercado e Guias Setoriais. **Automação e Gerenciamento de Energia**. Edição 95, dezembro de 2013. Publicação Digital. Disponível em: <http://www.osetoreletrico.com.br/web/documentos/guias_setoriais/ed-95_Pesquisa_Automacao-e-gerenciamento-de-energia.pdf>
Acesso em 10 de dezembro de 2015.

PIRES, Paulo Sérgio Motta; OLIVEIRA, Luiz Affonso H. Guedes de; BARROS, Diogo Nascimento. **Aspectos de segurança em sistemas SCADA – Uma visão geral**. 4º Congresso Internacional de Automação, Sistemas e Instrumentação. In: Controle & Instrumentação, Maio de 2005, p.112-119.

REYMASTER. Blog. **Automação e redes industriais**.
Disponível em: < http://d705243685.tecla337.tecla.com.br/blog/142-automacao-e-redes-industriais>
Acesso em: 10 de dezembro de 2015.

RAJ, Ritika; ANNADATE, S.A. **Development of SCADA like application using Arduino with .net interface.** International Journal of Communication Network Security, ISSN: 2231 – 1882, Volume-2, Issue-2, 2013.

ROSÁRIO, João Maurício. Automação Industrial. São Paulo: Baraúna, 2009.

ROQUE, Luiz Alberto Oliveira Lima. **Automação de Processos com Linguagem Ladder e Sistemas Supervisórios**. 1ed – Rio de Janeiro: LTC, 2014.

SEIXAS FILHO, Constantino. **A produção em foco.** In: Scantech News, Rio de Janeiro, Setembro 1999, p. 26-30.

SIEMENS, VIPA. **Driver Siemens M-PROT**. Versão 3.1.1. Última Atualização em 19/09/2014.

SIEMENS. Controlador Simatic. **Simatic s7 – 1200**. Setembro 2012
Disponível em: <www.siemens.com.br/simatic-s7-1200>
Acesso em: 09 de dezembro de 2015.

SILVA, Ana Paula Gonçalves da; SALVADOR, Marcelo. **O que são sistemas supervisórios?**. Atualizado em 20/12/2005
Disponível em: <http://www.wectrus.com.br/artigos/sist_superv.pdf>
Acesso em: 09 de dezembro de 2015.

SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da pesquisa e elaboração de dissertação**. – 4ª ed. rev. atual. – Florianópolis: UFSC, 2005. 138p.

TUNG, S. C.; LI, W. J.; HUANG, S. M.; LAI, Y. T. **A Web-Based Arduino Supervisory Control System**. Vols. 284-287, pp. 3216-3220, Jan. 2013.