

UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

DANIEL KHÉDE DOURADO VILLA

**COMUNICAÇÃO MODBUS VIA RS-485 NO RELÉ PROGRAMÁVEL
CLIC02-WEG PARA CRIAÇÃO DE INTERFACES GRÁFICAS EM
APLICAÇÕES DE DOMÓTICA**

VIÇOSA
2014

DANIEL KHÉDE DOURADO VILLA

**COMUNICAÇÃO MODBUS VIA RS-485 NO RELÉ PROGRAMÁVEL
CLIC02-WEG PARA CRIAÇÃO DE INTERFACES GRÁFICAS EM
APLICAÇÕES DE DOMÓTICA**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Alexandre Santos Brandão.

VIÇOSA
2014

DANIEL KHÉDE DOURADO VILLA

**COMUNICAÇÃO MODBUS VIA RS-485 NO RELÉ PROGRAMÁVEL
CLIC02-WEG PARA CRIAÇÃO DE INTERFACES GRÁFICAS EM
APLICAÇÕES DE DOMÓTICA**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em ___ de _____ de 2014.

COMISSÃO EXAMINADORA

Prof. Dr. Alexandre Santos Brandão
Universidade Federal de Viçosa

Prof. M.Sc. José Vítor Nicacio
Universidade Federal de Viçosa

Prof. B. Sc. Paulo Fábio Figueiredo Rocha
Universidade Federal de Viçosa

*“And be a simple, kind of man.
Oh be something, you love and understand.
Baby be a simple, kind of man.
Oh, won't you do this for me son,
If you can?”*

(Lynyrd Skynyrd)

Resumo

O grande avanço e a enorme importância que os sistemas supervisórios possuem nos sistemas automatizados são a motivação por trás do trabalho. Desde sua concepção, IHM's transformaram a forma de como a automação é feita na indústria em todo mundo. Procedimentos que antes eram dispendiosos e difíceis de serem executados por limitações físicas ou tecnológicas foram possíveis de serem concebidos com o seu advento.

Juntamente a isso, almeja-se implementar os conceitos de sistema supervisório tão difundidos nos dias atuais nas indústrias em outros ramos do mercado, como o imobiliário. As chamadas casas do futuro são objetos de consumo desejados cada vez mais no mercado, em que a automação faz com que seja possível o usuário interagir com a própria casa, aumentando a segurança, comodidade e o conforto que ela proporciona.

Para isso a implementação de uma automação residencial foi feita pelo CLP Clic-02 a uma maquete de forma a simular a planta real de uma residência. Foram abordadas automações nas áreas de segurança, iluminação, irrigação, climatização e acionamento de portas, todas estas monitoradas via IHM a ser elaborada neste trabalho. Fazer a ponte entre o CLP e esta IHM, utilizando o protocolo Modbus (RTU), foi o objetivo principal do texto.

Este trabalho teve como meta estabelecer a comunicação entre um controlador lógico programável e um computador pessoal, tornando possível elaborar um Sistema Supervisório atuante sobre o sistema no qual o CLP pertence.

Abstract

The motivation behind this work is the importance that supervisory systems have today in automated systems. Since his conception, the man-machine interface transformed the way automation has been done in industry worldwide. Procedures that were previously difficult or expensive because of physical and technological limitation nowadays are able to be achieved because of its advent.

The concepts of supervisory systems so widespread in industry are aimed by different kinds of business, like the house automation business. The so called houses of the future are a success around the world, with customers interest rising year after year. In that kind of houses, automation makes possible for the user to interact with his own house, increasing security, convenience and the comfort that it provides.

A model was used to simulate a real plant of a residence and for the home automation was used a PLC WEG Clic-02. In this work the focus were the automation about security, lighting, irrigation, climatisation and doors activation, all these monitored via the HMI. Making a connection between PLC and MMI (Man-Machine Interface) was the main goal of the text, for that is used the Modbus protocol (RTU).

This work aimed to establish communication between a programmable logic controller and a personal computer, making possible to elaborate a supervisory system acting on the system on which the PLC belongs.

Sumário

| | | |
|-------|--|----|
| 1 | Introdução..... | 15 |
| 1.1 | Redes Industriais | 16 |
| 1.2 | Controlador Lógico Programável (CLP) | 18 |
| 1.2.1 | Introdução..... | 18 |
| 1.2.2 | Clic02 - WEG..... | 18 |
| 1.3 | Interface Homem-Máquina..... | 20 |
| 1.3.1 | Tipos de IHM | 21 |
| 1.3.2 | Funcionamento e Aplicações da IHM | 21 |
| 1.3.4 | Vantagens e Conveniências | 22 |
| 1.3.5 | Flexibilidade da Interface | 22 |
| 1.4 | Expansão das Redes Industriais: Domótica..... | 22 |
| 1.5 | Objetivos..... | 23 |
| 1.6 | Contribuição | 23 |
| 2 | Desenvolvimento..... | 25 |
| 2.1 | Protocolo MODBUS | 25 |
| 2.1.1 | Tipos de Modbus | 25 |
| 2.1.2 | Modbus RTU e ASCII..... | 25 |
| 2.1.3 | Modbus RTU | 26 |
| 2.1.4 | Estrutura de perguntas e respostas no protocolo Modbus RTU | 28 |
| 2.1.5 | Aplicação do protocolo ao Clic-02 VT-D | 35 |
| 3 | Aplicação em Domótica..... | 45 |
| 3.2 | IHM | 49 |
| 3.3 | Resultados Finais..... | 52 |
| 3.4 | Outras aplicações: Partida de motor Estrela-Triângulo..... | 57 |
| 4 | Conclusão..... | 60 |
| 5 | Referências Bibliográficas..... | 60 |

Lista de Figuras

| | |
|---|----|
| Figura 1 - Tecnologia dominante..... | 16 |
| Figura 2 - Sistema Integrado de Gestão Empresarial..... | 16 |
| Figura 3 - Classificação das redes industriais e estrutura funcional..... | 17 |
| Figura 4 - CLP Click-02, estrutura básica..... | 19 |
| Figura 5 - Frame de mensagem padrão. | 26 |
| Figura 6 - Ligação computador e CLP..... | 35 |
| Figura 7 - Configurações do Sistema..... | 36 |
| Figura 8 - Exemplo de software mestre..... | 37 |
| Figura 9 - Leitura registrador 1544 do CLP Clic-02..... | 42 |
| Figura 10 - Escrita do valor 511 decimal no registrador 1544 do dispositivo 01..... | 43 |
| Figura 11 - Leitura registrador 1544 do dispositivo 01..... | 44 |
| Figura 12 - Ladder iluminação interna..... | 45 |
| Figura 13 - Circuito à transistor. | 46 |
| Figura 14 - Ladder iluminação externa..... | 46 |
| Figura 15 - Ladder para o sistema de segurança..... | 47 |
| Figura 16 - Configuração do ganho na entrada analógica A1..... | 47 |
| Figura 17 - Ladder sistema de temperatura..... | 48 |
| Figura 18 - Ladder irrigação, com RTC ligado de domingo a domingo..... | 49 |
| Figura 19 - Ladder porta da garagem..... | 49 |
| Figura 20 - Conteúdo do menu Início..... | 50 |
| Figura 21 - Conteúdo do menu Ações..... | 51 |
| Figura 22 - IHM da casa automatizada..... | 51 |
| Figura 23 - LEDs funcionando no primeiro estágio de falta de luminosidade..... | 52 |
| Figura 24 - LEDs funcionando no segundo estágio de falta de luminosidade..... | 53 |
| Figura 25 - LEDs funcionando no terceiro estágio de falta de luminosidade..... | 53 |
| Figura 26 - Iluminação externa ativada..... | 53 |

| | |
|--|----|
| Figura 27 - : Sistema de iluminação interna e externa em conjunto..... | 54 |
| Figura 28 - Registro da IHM no momento em que foi tirada a foto da Figura 24..... | 54 |
| Figura 29 - Alarme ativado sem a presença indesejada..... | 55 |
| Figura 30 - Alarme ativado detectando a presença..... | 55 |
| Figura 31 - Cooler instalado para a refrigeração do sistema..... | 55 |
| Figura 32 - Sensor da garagem e LED de portão fechado..... | 56 |
| Figura 33 - Sensor da garagem e LED portão aberto..... | 56 |
| Figura 34 - LED azul indicando que a automação da irrigação..... | 57 |
| Figura 35 - Após pressionado o botão ligar motor, este liga-se em estrela..... | 58 |
| Figura 36 - Após passado o tempo pré-determinado e atingido a velocidade requerida é feito o chaveamento que pode ser visto pela IHM que supervisiona o estado do motor..... | 58 |
| Figura 37 - O motor desligado após alto nível de temperatura detectado pelo sensor..... | 59 |

Lista de Tabelas

| | |
|--|----|
| Tabela 1 – Especificações do controlador Clic-02 (1/2). | 19 |
| Tabela 2 - Especificações do controlador Clic-02 (2/2). | 20 |
| Tabela 3 – Pergunta para leitura dos bits 20-70 no dispositivo 18. | 29 |
| Tabela 4 – Resposta para leitura dos bits 20-70 no dispositivo 18. | 30 |
| Tabela 5 – Pergunta para leitura dos registradores 1555-1557 no dispositivo escravo 25. | 31 |
| Tabela 6 – Resposta para leitura dos registradores 1555-1557 no dispositivo escravo 25. | 32 |
| Tabela 7 – Pergunta para escrita do bit 11136 no dispositivo 78. | 32 |
| Tabela 8 – Resposta para escrita do bit 11136 no dispositivo 78. | 32 |
| Tabela 9 - Pergunta para escrita do valor 31546 no registrador 2832 do dispositivo #7. | 33 |
| Tabela 10 – Resposta para escrita do valor 31546 no registrador 2832 do dispositivo #7. | 34 |
| Tabela 11 - Pergunta para escrita nos registradores 7 e 8 do dispositivo 1. | 34 |
| Tabela 12 – Resposta para escrita nos registradores 7 e 8 do dispositivo 1. | 35 |
| Tabela 13 - Parâmetros de configuração obrigatórios para comunicação. | 36 |
| Tabela 14 - Variáveis digitais | 38 |
| Tabela 15 - Variáveis analógicas | 38 |
| Tabela 16 - Estado dos registradores. | 39 |
| Tabela 17 - Endereço bit a bit das variáveis digitais. | 39 |
| Tabela 18 - Leitura registrador 1544 do dispositivo 01. | 41 |
| Tabela 19 - Escrita do valor 511 decimal no registrador 1544 do dispositivo 01. | 42 |
| Tabela 20: Leitura do registrador 1544 do dispositivo 01. | 43 |

1 Introdução

Pode-se dizer que o desejo de controlar os processos industriais acompanha o homem desde a criação das primeiras máquinas. Até a década de 1940, as plantas eram operadas manualmente por um grande número de operadores, os quais valiam-se de alguns poucos instrumentos mecânicos elementares que realizavam controle local (GUTIERREZ; PAN, 2008).

O termo Automação, do inglês automation, foi um termo inventado pelo marketing da indústria de equipamentos da década de 1960. O neologismo, sem dúvida, buscava enfatizar a participação de computadores no controle automático industrial (MORAES; CASTRUCCI, 2007).

Automação descreve um conceito amplo, envolvendo um conjunto de técnicas de controle, das quais é criado um sistema ativo, capaz de fornecer a resposta adequada em função das informações que recebe do processo em que está atuando. Dependendo das informações, o sistema calculará a melhor ação corretiva a ser executada (WEG, 2010).

Inicialmente, a implantação de processos automatizados na indústria tinha o objetivo de alcançar maior produtividade e redução de custos. O investimento para implantação de sistemas automáticos é elevado e, além disso, a nova instalação requer recursos, inclusive humanos, dispendiosos para sua manutenção. Atualmente, o principal motor da automação é a busca de maior qualidade dos processos, para reduzir perdas, (com reflexo em custos) e possibilitar a fabricação de bens que de outra forma não poderiam ser produzidos, bem como do aumento da sua flexibilidade. Outra justificativa para os pesados investimentos em automação que têm sido feitos é a segurança de processos industriais e de infra-estrutura críticos, pois a automação tem sido vista como uma forma de minimizar o erro humano (GUTIERREZ; PAN, 2008).

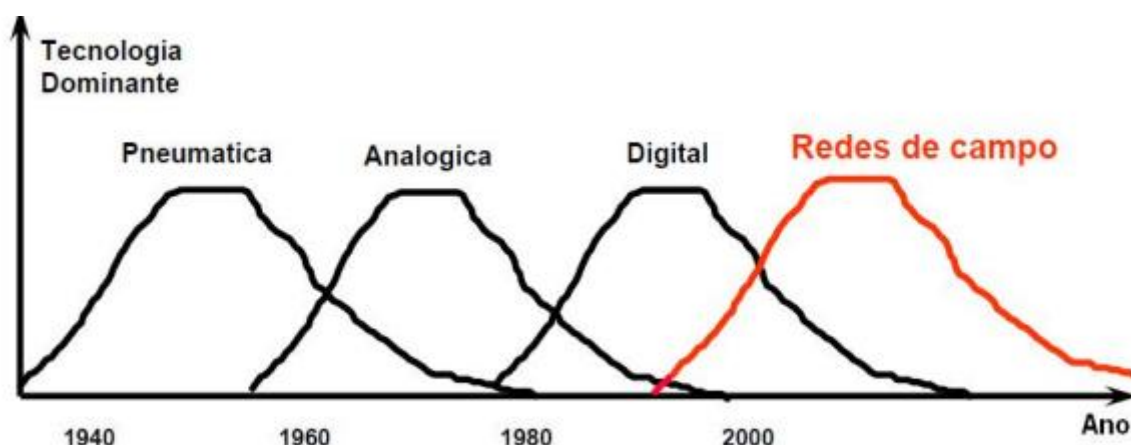


Figura 1: Tecnologia dominante (GUEDES, 2005)

1.1 Redes Industriais

No processo de produção atual as informações passam através de uma série de níveis corporativos, de tomada de decisão, e estes retransmitem as informações com o “campo”, ou seja, os departamentos, de forma que todas as decisões saem do coração da empresa, a administração.

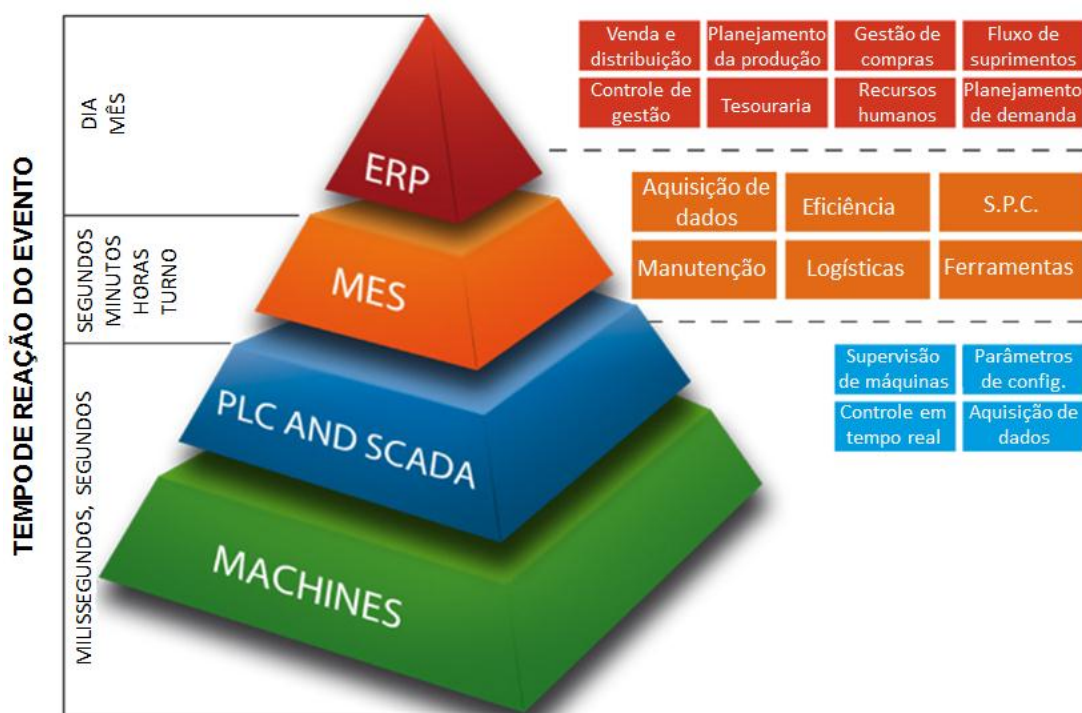


Figura 2: Sistema Integrado de Gestão Empresarial

No topo da pirâmide está o Sistema Integrado de Gestão (SIG), em inglês Enterprise Resource Planning (ERP), sendo este uma ferramenta indispensável permitindo que os fabricantes classifiquem e informatizem fluxos de negócios e logística. Na base da pirâmide estão os diversos departamentos, com suas máquinas e equipamentos, CLPs, instrumentos, documentos, depósitos, sistemas de supervisão e todas outras ferramentas igualmente importantes para garantir o controle do processo. A união destas diferentes áreas dentro da empresa é o caminho para a sobrevivência do próprio negócio.

Controlar as ligações do processo lógico e físico, identificando os responsáveis por entregar cada ordem em cada setor e a sincronização dentro da empresa entre os setores de baixo para cima, e vice-versa, significa reduzir ineficiências, aumentar a qualidade e reduzir o desperdício, incluindo os custos ocultos.

Como todo esse processo trata de um sistema de computação distribuída, as redes de comunicação industriais estão presentes em todos os níveis de interação. A fim de tornar possível a comunicação entre os computadores, PLCs, relés de proteção, controladores de processo, atuadores de válvulas, transdutores de energia, etc, estas redes tiveram que se submeter a padronizações, tornando possível uma maior integração de todas estas ferramentas.

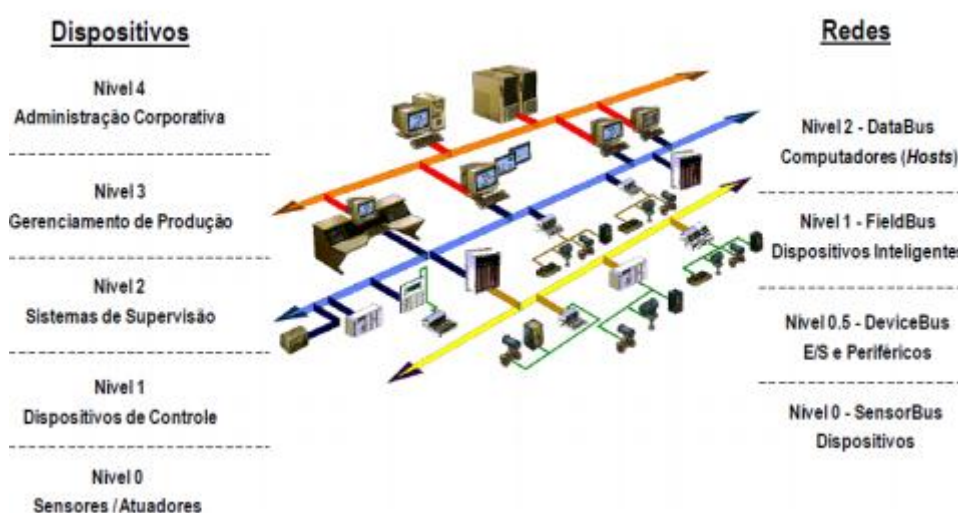


Figura 3: Classificação das redes industriais e estrutura funcional

Diversos protocolos foram então criados, cada um com suas características não possuindo um protocolo certo ou errado. Estes são apenas um conjunto de regras sobre como

se dará a comunicação entre as partes envolvidas. Os protocolos mais utilizados são: MODBUS, PROFIBUS, ETHERNET e FIELDBUS FOUNDATION.

1.2 Controlador Lógico Programável (CLP)

1.2.1 Introdução

A automação de diversos processos eletromecânicos, tais como o movimento de máquinas em uma linha de montagem, é feito através do uso de pequenos computadores chamados controladores lógicos programáveis (CLPs). Um CLP contém um microprocessador que é programado utilizando uma linguagem de computador específica. Normalmente, o programa que será utilizado para automação de um processo é escrito em um computador e, em seguida, é transferido para o CLP. O programa é armazenada no controlador em uma memória não-volátil.

Controladores lógicos programáveis normalmente contêm um número variável de portas de entrada e saída (E/S) e geralmente utilizam um reduzido set de instruções, possibilitando uma execução mais rápida da ação. CLPs são projetados para aplicações em tempo real e muitas vezes devem resistir aos ambientes hostis encontrados nas fábricas, tais como altos níveis de vibração e ruído. O circuito possuidor do controlador lógico programável monitora o status de vários sensores, que controlam os atuadores de starters de motores, solenóides, lâmpadas, monitores e válvulas.

Este tipo de controlador é responsável por grandes avanços no campo da automação industrial. Sistemas de automação anteriores tinham que usar dezenas (ou até centenas) de relés individuais, temporizadores e sequenciadores, que tinham que ser substituídos ou rearranjados sempre que o processo automatizado sofria alguma mudança. Em muitos casos, um CLP permite a substituição de todos os relês e temporizadores dentro de uma indústria.

1.2.2 Clic02 - WEG

O CLP Clic02 da WEG é um controlador bastante utilizado em automações de pequeno e médio porte, não possuindo a robustez de um CLP modular industrial, mas sendo capaz de desempenhar uma infinidade de aplicações em ambientes não tão agressivos. Este é possuidor de boa velocidade, fácil configuração, além de possuir uma grande gama de funções capazes de realizar tarefas como intertravamento, temporização, contagem, comparação entre variáveis e com o tempo, PWM e operações matemáticas básicas.



Figura 4: CLP Click-02, estrutura básica

Esta linha de CLPs possui diversos modelos, que diferenciam-se entre si em características como número de entradas e saídas, por possuírem protocolos de comunicação embarcados, tensão de alimentação, tipos de saída (a transistor ou a relé) e possibilidade quanto a módulos de expansão. As tabelas abaixo demonstram as características destes CLPs.

Tabela 1 – Especificações do controlador Clic-02 (1/2)

| Modelo | Tensão de Alimentação | Entradas | | Saídas Digitais | | Display & Teclado | Comunicação RS-485 | Máximo de E/S |
|-----------------|-----------------------|-----------|-----------|-----------------|------------|-------------------|--------------------|---------------|
| | | Digital | Análogica | Relé | Transistor | | | |
| CLW-02/12HR-D | 24 Vcc | 6 (8) ** | 2 ** | 4 | - | √, Z01-Z04 | - | 36 + 4 ** |
| CLW-02/12HT-D | | 6 (8) ** | 2 ** | - | 4 | √, Z01-Z04 | - | 36 + 4 ** |
| CLW-02/20HR-D | | 8 (12) ** | 4 ** | 8 | - | √, Z01-Z04 | - | 44 + 4 ** |
| CLW-02/20HT-D | | 8 (12) ** | 4 ** | - | 8 | √, Z01-Z04 | - | 44 + 4 ** |
| CLW-02/20VR-D | | 8 (12) ** | 4 ** | 8 | - | √, Z01-Z04 | MODBUS incorporado | 44 + 4 ** |
| CLW-02/20VT-D | | 8 (12) ** | 4 ** | - | 8 | √, Z01-Z04 | MODBUS incorporado | 44 + 4 ** |
| CLW-02/20HR-12D | 12 Vcc | 8 (12) ** | 4 ** | 8 | - | √, Z01-Z04 | - | 44 + 4 ** |
| CLW-02/10HR-A | 100 – 240 Vca | 6 | - | 4 | - | √, Z01-Z04 | - | 34 + 4 ** |
| CLW-02/20HR-A | | 12 | - | 8 | - | √, Z01-Z04 | - | 44 + 4 ** |

Tabela 2 – Especificações do controlador Clic-02 (2/2)

| Modelo | Tensão de Alimentação | Entradas | | Saídas | | |
|--------------|-----------------------|---|-----------|--------|------------|-----------|
| | | Digital | Analogica | Relé | Transistor | Analogica |
| CLW-02/8ER-A | 100 – 240Vca | 4 | - | 4 | - | - |
| CLW-02/8ER-D | 24 Vcc | 4 | - | 4 | - | - |
| CLW-02/8ET-D | | 4 | - | - | 4 | - |
| CLW-02/4AI | | - | 4 | - | - | - |
| CLW-02/4PT | | - | 4 | - | - | - |
| CLW-02/2AO | | - | - | - | - | 2 |
| CLW-02/MBUS | | Módulo de Comunicação, RS-485, Escravo ModBus RTU | | | | |
| CLW-02/DNET | | Módulo de Comunicação, Escravo DeviceNet | | | | |
| CLW-02/PBUS | | Módulo de Comunicação, Escravo Profibus-DP | | | | |
| CLW-02/EN01 | | Módulo de Comunicação, TCP/IP | | | | |

Outra característica é em relação à facilidade de programação de suas funções, programação esta podendo ser feita utilizando o display do CLP ou através do programa CLIC02 Edit via cabo serial utilizando um computador. Neste documento o CLP utilizado será o CLIC02 20VT-D, possuidor do protocolo MODBUS incorporado, possibilitando sua comunicação com o computador através do mesmo.

1.3 Interface Homem-Máquina

Interface homem-máquina ou IHM pode ser definida uma interface entre o usuário e a máquina, como o próprio nome diz. O termo em si pode ser muito amplo e esta interface ser encontrada nos mais variados equipamentos, como, por exemplo, aparelhos de MP3, computadores, utensílios domésticos, etc. Entretanto quando se fala em IHM no âmbito de processos de produção e controle o termo se torna muito mais específico. Uma interface homem-máquina neste caso provê a representação visual do sistema controlado, com aquisição de dados em tempo real, sendo então uma importante ferramenta no aumento da produtividade.

Sendo então uma unidade de controle centralizada na linha de produção, elas podem ser equipadas com históricos de eventos, database, visualização de câmeras, controladores de eventos, etc, para que o usuário possa acessar o sistema a qualquer momento, com qualquer propósito. Normalmente estas IHMs são integradas a CLPs, já que estes são responsáveis por adquirir toda a informação dos sensores, transformar os dados obtidos e dar entrada neles na IHM.

1.3.1 Tipos de IHM

Dentro de uma empresa a IHM pode ter diversas formas e aplicações, como substituir um painel repleto de botões, exibir dados e planilhas do processo produtivo, ou conter sumários e informações a cerca de toda empresa de forma centralizada. Dessa forma ela pode desempenhar de forma muito mais organizada o controle de todo o processo fabril, centralizando todos os botões com suas respectivas funções em apenas um lugar, conter constante feedback do sistema acerca do processo produtivo, com capacidade de emitir relatórios a qualquer momento, além de poder oferecer gráficos, análises de rendimento e resumos de todo o processo produtivo. Junto a isso, ela pode desempenhar ainda papel importante na manutenção dos equipamentos e ferramentas, podendo apontar qual setor está parado ou em déficit, agilizando seu reparo e contribuindo para manutenção preventiva dos equipamentos.

1.3.2 Funcionamento e Aplicações da IHM

Basicamente a IHM integra vários conjuntos que são necessários para que ela desempenhe seu papel de forma eficiente. A linha de produção consiste em todo maquinário que faz o trabalho requerido para a confecção do produto, por trás disso há uma série de sensores monitorando temperatura, velocidade, pressão, peso e velocidade de abastecimento. Controlando todo esse processo estão os CLPs, que recebem dados destes sensores de entrada/saída e transformam em dados, combinações lógicas, que por fim, alimentam a IHM.

Devido sua aplicabilidade IHMs são usadas nos mais diversos ramos industriais, como alimentício, farmacêutico, utilitários, de maquinário pesado, entre outros. CLPs juntamente com IHMs são o coração da linha de produção neste tipo de indústria. A evolução possibilitada pela capacidade de supervisão, aquisição de dados e controle de todas as partes do sistema na mão do operador. Por exemplo numa metalúrgica uma IHM pode controlar como os metais são cortados e moldados e com que velocidade fazer isso, útil para qualquer ramo industrial ela pode fazer o controle e reposição de estoques a fim de otimizar a relação com os fornecedores. De forma geral, ela pode proporcionar o controle de todos os aspectos da linha de produção, isto é, velocidade, eficiência, detecção e correção de erros.

1.3.4 Vantagens e Conveniências

A grande vantagem da IHM é sua interface gráfica de fácil entendimento e adaptação. Possuindo cores de identificação, imagens, ícones, a fim de facilitar, mesmo para leigos, a leitura da situação (por exemplo: vermelho para problemas), contornando situações até como a falta de conhecimento e formação por parte do operador. Devido suas características estas são capazes de aumentar potencialmente a margem de lucro e diminuir custos e perdas no processo produtivo.

Antes as indústrias dependiam apenas do CLP para o controle da sua linha de produção, não possuindo retorno em tempo real do que está acontecendo no chão da fábrica, não podendo acionar/desligar alarmes ou modificar cada ramo do sistema sem ter que reprogramar o CLP manualmente. A capacidade da IHM de poder ser funcional em tantas camadas do processo, concentrando todas as etapas do processo de automação industrial em apenas um determinado lugar, podendo este ter seu design escolhido conforme o gosto do usuário, tornam dela uma ferramenta chave.

1.3.5 Flexibilidade da Interface

Uma das maiores conveniências que a IHM possui é sua capacidade de ser personalizável para atender o que o operador quer/precisa visualizar. Ela pode dar suporte a aplicações da maior das complexidades, contendo então múltiplas telas e janelas, com várias rotinas rodando simultaneamente, ou aplicações mais simples, destinadas a áreas da indústria em que poucas, mas cruciais, informações são necessárias. Cada IHM possui suas peculiaridades, elas podem conter sons, vídeos, controle de acesso remoto, etc.

O importante é que cada interface seja projetada de forma a otimizar da melhor forma possível o setor em que ela é empregada, levando em consideração quem irá fazer as leituras e utilizar as informações, assim como todo o contexto em que ela está inserida, como ruídos, iluminação, poeira e outros riscos.

1.4 Expansão das Redes Industriais: Domótica

Com o constante avanço das redes industriais, suas inegáveis vantagens e facilidades são cobiçadas por outros ramos. Um deles é o da domótica ou também conhecida como

automação residencial. A automação residencial visa integrar e comunicar todos os dispositivos eletrônicos de uma casa. Apesar do elevado preço as chamadas “casas do futuro” vêm fazendo enorme sucesso ao redor do mundo, principalmente em exposições, sendo um cobiçado bem de consumo. O conceito de domótica visa gerir todos os recursos habitacionais, como iluminação, climatização e segurança, através de um sistema que integre todos eles.

Através da integração de tecnologias de informação com o ambiente da casa, os sistemas e aplicativos podem se comunicar de forma integrada resultando em vários benefícios, como segurança, economia de energia e conforto para o usuário. Sabendo que o número de dispositivos de uma casa tende a aumentar com o tempo, interconexão e comunicação se mostram características úteis e desejadas. Por exemplo, sua geladeira manda uma mensagem quando precisa ser descongelada, ou a luz do seu quarto acende quando você entra. Caso ninguém esteja em casa o sistema de alarme é ativado, ligando para o dono e para os vizinhos caso alguém entre.

A automação residencial também pode proporcionar ao dono monitoramento remoto da casa, e de suas instalações, via internet. Com isso o proprietário pode controlar sua residência através de um navegador de internet ou de um smartphone.

1.5 Objetivos

Este trabalho tem como objetivo fazer a comunicação entre o CLP Clic02 VT-D da WEG e um computador via protocolo MODBUS incorporado neste controlador. Com isso, conceitos de redes industriais foram utilizados de forma a confeccionar uma IHM para aplicação no ramo da automação residencial.

1.6 Contribuição

O estabelecimento dessa comunicação proporciona inúmeras possibilidades na criação de interfaces gráficas através dos diversos softwares específicos para esta finalidade disponíveis para PC, sendo estas interfaces aplicáveis em diversos ramos, tanto industriais quanto residenciais, pois o Clic02 VT-D mesmo não possuindo a robustez de um CLP modular industrial, é capaz de desempenhar uma infinidade de aplicações em ambientes não tão agressivos. Juntamente a isso este trabalho proporciona ao Departamento de Engenharia

Elétrica da Universidade Federal de Viçosa novas possibilidades de pesquisa e extensão envolvendo este controlador em suas diversas finalidades.

2 Desenvolvimento

2.1 Protocolo MODBUS

2.1.1 Tipos de Modbus

O protocolo Modbus pode ser implementado por três tipos básicos: Modbus TCP, Modbus RTU, Modbus ASCII. A grande diferença entre o Modbus TCP e os outros dois (RTU, ASCII) é que ele se comunica através de redes Ethernet, enquanto os demais utilizam comunicação serial RS232 ou RS485.

Durante a fabricação do controlador é escolhido o tipo de Modbus que ele utiliza para se comunicar juntamente com outros parâmetros de configuração, no caso do Modbus serial, por exemplo, *baud rate*, *parity*, etc. O importante a se considerar é que em toda rede Modbus apenas um dos tipos pode ser utilizado.

2.1.2 Modbus RTU e ASCII

As diferenças básicas entre as comunicações serial Modbus RTU e ASCII é a forma de como as mensagens são construídas, como a informação ocupa o frame de mensagem e como esta mensagem é decodificada.

Em Modbus ASCII as mensagens são codificadas utilizando seu valor em hexadecimal, representadas por caracteres ASCII. Os caracteres então permitidos são números de 0-9 e letras de A-F. Como os bytes de informação serial utilizados pelo protocolo Modbus são de 16 bits, duas letras em hexadecimal são necessárias, portanto, para cada byte de informação, é necessário enviar dois bytes entre mestre/escravo. Já o modo Modbus RTU trabalha no sistema binário e para cada byte de informação um byte é trocado entre mestre/escravo.

Outra característica é que as mensagens são construídas de forma que quem as recebe facilmente identifica o fim da mensagem e o início da próxima. No modo ASCII o início da mensagem é marcado com “:” e o fim com uma combinação CR/LF (em outras palavras enviando o código referente a pressionar a tecla ENTER duas vezes). Já o modo RTU utiliza

gaps de silêncio para indicar o início e término da mensagem. Antes de ser enviada a mensagem deve haver ao menos 3,5 caracteres de silêncio em relação à mensagem anterior e para se preparar para receber uma mensagem um dispositivo receptor limpa seu buffer após 1.5 caracteres de silêncio, sendo que este tempo de silêncio de 1 caractere varia conforme o *baud rate* utilizado na comunicação. Uma grande diferença que estas características proporcionam é que o modo ASCII aceita gaps de até 1 segundo, enquanto o modo RTU trabalha sobre regime contínuo de mensagem.

2.1.3 Modbus RTU

Os controladores que utilizam o Modbus RTU (*Remote Terminal Unit*), constroem suas mensagens de forma que cada byte de 8-bits carrega dois caracteres de 4-bits em hexadecimal. Dessa forma há uma maior densidade de informação fazendo com que este modo seja capaz de transmitir mais informação que o modo ASCII utilizando o mesmo *baud rate*. Segue na Figura 5 abaixo um frame básico de mensagem, contendo endereço do escravo, o comando a ser executado, uma quantidade variável de dados e uma verificação de consistência de dados (CRC).

| ADDRESS | FUNCTION | DATA | CRC CHECK |
|---------|----------|-------------------|-----------|
| 8 BITS | 8 BITS | $n \times 8$ BITS | 16 BITS |

Figura 5 – Frame de mensagem padrão

Os caracteres permitidos são hexadecimais de 0-9, A-F, e esta mensagem deve ser enviada em um fluxo contínuo, respeitando o tempo de 3,5 caracteres para o início da mensagem e 1,5 caracteres para o fim, de forma que caso haja um intervalo maior que o tempo de 1,5 caracteres o dispositivo receptor descarta a mensagem incompleta e já entende que o próximo byte que chegará até ele é o byte de endereço da próxima mensagem. De forma análoga, caso uma nova mensagem chegue antes do tempo de 3,5 caracteres o dispositivo receptor irá entender que esta mensagem é uma continuação da mensagem anterior, o que gerará um erro.

Uma outra peculiaridade do Modbus RTU é que ele é um protocolo que trabalha com RS485, possuindo apenas um canal para transmitir pergunta e resposta, havendo então a restrição de apenas um dispositivo enviar mensagens por vez, diferentemente do padrão RS232 onde é possível obter uma comunicação *full duplex*.

O campo de endereço

O campo de endereço da mensagem é composto por um byte de 8 bits, sendo que são válidos endereços de 0 – 247 em decimal. Dispositivos escravos podem possuir endereços entre 1 – 247, sendo o endereço 0 destinado a mensagens enviadas em modo *broadcast*, ou seja, para todos os escravos.

Quando um escravo recebe a mensagem, ele compõe a mensagem de resposta com o próprio endereço fazendo com que o mestre possa entender que escravo está respondendo a ele.

O campo da função

Assim como o campo do endereço, o campo da função é formado por um byte de 8 bits, aceitando valores entre 1 – 255 decimal. Estes códigos variam de dispositivo para dispositivo, alguns aceitando certas funções e outros não, dependendo da estrutura física e lógica do dispositivo e suas limitações.

Exemplos de ações são fazer escrita e leitura de registradores, entradas e saídas, sendo esta leitura e escrita podendo ser feita em um registrador de cada vez ou vários registradores ao mesmo tempo, dependendo apenas do tipo de código de função utilizado.

Quando um escravo responde a pergunta do mestre, ele usa o campo de função para informar se a resposta enviada é uma resposta normal (que não contém erros) ou se algum erro ocorreu durante a transmissão. Para uma resposta sem erros, o dispositivo escravo apenas repete o código de função utilizado para a pergunta, caso um erro tenha acontecido o escravo envia um código de exceção, e este código pode ser entendido através de valores pré-definidos de forma ao mestre ser informado de qual tipo de erro ocorreu.

O campo de dados

O campo de dados é formado utilizando dois caracteres em hexadecimal, cada um deles variando entre 00 e FF. Este campo de dados tem seu tamanho variável, depende do tipo de ação que o mestre enviou para o dispositivo escravo. Dependendo da função utilizada este

campo de dados é responsável por conter outros parâmetros como número de bytes enviados na resposta, endereço de registradores ou quantidade de registradores envolvidos. Por exemplo, para um dispositivo mestre requisitar a leitura de um grupo de registradores de um dispositivo escravo, nesta pergunta deve conter o endereço do primeiro registrador a ser lido e também quantos registradores são requisitados na leitura. Outro exemplo pode ser a escrita de um grupo de registradores, onde a pergunta do mestre deve conter o endereço do primeiro registrador, em quantos registradores ele quer escrever, a contagem de bytes de dados que serão enviados para escrita e todos os dados que devem ser escritos nos especificados registradores.

Campo de checagem de erros

No modo RTU a checagem de erros é feita através de dois bytes de 8-bits. Estes bytes contém o valor da verificação de redundância cíclica (*Cyclic redundancy check – CRC*), que é um método para identificação de erros normalmente utilizada em redes digitais binárias. Este método de verificação é bastante popular devido sua simplicidade de implementação e por sua excelente (0,004% de erro) capacidade de detectar erros comuns causados por ruído nas redes. O método de implementar este algoritmo é apresentada no manual *Modicon Modbus Protocol Reference Guide* em http://modbus.org/docs/PI_MBUS_300.pdf.

Códigos de função aplicáveis ao CLP Clic-02

Vários são os códigos de função possíveis no protocolo Modbus RTU, cabendo ao equipamento limitar quais deles este suporta. O CLP abordado neste trabalho, Clic-02/VT-D, é capaz de desempenhar as seguintes ações: Leitura de múltiplos bits, leitura de múltiplos registradores, escrita de um único bit, escrita de um único registrador e escrita de múltiplos registradores. (WEG, 2009)

2.1.4 Estrutura de perguntas e respostas no protocolo Modbus RTU

2.1.4.1 Leitura de bits

Este código de função possui o valor de 01 Hex e é responsável por ler o estado ON/OFF (0 ou 1) de bits no dispositivo escravo.

A pergunta

A pergunta a ser feita deve especificar o endereço de início da leitura e quantos bits serão lidos a partir do endereço pré-determinado. Bits são endereçados começando em zero, portanto os bits 1-16 são endereçados em 0-15. Segue na Tabela 3 abaixo o exemplo de como ler os bits de 20-70 do dispositivo escravo número 18.

Tabela – 3: Pergunta para leitura dos bits 20-70 no dispositivo 18.

| Campo | Bytes de 4-bits (Hex) |
|---|------------------------------|
| Endereço escravo | 12 |
| Código de função | 01 |
| Endereço de início (Superior) | 00 |
| Endereço de início (Inferior) | 14 |
| Número de endereços (Superior) | 00 |
| Número de endereços (Inferior) | 32 |
| Verificação de Erros (CRC16) (Superior) | FF |
| Verificação de Erros (CRC16) (Inferior) | 78 |

Portanto, a pergunta seria enviada desta forma, respeitando o frame padrão mostrado na Figura 5: um byte de 8 bits para pergunta, um byte de 8 bits para o código de função e n bytes de 8 bits para dados, em que, neste caso, n vale 4. Lembrar que, conforme dito anteriormente, os 8 bits de dados em binário devem ser escritos utilizando 2 caracteres em hexadecimal. O formato da pergunta então seria: 12 01 00 14 00 32 FF 78.

A resposta

A resposta possui o endereço do escravo que foi perguntado, juntamente com o eco da função. Na sequência vem a contagem de bytes necessários para escrever todo o frame de dados da resposta.

A resposta de uma pergunta feita com o código de função 01 possui, para cada byte de dados de resposta, um bit designado para cada bit de memória do escravo, ou seja, a resposta é entregue em bytes de dados de 8 bits, podendo possuir então o estado de 8 bits de memória do dispositivo escravo. Os estados, ON/OFF, são mostrados, respectivamente, por 1 e 0. Caso seja requisitado o estado de menos de 8 bits do escravo, o byte de dados é completado com 0's. O que quer dizer que se são requeridos, por exemplo, 5 bits de memória do escravo, bit0 ao bit4 e os estados dos bits bit4-bit3-bit2-bit1-bit0 são, respectivamente, ON-OFF-OFF-ON-ON, o valor da resposta em hexadecimal será 13h, ou seja, 00010011 em binário. O bit especificado como endereço de início na pergunta ocupa a posição LSB do byte.

Caso mais de 8 bits sejam requisitados a lógica da resposta permanece a mesma com os endereços e estados dos bits se mostrando de forma crescente e possuindo o sentido do MSB → LSB na resposta. A resposta à pergunta construída conforme mostra a Tabela 3 é um exemplo de como ficaria a resposta para uma pergunta requisitando mais do que 8 bits (no caso a pergunta requisita 50 bits!):

Tabela – 4: Resposta para leitura dos bits 20-70 no dispositivo 18.

| Campo | Bytes de 4-bits (Hex) |
|---|------------------------------|
| Endereço do escravo | 12 |
| Código da função | 01 |
| Número de Bytes | 07 |
| Dados (Bits 27-20) | CD |
| Dados (Bits 35-28) | 2F |
| Dados (Bits 43-36) | 01 |
| Dados (Bits 51-44) | AB |
| Dados (Bits 59-52) | B2 |
| Dados (Bits 65-58) | 6C |
| Dados (Bits 70-66) | 1B |
| Verificação de Erros (CRC16) (Superior) | 57 |
| Verificação de Erros (CRC16) (Inferior) | B5 |

Os estados dos bits 27-20 são mostrados pela conversão do valor em hexadecimal CD em binário, 1100 1101. Portanto, o estado dos bits 27-20 é: ON-ON-OFF-OFF-ON-ON-OFF-ON, de forma que o bit 27 se encontra no MSB do byte e o bit 20 no LSB do byte. Já no último byte, 1B em hexadecimal ou 0001 1011 em binário, o estado de 5 bits apenas são requisitados, do bit 66 ao 70, de forma que o resto do byte é completado por 0's. O estado das variáveis 70-66 são então ON-ON-OFF-ON-ON. Portanto a resposta dada pelo dispositivo escravo 18 à pergunta feita conforme mostra a Tabela 3 tem o seguinte formato: 12 01 07 CD 2F 01 AB B2 6C 1B 57 B5.

2.1.4.2 Leitura de registradores

Este código de função possui o valor de 03 Hex e faz a leitura dos registradores do dispositivo, fazendo a leitura de words de 16 bits.

A pergunta

A pergunta se organiza de forma que o campo de dados deve conter qual endereço irá se iniciar a leitura seguido de quantos registradores serão lidos, sendo que para cada uma

dessas informações 2 bytes de 8 bits são utilizados, ou seja, 2 conjuntos de 2 caracteres em hexadecimal por informação. Assim como os bits, registradores são endereçados começando em zero, portanto os registradores de 1-8 são endereçados 0-7.

Na tabela abaixo segue um exemplo da leitura dos registradores 1555 até 1557 do dispositivo escravo 25.

Tabela – 5: Pergunta para leitura dos registradores 1555-1557 no dispositivo escravo 25.

| Campo | Bytes de 4-bits (Hex) |
|---|------------------------------|
| Endereço escravo | 19 |
| Código de função | 03 |
| Endereço de início (Superior) | 06 |
| Endereço de início (Inferior) | 13 |
| Número de endereços (Superior) | 00 |
| Número de endereços (Inferior) | 03 |
| Verificação de Erros (CRC16) (Superior) | F7 |
| Verificação de Erros (CRC16) (Inferior) | 5E |

Portanto a pergunta feita tem o seguinte formato: 19 03 06 13 00 03 F7 5E.

A resposta

A resposta irá conter o número do dispositivo que está respondendo a pergunta, juntamente com um eco da função. No frame de dados, o primeiro byte irá dizer o número de bytes necessários para escrever a resposta, lembrando que cada registrador possui 16 bits ou seja 2 bytes de 8 bits. Para o exemplo dado conforme mostra a Tabela 5, a resposta do escravo teria a seguinte forma:

Tabela – 6: Resposta para leitura dos registradores 1555-1557 no dispositivo escravo 25.

| Campo | Bytes de 4-bits (Hex) |
|---|------------------------------|
| Endereço do escravo | 19 |
| Código da função | 03 |
| Número de Bytes | 06 |
| Dados (Superior – Registrador 1555) | 00 |
| Dados (Inferior – Registrador 1555) | 00 |
| Dados (Superior – Registrador 1556) | 00 |
| Dados (Inferior – Registrador 1556) | EC |
| Dados (Superior – Registrador 1557) | 13 |
| Dados (Inferior – Registrador 1557) | CB |
| Verificação de Erros (CRC16) (Superior) | 9A |
| Verificação de Erros (CRC16) (Inferior) | F4 |

Portanto a resposta dada pelo dispositivo escravo 25 é 19 03 06 00 00 00 EC 13 CB 9A F4. Essa resposta nos diz que o conteúdo do registrador 1555 é 0, do registrador 1556 é 00 EC ou 236 em decimal e a do registrador 1557 é 13 CB em hexadecimal, 5067 em decimal e 00010011 11001011 em binário.

2.1.4.3 Escrita de um único bit

Este código de função possui o valor 05 Hex e ele escreve o estado de um bit de memória no dispositivo escravo.

A pergunta

A pergunta é feita preenchendo o frame de dados com o endereço do bit que se deseja escrever e qual valor se deseja escrever, para cada especificação são usados 2 bytes de 8 bits. Os estados ON e OFF são setados da seguinte forma: FF 00 hex seta o bit para ON, enquanto 00 00 hex seta o bit para OFF. Todos outros valores não são aceitos e não surtem efeito sobre o bit. Segue abaixo um exemplo de como setar para ON o bit 11136 no dispositivo 78:

Tabela – 7: Pergunta para escrita do bit 11136 no dispositivo 78.

| Campo | Bytes de 4-bits (Hex) |
|---|------------------------------|
| Endereço escravo | 4E |
| Código de função | 05 |
| Endereço de início (Superior) | 2B |
| Endereço de início (Inferior) | 80 |
| Dados(Superior) | FF |
| Dados (Inferior) | 00 |
| Verificação de Erros (CRC16) (Superior) | 8A |
| Verificação de Erros (CRC16) (Inferior) | 09 |

Portanto a pergunta feita pelo dispositivo mestre ao dispositivo escravo 78 é: 4E 05 2B 80 FF 00 8A 09.

A resposta

A resposta para o código de função de escrita é simplesmente um eco da pergunta. Portanto, para de pergunta feita conforme a Tabela 7 é obtido como resposta:

Tabela – 8: Resposta para escrita do bit 11136 no dispositivo 78.

| Campo | Bytes de 4-bits (Hex) |
|-------------------------------|------------------------------|
| Endereço escravo | 4E |
| Código de função | 05 |
| Endereço de início (Superior) | 2B |
| Endereço de início (Inferior) | 80 |

| | |
|---|----|
| Dados(Superior) | FF |
| Dados (Inferior) | 00 |
| Verificação de Erros (CRC16) (Superior) | 8A |
| Verificação de Erros (CRC16) (Inferior) | 09 |

Com isso a resposta obtida pelo mestre foi um eco de sua pergunta: 4E 05 2B 80 FF 00 8A 09.

2.1.4.4 Escrita de um único registradore

Este código de função possui o valor 06 Hex e faz a ação de escrever sobre um registrador do dispositivo.

A pergunta

Similar à estrutura de escrita sobre um único bit, na escrita sobre um único registrador o frame de dados deve conter o endereço do registrador e o que escrever nele. Vale a pena ressaltar que cada registrador possui 2 bytes de 8 bits de tamanho sendo necessário 2 grupos de 2 caracteres hexadecimais para escrever os 16 bits.

A Tabela 9 abaixo demonstra um exemplo de como escrever o valor 31546 no registrador 2832 do dispositivo 7.

Tabela – 9: Pergunta para escrita do valor 31546 no registrador 2832 do dispositivo #7.

| Campo | Bytes de 4-bits (Hex) |
|---|------------------------------|
| Endereço escravo | 07 |
| Código de função | 06 |
| Endereço de início (Superior) | 0B |
| Endereço de início (Inferior) | 10 |
| Dados(Superior) | 7B |
| Dados (Inferior) | 3A |
| Verificação de Erros (CRC16) (Superior) | 28 |
| Verificação de Erros (CRC16) (Inferior) | AE |

A resposta

A resposta à escrita é apenas um eco da pergunta após o registrador ter seu valor alterado com sucesso. Assim, a resposta para uma pergunta feita conforme a Tabela 9 seria:

Tabela – 10: Resposta para escrita do valor 31546 no registrador 2832 do dispositivo #7.

| Campo | Bytes de 4-bits (Hex) |
|---|------------------------------|
| Endereço escravo | 07 |
| Código de função | 06 |
| Endereço de início (Superior) | 0B |
| Endereço de início (Inferior) | 10 |
| Dados(Superior) | 7B |
| Dados (Inferior) | 3A |
| Verificação de Erros (CRC16) (Superior) | 28 |
| Verificação de Erros (CRC16) (Inferior) | AE |

2.1.4.5 Escrita de Múltiplos Registradores

Este código de função possui o valor 10 Hex e escreve sobre uma sequência de registradores.

A pergunta

Para este código de função o campo de dados deve conter várias informações, que são ordenadas com a seguinte sequência: endereço de onde começará a escrita, número de registradores que serão escritos, número de bytes que serão alterados e dados a serem alterados nos registradores desejados especificados individualmente. Vale ressaltar que os dados são ordenados de forma que os primeiros dados escritos irão para o endereço que iniciará a escrita e assim sucessivamente.

Na Tabela 11 abaixo é mostrado um exemplo de como escrever, no dispositivo escravo 1, os dados 00 AA e 2B 47 nos registradores 7 e 8, respectivamente.

Tabela – 11: Pergunta para escrita nos registradores 7 e 8 do dispositivo 1.

| Campo | Bytes de 4-bits (Hex) |
|---|------------------------------|
| Endereço do escravo | 01 |
| Código de função | 10 |
| Endereço de início (Superior) | 00 |
| Endereço de início (Inferior) | 07 |
| Número de endereços (Superior) | 00 |
| Número de endereços (Inferior) | 02 |
| Contagem de Bytes | 04 |
| Dados (Registrador 7 - Superior) | 00 |
| Dados (Registrador 7 - Inferior) | AA |
| Dados (Registrador 8 - Superior) | 2B |
| Dados (Registrador 8 - Inferior) | 47 |
| Verificação de Erros (CRC16) (Superior) | D2 |
| Verificação de Erros (CRC16) (Inferior) | 23 |

A resposta

A resposta à escrita de múltiplos registradores contém no seu frame de dados o endereço de início da escrita, juntamente com o número de registradores que foram escritos no escravo. Para o exemplo da pergunta enviada conforme a Tabela 11 o escravo 01 enviaria a seguinte resposta:

Tabela – 12: Resposta para escrita nos registradores 7 e 8 do dispositivo 1.

| Campo | Bytes de 4-bits (Hex) |
|---|------------------------------|
| Endereço escravo | 01 |
| Código de função | 10 |
| Endereço de início (Superior) | 00 |
| Endereço de início (Inferior) | 07 |
| Números de endereços(Superior) | 00 |
| Números de endereços (Inferior) | 02 |
| Verificação de Erros (CRC16) (Superior) | F0 |
| Verificação de Erros (CRC16) (Inferior) | 09 |

2.1.5 Aplicação do protocolo ao Clic-02 VT-D

Ligação com o computador

A conexão entre computador e controlador foi feita utilizando um cabo de duas vias RS485, um conversor RS485 para RS232 da marca Hexin juntamente com um cabo serial DB9 RS232. A figura 6 abaixo demonstra a ligação:

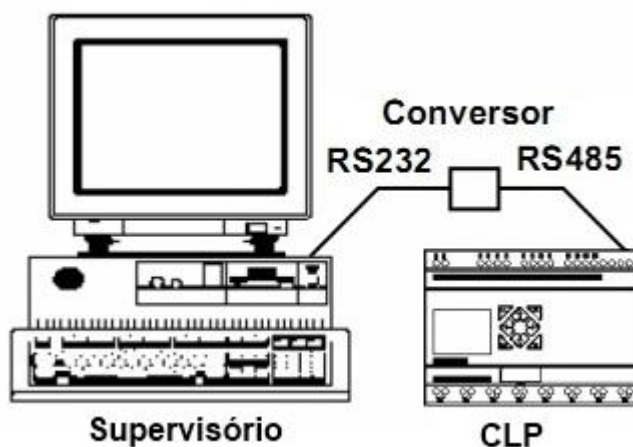


Figura 6: Ligação computador e CLP

Para fazer esta conexão são necessários certos ajustes tanto no controlador, quanto no computador supervisor, ou seja, ambos devem estar enviando/recebendo dados configurados utilizando o mesmo *baud rate*, número de bits de paridade e número de *stop bits*. O Clic-02 VT-D só se comunica por Modbus RTU se especificamente configurado com os seguintes parâmetros:

Tabela – 13: Parâmetros de configuração obrigatórios para comunicação.

| Parâmetros | Valores |
|------------|---------|
| E/S Remota | Não |
| ID | 01~99 |
| Modo | 8/N/2 |
| Baud Rate | 38400 |

Todas estas configurações podem ser implementadas utilizando o software Clic02 Edit ou diretamente na interface do CLP. Para configurar utilizando o software, basta acessar a aba **Operação/Configuração do Sistema** na barra de menu e configurar os parâmetros conforme a Tabela 13 específica.

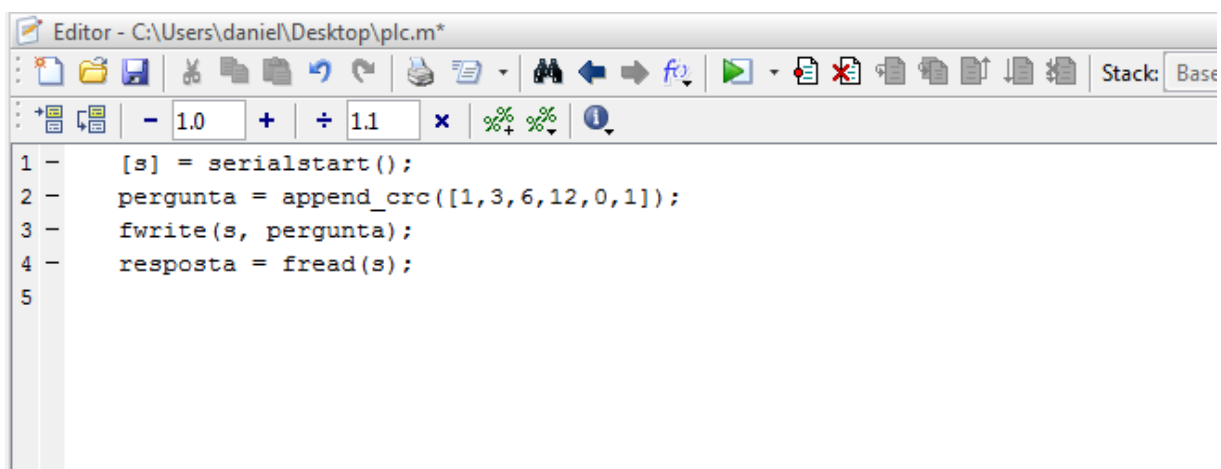
Figura 7 – Configurações do Sistema

O software de comunicação

O software de comunicação a ser utilizado no computador fará o papel mestre RTU, ou seja, ele irá enviar perguntas ao escravo e receber suas respostas. Como os escravos são dispositivos como CLP's, a leitura e escrita da memória destes dispositivos torna possível a elaboração de um sistema supervisório no mestre. Inúmeros softwares podem fazer este papel, papel este que gira em torno de executar ações para abrir a comunicação serial com o escravo, fazer a pergunta, obter a resposta e tratar essa resposta criando gráficos, bancos de dados, mostradores de condição e todas outras características que uma IHM pode conter.

O script a seguir utiliza a estrutura explicada acima. O comando *serialstart* faz a abertura de comunicação serial, trazendo dentro das suas configurações o *baud rate*, modo de paridade e *stop bits*. Conforme visto na anteriormente tanto o CLP quanto o computador devem comunicar utilizando os mesmos parâmetros, e, no caso do CLP Clic-02, estes são limitados conforme listado pela Tabela 13. Já a pergunta é montada através do script *append_crc*, que faz o cálculo do CRC-16 necessário para ser enviado na mensagem. O comando *fwrite* é responsável por enviar a pergunta ao escravo e o comando *fread* é responsável por mostrar o valor obtido na resposta.

Os valores que são utilizados na estrutura da função *append_crc* e os obtidos com *fwrite* e *fread* são conversões dos valores típicos das perguntas e respostas em hexadecimal, mostrado nos exemplos da seção 2.1.4, para decimal.



```
Editor - C:\Users\daniel\Desktop\plc.m*
[Icons: Save, Print, Copy, Paste, Undo, Redo, Find, Run, Stop, Refresh, etc.]
Stack: Base

1 - [s] = serialstart();
2 - pergunta = append_crc([1,3,6,12,0,1]);
3 - fwrite(s, pergunta);
4 - resposta = fread(s);
5
```

Figura 8 – Exemplo de software mestre

O script mostrado na Figura 8 portanto faz a pergunta ao dispositivo 1, com código de função 3 (leitura de registradores), com o endereço do registrador de onde se iniciará a leitura 060C (conforme dito, o argumento do comando *append_crc* são os valores em hexadecimal agrupados dois a dois convertidos em decimal, portanto o terceiro elemento do argumento “6” representa 06 hex da mensagem e o quarto argumento “12” representa 0C hex) e quantos registradores devem ser lidos a partir do endereço de início (“0” representa 00 hex e “1” representa 01 hex).

Tabelas de endereços para bits e registradores

O manual Micro Controlador Programável Clic-02 encontrado em: <http://ecatalog.weg.net/files/wegnet/WEG-rele-programavel-clic-02-3rd-manual-portugues-br.pdf> possui todas as informações a cerca dos bits e registradores, seus endereços e que tipo de código de função o bit ou registrador aceita. As Tabelas 14 e 15 trazem as características de alguns registradores e bits, enquanto a tabela 16 e 17 traz o endereço de alguns registradores e bits que são utilizados nos exemplos a seguir.

Tabela 14: Variáveis digitais

| | Símbolo | Contatos + - -/+ | Quantidade | Faixa Válida |
|-----------------------------|---------|-------------------------|------------|--------------|
| Entrada Digital | I | I / i | 12 | I01 – I0C |
| Saída Digital | Q | Q / q | 8 | Q01 – Q08 |
| Entrada via Teclado | Z | Z / z | 4 | Z01 – Z04 |
| Entrada Digital de Expansão | X | X / x | 12 | X01 – X0C |
| Saída Digital de Expansão | Y | Y / y | 12 | Y01 – Y0C |
| Marcador Auxiliar | M | M / m | 63 | M01 – M3F |
| | N | N / n | 63 | N01 – N3F |
| Temporizador | T | T / t | 31 | T01 – T1F |
| Contador | C | C / c | 31 | C01 – C1F |
| RTC | R | R / r | 31 | R01 – R1F |
| Comparador | G | G / g | 31 | G01 – G1F |

Tabela 15: Variáveis analógicas

| | Símbolo | Quantidade | Faixa Válida |
|-------------------------|----------------|-------------------|---------------------|
| Entrada Analógica | A | 8 | A01 ~ A08 |
| Ganho Entrada Analógica | V | 8 | V01 ~ V08 |
| Temporizador | T | 31 | T01 ~ T1F |
| Contador | C | 31 | C01 ~ C1F |
| Entrada de Temperatura | AT | 4 | AT01 ~ AT04 |
| Saída Analógica | AQ | 4 | AQ01 ~ AQ04 |
| Adição-Subtração | AS | 31 | AS01 ~ AS1F |
| Multiplicação-Divisão | MD | 31 | MD01 ~ MD1F |
| PID | PID | 15 | PI01 ~ PI0F |
| Multiplexador de Dados | MX | 15 | MX01 ~ MX0F |
| Rampa Analógica | AR | 15 | AR01 ~ AR0F |
| Registrador de Dados | DR | 240 | DR01 ~ DR0F |

Tabela 16: Estado dos registradores

| Endereço Modbus | | Função Modbus | Conteúdo das Words | | | | | | | | | | | | | | | | |
|------------------------|--------------------|----------------------|---------------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----|
| Decimal | Hexadecimal | | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 1536 | 0600 | 03H | R10 | R0F | R0E | R0D | R0C | R0B | R0A | R09 | R08 | R07 | R06 | R05 | R04 | R03 | R02 | R01 | |
| 1537 | 0601 | | R1F | R1E | R1D | R1C | R1B | R1A | R19 | R18 | R17 | R16 | R15 | R14 | R13 | R12 | R11 | | |
| 1538 | 0602 | | G10 | G0F | G0E | G0D | G0C | G0B | G0A | G09 | G08 | G07 | G06 | G05 | G04 | G03 | G02 | G01 | |
| 1539 | 0603 | | G1F | G1E | G1D | G1C | G1B | G1A | G19 | G18 | G17 | G16 | G15 | G14 | G13 | G12 | G11 | | |
| 1540 | 0604 | | T10 | T0F | T0E | T0D | T0C | T0B | T0A | T09 | T08 | T07 | T06 | T05 | T04 | T03 | T02 | T01 | |
| 1541 | 0605 | | T1F | T1E | T1D | T1C | T1B | T1A | T19 | T18 | T17 | T16 | T15 | T14 | T13 | T12 | T11 | | |
| 1542 | 0606 | | C10 | C0F | C0E | C0D | C0C | C0B | C0A | C09 | C08 | C07 | C06 | C05 | C04 | C03 | C02 | C01 | |
| 1543 | 0607 | | C1F | C1E | C1D | C1C | C1B | C1A | C19 | C18 | C17 | C16 | C15 | C14 | C13 | C12 | C11 | | |
| 1544 | 0608 | | M10 | M0F | M0E | M0D | M0C | M0B | M0A | M09 | M08 | M07 | M06 | M05 | M04 | M03 | M02 | M01 | |
| 1545 | 0609 | | M20 | M1F | M1E | M1D | M1C | M1B | M1A | M19 | M18 | M17 | M16 | M15 | M14 | M13 | M12 | M11 | |
| 1546 | 060A | | M30 | M2F | M2E | M2D | M2C | M2B | M2A | M29 | M28 | M27 | M26 | M25 | M24 | M23 | M22 | M21 | |
| 1547 | 060B | | M3F | M3E | M3D | M3C | M3B | M3A | M39 | M38 | M37 | M36 | M35 | M34 | M33 | M32 | M31 | | |
| 1548 | 060C | | N10 | N0F | N0E | N0D | N0C | N0B | N0A | N09 | N08 | N07 | N06 | N05 | N04 | N03 | N02 | N01 | |
| 1549 | 060D | | N20 | N1F | N1E | N1D | N1C | N1B | N1A | N19 | N18 | N17 | N16 | N15 | N14 | N13 | N12 | N11 | |
| 1550 | 060E | | N30 | N2F | N2E | N2D | N2C | N2B | N2A | N29 | N28 | N27 | N26 | N25 | N24 | N23 | N22 | N21 | |
| 1551 | 060F | | 10H | - | N3F | N3E | N3D | N3C | N3B | N3A | N39 | N38 | N37 | N36 | N35 | N34 | N33 | N32 | N31 |
| 1552 | 0610 | | - | - | - | - | I0C | I0B | I0A | I09 | I08 | I07 | I06 | I05 | I04 | I03 | I02 | I01 | |
| 1553 | 0611 | | - | - | - | - | X0C | X0B | X0A | X09 | X08 | X07 | X06 | X05 | X04 | X03 | X02 | X01 | |
| 1554 | 0612 | | - | - | - | - | Y0C | Y0B | Y0A | Y09 | Y08 | Y07 | Y06 | Y05 | Y04 | Y03 | Y02 | Y01 | |
| 1555 | 0613 | | - | - | - | - | - | - | - | - | Q08 | Q07 | Q06 | Q05 | Q04 | Q03 | Q02 | Q01 | |
| 1556 | 0614 | | - | - | - | - | - | - | - | - | - | - | - | - | Z04 | Z03 | Z02 | Z01 | |
| 1557 | 0615 | | H10 | H0F | H0E | H0D | H0C | H0B | H0A | H09 | H08 | H07 | H06 | H05 | H04 | H03 | H02 | H01 | |
| 1558 | 0616 | | - | H1F | H1E | H1D | H1C | H1B | H1A | H19 | H18 | H17 | H16 | H15 | H14 | H13 | H12 | H11 | |
| 1559 | 0617 | | - | - | - | - | - | - | - | - | L08 | L07 | L06 | L05 | L04 | L03 | L02 | L01 | |
| 1560 | 0618 | | - | - | - | - | - | - | - | - | - | - | - | - | - | S01 | P02 | P01 | |
| 1561 | 0619 | | W10 | W0F | W0E | W0D | W0C | W0B | W0A | W09 | W08 | W07 | W06 | W05 | W04 | W03 | W02 | W01 | |
| 1562 | 061A | | W20 | W1F | W1E | W1D | W1C | W1B | W1A | W19 | W18 | W17 | W16 | W15 | W14 | W13 | W12 | W11 | |
| 1563 | 061B | | W30 | W2F | W2E | W2D | W2C | W2B | W2A | W29 | W28 | W27 | W26 | W25 | W24 | W23 | W22 | W21 | |
| 1564 | 061C | | W40 | W3F | W3E | W3D | W3C | W3B | W3A | W39 | W38 | W37 | W36 | W35 | W34 | W33 | W32 | W31 | |

Tabela 17: Endereço bit a bit das variáveis digitais

| Endereço Modbus | | Funções Suportadas | Conteúdo |
|-----------------|-------------|--------------------|----------|
| Decimal | Hexadecimal | | |
| 11008 ~ 11038 | 2B00 ~ 2B1E | 01H 05H | R01~R1F |
| 11040 ~ 11070 | 2B20 ~ 2B3E | | G01~G1F |
| 11072 ~ 11102 | 2B40 ~ 2B5E | | T01~T1F |
| 11104 ~ 11134 | 2B60 ~ 2B7E | | C01~C1F |
| 11136 ~ 11198 | 2B80 ~ 2BBE | | M01~M3F |
| 11200 ~ 11262 | 2BC0 ~ 2BFE | | N1~N3F |
| 11264 ~ 11275 | 2C00 ~ 2C0B | | I1~IC |
| 11280 ~ 2C1B | 2C10 ~ 2C1B | | X1~XC |
| 11296 ~ 11307 | 2C20 ~ 2C2B | | Y1~YC |
| 11312 ~ 11319 | 2C30 ~ 2C37 | | Q1~Q8 |
| 11328 ~ 11331 | 2C40 ~ 2C43 | | Z1~Z4 |

A relevância do valor do endereço em decimal é dar a noção de quantos endereços certo dispositivo possui, ou facilitar ao programador memorizar os endereços de memória de certo dispositivo, já que bits e registradores começam em 0 e vão até 65535. O dispositivo em questão não utiliza valores de registradores e bits mais baixos, começando em 0 por exemplo, pois o firmware utilizado por este é uma evolução de firmwares anteriores embarcados no mesmo hardware, fazendo com que, devido as novas funções implementadas, todo um remapeamento da memória tenha sido feito. Firmwares anteriores do Clic-02 VT-D possuía registradores e bits começando em 0.

Já o valor em hexadecimal é chave para a construção das mensagens. Como o endereço das mensagens são enviados em 16 bits, 4 caracteres em hexadecimal divididos 2 a dois, as tabelas já trazem este valor fazendo com que o operador apenas utilize os 2 primeiros dígitos do endereço no campo “Endereço de Início (Superior)” e os 2 dígitos finais no campo “Endereço de Início (Inferior)” conforme os exemplos da seção 2.1.4. Outra facilidade trazida pelo endereçamento hexadecimal é que, como os registradores possuem o estado de vários bits em uma única word, quando o usuário lê os dois primeiros dígitos em hexadecimal de uma resposta vinda de um escravo trazendo informações sobre o CLP, ele pode converter apenas os dois primeiros dígitos para binário e obter os estados dos bits por estes designado. O mesmo pode ser feito para os dois últimos dígitos.

Por exemplo, caso o usuário deseje fazer a leitura do estado das variáveis auxiliares N01~N10 (registrador de 16 bytes – endereço 1548 decimal / 060C hexadecimal) e a resposta obtida da pergunta feita ao escravo traga o valor 0C22 por exemplo, o usuário pode fazer a

leitura dos primeiros 8 registradores (N01~N08) convertendo os dois últimos dígitos (22 hex – 00100010 bin) e fazer a leitura dos estados destes 8 bits: OFF-OFF-ON-OFF-OFF-OFF-ON-OFF. Caso queira fazer a leitura dos outros 8 bits (N09~N10) basta converter os dois primeiros dígitos (0C hex – 00001100). Se a resposta fosse entregue em decimal ao invés de hexadecimal o resultado do valor do registrador seria 3106, que é a conversão de 0C22 hex para decimal, o que implica em uma conversão para binário, a fim de fazer as leituras dos bits, menos intuitiva e mais trabalhosa.

Exemplo: Leitura de Registrador no Clic-02

Neste primeiro exemplo será executada a leitura de um registrador utilizando a comunicação Modbus RTU. O registrador M01~M10 do PLC #01 será lido (16 bytes). O endereço deste registrador pode ser obtido via Tabela 16 e é 1544 decimal ou 0608 em hexadecimal. É utilizada a estrutura de pergunta explicada na seção 2.1.4.2.

Tabela – 18: Leitura registrador 1544 do dispositivo 01.

| Campo | Bytes de 4-bits (Hex) |
|---|------------------------------|
| Endereço escravo | 01 |
| Código de função | 03 |
| Endereço de início (Superior) | 06 |
| Endereço de início (Inferior) | 08 |
| Número de endereços (Superior) | 00 |
| Número de endereços (Inferior) | 01 |
| Verificação de Erros (CRC16) (Superior) | 7C |
| Verificação de Erros (CRC16) (Inferior) | 80 |

```

Editor - C:\Users\daniel\Desktop\plc.m
1 - [s] = serialstart();
2 - pergunta = append_crc([1,3,6,8,0,1]);
3 - fwrite(s, pergunta);
4 - resposta = fread(s,7);
5

serialstart.m x plc.m x append_crc.m x IHM_Principal.m x PartidaEstrelaTrianguloTimer.m

Command Window

resposta =

     1
     3
     2
     0
     0
    184
     68

fx >>

```

Figura 9: Leitura registrador 1544 do CLP Clic-02

De acordo com a resposta é possível verificar que o endereço do dispositivo e o código de função são ecos da pergunta, que foram lidos 2 bytes de 8 bits e que o valor destes bytes são 00 00 hex. Por último na mensagem vem o valor do CRC convertidos para decimal.

Exemplo: Escrita de Registrador no Clic-02

Neste segundo exemplo será feita a escrita do valor 01 FF hex (511 decimal) no registrador do exemplo anterior e depois será feita a leitura do mesmo, a fim de verificar se o valor escrito de fato pôde ser verificado na memória do dispositivo. As estruturas de pergunta seguem os exemplos da seção 2.1.4.2 e 2.1.4.4.

Tabela – 19: Escrita do valor 511 decimal no registrador 1544 do dispositivo 01

| Campo | Bytes de 4-bits (Hex) |
|---|-----------------------|
| Endereço escravo | 01 |
| Código de função | 06 |
| Endereço de início (Superior) | 06 |
| Endereço de início (Inferior) | 08 |
| Dados(Superior) | 01 |
| Dados (Inferior) | FF |
| Verificação de Erros (CRC16) (Superior) | 49 |
| Verificação de Erros (CRC16) (Inferior) | 50 |

```

Editor - C:\Users\daniel\Desktop\plc.m
[Icons: Save, Print, Copy, Paste, Undo, Redo, Run, Stop, etc.]
Stack: Bas

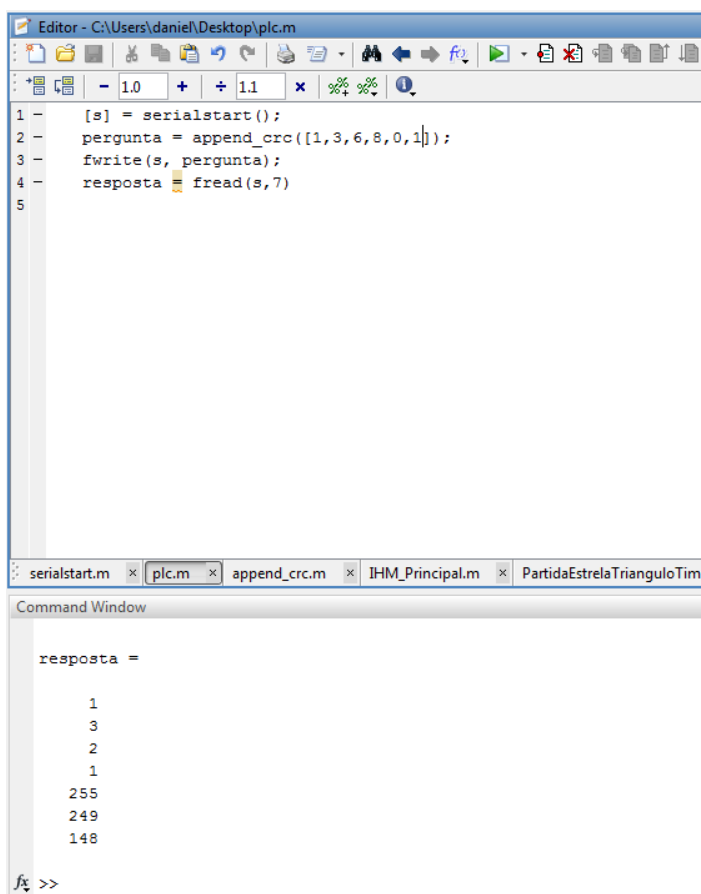
1 - [s] = serialstart();
2 - pergunta = append_crc([1,6,6,8,1,255]);
3 - fwrite(s, pergunta);
4 - resposta = fread(s,7)
5

```

Figura 10: Escrita do valor 511 decimal no registrador 1544 do dispositivo 01

Tabela – 20: Leitura do registrador 1544 do dispositivo 01.

| Campo | Bytes de 4-bits (Hex) |
|---|------------------------------|
| Endereço escravo | 01 |
| Código de função | 03 |
| Endereço de início (Superior) | 06 |
| Endereço de início (Inferior) | 08 |
| Número de endereços (Superior) | 00 |
| Número de endereços (Inferior) | 01 |
| Verificação de Erros (CRC16) (Superior) | 7C |
| Verificação de Erros (CRC16) (Inferior) | 80 |



```
Editor - C:\Users\daniel\Desktop\plc.m
[s] = serialstart();
pergunta = append_crc([1,3,6,8,0,1]);
fwrite(s, pergunta);
resposta = fread(s,7)
5

serialstart.m x plc.m x append_crc.m x IHM_Principal.m x PartidaEstrelaTrianguloTim

Command Window

resposta =

     1
     3
     2
     1
    255
    249
    148

fx >>
```

Figura 11: Leitura registrador 1544 do dispositivo 01.

Pode-se verificar na Figura 11 que na resposta o conteúdo que foi escrito está presente, já que esta resposta contém, na seguinte ordem, endereço do dispositivo, código de função (leitura), número de bytes de dados (dois), primeiros 8 bits de dados (1 em decimal ou 01 hex), últimos 8 bits de dados (255 em decimal ou FF hex) e CRC. Dessa forma, o valor escrito 01FF hex ou 511 decimal foi escrito com sucesso. Para o CLP, todos os bits de M01~M08 estão setados (00000000 11111111 em binário).

3 Aplicação em Domótica

De posse de uma ferramenta capaz de fazer a leitura de CLP's e gerar interfaces é possível criar modelos de IHM aplicáveis a inúmeros ramos do mercado. Este trabalho traz como aplicação uma IHM voltada à automação residencial.

A fim de produzir um protótipo de uma casa inteligente, foi montada a maquete de uma residência feita de madeira MDF. A casa a ser estudada é composta de quatro cômodos, uma varanda e os muros.

A automação irá cobrir os mais básicos, e principais, ramos da automação residencial: iluminação, segurança e temperatura. Também iremos trabalhar na simulação da irrigação do jardim, e na automação da porta da garagem. Uma interface homem-máquina irá mostrar ao proprietário o estado de todas essas variáveis e trará também possibilidade de controlar os diversos setores da planta, fazendo com que haja uma comunicação entre a casa e o dono.

3.1 Estrutura da casa automatizada e diagramas ladder

3.1.1 Iluminação

A iluminação foi dividida em dois subgrupos: iluminação interna e iluminação externa.

Para iluminação interna foi feito um sistema de tal forma que, dependendo da luminosidade exterior a casa, as luzes ficam mais fortes ou mais fracas, economizando energia. Para acendê-las é necessário o acionamento de um interruptor.

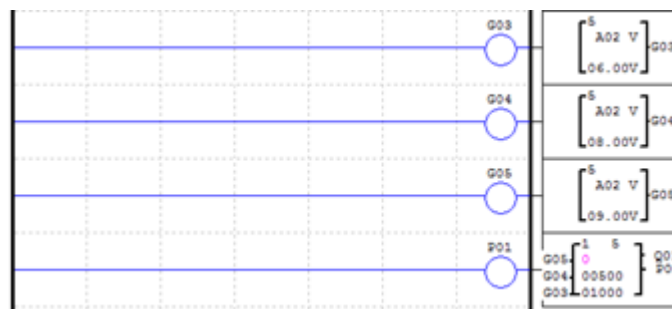


Figura 12: Ladder iluminação interna

Esse ladder utiliza duas funções do CLP, comparadores e PWM. As entradas G03, G04, G05 são entradas analógicas ligadas a um circuito LDR. Este circuito LDR capta a luz externa, e, através de um circuito divisor de tensão, a resistência variável do LDR é convertida em tensão analógica para o CLP. No controlador, essa tensão entra nos comparadores (três comparadores para a produção de três estágios de luminosidade) e dependendo do seu valor a função PWM é ativada com o duty cycle respectivo. Foram configurados três duty cycles diferentes, com ativação dependendo do estado de cada comparador: 10%, 50%, 100%, todos proporcionais a luz externa, i.e, aumento da resistência do LDR por falta de iluminação, e, portanto, maior valor de tensão de entrada nos comparadores.

Já a iluminação externa apresenta apenas dois modos de funcionamento: ligado e desligado. A iluminação externa é ativada assim que seja detectada uma luminosidade baixa por um circuito LDR. Este, diferente do anterior, utiliza um circuito transistorizado de forma que o circuito chaveia ao escurecer demais, ligando a luz na parte externa da casa. Segue abaixo as figuras 13 e 14, apresentando o circuito de transistor e o ladder aplicado:

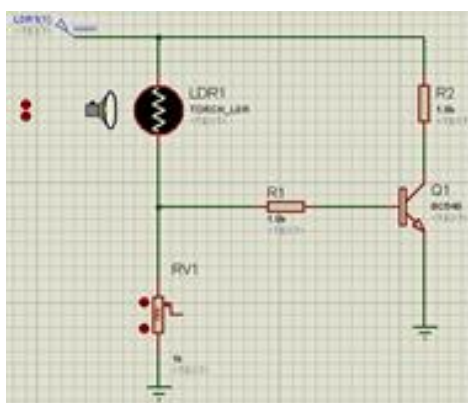


Figura 13: Circuito à transistor.



Figura 14: Ladder iluminação externa

3.1.2 Segurança

Nesta etapa foi feito um sistema de segurança que funciona da seguinte forma: o dono da casa, ao seu dispor, ativa o sistema de alarme da casa. Quando o alarme está ativado, um sensor de presença localizado no muro da casa detecta a presença indesejada, fazendo com que 3 LEDs vermelhos acendam, fazendo o papel da sirene do alarme. Para desligar a sirene, basta o proprietário desligar a chave de acionamento do alarme. Um novo acionamento não irá religar os LEDs, apenas rearmar o alarme.

Para fazer o papel do sensor, recorreu-se novamente ao circuito transistorizado utilizado na iluminação externa (Figura 13). No entanto dessa vez um LED alto brilho foi posicionado diretamente para o LDR, fazendo com que caso haja uma interrupção dessa luz, o transistor entre em corte, energizando os LEDs vermelhos.

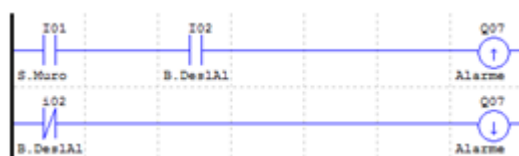


Figura 15: Ladder para o sistema de segurança

3.1.3 Temperatura

Para o controle da temperatura, foi acoplado ao sistema um sensor de temperatura LM35, capaz de entregar 10mV/°C. Essa característica dele serviu de entrada para o CLP, novamente na função comparador. Foram feitas duas rotinas, uma comparando a entrada do CLP referente à saída do LM35 ao valor 3V e outra comparando a mesma porta com o valor 2.5V. Em ambos os métodos foi aplicado um ganho de 10 vezes no valor da saída do LM35, uma manobra possível no Clic-02.

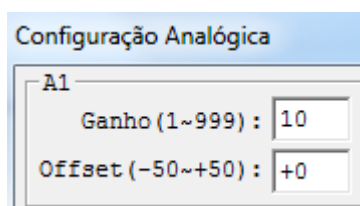


Figura 16: Configuração do ganho na entrada analógica A1

Uma outra característica do controlador utilizada aqui foi a utilização de dois métodos distintos de comparação. No primeiro caso, a função comparação energizaria a bobina de saída do CLP caso o valor de Volt do LM35 fosse maior a 3V, ou seja, caso a temperatura fosse maior do que 30°C. No entanto, através do comando SET, o valor da variável comparação não irá voltar ao estado baixo caso a temperatura venha a cair. A bobina energizada foi ligada a um cooler, responsável pelo resfriamento do ambiente.

Para que o cooler não funcionasse por tempo indeterminado, já que o comando SET mantém a variável em alto até que um comando RESET seja efetuado, foi programado um segundo comparador efetuando o RESET na bobina energizada quando o Volt do LM35 fosse menor do que 2.5V (25°C). Dessa forma o ambiente sempre estará com sua temperatura variando entre 25° e 30°C.



Figura 17: Ladder sistema de temperatura

3.1.4 Automações

Duas automações foram feitas, uma em relação à porta da garagem e outra no sistema de irrigação do jardim.

No jardim foi utilizada a função RTC (real time clock). Essa função energiza uma saída do CLP em uma determinada hora do dia. O nome real time clock vem dessa característica de utilizar o tempo real para executar a tarefa. Na maquete foi escolhido de 8:00h às 9:00h da manhã como sendo o tempo em que a irrigação ocorreria e um LED azul foi ligado representando o evento. Junto ao clock de tempo real uma chave foi adicionada, dando maior autonomia ao evento.



Figura 18: Ladder irrigação, com RTC ligado de domingo a domingo

Também do lado de fora da casa uma outra automação foi feita, esta em relação à garagem. Foi utilizado o sensor de presença a transistor também nesta etapa. Este sensor é responsável por detectar a presença do veículo que se aproxima, abrindo o portão da garagem para que ele entre, e, após 10s que ele tenha saído de frente do sensor, o circuito entende que ele entrou na garagem e que tudo ocorreu bem, fechando a porta.

No ladder abaixo I03 é o sensor da porta da garagem e Q02 e Q03 são os motores de abre portão e fecha portão, respectivamente. T01 é o timer responsável pelo tempo de 10s entre a operação do fechamento do portão.



Figura 19: Ladder porta da garagem

3.2 IHM

O sistema supervisorio confeccionado para a automação desta maquete consiste em apresentar o estado das automações, se estão ligadas ou desligadas, e também manipular o estado dos acionamentos monitorados e controlados pelo CLP. Tudo isso pode ser obtido

através da leitura de registradores e bits (códigos de função 01 e 03) e escrita de registradores e bits (códigos de função 05 e 06).

Este sistema supervisorio foi feito de forma a permitir que outras IHM's sejam carregadas no mesmo computador, dando a este uma maior generalidade. Após carregada a IHM deve-se escolher a opção **Início/Conectar CLP** e especificar a porta COM na qual o CLP está conectado ao computador.

Já carregada a IHM no menu **Ações** pode-se escolher colocar o CLP no modo **RUN/STOP**, ou seja, rodando o ladder e executando a automação ou não. Feito isso, basta clicar em **Ações/Iniciar Scan** para que a IHM passe a coletar os dados do CLP para informar ao usuário o estado das variáveis e também para interagir com o usuário caso ele escolha mudar o estado de alguma delas. Seguem as Figuras 20, 21 e 22 ilustrando o que foi explicado.

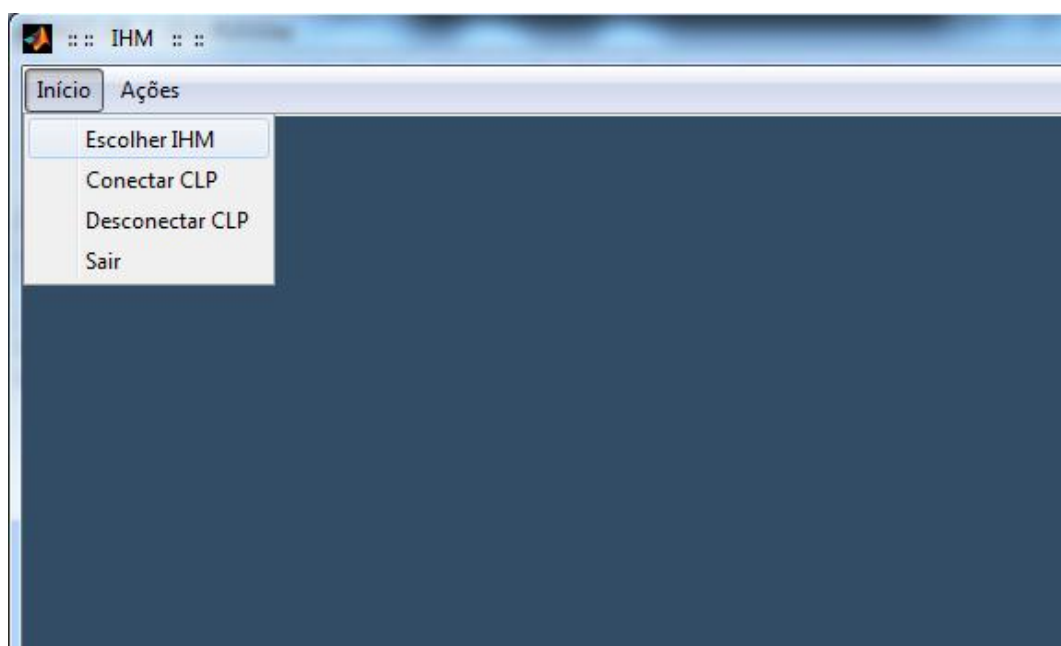


Figura 20: Conteúdo do menu Início

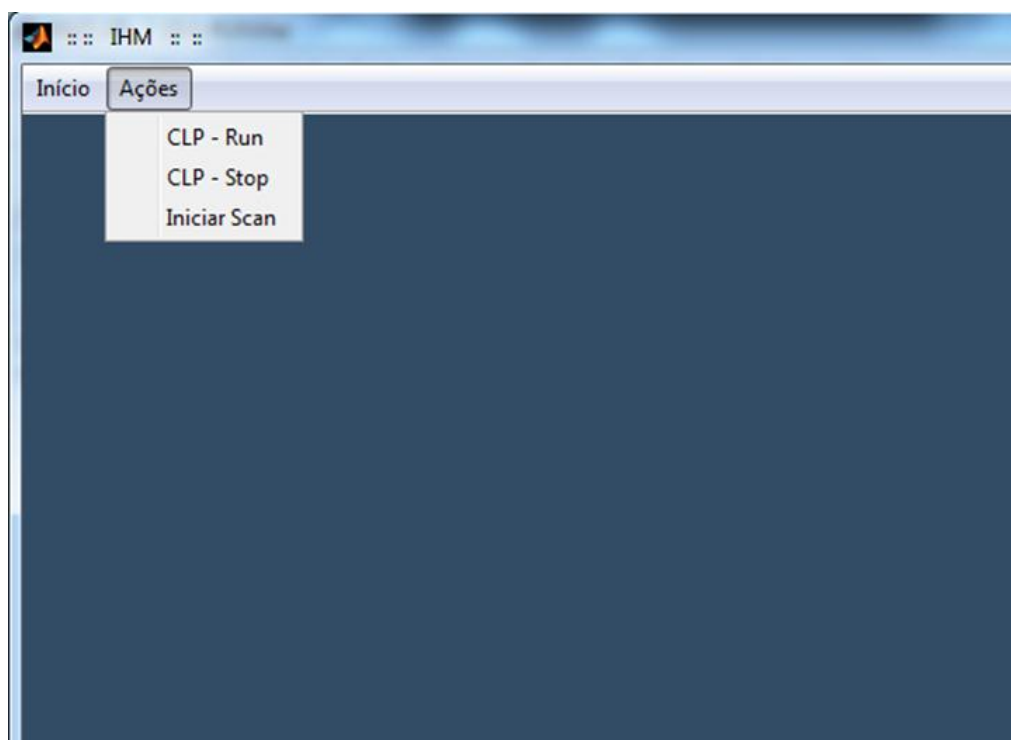


Figura 21: Conteúdo do menu Ações

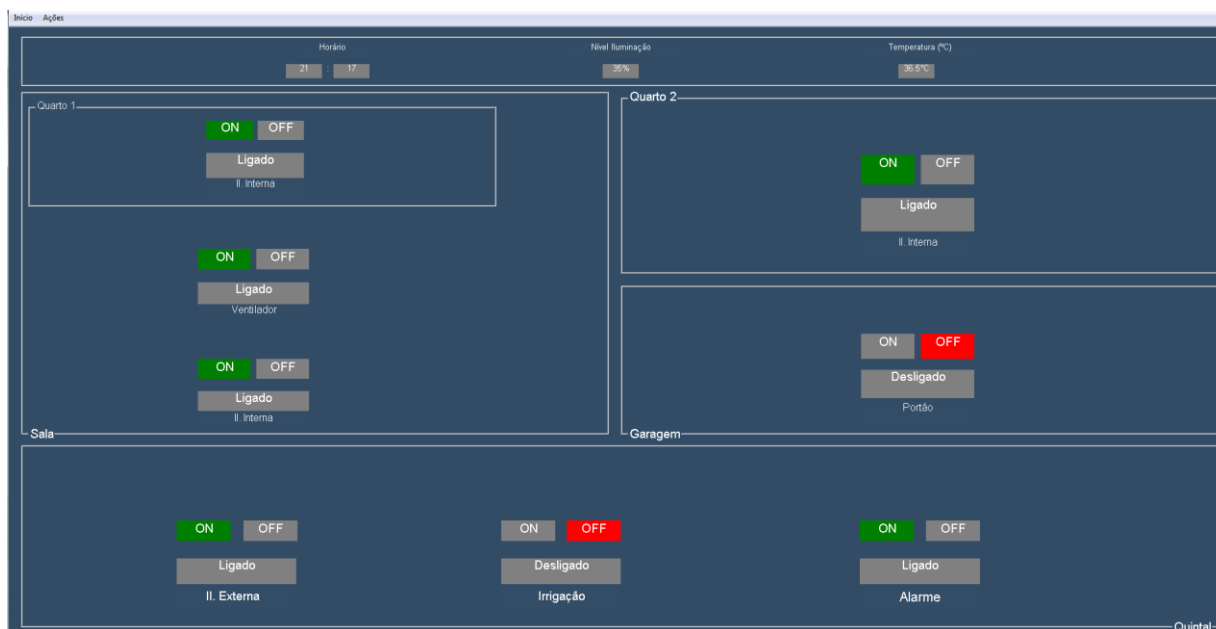


Figura 22: IHM da casa automatizada

A Figura 22 em questão é a IHM desenvolvida para a maquete e aonde é feita a interação com o usuário. Menus de temperatura, nível de luminosidade e horário estão

presentes já que estes estados controlam, ventilação, luzes internas e externas e irrigação, respectivamente.

Para que a IHM pudesse atuar sobre a automação residencial foram adicionadas novas chaves no ladder proposto na seção 3.1, de forma que haja uma autonomia de controle tanto fisicamente quanto via sistema supervisorio.

3.3 Resultados Finais

Montada a maquete e feita as ligações, em conjunto com o CLP e IHM, é possível aplicar, conforme objetivo do trabalho, o conceito de redes industriais à automação residencial devido a utilização do protocolo Modbus. Seguem os resultados obtidos.

3.3.1 Iluminação

Conforme antes mencionado a iluminação interna funciona via PWM, aumentando a luminosidade devido à ausência de luz. Para observar os resultados foi coberto o sensor LDR da iluminação interna a fim de simular a falta de luz.

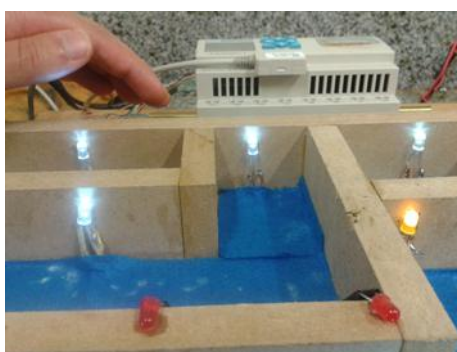


Figura 23: LEDs funcionando no primeiro estágio de falta de luminosidade

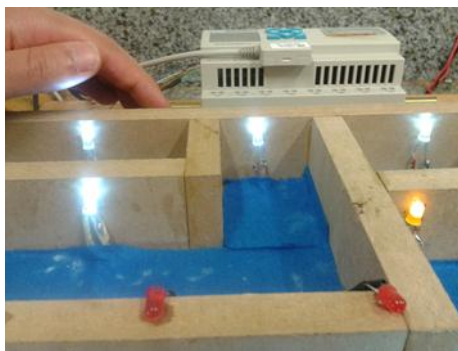


Figura 24: LEDs funcionando no segundo estágio de falta de luminosidade

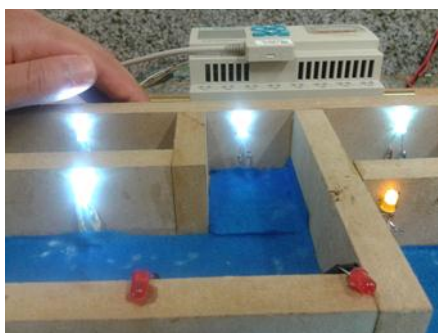


Figura 25: LEDs funcionando no terceiro estágio de falta de luminosidade

A análise da iluminação externa também foi satisfatória. Nas fotos abaixo pode-se perceber a atuação do sensor numa simulação da noite feita no laboratório.

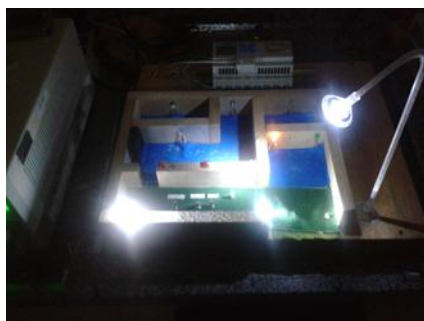


Figura 26: Iluminação externa ativada



Figura 27: Sistema de iluminação interna e externa em conjunto.

Em todos estes testes o nível de luminosidade é levado em consideração na automação, juntamente com o estado das chaves no sistema supervisor. Um exemplo de como estaria o estado da IHM caso a luminosidade esteja conforme mostrado Figura 24, pode ser visto na Figura 28 abaixo:

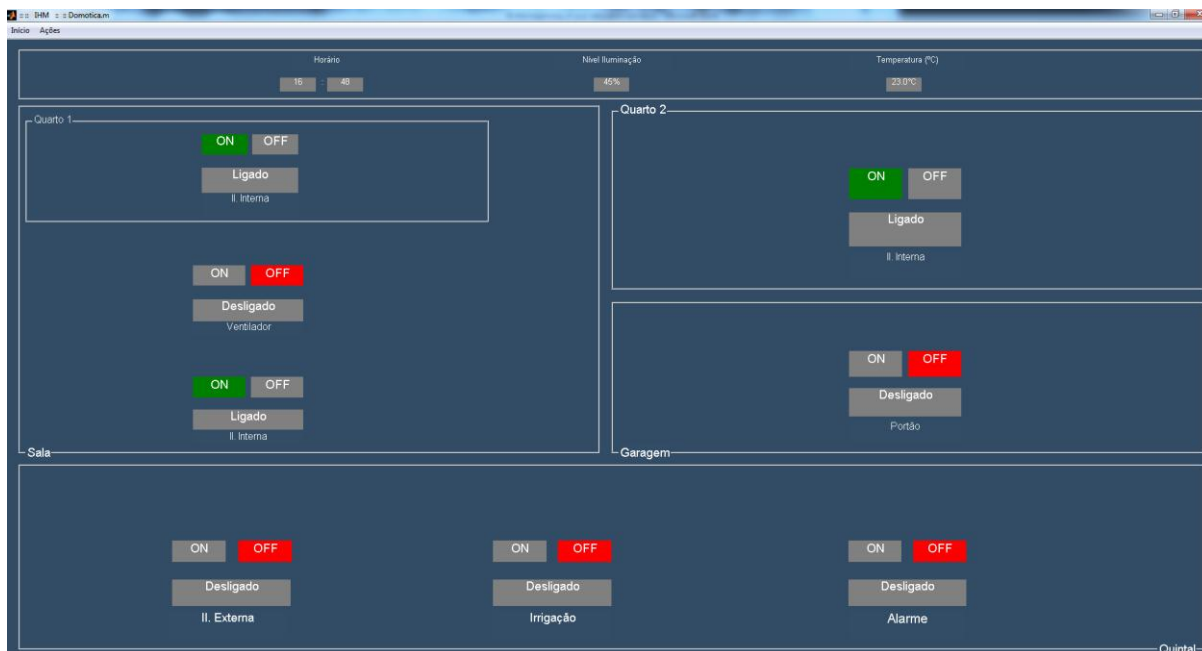


Figura 28: Registro da IHM no momento em que foi tirada a foto da Figura 24.

É interessante notar também que pela temperatura no momento medir menos de 25°C o ventilador se encontra no estado Desligado.

3.3.2 Segurança

As luzes do alarme, simbolizando uma presença indesejada, revelem que o circuito está funcionando conforme esperado. Ao atravessar o sensor, as luzes dos LEDs vermelhos se acenderam. Vale ressaltar que para que o alarme funcione este deve estar ativado antes.

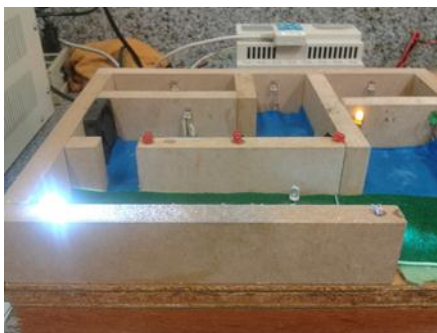


Figura 29: Alarme ativado sem a presença indesejada.

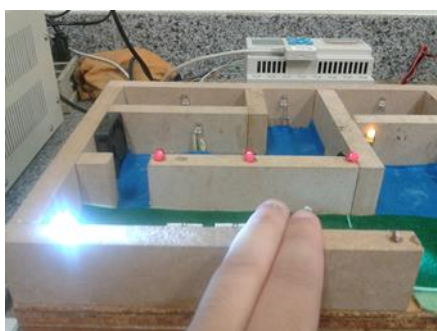


Figura 30: Alarme ativado detectando a presença.

3.3.3 Temperatura

O sistema de temperatura também funcionou conforme o esperado, ativando o cooler em 30°C e desativando em 25°C.

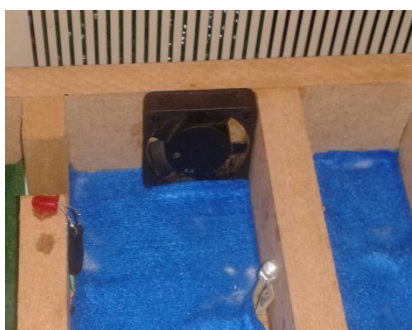


Figura 31: Cooler instalado para a refrigeração do sistema.

3.3.4 Automações

Na seção de automações pode-se analisar o funcionamento dos sistemas propostos da garagem e da irrigação. A foto abaixo percebemos que não há nada atravessando o sensor e o LED amarelo, simbolizando o portão fechado, está ligado.



Figura 32: Sensor da garagem e LED de portão fechado

Quando se detecta a presença, acontece o chaveamento e o LED verde acende, simbolizando a abertura do portão da garagem.



Figura 33: Sensor da garagem e LED portão aberto.

Na irrigação, conforme pré-programado, um LED azul acende indicando quando a irrigação está ocorrendo. Lembrando que é necessário que a chave da irrigação esteja ativada para o funcionamento no horário proposto.



Figura 34: LED azul indicando que a automação da irrigação

3.4 Outras aplicações: Partida de motor Estrela-Triângulo

Uma segunda aplicação seria a confecção de uma IHM para a automação feita na partida estrela-triângulo de um motor. A partida estrela-triângulo faz com que o motor realize uma partida mais suave, reduzindo em mais da metade a corrente de partida.

Um ponto importante deste tipo de partida de motor elétrico é que é necessário o fechamento para triângulo em um tempo específico (o motor deve atingir pelo menos 90% da sua rotação nominal). Para isso, CLPs são utilizados acoplados a tacômetros de forma que é possível ele fazer o sensoriamento e fazer o fechamento para triângulo no momento correto.

Foi então confeccionada também uma IHM para supervisionar este processo, contendo o estado do motor, ligado ou desligado, sua configuração, estrela ou triângulo e sua temperatura. Além disso um botão de segurança também foi adicionado à IHM para que seja feito o desligamento do motor caso necessário. O religamento só é efetuado após o botão de segurança ser liberado. O mostrador **Ventilador** e **Alarme**, informam através de cores o estado do ventilador e caso o automático tenha sido acionado. O ventilador é acionado acima de 50°C, já o alarme depende de alguma falha mecânica na ligação do motor ou caso a temperatura esteja acima dos 80°C. Um botão de status do CLP pode ser visto também na parte superior da IHM, informando se o CLP está no modo **RUN** ou **STOP**.

Segue nas figuras abaixo as fotos tiradas da IHM em funcionamento.



Figura 35: Após pressionado o botão ligar motor, este liga-se em estrela.

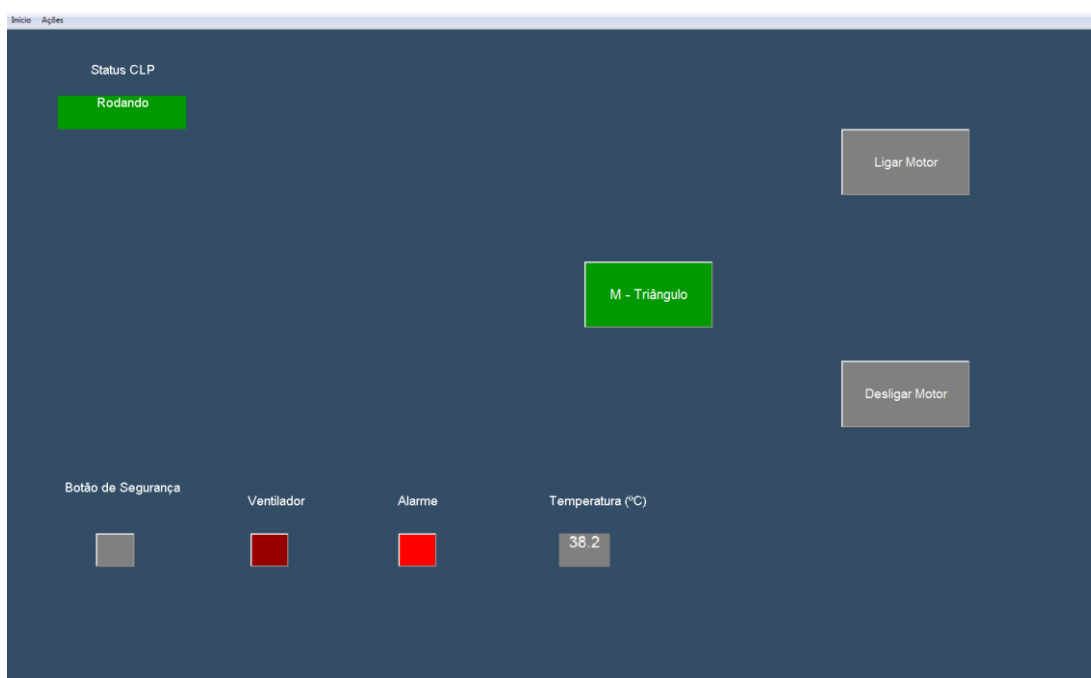


Figura 36: Após passado o tempo pré-determinado e atingido a velocidade requerida é feito o chaveamento que pode ser visto pela IHM que supervisiona o estado do motor

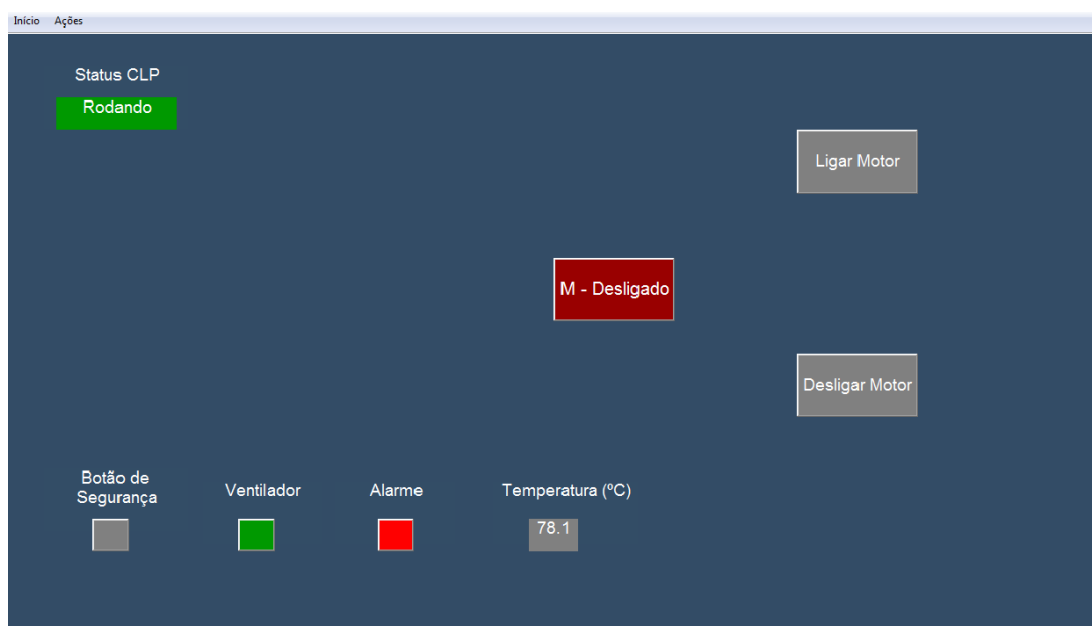


Figura 37: O motor desligado após alto nível de temperatura detectado pelo sensor

4 *Conclusão*

Este trabalho cumpriu o objetivo de fazer a comunicação entre o CLP Clic02 VT-D da WEG e um computador via protocolo MODBUS incorporado neste controlador, aplicando conceitos de redes industriais em outros ramos comerciais, como o imobiliário.

A conexão de um computador a um controlador tão versátil como são os CLP's estabelecem inúmeras possibilidades de criação de Sistemas Supervisórios, otimizando o processo de automação e gerando novas opções de controle que antes eram limitadas pelo dispositivo ou por dificuldades físicas.

Juntamente a isso este trabalho proporciona ao Departamento de Engenharia Elétrica da Universidade Federal de Viçosa novas possibilidades de pesquisa e extensão, como outras monografias baseadas na comunicação abordada neste texto ou novas implementações nas aulas práticas de automação. Além disso, passos como estes estimulam novos estudos dentro do departamento, ampliando ainda mais seu campo de atuação.

5 *Referências Bibliográficas*

GODOY, E. P. **Metodologias de Análise, Simulação e Controle para Sistemas de Controle via Redes**. 2007. Disponível em: <http://www.simulacao.eesc.usp.br/~egodoy/doutorado.html> Acesso em 01 jan. 2014.

GUEDES, L.A. **Classificação das redes para automação industrial**. 2005. Disponível em: http://www.dca.ufrn.br/~affonso/DCA0447/aulas/rai_cap3_part1.pdf. Acesso em 30 jan. 2014.

GUTIERREZ, R.M.V.; PAN, S.S.K. **Complexo Eletrônico: Automação do Controle Industrial** Disponível em: <http://www.bndes.gov.br/conhecimento/bnset/set2807.pdf>. Acesso em 06 jan. 2014.

MORAES, C. C. de; CASTRUCCI, P. L. **Engenharia de Automação Industrial**. 2.ed. LTC, 2007.

WEG S.A **Micro Controlador Programável Clic-02.** Disponível em: <http://ecatalog.weg.net/files/wegnet/WEG-rele-programavel-clic-02-3rd-manual-portugues-br.pdf>. Acesso em 12 de dez. 2013.

NOGUEIRA, A. **Redes de Comunicação para Sistemas de Automação Industrial.** 2009. 95p. Monografia (Monografia Graduação – Engenharia de Controle e Automação) UFOP. Ouro Preto.

MODBUS.ORG, **Modicon Modbus Protocol Reference Guide.** Disponível em [:http://modbus.org/docs/PI_MBUS_300.pdf](http://modbus.org/docs/PI_MBUS_300.pdf). Acesso em 08/02/2014.

MODBUS.ORG, **MODBUS Application Protocol Specification,** Disponível em: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf. Acesso em 08/02/2014.