

Guilherme Henrique Guerra Miranda

**SISTEMA DE SUPERVISÃO E
POSICIONAMENTO DE PAINÉIS
FOTOVOLTAICOS BASEADO EM REDE
MODBUS**

UBERLÂNDIA

2018

Guilherme Henrique Guerra Miranda

**SISTEMA DE SUPERVISÃO E POSICIONAMENTO DE
PAINÉIS FOTOVOLTAICOS BASEADO EM REDE
MODBUS**

Trabalho de Conclusão de Curso da Engenharia de Controle e Automação da Universidade Federal de Uberlândia - UFU - Faculdade de Engenharia Elétrica, como requisito para obtenção do título de Graduação em Engenharia de Controle e Automação.

Orientador: Prof. Me. Renato Santos Carrijo

UBERLÂNDIA

2018

Miranda, Guilherme Henrique Guerra

SISTEMA DE SUPERVISÃO E POSICIONAMENTO DE PAINÉIS FOTOVOLTAICOS BASEADO EM REDE MODBUS/ **Guilherme Henrique Guerra Miranda.**
– **UBERLÂNDIA, 2018-**

58 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Me. Renato Santos Carrijo

Trabalho de Conclusão de Curso – . **2018.**

Inclui bibliografia.

1. Sistemas Embarcados. 2. Modbus TCP. 2. Sistema de Supervisão. I. Renato Santos Carrijo. II. Universidade Federal de Uberlândia. III. Faculdade de Engenharia Elétrica. IV. Engenharia de Controle e Automação.

A Deus que me guiou em todos os momentos da vida.
A minha família e amigos que sempre estiveram ao meu lado e principalmente
a meus mestres que me ensinaram sobre minha profissão e também sobre a vida.

Agradecimentos

É primordial agradecer aqueles que estiveram presentes na trajetória da graduação. Agradeço principalmente a minha família por estar presente nesses e nos momentos mais difíceis vividos até agora. Agradeço todos os meus colegas que apoiaram e incentivaram a sempre buscar o melhor desenvolvimento, tanto na faculdade quanto na vida. Agradeço também a todos os professores do curso de Engenharia de Controle e Automação, em especial os professores Renato Carrijo, Renato Fernandes, Márcio, Marcelo, Carlos e Josué que me orientaram durante os períodos, com dedicação e carinho, ensinando não só as matérias mas como me tornar um exemplo de profissional de excelência. Por fim, agradeço a todos que com seu carinho e dedicação tornaram inesquecível esta etapa da vida.

*“Fracasso é uma possibilidade por aqui.
Se as coisas não estão fracassando,
você não está inovando o suficiente.”,
(Elon Musk)*

Resumo

Sistemas residenciais automatizados são uma realidade crescente onde a comodidade e a eficiência são objetivos a serem alcançados. Nesses sistemas, a necessidade da melhor otimização de recursos naturais e energéticos aliada a soluções inteligentes e cômodas trazidas pela automação ainda são complexas. Por outro lado, o crescente desejo por automação aliado as inovações tecnológicas torna mais acessíveis os equipamentos inteligentes, sistemas de automação e ainda sensores e atuadores diversos, integrando todo o ambiente residencial e deixando o usuário mais próximo do ambiente automatizado.

Aliado à necessidade de um ambiente inteligente, as energias renováveis seguem a mesma curva crescente. Hoje, a integração de um sistema residencial inteligente com a geração residencial de energia é inevitável. Os sistemas fotovoltaicos são um exemplo vigoroso desse relacionamento entre geração de energia e utilização residencial.

Este trabalho propõe o desenvolvimento de um sistema de automação embarcado em um microcontrolador AM335x ARM® Cortex-A8 disposto em uma placa de desenvolvimento do tipo BeagleBone Black® para gerenciamento e posicionamento dos painéis fotovoltaicos, comunicando com um sistema mestre como um controlador lógico programável através de um protocolo industrial, o Modbus TCP , levando robustez ao sistema tornando-o mais seguro e funcional. Nele, os painéis serão direcionados e angulados automaticamente para que recebam sempre a maior taxa de iluminação possível, melhorando a eficiência. O sistema de gerenciamento controla a temperatura dos painéis, dando ao usuário uma interface simples de monitoração dos dados através de uma interface homem máquina com toque na tela.

Palavras-chave: Sistema Embarcado. Modbus TCP. CLP. IHM.

Abstract

Automated residential systems are a growing reality where convenience and efficiency are goals to be achieved. In these systems, optimization of energy resources coupled with intelligent and convenient solutions brought about by automation are still complex. On the other hand, the growing desire for automation combined with technological innovations makes intelligent equipment, automation systems, sensors and actuators more accessible, integrating the entire residential environment and leaving the user closer to the automated environment.

At the same way, renewable energies follow the same growing curve. Today the integration of a smart residential system with residential power generation is inevitable. Photovoltaic systems are a powerful example of this relationship between energy generation and residential use.

This work proposes the development of an automation system embedded in an AM335x ARM® Cortex-A8 microcontroller arranged in a BeagleBone Black® type development board for the management and positioning of photovoltaic panels, communicating with a master system as a programmable logic controller through a Modbus TCP, an industrial protocol, bringing robustness to the system making it more secure and functional. In this system, the panels will be automatically oriented and angled so that they always receive the highest possible illumination rate, improving the efficiency, the management system would also control the panel temperature, giving the user a simple interface to monitor the data through a touch screen human machine interface .

Keywords: Embedded system. Modbus TCP. CLP. HMI.

Lista de ilustrações

Figura 1 – Sistema de Posicionamento versus sistema Fixo de painéis	17
Figura 2 – Representação entrada e saidas de um CLP Siemens S7 - 200	20
Figura 3 – Representação da linguagem ladder	21
Figura 4 – IHM Siemens como Sistema Supervisório	22
Figura 5 – Modelo OSI	23
Figura 6 – Tipos de mensagens - Modbus TCP	25
Figura 7 – Application Data Unit	25
Figura 8 – Exemplo de Ciclo de Mensagens Modbus	26
Figura 9 – Exemplo de utilização da rede Profinet	27
Figura 10 – Modelo de Blocos Fundamental - Cortex A8	28
Figura 11 – Diagrama de Blocos AM355x	28
Figura 12 – Sensor de temperatura LM35	29
Figura 13 – Resistor Dependente de Luz - LDR	30
Figura 14 – Malha fechada de controle de um Servo Motor	31
Figura 15 – Servo motor 9g	31
Figura 16 – Modulação por largura de pulso - PWM	32
Figura 17 – Altitude e Azimute Solar	33
Figura 18 – Efeito Fotoelétrico	33
Figura 19 – Construção de um Pannel Fotovoltaico	34
Figura 20 – Diagrama estrutural do projeto	35
Figura 21 – Placa de Desenvolvimento BeagleBone Black	36
Figura 22 – Arquitetura conceitual do modelo de mensagens Modbus TCP/IP e Modelo OSI	39
Figura 23 – Modbus TCP/IP ADU	39
Figura 24 – Definição do MBAP do Escravo TCP	40
Figura 25 – Porta de comunicação com o CLP	40
Figura 26 – Teste de Leitura dos quatro parâmetros AI	41
Figura 27 – Código parcial de configuração dos valores nos holding registers	42
Figura 28 – Teste de Escrita no canal DO	43
Figura 29 – Montagem do Sistema para Teste	43
Figura 30 – Configuração e teste do bloco MbClient do software TIA Portal	44
Figura 31 – WatchTable do mestre TCP - TIA PORTAL	44
Figura 32 – Vista isométrica do compartimento dos sensores	45
Figura 33 – (a)Vista frontal do compartimento de Sensores. (b) Fluxograma da Lógica de posicionamento	47
Figura 34 – Modelo 3D do sistema	47

Figura 35 – Diagrama elétrico do sistema protótipo	48
Figura 36 – Desenvolvimento IHM - TIA portal	49
Figura 37 – Teste de leitura das variáveis da IHM	49
Figura 38 – Tela inicial de utilização da IHM	50
Figura 39 – Tela de monitoração do posicionamento do painel fotovoltaico	51
Figura 40 – Tela de monitoração da temperatura do painel fotovoltaico	51
Figura 41 – Tela de monitoração da temperatura do painel fotovoltaico - Alarme de temperatura crítica	52
Figura 42 – Tela do sistema de resfriamento do painel fotovoltaico	52
Figura 43 – Tela de monitoração da luminância incidente no painel fotovoltaico - iluminância baixa	53
Figura 44 – Tela de monitoração da luminância incidente no painel fotovoltaico - iluminância Alta	53
Figura 45 – Protótipo 3D de suporte da placa fotovoltaica	54
Figura 46 – Compartimento dos Sensores LDR no protótipo	54

Lista de tabelas

Tabela 1 – Tabela de Requisitos funcionais do sistema	37
Tabela 2 – Tags do Projeto	38

Lista de abreviaturas e siglas

AI	<i>Analog Input</i>
ADU	<i>Application Data Unity</i>
ASCII	<i>American Code for Information Interchange</i>
CLP	<i>Controlador Lógico Programável</i>
HMI	<i>Human Machine Interface</i>
IHM	<i>Interface Homem-Máquina</i>
IP	<i>Internet Protocol</i>
IRT	<i>Isochronous Real Time</i>
ISO	<i>International Organization for Standardization</i>
LDR	<i>Light Dependent Resistor</i>
PDU	<i>Protocol Data Unity</i>
PWM	<i>Pulse Width Modulation</i>
RS-485	<i>Recommended Standard 485</i>
RTU	<i>Remote Terminal Unit</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
SRT	<i>Soft Real Time</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>

Sumário

1	INTRODUÇÃO	14
1.1	Justificativa	14
1.2	Objetivo	15
2	ESTADO DA ARTE	16
3	REFERENCIAL TEÓRICO	19
3.1	O cenário da Automação	19
3.2	Controlador Lógico Programável	19
3.2.1	Linguagem de programação por diagrama de contatos - Ladder	20
3.3	Interface Homem Máquina	21
3.4	Redes Industriais	22
3.4.1	Modbus	24
3.4.2	Rede Modbus TCP	24
3.4.3	Ethernet industrial : Profinet	26
3.5	Sistema Embarcados	27
3.5.1	Microcontroladores	27
3.5.2	Microcontroladores ARM	28
3.6	Sensores	29
3.6.1	Sensor de Temperatura LM35	29
3.6.2	Lighting Dependent Resistor – LDR	30
3.7	Atuadores	30
3.8	Energia Solar	32
4	METODOLOGIA	35
4.1	Visão geral do Sistema	35
4.2	Requisitos do Sistema	36
4.2.1	Tags do projeto	37
4.3	Desenvolvimento da rede no sistema embarcado	38
4.3.1	Teste de comunicação do Escravo TCP	40
4.4	Configuração do Mestre Modbus TCP	43
4.4.1	Desenvolvimento do protótipo para o sistema físico	45
4.5	Desenvolvimento da IHM	48
4.5.1	Teste da IHM	49
5	RESULTADOS E DISCUSSÕES	50

6	CONCLUSÃO E TRABALHOS FUTUROS	55
	REFERÊNCIAS	56

1 Introdução

As soluções de Automação Residencial levam ao usuário comodidade, eficiência e segurança TEZA (2002). Quando se alia um sistema inteligente de automação a um sistema de geração de energia o resultado visa a maior eficiência possível.

Os projetos de Automação atuais visam situações de controle mais comuns como o controle de iluminação, temperatura, sistemas de segurança e mídias. Os crescentes estudos estão direcionados a encontrar mais funcionalidades plausíveis de automação e consequentemente melhor eficiência desses novos sistemas, aliando ainda ao estudo de geração de energia residencial. Para tal, é proposto um sistema integrado de automação que visa supervisionar e posicionar painéis fotovoltaicos garantindo maior controle e eficiência na geração de energia, tornando mais fácil a supervisão e a integração com outros dispositivos inteligentes utilizando-se da robustez de sistemas de comunicação anteriormente apenas industriais e que agora tornam-se cada vez mais difundidos. Os sistemas fotovoltaicos geram energia elétrica através de células, utilizando-se de um efeito fotovoltaico. Esse efeito acontece quando os fótons da luz solar são absorvidos pela célula fotovoltaica. A energia presente nos fótons é transferida para os elétrons o que permite sua movimentação e gera a corrente elétrica Gazoli e Villalva (2015).

Nos sistemas de geração de energia fotovoltaica a eficiência está relacionada a taxa de iluminação dos painéis e a temperatura dos painéis solares. A importância da automação desses sistemas é indiscutível, bem como tornar esses sistemas mais eficientes e mais difundidos. Sistemas inteligentes, eficientes e limpos de geração de energia são os principais focos da questão energética do futuro.

1.1 Justificativa

Apesar das inúmeras pesquisas relativas a Automação industrial e também sobre energia solar e eficiência energética, existem poucas referências que integram esses temas e abordam um sistema inteligente de geração de energia residencial ou industrial de pequeno porte.

O sistema de automação do mercado, que são direcionados ao uso e eficiência solar possuem custo elevado, sendo viáveis apenas em grandes fábricas ou usinas de energia.

A motivação para realizar este trabalho é estudar as dificuldades de elaborar um sistema de automação simples, mas comum em indústrias, utilizando um sistema embarcado, com protocolo industrial de comunicação e integrando uma interface de supervisão para o usuário.

1.2 Objetivo

Implementar um sistema de automação para gerenciamento e posicionamento dos painéis fotovoltaicos embarcado em um microcontrolador do tipo AM335x ARM® Cortex-A8 disposto em uma placa de desenvolvimento do tipo Beaglebone Black® e comunicando com um sistema mestre presente no controlador lógico programável através de um protocolo de rede industrial, o ModBus TCP. O sistema citado deve ser capaz de direcionar e angular os painéis automaticamente para que recebam sempre a maior taxa de iluminação possível melhorando sua eficiência. O sistema deve ainda monitorar e atuar sobre a temperatura dos painéis e fornecer ao usuário um sistema de monitoração através de uma Interface homem máquina.

2 Estado da Arte

O uso de protocolos de comunicação industrial é algo bastante difundido e alvo de inúmeros estudos comerciais e educacionais. A inovação e tecnologia atual, trouxe a sociedade sistemas mais integrados e acessíveis levando esses protocolos que antes só eram vistos nos ambientes industriais a lugares mais próximos como as residências e comércios.

Como comprovado por (NOGUEIRA, 2009), os sistemas de automação e controle tem utilizado cada vez mais em redes de comunicação industriais, advento da necessidade de interoperabilidade entre Tecnologia de Automação e a de informação. Possibilitando que o processo industrial e a rede de negocio da empresa estejam interligados, melhorando a percepção e gestão dos resultados das fábricas. O mesmo ganho é apresentado quando se utiliza no âmbito residencial, sistemas mais conectados dão ao usuário maior controle dos sistemas de sua casa e assim mais conforto a quem utiliza.

O trabalho proposto pretende criar um sistema de automação para painéis fotovoltaicos utilizando-se de equipamento e protocolos industriais como o CLP, IHM e a rede Modbus, levando a robustez e a segurança utilizados na indústria para o ambiente residencial. Nesse sistema o CLP seria o mestre da rede e a placa de desenvolvimento seria o escravo, informando os dados coletados por seus sensores sempre que fosse solicitado. Todo o sistema seria integrado a um ambiente de supervisão disposto na IHM tornando o processo que foi automatizado mais visual, facilitando o controle e o entendimento do usuário sobre o mesmo. Um motivador é que nesse tipo de sistema, haverá certa flexibilidade para se adaptar as diferentes residências e aos tipos de sensores e atuadores de modo semelhante com o que acontece na indústria.

Para se obter um projeto de sistema confiável para a aquisição de dados e atuação, a seleção do modo e protocolo de comunicação é primordial de modo que o próximo passo seja a construção da melhor arquitetura para suportar o projeto (PATEL, 2017). O protocolo de comunicação utilizado, é um dos mais difundidos atualmente por conta da simplicidade de implementação e robustez. Por possuir um protocolo de transporte e rede do tipo TCP/IP futuramente ele também poderia funcionar sobre a internet, integrando outros conceitos do futuro e conectando o controle e os pontos da residência com o usuário ao redor do mundo.

Semelhante ao estudado por (PATEL, 2017), para o projeto proposto o microcontrolador utilizado teve um papel de escravo na rede de comunicação, esse propôs uma rede Modbus sob RS485, um meio físico diferente mas que ainda possibilitaria diversos escravos submetidos a um ou mais mestres de Rede.

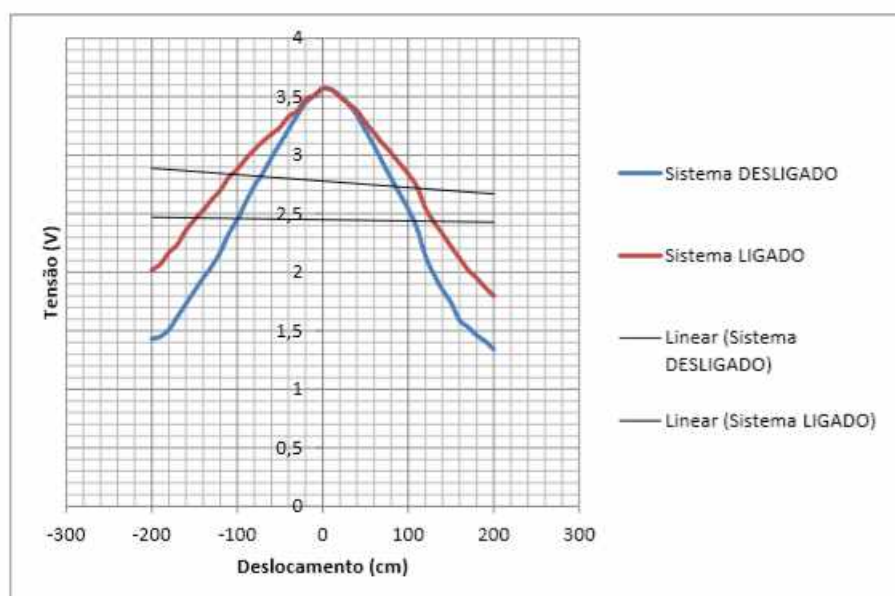
O protocolo de comunicação utilizado, é um dos mais difundidos atualmente

por conta da simplicidade de implementação e robustez. Por possuir um protocolo de transporte e rede do tipo TCP/IP futuramente ele também poderia funcionar sobre a internet, integrando outros conceitos do futuro e conectando o controle e os pontos da residência com o usuário ao redor do mundo.

NICÁCIO e CARVALHO (2011) desenvolveram um protótipo de um sistema de posicionamento de painéis fotovoltaicos utilizando microcontrolador Arduino, embora o objetivo seja aproximado, deste difere-se o microcontrolador utilizado bem como o sistema de automação, para o trabalho a ser desenvolvido existirá uma rede de comunicação e um sistema de supervisão e gerenciamento, tornando o escopo mais amplo e mais completo. Para o sistema aqui proposto, decidiu-se pela utilização da placa de desenvolvimento Beaglebone Black®, mais robusta que a proposta no trabalho citado e que por ser uma placa que atende as expectativas quanto ao hardware, mostra-se como um ambiente que facilita o desenvolvimento e ainda possui baixo custo quando comparado a outros sistemas semelhantes de automação.

Nos estudos de NICÁCIO e CARVALHO (2011), comprovou-se experimentalmente a eficiência de funcionamento da placa fotovoltaica com posicionamento automático em relação a placa fixa. Ao longo do dia a placa posicionada de acordo com a iluminação recebe ao menos 16% mais iluminação, fato também comprovado por (VALLDOREIX GREENPOWER, 2015) e que complementa que para sistemas eficientes, o ganho pode ultrapassar os 30%.

Figura 1 – Sistema de Posicionamento versus sistema Fixo de painéis



Fonte: Adaptada de NICÁCIO e CARVALHO (2011)

TAMBOLI (2015) utilizou o protocolo Modbus TCP para comunicação e controle de processos do tipo batelada, assim como SILVA e FEIJOO (2017) realizaram uma comunicação via Modbus TCP entre o CLP Siemens S7-1200 e o CLP Twido, da Schneider Electric utilizando ainda como sistema de supervisão uma IHM. (TAPIA, 2013) constatou que existem grandes vantagens em utilizar uma IHM, desde a melhor visualização e entendimento do processo quanto o acompanhamento em tempo real das variáveis, tornando o sistema mais verossímil. As percepções dos trabalhos citados instigaram ainda mais a utilização de uma interface de monitoração do projeto.

Esses estudos são fundamentais na elaboração deste trabalho, utilizando dos resultados desses para desenvolver um sistema que reúne a melhor das características dos trabalhos anteriores, embasando os conhecimentos para os tópicos objetivados a implementação.

3 Referencial teórico

Neste tópico serão referenciadas parte das teorias principais cujo conhecimento são necessários para melhor compreensão do trabalho realizado. A princípio será contextualizado o cenário da automação. Em seguida serão evidenciados os conceitos de redes industriais, o dispositivo de controle CLP e da lógica utilizada para a programação do sistema e ainda características do microcontrolador utilizado bem como o funcionamento dos painéis fotovoltaicos. Neste tópico também será fundamentado o conceito de interface homem-máquina, sua função e características. Por fim, serão definidos os conceitos de sistema embarcado, atuadores e sensores.

3.1 O cenário da Automação

A automação industrial não é novidade apenas do século XX, ela remete a períodos longínquos na história. (ROGGIA; FUENTES, 2016) disse “Entretanto, a automação industrial começou a conquistar destaque na sociedade no século XVIII, com o início da Revolução Industrial, originada na Inglaterra. Devido a uma evolução no modo de produção, o homem passou a produzir mercadorias em maior escala.” Com o passar dos séculos e o destaque da energia elétrica, as indústrias do aço, a química e ainda outras se desenvolveram como os setores de comunicação e transporte em decorrência da busca por melhorar o processo produtivo, automatizando-o como possível.

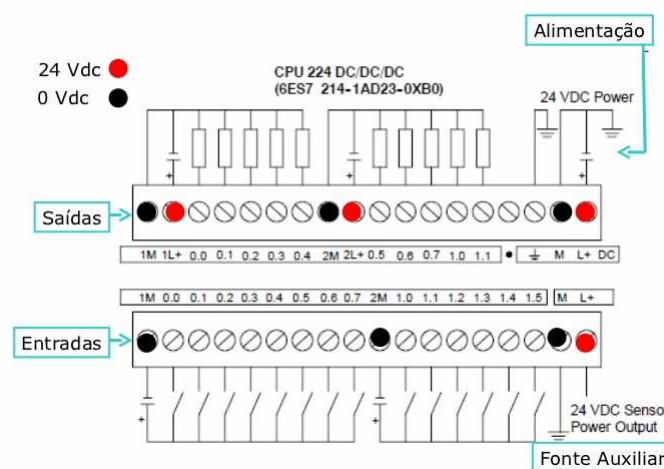
Já no século XX, os computadores moldam a automação. Utilizar servomecanismos e controladores programáveis passou a fazer parte de grande parte das aplicações da automação, tanto industriais como comerciais ou residenciais. Os computadores e microprocessadores formam a base da tecnologia atual e sua tecnologia são fundamentais para a automação contemporânea e exemplos de sua aplicação estão presentes em praticamente todas as áreas. Em seguida serão apresentados os conceitos dos equipamentos e tecnologias utilizadas, importantes para o desenvolvimento do trabalho.

3.2 Controlador Lógico Programável

(ROGGIA; FUENTES, 2016) definiu o Controlador Lógico Programável, o CLP, como um equipamento eletrônico digital, com hardware e software compatível com as aplicações industriais. Os comandos e controles industriais foram revolucionados após o surgimento do CLP na década de 70. A principal vantagem é a possibilidade de reprogramação e adaptação ao projeto, permitindo transferir as modificações de hardware em modificações de software.

Antes do surgimento dos controladores, a realização de tarefas de comando e controle só era possível através de painéis compostos de relés eletromagnéticos. Como cada painel era especificamente projetado para uma aplicação específica, para realizar modificações no processo havia a necessidade de alterações de âmbito estrutural, muitas vezes complexas e onerosas. O CLP Possui entradas para aquisição de dados e saídas para acionar diversos tipos de dispositivos ou processos 2, facilitando a modificação pela lógica de controle, do seu funcionamento no controle do processo.

Figura 2 – Representação entrada e saídas de um CLP Siemens S7 - 200



Fonte: Adaptada de Júnior (2011)

A programação pode ser realizada através de diversas linguagens de modo a contribuir de melhor maneira na execução correta das funções aritméticas, lógicas, de temporização, de contagem, ou ainda outras. A IEC 61131-3 define cinco linguagens de programação, elas são:

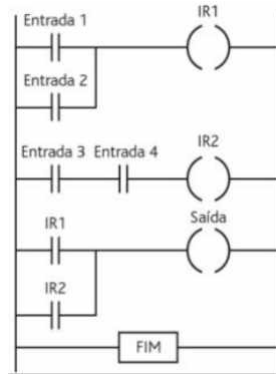
- ST (Structured Text) - Texto Estruturado
- IL (Instruction List) - Lista de Instruções
- LD (Ladder) - Linguagem ladder
- FBD (Function Block Diagram) - Diagrama de bloco
- SFC (Sequential Flow Chart) - Diagrama de Fluxo

3.2.1 Linguagem de programação por diagrama de contatos - Ladder

A linguagem Ladder é uma forma de programação gráfica, onde são representados contatos e bobinas. Os diagramas são compostos por duas linhas verticais e de linhas

horizontais, as quais são colocadas as instruções que serão executadas na rotina do programa, figura 3. As instruções podem ser contatos, bobinas, temporizadores, contadores ou ainda blocos de função (ROGGIA; FUENTES, 2016).

Figura 3 – Representação da linguagem ladder



Fonte: Adaptada de Júnior (2016)

A construção do programa parte da ideia de energização partindo de um caminho a ser percorrido, um fluxo semelhante ao que acontece com a corrente elétrica, fluindo no sentido da barra da esquerda para direita. Cada uma das linhas horizontais é uma instrução lógica onde os contatos são as entradas das sentenças, por exemplo os sensores, e as bobinas ao final da linha são as saídas ou atuadores. A lógica do programa é construída conforme a associação desses contatos e bobinas.

3.3 Interface Homem Máquina

A necessidade de centralizar informações, de modo a se obter a maior quantidade possível de dados é fundamental em todos os projetos de automação. A monitoração das variáveis do sistema é muito importante para assegurar a qualidade do sistema proposto (TAPIA, 2013). O sistema de supervisão, também conhecido como SCADA (Supervisory Control and Data Acquisition), coleta dados do processo através de determinados dispositivos industriais, os formata e apresenta ao operador em uma multiplicidade de formas.

(ROGGIA; FUENTES, 2016) evidenciou os principais benefícios do uso de sistemas de supervisão como a velocidade das informações e da detecção e correção de falhas, aumentando a segurança, a qualidade e a produtividade no processo. Em um sistema como esse, a tela de visão geral de processo apresenta ao operador uma visão global do sistema, sob visualização imediata na operação do proposto em projeto. Nestas telas são apresentados os dados mais significantes à operação e objetos que o representam. As principais características que um sistema de supervisão deve possuir são:

- Interface propícia a fácil visualização gráfica e de operação do sistema, ou seja, visualização dos valores coletados por sensores e estados atuais de atuadores.
- Gerenciamento de condições de alarme, mensagens de alerta, definidas pelo usuário, alertando ao utilizador sobre alguma situação específica.

(ROGGIA; FUENTES, 2016) também reforçou que a automação é cada vez mais comum e complexa e que garantir o melhor uso das informações é vital para a segurança e confiabilidade. A implementação de mecanismos de acessibilidade, segurança contra falhas dos sistemas de supervisão é fundamental em projetos de automação.

Figura 4 – IHM Siemens como Sistema Supervisório



Fonte: Adaptada de *Siemens (2018)*

A IHM é um sistema de supervisão e atuação disposto em uma tela, tornando possível a interação do sistema com o usuário, seja por visualização das variáveis, dos alarmes, ou acionamento na tela de atuadores no sistema. Essa é a interface do homem com o projeto e suas variáveis.

3.4 Redes Industriais

Uma rede de computadores é a interconexão de computadores autônomos e interconectados. A capacidade de trocar informações, através de um meio físico de comunicação é essa interconexão, apenas em sistemas com relacionamento definido essa autonomia não é um requisito (TANENBAUM; WETHERALL; TRANSLATIONS, 2011).

As redes industriais surgem da necessidade de interligar computadores e controladores permitindo o compartilhamento de recursos e bases de dados, que passam a ser únicos, conferindo segurança as informações. As redes foram desenvolvidas originalmente para a troca de dados entre computadores, no entanto com o advento da tecnologia, o uso de microprocessadores tornou possível realizar essa troca utilizando até mesmo instrumentos sensores, no chão de fábrica ou em outras aplicações (NOGUEIRA, 2009).

Como ao longo da história o número de aplicações e dispositivos aumentava, tornou-se importante que padrões fossem criados, possibilitando que diferentes dispositivos e fabricantes fossem utilizados em um mesmo processo. Um considerável esforço internacional aconteceu para a padronização de redes. A ISO (*International Organization for Standardization*) definiu um modelo referência para a interconexão de sistemas em 1978. O padrão OSI utiliza-se de uma filosofia de arquitetura em camadas, gerenciando a estruturação da comunicação de dados, através das sete camadas mostradas pela figura 5.

Figura 5 – Modelo OSI

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data link
1	Physical

Fonte: Adaptada de *Briscoe (2000)*

Uma interface entre as camadas define as operações elementares e os serviços que uma camada inferior oferece para a superior. Nesse modelo, cada camada corresponde a um nível de abstração, possui suas funções próprias e bem definidas.

(NOGUEIRA, 2009) definiu a camada de Aplicação como a interface do protocolo de comunicação a aplicação do usuário. Ela é responsável pela interface com o aplicativo que receberá as informações da transmissão. A camada de Apresentação é responsável pela formatação, sintaxe e semântica dos dados transmitidos que permitirá a interpretação da informação pelo receptor; essa camada converte o dado recebido em um formato entendido pelo protocolo usado. A camada de Sessão é responsável pela troca de dados, ela controla a comunicação entre os usuários, agrupando as mensagens, e gerenciando a transferência de dados entre as camadas. A camada de transporte no emissor recebe os dados enviados pela camada de sessão e os segmenta para que sejam enviados a de rede. No receptor, a de Transporte realiza o processo inverso. A camada de transporte separa as camadas de nível de aplicação das camadas físicas. A camada de Rede gerencia o tráfego e o roteamento dos dados. O roteamento utiliza-se de priorização e tráfego de dados (NOGUEIRA, 2009). A camada de Enlace de Dados assegura o conteúdo da mensagem. A mensagem deve ser a mesmo no local de destino e na origem. Esta camada é responsável pela validação dos dados (NOGUEIRA, 2009). A camada Física é responsável pela transferência de bits num circuito de comunicação é a interface elétricas e física.

3.4.1 Modbus

Desenvolvido e publicado pela Modicon Industrial Automation Systems em 1979 , é um dos mais antigos protocolos de rede utilizados em controladores lógicos programáveis para aquisição de sinais e comando de atuadores. O grupo Schneider Electric, ao qual a Modicon faz parte colocou as especificações e normas que definem o Modbus em domínio público, tornando o protocolo mais popular e utilizado em vários equipamentos existentes, além de tornar as soluções de rede mais baratas para utilização (NOGUEIRA, 2009). Em um modelo mestre escravo, a comunicação é sempre iniciada pelo mestre, e os nós escravos não se comunicam entre si. O mestre pode transmitir dois tipos de mensagens aos escravos, dentro de uma mesma rede:

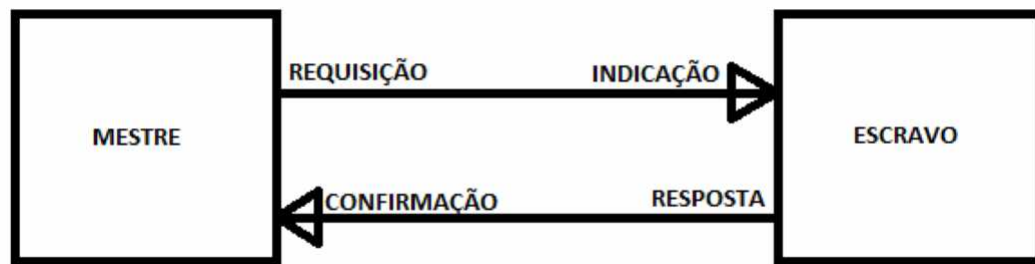
- Mensagem tipo *unicast*: o mestre envia uma requisição ao escravo definido e este retorna uma mensagem-resposta ao mestre.
- Mensagem tipo *broadcast*: o mestre envia a requisição para todos os escravos, e não é enviada nenhuma respostas para o mestre (MORAIS; CASTRUCCI, 2007). Existem dois modos de transmissão: ASCII (*American Code for Information Interchange*) e RTU (*Remote Terminal Unit*)

Para a o modo ASCII, cada byte de 8 bits em uma mensagem é enviado como dois caracteres ASCII. A principal vantagem deste modo é que ele permite que intervalos de tempo de até um segundo ocorram entre os caracteres sem causar um erro. Já o modo RTU tem cada byte de 8 bits em uma mensagem contendo dois caracteres hexadecimais de 4 bits. A principal vantagem deste modo é que sua maior densidade de caracteres permite melhor rendimento de dados do que o ASCII utilizando mesma taxa de transmissão (MODBUS-IDA, 2006).

3.4.2 Rede Modbus TCP

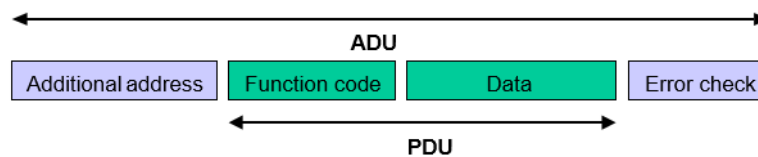
O protocolo Modbus também pode ser desenvolvido sobre uma rede Ethernet TCP/IP. O modelo é baseado em quatro tipos de mensagens, requisição, confirmação, indicação e resposta.

Figura 6 – Tipos de mensagens - Modbus TCP



O protocolo Modbus define uma unidade de dados de protocolo (PDU) simples e independente das camadas de comunicação subjacentes. O mapeamento do protocolo Modbus em barramentos ou redes específicos pode introduzir alguns campos adicionais na unidade de dados da aplicação (ADU).

Figura 7 – Application Data Unit



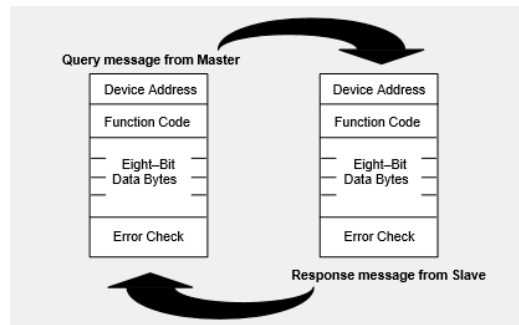
Fonte: Adaptada de *Modbus-IDA (2006)*

O código de função na requisição informa ao dispositivo escravo endereçado que tipo de ação deve ser executada. Os bytes de dados contêm qualquer informação adicional que o escravo precisará para executar a função. Por exemplo, o código de função 03 consultará o escravo para ler registros do tipo holding e responder com seu conteúdo. O campo de dados deve conter a informação informando ao escravo qual registrador iniciar e quantos registros devem ser lidos. O campo de verificação de erros fornece um método para o escravo validar a integridade do conteúdo da mensagem.

Para resposta normal, o código de função na resposta é um eco do código de função na requisição consulta. Os bytes de dados contêm os dados coletados pelo escravo, como valores de registro ou status. Se ocorrer um erro, o código da função é modificado para indicar que a resposta é uma resposta de erro e os bytes de dados contêm um código que descreve o erro. O campo de verificação de erros permite ao mestre confirmar que o conteúdo da mensagem é válido. No nível da mensagem, o protocolo Modbus ainda aplica o princípio mestre-escravo, embora o método de comunicação de rede seja ponto-a-ponto. Se um controlador originar uma mensagem, ele o faz como um dispositivo mestre e espera uma resposta de um dispositivo escravo. Da mesma forma, quando um controlador recebe

uma mensagem, ele constrói uma resposta escrava e a retorna para o controlador de origem (MODBUS-IDA, 2006).

Figura 8 – Exemplo de Ciclo de Mensagens Modbus



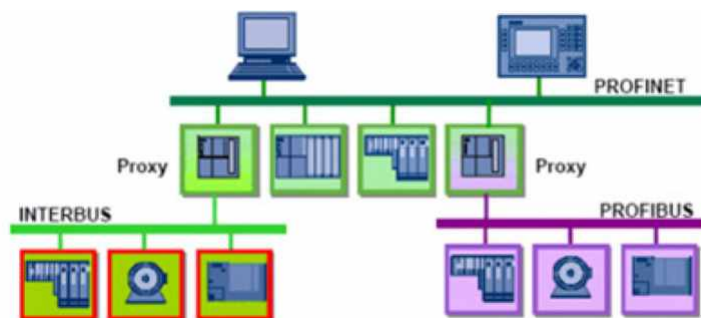
Fonte: Adaptada de *Modbus-IDA (2006)*

3.4.3 Ethernet industrial : Profinet

O Profinet é uma rede padronizada pela Associação Profibus Internacional como uma rede de Ethernet industrial. A norma que regulamenta o Profinet é a IEC61158-5 e IEC61158-6.

Segundo (LUGLI, 2008), o Profinet pode ter três formas distintas de operação, sendo duas delas para tempo real. Chamada de *Non-real time (Non-RT)*, a primeira maneira baseia-se na arquitetura TCP/IP pura, utilizando Ethernet na camada um e dois, o IP na camada três e o TCP ou UDP na camada quatro. Essa arquitetura não é utilizada em aplicações que requerem tempo real pois o tempo de processamento aproxima-se dos 100ms. A segunda maneira baseia-se no chamado *Soft Real Time (SRT)*, caracterizando-se por ser um canal que interliga diretamente a camada da Ethernet à aplicação. Eliminando vários níveis de protocolo, reduzindo o comprimento das mensagens transmitidas e o tempo de transmissão dos dados. A terceira maneira baseia-se no conceito de *Isochronous Real Time (IRT)*, para aplicações em que o tempo de resposta é crítico e deve ser menor do que 1ms. Uma aplicação típica deste conceito é o controle de movimento de robôs, quando o tempo de atualização dos dados deve ser pequeno (LUGLI, 2008).

Figura 9 – Exemplo de utilização da rede Profinet



Fonte: Adaptada de *Lugli (2008)*

3.5 Sistema Embarcados

3.5.1 Microcontroladores

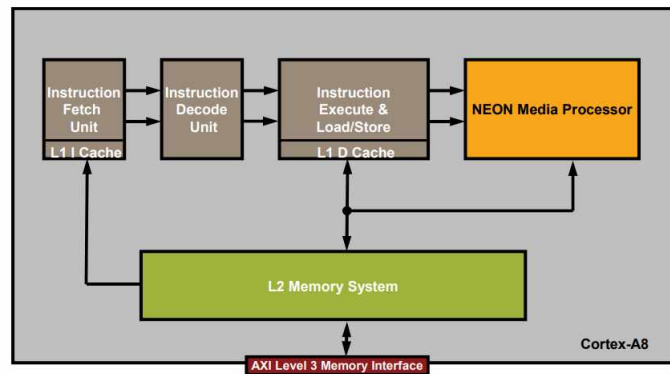
Segundo (PENIDO; TRINDADE, 2013), Um microcontrolador pode ser descrito como um computador em um único chip. Esse chip contém um processador, memórias, entradas analógicas e digitais, saídas, temporizadores, dispositivos para comunicação serial, e ainda outros.

O microcontrolador normalmente é composto dos seguintes dispositivos:

- Uma CPU (Central Processor Unit ou Unidade de Processamento Central), cuja finalidade é interpretar as instruções de programa.
- Uma memória PROM (Programmable Read Only Memory ou Memória Programável Somente de Leitura) onde são gravadas as instruções do programa.
- Uma memória RAM (Random Access Memory ou Memória de Acesso Aleatório) onde se armazenam as variáveis utilizadas pelo programa.
- Um conjunto de portas de entrada e saída, conhecidas como portas I/O, que são interfaces dispositivos externos.
- Geradores de clock, contadores, interfaces para comunicação serial, Conversores A/D e interfaces de rede.

Os processadores e microcontroladores podem ser classificados em dois modelos quanto a sua arquitetura, Harvard e Von Neuman. Quando um sistema possui uma única área de memória na qual ficam armazenados as variáveis e o programa a ser executado, diz-se que o sistema segue a arquitetura de Von Neuman. Para a arquitetura Harvard, as variáveis e o programa são armazenados em áreas distintas.

Figura 10 – Modelo de Blocos Fundamental - Cortex A8

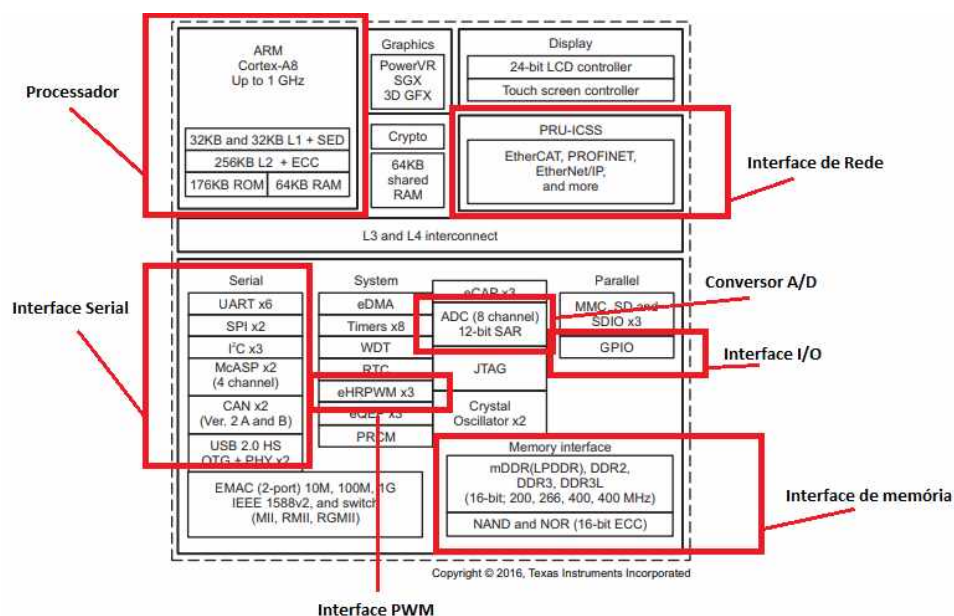


Fonte: ARM (2006)

3.5.2 Microcontroladores ARM

Os microcontroladores ARM, são microcontroladores baseados na arquitetura RISC, que favorece um conjunto pequeno e mais simples de instruções, favorecendo o tempo de execução de rotinas. Mais robustos que os controladores PIC, microcontroladores da arquitetura Harvard fabricados originalmente pela Microchip Technology, são mais rápidos por possuírem maior frequência e largura de dados. Semelhante ao descrito anteriormente, os microcontroladores do tipo ARM, possuem os dispositivos característicos como visto na figura 11 e se enquadram, no caso do o AM335x , modelo utilizado no projeto, a arquitetura fundamental de Von Neuman figura 10.

Figura 11 – Diagrama de Blocos AM355x



Fonte: Adaptada de Texas instruments (2016)

Os microprocessadores AM355x, baseados no processador ARM Cortex-A8, são aprimorados com opções de imagem, processamento gráfico, periféricos e interface industrial, como EtherCAT e PROFIBUS. Os dispositivos suportam sistemas operacionais de alto nível como Linux e Android (TEXAS INSTRUMENTS, 2016) .

3.6 Sensores

O sensor é um elemento sensível a uma forma de energia do ambiente. Ele relaciona informações sobre uma grandeza que precisa ser medida. Um sensor nem sempre tem as características elétricas necessárias para ser utilizado em um sistema de controle. Normalmente o sinal de saída deve ser manipulado e isso geralmente é realizado com um circuito de interface para produção de um sinal que possa ser lido pelo controlador. Quando o essa interface está acoplada a um sensor, o dispositivo recebe o nome de transdutor.

De acordo com a natureza do sinal de saída, os sensores podem ser classificados em sensores digitais e sensores analógicos. Sensores digitais monitoram a ocorrência ou não de um evento, apresentando em sua saída apenas dois níveis lógicos distintos, como ligado ou desligado, detectando a presença ou ausência de determinada grandeza. Já os sensores analógicos são utilizados para monitorar uma grandezas em uma faixas contínuas de valores, estabelecidas entre os limites mínimo e máximo do componente. Apresentam em sua saída um sinal de tensão, corrente ou resistência proporcional à grandeza física sensoreada.

3.6.1 Sensor de Temperatura LM35

Fabricado pela National Semiconductor pertencente ao grupo Texas Instruments, o sensor de temperatura LM35 realiza a medida da temperatura com saída linearmente proporcional a valores centígrados. Sua utilização é pertinente em variadas aplicações, relacionadas diretamente com a escala de temperatura desejada. As principais utilizações vão de termômetros, até controles, termostatos e detectores. Ele possui baixa impedância, saída linear e calibração inerente precisa, tornando o interfaceamento com os circuitos de controle e leitura mais simplificado.

Figura 12 – Sensor de temperatura LM35

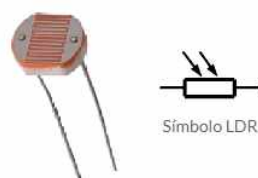


Fonte: *Filipe Flop (2017)*

3.6.2 Lighting Dependent Resistor – LDR

Os Resistores dependentes de luz são componentes eletrônicos que variam a sua resistência elétrica em função da intensidade da radiação eletromagnética visível incidente em sua superfície sensível (NICÁCIO; CARVALHO, 2011).

Figura 13 – Resistor Dependente de Luz - LDR



Fonte: Adaptada de *Filipe Flop (2017)*

Com o diminuição da luz incidente a resistência elétrica aumenta alcançando a faixa de Mega Ohms quando existe pouca ou nenhuma luz e diminui à algumas centenas de Ohms quando iluminado diretamente. Esse tipo de Resistor é então um sensor de luminosidade, detectando a presença da luz. Utilizando um circuito apropriado, o LDR varia a tensão analógica e essa, através de uma entrada analógica e de uma lógica de programação, informa o sistema de controle da presença e da quantidade incidente.

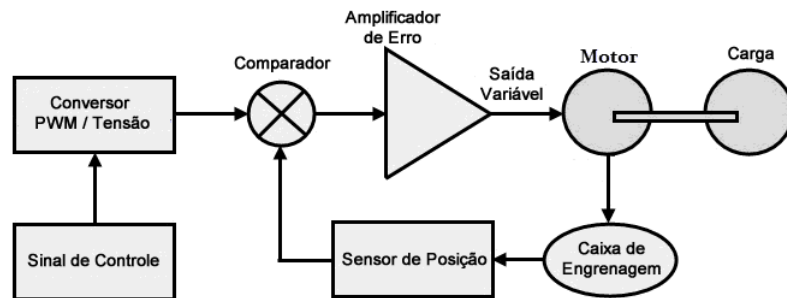
3.7 Atuadores

Atuadores são elementos que produzem ações motoras atendendo a comandos, essas podem ser manuais, elétricas ou mecânicas. Um dos modelos mais comuns de atuadores de posição é o servo motor (ROGGIA; FUENTES, 2016). Esse tipo de motor é utilizado principalmente em sistemas controle de posicionamento em projetos de automação. Um servo motor é um atuador que garante precisão em aplicações de controle de posição em malha fechada (ELECTRONICS HUB, 2015).

O servo motor trabalha com servo-mecanismo que usa o feedback de posição para controlar a velocidade e a posição final do motor. Internamente, um servo motor combina um motor com um circuito de realimentação, um controlador e outros circuitos complementares, ele usa um codificador ou sensor de velocidade conhecido como encoder, que tem a função de fornecer o feedback de velocidade e posição.

O sinal de realimentação por sua vez é comparado com a posição de comando de entrada, a posição desejada do motor, e produz o sinal de erro baseado na diferença entre elas. O sinal de erro disponível então gira o eixo do motor para a posição desejada, levando o novo erro a próximo de zero.

Figura 14 – Malha fechada de controle de um Servo Motor



Fonte: Adaptada de *Electronics Hub* (2015)

Os servo motores classificam-se em corrente alternada CA e corrente contínua CC, dependendo da natureza da alimentação necessária para operação. Os servo motores CC, menores, são compostos de ímã permanente, possuem escovas e são normalmente empregados em projetos menores, relacionando menores custos e maior simplicidade. Os servos com corrente alternada, mais comuns em indústrias, suportam aplicação que demandam por potência e exatidão no controle (ROGGIA; FUENTES, 2016).

No tipo de servomotor utilizado, uma tensão de referência CC é ajustada para o valor correspondente à saída desejada. Esta tensão pode ser aplicada utilizando um potenciômetro, um gerador de largura de pulso de controle (PWM) para o conversor de tensão, ou através de temporizadores dependendo do circuito de controle. A regulação do potenciômetro produz uma tensão correspondente que é então aplicada na entrada do amplificador de erro.

Figura 15 – Servo motor 9g

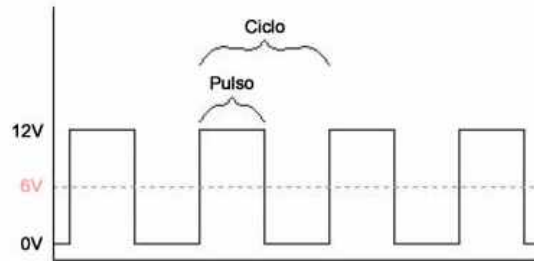


Fonte: Adaptada de *Filipe Flop* (2017)

Para realizar o controle digital, são utilizados microprocessadores ou microcontroladores para gerar os pulsos modulados, produzindo um controle mais preciso. A modulação por largura de pulso conhecida como PWM é a modulação da tensão de saída. Esse controle implica diretamente na razão cíclica, ou duty cycle, que é a fração de tempo em que este

se encontra em estado ativo, aumentando ou reduzindo a tensão média de saída do sistema (CITISYSTEMS, 2016).

Figura 16 – Modulação por largura de pulso - PWM



Fonte: Adaptada de *Citisystems* (2016)

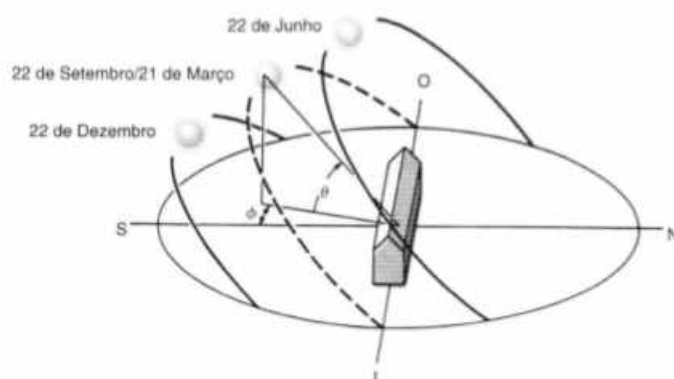
Para a imagem 16, um duty cycle de 50% é empregado, neste caso a tensão que variava inicialmente de 0V a 12V agora estará a 6V. O PWM é utilizado para gerar a tensão de referência que é seguida pelo Servo motor, posicionando-o conforme o necessário.

3.8 Energia Solar

Segundo (HINRICHS; KLEINBACH, 2003) as fontes renováveis de energia fornecem mais de 9% da energia mundial. As fontes ou recursos energéticos renováveis estão presentes na forma solar, eólica, hídrica, biomassa e geotérmica. A energia solar radiante pode ser utilizada desde o aquecimento de edificações, da água para consumo residencial e ainda para a produção de eletricidade através de células solares e geradores de calor.

A insolação recebida em um determinado local da superfície terrestre varia conforme a posição relativa ao sol, dependendo da latitude, da estação, do horário do dia e do grau de nuvens presentes. O sistema horizontal de coordenadas utiliza o horizonte de análise como um plano de referência. Ele é expresso em termos de dois ângulos: a altura ou elevação, representado por θ e o azimuth por Φ . A posição referencial de estrelas e planetas pode ser determinada pelo azimuth e elevação conhecidos através de fórmulas astronômicas para qualquer dia, local e horário.

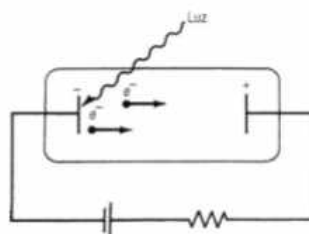
Figura 17 – Altitude e Azimute Solar



Fonte: Adaptada de *Hinrichs e Kleinbach (2003)*

A geração fotovoltaica, é a conversão de energia dos raios solares em eletricidade. O princípio de uso de energia solar para produção de energia elétrica foi descoberto em 1887 por Heinrich Hertz e explicado por Albert Einstein em 1905. Foi observado que, quando a luz atinge determinados metais, elétrons são emitidos. A este fenômeno deu-se o nome de efeito fotoelétrico.

Figura 18 – Efeito Fotoelétrico

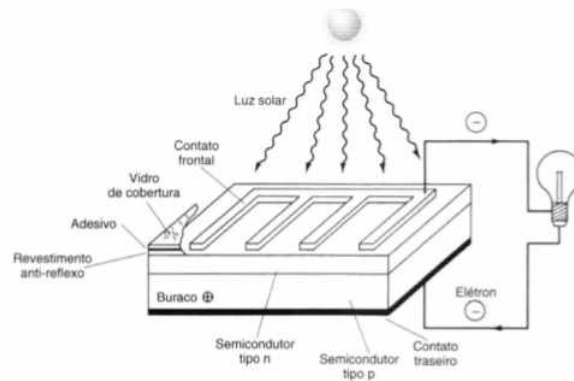


Fonte: Adaptada de *Hinrichs e Kleinbach (2003)*

Einstein explicou este efeito ao pressupondo que, a luz se comporta mais como uma partícula do que como uma onda. A maior parte das células solares é feita pelo agrupamento de camadas de silício cristalino com tratamento especial. Por de um processo denominado de dopagem, são adicionadas impurezas ao silício, como o fósforo, alterando suas propriedades e tornando-o um condutor melhor. Deste modo um semiconductor do tipo negativo, ou tipo N, no qual a carga da corrente é de elétrons negativos será criado. De modo semelhante, ao adicionar boro, menos elétrons estarão disponíveis no silício e, assim, surgirão vazios no cristal, produzindo um semiconductor do tipo positivo, ou do tipo P. Quando estes dois tipos de semicondutores são colocados juntos, eles formam uma junção P-N. Em uma célula solar atingida pela luz, elétrons e vazios são criados pelo efeito

fotoelétrico. Estas cargas são separadas pela barreira potencial na junção P-N. Como os lados P e N da célula solar estão conectados por um circuito externo, os elétrons irão fluir para fora do eletrodo N através de uma carga, realizando trabalho e para dentro do lado tipo p, onde se recombinarão aos vazios.

Figura 19 – Construção de um Pannel Fotovoltaico



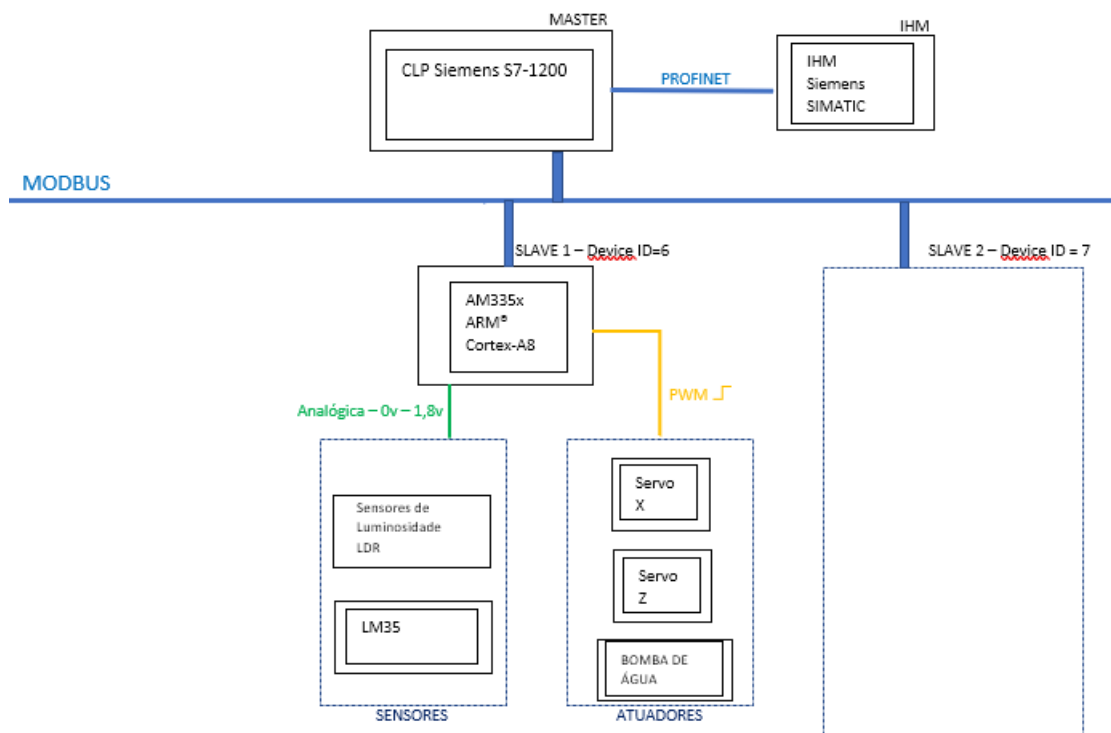
Fonte: Adaptada de *Hinrichs e Kleinbach (2003)*

4 Metodologia

4.1 Visão geral do Sistema

A realização deste trabalho aconteceu em três etapas: configuração da rede do sistema embarcado e sensoriamento, configuração da rede do mestre Modbus e por último a configuração da comunicação e das telas na IHM. a figura 20, ilustra a interface do projeto com o seu ambiente de aplicação e outros sistemas com os quais ele interagiu.

Figura 20 – Diagrama estrutural do projeto



O Hardware utilizado no projeto foi:

- IHM Siemens SIMATIC 6", modelo 6AV6372-2DF06-0DX0, versão 11.0.0
- CLP Siemens S7-1200, modelo 6ES7214-1AG40-0XB0, CPU 1214C, versão 2.0
- Microcontrolador AM335x ARM® Cortex-A8 - disposto na placa de desenvolvimento Beaglebone Black®
- Sensores de luminosidade LDR
- Sensor de temperatura LM35

- Servo motor Tower Pro 9g - SG90

Na placa de desenvolvimento foi estruturado o sensoriamento e atuação do projeto. Foram utilizados sensores de luminosidade do Tipo LDR para garantir a leitura da luminosidade e do funcionamento da lógica de posicionamento do painel fotovoltaico. Como sensor de temperatura foi utilizado o LM35. No projeto dos atuadores de posição, utilizou-se de servo-motores do tipo Tower Pro 9g - SG90.

Figura 21 – Placa de Desenvolvimento BeagleBone Black



Fonte: Adaptada de *BeagleBoard* (2014)

No sistema foram utilizadas duas redes de comunicação de protocolo aberto, uma do tipo Modbus TCP/IP e outra do tipo Profinet. A rede Modbus foi utilizada na comunicação entre o mestre da rede, o CLP, com o controlador e transmissor remoto que era escravo, aqui representado pela BeagleBone Black®, estipulou-se que a comunicação seguiria o protocolo de rede ModBus TCP por permitir uma melhor gestão e controle do envio e recepção dos dados (SILVA; FEIJOO, 2017). Já a rede Profinet, foi utilizada na comunicação entre o CLP e a IHM.

4.2 Requisitos do Sistema

Para o desenvolvimento do projeto, o primeiro passo foi estipular os requisitos funcionais. Tais requisitos modelaram desde o começo o projeto, estipulando também de modo indireto, o desenvolvimento dos testes e a validação do projeto. Eles foram estipulados através do estudo do protocolo Modbus, de modo a simular o processo de requisitos exigidos em projetos comerciais.

Tabela 1 – Tabela de Requisitos funcionais do sistema

Identificação	Nome	Prioridade
RF-01	O sistema deve possuir até 10 endereços	Essencial
RF-02	0(Zero) deve ser o endereço de Broadcast	Essencial
RF-03	1 até 10 devem ser endereços disponíveis para os escravos	Essencial
RF-04	10 até 50 devem ser endereços reservados	Essencial
RF-05	O endereço(ID) do Escravo na rede deve ser igual a 6	Essencial
RF-06	A placa de desenvolvimento Beaglebone Black deve se comunicar com o CLP através da porta 502	Essencial
RF-07	O IP da placa de desenvolvimento Beaglebone Black deve ser 192.168.0.55	Essencial
RF-08	Deve ser implementada a Function Code 03 - Read holding registers	Essencial
RF-09	Deve ser implementada a Function Code 06 - Write single register	Essencial
RF-10	O byte order deve ser Big endian	Essencial
RF-11	O frame de comunicação deve possuir a estrutura {MBAP Header} - {Function Code} - {Data}	Essencial
RF-12	O sistema deve possuir 5 entradas analógicas (AI) e 1 Saída digital (DO)	Essencial
RF-13	A posição 1 do Holding Register é correspondente ao valor médio adquirido através dos sensores de luminosidade	Essencial
RF-14	A posição 2 do Holding Register é correspondente ao valor adquirido através do sensor de temperatura	Essencial
RF-15	A posição 3 do Holding Register é correspondente ao valor da inclinação no eixo X do painel	Essencial
RF-16	A posição 4 do Holding Register é correspondente ao valor da inclinação no eixo Z do painel	Essencial
RF-17	A posição 7 do Holding Register é correspondente ao atuador do sistema de resfriamento do painel	Essencial
RF-18	A luminosidade deve ter um range máximo entre 0 e 4095 lúmens	Essencial
RF-19	A Temperatura pode variar entre 0°C e 180°C	Essencial
RF-20	A inclinação X pode variar entre 0° e 180°	Essencial
RF-21	A inclinação Z pode variar entre 0° e 180°	Essencial
RF-22	O resfriamento deve variar entre nível lógico alto ou 1 = Ligado e nível lógico baixo ou 0 = desligado	Essencial

4.2.1 Tags do projeto

Para o projeto, foram estipuladas variáveis para o sistema, todas do tipo Holding Register, sendo portanto seu endereço do tipo 4000X. A TAG utilizada foi escolhida pelo que definiu a sua funcionalidade no sistema, o tipo da variável determinou as entradas analógicas (AI) e a saída digital (DO). Por fim, foi determinado o range de trabalho.

Tabela 2 – Tags do Projeto

Posição do Holding Register	Endereço	TAG	Tipo	Range
Holding Register[1]	40001	Luminosidade	AI	0 - 4095
Holding Register[2]	40002	Temperatura	AI	0°C - 180°C
Holding Register[3]	40003	Ângulo X	AI	0° - 180°
Holding Register[4]	40004	Ângulo Z	AI	0° - 180°
Holding Register[7]	40007	Resfriamento	DO	0 - 1

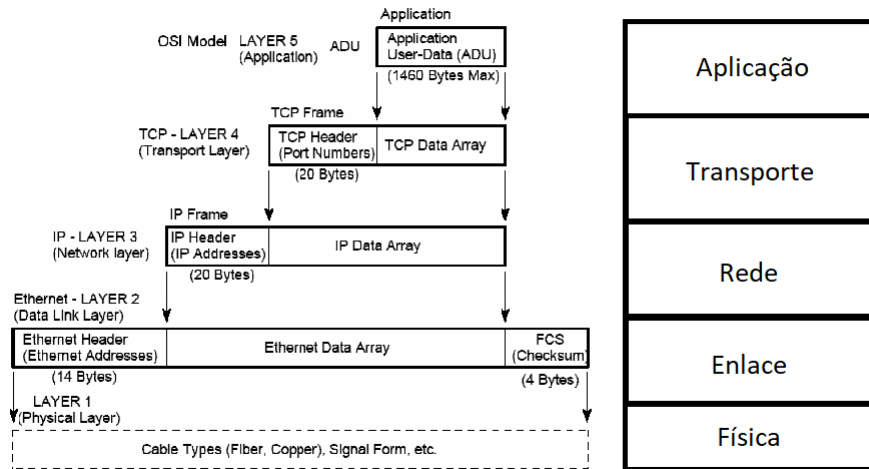
Entre as funções disponíveis no protocolo, para o projeto, foram implementadas apenas as funções Read Holding Registers (0x03) e Write Single Register (0x06) do protocolo Modbus.

4.3 Desenvolvimento da rede no sistema embarcado

Para esta etapa foi utilizado o microcontrolador AM335x ARM® Cortex-A8 disposto na placa de desenvolvimento Beaglebone Black® citado anteriormente. Primeiramente foi configurada a rede do sistema que atende como escravo, utilizando FreeModbus, uma implementação livre e gratuita do protocolo Modbus, desenvolvida para sistemas embarcados.

O FreeMODBUS implementa o protocolo de comunicação industrial de modo especialmente direcionado para sistemas embarcados. A implementação utilizada tem seu recebimento e transmissão, implementado como uma máquina de estados que é acionada por retornos de chamada da camada de abstração de hardware, fato que beneficia sua principal característica, a portabilidade (WALTER, 2010) . O Modbus TCP utilizado, envia um evento para a pilha quando um novo frame de comunicação estiver pronto para ser processado. A pilha de protocolo, em seguida, chama uma função que retorna o frame TCP Modbus recebido e o processa. Se for válida, uma resposta é criada e a camada de portabilidade é fornecida com a resposta do Modbus, enviando em seguida, a resposta de volta ao cliente ou mestre.

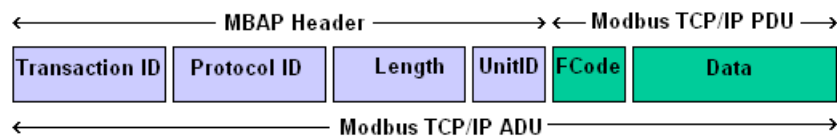
Figura 22 – Arquitetura conceitual do modelo de mensagens Modbus TCP/IP e Modelo OSI



Fonte: Adaptada de *SimplyModbus (2017)*

O protocolo utilizado possui um modelo de encapsulamento quando uma solicitação ou resposta é transportada em uma rede Modbus TCP / IP este encapsulamento é descrito conforme figura 23. O primeiro passo para a criação da rede do sistema embarcado, foi definir o MBAP nas definições do FreeModbus implementado na BeagleBone Black (figura 24).

Figura 23 – Modbus TCP/IP ADU



Fonte: Adaptada de (*MODBUS-IDA, 2006*)

Figura 24 – Definição do MBAP do Escravo TCP

```

47  /* ----- Defines ----- */
48
49  /* ----- MBAP Header ----- */
50  /*
51  *
52  * <----- MODBUS TCP/IP ADU(1) ----->
53  *      <----- MODBUS PDU (1') ----->
54  * +-----+-----+-----+-----+
55  * | TID | PID | Length | UID | Code | Data |
56  * +-----+-----+-----+-----+
57  * |      |      |      |      |      |
58  * (2)  (3)  (4)  (5)  (6)
59  *
60  * (2) ... MB_TCP_TID      = 0 (Transaction Identifier - 2 Byte)
61  * (3) ... MB_TCP_PID      = 2 (Protocol Identifier - 2 Byte)
62  * (4) ... MB_TCP_LEN      = 4 (Number of bytes - 2 Byte)
63  * (5) ... MB_TCP_UID      = 6 (Unit Identifier - 1 Byte)
64  * (6) ... MB_TCP_FUNC
65  *
66  * (1) ... Modbus TCP/IP Application Data Unit
67  * (1') ... Modbus Protocol Data Unit
68  */
69
70  #define MB_TCP_TID          0
71  #define MB_TCP_PID          2
72  #define MB_TCP_LEN          4
73  #define MB_TCP_UID          6
74
75
76  #define MB_TCP_PROTOCOL_ID  0 /* 0 = Modbus Protocol */

```

Utilizando como referência a figura 23, configurou-se o ID de transação, MB_TCP_TID como zero, o ID de protocolo MB_TCP_PID foi configurado como dois, pois esta ID é o definido para o protocolo Modbus/TCP no mestre utilizado (SIEMENS, 2017). Foi configurado também o número de Bytes MB_TCP_LEN como quatro e o ID do escravo MB_TCP_UID como seis.

Com base na tabela 1 também foi definida a porta de comunicação com o CLP.

Figura 25 – Porta de comunicação com o CLP

```

61  /* ----- Defines ----- */
62  #define MB_TCP_DEFAULT_PORT 502 /* TCP listening port. */

```

Para o projeto, foram implementadas as funções Read Holding Register (0x03) e Write single Register (0x06) do protocolo.

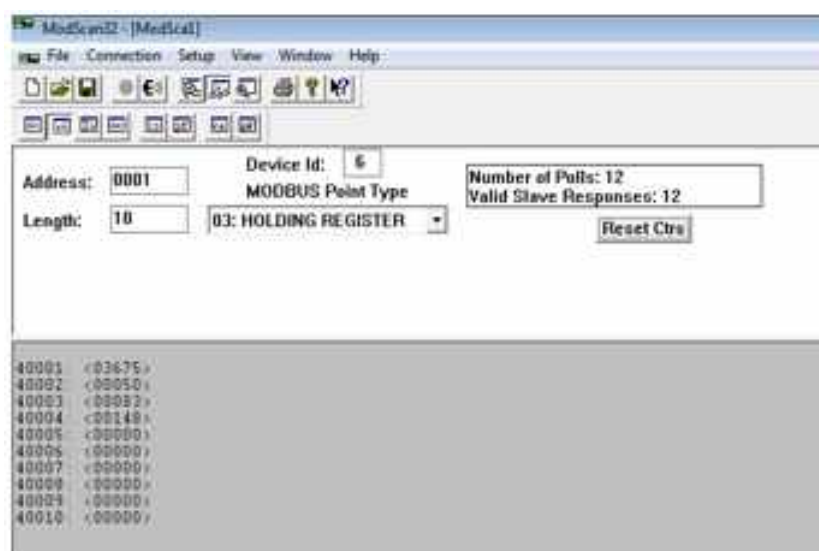
4.3.1 Teste de comunicação do Escravo TCP

Para garantir o funcionamento correto da Comunicação de rede, para o primeiro momento foram adicionados números fixos aos registradores, na finalidade de verificar a

efetividade da comunicação de rede. O passo seguinte foi realizar um teste de comunicação de rede com um software instalado no computador.

Para o teste foi utilizando um Software Mestre para a comunicação, o ModScan32 sobre uma versão gratuita de avaliação. Foi necessário configurar o software considerando o device ID, IP e Porta utilizados. Depois de iniciada a comunicação, foi verificado se os valores recebidos pelo software nos registradores correspondiam aos respectivos valores fixos adicionados no escravo.

Figura 26 – Teste de Leitura dos quatro parâmetros AI



O teste validou dois pontos importantes, o primeiro era a abertura do processo de comunicação e o segundo era do funcionamento correto da função através da correta leitura dos Holding Registers. Depois de validada, os valores de leitura dos sensores foram adicionados nos respectivos registradores (figura 27).

Podemos verificar na figura 26 os valores das leituras nos registradores 40001, 40002, 40003 e 40004.

Figura 27 – Código parcial de configuração dos valores nos holding registers

```

283 int ADC_read ( UCHAR sensor, unsigned int pin )
284 {
285     snprintf(buf,sizeof(buf),SYSFS_ADC_DIR"/in_voltage%d_raw",pin);
286     fd = open(buf,O_RDONLY);
287     read(fd,ch0,4);
288     sensor = atoi(ch0);
289     close(fd);
290     return(sensor);
291 }
292 eMBCErrorCode
293 eMBRegHoldingCB( UCHAR * pucRegBuffer, USHORT usAddress, USHORT usNRegs, eMBRegisterMode eMode )
294 {
295     eMBCErrorCode eStatus = MB_ENOERR;
296     int iRegIndex;
297     int fd;
298     char buf[MAX_BUF];
299     char ch0[5];
300     int ich0;
301     char ch;
302     int in_min = 0, in_max= 4095, out_min, out_max, aux;
303
304     vLDR1 = ADC_read (LDR1,LDR1_pin);
305     vLDR2 = ADC_read (LDR2,LDR2_pin);
306     vLDR3 = ADC_read (LDR3,LDR3_pin);
307     vLDR4 = ADC_read (LDR4,LDR4_pin);
308
309     AvLDR = (vLDR1 + vLDR2 + vLDR3 + vLDR4)/4;
310
311     //usRegHoldingBuf[0] = atoi(ch0);
312     usRegHoldingBuf[0] = AvLDR;
313     usRegHoldingBuf[1] = ADC_read(LM35,LM35_pin);
314     usRegHoldingBuf[2] = Xangle;
315     usRegHoldingBuf[3] = Zangle;
316
317
318     if( ( usAddress >= REG_HOLDING_START ) &&
319         ( usAddress + usNRegs <= REG_HOLDING_START + REG_HOLDING_NREGS ) )
320     {
321         iRegIndex = ( int )( usAddress - usRegHoldingStart );
322         switch ( eMode )
323         {
324             /* Pass current register values to the protocol stack. */
325             case MB_REG_READ:
326                 while( usNRegs > 0 )
327                 {
328                     *pucRegBuffer++ = ( UCHAR ) ( usRegHoldingBuf[iRegIndex] >> 8 );
329                     *pucRegBuffer++ = ( UCHAR ) ( usRegHoldingBuf[iRegIndex] & 0xFF );
330                     iRegIndex++;
331                     usNRegs--;
332                 }
333                 break;
334
335             /* Update current register values with new values from the
336              * protocol stack. */

```

Na imagem 27, a cada request recebido, a leitura dos sensores através da função `ADC_read` é realizada. Posteriormente são atribuídos os valores obtidos na leitura aos respectivos registradores.

A segunda etapa de testes para o escravo, também foi utilizado o software ModScan32 para avaliar a função de escrita dos registradores. Para validar a escrita, um led seria ligado indicando que o processo foi realizado. O valor escrito no led pode ser verificado no

registrador 40007.

Figura 28 – Teste de Escrita no canal DO

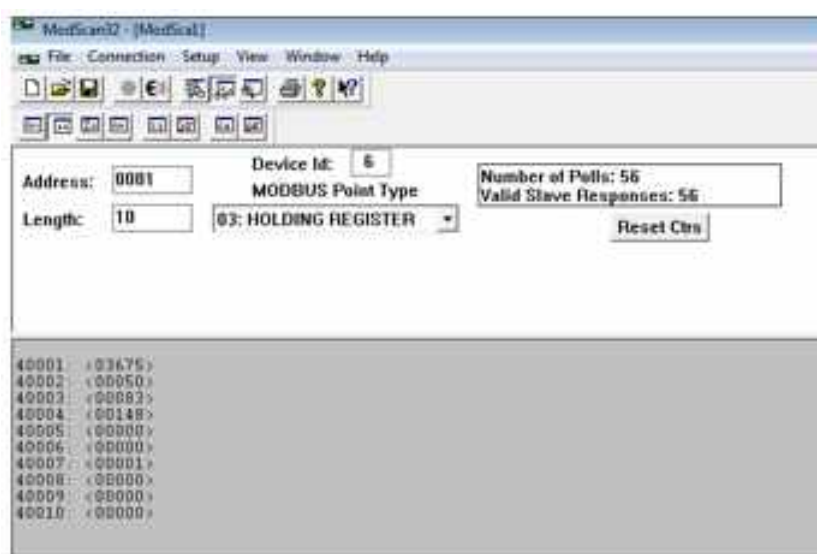
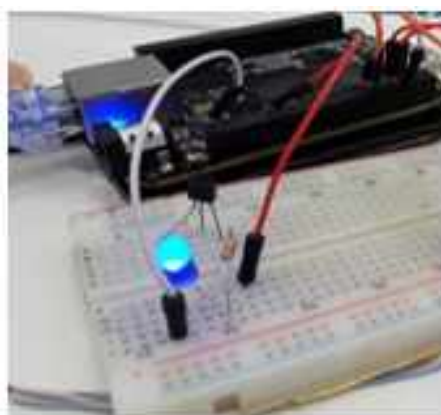


Figura 29 – Montagem do Sistema para Teste



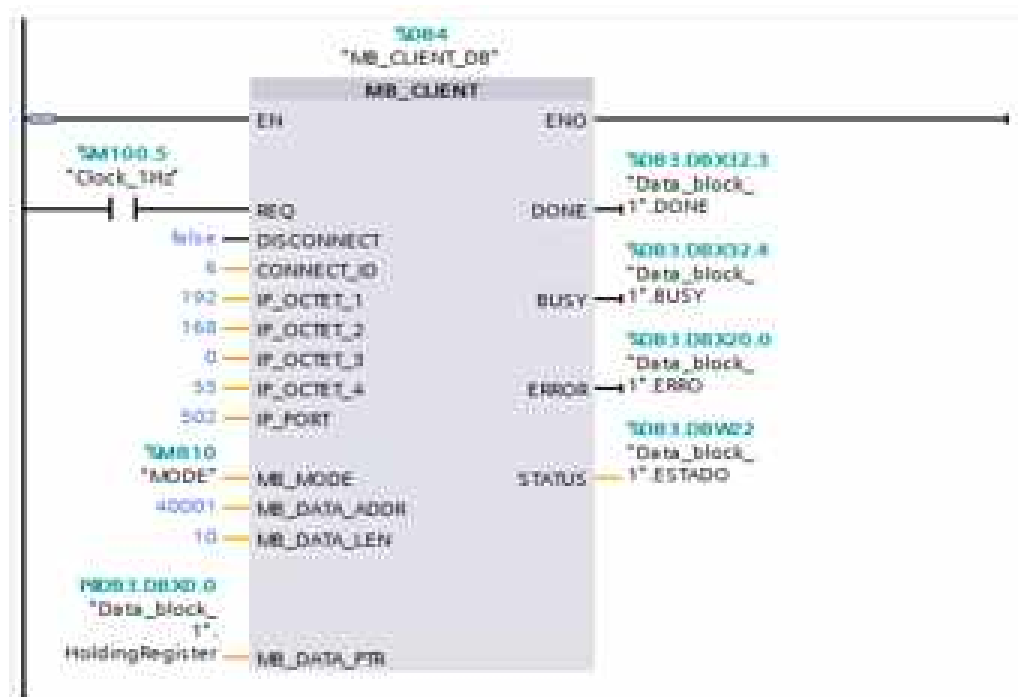
Como o sistema desenvolvido comunicou sem erros ao software ModScan32, foi então validada a comunicação por parte do escravo e foi possível prosseguir o desenvolvimento do projeto.

4.4 Configuração do Mestre Modbus TCP

No modelo desenvolvido, o mestre tem o papel de coletar as informações dos pontos Remotos ou escravos e publicá-los nas IHM. Assim como (SILVA; FEIJOO, 2017) foi utilizado para esse projeto um CLP Siemens S7-1200. Foi então desenvolvida uma lógica utilizando o bloco MbClient presente no Software TIA portal. Para o projeto estipulou-se

que a cada segundo um request seria mandado, deste modo foi colocado um Clock de 1 Hz na entrada REQ do bloco. De modo semelhante ao que aconteceu na fase de testes com o escravo disposto na BeagleBone Black, foi necessário configurar o bloco com os parâmetros do escravo, device ID, IP , a porta de comunicação, o Data Address e o Data Length. Após a Configuração do Mestre ModBus, foi necessário validar o processo de abertura de comunicação da rede e envio do Request para o escravo TCP. Foi observado o Estado do bloco Mbclient. Como o Bloco não apresentava erro, validou-se a abertura da comunicação.

Figura 30 – Configuração e teste do bloco MbClient do software TIA Portal



A segunda fase de teste utilizou a ferramenta *WatchTable* do software de programação TIA portal para observar se os valores lidos dos registradores, são os valores fixos colocados nos registradores do escravo.

Figura 31 – WatchTable do mestre TCP - TIA PORTAL

Sunpower > PLC_1 [CPU 1214C AGDCH] > Watch and force tables > Watch table_1

	Name	Address	Display format	Monitor value
1	"Anima"	%MW1	DEC	1024
2	"Data_block_1".HoldingRegister[1]	%DB3.DBW0	DEC	3675
3	"Data_block_1".HoldingRegister[2]	%DB3.DBW2	DEC	50
4	"Data_block_1".HoldingRegister[3]	%DB3.DBW4	DEC	83
5	"Data_block_1".HoldingRegister[4]	%DB3.DBW6	DEC	146
6	"Data_block_1".HoldingRegister[7]	%DB3.DBW12	DEC	1

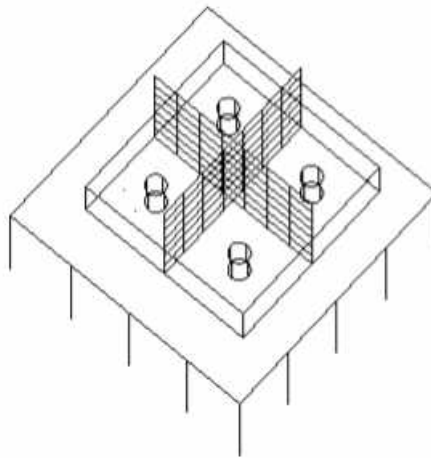
Os valores foram verificados validando esta etapa no desenvolvimento.

4.4.1 Desenvolvimento do protótipo para o sistema físico

O próximo Passo foi desenvolver um protótipo para o sistema físico, que abrigou o sensoriamento de iluminação, temperatura e atuação através da placa de desenvolvimento BeagleBone Black. Como o objetivo de estudo era o sistema de controle e redes de comunicação industriais, não houve a necessidade de instalar e testar a efetividade energética do sistema, visto que isso já foi comprovado em estudos como o de (NICÁCIO; CARVALHO, 2011) mostrado anteriormente.

Semelhante ao proposto por (NICÁCIO; CARVALHO, 2011), foi desenvolvido um compartimento para os sensores de luminosidade, utilizando anteparos entre eles. O modo com que são dispostos os sensores, possibilita a interpretação da direção dos raios, e assim fica possível indicar a direção correta de posicionamento do painel. Para esse projeto, o protótipo atua em 2 eixos de inclinação, ambos limitados ao movimento de 180°.

Figura 32 – Vista isométrica do compartimento dos sensores



Fonte: Adaptada de (NICÁCIO; CARVALHO, 2011)

A posição do painel é indicada pela posição dos servos, utilizando-se da lógica e do duty cycle enviado, para obter o valor correspondente aproximado em graus. Para os servos utilizados, com um PWM de 50Hz, a posição totalmente a esquerda era dada por um duty cycle de 2%, enquanto com aproximados 13% a posição toda a direita era definida. Para especificar o ângulo utilizando o duty cycle, ajustou-se uma função para definir o desejado.

Para o ângulo de 0 graus no servo, deve-se aplicar um duty Cycle de 2% . De modo semelhante para 180 graus, deve ser aplicado um duty cycle de 13%.

Pode-se então ajustar uma função qualquer que liga esses pontos, permitindo o cálculo do duty cycle para qualquer ângulo desejado. A inclinação dos dois pontos acima seria:

$$M = (y2 - y1)/(x2 - x1) = (13 - 2)/(180 - 0) = 11/180 \approx 0.6 \quad (4.1)$$

Onde,
M = Inclinação da Reta
y1 = duty cycle posição a esquerda
y2 = duty cycle posição a direita
x1 = ângulo posição a esquerda
x2 = ângulo posição a direita

Assim fica:

$$y - y1 = M(X - x1) \quad (4.2)$$

$$y - 2 = 0.6(X - 0) \quad (4.3)$$

$$y = 0.6 * X + 2 \quad (4.4)$$

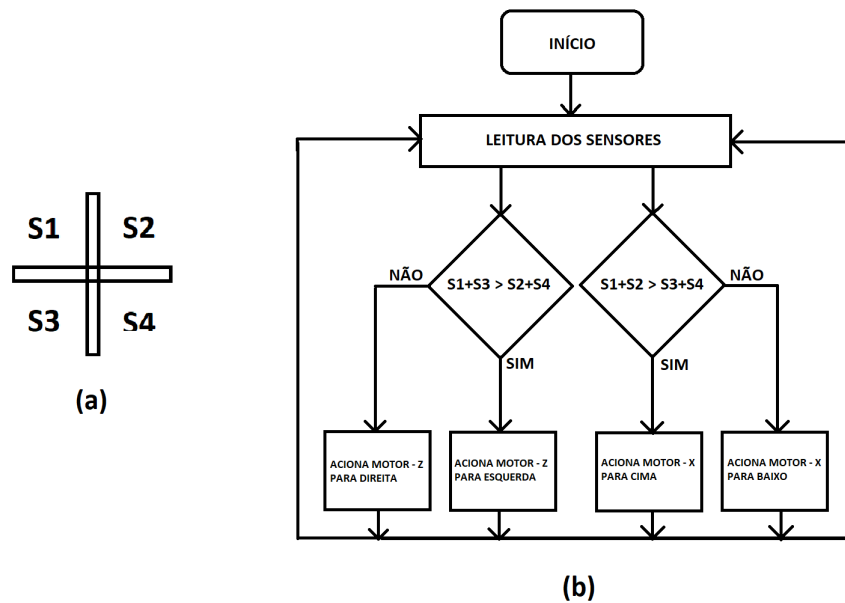
Onde,
y = duty cycle
x = ângulo desejado

Por fim, fica:

$$\hat{AnguloDesejado} = (DutyCycle - 2)/0.6 \quad (4.5)$$

Os sensores luminosos foram utilizados para a lógica de controle de movimento da placa e a média da leitura deles é enviada para o registrador respectivo.

Figura 33 – (a) Vista frontal do compartimento de Sensores. (b) Fluxograma da Lógica de posicionamento



Para a confecção da estrutura que suportaria o projeto, utilizou-se como base o projeto de modelagem 3D do posicionador solar proposto por (GRANDPIERRE NICOLAS AND CREATIVE COMMONS, 2015) , de desenvolvimento aberto e gratuito.

Figura 34 – Modelo 3D do sistema



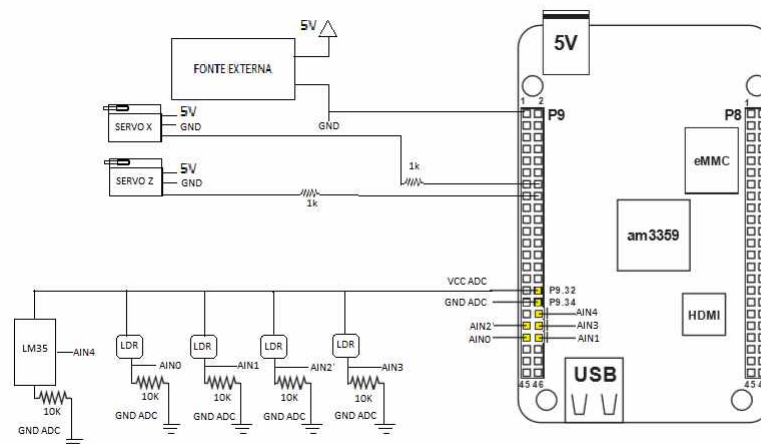
Fonte: Adaptada de (GRANDPIERRE NICOLAS AND CREATIVE COMMONS, 2015)

O tamanho reduzido, o baixo custo e o ótimo desempenho além da extrema adaptabilidade da placa de desenvolvimento BeagleBone tornaram-na uma plataforma de desenvolvimento excepcional. A BeagleBone tem diversas aplicações em sistemas de

domótica, robótica, sensoriamento ambiental e ainda diversas aplicações que quando utilizam do sensoriamento correto aumentam drasticamente as possibilidades de utilização (MOLLOY, 2014).

O sistema físico proposto foi pensado apenas para a situação de protótipo, para a construção de um sistema real de posicionamento, seria necessário o correto dimensionamento do projeto, no entanto a lógica de funcionamento de todo o sistema poderia ser mantido, mudando apenas o acionamento dos drives, motores e o sistema de posicionamento. No protótipo foram utilizadas cinco entradas analógicas, do modo como foi supracitado conforme a tabela 1, as entradas foram utilizadas para os sensores do sistema, quatro delas para o sensoriamento de luminosidade e uma para o sensor de temperatura. As três saídas digitais utilizadas foram direcionadas para o acionamento dos atuadores do projeto.

Figura 35 – Diagrama elétrico do sistema protótipo



Como o sistema criado para esse projeto era um protótipo, foram utilizados sensores de prototipagem, como os LDRs e o LM35. No entanto, respeitando os limites da placa e utilizando-se de circuitos adequados seria possível utilizar sensores comerciais para o sistema proposto, assim como o acionamento de motores maiores.

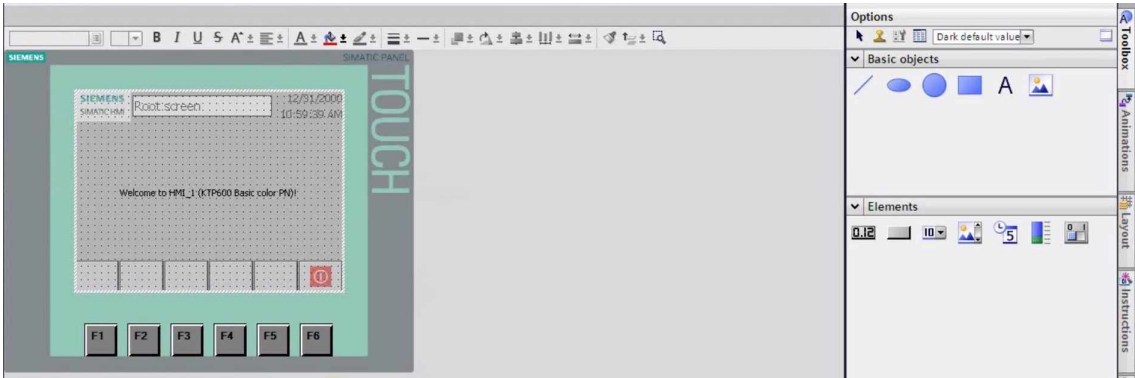
4.5 Desenvolvimento da IHM

Na etapa final de desenvolvimento do projeto foram criadas as telas de supervisão em na IHM SIMATIC da Siemens. A IHM conectou os sistemas anteriormente criados a um ambiente de supervisão intuitivo e responsivo ao toque, levando ao usuário as informações dos sensores, como o status do posicionamento do sistema, temperatura e alarmes.

Assim como (TAPIA, 2013) o desenvolvimento das telas foi realizado através do software TIA portal, assim como a aquisição inicial do dados dos sensores no processo de

validação.

Figura 36 – Desenvolvimento IHM - TIA portal



As tags do projeto foram associadas as variáveis criadas na IHM e pode-se prosseguir para o teste.

4.5.1 Teste da IHM

Os testes para a IHM foram realizados ao mesmo tempo que os teste do mestre TCP, enquanto os valores nos registradores ainda estavam fixados . Neste teste foi validado se as variáveis associadas eram lidas corretamente, uma vez que a TAG estivesse associada, ela deveria aparecer com valor do Holding Register correspondente(figura 37).

Figura 37 – Teste de leitura das variáveis da IHM

Sunpower ▶ PLC_1 [CPU 1214C AC/DC/Rly] ▶ Watch and force tables ▶ Watch table_1

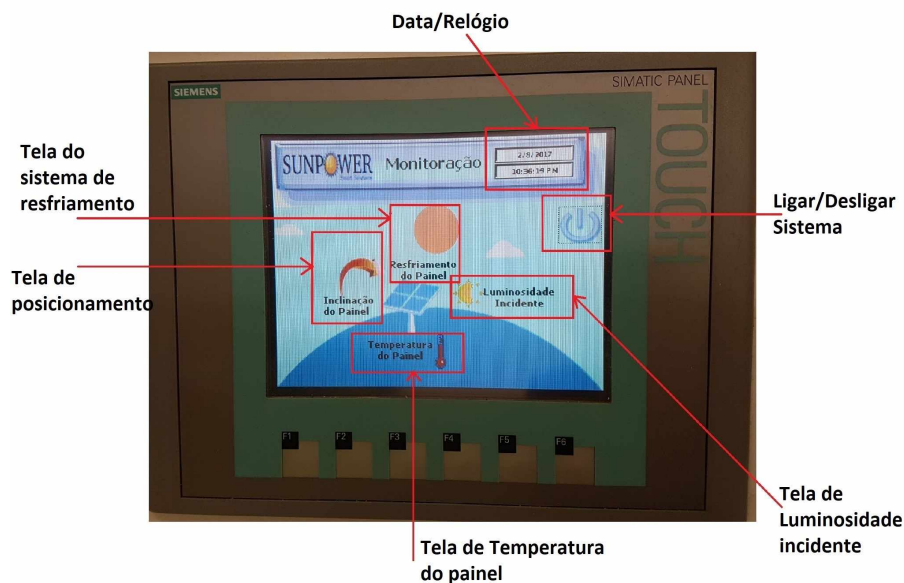
	i	Name	Address	Display format	Monitor value
1		"Anima"	%MW1	DEC	1024
2		"Data_block_1".HoldingRegister[1]	%DB3.DBW0	DEC	3675
3		"Data_block_1".HoldingRegister[2]	%DB3.DBW2	DEC	50
4		"Data_block_1".HoldingRegister[3]	%DB3.DBW4	DEC	83
5		"Data_block_1".HoldingRegister[4]	%DB3.DBW6	DEC	146
6		"Data_block_1".HoldingRegister[7]	%DB3.DBW12	DEC	1
7		"Luminosidade"	%IW102	DEC	3675
8		"Temperatura"	%IW104	DEC	50
9		"InclinaX"	%IW128	DEC	83
10		"InclinaZ"	%IW130	DEC	146
11		"Resfriamento"	%IW126	DEC	1

5 Resultados e Discussões

O sistema montado para o projeto integrou diversas tecnologias a uma rede industrial. Como a interface do projeto com o usuário foi a IHM, a melhor forma de demonstrar o funcionamento do sistema e os resultados obtidos no projeto se dá através das telas. Estas contemplam todas as funcionalidades propostas no trabalho, as quais serão exibidas através das imagens nesta seção.

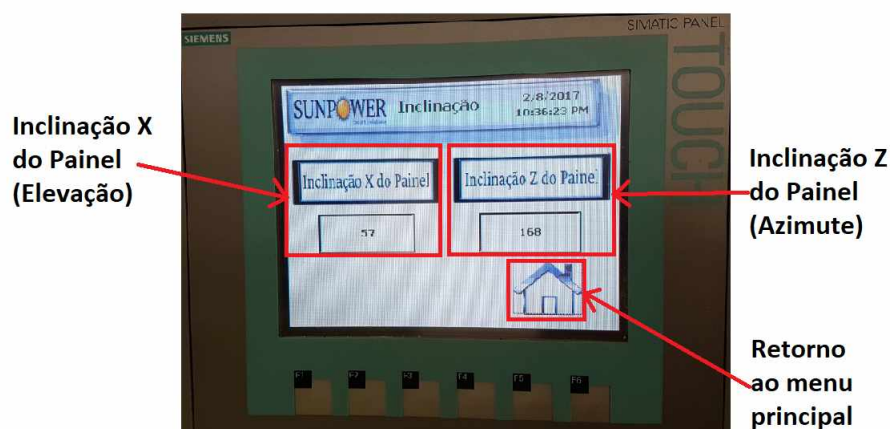
A tela inicial agrupa as funcionalidades do sistema, possibilitando que o usuário escolhesse a função desejada a ser monitorada. Diferente de um sistema de supervisão industrial, escolheu-se que a interface tivesse um sistema de subtelas, semelhante ao que acontece em um smartphone, tornando a experiência mais agradável ao usuário doméstico.

Figura 38 – Tela inicial de utilização da IHM



Caso o usuário escolha para ser monitorada seja a inclinação dos painéis, a tela da figura é aberta mostrando a posição atual do painel, através da inclinação X e da inclinação Z, que variam de 0 a 180 graus.

Figura 39 – Tela de monitoração do posicionamento do painel fotovoltaico



No caso da variável de monitoração desejada ser a temperatura, a tela desta variável irá se abrir como mostrado na figura . Nela, o usuário pode monitorar o valor pelo display ou ainda por um gráfico tipo barra. Se durante a monitoração da temperatura ela passar dos 100 graus Celsius um alarme de temperatura crítica será mostrado na tela e o sistema de resfriamento será automaticamente ligado como mostrado na figura.

Figura 40 – Tela de monitoração da temperatura do painel fotovoltaico

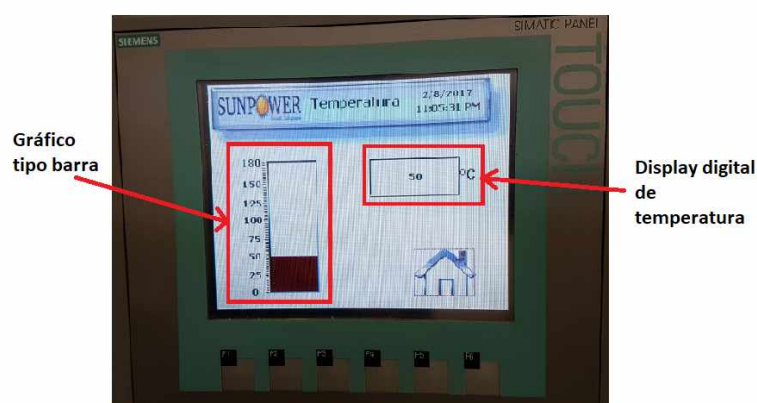


Figura 41 – Tela de monitoração da temperatura do painel fotovoltaico - Alarme de temperatura crítica



Figura 42 – Tela do sistema de resfriamento do painel fotovoltaico



Para o sistema de resfriamento, o acionamento acontece automaticamente por alarme de temperatura e ainda é possível acionar-lo através do botão presente na tela (figura 42), de modo que o usuário periodicamente possa verificar o acionamento correto do sistema de resfriamento da placa, inspecionando o sistema de modo a prevenir falhas.

Outra tela que integra o sistema é a de monitoração da Iluminação incidente dos painéis, esta mostra através de um display a iluminância captada pelos sensores. Uma animação também compõe a indicação da leitura dos sensores.

Figura 43 – Tela de monitoração da luminância incidente no painel fotovoltaico - iluminação baixa

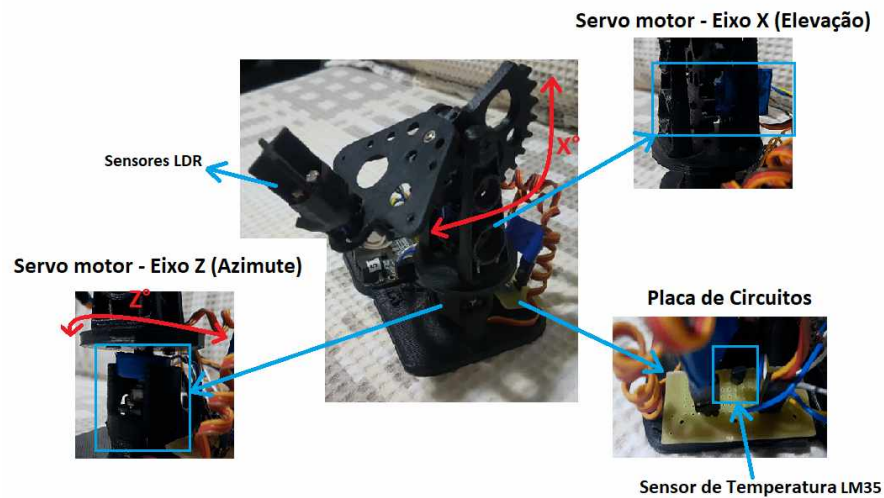


Figura 44 – Tela de monitoração da luminância incidente no painel fotovoltaico - iluminação Alta



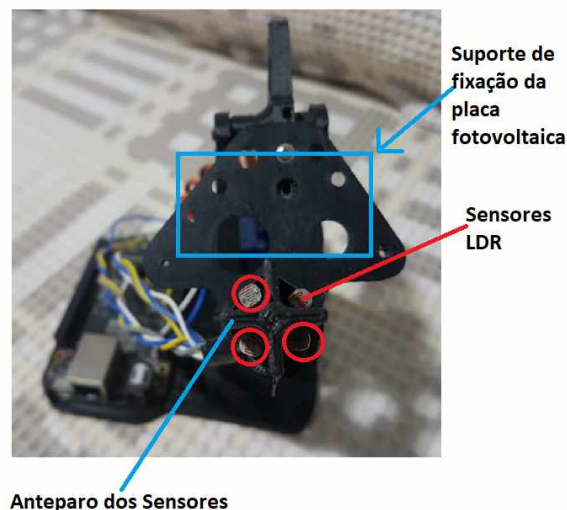
Para a validação completa do projeto foi necessário a criação de um protótipo que permitisse que o sistema criado, utilizando dos sensores e atuadores propostos, pudesse posicionar corretamente a placa de modo a receber a maior iluminação possível. O protótipo foi construído a partir de um modelo 3D e impresso utilizando as dimensões necessárias para abrigar a placa de desenvolvimento e os servo motores utilizados. Para o sistema de posicionamento da placa, no projeto foi incorporada uma relação de transmissão por engrenagens garantindo um maior torque para o posicionamento do eixo Z, possibilitando que o motor de passo não ficasse sobrecarregado.

Figura 45 – Protótipo 3D de suporte da placa fotovoltaica



O sistema que abrigava os sensores LDR, foi projetado de modo semelhante ao proposto por (NICÁCIO; CARVALHO, 2011), e mostrou-se eficiente na aquisição dos valores, possibilitando que a lógica de posicionamento funcionasse corretamente.

Figura 46 – Compartimento dos Sensores LDR no protótipo



Por fim, com todo o sistema integrado foi possível posicionar automaticamente o suporte do painel fotovoltaico, garantindo que a maior iluminância possível fosse obtida. O usuário foi capaz de supervisionar o processo através da tela da IHM e observou-se a variação correta da leitura dos sensores e acionamento do led que representava um relé de acionamento da bomba de resfriamento. O sistema final se mostrou viável e escalável, concluindo o objetivo deste projeto.

6 Conclusão e Trabalhos Futuros

A crescente demanda pela automação, aliada a necessidade de se reduzir os custos foram motivadoras desse projeto. O estudo aqui desenvolvido proporcionou principalmente a análise das dificuldades e também eficácia de sistemas mais simples e de baixo custo, aliados a tecnologias inovadoras.

Notou-se que no desenvolver do projeto, o microcontrolador AM335x ARM® Cortex-A8 disposto na placa de desenvolvimento Beaglebone Black® realizou com êxito sua função, sendo capaz de responder aos requisitos de rede e abrigando os sistemas físicos do projeto como sensoramento e sinais para os atuadores. O protótipo criado, utilizava sensores de luminosidade e temperatura para protótipos e atuadores de posição como servo-motores mais simples, porém representava um modelo fiel ao que pode ser criado utilizando de sensores comerciais de aplicação, tornando possível a construção real do aqui proposto.

De modo semelhante ao microcontrolador, a IHM mostrou-se eficaz em permitir a melhor monitoração do usuário, promovendo um ambiente mais agradável e simples de visualização das variáveis do projeto. Da mesma forma, a rede Modbus utilizada, mostrou-se robusta e eficiente para o que era proposto, levando confiabilidade ao projeto. Os objetivos propostos neste estudo foram totalmente realizados, mostrando os desafios de uma implementação que incorporava o estudo de redes industriais, sistemas embarcados e energias renováveis, aliando temas muito importantes e que estiveram presentes no decorrer dos estudos do curso.

Considerando a ampla aplicação do aqui proposto, em pequenas industriais e até mesmo o residencial, para trabalhos futuros sugere-se a implementação de outros sistemas de automação e sua integração a rede que aqui foi proposta e criada, tornando possível um ambiente integrado de automação em diferentes funcionalidades e utilização. Todo o projeto acessível através da IHM para a facilidade do usuário.

Deste modo, os avanços tecnológicos que possibilitam estudos como esse, permitem que a cada dia surjam sistemas ainda mais eficientes e inteligentes, tornando possível soluções automatizadas, mais rentáveis e utilizáveis por todas as pessoas.

Referências

- ARM. *The ARM Architecture*. [S.l.], 2006. Disponível em: <<https://www.arm.com>>. Citado na página 28.
- BEAGLEBOARD. Beaglebone black. 2014. Acesso em 12 de abril de 2018. Disponível em: <<https://beagleboard.org/black>>. Citado na página 36.
- BRISCOE, N. *Understanding The OSI 7-Layer Model*. [S.l.]: ITP journals, 2000. Citado na página 23.
- CITISYSTEMS. *PWM*. [S.l.], 2016. Acesso em 19 de junho de 2018. Disponível em: <<https://www.citisystems.com.br>>. Citado na página 32.
- ELECTRONICS HUB. *Servo Motors*. [S.l.], 2015. Acesso em 19 de junho de 2018. Disponível em: <<https://www.electronicshub.org/servo-motors>>. Citado 2 vezes nas páginas 30 e 31.
- FILIPE FLOP. *Sensores*. [S.l.], 2017. Acesso em 19 de junho de 2018. Disponível em: <<https://www.filipeflop.com/produto/>>. Citado 3 vezes nas páginas 29, 30 e 31.
- GAZOLI, J. R.; VILLALVA, M. G. *Energia Solar Fotovoltaica - Conceitos e Aplicações - Sistemas Isolados e Conectados À Rede*. [S.l.]: Editora Érica, 2015. v. 2. Citado na página 14.
- GRANDPIERRE NICOLAS AND CREATIVE COMMONS. *Mini Arduino Solar Tracker 2 Axis*. [S.l.], 2015. Acesso em 19 de junho de 2018. Disponível em: <<https://www.thingiverse.com/thing:736972>>. Citado na página 47.
- HINRICHS, R.; KLEINBACH, M. *Energia e Meio Ambiente*. [S.l.]: Pioneira Thomson Learning, 2003. ISBN 9788522103379. Citado 3 vezes nas páginas 32, 33 e 34.
- JÚNIOR, A. S. *CLP - Linguagens de Programação*. [S.l.]: IFTM - Rio Grande do Norte, 2016. Citado na página 21.
- JÚNIOR, J. P. da S. *Controladores Lógico Programáveis*. [S.l.]: SENAI CETAFR, 2011. Citado na página 20.
- LUGLI, A. B. *Uma visão do protocolo industrial Profinet e suas aplicações*. Sense Eletrônica LTDA - INATEL, 2008. Disponível em: <http://www.profibus.org.br/artigos_tecnicos/uma-visao-do-protocolo-industrial-profinet-e-suas-aplicacoes>. Citado 2 vezes nas páginas 26 e 27.
- MODBUS-IDA. *MODBUS Messaging on TCP/IP Implementation Guide*. [S.l.], 2006. Disponível em: <<http://www.modbus.org>>. Citado 4 vezes nas páginas 24, 25, 26 e 39.
- MOLLOY, D. *Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux*. Wiley, 2014. ISBN 1118935128. Disponível em: <<http://www.exploringbeaglebone.com/>>. Citado na página 48.

- NICÁCIO, L. D. S.; CARVALHO, S. S. D. Sistema de posicionamento de painéis fotovoltaicos utilizando microcontrolador. 2011. Citado 4 vezes nas páginas 17, 30, 45 e 54.
- NOGUEIRA, T. A. *Redes de Comunicação para sistemas de automação industrial*. Dissertação (Trabalho de Conclusão de Curso) — Universidade Federal de Ouro Preto, 2009. Citado 4 vezes nas páginas 16, 22, 23 e 24.
- PATEL, D. N. Modbus protocol implementation for microcontroller. *International Journal of Electronics, Electrical and Computational System*, v. 6, 2017. Citado na página 16.
- PENIDO, d. d. C. C.; TRINDADE, R. S. *Microcontroladores*. IFTM - Ouro Preto, 2013. Disponível em: <http://estudio01.proj.ufsm.br/cadernos/ifmg/tecnico_automacao_industrial/microcontroladores.pdf>. Citado na página 27.
- ROGGIA, L. .; FUENTES, R. *Automação Industrial*. [S.l.]: Colégio Técnico Industrial da Universidade Federal de Santa Maria - Rede E-Tec Brasil, 2016. Citado 5 vezes nas páginas 19, 21, 22, 30 e 31.
- SIEMENS. *Modbus/TCP with Instructions*. [S.l.], 2017. Disponível em: <<https://support.industry.siemens.com/cs/document/102020340>>. Citado na página 40.
- SIEMENS. *SIMATIC HMI Panel*. [S.l.], 2018. Acesso em 19 de junho de 2018. Disponível em: <<https://w3.siemens.com/mcms/human-machine-interface/en/special-hmi-devices/ex-zone-devices/hmi-panel-pc-ex/pages/default.aspx>>. Citado na página 22.
- SILVA, C. A. C.; FEIJOO, J. E. A. *Implementación de una Red MODBUS TCP Wireless con integración de tecnología Siemens Y Schneider para prácticas estudiantiles del laboratorio de automatización industrial*. Dissertação — Universidad Politécnica Salesiana, 2017. Citado 3 vezes nas páginas 18, 36 e 43.
- SIMPLYMODBUS. *About Modbus TCP/IP*. [S.l.], 2017. Acesso em 12 de abril de 2018. Disponível em: <<http://www.simplymodbus.ca/TCP.htm>>. Citado na página 39.
- TAMBOLI, S. Implementation of modbus rtu and modbus tcp communication using siemens s7-1200 plc for batch process. *Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on. IEEE*, p. 258–263, 2015. Citado na página 18.
- TANENBAUM, A.; WETHERALL, D.; TRANSLATIONS, O. *Redes de computadores*. [S.l.]: PRENTICE HALL BRASIL, 2011. ISBN 9788576059240. Citado na página 22.
- TAPIA, M. X. D. L. C. *Implementación de un HMI utilizando un Touch Panel KTP 600 PN y el PLC S7-1200 para el monitoreo y control de temperatura*. Dissertação (Trabalho de Conclusão de Curso) — Universidad de las Fuerzas Armadas ESPE, 2013. Citado 3 vezes nas páginas 18, 21 e 48.
- TEXAS INSTRUMENTS. *AM335x Sitara™ Processors datasheet*. [S.l.], 2016. Acesso em 19 de junho de 2018. Disponível em: <<http://www.ti.com/product/am3358>>. Citado 2 vezes nas páginas 28 e 29.
- TEZA, V. R. *Alguns aspectos sobre a automação residencial: domótica*. Tese (Doutorado), 2002. Citado na página 14.

VALLDOREIX GREENPOWER. *The Benefits of Solar Trackers*. [S.l.], 2015. Acesso em 12 de junho de 2018. Disponível em: <<http://www.valldoreix-gp.com/the-benefits-of-solar-trackers/>>. Citado na página 17.

WALTER, C. *FreeMODBUS - A Modbus ASCII/RTU and TCP implementation*. 2010. Disponível em: <<https://www.freemodbus.org/>>. Citado na página 38.