
Centro Estadual de Educação Tecnológica Paula Souza
Etec Prof. Dr. José Dagnoni
Desenvolvimento de Sistemas

PLC DE BAIXO CUSTO:O Futuro da Automação

Heitor Franco Miranda da Silva, Matheus Henrique Giusti e Matheus Lima Isenco

RESUMO: O projeto visa desenvolver um controlador lógico programado (plc) de baixo custo para automação industrial, focando em pequenas e médias empresas que enfrentam desafios junto a tecnologias obsoletas e custos elevados. a pesquisa inclui a utilização de microcontroladores e componentes acessíveis, simultaneamente a estruturação de um protótipo que demonstra a eficiência da solução. a proposta é viável e se mostra mais econômica em comparação aos plcs tradicionais, oferecendo uma alternativa competente para o setor industrial.

PALAVRAS-CHAVE: Automação; Custo; Protótipo.

1 INTRODUÇÃO

1.1. Tema

Desenvolvimento de um PLC de Baixo Custo para Automação Industrial: Um Estudo de Viabilidade e Implementação Prática.

1.2. Justificativa

Atualmente nas indústrias brasileiras, PLC's ou CLP'S, Controladores Lógicos Programados no geral, se tornaram comuns e passaram a exercer um importante papel nos mais diversos setores. Porém, há empresas de pequeno e até mesmo

médio porte as quais não possuem PLC ou o possuem com hardware obsoleto e mais simples. Isso acontece, pois, apesar de não ser uma tecnologia tão cara, o custo de implementação em conjunto com gastos com acessórios, sensores e atuadores que serão utilizados pode aumentar consideravelmente o preço e dependendo da qualidade dos componentes e da complexidade do processo o qual será automatizado, esse preço se torna exorbitante.

1.3. Problema

Existe uma alternativa de menor custo em relação ao CLP que atenda às necessidades das empresas de pequeno e médio porte?

1.4. Hipótese

Uma possível solução é procurar por microcontroladores, interfaces de comunicação, sensores e atuadores todos de baixo custo e utilizá-los na montagem de um CLP mais acessível. (GOMES, 2020)

1.5. Objetivos

1.5.1. Objetivo Geral

O objetivo geral deste projeto é desenvolver uma solução para o alto custos dos PLC's, que tenha todas as funções dos atuais na industrial. O PLC buscará fornecer uma plataforma barata e eficiente para auxiliar tanto as empresas quanto todo processo automatizado.

1.5.2. Objetivos Específicos

- Pesquisar a demanda da resolução do problema;
- Definir um plano de pesquisa condizente com a problemática;
- Estabelecer os materiais do protótipo-conceito;
- Comprar materiais;
- Montar protótipo-conceito;
- Montar o refrigerador que será medido.

1.6. Revisão Bibliográfica

Controles industriais e, principalmente, Controladores Lógicos Programados (CLP ou

PLC) são, atualmente, uma base tecnológica importante para a automação de processos industriais. Mesmo com a indústria 4.0 e a internet industrial, assume-se que tais controladores vão continuar a ser requeridos para a produção do amanhã. (LANGMANN, 2019)

PLCs são amplamente usados em instalações industriais e fábricas automatizadas incluindo estruturas críticas, como tratamento de água, reatores nucleares etc. Apesar de suas qualidades, os PLCs estão se tornando uma grande preocupação depois do ataque malicioso Stuxnet ocorrido em junho de 2010. O malware Stuxnet demarcou as vulnerabilidades do PLC. Tal malware foi capaz de, furtivamente, espionar, atacar e comprometer os códigos e dispositivos relacionados ao PLC, causando sérios danos. (SERHANE, 2018)

A atual prática de desenvolvimento de controles para sistemas industriais é baseada em descrições informais e Implementação de soluções obtidas de forma empírica. Porém, a aplicação de métodos formais para síntese de controles e desenvolvimento de programas de aplicação de PLCs está se tornando crucial, por causa do aumento dos problemas de complexidade de controle associados com a alta demanda de soluções de alta qualidade, tempo reduzido de desenvolvimento e etc. (VIEIRA, 2016)

1.7. Metodologia

As tarefas serão divididas entre os 3 integrantes do grupo, sendo cada um responsável por uma parte: coleta de artigos, cronograma do protótipo e documentação do projeto. A pesquisa se baseará em livros, artigos e estudos científicos. A documentação terá base na NBR IEC 61131-3 e NBR ISO 9001. Um protótipo-conceito será construído para testes e demonstração do funcionamento do PLC em um refrigerador, já que ele é capaz de mostrar a execução de inúmeros sensores, mostrando a eficácia do PLC e a sua importância para indústria apesar de seu preço exacerbado.

1.8. Cronograma

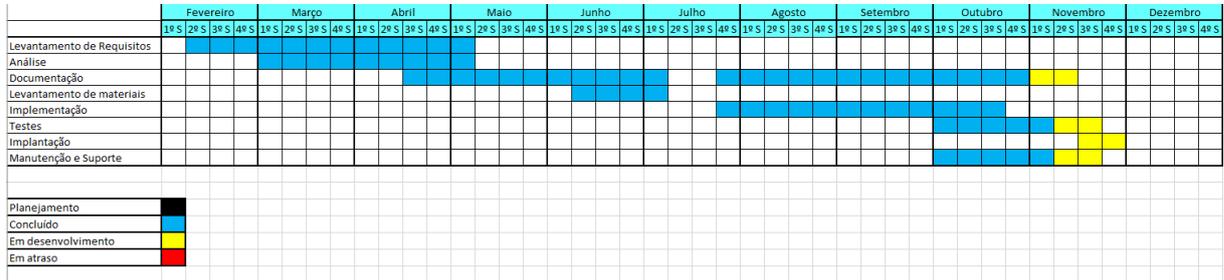


Imagem 01: Cronograma

Fonte: (Autoria Própria, 2024)

1.9. Análise de dados

A pesquisa teve como objetivo desenvolver uma solução para o alto custo dos PLC's, que atenda às necessidades das empresas de pequeno e médio porte. A hipótese é que uma possível solução seria utilizar microcontroladores, interfaces de comunicação, sensores e atuadores de baixo custo na montagem de um CLP mais acessível.

A coleta de dados envolveu a análise de livros, artigos e estudos científicos, além de informações sobre o custo de implementação, acessórios, sensores, atuadores e a complexidade dos processos a serem automatizados. As técnicas de análise incluíram estatística descritiva, análise de séries temporais e machine learning. Durante as pesquisas, foram identificadas normas importantes: a NBR IEC 61131-3, que exige documentação com diagramas Structured Text, Function Block Diagram e Ladder Diagram, e a NBR ISO 9001, que requer relatórios do projeto e documentação detalhada conforme a NBR IEC 61131-3, assegurando conformidade e rastreabilidade.

Os resultados indicaram que a solução proposta pode ser mais eficiente em termos de custo em comparação a um PLC convencional. O custo de implementação da solução proposta é menor, incluindo o custo do microcontrolador, interfaces de comunicação, sensores e atuadores. Além disso, a solução pode ser mais adequada para processos simples.

Concluiu-se que a solução proposta é viável e eficaz em termos de custo, eficiência e normas, podendo ser implementada em empresas de pequeno e médio porte que necessitem de automação industrial. No entanto, é importante monitorar

continuamente os resultados e ajustar a estratégia de análise de acordo com as necessidades.

2. DESENVOLVIMENTO

2.1. Ferramentas De Desenvolvimento De Software

2.1.1 Linguagens de Programação

- C++: Uma das linguagens de programação mais populares, C++ combina a eficiência da programação de baixo nível com as abstrações da programação de alto nível. É utilizada em diversas áreas, incluindo desenvolvimento de sistemas, jogos e aplicações de tempo real, devido à sua capacidade de manipular diretamente a memória e os recursos do hardware.

Exemplo de Aplicação: Desenvolvimento de sistemas operacionais, jogos, simulações em tempo real.

2.1.2 Frameworks

- Biblioteca MQTT (PubSubClient): Esta biblioteca é amplamente utilizada para comunicação via MQTT, um protocolo leve ideal para dispositivos com recursos limitados. A PubSubClient permite que dispositivos Arduino publiquem e recebam mensagens de um broker MQTT, facilitando a troca de dados em tempo real. Exemplo de Aplicação: Integração de sensores em um sistema de automação residencial, comunicação entre dispositivos IoT.
- Biblioteca para o Módulo Wi-Fi (ESP8266WiFi): Esta biblioteca fornece suporte para conectar placas Arduino ao Wi-Fi, permitindo que dispositivos se conectem à internet e se comuniquem com outros serviços online. É essencial para projetos que requerem conectividade em rede.

Exemplo de Aplicação: Monitoramento remoto de dados, controle de dispositivos via internet.

- DHT22: Biblioteca específica para o sensor DHT22, que mede temperatura e umidade. Permite a leitura fácil dos dados do sensor e a integração com aplicações que monitoram condições ambientais.

Exemplo de Aplicação: Sistemas de climatização, monitoramento de estufas.

- BMP280: Esta biblioteca é utilizada para o sensor BMP280, que mede pressão

atmosférica e temperatura. É ideal para aplicações que requerem medições precisas de altitude e condições climáticas.

Exemplo de Aplicação: Estações meteorológicas, drones.

- MQ-135: Biblioteca para o sensor MQ-135, que é utilizado para medir a qualidade do ar, detectando gases como amônia, benzeno e outros poluentes. Facilita a implementação de sistemas de monitoramento ambiental.

Exemplo de Aplicação: Monitoramento da qualidade do ar em ambientes urbanos, sistemas de alerta de poluição.

- OpenWatherMap(API): é uma plataforma de serviços que fornece acesso a dados meteorológicos em tempo real, previsões e históricos de tempo. Ela oferece uma série de recursos úteis para desenvolvedores, como a integração de informações climáticas em aplicativos, sites e sistemas de monitoramento.

2.1.3. Ambientes de Desenvolvimento Integrado (IDEs)

- Arduino IDE 2.0: A versão mais recente do ambiente de desenvolvimento, que apresenta uma interface moderna e responsiva. Inclui recursos como auto completar, navegação de código e um depurador em tempo real, embora a funcionalidade de depuração esteja limitada a algumas placas específicas, como a Arduino Zero.

Exemplo de Aplicação: Desenvolvimento de projetos de automação residencial, controle de robôs, e interação com sensores e atuadores.

2.1.4. Ferramentas Para Desenvolvimento Web(Frontend E Backend)

- Node-RED: Uma ferramenta de programação visual baseada em fluxo, que permite conectar dispositivos, APIs e serviços online de forma simples. Utiliza uma interface gráfica onde os usuários podem arrastar e soltar nós (nodes) para criar fluxos de dados, facilitando a automação e a integração de sistemas. Exemplo de Aplicação: Automação de processos industriais, integração de dispositivos IoT, monitoramento de dados em tempo real.
- JavaScript: A linguagem de programação utilizada no Node-RED, permitindo a manipulação de dados e a lógica de controle dentro dos fluxos. Os desenvolvedores podem escrever funções personalizadas para processar dados e interagir com APIs externas.

Exemplo de Aplicação: Criação de dashboards interativos, manipulação de dados em tempo real, desenvolvimento de aplicações web dinâmicas.

2.1.5. Materiais:

- DHT11 como Sensor de Temperatura: Utilizado para medir a temperatura ambiente com media precisão.
- Exemplo de Aplicação: Um termômetro digital que mostra a temperatura atual em um display LCD e envia alertas por e-mail se a temperatura exceder um limite predefinido.
- DHT11 como Sensor de Umidade: mede a umidade relativa do ar, permitindo monitoramento ambiental.
- Exemplo de Aplicação: Sistema de monitoramento de umidade para uma estufa, que ajusta a umidificação automaticamente para manter as condições ideais para plantas ou animais.
- Sensor de Pressão BMP280: Um sensor que mede a pressão atmosférica e a temperatura, útil para aplicações meteorológicas e de altitude.
- Exemplo de Aplicação: Uma estação meteorológica caseira que coleta dados de pressão atmosférica e temperatura, fornecendo previsões do tempo e alertas para mudanças climáticas.
- Sensor de Qualidade do Ar MQ-135: Detecta a presença de poluentes e gases nocivos, essencial para monitoramento ambiental.
- Exemplo de Aplicação: Sistema de monitoramento da qualidade do ar em ambientes internos, que alerta os usuários quando os níveis de poluentes atingem valores prejudiciais à saúde e sugere a ventilação do espaço.
- Shield IoT R2: O Shield IoT R2 é uma placa didática baseada no ESP32 (30 pinos), com entradas e saídas digitais, analógicas e I2C. Oferece conectores para sensores, display OLED e DHT11. Alimentada por 12V, inclui relés, reguladores para 5V e 3,3V e bornes KRE. Ideal para projetos IoT, destaca-se pela versatilidade e atenção aos detalhes em sua construção artesanal (CLUBE DA ELETRÔNICA, 2024).
- Exemplo de Aplicação: em um sistema de monitoramento ambiental para estufas agrícolas. Ele pode controlar ventiladores ou sistemas de irrigação

usando os relés, monitorar temperatura e umidade com o sensor DHT11 e exibir os dados em tempo real no display OLED. Sensores adicionais podem ser conectados para medir luminosidade ou níveis de CO₂, otimizando o cultivo.

- Módulo Wi-Fi ESP32: Um microcontrolador com conectividade Wi-Fi, ideal para projetos de IoT, permitindo que o Arduino se conecte à internet.
- Exemplo de Aplicação: Sistema de automação residencial que permite monitorar e controlar dispositivos em sua casa pela internet, como ajustar a temperatura do termostato ou acender as luzes remotamente através de um aplicativo. Fonte de Alimentação com Proteção (5V ou 12V): Fornece a energia necessária para os componentes, garantindo proteção contra sobrecargas.
- Protoboard e Jumpers: Facilita a montagem de circuitos temporários sem a necessidade de soldagem, permitindo testes e protótipos rápidos.
- Exemplo de Aplicação: Montagem de um circuito de teste para um novo sensor de distância ultrassônico, permitindo ajustes e verificações rápidas antes da construção do circuito final em uma placa de circuito impresso (PCB).
- Resistores e Capacitores: Componentes eletrônicos básicos que ajudam a controlar a corrente e estabilizar circuitos.
- Exemplo de Aplicação: Uso de resistores em série para limitar a corrente em LEDs em um projeto de indicador luminoso e capacitores para estabilizar a tensão em um circuito de alimentação de um motor de pequeno porte.
- Caixa de Montagem Resistente a Impactos: Protege os componentes eletrônicos e o Arduino, tornando o projeto mais durável e portátil.
- Exemplo de Aplicação: Encaixar um sistema de controle de irrigação automatizado em uma caixa resistente a impactos, garantindo que o dispositivo possa ser usado em ambientes externos sem sofrer danos devido à chuva ou quedas.
- Ventoinha ou Sistema de Resfriamento: Necessário para projetos que geram calor, garantindo que os componentes funcionem em temperaturas seguras.
- Exemplo de Aplicação: Implementação de um sistema de resfriamento com

refrigerador. Esse protótipo evidenciará a capacidade do PLC de gerenciar diversos sensores, destacando sua eficácia e importância para a indústria, mesmo diante de seu custo elevado.

2.4 Custos com Recursos Humanos

- Pesquisador em P&D (Matheus Lima): Salário médio pode variar entre R\$ 8.000 a R\$ 15.000, dependendo da experiência e da complexidade dos projetos.
- Engenheiro e Designer de Sistemas (Matheus Giusti): Engenheiro de Hardware e Software, a média salarial varia de R\$ 10.000 a R\$ 20.267.
- Gestor de Projetos (Heitor Franco): O salário médio varia entre R\$ 12.000 a R\$ 20.000, dependendo do porte da empresa e da complexidade dos projetos.

2.5 Interfaces (Telas do Software)

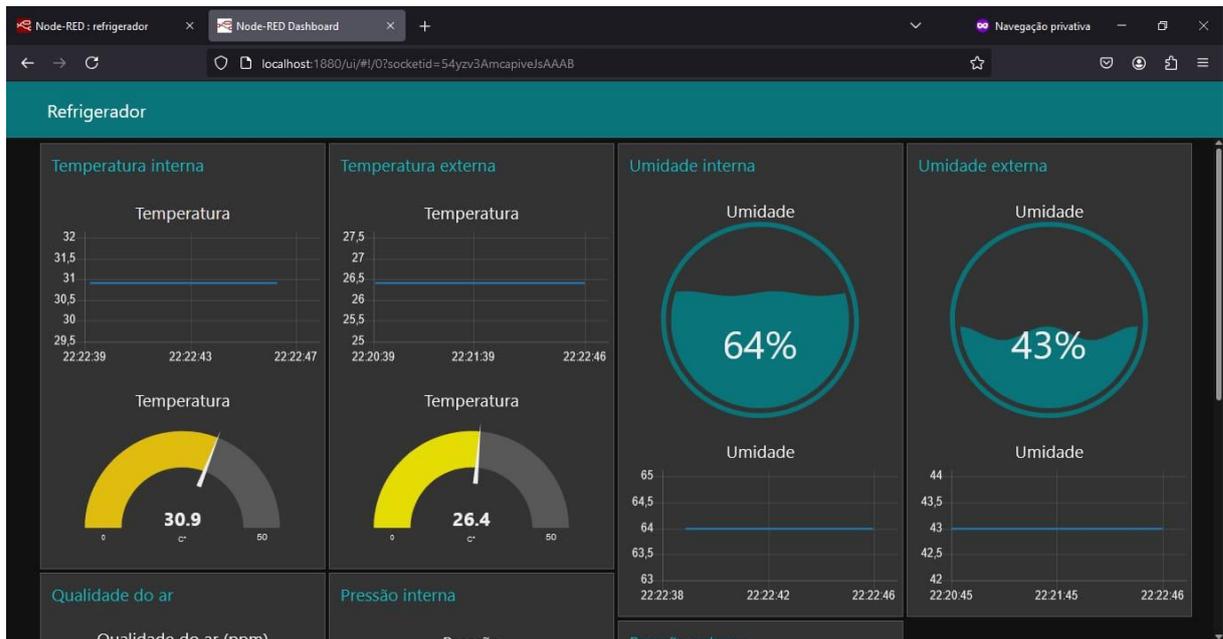


Imagem 02: Front-end web 1

Fonte: (Autoria Própria, 2024)

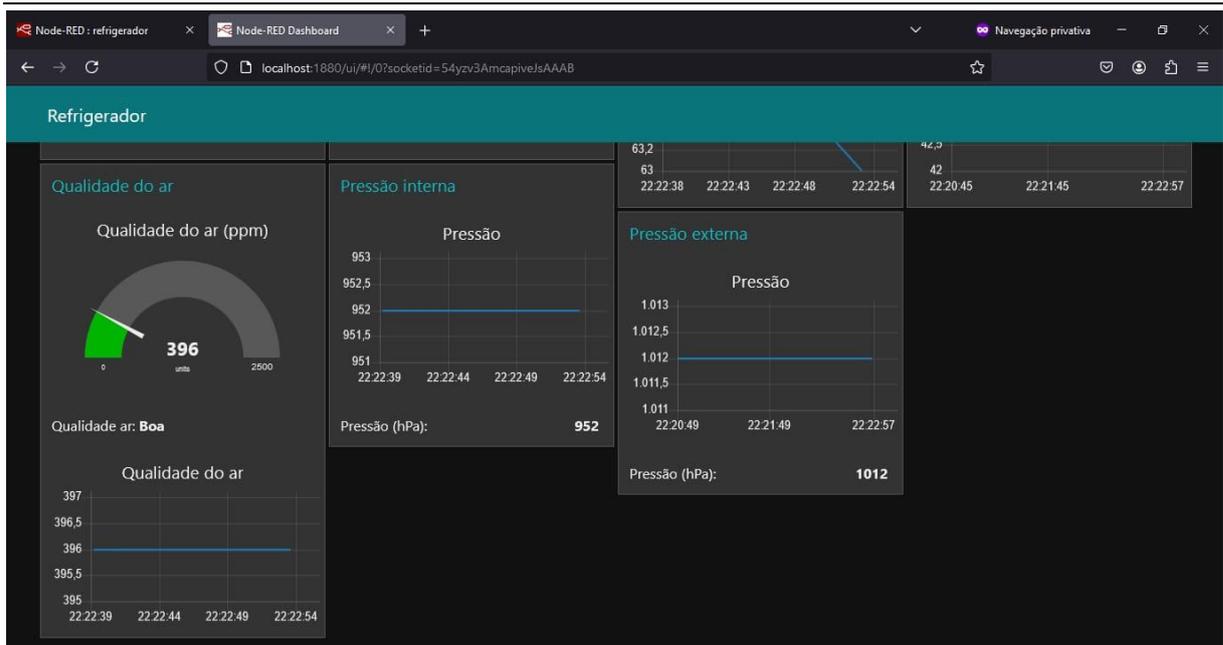


Imagem 03: Front-end web 2

Fonte: (Autoria Própria, 2024)

2.6. Missão, Valores e Visão

2.6.1. Missão

Oferecer ferramentas de automação eficientes e modernas por um preço acessível as empresas de pequeno e médio porte.

2.6.2. Valores

Empresa empenhada em oferecer soluções simples, eficientes, atuais e acessíveis, que sejam adequadas ao método de produção a ser automatizado.

2.6.3. Visão

Ser a maior empresa de inovação no crescimento da indústria Brasileira.

2.7. Logo

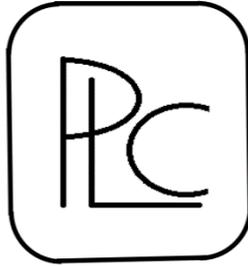


Imagem 04: Logo

Fonte: (Autoria Própria, 2024)

2.8. NBR IEC 61131-3

2.8.1. Structured Text

// Configuração Inicial

VAR

// Variáveis dos sensores

temp : REAL := 0.0; // Temperatura

umid : REAL := 0.0; // Umidade

qld_ar : INT := 0; // Qualidade do ar

pressao : INT := 0; // Pressão

// Variáveis de Qualidade do Ar

Qualidade_ar_severa : INT := 2000;

Qualidade_ar_bem_ruim : INT := 1800;

Qualidade_ar_ruim : INT := 1200;

Qualidade_ar_moderada : INT := 800;

// Variáveis de Conexão

ssid : STRING := 'your_SSID';

password : STRING := 'your_PASSWORD';

mqtt_server : STRING := 'your_MQTT_broker_IP';

```
// Estado de Conexão
isConnected : BOOL := FALSE;

// Temporizadores
delay1 : TON;
delay2 : TON;
delay3 : TON;

// Strings para publicação MQTT
umidString : STRING(10);
tempString : STRING(10);
qldArString : STRING(10);
pressaoString : STRING(10);
END_VAR

// Configuração do Sistema
METHOD setup : VOID
VAR
    status : BOOL;
END_VAR
BEGIN
    // Inicialização de Sensores
    DHT11.Initialize();
    MQ135.Calibrate();
    MQ135.GetRo();

    BMP280.Initialize(16#76);
    status := BMP280.IsConnected();

    IF NOT status THEN
        // Exibir erro caso o BMP280 não esteja conectado
        SysLog.WriteLogEntry('Erro: BMP280 não detectado.');
```

```
ELSE
    BMP280.ConfigureSampling(MODE_NORMAL, SAMPLING_X2,
    SAMPLING_X16, FILTER_X16, STANDBY_MS_500);
    END_IF

    // Configuração da Conexão WiFi
    setup_wifi();
END_METHOD

// Configuração da Conexão WiFi
METHOD setup_wifi : VOID
BEGIN
    SysLog.WriteLogEntry('Conectando ao WiFi...');
    WiFi.Connect(ssid, password);

    WHILE NOT WiFi.IsConnected() DO
        SysLog.WriteLogEntry('.');
        delay1(IN := TRUE, PT := T#500ms);
        IF delay1.Q THEN
            delay1(IN := FALSE);
        END_IF
    END_WHILE

    isConnected := TRUE;
    SysLog.WriteLogEntry('WiFi conectado.');
```

```
END_METHOD

// Reconexão com MQTT
METHOD reconnect : VOID
VAR
    clientId : STRING(30);
END_VAR
BEGIN
```

```
WHILE NOT MQTT.Client.IsConnected() DO
  clientId := CONCAT('ESP32Client-', TO_STRING(RANDOM(0..65535)));
  IF MQTT.Client.Connect(clientId) THEN
    MQTT.Client.Subscribe('refrigerador/comando');
  ELSE
    SysLog.WriteLogEntry('Erro de conexão MQTT. Tentando novamente em 5s.');
```

```
    delay2(IN := TRUE, PT := T#5s);
    IF delay2.Q THEN
      delay2(IN := FALSE);
    END_IF
  END_IF
END_WHILE
END_METHOD

// Ciclo Principal
METHOD loop : VOID
BEGIN
  // Verificar Conexão
  IF NOT MQTT.Client.IsConnected() THEN
    reconnect();
  END_IF

  // Leitura dos Sensores
  umid := DHT11.ReadHumidity();
  temp := DHT11.ReadTemperature();
  qld_ar := MQ135.ReadCO2();
  pressao := BMP280.ReadPressure() / 100;

  // Formatar Valores como String
  umidString := REAL_TO_STRING(umid);
  tempString := REAL_TO_STRING(temp);
  qldArString := INT_TO_STRING(qld_ar);
  pressaoString := INT_TO_STRING(pressao);
```

```
// Exibição e Publicação de Dados
IF qld_ar > Qualidade_ar_severa THEN
    SysLog.WriteLogEntry('Qualidade do Ar: Severa');
    MQTT.Client.Publish('refrigerador/status_ar', 'Severa');
ELSIF qld_ar > Qualidade_ar_bem_ruim THEN
    SysLog.WriteLogEntry('Qualidade do Ar: Bem Ruim');
    MQTT.Client.Publish('refrigerador/status_ar', 'Bem Ruim');
ELSIF qld_ar > Qualidade_ar_ruim THEN
    SysLog.WriteLogEntry('Qualidade do Ar: Ruim');
    MQTT.Client.Publish('refrigerador/status_ar', 'Ruim');
ELSIF qld_ar > Qualidade_ar_moderada THEN
    SysLog.WriteLogEntry('Qualidade do Ar: Moderada');
    MQTT.Client.Publish('refrigerador/status_ar', 'Moderada');
ELSE
    SysLog.WriteLogEntry('Qualidade do Ar: Boa');
    MQTT.Client.Publish('refrigerador/status_ar', 'Boa');
END_IF

// Publicar Dados MQTT
MQTT.Client.Publish('refrigerador/umidade', umidString);
MQTT.Client.Publish('refrigerador/temperatura', tempString);
MQTT.Client.Publish('refrigerador/qualidadeAr', qldArString);
MQTT.Client.Publish('refrigerador/pressao', pressaoString);

delay3(IN := TRUE, PT := T#10s);
IF delay3.Q THEN
    delay3(IN := FALSE);
END_IF
END_METHOD
```

Adaptações:

- Definição de Variáveis: As variáveis de sensores e valores são declaradas em uma seção VAR.
- Métodos (Funções): As funções como setup, setup_wifi e reconnect são representadas como métodos que executam a lógica inicial de configuração e conexão.
- Controle de Fluxo: O loop de WHILE é usado para simular a lógica de conexão e monitoramento de sensores.
- Sensor de Temperatura e Umidade: A instância do sensor DHT11 é criada e inicializada dentro do método setup.
- MQ135 e BMP280: Ambos os sensores são inicializados e verificados no método setup.

2.8.2. Function Block Diagram

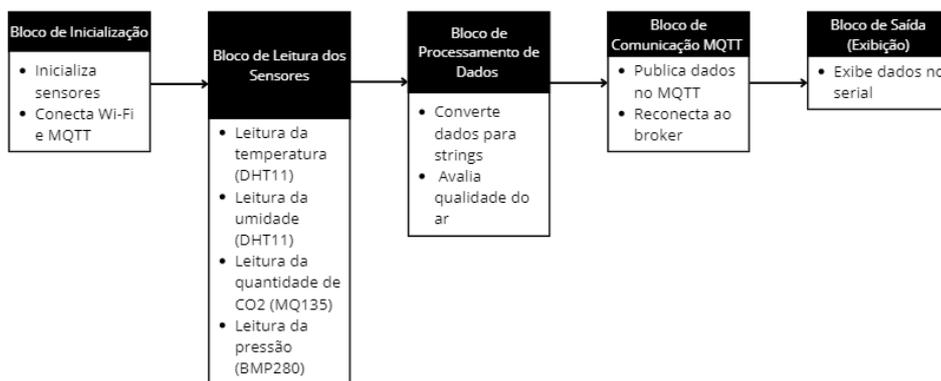


Imagem 05: Function Block Diagram

Fonte: (Autoria Própria, 2024)

2.8.3. Ladder Diagram

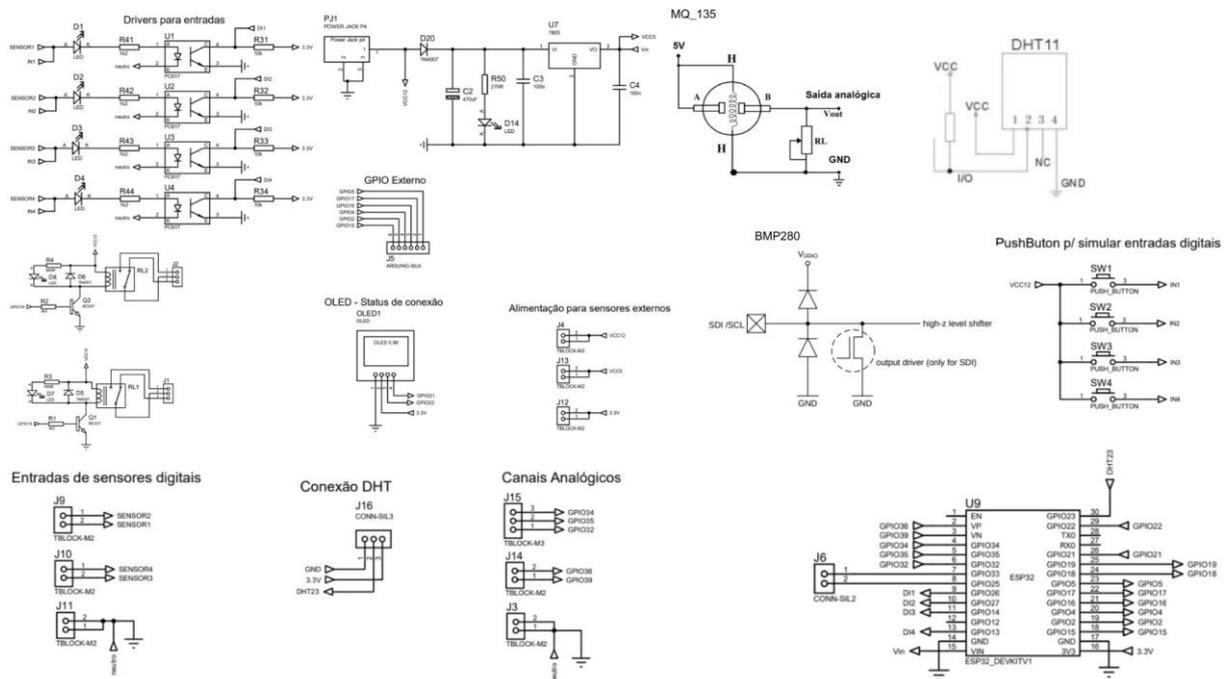


Imagem 06: Ladder Diagram

Fonte: (Autoria Própria, 2024)

2.9. NBR ISO 9001

2.9.1. Relatórios

- 04/11 – Criação do código inicial.
- 08/11 – Teste dos sensores.
- 10/11 – Implementação do protocolo de comunicação.
- 14/11 – Testes dos sensores e protocolo de comunicação em conjunto.
- 18/11 – Finalização do código.
- 19/11 – Começo do desenvolvimento da documentação em função do desenvolvimento do protótipo conceito.
- 20/11 – Desenvolvimento do Structured Text.
- 23/11 – Testes e manutenções finais.
- 24/11 – Desenvolvimento do Function Block Diagram.
- 25/11 – Adição da API OpenWaterMap.
- 26/11 – Desenvolvimento do Ladder Diagram.

3. CONSIDERAÇÕES FINAIS

A implementação deste projeto pode converter a realidade de muitas indústrias brasileiras, permitindo que empresas de ínfimo porte acessem tecnologias que anteriormente estavam disponíveis unicamente para grandes corporações. A abordagem adotada, que envolve averiguação detalhada e a edificação de um protótipo, proporciona um exemplar que pode ser replicado e apropriado a diferentes contextos e necessidades específicas.

É de suma importância que, ao correr do processo de implementação, as empresas mantenham um seguimento constante dos resultados e ajustem suas estratégias conformato ao primordial. Além disso, a disseminação da informação sobre automação acessível e a educação sobre o uso de tecnologias emergentes são cruciais para maximizar os benefícios desse projeto.

Por destino, este trabalho não só contribui para o aperfeiçoamento tecnológica no domínio industrial, mas igualmente alinha-se aos valores de acessibilidade e eficiência, promovendo uma atmosfera industrial mais sustentável e competitivo no Brasil. A continuidade da pesquisa e o aperfeiçoamento das soluções propostas são passos essenciais para caucionar que a automação industrial se torne uma realidade para todas as empresas, independentemente de seu porte.

4. REFERÊNCIAS

ABRAHAM SERHANE, Mohamad Raad, Raad Raad e Willy Susilo. PLC Code-Level Vulnerabilities. Disponível em: <https://ieeexplore.ieee.org/document/8460287>. Acesso em: 18 de maio, 2024.

AGNELLO DENIS VIEIRA, Eduardo Alves Portela Santos, Max Hering de Queiroz, André B. Leal, Amaro Duarte de Paula Neto, José E. R. Cury. A Method for PLC Implementation of Supervisory Control of Discrete Event Systems. Disponível em: <https://ieeexplore.ieee.org/document/7452406>. Acesso em: 18 de maio, 2024.

ANHANGUERA. Descubra qual é o salário de um engenheiro de software. Disponível

em: <https://blog.anhanguera.com/engenharia-de-software-salario/>. Acesso em: 20 nov. 2024.

AWARI. Salário de Engenheiro de Sistemas de Design. Disponível em: <https://awari.com.br/salario-de-engenheiro-de-sistemas-de-design/>. Acesso em: 20 nov. 2024.

CATHO. Salários na área de Engenharia: uma visão geral do mercado brasileiro. Disponível em: <https://www.catho.com.br/salarios/engenharia>. Acesso em: 20 nov. 2024.

CLUBE DA ELETRÔNICA. PDA IoT 2R. Disponível em: <https://www.clubedaeletronica.com.br/p/pda-iot-2r.html>. Acesso em: 27 nov. 2024.

OLX BRASIL. Salário médio de engenheiros no Brasil: uma análise por estado e especialização. Disponível em: <https://www.olx.com.br/dinheiro/salario-medio-engenheiros-brasil>. Acesso em: 20 nov. 2024.

OPENWEATHERMAP. OpenWeatherMap. Disponível em: <https://openweathermap.org/>. Acesso em: 27 nov. 2024.

QUERO BOLSA. Quanto ganha um Engenheiro de Software? (Salário 2024). Disponível em: <https://querobolsa.com.br/cursos-e-faculdades/engenharia-de-software/quanto-ganha-engenheiro-de-software-salario>. Acesso em: 20 nov. 2024.

QUERO BOLSA. Quanto ganha um Engenheiro de sistemas? Salário de Engenharia. Disponível em: <https://querobolsa.com.br/cursos-e-faculdades/engenharia-de-sistemas/quanto-ganha-engenheiro-de-sistemas-salario>. Acesso em: 20 nov. 2024.

REINHARD LANGMANN e Michael Stiller. The PLC as a Smart Service in Industry 4.0 Production Systems. Disponível em: <https://www.mdpi.com/2076-3417/9/18/3815>. Acesso em: 18 de maio, 2024.

ROBOCORE. Sensor de pressão e temperatura BMP280. Disponível em: https://www.robocore.net/sensor-ambiente/sensor-de-pressao-e-temperatura-bmp280?srsItd=AfmBOoqnEC0QmKitUFQaUTEAxoxBYjV5u2mgL-ZPS8bSoe6D_UXA5zMw. Acesso em: 27 nov. 2024.

ROBOCORE. Sensor de temperatura DHT11. Disponível em: <https://www.robocore.net/sensor-ambiente/sensor-de-temperatura-dht11>. Acesso em: 27 nov. 2024.

SALARIO.COM.BR. Engenheiro de Software - Salário Brasil. Disponível em: <https://www.salario.com.br/profissao/engenheiro-de-software-computacional-basico->

cbo-212215/. Acesso em: 20 nov. 2024.

UNIVERSIDADE DE SÃO PAULO. Aula 2: Controladores Lógicos Programáveis.

Disponível em:

https://edisciplinas.usp.br/pluginfile.php/1881421/mod_resource/content/0/Aula2_1131.pdf. Acesso em: 20 nov. 2024.

UNIVERSIDADE DE SÃO PAULO. NBR ISO 9001: Sistema de Gestão da Qualidade.

Disponível em:

https://edisciplinas.usp.br/pluginfile.php/8364638/mod_resource/content/1/nbriso9001.pdf. Acesso em: 20 nov. 2024.

VAGAS.COM.BR. Pesquisa salarial para engenheiros no Brasil: tendências e dados atualizados. Disponível em: <https://www.vagas.com.br/pesquisa-salarial-engenheiros-brasil>. Acesso em: 20 nov. 2024.

VIDA DE SILÍCIO. Sensor de gás MQ-135. Disponível em:

<https://portal.vidadesilicio.com.br/sensor-de-gas-mq-135/>. Acesso em: 27 nov. 2024.

ZENZORCONTROL. Nível de CO2: como detetar e quais os efeitos. Disponível em:

<https://zenzorcontrol.pt/pt/nivel-de-co2-como-detetar-quais-os-efeitos>. Acesso em: 27 nov. 2024.