

DESENVOLVENDO UM CLP COM EXPANSÃO E PROGRAMADO COM LINGUAGEM OPEN SOURCE

DEVELOPING A PLC WITH EXPANSION AND PROGRAMED WITH OPEN-SOURCE LANGUAGE

DESARROLLO DE UN CLP CON EXPANSIÓN Y PROGRAMADO EN UN LENGUAJE OPEN SOURCE

Ederson Batista Berti¹

Eliane Silva Custódio²

Resumo

Desenvolver um pequeno sistema de automação utilizando componentes eletromecânicos é possível, porque há algumas décadas a automação era desenvolvida dessa forma, no entanto, caso necessite de mudanças no sistema de automação, a dificuldade em fazer tal alteração é imensa, porque envolve alterações no circuito elétrico e necessita de profissionais com conhecimento específico. Com o desenvolvimento do microcomputador PC de uso pessoal, a indústria também ganhou seu microcomputador dedicado ao controle de processos e automação chamado de PLC do inglês ou Controlador Lógico Programável (CLP), ou seja, um computador com propósito específico para controle de sistemas de automação. Com o surgimento do CLP, a lógica da automação passou para código gravado no CLP, tornando mais acessível para alterações, melhoria de processo, diagnóstico de manutenção dos sistemas automatizados. Com o avanço da tecnologia de comunicação, foi possível coletar informações ricas sobre o chão de fábrica, tornando os processos produtivos mais eficientes em custos e qualidade. No entanto, ainda há sistemas antigos eletromecânicos na indústria, que podem ser atualizados com *retrofit* por meio de CLP a custo acessível, e é o que será explorado com essa pesquisa, desenvolvendo um *hardware* e se baseando no que se tem disponível de ferramentas de programação Open Source.

Palavras-chave: CLP; entradas; saídas; CPU; *firmware*; automação; *retrofit*.

Abstract

Developing a small automation system using electromechanical components is possible, because a few decades ago automation was developed in this way. However, if changes to the automation system are necessary, the difficulty in making such a change is immense because it involves changes in the electrical circuit and the need for professionals with specific knowledge. When the PC microcomputer for personal use was developed, the industry also gained its own microcomputer dedicated to process control and automation called a PLC or Programmable Logic Controller PLC, that is, a computer with a specific purpose for controlling automation systems. With the invention of the PLC, the automation logic was transferred to the code programmed in the PLC, making it more accessible for changes, process improvement, and maintenance diagnosis of automated systems and with the advancement of communication technology, it was possible to discover rich information about the factory floor, making production processes more efficient in terms of costs and quality. However, there are still older electromechanical systems in the industry that can be updated with retrofit through PLC at an affordable cost, and this is what will be explored with this research by developing hardware and based on what we currently have available in terms of Open-Source programming.

Keywords: Plc; inputs; output; cpu; firmware; automation; retrofit;

Resumen

Desarrollar un pequeño sistema de automatización utilizando componentes electromecánicos es posible, ya que hace unas décadas que la automatización se desarrollaba de esa forma. Sin embargo, si es necesario realizar cambios en el sistema de automatización, la dificultad para realizar dicho cambio es inmensa, ya que implica

¹ Estudante de Engenharia da Computação pelo Centro Universitário Internacional – UNINTER. E-mail: ederson@mastermoldes.com.br

² Professora no Centro Universitário Internacional – UNINTER. E-mail: eliane.s@uninter.com

cambios en el circuito eléctrico y requiere profesionales con conocimientos específicos. Con el desarrollo del microordenador PC para uso personal, la industria también logró su propio microordenador dedicado al control y la automatización de procesos, denominado PLC o Controlador Lógico Programable (PLC), es decir, un ordenador con una finalidad específica para controlar sistemas de automatización. Con la llegada del PLC, la lógica de automatización se convirtió en código grabado en el PLC, lo que la hizo más accesible para realizar cambios, mejoras en los procesos y diagnósticos de mantenimiento de los sistemas automatizados. Con el avance de la tecnología de la comunicación, ha sido posible recopilar muchas informaciones sobre el taller, haciendo que los procesos de producción sean más eficientes en términos de costes y calidad. Sin embargo, todavía existen en la industria antiguos sistemas electromecánicos que pueden ser reequipados utilizando un PLC a un coste asequible, y esto es lo que se explorará con esta investigación, desarrollando hardware y aprovechando lo disponible en herramientas de programación Open Source.

Palabras clave: PLC; entradas; salidas; CPU; *firmware*; automatización; *retrofit*.

1 Introdução

A automação está cada vez mais presente em nosso cotidiano, tornando tarefas como ligar e desligar aparelhos em horários específicos uma prática comum. Na indústria, o CLP (Controlador Lógico Programável), desempenha um papel fundamental na automação de processos, como nas linhas de montagem de veículos, em que robôs realizam tarefas de soldagem.

Esse avanço teve início na década de 60, quando a General Motors desenvolveu o primeiro CLP para suas linhas de produção (Zancan, 2011). Desde então, o mercado de CLP movimenta bilhões de dólares anualmente, impulsionado pela busca por competitividade, controle de qualidade e redução de custos.

Com o advento da indústria 4.0, a conectividade entre sistemas e máquinas se tornou essencial, permitindo o acesso a informações de produção de qualquer lugar. Muitos equipamentos antigos podem ser atualizados por meio de *retrofit*³, uma técnica de reforma ou melhoria, que traz benefícios financeiros às empresas ao prolongar a vida útil dos equipamentos a um custo mais acessível (Peixoto, 2016).

No entanto, o alto custo de implementação de novas tecnologias e a importação de equipamentos novos representam desafios para a adoção generalizada de CLPs em projetos de automação (Grams; Cetnarowski, 2014). Diante desse cenário, surge a necessidade de desenvolver um CLP de baixo custo, com possibilidade de expansão e programação fácil. Essa iniciativa visa atender a pequenos e médios projetos de automação, utilizando linguagem de programação *Open Source*.

³ Retrofit é um termo do inglês para reforma e no contexto industrial é um procedimento de reformar e modernizar um equipamento.

2 Fundamentação teórica

O controlador lógico programável é um dispositivo de processamento de dados com características de funcionamento semelhantes ao computador pessoal, porém, o equipamento é desenvolvido para aplicações industriais como controle de máquinas, células de produção, coleta de dados de sensores, que não exige *hardware* com grande poder de processamento e enormes quantidades de memória. O equipamento possui memória reduzida e roda um conjunto reduzido de instruções, orientado a sinais externos e dados fornecidos pelo usuário, posteriormente, processa os dados e faz acionamento de atuadores.

Assim, apresenta-se qual a definição de CLP pela IEC e NEMA. Segundo Franchi e Camargo (2010), CLP é entendido pelo *International Electrotechnical Commission (IEC)* como um sistema eletrônico que, operando digitalmente, é projetado para o uso em um ambiente industrial, o qual usa uma memória programável para a armazenagem interna de instruções orientadas para o usuário implementar funções específicas, tais como lógica sequencial, temporização, contagem e aritmética, a fim de controlar, por meio de entradas e saídas digitais ou analógicas, vários tipos de máquinas ou processos. O controlador programável e seus periféricos associados são projetados para serem facilmente integráveis em um sistema de controle industrial, usados em todas suas funções previstas.

Já de acordo com a *National Electrical Manufacturers Association (NEMA)*, trata-se de um equipamento eletrônico que funciona digitalmente, utilizando uma memória programável para o armazenamento interno de instruções que implementa funções específicas, tais como lógica, sequenciamento, registro e controle de tempos, contadores e operações aritméticas para controlar, por meio de módulos de entrada/saída digitais (LIGA/DESLIGA) ou analógicos (1-5VCC, 4-20mA etc.), vários tipos de máquinas ou processos.

3 Metodologia

O processo adotado para o desenvolvimento do Controlador Lógico Programável (CLP) de baixo custo envolve uma metodologia cuidadosa para garantir eficiência, economia e funcionalidade. Uma metodologia básica para o desenvolvimento desse tipo de CLP conta com alguns processos, como a definição de requisitos, a pesquisa de componentes, a seleção de plataforma e o desenvolvimento de *Firmware*.

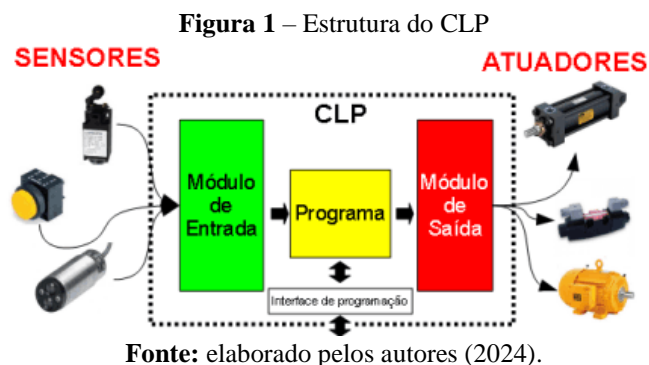
Para Definir os Requisitos, foi identificado claramente os requisitos do CLP, incluindo as entradas e saídas necessárias, capacidade de processamento, interface de usuário, comunicação de rede e qualquer funcionalidade específica exigida pelo sistema que será

controlado. Na pesquisa de componentes, realiza-se uma pesquisa detalhada para identificar os componentes eletrônicos e microcontroladores disponíveis no mercado que atendam aos requisitos do CLP e sejam acessíveis em termos de custo.

Com relação à seleção de plataforma, com base na pesquisa de componentes, foi selecionada a plataforma de *hardware* mais adequada para o CLP. Com uso da ESP32 e desenvolvimento de uma placa (PCB) personalizada. Finalmente, o desenvolvimento de *Firmware* foi realizado com o desenvolvimento do *firmware* do CLP, que inclui o programa de controle e lógica para processar as entradas, tomar decisões com base nelas e acionar as saídas correspondentes. A linguagem utilizada para programação foi o C/C++.

4 Desenvolvimento

A estrutura de *hardware*, necessária para o CLP, são interfaces de coleta de dados, acionamento, módulo de processamento e uma interface de troca de informações com o usuário, conforme mostrado na figura 1.



4.1 Placas CPU, IHM, entradas e saídas digitais

A Unidade de processamento é onde fica armazenada a lógica do programa, portas de comunicação para coleta de informações das placas de entradas, do circuito integrado RTC responsável pela hora, para fazer a troca de informações com a IHM, enviando comandos para as placas de saídas. Essa placa possui entrada de alimentação 12 V de tensão contínua que é utilizada somente para a CPU, sendo que essa fonte utiliza um conversor DC/DC do tipo Buck que rebaixa a tensão para 5 V necessário para o microcontrolador e demais circuitos.

As gravações de programas na CPU podem ser feitas com gravador ISP USBASP de baixo custo facilmente encontrado em fornecedores de componentes eletrônicos. A IHM mostrada na figura 2, traz recursos importantes como *display* de LCD que possibilita

acompanhar e inserir valores utilizados no processo. Também há a disposição um teclado com oito teclas para interação com a lógica do projeto e facilidade de acionamento direto de cada saída digital.

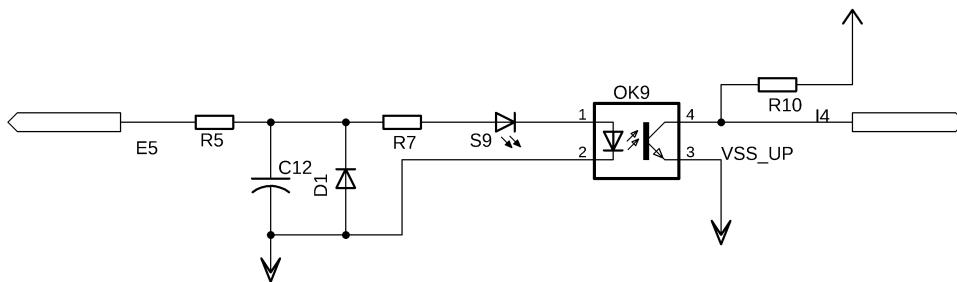
Figura 2 - IHM do CLP



Fonte: elaborado pelos autores (2024).

O módulo de entradas digitais possui oito entradas digitais e foi desenvolvido para trabalhar com sinais de 24 V oriundos de chave fim de curso, botoeiras e sensores do tipo PNP. Os sinais externos são isolados da CPU por opto acopladores, os quais evitam que interferências externas possam interferir no processamento e causar falhas. Na figura 3, há um esquemático que será usado no dimensionamento dos valores dos componentes.

Figura 3 - Entradas Digitais



Fonte: elaborado pelos autores (2024).

Parâmetros admissíveis para os sinais de entradas digitais podem ser identificados conforme dos dados da tabela 1:

Tabela 1 – Parâmetros dos sinais de entrada			
Variável	Símbolo	Valor	Unidade
Tensão mínima de entrada	$V_{inMÍN}$	22	[V]
Tensão máxima de entrada	$V_{inMÁX}$	24	[V]
Corrente definida para o LED	I_{MAXLED}	5	[mA]
Corrente de acionamento do Opto acoplador	I_{maxOPT}	5	[mA]
Frequência de parâmetros para cálculos do filtro	F_c	100	[Hz]

Tensão Led Opto acoplador	V_{opto}	1,2	[V]
---------------------------	------------	-----	-----

Fonte: elaborado pelos autores (2024).

Os resistores R5 e R7 do circuito de entrada da figura 3 limitam a corrente no opto acoplador e do LED indicador de entrada digital ativa, nesse caso, precisam ser dimensionados conforme os parâmetros de tensão mínima e máxima das equações. Calculando, o filtro passa baixo, admitindo uma frequência de corte de 100Hz. Com o valor do resistor R5 dimensionado em 3,3K Ω , utiliza-se a equação abaixo para calcular o capacitor para o filtro passa baixo.

$$C_{12_{filtro}} = \frac{1}{2\pi f R_5} = \frac{1}{2\pi \cdot 100 \cdot 3300} = 482nF$$

Foi adotado o valor comercial de 470nF, então a frequência de corte precisa ser recalculada:

$$F_c = \frac{1}{2\pi R_5 C_{12}} = \frac{1}{2\pi \cdot 3300 \cdot 470 \times 10^{-9}} = \frac{1}{0,00974028} = 102,67Hz$$

O modulo de saídas digitais possui oito saídas digitais com relé contato NA, o que possibilita o acionamento de cargas com tensões diferentes. Esse modulo de saída é isolado por opto acopladores entre barramento e acionamento, evitando que interferências geradas pelas cargas possam chegar ao barramento de comunicação. Para indicação de saídas ativas, há LED sinalizador para cada saída.

O dimensionamento do resistor R20 LED do opto acoplador foi calculado conforme a equação abaixo e o valor adotado o valor comercial de 820 Ω . A Figura 4 mostra o circuito do opto montado.

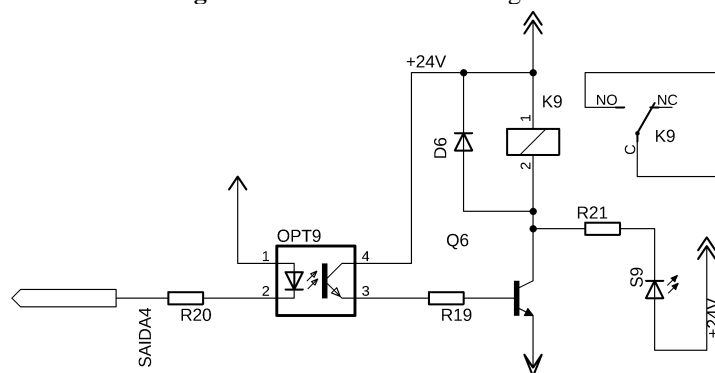
$$R_{led} = \frac{5 - 1,2}{5 \times 10^{-3}} = 760\Omega$$

Dimensionando o acionamento do relé modelo: SRA-24VDC-CL - SONGLE RELAY, esse componente fará acionamento de cargas com os dados que estão referenciados na tabela2.

Tabela 2 – Parâmetros do relé SRA-24VDC-CL

Variável	Símbolo	Valor	Unidade
Tensão da bobina	V_{in}	24	[V]
Corrente da bobina	I_{bob}	33,4	[mA]

Fonte: elaborado pelos autores (2024).

Figura 4 - Circuito de saídas digitais

Fonte: elaborado pelos autores (2024).

Foi calculado o valor comercial de $4,7K\Omega$ para R19. O diodo D6 foi selecionado o modelo 1N4007, diodo retificador de uso geral. A função do diodo D6 é roda livre⁴, para proteção do transistor devido ao surto produzido pela bobina do relé.

4.2 Estrutura da biblioteca

Uma das premissas do projeto foi a possibilidade de utilizar ferramentas *Open Source* para programação, então, optou-se pela IDE Arduino para o desenvolvimento da biblioteca, visto que o microcontrolador utilizado no projeto é o ATMEGA328, mesmo modelo utilizado no Arduino UNO. Isso também dá a possibilidade de carregar o *bootloader* do Arduino UNO no CLP e posterior uso de um conversor USB/Serial para carregar códigos compilados.

A segunda ferramenta utilizada é o Visual Studio Code (VS Code) que é uma suíte de desenvolvimento para uma diversidade grande de linguagens de programação, que foi desenvolvida pela Microsoft e é gratuita. Diferente do Arduino IDE, o VS Code utiliza uma extensão chamada PlatformIO e com a combinação de ambas as ferramentas, tem-se à disposição muitos recursos de programação e depuração de código.

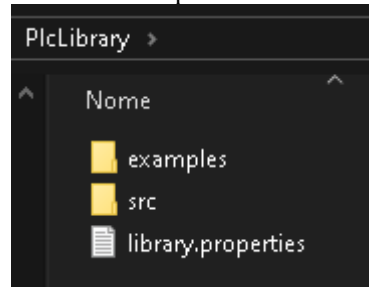
O desenvolvimento da biblioteca em linguagem C++ necessita de pelo menos dois arquivos, com extensão *PlcLibrary.h* e *PlcLibrary.cpp*. O arquivo *PlcLibrary.h* possui uma classe em que são declaradas as funções, macros, variáveis públicas e variáveis privadas. Esse mesmo arquivo será utilizado como cabeçalho da biblioteca, quando estiver em uso em algum projeto que faz uso do CLP.

⁴ Diodo de roda livre

Sugestão: Um diodo de roda livre, também chamado de diodo de recuperação, é um componente eletrônico empregado em circuitos de potência para salvaguardar dispositivos semicondutores, como transistores de potência e tiristores, contra sobretensões reversas. Sua função é permitir que a corrente induzida pelo campo magnético de um indutor (ou bobina) retorne ao circuito de maneira controlada, prevenindo danos aos componentes sensíveis (Hart, 2012).

Para complementar a biblioteca, é necessário implementar o arquivo *PlcLibrary.cpp*, o qual faz o “trabalho” de acionamento do *hardware*. No entanto, para seguir o padrão das bibliotecas utilizadas pelo projeto Arduino, conforme mostra a Figura 5, necessita-se de uma pasta principal com o nome da biblioteca a ser criada. Internamente a essa pasta, necessita-se de uma pasta *examples*, na qual estarão alguns exemplos de uso, uma pasta *src* que terá os códigos da biblioteca e um arquivo, *Keywords properties*.

Figura 5 - Formato pasta biblioteca Arduino



Fonte: elaborado pelos autores (2024).

5 Resultado e discussão

5.1 Desafios encontrados no desenvolvimento do *hardware*

O equipamento que está em desenvolvimento é para uso de ambiente industrial, então, há uma grande preocupação com sinais de interferências e temperatura gerados pelo chão de fábrica. O equipamento precisa de mais procedimentos de testes para confirmar a eficácia no funcionamento, pois podem ainda haver necessidades de melhorias no *hardware*.

O protótipo atual que está em uso para o desenvolvimento da biblioteca de programação e pode ser visualizado na Figura 6. Esse protótipo da imagem está com um módulo com oito de entradas digitais e dois módulos com oito saídas cada, totalizando 16 saídas digitais. A outra placa disponível no CLP é placa CPU, mais à esquerda na imagem, que corresponde ao módulo que é responsável pelo processamento dos dados coletados pelos módulos de E/S, que também faz comunicação com a IHM mostrada na figura 2. Outro detalhe importante é que a CPU possui sua própria fonte de alimentação, não compartilhando com os demais módulos, o que evita que sinais externos possam interferir no processamento dos dados.

Figura 6 – CPU e módulos de E/S



Fonte: elaborado pelos autores (2024).

Os circuitos que foram desenvolvidos durante esse trabalho de pesquisa estão relacionados nas imagens das Figura 7 até a Figura 9, que são as placas individuais. Já a imagem da Fonte: elaborado pelos autores (2024).

Figura 10 mostra como as placas estão encaixadas em um barramento que é utilizado para comunicação entre os módulos e a CPU.

Pode-se ver na Fonte: elaborado pelos autores (2024).

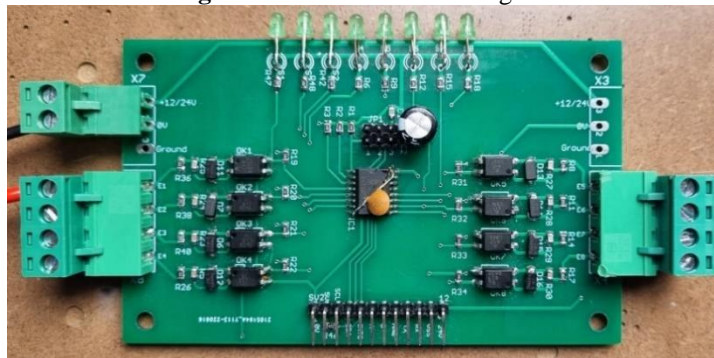
Figura 11 o protótipo utilizado para os testes do código da biblioteca durante o seu desenvolvimento. Isso se torna prático, pois durante o desenvolvimento, basta compilar o código, carregar no hardware e fazer os primeiros testes.

Figura 7 – Placa CPU



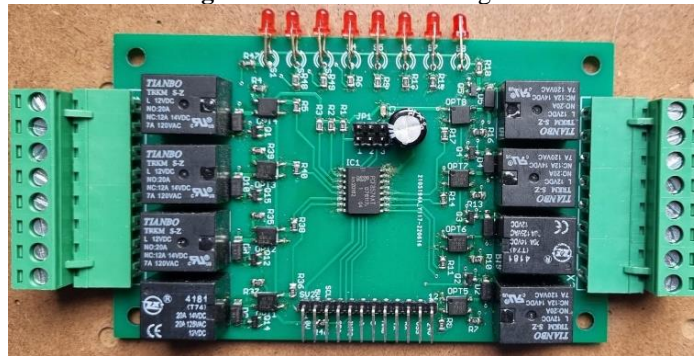
Fonte: elaborado pelos autores (2024).

Figura 8 – Placa 8 entradas digitais



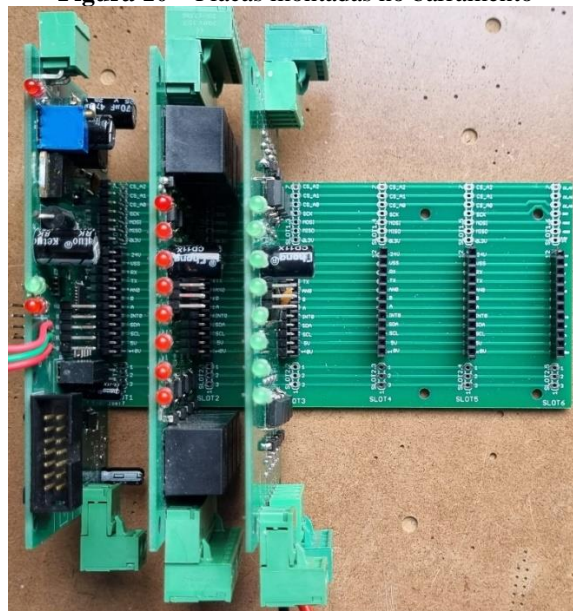
Fonte: elaborado pelos autores (2024).

Figura 9 – Placa 8 saídas digitais



Fonte: elaborado pelos autores (2024).

Figura 10 – Placas montadas no barramento



Fonte: elaborado pelos autores (2024).

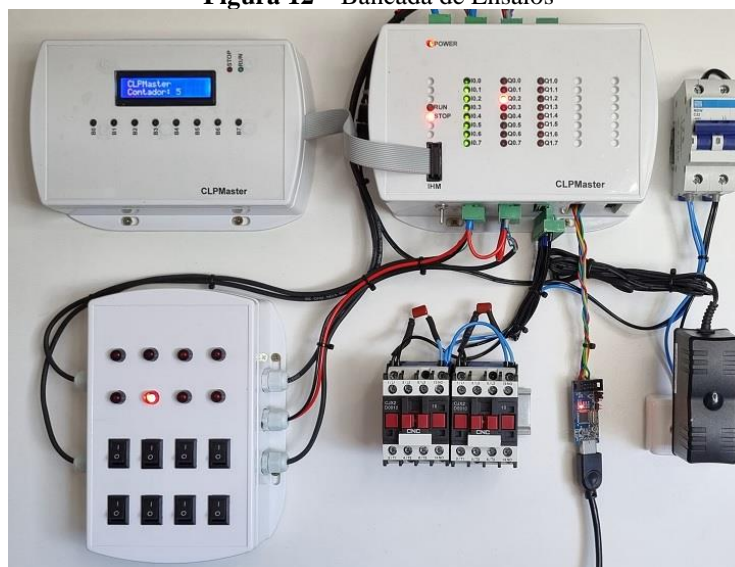
Figura 11 – Protótipo inicial do CLP



Fonte: elaborado pelos autores (2024).

Durante os ensaios de desenvolvimento, foram montadas as placas de circuito impresso, as quais foram acomodadas em um gabinete. No entanto, para visualização do produto em andamento, necessita-se acionar dispositivos quando são testadas as saídas digitais e necessita-se fazer leitura para fazer teste das entradas digitais, além do mais, é necessário interagir também com o CLP. Para isso foi montado a IHM com teclado. Todos esses módulos, incluindo a fonte de alimentação, foram montados em um painel conforme a Figura 12, o que tornou mais fácil, mais seguro e o visual de um produto.

Figura 12 – Bancada de Ensaio



Fonte: elaborado pelos autores (2024).

5.2 Desafios encontrados no desenvolvimento do software – LIB

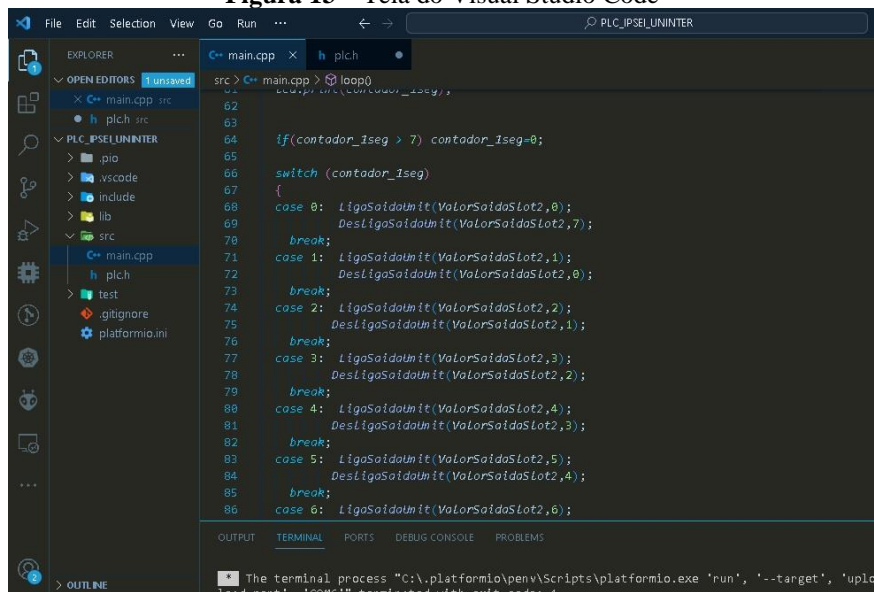
Quando se desenvolve uma aplicação com sistemas embarcados, o projeto é desenvolvido por completo, deixando o usuário final apenas com parâmetros de uso. No caso do CLP é um pouco diferente, porque é desenvolvido o *hardware* e a biblioteca de uso e o usuário, que nesse caso será alguém que tenha conhecimento, irá desenvolver a lógica do seu projeto se utilizando da biblioteca.

O código da biblioteca durante o projeto comporta leitura de entradas digitais, acionamento de saídas digitais, leitura das teclas e envio de mensagem para a IHM, RTC com geração de hora e calendário e ainda geração de tempos 100ms e 1 segundo que podem ser utilizados para monitoramento de tempos de processos em automação.

O desenvolvimento da biblioteca traz alguns desafios, pois o código necessita estar dentro dos padrões, otimizado e sem falhas. Além do mais, são necessários ajustes para deixar o código da biblioteca com o mínimo de erros possíveis e armazenar para *download* em sistemas como Github⁵, para que usuários possam fazer uso dessa juntamente com o CLP.

A biblioteca está sendo desenvolvida para uso na IDE arduino, mas o Visual Studio Code mostrado na Figura 13 é uma suíte de desenvolvimento, desenvolvida pela Microsoft, que possui um recurso chamado PlataformaIO. Nela é possível utilizar a estrutura de programação semelhante a IDE arduino com a diferença da ferramenta de depuração que ajuda muito no desenvolvimento de código.

Figura 13 – Tela do Visual Studio Code



⁵ O GitHub é um serviço baseado em nuvem que hospeda um sistema de controle de versão (VCS) chamado Git.

Fonte: elaborado pelos autores (2024).

5.3 Relação de custo

Os custos aproximados de componentes e placas para a montagem de uma unidade está descrito na tabela 3. Nessa tabela não estão sendo levados em conta outros custos como caixa para colocar o equipamento, manuais, cabos, adesivos, acessórios que seriam necessários para um produto finalizado.

Tabela 3 - Levantamento de Custos

Descrição dos itens	Quantidades	Custo aproximado (R\$)
Placa de entradas digitais	1	62,51
Placa de saídas digitais:	1	110,75
Placa CPU	1	124,19
IHM	1	96,89
Barramento	1	31,68
Gabinete CLP+IHM	1	50,0
Itens para montagem	1	10,0
TOTAL		486,02

Fonte: elaborado pelos autores (2024).

6 Considerações finais

O desenvolvimento de qualquer aplicação de tecnologia focada na indústria se torna crítica devido ao ambiente de intempéries que podem prejudicar o funcionamento de um equipamento eletrônico, sejam elas temperatura ou causada por meios eletrônicos.

O CLP passou por alguns testes de funcionamento em bancada e há um protótipo em funcionamento instalado desde dezembro de 2022, mas para tornar-se um produto aplicável na indústria, serão necessários mais testes em ambiente fabril, a fim de observar possíveis melhorias de *hardware*, tratando interferências e compatibilidade eletromagnética. No entanto, para abranger uma gama maior de aplicações, é necessário desenvolvimento de *hardware* de leitura de sensores de temperatura, sensores lineares, sistemas de comunicação, entre outros.

Na parte de *software*, a biblioteca está dando seus primeiros passos e são necessários mais testes, mais ajustes dos códigos para deixar o *software* eficiente. No projeto atual, o *hardware* tem disponível porta RS485 para comunicação Modbus Serial, porta Serial RS232, as quais podem ser exploradas para meios de comunicação, podendo ser testados e implementado na biblioteca de *software*. Materiais de apoio como instruções de uso da biblioteca ainda precisam ser criados, além treinamento, que posteriormente servirão para os usuários do CLP em aplicações de automação.

Referências

ARDUINO. O que é um Arduino? - **HomeArduino**. 2023. Disponível em: <https://www.arduino.cc/>. Acesso em: 02 abr. 2024.

FRANCHI, C. M.; CAMARGO, V. L. A. **Controladores Lógicos Programáveis: Sistemas discretos**. São Paulo: Érica, 2010.

GITHUB. Plataforma de hospedagem de códigos-fonte e arquivos. **GitHub**. Disponível em: <https://github.com/>. Acesso em: 02 abr. 2024.

GRAMS, C. A.; CETNAROWSKI, E. **Retrofit em máquinas industriais: estudo de caso**. 2014. 61 f. Trabalho de Conclusão de Curso (Graduação em Mecatrônica) — Universidade Tecnológica Federal do Paraná, Curitiba, 2014. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/9475>. Acesso em: 11 dez. 2024.

HART, D. W. **Eletrônica de potência: análise e projetos de circuitos**. Porto Alegre: AMGH, 2012.

PEIXOTO, W. C. **Implementação e análise de um retrofitting aplicado em uma máquina de 3 eixos**. 2016. 122 f. Dissertação (Mestrado em Engenharia Mecânica) — Universidade Federal da Paraíba, João Pessoa, 2016. Disponível em: https://repositorio.ufpb.br/jspui/handle/tede/8956?locale=pt_BR. Acesso em: 11 dez. 2024.

VISUAL Studio Code. **Suíte de Desenvolvimento de código**. 2024. Disponível em: <https://visualstudio.microsoft.com/pt-br/>. Acesso em: 23 fev. 2024.

ZANCAN, M. D. **Controladores programáveis**. 3. ed. Santa Maria: Universidade Federal de Santa Maria, 2011.