

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
MINAS GERAIS - *CAMPUS* IBIRITÉ
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Lucas Gonçalves Moreira

**PROJETO DE CONTROLADOR LÓGICO PROGRAMÁVEL
DIDÁTICO ABERTO**

Ibirité - MG

2023

LUCAS GONÇALVES MOREIRA

**PROJETO DE CONTROLADOR LÓGICO PROGRAMÁVEL
DIDÁTICO ABERTO**

Trabalho de conclusão de curso apresentado ao Curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - *Campus* Ibirité para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Carlos Dias da Silva Junior

Coorientador: Prof. Pedro Henrique Ferreira Machado

Ibirité - MG
2023

M838p

Moreira, Lucas Gonçalves.

Projeto de controlador lógico programável didático aberto. [recurso eletrônico] / Lucas Gonçalves Moreira. – Ibité, MG, 2023.

54 p. : il. color.

Orientador: Prof. Carlos Dias da Silva Junior.

Coorientador: Prof. Pedro Henrique Ferreira Machado

Trabalho de Conclusão de Curso (Graduação) – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, *Campus* Ibité, Bacharelado em Engenharia de Controle e Automação, 2023.

1. Controladores programáveis 2. Microcontroladores. 3. Automação industrial. I. Silva Junior, Carlos Dias da. II. Machado, Pedro Henrique Ferreira. III. Instituto Federal de Minas Gerais. *Campus* Ibité. IV. Título.

CDD 629.8

Lucas Gonçalves Moreira

**PROJETO DE CONTROLADOR LÓGICO PROGRAMÁVEL
DIDÁTICO ABERTO**

Trabalho de conclusão de curso apresentado ao Curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - *Campus* Ibirité para a obtenção do título de Engenheiro de Controle e Automação.

Aprovado em: 15/ 12/ 2023 pela banca examinadora:

Prof. Carlos Dias da Silva Junior - IFMG (Orientador)

Prof. Pedro Henrique Ferreira Machado - IFMG (Coorientador)

Prof. Thiago Henrique Barbosa de Carvalho Tavares - IFMG

Prof. Diego Henrique de Souza Chaves - IFMG

Dedico este trabalho aos meus queridos pais e avós, cujo amor incondicional e apoio constante foram a luz que guiou meu caminho ao longo desta jornada acadêmica. Suas palavras de encorajamento, sabedoria e carinho moldaram meu caráter e me inspiraram a buscar conhecimento.

AGRADECIMENTOS

Gostaria de expressar minha sincera gratidão ao Professor Carlos Dias, meu orientador, e ao Professor Pedro Machado, meu coorientador, pela paciência, orientação e profissionalismo ao longo deste trabalho. Seu suporte e conhecimento foram fundamentais para o desenvolvimento deste projeto.

Quero estender meus agradecimentos à minha melhor amiga Julia Freitas, cujo apoio, conselhos e incentivo foram inestimáveis durante os momentos desafiadores deste percurso acadêmico.

Não posso deixar de mencionar minha família, principalmente meus pais, que sempre me incentivaram a buscar conhecimento e a seguir meus sonhos. Sua dedicação e apoio incondicional foram a força motriz por trás de cada conquista neste trabalho.

“O progresso é impossível sem mudança, e aqueles que não podem mudar suas mentes não podem mudar nada.”

George Bernard Shaw

RESUMO

Este trabalho de pesquisa aplicada se concentra no desenvolvimento de um Controlador Lógico Programável (CLP) utilizando o microcontrolador ESP32. O CLP é um dispositivo amplamente empregado na automação industrial para controle e supervisão de processos. A adoção de microcontroladores, como o ESP32, oferece vantagens notáveis, tais como custos reduzidos, maior flexibilidade e integração com tecnologias de comunicação sem fio. Existem diversos estudos e pesquisas relacionados ao desenvolvimento de CLPs baseados em microcontroladores, como o ESP32, em várias áreas da automação industrial. Além disso, existe uma base sólida de pesquisas que examinam as vantagens e desafios do emprego de microcontroladores nesse contexto. Este trabalho busca ampliar e aprofundar o conhecimento existente, proporcionando uma análise abrangente do desenvolvimento de um CLP baseado no microcontrolador ESP32. Esta pesquisa contribuirá para o progresso do conhecimento na área, bem como permitirá uma atualização de equipamentos do Laboratório de Automação do IFMG campus Ibirité, impactando positivamente nas atividades didáticas do curso de Engenharia de Controle e Automação, e permitindo novas atividades de pesquisa. Em complemento, com a implementação deste projeto, espera-se que o Laboratório de Automação permita uma maior interação entre os dispositivos simulação de controle de processos industriais, alinhando-se às estratégias industriais mais modernas, como Internet das Coisas e controle descentralizado

Palavras-chave: Controlador Lógico Programável (CLP). Microcontrolador ESP32. Automação Industrial. Engenharia. Controle.

ABSTRACT

This applied research work focuses on the development of a Programmable Logic Controller (PLC) using the ESP32 microcontroller. The PLC is a widely employed device in industrial automation for process control and supervision. The adoption of microcontrollers, such as the ESP32, offers notable advantages such as reduced costs, increased flexibility, and integration with wireless communication technologies. There are several studies and research efforts related to the development of microcontroller-based PLCs, like the ESP32, across various areas of industrial automation. Moreover, there exists a solid base of research examining the advantages and challenges of employing microcontrollers in this context. This work aims to expand and deepen existing knowledge by providing a comprehensive analysis of developing a PLC based on the ESP32 microcontroller. This research will contribute to advancing knowledge in the field and facilitate an upgrade of equipment in the Automation Laboratory at IFMG Ibirité campus, positively impacting the teaching activities of the Control and Automation Engineering course and enabling new research endeavors. Additionally, through the implementation of this project, it is expected that the Automation Laboratory will allow greater interaction among devices simulating industrial process control, aligning with more modern industrial strategies such as the Internet of Things and decentralized control.

Keywords: Programmable Logic Controller (PLC). ESP32 Microcontroller. Industrial Automation. Engineering. Control.

LISTA DE ILUSTRAÇÕES

Figura 1 – PLC Modicon-084, o primeiro PLC comercial.	17
Figura 2 – Microcontrolador ESP32.	19
Figura 3 – Ciclo SCAN.	20
Figura 4 – Estrutura MODBUS.	21
Figura 5 – Painel de configuração de um novo projeto.	24
Figura 6 – Painel de configuração e compilação do programa.	24
Figura 7 – Arduino Mega.	25
Figura 8 – STM32F103C8T6.	25
Figura 9 – ESP32-DEVKIT-WROOM-32U.	26
Figura 10 – Programa do primeiro teste do OpenPLC.	27
Figura 11 – Circuito de teste.	28
Figura 12 – Saída digital do CLP.	29
Figura 13 – Saída digital responsável por selecionar a tensão.	30
Figura 14 – Entrada digital do CLP.	31
Figura 15 – Associação de resistores da saída analógica do CLP.	32
Figura 16 – Amplificador operacional da saída analógica do CLP.	32
Figura 17 – Entrada analógica do CLP.	33
Figura 18 – Regulador de tensão de 5V.	34
Figura 19 – Regulador de tensão de 3,3V.	34
Figura 20 – Optoacoplador 4n25.	35
Figura 21 – Diodo zener.	35
Figura 22 – Fusível de vidro de 2A.	36
Figura 23 – Configuração do Modbus RTU.	37
Figura 24 – Configuração do Modbus TCP no SCADA BR.	38
Figura 25 – Configuração da entrada e saída no SCADA BR.	38
Figura 26 – Supervisório para teste de comunicação.	39
Figura 27 – Esquemático de ligações do ESP32.	40
Figura 28 – Layout da saída digital.	41
Figura 29 – Programa ladder do ensaio.	43
Figura 30 – Sistema supervisório do ensaio.	43
Figura 31 – Fluxograma do ensaio	44
Figura 32 – Esquemático do ESP32.	49
Figura 33 – Esquemático das entradas digitais.	50
Figura 34 – Esquemático das saídas digitais.	50
Figura 35 – Esquemático das entradas analógicas.	51
Figura 36 – Esquemático da saídas analógicas.	51
Figura 37 – Esquemático dos reguladores de tensão.	52
Figura 38 – Layout da PCB.	53

Figura 39 – Layout da PCB em 3D.	53
Figura 40 – Protótipo.	53
Figura 41 – Protótipo.	54

LISTA DE TABELAS

Tabela 1 – Comparação entre ESP32, Arduino Mega e STM32F103C8T6	27
Tabela 2 – Relação dos terminais do ESP32 e endereços do OpenPLC.	28
Tabela 3 – Mapeamento de terminais e funções nos esquemáticos PLC e ESP.	42

LISTA DE ABREVIATURAS E SIGLAS

TCC	Trabalho de Conclusão de Curso
CLP	Controlador Lógico Programável
PLC	Programmable Logic Controller
CPU	Unidade Central de Processamento
RTU	Remote Terminal Unit
ASCII	American Standard Code for Information Interchange
TCP	Transmission Control Protocol
IP	Internet Protocol
IoT	Internet of Things
SoC	System on a Chip
BLE	Bluetooth Low Energy
GPIO	General Purpose Input/Output
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
UART	Universal Asynchronous Receiver/Transmitter
Wi-Fi	Wireless Fidelity
SSID	Service Set Identifier
THT	Through-Hole Technology
PCB	Print Circuit Board

LISTA DE SÍMBOLOS

Ω Letra grega maiúscula omega

SUMÁRIO

1	INTRODUÇÃO	14
2	REVISÃO BIBLIOGRÁFICA	16
2.1	O que é PLC	16
2.2	Princípios de construção e funcionamento do PLC	18
2.2.1	<i>Fonte de alimentação</i>	18
2.2.2	<i>Módulos de entrada e saída</i>	19
2.2.3	<i>Unidade Central de Processamento (CPU)</i>	19
2.2.4	<i>Ciclo de SCAN</i>	20
2.3	Redes e PLC	21
2.4	Desenvolvimento aberto de PLC	22
2.5	Circuitos de proteção	22
3	METODOLOGIA	24
3.1	Preparação do ambiente	24
3.2	Escolha do Microcontrolador	25
3.3	Teste de acionamento básico	27
3.4	Esquemático das entradas e saídas	28
3.4.1	<i>Saídas digitais</i>	29
3.4.2	<i>Entradas digitais</i>	30
3.4.3	<i>Saídas analógicas</i>	31
3.4.4	<i>Entradas analógicas</i>	32
3.5	Fonte de alimentação	33
3.6	Circuitos de proteção	34
3.7	Teste de comunicação	36
3.8	Desenvolvimento do layout da PCB	39
4	RESULTADOS	43
5	CONCLUSÃO	45
	REFERÊNCIAS	46
	APÊNDICE A – ESQUEMÁTICOS DO PROTÓTIPO	49

APÊNDICE B – LAYOUT DA PCB 53

1 INTRODUÇÃO

A revolução tecnológica e a crescente automação têm sido pilares fundamentais na evolução de diversas indústrias, transformando substancialmente os processos de produção e operação. Em meio a esse contexto dinâmico, a engenharia de controle e automação desempenha um papel crucial, fornecendo as ferramentas e conhecimentos necessários para conceber, implementar e otimizar sistemas automatizados e processos industriais. No entanto, diante da constante evolução tecnológica, é imperativo que os profissionais e futuros engenheiros dessa área estejam adequadamente preparados para enfrentar os desafios da crescente automação industrial.

A formação acadêmica assume um papel crucial na capacitação dos estudantes de engenharia de controle e automação. Entretanto, os métodos de ensino convencionais frequentemente não conseguem oferecer a experiência prática essencial para compreender plenamente os sistemas e processos complexos que esses futuros profissionais encontrarão. A limitação de experimentos em ambientes reais de produção industrial, devido a questões de segurança, altos custos e restrições no currículo acadêmico, tem impulsionado a necessidade de abordagens alternativas para promover um aprendizado mais efetivo e prático.

Os laboratórios didáticos exercem um papel central na formação de engenheiros de controle e automação, possibilitando que os alunos interajam com tecnologias reais, desenvolvam habilidades práticas e consolidem conceitos teóricos. Contudo, os desafios relacionados aos custos, manutenção e disponibilidade de equipamentos em laboratórios tradicionais podem limitar sua acessibilidade e eficácia. Diante dessas restrições, a criação de laboratórios físicos bem equipados é dificultada pelas restrições orçamentárias das instituições de ensino.

Uma alternativa promissora para enfrentar esses desafios é a implementação de laboratórios virtuais e experimentações em ambientes simulados. Apesar de oferecerem flexibilidade e custos reduzidos em comparação com laboratórios físicos, a exposição excessiva a simulações pode desvincular os alunos da realidade física. Além disso, as licenças de software para laboratórios virtuais frequentemente acarretam custos significativos, limitando sua acessibilidade.

Nesse contexto, emerge a necessidade de desenvolver abordagens híbridas que unam elementos do mundo real com simulações virtuais. Os laboratórios híbridos buscam superar as limitações dos laboratórios tradicionais e virtuais, proporcionando uma experiência mais próxima da realidade industrial a um custo reduzido. Esses ambientes oferecem oportunidades para experimentação prática com segurança, flexibilidade e controle sobre os processos. A implementação de soluções de laboratórios híbridos acessíveis e inovadoras é essencial para garantir que os alunos estejam bem preparados para atender às demandas em constante evolução da indústria de automação.

Dentro deste contexto, este trabalho de conclusão de curso (TCC) propõe a criação e implementação de uma arquitetura de CLP baseado em microcontrolador, de forma a construir um laboratório híbrido de baixo custo para o ensino de controle e automação. Mais especificamente,

o objetivo deste projeto é proporcionar uma plataforma prática que permita aos estudantes da área de automação industrial realizar experimentos práticos, aprofundando seu entendimento dos sistemas automatizados. Este laboratório híbrido será baseado em tecnologias de código aberto, como a plataforma ESP32 e o OpenPLC, tornando-o acessível e customizável. Acreditamos que essa abordagem inovadora fornecerá uma base sólida para o aprendizado e a preparação dos futuros profissionais de automação, capacitando-os para enfrentar os desafios em constante evolução da indústria.

Nos próximos capítulos deste TCC, será detalhado a configuração e a implementação do laboratório híbrido, destacando os benefícios do uso de tecnologias de código aberto e as estratégias de ensino relacionadas. Além disso, abordaremos as considerações de segurança e os resultados esperados do projeto. Este trabalho contribuirá para promover uma educação mais prática e acessível na área de controle e automação, proporcionando um melhor preparo aos estudantes para uma carreira de sucesso na indústria automatizada.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, será apresentada a revisão da literatura relacionada ao desenvolvimento de um Controlador Lógico Programável baseado no microcontrolador ESP32. A revisão será organizada em quatro seções principais.

2.1 O que é PLC

Os CLP's - Controladores Lógicos Programáveis têm suas origens nos sistemas de controle baseados em relés amplamente utilizados na automação industrial desde meados do século XX (MORAD, 2019). Os sistemas baseados em relés, embora confiáveis, eram complexos, ocupavam muito espaço e exigiam uma grande quantidade de dispositivos eletromecânicos interconectados para controlar os processos industriais (CARRASCO *et al.*, 2018).

A necessidade de uma alternativa mais eficiente e programável levou ao desenvolvimento dos primeiros PLCs. Acredita-se que o conceito inicial dos CLPs tenha surgido na década de 1960, quando a indústria começou a buscar soluções mais flexíveis para o controle de processos industriais (LEE, 2017).

Um marco crucial foi a criação do primeiro CLP comercial pela Modicon (Modicon Manufacturing Company) em 1969, liderada por Dick Morley (MORAES, 2020). Esse CLP, conhecido como "084", foi desenvolvido para substituir os sistemas baseados em relés e foi revolucionário por ser o primeiro dispositivo a integrar a lógica de controle em um único dispositivo programável (MORAD, 2019). O CLP Modicon-084 está ilustrado na Figura 1.

Figura 1 – PLC Modicon-084, o primeiro PLC comercial.



Fonte: (ELECTRIC, 2018)

O CLP 084 da Modicon foi projetado para oferecer um método de controle mais versátil e programável, utilizando uma linguagem de programação chamada "Ladder Logic". Essa linguagem, baseada na representação gráfica dos circuitos elétricos usando símbolos, tornou-se a base para a programação de CLPs e ainda é amplamente utilizada nos dias de hoje (LEE, 2017).

A evolução dos CLPs desde o 084 até os sistemas mais modernos tem sido impressionante. Ao longo dos anos, os CLPs se tornaram mais poderosos, com capacidades de processamento mais avançadas, integração de comunicação em rede e suporte para uma variedade de linguagens de programação (MORAES, 2020).

A facilidade de programação e reprogramação dos CLPs é fundamental na indústria, especialmente quando essa modificação pode ser realizada diretamente na planta. Essa característica simplifica ajustes operacionais sem a necessidade de interromper a produção. A utilização de linguagens de alto nível e amigáveis ao operador facilita a compreensão e a manutenção dos programas, permitindo uma operação mais eficiente e ágil (MORAES; CASTRUCCI, 2007).

Além disso, a modularidade dos CLPs, resumindo a funcionalidade em conjuntos de entradas e saídas, torna a manutenção e reparo mais diretos, permitindo a substituição de módulos defeituosos sem afetar a operação geral do sistema (FRANCHI; CAMARGO, 2008).

A capacidade de expansão da memória programável é uma vantagem fundamental dos CLPs modernos. Essa característica possibilita a adaptação do dispositivo a novas demandas e processos, permitindo a introdução de novas funcionalidades sem a necessidade de substituição completa do hardware (MORAES; CASTRUCCI, 2007).

A integração dos dados do CLP em bancos de dados gerenciais é um recurso valioso para otimizar o planejamento da produção. Ao disponibilizar informações do chão de fábrica para os departamentos de planejamento, esses dispositivos ajudam na tomada de decisões mais informadas e eficazes (FRANCHI; CAMARGO, 2008). A capacidade de comunicação em rede dos CLPs é essencial para a coleta e troca de dados. Essa funcionalidade possibilita a integração com outros sistemas industriais, contribuindo para a automatização e a eficiência operacional (SOUZA; VIEIRA, 2015). Por fim, a capacidade de informar ao operador sobre falhas por meio de sinalizadores visuais no CLP é crucial para identificar rapidamente a parte do sistema que está defeituosa, permitindo uma intervenção ágil e assertiva para minimizar o impacto na produção (MORAES; CASTRUCCI, 2007).

Essa história de evolução dos CLPs é essencial para entender como esses dispositivos se tornaram a espinha dorsal da automação industrial moderna, proporcionando controle preciso e flexível em uma ampla gama de processos industriais (CARRASCO *et al.*, 2018).

2.2 Princípios de construção e funcionamento do PLC

O CLP é um dispositivo crucial na automação industrial, composto por componentes fundamentais que possibilitam seu funcionamento. Um CLP é composto por uma fonte de alimentação, módulos de entrada e saída e uma unidade central de processamento (CPU) (BOLTON, 2016). Cada um desses elementos desempenha um papel vital na automação industrial.

O funcionamento do CLP é definido por uma rotina interna denominada ciclo de SCAN, a qual é responsável pela execução das funções de controle.

2.2.1 Fonte de alimentação

A fonte de alimentação é um componente essencial que converte a energia elétrica de uma fonte externa em tensões apropriadas para o funcionamento do CLP. Conforme pesquisas recentes, a estabilidade e confiabilidade da fonte de alimentação são cruciais para garantir o funcionamento contínuo do sistema, evitando interrupções nos processos industriais (TERREL, 2019).

2.2.2 Módulos de entrada e saída

Os módulos de entrada e saída atuam como interfaces entre o CLP e o ambiente externo, convertendo os sinais físicos em informações digitais processáveis e vice-versa. Os módulos de entrada recebem sinais de sensores e dispositivos externos, enquanto os módulos de saída enviam sinais de controle para dispositivos periféricos, garantindo a interação entre o sistema CLP e os processos industriais (MANSFIELD, 2018).

2.2.3 Unidade Central de Processamento (CPU)

A CPU é o “cérebro” do CLP, responsável pelo processamento de informações e execução de programas. A CPU contém o processador, memória e interfaces de comunicação, realizando operações lógicas de acordo com a lógica de controle programada, garantindo assim a eficiência na tomada de decisões e controle dos processos industriais (JOHNSON, 2020).

O ESP32 (Figura 9) é um microcontrolador desenvolvido pela Espressif Systems, uma empresa chinesa conhecida por suas soluções em IoT (Internet das Coisas). Lançado em 2016, o ESP32 é uma evolução do ESP8266 e se destaca por sua versatilidade e capacidades avançadas. Ele é construído em torno de um chip de sistema em pacote (SoC) com um processador Xtensa de 32 bits, oferecendo não apenas conectividade Wi-Fi, mas também Bluetooth de baixa energia (BLE), ampliando suas possibilidades de aplicação em sistemas IoT (SYSTEMS, 2020).

Figura 2 – Microcontrolador ESP32.



Fonte: (ELETROGATE, 2023a)

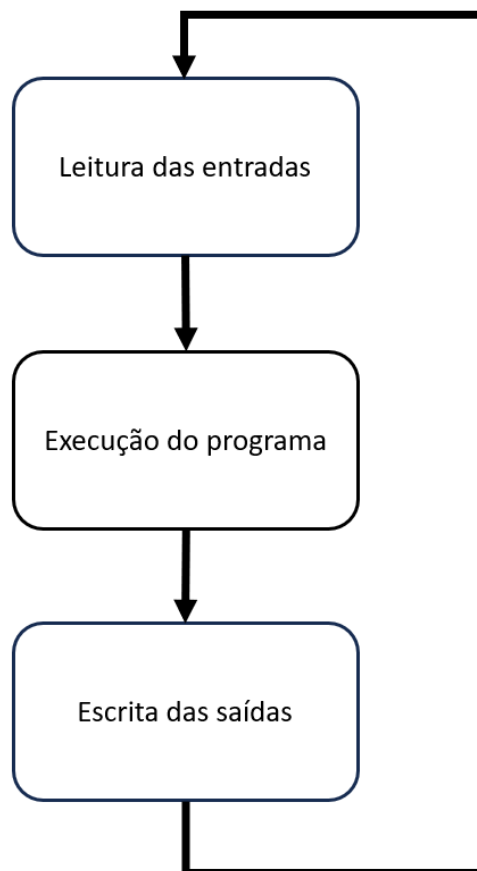
Com uma arquitetura dual-core e uma ampla variedade de periféricos, como GPIOs (General Purpose Input/Output), interfaces SPI, I2C e UART, além de sensores integrados, o ESP32 oferece uma base sólida para projetos de automação industrial e aplicações IoT mais complexas (SYSTEMS, 2020). Suas portas permitem uma variedade de conexões para sensores, atuadores e outros dispositivos, possibilitando a interação com o ambiente físico e a coleta de dados em tempo real.

Além disso, o ESP32 possui recursos de segurança embutidos, como criptografia de hardware e suporte para certificados digitais, atendendo às demandas de segurança para aplicações críticas (BROWN, 2020). Sua capacidade de processamento e a presença de uma ampla memória flash tornam o ESP32 uma opção viável para soluções industriais que requerem confiabilidade, conectividade e capacidade de processamento avançada (ROBERTS, 2019).

2.2.4 Ciclo de SCAN

O ciclo de SCAN (Figura 3) é o processo fundamental no funcionamento do CLP. Conforme estudos de Petruzella (2015), esse ciclo inicia com a leitura dos valores de entrada, seguido pela execução do programa armazenado na memória da CPU, onde a lógica de controle é aplicada. Após o processamento, os valores de saída são atualizados nos módulos de saída, acionando dispositivos externos. As instruções são executadas de cima para baixo e da direita para a esquerda. Esse ciclo ocorre de forma contínua e rápida, garantindo a atualização constante e precisa do controle sobre os processos industriais.

Figura 3 – Ciclo SCAN.



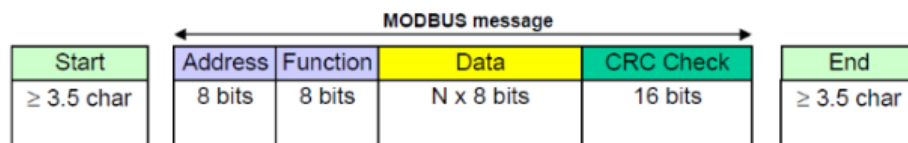
2.3 Redes e PLC

Os Controladores Lógicos Programáveis (CLPs) são frequentemente integrados em redes industriais para facilitar o controle e monitoramento de sistemas complexos (SMITH, 2020). Uma das principais tecnologias de comunicação nessas redes é o protocolo MODBUS, um padrão de comunicação serial amplamente utilizado em sistemas de automação industrial (GUPTA; AL., 2019).

O MODBUS opera com um esquema mestre/escravo, onde um dispositivo mestre, como um PLC ou um computador, faz requisições para dispositivos escravos, como sensores, atuadores ou outros PLCs, solicitando dados ou enviando comandos. O protocolo MODBUS suporta diferentes variantes, como MODBUS RTU (Remote Terminal Unit), MODBUS ASCII (American Standard Code for Information Interchange) e MODBUS TCP/IP (Transmission Control Protocol/Internet Protocol), que são adaptadas para diferentes meios de transmissão, como serial (RS-232/485) ou Ethernet (GUPTA; AL., 2019).

A simplicidade da estrutura de mensagem do MODBUS, que contém endereço de dispositivo, código de função, dados e verificação de erro, contribui significativamente para sua eficiência e confiabilidade (GUPTA; AL., 2019). Isso permite uma comunicação ágil e precisa entre os dispositivos, essencial para o funcionamento harmonioso dos sistemas industriais. A Figura 4 detalha a estrutura do MODBUS, evidenciando a organização clara de seus elementos, desde o início até o fim da comunicação: inicia com o endereço do dispositivo, seguido pelo código de função, dados, verificação de erro (CRC) e finaliza a transmissão. Essa representação visual da estrutura reforça a compreensão dos componentes-chave do protocolo, facilitando o entendimento e a implementação em ambientes industriais.

Figura 4 – Estrutura MODBUS.



Fonte: (DOCPLAYER, 2023).

Sua implementação simples facilita a integração em sistemas industriais e a interoperabilidade entre dispositivos de diferentes fabricantes (GOMEZ, 2020). Isso proporciona uma comunicação efetiva mesmo em ambientes com múltiplos dispositivos de variados fabricantes, promovendo a interconexão necessária para a automação industrial.

2.4 Desenvolvimento aberto de PLC

O avanço das tecnologias de código aberto tem desempenhado um papel significativo na evolução dos Controladores Lógicos Programáveis (CLPs), introduzindo novas abordagens para o desenvolvimento de hardware e software de automação industrial (ERFANIAN; SMITH, 2020). Um dos projetos notáveis nesse contexto é o OpenPLC, uma plataforma de código aberto que oferece um ambiente flexível para desenvolvimento de CLPs (CAVALCANTE; SILVA, 2019).

O OpenPLC é um software livre que permite a programação de CLPs utilizando uma variedade de linguagens, incluindo lógica Ladder, texto estruturado e diagrama de blocos, além de suportar comunicações com protocolos industriais comuns, como o MODBUS (CORRÊA; SANTOS, 2021). Esse projeto é uma resposta à demanda crescente por soluções de automação mais acessíveis, permitindo que engenheiros e entusiastas de automação experimentem e desenvolvam sistemas sem as restrições associadas aos softwares proprietários.

Além do OpenPLC, a comunidade de desenvolvimento aberto tem se concentrado no conceito de CLP open hardware, que se baseia na disponibilização de especificações e designs de hardware abertos para a criação de CLPs (CHIU; LEE, 2018). Projetos como o Olimex e o Raspberry Pi na automação industrial têm sido utilizados como base para o desenvolvimento de CLPs open hardware, fornecendo alternativas de baixo custo e altamente customizáveis para aplicações industriais.

Essas iniciativas estão redefinindo o panorama da automação industrial, permitindo o acesso a soluções de automação mais acessíveis, customizáveis e expansíveis para uma variedade de aplicações industriais.

2.5 Circuitos de proteção

Os Controladores Lógicos Programáveis (CLPs) são amplamente utilizados na automação industrial, desempenhando um papel vital no As instruções são eecutadas de cima para baixo e da direita para a esquerda. Porém, sua aplicação em ambientes industriais expõe esses dispositivos a diversos riscos, como surtos de corrente, variações de tensão e potenciais erros de operação que podem comprometer seu funcionamento adequado e, conseqüentemente, a estabilidade do processo industrial (SILVA, 2019).

Os surtos de corrente e tensão são fenômenos comuns em ambientes industriais, podendo danificar componentes sensíveis dos CLPs e, conseqüentemente, interromper a produção. Esses surtos podem ser causados por descargas atmosféricas, picos de energia na rede elétrica ou até mesmo operações inadequadas. Portanto, a implementação de circuitos de proteção é crucial para salvaguardar os CLPs contra tais eventos (BARBOSA, 2020).

Além disso, possíveis erros de operação, como curtos-circuitos ou falhas nos processos

de programação, podem comprometer o funcionamento correto dos CLPs. Portanto, é essencial adotar estratégias para minimizar esses riscos, como a implementação de fusíveis, dispositivos de proteção contra surtos e a utilização de protocolos de segurança específicos para a programação dos CLPs (SILVA, 2019).

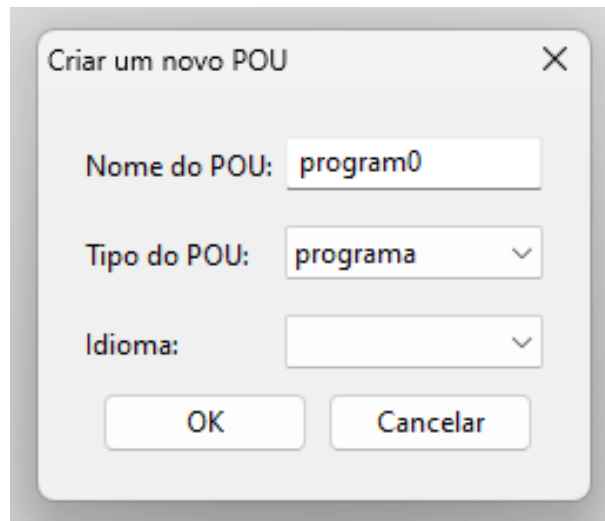
A proteção eficaz dos CLPs em ambientes industriais é fundamental para garantir a continuidade das operações. Estratégias de proteção bem planejadas não apenas preservam a integridade dos dispositivos, mas também contribuem para a estabilidade e segurança dos processos industriais.

3 METODOLOGIA

3.1 Preparação do ambiente

A metodologia adotada para a realização eficaz da implementação do OpenPLC em projetos de controle industrial segue uma abordagem sistemática, buscando aprimorar a eficiência durante as fases de instalação e configuração. Inicialmente, o procedimento compreende o *download* do OpenPLC Editor e Runtime a partir da fonte oficial, garantindo assim o acesso às versões mais recentes disponíveis. Posteriormente, foram seguidas as instruções dos assistentes de instalação para configurar o OpenPLC Editor, iniciando um novo projeto com a criação de uma pasta vazia e a seleção de uma linguagem de programação, como ilustrado na Figura 5. As opções de linguagem incluem IL (Instruction List), ST (Structured Text), LD (Ladder Diagram), FBD (Function Block Diagram) e SFC (Sequential Function Chart).

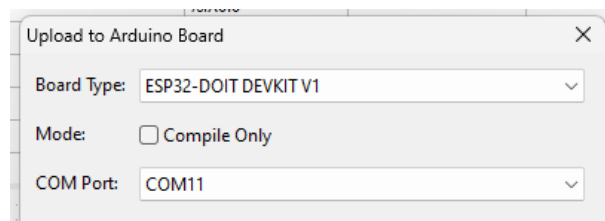
Figura 5 – Painel de configuração de um novo projeto.



Fonte: Autoria própria.

Um passo crucial envolve a configuração da placa principal, identificada como "Board Type" na Figura 6, para selecionar o microcontrolador específico que estamos utilizando. Além disso, é essencial especificar a localização de leitura deste microcontrolador, denominada "COM Port", que corresponde à entrada USB onde o microcontrolador está conectado.

Figura 6 – Painel de configuração e compilação do programa.



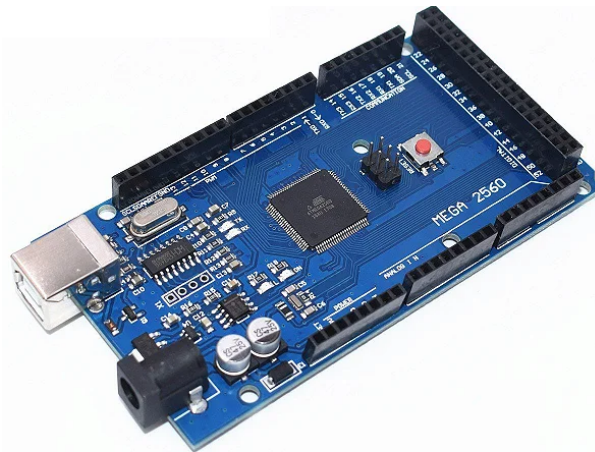
Fonte: Autoria própria.

3.2 Escolha do Microcontrolador

A seleção do microcontrolador fundamentou-se na análise dos principais dispositivos empregados no desenvolvimento de protótipos. Considerando o Arduino Mega, reconhecido por possuir diversas entradas e saídas; o STM32F103C8T6, destacado pelo seu desempenho e eficiência; e o ESP32, notável por suas capacidades de conectividade. Esses três modelos representam escolhas proeminentes de microcontroladores, cada um oferecendo distintas vantagens para aplicações diversas.

No contexto do Arduino Mega (Figura 7), destaca-se sua simplicidade operacional e a vasta comunidade de desenvolvedores. Entretanto, limitações em termos de poder de processamento e capacidade de comunicação podem impactar a escolha, especialmente em projetos mais complexos.

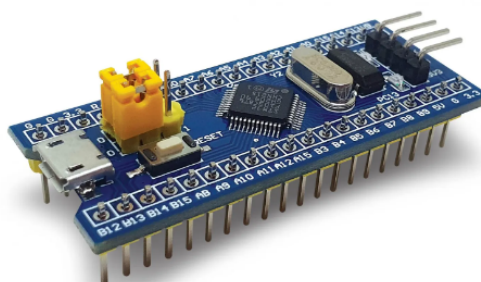
Figura 7 – Arduino Mega.



Fonte: (ROBÓTICA, 2023).

O STM32F103C8T6 (Figura 8), fundamentado na arquitetura ARM Cortex-M3, sobressai-se pelo poder de processamento superior e recursos periféricos avançados. No entanto, a complexidade associada a esses recursos pode representar um desafio para desenvolvedores menos experientes.

Figura 8 – STM32F103C8T6.



Fonte: (ROBOTICS, 2023)

Por sua vez, o ESP32 emerge como uma escolha vantajosa, oferecendo poder de processamento comparável ao STM32F103C8T6 e integrando conectividade Wi-Fi e Bluetooth. Essa característica singular torna o ESP32 atrativo em projetos de Internet das Coisas (IoT) e em aplicações que demandam comunicação sem fio.

As vantagens do ESP32 (Figura 9) evidenciam-se não apenas em sua flexibilidade, mas também em sua relação custo-benefício favorável. A presença de conectividade sem fio integrada, aliada a um desempenho sólido, o posiciona como uma escolha promissora para diversas aplicações, especialmente em cenários onde a comunicação e a conectividade são críticas para o sucesso do projeto.

Figura 9 – ESP32-DEVKIT-WROOM-32U.



Fonte: (USINAINFO, 2023).

A análise criteriosa (Tabela 1) dos principais microcontroladores para o desenvolvimento de protótipos - Arduino Mega, STM32F103 e ESP32 - conduziu à escolha do ESP32 como a solução mais adequada. Optou-se por utilizar o módulo de desenvolvimento ESP32-DEVKIT-WROOM-32U, reconhecendo sua versatilidade e excelente custo-benefício. Este módulo se destaca não apenas por sua capacidade de conectividade integradas, como Wi-Fi e Bluetooth, mas também por atender às demandas de processamento exigidas pelo projeto. Essa escolha estratégica foi fundamentada na busca por uma solução que não só atendesse aos requisitos de processamento, mas também garantisse uma comunicação eficiente, elemento crucial para os objetivos deste projeto. Especificamente, a necessidade de integração com Wi-Fi para a comunicação ModBus direcionou a escolha do ESP32, atendendo de forma fundamental aos requisitos de conectividade exigidos pelo protocolo ModBus. Dessa forma, o ESP32-DEVKIT-WROOM-32U foi estabelecido como a seleção mais apropriada para as demandas deste projeto, possibilitando uma comunicação eficaz e confiável.

Tabela 1 – Comparação entre ESP32, Arduino Mega e STM32F103C8T6

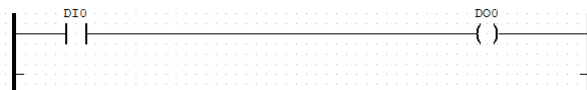
Características	ESP32	Arduino Mega	STM32F103C8T6
Número de GPIO	36	54	37
Entradas digitais	36	54	37
Saídas digitais	36	15	37
Entradas analógicas	18	16	10
Saídas analógicas	02	00	00
WiFi integrado	Sim	Não	Não
Bluetooth integrado	Sim	Não	Não
Alimentação	3.3V/5V	5V/12V	3.3V/5V
Tamanho	Pequeno	Grande	Pequeno
Valor	R\$40-R\$80	R\$150-R\$200	R\$30-R\$60

Fonte: Autoria própria.

3.3 Teste de acionamento básico

A fim de validar a funcionalidade do OpenPLC no ESP32, foi conduzido um teste preliminar utilizando uma *protoboard* e alguns componentes eletrônicos. O procedimento inicial foi direto e consistiu na elaboração de um programa em linguagem Ladder, onde um contato normalmente aberto (NA) foi utilizado como entrada e uma bobina como saída. O objetivo desse teste foi simples, ao acionar o botão o programa acende um LED, e pode ser observado na Figura 10.

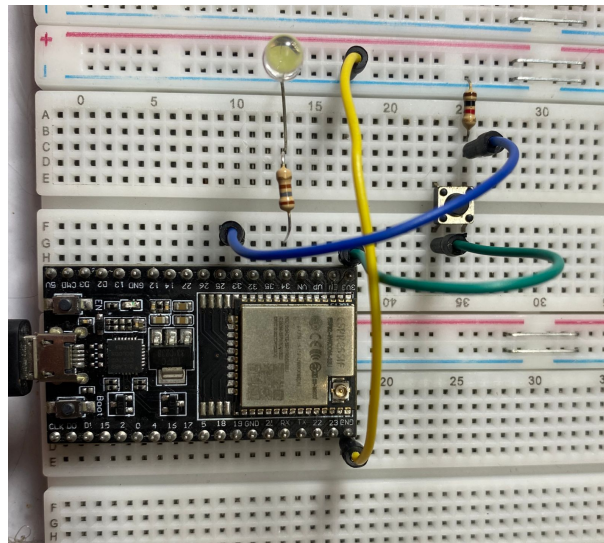
Figura 10 – Programa do primeiro teste do OpenPLC.



Fonte: Autoria própria.

Para realizar essa operação, foi montado um circuito básico (Figura 11) com os seguintes componentes: o microcontrolador ESP32-DEVKIT-WROOM-32U, um LED, um resistor de 180 Ω e outro de 1 $k\Omega$, além de um botão de pressão e alguns jumpers para conexão. O circuito foi alimentado usando a própria saída de tensão do ESP32.

Figura 11 – Circuito de teste.



Fonte: Autoria própria.

Ao validar a funcionalidade do OpenPLC no ESP32 por meio de um teste preliminar, os resultados obtidos foram satisfatórios, atendendo o objetivo estabelecido, que consistia em acender um LED ao pressionar um botão

3.4 Esquemático das entradas e saídas

A etapa inicial na elaboração do esquemático das entradas e saídas do protótipo envolveu a definição da capacidade de entradas e saídas digitais e analógicas viáveis no ESP32. Esta fase teve foco na a pesquisa na documentação do OpenPLC, visando entender a correlação dos terminais do ESP32 com a alocação de entradas, saídas e memórias. Conforme detalhado na documentação do OpenPLC, o ESP32 permite o uso de até 9 entradas digitais, 10 saídas digitais, 4 entradas analógicas e 2 saídas analógicas. Além disso, a documentação delineou a relação específica dos pinos com seus respectivos endereços, destacada na Tabela 2.

Tabela 2 – Relação dos terminais do ESP32 e endereços do OpenPLC.

Tipo	Terminais	Endereço
Entradas digitais	17, 18, 19, 21, 22, 23, 27, 32 33	%IX0.0 – %IX0.7 %IX1.0 – %IX1.0
Saídas digitais	01, 02, 03, 04, 05, 12, 13, 14 15, 16	%QX0.0 – %QX0.7 %QX1.0 – %QX1.1
Entradas analógicas	34, 35, 36, 39	%IW0 – %IW3
Saídas analógicas	25, 26	%QW0 – %QW1

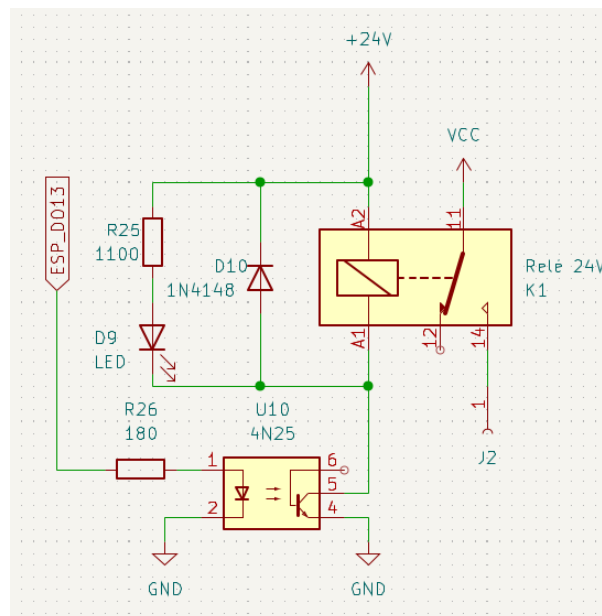
Com base nas informações contidas na tabela 2, foi determinado a utilização de 8 en-

tradas digitais, 9 saídas digitais, além das entradas e saídas analógicas disponíveis. No entanto, uma das saídas digitais foi designada para uma função específica interna ao protótipo. Essa saída digital específica terá a responsabilidade de definir o valor da tensão nas demais saídas, estabelecendo entre 5V ou 24V. Os esquemáticos completos podem ser observado no Apêndice A.

3.4.1 Saídas digitais

No processo de desenvolvimento do circuito, foi estabelecido um padrão para as saídas digitais, optando pelo uso de relés devido à sua praticidade e eficiência. Para garantir a proteção do circuito contra picos de tensão gerados pelo desligamento dos relés, empregou-se um relé de 24V em paralelo com um diodo 1N4148, atuando como um diodo de supressão de tensão. Além disso, integrou-se um LED em série com um resistor de 1,1 k Ω , também conectado em paralelo com os relés, para indicar o estado de funcionamento das saídas. O acionamento dos relés foi viabilizado por meio de um optoacoplador 4N25, permitindo a ativação do relé de 24V a partir do sinal de 3,3V do ESP32. Para limitar a corrente de saída do ESP32, foi incorporado um resistor de 180 Ω em série com o LED do 4N25. O esquemático completo desse circuito pode ser visualizado na Figura 12.

Figura 12 – Saída digital do CLP.

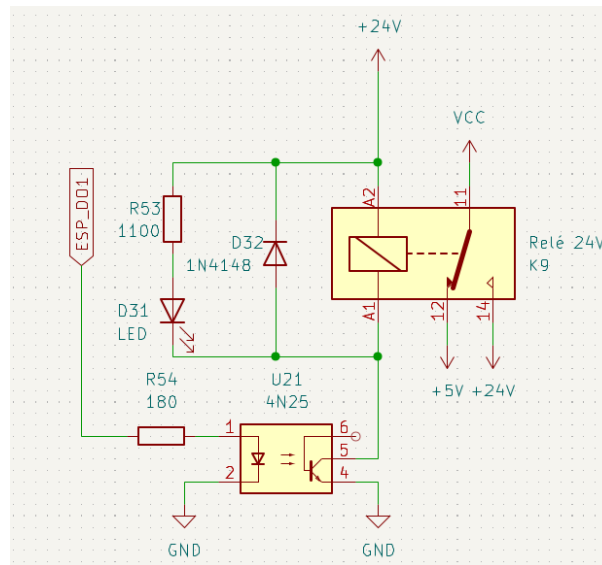


Fonte: A autoria própria.

Para a elaboração da saída digital encarregada da configuração interna para selecionar a tensão das outras saídas digitais, foi desenvolvido um circuito similar, mas com conexões de relé distintas. Neste caso, os relés foram conectados às fontes de tensão de 24V e 5V, e não ao conector da saída digital como nos demais circuitos. Esta disposição foi estabelecida para permitir que esta saída digital controle a tensão de operação das outras saídas, sem diretamente influenciar

os conectores das saídas digitais. Essa configuração permitirá a seleção da tensão desejada entre 24V e 5V conforme necessário para as outras saídas digitais, conferindo versatilidade ao sistema e possibilitando diferentes cenários de operação.

Figura 13 – Saída digital responsável por selecionar a tensão.



Fonte: A autoria própria.

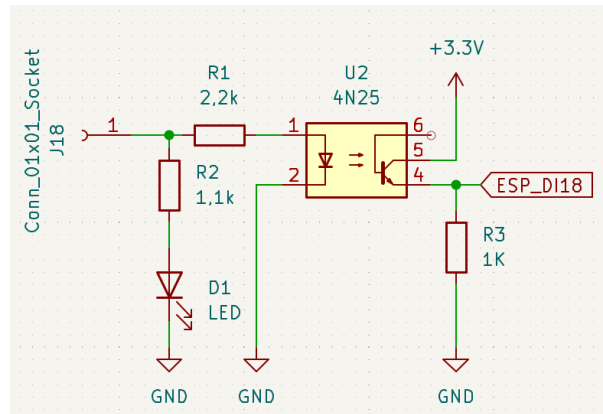
3.4.2 Entradas digitais

No desenvolvimento das entradas digitais, foi adotada a utilização do optoacoplador 4n25 como elemento de conversão do sinal de entrada de 24V para os 3,3V aceitos pelo ESP32. Adicionalmente, foi introduzido na entrada do sinal digital um LED em série com um resistor de 1,1kΩ, o qual foi calculado seguindo a Lei de Ohm (Equação 3.1) e considerando a tensão de operação do LED de 1,8V e corrente de 20 mA. Essa configuração tem a função de indicar o estado de funcionamento da entrada, proporcionando uma visualização clara do seu status operacional.

$$V = I \times R \quad (3.1)$$

Na saída do 4N25, foi empregado um resistor de pull-down, usada para manter um pin de entrada em um nível de tensão baixo, de 1kΩ, responsável por fornecer o sinal de tensão de 3,3V para a entrada digital do ESP32, garantindo a comunicação adequada entre os componentes. O diagrama completo deste circuito está disponível na Figura 14.

Figura 14 – Entrada digital do CLP.



Fonte: Autoria própria.

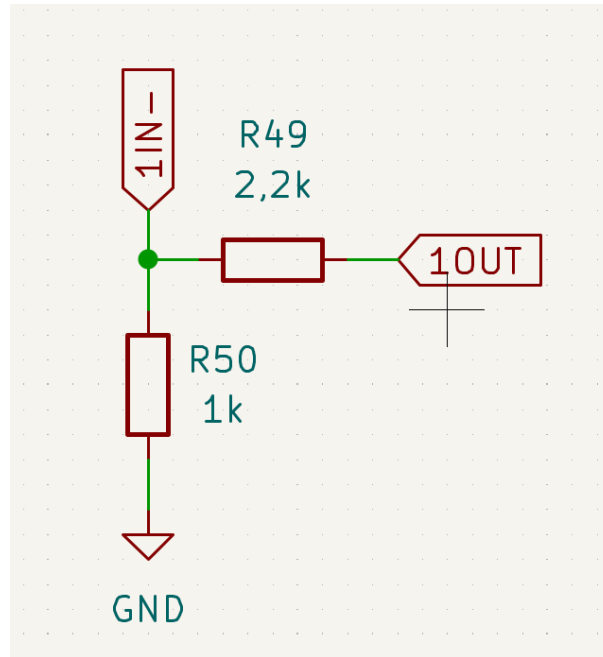
3.4.3 Saídas analógicas

As saídas analógicas foram projetadas para operar com um sinal de 0 a 10V. Para atender a esse requisito, foi necessário empregar um amplificador LM358, responsável por converter o sinal analógico variável de 0 a 3,3V gerado pelo ESP32. A determinação dos valores dos resistores seguiu a fórmula do amplificador operacional não inversor, demonstrado na Equação 3.2.

$$\text{Ganho} = 1 + \frac{R2}{R1} \quad (3.2)$$

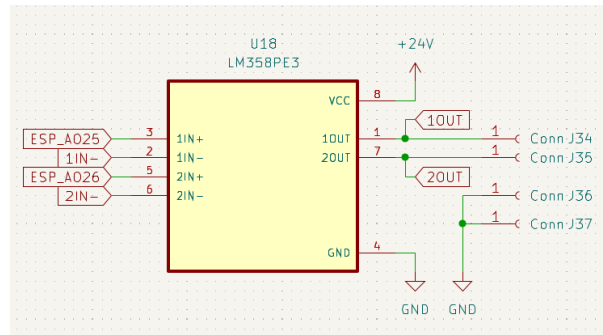
Para isso, foi definido o R2 como 2,2kΩ por ser um resistor de comercial, e calculado o R1. Após os cálculos, ficou estabelecido que o R1 deveria ser de 1,1kΩ; o resistor comercial mais próximo adotado foi o de 1kΩ. O LM358 foi alimentado com uma tensão de 24V para garantir o correto funcionamento do circuito. Os esquemáticos detalhados das saídas analógicas foram representados nas Figuras 15 e 16, demonstrando a configuração adotada para permitir a conversão do sinal de 0 a 3,3V do ESP32 para um sinal de 0 a 10V, essencial para o correto desempenho das saídas analógicas.

Figura 15 – Associação de resistores da saída analógica do CLP.



Fonte: Autoria própria.

Figura 16 – Amplificador operacional da saída analógica do CLP.



Fonte: Autoria própria.

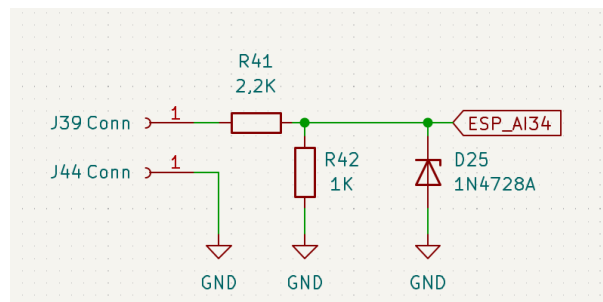
3.4.4 Entradas analógicas

As entradas analógicas foram planejadas para receber um sinal de tensão variável de 0 a 10V. Para viabilizar essa funcionalidade, foi implementado um arranjo com resistores e diodos Zener, operando como um divisor de tensão. Essa configuração foi projetada para converter a faixa de entrada de 0-10V para uma faixa mais adequada de leitura de 0-3,3V, alinhada com as capacidades da entrada analógica do ESP32.

$$V_{\text{saida}} = V_{\text{entrada}} \times \frac{R2}{R1 + R2} \tag{3.3}$$

No processo de determinação dos valores dos resistores, foi utilizado a fórmula do divisor de tensão, Equação 3.3, como guia. Foi definido um valor inicial de R1 como 2,2 k Ω . Através dos cálculos, foi encontrado o valor teórico para R2 como 1083 Ω . Contudo, dado que não há resistores comerciais disponíveis com esse valor específico, optou-se por empregar um resistor de 1 k Ω como uma solução prática para desempenhar o papel de R2. Assim como podemos observar na Figura 17, o esquemático da saída analógica foi projetado utilizando um divisor de tensão com R1 de 2,2 k Ω , um R2 de 1 k Ω e um diodo Zener de 3,3v para a proteção da entrada analógica do ESP32.

Figura 17 – Entrada analógica do CLP.



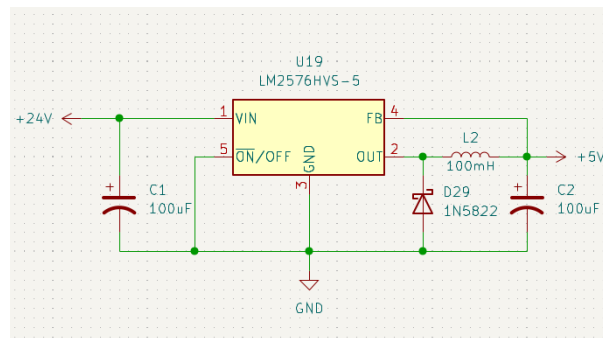
Fonte: Autoria própria.

3.5 Fonte de alimentação

Para a fonte de alimentação do circuito, identificou-se a necessidade de três níveis de tensão distintos: 24V, 5V e 3,3V. Para atender a essa demanda, foram desenvolvidos dois circuitos para reduzir a tensão de entrada de 24V, utilizada como alimentação do protótipo.

O primeiro circuito, apresentado na Figura 18, emprega um regulador de tensão LM2576-5V para reduzir a tensão de 24V para 5V. Esse regulador é capaz de fornecer até 3A na saída. Em conformidade com as especificações do LM2576-5V, foram empregados dois capacitores eletrolíticos de 100 μ F, um na entrada e outro na saída, responsáveis por filtrar o sinal de entrada e saída do regulador. Além disso, é recomendado o uso de um diodo zener 1N5822 e um indutor de 100mH na saída do regulador para garantir a estabilidade do circuito.

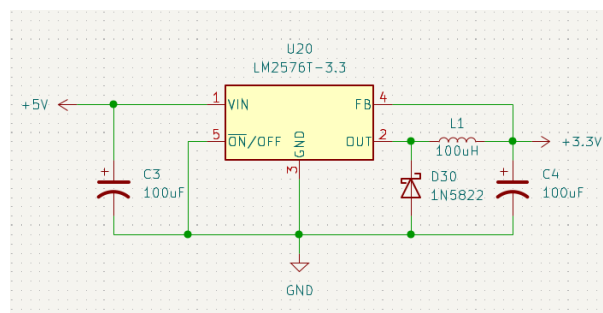
Figura 18 – Regulador de tensão de 5V.



Fonte: Autoria própria.

O segundo circuito utiliza um LM2576-3,3V para reduzir a tensão de 5V para 3,3V. Por fazer parte da mesma família do LM2576-5V mencionado anteriormente, as recomendações de componentes, como capacitores, diodo e indutor, permanecem semelhantes. Assim, foram empregados capacitores eletrolíticos de 100µF para filtrar o sinal, juntamente com o diodo zener 1N5822 e o indutor de 100mH. O circuito responsável por fornecer a tensão de 3,3V está ilustrado na Figura 19.

Figura 19 – Regulador de tensão de 3,3V.



Fonte: Autoria própria.

3.6 Circuitos de proteção

Para a proteção do circuito, foram adotados dispositivos simples, porém eficazes, para garantir a segurança do sistema. O primeiro elemento considerado para a proteção foi o optoacoplador 4N25 (Figura 20), responsável por chavear a corrente dos relés das saídas digitais. Ele atua como uma barreira de proteção para o ESP32, uma vez que a corrente proveniente da fonte de 24V, passando pelo relé, não apresenta risco de afetar diretamente o microcontrolador. Isso se deve à ausência de conexão física, já que o acionamento é realizado por um LED e um fototransistor.

Figura 20 – Optoacoplador 4n25.



Fonte: (ELETROGATE, 2023b).

Outros dispositivos de proteção amplamente empregados no protótipo foram os diodos. Eles foram utilizados em configuração paralela como diodos de supressão de tensão, prevenindo picos de tensão induzidos pela bobina do relé. Além disso, diodos zener (Figura 21) foram empregados nas entradas analógicas. Os diodos zener utilizados nessas entradas possuem uma tensão de 3,3V, protegendo, assim, as entradas analógicas do ESP32 contra sobretensões.

Figura 21 – Diodo zener.



Fonte: (PROESI, 2023a)

O último dispositivo de proteção implementado foi um fusível de 2A (Figura 22), localizado na alimentação do protótipo. Ele desempenha o papel de proteger o circuito contra picos

de corrente e curtos-circuitos, contribuindo para a segurança geral do sistema.

Figura 22 – Fusível de vidro de 2A.

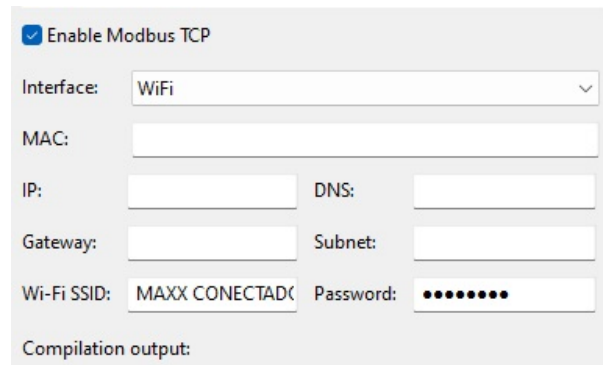


Fonte: (PROESI, 2023b).

3.7 Teste de comunicação

O teste de conectividade e comunicação em rede foi conduzido por meio do emprego do *software open source* SCADA BR, um sistema supervisor robusto que oferece suporte à comunicação MODBUS TCP. Esse teste exigiu a elaboração de um programa em linguagem Ladder no OpenPLC Editor, posteriormente compilado e carregado em um dispositivo ESP32. Durante o procedimento de *upload* do código para o ESP32, foi necessário habilitar a opção 'Enable Modbus TCP' no OpenPLC Editor. Além disso, para garantir a integração do ESP32 à rede, foram inseridos os detalhes de conexão Wi-Fi, incluindo o SSID (Service Set Identifier - identificador de rede Wi-Fi) e a senha correspondente. Essa configuração, detalhada na Figura 23, permitiu que o ESP32 se conectasse à rede Wi-Fi, estabelecendo um canal para a comunicação via Modbus TCP.

Figura 23 – Configuração do Modbus RTU.



Enable Modbus TCP

Interface: WiFi

MAC:

IP: DNS:

Gateway: Subnet:

Wi-Fi SSID: MAXX CONECTAD Password: ●●●●●●●●●●

Compilation output:

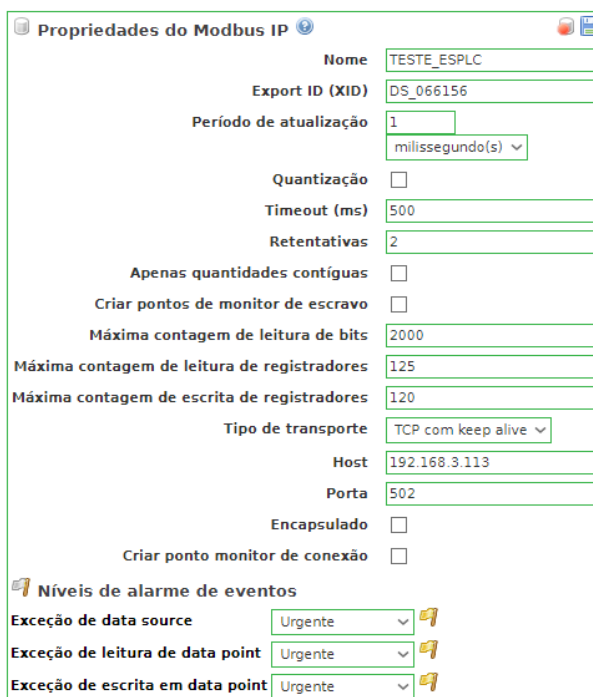
Fonte: Autoria própria.

Durante a execução deste estudo, a configuração do *software* SCADA BR foi crucial para estabelecer uma comunicação eficiente com o dispositivo ESP32, utilizando o protocolo Modbus TCP. O processo começou com a obtenção do *software* a partir do repositório do GitHub, seguido pela instalação conforme as instruções fornecidas. Após a instalação, o acesso inicial ao SCADA BR foi feito, utilizando as credenciais padrão ('admin') para o login inicial, permitindo o início do processo de configuração.

Uma etapa fundamental foi a criação de um Data Source no SCADA BR, um passo necessário para viabilizar a comunicação via Modbus TCP. Durante essa fase, informações específicas foram requeridas para estabelecer a conexão desejada entre o *software* e o dispositivo ESP32. Com base nos objetivos do estudo, os parâmetros foram ajustados para otimizar a comunicação: o período de atualização foi configurado para 1ms e o tipo de transporte foi definido como 'TCP com keep alive'. O HOST foi preenchido com o endereço IP atribuído ao ESP32 na rede local, possibilitando o estabelecimento da conexão necessária.

Para determinar o endereço IP do ESP32 na rede local, um procedimento específico foi adotado. Utilizando o *prompt* de comando no Windows, primeiramente o comando "ipconfig" foi executado para acessar as configurações de rede. Em seguida, o comando "arp -a" permitiu visualizar os IPs dos dispositivos conectados à rede. Esse teste foi realizado antes e depois do *upload* do programa no ESP32. Após a configuração do Modbus na porta 502, padrão para essa comunicação, foi possível observar o novo IP atribuído ao dispositivo, permitindo a identificação específica do ESP32 na rede. As configurações realizadas para a comunicação Modbus podem ser observadas na Figura 24.

Figura 24 – Configuração do Modbus TCP no SCADA BR.



Fonte: Autoria própria.

Após a configuração do protocolo Modbus, foi necessário delimitar os tipos de variáveis a serem monitoradas no sistema. Isso foi alcançado por meio da criação de um Data Point no ambiente do SCADA BR. Para as variáveis de saída, identificadas como bobinas no programa, os endereços foram especificados removendo o prefixo e inserindo apenas o número do *offset*. Por exemplo, a variável com o endereço *%QX0.5* no programa foi adicionada no sistema como "DO5", utilizando apenas o "5" como identificador.

Da mesma forma, para as variáveis de entrada, identificadas pelo endereço *%IX0.0*, foi utilizado somente o número do *offset*, neste caso, "0". Essa variável foi designada no sistema como "DI0".

Na Figura 25, estão documentadas as configurações mencionadas anteriormente para os Data Points no ambiente do SCADA BR.

Figura 25 – Configuração da entrada e saída no SCADA BR.

Data points						
Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)	
DI0	Binário		1	Entrada Discreta (Input Status)	0	
DO5	Binário		1	Bobina (Coil Status)	5	

Fonte: Autoria própria.

Na Figura 26, é apresentado um sistema supervisorio simplificado, concebido para moni-

torar o estado das variáveis do sistema. Este sistema consiste em elementos básicos de interface, contendo um gráfico para visualização dos níveis lógicos das variáveis e um botão indicativo do status de ligado ou desligado da entrada. A simplicidade da interface permite uma rápida verificação visual dos estados das variáveis, oferecendo uma representação visual direta e clara para o acompanhamento em tempo real dos dados provenientes do sistema monitorado.

Figura 26 – Supervisório para teste de comunicação.



Fonte: Autoria própria.

O teste de conexão utilizando o protocolo Modbus e a rede *wireless* local foi conduzido com sucesso, apesar da limitação identificada no ESP32, que não permitiu a escrita de variáveis através desse protocolo. O foco do teste concentrou-se na leitura das variáveis por meio de um sistema supervisorio de código aberto, possibilitando a visualização e monitoramento das variáveis em um ambiente prático.

3.8 Desenvolvimento do layout da PCB

O processo de decisão para determinar os tipos de componentes a serem utilizados no layout da PCB foi baseado nas necessidades específicas do protótipo, levando em consideração a montagem artesanal em uma placa de fenolite de dupla face. Optou-se por componentes THT (Through-Hole Technology) para simplificar o processo de soldagem e manter a continuidade das trilhas entre as faces da placa.

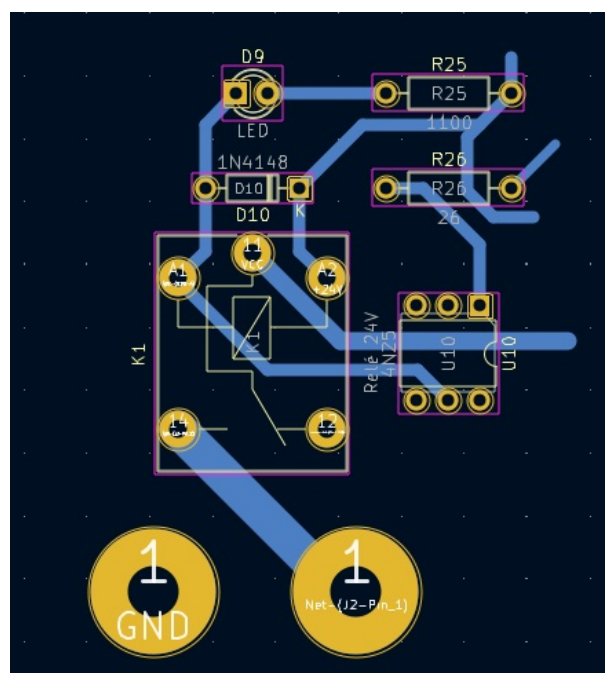
O desenvolvimento do layout da PCB foi conduzido no KiCad, um software de código aberto voltado para o design de PCB. Os esquemáticos previamente trabalhados foram replicados conforme a quantidade necessária de entradas e saídas, sejam elas analógicas ou digitais. Além disso, os circuitos para regular a tensão para 5V e 3,3V, bem como o circuito do fusível para proteção do protótipo, foram incorporados ao projeto.

Adicionalmente, foi criado um esquemático (Figura 27) dedicado a endereçar todas as entradas e saídas aos terminais correspondentes do ESP32. Essa etapa é fundamental para garan-

incluem informações detalhadas sobre vários aspectos do design, como a largura das trilhas, a qual é determinada com base na corrente elétrica esperada, visando garantir a funcionalidade e confiabilidade do circuito impresso.

Para organizar e agrupar eficientemente o circuito como um todo, foi imprescindível desenvolver layouts individuais para cada circuito do esquemático. Como exemplificado na Figura 31, foi elaborado o layout específico para o circuito da saída digital. Esta abordagem permitiu uma organização mais clara e uma visão detalhada de cada componente e conexão, facilitando a compreensão e a montagem posterior do circuito.

Figura 28 – Layout da saída digital.



Fonte: Autoria própria.

No processo de integração dos circuitos individuais, a conexão e a organização das saídas e entradas foram essenciais para estabelecer um padrão lógico. As saídas foram dispostas na parte inferior, enquanto as entradas foram posicionadas na parte superior, criando uma disposição coesa. O layout final da PCB, observado na Figura 38 e Figura 41 dispostas na seção APÊNDICE B, contemplou as 8 saídas digitais, 8 entradas digitais, 4 entradas analógicas e 2 saídas analógicas, todas definidas e mapeadas conforme descrito na Tabela 3.

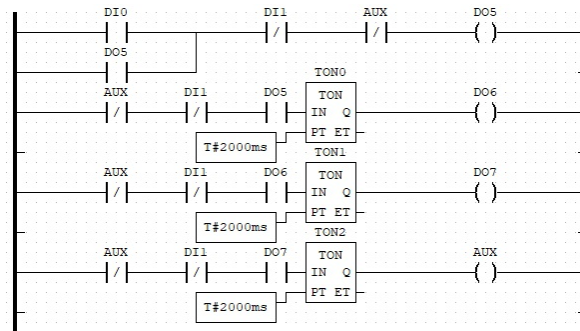
Tabela 3 – Mapeamento de terminais e funções nos esquemáticos PLC e ESP.

Terminal	Nome	Função 2	Nome Esquemático PLC	Nome Esquemático ESP
3	VP	%IW2	PLC_AI3	ESP_AI36
4	VN	%IW3	PLC_AI4	ESP_AI39
5	IO34	%IW0	PLC_AI1	ESP_AI34
6	IO35	%IW1	PLC_AI2	ESP_AI35
7	IO32	%IO0.7	PLC_DI8	ESP_DI32
9	IO25	%QW0	PLC_AO1	ESP_AO25
10	IO26	%QW1	PLC_AO2	ESP_AO26
11	IO27	%IO0.6	PLC_DI7	ESP_DI27
12	IO14	%QX0.7	PLC_DO3	ESP_DO14
13	IO12	%QX0.5	PLC_DO2	ESP_DO12
15	IO13	%QX0.6	PLC_DO1	ESP_DO13
23	IO15	%QX1.0	PLC_DO8	ESP_DO15
24	IO2	%QX0.1	PLC_DO7	ESP_DO2
26	IO4	%QX0.3	PLC_DO6	ESP_DO4
28	IO17	%IO0.0	PLC_DI1	ESP_DI17
29	IO5	%QX0.4	PLC_DO5	ESP_DO5
30	IO18	%IO0.1	PLC_DI2	ESP_DI18
31	IO19	%IO0.2	PLC_DI3	ESP_DI19
33	IO21	%IO0.3	PLC_DI4	ESP_DI21
34	RX	%QX0.2	PLC_DO4	ESP_DO3
35	TX	%QX0.0	PLC_24V_5V	ESP_DO1
36	IO22	%IO0.4	PLC_DI5	ESP_DI22
37	IO23	%IO0.5	PLC_DI6	ESP_DI23

4 RESULTADOS

O protótipo foi submetido a um ensaio prático para garantir seu funcionamento. Foi elaborado um código em linguagem Ladder, como ilustrado na Figura 29, composto por três bobinas e duas entradas. A lógica operacional consiste em ativar a entrada DI0 para acionar a primeira bobina (DO5) e selar o circuito, enquanto um temporizador TON é iniciado, contando 2000ms. Após a contagem, a segunda bobina (DO6) é ativada juntamente com outro temporizador TON de 2000ms. Esse processo se repete para a terceira bobina (DO7). No término da contagem, as três bobinas são desativadas. A segunda entrada, DI1, é destinada a desligar o sistema em qualquer fase do processo. O programa Ladder foi carregado juntamente com a configuração do Modbus TCP para garantir o funcionamento adequado do sistema.

Figura 29 – Programa ladder do ensaio.



Fonte: Autoria própria.

O sistema supervisório (Figura 30) desenvolvido no SCADA BR desempenha a função de monitorar o status das variáveis. Utilizando uma imagem de fundo como base, configuramos animações que refletem o estado atual do sistema. Os LEDs no painel indicam se as entradas foram ativadas, enquanto os ícones de ventiladores representam os estados das saídas. Foi configurado e conectado cada uma das entradas e saídas de acordo com as especificações da rede *wireless* local, permitindo uma visualização clara e precisa do funcionamento do sistema.

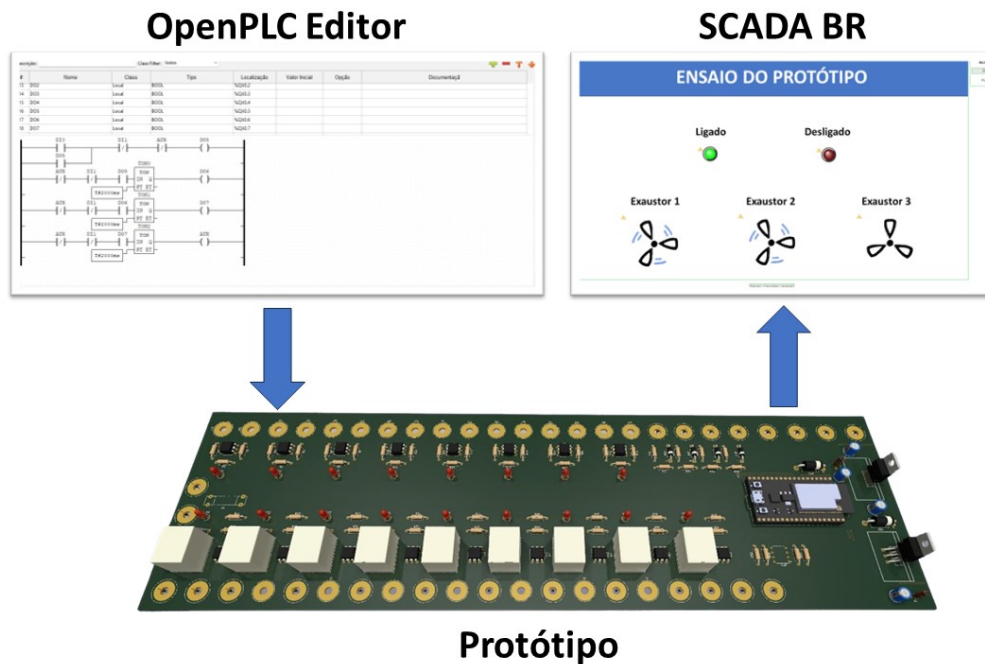
Figura 30 – Sistema supervisório do ensaio.



Fonte: Autoria própria.

O fluxo de comunicação entre o protótipo e os softwares *open source* é ilustrado na Figura 31. O OpenPLC Editor, responsável pelo desenvolvimento do ladder, compila o programa e o carrega no protótipo. Enquanto isso, o SCADA BR, encarregado de monitorar o estado das variáveis, estabelece a comunicação Modbus TCP por meio da rede local para receber e exibir os dados do sistema em tempo real.

Figura 31 – Fluxograma do ensaio



Fonte: Autoria própria.

Os resultados obtidos através dos ensaios práticos e da integração dos *softwares open source* com o protótipo demonstram a viabilidade e eficácia do sistema desenvolvido. A execução bem-sucedida do código ladder, aliada à configuração precisa do sistema supervisor no SCADA BR, oferece uma visualização detalhada e em tempo real do comportamento das variáveis. Além disso, a fluidez do fluxo de comunicação entre o OpenPLC Editor e o SCADA BR, por meio do Modbus TCP, ressalta a robustez e a confiabilidade da estrutura implementada. Esses resultados validam a funcionalidade do protótipo, destacando seu potencial para aplicação em ambientes que demandem controle e supervisão de processos industriais de forma eficiente e acessível.

5 CONCLUSÃO

Este trabalho destaca a construção e aplicabilidade de um protótipo de *hardware* integrado a *softwares open-source*, destinado a laboratórios acadêmicos no campo de controle e automação. Esse ambiente oferece melhorias substanciais no ensino, proporcionando uma experiência abrangente aos alunos, desde a programação até a implementação do protocolo Modbus para utilização em sistemas supervisórios.

O funcionamento eficaz dos *softwares open-source* demonstrou que ferramentas de baixo custo têm o potencial de enriquecer o aprendizado em automação de processos. Durante o desenvolvimento do *hardware*, o *software* KiCad foi fundamental. A escolha do ScadaBR como supervisório foi apropriada devido à sua robustez e suporte comunitário, enquanto o servidor OpenPLC destacou-se por sua simplicidade e eficácia na programação.

Entretanto, há perspectivas de aprimoramento e avanços para o protótipo. A adição de um display OLED para visualização do IP do ESP32 pode aprimorar significativamente a usabilidade do sistema, facilitando a consulta e configuração do dispositivo. Além disso, a adaptação das entradas e saídas analógicas para o sinal de 4-20mA expandiria a gama de sensores e dispositivos compatíveis, proporcionando uma integração mais abrangente com equipamentos industriais padronizados.

Essas melhorias potenciais representam uma evolução significativa para o sistema, tornando-o mais acessível, adaptável e compatível com uma variedade maior de dispositivos e ambientes industriais. Considerando essas perspectivas futuras, a continuidade dos aprimoramentos pode fortalecer ainda mais a aplicabilidade e eficiência do sistema para uma ampla gama de aplicações didáticas e industriais.

REFERÊNCIAS

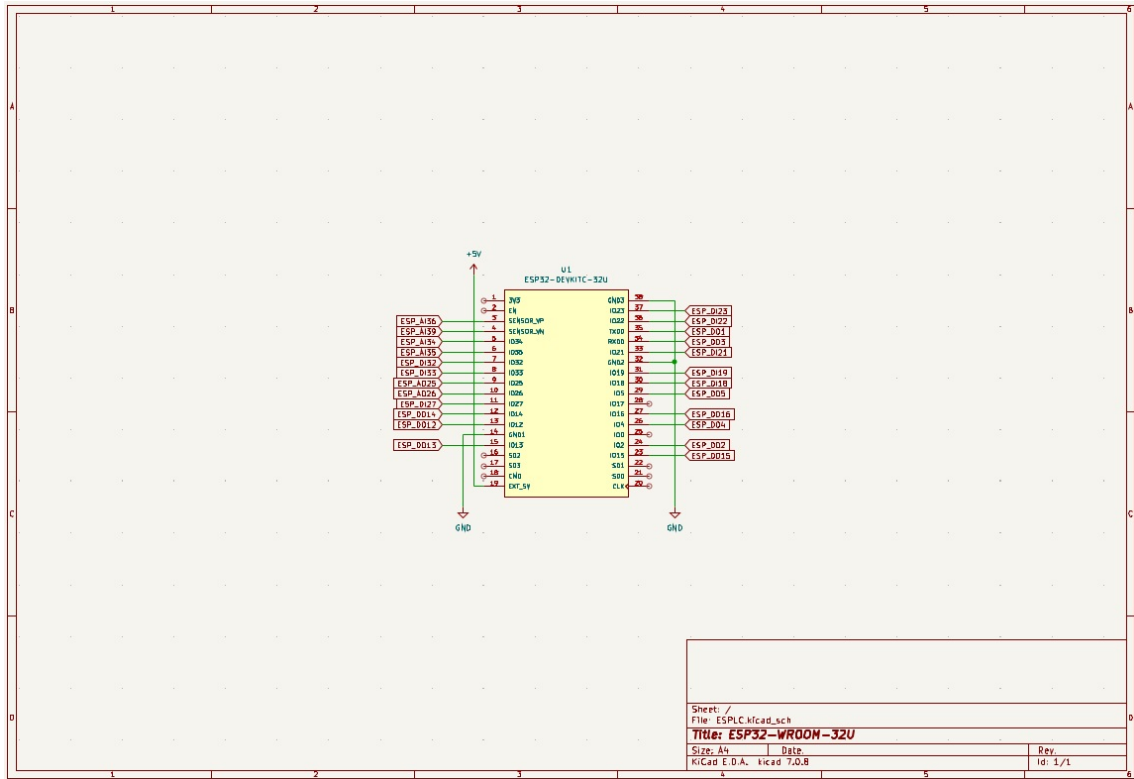
- BARBOSA, A. M. Ensuring reliability of plcs in industrial environments. In: **Proceedings of IEEE Industrial Automation Conference**. [S.l.: s.n.], 2020. Citado na página 22.
- BOLTON, W. **Programmable Logic Controllers**. [S.l.]: Elsevier, 2016. Citado na página 18.
- BROWN, T. Security features in esp32. **Journal of IoT Security**, v. 5, n. 2, p. 45–56, 2020. Citado na página 20.
- CARRASCO, J. M. *et al.* Analysis and design of plc-based control systems. **Control Engineering Practice**, v. 77, p. 10–22, 2018. Citado 2 vezes nas páginas 16 e 18.
- CAVALCANTE, R.; SILVA, A. Openplc: An open source alternative for plcs development. In: **Proceedings of the International Conference on Industrial Automation**. [S.l.: s.n.], 2019. Citado na página 22.
- CHIU, W.; LEE, S. Open hardware approach for programmable logic controllers. **Open Engineering Journal**, v. 7, p. 89–102, 2018. Citado na página 22.
- CORRÊA, L.; SANTOS, M. Openplc: A flexible open source platform for plcs supporting modbus communication. **Journal of Industrial Computing**, v. 15, n. 2, p. 112–125, 2021. Citado na página 22.
- DOCPLAYER. **Redes Industriais Modbus RTU**. 2023. <<https://docplayer.com.br/6403430-Redes-industriais-modbus-rtu.html>>. [Acessado em 4 de novembro de 2023]. Citado na página 21.
- ELECTRIC, S. **Modicon: 50 anos de pioneirismo e inovações**. 2018. <<https://blog.se.com.br/automacao-industrial/2018/11/30/modicon-50-anos-de-pioneirismo-e-inovacoes/>>. [Acessado em 4 de novembro de 2023]. Citado na página 17.
- ELETROGATE. **Chip ESP32 WROOM-32U**. 2023. <<https://www.eletrogate.com/chip-esp32-wroom-32u>>. [Acessado em 4 de novembro de 2023]. Citado na página 19.
- _____. **ESP32 WROOM 32U DevKitC V4 com WiFi e Bluetooth**. 2023. <<https://www.eletrogate.com/circuito-integrado-4n25-optoacoplador>>. [Acessado em 28 de novembro de 2023]. Citado na página 35.
- ERFANIAN, M.; SMITH, J. The role of open source technologies in the evolution of programmable logic controllers. **Industrial Automation Journal**, v. 8, n. 3, p. 45–56, 2020. Citado na página 22.
- FRANCHI, L.; CAMARGO, A. Plc: An overview of applications. **Automation Today**, v. 12, n. 4, p. 45–52, 2008. Citado na página 18.
- GOMEZ, E. Understanding the modbus protocol in industrial automation. **Industrial Communication Journal**, v. 12, n. 2, p. 78–91, 2020. Citado na página 21.
- GUPTA, S.; AL. *et.* Efficiency and reliability in modbus. **Industrial Communication Review**, v. 7, n. 4, p. 212–225, 2019. Citado na página 21.
- JOHNSON, D. Role of the cpu in plc systems. **Industrial Computing Review**, v. 25, n. 3, p. 45–57, 2020. Citado na página 19.

- LEE, S.-H. Evolution and outlook of plc technologies. **Journal of Electrical Engineering and Technology**, v. 12, n. 4, p. 1371–1381, 2017. Citado 2 vezes nas páginas 16 e 17.
- MANSFIELD, M. Role of input and output modules in plc systems. **Industrial Automation Journal**, v. 15, n. 2, p. 78–89, 2018. Citado na página 19.
- MORAD, A. Industrial control systems: From relay to plc. **IEEE Transactions on Industrial Informatics**, v. 15, n. 5, p. 2985–2993, 2019. Citado na página 16.
- MORAES, C. Dick morley: the creator of the first plc and a pioneer of the industrial revolution. **International Journal of Engineering Research and Technology**, v. 13, n. 9, p. 2008–2013, 2020. Citado 2 vezes nas páginas 16 e 17.
- MORAES, C.; CASTRUCCI, P. Automated production systems: Plcs. **International Journal of Industrial Automation**, v. 3, n. 2, p. 89–95, 2007. Citado 2 vezes nas páginas 17 e 18.
- PETRUZELLA, F. D. **Programmable Logic Controllers**. 5ª ed.. ed. Nova Iorque, NY: McGraw-Hill Education, 2015. ISBN 978-0073373843. Citado na página 20.
- PROESI. **DIODO ZENER DE 3,3V E 5W**. 2023. <<https://proesi.com.br/diodo-zener-de-33-v-e-5w-1n-5364-do-41.html>>. [Acessado em 28 de novembro de 2023]. Citado na página 35.
- _____. **FUSÍVEL VIDRO 2A 250V**. 2023. <<https://proesi.com.br/fusivel-vidro-fusibras-20agf-5x20mm-2a-rapido.html>>. [Acessado em 28 de novembro de 2023]. Citado na página 36.
- ROBERTS, A. Esp32 for industrial solutions. **Industrial Applications of IoT**, v. 8, n. 4, p. 112–125, 2019. Citado na página 20.
- ROBOTICS, A. **Placa de Desenvolvimento ARM STM32 STM32F103C8T6**. 2023. <<https://www.a2robotics.com.br/placa-de-desenvolvimento-arm-stm32-stm32f103c8t6>>. [Acessado em 28 de novembro de 2023]. Citado na página 25.
- ROBÓTICA, A. **Arduino Mega 2560 R3 Compatível CH340**. 2023. <<https://www.autocorerobotica.com.br/arduino-mega-2560-r3-compativel-ch340>>. [Acessado em 28 de novembro de 2023]. Citado na página 25.
- SILVA, J. O. Industrial programmable logic controllers: Protection and security measures. **International Journal of Industrial Automation**, 2019. Citado 2 vezes nas páginas 22 e 23.
- SMITH, J. Plcs in industrial networks. **Industrial Networking Journal**, v. 12, n. 3, p. 78–85, 2020. Citado na página 21.
- SOUZA, E.; VIEIRA, R. Communication networks in industrial automation. **Industrial Networking Journal**, v. 7, n. 3, p. 112–119, 2015. Citado na página 18.
- SYSTEMS, E. **ESP32 Technical Specifications**. 2020. Accessed: December 2023. Disponível em: <<https://www.espressif.com/en/products/socs/esp32>>. Citado na página 19.
- TERREL, T. The importance of power supply stability in industrial automation. **Industrial Engineering Journal**, v. 12, n. 3, p. 45–58, 2019. Citado na página 18.

USINAINFO. **ESP32 WROOM 32U DevKitC V4 com WiFi e Bluetooth**. 2023. <<https://www.usinainfo.com.br/esp32/esp32-wroom-32u-devkitc-v4-com-wifi-e-bluetooth-7077.html>>. [Acessado em 28 de novembro de 2023]. Citado na página 26.

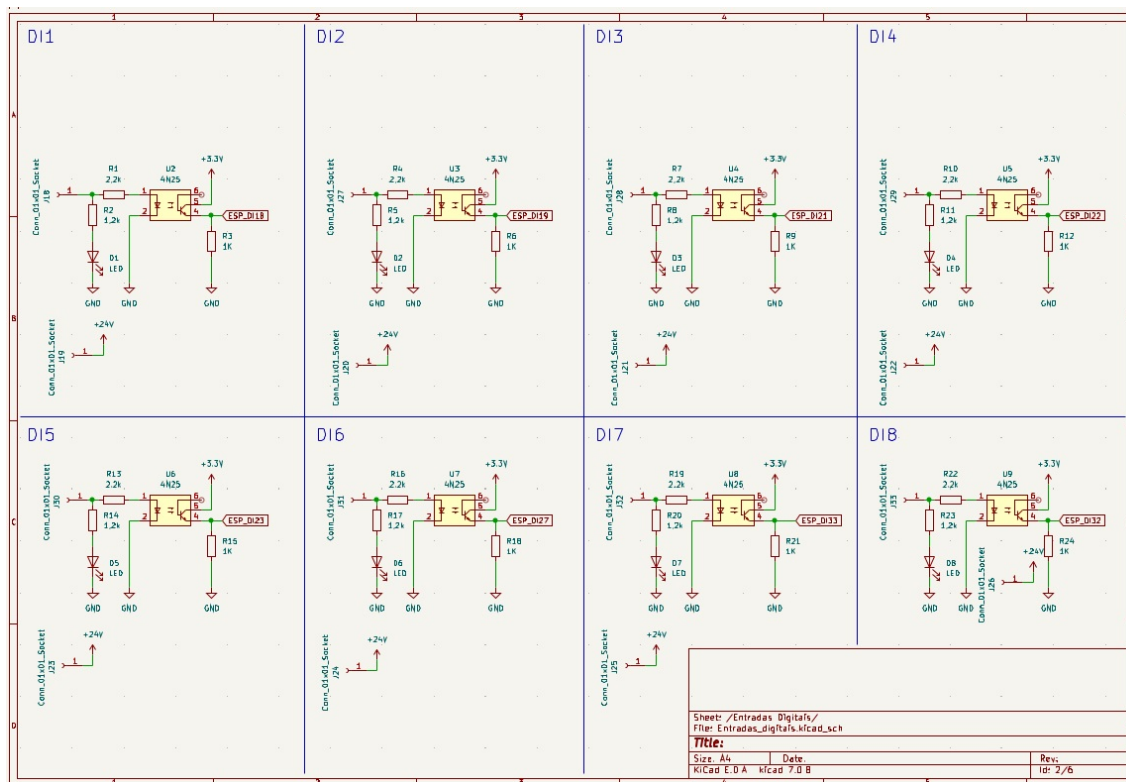
APÊNDICE A – ESQUEMÁTICOS DO PROTÓTIPO

Figura 32 – Esquemático do ESP32.



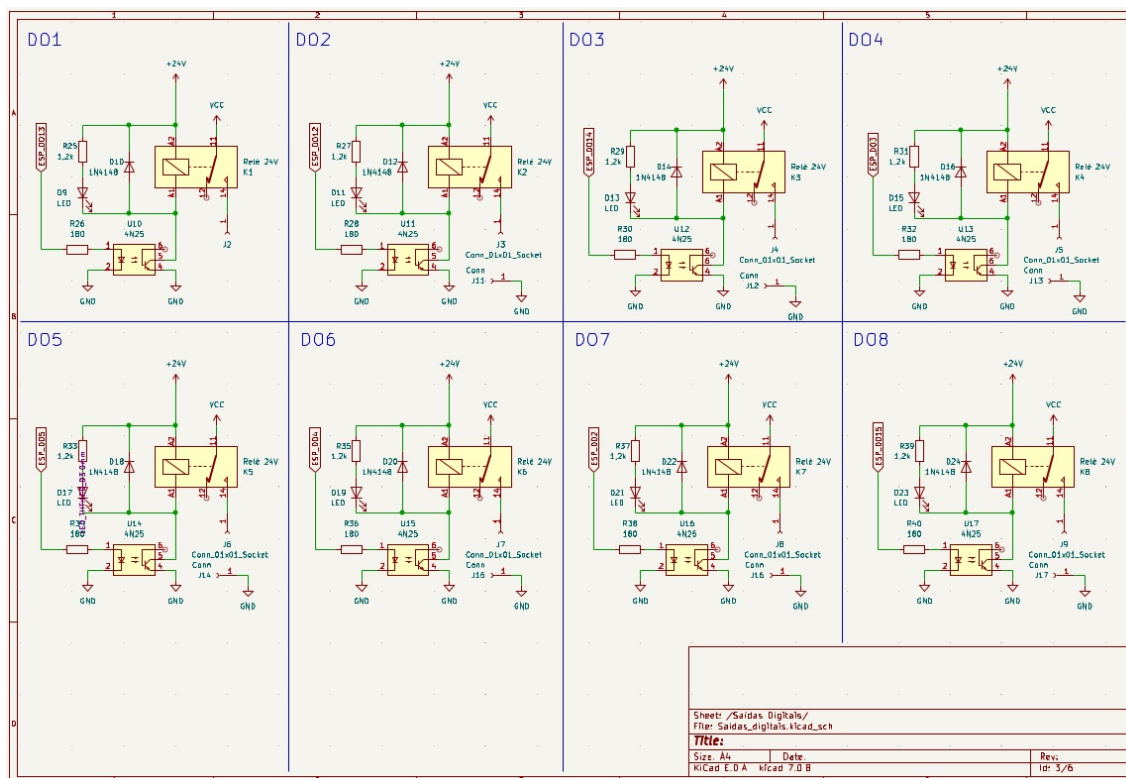
Fonte: Autoria própria.

Figura 33 – Esquemático das entradas digitais.



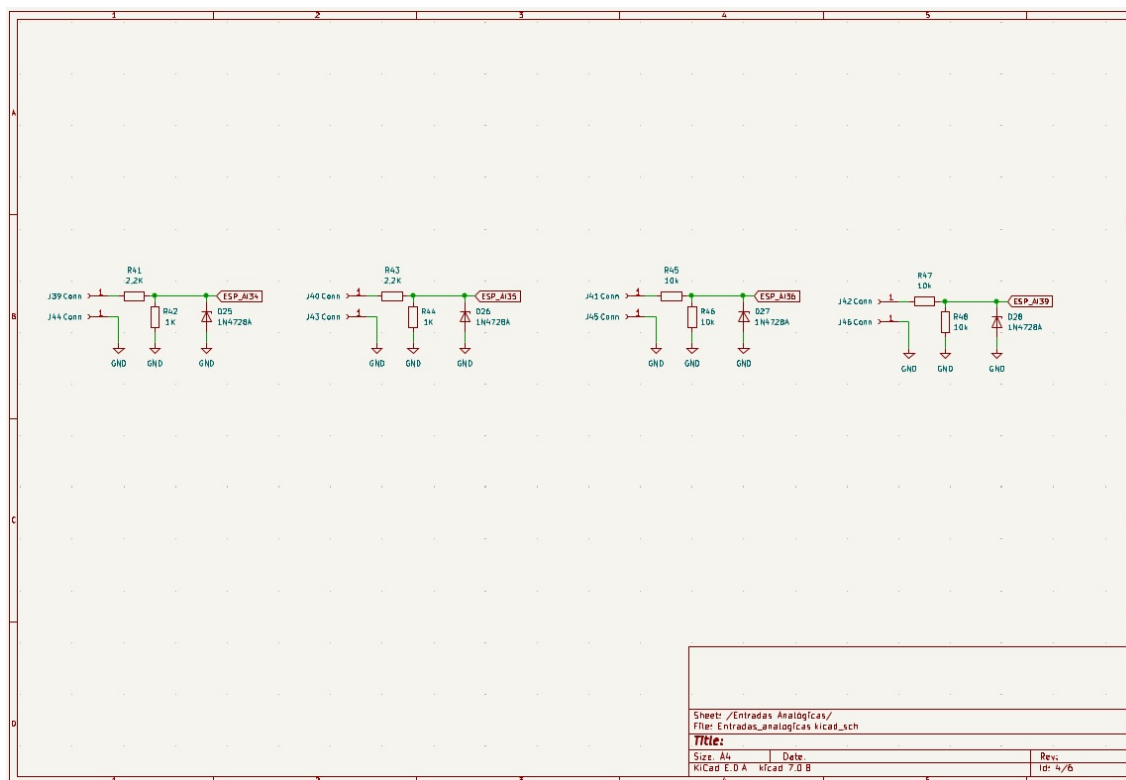
Fonte: Autoria própria.

Figura 34 – Esquemático das saídas digitais.



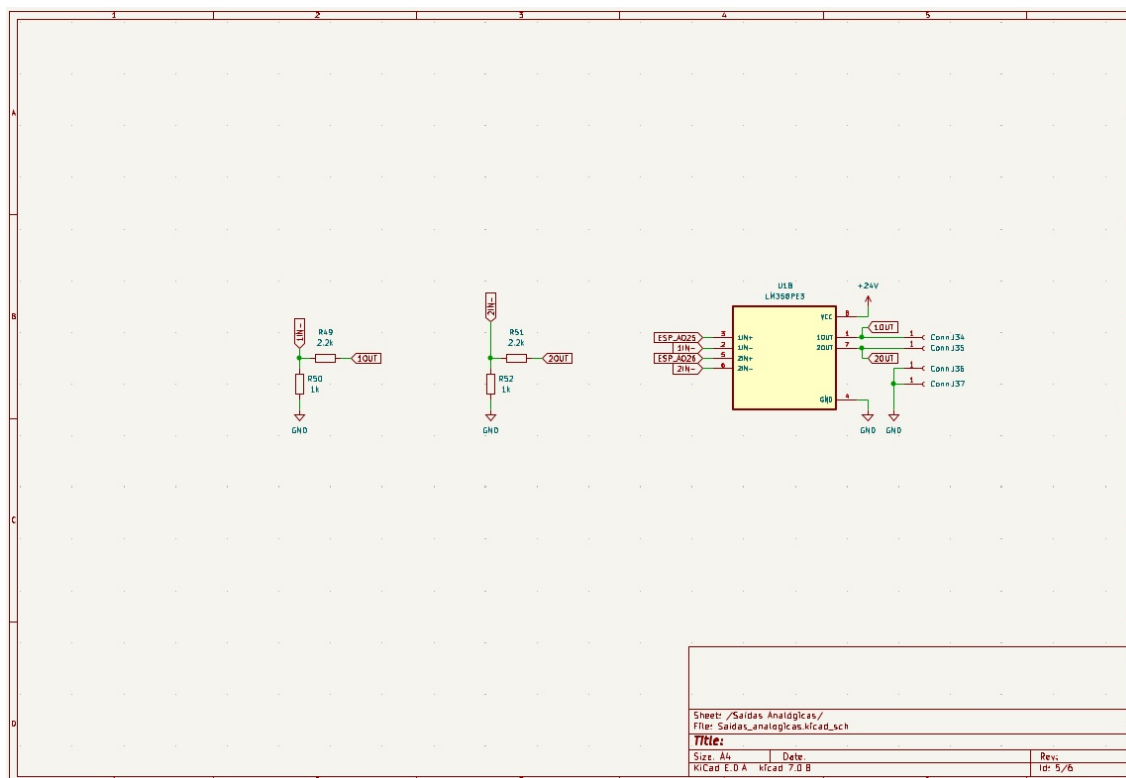
Fonte: Autoria própria.

Figura 35 – Esquemático das entradas analógicas.



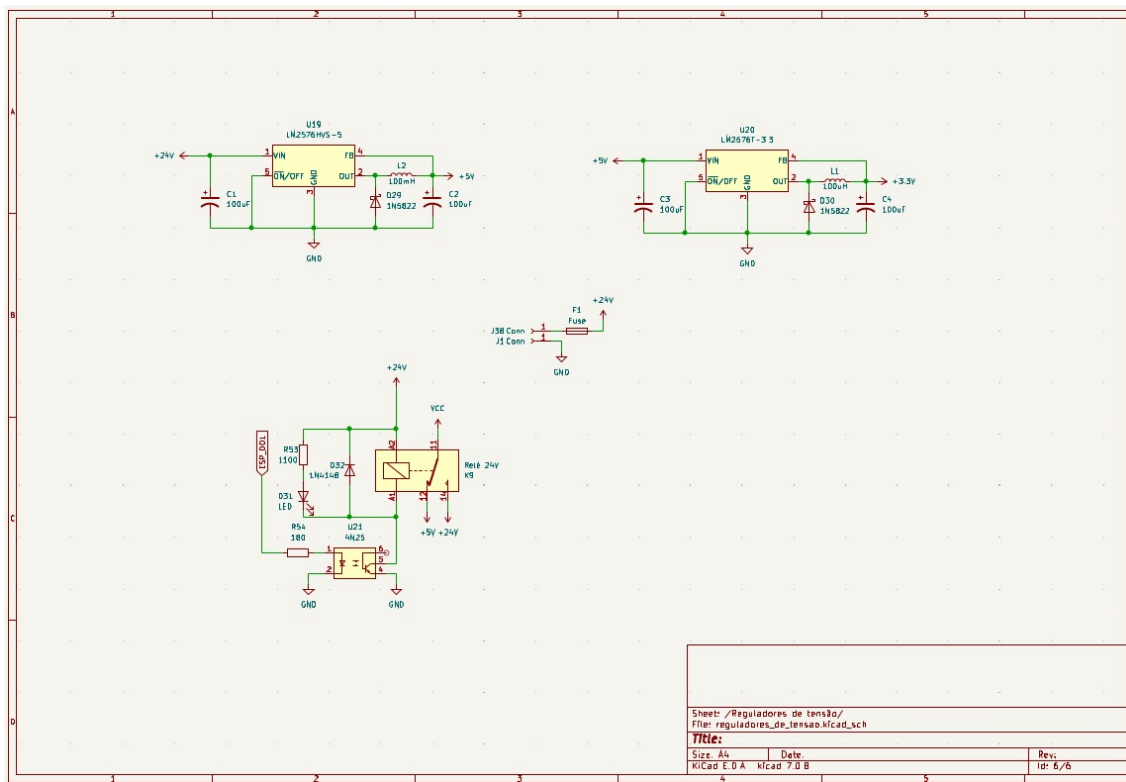
Fonte: Autoria própria.

Figura 36 – Esquemático da saídas analógicas.



Fonte: Autoria própria.

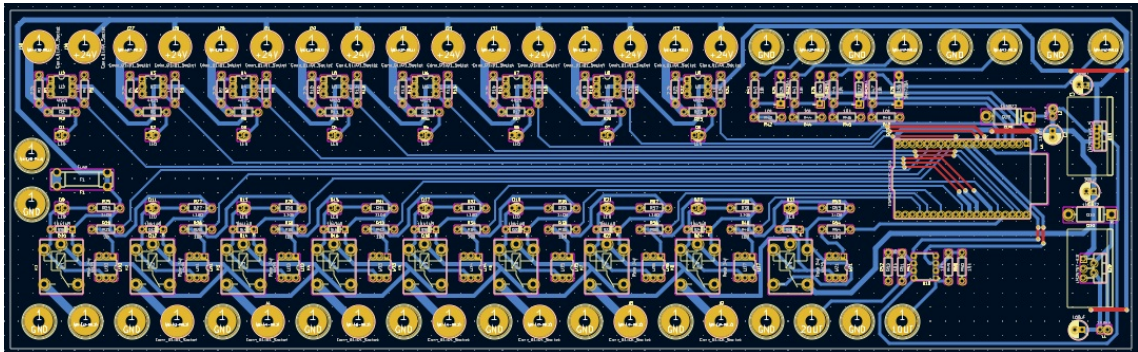
Figura 37 – Esquemático dos reguladores de tensão.



Fonte: Autoria própria.

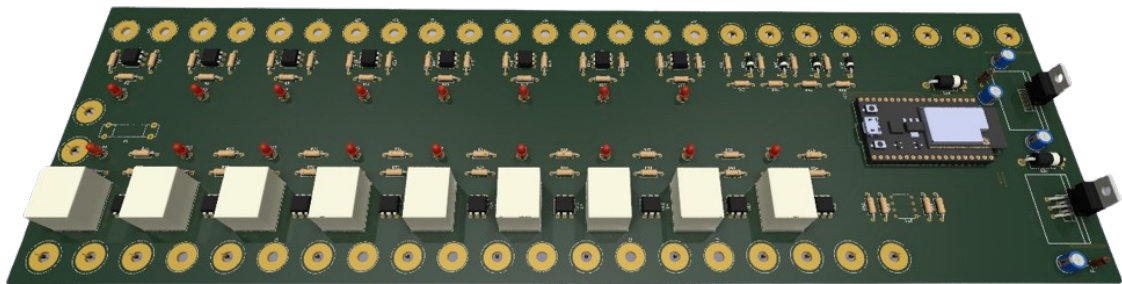
APÊNDICE B – LAYOUT DA PCB

Figura 38 – Layout da PCB.



Fonte: Autoria própria.

Figura 39 – Layout da PCB em 3D.



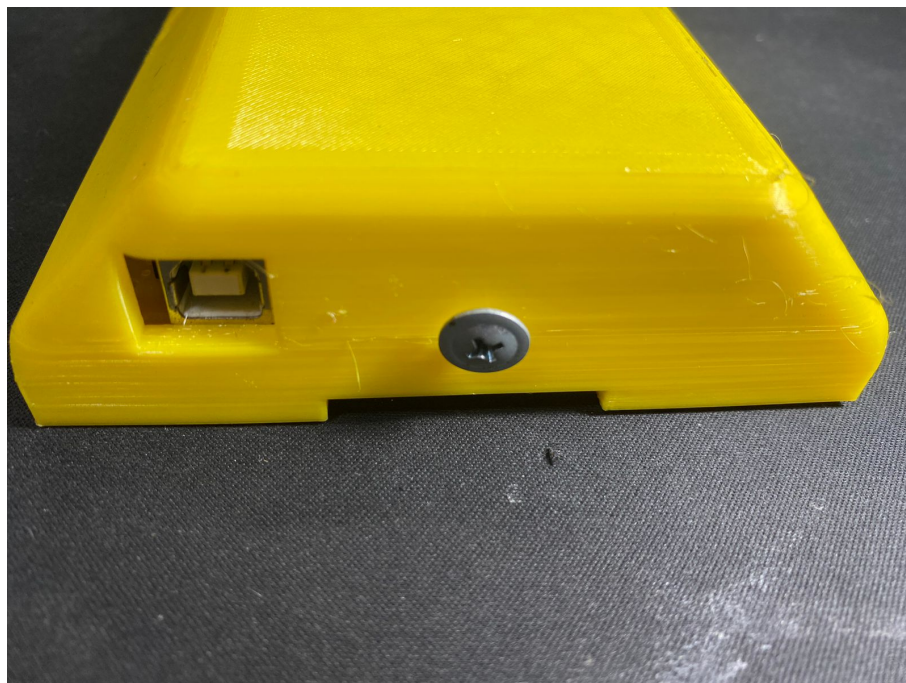
Fonte: Autoria própria.

Figura 40 – Protótipo.



Fonte: Autoria própria.

Figura 41 – Protótipo.



Fonte: Autoria própria.