

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

WILLIAN FELIPE DREHER QUEIROZ

**CONTROLADOR LÓGICO PROGRAMÁVEL ALTERNATIVO DE LIVRE ACESSO
BASEADO EM UM MICROCONTROLADOR PIC**

MEDIANEIRA

2022

WILLIAN FELIPE DREHER QUEIROZ

**CONTROLADOR LÓGICO PROGRAMÁVEL ALTERNATIVO DE LIVRE ACESSO
BASEADO EM UM MICROCONTROLADOR PIC**

**Alternative open access programmable logic controller based on a PIC
microcontroller**

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Alberto Noboru Miyadaira.

MEDIANEIRA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

WILLIAN FELIPE DREHER QUEIROZ

**CONTROLADOR LÓGICO PROGRAMÁVEL ALTERNATIVO DE LIVRE ACESSO
BASEADO EM UM MICROCONTROLADOR PIC**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título
de Bacharel em Engenharia Elétrica da Universidade
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 23/junho/2022

Alberto Noboru Miyadaira
Doutorado
Universidade Tecnológica Federal do Paraná

Leandro Antonio Pasa
Doutorado
Universidade Tecnológica Federal do Paraná

Yuri Ferruzzi
Mestrado
Universidade Tecnológica Federal do Paraná

MEDIANEIRA

2022

RESUMO

O controlador lógico programável (CLP) surgiu no fim da década de 60 para substituir a lógica de relés, trazendo a capacidade de reprogramação da lógica de operação do sistema, ocupando menos espaço e assim foram amplamente aceitos pela indústria. Com a evolução da eletrônica, deixaram de apenas substituir a lógica de relés e agregaram novas funções de controle como temporização, operações aritméticas, aquisição de dados, comunicação com outros dispositivos, entre outras. Mas devido a essa evolução, o controlador lógico programável se tornou um dispositivo caro, o que o torna viável para controle de grandes sistemas de automação, mas para pequenas aplicações, se torna uma alternativa cara e inviável. Neste contexto, este trabalho tem por objetivo desenvolver um controlador lógico programável baseado em microcontroladores PIC, como alternativa de baixo custo, visando sua aplicação em sistemas de automação de pequeno porte. Este dispositivo reúne as principais características esperadas em um controlador lógico programável, como entradas e saídas digitais, entradas e saídas analógicas e interface de comunicação com outros dispositivos. Foram desenvolvidos os circuitos elétricos e as placas de circuito impresso, seguidos da construção do dispositivo e da avaliação do seu desempenho. Este projeto resultou em um controlador lógico programável alternativo dividido em módulos, com um custo de aproximadamente 290 reais para a produção de uma unidade. A partir dos testes de desempenho, comprovou-se a capacidade deste dispositivo de detectar sinais enviados por sensores industriais PNP, acionar cargas de corrente alternada ou contínua, medir e gerar sinais analógicos de tensão com amplitude entre 0 e 10 V, além de se comunicar com outros dispositivos através dos padrões RS-232 e RS-485.

Palavras-chave: controlador lógico programável; microcontrolador; automação.

ABSTRACT

The programmable logic controller appeared in the late 60's to replace the relay logic, bringing the ability to reprogram the system's operating logic, taking up less space and thus were widely accepted by the industry. With the evolution of electronics, They stopped just replacing relay logic and added new control functions such as temporization, arithmetic operations, data acquisition, communication with other devices, amongst others functions. But due to this evolution, the programmable logic controller has become an expensive device, which makes it viable for controlling large automation systems, but for small applications, it becomes an expensive and unfeasible alternative. In this context, this work aims to develop a programmable logic controller based on PIC microcontrollers, as a low-cost alternative, aiming its application in small automation systems. This device brings together the main features expected in a programmable logic controller, such as digital inputs and outputs, analog inputs and outputs and communication interface with other devices. Electrical circuits and printed circuit boards were developed, followed by the construction of the device and the evaluation of its performance. This Project has resulted in an alternative programmable logical controller divided in modules, costing approximately 290 brazilian reais to produce a unit. From the performance tests, it has proven the capacity of the device to detect signals sent by industrial sensors PNP, trigger charges of AC or DC current, measure and generate analog voltage signals with amplitude between 0 and 10 V, in addition to communicating with other devices through RS-232 and RS-485 standards.

Keywords: programmable logic controller; microcontroller; automation.

LISTA DE ILUSTRAÇÕES

Figura 1 - Estrutura de um CLP.....	20
Figura 2 - Ambiente de programação do software LDMICRO	24
Figura 3 - Exemplo de simulação de um programa no software LDMICRO	25
Figura 4 - Pinagem do modelo PIC16F877A.....	28
Figura 5 - Arquitetura interna do modelo PIC16F877A	28
Figura 6 - Transmissão assíncrona do caractere "A"	29
Figura 7 - Pinagem do conector DB9 (fêmea).....	34
Figura 8 - Conexão null-modem	34
Figura 9 - Topologia de rede em barramento	36
Figura 10 - Conexão Full-Duplex.....	36
Figura 11 - Conexão Half-Duplex.....	36
Figura 12 - Comprimento da linha x Taxa de transmissão de dados (RS-485).....	38
Figura 13 - Conversores buck com ajuste de tensão	39
Figura 14 - Conversores buck com ajuste de tensão	40
Figura 15 - Circuito de alimentação.....	41
Figura 16 - Circuito oscilador do microcontrolador	42
Figura 17 - Circuito de <i>reset</i> do microcontrolador	42
Figura 18 - Conector de gravação do microcontrolador	43
Figura 19 - Circuito conectado ao pino MCLR do microcontrolador	44
Figura 20 - Circuito de entrada digital	45
Figura 21 - Entradas das portas lógicas não utilizadas	46
Figura 22 - Circuito de saída digital.....	47
Figura 23 - Circuito de entrada analógica	48
Figura 24 - Circuito de saída analógica.....	52
Figura 25 - Módulo conversor RS-485	53
Figura 26 - Módulo conversor RS-232	54
Figura 27 - Circuito de seleção com portas lógicas.....	55
Figura 28 - Alternativa com transistor a uma porta lógica NOT	56
Figura 29 - Alternativa com transistores a uma porta lógica OR	56
Figura 30 - Circuito de seleção do módulo de comunicação.....	57
Figura 31 - Conexão do módulo de comunicação RS-485.....	57
Figura 32 - Conexão do módulo de comunicação RS-232.....	58
Figura 33 - Circuito de comunicação RS-232 e RS-485.....	58
Figura 34 - <i>Layout</i> do módulo principal	61
Figura 35 - <i>Layout</i> do módulo de alimentação	61
Figura 36 - <i>Layout</i> do módulo de entradas digitais.....	62
Figura 37 - <i>Layout</i> do módulo de saídas digitais	63
Figura 38 - <i>Layout</i> do módulo de comunicação.....	63

Figura 39 - <i>Layout</i> do dispositivo.....	64
Figura 40 - Controlador lógico programável fabricado	65
Figura 41 - Teste sem obstáculo à frente do sensor óptico.....	66
Figura 42 - Teste com obstáculo à frente do sensor óptico.....	66
Figura 43 - Teste de continuidade do contato NA	67
Figura 44 - Teste de leitura de um sinal analógico	68
Figura 45 - Plotagem de um sinal analógico no <i>software</i> SerialPlot.....	68
Figura 46 - Medição de um sinal analógico em um osciloscópio.....	69
Figura 47 - Leitura analógica com e sem filtro de média móvel	69
Figura 48 - Sinal na saída analógica e sinal PWM do microcontrolador	71
Figura 49 - Conexão do teste RS-232.....	72
Figura 50 - Conexão do teste RS-485.....	73
Figura 51 - Testes de comunicação	73
Figura 52 - Caracteres recebidos, pelo computador, via comunicação serial	74
Quadro 1 - Relação de pinos do gravador PICkit 3	43
Quadro 2 - Funções dos pinos do microcontrolador neste dispositivo	59

LISTA DE TABELAS

Tabela 1 - Resultados do teste de precisão da leitura analógica	70
Tabela 2 - Resultados do teste de precisão da saída analógica	71

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Delimitação do tema.....	12
1.2	Problema de pesquisa e hipótese.....	12
1.3	Justificativa.....	12
2	OBJETIVOS	14
2.1	Objetivo geral.....	14
2.2	Objetivos específicos.....	14
3	CONTROLADORES LÓGICOS PROGRAMÁVEIS MICROCONTROLADOS DE BAIXO CUSTO	15
3.1	Controlador lógico programável	18
3.2	Linguagem Ladder	22
3.2.1	Software LDMICRO	23
3.3	Microcontroladores	26
3.3.1	Microcontrolador PIC16F877A	27
3.4	Redes de comunicação.....	29
3.4.1	Protocolos de comunicação	30
3.4.1.1	<u>Protocolo RS-232</u>	<u>32</u>
3.4.1.2	<u>Protocolo RS-485.....</u>	<u>35</u>
4	MATERIAIS E MÉTODOS	39
4.1	Alimentação	40
4.2	Microcontrolador	41
4.3	Entradas digitais.....	44
4.4	Saídas Digitais	46
4.5	Entradas analógicas.....	48
4.6	Saídas analógicas	49
4.7	Comunicação	52
4.8	Configuração do microcontrolador	58
4.9	Projeto das placas de circuito impresso	60
4.10	Construção do dispositivo	64
5	RESULTADOS E DISCUSSÕES	65
5.1	Controlador lógico programável	65
5.2	Detecção de sinais nas entradas digitais.....	66
5.3	Acionamento das saídas digitais	67

5.4	Leitura de sinais analógicos.....	67
5.5	Geração de sinais analógicos	70
5.6	Comunicação com outros dispositivos.....	72
6	CONSIDERAÇÕES FINAIS	75
	REFERÊNCIAS.....	76
	APÊNDICE A - Lista de componentes	81
	APÊNDICE B - Esquemático do circuito elétrico.....	84

1 INTRODUÇÃO

Os sistemas de automação da década de 60 utilizavam amplamente a lógica a relé, onde os sistemas mais complexos requeriam grandes painéis de relés interconectados por uma grande quantidade de fios. Sempre que a lógica precisava ser modificada, os painéis eram alterados ou substituídos, o que interrompia a produção por bastante tempo e gerava custos. Como solução a este problema, em 1968, a General Motors requisitou o desenvolvimento de um sistema flexível, que pudesse ser reprogramado quando necessário e que fosse resistente às condições adversas existentes no ambiente industrial, como poeira, impacto, vibração, etc. Este projeto foi liderado por Richard E. Morley e resultou no primeiro controlador lógico programável da história. Esta tecnologia foi amplamente aceita pela indústria e se espalhou em pouco tempo. (FRANCHI e CAMARGO, 2008; MORAES e CASTRUCCI, 2010; PAREDE e GOMES, 2011).

Ao longo do tempo, surgiram novas tecnologias e o controlador lógico programável, além substituir a lógica de relés, foi agregando novas funções como temporização, aquisição e manipulação de dados, operações aritméticas, controle PID, etc. A evolução da eletrônica, com a criação dos circuitos integrados, permitiu a redução de tamanho dos dispositivos, maior precisão, confiabilidade e velocidade de operação (FRANCHI e CAMARGO, 2008; PAREDE e GOMES, 2011).

Após décadas de evolução, um controlador lógico programável (CLP) pode ser considerado um computador de uso industrial que realiza funções de controle conforme a sua programação. Atualmente, está presente na maioria dos sistemas de automação industrial, principalmente por conta de seu tamanho reduzido, facilidade de alterar o comportamento do sistema, alta confiabilidade, capacidade de coletar dados, executar funções complexas de controle e comunicar se com outros dispositivos (FRANCHI e CAMARGO, 2008; PETRUZELLA, 2014; SILVEIRA e SANTOS, 2010).

1.1 Delimitação do tema

A partir do assunto de controladores lógicos programáveis, o tema deste trabalho será delimitado nos controladores lógicos programáveis microcontrolados de baixo custo.

1.2 Problema de pesquisa e hipótese

O CLP se tornou um dispositivo complexo e repleto de funções, porém, uma das consequências desta evolução é o alto valor de mercado dos CLPs industriais. Por conta disso, é uma ótima alternativa de controle para grandes sistemas de automação, porém, para simples aplicações de automação, onde não há necessidade de funções tão complexas, se torna uma alternativa cara e inviável. Este alto custo limita o acesso a esta tecnologia e, em consequência, limita o aprendizado na área. Então questiona-se: É possível obter um modelo de controlador lógico programável com custo acessível e capaz de controlar sistemas de automação de pequeno porte?

Uma hipótese seria desenvolver um controlador lógico programável com uma arquitetura mais simples, baseada em um microcontrolador, mantendo as principais características de um CLP.

1.3 Justificativa

O alto custo dos controladores lógicos programáveis limita a quantidade de dispositivos disponíveis nas aulas práticas nas universidades, deste modo, vários alunos precisam dividir um mesmo CLP, o que dificulta o aprendizado. Então este projeto visa oferecer uma alternativa de menor custo que permita aumentar a quantidade de controladores disponíveis, resultando em uma melhor distribuição dos dispositivos entre os alunos. Este dispositivo poderá ser produzido na própria universidade ou até mesmo pelos próprios alunos. Ao fabricarem os dispositivos, os alunos terão oportunidade de aprender mais sobre a arquitetura e o funcionamento interno de um CLP, e aplicarão conceitos de eletrônica, sistemas embarcados,

confeção de placas de circuito impresso, soldagem e montagem de placas eletrônicas.

Apesar de a ideia ter surgido para suprir a necessidade das universidades, este dispositivo não se limitará a aplicações didáticas. A intenção é que qualquer indivíduo com conhecimento suficiente, possa fabricar o seu próprio CLP, para utilizá-lo nas mais diversas aplicações de pequeno porte, seja para aprendizado, automações residenciais ou industriais. Portanto, a lista de componentes utilizados e o projeto do circuito elétrico serão disponibilizados no apêndice, permitindo a fabricação ou até a modificação deste dispositivo conforme a necessidade, por qualquer indivíduo que tenha conhecimento suficiente para tal.

Deste modo, este projeto busca facilitar o acesso a esta tecnologia, fornecendo uma alternativa de baixo custo a quem busca um controlador lógico programável para o controle de pequenos projetos de automação ou para auxiliar no aprendizado de automação e controladores lógico programáveis.

2 OBJETIVOS

Nesta seção, o objetivo geral e os objetivos específicos deste projeto de pesquisa serão apresentados.

2.1 Objetivo geral

O objetivo geral deste trabalho é desenvolver um controlador lógico programável de baixo custo com um microcontrolador PIC, visando a automação de projetos de pequeno porte.

2.2 Objetivos específicos

Estes são os objetivos específicos que devem ser alcançados ao longo do desenvolvimento da pesquisa:

- Identificar os principais recursos presentes nos controladores lógicos programáveis;
- Projetar o circuito elétrico;
- Projetar o layout da placa de circuito impresso;
- Construir um modelo funcional de controlador lógico programável;
- Avaliar o desempenho do dispositivo na automação de sistemas.

3 CONTROLADORES LÓGICOS PROGRAMÁVEIS MICROCONTROLADOS DE BAIXO CUSTO

Este capítulo tem por objetivo descrever os principais trabalhos relacionados a controladores lógicos programáveis microcontrolados de baixo custo, destacando suas principais características e recursos, e, em seguida, apresentar os principais conceitos envolvidos neste trabalho.

Os trabalhos de Eleutério, Hovadich e Braga (2011), Rohde, Kieckow e Machado (2011), Micaloski (2012), Suresh *et al.* (2020), Sanver *et al.* (2018), EG Projects (2019) e o livro de Uzam (2021) apresentam o desenvolvimento de controladores lógicos programáveis de baixo custo baseados em vários modelos de microcontroladores PIC. Estes controladores lógicos programáveis foram construídos com arquiteturas semelhantes, basicamente compostas por um microcontrolador PIC, interface de entrada/saída digital e interface de gravação do microcontrolador. Além destes recursos, alguns destes controladores ainda contém entradas e saídas analógicas e interface de comunicação serial.

O projeto de EG Projects (2019) além de desenvolver um circuito simples de controlador lógico programável, ainda adiciona um *display* LCD que informa o estado das entradas e saídas. A lógica de programação em Ladder é elaborada no *software* LDmicro.

O artigo de Suresh *et al.* (2020) se destaca por apresentar um controlador lógico programável com 16 portas digitais que podem ser configuradas como entrada ou saída dependendo da necessidade do usuário. A programação pode ser feita tanto em Ladder, utilizando o *software* LDmicro, como também em linguagem C, utilizando o *software* MPLAB.

O livro de Uzam (2021) apresenta módulos de entrada/saída digitais, com oito entradas e oito saídas a relé, conectados ao microcontrolador através de comunicação serial, o que possibilita a adição de até quatro módulos de entrada/saída em série. Ainda suporta adição de até quatro módulos de entrada analógica e um módulo de saída analógica, além de módulos de filtro passa-baixa, de tensão de referência de 5 VCC e de regulador de tensão. A programação é realizada através de macros em linguagem Assembly, criadas pelo autor do projeto.

Já o trabalho de Fernando (2017) apresenta o desenvolvimento de um editor Ladder para navegador *web*, interligado através de um servidor *web* a uma máquina

virtual que pode ser utilizada em diferentes plataformas de *hardware* que possuam suporte a comunicação Ethernet. A continuidade deste trabalho foi realizada por Zanzoti (2019), onde foi desenvolvido um controlador lógico programável de baixo custo capaz de executar a máquina virtual desenvolvida por Fernando (2017). O microcontrolador utilizado foi o modelo STM32L433CCT6. O *hardware* é alimentado em 35 VCC, contém quatro entradas digitais e quatro saídas digitais, sendo duas saídas a transistor e duas saídas a relé, ainda possui duas entradas analógicas e duas saídas analógicas. Quanto à interface de comunicação, possui um controlador Ethernet, comunicação por RS-485, possibilidade de conexão via I2C e uma porta USB destinada à gravação do *firmware* no microcontrolador. O desenvolvimento deste trabalho foi apresentado de forma clara e bem detalhada.

O artigo de Gevaerd, Silva e Bonacorso (2010) desenvolveu um controlador lógico programável de baixo custo visando projetos de automação industrial. O circuito é baseado no microcontrolador AT89S8253, alimentado a uma tensão de 24 VCC, permite a conexão de até oito módulos de entrada/saída digital, com quatro entradas e quatro saídas, totalizando 32 entradas e 32 saídas digitais possíveis. Possui um circuito de amplificação de corrente com oito saídas e uma interface de comunicação serial RS-232. Uma interface SPI é utilizada para gravação do código no microcontrolador.

O projeto de Kay *et al.* (2012) apresenta o desenvolvimento de um controlador lógico programável de baixo custo, visando aplicações simples de automação. Uma tensão de 15 VCC alimenta o circuito, o modelo C8051F320 foi o microcontrolador escolhido, conta com seis entradas digitais, quatro saídas digitais, quatro entradas analógicas, três saídas analógicas, memória EEPROM, RTC, interface de comunicação USB e RS-485. A programação em C foi feita através do *software* Proview32. A gravação do *firmware* no microcontrolador foi feita através da porta USB. Este projeto resultou em um CLP bem completo mantendo o baixo custo.

O trabalho de Alves *et al.* (2014) apresenta o desenvolvimento de um controlador lógico programável de baixo custo de código aberto. Utiliza um microcontrolador AVR ATmega2560, possui oito entradas digitais em cada módulo de entrada, oito saídas a relé em cada módulo de saída, a comunicação entre os módulos segue o protocolo RS-485 e um protocolo criado pelos autores, chamado de OPLC Protocol. Uma porta USB é utilizada para gravação de programas no microcontrolador e para comunicação com o PC. Possui suporte a comunicação com

outros dispositivos através do protocolo MODBUS-TCP. O editor Ladder utilizado neste CLP é uma versão modificada do editor LDmicro, chamada de OpenPLC Ladder Editor. O resultado deste projeto foi um CLP bem completo e robusto, capaz de substituir facilmente um CLP presente no mercado, mas com um menor custo e com as vantagens de ter seu código aberto.

O artigo de Neves e Freitas (2019) consiste no desenvolvimento de um controlador lógico programável de baixo custo, visando aplicações didáticas. O microcontrolador utilizado como CPU foi o ATMEGA328P incluído na placa do Arduino Nano. O CLP desenvolvido opera a uma tensão de 24 VCC, possui quatro entradas digitais de 24 VCC, quatro saídas a relé, duas entradas analógicas de 4 a 20 mA, e duas saídas analógicas de 4 a 20 mA. O software Ladder Maker foi utilizado para a programação em Ladder deste CLP. Não apresenta interface de comunicação com outros dispositivos.

Existem diferentes abordagens dentre os trabalhos supracitados, onde alguns autores desenvolveram plataformas simples e baratas de ensino, enquanto outros, elaboraram dispositivos mais complexos e robustos, mas mantendo o baixo custo, visando competir com controladores existentes no mercado.

Apesar das diferentes abordagens, notam-se algumas semelhanças nos projetos, como o emprego de opto acoplador na isolação elétrica das entradas digitais e uso de saídas digitais a relé. No geral, as tensões de alimentação utilizadas foram de 12 VCC ou 24 VCC. As linguagens de programação mais empregadas foram as linguagens C e Ladder. Dentre os que se basearam em linguagem Ladder, nota-se a predominância do uso do software LDmicro ou de uma versão modificada dele. As interfaces de comunicação serial RS-232 e RS-485 foram as mais presentes, mas nem todos os trabalhos apresentaram interface de comunicação.

O presente trabalho propõe o desenvolvimento de um CLP de baixo custo utilizando um microcontrolador PIC, cujas características sejam suficientes para a aplicação em projetos de automação de pequeno porte. O CLP reunirá os principais recursos requeridos neste tipo de dispositivo, apresentando interfaces de entradas e saídas digitais optoacopladas, entradas e saídas analógicas, interface de comunicação serial e programação em Ladder. Deve ser uma alternativa interessante para redução de custos em pequenos projetos de automação, mantendo as principais características esperadas em um CLP.

Nas próximas seções, serão apresentados os principais conceitos incluídos no embasamento teórico deste trabalho.

3.1 Controlador lógico programável

Todo sistema de automação industrial requer algum tipo de controlador para gerenciar e comandar o processo, desde os mais simples, como controlar a velocidade de um motor, até os mais complexos, como controlar toda uma planta industrial. Além do controlador, outros dispositivos que são essenciais para os sistemas de automação, são os sensores e atuadores, que relacionam as grandezas físicas com as grandezas elétricas do processo. Os sensores convertem alguma grandeza física em sinais elétricos. Alguns exemplos de sensores são: botões, sensores de presença, de temperatura, etc. Já os atuadores, são equipamentos que convertem os sinais elétricos em ações físicas, como motores, válvulas, aquecedores, entre outros (FRANCHI e CAMARGO, 2008; SILVEIRA e SANTOS, 2010).

Dentre as várias opções de controladores, a mais viável atualmente, para a maioria dos sistemas de controle, é o controlador lógico programável, principalmente por conta de seu tamanho reduzido, facilidade de alterar o comportamento do sistema, alta confiabilidade, possibilidade de coletar dados e comunicar se com outros dispositivos (FRANCHI e CAMARGO, 2008; SILVEIRA e SANTOS, 2010).

Segundo Petruzella (2014), um controlador lógico programável (CLP) pode ser considerado um computador de uso industrial que realiza funções de controle conforme a sua programação, que surgiu para substituir os circuitos de controle via relé. É distinto de um computador pessoal, já que foi desenvolvido para realizar tarefas específicas em indústrias e então possui interfaces de entrada/saída, maior resistência a altas temperaturas, impacto e vibração, imunidade a ruídos elétricos e descargas eletrostáticas. As suas funções de controle não se limitam a monitorar as entradas e modificar as saídas, também possui funções de temporização e de contagem de eventos, capacidade de armazenar, manipular e processar dados, entre outras.

A definição de um controlador lógico programável, segundo a norma IEC 61131-1, é:

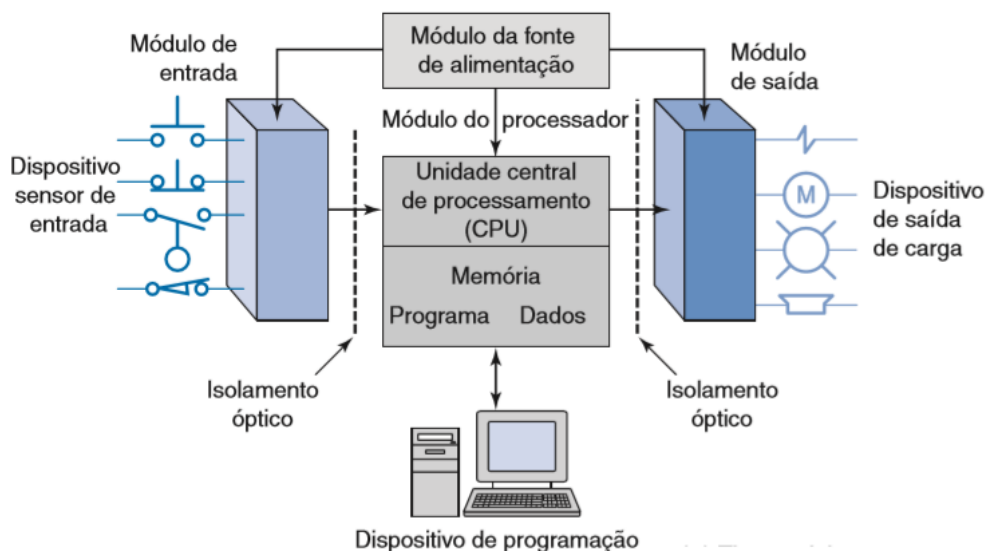
Sistema eletrônico operado digitalmente, projetado para uso em um ambiente industrial, que utiliza uma memória programável para o armazenamento interno das instruções orientadas pelo usuário para implementação de funções específicas como lógica, sequenciamento, temporização, contagem e aritmética, para controlar, por meio de entradas e saídas digitais ou analógicas, vários tipos de máquinas e processos. Tanto o CLP como seus periféricos associados são projetados para que possam ser facilmente integrados a um sistema de controle industrial e facilmente utilizados em todas as suas funções pretendidas. (INTERNATIONAL ELECTROTECHNICAL COMMISSION, 2003, p. 7, tradução nossa).

As vantagens do CLP em relação aos outros sistemas de controle, conforme Petruzella (2014), são:

- A maior parte da fiação externa não é mais necessária, pois a lógica de operação está na sua memória, o que elimina as chances de erro lógico na fiação, tornando o processo mais confiável;
- Como está armazenada na memória, a lógica pode ser facilmente transferida para outros CLPs;
- Já que as relações entre entradas e saídas são feitas pelo programa, a modificação da lógica se torna muito mais fácil do que modificar a fiação de um circuito;
- Devido a sua capacidade de substituir vários relés e reduzir a fiação, o CLP pode ser considerado uma alternativa de menor custo para certas aplicações;
- Possui capacidade de se comunicar com outros equipamentos, podendo realizar monitoramento, supervisão e coleta de dados;
- Tem um bom tempo de resposta, atualizando as variáveis do sistema em tempo real;
- O usuário é capaz de detectar defeitos de forma eficaz, utilizando as funções de diagnóstico do CLP.

De acordo com Petruzella (2014), um CLP é composto por quatro partes: fonte de alimentação, unidade central de processamento (CPU), dispositivos de entrada e saída, e o dispositivo de programação. A estrutura de um CLP pode ser visualizada na Figura 1.

Figura 1 - Estrutura de um CLP



Fonte: PETRUZELLA, 2013, p. 4.

A fonte de alimentação tem a função de fornecer energia suficiente para alimentar os componentes do CLP. Lembrando que em alguns casos, é necessário adicionar uma segunda fonte, por conta da expansão dos módulos. Dependendo do modelo do CLP, pode ser alimentado com uma tensão de 220 V ou com tensão contínua de 24 V (FRANCHI e CAMARGO, 2008).

A unidade central de processamento é o “cérebro” do CLP, já que é responsável por gerenciar o processo e pela execução do programa desenvolvido pelo usuário em um processo repetitivo conhecido como varredura ou *scan*. Onde recebe os sinais de entrada e comandos de comunicação, executa as operações lógicas, aritméticas e de controle, e então atualiza o estado das saídas (FRANCHI e CAMARGO, 2008; PETRUZELLA, 2014).

Os dispositivos de entrada e saída são responsáveis por conectar os elementos do mundo exterior ao controlador, e condicionam os sinais recebidos ou enviados. Também são conhecidos como módulos I/O, referindo-se ao termo da língua inglesa “*Input/Output*”, que significa “Entrada/Saída”. Nos terminais de entrada, são conectados dispositivos de entrada, como botoeiras, chaves e sensores, e nos terminais de saída, são conectados os dispositivos de saída, como válvulas, contadores e lâmpadas. Para garantir o isolamento elétrico entre o mundo exterior e os componentes internos do controlador, a maioria dos CLPs utilizam acopladores ópticos, que acoplam circuitos por meio da luz (FRANCHI e CAMARGO, 2008; PETRUZELLA, 2014).

As interfaces de entrada digital em corrente contínua definem se operam em modo de fonte (*source*) ou dreno (*sink*) de corrente. Os dispositivos de entrada também são especificados como tipo dreno (NPN) ou fonte (PNP) de corrente. Um dispositivo do tipo dreno irá receber a corrente, e um dispositivo do tipo fonte irá fornecer a corrente. Para correta operação, os dispositivos de entrada do tipo dreno devem ser conectados a interfaces de entrada do tipo fonte e os dispositivos de entrada do tipo fonte devem ser conectados a interfaces de entrada do tipo dreno (PETRUZELLA, 2014).

O dispositivo de programação é responsável por inserir o programa na memória do CLP. Existem dispositivos de programação portáteis, que são normalmente utilizados em CLPs de menor porte, por conta de seu custo reduzido. A maioria dos fabricantes fornece um *software* que permite que a elaboração e edição dos programas sejam realizadas em um computador. Assim que o programa estiver finalizado ou corrigido, o dispositivo de programação insere o programa na memória do CLP (FRANCHI e CAMARGO, 2008; PETRUZELLA, 2014).

A maioria dos CLPs possuem sistemas de comunicação, que permitem a comunicação do CLP com outros dispositivos presentes na rede de comunicação. Também podem ser utilizados para inserir o programa na memória do CLP (FRANCHI e CAMARGO, 2008).

O programa contém as instruções que o CLP deve seguir para realizar as ações e atualizar as saídas. É desenvolvido pelo usuário, geralmente na linguagem Ladder, embora existam outras linguagens, tais como diagrama de blocos de função, mapa de função sequencial, lista de instruções e texto estruturado (PETRUZELLA, 2014).

Visto a importância do CLP em processos industriais, o presente trabalho propõe o desenvolvimento de um controlador lógico programável de baixo custo, que permita o controle de sistemas de automação de pequeno porte. Por ser a linguagem de programação mais comum, a programação deste CLP será feita por diagrama Ladder, utilizando o *software* LDmicro que implementa esta linguagem em microcontroladores.

3.2 Linguagem Ladder

Segundo Petruzella (2014), existem cinco linguagens de programação definidas pela norma IEC 61131, são elas: Diagrama Ladder (LD), Diagrama de Blocos de Função (FBD), Mapa de Função Sequencial (SFC), Lista de Instruções (IL) e Texto Estruturado (ST). Destas linguagens, a mais utilizada é a linguagem Ladder, que é projetada de forma similar à lógica de relés.

Ladder é uma linguagem gráfica inspirada na lógica de relés e nos diagramas de contatos elétricos utilizados nos comandos de acionamento de máquinas. Foi desenvolvida para que engenheiros e técnicos, apenas com conhecimento em lógica de relés, pudessem programar o CLP. Por conta disso, e por ser a primeira linguagem utilizada nos CLPs, é a mais popular e pode ser encontrada na maioria dos CLPs atuais (FRANCHI e CAMARGO, 2008; PAREDE e GOMES, 2011).

De acordo com Petruzella (2014), a linguagem Ladder é um conjunto de instruções representadas de forma gráfica, que é utilizado para desenvolver o programa. Os símbolos básicos deste conjunto de instruções são contatos e bobinas, que são organizados de tal modo que resulta na lógica de controle.

As variáveis que representam os elementos de entrada, saída, memória, temporizadores e contadores, são denominadas operandos. Estes operandos são utilizados pelo programa para executar as operações lógicas e aritméticas. Cada linha possui um conjunto de instruções, que são processadas e o seu resultado é atribuído a um bloco de saída, presente no final da linha, à direita (PAREDE e GOMES, 2011).

Segundo Parede e Gomes (2011), os operandos podem ser divididos em três categorias:

- **Entradas (I):** Variáveis que representam os módulos de entrada e que podem ser lidos pelo programa, mas só podem ser alteradas pelos módulos de entrada;
- **Saídas (Q):** Variáveis que representam os módulos de saída, e podem ser lidos e escritos pelo programa;
- **Memória (M):** Variáveis que armazenam dados e resultados parciais, e podem ser lidos e escritos pelo programa;

De acordo com Moraes e Castrucci (2010), as principais instruções da lógica de relés operam da seguinte forma:

- **Instrução Normalmente Aberto (NA):** Caso o *bit* esteja em nível lógico baixo, a instrução retorna valor falso, e se o *bit* estiver em nível lógico alto, a instrução retorna valor verdadeiro;
- **Instrução Normalmente Fechado (NF):** Quando o *bit* estiver em nível lógico baixo, a instrução retorna valor verdadeiro, e caso o *bit* esteja em nível lógico alto, a instrução retorna valor falso;
- **Instrução de bobina energizada (OTE):** Se houver continuidade lógica na linha, o *bit* será colocado em nível lógico alto, e caso contrário, será colocado em nível lógico baixo;
- **Instrução One Shot Rising (OSR):** Esta instrução será verdadeira apenas quando houver transição de condição falsa para verdadeira na linha e permanece verdadeira durante apenas um ciclo de varredura;
- **Instrução de bobina energizada com retenção (OTL):** Esta instrução coloca a saída em nível lógico alto, sendo que esta saída só será desabilitada caso uma instrução OTU (direcionada ao mesmo *bit*) seja acionada;
- **Instrução de desabilitar bobina com retenção (OTU):** Se a saída estiver em nível lógico alto, esta instrução desabilita a saída habilitada pela instrução OTL.

Ainda existem as instruções de temporização, contagem e operações aritméticas, mas elas não serão detalhadas neste tópico.

3.2.1 Software LDMICRO

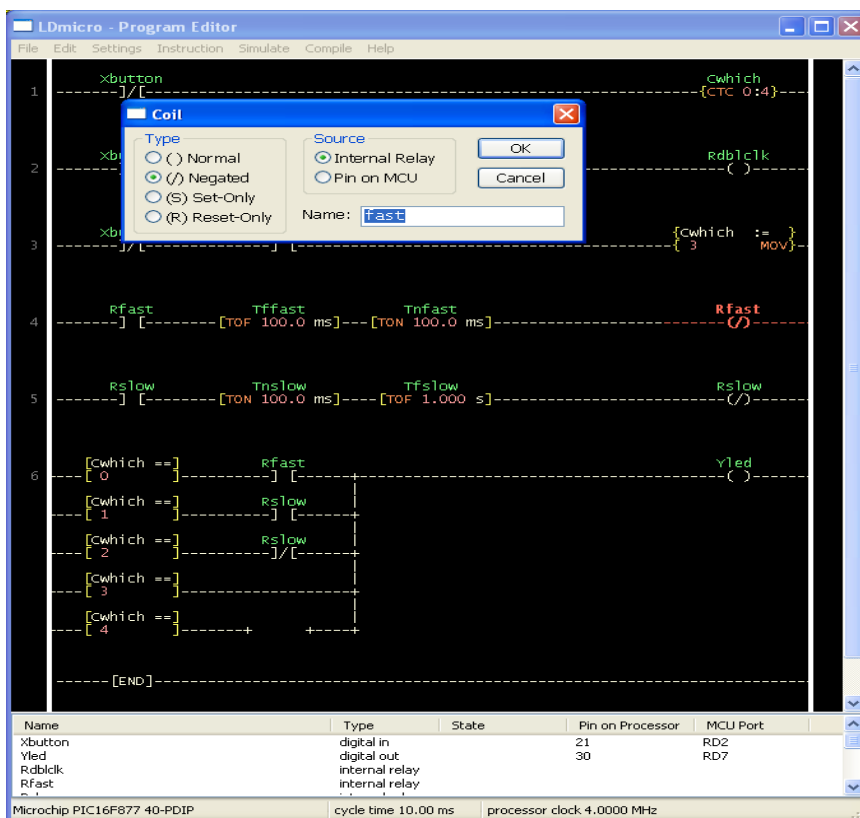
Embora as linguagens mais empregadas na programação de microcontroladores sejam Assembly, C e C++, neste trabalho será adotada a linguagem Ladder por ser a mais utilizada na programação de controladores lógicos programáveis. Portanto, se faz necessário um programa que permita a programação de microcontroladores PIC nesta linguagem.

O *software* LDmicro resolve este problema, pois permite a programação via diagrama Ladder para microcontroladores PIC16 ou AVR. Além das entradas e

saídas digitais, também são suportadas funções como temporizadores (TON, TOF e RTO), contadores (CTU, CTD e contadores circulares), entradas analógicas, saídas de modulação por largura de pulso (PWM), comunicação serial, criação de variáveis, execução de operações aritméticas (adição, subtração, multiplicação e divisão), comparação de valores e armazenamento de dados (WESTHUES, 2016).

A programação começa com uma linha vazia, onde é possível adicionar elementos como contatos, bobinas e diversas funções suportadas. Cada elemento tem várias configurações a serem feitas, por exemplo, uma bobina pode ser configurada como um relé interno ou um pino do microcontrolador, que podem ser do tipo normal, negado, *set* ou *reset*. Os elementos podem ser adicionados ao circuito em série ou paralelo. As entradas e saídas do diagrama podem ser associadas a elementos de memória ou aos pinos do microcontrolador. É necessário configurar o ciclo de varredura do código, a frequência do cristal que será utilizado no microcontrolador e a taxa de transferência da comunicação serial. A Figura 2 mostra o ambiente de programação Ladder (WESTHUES, 2016).

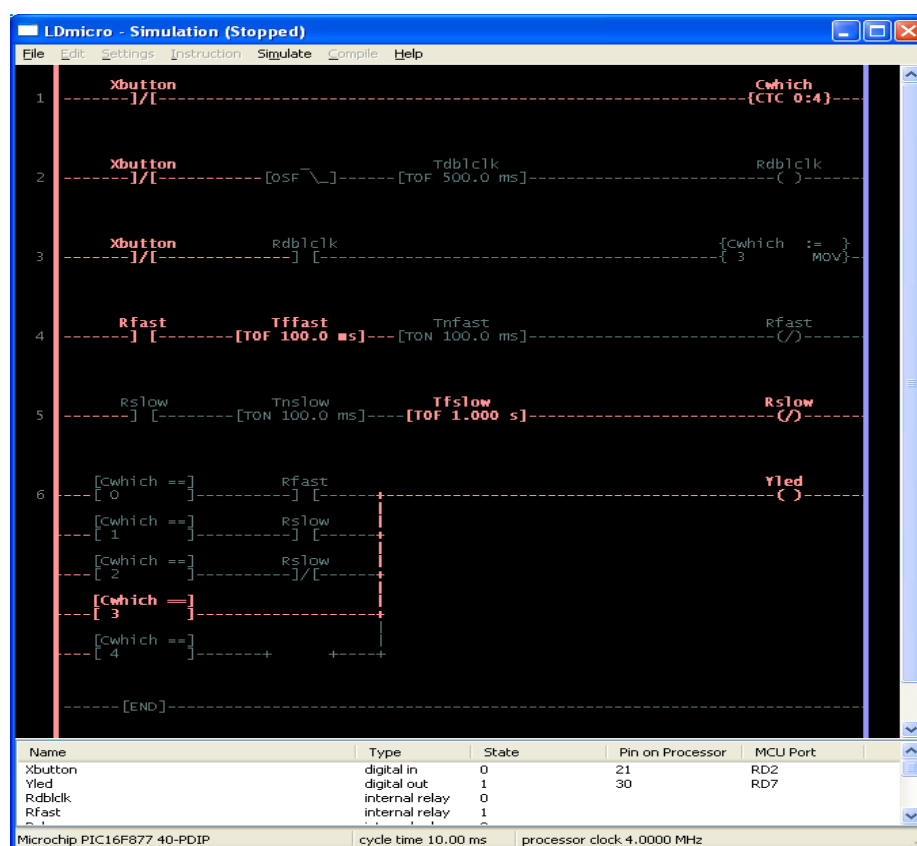
Figura 2 - Ambiente de programação do software LDMICRO



Fonte: WESTHUES, 2016.

Além da programação de forma gráfica, o *software* permite a simulação do programa, com o intuito de testar o programa antes mesmo de gravá-lo no microcontrolador. Durante a simulação, os elementos ativos e linhas energizadas serão representados com cor mais intensa e ainda será mostrada uma lista de entradas e saídas, contendo os estados das variáveis, o que facilita a verificação de erros do programa. Um exemplo de simulação está representado na Figura 3 (WESTHUES, 2016).

Figura 3 - Exemplo de simulação de um programa no software LDMICRO



Fonte: WESTHUES, 2016.

Assim que o programa for testado e estiver funcionando corretamente, as entradas e saídas devem ser associadas aos pinos do microcontrolador e quando o programa for compilado, será gerado um arquivo no formato Intel IHEX, que pode ser gravado no microcontrolador através de um gravador específico (PICkit 3, por exemplo). Também possui a opção de gerar um código em linguagem C, que pode ser editado e compilado em algum outro *software* (WESTHUES, 2016).

Segundo Westhues (2016), o software LDMICRO é livre, tendo o código fonte e executáveis disponibilizados para livre acesso. Isso permite que caso ocorra

algum erro ou alguma melhoria for necessária, o próprio usuário possa resolver. Mesmo o software não tendo funções tão complexas como os softwares específicos de CLP (desenvolvidos pelo fabricante), possui as funções necessárias para a aplicação a pequenos sistemas de automação, que é o intuito deste trabalho, cujos modelos suportados pelo software são: PIC16F628(A), PIC16F88, PIC16F819, PIC16F877(A), PIC16F876(A), PIC16F887, PIC16F886, ATmega128, ATmega64, ATmega162, ATmega32, ATmega16 e ATmega8 (WESTHUES, 2016).

3.3 Microcontroladores

Os microcontroladores podem ser definidos como pequenos dispositivos eletrônicos que são inteligentes e podem ser programados para realizar o controle de processos lógicos. Como pode ser programado, a sua lógica de operação é definida por um programa que é gravado no componente e sempre que o componente for alimentado, este programa será executado (SOUZA e SOUZA, 2012).

São formados por Unidade Central de Processamento (CPU), pinos de entrada e saída, memória de dados e de programa, e periféricos como: *timers*, contadores, PWM, conversores analógico-digitais, comunicação serial, etc. O que diferencia um microcontrolador de um microprocessador é que o microcontrolador reúne o microprocessador e os periféricos necessários para seu funcionamento em um único *chip* (MIYADAIRA, 2013; NICOLOSI, 2013; SOUZA e SOUZA, 2012).

As principais arquiteturas utilizadas em microcontroladores são Harvard e Von-Neumann, que se diferenciam por existir um barramento de acesso à memória de dados e outro para a memória de programa no caso da arquitetura Harvard, e na Von-Neumann, as memórias utilizam o mesmo barramento. (MIYADAIRA, 2013).

De acordo com Miyadaira (2013), a memória de programa tem a função de armazenar o programa que define a lógica de operação, é do tipo não volátil, portanto seus dados não são perdidos caso seja desenergizada, e podem ser de quatro tipos:

- ROM: permite apenas leitura pelo usuário;
- EPROM: permite ser gravada e é apagada através de exposição da janela de quartzo à luz ultravioleta;

- OTP: pode ser gravada apenas uma vez;
- FLASH: permite se apagada eletricamente e regravada em torno de 100 mil a um milhão de vezes, portanto é a mais flexível.

Diferente da memória de programa, a memória de dados é volátil, pois armazena os dados de operação do sistema somente enquanto estiver energizada e é definida como memória RAM (do inglês, *Random Access Memory*). As variáveis devem ser carregadas para este tipo de memória quando o dispositivo for energizado. (MIYADAIRA, 2013).

Todos os microcontroladores possuem pinos de entrada/saída (também conhecidos como Pinos I/O) que permitem a comunicação do sistema com o mundo exterior. Podem ser definidos como entrada, onde realizará a leitura do sinal do pino, ou como saída, onde definirá o nível de tensão do pino para controlar periféricos (MIYADAIRA, 2013).

3.3.1 Microcontrolador PIC16F877A

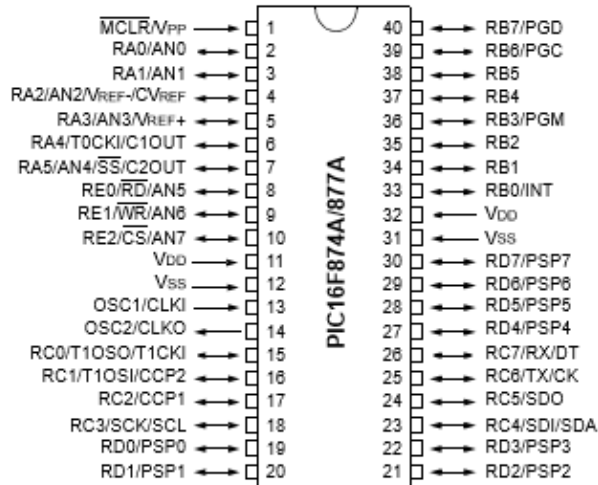
De acordo com os dados obtidos a partir do datasheet disponibilizado por Microchip Technology Inc. (2013), o PIC16F877A é um microcontrolador de 40 pinos que apresenta arquitetura Harvard, com um barramento de dados de 8 *bits* e um barramento de programa de 14 *bits*. Pode operar com uma frequência de até 20 MHz, com um período de ciclo de máquina de 200 ns. Como conta com uma CPU RISC de alto desempenho, possui apenas 35 instruções que necessitam ser aprendidas. As principais características do modelo PIC16F877A são:

- Memória de programa FLASH de 8K *words*;
- Memória RAM de 368 *bytes* e memória EEPROM de 256 *bytes*;
- 33 pinos de entrada e saída, divididos entre cinco portas digitais;
- Conversor Analógico-Digital com resolução de 10 *bits* em 8 pinos;
- Dois módulos CCP (*Capture, Compare, PWM*);
- Porta Serial Síncrona (SSP), Interface Periférica Serial (SPI) e I2C;
- Módulo USART para comunicação serial assíncrona;
- Dois *Timers* de 8 *bits* e um *Timer* de 16 *bits*;
- *WatchDog Timer* (WDT);
- Modo de operação de baixo consumo de energia.

A Figura 4 mostra a pinagem do modelo PIC16F877A com encapsulamento DIP de 40 pinos.

Figura 4 - Pinagem do modelo PIC16F877A

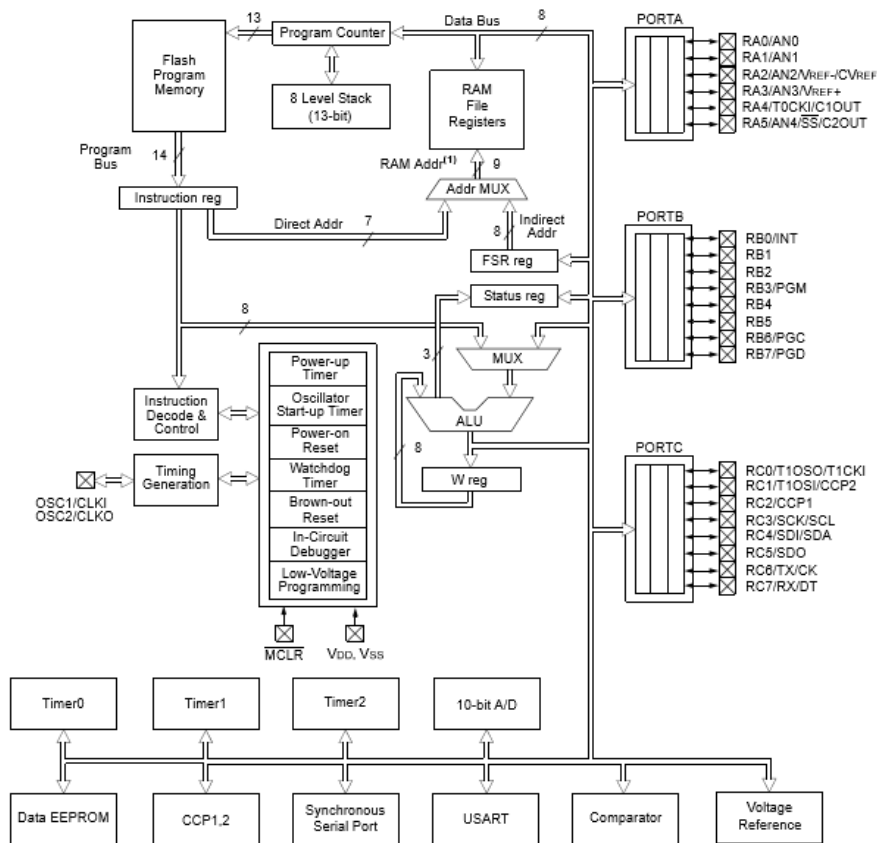
40-Pin PDIP



Fonte: MICROCHIP TECHNOLOGY INC., 2013, p. 3.

A arquitetura interna deste microcontrolador é representada pelo diagrama de blocos presente na Figura 5.

Figura 5 - Arquitetura interna do modelo PIC16F877A



Fonte: MICROCHIP TECHNOLOGY INC., 2013, p. 6.

3.4 Redes de comunicação

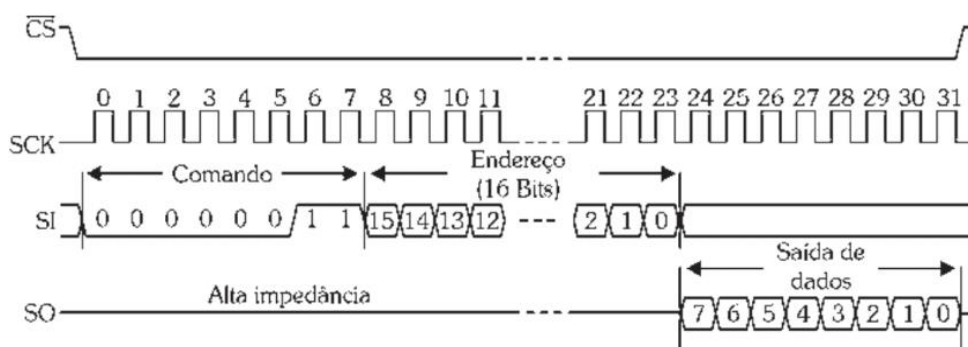
A necessidade de troca de informações entre dispositivos, junto com o aumento da quantidade de dispositivos digitais de controle, provocou a inserção do conceito de redes de comunicação, que antes era mais comum no âmbito administrativo, também no chão de fábrica (FOROUZAN, 2010; SILVEIRA, 2010; PAREDE e GOMES, 2011).

Uma rede de comunicação pode ser definida como um conjunto de dispositivos conectados por um meio físico (fios, radiofrequência, etc.), que podem trocar informações entre si. Os dispositivos presentes nestas redes são normalmente conhecidos como nós (FOROUZAN, 2010; PAREDE e GOMES, 2011; TANENBAUM, 2003).

Existem diversas formas de enviar informações, mas a forma mais utilizada é a comunicação serial. A comunicação serial consiste em transmitir a informação dividida em *bits*, que são transmitidos em série, ou seja, um após o outro, até que toda a informação seja transmitida. Esta comunicação pode ser estabelecida de forma síncrona ou assíncrona (PAREDE e GOMES, 2011; PEREIRA, 2009).

Na comunicação assíncrona, a sincronização entre o transmissor e receptor é feita pelo *clock* de cada dispositivo e por marcadores de início e fim da transmissão. A transmissão começa com um *bit* de início (*start bit*), em seguida são enviados os *bits* de dados, começando pelo *bit* menos significativo, e para finalizar a transmissão é enviado o bit de fim de transmissão (*stop bit*). A Figura 6 ilustra a transmissão do caractere “A”, segundo o código da tabela ASCII, pelo modo de transmissão serial assíncrona do protocolo RS-232. (PEREIRA, 2009).

Figura 6 - Transmissão assíncrona do caractere “A”



Fonte: PEREIRA, 2009, p.270.

Podemos observar na Figura 6, que a cada transição de *clock* (linha SCK), um *bit* de informação é enviado na linha de dados (linha SI).

Para que a comunicação entre os dispositivos ocorra de forma correta, com o mínimo de perdas, se faz necessário estabelecer regras que definam e orientem o processo de transmissão de informações pela rede, este conjunto de regras é definido como protocolo de comunicação (SILVEIRA, 2010).

3.4.1 Protocolos de comunicação

Reunindo as ideias de Tanenbaum (2003), Silveira (2010) e Parede e Gomes (2011), um protocolo de comunicação pode ser definido como um conjunto de regras que permitem a comunicação entre dois ou mais dispositivos, que determina o formato da mensagem ou dos pacotes, as ações que devem ser executadas quando as informações são enviadas ou recebidas, a velocidade em que as informações são transmitidas, entre outras definições necessárias para uma boa comunicação.

De acordo com Silveira (2010), o conceito de protocolo de comunicação pode ser comparado com o processo de envio de uma carta. A mensagem escrita na carta deve estar em um idioma que o remetente e o destinatário compreendam, e precisa ser escrita de forma legível. Então, a carta deve ser colocada dentro de um envelope, contendo as informações necessárias, como nome do destinatário e seu endereço, e este envelope deve ser colocado em uma caixa de correio, para que os correios possam realizar sua função de forma correta e a mensagem possa ser entregue sem problemas.

Segundo Silveira (2010) e Mondadori (2016), a ISO (do inglês, *International Standard Organization*, e em português, Organização de Padronização Internacional) é uma organização que tem como objetivo a padronização, e atua em vários setores, incluindo a comunicação. Visando a padronização e interconectividade, desenvolveu o modelo OSI (do inglês, *Open System Interconnection*, e em português, Interconexão de Sistemas Abertos). O modelo OSI define sete camadas que definem as funções de um protocolo de comunicação em todos os níveis de uma rede.

Baseando-se nas ideias de Tanenbaum (2003), Mondadori (2016) e Silveira (2010), as sete camadas serão citadas e explicadas a seguir:

- Camada física: define como os *bits* serão transmitidos fisicamente, especificando as características elétricas e mecânicas, como cabeamento, conectores e níveis de tensão;
- Camada de enlace: responsável pela detecção de erros de comunicação, controle da taxa de transferência de frames e gerenciamento de acesso ao meio físico;
- Camada de rede: gerencia o roteamento da rede, ou seja, encontra o melhor caminho para a mensagem seguir até seu destino, e envolve a função de endereçamento;
- Camada de transporte: assegura que a mensagem chegará completa ao seu destino. Controla o fluxo e define a melhor maneira de transportar a mensagem, dividindo-a em partes menores se necessário;
- Camada de sessão: responsável por estabelecer a conexão, controlar quem deve transmitir a cada momento, e sincronização, verificando transmissões longas, para que possa restabelecer a conexão a partir do ponto onde pararam, caso ocorra algum problema;
- Camada de apresentação: sua principal função é converter as informações recebidas em formato de linguagem de máquina para uma linguagem que possa ser compreendida facilmente pelo usuário. E da mesma forma, convertendo informações inseridas pelo usuário para formato de linguagem de máquina;
- Camada de aplicação: Administra e fornece os recursos necessários para a transferência da mensagem. É uma camada mais próxima do entendimento do usuário.

Segundo Parede e Gomes (2011), Stevan Junior, Leme e Santos (2018), Fonseca, Santos e Coelho (2016), os principais protocolos utilizados nos ambientes industriais são: Modbus, Profibus, CAN, DeviceNET, Hart, AS-Interface e Ethernet. A maioria destes protocolos possuem várias versões adaptadas para cada aplicação, e serviram como base para criação de outros protocolos. Os protocolos que não definem as características físicas de transmissão utilizam padrões elétricos já existentes, como RS-232, RS-422 e RS-485.

Como o microcontrolador escolhido como base apresenta um módulo USART (do inglês, *Universal Synchronous Asynchronous Receiver Transmitter*) que permite a aplicação dos protocolos de camada física RS-232 e RS-485, manteremos o foco nestes protocolos.

3.4.1.1 Protocolo RS-232

O padrão RS-232 foi criado em 1962 para padronizar a comunicação entre os dispositivos DTE e DCE, utilizando comunicação serial assíncrona e topologia ponto a ponto, ou seja, uma conexão direta entre os equipamentos. Um DTE, do inglês, *Data Terminal Equipment*, pode ser um computador ou uma impressora. Um DCE, do inglês, *Data Communication Equipment*, pode ser um modem. O DTE é responsável por fornecer os dados, e o DCE tem a função de controle, estabelecendo, mantendo e terminando a conexão, além de decodificar os sinais recebidos do DTE. Ao longo do tempo, este padrão foi adotado para outras aplicações e quando o computador passou a ser utilizado na indústria, o RS-232 se popularizou com um padrão de baixo custo para comunicação entre um DTE e outros dispositivos (PECKOL, 2019; REYNDERS, 2005; TEXAS INSTRUMENTS, 2002).

O padrão RS-232 define as características elétricas e mecânicas, e o controle da troca de dados, portanto faz parte da primeira camada do modelo OSI, ou seja, da camada física (REYNDERS, 2005; TEXAS INSTRUMENTS, 2002).

Os níveis de tensão não podem exceder os limites de -25 V a +25 V. Para o transmissor, uma tensão entre -5 V e -25 V representa nível lógico alto, uma tensão entre +5 V e +25 V representa nível lógico baixo, e a faixa entre -5 V e +5 V é considerada como nível lógico indefinido. Já para o receptor, o nível lógico alto é representado por uma tensão entre -3 V e -25 V, o nível lógico baixo consiste em uma tensão entre +3 V e +25 V, e a tensão entre -3 V e +3 V é a faixa de tensão onde o nível lógico é indefinido. Essa diferença de 2 V entre o transmissor e o receptor, existe para evitar que ruídos ou a queda de tensão ao longo da linha interfiram na transmissão de dados (REYNDERS, 2005; TEXAS INSTRUMENTS, 2002).

O padrão RS-232 completo, especificava a utilização de conectores de 25 pinos, mas como nem todos os pinos são necessários para a comunicação na indústria e era necessário que o conector fosse menor para poder ser colocado em computadores e notebooks, o conector de 9 pinos (DB9) passou a ser utilizado. Mas nos dois casos, a conexão padrão é paralela, sem cruzamento, e o sinal tem sentido único em cada pino (PECKOL, 2019; REYNDERS, 2005; TEXAS INSTRUMENTS, 2002).

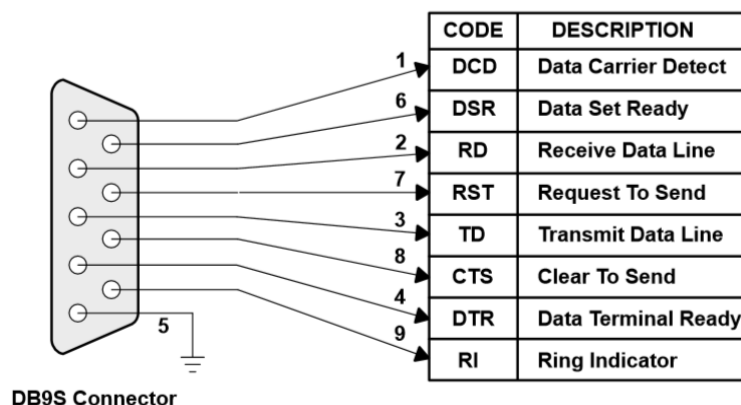
Segundo Reynders (2005) e Texas Instruments (2002), essas são as funções de cada pino do conector DB9:

- **Protective ground (shield):** o aterramento garante que as carcaças dos dois conectores estejam em mesmo potencial;
- **Transmitted data (TxD):** nesta linha, os dados são transmitidos, de forma serial, partindo do DTE em direção ao DCE. Em condição de *idle*, esta linha é mantida com tensão negativa;
- **Received data (RxD):** por esta linha, o DCE envia dados, de forma serial, para o DTE;
- **Request to send (RTS):** este sinal é ativo pelo DTE quando ele está pronto para transmitir dados. Então o DCE deve se preparar para receber estes dados e ativar a linha CTS (*Clear to Send*);
- **Clear to send (CTS):** assim que o DTE informar que está pronto para transmitir dados, ativando o RTS, o DCE se prepara e torna esta linha ativa, informando que está pronto para receber estes dados;
- **Data set ready (DSR):** este sinal é enviado pelo DCE ao DTE, para indicar que está pronto e conectado na linha;
- **Signal ground (comum):** é o pino que fornece uma referência de terra comum para a comunicação entre todos os circuitos;
- **Data carrier detect (DCD):** este sinal é ativo pelo DCE quando ele detecta que há transferência de dados, e continua ativo até que esta transferência termine;
- **Data Terminal Ready (DTR):** o DTE torna este sinal ativo para informar ao DCE, que está pronto para se comunicar, seja para transmitir ou para receber dados;

- **Ring Indicator (RI):** se torna ativo quando o DCE detecta um sinal de chamada. Oriundo da sua aplicação padrão como telefone, este sinal é ativo quando o telefone está tocando.

A Figura 7 ilustra a relação de cada pino do conector DB9 (fêmea) com sua respectiva função.

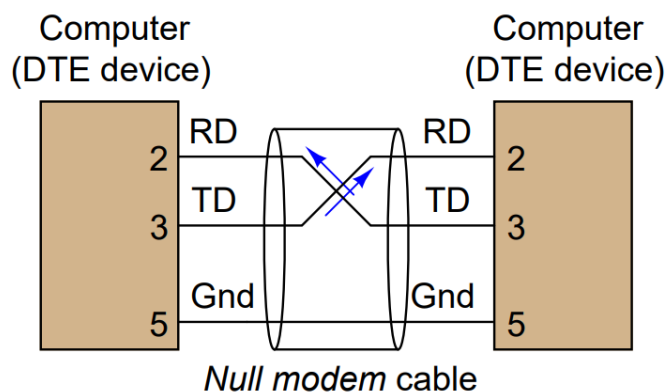
Figura 7 - Pinagem do conector DB9 (fêmea)



Fonte: TEXAS INSTRUMENTS, 2002, p. 7.

Como a maioria dos dispositivos utilizam uma interface do tipo DTE, se torna necessário realizar a comunicação entre eles, mas como o protocolo foi desenvolvido para a comunicação entre um DTE e um DCE, é preciso fazer alguns ajustes. Para resolver este problema, existe o modo de conexão conhecido como *null-modem*, que utiliza apenas três linhas: RD, TD e terra comum. Neste modo de conexão, as linhas 2 (RD) e 3 (TD) são cruzadas, ou seja, a linha TD de cada DTE deve ser conectada ao pino RD do outro DTE. A Figura 8 ilustra esta conexão (KUPHALDT, 2019; REYNDERS, 2005; TEXAS INSTRUMENTS, 2002).

Figura 8 - Conexão null-modem



Fonte: KUPHALDT, 2019, p. 1055.

3.4.1.2 Protocolo RS-485

Segundo Weis (2019), Kugelstadt (2016), Soltero (2010) e Marais (2008), o padrão de comunicação serial RS-485 é um dos mais utilizados em redes industriais, e pode ser considerado um protocolo de camada física no modelo OSI, já que padroniza apenas as características elétricas necessárias para a comunicação. Geralmente opera em conjunto com protocolos de camadas superiores, como Modbus, Profibus, ARCNET, etc. As suas principais características são:

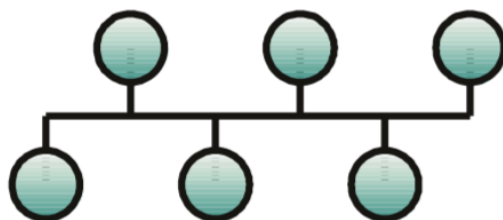
- Comunicação *Half-Duplex* ou *Full-Duplex*;
- Proteção contra interferências e ruídos elétricos;
- Suporta até 32 *Unit Loads* (UL);
- Permite que o barramento seja longo (até 1200 m);
- Alta velocidade de transmissão (100 Kbps a 10 Mbps).

Este padrão opera baseado na diferença de tensão entre dois fios, chamados de A e B. Se a tensão no fio A for maior que a tensão no fio B ($V_A > V_B$), é considerado nível lógico alto na linha, e caso a tensão no fio A for menor que a tensão no fio B ($V_A < V_B$), considera-se nível lógico baixo na linha. Por conta de operar desta maneira, este padrão é capaz de rejeitar ruídos, já que se houverem interferências, afetarão os dois fios da linha, mantendo a mesma diferença de tensão entre eles (MARAIS, 2008).

Os dispositivos emissores devem fornecer uma tensão de no mínimo 1,5 V para uma carga de 54 Ω , enquanto os receptores devem receber pelo menos 200 mV. Essas condições permitem que ocorra a transmissão de dados mesmo em ambientes onde há diminuição da amplitude do sinal ao longo da linha. Esse é um dos motivos que torna o RS-485 eficiente para transmissão a longas distâncias em ambientes com muito ruído (KUGELSTADT, 2016).

De acordo com Kugelstadt (2016), a topologia de rede, ou seja, a forma com que os dispositivos são conectados na rede, pode ser do tipo barramento ou sua variação conhecida como *daisy-chain*. A topologia em barramento é constituída por cabos formando o “tronco” principal e os dispositivos são conectados ao tronco por meio de derivações, como pode ser visto na Figura 9.

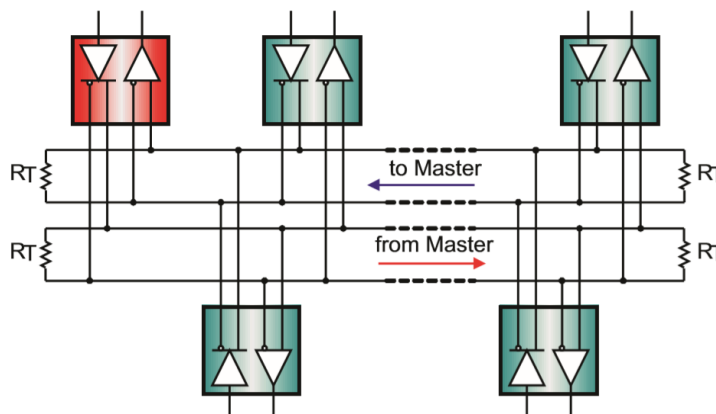
Figura 9 - Topologia de rede em barramento



Fonte: KUGELSTADT, 2016, p. 2.

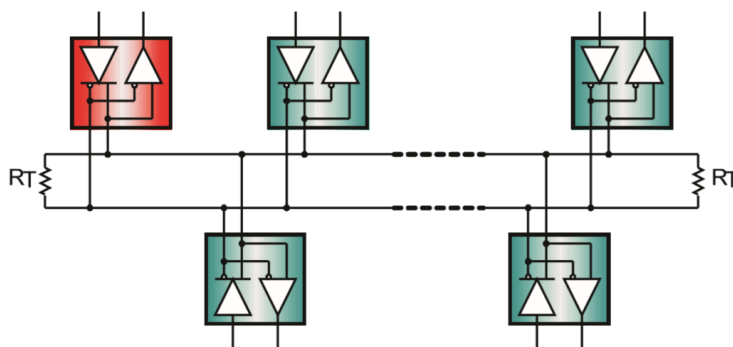
O barramento RS-485 pode ser *full-duplex* ou *half-duplex*. A conexão *full-duplex* separa um barramento para o transmissor e outro para o receptor, portanto precisa de dois pares de fios. Isso permite que os dispositivos transmitam e recebam dados simultaneamente. Já a conexão *half-duplex* utiliza um único par de fios, portanto a transmissão e recepção de dados não podem ocorrer ao mesmo tempo. As figuras 10 e 11 representam as conexões *full-duplex* e *half-duplex*, respectivamente (KUGELSTADT, 2016; MARAIS, 2008).

Figura 10 - Conexão Full-Duplex



Fonte: KUGELSTADT, 2016, p. 2.

Figura 11 - Conexão Half-Duplex



Fonte: KUGELSTADT, 2016, p. 2.

O cabeamento de uma rede RS-485 deve ser feito por meio de pares de fios trançados, porque o ruído externo afetará igualmente os dois fios, mantendo a integridade das informações transmitidas. Estes cabos podem ser do tipo blindado ou não blindado (KUGELSTADT, 2016).

Em cabos longos, o receptor pode fazer a leitura incorreta do nível lógico da linha, por conta do efeito de reflexão do sinal no fim da linha. Para evitar que o sinal seja refletido, são necessários resistores de terminação com o mesmo valor da impedância característica da linha, que no caso dos cabos recomendados para o RS-485, tem um valor de 120Ω . No modo de terminação mais utilizado, são colocados dois resistores de 120Ω , um em cada ponta da linha. Na configuração *half-duplex*, as duas pontas da linha devem ter resistores de terminação, e na configuração *full-duplex*, apenas os receptores do mestre e do escravo mais distante precisam do resistor de terminação (KUGELSTADT, 2016; MARAIS, 2008).

Quando nenhum dispositivo da rede está ativo, tem-se uma condição conhecida como *idle*. Nesta condição, os receptores podem detectar valores aleatórios de tensão, causando a leitura incorreta do nível lógico. Uma maneira de resolver este problema é inserir em um ponto da linha, um divisor de tensão resistivo que seja capaz de fornecer a tensão mínima necessária para que o receptor detecte corretamente o estado lógico da linha. Esta técnica é conhecida como *Failsafe Biasing* (KUGELSTADT, 2016; MARAIS, 2008; SOLTERO, 2010).

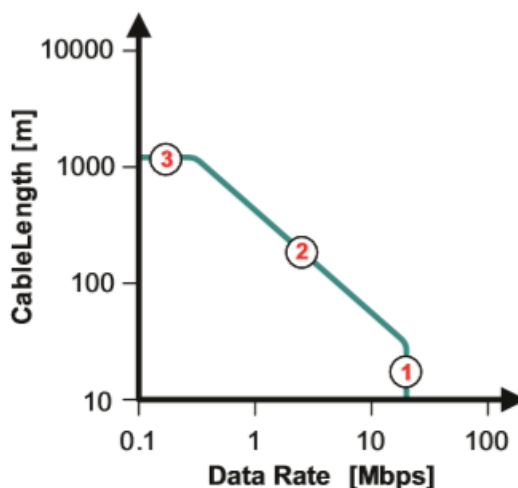
Como a saída dos dispositivos dependem da corrente que ele precisa fornecer para uma carga, quanto mais dispositivos forem adicionados à rede, maior a corrente necessária. A impedância de um receptor do padrão RS-485 é definida como *Unit Load* (UL), e seu valor é especificado em $12 \text{ K}\Omega$. O padrão RS-485 especifica que a linha deve ser capaz de suportar até 32 ULs. Mas alguns modelos de receptores tem a impedância equivalente a $\frac{1}{4}$ UL ou $\frac{1}{8}$ UL, e isso permite que mais dispositivos sejam adicionados à rede. Por exemplo, se os receptores possuírem $\frac{1}{4}$ UL, a quantidade máxima de UL quadruplica ($4 \times 32 = 128$), e se possuírem $\frac{1}{8}$ UL, a quantidade máxima de UL é oito vezes maior ($8 \times 32 = 256$) (KUGELSTADT, 2016; MARAIS, 2008; SOLTERO, 2010).

Mas segundo Kugelstadt (2016), como a técnica de *Failsafe Biasing* ocupa até 20 ULs, o número máximo de *transceivers* que podem ser conectados na rede seguem a Equação 1:

$$N = \frac{32 UL_{padr\tilde{a}o} - 20 UL_{Failsafe}}{UL_{por\ transceiver}} \quad (1)$$

A máxima distância da linha varia conforme a velocidade de transmissão de dados. Esta distância é limitada pela perda de tensão na linha e pela oscilação do sinal em determinada velocidade de transmissão. A máxima velocidade é de 10 Mbps e a mínima velocidade é de 100 Kbps, tendo 4000 pés como distância máxima, que equivale a aproximadamente 1219 metros. A Figura 12 mostra a variação do comprimento da linha conforme o aumento da taxa de transmissão de dados no protocolo RS-485 (KUGELSTADT, 2016; MARAIS, 2008).

Figura 12 - Comprimento da linha x Taxa de transmissão de dados (RS-485)



Fonte: KUGELSTADT, 2016, p. 5.

4 MATERIAIS E MÉTODOS

Este capítulo apresenta a arquitetura e as características básicas propostas para o controlador lógico programável, e também descreve os procedimentos e os programas envolvidos no seu desenvolvimento e na sua construção.

O primeiro passo do desenvolvimento foi a elaboração do circuito elétrico, com o apoio de simulações no *software* PROTEUS e de testes em protótipos montados em *protoboards*. Em seguida, as placas de circuito impresso (PCI) foram projetadas no PROTEUS, o qual também forneceu a lista de todos os componentes da PCI, a qual está disponível no Apêndice A.

A programação em Ladder foi realizada através do software livre LDmicro, desenvolvido por Jonathan Westhues. A utilização deste software visou economizar tempo e investimento na elaboração de um software de programação Ladder específico para este CLP, enquanto que a lógica de programação foi transferida ao CLP através do gravador PICkit3 apresentado na Figura 13.

Figura 13 - Conversores buck com ajuste de tensão



Fonte: Aatoria Própria.

Este controlador lógico programável foi baseado no microcontrolador PIC16F877A, pois este é um dos modelos mais completos suportados pelo *software* de programação Ladder e também por conta de seu baixo custo, popularidade e facilidade de aquisição.

A partir do estudo das principais características de um CLP, da análise da arquitetura do microcontrolador escolhido e das opções disponíveis no software de programação, foram propostas as características mínimas para este controlador lógico programável.

As características mínimas propostas, são apresentadas a seguir:

- Tensão de Alimentação a 24 Vcc;
- Programação em Ladder;
- 8 entradas digitais opto acopladas e 8 saídas digitais a relé;
- 4 entradas analógicas e 2 saídas analógicas, com faixa de tensão de 0 a 10 V;
- Interface de comunicação serial RS-232 e/ou RS-485.

Os circuitos desenvolvidos para atender a estas especificações e os projetos das placas de circuito impresso serão apresentados e explicados a seguir. O esquemático do circuito está disponível no Apêndice B.

4.1 Alimentação

O circuito é alimentado com 24 V, mas alguns componentes precisam de alimentação com outros valores de tensão, como 12 V e 5 V. Para fornecer estes valores de tensão, foram utilizados dois conversores *buck* comerciais com ajuste da tensão de saída, conforme a Figura 14, onde a tensão de saída de um conversor foi regulada em 5 V e no outro conversor ajustou-se em 12 V.

Figura 14 - Conversores buck com ajuste de tensão

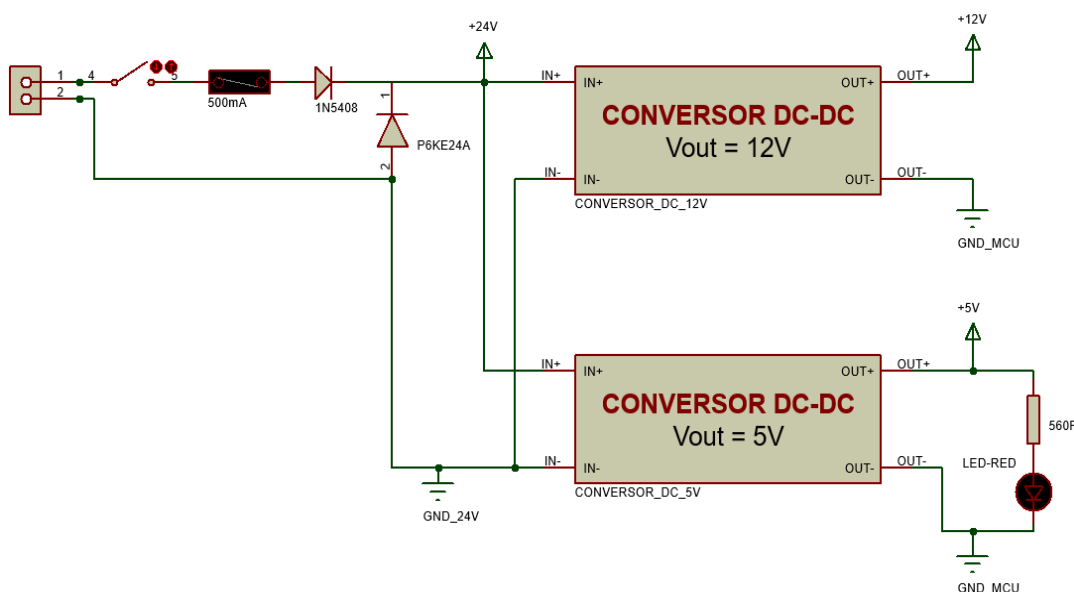


Fonte: Autoria Própria.

A fim de proteger os conversores e o restante do circuito, foi adicionado um circuito simples de proteção, formado por um fusível de 500 mA, um diodo 1N5458 e um diodo supressor P6KE24A. O diodo 1N5458 protege o circuito contra inversão de polaridade na alimentação, o diodo supressor protege contra sobretensão e o fusível

se rompe em caso de sobrecorrente. Entre o fusível e o terminal de alimentação, foi colocado um botão para ligar e desligar o circuito quando necessário. Na saída do conversor *buck* ajustado em 5 V, foi adicionado um LED, para indicar que o circuito está energizado, e um resistor de 560 Ω , para limitar a corrente do LED. A Figura 15 apresenta o circuito de alimentação do dispositivo.

Figura 15 - Circuito de alimentação



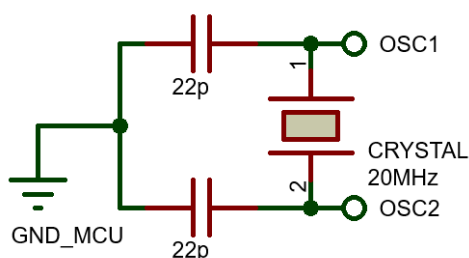
Fonte: Autoria Própria.

Desta forma, este circuito de alimentação é capaz de fornecer a tensão de entrada (24 V) e as tensões de saída dos conversores *buck* (5 e 12 V).

4.2 Microcontrolador

O circuito de operação do microcontrolador foi desenvolvido de forma que garanta seu bom funcionamento e a gravação do código de forma simples. O microcontrolador foi alimentado com 5 V e utiliza dois capacitores de desacoplamento, com valor de 100 nF, conectados próximos aos pinos de alimentação. A sua frequência de operação foi definida pelo seu circuito oscilador, formado por um cristal oscilador de 20 MHz e dois capacitores de 22 pF, conforme a Figura 16.

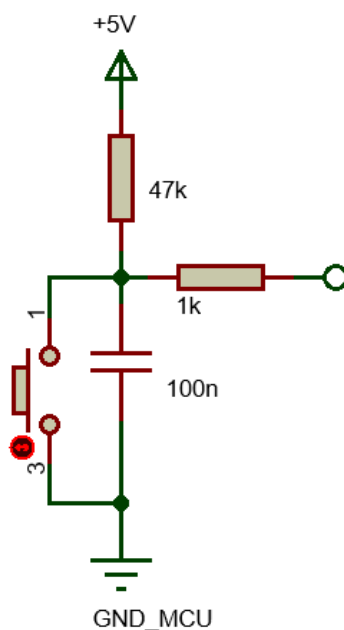
Figura 16 - Circuito oscilador do microcontrolador



Fonte: Autoria Própria.

Conforme recomendação do *datasheet* do componente, o seu circuito de *reset* foi formado por um resistor de 47 k Ω , um capacitor de 100 nF, um resistor de 1 k Ω e um botão, conectados conforme a Figura 17. Os resistores e o capacitor formam um filtro contra ruídos e descargas eletrostáticas, já o botão, quando é pressionado, causa o *reset* do microcontrolador.

Figura 17 - Circuito de *reset* do microcontrolador



Fonte: Autoria Própria.

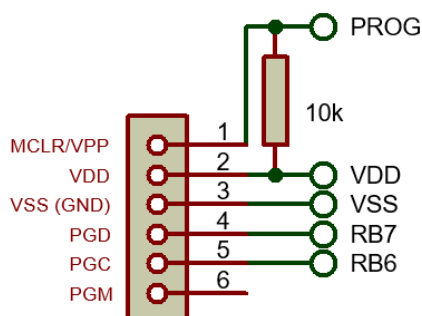
Para transferir a lógica de operação para o microcontrolador, foi utilizado o gravador PICkit 3. Este gravador possui seis pinos que devem ser conectados ao microcontrolador, conforme apresentado no Quadro 1.

Quadro 1 - Relação de pinos do gravador PICKit 3

Pinos PICKit 3	Pinos Microcontrolador
1	MCLR/VPP
2	VDD
3	GND
4	PDG (ICSPDAT)
5	PGC (ICSPCLK)
6	PGM (LVP)

Fonte: Microchip Technology Inc, 2012, p.21.

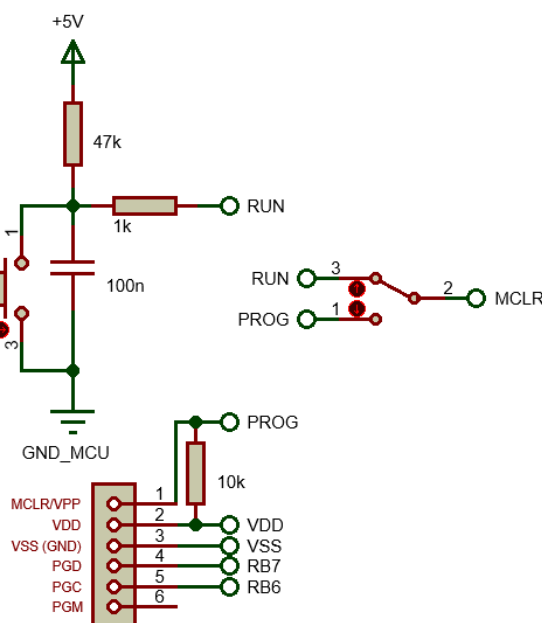
Um conector com 6 pinos, foi utilizado para a conexão com o gravador. Entre o pino VDD e MCLR, foi conectado um resistor de *pull-up* com valor de 10 k Ω , conforme recomendado pelo guia de usuário de Microchip Technolgy Inc. (2012), e apresentado na Figura 18. Também é recomendado evitar a conexão de capacitores no pino MCLR, pois pode atrapalhar as rápidas variações de tensão que ocorrem durante a gravação. Isso gera um problema, já que é necessário um capacitor no filtro de ruídos do pino MCLR.

Figura 18 - Conector de gravação do microcontrolador

Fonte: Aatoria Própria.

Buscando contornar este problema, foi incluída uma chave de seleção com dois polos, conectada ao circuito de *reset* e ao circuito de gravação, com sua saída conectada ao pino MCLR do microcontrolador, conforme a Figura 19.

Figura 19 - Circuito conectado ao pino MCLR do microcontrolador



Fonte: Autoria Própria.

Desta forma, é feita a seleção entre dois modos de operação do microcontrolador, o modo de operação normal (RUN) e o modo de gravação (PROG). No modo RUN, o microcontrolador executa o código programado, possui um filtro contra ruídos no pino MCLR e permite o *reset* do microcontrolador. Já no modo PROG, é possível gravar as instruções no microcontrolador, que ainda é capaz de executar o que foi programado, porém não é possível resetar o microcontrolador. Recomenda-se que o microcontrolador permaneça em modo RUN para a execução das suas instruções programadas, mudando para o modo PROG apenas durante a gravação do código no microcontrolador.

4.3 Entradas digitais

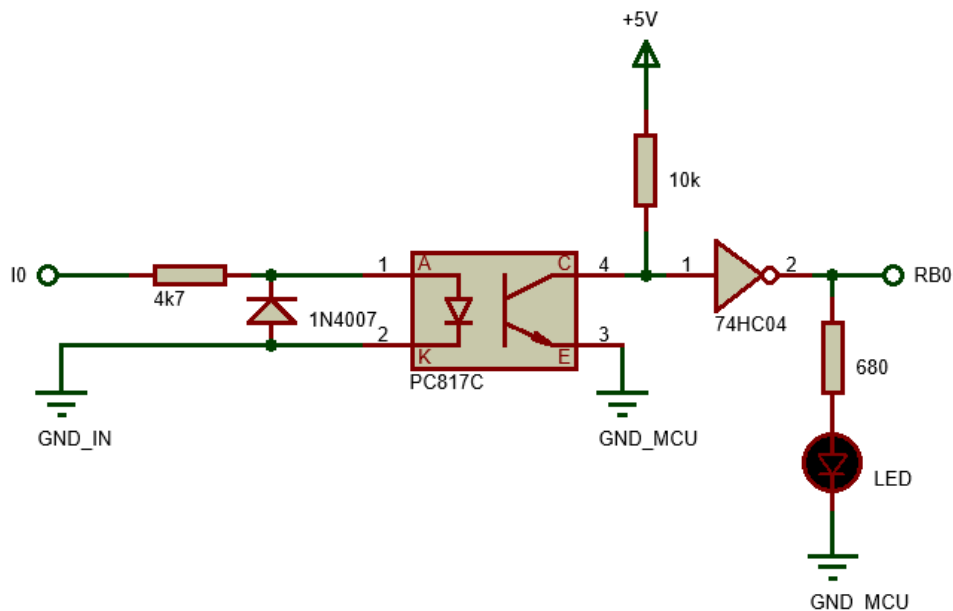
As entradas digitais foram desenvolvidas como tipo dreno, para que sejam compatíveis com sensores do tipo fonte (PNP), recebendo sinais de corrente contínua, a uma tensão de 24 V.

Visando uma programação mais intuitiva, este circuito foi desenvolvido de forma que quando um sinal de 24 V for aplicado na entrada digital, o pino do microcontrolador, correspondente a esta entrada, reconheça um nível lógico alto.

Cada circuito de entrada digital é formado por um resistor de 4k7 Ω , um optoacoplador PC817C, um resistor de 10 k Ω , uma porta lógica NOT de um circuito

integrado 74HC04, um resistor de 680 Ω e um LED, como pode ser observado na Figura 20.

Figura 20 - Circuito de entrada digital



Fonte: Autoria Própria.

A Figura 20 apresenta o circuito de uma das entradas digitais. Ao lado esquerdo do optoacoplador, há um circuito operando com uma tensão de 24 V e ao lado direito, há um circuito alimentado por 5 V. O optoacoplador tem a função de acoplar estes dois circuitos, mantendo o isolamento elétrico entre eles. O resistor de 4k7 Ω limita a corrente que flui através do diodo de entrada do optoacoplador quando um sinal é aplicado na entrada I0. Para um sinal de 24 V, a corrente de entrada do optoacoplador é limitada em aproximadamente 5 mA, que é suficiente para saturar o fototransistor.

O resistor de 10 k Ω opera como resistor de *pull-up*, mantendo a entrada da porta lógica em nível lógico alto quando o fototransistor do optoacoplador não está em condução. A porta lógica NOT inverte o nível lógico apresentado em sua entrada. Desta forma, quando um sinal é aplicado na entrada I0, o fototransistor entra em condução e assim aplica um nível lógico baixo na entrada da porta lógica, resultando em um nível lógico alto na saída da porta lógica e no pino de entrada digital do microcontrolador (neste caso, RB0).

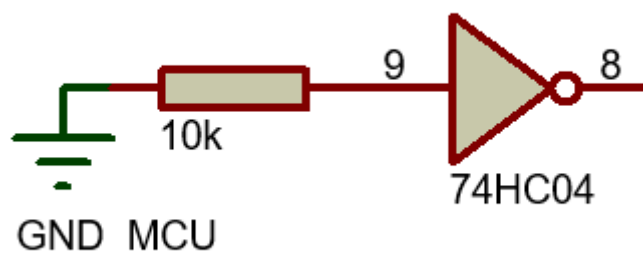
O resistor de 680 Ω limita a corrente do LED que sinaliza quando um sinal é recebido na entrada do circuito. O diodo em paralelo com o diodo do optoacoplador

serve como proteção contra uma possível inversão de polaridade na entrada do circuito, o que poderia danificar o optoacoplador.

Foram utilizados dois circuitos integrados 74HC04 para fornecer as portas lógicas ao circuito, sendo que cada circuito integrado tem seis portas lógicas NOT. Antes do desenvolvimento do circuito, esperava-se que o módulo de entradas digitais fosse formado por oito entradas, porém visando aproveitar melhor as portas lógicas e os pinos do microcontrolador que não foram utilizados, optou-se por um circuito com dez entradas.

Assim, cada circuito integrado 74HC04 atendeu cinco circuitos de entrada digital, sobrando apenas uma porta lógica em cada circuito integrado. Seguindo o *datasheet* do componente, as entradas das portas lógicas não utilizadas foram mantidas em nível lógico baixo, utilizando resistores de *pull-down* com valor de 10 k Ω , visando proteger o circuito integrado de possíveis danos devido à flutuação do nível lógico na entrada, conforme ilustrado na Figura 21.

Figura 21 - Entradas das portas lógicas não utilizadas



Fonte: Autoria Própria.

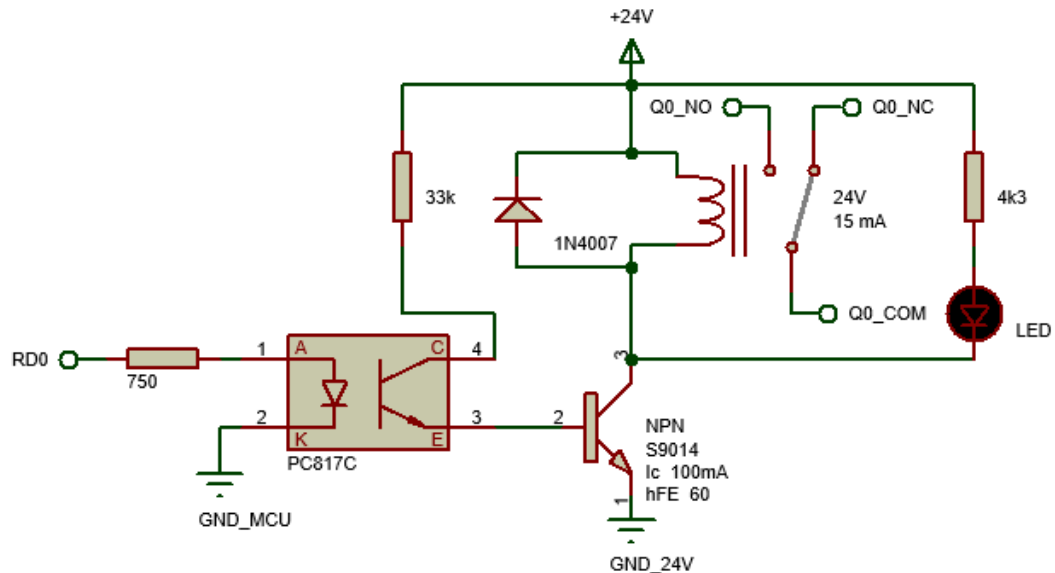
4.4 Saídas Digitais

Os circuitos de saídas digitais foram baseados em relés acionados pelo microcontrolador, o que permite que as cargas sejam acionadas tanto por corrente alternada quanto por corrente contínua.

O módulo de saídas digitais é formado por oito circuitos idênticos de acionamento de relés, cada um acionado por um pino diferente do microcontrolador. Cada circuito de saída digital é composto por um resistor de 750 Ω , um optoacoplador PC817C, um resistor de 33 k Ω , um transistor S9014 (NPN), um relé

24 V, um diodo 1N4007, um resistor de $4k3 \Omega$ e um LED, conforme apresentado na Figura 22.

Figura 22 - Circuito de saída digital



Fonte: Autoria Própria.

Neste circuito, o resistor de 750Ω limita a corrente na entrada do optoacoplador, o optoacoplador acopla os circuitos alimentados por 5 V e por 24 V, o resistor de $33 k\Omega$ limita a corrente de base do transistor NPN, o qual é responsável por controlar a corrente que flui pelo relé. O relé fecha ou abre os contatos conforme a corrente flui por sua bobina, o diodo 1N4007 opera como diodo de roda livre, evitando picos de tensão reversa no desligamento do relé, o resistor de $4k3 \Omega$ limita a corrente do LED que sinaliza quando o relé está acionado.

Quando o pino de saída digital do microcontrolador, correspondente a este circuito (neste caso, RD0), é definido como nível lógico alto, uma corrente flui pelo diodo do optoacoplador, o seu fototransistor permite que a corrente de base do transistor NPN flua e coloque-o em condução. Assim o transistor NPN opera como uma chave fechada, permitindo que flua corrente pela bobina do relé e pelo LED, acionando ambos.

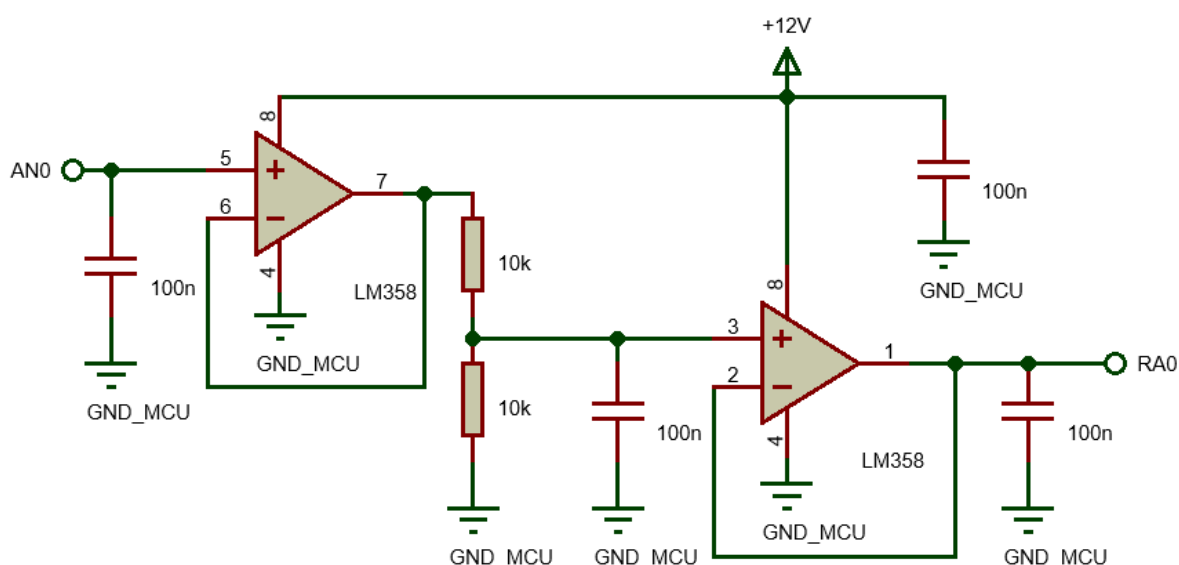
Se o pino de saída digital do microcontrolador estiver em nível lógico baixo, o fototransistor do optoacoplador e o transistor NPN operam na região de corte, portanto se comportam como uma chave aberta, impedindo a passagem de corrente e, conseqüentemente, impedindo o acionamento do relé e do LED.

4.5 Entradas analógicas

As entradas analógicas deste controlador lógico programável utilizam o conversor analógico digital interno do microcontrolador. Estas entradas devem medir sinais de 0 até 10 V, porém como o conversor analógico digital do microcontrolador só é capaz de medir sinais entre 0 e 5 V, se fez necessário desenvolver um circuito condicionador de sinal, capaz de converter o sinal de entrada para o nível de tensão compatível com o microcontrolador.

Este circuito condicionador de sinal é composto por dois resistores de 10 k Ω , dois amplificadores operacionais inclusos no circuito integrado LM358 e quatro capacitores de 100 nF, conforme apresentado na Figura 23.

Figura 23 - Circuito de entrada analógica



Fonte: Autoria Própria.

Neste circuito condicionador de sinal, um sinal com amplitude de 0 a 10 V é aplicado na entrada não-inversora de um amplificador operacional (LM358) conectado como seguidor de tensão (*buffer*). Na saída deste *buffer*, tem-se um sinal de mesma amplitude chegando ao divisor de tensão formado pelos dois resistores de mesmo valor (10 k Ω). Após o divisor de tensão, o sinal passa a ter metade de sua amplitude anterior e então chega a outro amplificador operacional, também conectado como seguidor de tensão, que fornece em sua saída, um sinal que vai até o pino de entrada analógica do microcontrolador com o nível de tensão adequado. Capacitores de desacoplamento foram conectados próximos à alimentação,

entradas não-inversoras dos amplificadores operacionais e entrada analógica do microcontrolador, a fim de filtrar possíveis ruídos.

O conversor analógico digital do microcontrolador tem uma resolução de 10 *bits*, portanto existem 1024 possíveis valores de leitura analógica. Como o sinal que será aplicado na entrada analógica do CLP varia entre 0 e 10 V, a resolução em Volts pode ser calculada conforme a Equação 2:

$$Resolução_{Volts} = \frac{V_{ref+} - V_{ref-}}{2^{n_{bits}-1}} \quad (2)$$

Substituindo os valores $V_{ref+} = 10\text{ V}$, $V_{ref-} = 0\text{ V}$ e $n_{bits} = 10$ na Equação 2, tem-se a Equação 3:

$$Resolução_{Volts} = \frac{V_{ref+} - V_{ref-}}{2^{n_{bits}-1}} = \frac{10-0}{2^{10-1}} = \frac{10}{2^9} = \frac{10}{1023} = 9,7752\text{ mV} \quad (3)$$

A resolução em Volts representa a mínima variação de tensão necessária para alterar o valor obtido da leitura do conversor A/D. O resultado da conversão é armazenado como um valor de 10 *bits* nos registradores ADRESH e ADRESL do microcontrolador. A partir deste valor, é possível obter o valor da tensão aplicada na entrada analógica do dispositivo, multiplicando o valor de resolução em Volts pelo valor de 10 *bits* obtido como resultado da conversão A/D, como mostra a Equação 4:

$$V_{entrada\ analógica} = Resultado_{Conversão\ \frac{A}{D}} * Resolução_{Volts} \quad (4)$$

4.6 Saídas analógicas

As saídas analógicas deste CLP se baseiam em um circuito onde um sinal PWM (modulação por largura de pulso), gerado pelo microcontrolador, passa por um filtro passa-baixas, formado por um resistor e um capacitor, resultando em um sinal analógico de tensão. A amplitude deste sinal depende da amplitude e do *duty cycle* do sinal PWM, conforme representado na Equação 5.

$$Tensão\ DAC = (Amplitude\ PWM) * (duty\ cycle\ PWM\ \%) \quad (5)$$

Sabe-se que este sinal analógico resultante possui *ripple*, que faz com que a amplitude do sinal varie a cada período do sinal PWM. Uma forma de reduzir este efeito, é a redução da frequência de corte do filtro passa-baixas, porém o tempo de resposta do sinal é aumentado devido ao aumento de impedância do circuito. Outra solução é o aumento da frequência do sinal PWM, mas o módulo CCP do

microcontrolador tem limitações. A resolução máxima deste módulo é 10 *bits*, porém quanto maior a frequência do sinal PWM, menor é a resolução. A Equação 6 demonstra esta relação.

$$Resolução\ PWM = \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)}\ bits \quad (6)$$

Para um microcontrolador com uma frequência de operação a 20MHz, o maior valor de frequência do sinal PWM, com uma resolução de 10 *bits*, seria de 19,53 kHz, mas como esta é uma frequência baixa, seria necessária uma grande redução na frequência de corte do filtro e conseqüentemente aumentaria o tempo de resposta do sinal.

Sabendo que o valor de *duty cycle* pode ser visto como uma porcentagem do período do sinal PWM, optou-se por elevar a frequência até o ponto em que a faixa de valores possíveis do *duty cycle* seja de 0 a 100. Assim é possível variar a tensão a um passo de 1% da tensão máxima, o que significa um passo de 50 mV para uma tensão máxima de 5 V ou um passo de 100 mV para uma tensão máxima de 10 V.

Sabendo que o valor máximo de *duty cycle*, em segundos, é igual ao período do sinal PWM, podemos encontrar a frequência necessária para obter a faixa de valores desejada, utilizando as equações 7, 8 e 9, a seguir.

$$Duty\ cycle\ PWM = (CCPR1L:CCP1CON < 5:4 >) * T_{osc} * (TMR2\ Prescale) \quad (7)$$

$$T_{osc} = \frac{1}{F_{osc}}; \quad (8)$$

$$Freq.PWM = \frac{1}{Período\ PWM}; \quad (9)$$

Manipulando estas equações, tem-se a Equação 10:

$$Freq.PWM = \frac{F_{osc}}{(CCPR1L:CCP1CON < 5:4 >) * (TMR2\ Prescale)} \quad (10)$$

Os bits CCPR1L:CCP1CON<5:4> especificam o *duty cycle*, portanto para uma faixa de valores de 0 até 100, o valor armazenado deve ser igual a 100 quando o *duty cycle* é igual ao período do sinal PWM. Utilizando um *prescale* 1:1 no Timer2 e sabendo que a frequência de operação do microcontrolador é igual a 20 MHz, tem-se a Equação 11.

$$Freq.PWM = \frac{20 * 10^6}{(100) * (1)} = 200\ kHz \quad (11)$$

Portanto, para uma faixa de valores de *duty cycle* de 0 até 100, a frequência do sinal PWM deve ser igual a 200 kHz. Então a frequência de corte do filtro passa

baixas foi definida como igual a 590 Hz, buscando o equilíbrio entre atenuação do *ripple* e um tempo de resposta aceitável. Definindo o valor do capacitor em 100 nF, tem-se a Equação 12:

$$R = \frac{1}{2\pi f_c C} = \frac{1}{2\pi * 590 * 100 * 10^{-9}} = 2697,54 \Omega \cong 2700 \Omega \quad (12)$$

Desta forma, o filtro passa baixas é formado por um resistor de 2k7 Ω e um capacitor de 100 nF. O sinal analógico resultante varia entre 0 e 5 V, porém as saídas analógicas de CLPs industriais geralmente operam em uma faixa de 0 a 10 V, então se faz necessário amplificar este sinal.

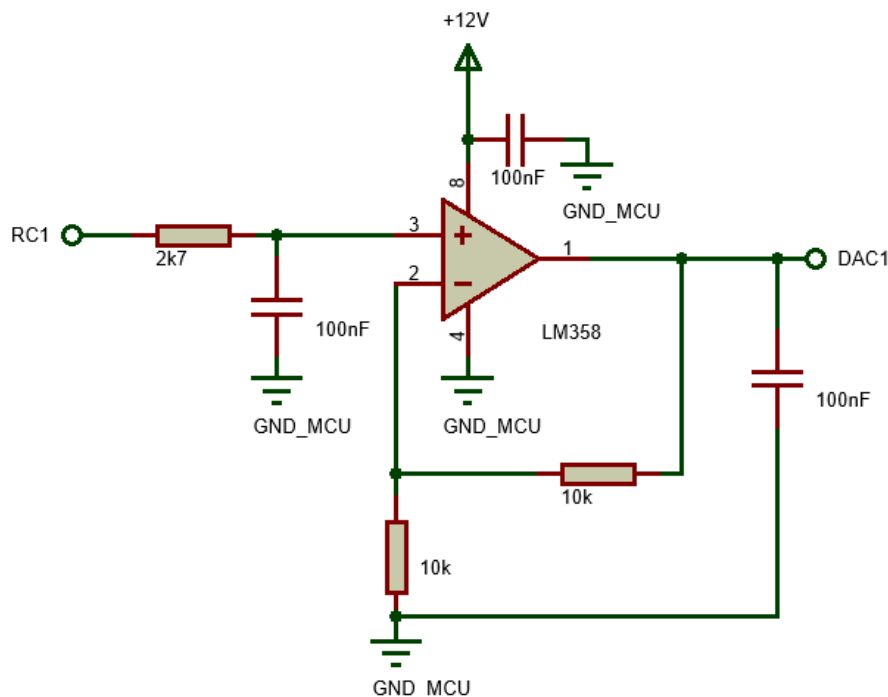
A amplificação do sinal é obtida colocando, logo após o filtro, um amplificador operacional (LM358) configurado como amplificador não-inversor com um ganho igual a 2, como pode ser observado na Figura 23. Para definir o ganho do amplificador, é preciso atender a equação de ganho do amplificador não-inversor apresentada na Equação 13.

$$Ganho = 1 + \frac{R3}{R2} \quad (13)$$

Para um ganho igual a 2, são necessários dois resistores de mesmo valor, então definiu-se que ambos os resistores tenham um valor de 10 k Ω . Recomenda-se o uso de resistores de precisão para que o divisor de tensão seja mais preciso. Na saída do amplificador, foi conectado um capacitor de 100 nF com a função de atenuar o *ripple* do sinal analógico de saída. Também foi utilizado um capacitor de desacoplamento de 100 nF na alimentação do amplificador operacional.

A Figura 24 apresenta o circuito de saída analógica desenvolvido, com o filtro passa baixas formado pelo resistor de 2k7 Ω e um capacitor de 100 nF, o amplificador operacional, configurado pelos dois resistores de 10 k Ω , um capacitor de 100 nF conectado à saída do amplificador e ainda um capacitor de desacoplamento.

Figura 24 - Circuito de saída analógica



Fonte: Autoria Própria.

Foram projetados dois circuitos de saída analógica, um conectado ao pino RC1 e outro ao pino RC2, utilizando os dois módulos CCP (*Capture, Compare, PWM*) do microcontrolador.

Para evitar que a carga opere com uma corrente maior do que o limite máximo de corrente de saída do amplificador operacional, é necessário definir uma impedância de carga mínima. Sabendo que o LM358 é capaz de fornecer uma corrente típica de 20 mA na sua saída e que o valor máximo de tensão na saída será 10 V, calcula-se a impedância mínima de carga utilizando a Equação 14:

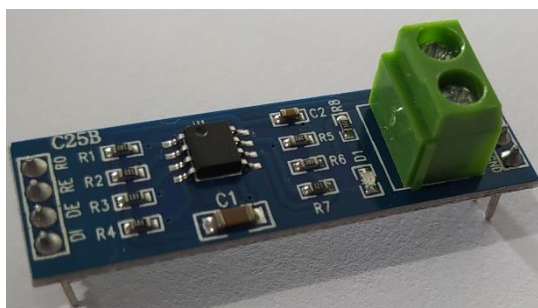
$$Z_{L_{min}} = \frac{V_{max}}{I_{typ}} = \frac{10}{20 \times 10^{-3}} = 500 \Omega \quad (14)$$

4.7 Comunicação

O módulo de comunicação desenvolvido é capaz de realizar a comunicação do microcontrolador através das interfaces RS-485 e RS-232, com a opção de seleção entre os *transceivers*. Foram utilizados módulos comerciais para a conversão dos sinais dos padrões RS-485 e RS-232 em sinais do padrão TTL.

O módulo RS-485 converte um sinal padrão TTL em um sinal padrão RS-485 e vice-versa. Possui comunicação *half-duplex*, ou seja, permite a comunicação bilateral, porém não ao mesmo tempo. Seu circuito é basicamente formado por um circuito integrado MAX485 e por resistores e capacitores que garantem o correto funcionamento deste circuito integrado. A Figura 25 mostra o módulo RS-485 utilizado.

Figura 25 - Módulo conversor RS-485



Fonte: Aatoria Própria.

O módulo possui oito terminais para a operação do módulo, os quais são coincidentes com os pinos do circuito integrado. A seguir são apresentadas as funções de cada pino, baseadas no *datasheet* do componente MAX485 (MAXIM INTEGRATED PRODUCTS INC, 2014).

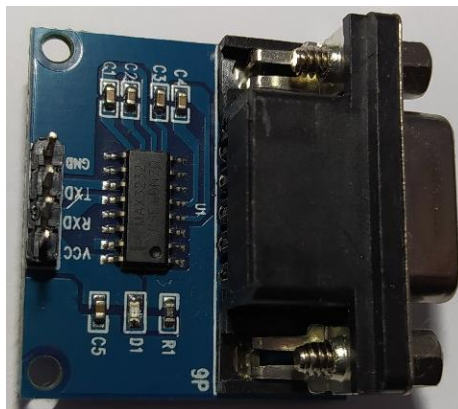
- **RO (*Receiver Output*):** Quando A é 200 mV maior que B, este pino assume nível lógico alto. Já quando A é 200 mV menor que B, assume nível lógico baixo. Deve ser conectado ao pino RX do microcontrolador.
- **RE (*Receiver Output Enable*):** É um *bit* de controle do pino RO. Quando está em nível lógico baixo, habilita o pino RO, já quando está em nível lógico alto, RO assume uma alta impedância.
- **DE (*Driver Output Enable*):** Se estiver em nível lógico alto, A e B funcionam como transmissores de dados. Quando está em nível lógico baixo, A e B se tornam de alta impedância, funcionando como receptores de dados caso RE esteja em nível lógico baixo.
- **DI (*Driver Input*):** Sua operação depende que o pino DE esteja em nível lógico alto. Se DI estiver em nível lógico baixo, A assume nível lógico baixo e assume nível lógico baixo. Um nível lógico baixo em DI induz nível lógico alto em A e nível lógico baixo em B. Deve ser conectado ao pino TX do microcontrolador.

- **A:** Entrada receptora não inversora e saída transmissora não inversora.
- **B:** Entrada receptora inversora e saída transmissora inversora.
- **VCC e GND:** Pinos de alimentação.

Como o pino RE opera com lógica inversa (NOT), os pinos RE e DE podem ser colocados em curto, desta forma quando estes pinos estão em nível lógico alto, o módulo opera enviando dados e quando estão em nível lógico baixo, o módulo opera recebendo dados. O pino RE0 do microcontrolador foi escolhido para controlar o nível lógico dos pinos RE e DE, assim controlando o sentido da comunicação do módulo RS-485. A e B são os terminais que conectam este módulo a *transceivers* de outros dispositivos.

O módulo RS-232 é capaz de converter um sinal padrão TTL em um sinal padrão RS-232 e vice versa. Permite o envio e recepção de dados ao mesmo tempo, ou seja, apresenta comunicação *full-duplex*. Um circuito integrado MAX3232 e alguns capacitores auxiliares formam o circuito responsável pela conversão dos sinais. A Figura 26 mostra o módulo RS-232 adquirido.

Figura 26 - Módulo conversor RS-232



Fonte: Autoria Própria.

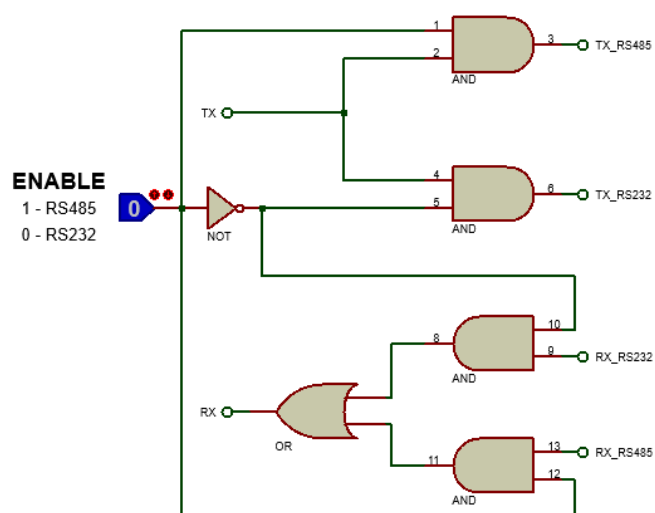
Os pinos disponíveis para conexões externas são apresentados e detalhados a seguir, baseando-se no datasheet do circuito integrado MAX3232 (TEXAS INSTRUMENTS, 2021).

- **T1OUT:** Envio dos dados, recebidos no pino T1IN, convertidos em padrão RS-232. Conectado ao pino 2 do conector DB9 fêmea;
- **R1IN:** Recepção de dados em formato RS-232, que serão convertidos em padrão TTL. Conectado ao pino 3 do conector DB9 fêmea;

- **R1OUT**: Envio dos dados, recebidos no pino R1IN, convertidos em padrão TTL. Conectado ao pino RXD do módulo, conecta-se ao pino RX da UART do microcontrolador;
- **T1IN**: Recebe, em formato TTL, os dados que serão enviados em formato RS-232 pelo pino T1OUT. Conectado ao pino TXD do módulo, deve ser conectado ao pino TX da UART do microcontrolador;
- **VCC**: Alimentação em 3.3 ou 5 V, conectado ao pino VCC do módulo;
- **GND**: Terra, conectado ao pino 5 do conector DB9 e ao pino GND do módulo.

O circuito de seleção entre os *transceivers* RS-232 e RS-485 foi desenvolvido com portas lógicas AND, NOT e OR, organizadas conforme a Figura 27.

Figura 27 - Circuito de seleção com portas lógicas



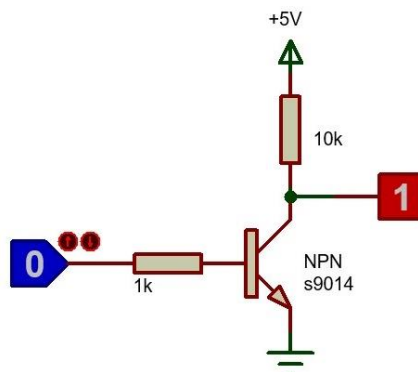
Fonte: Autoria Própria.

Neste circuito de portas lógicas, foi utilizado um *bit* de controle para definir qual módulo de comunicação estará ativo. Quando o *bit* estiver em nível lógico baixo, o módulo RS-232 estará ativo e o módulo RS-485 inativo. Já quando estiver em nível lógico alto, o módulo RS-458 estará ativo e o módulo RS-232 inativo.

Devido à dificuldade de encontrar circuitos integrados com um número menor de portas lógicas, nas portas lógicas NOT e OR, onde somente é necessária uma porta lógica de cada, foram utilizados circuitos lógicos formados por transistores.

A Figura 28 apresenta uma alternativa simples à porta lógica NOT, utilizando um transistor NPN S9014, um resistor de 1 k Ω e um resistor de 10 k Ω .

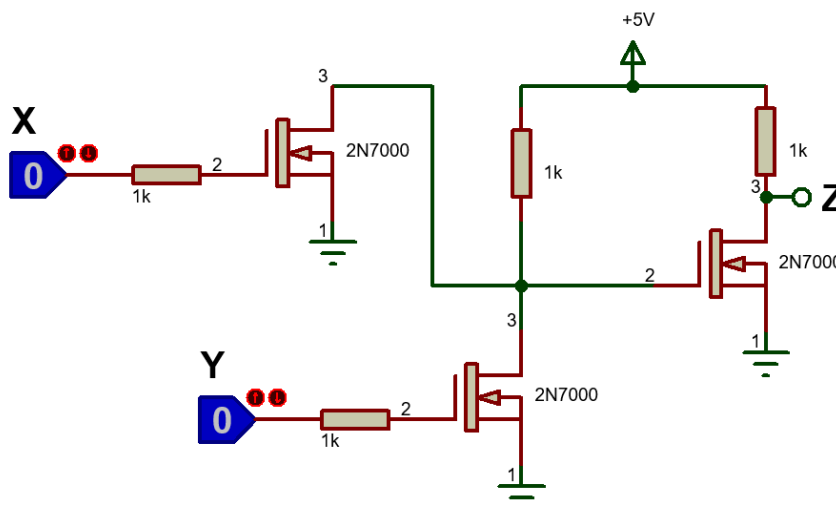
Figura 28 - Alternativa com transistor a uma porta lógica NOT



Fonte: Autoria Própria.

Como alternativa a uma porta lógica OR, pode-se utilizar três transistores 2N7000 e quatro resistores de 1 k Ω , conectados conforme o circuito apresentado na Figura 29. Nesta figura, as entradas da porta lógica estão identificadas como X e Y, já a saída está identificada pela letra Z.

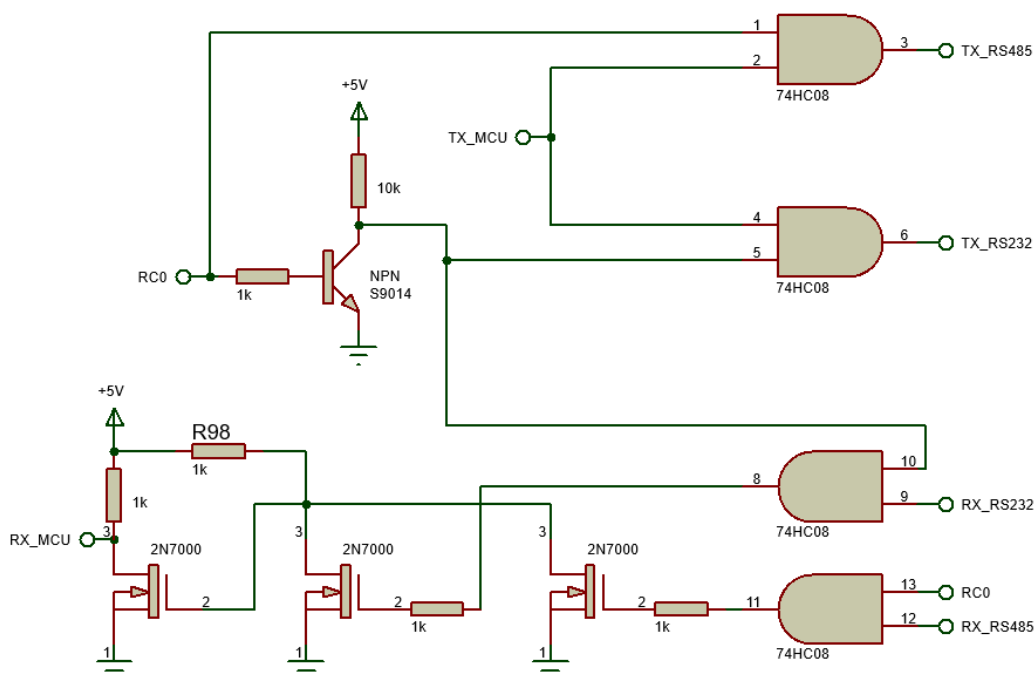
Figura 29 - Alternativa com transistores a uma porta lógica OR



Fonte: Autoria Própria.

Substituindo as portas lógicas NOT e OR pelas suas alternativas de circuitos com transistores e utilizando um circuito integrado 74HC08 para fornecer as portas lógicas AND, forma-se o circuito de seleção do módulo de comunicação, apresentado na Figura 30. O pino RC0 do microcontrolador foi definido como o *bit* de seleção do módulo de comunicação.

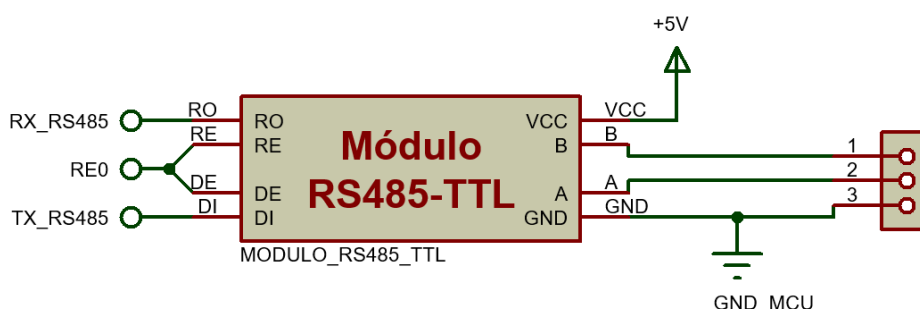
Figura 30 - Circuito de seleção do módulo de comunicação



Fonte: Autoria Própria.

Com o circuito de seleção finalizado, os módulos de comunicação foram conectados a este circuito. A Figura 31 mostra que o módulo RS-485 foi alimentado em 5V entre os pinos VCC e GND, com o pino RO conectado à entrada de uma porta AND (RX_RS485), com os pinos RE e DE curto circuitados e conectados ao pino RE0 do microcontrolador, com o pino DI conectado à saída de uma porta AND (TX_RS485) e os pinos A, B e GND foram conectados a um terminal, onde será feita a conexão com outros dispositivos.

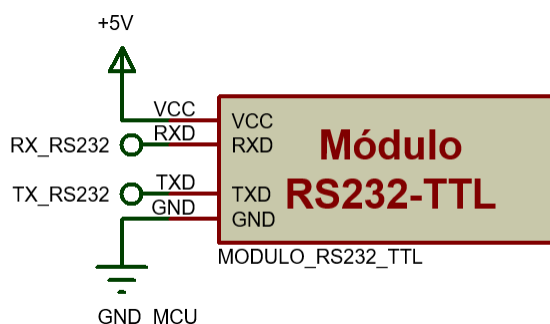
Figura 31 - Conexão do módulo de comunicação RS-485



Fonte: Autoria Própria.

O módulo RS-232 teve seu pino RXD conectado a uma entrada de uma porta lógica AND (RX_RS232) e seu pino TXD conectado à saída de uma porta lógica AND (TX_RS232), também teve sua alimentação em 5 V entre os pinos VCC e GND, conforme a Figura 32.

Figura 32 - Conexão do módulo de comunicação RS-232



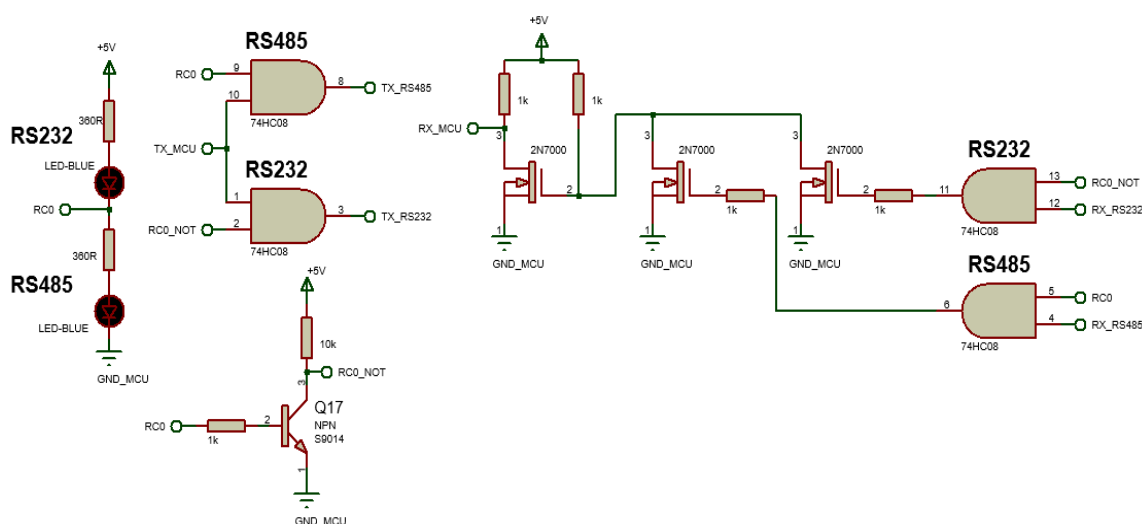
Fonte: Autoria Própria.

Desta forma, a comunicação é controlada por dois pinos de saída digital do microcontrolador:

- **Bit RC0**: Seleção do módulo de comunicação (RS-232 ou RS-485);
- **Bit RE0**: Seleção do modo de operação do módulo RS-485 (recepção ou envio de dados).

Para a sinalização de qual módulo de comunicação está ativo, foi adicionado um circuito simples conectado ao pino RC0 do microcontrolador, com dois LEDs e dois resistores de 360 Ω . A Figura 33 apresenta o circuito completo de comunicação.

Figura 33 - Circuito de comunicação RS-232 e RS-485



Fonte: Autoria Própria.

4.8 Configuração do microcontrolador

Finalizado o desenvolvimento do esquemático do circuito, é preciso listar as funções de cada pino utilizado do microcontrolador, a fim de facilitar o entendimento

do circuito e a programação deste controlador lógico programável. O Quadro 2 lista e associa os pinos do microcontrolador às suas funções no circuito elaborado.

Quadro 2 - Funções dos pinos do microcontrolador neste dispositivo

Pino		Função neste dispositivo
Nº	Identificador	
1	MCLR/VPP	Reset
2	RA0/AN0	Entrada Analógica AN0
3	RA1/AN1	Entrada Analógica AN1
4	RA2/AN2	Entrada Analógica AN2
5	RA3/AN3	Entrada Analógica AN3
6	RA4	Não utilizado
7	RA5/AN4	Não utilizado
8	RE0/AN5	Seleção do modo de operação do módulo RS-485
	RE1/AN6	Entrada Digital I8
10	RE2/AN7	Entrada Digital I9
11	VDD	Alimentação
12	VSS	Alimentação
13	OSC1	Oscilador
14	OSC2	Oscilador
15	RC0	Seleção do módulo de comunicação
16	RC1/CCP2	Saída Analógica 1
17	RC2/CCP1	Saída Analógica 2
18	RC3	Não utilizado
19	RD0	Saída Digital Q0
20	RD1	Saída Digital Q1
21	RD2	Saída Digital Q2
22	RD3	Saída Digital Q3
23	RC4	Não utilizado
24	RC5	Não utilizado
25	RC6/TX	Comunicação (TX)
26	RC7/RX	Comunicação (RX)
27	RD4	Saída Digital Q4
28	RD5	Saída Digital Q5
29	RD6	Saída Digital Q6
30	RD7	Saída Digital Q7
31	VSS	Alimentação
32	VDD	Alimentação
33	RB0	Entrada Digital I0
34	RB1	Entrada Digital I1
35	RB2	Entrada Digital I2
36	RB3	Entrada Digital I3
37	RB4	Entrada Digital I4
38	RB5	Entrada Digital I5
39	RB6/PGC	Entrada Digital I6
40	RB7/PGD	Entrada Digital I7

Fonte: Autoria Própria

Esta relação deve ser utilizada durante a elaboração das lógicas de programação que serão gravadas no dispositivo.

4.9 Projeto das placas de circuito impresso

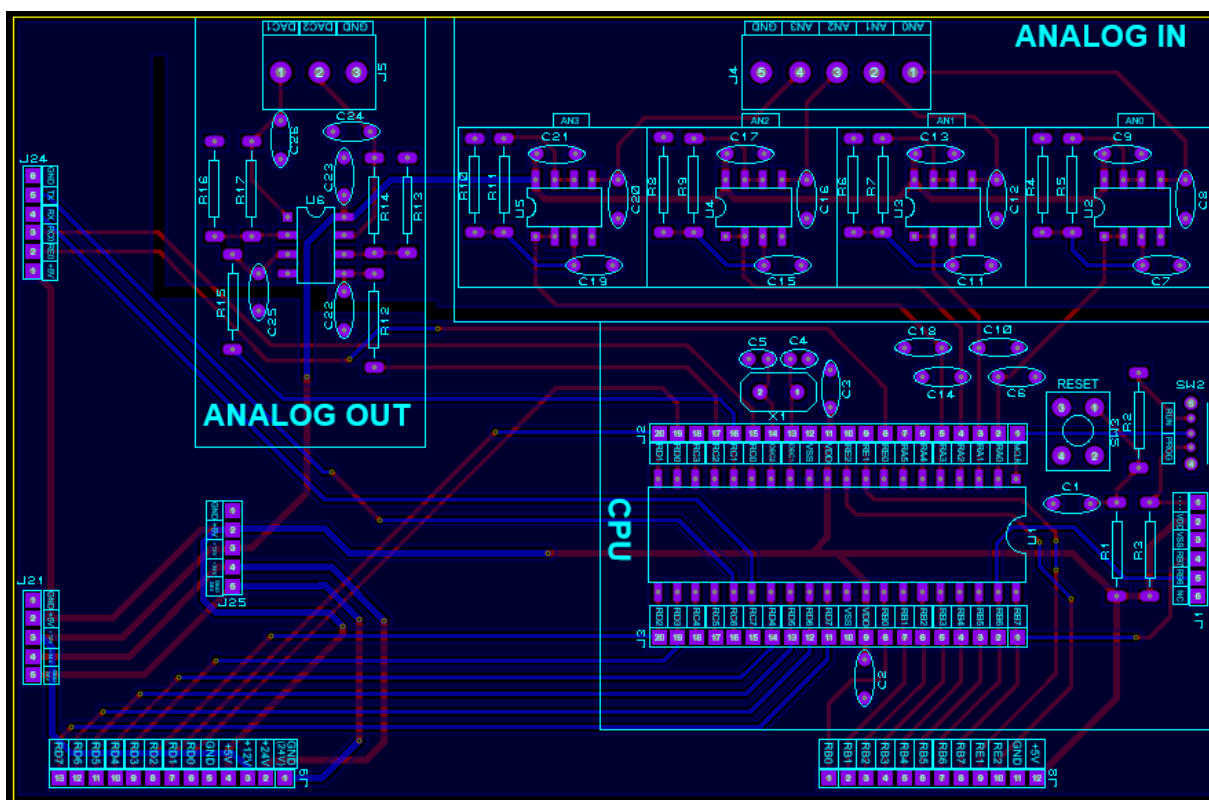
Após o desenvolvimento do esquemático do circuito do controlador lógico programável, foi elaborado o *layout* das placas de circuito impresso, utilizando o *software* PROTEUS. O circuito foi dividido em cinco módulos: módulo de alimentação, módulo principal, módulo de entradas digitais, módulo de saídas digitais e módulo de comunicação. Essa divisão em módulos teve como objetivo tornar o dispositivo mais versátil, facilitando a identificação de erros, a manutenção e a reposição de placas eletrônicas danificadas, além de oferecer a possibilidade de desenvolvimento de novos módulos compatíveis com o dispositivo.

As placas foram projetadas com núcleo de material FR-4, dupla face de cobre, furos metalizados e máscara de solda. Na camada inferior foram projetados planos de terra afim de reduzir a impedância de terra, reduzindo ruídos e melhorando a qualidade do sinal. Seguindo as orientações presentes em Microchip Technology Inc. (2005), os componentes foram posicionados de forma que os componentes analógicos e digitais fiquem separados, na medida do possível, assim como os componentes que operam em alta frequência foram mantidos afastados dos componentes que operam em baixa frequência, a fim de reduzir possíveis ruídos e interferências entre os circuitos.

O formato das placas e o posicionamento de conectores e terminais foram projetados de modo que os módulos de alimentação, de comunicação, de entradas digitais e de saídas digitais sejam conectados diretamente à placa principal.

O módulo principal é formado pelo circuito do microcontrolador, pelos circuitos de entradas e saídas analógicas, junto com os conectores utilizados na gravação e conexão com os outros módulos do dispositivo. As entradas e saídas analógicas foram mantidas na mesma placa do microcontrolador com o intuito de reduzir os erros de medição. Nesta placa, o plano de terra dos circuitos analógicos se interliga ao plano de terra do restante do circuito em um local próximo à conexão do módulo de alimentação, a fim de evitar possíveis ruídos nos sinais analógicos. A Figura 34 apresenta o *layout* da placa principal.

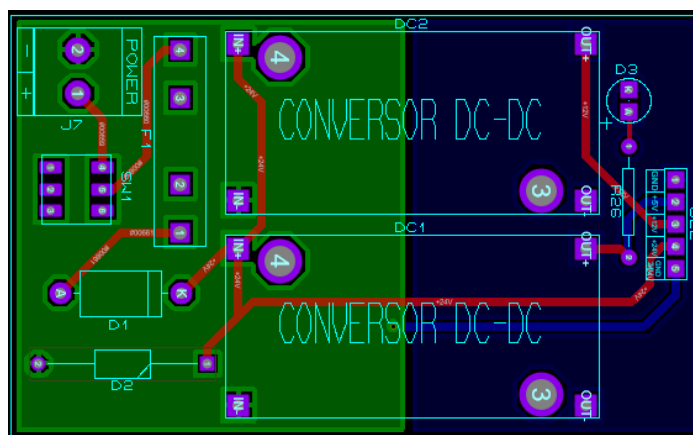
Figura 34 - *Layout* do módulo principal



Fonte: Autoria Própria.

O módulo de alimentação é formado inteiramente pelo circuito de alimentação apresentado anteriormente. Sua placa foi projetada com o terminal de alimentação em 24 V posicionado próximo à borda da placa e ao botão de liga e desliga, e também com pinos para a conexão com o módulo principal. A Figura 35 apresenta o *layout* da placa de alimentação.

Figura 35 - *Layout* do módulo de alimentação

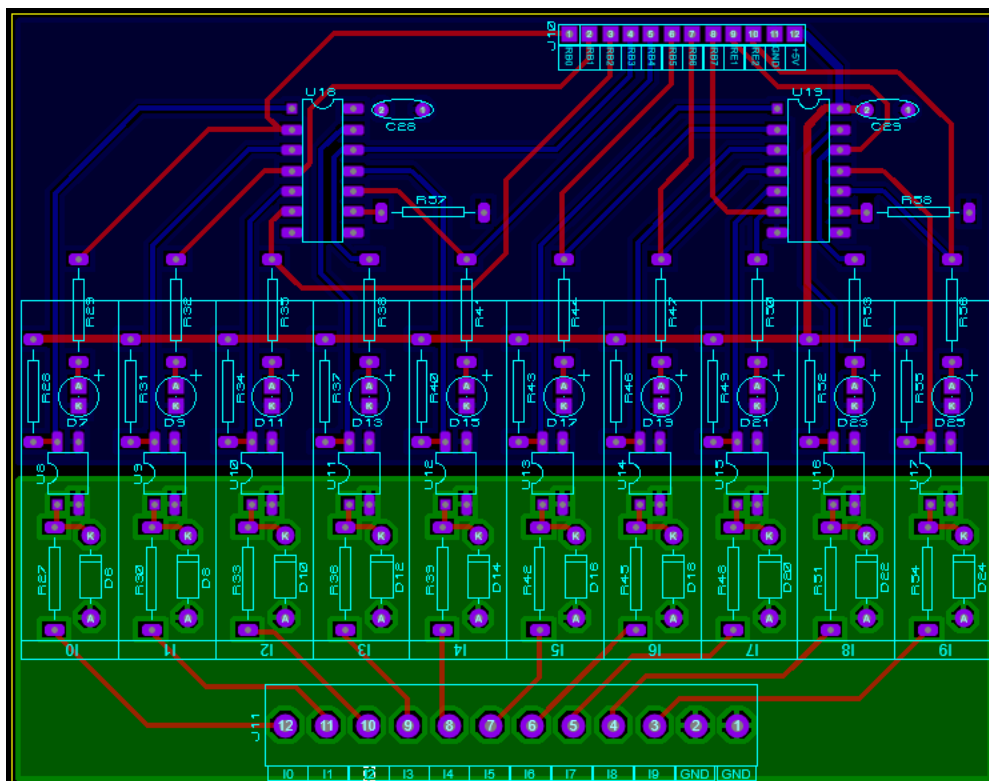


Fonte: Autoria Própria.

O módulo de entradas digitais é formado pelo circuito de dez entradas digitais, com um conector para a conexão com a placa principal e um terminal para a

conexão com os dispositivos de entrada. O plano de terra do circuito que será conectado aos sinais dos dispositivos de entrada, é isolado do plano de terra do circuito alimentado em 5 V. A Figura 36 apresenta o *layout* da placa de entradas digitais.

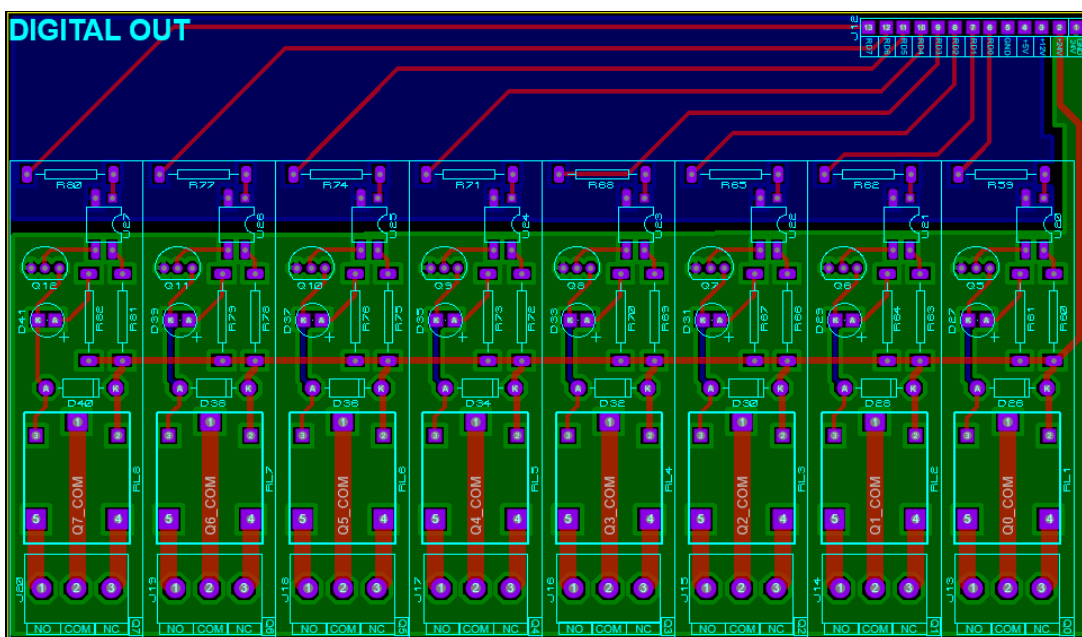
Figura 36 - Layout do módulo de entradas digitais



Fonte: Autoria Própria.

A placa de saídas digitais acomoda os oito circuitos de saídas digitais a relé, com os pinos de conexão com a placa principal e oito terminais para a conexão com os contatos COM, NO e NC dos relés de cada saída, conforme a Figura 37. O plano de terra do circuito alimentado em 24 V é separada do plano de terra do circuito alimentado em 5 V.

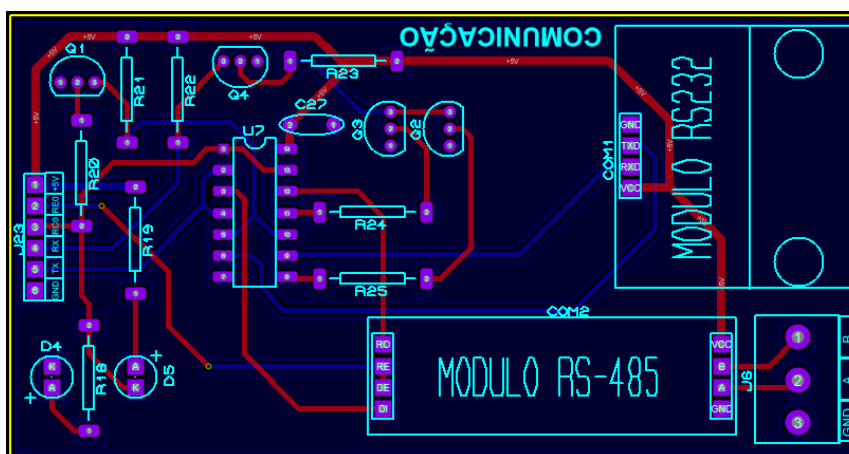
Figura 37 - Layout do módulo de saídas digitais



Fonte: Autoria Própria.

O módulo de comunicação é formado por todo o circuito de comunicação, incluindo o circuito de seleção, os *pads* para soldagem dos módulos RS-232 e RS-485, junto com os pinos de conexão com a placa principal e um terminal para a conexão dos fios de comunicação RS-485 com outros dispositivos. No caso da comunicação RS-232, foi utilizado o conector DB9 fêmea existente no próprio módulo. A Figura 38 mostra o *layout* da placa de comunicação.

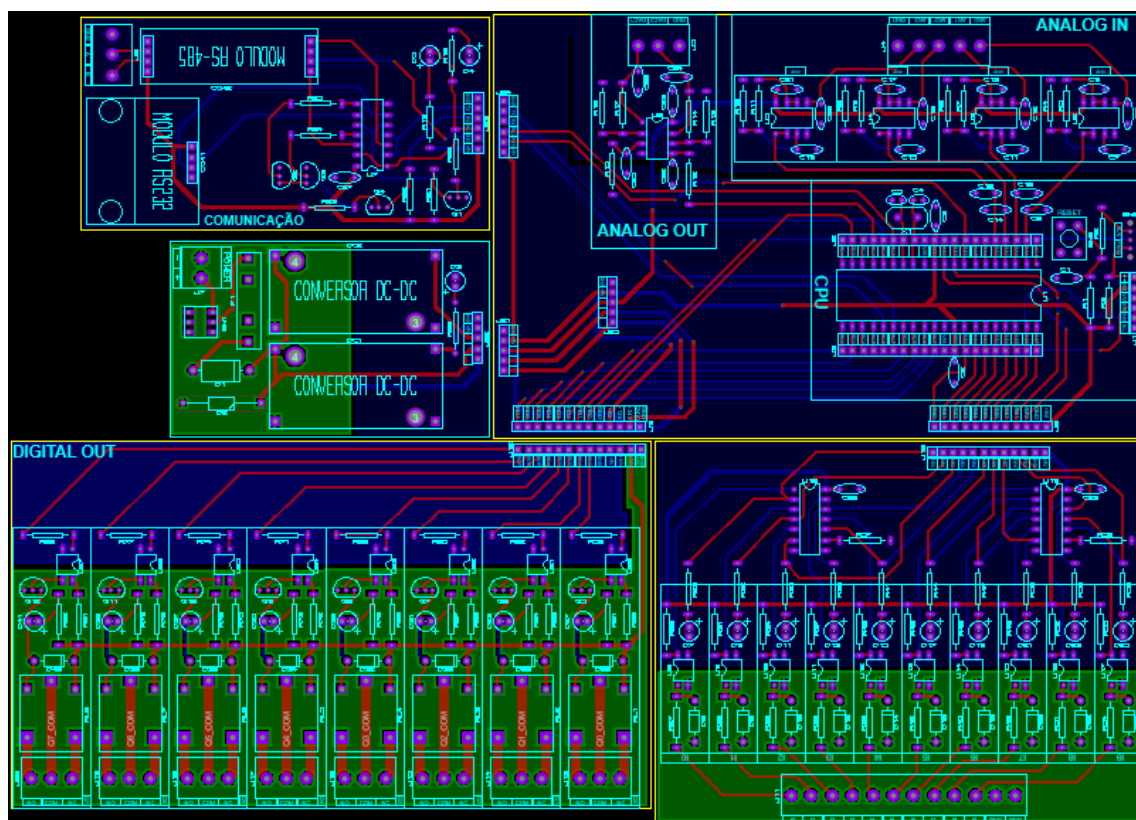
Figura 38 - Layout do módulo de comunicação



Fonte: Autoria Própria.

Encaixando todos os módulos, a placa principal foi centralizada, com os módulos de entradas digitais e de saídas digitais posicionados abaixo dela e os módulos de alimentação e de comunicação posicionados à esquerda dela, conforme a Figura 39.

Figura 39 - *Layout* do dispositivo



Fonte: Autoria Própria.

Assim que o projeto das placas de circuito impresso foi finalizado, foi feita a encomenda da produção das placas em uma empresa especializada.

4.10 Construção do dispositivo

Logo após a listagem dos componentes, a qual está disponível no Apêndice A, foi feita a aquisição dos mesmos. Com as placas e os componentes em mãos, o projeto foi posto em prática, com a soldagem dos componentes e montagem do controlador lógico programável. Os circuitos de entrada digital e de saída digital são circuitos que foram replicados diversas vezes em seus devidos módulos e não dependem de todos eles para funcionar. Portanto, visando poupar tempo na construção do dispositivo, foram soldados apenas os componentes necessários para o funcionamento de metade da quantidade de circuitos de cada módulo.

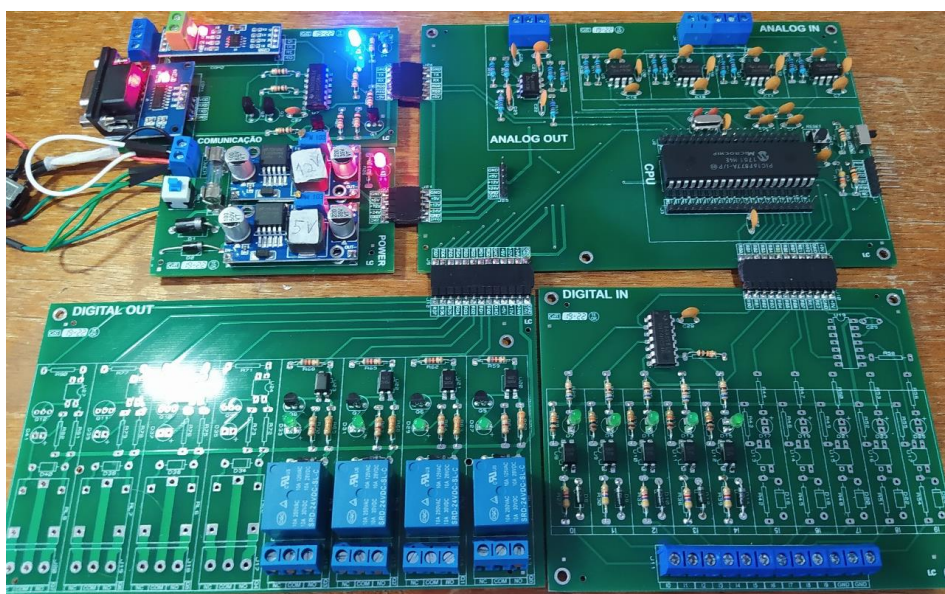
5 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados e analisados os resultados obtidos a partir dos procedimentos de desenvolvimento do dispositivo e também será apresentada a avaliação de desempenho do dispositivo.

5.1 Controlador lógico programável

A montagem e soldagem dos componentes nas placas projetadas resultaram em um controlador lógico programável, apresentado na Figura 40.

Figura 40 - Controlador lógico programável fabricado



Fonte: Autoria Própria.

Como citado anteriormente, os módulos de entradas digitais e de saídas digitais não foram montados por completo, a fim de economizar tempo. Porém a parte montada não depende dos componentes faltantes e funciona normalmente.

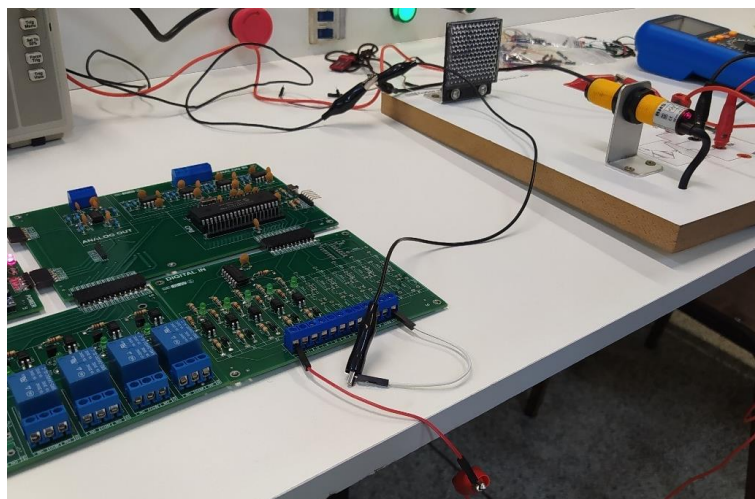
O custo total para a produção de uma unidade deste controlador lógico programável é de aproximadamente 290 reais, um valor bem abaixo dos controladores lógicos programáveis disponíveis no mercado.

5.2 Detecção de sinais nas entradas digitais

No teste com o sensor óptico com saída do tipo PNP, com seu contato normalmente aberto conectado à entrada digital, quando foi colocado um obstáculo à frente do sensor, a entrada digital detectou um sinal de entrada, o qual não foi detectado quando não havia nenhum obstáculo entre o sensor e a superfície refletora, conforme as Figuras 41 e 42.

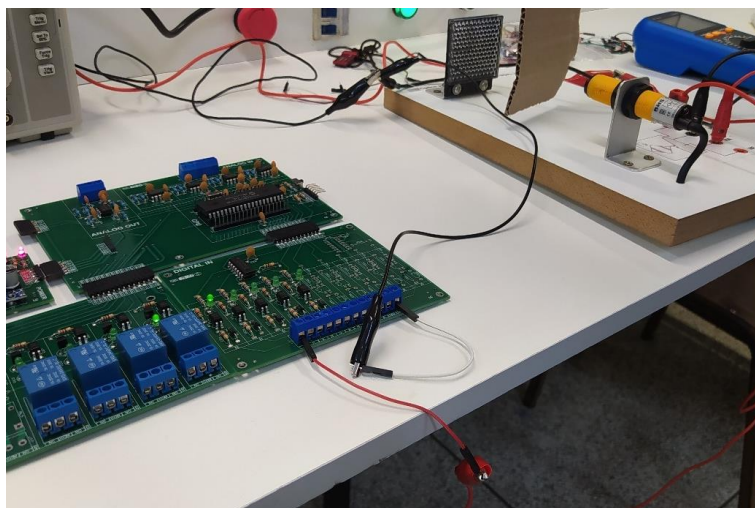
Para testar as entradas e saídas digitais, foi implementado um algoritmo que faz a leitura da entrada e reproduz o estado na saída correspondente, como pode ser observado nas Figuras 41 e 42, onde o estado do LED da entrada I0 e saída Q0 são iguais. Este comportamento foi observado nos testes de todas as entradas digitais montadas.

Figura 41 - Teste sem obstáculo à frente do sensor óptico



Fonte: Autoria Própria.

Figura 42 - Teste com obstáculo à frente do sensor óptico



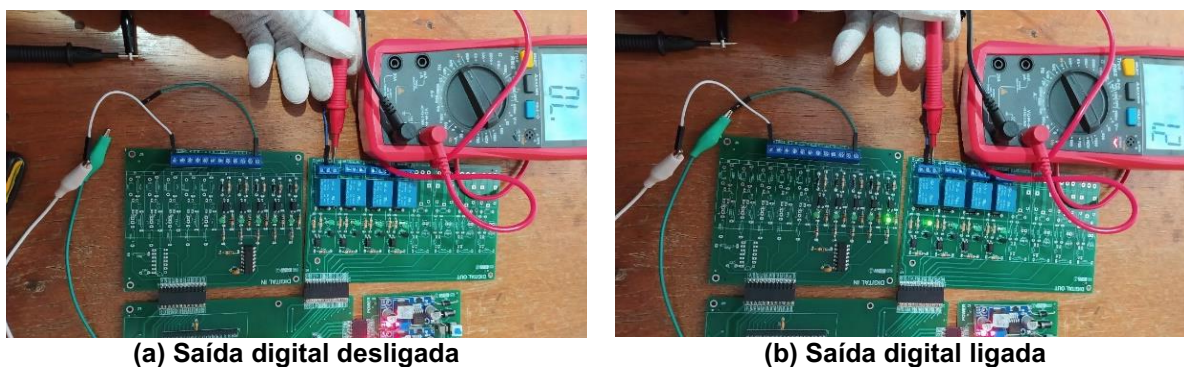
Fonte: Autoria Própria.

Em todos os testes nas entradas digitais, o dispositivo se comportou conforme esperado, reconhecendo os sinais aplicados e assim comprovando o correto funcionamento dos circuitos de entradas digitais.

5.3 Acionamento das saídas digitais

Os testes de continuidade entre os contatos do relé de uma das saídas foram realizados em dois momentos, primeiro com a saída inativa e depois com a saída ativa, conforme as Figura 43.

Figura 43 - Teste de continuidade do contato NA



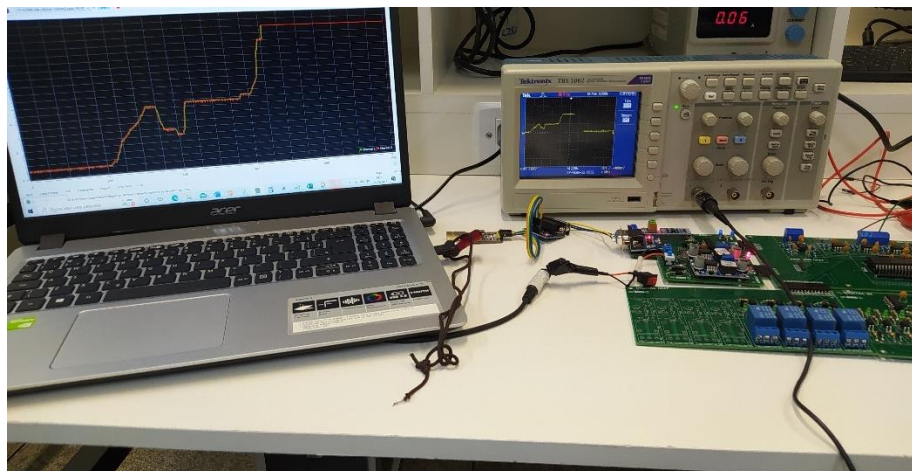
Fonte: Autoria Própria.

Neste teste, o multímetro indicou continuidade apenas quando a saída digital estava ativa. Nos testes de continuidade do contato normalmente fechado, o comportamento foi inverso ao do contato normalmente aberto.

5.4 Leitura de sinais analógicos

Um sinal analógico foi aplicado a uma entrada analógica do CLP e também foi medido por um osciloscópio Tektronix TBS1062. O resultado da leitura analógica realizada pelo controlador lógico programável foi enviada via comunicação serial para o computador e plotada em um gráfico, utilizando o *software* SerialPlot. A Figura 44 mostra o teste de leitura do sinal analógico, com o resultado desta leitura sendo exibido no computador e também mostra o resultado da medição do sinal pelo osciloscópio.

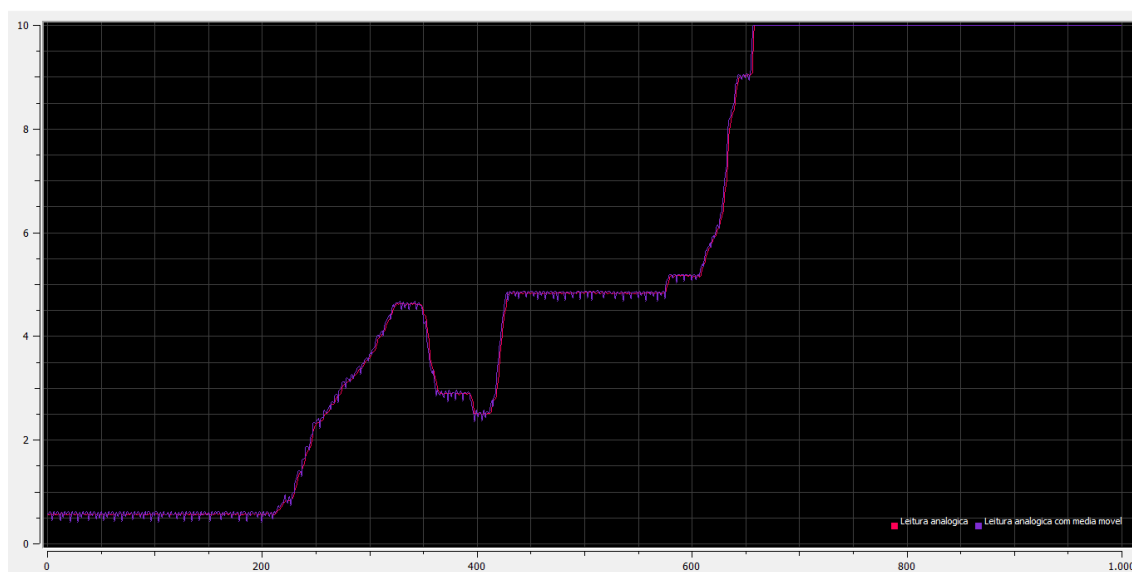
Figura 44 - Teste de leitura de um sinal analógico



Fonte: Autoria Própria.

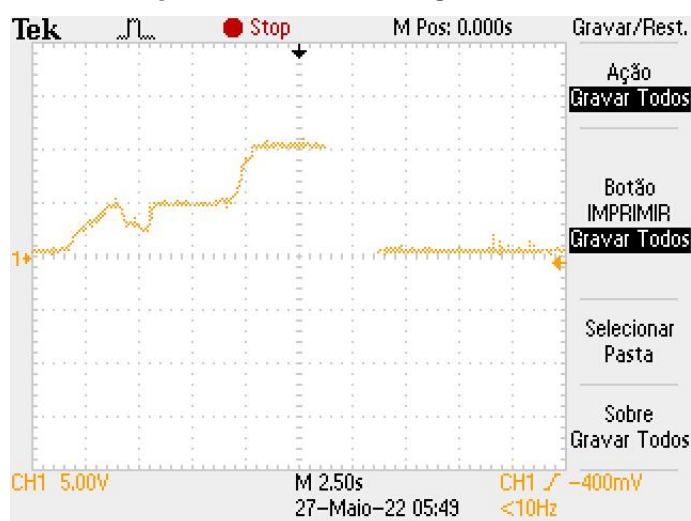
As Figuras 45 e 46 apresentam o sinal analógico plotado no computador, utilizando o *software* SerialPlot, e medido no osciloscópio.

Figura 45 - Plotagem de um sinal analógico no *software* SerialPlot



Fonte: Autoria Própria.

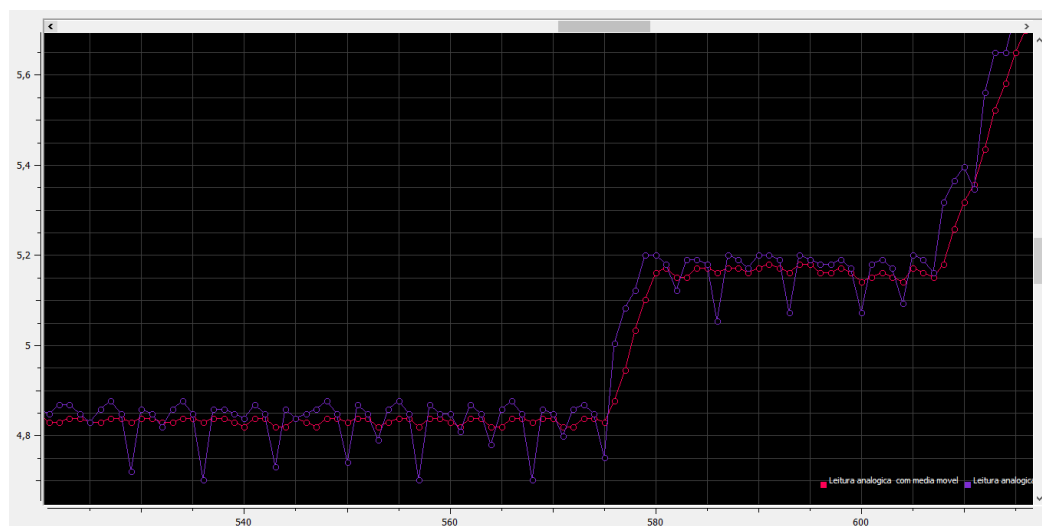
Figura 46 - Medição de um sinal analógico em um osciloscópio



Fonte: Autoria Própria.

Nota-se como os dois gráficos bem semelhantes, o que mostra uma boa leitura do sinal pelo circuito de entrada analógica. Aproximando o gráfico gerado no computador, fica mais clara a diferença entre a leitura analógica pura e a leitura analógica tratada por um filtro de média móvel, conforme apresentado na Figura 47.

Figura 47 - Leitura analógica com e sem filtro de média móvel



Fonte: Autoria Própria.

Na Figura 47, a curva de cor violeta representa a leitura analógica pura e a curva de cor rosa representa a leitura analógica tratada com um filtro de média móvel. Na leitura tratada com um filtro de média móvel, nota-se a redução significativa da variação causada por ruídos no sinal analógico, ao preço de um atraso na leitura do sinal.

Os resultados do teste de precisão da leitura analógica são apresentados na Tabela 1. Junto com os resultados de leitura analógica e medição com multímetro, também foram calculados os erros absolutos e relativos para cada valor medido.

Tabela 1 - Resultados do teste de precisão da leitura analógica

Multímetro	Conversão	Erro		Multímetro	Conversão	Erro	
Volts	Volts	Volts	%	Volts	Volts	Volts	%
0,000	0,010	0,010	9675,17%	4,006	4,018	0,012	0,29%
0,004	0,010	0,006	157,24%	4,502	4,516	0,014	0,31%
0,010	0,020	0,009	93,57%	5,004	5,015	0,011	0,21%
0,051	0,049	0,002	4,73%	5,499	5,513	0,014	0,26%
0,100	0,098	0,002	2,05%	6,008	6,022	0,014	0,22%
0,250	0,244	0,006	2,37%	6,500	6,520	0,020	0,31%
0,500	0,499	0,002	0,37%	7,000	7,019	0,019	0,27%
0,999	0,997	0,002	0,19%	7,510	7,527	0,017	0,22%
1,501	1,496	0,005	0,36%	8,000	8,016	0,016	0,20%
2,004	2,004	0,000	0,00%	8,500	8,514	0,014	0,17%
2,504	2,502	0,002	0,06%	8,990	9,003	0,013	0,14%
3,004	3,011	0,007	0,22%	9,500	9,511	0,011	0,12%
3,499	3,509	0,010	0,29%	10,000	10,000	0,000	0,00%

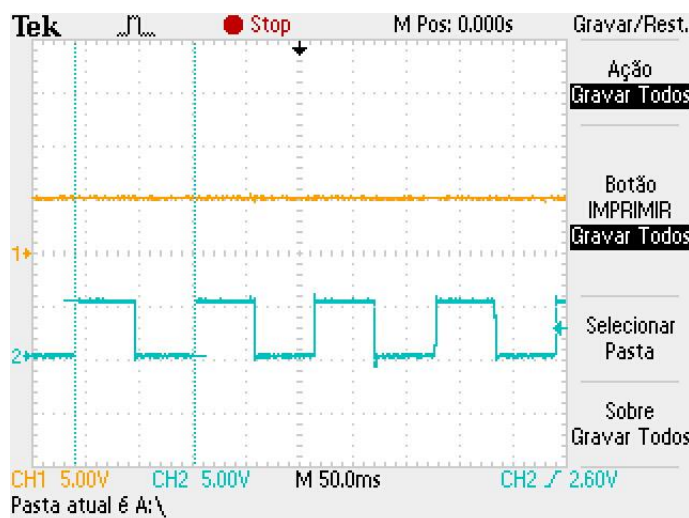
Fonte: Autoria Própria.

Analisando a Tabela 1, percebe-se que para o menor valor de tensão, o erro de medição foi de 9675,17%, por conta da tensão de *offset* existente nos amplificadores operacionais. Porém, para valores na faixa de 50 mV a 250 mV, a máxima variação foi de 4,73%, enquanto que na faixa de 500 mV a 10 V, a maior variação foi de 0,37%. A partir destes resultados, é possível afirmar que o circuito de entrada analógica atendeu as expectativas, uma vez que a tensão de entrada varia entre 0 e 10 V apresentando um erro relativo máximo de 0,37% dentro da faixa de 500 mV a 10 V.

5.5 Geração de sinais analógicos

A Figura 48 apresenta a medição, em um osciloscópio, do sinal da saída analógica DAC1, junto com a medição do sinal PWM gerado no pino RC1.

Figura 48 - Sinal na saída analógica e sinal PWM do microcontrolador



Fonte: Autoria Própria.

A Figura 48 mostra, em azul, um sinal PWM com *duty cycle* de 50%, que resulta em um sinal analógico, em amarelo, de aproximadamente 5 V, que representa metade da amplitude total da saída analógica.

Os resultados do teste de precisão da saída analógica são apresentados na Tabela 2. Para cada valor definido de *duty cycle*, foram obtidos o valor de tensão esperado para este *duty cycle*, o valor de tensão medido por um multímetro e os erros absolutos e relativos entre esses valores.

Tabela 2 - Resultados do teste de precisão da saída analógica

Duty cycle (%)	Teoria	Medição	Erro (0-10)		Duty cycle (%)	Teoria	Medição	Erro (0-10)	
	0-10 (V)	0-10 (V)	Abs.	%		0-10 (V)	0-10 (V)	Abs.	%
0%	0	0,0051	0,0051	--	50%	5	4,972	0,028	0,56%
1%	0,1	0,0994	0,0006	0,60%	55%	5,5	5,47	0,03	0,55%
5%	0,5	0,4922	0,0078	1,56%	60%	6	5,971	0,029	0,48%
10%	1	0,992	0,008	0,80%	65%	6,5	6,46	0,04	0,62%
15%	1,5	1,485	0,015	1,00%	70%	7	6,97	0,03	0,43%
20%	2	1,98	0,02	1,00%	75%	7,5	7,47	0,03	0,40%
25%	2,5	2,477	0,023	0,92%	80%	8	7,97	0,03	0,38%
30%	3	2,977	0,023	0,77%	85%	8,5	8,47	0,03	0,35%
35%	3,5	3,473	0,027	0,77%	90%	9	8,98	0,02	0,22%
40%	4	3,971	0,029	0,72%	95%	9,5	9,49	0,01	0,11%
45%	4,5	4,47	0,03	0,67%	100%	10	9,99	0,01	0,10%

Fonte: Autoria Própria.

Analisando os resultados, percebe-se que nenhum erro ultrapassou o passo de 100 mV definido para a saída analógica, portanto, os resultados comprovam o

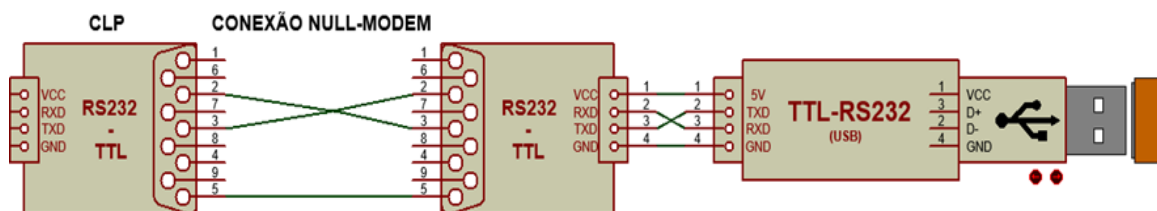
funcionamento do circuito de saída analógica, apresentando um erro relativo máximo de 1,56% onde o *duty cycle* é igual a 5%.

5.6 Comunicação com outros dispositivos

Os testes de comunicação com outros dispositivos foram realizados por meio do envio e recepção de caracteres entre o controlador lógico programável e um computador. Foi desenvolvido um código em Ladder, onde a cada um segundo é enviado um caractere, dependendo de qual módulo de comunicação está selecionado, sendo “a” para comunicação RS-232 e “b” para a comunicação RS-485. Neste mesmo código, foi definido que uma das saídas digitais deve ter seu estado lógico trocado quando receber um caractere. A saída digital que será acionada depende de qual caractere foi recebido, onde “a” aciona a saída Q0 e “b” a saída Q1.

Primeiramente foi realizado o teste do módulo RS-232 do CLP, o qual foi conectado no computador a partir de um conversor USB serial TTL RS-232, cujo esquemático desta conexão pode ser observado na Figura 49.

Figura 49 - Conexão do teste RS-232

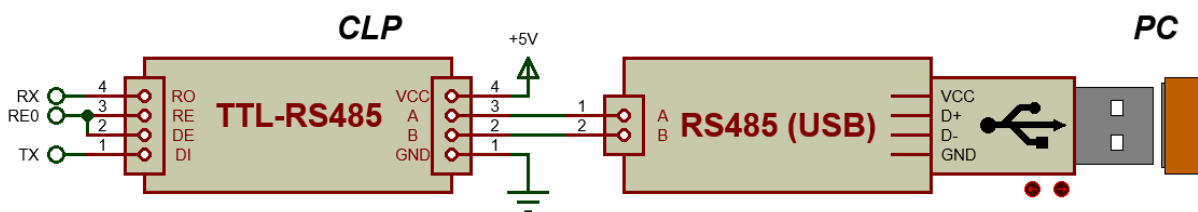


Fonte: Autoria Própria.

O cabo que conecta os dois módulos RS-232 foi montado com uma conexão *null-modem*, uma ligação cruzada entre os pinos 2 e 3 do conector DB9. Após a conexão com o computador, foi selecionado o módulo RS-232 através do *bit* de controle RC0 e assim o CLP deve enviar o caractere “a” a cada um segundo. Em seguida, através do *software* Terminal foi enviado o caractere “a” para o CLP.

No teste de comunicação RS-485, foi utilizado o mesmo código do teste RS-232. O conversor RS-485 do CLP foi conectado a um outro conversor RS-485 conectado à porta USB do computador, conforme a Figura 50.

Figura 50 - Conexão do teste RS-485

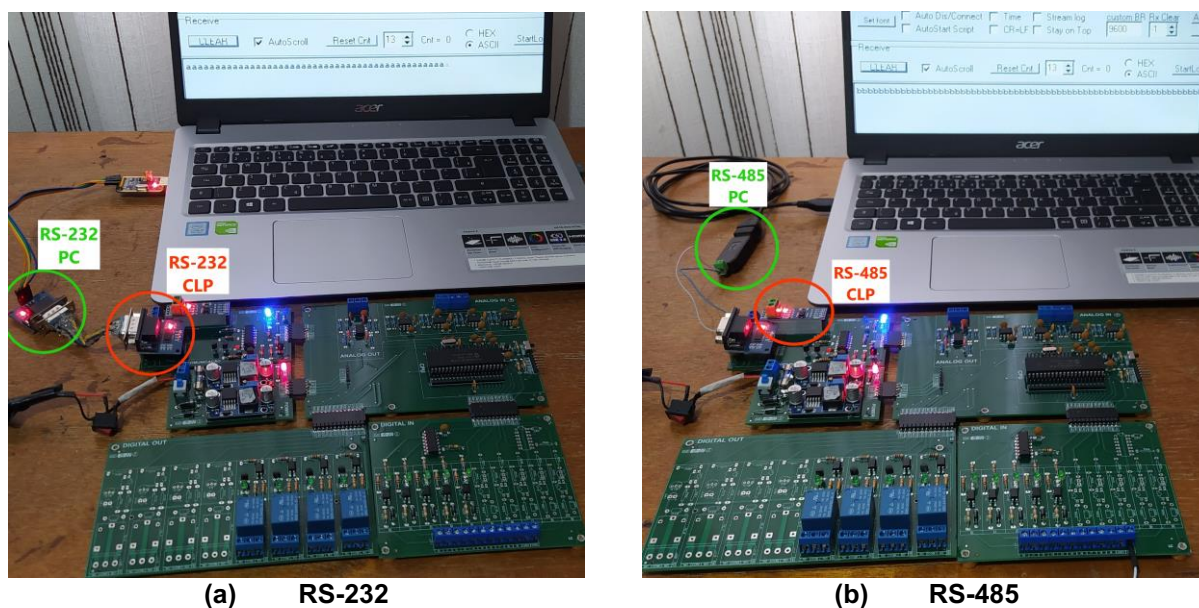


Fonte: Autoria Própria.

Este módulo RS-485 conectado à porta USB do computador possui comunicação *full-duplex*, portanto não precisa de seleção do sentido da comunicação. O módulo RS-485 do CLP foi selecionado através do *bit* RC0 e foi enviado o caractere “b” a cada um segundo. Em seguida, através do *software* Terminal foi enviado o caractere “b” para o CLP.

Os testes de comunicação propostos anteriormente foram realizados tanto para a comunicação RS-232 como também para comunicação RS-485, conforme apresentado na Figura 51.

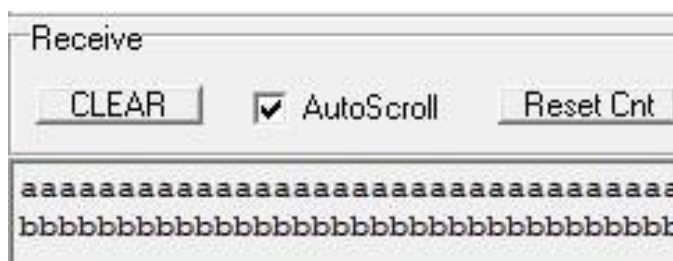
Figura 51 - Testes de comunicação



Fonte: Autoria Própria.

A Figura 52 apresenta os caracteres recebidos pelo *software* Terminal para comunicação serial RS-232 e RS-485, onde durante a comunicação RS-232 foi recebido o caractere “a” e na comunicação RS-485 foi recebido caractere “b”.

Figura 52 - Caracteres recebidos, pelo computador, via comunicação serial



Fonte: Autoria Própria.

A saída Q0 foi ligada e desligada conforme o envio do caractere “a” através do software Terminal e saída Q1 apresentou o mesmo comportamento quando o caractere “b” foi enviado.

Baseado nos resultados apresentados, nota-se que o circuito proposto no trabalho funciona conforme esperado, realizando a comunicação entre os dispositivos, tanto pelo padrão RS-232 como também pelo padrão RS-485, com a seleção e indicação de qual padrão de comunicação está ativo.

6 CONSIDERAÇÕES FINAIS

Este trabalho buscou desenvolver um controlador lógico programável mais acessível, que fosse capaz de controlar sistemas de automação de pequeno porte, fornecendo uma alternativa de baixo custo a quem busca um controlador lógico programável para fins didáticos ou para aplicações simples de automação.

Para alcançar o objetivo geral de desenvolver um controlador lógico programável baseado em um microcontrolador PIC, foram alcançados todos os objetivos específicos definidos, com a identificação dos principais recursos dos controladores lógicos programáveis, que serviu como base para o projeto do circuito elétrico, que foi implementado através do projeto de *layout* e fabricação das placas de circuito impresso e também da construção do dispositivo.

Ao fim do desenvolvimento deste projeto, obteve-se um controlador lógico programável com uma arquitetura simples, que custou aproximadamente 290 reais para a construção de uma unidade. A partir dos testes realizados, comprovou-se que este dispositivo é capaz de detectar sinais de 24 V enviados por sensores industriais do tipo PNP, acionar cargas de corrente alternada ou contínua, medir e gerar sinais analógicos de tensão com amplitude entre 0 e 10 V, além de se comunicar com outros dispositivos utilizando os padrões de comunicação RS-232 e RS-485.

Em pesquisas futuras, sugere-se aprimorar este dispositivo, adicionando novos recursos, como comunicação via interface USB, *bluetooth* ou *wi-fi*, relógio de tempo real com calendário (RTC), utilizar componentes SMD para reduzir a dimensão das placas de circuito impresso e reduzir custos, fabricar uma caixa de proteção ao circuito, além de desenvolver módulos de entradas e saídas para fins específicos, como controle de ângulo de disparo de tiristores, leitura de sensores de temperatura, entre outros. Outra possibilidade é a utilização deste controlador lógico programável no controle de sistemas de automação ou em aplicações didáticas.

É esperado que este trabalho contribua com a ampliação do acesso à tecnologia de controladores lógicos programáveis e auxilie no aprendizado sobre o tema.

REFERÊNCIAS

ALVES, Thiago Rodrigues; *et al.* OpenPLC: An open source alternative to automation. *In: IEEE Global Humanitarian Technology Conference, 2014, San Jose, USA. Proceedings* [...]. San Jose, USA: IEEE, 2014. p.585-590. Disponível em: <https://ieeexplore.ieee.org/document/6970342>. Acesso em: 18 mar. 2021.

ELEUTÉRIO, Warley Alisson; HOVADICH, Wagner Antonio A.; BRAGA, Eduardo Queiroz. CONTROLADOR LÓGICO PROGRAMÁVEL UTILIZANDO PIC 18F4550. **E-xacta**, Belo Horizonte, v. 4, n. 3, p. 159-179, 2011. Disponível em: <https://revistas.unibh.br/dcet/article/view/695>. Acesso em: 18 mar. 2021.

FERNANDO, Jonas Mazza. **Desenvolvimento de um Controlador Lógico Programável com Interface de Programação Ladder via Aplicação Web Embarcada**. 2017. 74 f. Trabalho de Conclusão de Curso (Graduação) - Curso de Engenharia Elétrica, Universidade Federal de Uberlândia, Uberlândia, 2017. Disponível em: <https://repositorio.ufu.br/handle/123456789/20281>. Acesso em: 18 mar. 2021.

FONSECA, Fabrício Ramos da; SANTOS, Fábio Lobue dos; COELHO, Marcelo Saraiva. **Sistemas digitais de controle industrial**. São Paulo: SENAI-SP Editora, 2016. Disponível em: <https://books.google.com.br/books?id=erRCDwAAQBAJ&lpg=PP1&hl=pt-BR&pg=PP1#v=onepage&q&f=false>. Acesso em: 30 abr. 2021.

FOROUZAN, Behrouz A. **Comunicação de dados e redes de computadores**. 4. ed. Porto Alegre: AMGH, 2010. Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788563308474>. Acesso em: 30 abr. 2021.

FRANCHI, Claiton Moro; CAMARGO, Valter Luís de. **Controladores lógicos programáveis: sistemas discretos**. 1. ed. São Paulo: Erica, 2008.

GEVAERD, Bruno de Melo; SILVA, Fernando Costenaro; BONACORSO, Nelso Gauze. Desenvolvimento de um controlador lógico programável modular dedicado ao controle e ao acionamento de equipamentos automatizados. **Caderno de Publicações Acadêmicas**, Florianópolis, v. 2, n. 1, p. 72-78, 2010. Disponível em: <https://periodicos.ifsc.edu.br/index.php/publicacoes/article/view/85>. Acesso em: 18 mar 2021.

INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC-61131-1: Programmable controllers – Part 1: General information**. 2nd ed. Geneva: IEC, 2003.

KAY, Marcel Seiji, et al. Development of a Low Cost Programmable Logic Controller. *In: Brazilian Symposium on Computing System Engineering, 2012, Natal, RN. Proceedings* [...]. Natal, RN: IEEE, 2013. p. 49-52. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6473631>. Acesso em: 18 mar 2021.

KUGELSTADT, Thomas. **The RS-485 Design Guide**. Dallas: Texas Instruments, 2016. Disponível em: <http://www.ti.com/lit/an/slla272c/slla272c.pdf>. Acesso em: 30 abr. 2021.

KUPHALDT, Tony R. **Lessons In Industrial Instrumentation**. Version 2.32. California: Creative Commons, 2019. Disponível em: http://ibiblio.org/kuphaldt/socratic/sinst/book/liii_2v32.pdf. Acesso em: 30 abr. 2021.

MARAIS, Hein. **RS-485/RS-422 Circuit Implementation Guide**. Norwood: Analog Devices, 2008. Disponível em: <https://www.analog.com/media/en/technical-documentation/application-notes/AN-960.pdf>. Acesso em: 30 abr. 2021.

MAXIM INTEGRATED PRODUCTS INC. **MAX481/MAX483/MAX485/ MAX487–MAX491/MAX1487** Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers. set. 2014. Disponível em: <https://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf>. Acesso em: 03 jun. 2022.

MICALOSKI, Odirlei. **Mini controlador lógico programável**. 2012. 45 f. Trabalho de Conclusão de Curso (Graduação) – Universidade Tecnológica Federal do Paraná, Curitiba, 2012. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/handle/1/904>. Acesso em: 18 mar. 2021.

MICROCHIP TECHNOLOGY INC. **Analog and Interface Guide**. Volume 1. 2005. Disponível em: <http://ww1.microchip.com/downloads/en/devicedoc/00924b.pdf>. Acesso em: 31 mai. 2022.

MICROCHIP TECHNOLOGY INC. **PICkit™ 3 In-Circuit Debugger/Programmer User's Guide**. 29 nov. 2012. Disponível em: <http://ww1.microchip.com/downloads/en/devicedoc/52116a.pdf>. Acesso em: 31 mai. 2022.

MICROCHIP TECHNOLOGY INC. **Data Sheet PIC16F87XA (DS39582C)**. Microchip Technology Inc.: USA, 2013. Disponível em: <https://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf>. Acesso em: 30 abr. 2021.

MIYADAIRA, Alberto Noboru. **Microcontroladores PIC18: aprenda e programe em linguagem C**. 4. ed. São Paulo: Érica, 2013.

MONDADORI, Jorge Augusto Pessatto. **Redes Industriais**. Uberaba: Editora Universidade de Uberaba, 2016.

MORAES, Cícero Couto de; CASTRUCCI, Plínio de Lauro. **Engenharia de automação industrial**. 2. ed. Rio de Janeiro: LTC, 2010. Disponível em: <https://integrada.minhabiblioteca.com.br/books/978-85-216-1976-5>. Acesso em: 30 abr. 2021.

NEVES, Y.; FREITAS, D. Construção de Controlador Lógico Programável de Baixo Custo para Fins Didáticos. **Revista de Engenharia e Pesquisa Aplicada**, v. 4, n. 4, p. 28-39, 10 nov. 2019. Disponível em: <http://revistas.poli.br/index.php/repa/article/view/1172/523>. Acesso em: 18 mar. 2021.

NICOLOSI, Denys Emílio Campion. **Microcontrolador 8051 Detalhado**. 9. ed. São Paulo: Érica, 2013. Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788536519876>. Acesso em: 30 abr. 2021.

PAREDE, Ismael Moura; GOMES, Luiz Eduardo Lemes. **Eletrônica: Automação Industrial**. São Paulo: Fundação Padre Anchieta, 2011. (Coleção Técnica Interativa. Série Eletrônica, v. 6).

PECKOL, James K. **Embedded Systems: A Contemporary Design Tool**. 2nd ed. Hoboken, NJ: John Wiley & Sons, 2019. Disponível em: <https://books.google.com.br/books?id=ummPDwAAQBAJ&pg=PR1&hl=pt-BR&pg=PR1#v=onepage&q&f=false>. Acesso em: 30 abr. 2021.

PEREIRA, Fábio. **Microcontroladores PIC: programação em C**. 7. ed. São Paulo: Érica, 2007. Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788536519937>. Acesso em: 30 abr. 2021.

PETRUZELLA, Frank D. **Controladores lógicos programáveis**. 4. ed. Porto Alegre: AMGH, 2014. Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788580552836>. Acesso em: 30 abr. 2021.

REYNDERS, Deon; MACKAY, Steve; WRIGHT, Edwin. **Practical Industrial Data Communications: Best Practice Techniques**. 1st ed. Amsterdam: Butterworth-Heinemann, 2005. Disponível em: <http://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=123446&lang=pt-br&site=eds-live&scope=site>. Acesso em: 30 abr. 2021.

ROHDE, Regis Almir; KIECKOW, Flávio; MACHADO, Filipe Molinar. PROJETO DE UM MICRO-CLP DE BAIXO CUSTO COM PROGRAMAÇÃO EM LADDER. *In*: Congresso Brasileiro de Educação em Engenharia (COBENGE), 2011, Blumenau. **Anais** [...]. Brasília: Editora ABENGE, 2011. Disponível em: <http://www.abenge.org.br/cobenge/legado/arquivos/8/ressoestec/art1792.pdf>. Acesso em: 30 abr. 2021.

SANVER, Ufuk, et al. Design and implementation of a programmable logic controller using PIC18F4580. *In*: IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2018, St. Petersburg and Moscow. **Proceedings** [...]. St. Petersburg and Moscow: IEEE, 2018. p. 231-235. Disponível em: <https://ieeexplore.ieee.org/document/8317072>. Acesso em: 18 Mar. 2021.

SILVEIRA, Paulo Rogério da; SANTOS, Winderson E. **Automação e Controle Discreto**. 9. ed. São Paulo: Érica, 2010.

SOLTERO, Manny; ZHANG, Jing; COCKRIL, Chris. **RS-422 and RS-485 Standards Overview and System Configurations**. Dallas: Texas Instruments, 2010. Disponível em: <http://www.ti.com/lit/an/slla070d/slla070d.pdf>. Acesso em: 30 abr. 2021.

SOUZA, Daniel Rodrigues de; SOUZA, David José de. **Desbravando o PIC18: ensino didático**. 1. ed. São Paulo: Érica, 2012. Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788536518329>. Acesso em: 30 abr. 2021.

STEVAN JUNIOR, Sergio Luiz; LEME, Murilo Oliveira; SANTOS, Max Mauro Dias. **Indústria 4.0: fundamentos, perspectivas e aplicações**. São Paulo: Érica, 2018. Disponível em: <https://books.google.com.br/books?id=OYuwDwAAQBAJ&lpg=PP1&hl=pt-BR&pg=PP1#v=onepage&q&f=false>. Acesso em: 30 abr. 2021.

SURESH, G.; KAILASH, N. K.; VARSHAA, D.; SAMEER, J. Mohamed. Design and Implementation of a Configurable Programmable Logic Controller Using Microchip PIC. **International Journal of Research in Engineering, Science and Management**, Andhra Pradesh, India, v. 3, n. 3, p. 311-315, 2020. Disponível em: <https://ieeexplore.ieee.org/document/8317072>. Acesso em: 18 mar 2021.

TANENBAUM, Andrew S. **Redes de Computadores**. Tradução de Vandenberg D. de Souza. 4. ed. Rio de Janeiro: Campus, 2003.

TEXAS INSTRUMENTS. **Interface Circuits for TIA/EIA-232-F**: Design Notes. Dallas: Texas Instruments, 2002. Disponível em: <https://www.ti.com/lit/an/slla037a/slla037a.pdf>. Acesso em: 30 abr. 2021.

TEXAS INSTRUMENTS. **MAX3232**: 3-V to 5.5-V Multichannel RS-232 Line Driver and Receiver With ± 15 -kV ESD Protection. Dallas: Texas Instruments, 2021. Disponível em: https://www.ti.com/lit/ds/symlink/max3232.pdf?ts=1654387212199&ref_url=https%253A%252F%252Fwww.ti.com%252Fdocument-viewer%252FMAX3232%252Fdatasheet. Acesso em: 30 abr. 2022.

UNDERSTANDING EIA-485 Networks. **The Extension**, [s. l.], v. 1, n. 1, Spring 1999. Disponível em: <https://www.ccontrols.com/pdf/ExtV1N1.pdf>. Acesso em: 30 abr. 2021.

UZAM, Murat. **PIC16F1847 Microcontroller-Based Programmable Logic Controller**: Hardware and Basic Concepts. 1st ed. New York: CRC Press, 2021.

WEIS, Olga. O que é o RS485? Guia de comunicação RS485. **Eltima Publishing**, 2019. Disponível em: <https://www.eltima.com/pt/article/rs485-communication-guide/>. Acesso em: 30 abr. 2021.

WESTHUES, Jonathan. **LDmicro**: Ladder Logic for PIC and AVR, 2016. Disponível em: <https://cq.cx/ladder-pt.html>. Acesso em: 30 abr. 2021.

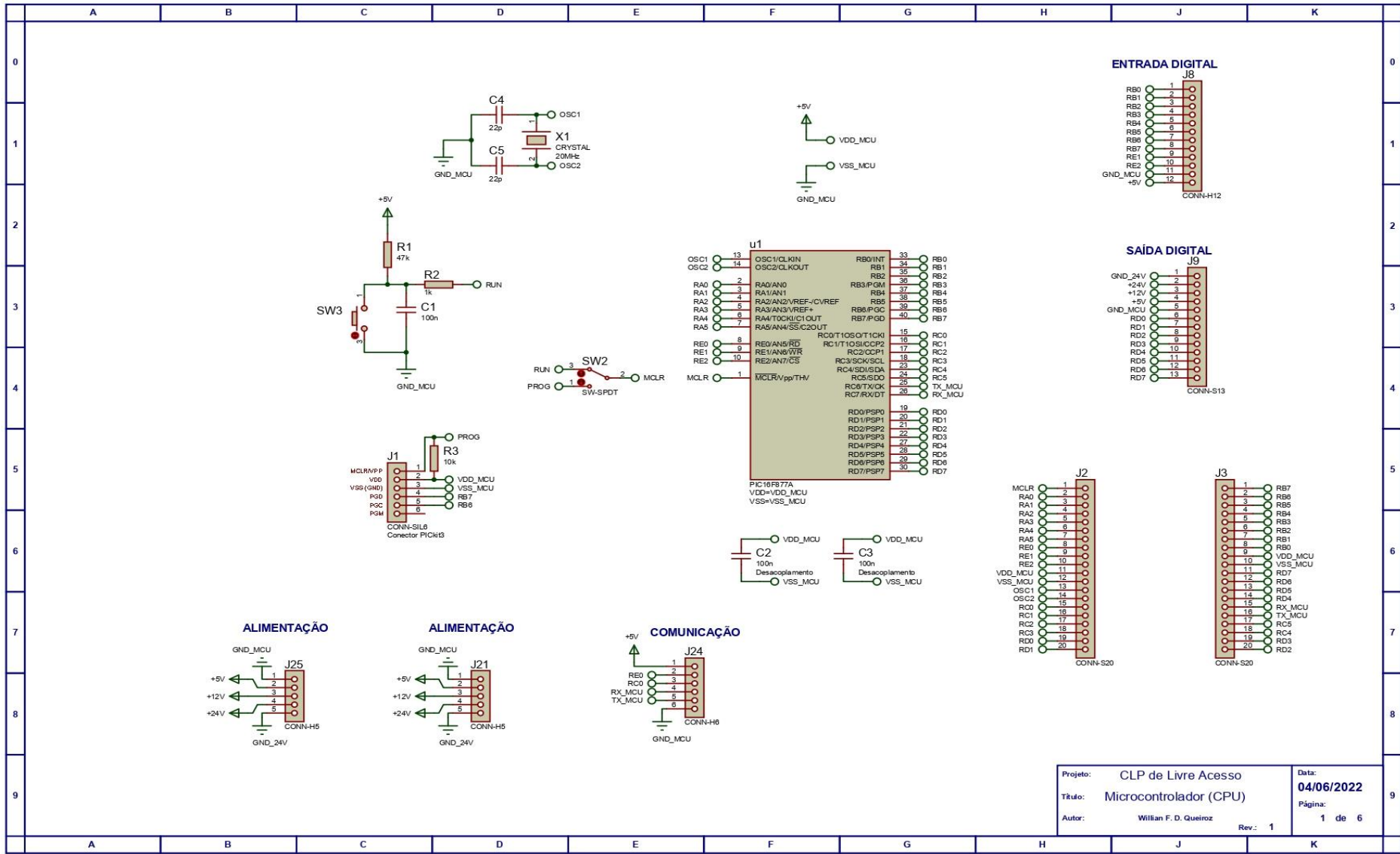
ZANZOTI, Fernando Henrique Oliveira. **Desenvolvimento do protótipo de hardware de um micro PLC**. 2019. 68 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Uberlândia, Uberlândia, 2019. Disponível em: <https://repositorio.ufu.br/handle/123456789/26392>. Acesso em: 18 mar. 2021.

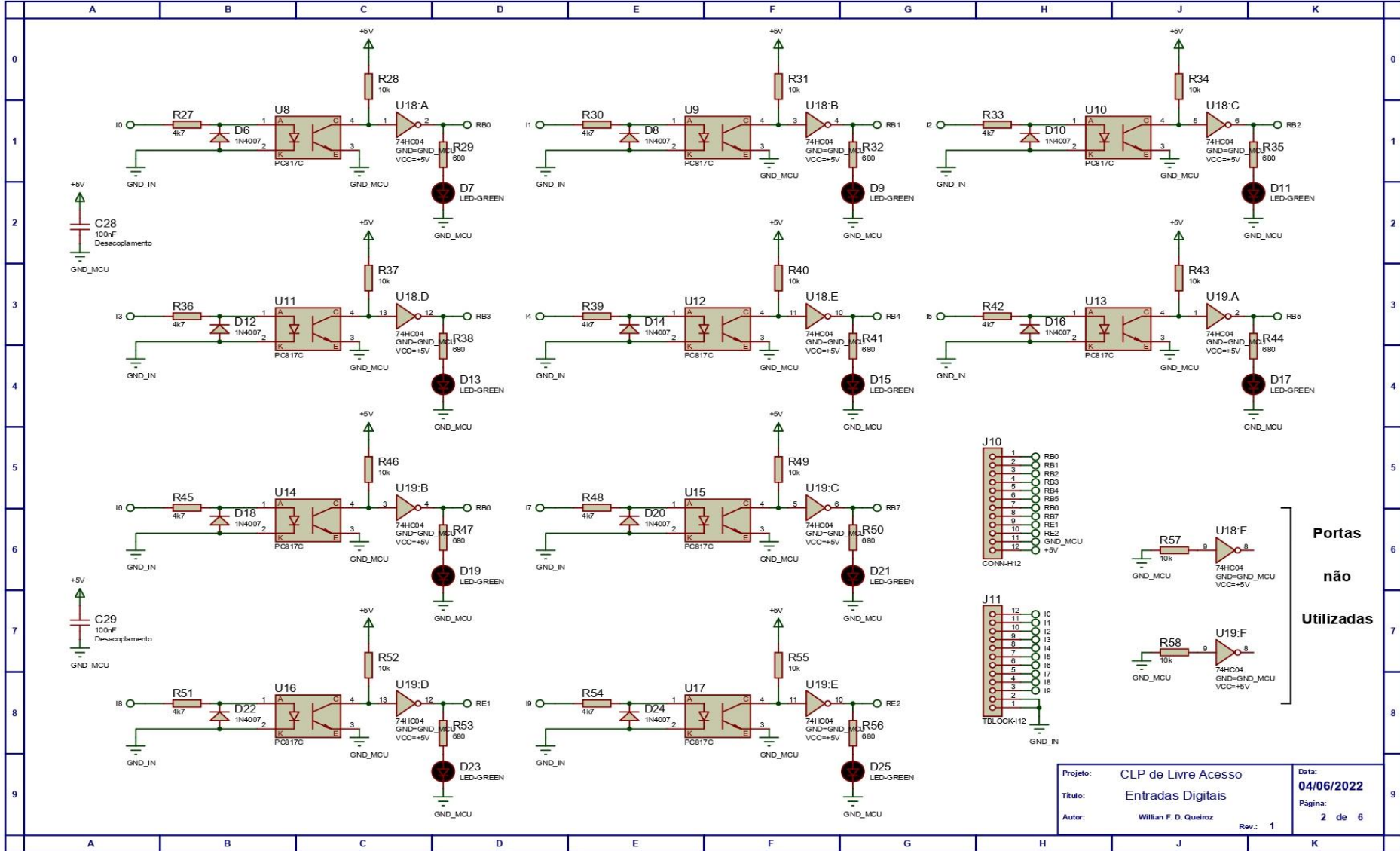
APÊNDICE A - Lista de componentes

Descrição do componente	Quantidade
Capacitor Cerâmico Disco 100nf 50V	27
Capacitor Cerâmico Disco 22pf 50V	2
Resistor 47k Ω 1/4 W 5%	1
Resistor 1 k Ω 1/4 W 5%	6
Resistor 10 k Ω 1/4 W 5%	14
Resistor de precisão 10 k Ω 1/4 W 1%	12
Resistor de precisão 2k7 Ω 1/4 W 1%	2
Resistor 360 Ω 1/4 W 5%	2
Resistor 560 Ω 1/4 W 5%	1
Resistor 4k7 Ω 1/4 W 5%	10
Resistor 680 Ω 1/4 W 5%	10
Resistor 750 Ω 1/4 W 5%	8
Resistor 33 k Ω 1/4 W 5%	8
Resistor 4k3 Ω 1/4 W 5%	8
Microcontrolador PIC16F877A	1
Amplificador Operacional LM358	5
Circuito Integrado 74HC08	1
Circuito Integrado 74HC04	2
Optoacoplador PC817C	18
Transistor S9014	9
Transistor 2N700	3
Diodo 1N5408	1
Diodo 1N4007	18
Diodo P6KE24A	1
Led Difuso 3mm - Vermelho	1
Led Difuso 3mm - Azul	2
Led Difuso 3mm - Verde	18
Módulo RS232-TTL (MAX3232)	1
Módulo RS485-TTL (MAX485)	1
Conversor DC-DC Buck LM2596S	2
Fusível 500 mA	1
Relé 24V SRD-24VDC-SL-C	8

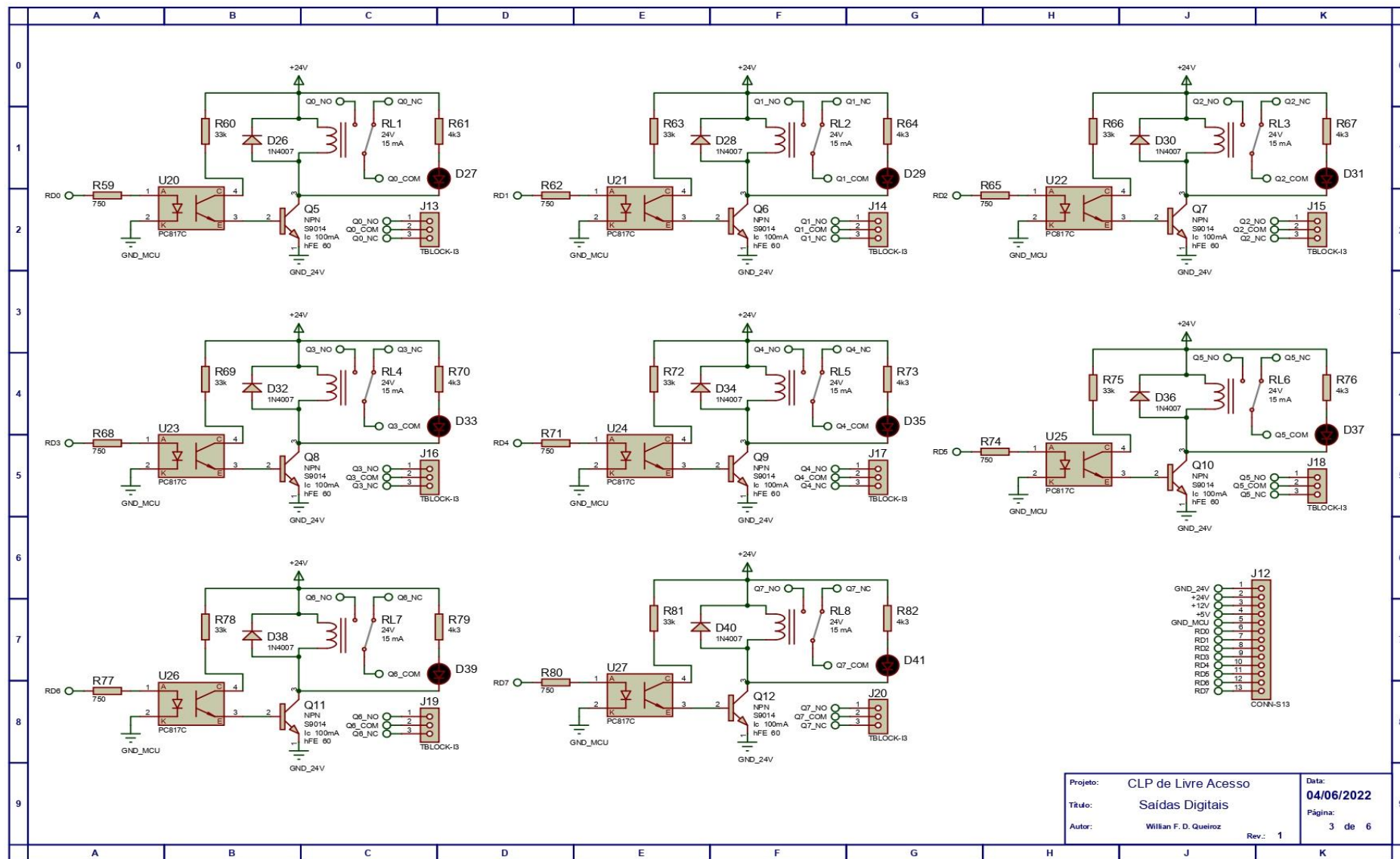
Descrição do componente	Quantidade
Chave Táctil KFC-A06 6x6x6mm	1
Chave SPDT SK12D07VG3	1
Chave com trava 8x8mm	1
Cristal Oscilador 20 MHz	1
Borne KRE 3T	15
Borne KRE 2T	2
Pinos SIL 0° Macho	45
Pinos SIL 90° Macho	42
Pinos SIL 90° Fêmea	42

APÊNDICE B - Esquemático do circuito elétrico





Projeto:	CLP de Livre Acesso	Data:	04/06/2022
Título:	Entradas Digitais	Página:	2 de 6
Autor:	Willan F. D. Queiroz	Rev.:	1



Projeto: CLP de Livre Acesso
 Título: Saídas Digitais
 Autor: Willian F. D. Queiroz
 Data: 04/06/2022
 Página: 3 de 6
 Rev.: 1

