FACULDADE SENAI DE TECNOLOGIA MECATRÔNICA

CONTROLADOR LÓGICO PROGRAMÁVEL PARA ENSAIOS DE DESEMPENHO DE MICROCONTROLADORES

PROGAMABLE LOGIC CONTROLLER FOR PERFORMANCE TESTS OF MICROCONTROLLERS

Vítor Eduardo Sabadine da Cruz[1] [i]
Paulo Sebastião Ladivez[2] [ii]
José Roberto dos Santos[3] [iv]
Sérgio Luiz Volpiano[4] [v]
Vicente Gomes de Oliveira Junior[5] [vi]

RESUMO

Com o objetivo de testar o desempenho de diferentes tipos de tecnologias de microcontroladores e sistemas embarcados, este trabalho propõe um circuito de um Controlador Lógico Programável (CLP), que permita que seu núcleo de processamento seja removível, e uma estrutura de *software* para comparação dessas tecnologias. O circuito permitirá que placas compatíveis com as conexões do Arduino Due sejam usadas no CLP, sem a necessidade de nenhuma alteração nos demais circuitos da placa. Para confeccioná-la, foram escolhidos circuitos reguladores, de entrada e saída, que permitam a compatibilidade elétrica e funcional entre diferentes sistemas embarcados.

ABSTRACT

In order to test the performance of different kinds of microcontrollers technology and embedded systems, this work proposes a Programable Logic Controller (PLC) circuit, allowing its processing core to be removable, and a software structure for comparison of these technology. The circuit must allow that boards compatible with Arduino Due connections can be used in the PLC, without the necessity of any interaction with the other board circuits. During the design of this board, it was chosen circuits of regulators, inputs and outputs that allow electric and functional compatibility between different embedded systems.

[1]Pós-graduando em Automação Industrial na Faculdade SENAI de Tecnologia Mecatrônica. E-mail: vitorengcruz@gmail.com
[2]Professor da Faculdade SENAI de Tecnologia Mecatrônica. E-mail: paulo.ladivez@sp.senai.br
[3] Docente na Faculdade SENAI de Tecnologia Mecatrônica. E-mail: joseroberto@sp.senai.br
[4] Orientador. Docente e Me. em Engenharia Elétrica em Sistema de Potência da Faculdade Senai de Tecnologia Mecatrônica. E-mail: sergio.volpiano@sp.senai.br
[5] Mestre em Engenharia Mecânica. Professor da Faculdade SENAI de Tecnologia Mecatrônica. E-mail: vgomes@sp.senai.br.

1 INTRODUÇÃO

Atualmente o mercado de microcontroladores e sistemas embarcados está crescendo cada vez mais, graças à sua densidade de recursos em *chips* e placas compactas, possibilitando que sejam aplicados em soluções exclusivas, portáveis, com restrições de consumo e alto volume de produção, como nos *smartphones*, eletrodomésticos e eletrônicos de consumo em aplicações industriais, aeroespaciais, militares, saúde, entre outras. Mas junto com essa tendência, crescem também as dúvidas dos desenvolvedores em responder uma simples pregunta: Qual tecnologia embarcada devemos usar no desenvolvimento de cada tipo de trabalho?

Segundo Ganssle e Barr (2003):

> [Sistema embarcado é] uma combinação de *hardware* e *software* de computador, e possivelmente adição de partes mecânicas entre outros, projetado para uma função dedicada. Em alguns casos, sistemas embarcados são partes de um sistema ou produto maior, como no caso de um Sistema de Antitravamento de Freios (ABS) em um carro. (GANSSLE; BARR, 2003, p. 90).

Neste trabalho, será desenvolvido uma placa que simule um CLP a nível acadêmico, de baixo custo e sem o compromisso de substituir um CLP industrial onde, além das entradas e saídas, tanto analógicas quanto discretas, existirá uma conexão com o padrão da placa de desenvolvimento Arduino Due. Esta característica permitirá a troca do núcleo de processamento do CLP e, consequentemente, a possibilidade de estudar as aplicações de diferentes tecnologias, de microcontroladores a processadores de sinais digitais disponíveis no mercado, em um mesmo contexto de aplicação industrial.

2 DESENVOLVIMENTO

De acordo com Petruzella (2014), um CLP possui características vantajosas para a automação industrial:

> Esses controladores (CLPs) reduziram muito a fiação associada aos circuitos de controle convencional a relé, além de apresentar outros benefícios, como a facilidade de programação e instalação, controle de alta velocidade, compatibilidade de rede, verificação de defeitos e conveniência de teste e alta confiabilidade. (PETRUZELLA, 2014, p. 1).

Para que o programa execute um processo é necessário produzir a interface física entre o microcontrolador e as entradas e saídas do CLP. Para isso, será descrito como esta interface foi idealizada.

2.1 *Hardware*

Com o objetivo de comparar o tempo de processamento de um programa usando diferentes soluções de *hardware* e *software* embarcados, foi desenvolvido uma placa de circuito impresso que simula um CLP de mercado, o que facilita sua instalação em máquinas já existentes. Por se tratar de um *hardware* simplificado, os circuitos analógicos necessitam de uma calibração via *software*.

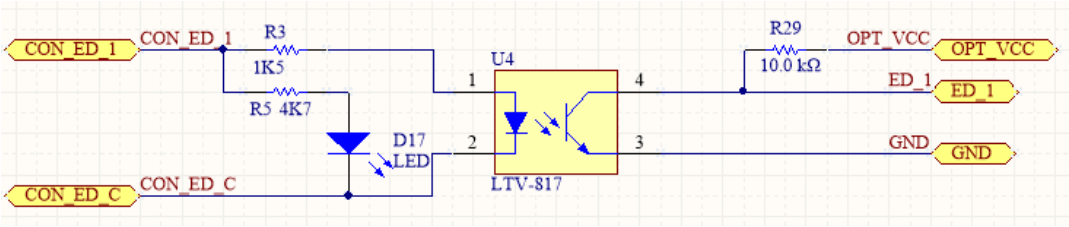
2.1.1 Entradas

Para coletar informações oriundas de grandezas físicas, é necessário que esta seja convertida em sinais elétricos, através de dispositivos chamados de sensores. Após a conversão, existe a necessidade dos sinais elétricos serem transmitidos a média e longas distâncias, o que exige que sejam usadas tensões elevadas, já que as perdas ocorridas nos cabos de transmissão podem deteriorar a integridade da informação a ser transmitida para o CLP.

Existem três formas de sinais elétricos oriundas dos sensores: analógicas, discretas e digitais. Segundo Bolton (2009, p.81), “quando estiver conectando sensores que geram sinais digitais ou discretos para uma unidade de entrada, cuidados devem ser tomados para garantir que os níveis de tensão sejam compatíveis”. Então neste artigo são apresentados circuitos de conversão para cada tipo de sinal externo, com a finalidade de torná-los compatíveis com os níveis de tensão dos microcontroladores escolhidos, variando entre 3,3 a 5 volts.

2.1.1.1 Entradas discretas

Figura 1 – Circuito para entrada discreta isolada do CLP



Fonte: Elaborado pelo autor

Para condicionar as entradas discretas, representadas na figura 1, foram utilizados acopladores ópticos com isolação elétrica de 5.000V, que polarizam fototransistores NPN em suas saídas quando energizados com 24VDC entre os dois pontos CON_ED (Conexão Entrada Discreta).

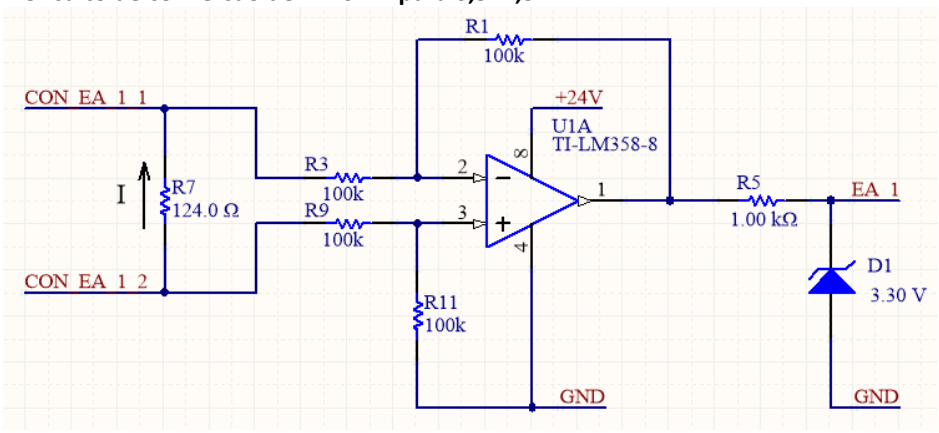
2.1.1.2 Entradas analógicas

De acordo com Bolton (2009):

> Para evitar ter múltiplos tipos de canais de entrada, que sejam compatíveis com a grande variedade de sinais analógicos, que podem ser geradas pelos sensores, condicionadores de sinais externos são também usados para trazer os sinais analógicos para um nível comum e assim permitir uma forma padrão de uso de entradas de sinais analógicos. (BOLTON, 2009, p. 81).

Um padrão comum na indústria é o sinal analógico com uma corrente de 4 a 20 mA passar por um resistor de 250 ohms para gerar um sinal de entrada de 1 a 5 V, que posteriormente será convertido por um circuito Conversor Analógico para Digital (ADC). Porém como os microcontroladores possuem diferentes níveis de tensão de entrada, foi decidido usar um resistor de 124 ohms e converter a corrente para um nível de 0,5 a 2,5 V, mantendo assim a possibilidade de substituição do microcontrolador sem a necessidade de alterações na placa principal do CLP proposto neste artigo. A figura 2 mostra o circuito dimensionado para a aplicação.

Figura 2 – Circuito de conversão de 4~20mA para 0,5~2,5V.



Fonte: Elaborado pelo autor.

2.1.2 Saídas

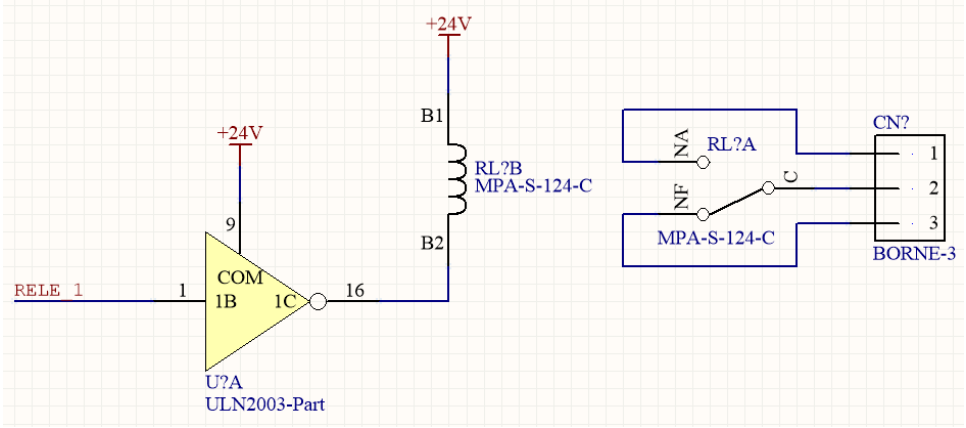
Após a leitura e o processamento da informação, o CLP precisa externar o resultado da operação para acionar os dispositivos da máquina e fazer com que ela cumpra a função desejada pelo programador. Apesar do microcontrolador dispor de circuitos periféricos internos para realizar esta função, estes não possuem capacidade para fornecer tensão e corrente elétrica suficiente para acionar cargas de alta potência, como lâmpadas e motores. Além disso, existe a necessidade de controlar dispositivos de forma analógica, mas que estão distantes do CLP.

2.1.2.1 Saídas discretas

Para acionamento de atuadores e indicadores discretos, foi escolhido um relé com contato reversível, que é um componente comum, de baixo custo e que garante robustez na saída do CLP. Foi utilizado o circuito integrado (CI) ULN2803, que possui um conjunto de 8 transistores Darlington para ligar as bobinas destes relés.

Uma representação simbólica comum desse circuito é uma porta inversora digital, já que quando aplicado tensão acima de 1,4V em sua base (entrada), o coletor fica com uma tensão próxima do zero. Abaixo está a representação de uma saída discreta do CLP, na figura 3.

Figura 3 – Esquemático de uma saída discreta do CLP, com ULN2803.

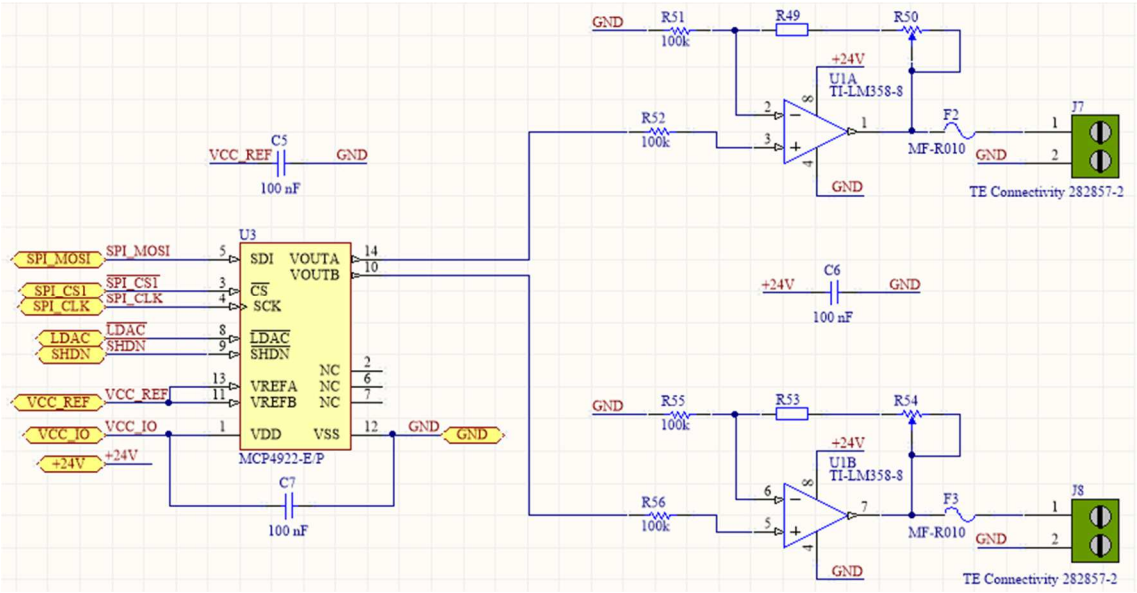


Fonte: Elaborado pelo autor

2.1.2.2 Saídas analógicas

O circuito para saída analógica foi dimensionado para uma escala de 0~10V, tensão de operação de uma válvula proporcional disponível no SENAI. Para manter a compatibilidade da placa principal do CLP com qualquer processador que for utilizado para estudo, optou-se por usar o CI Conversor Digital para Analógico (DAC) MCP4922, conectado via *Serial Peripheral Interface* (SPI). Na figura 4 pode-se observar o circuito amplificador, conectado nas saídas dos conversores DAC, para elevar a tensão de 3,3V para 10V, R49=R53=180kΩ e R50=R54=47kΩ.

Figura 4 – Esquemático das duas saídas analógicas do CLP, com MCP4922

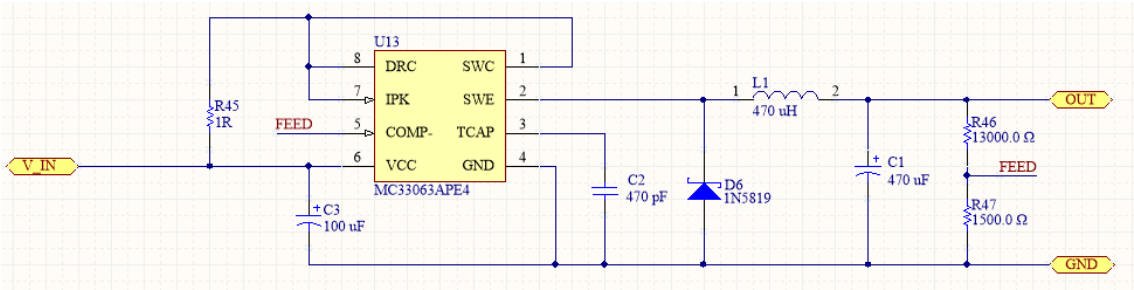


Fonte: Elaborado pelo autor

2.1.3 Regulador de tensão para Vin

Para regular a tensão de alimentação do *Arduino*, por se tratar de uma diferença de tensão de 24V para 12V, foi utilizada o CI regulador chaveado MC33063APE4 no modo *step-down*, pois este circuito integrado aceita uma tensão de entrada elevada e, ao mesmo tempo, conta com uma eficiência energética maior que um regulador linear, diminuindo o aquecimento deste circuito e eliminando a necessidade de adicionar um dissipador térmico no CI. O circuito da figura 5 tem 12V de saída e alimentará uma carga de, no máximo, 150mA, operando em uma frequência de 57,65kHz e com *ripple* de 33,75mV.

Figura 5 – Esquemático do regulador de tensão de Vin, com o MC33063APE4



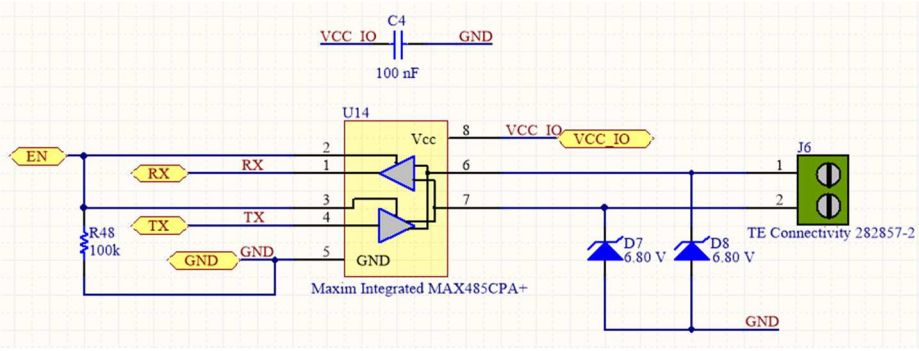
Fonte: Elaborado pelo autor

Quanto ao regulador de 3,3V, foi decidido utilizar o próprio regulador interno da placa do Arduino (ou compatível), pois os únicos circuitos externos que serão alimentados por este componente serão os resistores de *pull-up* e o conversor DAC, sendo que estes possuem uma corrente de consumo baixo.

2.1.4 Comunicação com computador e sistemas externos

Para realizar a transferência de dados da planta para uma interface gráfica, é necessária uma conexão física para transferência de dados digitais. Esta interface deve permitir que o computador ou sistemas externos se conectem de média à longa distância da planta, permitindo a transmissão de dados em velocidades suficientes para que o sistema tenha um funcionamento bem fluido, sem travas ou demoras nas respostas da comunicação. Foi usado o um circuito padrão TIA/EIA 485 na montagem, com topologia mestre-escravo para atender essa funcionalidade, conforme a figura 6.

Figura 6 – Esquemático do driver de comunicação RS-485

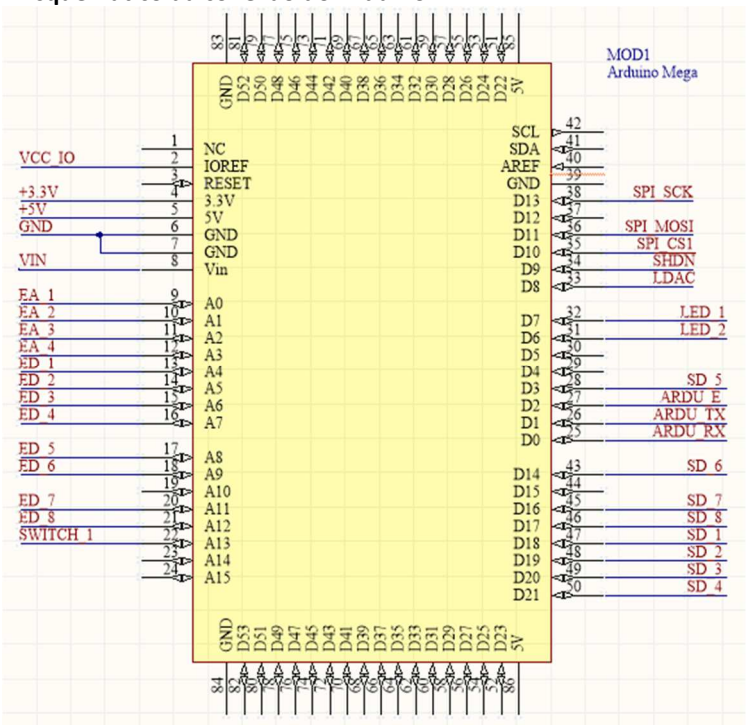


Fonte: Elaborado pelo autor

2.1.5 Conexão do Arduino

Na conexão entre a placa do Arduino e a placa do CLP, serão utilizadas barras de pinos para conexão entre circuitos impressos.

Figura 7 – Esquemático da conexão do Arduino



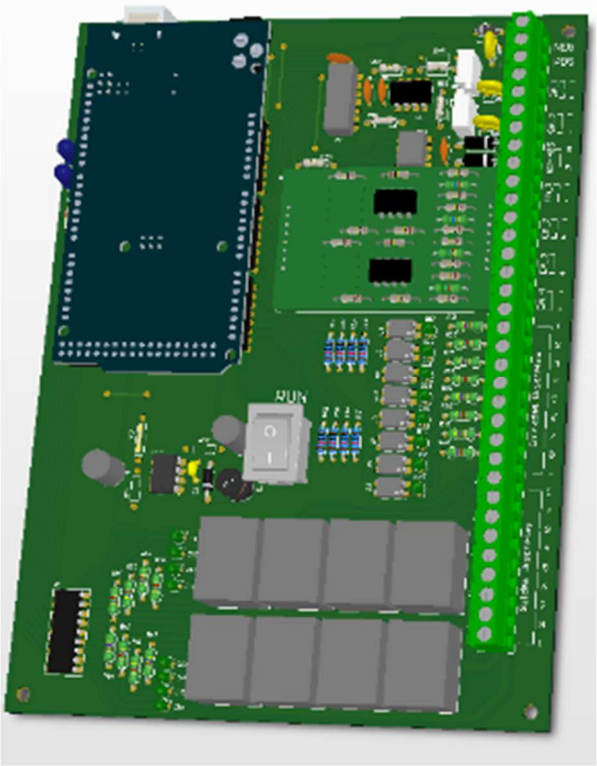
Fonte: Elaborado pelo autor

Na figura 7, pode-se observar as ligações para os circuitos: entradas analógicas (EA_X), entradas discretas (ED_X), saídas analógicas (SPI), saídas discretas (SD_X) e comunicação digital (ARDU).

2.1.6 Circuito impresso

Algumas restrições foram encontradas para realizar o desenho do circuito, entre elas a necessidade de ser instalado dentro do quadro elétrico na planta de manipulação de líquidos, para isso foi levantado o tamanho máximo da placa de circuito impresso a partir da disposição dos componentes no quadro elétrico. Em seguida, foi decidido fazer a placa de circuito impresso em face simples, possibilitando sua fabricação usando maquinário disponível na Escola SENAI. A figura 8 mostra prévia da placa completa.

Figura 8 – Foto do modelo 3D da placa de circuito impresso no *software* CircuitMaker

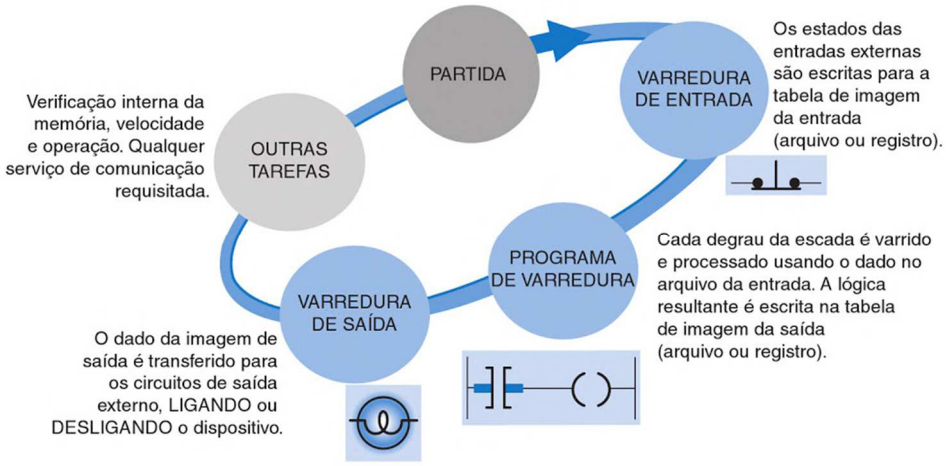


Fonte: Elaborado pelo autor

2.2 *Software*

A figura 9 mostra a representação descrita por Petruzella (2014, p. 73) de “[...]um ciclo de varredura de um CLP simples, que consiste em varredura de entrada, varredura do programa, varredura de saída e outras tarefas.”

Figura 9 – Ciclo de varredura do programa do CLP



Fonte: Petruzella (2014, p. 74)

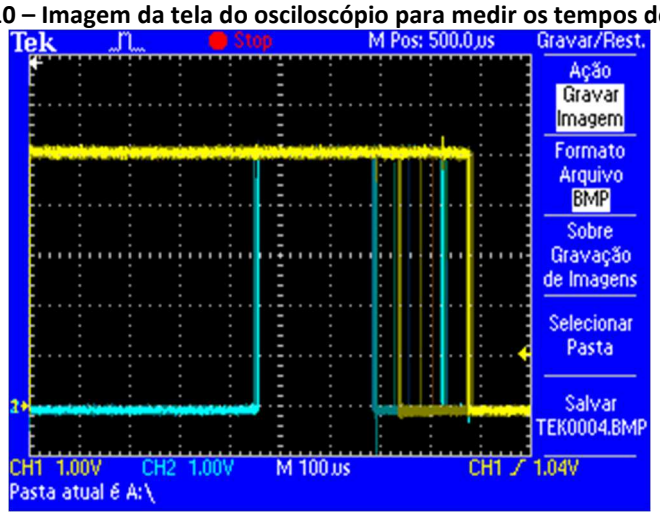
Ao ler as entradas físicas, o programa faz o uso de memória de imagem para armazenar o estado de todas as entradas, tanto analógicas quanto digitais, lidas quando ocorre a varredura de entradas. Isso permite ao processo trabalhar com valores capturados em uma única fração de tempo, garantindo que esses números estejam sincronizados e não sofrerão alteração a medida em que a varredura do programa é executada, evitando assim inconsistências lógicas, além de isolar a camada física do programa do usuário.

Durante a execução, é necessário que os resultados das operações sejam escritos nas correspondentes saídas analógicas, assim fazendo com que a planta receba o sinal calculado. A técnica de memória imagem foi usada para criar uma estrutura para as saídas, similar à varredura de entradas, obtendo-se assim as mesmas vantagens do sincronismo e isolamento entre programa e *hardware*. Estes valores, por sua vez, só são de fato aplicados no *hardware* após a finalização do programa de varredura.

Após esta estrutura de processo, foi necessário adicionar um atraso de tempo ocioso restante para que os cálculos Proporcional Integral Derivativo (PID) tenham uma base de tempo fixo e determinado em suas iterações. No caso usado para comparar as tecnologias neste trabalho, foi usado um período fixo de 1ms para a execução de uma iteração do programa.

A figura 10 mostra como os tempos de uma iteração do cálculo pode ser capturado para ser analisado, utilizando um osciloscópio, com dois canais, ligados em dois pinos de saída digital sobressalentes do microcontrolador. O primeiro pino foi programado para indicar quando o ciclo de varredura do programa começa e termina, antes de começar a contar o tempo ocioso restante da iteração (linha amarela), e o outro pino para quando o programa de varredura começou e terminou (linha azul).

Figura 10 – Imagem da tela do osciloscópio para medir os tempos de cada rotina.



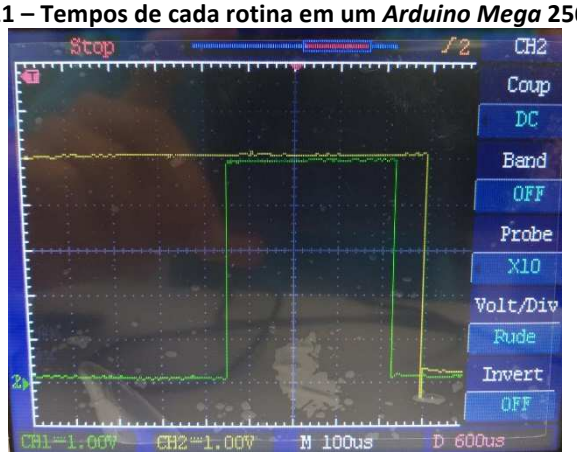
Fonte: Elaborado pelo Autor

É possível observar que uma tela do osciloscópio corresponde a uma iteração de 1ms, dividido em 10 períodos de 100us. O *trigger* foi posicionado no começo da tela e configurado no canal um com inclinação de subida, desta forma a captura da tela se inicia quando o pino correspondente ao ciclo de varredura transitar de 0 para 1.

Ainda na figura 10, para usar como base de estudo, o código foi executado no microcontrolador *Arduino Uno*, usando o compilador presente na interface do *Arduino IDE*. Um programa de varredura que calcula, em ponto flutuante, dois sistemas PID diferentes foi codificado para este teste, sendo esse mesmo programa executado em todos os ensaios deste artigo. É possível observar que o *Arduino Uno*, por realizar os cálculos de ponto flutuante por *software*, apresentou uma grande variação de tempo de processamento entre iterações do ciclo de varredura.

No segundo ensaio, foi comparado outros dois *hardwares*, *Arduino* Mega 2560 e Due, ambos compatíveis com a interface do Arduino IDE e usando o mesmo projeto ensaiado com o Arduino Uno, mostrados nas figuras 11 e 12, respectivamente.

Figura 11 – Tempos de cada rotina em um *Arduino Mega* 2560.



Fonte: Elaborado pelo Autor

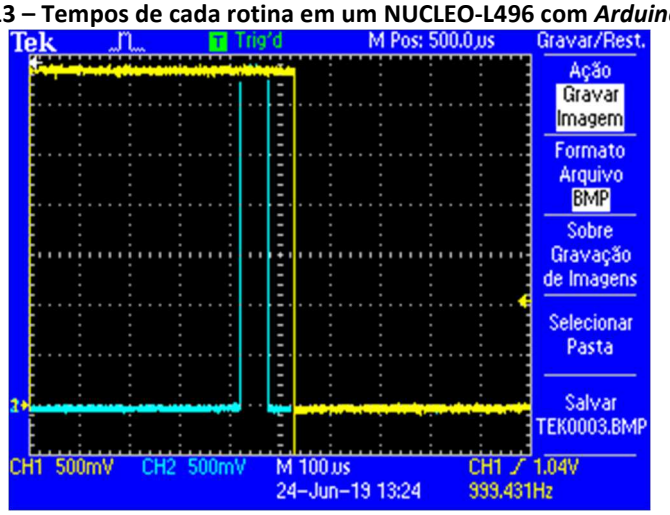
Figura 12 – Tempos de cada rotina em um *Arduino Due*.

Fonte: Elaborado pelo Autor

Pode ser observado, na figura 11, que o comportamento do ciclo de varredura do Mega não sofreu alterações em comparação com o Uno, pois as arquiteturas de ambos são muito similares. Já no ensaio do Due, na figura 12, observou-se um ganho significativo na leitura das entradas e no cálculo do ponto flutuante, justificado pelo *hardware* e *software* preparados para captura mais rápida das entradas analógicas e o processador *ARM Cortex-M3* que possui um módulo para cálculos em ponto flutuante, utilizado na arquitetura do microcontrolador SAM3X8E do *Due*.

Em seguida foi testado um kit de desenvolvimento compatível com *Arduino IDE*, o STM NUCLEO-L496, que possui um processador *ARM Cortex-M4*. Essa placa foi desenvolvida pela fabricante de microeletrônicos ST, com a intenção de atrair o público acostumado com desenvolvimento em *Arduino*. Na figura 13, foi trocado apenas a placa do CLP e alterado a seleção da placa para NUCLEO-L496 na interface *Arduino IDE*.

Figura 13 – Tempos de cada rotina em um NUCLEO-L496 com *Arduino*.

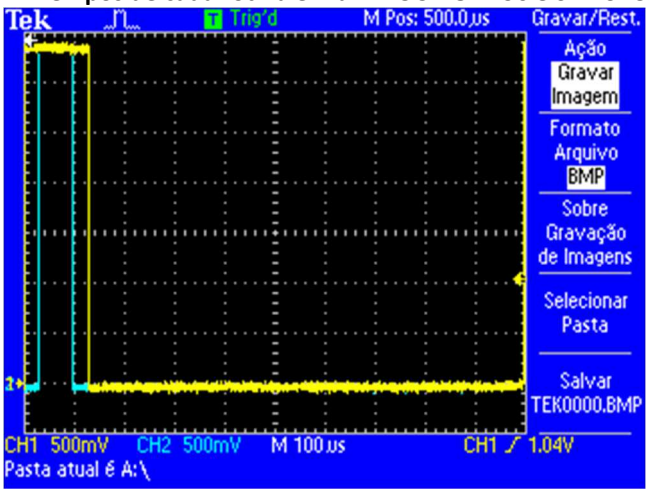


Fonte: Elaborado pelo Autor

Nessa figura nota-se que o programa de varredura é executado com o tempo similar ao *Cortex-M3*, apesar do processador em questão possuir um Processador de Sinais Digitais (DSP). Observou-se que os tempos de leitura das entradas analógicas se mantiveram muito próximos aos Arduino Uno e Mega, apesar da arquitetura suportar uma capacidade de captura mais rápida.

Neste próximo ensaio, o código fonte do processo foi adaptado para usar o programa STM32CubeIDE, uma solução de *software* de desenvolvimento fornecido pelo fabricante do NUCLEO-L496. A figura 14 mostra a performance do NUCLEO-L496 com a camada HAL da STM32Cube programada para a aplicação.

Figura 14 – Tempos de cada rotina em um NUCLEO-L496 e STM32Cube.

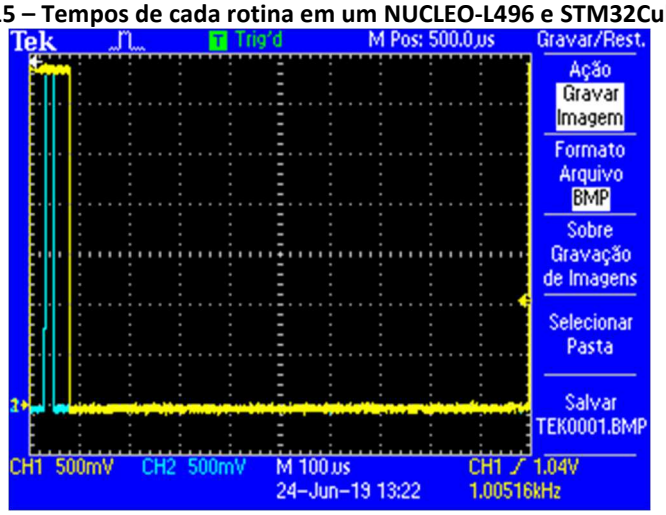


Fonte: Elaborado pelo Autor

No cenário da figura 14, podemos observar o ganho de tempo com as configurações de *hardware* programadas no STM32CubeMX, agilizando os processos de varredura de entradas e de saídas.

E por último, foi ajustado a configuração de otimização do código para a opção "Otimizar para Velocidade" (*Optimize for speed*), opção presente na interface da IDE, em "*Propriedities->C/C++ Build->Settings",* aba *"Tool Settings",* menu *"MCUGCC Compiler->Optimization",* item *"Optimization level->Optimize for speed (-Ofast)".*

Figura 15 – Tempos de cada rotina em um NUCLEO-L496 e STM32Cube com otimização ativa.



Fonte: Elaborado pelo Autor

Como pode ser observado na figura 15, graças a estas novas configurações de *software*, obteve-se maior velocidade na execução do cálculo em ponto flutuante. Outro ganho considerável foi a possibilidade de usar a ferramenta *Serial Wire Debug* (SWD), presentes nas arquiteturas *ARM Cortex-M* e suportado pela interface do programa

STM32CubeIDE, que permite realizar testes e validações do código com a placa funcionando em sua aplicação final, como o uso do TRACE, levando em consideração todos os devidos cuidados a serem tomados durante tal procedimento.

2.3 Trabalhos futuros

Do ponto de vista do software, pode-se desenvolver uma pilha de protocolo Modbus para aplicações de controle em rede e comunicação com supervisórios, usando a interface serial RS-485 embutida no simulador de CLP cujo *hardware* já foi desenvolvido neste trabalho.

Outra sugestão é modificar as entradas e saídas discretas para permitir a conexão de dispositivos via PWM, modulação por largura de pulso, como servomotores ou sensores de temperatura e umidade.

Por fim, pode-se fazer o programa do processo para diferentes fabricantes de microcontroladores e realizar novos ensaios para comparação de performance e complementar os testes deste trabalho.

CONCLUSÃO

Após os estudos das tecnologias de microcontroladores disponíveis para a concepção do projeto do simulador de CLP, foi concluído que, apesar das facilidades que a ferramenta Arduino provém, tanto no *hardware* quanto no *software*, existem necessidades que só podem ser atendidas por outras tecnologias, como o uso de técnicas e ferramentas que permitam extrair o máximo de desempenho do sistema em sua aplicação e a testabilidade do *software* por meio da ferramenta TRACE, não presentes na plataforma, o que evidencia as limitações do Arduino no uso acadêmico para o desenvolvimento de projeto por pessoas que não tenham vivência com a tecnologia e também para o estudo de arquiteturas de microcontroladores, já que a ferramenta não permite uma maior interação do aluno com o *hardware*.

REFERÊNCIAS

ARDUINO. **Arduino DUE**, 2019. Disponível em: https://store.arduino.cc/usa/arduino-due. Acesso em: 13 jun. 2019.

ARDUINO. **Arduino IDE**, 2019. Disponível em: https://www.arduino.cc/en/Main/Software. Acesso em: 10 mai. 2019.

BOLTON, William. ***Programmable logic controllers*.** 5. ed. Oxford: Newnes, 2009. 416 p.

GANSSLE, Jack e BARR, Michael. ***Embedded systems dictionary*.** 1. ed. San Francisco CA: Crc Press, 2003. 256 p.

OSHANA, Robert. ***Software Engineering for Embedded Systems:*** *Methods, Practical Techniques, and Applications.* 1. ed. Waltham MA: Editora Newnes, 2013. 1200 p.

PETRUZELLA, Frank D. **Controladores lógicos programáveis**. 4. ed. Porto Alegre RS: Editora Bookman, 2014. 398 p.

STMICROELECTRONICS. ***STM32CubeIDE 1.0 development tool***. Disponível em: https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-ides/stm32cubeide.html. Acesso em: 25 jun. 2019

TEXAS INSTRUMENTS. **ULN2803A *Darlington Transistors Array***, revisado em fevereiro de 2017. Disponível em: http://www.ti.com/lit/ds/symlink/uln2803a.pdf. Acesso em: 13 jun. 2019.

AGRADECIMENTOS

Agradeço primeiramente a Deus pelas bênçãos em minha carreira, a ajuda dos colegas de sala e tutores. Agradeço também o apoio da família, que sempre esteve presente durante todas as fases de meu aprendizado.

Sobre os autores:

_____

[i] VÍTOR EDUARDO SABADINE DA CRUZ

Possui graduação em Engenharia de Computação pela Universidade São Judas Tadeu (2015). Atualmente está cursando Pós-graduação em Automação Industrial no Senai Armando de Arruda Pereira (2018).

[ii] PAULO SEBASTIÃO LADIVEZ

Possui graduação em Engenharia Elétrica pela Universidade Mogi das Cruzes (1984) com especialização em Tecnologias e Sistemas de Informação pela Universidade Federal do ABC (2013). Atualmente é professor da Faculdade SENAI de Tecnologia Mecatrônica, lecionando as disciplinas Projetos, Microcontroladores, Linguagem de Programação no curso Tecnológico em Mecatrônica Industrial e na Pós-Graduação em Automação Industrial. Tem experiência na área de Engenharia Eletrônica, com ênfase em Automação Industrial e Mecatrônica, atuando principalmente nos seguintes temas: Mecatrônica, Manufatura Digital, Redes Industriais, Automação Industrial, Microcontroladores e Controle. CV: http://lattes.cnpq.br/7235073442326291

[iii] JOSE ROBERTO DOS SANTOS

Atualmente ministra aulas na pós-graduação de Indústria 4.0 e na graduação em Tecnologia em Mecatrônica na Faculdade SENAI de Tecnologia Mecatrônica, que fica no SENAI Armando de Arruda Pereira. Assessora também o Instituto SENAI de Tecnologia Metalmecânica em projetos industriais com foco na Indústria 4.0. Durante 9 anos ministrou aulas pelo SENAI-SP, nos cursos de técnico em eletroeletrônica, cursos de aprendizagem industrial eletricista de manutenção e mecânico de usinagem, além de Formação Inicial e Continuada (FIC) com cursos voltados a área de redes de computadores e programação, possui treinamento de Linux, cisco e Microsoft. Possui Pós-graduação na área de segurança da informação pela Uninove (2016), graduação em tecnologia da informação e bacharel em sistema da informação (2009), além de superior em Automação industrial. Tem experiência na área de Segurança da informação, administração de ambientes de redes Windows e Linux, automação indústria.

[vi] SERGIO LUIZ VOLPIANO

Graduação em Engenharia Industrial Elétrica pela Universidade Santa Cecília dos Bandeirantes com especialização em Eletrônica Industrial pela Universidade São Judas Tadeu e Mestre em Engenharia Elétrica na área de Sistema de Potência pela Universidade de São Paulo. Trabalha como professor da Faculdade Senai de Tecnologia em Mecatrônica Industrial e na Faculdade de Tecnologia do Estado de São Paulo, (Fatec) em São Bernardo do Campo lecionando as disciplinas de Eletrônica de Potência, Máquinas Elétricas, Acionamento Eletrônico de Máquinas Elétricas, Eletrônica Digital, Instalações Elétricas Industriais, Eletrônica Geral e Análise de Circuitos em Corrente Contínua e Corrente Alternada nos cursos de graduação e Pós-graduação. Possui experiência na área de Engenharia Elétrica, com ênfase em Máquinas Elétricas, Conversão Eletromecânica de Energia, Sistemas de Potência e Eletrônica Industrial.

[v] VICENTE GOMES DE OLIVEIRA JUNIOR

Possui graduação em Tecnologia Elétrica pela Universidade Presbiteriana Mackenzie (1982). Complemento em pedagogia na Universidade Metodista de Piracicaba (1999), Mestrado em Engenharia Mecânica pela Universidade Estadual de Campinas (2006). Atualmente é professor na área de automação industrial da Faculdade Senai de Tecnologia Mecatrônica nos cursos de graduação e pós-graduação. Tem experiência na área de Automação Industrial, atuando principalmente nos seguintes temas: pneumática, eletropneumática, hidráulica, eletrohidráulica, controlador programável, robótica básica, sistema supervisório, algumas redes industriais.