

Laboratório Virtual de Sistemas Embarcados: Uma Abordagem de Ensino Remoto Usando Softwares Gratuitos

Paulo Felipe Possa Parreira * Taynan Roger Silva **
Heitor Augusto de Novais * Eduardo José da Silva Luz ***,***
Alan Kardek Rêgo Segundo *,**,*

* Departamento de Controle e Automação, Universidade Federal de Ouro Preto, Ouro Preto - MG, (e-mails: paulof@ufop.edu.br, heitor.novais@aluno.ufop.edu.br, alankardek@ufop.edu.br)

** Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração, Universidade Federal de Ouro Preto e Instituto Tecnológico Vale, Ouro Preto - MG, (e-mails: taynan.silva@aluno.itv.org, eduluz@ufop.edu.br).

*** Programa de Pós Graduação em Ciências da Computação, Universidade Federal de Ouro Preto, Ouro Preto - MG

Abstract: In the teaching of Embedded Systems, laboratory practices are essential for the training of new professionals. With the pandemic caused by the new coronavirus, social isolation compromises these present activities. Thus, this work presents a virtual laboratory based on free software applied to grad and undergrad courses to make feasible teaching Embedded Systems. The development of laboratory practices uses the Project-Based Learning methodology, aiming for transversal skills development by students. The work shows two project examples developed by students using the virtual laboratory. Results show that the virtual environment and the project-based approach are potential tools for the student's engagement in the project's development during the remote period since they fulfill the required expectations.

Resumo: No ensino de Sistemas Embarcados a realização de práticas em laboratórios é fundamental para a formação de novos profissionais. Com a pandemia causada pelo novo coronavírus a oferta dessas atividades foi prejudicada devido à necessidade de isolamento social. Assim, visando dar continuidade ao ensino desta disciplina de forma remota em cursos de graduação e pós-graduação, no presente trabalho apresenta-se um laboratório virtual de Sistemas Embarcados baseado em softwares gratuitos. Para o desenvolvimento de habilidades transversais pelos alunos foi utilizada a metodologia de Aprendizagem Baseada em Projetos nas atividades práticas do laboratório virtual. Dois exemplos de projetos desenvolvidos pelos discentes usando esse laboratório são apresentados neste artigo. Os resultados evidenciam que os recursos e a metodologia utilizada foram ferramentas potenciais para o engajamento dos discentes na execução dos trabalhos durante a oferta da disciplina de forma remota, uma vez que todos os alunos matriculados foram capazes de atingir os objetivos propostos para os projetos.

Keywords: Embedded Systems; Education; Engineering Teaching; Virtual Laboratory; Project-Based Learning; Free Educational Resources.

Palavras-chaves: Sistemas Embarcados; Educação; Ensino em Engenharia; Laboratório Virtual; Aprendizagem Baseada em Projeto; Recursos Educacionais Gratuitos.

1. INTRODUÇÃO

Devido à massiva utilização de sistemas microcontrolados em diversos produtos tecnológicos, é crescente o interesse de estudantes e entusiastas nesta área de atuação. Dada a necessidade do mercado pela formação de profissionais da área de sistemas embarcados, diversas instituições de ensino superior têm ofertado disciplinas de sistemas embarcados tanto em cursos de graduação como em cursos de pós-graduação dentro das áreas de engenharia.

Geralmente, espera-se que ao final da disciplina os alunos sejam capazes de projetar um sistema embarcado de acordo com os requisitos do sistema. Para isso, apresenta-se durante as aulas diversas técnicas de programação de microcontroladores, de simulação de projetos, de desenvolvimento de hardware e de criação de sistemas supervisórios.

Por ser caracterizada como uma área interdisciplinar e multidisciplinar, que engloba conhecimentos em eletrônica, algoritmos, sistemas operacionais, processamento de sinais e teoria de controle, o uso de laboratórios no ensino de

sistemas embarcados é fundamental para o processo de ensino-aprendizagem dos estudantes uma vez que, por meio das experiências realizadas nesses ambientes, os alunos têm a oportunidade de consolidar os conhecimentos teóricos do curso para a resolução de problemas reais da vida profissional. Dessa forma, diversas metodologias de ensino em laboratórios têm sido desenvolvidas por educadores para atrair e motivar os estudantes em uma práxis pedagógica que concilie os aspectos de teoria em situações práticas (Matos et al., 2019b; Segundo et al., 2015a,b).

Com a necessidade do isolamento social imposta pela pandemia do novo coronavírus (Sars-CoV-2) esse desafio de integração por meio de atividades laboratoriais tornou-se ainda maior, visto que muitas universidades passaram a ofertar essas disciplinas apenas de forma remota. Se nos laboratórios físicos de sistemas embarcados os estudantes tinham acesso à diversas plantas didáticas e contato direto com hardwares previamente adquiridos pelas instituições, com a pandemia essa interação ficou limitada ao computador pessoal do aluno. Aliado a isso, esse novo contexto evidenciou também a dificuldade no uso de softwares livres (free software) e gratuitos (freeware) para os estudantes, visto que a maioria das plataformas experimentais disponíveis e aplicáveis para a graduação e pós-graduação são muito caras e/ou tem uma arquitetura de hardware e software fechada (Torga et al., 2019).

Dentre as alternativas que visam complementar ou substituir a utilização de laboratórios físicos, as mais utilizadas são os laboratórios remotos e os laboratórios virtuais, ou a combinação de ambos (Bencomo, 2004).

Os laboratórios virtuais utilizam uma abordagem puramente via software para realizar a simulação da dinâmica de diversos tipos de plataformas físicas. Essa simulação normalmente é baseada em modelos matemáticos e seu grau de similaridade com o processo real depende da implementação (Karakasidis, 2013). Comparado com o modelo remoto, os laboratórios virtuais possuem um menor custo de implantação e manutenção, além de ser possível a realização de vários experimentos simultâneos e uma maior disponibilidade entre os estudantes.

Assim, na busca pela continuidade do ensino de Sistemas Embarcados adaptado ao novo contexto, no presente trabalho apresenta-se um projeto de laboratório virtual de sistemas embarcados a partir de softwares educacionais livres e gratuitos. A proposta do laboratório utiliza uma didática que mescla o ensino teórico com a metodologia de Aprendizagem Baseada em Projetos (PBL - *Project-Based Learning*), sendo esta uma abordagem que fundamenta-se no uso contextualizado de uma situação para o aprendizado autodirigido (Matos et al., 2019a). Nessa metodologia o professor assume a função de mediador da aprendizagem enquanto os estudantes assumem um papel ativo e central nos processos de aprendizagem visto que eles próprios são os responsáveis pela resolução dos problemas.

Dois exemplos de projetos são apresentados para ilustrar a estrutura flexível e versátil dos softwares selecionados. Os resultados relatados são experiências da utilização do laboratório virtual por alunos de graduação e pós-graduação em engenharia de controle e automação de uma universidade federal no segundo semestre de 2020.

2. SOFTWARES E BIBLIOTECAS UTILIZADOS NO LABORATÓRIO VIRTUAL

Para o laboratório virtual de Sistemas Embarcados proposto neste trabalho é utilizado o software emulador de placas didáticas *PICSimLab* junto com o IDE (ambiente de desenvolvimento integrado) *MPLAB X* e o compilador *XC8*. Esses programas foram escolhidos devido à característica acessível de uso gratuito dos softwares, sendo o primeiro um software livre/gratuito e os demais softwares gratuitos. A placa escolhida para emulação dos projetos é a placa virtual *PICGenios*, e o microcontrolador sugerido aos alunos é o *PIC18F4520*, que contém um processador de 8 bits. Para o desenvolvimento de sistemas supervisórios é proposto o uso do *Visual Studio Community* para aplicações baseadas em *Windows Form*. Da mesma forma, o protocolo de comunicação escolhido para o desenvolvimento dos projetos é o protocolo industrial *Modbus RTU*, que pode ser ensinado por meio de simulações com bibliotecas específicas, como *Easymodbus*, *NModbus* e *Modbus-Slave-for-PIC18F*. Por fim, para a emulação de portas serial no computador pessoal do aluno é utilizado o software *VSPE* que cria portas serial virtuais dentro do sistema operacional operado. Os programas escolhidos assim como a placa virtual, o microcontrolador e o protocolo de comunicação citados são brevemente relatados a seguir.

2.1 *PICSimLab*

O *PICSimLab*¹ (acrônimo de PIC Simulator Laboratory) é um software livre desenvolvido pelo Prof. Luis Claudio Gambôa Lopes, de código aberto e disponível nas plataformas Linux e Windows para simulação de sistemas microcontrolados. Trata-se de um emulador de tempo real de placas de desenvolvimento com suporte à depuração pelas IDEs *MPLAB X* e *Arduino*. O software permite a utilização de placas prontas e também a criação de modelos próprios por meio de componentes isolados em protoboards virtuais.

Atualmente o *PICSimLab* conta com diversas placas passíveis de emulação como, por exemplo, as placas *Arduino Uno*, *McLab1*, *McLab2* e a placa *PICGenios*. Tais placas possuem suporte para diversos modelos de PIC (de 8 e 16 bits) assim como suporte para o microcontrolador *ATmega328p*. Por meio do *PICSimLab* é possível realizar a simulação de diversas aplicações reais como, por exemplo, a análise de tensões de saída usando um osciloscópio virtual. O programa permite também a simulação de protocolos de comunicação pela porta serial. Por ser um software livre, os estudantes podem utilizá-lo de forma gratuita em seus computadores pessoais para a realização de práticas de ensino de microcontroladores.

A Placa Virtual *PICGenios* é uma emulação do kit educacional *PICGenios* dentro do software *PICSimLab*. Essa placa é compatível com diversos modelos de PIC da família 18F e 16F e contém diversos periféricos acoplados para pronta utilização, conforme exibido na Figura 1. Dentre alguns componentes presentes na placa estão quatro displays de 7 segmentos, um LCD, um teclado matricial, potênci-

¹ *PICSimLab* - Disponível em: <http://sourceforge.net/projects/picsim>. Acesso em: 29 ago. 2021

metros, uma ventoinha, uma resistência de aquecimento e um sensor de temperatura LM35.



Figura 1. Emulação da placa PICGenios dentro do software PICSimLab.

O PIC18F4520 é um microcontrolador desenvolvido pela Microchip, de arquitetura Harvard de 8 bits e 40 pinos. É um microcontrolador de baixo custo que consegue atender uma grande gama de aplicações. Possui 4 temporizadores, 1 conversor A/D, sistema PWM, suporte a interrupções e comunicação serial. É um dos chips que pode ser simulado dentro do PICSimLab nas placas virtuais PICGenios, Mclab2 e PQDB.

2.2 MPLAB X

O MPLAB X² é um IDE criado pela Microchip para auxiliar o desenvolvimento de códigos para diversos microcontroladores da família PIC. Baseado no Netbeans, ele possui versões tanto para uso em sistemas operacionais Linux quanto para Windows. Apesar de não ser um software livre, possui distribuição gratuita com diversos recursos livres para aplicações complexas. É uma excelente ferramenta para o desenvolvimento de software embarcado, principalmente por permitir o uso de diversos compiladores, assim como a integração com depuradores externos, como é feito no laboratório virtual com o depurador do PICSimLab. Ele permite também a geração automática de configurações para placas da Microchip. Além disso, possui uma interface amigável e facilmente configurável, sendo uma boa alternativa para estudantes e desenvolvedores iniciantes.

Dentre os compiladores suportados estão os compiladores da família XC de compiladores Microchip. No caso do laboratório virtual é utilizado o MPLAB XC8 — versão de 8 bits do compilador citado. A utilização desse compilador é gratuita para todos os PICs de 8 bits.

2.3 Visual Studio Community

O Visual Studio Community³ é um IDE da Microsoft para o desenvolvimento de software, especialmente dedicado ao

² Microchip - Disponível em: <http://www.microchip.com>. Acesso em: 29 ago. 2021

³ Visual Studio Community - Disponível em: <http://visualstudio.microsoft.com/pt-br/vs/community>. Acesso em: 29 ago. 2021

.NET Framework e às linguagens Visual Basic, C, C++, C# e F#. Por ele, é possível criar aplicativos baseados em Windows Form que exibem informações, solicitam entrada de dados de usuários e se comunicam com dispositivos remotos pela rede. É um software gratuito e de uso irrestrito, sendo uma versão simplificada do Visual Studio Professional e do Visual Studio Enterprise. No ensino de Sistemas Embarcados, pode-se utilizá-lo para a criação de sistemas supervisórios (SCADA) e sistemas de interface homem-máquina (IHM).

2.4 Modbus

Modbus é um protocolo de comunicação mantido pela Modicon Corporation. É atualmente o padrão mais utilizado em sistemas de automação industrial. O protocolo não define a interface física da comunicação, mas a forma como as informações são transportadas pela rede. Atualmente suporta os modos RTU, ASCII e TCP, além das tradicionais interfaces RS-232, RS-422, RS-485 e Ethernet (Peng et al., 2008).

O protocolo funciona por meio de um sistema mestre-escravo e temporizadores para tráfego de mensagens, permitindo a consistência dos dados com base na verificação pelo mestre dos estados dos escravos conectados.

Para o laboratório virtual, o Modbus RTU foi escolhido por ser um protocolo livre e também por ser um padrão industrial, permitindo que os alunos tenham uma experiência prática do que é utilizado no mercado de trabalho.

Para a utilização do protocolo no Visual Studio, para aplicações em sistemas supervisórios, são indicadas as bibliotecas EasyModbus⁴ e NModbus⁵ implementadas em C#. Já para o sistema embarcado é indicada a biblioteca Modbus Slave For PIC18F⁶, implementada em C.

2.5 Virtual Serial Ports Emulator

O Virtual Serial Ports Emulator⁷ (VSPE) é um software gratuito na versão de 32 bits que emula portas serial físicas dentro do sistema operacional operado. O uso do VSPE permite criar um par de dispositivos de porta serial virtual para comunicação entre o PicSimLab e o sistema supervisório criado dentro do Visual Studio Community.

3. METODOLOGIA

Para a utilização do laboratório virtual são propostos dois projetos nos quais os alunos devem utilizar os conceitos teóricos aprendidos durante o curso para a resolução de problemas práticos. Conforme descrito na Seção 1, neste trabalho utiliza-se uma abordagem que mescla o ensino teórico com a metodologia PBL. Esse tipo de aprendizagem é uma metodologia ativa que leva em consideração

⁴ EasyModBus - Disponível em: <http://easymodbustcp.net/en>, Acesso em: 29 ago. 2021

⁵ NModbus - Disponível em: <http://github.com/NModbus/NModbus>. Acesso em: 29 ago. 2021

⁶ Modbus Slave for PIC18F - Disponível em: <http://github.com/whchoi/Modbus-Slave-for-PIC18F>. Acesso em: 29 ago. 2021

⁷ VSP Emulator - Disponível em: <http://www.eterlogic.com/Products.VSPE.html>. Acesso em: 29 ago. 2021

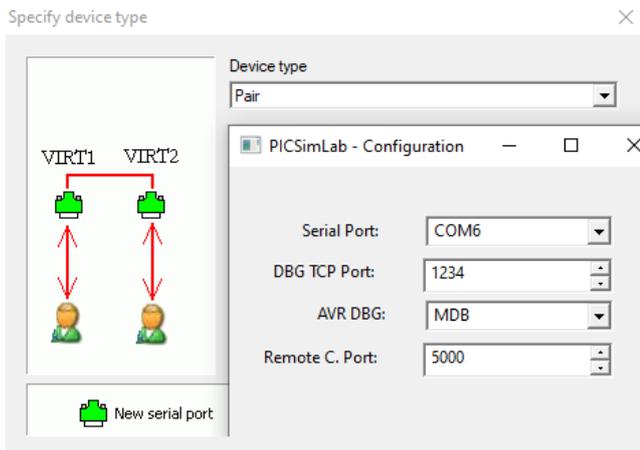


Figura 2. Emulação de par de portas serial para comunicação entre o PICSimLab e o sistema supervisorio.

o desenvolvimento da autonomia do estudante a partir da abordagem de projetos baseados em problemas reais. Segundo Souza et al. (2013), a metodologia ajuda os estudantes a visualizarem os conceitos, coletar e interpretar dados, e ligar a teoria com o mundo real. Esse conjunto de competências, associada a um projeto real, incentiva os alunos a aprender a aprender (Souza et al., 2013).

No início da disciplina, são apresentados dois tipos de projetos para que os estudantes executem ao longo do semestre, com base nas orientações e direcionamentos do professor, nos conteúdos das aulas teóricas e em outros conhecimentos adquiridos durante o curso. Visando a autonomia dos estudantes, durante o desenvolvimento dos projetos os alunos têm oportunidade de propor diferentes soluções para executá-los. Nesse sentido, uma apresentação prévia dos projetos é realizada pelos estudantes no meio do semestre, contendo o firmware do microcontrolador e a simulação preliminar do sistema na placa virtual. Essa etapa oportuniza uma discussão crítica e o repasse de conhecimentos específicos a cada grupo pelo professor. No final do semestre, os estudantes apresentam a resolução final dos projetos, contendo o software supervisorio e contemplando as pendências apontadas na etapa anterior.

Uma vez que o ensino de Sistemas Embarcados é ofertado tanto em cursos de Engenharia da Computação como em cursos de Engenharia Elétrica, Engenharia Eletrônica e Engenharia de Controle e Automação, os projetos sugeridos podem ser utilizados individualmente ou em conjunto, de acordo com os conteúdos programáticos de cada curso. Desta forma, no primeiro projeto os alunos são estimulados a desenvolverem técnicas de programação de microcontroladores na criação de sistemas embarcados que se comunicam com sistemas supervisorios, enquanto que no segundo projeto os alunos são estimulados a utilizarem os conhecimentos de sistemas embarcados para a solução de um problema de Teoria de Controle.

Os projetos sugeridos, assim como o roteiro das atividades são descritas a seguir.

3.1 Sistema de Bombeamento de Água

Para o desenvolvimento dos conhecimentos de leitura e acionamento de entradas e saídas digitais, interrupções,

temporizadores, contadores e comunicação entre dispositivos é proposto aos alunos uma atividade de acionamento de um sistema de bombeamento de água para o controle de nível de dois reservatórios.

Trata-se de um sistema composto por duas caixas d'água, uma inferior e outra superior, sendo ambas equipadas com sensores analógicos de nível. O sistema pode atuar de forma manual ou automática. No modo manual existe apenas a ação de ligar ou desligar a bomba que fornece água para a caixa superior. Já no modo automático, o sistema apenas pode ligar a bomba de acordo com os seguintes requisitos: (i) o nível da água na caixa inferior deve estar acima de 10% e (ii) o nível da água da caixa superior deve estar acima de 50%. A bomba deve ser acionada até que a caixa superior atinja o nível de 90% ou que a caixa inferior chegue no limite de 10%.

Nesta atividade o principal desafio proposto é a criação de um sistema supervisorio no Windows Form com interface gráfica e comunicação Modbus RTU. Para que a aplicação execute tarefas como exibir informações, armazenar dados e comunicar-se em rede de forma simultânea, é sugerido aos alunos que eles apliquem no desenvolvimento do código alguns conceitos de computação em tempo real como o paralelismo e a sincronia.

Além disso, visando o aprendizado de técnicas de programação de microcontroladores, é requisitado que os alunos programem o sistema embarcado de forma escalável e manutenível, utilizando boas práticas de programação e arquitetura como, por exemplo, o Princípio da Responsabilidade Única que tem como regra definir uma, e apenas uma, responsabilidade para uma classe (Martin, 2014).

Na Seção 4.1 são exibidos alguns sistemas supervisorios criados pelos alunos para a resolução desse projeto.

3.2 Sistema de Controle de Temperatura

Para o desenvolvimento de atividades práticas de sistemas de controle dentro do laboratório virtual de sistemas embarcados é proposto aos alunos um projeto de controle de temperatura da resistência de aquecimento presente na placa virtual PICGenios.

Embora a placa utilizada dentro do programa PICSimLab seja uma simulação virtual da placa real, ela simula de forma realista o comportamento dinâmico dos sensores e periféricos presentes, o que permite que atividades de controle sejam realizadas dentro do laboratório virtual.

Nessa atividade prática, o calor gerado pela resistência de aquecimento pode ser mensurado pelo sensor de temperatura LM35, que está ligado ao canal A/D do PIC18F4520. Esse sensor mede temperaturas dentro da faixa de -55°C à 150°C , com variação típica de 10 mV por grau Celsius. Uma ventoinha instalada acima da própria resistência (Figura 3) é capaz de refrigerar a resistência de aquecimento alterando a dinâmica de variação de temperatura.

A ventoinha pode ter sua velocidade de giro controlada via PWM por meio do canal CCP1 do PIC18F4520. Assim, se a ventoinha estiver ligada simultaneamente com a resistência de aquecimento, quanto maior a velocidade de giro da ventoinha menor será a temperatura da resistência de aquecimento.

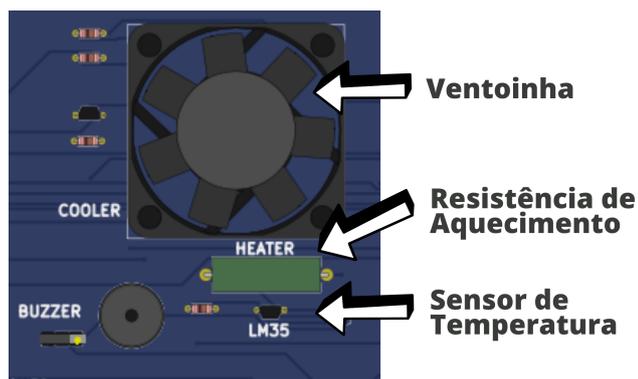


Figura 3. Sistema dinâmico ventoinha-resistência-de-aquecimento da placa PICGenios.

A proposta é que os alunos identifiquem e modelem o sistema dinâmico ventoinha-resistência-de-aquecimento para, em seguida, projetar e implementar um compensador PID digital com estratégia de controle para o seguimento de referência (temperatura) em tempo discreto.

O roteiro da atividade prática criada dentro do laboratório virtual de sistemas embarcados é dividido em duas etapas: (i) identificação e modelagem do sistema e (ii) projeto e implementação do controlador PID. As abordagens metodológicas sugeridas para os alunos nessas duas etapas são descritas a seguir. Já na Seção 4.2 são relatados alguns resultados obtidos por alunos para a resolução desse projeto.

Identificação e Modelagem Para a identificação do sistema representado, é sugerido aos alunos que eles façam a modelagem de forma empírica, isto é, utilizando-se um método de modelagem caixa preta. O método recomendado para esta atividade é o método de identificação por curva de reação, em que a função de transferência do sistema é estimada aplicando-se um sinal de entrada em malha aberta e analisando-se a resposta de saída.

Esse método é adequado para o laboratório virtual de sistemas embarcados uma vez que a dinâmica da resistência de aquecimento em estudo não apresenta integradores, sendo sua resposta ao degrau unitário uma curva característica em "S", o que faz com que o sistema possa ser modelado como uma função de transferência de 1ª ordem, conforme Equação (1), em que M_u é o ganho estático, τ_d é o tempo morto e τ é a constante de tempo dominante do sistema. Estando o valor de temperatura da resistência de aquecimento em estado estacionário é sugerido aos alunos que apliquem no sistema de rotação de giro da ventoinha um degrau (positivo ou negativo) de aproximadamente 20% do fundo de escala da ventoinha, conforme Figura 4, para que se obtenha os parâmetros M_u , τ_d e τ .

$$G(s) = \frac{M_u \cdot e^{-s \cdot \tau_d}}{1 + s \cdot \tau} \quad (1)$$

Controlador PID Para a implementação do controlador PID é sugerido aos alunos que eles projetem uma estratégia de controle a partir do método de ajuste conhecido como Ziegler-Nichols. As relações de sintonia do método são descritas na Tabela 1, sendo a equação do controlador PID definida por

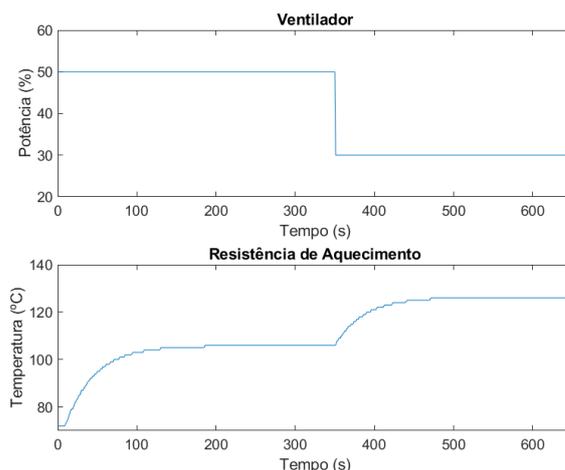


Figura 4. Modelagem caixa preta via método de identificação por curva de reação. Fonte: Autores.

$$G_c(s) = k_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (2)$$

Tabela 1. Parâmetros do Método de Ziegler-Nichols.

Controlador	k_p	T_i	T_d
P	$1,0 \cdot \frac{\tau}{M_u \cdot \tau_d}$	-	-
PI	$0,9 \cdot \frac{\tau}{M_u \cdot \tau_d}$	$3,33 \cdot \tau_d$	-
PID	$1,2 \cdot \frac{\tau}{M_u \cdot \tau_d}$	$2,00 \cdot \tau_d$	$0,50 \cdot \tau_d$

Nesse ponto, é importante que os alunos discutam qual é o melhor controlador para planta (P, PI, PD ou PID) tendo em vista que o objetivo da sintonia proposta é o ajuste do valor final da temperatura em regime permanente, com erro mínimo (limite máximo sugerido de 5°C).

Uma vez que o microcontrolador utilizado no laboratório virtual é um sistema digital, é sugerido aos alunos que discretizem o controlador projetado utilizando a aproximação de Tustin (transformada bilinear/integração trapezoidal).

4. RESULTADOS

4.1 Sistema de Bombeamento de Água

Seguindo as diretrizes propostas na Seção 3.1 é exibido na Figura 5 uma das telas de sistema supervisorio criada por um aluno do laboratório virtual. A aplicação foi desenvolvida de forma animada para que fosse possível reproduzir com fidelidade o processo real. Para isso, utilizou-se técnicas de animação, como transições de imagens para simular as dinâmicas do meio físico.

Para a comunicação entre o microcontrolador e a aplicação no Windows Form foi utilizada no lado do supervisorio a biblioteca NModbus4, que possui uma série de funções de leitura e escrita pré-definidas e que facilitam a conexão entre os dispositivos. Nessa configuração o sistema supervisorio foi configurado como o mestre Modbus, enquanto a placa virtual PICGenios foi configurada como o dispositivo escravo. Dessa forma, no lado do sistema embarcado

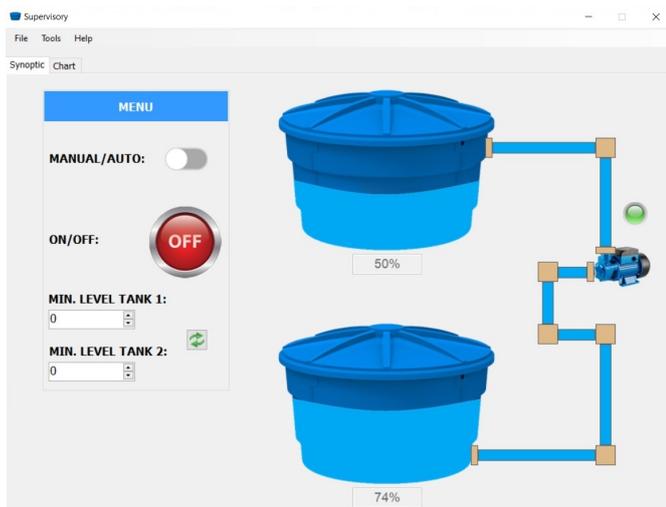


Figura 5. Sistema supervisorio desenvolvido para o projeto de bombeamento de água. Fonte: Autores.

foi utilizada a biblioteca Modbus Slave For PIC18F para configurar o sistema microcontrolado.

Sobre o uso dos conceitos de computação em tempo real na aplicação, para as diversas tarefas implementadas foram configuradas threads dedicadas concorrentes com a thread principal para, por exemplo, amostrar as medições em um gráfico e exibir os dados em tempo real, conforme realizado na Figura 6. Assim, não era necessário aguardar o término de uma tarefa para iniciar outra. Apesar da operação de leitura não ser uma aplicação crítica, foram implementados semáforos nessas situações para evitar que fossem executadas simultaneamente operações de leitura e escrita, protegendo a aplicação de situações inesperadas.



Figura 6. Exibição de dados em tempo real. Fonte: Autores.

Por fim, sobre as técnicas de programação de microcontroladores utilizadas no laboratório virtual, além do Princípio da Responsabilidade Única, utilizou-se também os conceitos da Clean Architecture, conforme elucidado por Martin (2012). Tais conceitos constituem um padrão arquitetural que tem como objetivo promover a implementação de sistemas que favorecem a reusabilidade de código, a coesão, a independência de tecnologia e a estabilidade, por meio de uma arquitetura em camadas.

No estudo de caso em questão, o driver Modbus, o banco de dados e a interface gráfica constituíram a camada mais

externa do modelo Clean Architecture, representando as tecnologias utilizadas na aplicação. Na camada intermediária programou-se os serviços de interface entre a regra de negócio e as tecnologias utilizadas. Por último, no núcleo da camada no modelo Clean Architecture, implementou-se as regras de negócio e as entidades que representam o modelo de domínio da aplicação.

4.2 Sistema de Controle de Temperatura

Seguindo o roteiro estipulado na Seção 3.2 é exibido a seguir, na Figura 7, uma das modelagens possíveis para o projeto proposto. Tal modelo foi obtido por um aluno do laboratório virtual que optou por utilizar um controlador PI para a solução do sistema de controle da temperatura da resistência de aquecimento.

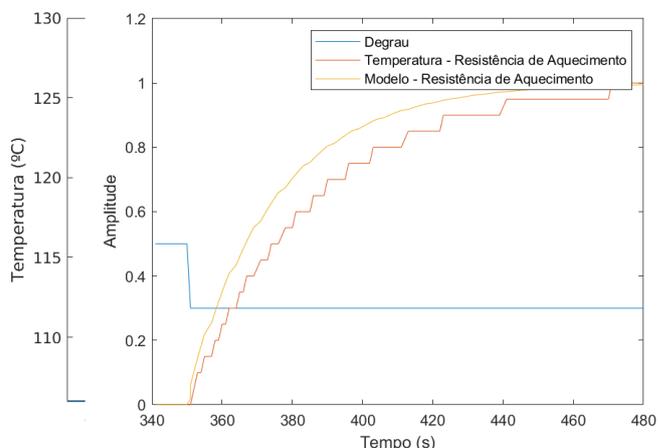


Figura 7. Modelagem obtida para obtenção dos parâmetros M_u , τ_d e τ . Fonte: Autores.

A equação do controlador discretizado com a aproximação de Tustin, para esse caso, foi definida por

$$u(k) = u(k-1) + k_p \cdot \left(\frac{2 \cdot T_i + T}{2 \cdot T_i} \right) \cdot e(k) + k_p \cdot \left(\frac{-2 \cdot T_i + T}{2 \cdot T_i} \right) \cdot e(k-1), \quad (3)$$

em que $u(k)$ é a potência da ventoinha no instante atual, $u(k-1)$ é a potência da ventoinha no instante anterior, T é o tempo de amostragem do sistema, $e(k)$ é o erro de temperatura no instante atual e $e(k-1)$ é o erro de temperatura no instante anterior.

Neste caso, implementou-se o algoritmo utilizando a biblioteca de acionamento da ventoinha disponibilizada como exemplo pelo desenvolvedor do PICSIMLab. A função utilizada (PWM_Set_Duty) foi capaz de alterar a velocidade de giro da ventoinha por meio do canal CCP1 do PIC18F4520, conforme relatado anteriormente. Uma vez que o PIC18F4520 é um microcontrolador de 8 bits, a função utilizada foi ajustada para saturar a potência de giro da ventoinha com argumento 255 ($2^8 - 1$) e zerar a potência com argumento nulo.

Como resultado para essa modelagem, obteve-se uma estratégia de controle eficaz, dentro da faixa de erro imposta (5°C) e tempo de resposta coerente com o modelado.

A validação do sistema de controle para seguimento de referência em 100 °C é exibida a seguir, na Figura 8, onde é mostrado também a atuação do sinal de controle no decorrer do tempo de simulação.

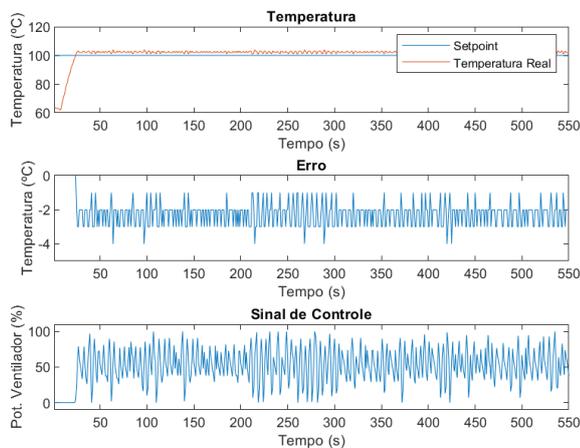


Figura 8. Validação do sistema de controle para setpoint de temperatura em 100 °C. Fonte: Autores.

Conforme relatado, uma vez que a dinâmica da resistência de aquecimento em estudo não apresenta integradores, percebe-se na Figura 8 oscilações da potência do ventilador em regime permanente. Isso acontece uma vez que, sem integradores, o sistema pode ser classificado como Tipo O, onde o erro em regime permanente para uma entrada em degrau é sempre finito. Nessa situação, para que o erro seja zero, deve-se colocar um integrador na malha.

Da mesma forma que no projeto tratado na Seção 3.1, para este projeto os alunos do laboratório também desenvolveram sistemas supervisórios utilizando a comunicação Modbus RTU entre dispositivos, conforme exemplificado na Figura 9.

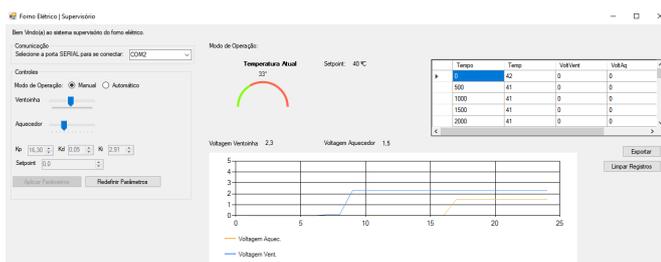


Figura 9. Sistema supervisório desenvolvido no laboratório virtual. Fonte: Autores.

5. CONSIDERAÇÕES FINAIS

O trabalho realizado apresentou uma alternativa para o ensino remoto de Sistemas Embarcados em cursos de graduação e pós-graduação em engenharia. A abordagem da criação de um laboratório virtual utilizando softwares livre e (ou) gratuitos, assim como a metodologia PBL, foram potenciais para atingir ao final do curso a capacidade dos estudantes projetarem sistemas embarcados e para abordar os conceitos previstos nas ementas das disciplinas dos cursos. O laboratório foi utilizado no segundo semestre de

2020 por 19 alunos, sendo que todos obtiveram bom índice de desempenho durante o desenvolvimento dos projetos propostos.

Sobre a plataforma e programas sugeridos pode-se concluir que eles são flexíveis e permitem a realização de diversas práticas de sistemas embarcados de forma totalmente remota, sem a ampliação de custos para a instituição, o que é fundamental para disciplinas que possuem um caráter majoritariamente prático.

Notou-se a importância do auxílio e orientação do professor durante o uso do laboratório virtual e desenvolvimento dos projetos pelos estudantes. Cabe ressaltar o potencial do presente trabalho em auxiliar os professores da área no uso dos softwares e bibliotecas apresentados tendo em vista o detalhamento abordado ao longo do manuscrito, que pode ser replicado em outros contextos de ensino.

Como trabalho futuro, pretende-se aplicar esta abordagem em um número maior de turmas, tanto de graduação quanto de pós-graduação, permitindo que os alunos avaliem o método de ensino e que, assim, seja possível validar o mesmo com outros projetos e condições de operação, principalmente com o ensino remoto sendo uma realidade enquanto a pandemia do novo coronavírus perdurar.

6. AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ), do Instituto Tecnológico Vale (ITV) e da Universidade Federal de Ouro Preto (UFOP).

REFERÊNCIAS

- Bencomo, S.D. (2004). Control learning: Present and future. *Annual Reviews in Control*, 28(1), 115–136.
- Karakasidis, T. (2013). Virtual and remote labs in higher education distance learning of physical and engineering sciences. In *2013 IEEE Global Engineering Education Conference (EDUCON)*, 798–807. IEEE.
- Martin, R. (2012). The clean architecture. URL <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>.
- Martin, R. (2014). The single responsibility principle. URL <https://blog.cleancoder.com/uncle-bob/2014/05/08/SingleResponsibilityPrinciple.html>.
- Matos, S., Gaigher Netto, G., Lage, V., Rêgo Segundo, A.K., and Braga, M. (2019a). Tratamento de dados: uma abordagem prática para aprendizagem de atenuação de ruídos e eliminação de outliers. doi:10.17648/sbai-2019-111570.
- Matos, S., Mattos, V., Leocádio, R., Rêgo Segundo, A.K., Cocota Júnior, J.A.N., Euzébio, T., and Viegas, D. (2019b). Bancada de transferência de calor para práticas de teoria de controle. doi:sbai-2019-111581.
- Peng, D.g., Zhang, H., Yang, L., and Li, H. (2008). Design and realization of modbus protocol based on embedded linux system. In *2008 International Conference on Embedded Software and Systems Symposia*, 275–280. IEEE.

- Segundo, A.K.R., Cocota, J.A.N., and Ferreira, D.V.M. (2015a). Development of an educational tool for control engineering. In *2015 IEEE Global Engineering Education Conference (EDUCON)*, 594–601. IEEE.
- Segundo, A.K.R., Cocota, J.A.N., Hilário, R.Q., de Oliveira Gomide, V., and Ferreira, D.V.M. (2015b). Low cost scada system for education. In *2015 IEEE Global Engineering Education Conference (EDUCON)*, 536–542. IEEE.
- Souza, I., Silva, S., Neves, D., Gomes Neves, D., Silveira, L., and Fernandes, M. (2013). Aprendizagem baseada em projetos aplicada ao ensino de sistemas embarcados.
- Torga, D., Cocota, J., Rêgo Segundo, A.K., and Martins, W. (2019). Desenvolvimento de uma bancada didática para práticas de controle de velocidade de motor de corrente contínua. doi:10.17648/sbai-2019-111612.