



DESENVOLVIMENTO DE UMA INTERFACE DE USUÁRIO PARA UM SISTEMA DE CONTROLE DE TEMPERATURA EDUCACIONAL COMPACTO, PORTÁTIL E DE BAIXO CUSTO

DOI: 10.37702/2175-957X.COBENGE.2024.5362

Autores: PEDRO HENRIQUE COUTINHO CONDE, DANIEL DE ALMEIDA FERNANDES, ANA SOPHIA CAVALCANTI ALVES VILAS BOAS

Resumo: Este artigo apresenta uma Interface de Usuário (IU) especialmente desenvolvida para simplificar o uso de um Sistema de Controle de Temperatura (SCT) educacional, compacto, portátil e de baixo custo, proposto como recurso didático para auxiliar no processo de ensino/aprendizagem de disciplinas de Engenharia Elétrica, principalmente aquelas relacionadas à área de controle automático de sistemas dinâmicos, porém não somente delas. O SCT auxilia na realização de demonstrações e experimentos que aproximam teoria e prática durante as aulas teóricas. Diversas teorias e técnicas podem ser exemplificadas, comparadas e testadas na prática. A IU substitui, com vantagens, uma IU desenvolvida inicialmente com base no Ambiente de Desenvolvimento Integrado (ADI) do Arduino. A nova IU foi escrita em linguagem de programação C#, através do ADI Visual Studio, aplicando-se o conceito de programação orientada a objetos. Por enquanto, ela funciona somente no sistema operacional Windows. Essa nova IU comunica-se com o SCT por meio de um protocolo de comunicação serial especialmente desenvolvido e permite a visualização em tempo real de alarmes, mensagens e valores de algumas das grandezas relativas à operação do SCT, além de apresentá-los de forma mais clara e objetiva.

Palavras-chave: Controle de temperatura, C#, Interface de usuário, Programação orientada a objetos, Sistema de controle automático, Visual Studio

DESENVOLVIMENTO DE UMA INTERFACE DE USUÁRIO PARA UM SISTEMA DE CONTROLE DE TEMPERATURA EDUCACIONAL COMPACTO, PORTÁTIL E DE BAIXO CUSTO

1 INTRODUÇÃO

Este artigo apresenta uma Interface de Usuário (IU) especialmente desenvolvida para simplificar o uso do Sistema de Controle de Temperatura (SCT) educacional compacto, portátil e de baixo custo proposto em Dalton; Lopes; Fernandes (2023b) e discutido em Dalton; Lopes; Fernandes (2023a) e Lopes; Dalton; Fernandes (2023). Ela deve substituir, com vantagens, a precária IU desenvolvida em um primeiro momento utilizando a ferramenta chamada “*Plotter serial*”, nativa do Ambiente de Desenvolvimento Integrado (ADI) — *Integrated Development Environment* (IDE) — do Arduino. Essa nova IU foi escrita em linguagem de programação C#, desenvolvida pela *Microsoft Corporation*, e, por enquanto, funciona somente no Sistema Operacional (SO) Windows. O desenvolvimento da IU foi o tema central de um projeto de Iniciação Científica (IC) que também serviu para solucionar algumas pendências apontadas nas seções “Conclusões” dos trabalhos de Dalton; Lopes; Fernandes (2023a; 2023b). Mais detalhes a respeito do SCT são fornecidos a seguir.

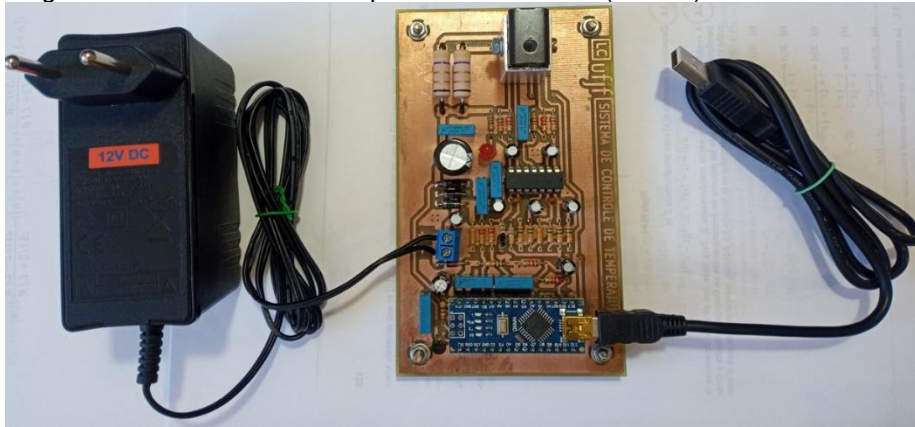
Os principais objetivos deste artigo são apresentar a nova IU que foi desenvolvida para o SCT e discutir alguns dos aspectos mais relevantes do seu desenvolvimento.

2 SISTEMA DE CONTROLE DE TEMPERATURA

O SCT, mostrado na Figura 1, é uma ferramenta prática bastante útil no apoio ao ensino de vários conteúdos de Engenharia Elétrica (ex.: circuitos elétricos, eletrônicas analógica e digital, microcontroladores, programação, entre outros), ainda que ela tenha sido criada mais voltada a servir de recurso didático auxiliar no processo de ensino/aprendizagem de disciplinas da área de controle automático (DALTON; LOPES; FERNANDES, 2023a; 2023b; LOPES; DALTON; FERNANDES, 2023). Diversas teorias e técnicas de controle automático de sistemas dinâmicos podem ser exemplificadas, comparadas e testadas na prática durante as aulas teóricas, aproximando teoria e prática dentro das salas de aula. Além disso, o funcionamento de cada parte do circuito eletrônico, a Placa de Circuito Impresso (PCI) e a programação do microcontrolador da placa Arduino Nano, entre outros aspectos interessantes, podem ser explorados durante as aulas teóricas de outras disciplinas também, isto é, de disciplinas não relacionadas à área de controle automático. O SCT é uma ferramenta versátil, segura, barata e fácil de construir e operar, que se tornou ainda mais fácil de operar através da IU dedicada apresentada neste artigo.

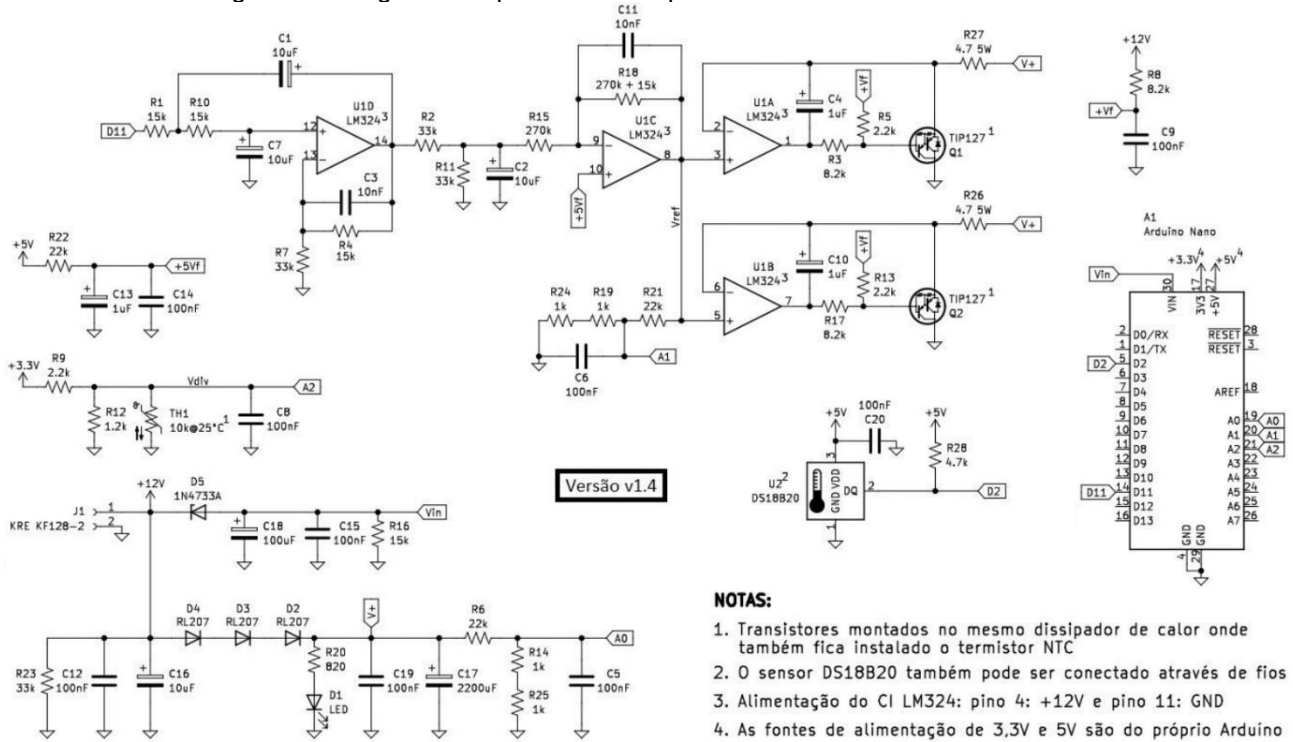
O circuito eletrônico, apresentado nos trabalhos supracitados, sofreu pequenas alterações com o objetivo de acelerar um pouco a resposta da planta. Houve basicamente um aumento da potência máxima que pode ser dissipada pelos transistores de potência, de modo que eles podem aquecer o dissipador de calor mais rapidamente, e tal modificação demandou outras ligeiras alterações no circuito eletrônico, cuja versão atual é mostrada na Figura 2. A fonte de alimentação (externa) deve ser mais potente do que era antes, ou seja, de 12 V, ≥ 24 W. A Figura 3 apresenta um diagrama que mescla um diagrama esquemático e um diagrama de blocos. A intenção com ele é facilitar a interpretação da Figura 2 e, dessa forma, facilitar a compreensão de todo o circuito eletrônico. Ênfase é dada à planta controlada, que ocupa toda a metade superior do diagrama. Considera-se, de maneira simplificada, que a planta possui somente uma entrada, $u(t)$, e uma saída, $y(t)$.

Figura 1 – SCT. Dimensões aproximadas da PCI (LxAxP): 131x76x45 mm.



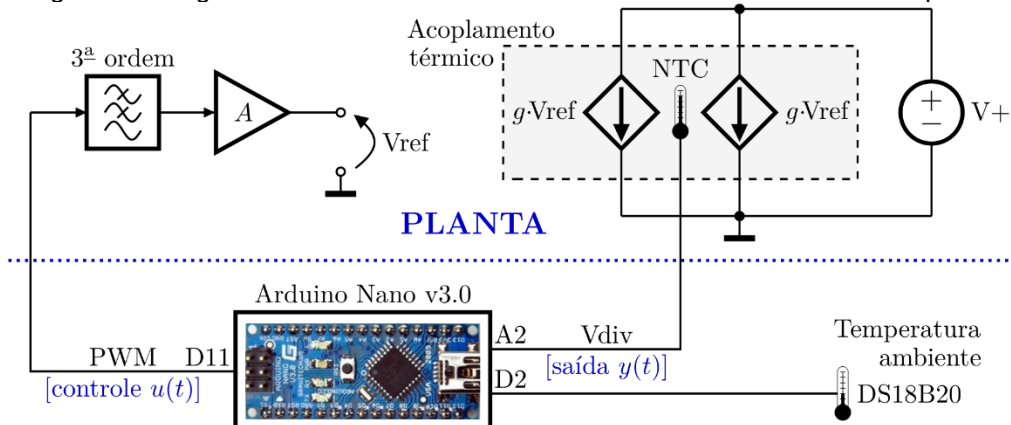
Fonte: Autoria própria, 2024.

Figura 2 – Diagrama esquemático completo do circuito eletrônico do SCT.



Fonte: Eng. Matheus do Sacramento Constantino (UFJF), 2024.

Figura 3 – Diagrama funcional do circuito eletrônico do SCT com ênfase na planta.



Fonte: Autoria própria, 2024.

3 LINGUAGEM DE PROGRAMAÇÃO C#

A IU foi inicialmente idealizada somente para o SO Windows e, para ele, a principal linguagem de programação disponível é C#, uma linguagem de programação moderna e multiparadigma (JOHNS, 2024; BOEHM; MURACH, 2021; PRICE, 2019). Já com ela em mente, o *Visual Studio* foi escolhido como ferramenta de desenvolvimento, pois é o principal ADI para C#. Ele também foi desenvolvido pela *Microsoft Corporation*. Duas linguagens de programação que poderiam ser utilizadas alternativamente são Java e *Python* (JOHNS, 2024; AWARI, 2023). Java é extremamente poderosa para o desenvolvimento de aplicações multiplataforma, porém ela não é a mais conveniente para a aplicação apresentada neste trabalho por se caracterizar por uma curva de aprendizado inicial íngreme. O mesmo pode ser dito a respeito de *Python*, excelente para aplicações de aprendizado de máquinas e análise de dados, mas relativamente complexa para o desenvolvimento de aplicações quando comparada a C#.

4 METODOLOGIAS DE DESENVOLVIMENTO

A IU foi desenvolvida, de uma certa maneira, influenciada pelos trabalhos de Barbosa; Azevedo Jr.; Normey-Rico (2023) e Guzmán *et al.* (2014). Ambos os trabalhos possuem cunho educacional, voltados ao ensino de controle automático em nível de graduação, porém baseiam-se apenas em simulações computacionais e não em sistemas reais.

Além disso, duas metodologias foram tomadas como base para o desenvolvimento da nova IU: **i)** as regras de ouro propostas por Oliveira; Oliveira (2015), principalmente em relação a consistência, *feedback* informativo, estratégias de prevenção de erros e utilização de menor carga de memória a curto prazo, e **ii)** alto desempenho visual, proposta por Goetz (2019), principalmente em relação à utilização de cores discretas e poucos objetos com cores chamativas, bem como o *design* centrado no ser humano. Os resultados da aplicação dessas metodologias vão além da estética. O intuito é que a IU ajude a prevenir falhas e possíveis erros operacionais, proporcionando maior conforto aos usuários, além de oferecer maior facilidade de aprendizado e memorização das funções.

5 INTERAÇÃO DA IU COM O SCT

Como foi mencionado na Seção 1, o intuito de desenvolver uma nova IU é operar o SCT sem depender do ADI do Arduino. Todas as funcionalidades previamente desenvolvidas deveriam ser reprogramadas na nova IU. Portanto, nessa etapa do projeto, todas as funcionalidades existentes foram devidamente testadas e documentadas, a fim de que ficasse bem claro como interagem com o SCT. Além disso, novas funcionalidades deveriam ser implementadas também, enriquecendo mais o recurso didático e melhorando a experiência dos usuários. Dessa forma, toda a documentação produzida tornou-se um importante guia para a programação da nova IU.

A nova IU:

- Comunica-se com a placa Arduino Nano através de uma porta serial universal — *Universal Serial Bus* (USB);
- Exibe em tempo real os alarmes e mensagens relativos à operação do SCT;
- Exibe em tempo real os valores de algumas das grandezas do SCT;
- Possibilita a seleção do modo de operação do SCT: Malha Aberta (MA) ou Malha Fechada (MF) (CASTRUCCI; BITTAR; SALES, 2018; DORF; BISHOP, 2018);
- Possibilita a seleção do tipo de controlador desejado, quando da operação do SCT em MF: Proporcional-Integral-Derivativo (PID) ou Liga-Desliga (LD) — *On-off* (CASTRUCCI; BITTAR; SALES, 2018; DORF; BISHOP, 2018);

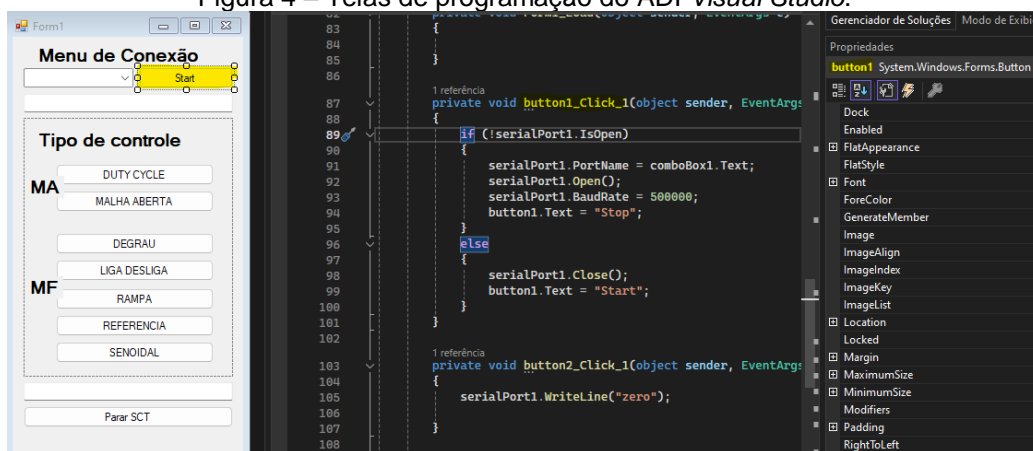
- f) Possibilita tanto a seleção do tipo de sinal de referência quanto o ajuste dos parâmetros do sinal selecionado, quando da operação do SCT em MF;
- g) Possibilita a alteração dos ganhos do controlador PID, quando da operação do SCT em MF. O controlador LD não necessita disso;
- h) Possibilita a definição direta da razão cíclica do sinal de entrada da planta $u(t)$ (ver Figuras 2 e 3), quando da operação do SCT em MA;
- i) Grava temporariamente os dados exibidos em tela e permite exportá-los na forma de arquivos com as extensões “.csv” e “.txt”, ambas bastante úteis e populares.

6 PROGRAMAÇÃO DA IU

Para o desenvolvimento do código em C# foi indispensável a aplicação do conceito de Programação Orientada a Objetos (POO), uma vez que o *Visual Studio* cria uma classe para cada uma de suas telas — *forms*. Assim, para facilitar as futuras edições, manter este conceito, juntamente com boas práticas de programação, torna-se indispensável.

Cada novo objeto adicionado à IU, aba *design* do ADI *Visual Studio*, gera uma nova função no código, na aba de programação, como mostrado na Figura 4. A figura mostra a primeira IU desenvolvida, através da qual foram realizados testes de comunicação entre a IU e a placa Arduino Nano, por meio do menu de conexão criado inicialmente. As partes do código destacadas em amarelo mostram a relação entre as telas. O botão 1 — *button1* —, quando clicado, executa o código da função *private void button1_click()*. Esse mesmo padrão foi utilizado na implementação de todos os outros botões presentes na IU.

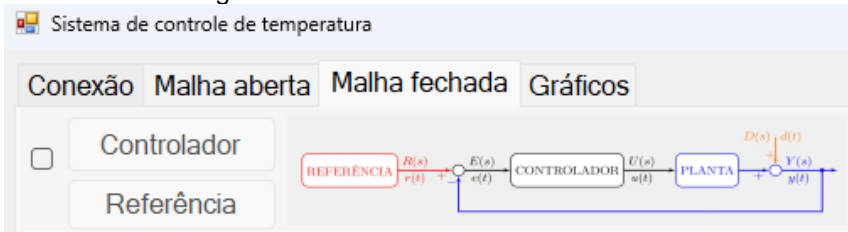
Figura 4 – Telas de programação do ADI *Visual Studio*.



Fonte: Autoria própria, 2024.

Para organizar todas as funções da IU, foi concebido um menu de abas, o qual foi posicionado no canto superior esquerdo da IU. Ele tem por finalidade a navegação entre 4 abas distintas, como mostrado na Figura 5. Na aba "Conexão", realiza-se a conexão com a placa Arduino Nano, através de uma das portas USB do computador. Todas as portas em uso são listadas automaticamente. A aba "Malha aberta" foi incluída exclusivamente para operação do SCT em MA. Ela possibilita definir a razão cíclica — *Duty Cycle* (DC) — do sinal de controle $u(t)$ diretamente (ver Figuras 2 e 3), tal que $0\% \leq DC \leq 100\%$. Através da aba "Malha fechada", opera-se o SCT em MF. Ela possibilita selecionar o tipo de controlador desejado para fechar a malha; possibilita também selecionar o tipo de sinal de referência de temperatura e ajustar alguns dos parâmetros do sinal selecionado. Por fim, na aba "Gráficos", personaliza-se os gráficos que exibem, em tempo real, os valores de algumas das grandezas, bem como o tempo decorrido, alarmes e mensagens relativos à operação do SCT.

Figura 5 – Detalhe da IU: menu de abas.



Fonte: Autoria própria, 2024.

6.1 Comunicação serial com a placa Arduino Nano

Para garantir o sincronismo e a segurança no envio de informações da IU à placa Arduino Nano, identificou-se a necessidade de desenvolver um protocolo de comunicação serial. Ele permitiu satisfazer todas as especificações apresentadas na Seção 5. A Tabela 1 apresenta o protocolo desenvolvido, que consiste em um vetor de *bytes* de tamanho fixo e igual a 9. Cada tarefa executada pelo Arduino requer uma quantidade predeterminada de informações, contidas entre o 3º e o 8º *bytes* desse vetor de *bytes*. *Bytes* de informação desnecessários à execução de uma tarefa específica são desprezados pelo Arduino. A versão inicial do código do Arduino, *firmware* original do SCT (DALTON; LOPES; FERNANDES, 2023b), precisou de ajustes para trabalhar junto com a nova IU (ver Seção 1).

Tabela 1 – Vetor de *bytes* do protocolo de comunicação serial desenvolvido.

Posição do <i>byte</i> no vetor	Significado
0	<i>Start byte</i>
1	Código da tarefa a ser executada (<i>byte</i>)
2	Valor 1 (<i>byte</i>)
3	Valor 2 (<i>byte</i>)
4	Valor 3 (<i>byte</i>)
5	Valor 4 (<i>byte</i>)
6	Valor 5 (<i>byte</i>)
7	Valor 6 (<i>byte</i>)
8	<i>End byte</i>

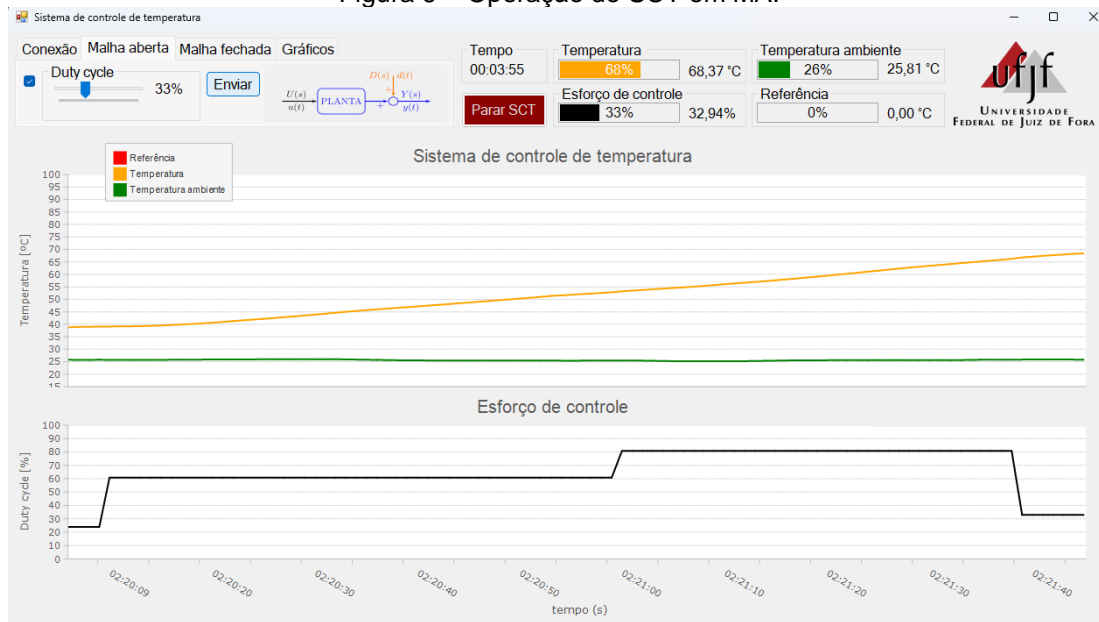
Fonte: Autoria própria, 2024.

Cada novo vetor de *bytes* que o Arduino recebe é analisado. Uma estrutura de verificação em etapas foi implementada para isso. Verifica-se inicialmente se foram recebidos todos os 9 *bytes* e, em caso afirmativo, a mensagem recebida é armazenada; caso contrário, ela é rejeitada. Verifica-se então os *bytes* de início e fim da mensagem — *start byte* e *end byte*, respectivamente — e, caso estejam corretos, a tarefa discriminada através do 2º *byte* da mensagem é identificada e devidamente processada pela placa Arduino Nano, com base nas informações necessárias, contidas do 3º ao 8º *bytes*, valores 1 a 6, respectivamente (ver Tabela 1). Algumas outras informações a respeito do *firmware* do SCT estão disponíveis em Dalton; Lopes; Fernandes (2023b).

6.2 Operação do SCT em MA

Quando o SCT opera em MA, a planta trabalha sozinha e não há realimentação do sinal de saída $y(t)$ (ver Figuras 2 e 3). Portanto, não são utilizados nem controlador nem sinal de referência, como se pode observar no diagrama de blocos exibido na imagem mostrada na Figura 6. Nessa tela, ativa-se a operação em MA por meio de uma caixa de seleção — *checkbox* —, localizada no canto superior esquerdo. Conseqüentemente, isso desabilita a operação em MF. Nessa mesma tela, através de um botão deslizante, ajusta-se a razão cíclica do sinal de entrada da planta $u(t)$. Mas o valor ajustado é efetivamente aplicado à planta somente quando se clica no botão “Enviar”.

Figura 6 – Operação do SCT em MA.



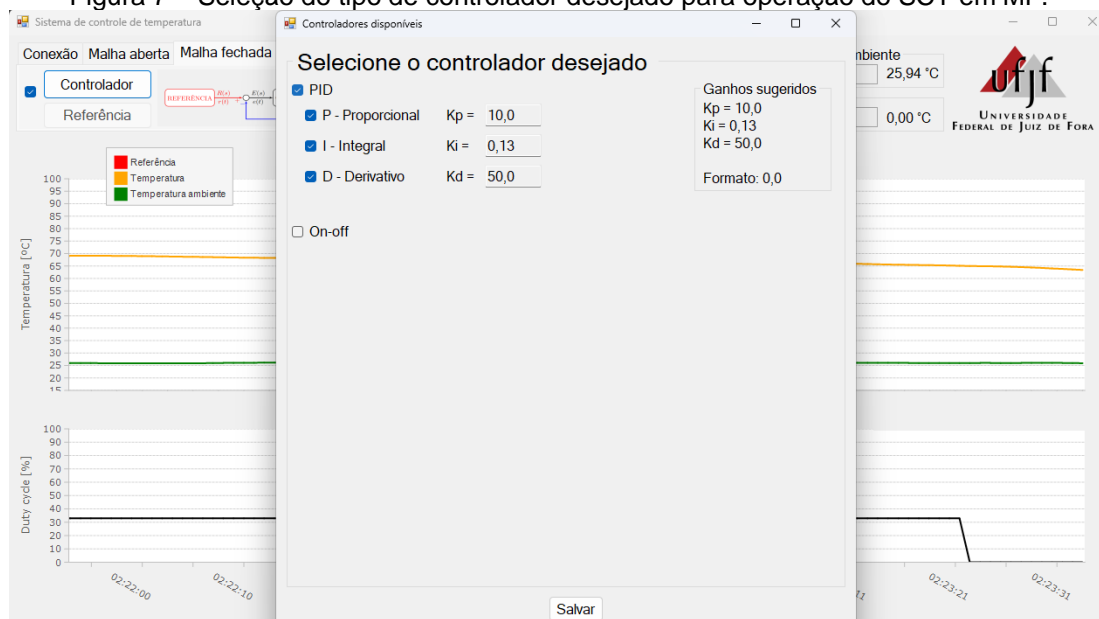
Fonte: Autoria própria, 2024.

Em MA, por diversos motivos, alcançar e manter uma temperatura desejada pode se tornar uma tarefa consideravelmente difícil de realizar (CASTRUCCI; BITTAR; SALES, 2018; DORF; BISHOP, 2018). Assim, mudanças recorrentes no valor da razão cíclica podem ser necessárias, como exemplificado na Figura 6, onde se buscava o valor de 80 °C.

6.3 Operação do SCT em MF

Quando o SCT opera em MF, a presença de um controlador e da realimentação do sinal de saída $y(t)$ levam a um cenário completamente diferente, possibilitando o rastreamento de sinais de referência (CASTRUCCI; BITTAR; SALES, 2018; DORF; BISHOP, 2018). Pode-se observar, integralmente na Figura 5 e apenas parcialmente na Figura 7, o diagrama de blocos exibido na IU quando da operação do SCT em MF, indicando o referido cenário.

Figura 7 – Seleção do tipo de controlador desejado para operação do SCT em MF.

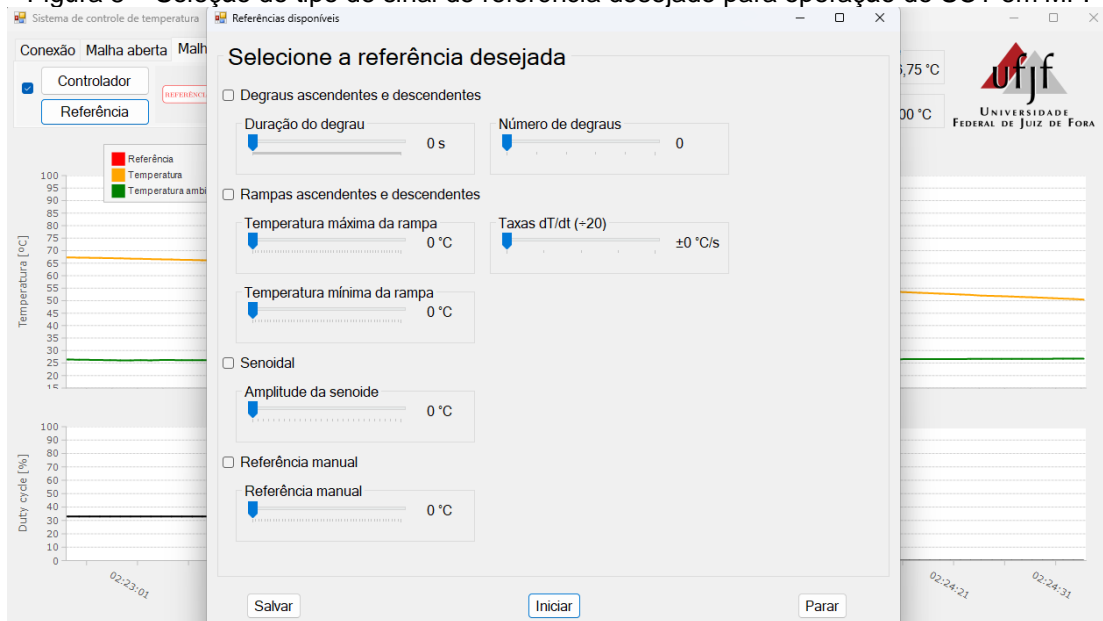


Fonte: Autoria própria, 2024.

Na aba “Malha fechada” do menu de abas (ver Figuras 5 e 7), ativa-se a operação em MF por meio da *checkbox* localizada no canto superior esquerdo. Isso desabilita a operação em MA. Através do botão “Controlador”, abre-se a janela que permite selecionar o tipo de controlador desejado. Caso seja definido o tipo PID, pode-se então ajustar os ganhos k_P , k_I e k_D digitando os valores desejados nos respectivos campos da janela. *Checkboxes* possibilitam definir variações tais como controladores dos tipos P, I, D, PI e PD. É necessário salvar as configurações primeiro, para que elas possam surtir efeito no SCT.

Após clicar no botão “Salvar”, os usuários recebem permissão para utilizar o botão “Referência”, através do qual abre-se uma janela que permite selecionar o tipo de sinal de referência desejado. Essa janela é mostrada na Figura 8. *Checkboxes* possibilitam selecionar o tipo de sinal de referência de temperatura desejado. Quando uma *checkbox* é selecionada, todas as demais são imediatamente desmarcadas e os parâmetros do sinal de referência correspondente a ela podem ser ajustados através de botões deslizantes. Todos os tipos de sinal de referência atualmente disponíveis são descritos a seguir.

Figura 8 – Seleção do tipo de sinal de referência desejado para operação do SCT em MF.



Fonte: Autoria própria, 2024.

Degraus ascendentes e descendentes

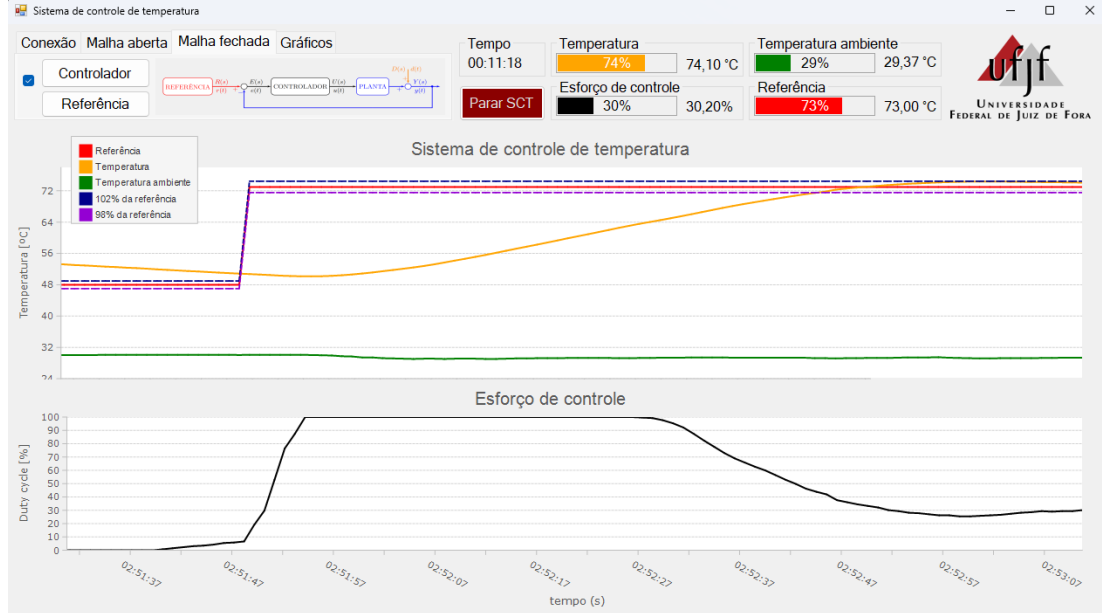
Dois valores são enviados ao Arduino: duração de cada degrau, em segundos, e número de degraus ascendentes/descendentes desejados (ver Figura 8). A amplitude de cada degrau é dada por

$$A_{deg} = (T_{m\acute{a}x} - T_{m\acute{i}n}) / (N_{deg} - 1) \quad (1)$$

onde N_{deg} representa a quantidade de degraus e $T_{m\acute{i}n} := 48 \text{ }^\circ\text{C}$ e $T_{m\acute{a}x} := 98 \text{ }^\circ\text{C}$ são duas constantes arbitradas para proporcionar uma excursão útil de $50 \text{ }^\circ\text{C}$ para operação satisfatória e segura do SCT. O valor do primeiro degrau é $T_{m\acute{i}n}$. Transcorrido o tempo de duração desejado, um novo degrau, de valor $T_{m\acute{i}n} + A_{deg}$, é iniciado junto com o reinício da contagem do tempo de duração, e assim sucessivamente, até que o último degrau ascendente, de valor $T_{m\acute{a}x}$, tem início. Transcorrido o tempo de duração, o primeiro degrau descendente, de valor $T_{m\acute{a}x} - A_{deg}$, é iniciado junto com o reinício da contagem do tempo de duração, e assim por diante, até que o último degrau descendente, de valor $T_{m\acute{i}n}$, tem iní-

cio. Isso se perpetua enquanto essa referência permanecer selecionada. A Figura 9 mostra um exemplo. A temperatura ambiente, medida pelo sensor de temperatura DS18B20, e as balizas de $\pm 2\%$ em torno do sinal de referência de temperatura também são exibidas.

Figura 9 – SCT em MF e sinal de referência do tipo degraus ascendentes e descendentes.



Fonte: Autoria própria, 2024.

Rampas ascendentes e descendentes

Três valores são enviados ao Arduino: temperaturas mínima e máxima das rampas e taxa de variação de temperatura desejadas (ver Figura 8). As rampas são simétricas; apresentam a mesma taxa de variação. O sinal de referência de temperatura é dado por

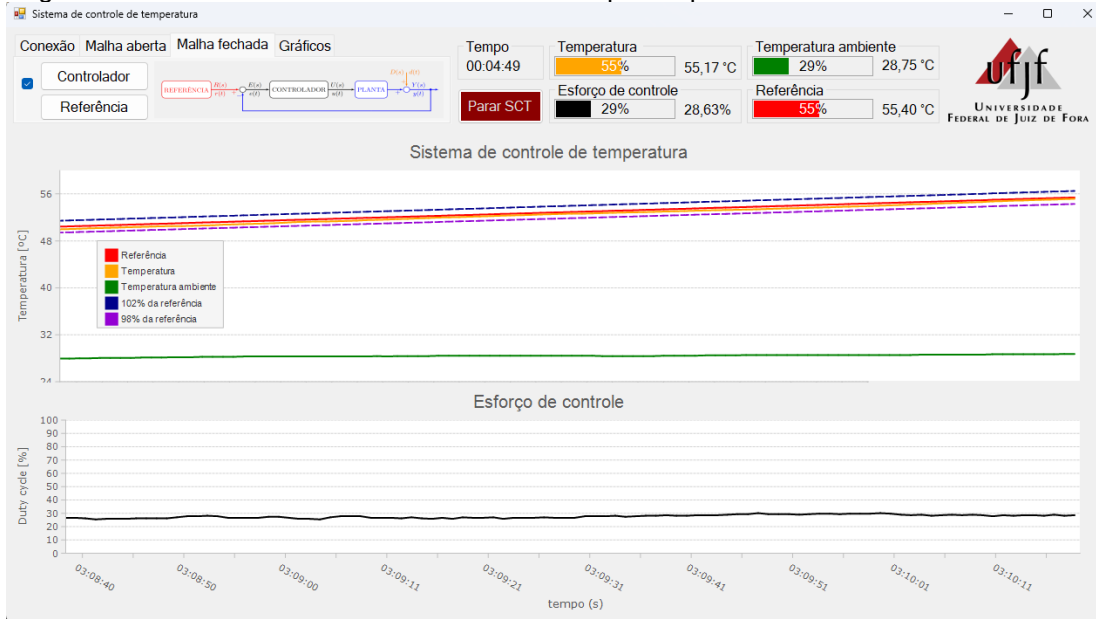
$$T_{ref}(t) = \begin{cases} T_{min}^{rampa} + (dT/dt) \cdot t' & \text{se a rampa é ascendente} \\ T_{máx}^{rampa} - (dT/dt) \cdot t' & \text{caso contrário} \end{cases} \quad (2)$$

onde T_{min}^{rampa} [°C] e $T_{máx}^{rampa}$ [°C] representam as temperaturas mínima e máxima atingidas pelas rampas, respectivamente, dT/dt [°C/s] é o valor absoluto da taxa de variação de temperatura que caracteriza as rampas e t' [s] é o tempo relativo, cuja contagem é reiniciada toda vez que uma nova rampa tem início. Um período inicial é aguardado antes da primeira rampa ascendente ter início, porque $y(t)$, isto é, a temperatura do dissipador de calor, ainda pode estar elevada quando esse sinal de referência for selecionado. Quando uma rampa ascendente atinge $T_{máx}^{rampa}$, uma rampa descendente tem início. Quando uma rampa descendente atinge T_{min}^{rampa} , uma rampa ascendente tem início. Esse movimento de alternância das rampas se perpetua enquanto esse tipo de sinal de referência permanecer selecionado. A Figura 10 mostra um exemplo. A temperatura ambiente e as balizas de $\pm 2\%$ em torno do sinal de referência também são exibidas na Figura 10.

Senoidal

Um único valor é enviado ao Arduino: amplitude desejada (ver Figura 8). Optou-se por trabalhar somente com a amplitude porque a resposta da planta é relativamente lenta, de modo que a resposta de MF é satisfatória somente para sinais de referência de baixas frequências. O sinal de referência de temperatura senoidal é dado por

Figura 10 – SCT em MF e sinal de referência do tipo rampas ascendentes e descendentes.



Fonte: Autoria própria, 2024.

$$T_{ref}(t) = T_{méd} + A_{sen} \text{sen}(\omega \cdot t) \quad (3)$$

onde $T_{méd} := 70 \text{ °C}$ é uma constante arbitrada para proporcionar uma boa excursão útil para operação satisfatória e segura do SCT, $A_{sen} [\text{°C}]$ é amplitude desejada, $\omega := \pi/300 \text{ rad/s}$ é uma constante arbitrada em conformidade com o exposto no parágrafo anterior a respeito da velocidade de resposta da planta e $t [\text{s}]$ é o tempo.

Manual

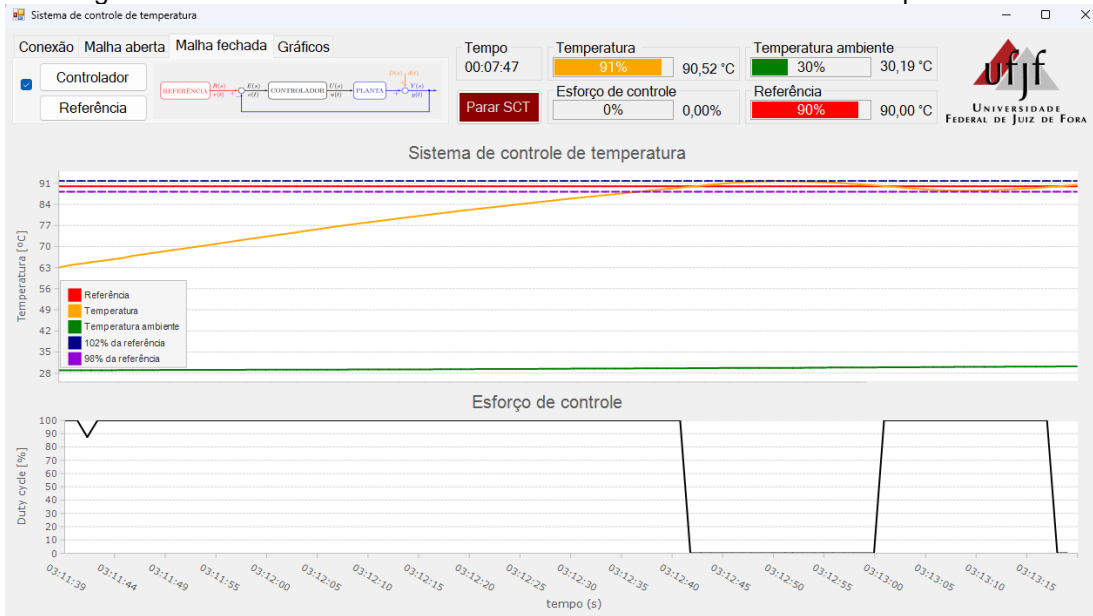
Um único valor é enviado ao Arduino: temperatura desejada (ver Figura 8). O sinal de referência corresponde a um único degrau, de duração indefinida e valor igual à temperatura desejada. Pode-se redefinir o valor desejado a qualquer momento. A Figura 11 mostra um exemplo. Nele, ao contrário dos exemplos mostrados nas Figuras 9 e 10, o controlador do tipo LD foi selecionado e, mais uma vez, a temperatura ambiente e as balizas de $\pm 2\%$ em torno do sinal de referência de temperatura também são exibidas.

6.4 Recursos gráficos

Diferentes recursos gráficos são utilizados concomitantemente para facilitar a percepção e a compreensão dos valores das grandezas exibidos através da IU. São eles: **i)** gráficos atualizados em tempo real, **ii)** barras de progresso percentual, acrescidas dos valores numéricos correspondentes com duas casas decimais e as unidades de medida, também atualizadas em tempo real e **iii)** indicador de tempo transcorrido de operação. A inclusão desse indicador foi essencial à leitura confortável dos gráficos, porque, neles, o tempo é assinalado pelas estampas de tempo correspondentes aos momentos em que os dados são recebidos pela IU, o que torna a percepção do tempo decorrido insatisfatória na falta do referido indicador direto. Observar atentamente as Figuras 6 a 12.

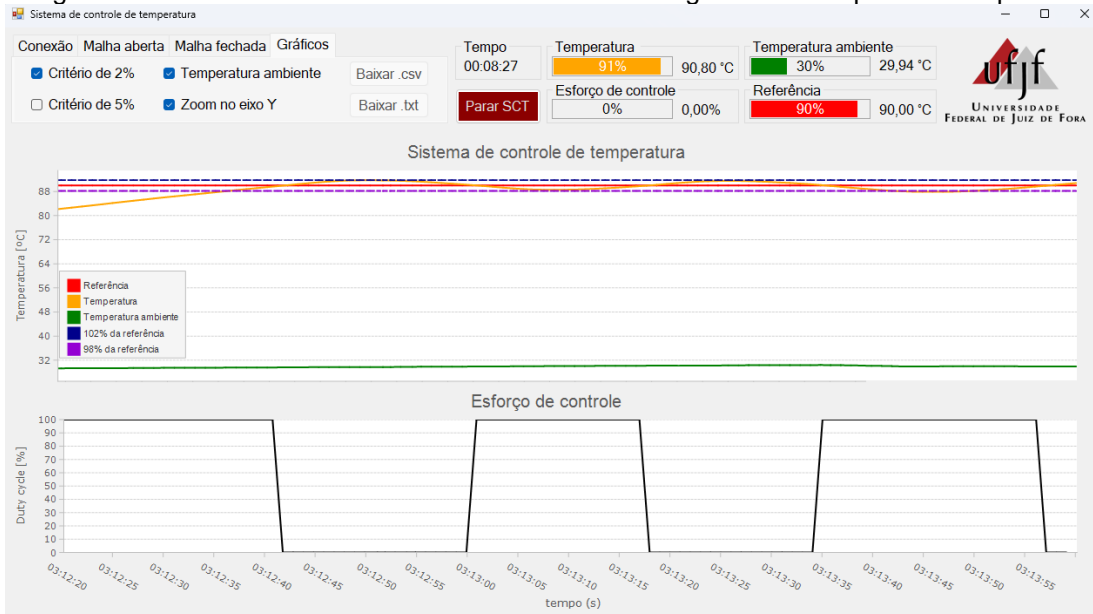
Na aba “Gráficos” do menu de abas (ver Figura 5), *checkboxes* permitem optar por exibir, ou não, **i)** a temperatura ambiente, medida pelo sensor de temperatura DS18B20 (ver Figuras 2 e 3), **ii)** as balizas de $\pm 2\%$ e $\pm 5\%$ em torno do sinal de referência de temperatura e **iii)** a ampliação — *zoom in* — somente do eixo Y do gráfico de temperatura. A Figura 12 mostra um exemplo, onde o SCT operava em MF com o controlador do tipo LD.

Figura 11 – SCT em MF com controlador LD e sinal de referência do tipo manual.



Fonte: Autoria própria, 2024.

Figura 12 – Aba “Gráficos” da IU e escala do eixo Y do gráfico de temperatura ampliada.



Fonte: Autoria própria, 2024.

7 CONCLUSÕES

Este artigo apresentou uma IU especialmente desenvolvida para simplificar o uso do SCT proposto em Dalton; Lopes; Fernandes (2023b). Os aspectos mais relevantes do desenvolvimento foram discutidos. A IU é eficiente, robusta e de fácil aprendizado. Ela trouxe uma série de melhorias significativas, tanto em termos de funcionalidade quanto de usabilidade, oferecendo uma experiência mais intuitiva, segura e confortável aos usuários.

A nova IU **i)** exibe valores, alarmes e mensagens de operação em tempo real, **ii)** possibilita selecionar entre os modos de operação do SCT em MA e MF, **iii)** possibilita selecionar entre os controladores PID e LD, quando da operação do SCT em MF, **iv)** possibilita selecionar o tipo do sinal de referência de temperatura, quando da operação do SCT

em MF, **v**) possibilita alterar os ganhos do PID e ajustar os parâmetros dos sinais de referência, quando da operação do SCT em MF, bem como possibilita definir diretamente a **ra**ção cíclica do sinal de entrada da planta, quando da operação do SCT em MA, e **vi**) grava e permite exportar os dados gerados durante a operação do SCT. A visualização gráfica foi aprimorada, com gráficos no tempo e mais formas de visualização dos dados em tempo real, tornando o acompanhamento e a análise dos dados mais intuitiva e compreensível. A utilização de técnicas de POO e de boas práticas de programação assegurou que o código ficasse organizado, fácil de corrigir e atualizar, facilitando futuras atualizações.

Espera-se que este artigo seja motivador de novos estudos na área de controle automático e que ele sirva ainda para aguçar a curiosidade dos discentes.

Trabalhos futuros devem explorar outras técnicas de controle automático, por exemplo, pré-alimentação de referência — *reference feedforward* —, controle nebuloso — *fuzzy control* —, controle baseado em aprendizagem por reforço — *reinforcement learning* —, entre outras, bem como a geração de novos tipos úteis de sinais de referência de temperatura, que exigirão, ambas, modificações na IU como ajustes no *firmware* do SCT. Outra importante iniciativa futura será tornar a IU multiplataforma, pois isso possibilitará seu funcionamento também em máquinas que executam os SOs Linux e macOS.

AGRADECIMENTOS

Os autores agradecem ao Eng. Matheus do Sacramento Constantino (UFJF) pela colaboração no projeto. Ele elaborou o diagrama esquemático do circuito eletrônico do SCT (ver Figura 2), roteou a PCI (ver Figura 1) e contribuiu decisivamente para dirimir algumas das dúvidas que surgiram durante o processo de desenvolvimento da IU.

REFERÊNCIAS

AWARI. **Python, Java e Javascript: qual a melhor linguagem de programação para você?** Awari, 2023. Disponível em: <https://awari.com.br/python-java-e-javascript-qual-a-melhor-linguagem-de-programacao-para-voce/>. Acesso em: 06/06/2024.

BARBOSA, Gabriel; AZEVEDO JR., Arnaldo P.; NORMEY-RICO, Julio E. Ferramentas iterativas de código aberto para estudo de controladores clássicos. In: XVI Simpósio Brasileiro de Automação Inteligente (SBAI), 2023, Manaus. **Anais**. Manaus.

BOEHM, Anne; MURACH, Joel. **Murach's C#**. 7ª ed. Fresno, EUA: Mike Murach and Associates, Inc., 2021.

CASTRUCCI, Plínio B. de L.; BITTAR, Anselmo; SALES, Roberto M. **Controle automático**. 2ª ed. Rio de Janeiro: LTC — Livros Técnicos e Científicos Editora Ltda., 2018.

DALTON, André de A.; LOPES, Bartolomeu H.; FERNANDES, Daniel de A. Aplicação de um sistema de controle de temperatura de baixo custo no ensino prático de engenharia elétrica. In: 51º Congresso Brasileiro de Educação em Engenharia (COBENGE), 2023a, Rio de Janeiro. **Anais**. Rio de Janeiro.

DALTON, André de A.; LOPES, Bartolomeu H.; FERNANDES, Daniel de A. Sistema de controle de temperatura educacional compacto, portátil e de baixo custo. In: XVI Simpósio Brasileiro de Automação Inteligente (SBAI), 2023b, Manaus. **Anais**. Manaus.

DORF, Richard C.; BISHOP, Robert H. **Sistemas de controle modernos**. 13ª ed. Rio de Janeiro: LTC — Livros Técnicos e Científicos Editora Ltda., 2018.

GOETZ, Helcker F. **Metodologia para desenvolvimento de IHMs de alta performance visual**. Elipse Knowledgebase, 2019. Disponível em: <https://kb.elipse.com.br/metodologia-para-desenvolvimento-de-ihms-de-alta-performance-visual/>. Acesso em: 06/06/2024.

GUZMÁN, J. L.; HÄGGLUND, T.; ÅSTRÖM, K. J.; DORMIDO, S.; BERENGUEL, M.; PIGUET, Y. Understanding PID design through interactive tools. In: 19th World Congress (WC) of the International Federation of Automatic Control (IFAC), 2014, Cidade do Cabo, África do Sul. **Anais**. Cidade do Cabo, v. 47, n. 3, pp. 12243–12248.

JOHNS, Robert. **The 20 best programming languages to learn in 2024**. Hackr.io, 2024. Disponível em: <https://hackr.io/blog/best-programming-languages-to-learn>. Acesso em: 06/06/2024.

LOPES, Bartolomeu H.; DALTON, André de A.; FERNANDES, Daniel de A. A utilização de um sistema de controle de temperatura de fácil construção para ensino prático nas disciplinas de engenharia elétrica. In: 51º Congresso Brasileiro de Educação em Engenharia (COBENGE), 2023, Rio de Janeiro. **Anais**. Rio de Janeiro.

OLIVEIRA, Francisco Carlos de M. B.; OLIVEIRA, Fernando Antônio de M. B. **Interação humano computador**. 2ª ed. Fortaleza: Editora da Univ. Estadual do Ceará (EdUECE), 2015.

PRICE, Mark J. **C# 8.0 and .NET Core 3.0: modern cross-platform development**. 4ª ed. Birmingham, Reino Unido: Packt Publishing Ltd., 2019.

DEVELOPMENT OF A USER INTERFACE FOR A COMPACT, PORTABLE AND LOW-COST EDUCATIONAL TEMPERATURE CONTROL SYSTEM

Abstract: *This paper presents a User Interface (UI) specially developed to simplify the use of an educational, compact, portable and low-cost Temperature Control System (TCS). The TCS was previously proposed as a didactic resource to assist in the teaching/learning process of several different subjects in Electrical Engineering, mainly those related to the area of automatic control of dynamic systems, but not only them. It helps to carry out practical demonstrations and experiments that bring theory and practice together during theoretical classes. Various theories and techniques can be exemplified, compared and tested in practice. The UI replaces, with advantages, a UI initially developed based on the Integrated Development Environment (IDE) of Arduino. The new UI was written in C#, through the Visual Studio IDE, by applying the object-oriented programming concept. For now, it only works on the Windows operating system. The new UI communicates with the TCS through a specially developed serial communication protocol and allows real-time visualization of alarms, messages and values of some of the quantities related to the TCS operation. In addition to that, it also presents them in a clearer and more efficient way.*

Keywords: *C#, Feedback control system, Object-oriented programming, Temperature control, User interface, Visual Studio*

