



APLICAÇÕES DIDÁTICAS EM CONTROLE E AUTOMAÇÃO UTILIZANDO SOFTWARES DE SIMULAÇÃO E SISTEMAS DE CONTROLE DIGITAL EMBARCADOS

Adriel Luiz Marques ^{*1}, Renato Santos Carrijo² e Renato Ferreira Fernandes Júnior³

¹FEELT - Universidade Federal de Uberlândia

²FEELT - Universidade Federal de Uberlândia

³FEELT - Universidade Federal de Uberlândia

Resumo - Este artigo tem como objetivo analisar o uso de plataformas de hardware aberto, como o Arduino, para aplicação em softwares de simulação utilizados no laboratório de controle e automação da Universidade Federal de Uberlândia. Diante disso, será avaliado o atual modelo empregado nesse laboratório, que utiliza o software de simulação SCILAB integrado com a plataforma Arduino Uno, observando as suas vantagens e desvantagens. Além disso, será examinada a proposta a utilização de placas ARM com CPU Cortex-M3 (STM32F103C8T6) para essa tarefa, levantando os benefícios de sua utilização e como adequar o atual modelo para o uso dessas placas. Por fim, são apresentadas ideias para futuros trabalhos, uma vez que este artigo é parte inicial de uma pesquisa de iniciação científica sobre aplicações didáticas em controle e automação.

Palavras-Chave- Arduino, controle digital, microcontroladores, SCILAB, softwares de simulação, STM32.

OPEN HARDWARE PLATFORM ANALYSIS FOR TEACHING APPLICATIONS IN SIMULATION SOFTWARE AND DIGITAL CONTROL SYSTEMS

Abstract - This paper aims to analyze the use of open hardware platforms, such as Arduino, for application in simulation software used in the control and automation laboratory of the Universidade Federal de Uberlândia. Therefore, the current model used in this laboratory, which uses SCILAB simulation software integrated with the Arduino Uno platform, will be evaluated, observing its advantages and disadvantages. In addition, the proposal to use Cortex-M3 CPU ARM boards (STM32F103C8T6) for this task will be examined, showing the benefits of their use and how to adapt the current model to be used on these boards. Finally, ideas for future work are presented, since this article is an initial part of a scientific research on teaching applications in control and automation.

Keywords - Arduino, digital control, microcontrollers,

*adrielluizmarques@hotmail.com

SCILAB, simulation software, STM32.

I. INTRODUÇÃO

O controle automático é fundamental no avanço da engenharia e da ciência, tornando-se indispensável nos processos industriais e de manufatura modernos. Os avanços nessa área de conhecimento propiciam meios para a obtenção do desempenho ótimo de sistemas dinâmicos, melhorias de produtividade e alívio em trabalhos manuais desgastantes [1].

O controle de sistemas físicos com o uso de um computador digital ou de um microcontrolador está se tornando cada vez mais comum. Entre as suas vantagens estão a flexibilidade dos programas de controle, que podem ser readequados facilmente de acordo com as exigências do projeto, e a reutilização do projeto de hardware com variações de softwares, o que reduz o tempo de projeto [2].

Diante desse cenário, é importante a utilização de softwares de simulação que facilitem a execução de técnicas de controle digital, especialmente no ambiente acadêmico. Tais ferramentas permitem que o aluno pratique a teoria aprendida de forma segura e controlada [3]. Associado ao software de simulação, um ambiente de aquisição de dados suficientemente adequado para analisar as características da planta a ser controlada é necessário para os estudos de técnicas de controle [4].

Atualmente, o laboratório de controle e automação utiliza o software SCILAB em conjunto com a plataforma Arduino como software de simulação e controle dos sistemas. Porém, esta plataforma apresenta problemas, como uma baixa precisão para a caracterização de sistemas com dinâmicas rápidas e uma dificuldade para acionar muitas cargas simultaneamente.

Desse modo, este trabalho visa analisar a utilização de hardwares de software aberto para aplicações didáticas em controle empregando softwares de simulação e controle avançados. A análise será realizada examinando o desempenho do software SCILAB utilizando diferentes plataformas de hardware, como a plataforma Arduino Uno e a STM32F103C876, em diferentes cenários de controle utilizados nas disciplinas, observando as suas vantagens e desvantagens. Além disso,

são citadas novas propostas para a continuação deste trabalho, como a modularização do código do firmware visando torná-lo portátil para diferentes plataformas de hardware e seja mais fácil a sua migração para uma nova plataforma.

Para isso, este trabalho está dividido da seguinte forma: No capítulo I é realizado uma breve introdução sobre o assunto. No capítulo II é descrito o sistema atual do laboratório. No capítulo III é detalhada toda a proposta deste trabalho. No capítulo IV são mostrados os resultados obtidos até o momento e, por fim, no capítulo V são realizadas as conclusões.

II. SISTEMA ATUAL DO LABORATÓRIO

Atualmente, o laboratório de controle e automação da Universidade Federal de Uberlândia utiliza o SCILAB como software de simulação e a plataforma Arduino Uno para a aquisição e controle de sinais externos, conforme a Figura 1.

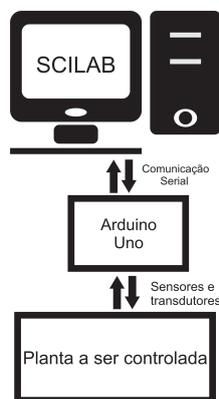
O SCILAB é um programa científico, de plataforma aberta, aplicado à computação numérica, desenvolvido para aplicações em controle de sistemas e processamento de sinais. Ele foi desenvolvido por pesquisadores do INRIA (Institut National de Recherche en Informatique et en Automatique) e do ENPC (École Nationale des Ponts et Chaussées [5]).

O software MATLAB é uma opção comumente utilizada no controle de processos. Contudo, apesar de possuir diversas ferramentas que facilitam aplicações de controle, ele é de domínio privado, o que gera custos com licenças e atualizações, tornando inviável a sua utilização na grande maioria das instituições de ensino e pesquisa [16]. Diante disso, o SCILAB apresenta-se como uma solução que atende às demandas técnicas e econômicas para os laboratórios de pesquisa.

O Arduino Uno é um sistema embarcado baseado no microcontrolador ATmega328P, de software livre (refere-se a qualquer programa de computador que pode ser usado, copiado, estudado, modificado e redistribuído sem restrições [3]), desenvolvida pelo Ivrea Interaction Design Institute, para aplicações educacionais [6].

A associação do SCILAB com o Arduino Uno é possível por meio do desenvolvimento de bibliotecas, conhecidas como “toolboxes”, que permitem a criação de diagramas de blocos no SCILAB que se comunicam, via serial, com o firmware do Arduino.

Figura 1: Estrutura de funcionamento do Arduino com o SCILAB.

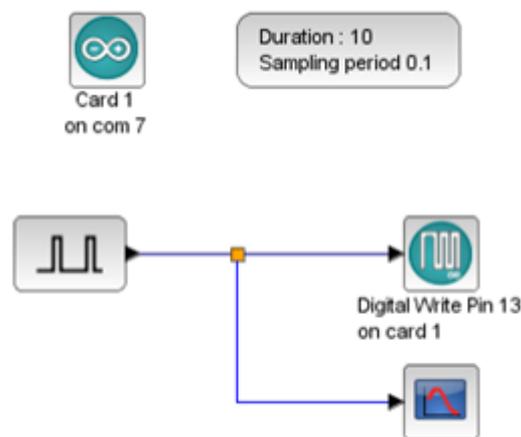


Na Figura 1, é possível observar com mais detalhes esse modelo utilizado no laboratório de controle e automação. O

Arduino tem a função de obter os dados da planta de controle por meio de sensores e enviá-los via serial para o SCILAB, o qual tem o papel de processar esses dados e enviar a resposta de controle para o Arduino via serial. Por fim, o Arduino utiliza-se de atuadores para promover a ação e controle sobre a planta.

O SCILAB possui uma ferramenta de simulação de sistemas dinâmicos contínuos e discretos, conhecida como XCOS. Com essa ferramenta é possível utilizar diagramas de blocos para programar técnicas de controle, sem a necessidade de alguma programação textual. Atuando em conjunto com o XCOS, o ATOMS (AuTomatic mOdules Management for Scilab) elaborou um módulo específico para a comunicação com o Arduino [8]. Esse módulo, ou “toolbox”, possibilita que placas do Arduino Uno possam interagir com o ambiente XCOS e controlar hardwares externos por meio da manipulação de blocos, permitindo leituras digitais e analógicas, acionamento de saídas com PWM, entre outras funções.

Figura 2: Exemplo de uma aplicação com o Arduino no XCOS [8].



Na Figura 2, observa-se um exemplo de uma aplicação com Arduino no XCOS. O bloco "Card 1 on com 7" refere-se à porta serial onde a placa do Arduino Uno está conectada. À direita desse bloco, encontra-se o bloco para a definição da duração da simulação e do período de amostragem. No canto inferior esquerdo está o bloco para a definição da forma de onda para o acionamento da saída PWM. O bloco “Digital Write Pin 13 on card 1” define qual a porta do Arduino a ser acionada. Por fim, o bloco no canto inferior direito fornece uma observação da forma de onda que está sendo utilizada no processo.

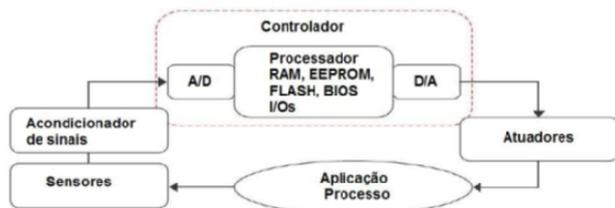
III. METODOLOGIA DE DESENVOLVIMENTO E ANÁLISE

Este trabalho visa analisar a utilização de hardwares de software aberto para aplicações didáticas em controle utilizando-se softwares de simulação. Para isso, será comparado o atual modelo utilizado no laboratório de controle e automação com outras opções de hardwares disponíveis no mercado, especialmente placas com o microcontrolador STM32F103C8T6.

Desse modo, para analisar a eficácia do atual modelo didático de se utilizar o SCILAB em conjunto com plataformas

Arduino, é necessário analisar as exigências requeridas para essa aplicação. Além disso, é necessário observar se é possível encontrar no mercado soluções que cumpram tais requerimentos como: Taxa Amostragem, Conversão Digital e Analógica, Viabilidade Econômica. Estes requerimentos podem ser observados na Figura 3 e serão analisados no decorrer desse trabalho.

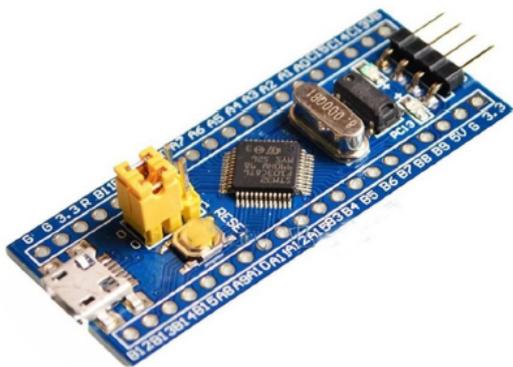
Figura 3: Funcionamento básico de um controlador digital [4].



Na figura 3, observa-se que o controlador digital atua processando dados procedentes do condicionador de sinais, o qual é responsável por realizar o tratamento de dados proveniente dos sensores. Logo, a qualidade do conversor analógico-digital (A/D) e a taxa de amostragem são fatores importantes para uma boa leitura de dados externos, além das características internas do microcontrolador, como as memórias RAM e a FLASH. Além disso, o controlador digital necessita de uma interface que o permita atuar sobre o ambiente exterior. Portanto, é necessário que ele possua um conversor digital-analógico (D/A) de modo a alimentar os atuadores e controlar o processo. Ademais, é necessário considerar os custos de cada equipamento, uma vez que esse fator pode inviabilizar algumas soluções.

Outro fator a ser considerado em um dispositivo que seja empregado nessa tarefa é a facilidade de seu manuseio, o que envolve a adaptação do firmware original da ATOM para que seja possível programa-lo por meio do SCILAB. Como a IDE (Ambiente de Desenvolvimento Integrado) do Arduino possui uma adaptação para programar microcontroladores da SMT32 [9], as adaptações necessárias para utilizar placas com o STM32F103C8T6 são mínimas. Diante disso, esse foi o hardware utilizado para essa análise, o qual pode ser visto na Figura 4 [14].

Figura 4: Placa STM103F103C8T6 - ARM Cortex-M3[14].



IV. RESULTADOS E DISCUSSÃO

De acordo com os quesitos apresentados anteriormente, é possível organizar os dados referentes a análise de cada sistema embarcado de acordo com a tabela a seguir:

Tabela 1: Comparação placas Arduino e STM32 [4],[10] e [11]

Característica	Arduino	STM32
A/D	10-bit, 8 canais	12-bit, 10 canais
Resolução[%]	0,10	0,02
Taxa de amostragem [KS/s]	1000	7000
SRAM	2 KB	20 KB
PWM	6	15
Preço médio	\$2,80	\$1,80

A. Amostragem

De acordo com a Tabela 1, a placa STM32 possui 10 canais A/D de 12 bits. Enquanto que a placa Arduino possui um A/D com 8 canais de 10 bits. Portanto, observa-se que o módulo da STM32 possui maior quantidade de pinos para essa aplicação, além de apresentar uma resolução 5 vezes maior, o que representa a possibilidade de obter dados de uma maior quantidade de periféricos e, principalmente, representar os sinais com maior precisão, o que melhora a qualidade do controle digital.

Além disso, a placa STM32 possui uma taxa de amostragem 7 vezes maior que o Arduino. Essa característica, associada a uma SRAM com 10 vezes mais capacidade, permite que o processamento da STM32 seja, geralmente, mais rápido que o do Arduino, o que é importante para determinados projetos de controle.

B. Conversão digital-analógica

Conforme a Tabela 1, observa-se que a placa STM32 possui 9 canais PWM a mais em relação ao Arduino, o que a torna mais versátil para as diversas aplicações em que é necessário acionar cargas com uma tensão variada, o que é comum em projetos de controle.

C. Viabilidade econômica

De acordo com [3], a análise da viabilidade econômica da utilização de softwares na área universitária é importante, pois pode impossibilitar certas propostas. No uso de hardwares não é diferente, uma boa proposta deve ser, primordialmente, viável economicamente.

Conforme Tabela 1, observa-se que a utilização de placas STM32 promoveria uma economia de 36% em relação ao uso de placas de Arduino. Sendo possível, desta forma, investir na aquisição de um maior número de componentes ou em outras necessidades do laboratório.

D. Ajustes para utilização da STM32 com a biblioteca Arduino do XCOS

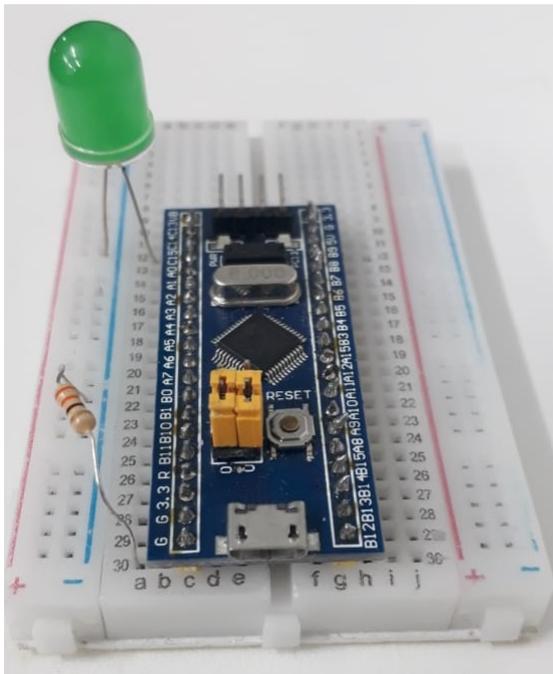
De modo a evitar a necessidade de se desenvolver uma biblioteca específica para se programar um microcontrolador genérico com o XCOS, é possível utilizar o módulo desenvolvido para o Arduino, com algumas pequenas adequações.

Uma vez que já existe uma biblioteca para se programar placas da STM32 com a IDE original do Arduino [9], só é necessário observar a numeração dos pinos realizada por essa biblioteca. À vista disso, observa-se que o pino “0” da biblioteca [9] refere-se ao pino “PA0” da STM32, seguindo essa numeração até o pino “34” que refere-se ao pino “PC15”. Além disso, é possível alterar a biblioteca fornecida pela ATOMS, de modo a apresentar a relação de pinos do microcontrolador da STM32 para o usuário do XCOS, facilitando o trabalho nesse software.

E. Testes da integração da STM32 com o XCOS

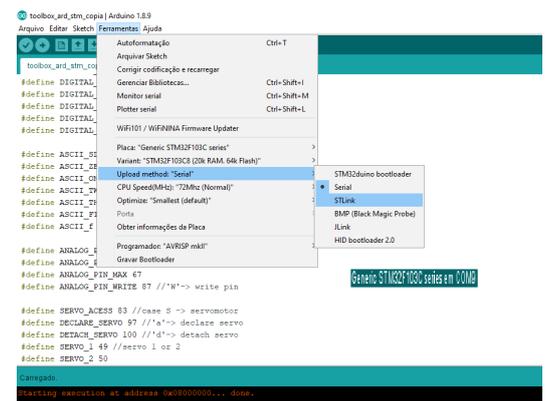
Para avaliar o uso do XCOS com microcontroladores STM32F103C8T6, será utilizado o tutorial “Scilab Arduino Blink LED Tutorial” [17], adaptando-o para o uso desse novo microcontrolador. A ideia é fazer com que um LED pisque por um determinado tempo, de acordo com uma forma de onda definida nos blocos do XCOS. A montagem desse circuito pode ser vista na Figura 5. O LED (diodo emissor de luz) tem o seu anodo ligado ao pino PC13 da STM32 e seu catodo ligado ao resistor, o qual está ligado ao terra.

Figura 5: Circuito para controle de LED.



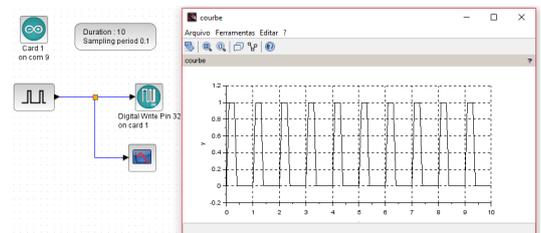
A primeira tarefa necessária é transferir o firmware desenvolvido pela ATOM para o microcontrolador da STM32. Como a comunicação com o software SCILAB é via serial, será utilizado o conversor HW- 597 USB TTL para a transferência do firmware, uma vez que ele será utilizado em etapas futuras, sendo necessário colocar o microcontrolador em modo bootloader (mover o primeiro conector para a posição 1 e pressionar o botão reset da placa) e escolher o método serial para o upload na IDE. A Figura 6 mostra a realização dessa etapa utilizando-se a IDE do Arduino com a biblioteca para STM32 [9].

Figura 6: Transferência do firmware para o microcontrolador.



Diante disso, o XCOS já poderá se comunicar com a placa da STM32. Desse modo, basta selecionar os blocos necessários para a construção do circuito, de acordo com a Figura 2, definir a porta COM utilizada e determinar o a saída a ser acionada. O resultado pode ser visto na Figura 7.

Figura 7: Controlando um LED com o SCILAB e a STM32.



De acordo com a Figura 7, verifica-se o funcionamento desse modelo, sendo possível verificar pelo gráfico os momentos em que o LED acendeu e apagou. Portanto, conclui-se que esse método cumpriu a sua proposta e pode ser utilizado para aplicações semelhantes.

V. CONCLUSÕES

Este trabalho faz parte de um projeto de iniciação científica desenvolvido no Laboratório de Automação Sistemas Eletrônicos e Controle (LASEC) da Universidade Federal de Uberlândia que visa trabalhar no desenvolvimento de drivers de integração de placas eletrônicas didáticas aplicados a plataformas open source (com código aberto) de simulação de sistemas de controle. Por meio desse artigo foi possível analisar o uso de plataformas de hardware aberto aplicadas a softwares de simulação, especialmente a integração da plataforma SCILAB com o Arduino Uno, levantando as suas vantagens e desvantagens.

Primeiramente, observa-se que a plataforma Arduino Uno é útil para o ensino de técnicas de controle por meio de softwares de simulação, uma vez que possui uma vasta documentação e bibliotecas prontas para uso, como é o caso da biblioteca Arduino XCOS, além de possuir uma programação aberta para alterações. Desse modo, ela é uma ferramenta importante no desenvolvimento inicial de laboratórios de controle. Contudo, o Atmega328P, do Arduino Uno, apresenta limitações se comparado a outros microcontroladores existentes no mercado. Nesse sentido, placas com o microcontrolador STM32F103C8T6 recebem destaque, pois apresentam melhor

desempenho técnico, são viáveis economicamente e possuem ferramentas que facilitam a sua implantação em laboratórios didáticos.

Todavia, é oportuno ressaltar que, apesar das importantes conclusões apresentadas nesse artigo, elas fazem parte de resultados iniciais da pesquisa de iniciação científica. Desse modo, melhores contribuições são esperadas na continuação desse projeto. Para trabalhos futuros, propõe-se o desenvolvimento de um firmware com maior otimização para placas da STM32, utilizando as plataformas de desenvolvimento da STMicroelectronics como o STM32CubeMX [12]. Posteriormente, é indicada a alteração a metodologia de programação do firmware, utilizando um modelo multitarefa, no qual um RTOS (Sistema Operacional de Tempo Real ou Real-Time Operating System, em inglês) possa ser usado para o gerenciamento de tarefas, dividindo o tempo do processador em intervalos de tempo de acordo com os processos requeridos [13]. Desse modo, o microcontrolador pode gerenciar com maior eficácia a sua comunicação com o software de simulação, continuando a sua aquisição e processamento de dados externos.

Além disso, em trabalhos futuros, sugere-se a elaboração de um firmware portátil para diferentes plataformas, tanto em relação ao software de simulação quanto em relação ao hardware de aquisição de sinais. Tal aplicação será útil para diversas aplicações didáticas, em diferentes disciplinas de controle.

Por fim, tendo em vista a aplicação didática desse projeto, incentiva-se o desenvolvimento das SMO's modificadas (Séries Metódicas de Oficina) de acordo com os hardwares e softwares utilizados. Essa metodologia, desenvolvida na década de 1940 e aplicada no SENAI, consiste na elaboração de folhas de instruções que expliquem de forma clara e objetiva o trabalho prático que será realizado [15]. Tal recurso, que já trouxe bons resultados em algumas disciplinas, pode facilitar o aprendizado do aluno e orientar o papel dos professores nessas atividades.

AGRADECIMENTOS

Os autores agradecem ao Laboratório de Automação Sistemas Eletrônicos e Controle (LASEC) da Universidade Federal de Uberlândia, que proporcionou a estrutura física e tecnológica necessária para a pesquisa de iniciação científica que motivou a elaboração deste trabalho.

REFERÊNCIAS

- [1] OGATA, K. *Engenharia de Controle Moderno*. 3a Edição. Rio de Janeiro, 2000.
- [2] FRANKLIN, G. F.; POWEL J. D.; WORKMAN M. L. *Digital Control of Dynamic Systems*. 3a Edição.
- [3] AGOSTINHO N. U., et al. *Softwares livres no ensino de engenharia: uma atitude socialmente justa, economicamente viável e tecnologicamente sustentável*. Anais do XXXIV COBENGE. Passo Fundo: Ed. Universidade de Passo Fundo, Setembro de 2006.
- [4] SILVA, W. S., FERNANDES R. F. *Análise de sistema de aquisição de dados para controle de processos*. XIV CEEL, Uberlândia, 2016.
- [5] MAROMISATO, G. D. Y., et al. *A utilização de um "software" livre no ensino de sistemas de controle*. In: International Conference on Engineering and Computer Education. São Paulo.
- [6] *What is Arduino?*. Disponível em: <https://www.arduino.cc/en/Guide/Introduction#>. Acesso em: 07 ago. 2019.
- [7] *STM32 - STM32F103x8, STM32F103xB User Guide*. Disponível em: <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>. Acesso em: 7 de agosto de 2019.
- [8] *ATOMS : Arduino details*. Disponível em: <https://atoms.scilab.org/toolboxes/arduino/1.6.2>. Acesso em: 9 de agosto de 2019.
- [9] CLARK, R. *Arduino_STM32*. Disponível em: https://github.com/rogerclarkmelbourne/Arduino_STM32. Acesso em: 9 de agosto de 2019.
- [10] *STM32 com IDE Arduino: Primeiros Passos*. Disponível em: <https://www.filipeflop.com/blog/stm32-com-ide-arduino-primeiros-passos/>. Acesso em: 9 de agosto de 2019.
- [11] *CONHECENDO O STM32 OU BLUEPILL*. Disponível em: <https://www.curtocircuito.com.br/blog/conhecendo-o-bluepill-stm32/>. Acesso em: 10 de agosto de 2019.
- [12] *STM32CubeMX*. Disponível em: <https://www.st.com/en/development-tools/stm32cubemx.html>. Acesso em: 10 de agosto de 2019.
- [13] YIU, Joseph. *The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors*. 3a Edição. Cambridge, UK: Elsevier, 2014.
- [14] BARTSCH, Arthur Garcia et al. *Computational Cost Evaluation Method to Embedded Digital Control Systems*. IEEE Latin America Transactions, v. 14, n. 2, p. 437-444, 2016.
- [15] CORAIM, Aline. *O SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL (SENAI) E AS SÉRIES METÓDICAS DE ENSINO*. XVI ENDIPE - Encontro Nacional de Didática e Práticas de Ensino. 2012.
- [16] LONGO, Guilherme Augusto Rodrigues; LAMBLÉM, Regina Litz; DUARTE, Marco Aparecido Queiroz. *USO DO SCILAB E DO OCTAVE COMO OPÇÕES PARA O ENSINO DA MATEMÁTICA E RESOLUÇÃO DE PROBLEMAS EM ANÁLISE NUMÉRICA*. ANAIS DO ENIC, v. 1, n. 4, 2012.
- [17] DEBRAY, Y. *Scilab Arduino Blink LED Tutorial*. Disponível em: https://atoms.scilab.org/toolboxes/arduino/1.5/files/Scilab_Arduino_Blinking_LED_Tutorial.pdf. Acesso em: 15 de agosto de 2019.