

V Arduino: Um Componente Virtual do Arduino UNO como Dispositivo Periférico do Simulador CompSim

Lucas Fontes Cartaxo¹, Cícero Samuel Mendes¹, Edson Barbosa Lisboa²,
Guilherme Álvaro Esmeraldo¹

¹Laboratório de Sistemas Embarcados e Distribuídos – Instituto Federal do Ceará
campus Crato – Crato – CE – Brazil

²Laboratório de Estudos Avançados em Eletrônica – Instituto Federal de Sergipe
campus Aracaju – Aracaju – SE – Brazil

{lfontesc, mr.samuelmendes}@gmail.com,
edson.lisboa@academico.ifs.edu.br, guilhermealvaro@ifce.edu.br

Abstract. *Computer Architecture and Organization is a course that deals with computer design, whereas the Input/Output Subsystems is one of its most complex and extensive topics. The scientific literature has opted for the use of computer simulators to support pedagogical practices, however, due to the use of very abstract simulation tools, the input/output topic is treated in a purely conceptual way, leaving gaps in its learning. This paper presents a new learning tool, which emulates different electronic components to allow the creation of virtual peripherals and thereby support the teaching-learning process in Input/Output subsystems.*

Resumo. *Organização e Arquitetura de Computadores é uma disciplina que trata do projeto de computadores, tendo o tópico Subsistemas de Entrada/Saída como um dos mais complexos e extensos. A literatura científica tem optado pelo uso de simuladores computacionais para apoio às práticas pedagógicas, porém, devido ao uso de ferramentas de simulação muito abstratas, o tópico de entrada/saída é tratado de forma puramente conceitual, deixando lacunas no seu aprendizado. Neste artigo é apresentada uma nova ferramenta de aprendizagem, a qual emula diferentes componentes eletrônicos para permitir a criação de periféricos virtuais e com isso apoiar o processo de ensino-aprendizagem em subsistemas de Entrada/saída.*

1. Introdução

Em cursos técnicos e superiores nas áreas de computação e eletrônica, a disciplina de Arquitetura e Organização de Computadores (AOC) trata do projeto de sistemas computacionais, cujos conteúdos variam desde o estudo das estruturas e comportamento dos componentes do computador até sua programação em baixo nível [ACM, 2013] [Zorzo et al., 2017]. Entre esses conteúdos, um dos mais importantes consiste em Subsistemas de Entrada/Saída, o qual refere-se à interação do sistema computacional com seus periféricos. Essa interação envolve diversos conceitos, como os de interfaces, barramentos e protocolos de comunicação, módulos de Entrada/Saída (E/S) e os próprios periféricos. Esses conceitos são considerados amplos e complexos e têm sido frequentemente abordados de forma puramente conceitual [Larraza-Mendiluze e Garay-Vitoria, 2014].

Na literatura tem-se optado pela utilização de simuladores computacionais para apoio ao aprendizado dos diferentes aspectos de AOC [Fernandes e Silva, 2017]. Os simuladores são ferramentas que buscam representar cenários reais no projeto de sistemas computacionais e têm como principais benefícios a abstração dos diferentes recursos do computador, não necessitarem de laboratórios de hardware e de apoio de técnicos especializados e permitem rápido *feedback* nas simulações. Entretanto, segundo Esmeraldo et al. (2019), por conta das abstrações das ferramentas de simulação, não é possível a exploração dos detalhes técnicos que os periféricos reais apresentam. Nesse sentido, na literatura, alguns simuladores, como o SimuS [Neto, Borges e Silva, 2017] e CompSim [Lisboa et al., 2018], vêm com a proposta de oferecer simulações com interações com hardware físico. O Simulador CompSim, particularmente, explora, de forma prática, a interação entre um computador virtual simulável e seus periféricos, que podem ser virtuais ou reais (físicos). Para tanto, o CompSim dispõe de uma interface de E/S padronizada que permite conectar o sistema computacional simulado a diferentes periféricos. Entre seus periféricos, há uma interface para controle de uma plataforma de prototipação Arduino, com a qual é possível realizar experimentos que envolvam desde o projeto físico de periféricos eletrônicos até sistemas computacionais reais completos.

O uso do Arduino como uma ferramenta para o ensino de microcontroladores e subsistemas de E/S tem se tornado bastante popular, devido a sua simplicidade e baixo custo. No entanto, alguns estudos, como os apontados nos trabalhos [Gonçalves e Duraes, 2019] e [Katayama et al., 2019], demonstram que a utilização de Arduino pode apresentar desvantagens, tais como: alta demanda de tempo para configuração e montagem dos experimentos, considerando o curto período de tempo das aulas práticas em laboratório; dependendo do cenário de projeto, pode necessitar de maiores habilidades técnicas em eletrônica; com o tempo, há o desgaste de seus componentes eletrônicos; dependência de vários componentes eletrônicos complementares que podem elevar rapidamente o custo; e, estudantes que não possuem o nível adequado em eletrônica podem projetar circuitos sem os devidos cuidados técnicos que podem danificá-lo e/ou a seus módulos complementares.

Neste artigo é apresentado um novo simulador da plataforma Arduino, chamado de “VArduino” (Virtual Arduino), o qual foi concebido como um periférico virtual do CompSim, para apoiar o aprendizado em subsistemas de E/S na disciplina de AOC. O simulador proposto consiste de uma ferramenta de software com interface gráfica que remete a uma plataforma Arduino e a diferentes componentes eletrônicos básicos, tais como, LEDs, botões, *display* e sensores. O simulador de Arduino aqui proposto, quando integrado ao CompSim, permite a criação de cenários virtuais de projetos de sistemas computacionais semelhantes aos reais, onde pode-se explorar os detalhes de subsistemas de E/S, e com isso abstrair a necessidade da utilização do hardware físico e, consequentemente, tornar o aprendizado mais dinâmico.

O restante deste artigo está dividido da seguinte maneira: A Seção 2 apresenta os trabalhos relacionados; A Seção 3 apresenta os materiais e métodos utilizados no desenvolvimento deste trabalho; Na Seção 4, introduz-se o simulador CompSim e a sua interface com periféricos; A Seção 5 apresenta o simulador de Arduino aqui proposto; Na Seção 6, realiza-se uma discussão sobre o potencial de aprendizado em subsistemas de E/S pelo uso do simulador de Arduino aqui proposto; e, por fim, a Seção 7 demarca as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

O avanço das tecnologias de circuitos integrados e a alta demanda por novas aplicações embarcadas, fez com que a indústria apresentasse diferentes soluções para projetos de sistemas computacionais. Entre as soluções, estão as plataformas de prototipação que, em termos gerais: são placas de circuitos que incluem componentes eletrônicos comerciais, frequentemente, baseados em microcontroladores ou microprocessadores; são utilizadas para atender a requisitos de processamento e/ou de E/S do sistema em desenvolvimento; bem como simplificam o desenvolvimento e testes de novos projetos eletrônicos [Gretlein, 2013]. As plataformas de prototipação são regularmente utilizadas em estágios iniciais do projeto de um sistema computacional por trazerem *feedback* rápido na verificação/validação dos requisitos, bem como possibilitam a realização de refinamentos antes da produção do sistema final. A Tabela I apresenta opções de plataformas de prototipação com suas respectivas características.

Tabela 1. Comparativo entre plataformas abertas de prototipação.

Característica	Plataforma de Prototipação				
	Raspberry Pi 3 Model A+	BeagleBone Black RevC	ESP32 Bluetooth	NodeMCU ESP8266	Arduino UNO
Preço médio (R\$)*	299,00	500,00	85,00	50,00	55,00
Área física (mm x mm)	60 x 56	86,4 x 54,6	52 x 28	49 x 25,5	76 x 64
Capacidade de memória	512 MB SDRAM	512 MB SDRAM	520 KB SRAM	128 kB SRAM	2 KB SRAM
Frequência da CPU (MHz)	1.400	1.000	60 a 240	80	16
Número de pinos de E/S	40	65	11	11	19
Sistema Operacional	Sim	Sim	Não	Não	Não

*Computou-se as médias a partir de sites especializados em vendas de componentes eletrônicos.

Analisando a Tabela 1, percebe-se que os modelos Raspberry Pi 3 Model A+ e BeagleBone Black RevC apresentaram as melhores configurações de capacidade de armazenamento em memória, frequência de operação e número de pinos de E/S, porém, seus custos médios de aquisição são bem superiores ao das demais, além de necessitarem de suporte de um sistema operacional embarcado para operação (este último fator agrega complexidade ao uso da plataforma). Os modelos Esp32 Bluetooth e NodeMCU ESP8266 apresentaram menores áreas físicas, maiores capacidades de memória e maiores frequências de operação em relação ao Arduino UNO. Ainda comparando ao Arduino UNO, ressalta-se que o NodeMCU ESP8266 apresenta menor custo médio de aquisição. Por fim, a plataforma Arduino UNO, em relação a essas duas últimas, se destacou com maior número de pinos de E/S, bem como custo médio de aquisição inferior ao ESP32 Bluetooth.

Considera-se que a plataforma de prototipação escolhida será utilizada como periférico do simulador CompSim, ou seja, será utilizada para construção de periféricos básicos. Ao estabelecer um *trade-off* entre as métricas apresentadas na Tabela 1, pode-se concluir que o Arduino UNO apresenta o melhor custo-benefício, pois além de não necessitar de um sistema operacional embarcado para operação, possui um baixo custo médio de aquisição e um número intermediário de pinos de E/S.

O gráfico da Figura 1 mostra as tendências de pesquisas (popularidade), nos últimos 5 anos, das plataformas de prototipação apresentadas na Tabela 1. Percebe-se que, ao longo de todo esse período, Arduino UNO permaneceu a mais popular, pois, além de estar disponível para aquisição em vários web sites de vendas de produtos eletrônicos, há uma farta disponibilidade de materiais didáticos digitais, tais como livros, tutoriais, apostilas, exemplos de experimentos e de projetos completos, fóruns de discussões, entre outros. Alguns estudos, como o apresentado em [Costa e Silva, 2018], citam que o Arduino UNO é uma das plataformas mais utilizadas no mundo.

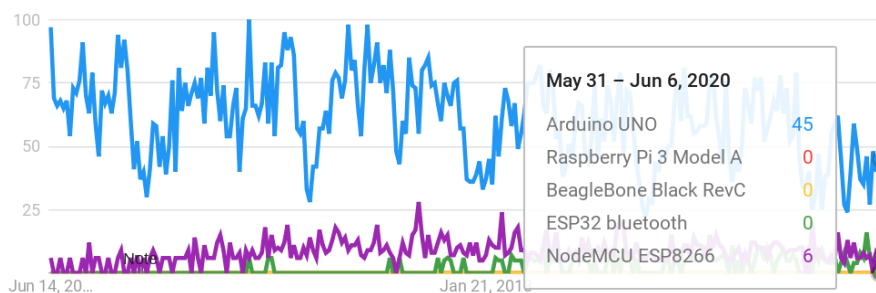


Figura 1. Gráfico comparativo de popularidade entre plataformas de prototipação.

Arduino UNO é uma placa de circuitos eletrônicos baseada no microcontrolador Atmega328 da Atmel. Ele possui 19 pinos de E/S de propósito geral (*General Purpose Input/Output* - GPIO), que são utilizados para interagir com outros componentes eletrônicos por meio de operações de escrita (envio) e leitura (recepção) de dados digitais e analógicos [D'Ausilio, 2012]. O Arduino UNO pode ser programado e utilizado mediante uma conexão via USB com um computador. Assim, ao conectar um Arduino UNO à porta USB do computador, com seus pinos de GPIO e com disponibilidade de componentes eletrônicos complementares, é possível compor diversos tipos de projetos eletrônicos, tais como acender LEDs, ler de dados de sensores e acionar motores.

Na literatura, há diferentes simuladores do Arduino UNO. O estudo apresentado por Gonçalves e Duraes (2019), por exemplo, apresenta um simulador que possui uma interface web e possibilita a criação de diferentes projetos com suporte de alguns componentes eletrônicos virtuais, tais como LEDs, botões, displays e motores. Esse simulador possibilita programar a aplicação na linguagem C/C++ nativa do Arduino e, durante uma simulação, é possível visualizar e interagir com os componentes. Além desse, pode-se encontrar diferentes simuladores de Arduino UNO, tais como: Arduino IO Simulator (2020), que apresenta uma interface para *desktop*, pode ser instalado em diferentes sistemas operacionais e inclui uma gama de complementos para criação de variados tipos de projetos; e Autodesk Tinkercad [PROGRAMMING ELECTRONICS ACADEMY, 2019], que consiste de um portal web educacional para aprendizado em projetos *Maker*, e que busca simplificar a construção de experimentos eletrônicos e compartilhá-los entre usuários do portal. Uma vez que os simuladores de Arduino UNO, presentes na literatura, necessitam ser programados a partir da linguagem C/C++ nativa do Arduino, percebe-se que, no contexto do CompSim, em que o Arduino UNO não é utilizado como plataforma computacional do sistema – ele é utilizado apenas para construção de periféricos e não há necessidade de programá-lo –, conclui-se que a criação de um novo simulador torna-se uma tarefa mais proveitosa, em função de buscar integrar com os já existentes.

3. Materiais e Métodos

Para o desenvolvimento do simulador aqui proposto, o primeiro passo consistiu no aprofundamento teórico em projetos de sistemas computacionais embarcados. Fez-se um levantamento bibliográfico e verificou-se que, dentre as opções, o Arduino UNO é uma das mais populares e possui recursos suficientes para o desenvolvimento de variados projetos eletrônicos. Em seguida, realizou-se um estudo detalhado dos recursos do Arduino UNO, visando compreender todos os seus fundamentos e características para criação de projetos eletrônicos, bem como levantou-se os requisitos considerados no desenvolvimento do novo simulador. Neste passo, foram utilizados os seguintes materiais: uma placa de circuitos do Arduino UNO e componentes eletrônicos analógicos diversos, entre eles: LEDs, botões, resistores, potenciômetros, *display* de 7 segmentos, *buzzer* (campainha), *jumpers* (fios), sensores de luminosidade (LDR), de temperatura (NTC) e de umidade (higrômetro), entre outros.

O simulador CompSim possui um modelo de interface padronizada de E/S que permite a inclusão de novos periféricos a sua plataforma computacional virtual (ver Subseção 4.1). Desta forma, o simulador do Arduino UNO aqui proposto buscou seguir esse modelo de interface, de forma a simplificar sua integração. Para a codificação do simulador proposto, utilizou-se as mesmas tecnologias empregadas no desenvolvimento do simulador CompSim, como a linguagem Python v3.x e o *framework* gráfico Tkinter.

Após a codificação do novo simulador Arduino UNO, foram realizados diferentes estudos para verificar se ele seguiu os requisitos levantados para o projeto (e.g. integração com interface de E/S, suporte de interface gráfica, protocolos de comunicação e desempenho) e o comportamento de um Arduino UNO real.

4. O Simulador CompSim

CompSim é um simulador de sistema completo [Esmeraldo et al., 2019], o qual inclui uma Plataforma Virtual de Hardware (PVH) simulável composta por processador, memórias cache e RAM, barramento de sistema (comunicação entre o processador e as memórias) e um barramento de periféricos (comunicação entre processador e periféricos). No barramento de periféricos, há o Subsistema de E/S, o qual consiste de uma interface padronizada que simplifica a inclusão de novos periféricos e sua comunicação com o processador. A Figura 1 ilustra a interface gráfica do CompSim.

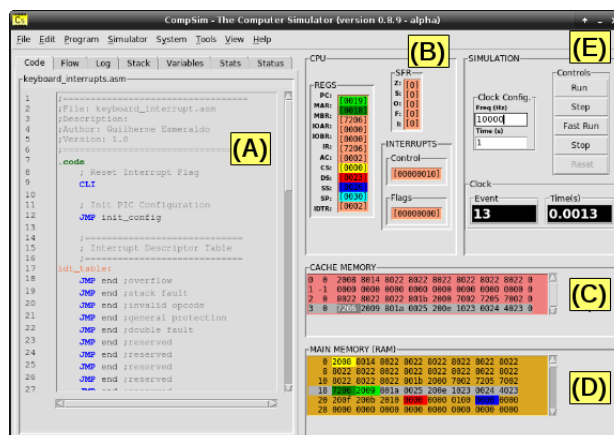


Figura 1. Interface gráfica do Simulador CompSim.

A interface gráfica do CopmSim permite configurar os componentes da PVH,

suportar a codificação de novos programas, realizar simulações e visualizar o comportamento da PVH durante uma simulação. A Figura 1 mostra a interface gráfica do CompSim, onde pode-se visualizar os componentes gráficos: A) Editor de código; B) Painel de visualização do Processador; C) Painel de visualização da Memória cache; D) Painel de visualização da Memória RAM; e E) Componentes de configuração e execução de simulações.

4.1. Interface Padronizada para Conexão com Periféricos

O simulador CompSim conta com uma interface de E/S padronizada para permitir a conexão simplificada de novos periféricos à PVH. Um periférico torna-se “plugável” à PVH ao implementar essa interface, sendo possível, por exemplo, plugar determinados periféricos para compor um sistema computacional específico.

Cada periférico plugável deve incluir um arquivo de especificação de sua interface (arquivo com extensão “.csd”), o qual deve incluir as seguintes definições: número(s) da(s) porta(s) de E/S; número da interrupção (IRQ), caso o periférico suporte; dados para instanciar o componente de software (nome do pacote e da classe de software); e uma descrição textual para identificação do periférico. Além disso, o software do periférico deve ser codificado com a linguagem de programação Python v3.x, assim, além do arquivo “.csd”, também haverá um outro arquivo com extensão “.py”. Desta maneira, ao se criar uma PVH no simulador, sua interface de E/S padronizada buscará os arquivos de descrição de interface de periférico (“.csd”) e, de forma automatizada, instanciará os respectivos componentes de software (via arquivos “.py”), que serão plugados ao barramento virtual de periféricos, como está ilustrado na Figura 2. Após esse processo, os periféricos virtuais já estarão aptos para interação com o restante dos componentes de hardware do sistema computacional simulado.

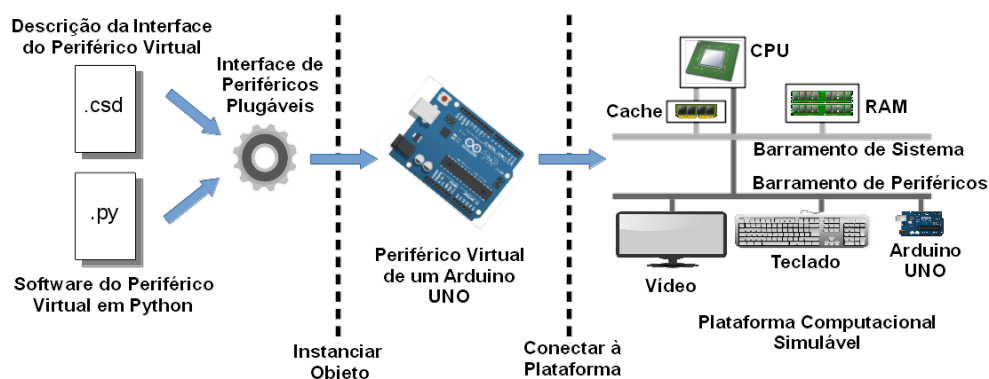


Figura 2. Ilustração da criação de um periférico virtual plugável e conexão com a plataforma computacional.

Atualmente, o CompSim conta oficialmente com os periféricos Teclado, Vídeo e Arduino UNO, como podem ser vistos nas Figuras 3-A, 3-B e 3-C, respectivamente. Os componentes Teclado e Vídeo são do tipo virtual, pois consistem de software para emular os comportamentos dos respectivos periféricos reais. Já o periférico Arduino UNO é dividido em dois submódulos, sendo que o primeiro consiste de um software que implementa a integração do dispositivo ao barramento de periféricos, e o segundo consiste de uma placa de circuitos física da plataforma aberta de prototipação Arduino UNO [Badamasi, 2014]. Com este periférico do Arduino UNO é possível criar cenários de projetos de sistemas computacionais em que pode-se realizar a simulação da PVH interagindo com hardware real.

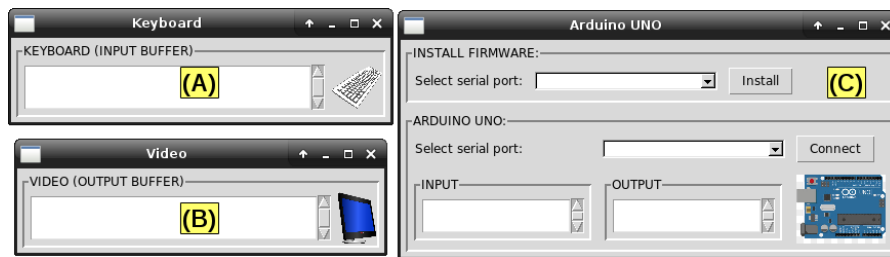


Figura 3. Periféricos virtuais plugáveis do CompSim.

5. O Componente VArduino

O simulador VArduino foi desenvolvido como um componente virtual gráfico, plugável à PVH do CompSim, para apoiar o estudo das interações de E/S entre o sistema computacional simulado e periféricos montados com apoio do Arduino UNO. A ideia básica é permitir a criação de cenários virtuais de projetos de sistemas computacionais semelhantes aos reais, abstraindo a necessidade do uso de um Arduino UNO físico.

A interface gráfica do VArduino, que pode ser vista na Figura 4, é composta por um Arduino UNO virtual e por diferentes componentes gráficos que emulam dispositivos eletrônicos reais, denominados aqui de Dispositivos Eletrônicos Virtuais (DEV).

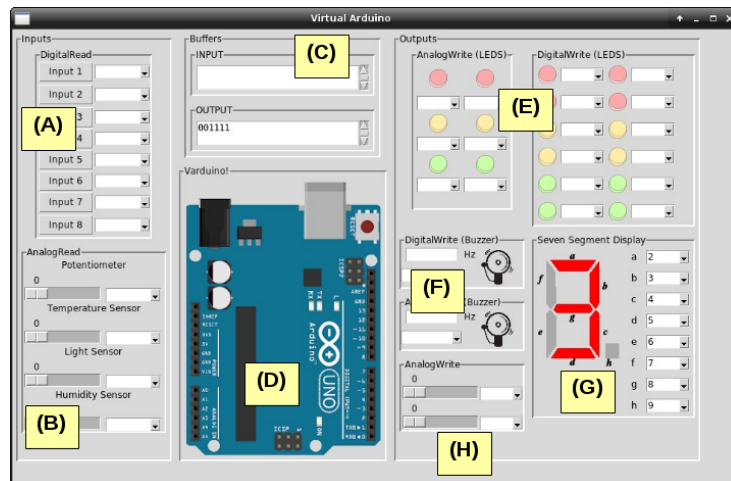


Figura 4. Interface gráfica do Arduino Virtual.

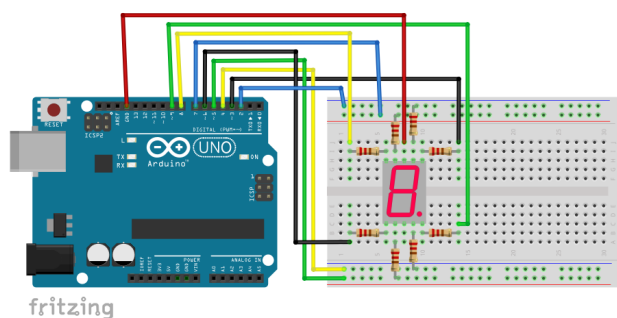
A interface gráfica está dividida em três macroblocos de componentes gráficos, onde: do lado esquerdo, estão os DEVs de entrada de dados, que são 8 botões (*push buttons*) (Figura 4-A), 1 potenciômetro e 3 sensores analógicos (temperatura, luminosidade e umidade) (Figura 4-B); ao centro, os componentes gráficos apresentam os *buffers* de entrada (*input*) e saída (*output*) (Figura 4-C), usados na comunicação entre CompSim e VArduino, e, logo abaixo, há o diagrama do Arduino UNO virtual (Figura 4-D); por fim, à direita, estão os DEVs de saída de dados, que são 18 LEDs (Figura 4-E) e *buzzers* (Figura 4-F), os quais podem ser utilizados em operações de escrita digital/analgica, um *display* de 7 segmentos (Figura 4-G), e dois dispositivos genéricos de escrita analógica (Figura 4-H), que podem simular, por exemplo, o acionamento de *coolers*/exaustores de computadores, acionar e controlar o movimento de motores, gerar sinais modulados para controle de um LED infravermelho (e.g. para uso como controle remoto), gerar efeitos sonoros, entre outros.

Para o controle dos DEVs, é necessário inicialmente conectá-los ao VArduino. Para tanto, cada DEV inclui uma ou mais caixas de combinação (*combo boxes*), nas quais é possível escolher em quais pinos de GPIO do VArduino os dispositivos estarão conectados (ver Figura 4). O VArduino simula os quatro modos de operação dos pinos de GPIO de um Arduino UNO real, que são escrita e leitura digital, bem como escrita e leitura analógica. A partir do suporte de conexões aos pinos de GPIO e dos modos de operação do VArduino, uma PVH poderá interagir com os DEVs.

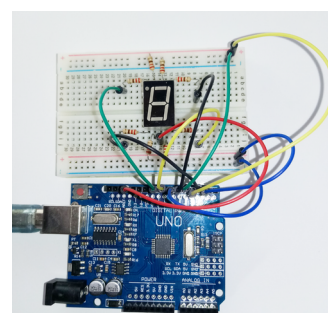
6. Discussão e Resultados

Para demonstrar os benefícios do uso do simulador VArduino, foi definido um estudo de caso que consistiu na criação de um sistema computacional virtual, no simulador CompSim, para controlar um periférico com um *display* de 7-segmentos. Para a criação do sistema computacional virtual, inicialmente, no CompSim, foi criada e configurada uma PVH e, em seguida, codificou-se o software da aplicação de controle do *display* de 7-segmentos.

Em um primeiro instante, para compor o periférico do estudo de caso, optou-se por utilizar um periférico de hardware físico. Assim, foi necessário, além do periférico plugável do Arduino UNO (ver Figura 3-C), o suporte dos seguintes componentes eletrônicos: uma placa de circuitos do Arduino Uno rev3, um *display* de 7-Segmentos (do tipo ânodo comum), 8 resistores de 220 Ohms, uma *proto board* e 9 *jumpers* (fios). O diagrama de montagem e o periférico físico montado podem ser vistos na Figura 5 (a) e (b), respectivamente. Considerando que o cenário proposto para o estudo de caso consiste de uma aplicação simples (controle de um *display* de 7-segmentos), analisando a Figura 5, percebe-se que a confecção do periférico físico, por um estudante não introduzido em eletrônica, pode se tornar uma tarefa complexa, pois envolve o conhecimento das especificidades e a aquisição de cada componente eletrônico, além dos procedimentos necessários para montagem do circuito com auxílio da *proto board*.



(a) Diagrama de montagem do periférico



(b) Periférico montado.

Figura 5. Periférico físico para controle de display de 7-segmentos.

Em um segundo momento, considerando que foi utilizado o mesmo sistema computacional virtual criado anteriormente no CompSim, optou-se por utilizar o VArduino para implementação do periférico do estudo de caso. Sabendo que o VArduino inclui um Arduino UNO e um *display* de 7-segmentos virtuais, o passo seguinte consistiu em configurar as conexões entre eles. Esse procedimento foi realizado pelo uso das caixas de combinação, selecionando quais pinos de GPIO do Arduino UNO estão conectados a cada um dos LEDs do *display* de 7-segmentos. Na Figura 4-G, à direita do *display* de 7-segmentos, pode-se visualizar as caixas de

combinação com a configuração de conexão dos pinos de GPIO com os segmentos (LEDs). Ressalta-se que cada um desses segmentos recebe uma identificação, uma letra de “a” à “h”, e que os números nas caixas de combinação ao lado consistem dos respectivos pinos de GPIO utilizados nas conexões.

Analisando as configurações de conexão dos pinos de GPIO do Arduino UNO com os LEDs do *display* de 7-segmentos nos periféricos físico e VArduino, apresentadas nas Figuras 5 e 4-G, respectivamente, percebe-se que são idênticas. No entanto, observa-se que ao abstrair o uso de dispositivos eletrônicos, a solução aqui proposta para simulação de uma plataforma Arduino UNO, VArduino, trata diretamente os problemas apontados em [Gonçalves e Duraes, 2019] e [Katayama et al., 2019], eliminando assim os custos na aquisição e manutenção dos componentes eletrônicos, bem como reduzindo os ônus no aprendizado dos conceitos básicos de eletrônica e no manuseio desses dispositivos.

7. Conclusões

Neste artigo apresentou-se o VArduino, um simulador criado para apoiar o processo de ensino-aprendizagem em conceitos de subsistemas de E/S nas disciplinas de AOC. VArduino inclui componentes gráficos que emulam um Arduino UNO e diferentes componentes eletrônicos, e pode ser utilizado na criação de diferentes periféricos virtuais para o simulador CompSim. O estudo de caso apresentado mostrou que, em relação a uma placa física do Arduino UNO, ao utilizar o VArduino, pode-se eliminar custos de aquisição e manutenção de componentes eletrônicos, abstrair a necessidade de conhecimentos prévios de eletrônica básica e tornar a criação de novos periféricos mais dinâmica.

O simulador CompSim, que já vem sendo utilizado por turmas de diferentes cursos técnicos e de graduação, nas áreas de computação e eletrônica, tem se mostrado efetivo no apoio ao processo de ensino-aprendizagem dos conceitos de AOC. Da mesma maneira, espera-se, no futuro, também validar o VArduino junto à comunidade estudantil, através da criação de diferentes cenários de aplicações computacionais com requisitos de E/S, para aplicação em práticas laboratoriais. Desta maneira, espera-se que os *feedbacks* possam contribuir na correção de falhas, adição de novos recursos, no aumento da qualidade da solução e no aperfeiçoamento do apoio ao aprendizado em subsistemas de E/S.

Referências

- ACM. Association for Computing Machinery; IEEE Computer Society (2013) “Curriculum Guidelines for Undergraduate Degree Programs in Computer Science”. Disponível em: <https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf>. Acesso em: 8 jun. 2020.
- ARDUINO IO SIMULATOR (2020) Disponível em: <<https://xevro.be/products/arduino%2520simulator%25201.5.html>>. Acesso em: 8 jun. 2020.
- PROGRAMMING ELECTRONICS ACADEMY. “The Arduino Simulator you’ve been looking for!” (2020) Disponível em: <<https://www.programmingelectronics.com/arduino-simulator-tinkercad/>>. Acesso em: 8 jun. 2020.

- Badamasi, Y. A. (2014) "The working principle of an Arduino", In: 11th International Conference on Electronics, Computer and Computation (ICECCO), p. 1-4.
- Costa, E.; Silva, G. P. (2018) "Um Simulador Didático para o Ensino de Arquitetura de Computadores e Internet das Coisas", In: International Journal of Computer Architecture Education (IJCAE), v. 7, n. 1, p. 29-38.
- D'Ausilio, A. (2012) "Arduino: A low-cost multipurpose lab equipment", In: Behavior research methods, 44(2), p. 305-313.
- Esmeraldo, G. A. R. M.; Mendes, C. S. R.; Cartaxo, L. F.; Lisboa, E. B. (2019) "Apoio ao Aprendizado em Arquitetura e Organização de Computadores: Um Estudo Comparativo entre Simuladores Computacionais", In: Revista Tecnologias na Educação, 2019.
- Fernandes, S. R.; Silva, I. S. (2017) "Relato de Experiência Interdisciplinar Usando MIPS", In: International Journal of Computer Architecture Education (IJCAE), V. 6, n. 1, p. 52-61.
- Gonçalves, P. F.; Duraes, J. (2019) "An Arduino Simulator for Practical Embedded Programming Teaching", In: International Symposium on Ambient Intelligence and Embedded Systems.
- Gretlein, S. (2013) "Software Modeling for Embedded Systems", In: Software Engineering for Embedded Systems. Newnes, p. 59-91.
- Katayama, T.; Nishida, T.; Kita, Y.; Yamaba, H.; Aburada, K.; Okazaki, N. (2019) "Implementation of Arduino Simulator ADVIS Visualizing the Value of Voltage on the Circuit", In: Journal of Robotics, Networking and Artificial Life, 5(4), p. 249-252.
- Larrazza-Mendiluz, E.; Garay-Vitoria, N. (2014) "Approaches and tools used to teach the computer input/output subsystem: A survey", In: IEEE Transactions on Education, v. 58, n. 1, p. 1-6.
- Lisboa, E. B.; Cartaxo, L. F.; Mendes, C. S. R.; Esmeraldo, G. A. R. M. (2018) "Ambiente Integrado de Hardware e Software Aplicado ao Ensino de Projeto de Sistemas Computacionais", In: III Congresso sobre Tecnologias na Educação (Ctrl+E 2018).
- Neto, A. M. A.; Borges, J. A. dos S.; Silva, G. P. (2017) "Extensão do Simulador SimuS com uso do Protocolo Firmata", In: XVIII Workshop de Iniciação Científica do XVII Simpósio em Sistemas Computacionais de Alto Desempenho (WIC-WSCAD), p.123-128.
- Zorzo, A. F.; Nunes, D.; Matos, E.; Steinmacher, I.; Leite, J.; Araujo, R. M.; Correia, R.; Martins, S. (2017) "Referenciais de formação para os cursos de graduação em computação", In: Sociedade Brasileira de Computação (SBC), 153p.