

Ambiente Integrado de Hardware e Software Aplicado ao Ensino de Projeto de Sistemas Computacionais

Edson Lisboa¹, Lucas Fontes², Cícero Samuel Mendes², Guilherme Esmeraldo²

¹Instituto Federal de Educação, Ciência e Tecnologia de Sergipe – Aracaju

²Instituto Federal de Educação, Ciência e Tecnologia do Ceará – Crato

edson.lisboa@academico.ifs.edu.br, lfontesc@gmail.com,
mendes.samuel99@gmail.com, guilhermealvaro@ifce.edu.br

Abstract. *Computer Organization and Architecture is a complex discipline for dealing with the design of computer systems having to be aligned with new technological trends. The scientific literature has opted for the use of simulation tools in pedagogical practices, due to the high cost for creating and maintaining specialized laboratories. However, due to purely theoretical classes and/or the use of very abstract simulation tools, there are gaps in the teaching-learning process of these concepts. This paper presents a new learning tool that combines simulation to interactions with physical and low-cost devices to establish complete scenarios of hardware projects.*

Resumo. *Organização e Arquitetura de Computadores é uma disciplina complexa por tratar do projeto de sistemas computacionais tendo que estar alinhada às novas tendências tecnológicas. A literatura científica tem optado pelo uso de ferramentas de simulação, nas práticas pedagógicas, em função do alto custo para criação e manutenção de laboratórios especializados. Porém, devido às aulas puramente teóricas e/ou ao uso de ferramentas de simulação muito abstratas, surgem lacunas no processo de ensino-aprendizagem desses conceitos. Este artigo apresenta uma nova ferramenta de aprendizagem que alia simulação a interações com dispositivos físicos e de baixo custo, para estabelecer cenários completos de projetos de hardware.*

1. Introdução

Organização e Arquitetura de Computadores (OAC) é uma das disciplinas mais complexas presentes em cursos de Computação [Zorzo et al. 2017], pois além de tratar de assuntos extensos, e não triviais, relacionados ao projeto e análise de sistemas digitais e de computadores, deve estar alinhada às novas tendências tecnológicas, como são os casos de Internet das Coisas (IoT), Sistemas Embarcados, Sistemas de Computação Paralela e Sistemas Reconfiguráveis [Ghaffarian 2016]. A soma desses fatores resulta em grandes desafios para o processo de ensino-aprendizagem nessa área e demanda novas abordagens, métodos e ferramentas de hardware e software, inclusive, fazendo uso de artefatos resultantes das inovações tecnológicas.

O uso de simuladores para apoiar o aprendizado em OAC é uma prática comum e, na literatura, há variados tipos e com diferentes características funcionais e não funcionais. Alguns trabalhos, como os apresentados em [Nikolic et al. 2009] e [Penna, Freitas 2013], realizam comparações entre diferentes simuladores, buscando estabelecer critérios para adoção de algum deles para apoio em cursos de OAC. Entre os critérios de

comparação, destacam-se: granularidade (nível de detalhes, opções de configuração e variabilidade de componentes de hardware); desempenho e precisão dos resultados da simulação; suporte à simulação de aplicações de usuário; e interfaces gráficas interativas.

Contudo, de acordo com o estudo apresentado em [Larraza-Mendiluze, Garay-Vitoria 2015], dentre os tópicos abordados em OAC, há lacunas no processo de ensino-aprendizagem em “subsistemas de entrada/saída”, devido aos métodos de ensino empregados que limitam-se a aulas puramente teóricas e/ou aos altos níveis de abstração das ferramentas de simulação utilizadas, não possibilitando a exploração dos detalhes técnicos que os dispositivos reais apresentam. Esse mesmo trabalho aponta ainda que existe a necessidade de mais estudos para a educação nessa área e que é um campo importante não só para OAC, mas também para outras áreas, como, por exemplo, projeto de sistemas computacionais, sistemas operacionais e interação homem-computador.

Outro problema comum é a ausência de exposição dos estudantes ao hardware físico, onde, o contato com os componentes eletrônicos físicos incentiva o aprendizado de todas as fases de desenvolvimento de um sistema digital ou periférico [Black 2016]. Nesse sentido, o desafio maior torna-se a composição de laboratórios especializados, com suporte de profissionais experientes, e a disponibilidade dos materiais a serem utilizados nas aulas práticas. Alguns trabalhos, como os apresentados em [Black 2016][Neto, Borges, Silva 2017], buscaram contornar esse problema pelo uso do Arduino em práticas laboratoriais. Arduino é uma plataforma eletrônica baseada em microcontrolador, com especificação aberta e tem sido largamente utilizada em projetos de sistemas eletrônicos por estudantes, hobistas e profissionais [Arduino 2018]. Seu grande sucesso vem sendo atribuído à sua simplicidade, facilidade de uso, o baixo custo e à diversidade de produtos e complementos criados pela indústria [Kushner 2011].

Diante do exposto, torna-se importante buscar soluções que reúnam a dinâmica e versatilidade dos processos baseados em simulação, acompanhada da importância e dos desafios da exposição ao hardware real. Com esse intuito, este trabalho apresenta o CompSim, uma ferramenta de apoio ao aprendizado em projeto de sistemas computacionais, abrangendo hardware e software de baixo nível, com funcionalidades de simulação e interação com dispositivos de hardware real. O simulador proposto inclui componentes virtuais de hardware, como processador, memórias, barramentos e periféricos, bem como permite a integração ao Arduino, onde este passa a se comportar como um periférico do sistema computacional simulado. Com esse suporte de integração, espera-se tornar o aprendizado em subsistemas de entrada/saída mais atrativo e dinâmico, além de fornecer subsídios para o desenvolvimento de projetos de sistemas digitais físicos, estes com complexidades escaláveis de acordo com as necessidades do processo de ensino-aprendizagem e da metodologia adotada.

Este artigo está estruturado da seguinte maneira: Na seção a seguir, apresenta-se o simulador CompSim; Na Seção 3, é tratado o modelo de interface de entrada/saída do Arduino; A Seção 4 apresenta o modelo de integração CompSim-Arduino; A Seção 5 discute os potenciais dessa integração e, por fim, a Seção 6 apresenta as conclusões e demarca os trabalhos futuros.

2. O Simulador CompSim

CompSim¹ inclui uma plataforma de hardware simulável, chamada de “Mandacaru”, a qual inclui um processador conceitual, memórias cache e RAM, barramentos de sistema e de periféricos, e dispositivos de entrada (Teclado) e de saída (Vídeo). Esses componentes virtuais foram concebidos para suportar o estudo de diferentes conceitos de OAC, como, por exemplo, fases do ciclo de instrução, representação em memória de dados numéricos, não-numéricos e de estruturas de dados, conjunto de instruções da arquitetura (tipos, tamanhos e formatos de instruções), modos de endereçamento, programação em nível de máquina, representação de construções de linguagens de alto nível, cálculo dos tempos de acesso aos dados, entre outros.

O CompSim possui uma interface gráfica que permite configurar os componentes da plataforma Mandacaru, programar uma aplicação para execução no processador conceitual, chamado de “Cariri”, configurar, executar e visualizar a simulação, bem como analisar o desempenho do sistema após uma simulação. A Figura 1 mostra a interface gráfica do simulador CompSim.

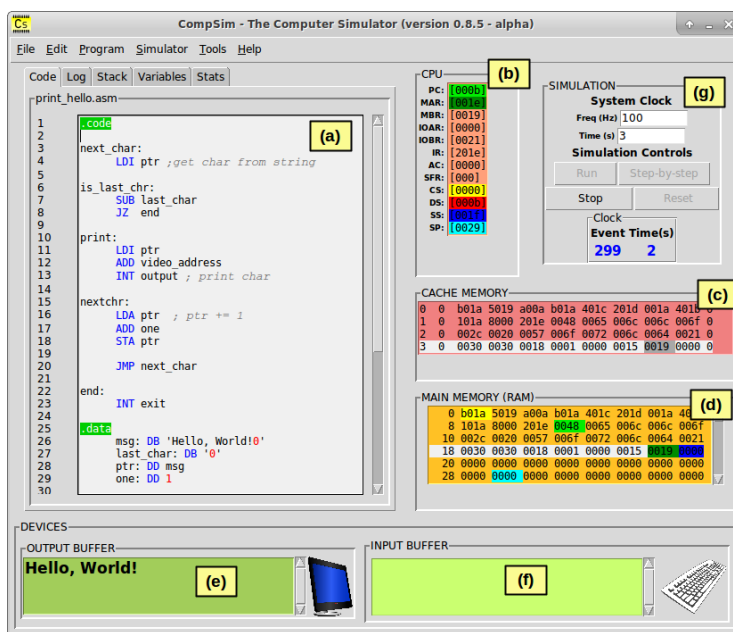


Figura 1 - Interface gráfica do simulador CompSim.

Na Figura 1, pode-se observar os seguintes componentes gráficos: (a) Editor de código: inclui recursos para simplificar a codificação de uma aplicação, como número de linhas, teclas de atalho, um assistente de codificação, entre outros. Este componente está integrado a um montador (*Assembler*) que realiza análises léxica, sintática e semântica no código-fonte da aplicação, tradução deste para *bytecodes*, com cópia para memória RAM; (b) CPU: durante uma simulação, exibe os registradores do processador Cariri e respectivos valores assumidos. Os registradores de endereçamento possuem cores diferenciadas, onde as respectivas cores são utilizadas para referenciar diferentes posições da memória RAM.; (c) Memória cache: exibe as linhas da cache e respectivos

¹ CompSim. Apresentação. Disponível em: <<https://www.youtube.com/watch?v=eQTsdwTi5ts>>. Acesso em: 06 abr. 2018.

valores. Este componente gráfico destaca a linha e a palavra endereçada pelo processador durante uma simulação; (d) Memória RAM: exibe os conteúdos (instruções e dados) de todos os endereços da componente virtual memória RAM. Os componentes gráficos Memória RAM e CPU estão integrados, de forma que, durante uma simulação, as posições de memória são coloridas de acordo com as respectivas cores dos registradores de endereçamento; (e) Vídeo: exibe os caracteres enviados para o componente virtual de vídeo; (f) Teclado: disponibiliza uma linha de comandos que permite ao usuário interagir com uma aplicação através do componente virtual teclado; e (g) Controles de simulação: inclui controles que permitem configurar o tempo de simulação e frequência de relógio de sistema (*clock*), iniciar, executar passo-a-passo, parar e reiniciar uma simulação.

Na Figura 1, está ilustrada a execução da aplicação “Hello, World!”, a qual levou 3 segundos - um total de 300 ciclos de relógio, a uma frequência de 100 Hz de relógio de sistema - para impressão da *string* no componente virtual de Vídeo (e).

Analisando este cenário, percebe-se que o aprendizado em “subsistemas de entrada/saída” estaria bastante limitado à interações com os programas de usuário, utilizando apenas os componentes gráficos de Vídeo e Teclado. Sendo assim, é fundamental acrescentar recursos que possibilitem a integração de diferentes dispositivos de entrada e saída, tornando-os mais próximos ao que se aplica no desenvolvimento de projetos de sistemas eletrônicos reais. As seções a seguir detalham a integração do simulador CompSim com o Arduino, e discutem as potencialidades resultantes dessa conjunção no aprendizado em entrada/saída e projetos de sistemas digitais.

3. Interface de Entrada/Saída do Arduino UNO

Os microcontroladores Atmega8, Atmega168 e Atmega328p, presentes nas diferentes plataformas Arduino, incluem as portas² B, C e D, que são utilizadas em operações de entrada/saída digitais e analógicas. As operações em cada uma dessas portas podem ser realizadas a partir da manipulação de 3 registradores dos controladores Atmega, que são: 1) DDRx: determina se os bits da porta x serão utilizados para operação de entrada de dados (INPUT ou leitura) ou de saída (OUTPUT ou escrita); 2) PORTx: controla a tensão dos bits relacionados à porta x, onde poderão estar com voltagem alta (HIGH, nível lógico “1”) ou baixa (LOW, nível lógico “0”); e 3) PINx: realiza a leitura do estado dos bits da porta x, quando configurados como entrada (INPUT). Cabe ressaltar que os registradores DDRx e PORTx são de leitura/escrita de dados, enquanto que PIN é apenas de leitura.

Cada bit desses registradores corresponde a um único pino em uma placa de circuito impresso (PCI) do Arduino. Por exemplo, na PCI do Arduino UNO, a qual é ilustrada na Figura 2, o bit 0, dos registradores DDRB, PORTB e PINB, refere-se ao pino PB0 (*Port B pin 0*), que corresponde ao pino digital 8 da interface de entrada/saída na sua PCI.

2

Porta é um espaço de endereçamento que é frequentemente utilizado em operações de entrada/saída.

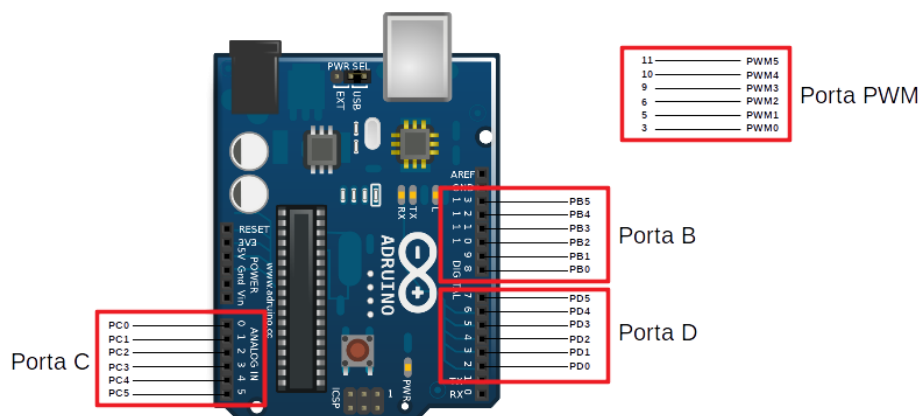


Figura 2 – Interface de entrada/saída do Arduino UNO.

As portas B e D são utilizadas em operações de entrada/saída digitais, enquanto que a porta C utiliza-se apenas em operações de entrada analógica. Os pinos 3, 5, 6 (bits 4, 6 e 7 da porta D) e 9, 10, 11 (bits 1, 2 e 3 da Porta B) podem ainda ser utilizados para saída analógica via PWM (*Pulse With Modulation*³) (a Figura 2 mostra a posição das portas e respectivos pinos em uma PCI Arduino UNO).

4. Interface de Integração CompSim-Arduino UNO

O processador Cariri, presente na plataforma Mandacaru, em seu conjunto de instruções da arquitetura, possui a instrução INT para operações de entrada/saída, com o código de operação 20 para leitura de um caractere a partir de um determinado periférico e código 21 para escrita de um caractere em um determinado periférico.

Essas operações de entrada/saída devem atender às seguintes condições: 1) Em operações de leitura/escrita, os 8 bits mais significativos do Acumulador serão utilizados para endereçamento de periféricos. Com isso, podem ser endereçados até 256 periféricos diferentes (8 bits $\rightarrow 2^8 = 256$ periféricos); 2) Em operações de escrita, os 8 bits menos significativos do Acumulador serão utilizados para guardar o dado que será escrito em algum periférico; e 3) Em operações de entrada, após a leitura de periférico, o dado lido estará disponível no Acumulador. Desta forma, nas operações de entrada/saída, o registrador Acumulador do processador Cariri passa a incluir os campos *AC High* e *AC Low*, que podem ser vistos na Figura 3.

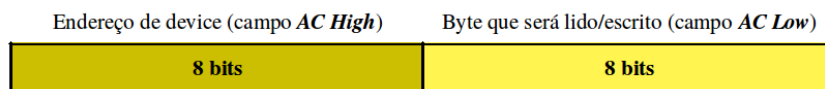


Figura 3 - Campos do registrador Acumulador em operações de Entrada/Saída.

Uma vez que se conecta uma PCI Arduino UNO ao computador onde está sendo executado o CompSim, automaticamente, as instruções “INT 20” e “INT 21” passam a suportar operações de entrada/saída em suas portas. A Figura 4 ilustra a organização da plataforma Mandacaru conectada ao Arduino UNO, onde pode-se observar a existência

3

PWM é uma técnica que utiliza diferentes amplitudes de sinais digitais para representar um sinal analógico.

de controladores entre o barramento de periféricos e os dispositivos Vídeo, Teclado e Arduino UNO. Os controladores são importantes pois traduzem o formato das requisições do processador Cariri para o formato específico de cada periférico.

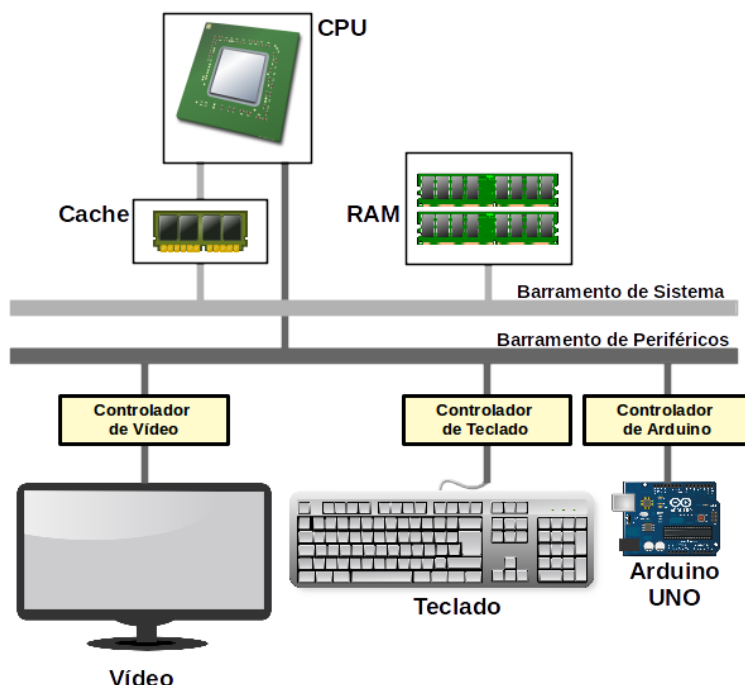


Figura 4 - Plataforma Mandacaru conectada ao Arduino UNO.

Já a Figura 5 detalha a interface do controlador virtual que realiza a integração do barramento de periféricos da plataforma Mandacaru à PCI Arduino UNO. Ao receber uma requisição de entrada/saída proveniente do processador, via barramento de periféricos, e dependendo do tipo de operação e da porta especificada, o controlador realizará uma escrita ou leitura na respectiva interface de pinos da PCI Arduino UNO conectada a ele.

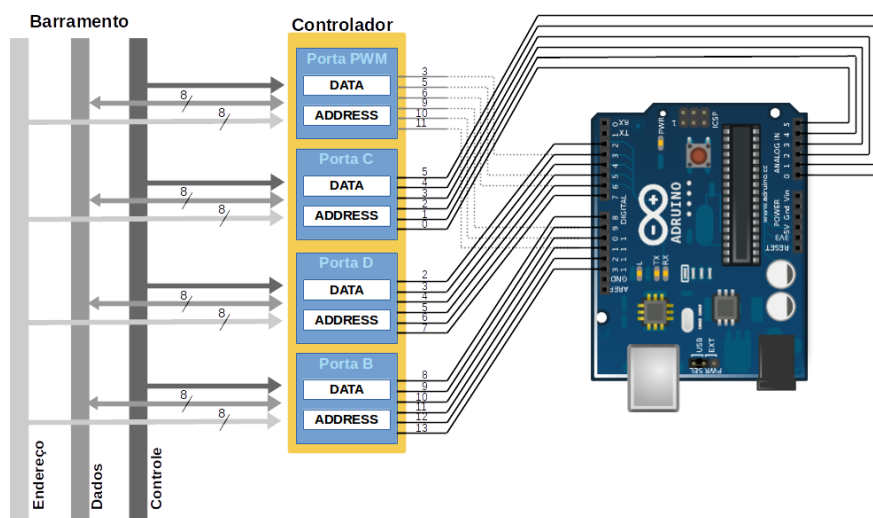


Figura 5 - Detalhe da interface de conexão do Arduino UNO à plataforma Mandacaru via controlador.

Com a integração de uma PCI Arduino UNO ao barramento de periféricos, a plataforma Mandacaru passa a contar com uma gama maior de periféricos de entrada/saída. A Tabela 1 sumariza todos os periféricos, respectivos tipos, modos de operação e número portas.

Tabela 1 - Periféricos do CompSim.

Periférico	Tipo	Modo de Operação	Porta
Vídeo	Virtual	Saída Digital	0
Teclado	Virtual	Entrada Digital	1
Arduino - Porta B	Físico	Entrada/Saída Digital	2
Arduino - Porta C	Físico	Entrada Analógica	3
Arduino - Porta D	Físico	Entrada/Saída Digital	4
Arduino - Porta PWM (Pino 3)	Físico	Saída Analógica	5
Arduino - Porta PWM (Pino 5)	Físico	Saída Analógica	6
Arduino - Porta PWM (Pino 6)	Físico	Saída Analógica	7
Arduino - Porta PWM (Pino 7)	Físico	Saída Analógica	8
Arduino - Porta PWM (Pino 10)	Físico	Saída Analógica	9
Arduino - Porta PWM (Pino 11)	Físico	Saída Analógica	10

Na tabela, pode-se observar que os periféricos Vídeo e Teclado são do tipo virtual, enquanto que os demais são físicos, por estarem relacionados às portas do Arduino UNO. Outro detalhe importante está relacionado à porta PWM, a qual inclui 6 pinos independentes, e, por causa disso, cada um deles recebeu um endereço de porta específico, podendo comportar até 6 periféricos.

5. Discussão de Resultados

Uma vez que a plataforma Arduino passa a integrar o simulador CompSim, o estudo de OAC sofre em incremento no potencial de ensino-aprendizagem em subsistemas de entrada/saída. Agora, os estudantes podem agregar, às experiências práticas em laboratório, projetos de periféricos e de sistemas eletrônicos com Arduino.

Um exemplo básico de periférico físico do CompSim utilizando Arduino UNO pode ser visto na Figura 6. Nesse exemplo, o periférico consiste em um LED ligado a um resistor de 220 Ohms e conectado ao pino 8 da PCI, que pode ser ligado/desligado a partir de operações de entrada/saída pelo processador Cariri.

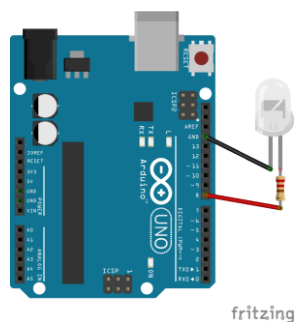


Figura 6 – Diagrama de um periférico com LED conectado ao pino 8 da PCI Arduino UNO.

Os códigos-fonte, ilustrados na Figura 7, mostram como ligar o LED conectado ao pino 8 da PCI Arduino UNO, sendo que em (a) o código é descrito C/C++ nativo da IDE Arduino, e em (b), é descrito em Assembly do processador Cariri.

Observando os dois códigos, nas Figuras 7 (a) e (b), percebe-se que o código Assembly inclui maior número de linhas, porém ao considerarmos apenas as instruções que operam o periférico, percebe-se que: na Linha 5, carrega-se no Acumulador o dado que será escrito na porta B (porta na qual está conectado o LED); na Linha 7, o endereço da porta B é adicionado ao Acumulador; e na Linha 11, realiza-se a operação de saída (nela, o comando de escrita é enviado ao controlador do Arduino UNO, para que ele ative o LED).

```

1. //LED conectado ao pino 8
2. int led = 8;
3.
4. void setup() {
5.     //Seta pino 8 como saída
6.     pinMode(led, OUTPUT);
7. }
8.
9. //rotina para ligar o LED
10. void loop() {
11.     //Realiza escrita digital
12.     digitalWrite(led, HIGH);
13. }
14.

```

(a) Código C/C++ Arduino

```

1. ;Codigo do programa
2. .code
3.
4. ;Carrega dado no AC para escrita
5. LDA high
6.
7. ;Add endereço da porta B ao AC
8. ADD arduino_portB
9.
10. ;Realiza operacao de escrita
11. INT output
12.
13. ;encerra a aplicacao
14. INT exit
15.
16. ; Dados do programa
17. .data
18. high: DD 1;
19. arduino_portB: DD 512
20. output: DD 21
21. exit: DD 25
22.
23. ; Pilha do programa
24. .stack 10

```

(b) Código Assembly CompSim

Figura 7 – Comparativo entre código Arduino e CompSim para periférico da Figura 6.

A aplicação ilustrada nas Figuras 6 e 7 consiste de um exemplo básico de construção de periféricos. Com o mesmo conceito proposto para as operações de entrada/saída digital e analógicas no CompSim, os estudantes poderão criar periféricos mais complexos, como, por exemplo, impressoras 3D, estações meteorológicas, veículos programáveis, robôs, etc. Naturalmente que, para construção de dispositivos eletrônicos mais elaborados, os estudantes necessitarão ser introduzidos à eletrônica

analógica/digital básica ou fazer uso de subsistemas eletrônicos complementares pré fabricados, também conhecidos por *shields*. A indústria atualmente oferece variados tipos de *shields*, como *displays* gráficos, motores, sensores, interfaces de comunicação com/sem fio, conversores, acionadores elétricos, subsistemas de armazenamentos de dados, entre outros.

Além disso, há potencial para a criação de sistemas eletrônicos completos, como sistemas para automação em geral, como horta inteligente ou automação residencial; sistemas de segurança com controle de acesso via digitação de senha, RFID, Biometria ou reconhecimento facial; jogos eletrônicos, com interfaces de *joysticks* especializados ou sensores de movimento; entre outros.

Atualmente, o simulador CompSim vem sendo utilizado, em práticas laboratoriais, como apoio pedagógico à disciplina de OAC de um curso de Bacharelado em Sistemas de Informação, como pode ser visto na Figura 8. Com a recente integração CompSim-Arduino UNO, propôs-se aos estudantes de uma nova turma, como projeto de disciplina, a criação de um novo periférico físico para a plataforma Mandacaru. Serão disponibilizados 10 kits “Arduino Iniciante” e espera-se que, com esse experimento, até o final do semestre letivo, os estudantes tenham compreendido com maior profundidade os conceitos de subsistemas de entrada/saída e tenham sido introduzidos ao projeto de sistemas digitais, empregando-os na criação e aplicação de novos periféricos.



Figura 8 – Aula de AOC em laboratório com suporte do CompSim.

6. Conclusões

Este artigo apresentou uma proposta de integração CompSim-Arduino, na qual uma PCI Arduino UNO torna-se um periférico físico da plataforma virtual Mandacaru para compor um ambiente completo de simulação de sistemas computacionais com hardware e software integrados. Essa proposta busca agregar novos perfis de formação na disciplina de Organização e Arquitetura de Computadores, onde os estudantes podem empregar os conceitos aprendidos em subsistemas de entrada/saída e fundamentos de eletrônica básica para projetar dispositivos físicos reais.

Os trabalhos futuros incluem a adição de novos componentes à plataforma Mandacaru, visando disponibilizar cenários reais e fluxo completo de projetos de sistemas computacionais, a composição de projetos diversificados para o

desenvolvimento de novos sistemas eletrônicos reais e integração com novas tecnologias em dispositivos eletrônicos, como por exemplo, hardware reconfigurável FPGA (*Field Programming Gate Array*).

Agradecimentos

Agradecemos a IFS/PROPEX/Petrobrás e IFCE/FUNCAP por fomentarem esta pesquisa através da concessão de bolsas e apoio financeiro.

Referencias

- Arduino. (2018) "Arduino - Introduction". Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 26 mar. 2018.
- Black, M. (2016) "Export to arduino: a tool to teach processor design on real hardware". *Journal of Computing Sciences in Colleges*, 31(6), pp.21-26.
- Ghaffarian, R. (2016) "Microelectronics packaging technology roadmaps, assembly reliability, and prognostics". *Facta universitatis-series: Electronics and Energetics*, 29(4), pp.543-611.
- Neto, A. M. A., Borges, J. A. dos S. and Silva, G. P. (2017) "Extensão do Simulador SimuS com uso do Protocolo Firmata". XVIII Workshop de Iniciação Científica do XVII Simpósio em Sistemas Computacionais de Alto Desempenho (WIC-WSCAD), pp.123-128.
- Nikolic, B.; Radivojevic, Z.; Djordjevic, J. and Milutinovic, V. (2009) A Survey and Evaluation of Simulators Suitable for Teaching Courses in Computer Architecture and Organization. *IEEE Transactions on Education*, Vol. 52, No. 4.
- Kushner, D. (2011) "The making of arduino". *IEEE Spectrum*, 26.
- Penna, P. H. M. M. and Freitas, H. C. (2013) Análise e Avaliação de Simuladores de Sistemas Completos para o Ensino de Arquitetura de Computadores. *International Journal of Computer Architecture Education (IJCAE)*, v. 2, no. 1, pp 13-16.
- Zorzo, A. F., Nunes, D., Matos, E. S., Steinmacher, I., Leite, J. C., Araujo, R., Correia, R. C. M. and Martins, S. (2017) "Referenciais de Formação para os Cursos de Graduação em Computação". *Sociedade Brasileira de Computação (SBC)*. 153p.