

Desenvolvimento de hardware, software e sistema de controle de robôs educacionais

Development of hardware, software and controlling system of educational robots

RESUMO

OBJETIVO: Através da interação com dispositivos robóticos físicos ou simulados, capacitar estudantes do ensino básico a entender os principais fundamentos de programação. **MÉTODOS:** Desenvolvimento de um ambiente gráfico de programação por fluxogramas e um robô de baixo custo e mecânica prática. **RESULTADOS:** Incentiva-se a participação dos estudantes em eventos e competições de robótica e desperta-se o interesse dos alunos para a área da robótica. **CONCLUSÕES:** O uso da ferramenta desenvolvida no projeto proporciona várias opções para o ensino de robótica nas escolas, atraindo também maior interesse para toda a área tecnológica.

PALAVRAS-CHAVE: Robótica. Educação. Fluxograma.

ABSTRACT

OBJECTIVE: Through the interaction with robotics, enable students to understand the fundamentals of computer programming. **METHODS:** Development of a graphical environment of programming by flowcharts and develop a low cost robot. **RESULTS:** Students are encouraged to participate in robotics competitions and stay more interested in robotics in general. **CONCLUSIONS:** The use of the tools developed in this project provides many options for teaching robotics in schools, thus also attracting greater interest in the whole technological field.

KEYWORDS: Robotics. Education. Flowcharts.

Misael Rosa da Costa
misael@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

João Alberto Fabro
fabro@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Recebido: 19 ago. 2020.

Aprovado: 01 out. 2020.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

Com o avanço da tecnologia e a facilidade de seu acesso pelos jovens nos dias atuais, está sendo difícil aos professores prenderem a atenção de seus alunos, principalmente quando o conteúdo é programação. Visto que esses jovens com um rápido acesso conseguem o que precisam, não sentindo necessidade de aprender programação do jeito tradicional (OLIVEIRA et al, 2012). Nesse cenário, a robótica educacional tem ganhado cada vez mais espaço na área educacional. Pesquisas mostram que esses métodos têm grande aceitação pelos alunos, além de incentivarem o estudo e entendimento de outras disciplinas, como física e matemática (VARGAS et al, 2012). Essas estratégias são eficazes pois possibilitam que o aluno visualize o resultado do que fez, além de ajudar a integrar diferentes áreas de conhecimento e despertar o interesse dos estudantes, visto que os mesmos aprendem mais depressa quando lidam com objetos em vez de fórmulas e abstrações (OLIVEIRA et al, 2012). Sendo assim, a robótica se apresenta como um meio eficiente de apresentar as abstrações necessárias à compreensão da programação (VARGAS et al, 2012).

Esses métodos também facilitam a identificação e correção de erros, pois estes são fisicamente visíveis, ao contrário da programação normal e, além disso ajudam a melhorar a autoestima dos alunos. Todavia, muitas dessas propostas envolvem alunos com alguma noção de programação e/ou utilizam kit de desenvolvimento da empresa Lego, que possui um custo relativamente alto. Por isso se optou por desenvolver um kit de desenvolvimento de baixo custo, usando componentes eletrônicos baratos e usando fluxogramas como programação, considerada uma maneira mais simples de introduzir alunos à programação (MASCHIO; DIRENE, 2007).

Este projeto é desenvolvido no LASER, Laboratório Avançado de Sistemas Embarcados e Robótica, na UTFPR, Campus Curitiba, Sede Centro.

MATERIAL E MÉTODOS

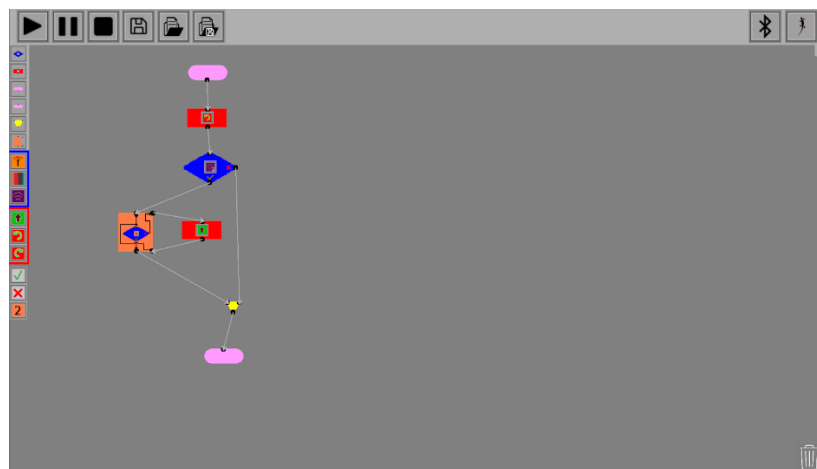
Para o ensino dos fundamentos de programação, foi desenvolvido um ambiente gráfico para criação de fluxogramas. O mesmo é dividido em duas partes: *Fluxprog* e *FluxprogBackEnd*. No *Fluxprog* há blocos que podem ser arrastados pelo programador de modo a formar o programa (Figura 1). Ao final, o fluxograma é interpretado e convertido nos comandos que são enviados e executados pelo robô, são eles: andar um passo para frente, virar à direita e virar à esquerda. Todas as partes do projeto foram desenvolvidas em C++ e a interface gráfica foi desenvolvida com uso da biblioteca *Allegro*.

O *Fluxprog* permite o controle tanto de um robô real, quanto de um robô “simulado” no ambiente V-REP (*Virtual Robotics Experimentation Platform*)(Coppelia Robotics, V-REP), que é um simulador tridimensional realístico de robótica, e uso livre e gratuito para fins educacionais e de pesquisa, no qual foi modelado um robô e um ambiente próximo ao utilizado na realidade. Este ambiente apresenta um robô com duas rodas, 5 sensores de proximidades simulando ultrassom e 5 sensores de cor simulando sensores de fita preta. Esses sensores também podem ser usados na leitura de cor. No caso, a cor verde, já que

consta em uma das regras da OBR (Olimpíada Brasileira de Robótica) que o robô deve virar ao ver algo verde no chão.

O *FluxprogBackEnd* capta os comandos enviados pelo *Fluxprog* e os envia para o robô simulado ou físico. Para comunicação ente o *FuxprogBackEnd* e o *Fluxprog* é usado o compartilhamento de memória, via biblioteca *Boost::SharedMemory*. Cabe ao *Fluxprog* abrir o programa intermediário, selecionando através de uma posição de memória específica se deseja utilizar o robô real ou o simulado. Depois ambos leem e escrevem na memória de forma que o controle do robô físico ou simulado seja feito sem problemas.

Figura 1 – Exemplo de fluxograma



Fonte: Autoria própria.

Com os atuais modos de operação do programa, o usuário possui 3 níveis de controle do robô: no primeiro nível, o usuário apenas usa comandos de movimentação, não lendo os sensores do robô para tomar decisões. Assim, cabe ao *FluxProgBackend* cuidar para que o robô não colida com um obstáculo caso o usuário cometa algum erro na programação. No segundo, o robô segue linha de forma automática, mas não evita colisões. Com isso o usuário deve ler os sensores de ultrassom para tomar decisões em caso de obstáculos. No terceiro nível o robô não possui mais nenhuma “inteligência pré-programada”, cabe ao usuário controlar o robô, fazendo a lógica de seguir linha, entre outros casos, a partir da leitura dos sensores de fita e de cor.

Para o robô real, a plataforma de desenvolvimento usada é o Arduino, sendo que o modelo escolhido foi o MEGA, devido ao grande número de sensores e componentes do robô. A programação foi feita em C++, com base no algoritmo já funcionando no simulador, dessa forma o *FluxProg* envia um dos três comandos já citados, o robô identifica e age de formar a executá-lo. O ambiente físico usado é um banner impresso, contendo um caminho semelhante ao do V-REP, porém há também pistas para treinos para a OBR.

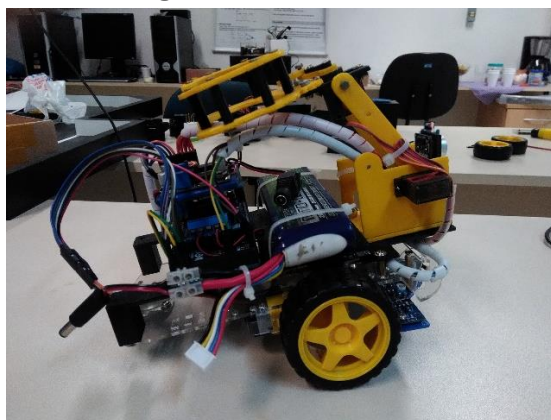
Para fazer a comunicação com o robô físico, utilizou-se o módulo HC-05, que usa tecnologia Bluetooth. Os sensores de fita preta utilizados no robô fazem parte do módulo BFD-1000. Já o sensor ultrassônico escolhido foi o do módulo HC-SR04, que possui distância efetiva de captação dentro do limite de trabalho escolhido para o robô. Na movimentação do robô são usados dois motores DC com tensão

de operação entre 3V e 6V. Para ligar e controlar a rotação dos motores é usado o Motor Shield L293D. No controle da velocidade e direção do robô, são utilizados dois sensores do tipo encoder, módulo MH-Sensor-Series, que simulam um encoder analógico usando infravermelho. Baterias de lítio de alto desempenho estão sendo usadas na alimentação.

A carcaça do robô, juntamente com os dois motores, respectivas rodas e discos encoder, foram adquiridos pelo kit 2WD. Para a confecção das bases dos diversos componentes do robô foi usado modelagem e impressão 3D.

Com o objetivo de deixar os robôs físicos aptos a participar da OBR, incluiu-se sensores de cor ao robô, módulo TC34725. A grande quantidade de sensores foi um problema no início, principalmente devido ao ultrassom, que gera um grande atraso ao sistema já que é dependente do ambiente. Isso foi resolvido usando interrupções e preempções, porém o espaço se tornou pequeno e com isso 2 sensores ultrassons foram retirados do projeto. Além disso, uma garra está sendo desenvolvida para captação de uma bolinha, simulando um resgate, que é um dos objetivos da OBR (Figura 2).

Figura 2 – Robô Físico



Fonte: A autoria própria.

Para avaliação do método proposto, desenvolveu-se um questionário, que é aplicado nos alunos ou professores que utilizam o *FluxProg*. Esse questionário é útil para que se possa fazer melhorias na metodologia, e entender a eficácia da mesma.

A metodologia completa proposta neste projeto ocorre da seguinte maneira: os usuários primeiro aprendem a lógica da programação, usando o *FluxProg*. Após isso, começam os testes básicos no Arduino, como acender um LED. Depois os usuários começam a montar seu próprio robô, e a programação para as atividades da OBR começam.

O FluxProg apenas envia “mensagens” para o robô executar um dos comandos pré-determinados, posteriormente é necessário que os alunos aprendam programar diretamente no Arduino. Para programação do Arduino, inicialmente se usa o *ArduBlock*, uma IDE que permite a programação do Arduino com blocos. Conforme a familiaridade com o microcontrolador aumenta, se passa para a IDE padrão do Arduino, com programação de forma textual.

RESULTADO E DISCUSSÃO

Em razão da pandemia, os encontros nos quais a metodologia é aplicada não foram realizados durante a vigência do projeto abrangida neste artigo. Entretanto, anteriormente houveram três encontros nos quais o projeto foi exposto e a metodologia aplicada.

O primeiro consistiu em apresentar e ensinar o uso do *FluxProg* para alunos do primeiro período dos cursos de Engenharia de Computação e Sistemas de Informação da UTFPR Campus Curitiba. Esta se deu em dois dias, um para cada curso, sendo que ao final da apresentação foi aplicado um questionário. Infelizmente para a primeira turma não foi possível apresentar uma boa usabilidade, já que o *FluxProg* estava travando devido a vazamento de memória, problema que foi corrigido para a segunda turma. Após a aplicação do questionário, percebeu-se que o método, mesmo para alunos da graduação, gera bons resultados, já que incentiva mais ainda a busca pelo aprendizado de programação, além de auxiliar na disciplina e motivar os alunos.

O segundo e terceiro foram encontros com professores da Rede de Ensino Municipal de Curitiba. Estes foram realizados com o objetivo de montar os robôs físicos e fazê-los se movimentar. Foi apresentado o *FluxProg*, o *ArduBlock* e realizado alguns testes com o Arduino, e ao final os robôs estavam se movimentando. O objetivo dos encontros com professores é ensiná-los a metodologia e o desenvolvimento de robôs físicos, de modo que eles possam passar para os seus alunos em sala de aula.

O resultado dessas exposições mostra que a metodologia consegue ensinar programação de uma forma chamativa e interessante, que é o objetivo pretendido, além de atrair também a atenção para montagem e entendimento da parte mecânica e eletrônica do robô.

CONCLUSÃO

Apesar de ainda estar em estado de desenvolvimento, o ambiente de programação está funcional, estando disponibilizado para download no repositório <https://github.com/laser-utfpr/FluxProg/>. Neste link podem ser encontradas instruções de download, compilação e execução do ambiente de programação desenvolvido, bem como instruções de execução do simulador.

O valor final do robô ficou na média de 350 reais. Apesar de parecer um valor relativamente alto para pessoas que não tem um conhecimento prático na área da computação, esse valor está bem abaixo dos kits de desenvolvimento da LEGO, por exemplo. O valor do robô é correspondente apenas aos componentes físicos, já que o software desenvolvido pela equipe é livre.

No mais, é necessário aplicar mais vezes a metodologia para poder tomar decisões sobre mudanças posteriormente necessárias, a fim de gerar melhores resultados.

AGRADECIMENTOS

O presente trabalho foi realizado com o apoio da Fundação Araucária FA - Paraná/Brasil.

REFERÊNCIAS

Coppelia Robotics. **V-REP**. Disponível em: <http://www.coppeliarobotics.com/>. Acesso em: 31 Ago. 2020.

MASCHIO, E.; DIRENE, A. I. Múltiplas representações externas para o ensino de programação de computadores. *In: WORKSHOP DE AMBIENTES DE APOIO À APRENDIZAGEM DE ALGORITMOS E PROGRAMAÇÃO*, 1., 2007, São Paulo. **Anais...** São Paulo: Sociedade Brasileira de Computação, 2007.

OLIVEIRA, D.; FERREIRA, S.; CELESTINO, H.; FERREIRA, S.; ABRANTES, P. Uma proposta de ensino e aprendizagem de programação utilizando a robótica educativa e *storytelling*. *In: CONGRESSO INTERNACIONAL TIC E EDUCAÇÃO*, 2., 2012, Lisboa. **Atas...** Lisboa: Instituto de Educação da Universidade de Lisboa, 2012.

VARGAS, M. N.; MENEZES, A. G. C.; MASSARO C. M.; GONÇALVES T. M. Utilização da robótica educacional como ferramenta lúdica de aprendizagem na engenharia de produção: introdução à produção automatizada. *In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA*, 40., 2012, Belém. **Anais...** Belém: ABENGE, 2012.