

Desenvolvimento de um Ambiente de Programação em Blocos para um Kit de Robótica Educacional

Marcel D. A. Siqueira, Márcio H. Rodrigues, Adriana Herden, Marcos B. R. Vallim

Centro de Experimentação Ninho de Pardais (NIPAR)
Universidade Tecnológica Federal do Paraná (UTFPR) – Campus Cornélio Procópio
Avenida Alberto Carazzai, 1640 – 86.300-000 – Cornélio Procópio – PR – Brasil

marceldaniilo@gmail.com, mvallim@utfpr.edu.br,
herden@utfpr.edu.br, marciohr13@gmail.com

Abstract. *This article presents a programming environment for an educational robotics kit. This kit is located in the final stage of development, and will be used as a tool in the teaching of robotics educational in public high schools. The article shows the methodology of the project contextualized for an application platform, but also discusses technical details of the kit developed, with a focus on software. The environment was developed in Visual Basic .NET platform to work with Windows systems, and does the programming of robots by programmable blocks.*

Resumo. *Este artigo mostra um ambiente de programação para um kit de robótica educacional. Este kit, situado em fase final de desenvolvimento, será utilizado como uma ferramenta didática no ensino de robótica educacional em escolas públicas ensino médio. O artigo mostra a metodologia do projeto contextualizada à plataforma de aplicação, e aborda detalhes técnicos do kit desenvolvido, com foco sobre o software. O ambiente foi desenvolvido na plataforma Visual Basic .NET para funcionamento em sistemas Windows, e realiza a programação de robôs por blocos programáveis.*

1. Introdução

Este artigo apresenta um ambiente de programação utilizado em um kit de robótica educacional. Este kit, em fase final de desenvolvimento, será uma ferramenta de trabalho em oficinas a serem realizadas com alunos de ensino médio como um incentivo ao ingresso nas carreiras de ciência e tecnologia. O projeto que comporta este trabalho é intitulado ‘Ninho de Pardais’, tem apoio da Financiadora de Estudos e Projetos – FINEP, e é formado por alunos de ensino médio e bacharelado, professores de engenharia e uma equipe de apoio psicopedagógico. [Vallim *et al.* 2009].

Em mercado, há desde kits de robótica simples e flexíveis como os kits da Lego® e Fischertechnik Robotics® até outros mais caros e sofisticados, como os kits da empresa Festo® [Enderle 2008]. Difundir o uso desses kits por escolas públicas é um desafio, haja que, como apontam Medeiros Filho e Gonçalves (2008), o elevado custo de aquisição faz com que eles não façam parte da realidade do ensino público brasileiro.

O projeto Ninho de Pardais tem como uma de suas metas físicas o desenvolvimento de um kit próprio para viabilizar a robótica em escolas públicas. O kit é projetado por alunos de engenharia elétrica e mecânica, de informática e de ensino

médio técnico, todos orientados por uma equipe de professores das áreas de educação e engenharia. O trabalho engloba desde o desenho e fabricação de peças mecânicas até o desenvolvimento de metodologias de ensino de robótica a alunos de ensino médio.

Neste artigo será apresentado o ambiente de programação desenvolvido para o kit Ninho de Pardais, cuja versão de testes está disponível em <<http://bit.ly/uxccRq>>. Ele aborda do conceito de programação em blocos, e para a sua realidade de uso (usuários sem conhecimento técnico de linguagens de programação), essa é uma forma amigável para programar um robô, fato esse é comprovado pela adoção do mesmo conceito por marcas consagradas de kits de robótica educacional. Serão apresentados o contexto de inserção do ambiente de programação, detalhes metodológicos e construtivos e os resultados obtidos.

2. Plataforma de Aplicação

A plataforma de robótica é definida por um conjunto de peças mecânicas (módulo mecânico), um conjunto de hardware (controlador lógico, sensores e motores), e um software para programação do robô (módulo de software). O *know-how* necessário ao desenvolvimento foi buscado com estudos e análises formais dos kits comerciais existentes em mercado, como kits Lego®, Fischertechnik Robotics® e PNCA® (nacional) [Sevinhago, Herden e Vallim 2009].

O módulo de hardware adotou uma ideia diferente dos kits comerciais: Prototipação em lógica reconfigurável. Isto foi adotado tanto para o estudo do próprio conceito quanto para a definição de um microcontrolador enxuto [Siqueira e Vallim 2010]. A topologia é centralizada em um processador virtual Nios II, da empresa Altera, e estudado em uma placa de desenvolvimento Altera DE2. A estrutura é definida pela etapa de comunicação (USB-Blaster), processamento (Nios II) e módulo de hardware I/O (controle de motores e sensores). A figura 1 reúne essa topologia.

O módulo mecânico é subdividido pelo critério de contribuição estrutural e tipo de encaixe. Elementos estruturais são responsáveis pela sustentação e forma do robô, e podem possuir encaixes dos tipos básico, bidirecional ou tridimensional. Já os elementos de ligação são responsáveis por realizar o encaixe entre dois ou mais elementos. E, por fim, os elementos motores são responsáveis por promover o movimento do robô [Moscato e Vallim 2010]. A figura 2 apresenta as peças principais mecânicas que compõem o kit, e que atualmente estão em processo de fabricação.

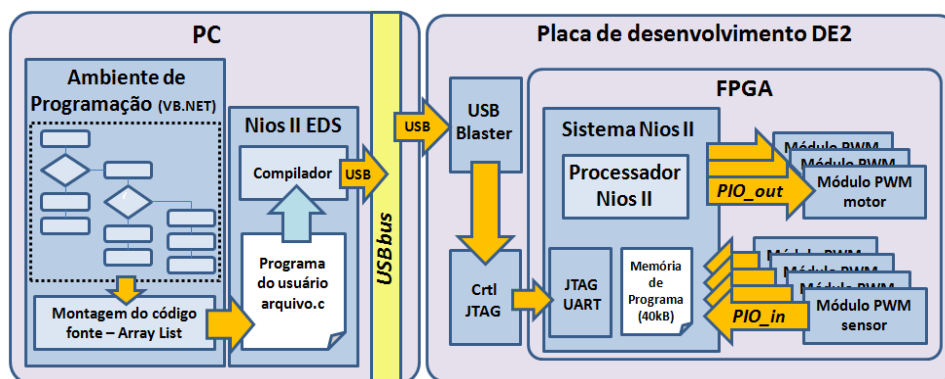


Figura 1. Estrutura de hardware com ligação ao ambiente de programação

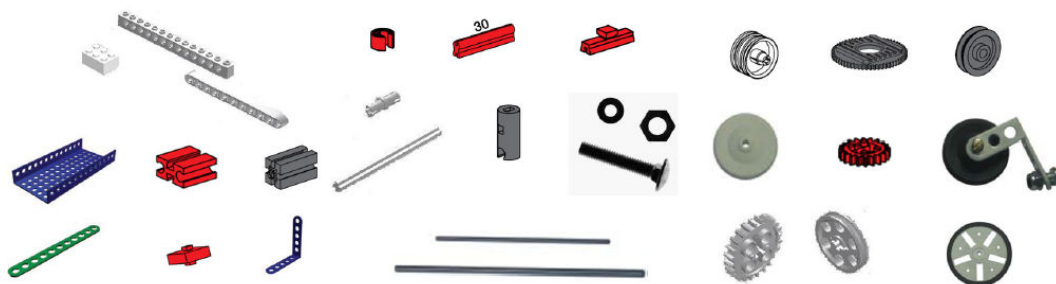


Figura 2. Elementos mecânicos desenvolvidos para o kit de robótica Ninho de Pardais

3. O Ambiente de Programação

O ambiente de programação foi projetado sob o paradigma de orientação a objetos (OO), e é responsável por capturar as informações inseridas pelo usuário, gerando um arquivo de saída (de extensão .c) como código fonte do controlador do robô. Uma vez definido este arquivo, o ambiente de programação interage com um segundo, dedicado exclusivamente à programação do sistema embarcado.

Os componentes de computação estão relacionados ao software do ambiente de programação de ações de um robô móvel. Primeiramente, o usuário se responsabiliza por programar os eventos do robô (seguindo as diretrizes de um desafio proposto) por meio da interface do software, e então transmite este conjunto de instruções para o controlador lógico do robô via *Universal Serial Bus* (USB).

Com o amadurecimento da pesquisa optou-se pela utilização da classe *ArrayList* do VB.NET na programação do software, com a geração de um código fonte compatível com o processador Nios II. O objetivo foi permitir ao processador executar um arquivo de programação gerado por um arquivo de formato '.c', que contém a sequência de instruções informadas pelo usuário [Coimbra, Machado e Herden 2010].

3.1. Programação em Blocos

Por meio de blocos gráficos, o ambiente de programação oferece ao usuário comandos básicos, como acionamento de motores, leitura de sensores, declaração e controle de variáveis e estruturas de controle. Esse modo de programação foi inspirado na linguagem NXT-G, do kit Lego Mindstorms®, em que “ao invés de se escrever um programa em texto, trabalha-se em blocos que representam funções independentes de instruções de programa. Esses blocos podem ser arrastados e dispostos conforme uma lógica desejada, seguindo a mesma ordem para execução” [Bishop 2008].

4. Resultados

Os resultados alcançados durante o trabalho incluem o desenvolvimento do ambiente de programação em si, dos blocos de programação e a comunicação com o ambiente de programação, isto é, realizar a integração entre as diversas etapas da programação de um robô, como ilustrou a figura 1. Nesta seção são mostrados apenas o ambiente de programação e um bloco funcional como exemplo.

A visão geral do ambiente de programação está disposta na figura 3. O usuário escolhe qual bloco irá alocar na sequência lógica do código, abre-o e o parametriza. Ao clicar em enviar, um trecho de código 'c' correspondente é acrescentado à *ArrayList*.

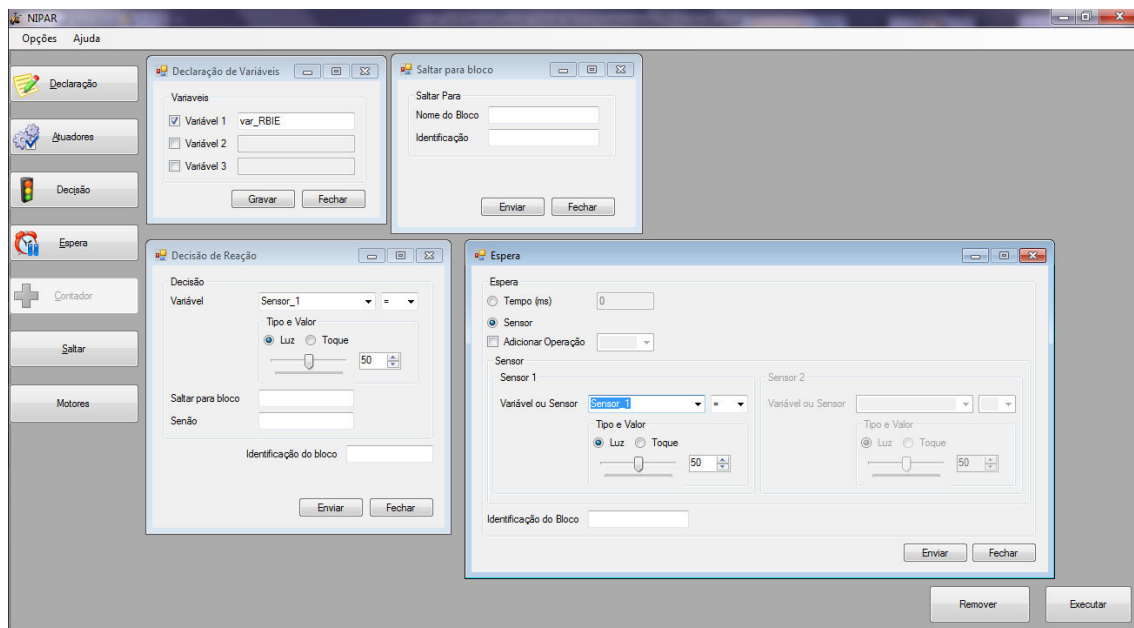
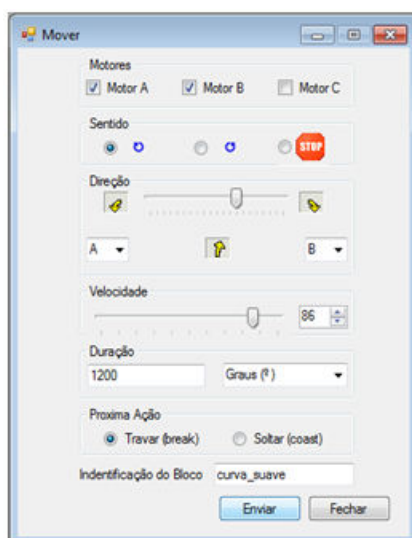


Figura 3. Tela do ambiente de programação com alguns blocos desenvolvidos

3.1. Blocos de Programação

Baseando-se nos kits de robótica experimentados, foram concebidos os blocos fundamentais para a programação, com funções de intituladas por: 'Declaração', 'Mover', 'Decisão', 'Espera', 'Contador'; Os blocos são genéricos, e permitem a resolução de desafios consideráveis de robótica. Não foram utilizadas estruturas prontas como 'seguir-faixa', comum em alguns kits comerciais. O propósito é que os alunos-usuários sejam instigados ao desenvolvimento de suas próprias funções. Atualmente, o projeto do software está focado na criação de novos dos blocos de programação. O bloco 'Mover' aparece na figura 4 como exemplo de um bloco desenvolvido: Na parte **a**, com o aspecto que é apresentado ao usuário, na parte **b**, seu trecho código '.c' gerado para a *ArrayList*, e na parte **c**, o comportamento do robô para essa parametrização.

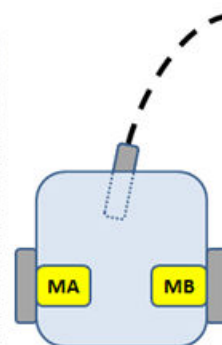


(a)

```
curva_suave:
motor_a = 86;
motor_b = 17;
envia();
angulo_a(1200);
while(ang_a_flag == 0){
recebe_encoders();
}

motor_a_block = 1;
motor_a = 0;
motor_b_block = 1;
motor_b = 0;
envia();
```

(b)



(c)

Figura 4. Exemplificação com o bloco 'Mover'. Parte (a): Parametrização do usuário; Parte (b): Trecho de código '.c' gerado pela parametrização; Parte (c): Reação do robô

8. Conclusão e Discussão

Este artigo apresentou um ambiente de programação e sua aplicação em um kit de robótica. O software foi desenvolvido em paralelo com as demais áreas de conhecimento, que dentro do projeto do kit tratam de todas as tarefas necessárias para a construção do kit. O projeto visa levar a robótica educacional à rede pública de ensino, e isso requer do kit baixo custo tanto de materiais quanto de desenvolvimento.

Os próximos passos do projeto são em rumo à implementação de novos blocos de programação, ao passo que o projeto de hardware se estende e ganha novas funcionalidades. Além disso, há a necessidade de criação de uma área de trabalho no ambiente de programação com a construção dos blocos parametrizados ao passo que estes são modificados. Após isso, serão realizadas oficinas piloto para teste do kit com alunos de ensino médio da própria universidade e da comunidade externa.

9. Agradecimentos

Este trabalho é financiado pela Financiadora de Estudos e Projetos (FINEP) (REF 4971/2006) e conta com o apoio da Fundação de Apoio à Educação, Pesquisa e Desenvolvimento Científico e Tecnológico da Universidade Tecnológica Federal do Paraná (FUNTEF-PR), da Fundação Araucária e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq Brasil).

Referências

- Bishop, O. (2008). *Programming Lego Mindstorms NXT*, Syngress, 1ª edição.
- Coimbra, V. A., Machado, P. H. A. e Herden, A. (2010). “Ambiente De Programação Para Kit De Robótica Educacional: Um Recurso Por ArrayList”. In: *Anais do 13º Encontro de Atividades Científicas da UNOPAR – XIII EAC UNOPAR*. Londrina.
- Enderle, S. (2008). “The qfix Robot Kits”. In: *Proceedings of 1st International Conference on Research and Education in Robotics*, Heidelberg. Springer.
- Medeiros Filho, D. A. e Gonçalves, P. C. (2008). “Robótica educacional de baixo custo: uma realidade para as escolas brasileiras”. In: *Anais do Workshop sobre Informática na Escola – WIE’2008*. Belém.
- Moscato, A. L. S. e Vallim, M. B. R. (2010). “Desenvolvimento Do Módulo De Elementos Mecânicos Para Um Kit De Robótica Educacional”. In: *Anais do 13º Encontro de Atividades Científicas da UNOPAR – XIII EAC UNOPAR*. Londrina.
- Sevinhago, R., Herden, A. e Vallim, M. B. R. (2009). “Análise de Usabilidade do Software de Programação de Três Kits Comerciais de Robótica Educacional”. In: *Anais do 1º Encontro Nacional de Informática e Educação – ENINED*. Cascavel.
- Siqueira, M. D. A. e Vallim, M. B. R. (2010). “Implementação de um Módulo de Controle Aplicado à Robótica Educacional Baseado em Hardware Reconfigurável”. In: *Anais do 24º Congresso Regional de Iniciação Científica e Tecnológica em Engenharia – XXIV CRICTE*. Rio Grande.
- Vallim, M. B. R., Herden, A., Gallo, R., Cardoso, L. R. e Bitencourt, L. C. (2009). “Incentivando Carreiras na Área Tecnológica Através da Robótica Educacional”. In: *Anais do 37º Congresso Brasileiro de Ensino de Engenharia – COBENGE*. Recife.