

AVALIAÇÃO DE AMBIENTES DE DESENVOLVIMENTO PARA AUTOMAÇÃO DO PROBLEMA DO CUBO MÁGICO PARA O ROBÔ LEGO MINDSTORMS NXT

Walter SIMÕES (1); Eliane COLLINS (2); Wanderlan ALBUQUERQUE (3), Rafael PADILLA (4), Rubens VALENTE (5), Vicente LUCENA Jr. (6)

- (1) UFAM – FT- PPGE – Universidade Federal do Amazonas-Faculdade de Tecnologia (FT)- Programa de Pós-Graduação em Engenharia Elétrica, Campus da universidade Federal do Amazonas – Setor Norte- Faculdade de Tecnologia- Av. Gal Rodrigo Octávio Jordão Ramos nº 3000. CEP 69077-000 – Bairro Aleixo – Manaus - AM - Brasil,
(2) UNINORTE – Centro Universitário do Norte, Rua Dez de Julho, nº 873. CEP 69010-060 – Centro, Manaus - AM - Brasil
(3) INDT– Instituto Nokia de Tecnologia (INdT), Rodovia Torquato Tapajós nº 7200. CEP69048-660 – Bairro Colônia Terra Nova, Manaus - AM – Brasil,

(1) e-mail: UFAM-FT-PPGE, UNINORTE, e-mail: waltersimoes@gmail.com

(2) UFAM-FT-PPGE, INDT, e-mail: elianecollins@gmail.com,

(3) UFAM-FT-PPGE, UNINORTE, e-mail: wcacarvalho@gmail.com

(4) UFAM-FT-PPGE e-mail: eng.rafaelpadill@gmail.com

(5) UFAM-FT-PPGE, e-mail: rubens.analista@gmail.com

(6) UFAM, CETELI E CEFET-AM, Centro Federal de Educação Tecnológica do Amazonas, e-mail: vicente@ufam.edu.br

RESUMO

Este trabalho apresenta os resultados de um experimento onde duas plataformas de desenvolvimento de aplicações para o robô LEGO MINDSTORMS NXT foram testadas para a escolha do ambiente adequado a equipe para implementação da automação do problema do Cubo Mágico. O objetivo deste experimento foi avaliar o melhor ambiente de desenvolvimento para o cenário de automação e avaliar os critérios escolhidos pela equipe para tornar possível a solução do problema. A metodologia aplicada foi a realização de testes de hardware e software na plataforma do robô LEGO MINDSTORMS NXT utilizando as linguagens de programação NXC e Java, sob a dependência do laboratório da Universidade Federal do Amazonas, bem como pesquisas de algoritmos que validassem a solução do problema. A principal contribuição deste trabalho é apresentar os resultados da avaliação de duas plataformas por uma equipe sem experiência nas práticas de programação dos ambientes que podem ser utilizadas na pesquisas de estudantes para simular diversos problemas enfrentados pela automação com utilização de software embarcado.

Palavras-chave: Automação, sistemas embarcado, robô LEGO, Java.

1 INTRODUÇÃO

A exigência de mercado de produzir cada vez mais produtos, em menos tempo e com maior qualidade faz com que a automação de processos industriais seja uma área em constante evolução. Neste contexto, a agregação de diferentes tecnologias é amplamente aplicada, desde a utilização de protocolos de rede, sistemas operacionais, interferência eletro-magnética, arquiteturas de computadores, software embarcados, onde a maioria das vezes os requisitos de tempo real devem ser cumpridos.

As atividades práticas dos conceitos de automação industrial quando utilizadas pelos estudantes contribuem para pesquisa e desenvolvimento de novas idéias e aplicações. A queda do custo de aquisição de microcontroladores e microprocessadores proporcionam a criação de kits de desenvolvimento que podem ser incorporados em projetos acadêmicos ou industriais. (MAIA *et. Al*, 2008).

Várias atividades práticas podem ser realizadas, porém uma das que mais atrai estudantes é a utilização do kit de desenvolvimento MINDSTORMS da LEGO para construção e programação de robôs que possam realizar alguma atividade humana. O LEGO MINDSTORMS NXT é uma linha do brinquedo LEGO, lançada comercialmente em 2006, voltada para a Educação tecnológica (MINDSTORMS, 2010).

Com o kit, é possível construir os robôs e programá-los utilizando as linguagens de programação criadas pelo Grupo LEGO RCX Code e ROBOLAB, ou utilizando as linguagens de programação que não foram desenvolvidas pela LEGO: Java, NQC, NXC, C++, pbFORTH, Visual Basic e XS (MINDSTORMS, 2010). A LEGO também oferece um sistema de desenvolvimento que permite a interação com qualquer linguagem de programação que seja compatível com a tecnologia activex.

O Kit possui uma CPU (Bricx) contendo um microcontrolador de 8 bits Renesas H8/300. Uma memória de 32K que é utilizada para armazenar o firmware e os programas. Esse kit vem sendo amplamente utilizado por universidades em todo mundo, onde até concursos são realizados e os projetos mais criativos são premiados.

O principal objetivo deste trabalho é fazer uma análise em diferentes plataformas de desenvolvimento utilizadas no kit MINDSTORMS para solucionar o problema do Cubo Mágico. Na seção 2 descrevemos o problema do cubo mágico, projeto de automação o qual está em desenvolvimento. Na seção 3 descrevemos a arquitetura do robô usada para a solução do problema. Na seção 4 mostraremos as plataformas utilizadas para teste, o conceito da plataforma NXC no ambiente Bricx, a plataforma LEGO NXJ e os testes realizados. Na seção 5 descrevemos os resultados obtidos com os testes realizados e na seção 6 nossa conclusão da pesquisa e lições aprendidas.

2 PROBLEMA DO CUBO MÁGICO

O cubo mágico (Rubik's Cube), considerado como um quebra-cabeça em 3D, foi criado por Ernő Rubik em 1974 e passou a ser comercializado em 1980. Até o Janeiro de 2009, foram vendidos mais de 350 milhões de cubos mágicos mundialmente, sendo considerado o quebra-cabeça mais vendido no mundo (ADAMS, 2009).

Existem diversas variações e modelos de cubos mágicos, 4x4x4, 5x5x5, 6x6x6, 7x7x7 e o mais popular, o cubo clássico 3x3x3. O cubo clássico, cada uma das 6 faces é coberta por 9 adesivos coloridos com uma das cores a seguir: branca, azul, laranja, vermelha, amarela e verde.

Diversos algoritmos foram desenvolvidos para a resolução do cubo mágico. Os algoritmos são aplicados à resolução de pequenas partes do cubo, como as bordas, por exemplo. Outros algoritmos resolvem cada face de cada vez, porém causam a desorganização das outras faces. Algoritmos um pouco mais complexos podem resolver partes do cubo, evitando o embaralhamento de algumas partes.

Algoritmos foram criados ao longo dos anos propondo uma solução para o cubo mágico. David Singmaster publicou o livro "Notes on Rubik's Magic Cube" em 1981 propondo a solução de cada camada por vez, tendo o problema resolvido após as 6 camadas estarem resolvidas. Singmaster propôs que a primeira camada (face do cubo) fosse resolvida primeiro, tendo a camada central sendo resolvida no segundo passo e, finalmente, a terceira camada (outra face do cubo) resolvida.

Resolver o cubo mágico com menos movimentos possíveis e com a utilização de poucos algoritmos e sequências de movimentos é ainda uma questão aberta. Gene Cooperman e Daniel Kunkle (KUNCLE, D.; COOPERMAN, C., 2007) afirmaram 26 movimentos é o máximo de movimentos necessários para se

conseguir resolver o problema do cubo mágico. Thomas Rokicki afirmou que praticamente não existem arranjos que necessitem de 26 movimentos. (TOM ROKICKI, 2008).

Philip Marshall (MARSHAL, 2005) propôs uma solução que implica a utilização de apenas 2 algoritmos, levando a média de 65 movimentos para resolução. Vários métodos podem ser utilizados para resolução do problema do cubo mágico 3x3. Este trabalho visa realizar testes de plataformas diferentes para automatizar este processo utilizando o robô MINDSTORMS.



Figura 1-Projeto do Robô¹

3 PROJETO ROBÔ LEGO MINDSTORMS NXT

De acordo com pesquisas entre vários modelos de montagem para o robô LEGO, escolhemos o projeto de montagem do robô que poderia demandar menos tempo e menos complexidade de montagem das peças. O robô foi projetado conforme a Figura 2.

Sua estrutura conta com um sensor de luz para ler as faces do cubo, um sensor ultra-sônico para detectar a presença da distância do cubo à mão robótica, uma cesta giratória que permite girar a lateral do cubo para ler outras peças do mesmo, uma mão robótica para girar as faces do cubo e uma unidade de processamento que permite embarcar e executar a aplicação e ainda, rodas e atuadores, ou também chamados motores, que acionam o movimento de cada ação no robô. A equipe levou cerca de uma semana para fazer a montagem completa do robô.

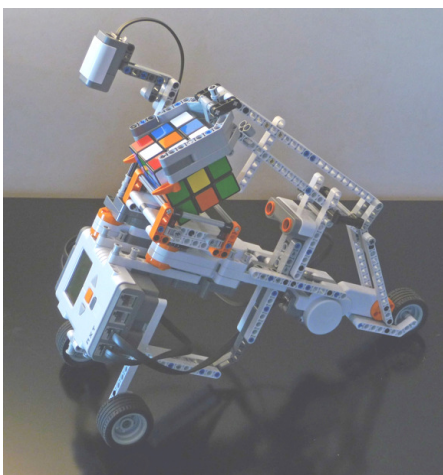


Figura 2-Projeto do Robô²

¹ Disponível em: <<http://www.puzzlingqueen.com>> Acesso em 21 Jul. 2010.

² Disponível em: <<http://www.titledtwister.com>> Acesso em 15 Jul. 2010.

4 PLATAFORMAS DE DESENVOLVIMENTO

Pelo fato do algoritmo de resolução do cubo mágico ser complexo, precisaríamos de um ambiente de desenvolvimento que nos proporcionasse recursos bons na edição de código, identificação rápida dos comandos dos componentes do robô, facilidade para a compilação de código e para embarcar a aplicação no robô. Seguindo esses critérios, de acordo com várias pesquisas a respeito das plataformas que deveríamos utilizar para a implementação do projeto do robô, consideramos duas principais: Plataforma NXC no ambiente de desenvolvimento Bricx e Plataforma LEGO NXJ.

Apresentaremos a seguir, os testes para o estudo de caso da resolução do problema do cubo mágico em um sistema embarcado nas duas plataformas, a partir destes foi feita a escolha da plataforma que mais se adequou às necessidades da equipe e do projeto.

4.1 Experimento com a Plataforma NXC e Bricx para desenvolvimento

NXC (*Not eXactly C*) é uma linguagem específica para a programação do LEGO MINDSTORMS NXT (MINDSTORMS, 2010). O NXT possui um interpretador de bytecodes que é usado para executar programas (LUDI, 2008). O NXC compila o código-fonte traduzindo para NXT bytecodes. A estrutura do NXC é similar a linguagem C, porém não é uma linguagem para propósito geral, possui restrições e limitações do interpretador NXT (HANSEN, 2010). A linguagem é suportada em todas as versões do firmware NXT padrão e é conhecida por ser de rápida execução e mais próxima ao hardware.

O ambiente que é usado para a linguagem NXC é o Bricx Command Center, que é open source e suporta variações do LEGO MINDSTORMS NXT (NOGA, 2010). Em linhas gerais é um ambiente com recursos simples de edição de código e de fácil instalação, pois é compatível com várias versões do windows e faz conexão com o robô LEGO MINDSTORMS.

Para avaliação da plataforma com NXC e Bricx, o primeiro passo foi nos certificar que o robô LEGO MINDSTORMS estava com um firmware NXT padrão compatível com a linguagem. Para este caso, nós atualizamos o firmware para a nova versão LEGO NXT 1.29, que está disponível no site do fabricante, para que suportasse a linguagem NXC tornando possível embarcar o software no dispositivo.

A instalação do ambiente Bricx no sistema operacional Windows Vista ocorreu sem problemas e sem necessidade de outras instalações adicionais, já que o sistema já possuía o drive da USB para conectar no robô. Com o ambiente instalado, iniciamos o estudo do ambiente seguindo os seguintes passos:

- Conectar o robô com o computador via USB;
- Inicializar o ambiente Bricx, a primeira tela pede que identifique a porta e o firmware que está sendo utilizado, no caso NXT standard;
- Se não houver problemas na conexão via USB a tela principal com o editor de código do ambiente Bricx será inicializada;
- No botão File no menu superior, clicamos em Open para abrir o código do cubo mágico na versão NXC;
- O próximo passo foi compilar o código, para isso o Bricx possui uma barra de atalho com o ícone Compile;
- Caso não haja erros de compilação, o código compilado já é enviado automaticamente para o robô via USB, as alterações no código são embarcadas sem necessidade de reconectar o robô;
- Para executar pelo ambiente, basta ir na barra de atalhos no ícone Run;
- O robô executa o código embarcado e para parar a execução o ambiente possui o ícone Stop.

Em dois dias de avaliação da plataforma, testamos primeiramente o funcionamento dos sensores. O sensor de luz (*lightsensor*) que está acoplado em um motor de rotação. Este sensor deveria realizar o movimento de leitura na peça central da face do cubo virada para cima e depois se inclinar ligeiramente para a lateral para que a cesta gire e assim o leitor identifique as outras peças.

Os testes aqui realizados envolveram a manipulação de três partes móveis do robô: A cesta, o braço que tem o sensor de luz acoplado e o leitor das faces do cubo. Com isso, deveria ser feita a salvaguarda da leitura das 6 faces. A partir daí, os recursos estariam preparados para a atuação do algoritmo de resolução do cubo.

Para isso, edições no código foram feitas para testarmos isoladamente os comandos da linguagem, primeiro fizemos o teste para acionar os motores do robô, em seguida inicializamos atribuindo valores para as variáveis de posição dos componentes do robô e por fim enviamos o comando para a leitura dos sensores para a identificação das cores do cubo. Como se tratava da avaliação da plataforma não foi testado o algoritmo lógico de resolução do cubo, visto que a complexidade deste na linguagem NXC dificultaram os testes.

4.2 Experimento com a Plataformas Lejos NXJ

A plataforma Lejos NXJ é um ambiente de programação que permite o uso da linguagem Java no Robô LEGO MINDSTORMS NXT (LEJOS,2010). Sua integração com ambiente de programação pode ser através de um ambiente de desenvolvimento como o Netbeans e o Eclipse que suportam várias plataformas de sistemas operacionais como: Windows e Linux. Para a integração com o robô, o *firmware* NXT inclui uma máquina virtual java possibilitando embarcar e executar as aplicações. A comunicação com o Robô e outros dispositivos pode ser realizada através de Bluetooth e via USB.

Para este estudo de caso foi realizado testes nos sistemas operacionais Windows XP e Windows Server, mas funcionou com êxito através do Windows XP, devido a este já possuir o driver de comunicação USB de 32 bits compatível com o *firmware* do Robô MINDSTORMS NXT. O ambiente de desenvolvimento utilizado foi o Eclipse, pois foi de fácil instalação de plugins para o *firmware* do Lejos. Para executar a aplicação foi feita a seguinte sequência de passos:

- Primeiramente configuramos o ambiente via Ms-Dos, executando na pasta do Lejos NXJ\bin o seguinte comando: nxjflash.bat;
- Depois de reconhecer o *firmware* no robô, na opção Project Explorer do eclipse clicou-se com botão direito sobre a aplicação com nome Teste.java e escolheu a opção *Convert to Lejos NXJ projet* e em seguida executamos a opção *Upload Program to The NXT Brick* para embarcar o código e executar a aplicação. Porém a opção de *Upload* não funcionou;
- A opção que executou sem erros e embarcou a aplicação foi via MS- DOS através da sequência de comandos: nxjc Teste.java para compilar a aplicação e para embarcar o comando usado é: nxj Teste.

Uma vez que o código era embarcado, tínhamos que refazer os passos descritos a cada alteração de código, pois ele não era compilado e embarcado automaticamente como no Brick. Os testes realizados nesta plataforma foram os mesmos descritos na seção 4.1, onde avaliamos o nível de complexidade dos comandos em Java para acionar os componentes do robô como motores e sensores separadamente.

5 RESULTADOS OBTIDOS

A equipe avaliou as plataformas dedicando dois dias para testar cada uma, com isso conseguimos, em linhas gerais a seguinte análise de resultados:

1. Na plataforma NXJ, que utiliza linguagem Java, por ser uma linguagem popularizada e de alto nível, a equipe melhor se familiarizou e o ambiente eclipse é mais estável e facilitou a edição do código, foi mais intuitiva a identificação das funções de movimento do sensor e dos parâmetros de localização, onde conseguimos em pouco tempo o movimento ideal de inicialização do robô e a leitura das cores do cubo.

As informações das cores foram armazenadas e nos foi possível testar o movimento da mão robótica de “ir para frente” e ir “para trás”. Porém, podemos destacar que tanto a linguagem quanto o ambiente possuem um bom suporte e várias literaturas de apoio, mas frizamos que o processo de configuração do ambiente, compilação do código e execução do programa embarcado é mais trabalhoso e demandou muito tempo nos testes.

2. Na plataforma NXC e Brick, destacamos que pelo fato da equipe não ser familiarizada com a linguagem C, as funções do NXC foram menos intuitivas, porém o ambiente brick proporciona um

bom Help e é mais fácil de embarcar o software no dispositivo pois é feito de maneira automática, os comandos NXC são bem explicados na documentação e exemplificados, porém o ambiente possui alguns erros e instabilidade, com isso demanda tempo para aprendizado do ambiente que ainda assim possui uma interface intuitiva para desenvolvedores.

Com esse ambiente, conseguimos com êxito executar a movimentação dos motores do sensor, a movimentação da bandeja e do braço, porém tivemos dificuldade apenas em ajustar os parâmetros de localização, isso devido à dificuldade inicial de entender a linguagem NXC, mas essa dificuldade foi contornada com a experiência e uso do ambiente.

3. Em relação à questão de desempenho, o algoritmo testado nas plataformas analisadas foi criado por Hans Anderson (TILTEDTWISTER,2010) e adaptado para as linguagens NXC e Java, o código nas duas linguagens está disponível em seu site, assim como o vídeo de sua execução.

Na plataforma NXC obtivemos um resultado melhor, conseguindo executar todo o algoritmo de resolução do cubo com média de 60 movimentos do braço robótico e bandeja giratória. Enquanto que na linguagem Java o algoritmo precisava ser revisado várias vezes e não conseguimos executar toda solução do algoritmo devido à falta de métodos nativos que existem no ambiente NXC, mas que no ambiente Java precisavam ser construídos, demandando muito tempo.

Com base nos resultados do experimento em cada plataforma, não identificamos problemas de tempo de execução das aplicações embarcadas no robô, a plataforma java não apresentou demora em relação ao NXC. Durante o experimento, a linguagem java (lejos) realizou os controles dos sensores e motores (atuadores), porém não possui todos os métodos nativos como na linguagem NXC, programada em C que consegue controlar mais de dispositivo numa mesma thread. A primeira versão de código foi desenvolvido em java, mas não obtivemos sucesso na montagem do cubo, então substituímos pelo código em NXC.

6 CONCLUSÃO

Neste artigo apresentamos a avaliação de duas plataformas de desenvolvimento para implementar o problema da solução do cubo mágico utilizando o robô LEGO MINDSTORMS NXT. A partir dos resultados dos testes realizados, temos como lições aprendidas, que é importante conhecermos as principais plataformas disponíveis de desenvolvimento, realizar testes e pesquisas de literaturas de suporte para analisarmos a que melhor nos cabe para a resolução do problema e aquisição de conhecimento.

Contudo destacamos que os pontos como a familiaridade com a linguagem, facilidades do ambiente de desenvolvimento, tempo de entrega do projeto, desempenho e material de suporte são fatores fundamentais na escolha de uma plataforma de desenvolvimento para avanços na solução de um problema complexo, pois com isso o avanço na resolução pode ser mais rápido e efetivo mesmo que o tempo com a configuração e instalação de ferramentas seja maior.

Com isso, a automação da solução do cubo mágico foi um experimento complexo e satisfatório que exigiu muito tempo para a análise e teste dos algoritmos, calibragem dos motores (atuadores) e sensores. O resultado positivo da solução do robô se deu em virtude dos testes, a análise da escolha da melhor plataforma, a adaptação e o desempenho do algoritmo.

REFERÊNCIAS

ADAMS, L. W. **The Rubik's Cube: A Puzzling Success**. Revista TIME. Publicada em 28 fev 2009.

HANSEN JOHN: **NXC Programmer guide**. Disponível em:

<<http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/index.html>> Acesso em: 14 jul 2010.

HIROTA, E.H. **Desenvolvimento de competências para a introdução de inovações gerenciais na construção através da aprendizagem na ação**. 2001. 205p. Tese (Doutorado em Engenharia) - Escola de Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2001.

KUNKLE, D., COOPERMAN, C. **Twenty-six moves suffice for Rubik's Cube**, 2007.

LEJOS. **Java for LEGO MINDSTORMS**. Disponível em:
<<http://lejos.sourceforge.net/nxt/nxj/tutorial/Preliminaries/Intro.htm>> Acesso em: 15 jul 2010.

LUDI, A. S., RECHLMAYR, T. **Developing inclusive outreach activities for students with visual impairments**, Technical Symposium on Computer Science Education archive. Proceedings of the 39th SIGCSE technical symposium on Computer science education, 2008.

MAIA, D., L., SILVA, J. V., ROSA, E. R., LUCENA, V., **A robótica como ambiente de programação utilizando o kit LEGO MINDSTORMS**, XVIII SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2008.

MARSHAL, P. **The Ultimate Solution to Rubik's Cube**, 2005

MINDSTORMS. **What is NXT?** Disponível em:
<<http://MINDSTORMS.LEGO.com>> Acesso em: 15 jul 2010.

NOGA, M. L. **Bricxos™ at source-forge**. Disponível em: <<http://Bricxos.sourceforge.net>> Acesso em: 10 jul 2010.

ROKICKI, T. **Twenty-five Moves Suffice for Rubik's Cube**, 2008.

TILTEDTWISTER. **LEGO MINDSTORMS Rubik's Cube Solver**. Disponível em:
<<http://tiltedtwister.com>> Acesso em: 15 jul 2010.