
CONCEPÇÃO DE UMA INTERFACE PARA PROGRAMAÇÃO DE ROBÔS MÓVEIS INTELIGENTES

Felipe Antunes^{*, †}

felipe@xbot.com.br

Anderson Dias[†]

anderson@xbot.com.br

Yan Freitas^{†, ‡}

yan@xbot.com.br

Antônio V. Netto^{†, ‡}

valerio@xbot.com.br

^{*}Departamento de Física, Universidade Federal de São Carlos (UFSCar)
Washington Luis, Km 235 - Monjolinho, São Carlos - SP

[†]XBot

Rua Monteiro Lobato, 2479 - Jardim Brasil, São Carlos - SP

[‡]Universidade Central Paulista (Unicep)

Rua Miguel Petroni, 5111 – Jardim Centenário, São Carlos - SP

Abstract This paper describes the Software Curumim Interface program that it controls a robot via block's programming. This program makes easier for people, that, don't have enough knowledge about the program languages, even those, they can determinate some action to the robot. The program has a menu that contain significative icons that sugest the comands. In order to write the program, it is only necessary ti clic the icon in the menu and the block will be insert in the program's window. In this way a block's sequency will be create, that the Curumim will execute. The interface makes enable to the user change any parameter of the block, delete it, insert a new block beetwen other already in the program. For the experienced user, exist a possibility to use the C/C++ languages to program the robot.

Keywords education software, robotic education, mobile robot, education tecnology, robotics interface program.

Resumo Este artigo descreve o software de controle do Curumim, o qual comanda o robô via programação em blocos. O software facilita a introdução à lógica de programação, pois, mesmo aqueles que nunca tiveram contato com o conceito, conseguem operar o robô para uma determinada ação através das instruções em blocos. Os blocos são separados por ícones intuitivos, cada um conjunto de instruções, que são inseridos linearmente através de alguns cliques, escrevendo um algoritmo para o controle do Curumim. A interface permite que o usuário altere os parâmetros destas funções ou insira um novo bloco ao longo do programa. Como diferencial, para usuários mais avançados, a interface traduz automaticamente a linguagem em blocos em um algoritmo em C/C++ com todas as instruções de controle do robô.

Palavras Chave: software educacional, robótica educacional, robôs móveis, educação tecnológica, programação em blocos.

1 INTRODUÇÃO

Com o intuito de promover o desenvolvimento educacional e o aprendizado de conceitos básicos de lógica por intermédio de um robô, foi elaborado o kit Curumim. Esse kit, ilustrado na Figura 1, inclui o robô Curumim, um par de baterias com carregador, o rádio-base para a comunicação entre o robô e o computador com cabo USB, receptor para os sinais da câmera do robô e seus cabos RCA, além de um CD com o software do Curumim junto com seu manual seu manual.



Figura 1 - Kit Curumim

O software é o intermediário entre o robô e o usuário, pois é por meio desse software que o usuário irá conseguir enviar os comandos desejados para a movimentação do Curumim. Para tanto, apresenta-se duas opções de programação: em blocos e em linguagem C/C++.

A primeira opção de programação é formada por um conjunto de blocos que o usuário escolhe um a um para montar seu

algoritmo, a fim de que o mesmo realize as ações planejadas através do envio das instruções pelo rádio-base.

Dessa forma, ao montar um programa, o usuário terá que raciocinar sobre quais blocos utilizar e em que ordem “encaixá-los”, além de usar criatividade para conseguir alcançar o objetivo pretendido [1].

O software possui ainda um “tradutor” de blocos para C/C++. A cada bloco incluído em seu programa existe um trecho de código equivalente que vai construindo o programa em C/C++ respectivo. Esse código é disponibilizado ao usuário para que o mesmo possa acessá-lo e até alterá-lo. O objetivo é levar o usuário a um nível mais elevado, pois tal funcionalidade propicia a familiarização com a linguagem C/C++. Assim, também há a possibilidade de construir programas mais elaborados para o Curumim.

2 O ROBÔ CURUMIM

Como visto na Figura 2, o robô Curumim é composto de alguns itens que facilitam a execução de suas tarefas:

- Uma pinça na sua frente que funciona no sentido vertical para cima e para baixo, onde o usuário pode colocar uma caneta e com seu posicionamento para baixo ter marcada a trajetória do robô;
- Uma câmera que fornece imagens da frente do robô, podendo o usuário solicitar uma foto ou a leitura de um QRCode durante a execução de cada uma das ações;
- Cinco sensores utilizados para detectar obstáculos que sejam próximos ao robô;
- Um conjunto de três motores utilizados para locomoção, por meio de rodas omnidirecionais, rodas estas que infligem uma composição de movimentos diferentes ao robô e que não poderiam ser efetuados por rodas convencionais.



Figura 2 - Características do robô

3 SOFTWARE DE PROGRAMAÇÃO EM BLOCOS

O software é o ambiente de programação do robô Curumim, no qual o usuário determina as ações que deseja que este robô execute. Possui uma interface amigável e simples de utilizar [4], descrita em cinco áreas, como ilustradas na Figura 3:

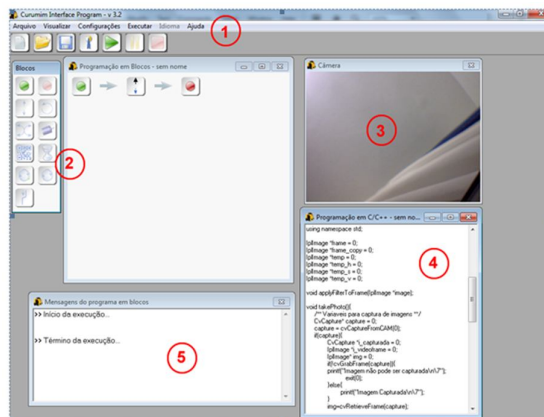


Figura 3 - Interface do Software Curumim

1. Área de Menu e Barra de Ferramentas: o menu é formado com as opções gerais disponível do software, desde o gerenciamento do programa gerado pelo usuário, assim como configurações, visualizações e ajuda para melhor utilização software. Algumas dessas opções ainda tem a combinação de teclas para acesso direto. A barra de ferramentas possui algumas das opções contidas no Menu com a finalidade de servirem de teclas de atalho rápido, sendo ilustradas com figuras que as identificam mais facilmente.
2. Área de Programação em Blocos: principalmente destinadas aos usuários iniciantes que nunca programaram um robô e não conhecem uma linguagem específica para tal. Isso porque conta com uma série de blocos em seu menu com as ações já pré-determinadas para o robô, o que torna transparente ao usuário os comandos que cada bloco envolve para que tais ações sejam realizada. Um desenho significativo serve de ilustração para cada um dos blocos, o que facilita sua identificação. A cada seleção de um bloco o mesmo é inserido na janela de programação ao lado do referente menu.
3. Área de Imagem: possui uma janela que exibe a imagem capturada pela câmera do robô. O usuário pode solicitar sua exibição ou fechamento sempre que quiser, exceto durante a execução do programa em blocos em que sua exibição é feita automaticamente pelo software.
4. Área de Programação C/C++: com a possibilidade de construção de um programa em C/C++ para o robô, tem-se a oportunidade de crescimento do usuário iniciante, seja a “tradução fiel” do programa em blocos ou a “tradução” acrescida de alterações realizadas pelo usuário. E, os usuários experientes tem nessa janela uma opção de programação mais avançada para o robô.
5. Área de Mensagens: possui uma janela que exibe as informações referentes ao resultado da compilação e linkagem do programa C/C++ ou a indicação de cada bloco executado pelo robô durante a execução de seu respectivo programa, fornecendo ao usuário um retorno das ações realizadas.

O menu e a barra de ferramentas possuem opções úteis tanto para a programação em blocos, quando para visualização da imagem capturada pela câmera do robô e do código em C/C++, etc.

O menu está sempre visível, podendo algumas de suas opções estarem desabilitadas em certos momentos, de acordo com cada situação, durante a execução do software, evitando que o usuário faça alguma solicitação indevida. As opções do menu são separadas em grupos, como ilustrado na Figura 4:

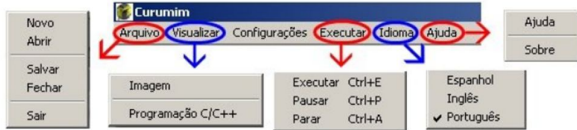


Figura 4 - Menu do software

- Arquivo: possui as opções referentes ao gerenciamento dos programas do usuário, como criar um novo, abrir, salvar ou fechar um programa, além da opção de sair do software Curumim;
- Visualizar: possui as opções de visualização da imagem gerada pela câmera do robô e a “tradução” do programa em blocos para C/C++;
- Configurações: menu de configuração da comunicação, do compilador e dos produtos da câmera;
- Executar: possui as opções de compilar um programa C/C++, desde que haja um aberto, executar um programa em blocos ou em C/C++ (no caso do programa em C/C++, é necessário que o programa já esteja compilado para que o mesmo possa ser executado), pausar e parar, desde que haja um programa em blocos em execução;
- Idioma: possui idiomas disponíveis para o software – Espanhol, Inglês ou Português;
- Ajuda: possui a opção de ajuda, que disponibiliza informações do modo de uso do software. Tem a finalidade de auxiliar o usuário com suas dúvidas e opção de sobre, que exhibe informações a respeito do software.

Dentre essas opções, as que estão disponíveis para o acesso rápido na Barra de Ferramentas são ilustradas na Figura 5:



Figura 5 - Barra de Ferramentas do Software

3.1 Configurações

A janela de configuração é responsável pelas determinação de valores a serem utilizados pelo software, especialmente na comunicação, pois é referente ao fluxo de informação entre o software e o robô. Possui duas abas: Comunicação e Câmera, como ilustrado na Figura 6.

A aba de comunicação é obrigatória para que o usuário possa ter a sua disposição as demais funcionalidades do software do Curumim. As outras abas de configuração podem ser preenchidas durante o uso do software quando o usuário desejar ou quando solicitado pelo próprio software.

Nessa aba tem-se campos para o preenchimento adequado da porta serial e endereços de origem e destino a serem utilizados durante a comunicação com o robô. Inicialmente o usuário deve preencher a porta serial em que está ligado a rádio-base. Caso a porta serial não esteja disponível, basta clicar no botão Detectar Portas para que o software possa identificá-la e o usuário então selecione-a.

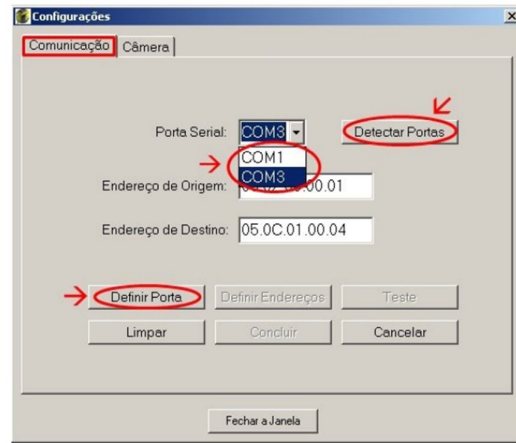


Figura 6 - Janela de configurações

Após a definição da porta serial, o usuário deve preencher os endereços de origem (identificação do rádio-base) e de destino (identificação do robô) e defini-los também. Em seguida o usuário deve poder concluir suas alterações ou, se preferir, realizar uma verificação da comunicação com o robô por meio do botão Teste.

Na aba Câmera o usuário determina o nome padrão para os arquivos de fotos que podem vir a ser geradas pelo robô durante a execução de seus movimentos. A nomeação dos arquivos de foto é feita pela junção do nome padrão determinado pelo usuário com uma numeração ordinária controlada pelo software. O usuário também pode determinar os locais onde tais arquivos serão armazenados.

3.2 Área de Programação em Blocos

A área de programação em blocos é o destaque do software, pois foi elaborada especialmente para tornar a determinação de ações para um robô como uma situação simples, intuitiva e independente de quaisquer conhecimentos em uma linguagem de programação específica.

Além das figuras, que tem por objetivo sua identificação imediata, os blocos contidos no menu ainda possuem hints que auxiliam os usuários a identificar o nome de cada bloco, quando passado o mouse por cima deles. Na Figura 7 é identificado cada um dos blocos por seus nomes.

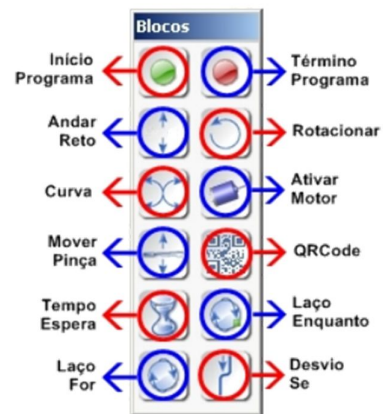


Figura 7 - Menu de Blocos

O bloco Tempo de Espera é um bloco que faz com que o robô permaneça parado, sem executar qualquer ação, durante um certo tempo.

Os blocos Laço PARA, Laço ENQUANTO e Desvio SE são blocos condicionais que englobam outros blocos, sendo que a avaliação da condição do bloco condicional em questão. A condição avaliada por cada condicional é:

- Laço PARA: enquanto o número de vezes determinado para sua repetição não for ultrapassado, os blocos envolvidos pelo Laço PARA são avaliados e executados;
- Laço ENQUANTO: enquanto a combinação do resultado dos sensores escolhidos do robô for verdadeira e o número máximo de vezes determinado para sua repetição não for ultrapassado, os blocos envolvidos pelo Laço ENQUANTO são avaliados e executados;
- Desvio SE: se a combinação do resultado dos sensores escolhidos for verdadeira, os blocos envolvidos pelo Desvio SE são avaliados e executados.

Ao inserir qualquer um dos blocos, exceto delimitadores, uma janela com as informações do bloco é aberta com valores padrões que podem ser alterados pelo usuário de forma simples e prática. Além disso, conta com uma verificação dos valores preenchidos para que o usuário não realize nenhum preenchimento equivocado. Como por exemplo, algumas dessas janelas são apresentadas a seguir.

A janela do bloco Andar Reto, exibida na Figura 8, possui os seguintes parâmetros:

- Direção: indicador da direção a ser seguida pelo robô (frente ou trás)
- Velocidade: indicador da velocidade a ser adotada pelo robô (1: baixa; 2: média; 3: alta);
- Distância: indicador da distancia a ser percorrida pelo robô (1 mm até 99.999 mm);
- Recurso de imagem: indicador de uso ou não do recurso de captura de imagem.



Figura 8 - Janela de parâmetros do bloco Andar Reto



Figura 9 - Janela de parâmetros do bloco Ativar Motor

A janela do bloco Ativar Motor, ilustrada na Figura 9, possui os mesmos parâmetros que a do bloco Andar Reto, porém o controle desses parâmetros é feito individualmente para cada uma das três rodas do robô que possui um motor cada. Isso, juntamente com a possibilidade de escolha de quais motores utilizar, permite a criação de movimentos variados a cada combinação de parâmetros.

Já a janela do bloco ENQUANTO, mostrada na Figura 10 apresenta uma área para construção de uma expressão lógica. O usuário pode fazer uso dos resultados dos sensores que quiser, combinando-os por meio dos operadores NOT, AND, OR, NAND ou XOR, além de ter a possibilidade do uso de parênteses. Essa expressão é avaliada durante a execução do programa e, se seu resultado for verdadeiro, os blocos que estiverem envolvidos por esse condicional serão executados. Caso contrário, serão ignorados.



Figura 10 - Janela de parâmetros do bloco ENQUANTO

A medida que o usuário for selecionado os blocos que deseja inserir no seu programa, os mesmos são incluídos na janela de programação em blocos. Além disso, o usuário ainda pode alterar os valores dos parâmetros de seus blocos por meio das janelas já mencionadas e/ou retirar um bloco de seu programa.

Com exceção da inserção sequencial de blocos, as demais funcionalidades mencionadas estão disponíveis no menu popup, diferenciado de acordo com cada bloco: para o bloco de Início é “Inserir um bloco depois”; para os blocos de movimento e condicionais são “Alterar os valores dos parâmetros do bloco”, “Excluir o bloco”, “Inserir um bloco antes” e “Inserir um bloco depois”; e, para o bloco de Término são “Excluir o bloco” e “Inserir um bloco antes”.

Adicionando bloco a bloco ao programa, o usuário determina as opções que deseja para o robô e em que ordem tais ações devem ser realizadas.

Na figura 11, é apresentado um exemplo de programa em blocos que pode ser construído no software. Esse programa foi elaborado por meio da solicitação de sete passos a serem cumpridos pelo robô:

- Baixar a pinça;
- Enquanto não houver obstáculo atrás do robô, andar 30cm para trás, firmando com velocidade média. Considerar essa condição somente válida por até 4 vezes a sua execução;
- Rotacionar para a esquerda 90°, tirando uma foto;
- Se houver obstáculos a sua direita, siga em frente;
- Repita por 4 vezes a curva para a direita a frente;
- Esperar por 2 segundos antes de continuar o próximo passo;

- E, por fim, subir a pinça.

Embora na Figura 11 não apareça diretamente os valores dos parâmetros alterados para tal solicitação, o usuário pode verificar cada um dos valores dos parâmetros dos blocos passando um hint com as informações. Caso queira alterar seus valores, excluir ou inserir novos blocos, basta clicar com o botão direito do mouse sobre o bloco desejado e solicitar a ação pretendida.

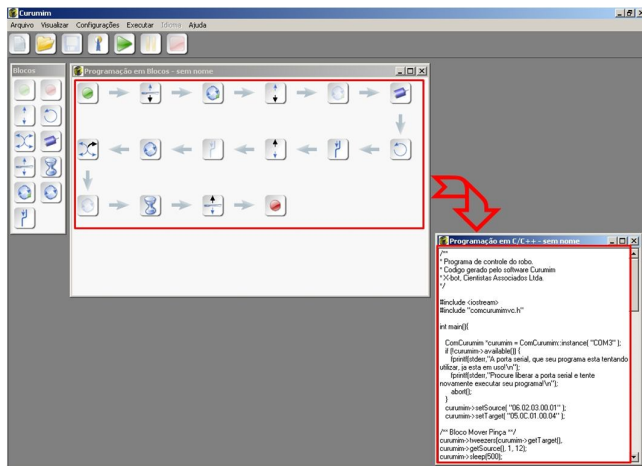


Figura 11 - Exemplo de um programa em blocos e sua tradução para C/C++

3.3 Execução do programa em blocos

Quando o usuário solicita a execução do programa em blocos, é aberta uma janela logo abaixo da janela de programação em bloco que indica as ações executadas. Isso possibilita ao usuário acompanhar uma a uma as ações realizadas pelo robô.

Ainda durante a execução do programa em blocos, o usuário pode pausar e, posteriormente, solicitar o retorno a execução do programa, além de poder parar a execução do mesmo.

3.4 Tradução do programa em blocos

O usuário pode solicitar a “tradução” do programa em blocos para linguagem CC++. Com isso, pode ir aprendendo a sintaxe da linguagem.

Mesmo para quem já está familiarizado com a lógica de programação, haverá o acesso aos métodos que enviam os comandos para os robôs, podendo alterar seus parâmetros diretamente. Assim, o usuário pode incluir outras estruturas mais complexas em seu programa. A compilação desse programa pode ser feita no próprio software, bastando fazer a configuração de um dos dois compiladores solicitados pelo software. Também é possível executar esse programa, já compilado.

Todavia, se preferir editá-lo em outro ambiente mais específico da linguagem, basta salvar esse programa, abri-lo no ambiente desejado e importar a respectiva biblioteca dos comandos para o robô.

4 CONSIDERAÇÕES FINAIS

O software do Curumim foi desenvolvido com uma interface simples e amigável de modo a alcançar o maior número de usuários possível. Também prevê-se a evolução dos usuários iniciantes, que em um primeiro contato só programariam o robô via blocos e, posteriormente, poderiam passar a utilizar a linguagem C/C++. Tal objetivo é esperado pois acredita-se que

o interesse pelo aprendizado de programação seja facilmente despertado por intermédio do robô Curumim.

Vale ressaltar que através do robô o usuário não estaria limitado apenas ao estudo de linguagens de programação, mas também a todas as outras tecnologias incluídas dentro da robótica, tais como mecânica, eletrônica, sensoriamento, entre outras. Desta forma, acreditamos ter desenvolvido um caminho de aprendizado muito eficiente, dinâmico e prático.

REFERÊNCIA BIBLIOGRÁFICA

- [1] Castilho, M. (2002). *Robótica na Educação: Com que Objetivos?* Monografia de conclusão de Pós-Graduação em informática na educação. Universidade Federal do Rio Grande do sul, Porto Alegre – RS.
- [2] Konzen, I. M. G., Cruz, M. E. J. K. (2007). *Kit de Robótica Educativa: desenvolvimento de aplicação metodológica.* II Escola Regional de Licenciatura em Computação, Universidade Federal de Santa Cruz do Sul, Santa Cruz do Sul – RS.
- [3] Rocha, H. V., Baranauskas M. C. C. (2000). *Design e Avaliação de Interface Humano-Computador.* XII Escola de Computação, São Paulo – SP.