



APLICAÇÃO DO SISTEMA OPERACIONAL ANDROID EM ROBÓTICA MÓVEL

Custódio Gastão da Silva Júnior - custodio.silva@cba.ifmt.edu.br

Ivoney Silva Borges – ivoneyborges@gmail.com

Leonardo Fernandes Cherubini – cherubini18@hotmail.com

Ronan Marcelo Martins – ronan.martins@cba.ifmt.edu.br

Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso – Campus Cuiabá

Rua Professora Zulmira Canavarros, 93 - Bairro Centro

CEP 78005-200 – Cuiabá - Mato Grosso/MT

Resumo: *O presente trabalho mostra o desenvolvimento de uma aplicação Android capaz de se comunicar com um robô Lego MindStorms NXT 2.0, utilizando o bluetooth como protocolo de comunicação. Esta aplicação está estruturada no modelo cliente-servidor, tendo o Robô Lego MindStorms NXT 2.0 como servidor e a aplicação Android como cliente. Os materiais e métodos utilizados envolvem a máquina virtual Java, ambiente de desenvolvimento Eclipse, Lego Mindstorms NXT 2.0, modelo 8547, sistema operacional Android, smartphone, linguagem de programação Java, máquina virtual Lejos e protocolo de comunicação bluetooth. Além disso, o desenvolvimento do presente trabalho está em sintonia com a atualidade referente as aplicações móveis que, de acordo com a União Internacional de Telecomunicações (ITU), aproximadamente, 6 bilhões de pessoas no mundo já possuem celulares e outros dispositivos móveis vem ganhando cada vez mais importância, integrando a realidade das mídias sociais, do mercado corporativo e das universidades. O resultado obtido neste trabalho e o seu uso em sala de aula tornou-se fácil e interessante de exemplificar o uso do sistema operacional Android com a robótica móvel, pois, permite demonstrar a abrangência e integração de várias tecnologias, contribuindo para que o estudante absorva com maior clareza os conteúdos empregados no ensino de engenharia. A fim de fortalecer e disseminar o conhecimento adquirido neste trabalho, todo o código fonte da aplicação e o vídeo do resultado obtido estão disponíveis na internet.*

Palavras chave: *robótica móvel, sistema operacional android, java, lego mindstorms NXT*

1.INTRODUÇÃO

Desde os tempos remotos o homem vem trabalhando em busca de novas soluções para a realização de tarefas do dia a dia de forma automática e confortável. E com o passar dos anos, devido às necessidades surgidas, foram acumulados conhecimentos que, atualmente, possibilitam o desenvolvimento e uso de equipamentos que para a sua concretização é necessário integrar conhecimentos de áreas de conhecimentos diversas. Um exemplo disso é a área da robótica cuja utilização crescente de robôs não se verifica apenas na indústria, mas também em diferentes ramos da nossa sociedade. Dois acontecimentos no ano de 2011 ilustram bem essa realidade, isto é, o uso de robô submarino na localização e no resgate do avião da Air France 447 que caiu no Oceano Atlântico em 2009 e a utilização dos robôs de



inspeção para verificar as condições estruturais da usina nuclear de Fukushima Daiichi, Japão, afetada por um tsunami. São exemplos que mostram a grande importância e a gama de aplicações atuais dos robôs e os interesses econômicos e sociais envolvidos em relação ao seu desenvolvimento e aplicação.

Além disso, o presente trabalho está inserido no contexto da educação pedagógica e tecnológica, principalmente, no que concerne a área da robótica, propiciando o aumento do potencial criativo dos estudantes; despertando com facilidade o entendimento do que é ciência; mostrando a importância do trabalho em grupo; contribuindo para o entendimento da aplicação da interdisciplinaridade, enfim, contribui para uma formação científica e tecnológica mais abrangente do estudante.

O tema Robótica ganha mais relevância quando a literatura científica que trata do assunto tem destacado e garantido o quão importante o desenvolvimento de pesquisa na área da Robótica Móvel para o futuro. Prevê-se que a robótica móvel autônoma será responsável por uma nova revolução no mundo moderno, trazendo inúmeros benefícios e grandes desafios para um futuro próximo. Além disso, o tema foco abre a possibilidade de explorar vários temas científicos como, por exemplo, a inteligência artificial que capacita os robôs para a execução autônoma e os sistemas computacionais que permitem dotar esses robôs de habilidades para reconhecer o ambiente onde se encontram estabelecer objetivo, planejar e modificar o ambiente onde estão para realizar os seus objetivos, fornecendo uma gama de aplicações como, por exemplo, inspeção de oleodutos, realização de tarefas em ambientes inóspitos e insalubres, etc.

Ao mesmo tempo, segundo (AGUIAR, 2011) é necessário formar mais e melhores engenheiros no Brasil, requerendo um esforço coordenado e em conjunto entre governo, academia e o setor empresarial. Afirma, ainda, que para se alcançar uma engenharia mais competitiva é necessário, entre outras ações, difundir o papel da engenharia com o propósito de elevar a motivação e a procura pelos cursos de engenharia como também promover uma maior inserção de estudos interdisciplinares nos cursos.

Em paralelo a essa realidade, o crescimento da tecnologia móvel relacionado a celulares, tablets entre outros dispositivos vem ganhando cada vez mais importância e se integrando a realidade das mídias sociais, do mercado corporativo e das universidades. De acordo com a União Internacional de Telecomunicações (ITU), aproximadamente, 6 bilhões de pessoas no mundo já possuem celulares. E com o objetivo de suprir a demanda deste mercado para melhor qualidade e custo benefício para as empresas, desenvolvedores e usuários finais, a empresa Google desenvolveu um sistema operacional denominado Android.

O Android é um sistema operacional de código aberto, possibilitando a contribuição de qualquer desenvolvedor em seu código fonte ou até mesmo corrigir erros do sistema. O sistema operacional Android é baseado no kernel 2.6 do Linux e é o responsável por gerenciar a memória, os processos, os threads e a segurança do sistema no qual o mesmo está operando.

Tomando como referência o cenário apresentado, o presente trabalho tem o objetivo de aplicar o sistema Android na Robótica Móvel, indo além da aplicação do Android em smartphones.

Em linhas gerais, o presente artigo mostra uma aplicação Android capaz de se comunicar com um robô Lego MindStorms utilizando o Bluetooth como protocolo de comunicação. Para tanto, desenvolveu-se uma aplicação estruturada no modelo cliente-servidor tendo o Robô Lego MindStorms como o servidor e a aplicação Android como cliente. Uma aplicação Java será executada no robô e armazenará a solicitação de conexão de um cliente. O aplicativo Android se conectará ao robô possibilitando a comunicação entre os dispositivos.

2.FUNDAMENTOS DO SISTEMA OPERACIONAL ANDROID

A arquitetura do sistema operacional Android é dividida em camadas, isto é: *Applications*, *Application Framework*, *Libraries*, *Android Runtime* e o *kernel* do *Linux* conforme ilustra Figura 1.

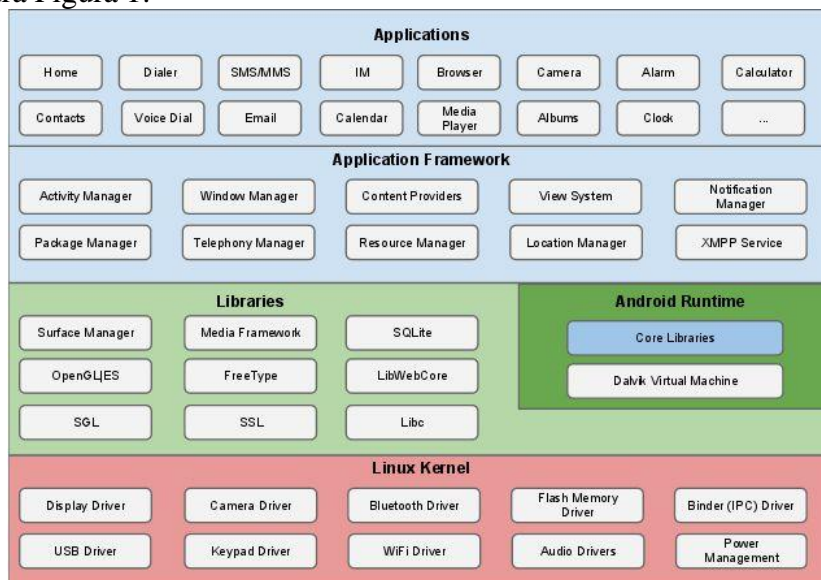


Figura 1: Arquitetura Android (OPEN, 2012).

Onde:

- Camada *Applications*: Onde estão os aplicativos nativos do Android que incluem aplicativos de *e-mails*, calendário, navegador de internet, programa *SMS*, mapa *GPS*, entre outros.
- Camada *Application Framework*: Apresenta os componentes que permitirão que novas estruturas sejam utilizadas para o desenvolvimento de novas aplicações.
- Camada de *Libraries*: São bibliotecas padrão do Android escrita em C/C++. Também possui as bibliotecas de multimídia, do acelerômetro, banco de dados (*SQLight*) e fontes *bitmap*.
- Camada *Android Runtime*: É a máquina virtual *Dalvik* responsável pela execução dos códigos Java das aplicações.
- *Kernel Linux 2.6*: Fornece os serviços do núcleo do sistema.

As aplicações Android são desenvolvidas pela linguagem de programação Java. Mas o Android não tem Máquina Virtual Java para “rodar” os arquivos compilados (extensão “.class”). O Android tem uma máquina virtual chamada *Dalvik* que foi desenvolvida pela *Google* com o objetivo de interpretar códigos desenvolvidos por dispositivos móveis, deixando para o sistema operacional o gerenciamento de memória e o isolamento de processos.

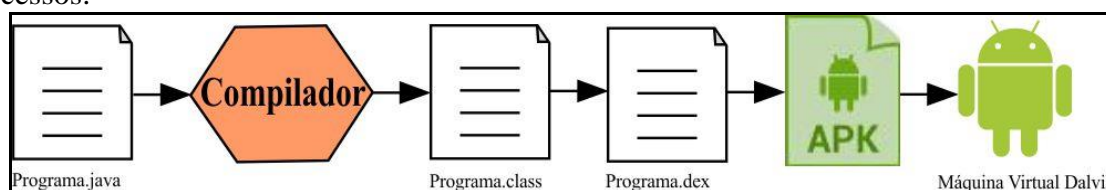


Figura 2: Processo de execução de um aplicativo Android.

Conforme ilustrado na Figura 2, após a compilação do arquivo Java para a extensão .class, uma ferramenta do SDK (*Software Development Kit*) Android chamada “dx” fará a conversão do arquivo de extensão “.class” para a extensão “.dex”, sendo este capaz de rodar



na máquina virtual *Dalvik*. Depois, todos os arquivos “.dex” são compactados em um único arquivo de extensão “.apk” que representa a versão final do aplicativo pronto para ser instalado.

3. MATERIAIS E MÉTODOS

3.1. Lego Mindstorm NXT

O robô *Lego Mindstorms NXT 2.0* pode ser usado para fins educacionais como, por exemplo, na introdução à robótica, automação e, etc. O kit *Lego Mindstorms NXT 2.0* consiste em um microcontrolador ARM e o mesmo representa, neste projeto, o sistema robótico a ser controlado.

3.2. Linguagem de programação Java

É uma linguagem de programação orientada a objetos. Ela é compilada para *bytecode*, podendo ser executada em qualquer sistema operacional e diferentes tipos de *hardware* por uma máquina virtual.

É o padrão mundial para o desenvolvimento de aplicativos web, celular, jogos e softwares corporativos. Com ferramentas abrangentes, um “ecossistema” sólido e um desempenho eficiente. O Java oferece a portabilidade de aplicativos mesmo entre ambientes computacionais distintos.

O Java é uma linguagem fácil de aprender, amplamente usada pelos desenvolvedores e é multiplataforma.

3.3. Máquina virtual Lejos

Lejos trata-se de uma pequena máquina virtual Java desenvolvida para o robô *Lego Mindstorms NXT*, possibilitando um ambiente de programação Java para a programação do robô.

Lejos NXJ possui todas as Classes da *API NXJ* e as ferramentas necessárias para executar no robô *Lego Mindstorms NXT*. O *Lejos NXJ* é um *firmware* substituto para o padrão do *Lego NXT*, incluindo a máquina virtual Java. A biblioteca Java que implementa a *API Lejos NXJ* fornece uma alternativa *Java Runtime* que é otimizada para o *NXT*.

Como o *Lejos* é um *firmware* substituto, ele deve ser atualizado para o *NXT*, e irá substituir o *firmware* padrão do *Lego Mindstorms NXT*. Isso apaga todos os arquivos atualmente detidos no *Lego*. O *firmware* padrão do *Lego* pode ser restaurados usando o software fornecido pela *Lego*.

Lejos é um projeto *open source* hospedado no repositório do *sourceforge*. Ele foi originalmente criado a partir do projeto *TinyVM* que implementou uma máquina virtual Java para o sistema *Lego Mindstorm RCX*.

As vantagens de usar *Lejos NXJ* ao invés de um outro, são:

- Linguagem orientada a objetos usando Java;
- Tipos primitivos da linguagem Java;
- Possui a maioria das classes *java.lang*, *java.util* e *java.io*;
- Tem *API* de robótica bem documentada;
- É um projeto de código aberto, com muitos contribuidores;
- Permite a escolha de ambientes de desenvolvimento integrado, incluindo *Eclipse* e *NetBeans*;

- Suporta os mais recentes recursos da linguagem Java 1.7;
- Tem suporte completo para *bluetooth*, USB, protocolos I2C e RS485;
- Suporta localização, incluindo localização de Monte Carlo (MCL);
- Suporte a som e, inclusive, permite a reprodução de arquivos WAV de 8 bits;
- E, sobretudo, é amplamente utilizado por universidades e estabelecimentos de ensino.

3.4. Ambiente de desenvolvimento *eclipse*

O ambiente de desenvolvimento de aplicativos para a plataforma Android proporciona muitas vantagens para os desenvolvedores. O ambiente de programação Android utiliza a poderosa linguagem de programação Java. O ambiente de programação está disponível para muitos sistemas operacionais *desktop* e possui baixo custo para a distribuição e comercialização de aplicativos. A plataforma não está limitada para celulares e *tablets*, existem televisores, relógios, óculos e vários tipos de eletrodomésticos que possuem o sistema operacional Android. O ambiente de programação para aplicativos Android podem ser desenvolvidos em várias *IDEs* (*Integrated Development Environment*), tendo as mais utilizadas o *Eclipse* e o *Netbeans*. A *IDE Eclipse* é a mais utilizada pela comunidade de programadores e é a recomendada pela *Google*. Para o desenvolvimento de aplicações Android no *Eclipse* deve ser baixado o *SDK* (*Software Development Kit*) e depois um *plugin* do Android deve ser instalado no *Eclipse*.

O desenvolvimento de aplicativo no *Eclipse* já apresenta um projeto inicial pré-definido, contendo pastas organizadas com vários arquivos prontos para o funcionamento de um aplicativo Android. A Figura 3 mostra a organização dos arquivos em destaque da pasta “Aplicativo Android”.

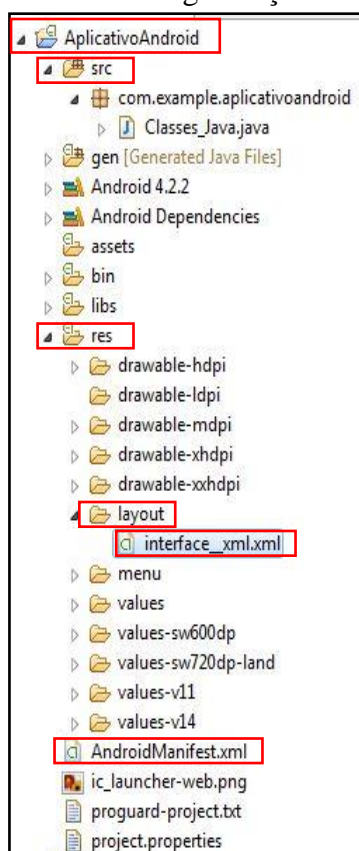


Figura 3: Estrutura do desenvolvimento de um aplicativo Android

A Figura 3 apresenta a estrutura de uma aplicação Android. As principais pastas deste projeto são a *src* e a *layout*. Na pasta *layout* está presente o arquivo *xml*, ou seja, o arquivo *interface_xml.xml*. Este arquivo será o responsável pela criação da interface do aplicativo Android.

A pasta *src* armazena o arquivo *java Classes_java.java*. Esse arquivo será o responsável pela programação do aplicativo. Estão presentes na pasta *src* todos os códigos em Java desenvolvidos no projeto Android.

Outro arquivo muito importante é o *AndroidManifest.xml*, presente na raiz do projeto. Todas as permissões e configurações necessárias para executar o aplicativo são geradas a partir deste arquivo.

3.5. Conexão Bluetooth

A aplicação consiste em utilizar o Android para comunicar com o robô *Lego Mindstorms NXT* por meio do protocolo *bluetooth*.

Essa aplicação baseia-se no modelo cliente-servidor, tendo o robô *Lego Mindstorms* como o servidor e a aplicação Android como cliente. Uma aplicação Java será executada no robô e armazenará a solicitação de conexão de um cliente. O aplicativo Android se conectará ao robô, possibilitando a comunicação entre os dispositivos. A Figura 4 mostra o esquema de aquisição e envio de dados do experimento.



Figura 4: Tabela de aquisição e envio de dados

Na Figura 4 são apresentados todos os passos para o funcionamento da comunicação entre o aplicativo Android e robô *Lego Mindstorms*, isto é:

-Passo 1: Emparelhamento

Como padrão de segurança em toda comunicação *bluetooth*, antes de uma comunicação entre dois dispositivos deve ocorrer o emparelhamento entre os dispositivos para que exista uma identificação entre eles.

O emparelhamento é a identificação entre os dispositivos. É obrigatório para que os dois dispositivos recebam informações suficientes um do outro para fornecer uma conexão segura. Para a execução do emparelhamento, um dispositivo deve enviar um código de segurança para o outro.

-Passo 2: Execução do Servidor

Depois do emparelhamento, o robô vai executar a classe Java desenvolvida para a comunicação de servidor e vai aguardar a conexão do aplicativo Android. Tendo o servidor executado, o aplicativo Android já pode se conectar ao robô. Como os dispositivos já estão emparelhados, o aplicativo tem os dados necessários para a conexão com o robô. Como o *bluetooth* é uma rede sem fio, o aplicativo Android necessita do endereço do robô para poder saber com qual dispositivo se conectar.

-Passo 3: Criptografando a Conexão

O endereço utilizado para este tipo de conexão é conhecido como *MAC (Media Access Control)*. O endereço *MAC* é um endereço físico único em todos os dispositivos conectados em algum tipo de rede. Este endereço é formado por 6 bytes separados por dois pontos (":"), representado por dois algarismos hexadecimais (exemplo: "00:00:4E:6R:G5:77"). Com o endereço *MAC* do robô, antes da conexão o aplicativo deve criptografar a conexão enviando o padrão de identificação *UUID (Universally Unique identifier)* para tornar a conexão única. O identificador único universal (*UUID*) consiste em um padrão alfanumérico hexadecimal dividido em 5 grupos, separados por hífen (exemplo: "00001101-0000-1000-8000-00805F9B34FB").

A linguagem de programação Java disponibiliza o canal de comunicação *socket* que também está presente na biblioteca do Android. Uma variável do tipo da Classe *BluetoothSocket* deverá ser instanciada e em sua configuração inicial ela receberá o identificador *UUID*.

-Passo 4: Conectando

Com a variável *socket* configurada, a conexão já está pronta para ser criptografada, gerando segurança para a comunicação entre o aplicativo e o robô *Lego MindStorms*. Para o aplicativo se conectar definitivamente com o robô, a variável *socket* deve chamar a função *connect* como apresentada na Figura 5.

```
socket = device.createRfcommSocketToServiceRecord(UUID
    .fromString("00001101-0000-1000-8000-00805F9B34FB"));
socket.connect();
```

Figura 5: Conexão *socket*.

-Passo 5: Enviando os dados

Nesta etapa, o celular já está conectado ao robô e pronto para a aquisição e envio de dados entre os dispositivos. Agora, a variável *socket* deverá ser usada para a configuração do fluxo de envio de dados do aplicativo Android.

Uma variável do tipo *DataOutputStream* deve ser instanciada para que o fluxo de envio de dados ser criado e os dados já possam ser enviados para o robô.

A Figura 6 mostra a configuração da variável *DataOutputStream* para enviar os dados para o robô a partir das funções *writeChar* e *writeByte*.

```
output = new OutputStream(socket.getOutputStream());  
output.writeChar(letra);  
output.writeByte(speed);
```

Figura 6: Configuração do output e o envio de dados.

Os comandos enviados pelo aplicativo ao robô serão dois valores, um caractere que representa a direção e um valor inteiro que representará a velocidade executada pelo motor do robô.

5. APLICAÇÃO

O resultado da aplicação dos materiais e métodos da seção anterior será mostrado nesta seção. Trata-se da implantação da comunicação com o robô *Lego MindStorms NXT* utilizando o *bluetooth* como protocolo de comunicação, tendo o robô *Lego MindStorms NXT* como o servidor e a aplicação Android como cliente.

O aplicativo Android é constituído por duas telas, uma é considerada a principal e denomina-se “Tela 1”. A “Tela 1” é onde estão localizados os controles dos movimentos do robô. Enquanto a outra tela tem o objetivo de pesquisar os dispositivos externos para a conexão e está designada como “Tela 2”.

Ao abrir o aplicativo e caso o *Bluetooth* do celular estiver desligado, o aplicativo pedirá a solicitação ao usuário para a ativação do *Bluetooth* como mostrado na Figura 6.



Figura 6: Tela inicial para quando o Bluetooth estiver desligado.

Por outro lado, no momento em que o aplicativo for inicializado e o celular estiver com o *Bluetooth* ligado não aparecerá a solicitação de ativação como mostrado na figura 6, mas sim a tela principal (Tela 1) será aberta normalmente como ilustrado na Figura 7.

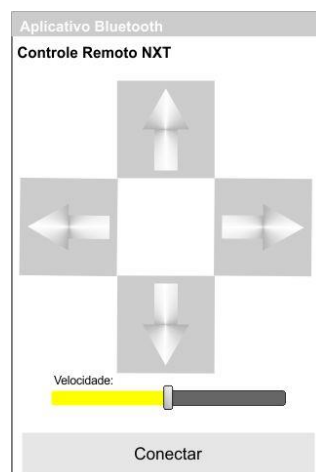


Figura 7: Tela de controle dos movimentos do robô – Tela 1.

Depois que a aplicação estiver em sua tela principal (Tela 1) como apresentada na Figura 7 com o *Bluetooth* já ativado será necessário a conexão com o dispositivo externo. O “botão” conectar da tela principal (Tela 1) deverá ser “clicado” e assim a tela de pesquisa (Tela 2) será chamada, deixando a “Tela 1” em modo de espera.

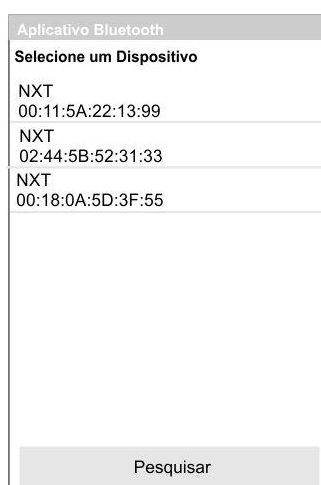


Figura 8: Tela de pesquisa vinculada a dispositivos externos – Tela 2.

Conforme apresentado na Figura 8, a tela de pesquisa (Tela 2) apresentará uma lista com os dispositivos externos encontrados em um raio de 10 metros e um botão para refazer a pesquisa.

O momento em que a tela de pesquisa (Tela 2) é chamada, ela ativará o modo de pesquisa do rádio *Bluetooth* do celular Android ocupando toda a banda de conexão e armazenará na lista da “Tela 2” os nomes dos dispositivos encontrados e seus respectivos endereços MAC (*Media Access Control*).

O botão pesquisar da Figura 8 tem a função de fazer a pesquisa por dispositivos externos como já é feito automaticamente quando a tela de pesquisa é inicializada. Depois do

dispositivo é encontrado que no caso desta pesquisa é o robô *Lego MindStorms NXT*, o item onde se encontra o nome do dispositivo deve ser clicado e, assim o endereço do dispositivo é retornado para a tela principal (Tela 1) e a tela de pesquisa (Tela 2) é finalizada. Depois do retorno para a tela principal, usando o endereço *MAC* do dispositivo selecionado para fazer a conexão.

Conforme explicado anteriormente, para se fazer a conexão, o aplicativo armazena um código identificador no *socket Java* da conexão e, posteriormente, pega o endereço *MAC* que veio da tela de pesquisa (Tela 2) e se conecta ao robô *Lego Mindstorms NXT*.

Os comandos utilizados para o controle do robô se encontram na Figura 7, apresentando as direções que serão executadas pelo robô e um medidor de velocidade para estabelecer um valor de 0 a 100, correspondentes à velocidade dos motores do robô *Lego Mindstorms NXT*.

Os dados enviados para o robô são um caractere e um valor inteiro, definindo a trajetória do robô e sua respectiva velocidade. Os caracteres são 'w' (frente), 'a' (esquerda), 'd' (direita) e 's' (trás). O robô já tem sua programação definida esperando apenas o caractere e o valor inteiro para executar seu movimento. Dessa forma, os movimentos do robô são controlados pelo celular e/ou *smartphone*.

Todo o código fonte desenvolvido está disponível na internet em (Cherubini, 2013) e o vídeo contendo o resultado da aplicação está em (Cherubini, 2013). No vídeo, fica demonstrado o controle de direção e de velocidade do robô.

6. CONSIDERAÇÕES FINAIS

Algumas dificuldades foram encontradas durante o desenvolvimento deste trabalho. Uma pela escassez de material bibliográfico disponível sobre as bibliotecas *Lejos*, tendo encontrado apenas o blog oficial do *Lejos* e o livro do autor *Brian Bagnall*. A outra foi relacionada com a comunicação *Bluetooth* com os dispositivos envolvidos no experimento, haja vista que a raridade bibliográfica sobre *Lejos* que aborde e ensine a operar com o pacote *bluetooth* para a comunicação com celulares e/ou *smatphones*. Diante disso, a solução foi recorrer a interpretação da documentação disponível sobre *Lejos*. O outro obstáculo também decorrente de bibliografia disponível refere-se à biblioteca *Bluetooth* do *Android*. As bibliografias pesquisadas apresentam um conteúdo muito superficial sobre esse assunto. Neste caso, a solução encontrada foi pesquisar o material bibliográfico de referência do portal de desenvolvedores oficial da *Google*.

Vencida a etapa das dificuldades, o resultado obtido neste trabalho e o seu uso em sala de aula de aula, tornou-se fácil e interessante exemplificar o uso do sistema operacional *Android* com a robótica móvel, pois, permite a um baixo custo demonstrar a abrangência e integração de várias tecnologias envolvendo Engenharia da Computação e a Engenharia de Controle e Automação, haja vista que várias tecnologias computacionais estão atuando de forma conjunta, tais como, linguagem de programação *Java*, aplicação do conceito de *socket*, comunicação por meio de *bluetooth*, sistema operacional *Lejos* e o *hardware* (*Lego MindStorms NXT*, *smatphone*, *mecânica*).

Essa aplicação evidenciou que a robótica associada a outras tecnologias fornece ao estudante uma ampla visão para a aplicação em outros estudos, favorecendo o espírito empreendedor dos estudantes de engenharia.

O estudo demonstrou, ainda, que utilizar robôs educacionais e empregar desafios como a integração das tecnologias, facilita a compreensão da aplicação da tecnologia na engenharia, fazendo com que o estudante absorva com uma maior clareza os conteúdos empregados no ensino de engenharia.

**Agradecimentos**

Os autores agradecem a participação do grupo PET AutoNet/IFMT na elaboração e execução dos experimentos ilustrados neste trabalho.

REFERÊNCIA BIBLIOGRÁFICA

AGUIAR, Benedito G. **Educação em Engenharia: Realidade Atual**. Anais: XXXIX – Congresso Brasileiro de Ensino de Engenharia. Blumenau/SC, 2011.

LECHETA, Ricardo R. **Google Android**. 2.ed. São Paulo: Novatec, 2009.

BAGNALL, Brian. **Intelligence unleashed**. Winnipeg: VARIANT PRESS, 2011.

CHERUBINI, L.. **Código fonte do cliente da aplicação do presente trabalho**. Disponível em: <<https://github.com/leocherubini/AndroidLegoNXT>>. Acesso em: 15/06/2013.

CHERUBINI, L.. **Código fonte do servidor da aplicação do presente trabalho**. Disponível em: <<https://github.com/leocherubini/ServidorLejos>>. Acesso em: 15/06/2013.

CHERUBINI, L.. **Vídeo do resultado da aplicação do presente trabalho**. Disponível em: <<http://www.youtube.com/watch?v=zyUyFrkNTIA&feature=youtu.be>>. Acesso em: 15/06/2013.

GOOGLE. **Portal de desenvolvedores para Android**. Disponível em: <<http://developer.android.com/index.html>>. Acesso em: 15/08/2012.

DARWIN, Ivan F. **Android Cookbook**. São Paulo: Novatec, 2012.

LEJOS. **Application Programming Interface Lejos**. Disponível em: <<http://lejos.sourceforge.net/nxt/nxj/api/index.html>>. Acesso em: 20/10/2012.

LEJOS. **Tutorial Lejos**. Disponível em: <<http://lejos.sourceforge.net/nxt/nxj/tutorial/index.htm>>. Acesso em: 20/09/2012.

OPEN Source Project. **Android**. Disponível em: <<https://source.android.com/tech/security/>>. Acesso em 30/09/2012.

APPLICATION OF ANDROID OPERATING SYSTEM IN MOBILE ROBOTICS

Abstract: *This paper presents the development of an Android application can communicate with a Lego MindStorms NXT 2.0 robot using bluetooth as the communication protocol. This application is structured in client-server model, and the Robot Lego MindStorms NXT 2.0 as server and Android application as a client. The materials and methods used involve the Java virtual machine, development environment Eclipse and Netbeans, Lego Mindstorms NXT 2.0 8547 model, Android OS smartphone, the Java programming language, virtual machine Lejos and bluetooth communication protocol. Furthermore, the development of this work is in*



keeping with today regarding mobile applications, according to the International Telecommunication Union (ITU), approximately 6 billion people in the world already have mobile phones and other mobile devices is gaining increasingly important, integrating the reality of social media, the corporate market and universities. The result obtained in this work and its use in the classroom lecture has become easy and interesting to exemplify the use of the Android OS with mobile robotics, since it allows to demonstrate the breadth and integration of various technologies, contributing to the student absorb more clearly the content used in engineering education. In order to strengthen and disseminate the knowledge gained in this work, all the source code of the application and the video of the result are available on the internet.

Key-words: *mobile robotics, android operating system, java, Lego Mindstorms NXT*