

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RODRIGO SILVA DA CUNHA

**DESENVOLVIMENTO DE UMA CENTRAL DE CONTROLE PARA CADEIRA DE
RODAS (CCR)**

NATAL – RN
2014

RODRIGO SILVA DA CUNHA

**DESENVOLVIMENTO DE UMA CENTRAL DE CONTROLE PARA CADEIRA DE
RODAS (CCR)**

Trabalho de Conclusão de Curso para
obtenção do título de Bacharel em Ciência
da Computação da Universidade do Estado
do Rio Grande do Norte – UERN.

**ORIENTADOR: Anderson Abner de S.
Souza**

**NATAL – RN
2014**

**Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.**

Cunha, Rodrigo Silva da
Desenvolvimento de uma central de controle para cadeira de rodas / Rodrigo
Silva da Cunha– Natal, RN, 2014.

51 f.

Orientador(a): Prof. Anderson Abner de S. Souza

Monografia (Bacharelado). Universidade do Estado do Rio Grande do Norte.
Campus de Natal. Curso de Ciência da Computação

1. Android. 2. Dispositivo móvel. 3. Cadeira de rodas. I. Souza, Anderson Abner
de S. II. Universidade do Estado do Rio Grande do Norte. III. Título.

UERN/ BC

CDD 004

RODRIGO SILVA DA CUNHA

**DESENVOLVIMENTO DE UMA CENTRAL DE CONTROLE PARA CADEIRA DE
RODAS (CCR)**

Monografia apresentada como pré-requisito para obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à avaliação da banca examinadora composta pelos seguintes membros:

Aprovada em ___ / ___ / _____

BANCA EXAMINADORA

Anderson Abner de Santana Souza (ORIENTADOR - UERN)

Rosierly da Silva Maia (UERN)

Bruno Cruz de Oliveira (UERN)

Dedico este presente trabalho aos meus queridos
pais Jório Correa da Cunha e Janeide Silva da
Cunha e toda a minha família que sempre me
motivou nos Estudos.

AGRADECIMENTOS

Primeiramente eu preciso agradecer à Deus, sem a graça Dele eu não teria tido forças para continuar as etapas deste trabalho. Foi Ele quem me motivou a permanecer firme e forte no caminho certo para obter o sucesso desta monografia.

Em segundo, mas não menos importante, agradeço aos meus pais, Jorio Correa da Cunha e Janeide Silva da Cunha, que, com carinho e dedicação, me sustentaram em todos os momentos, não apenas na vida acadêmica. São meus heróis, que me incentivavam a continuar mesmo quando eu preferia desistir. Sem a força deles eu não teria conseguido.

Ao meu professor orientador Anderson Abner de Santana Souza, que me acolheu como orientando. Agradeço sua paciência na orientação, obrigado por compartilhar seus conhecimentos de robótica comigo, sua mansidão, e seu incentivo, sem estas coisas eu não teria conseguido chegar tão longe.

À minha professora Rosiery da Silva Maia, obrigado por responder minhas dúvidas quanto a elaboração da monografia. Obrigado por dedicar seu tempo auxiliando minhas pesquisas e me encorajando a concluir.

Aos professores da UERN, que se mostraram dispostos a me ajudar com a minha pesquisa.

Aos funcionários da UERN, que abriam e fechavam os laboratórios para que eu pudesse estudar, as vezes até tarde da noite e nos fins de semana.

Aos meus amigos do peito Renato Carlos e Antônio Porto, que foram meu grupo em quase todos os trabalhos durante a universidade. Agradeço pela boa amizade que me proporcionaram nesses anos de curso e para o resto da vida. Bons companheiros de estudo e diversão como eles são raros presentes de Deus. Obrigado por me ajudarem nos estudos, esta ajuda foi fundamental pra mim.

Agradeço, por fim, todos aqueles que não pude citar, que contribuíram de alguma forma para o meu crescimento como pessoa e como estudante acadêmico ao final desta fase da minha vida, a conclusão do curso de bacharelado em ciência da computação.

“O maior inimigo do conhecimento não é a ignorância, mas a ilusão do conhecimento”.

(Stephen Hawking)

RESUMO

Este trabalho tem como objetivo a criação de uma central de Controle para Cadeira de Rodas (CCR) com *hardware* e *software* livres, de código aberto, que seja de baixo custo e de fácil controle pelos usuários. A CCR foi construída com o fim de validar a construção de uma cadeira de rodas que proporcione diferentes formas de controle e comodidade às pessoas que possuam algum tipo de deficiência motora. A CCR é controlada sem fios usando a tecnologia *bluetooth* através de um aplicativo desenvolvido para a plataforma Android, da Google, para dispositivos móveis. O controle por meio de tais dispositivos proporciona comodidade ao usuário que pode escolher controlar a CCR através dos recursos disponíveis no dispositivo, que podem ser: acelerômetro e comando de voz. O núcleo de hardware de interfaceamento usado no desenvolvimento da CCR foi uma placa Arduino UNO, de baixo custo e de programação fácil, munido de 4 motores conectados a uma controladora eletrônica e um sensor ultrassom para detecção de obstáculos, prevenindo situações de colisão. Ao final deste trabalho, foram elaborados testes de funcionamento da CCR e um cenário evolutivo, vislumbrando a construção de uma possível cadeira de rodas baseada nela.

Palavras-chaves: Cadeira de rodas. Arduino. Android. Controle por meio de dispositivos móveis

ABSTRACT

This work aims to create a Central Control to Wheelchair (CCR) with free hardware and software, open source, that is low cost and easy to control by users. The CCR was built in order to validate the construction of a wheelchair that provides different forms of control and comfort to people who have some kind of physical disability. The CCR is controlled wirelessly using Bluetooth technology through an application developed for the Android platform of Google for mobile devices. The control through such devices provides convenience to the user who can choose control the CCR through the resources available on the device, which can be: accelerometer and voice command. The core hardware interfacing used in developing the CCR was an Arduino UNO board, low cost and easy to program, equipped with 4 motors connected to an electronic controller and a sensor for ultrasound detection of obstacles, preventing collision situations. At the end of this work, functional tests of the CCR and an evolutionary scenario were developed, envisioning the construction of a possible wheelchair based on it.

Keywords: Wheelchair. Arduino. Android. Control by mobile devices.

LISTA DE FIGURAS

Figura 1: Protótipo de cadeira de rodas IntellWheels -----	18
Figura 2: Componentes de hardware do CCR -----	21
Figura 3: Módulo bluetooth -----	22
Figura 4: Placa Arduino UNO Fonte: Rodrigo Silva (2014) -----	24
Figura 5: Módulo ultrassônico HC-SR04-----	26
Figura 6: Esquema dos pinos da controladora de motores -----	27
Figura 7: Esquema de acoplagem do módulo bluetooth à placa Arduino -----	29
Figura 8: Esquema de acoplagem entre o Arduino e a placa controladora -----	30
Figura 9: Esquema de conexão entre o Arduino e o sensor ultrassom-----	31
Figura 10: Detalhe da protoboard-----	32
Figura 11: Processo de interação entre o usuário e o sistema -----	34
Figura 12: Ciclo de vida da activity -----	37
Figura 13: Diagrama de fluxo de telas-----	39
Figura 14: Diagrama de classe da UI-----	40
Figura 15: Diagrama de máquina de estados da programação do Arduino -----	41
Figura 16: UI com informação de conexão-----	43
Figura 17: Teste de conexão-----	43
Figura 18: Teste de controle por acelerômetro-----	44
Figura 19: Teste de controle por comando de voz -----	45
Figura 20: Teste de detecção de obstáculos -----	47

LISTAS DE ABREVIATURAS E SIGLAS

- ADT** *Android Development Tools*
- API** *Application Programming Interface*
- CAD** *Computer Aided Design*
- CC** Corrente Contínua
- FPGA** *Field Programmable Gate Array*
- IDE** *Integrated Development Environment*
- LCD** *Liquid Crystal Display*
- MAC** *Media Access Control*
- PIC** *Peripheral Interface Controller*
- SDK** *Software Development Kit*
- SO** Sistema Operacional
- UERN** Universidade do Estado do Rio Grande do Norte
- UI** *User Interface*
- USB** *Universal Serial Bus*
- XML** *eXtensible Markup Language*

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	15
1.1.1	Objetivo Geral	15
1.1.2	Objetivo Específico	15
2	TRABALHOS RELACIONADOS	17
3	COMPONENTES DE <i>HARDWARE</i> E MONTAGEM DA CCR	21
3.1	MÓDULO BLUETOOTH	21
3.2	PLACA ARDUINO	22
3.3	SENSOR DE ULTRASSOM	24
3.4	PLACA CONTROLADORA	26
3.5	MOTORES	28
3.6	MONTAGEM DOS COMPONENTES DE <i>HARDWARE</i>	28
4	CENTRAL DE CONTROLE PARA CADEIRA DE RODAS	33
4.1	PRINCÍPIO DE FUNCIONAMENTO	33
4.1.1	Android SO	35
4.1.2	Estrutura de um código Android	36
4.2	INTERFACE DE CONTROLE NO DISPOSITIVO MÓVEL	37
4.3	PROGRAMAÇÃO DO ARDUINO	40
5	TESTES E RESULTADOS	42
5.1	TESTE DE CONEXÃO COM DISPOSITIVO MÓVEL <i>SMARTPHONE</i>	42
5.2	TESTE DE CONTROLE POR ACELERÔMETRO	44
5.3	TESTE DE CONTROLE POR COMANDO DE VOZ	45
5.4	TESTE DE DETECÇÃO DE OBSTÁCULOS	46
5.5	COMPARATIVO DE PREÇOS	47
6	CONCLUSÕES	49
7	REFERÊNCIAS	50

1 INTRODUÇÃO

A robótica e a mecânica estão fortemente presentes nas atividades diárias do ser humano. As máquinas servem para facilitar as atividades humanas quando, por exemplo, um automóvel leva uma pessoa de um ponto a outro, quando um veículo não tripulado é usado para varredura do solo submarino a grandes profundidades ou quando robôs são utilizados em terrenos de guerra. Esses exemplos poupam o esforço humano e o protegem dos riscos, auxiliando na sua integridade e segurança.

Neste contexto, as cadeiras de rodas podem ser classificadas como máquinas criadas para facilitar a locomoção de pessoas com deficiência motora, ou seja, que não podem se mover livremente sozinhas. As cadeiras são fabricadas por várias empresas, cada projeto usa artifícios tecnológicos próprios para adaptar as cadeiras às necessidades dos usuários, seja luxo, conforto, da deficiência do portador, do ambiente de uso ou condições financeiras do portador.

A aplicação dos conhecimentos advindos do mundo da mecânica e da robótica no contexto da comodidade para cadeirantes resultam em componentes inteligentes que facilitam a vida dos usuários. As cadeiras de rodas inteligentes podem se tornar uma importante solução para o desafio de auxiliar os indivíduos que têm deficiência e são, portanto, um foco convergente de interesses entre cientistas e pesquisadores.

Existem projetos (GOMES, 2013), (REIS, 2012), (HOCKEY e MILLER, 2007), concluídos e em desenvolvimento, que produzem componentes inteligentes, elaborados por pesquisadores ao redor do mundo que visam diminuir as dificuldades encontradas pelos deficientes usuários de cadeiras de rodas.

A justificativa para pesquisas com cadeira de rodas inteligentes vem da preocupação com a comodidade dos cadeirantes, bem como pelo crescimento do número de usuários e suas exigências em qualidade de vida. Portadores de deficiências motoras, idosos e obesos são usuários em potencial dessas tecnologias.

Com isto, foi planejado a construção de uma Central de Controle de Cadeira de Rodas com base em Arduino e Android para gerar, em um futuro não distante, cadeiras de rodas comerciais com baixo custo. É importante ressaltar que as

tecnologias utilizadas são abertas para a comunidade, o que facilita a disponibilidade do equipamento aos usuários visto a aceitação dessas plataformas no mercado.

Como núcleo de processamento deste sistema, dentre as possíveis opções como FPGA (*Field Programmable Gate Array*) ou um PIC (*Peripheral Interface Controller*), foi adotado, para o sistema desenvolvido, a placa *Arduino*. Este *hardware*, original da Itália, já está bastante difundido no mundo como uma opção de plataforma de prototipagem eletrônica, *open-source*, apoiado nas facilidades de usabilidade e programação. Foi pensado para ser adotado por grandes projetistas ou por iniciantes, ser de baixo custo, flexível e reconfigurável. Possui linguagem de programação essencialmente C/C++ e possui bastante documentação e exemplos à disposição dos interessados.

No desenvolvimento do sistema de *hardware* foi estudado e o uso de sensores capazes de detectar obstáculos à frente e impedir o movimento da cadeira no sentido do obstáculo. Desta forma a cadeira de rodas auxiliará a locomoção não só de deficientes de membros inferiores como também pessoas com baixa visão, por exemplo. Novas funcionalidades, visando dotar a cadeira de rodas com mais inteligência, poderão ser acopladas ao projeto em trabalhos futuros.

Sistemas de aquisição e percepção para interpretar os sinais do usuário e do mundo podem ser implementados. Sensores diferentes resultam em diferentes abordagens para tomada de decisões. Dentre os sensores usados para calcular distâncias entre objetos e que podem servir para detecção de possíveis condições de colisões, existem os *lasers* e os ultrassônicos. Embora o sensor sonar capte ondas em forma cônica, abrangendo uma área maior do que a área de captação cilíndrica vista pelo sensor *laser* (RIBEIRO, 2013), é consideravelmente mais barato e isto torna este tipo de sensor a opção mais viável ao projeto.

O conceito do *open-source*, o grande número de pessoas com necessidades especiais no Brasil, os altos preços dos equipamentos de suporte aos portadores de deficiência são os fortes motivadores para a realização desta pesquisa.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo deste trabalho é mostrar o processo de desenvolvimento de uma Central de Controle para Cadeira de Rodas para auxiliar na solução de problemas referentes a usuários de cadeiras de rodas que tenham dificuldade em utilizá-las para deslocar-se, provendo a detecção de obstáculos pelo percurso. O sistema é implementado em módulos, permitindo assim que futuramente seja possível a inclusão de novas habilidades à cadeira. Como o sistema pode ser reprogramável, incrementado ou decrementado de novas funções, a cadeira pode ser adaptada para as necessidades particulares de cada cadeirante.

1.1.2 Objetivo Específico

Este projeto demonstra a criação de uma Central de Controle para Cadeira de Rodas motorizada com núcleo em Arduino e controlada por um *software* Android. Para que fosse possível realizar este projeto, foi preciso:

- Analisar a faixa de preços de cadeiras de rodas motorizadas existentes e procurar uma forma de construir um sistema que realizasse funções úteis para um cadeirante e que fosse de baixo custo;
- Estudar e utilizar um robô que possui a forma de acionamento diferencial, parecida a com a de uma cadeira de rodas;
- Configurar o ambiente de desenvolvimento e programação para Arduino;
- Configurar o ambiente de desenvolvimento e programação para Android;
- Montar, integrar e programar do Arduino com a plataforma mecânica e suas interfaces eletrônicas;
- Desenvolver o aplicativo de controle no Android, que possibilita o controle através de acelerômetro e voz por meio de dispositivos móveis;
- Implementar um sistema simples de detecção de colisão, baseado em um sensor de ultrassom;
- Validar o sistema desenvolvido através de experimentos e testes, vislumbrando as funções básicas necessárias para um cadeirante;

- Após a conclusão da montagem e programação do sistema, fazer um comparativo de preço entre o custo de uma cadeira de rodas motorizada normal e a possível cadeira de rodas gerada pela CCR.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma:

- Capítulo 1 – Apresenta uma contextualização do problema abordado e a solução desenvolvida;
- Capítulo 2 – Referencia outros trabalhos semelhantes já existentes;
- Capítulo 3 – Relata o processo de montagem dos componentes de *hardware* do sistema;
- Capítulo 4 – Relata o processo de criação e construção da interface gráfica no dispositivo móvel;
- Capítulo 5 – Apresenta resultados de testes funcionais sobre o CCR;
- Capítulo 6 – Descreve o que foi concluído após todo o processo;
- Capítulo 7 – Referências bibliográficas de artigos e sítios na área pesquisada.

2 TRABALHOS RELACIONADOS

Nos últimos anos, o crescimento da preocupação com os cadeirantes e com a melhoria em sua qualidade de vida foram motivações para o desenvolvimento de projetos de cadeira de rodas que auxiliam os cadeirantes em sua locomoção, promovendo-lhes maior comodidade. A seguir estão relatados alguns desses projetos, os quais se relacionam com o contexto abordado neste trabalho.

- **Cadeira guiada por conversação**

Nos Estados Unidos, pesquisadores observaram que as cadeiras de rodas motorizadas não eram simples de serem operadas por todos os usuários. Operar uma cadeira de rodas motorizada exige que o usuário perceba com precisão o ambiente em que está se movendo, isto é, o usuário deve reconhecer os perigos que estão presentes em seu trajeto, como obstáculos, e por fim, deve ser capaz de manifestar sua vontade de forma precisa para o *joystick* que controla a cadeira (HOCKEY e MILLER, 2007).

Sabendo disso, novas formas de controle para cadeiras de rodas motorizadas vem sendo desenvolvidas com o objetivo de auxiliar os usuários que têm dificuldades em operar um *joystick* normal. Entretanto, a maioria é apenas um reposicionamento do *joystick* e adicionando algum complemento para que esse possa ser operado com outra parte do corpo como o cotovelo, língua ou queixo. Existem, também, recursos como leitura ocular, que verifica os movimentos dos olhos para determinar um comando, também comando por voz, em que o sistema de controle da cadeira interpreta comandos simples do usuário como direita, esquerda, frente e trás.

Visando a melhoria do controle de cadeiras de rodas para essas pessoas, foi criado uma cadeira controlada por voz que entende comandos mais complexos, como uma conversa e não só comandos simples, como palavras. O usuário inicia uma conversa com o sistema de controle da cadeira, como por exemplo, “me leve para minha mesa”, este analisa o ambiente e detecta uma mesa. Na sequência o sistema de controle da cadeira responde ao usuário perguntando se a mesa à qual ele deseja ir é a mesa detectada, caso o usuário confirme, a cadeira se desloca de forma autônoma para perto da mesa.

Este sistema de conversação é uma evolução do sistema de reconhecimento de fala já existente. O sistema da cadeira compreende a linguagem natural e gera saídas para o usuário. Para isso, o controlador da cadeira deve interpretar o que o enunciado do usuário significa no contexto e as implicações dessa enunciação para a conversa.

A compreensão da linguagem natural e raciocínio sobre a interação conversacional tornam esses sistemas qualitativamente diferente de sistemas de comando de palavras-chave simples e por isto serviram de apoio à CCR. A CCR usa o controle por voz como uma das formas de comando e foi preciso estudar como este sistema de controle funciona (analisar palavras e comparar com palavras existentes para superar erros de dicção, pronúncia), portanto este trabalho relacionado serve como parâmetro para a CCR.

- **Intellwheels**

Em Portugal, na Faculdade de Engenharia da Universidade do Porto, um grupo de docentes criou um módulo de controle que pode ser acoplado a quase qualquer cadeira de rodas motorizada comum, possibilitando atribuir inteligência à cadeira. Ver figura 1.

Figura 1: Protótipo de cadeira de rodas IntellWheels



Fonte: <http://www.rehab.research.va.gov/jour/11/489/braga489.html>

Este módulo possui componentes de controle para a cadeira de rodas, como microfones, câmeras e componentes de reconhecimento de terreno por sensores que permitem à cadeira se desviar de obstáculos de forma autônoma.

O objetivo do IntellWheels é transformar uma cadeira de rodas elétrica popular e comercial em um equipamento inteligente, sem alterar grandemente as funções ergonômicas da cadeira e sem tornar o custo alto (REIS, 2012).

O usuário pode controlar a IntellWheels através de várias opções de entrada de dados, sendo elas: um *joystick* tradicional, a mais comum forma de controle de cadeiras de rodas motorizadas; um *joystick* USB com vários botões com funções personalizáveis; movimentos de cabeça, o usuário inclina a cabeça para a direção que pretende se mover e seus movimentos são captados por câmeras especiais para detectar a inclinação; teclado e *touch screen*; por meio de expressões faciais, onde uma *webcam* capta no rosto do cadeirante as expressões faciais que representam para o sistema o comando de seguir em uma determinada direção; e por fim, comando de voz, onde um *software* de reconhecimento de voz interpreta os comandos ditados pelo cadeirante através de um microfone.

Para o reconhecimento do terreno o IntellWheels possui uma barra em “U” com vários sensores ultrassônicos e infravermelhos dispostos na estrutura, cobrindo boa parte do entorno da cadeira motorizada.

Em resumo o IntellWheels implementa inteligência em uma cadeira de rodas elétrica, facilitando a sua utilização por diferentes tipos de usuários tornando-se assim uma rica fonte de referencial teórico para a CCR. A CCR tem um objetivo semelhante, a criação de uma central de Controle para Cadeira de Rodas que equipe uma cadeira de rodas comum com sensor ultrassom e uma nova forma de controle por dispositivo móvel mas tende a ser mais barato que a IntellWheels por se tratar de equipar uma cadeira ainda não motorizada.

- **Freedom**

No estado de Minas Gerais, alunos da Universidade do Vale do Rio Doce desenvolveram um projeto de conclusão de curso denominado Freedom, no ano de 2011. O projeto teve como objetivo proporcionar às pessoas que possuem dificuldade de locomoção ou que dependem de cadeira de rodas, uma maior comodidade e conforto a partir da adição e programação de componentes robóticos à uma cadeira de rodas comum (GOMES, 2013).

O produto final deste projeto foi um sistema que permite ao usuário controlar a própria cadeira de rodas por meio de um *tablet* com sistema operacional Android, com a finalidade de tornar o controle da cadeira de rodas em algo mais intuitivo e confortável. O custo do projeto final também foi calculado e ponderado para ser econômico. A escolha da plataforma Android para controle da cadeira justifica parte do processo de baratear o produto final.

O Freedom foi construído sobre uma cadeira de rodas simples, não motorizada de fábrica, e se utilizando de uma placa Arduino como circuito de controle. Esses elementos também favorecem o barateamento do custo final do produto, o que serviu de inspiração para este projeto.

A cadeira de rodas possui um braço robótico controlado pelo *tablet* que pode ser usado para alcançar objetos em locais de difícil acesso para um cadeirante sem que este precise pedir ajuda a alguém. Este componente robótico acoplado à cadeira garante uma maior independência ao cadeirante.

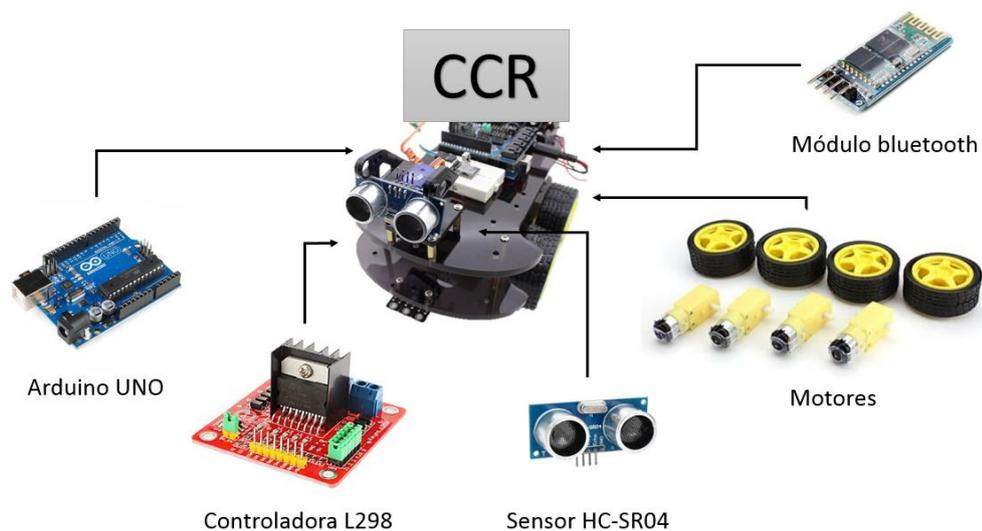
Em resumo, o Freedom tem como objetivo a adoção do sistema Android e o uso de uma placa de circuito em Arduino como forma de barateamento do custo de construção de uma cadeira de rodas modificada com um braço robótico. Esse atributo de barateamento de custos usando um sistema Android no dispositivo móvel de controle serviu de inspiração para a CCR.

3 COMPONENTES DE *HARDWARE* E MONTAGEM DA CCR

Em paralelo com a elaboração da Interface Gráfica (UI), a CCR precisava ser construída com um *hardware* capaz de interpretar os sinais da UI e executar movimentos, simulando uma cadeira de rodas que evitasse colisão com obstáculos.

Para este sistema, foi utilizada uma plataforma mecânica com *drive* diferencial (motores de cada lado acionados de forma independente) de baixo custo, que pudesse imitar os movimentos realizados por uma cadeira de rodas comum. Essa plataforma foi equipada com componentes que podem também ser utilizados em um projeto final de uma cadeira de rodas, adicionando-se alguns outros para adaptação. A figura 2 mostra alguns dispositivos utilizados no CCR.

Figura 2: Componentes de hardware do CCR



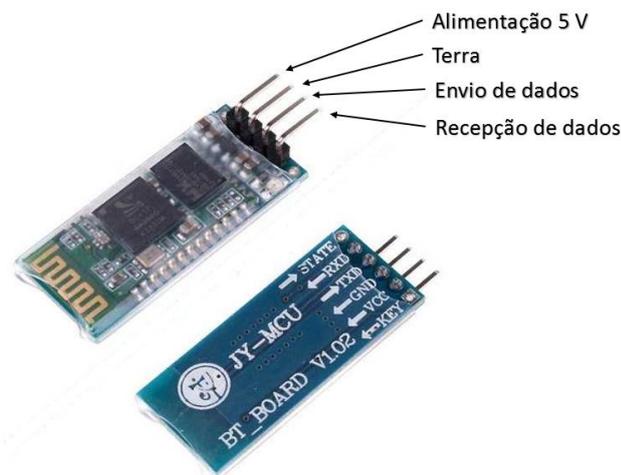
Fonte: Rodrigo Silva (2014)

3.1 MÓDULO BLUETOOTH

O sistema é controlado remotamente pelo usuário, isto significa que não existe um controle fixo conectado diretamente à CCR. Ela recebe os comandos de um dispositivo móvel por meio de uma conexão *bluetooth*. Portanto, a cadeira deve possuir um módulo receptor/transmissor de sinais *bluetooth* para receber os comandos do usuário indicando a direção de movimento do sistema.

O módulo *bluetooth* possui quatro pinos de comunicação, conforme figura 3. O pino de entrada RXD é o pino de recepção de sinais de forma serial, é por ele que o circuito controlador recebe os dados que estão sendo recebidos. O pino de saída TXD é usado para enviar sinais para um receptor. A placa controladora, comentada mais à frente, pode direcionar dados a serem enviados (caso necessário) para este pino. O módulo possui um pino de alimentação VCC de 5V de voltagem e um pino de aterramento GND.

Figura 3: Módulo *bluetooth*



Fonte: Rodrigo Silva (2014)

A conexão *bluetooth*, neste caso, é feita apenas entre o dispositivo móvel e o módulo *bluetooth* no Arduino. Uma vez estabelecida a conexão, o aplicativo no dispositivo móvel poderá se comunicar livremente com o CCR, sem interferência de outros dispositivos, desde que esteja no limite de alcance da conexão.

3.2 PLACA ARDUINO

Arduino é uma plataforma de prototipagem eletrônica de código aberto baseada na flexibilidade e facilidade de uso de *software* e *hardware* (ARDUINO, 2013).

Existem diferentes modelos de placas Arduino no mercado, cada um com diferentes capacidades de processamento, funcionalidades e quantidade de pinos. Também existem diversos complementos para o Arduino como telas LCDs, *shields*, e adaptadores para tornar o Arduino mais abrangente.

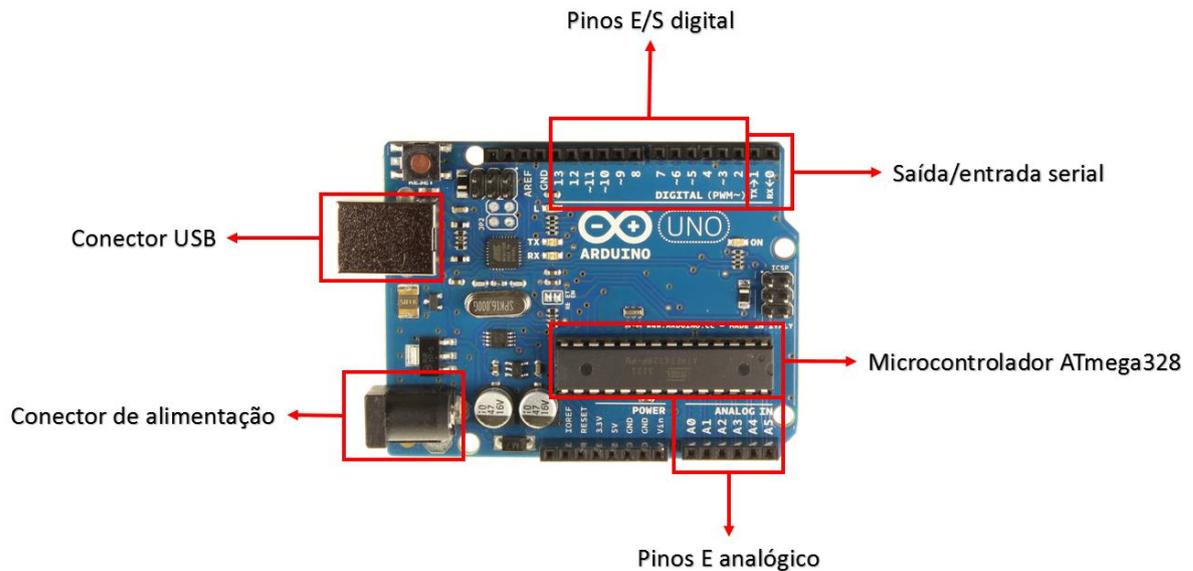
Uma placa Arduino pode sentir e se comunicar com o ambiente por meio de sensores e atuadores que podem ser conectados em seus pinos de entrada e saída de sinais. O microcontrolador da placa é programado por meio de uma linguagem de programação própria do Arduino, baseada em *Wiring* que é muito semelhante a C/C++, possuindo um ambiente de desenvolvimento próprio e disponível gratuitamente também. Seus projetos podem trabalhar isoladamente ou compartilhando funções com outros *softwares* (ARDUINO, 2013).

No endereço oficial eletrônico do Arduino é possível encontrar os motivos que justificam a utilização deste *hardware*:

- As placas podem ser construídas a mão ou compradas pré-montadas, o *software* pode ser baixado gratuitamente.
- Os *designs* de referência do *hardware* (arquivos de *Computer Aided Design* ou CAD) estão disponíveis sob uma licença de código aberto, o usuário é livre para adaptá-los às suas necessidades.

Para a construção da CCR foi utilizado o modelo UNO da placa Arduino, figura 4. É uma placa com um microcontrolador ATmega328, possui 14 pinos de E/S digital (destes, 6 podem ser usados como saídas PWM), 6 pinos de entrada analógicos, 32Kb de memória *flash* e um *clock* de 16MHz. Possui uma porta USB para conexão com um computador que além de permitir a programação da placa ainda fornece os 5V de voltagem necessários para o funcionamento da placa ou é possível a alimentação com uma fonte adaptadora de corrente ou baterias. A figura 4 mostra a composição do Arduino utilizado.

Figura 4: Placa Arduino UNO



Fonte: Rodrigo Silva (2014)

Esta placa possui configurações capazes de implementar as funcionalidades propostas para o CCR, sem que tenha sub ou sobrecarga de recursos. Seu valor está entre os mais baratos entre as placas Arduino do site oficial, sendo este outro motivo principal para a escolha de sua utilização no sistema.

3.3 SENSOR DE ULTRASSOM

O sistema possui um módulo de ultrassom capaz de detectar obstáculos em seu ângulo de visão, isso faz com que a Central de Cadeira de Rodas perceba uma possível colisão e tome atitudes com relação ao seu trajeto pelo ambiente.

O modelo escolhido de sensor de ultrassom foi o HC-SR04 (ver figura 5). Ele é um sensor de baixo custo e possui a precisão aceitável para exigido pelo projeto da CCR, a detecção de obstáculos. Existem outros modelos de sensores capazes de perceber o ambiente e identificar obstáculos além do ultrassom (como por exemplo o sensor a *laser*), adotou-se este modelo pois sua capacidade e seu preço eram adequadas para um sistema de cadeira de rodas de baixo custo. Isto não significa que o sistema não possa ser complementado de novos sensores melhores e mais caros desde que haja consentimento do usuário que optaria por um sistema mais caro, porém, mais preciso.

Um sensor de ultrassom funciona por meio de ondas sonoras de alta frequência. Geralmente o sensor emite um pulso de som e se prepara para ouvir o eco do pulso. Ele calcula distâncias à objetos por meio de uma fórmula simples que envolve a constante da velocidade do som e o tempo de retorno do eco. Sabendo que a velocidade do som é uma constante equivalente a $V_s = 340\text{m/s}$, pode-se encontrar a distância percorrida pela fórmula:

$$Velocidade = \frac{\Delta Espaco (m)}{\Delta Tempo (s)}$$

$$\Delta Espaco (m) = Velocidade \times \Delta Tempo (s)$$

$$\Delta Espaco (m) = 340 \text{ m/s} \times \Delta Tempo (s)$$

Então, basta o sensor ultrassom calcular o tempo em segundos em que um pulso emitido demora para ser ecoado, que é possível determinar a distância entre o sensor e o obstáculo que serviu de anteparo para o pulso sonoro. Como se trata de um pulso que vai até um anteparo e retorna na forma de eco, é necessário dividir o resultado da equação por 2, pois o pulso viaja até o anteparo e retorna, percorrendo duas vezes a mesma distância. Assim, a distância real entre o sensor e o obstáculo é estimada por:

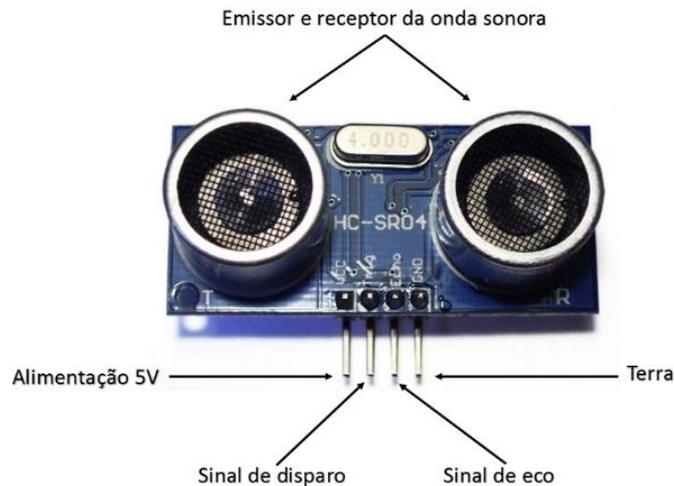
$$Distância do obstáculo (m) = \frac{340 \text{ m/s} \times \Delta Tempo (s)}{2}$$

Após obter a distância ao obstáculo, o CCR pode assumir uma postura de inteligência ao perceber que à sua frente encontra-se um obstáculo, com isto o sistema pode impedir que o usuário consiga mover a CCR na direção do obstáculo, caso este esteja a uma distância inferior à especificada como de segurança.

O módulo HC-SR04 possui um emissor e um receptor de ondas sonoras e quatro pinos em seu circuito, detalhes na figura 8. Pino de alimentação VCC que trabalha com 5V de voltagem, serve para receber energia e permitir o correto funcionamento do módulo. O pino de entrada TRIG, ou gatilho traduzindo do inglês, é por onde o circuito controlador sinaliza ao módulo quando e por quanto tempo o emissor de ondas sonoras deve emitir um pulso sonoro de frequência 40kHz. O pino de saída ECHO, ou eco traduzindo do inglês, é por onde o módulo envia ao circuito

controlador o tempo em que o receptor de ondas sonoras levou para receber o eco do pulso. E, por fim, o GND, ou terra traduzindo do inglês, é o aterramento da voltagem do módulo com 0V.

Figura 5: Módulo ultrassônico HC-SR04



Fonte: Rodrigo Silva (2014)

O módulo possui um alcance máximo de 4m e mínimo de 2cm. O ângulo de visão do ultrassom é de 15 graus. Para o correto funcionamento do cálculo de distancias o módulo sugere que o pulso indicado no pino TRIG dure no mínimo 10 μ s. Para garantir uma leitura correta da distância o plano deve ser o mais regular possível, não possuindo protuberâncias ou relevo irregular.

3.4 PLACA CONTROLADORA

Os motores são acionados por uma placa de circuito de acionamento responsável por controlar a corrente elétrica necessária para fazer os motores girarem, em uma direção especificada.

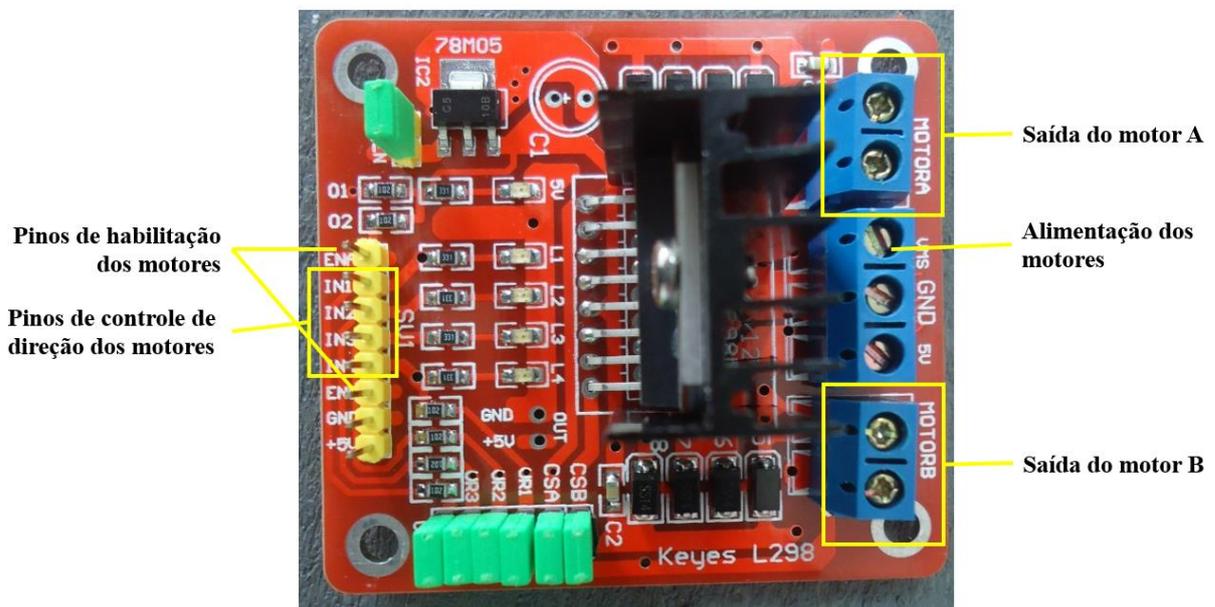
A tensão que o microcontrolador do Arduino usa é de 5V, e os motores de corrente contínua precisam de uma tensão maior. Então a placa controladora funciona como uma ponte (ou interface) entre um circuito com correntes elétricas baixas (Arduino) e um com correntes mais altas, evitando danos ao Arduino.

A placa controladora usada no CCR possui capacidade para controlar dois motores individualmente. Assim, é possível operar qual motor será ativado ou

desativado e a direção de giro de cada um dos motores separadamente. Para o caso da CCR, foram usados quatro motores, estes eram vistos pela controladora como cada par um só motor, sendo os pares controlados simultaneamente.

Uma visão geral da placa controladora pode ser constatada na figura 6. Este dispositivo possui 8 pinos: ENA e ENB são os pinos que habilitam ou desabilitam o funcionamento do motor A ou motor B, no caso da CCR, os motores do lado esquerdo ou os motores do lado direito. IN1, IN2, IN3 e IN4 são responsáveis por indicar o sentido de rotação de cada motor, no caso da CCR indica a direção de rotação de cada lado (par de motores) do sistema. Na prática isso é o que faz o robô se locomover em direções diferentes. Os demais pinos são uma saída de tensão opcional de 5V e um pino GND (terra). A placa controladora também possui saídas de tensão para os motores, uma saída para cada, no caso da CCR, uma saída para cada par de motores, e a entrada de alimentação das baterias de 9V para suprimento da placa controladora e dos motores, como ilustra a figura 6.

Figura 6: Esquema dos pinos da controladora de motores



Fonte: Rodrigo Silva (2014)

3.5 MOTORES

Os motores são os atuadores do sistema, ou seja, a forma pela qual ele se movimenta pelo ambiente. Para a CCR, foram usados 4 motores de Corrente Contínua (CC) que recebem alimentação separada de 6 pilhas AA de 1,5V, totalizando 9V, isto é o bastante para mover a CCR.

O desempenho dos motores varia de acordo com a carga das pilhas utilizadas e com o peso da CCR. O ideal é usar sempre pilhas novas e dedicadas apenas aos motores e sua controladora. Desta forma, praticamente toda a diferença de potencial de 9V gerada pelas pilhas, é consumida apenas pelos motores. Não havendo desvio de carga, os motores funcionam melhor.

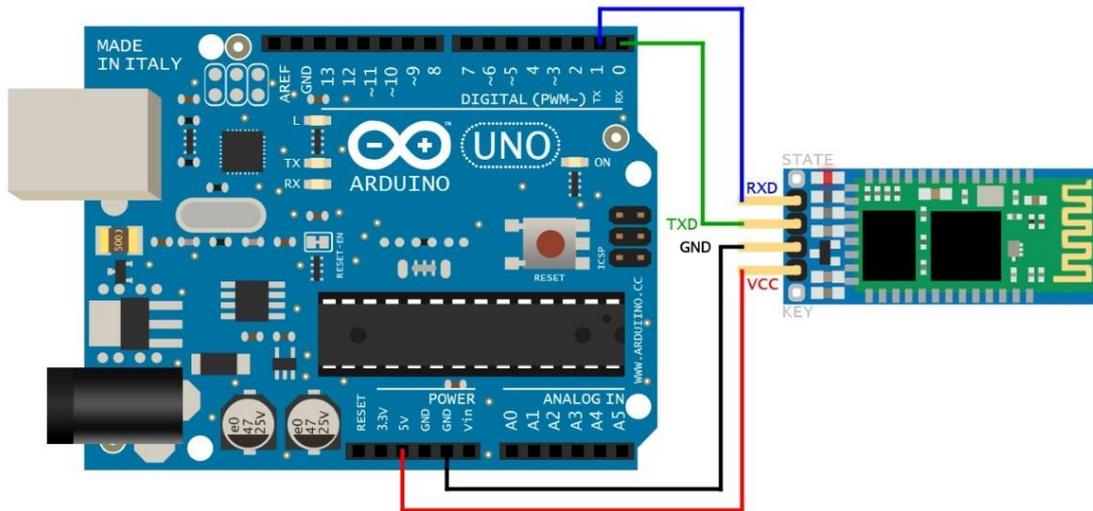
3.6 MONTAGEM DOS COMPONENTES DE *HARDWARE*

Com os componentes de *hardware* do sistema devidamente selecionados e estudados, deu-se início a etapa de montagem. Cada componente tem uma tensão de alimentação própria para funcionar e uma configuração correta para se conectar ao outro componente, e isto precisa ser analisado cuidadosamente para não danificar a CCR.

O componente base do sistema é a placa Arduino. Apenas os pinos digitais e os pinos de entrada e saída serial foram usados para a montagem com os demais componentes.

O primeiro componente a ser conectado ao Arduino foi o módulo *bluetooth*. O módulo possui dois pinos que se comunicam com o Arduino, um para envio e outro para recebimento de sinais. A forma de transmissão desses dados é de forma serial, por isto o módulo foi ligado aos pinos de entrada e saída serial do Arduino. Os outros pinos do módulo estão ligados ao VCC de 5V e ao GND como esquematizado na figura 7 a seguir.

Figura 7: Esquema de acoplamento do módulo *bluetooth* à placa Arduino

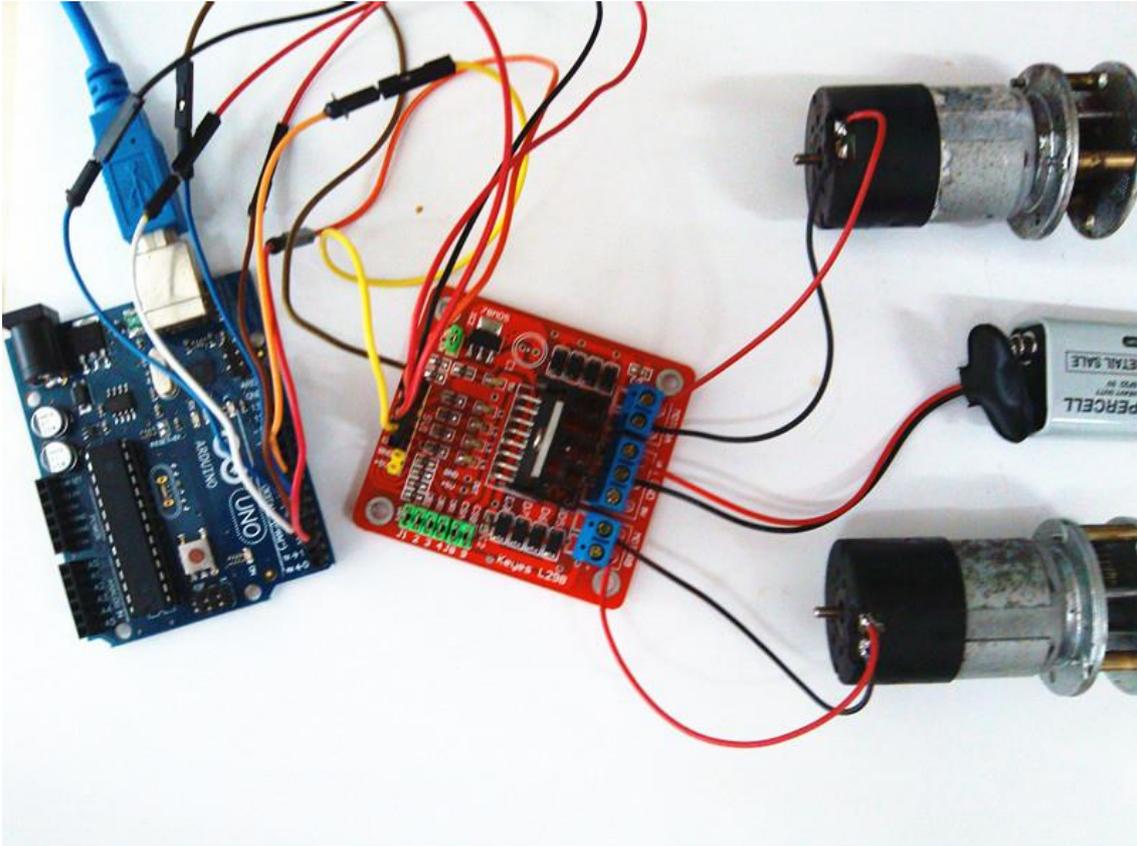


Fonte: <http://forum.arduino.cc/index.php?topic=191834.0>

O pino de envio de dados no *bluetooth* (TXD) se conecta ao pino serial de recebimento de dados (RX) no Arduino e o pino de recebimento de dados no *bluetooth* (RXD) se conecta ao pino serial de envio de dados (TX) no Arduino. Esta conexão é importante, pois é por ela que o Arduino vai receber o comando vindo do dispositivo móvel. A conexão entre o módulo *bluetooth* e o dispositivo móvel é realizada ao fornecer o endereço do módulo ao dispositivo móvel.

Com o Arduino recebendo as informações de direção, é preciso informar à placa controladora qual o sentido de rotação de cada motor para que a CCR siga na direção informada. Para isso, a placa controladora de motores foi acoplada aos pinos de entrada e saída digital de 2 a 7 do Arduino, cada um responsável por controlar a direção e habilitação de giro dos motores dispostos de cada lado da CCR. A figura 8 a seguir representa essa etapa de montagem da CCR.

Figura 8: Esquema de acoplamento entre o Arduino e a placa controladora



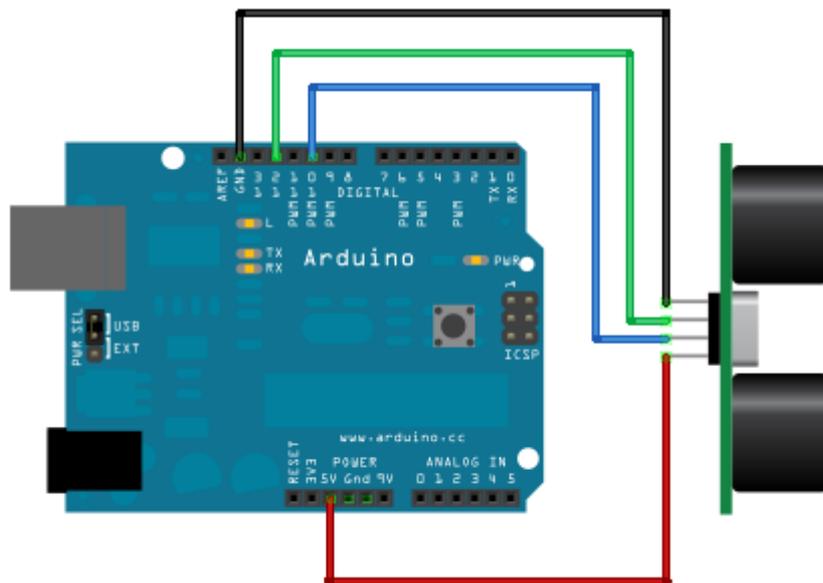
Fonte: http://www.geekonfire.com/wiki/index.php?title=Dual_H-Bridge_Motor_Driver

É importante ressaltar que a CCR possui 4 motores e que esta placa controladora apenas aciona um par de motores. Este problema foi resolvido clonando cada um dos motores, deixando assim, a CCR com 4 motores, mas é visto pela controladora como apenas um par de motores. Isto quer dizer que a controladora comanda cada par de motores como se fosse apenas um, ou seja, os motores do lado direito e os motores do lado esquerdo. Assim, quando a CCR receber o comando para ir para frente ou para trás, os dois lados, direito e esquerdo, se movem no mesmo sentido, ao passo que, quando recebe o comando para girar no sentido horário ou anti-horário os dois lados se movem em sentidos opostos.

Outro detalhe importante é que os motores consomem energia de forma considerável, ao ponto de ser mais eficiente uma fonte de energia de 9V dedicada a este conjunto. Assim, quando a bateria começar a se enfraquecer, não vai prejudicar o rendimento da placa Arduino, tendo ela uma alimentação também dedicada. Para testes ou demonstrações simples, com baterias em bom estado, pode-se alimentar ambos, Arduino e motores, com a mesma fonte.

A etapa final de montagem é a junção com o sensor de ultrassom. O sensor possui dois pinos de comunicação com o Arduino, um para o envio do pulso e outro para recebimento do eco. Cada um destes pinos foi conectado a uma das portas de entrada e saída digital restantes do Arduino. No caso deste projeto, o pino 10 está recebendo o sinal do eco do sensor e o pino 12 do Arduino está enviando o pulso para o sensor ultrassom. A figura 9 indica a conexão entre os dois componentes, o fio azul representa o sinal eco e o fio verde o sinal do pulso.

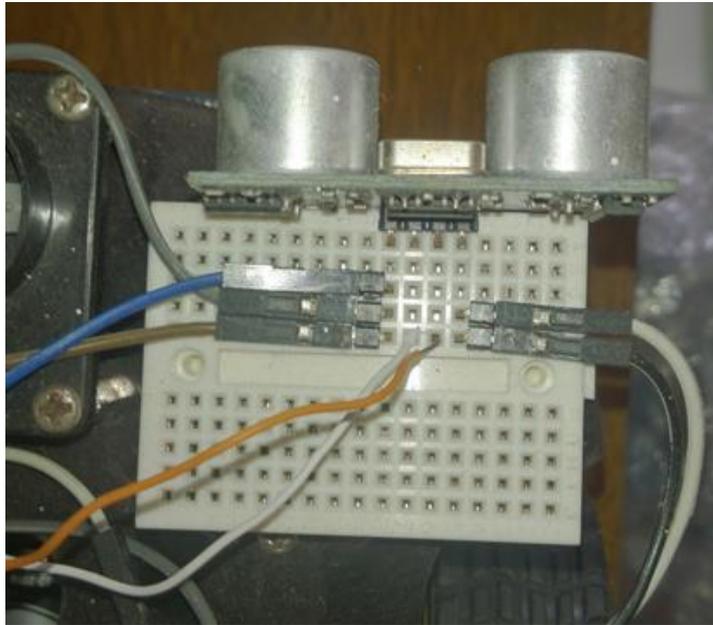
Figura 9: Esquema de conexão entre o Arduino e o sensor ultrassom



Fonte: <http://diasje.com/arduino-sonar-first-tests-with-hc-sr04/>

Os pinos do Arduino de GND e 5V são usados por mais de um componente, por isto foi preciso o uso de uma *protoboard* para distribuir os sinais para todos os componentes que precisaram. Para aproveitar ainda mais o *protoboard* o sensor ultrassom foi conectado diretamente à ele (ver figura 10) pois o *protoboard* possui uma base adesiva que garante estabilidade no corpo da CCR.

Figura 10: Detalhe da *protoboard*



Fonte: Rodrigo Silva (2014)

A compatibilidade dos componentes, o baixo custo associado aos mesmos e a facilidade de manipulação com vasto material disponível para consulta proveniente de outros desenvolvedores, foram motivações para a escolha dos mesmos.

Com todos os componentes conectados, o sistema de *hardware* precisa da lógica do *software* para funcionar. A etapa a seguir relata a elaboração dos componentes de *software* do CCR. Com estas duas etapas prontas o sistema pode então entrar na etapa de testes e resultados para análise de falhas e funcionalidades.

4 CENTRAL DE CONTROLE PARA CADEIRA DE RODAS

4.1 PRINCÍPIO DE FUNCIONAMENTO

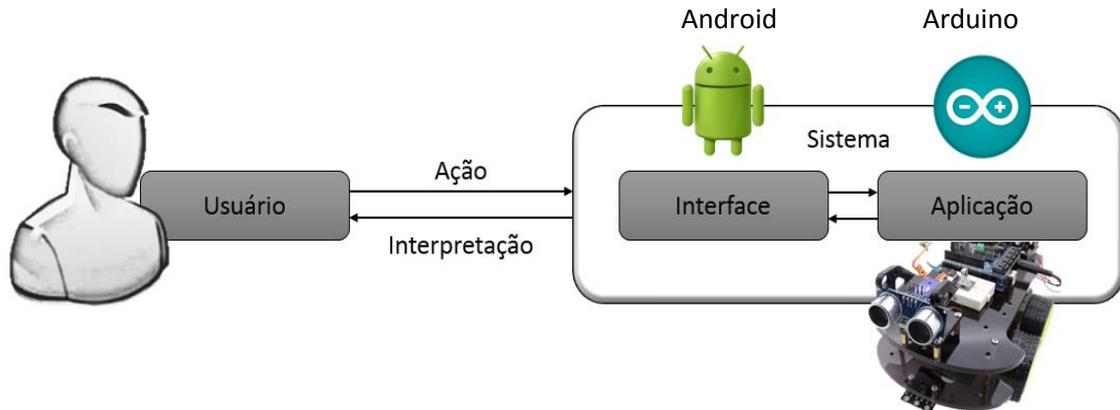
A etapa seguinte desenvolvida para a criação do sistema foi a sua Interface do Usuário (*User Interface - UI*). Moran (1981) criou uma das definições mais estáveis sobre interface. Ele disse que a interface de usuário deve ser entendida como sendo a parte de um sistema computacional com a qual uma pessoa entra em contato física, perceptiva e conceitualmente. Esta definição de Moran demonstra que a interface possui uma perspectiva física, em que o usuário percebe e manipula, e uma perspectiva conceitual, em que o usuário interpreta, processa e raciocina.

A Interface do Usuário possui componentes de *hardware* e *software*. O *hardware* engloba todos os dispositivos com os quais o usuário manipula e percebe o sistema, como por exemplo, uma tela, um teclado, entre outros. Já o *software*, implementa os processos computacionais responsáveis por controlar o *hardware* através da interpretação dos comandos do usuário.

Para facilitar o acesso das pessoas ao sistema desenvolvido, optou-se pela utilização de um dispositivo móvel que possuísse o sistema operacional da Google, o Android, de livre acesso para a comunidade, fácil programação e vasto material disponível. Portanto a UI é um programa desenvolvido para esta plataforma, com o *hardware* da interface sendo um dispositivo móvel (um *smartphone*, por exemplo) e seus sensores (acelerômetro e reconhecimento de voz), e o *software* da interface sendo o sistema operacional Android. Esta escolha não limita o sistema de poder ser projetado em outras plataformas, foi escolhido Android pelo baixo custo dos dispositivos e pela grande aceitação do público a eles.

A UI se comunica com uma placa Arduino. Esta placa também é programável, de *hardware* e *software* livres, e tem a função de receber as instruções do usuário advindas da UI no dispositivo móvel e os dados providos por um sensor de ultrassom e computa as informações para mover a CCR. Na figura 11 é mostrado o esquema de interação entre o usuário e o sistema, e aponta a forma como a interface serve de comunicador intermediário entre o usuário e a aplicação.

Figura 11: Processo de interação entre o usuário e o sistema



Fonte: Rodrigo Silva (2014)

O usuário se comunica com o sistema fornecendo ações, seja um movimento do acelerômetro ou um comando por voz, usando diretamente a interface Android. O sistema interpreta as ações e controla a aplicação. Um retorno visual é exibido ao usuário pela interface para melhorar a interação.

Como já mencionado, foi escolhido um dispositivo móvel com Sistema Operacional (SO) Android como sendo o *hardware* da interface devido à grande popularização destes dispositivos. Os dispositivos móveis inteligentes hoje são amplamente divulgados, estão ao acesso da maioria das pessoas por possuírem valores e atributos diversos. O atrativo destes aparelhos é o fato de juntarem várias funcionalidades antes encontradas em dispositivos distintos agora em um único aparelho, o que os torna dispositivos práticos para o dia a dia das pessoas.

Estes aparelhos são fabricados por várias empresas e cada uma delas concorre por compradores de seus modelos. Para gerenciá-los é preciso um sistema operacional próprio. Algumas empresas fabricam seus próprios sistemas operacionais capazes de controlar o *hardware* dos dispositivos feito por elas. Este tipo de SO é *software* proprietário, ou seja, não pode ser manipulado, copiado ou transferido por aqueles que não possuem permissão do autor. O produtor do *software* proprietário possui direitos autorais sobre o produto e isso priva o usuário final de alguns benefícios. Desta forma produzir uma interface para a CCR nestes requisitos é inviável mas não impossível.

Por outro lado, a mundialmente conhecida gigante da internet, Google, desenvolveu para a sua linha de *smartphones* um SO gratuito e de *software* livre, podendo ser editado pelo usuário final para que se adeque aos seus objetivos específicos. Este SO chama-se Android e é atualmente usado em vários modelos de dispositivos móveis.

Esta alternativa se mostrou mais viável para a construção da UI do sistema visto a liberdade de manipulação do sistema, gratuidade do mesmo e a forte consolidação do SO em dispositivos diversos.

4.1.1 Android SO

O Android é o sistema operacional para dispositivos móveis, como *smartphones* e *tablets*, desenvolvido e gerenciado pela Google, e que possui parte do seu código fonte disponível para a comunidade. Possui um *kernel* baseado em Linux e executa suas aplicações em uma máquina virtual Java melhorada, chamada Dalvik, para requisitar pouca memória e consumir pouca energia. Pesquisas apontam que o Android é o SO para *smartphones* que mais cresce no mundo (LECHETA, 2012).

A linguagem de programação dessa plataforma é a Java, para os componentes gráficos usa-se *eXtensible Markup Language* (XML), e possui um grande conteúdo de auxílio e exemplos na internet que os próprios desenvolvedores compartilham. O Android possui um *Software Development Kit* (SDK) próprio, que providencia as bibliotecas da *Application Programming Interface* (API) e as ferramentas de desenvolvimento necessárias para codificar, testar e gerenciar aplicações (ANDROID, 2014). Uma das ferramenta para desenvolvimento de aplicações em Android é o Eclipse IDE. Este, somado ao *plugin Android Development Tools* (ADT), permite a criação de projetos e pacotes segundo a API do Android, permite a visualização da UI, permite exportar as aplicações no formato “.apk” e distribuir para o dispositivo, emular um dispositivo, entre outras operações (ANDROID, 2014).

4.1.2 Estrutura de um código Android

Existem elementos denominados componentes de um programa Android. Um aplicativo em Android pode até não conter todos os componentes, mas tem que ter pelo menos um deles (NASCIMENTO, 2012). A seguir será mostrado dois pontos importantes em uma aplicação Android, a classe *activity* e os arquivos de *layout*.

Activity: uma aplicação Android interage com o usuário por meio de uma classe chamada *activity*, que é a representação de uma tela da aplicação onde o usuário tem acesso à aplicação e recebe informações dela. Cada tela da aplicação possui uma classe *activity* diferente responsável por gerenciá-la. A *activity* possui um ciclo de vida, representado na figura 12, que é gerenciado pelo SO e indica se a aplicação está, por exemplo, em *background* ou *foreground*.

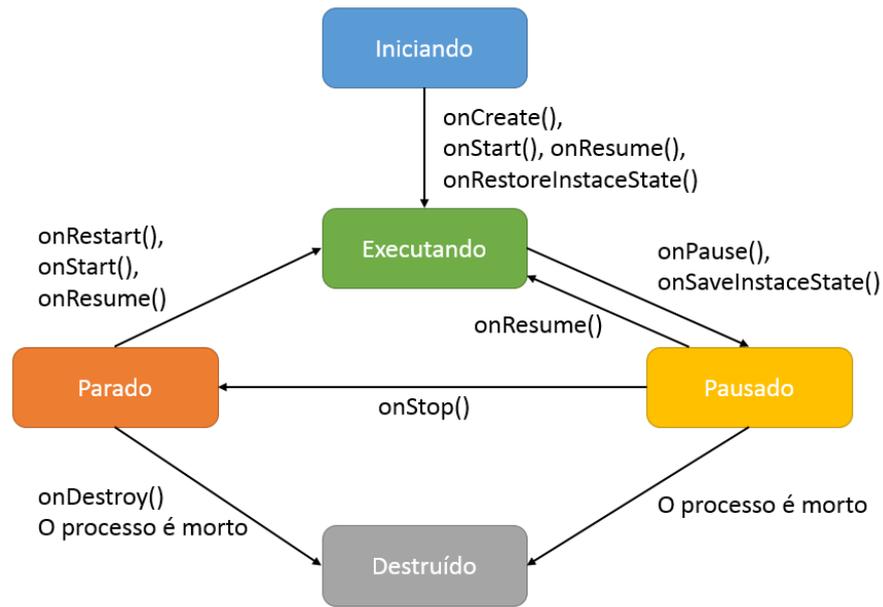
Uma *activity* pode iniciar outra com o objetivo de executar novas ações. Com isso o SO organiza as *activities* que estão em execução em uma pilha que segue a ordem *last in, first out* (último a entrar, primeiro a sair). Isto quer dizer que quando uma *activity* é destruída sua anterior passa para *foreground* (NASCIMENTO, 2012).

A *activity* possui um ciclo de vida em que a aplicação segue os seguintes passos:

- **Criação** – A *activity* é criada e o SO inicializa a tela correspondente à classe vinculando assim a tela ao seu comportamento.
- **Execução** – A *activity* está inicializada e é posta em foco (*foreground*).
- **Pausa** – A *activity* é posta em segundo plano (*background*), o SO salva o status da *activity* enquanto ela não está em foco.
- **Parado** – A *activity* tem seu funcionamento interrompido pelo SO, neste estado a *activity* está pronta para ser destruída ou para ser reinicializada, isto é, voltar ao estado de Execução.
- **Destruído** – O estado final, a *activity* é encerrada e o SO elimina os dados salvos por esta *activity* durante este ciclo. A *activity* não pode mais ser reiniciada no mesmo ciclo pois o SO não possui mais o status.

Existem métodos que são chamados para indicar o estado atual no ciclo de vida de uma *activity*. Estes métodos podem ser vistos também na figura 12. As classes que herdarem a classe *activity* do Android podem sobrescrever os métodos dela afim de realizar diferentes ações em determinados estados da *activity*.

Figura 12: Ciclo de vida da *activity*



Fonte: Rodrigo Silva (2014)

Layout: no Android, assim que uma *activity* é criada, é preciso definir um *layout* para ela. Um *layout* é um arquivo “.xml” que contém toda a estrutura gráfica da tela. Os componentes do *layout* são definidos de acordo com a versão do Android, assim, versões mais recentes do Android contém alguns componentes de *layout* que não existiam nas versões antigas. Por isso é importante ter cautela ao elaborar um *layout* que funcione na maioria dos dispositivos.

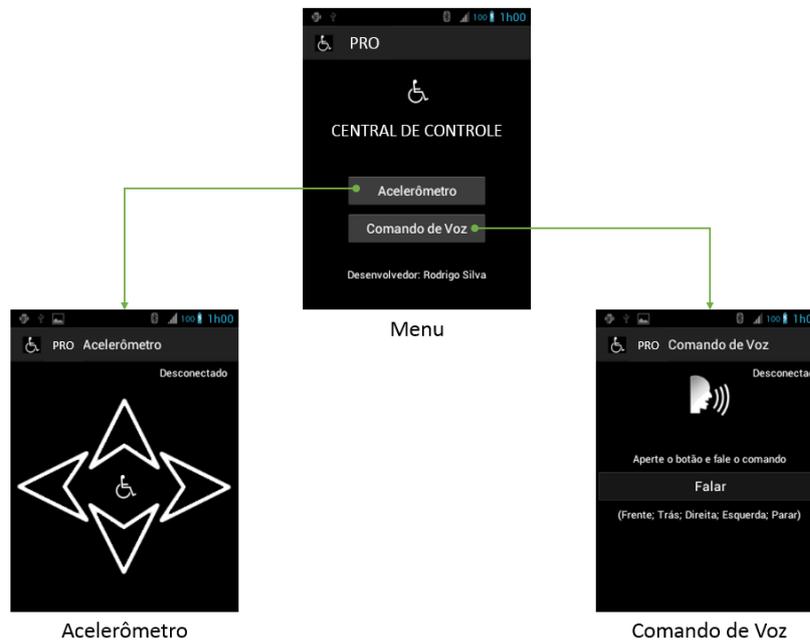
4.2 INTERFACE DE CONTROLE NO DISPOSITIVO MÓVEL

Para criar uma UI para o sistema foi usado o *software* Eclipse IDE – Juno. A intenção foi a construção de um programa que expõe telas intuitivas ao usuário, para que este tenha facilidade na manipulação do aplicativo. O programa possibilita ao usuário o controle dos movimentos da central de Controle para Cadeira de Rodas, via comunicação *bluetooth*, usando os recursos disponíveis no dispositivo móvel. Para este caso, o programa controla a CCR por meio de acelerômetro ou comando de voz.

Para implementar as funções desejadas, foi criada uma UI que possui três telas principais mostradas na figura 13, são elas:

- **Menu** – Nesta tela, o usuário pode escolher a forma de controle da CCR entre acelerômetro ou comando de voz. Aqui também o usuário é notificado caso tente usar umas dessas opções e o *bluetooth* do dispositivo móvel estiver desligado. Isto ocorre porque, no momento em que o usuário escolhe uma das opções do menu, a UI procura estabelecer uma conexão *bluetooth* usando o endereço *Media Access Control* (MAC) já conhecido do sistema. Caso a CCR esteja ligada e pronta para receber uma nova conexão, a UI da opção selecionada é então chamada, informando ao usuário se houve sucesso no processo. Caso esteja incapacitada de se conectar, a UI informa que a comunicação *bluetooth* entre programa e a CCR não foi ainda estabelecida.
- **Acelerômetro** – Nesta tela, o usuário recebe informações visuais do *status* da conexão entre o programa e a CCR, e também sobre o comando de direção que está sendo enviado a partir da inclinação medida pelo acelerômetro do dispositivo móvel. Isso pode ser visualizado na tela da UI por meio de um conjunto de setas direcionais que mudam de cor indicando o comando ativado com a inclinação.
- **Comando de voz** – Nesta tela também é informado ao usuário o *status* da conexão entre o programa e a CCR, e possui as instruções reconhecidas pelo comando de voz. A tela também possui um botão que, quando pressionado, abre uma *activity* nativa do Android, a qual, por meio de uma conexão com a Internet, recebe um comando de voz e o interpreta.

Figura 13: Diagrama de fluxo de telas



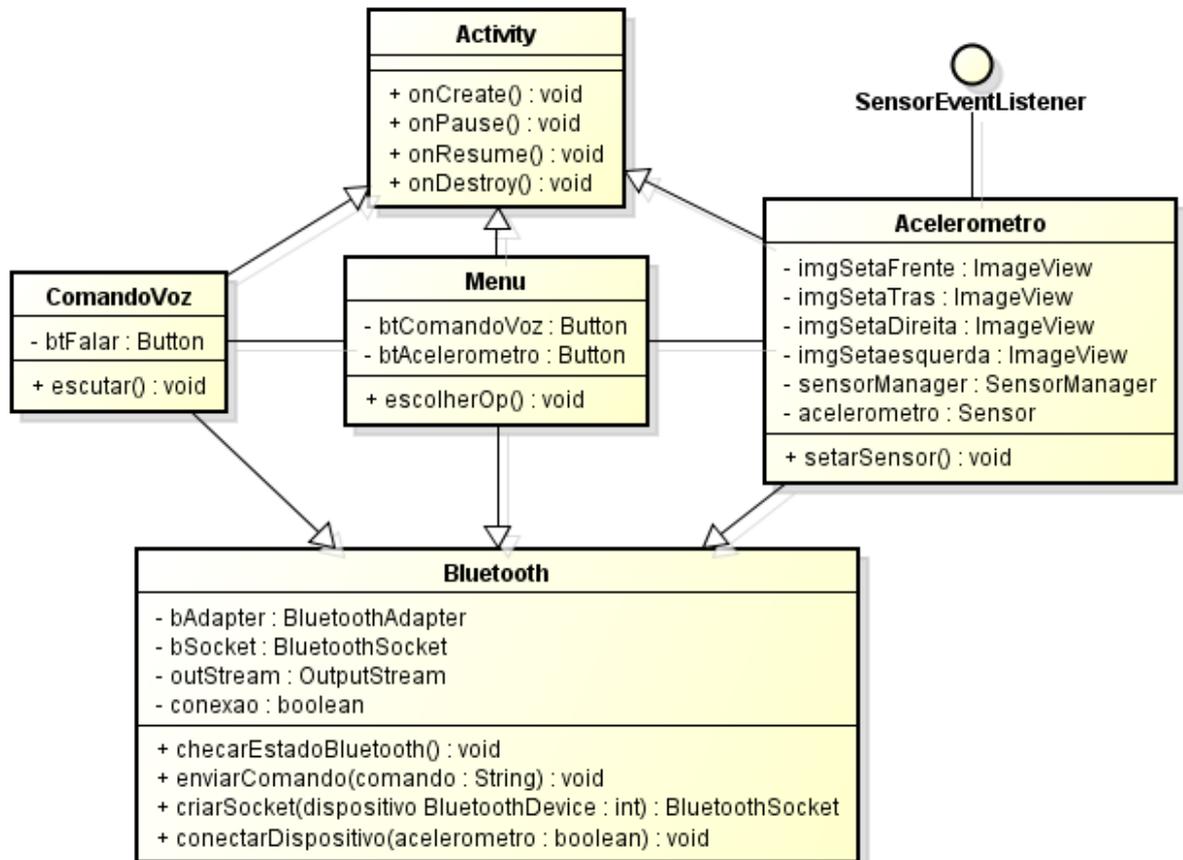
Fonte: Rodrigo Silva (2014)

O diagrama de classes representado pela figura 14 mostra a organização das classes da UI, mostrando as classes existentes e suas relações de consumo e herança, e a interface implementada pela classe Acelerômetro.

As telas principais são as telas que herdam a classe *activity*. Existe outra classe que não herda *activity*, que é a classe responsável pelo gerenciamento da conexão *bluetooth* com a CCR. Esta classe é apenas consumida pelas classes principais por possuir métodos que indicam, por exemplo, se o módulo *bluetooth* está ativo ou não, se a conexão com a CCR ocorreu corretamente, se o comando da direção foi enviado de forma correta. Essas indicações aparecem ao usuário para que este saiba o que está acontecendo com a aplicação e o motivo de algum erro.

A classe principal Acelerômetro implementa a interface *SensorEventListener*. Com isso a classe usa o método *onSensorChanged(SensorEvent evento)* pertencente à interface que realiza uma ação, envia à classe *bluetooth*, e em sequência à CCR, o comando referente à direção em que deve se mover, cada vez que o valor do sensor acelerômetro for alterado. Para que isso não sobrecarregue o sistema com envios de direções em excesso, foi estabelecido um valor mínimo de inclinação percebido pelo sensor para permitir o envio do comando.

Figura 14: Diagrama de classe da UI



Fonte: Rodrigo Silva (2014)

4.3 PROGRAMAÇÃO DO ARDUINO

Funcionando em paralelo com o aplicativo Android, existe um código sendo executado pela placa Arduino.

Uma placa Arduino consiste de um agrupamento de componentes eletrônicos em uma placa de circuito impresso caso não seja programada para realizar alguma função. A linguagem de programação é C/C++ com algumas adaptações, seguindo a metodologia de codificação baseada em funções (ARDUINO, 2013).

Existem duas funções primordiais em um código Arduino, a função `setup()` e a função `loop()`. A função `setup()` é executada uma vez, sempre que o Arduino é iniciado ou reiniciado. Serve para inicializar variáveis, configurar os pinos a serem usados como entrada ou saída de sinais e iniciar bibliotecas. A função `loop()` é executada após a `setup()` e faz a repetição de um trecho de código de forma

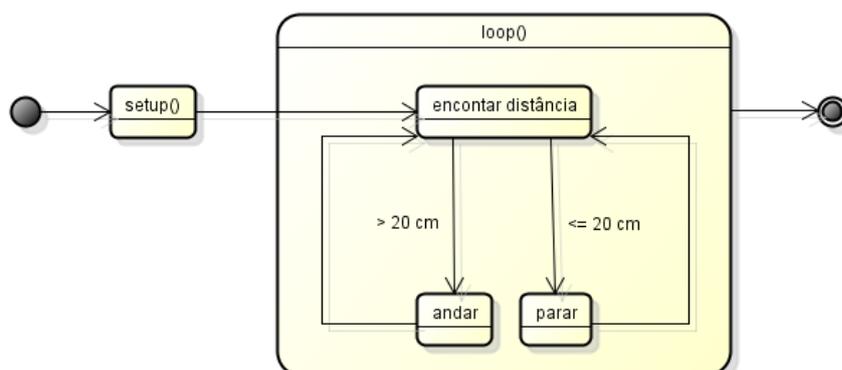
contínua que permite a execução do programa e o correto funcionamento da placa. Podem existir outras funções criadas pelo programador assim como em outras estruturas de código modularizadas. Essas funções são chamadas a partir da repetição da função *loop()*.

O Arduino se comunica com um módulo *bluetooth* e com um sensor de ultrassom. Estes componentes fornecem ao Arduino os meios necessários para controlar os motores, ao receber um comando do aplicativo no dispositivo móvel e parar, caso encontre um obstáculo em frente ao sensor.

O código implementado começa com a função *setup()*, onde é informado todos os pinos que serão usados e suas formas de trânsito de dados. Como esta função é executada uma vez ao iniciar a placa, é preciso informar o estado inicial de sinal para as portas que enviam o comando do Arduino para o circuito de acionamento dos motores. Isto serve para o sistema não iniciar em movimento.

Feito isto, é iniciada a função *loop()*. Esta função, primeiramente, verifica a distância de algum possível obstáculo à frente da CCR, através de uma leitura dos dados do sensor de ultrassom. Uma distância foi adotada arbitrariamente para simular uma distância segura de parada em caso de detecção de obstáculos. Caso seja percebido um obstáculo a uma distância igual ou menor que 20cm, o Arduino envia ao circuito de acionamento dos motores o sinal para não movimentá-los. Caso contrário, é chamada a função responsável por movimentar os motores. Esta função reconhece a direção do movimento que foi informada pelo dispositivo móvel e recebida pelo módulo *bluetooth* que está conectado ao Arduino, e gera os comandos necessários para que o circuito de acionamento mova a CCR na direção indicada. O esquema de funcionamento do programa do Arduino está representado na figura 15.

Figura 15: Diagrama de máquina de estados da programação do Arduino



Fonte: Rodrigo Silva (2014)

5 TESTES E RESULTADOS

Com todos os componentes conectados, o sistema estava finalmente pronto para a etapa de testes. Esta etapa visa estimar metas para pôr em prova as capacidades do sistema de realizar funções de uma cadeira de rodas.

As metas a serem testadas são:

- Teste de conexão com um dispositivo móvel (nos testes, um *smartphone*);
- Teste de controle por acelerômetro;
- Teste de controle por comando de voz;
- Teste de detecção de obstáculos.

5.1 TESTE DE CONEXÃO COM DISPOSITIVO MÓVEL *SMARTPHONE*

A primeira coisa a ser feita para a realização dos testes é a conexão entre a CCR e o dispositivo móvel. Nos testes realizados foi utilizado um *smartphone*. O módulo *bluetooth* acoplado à CCR possui um endereço para conexão único, um MAC, como em outros dispositivos de rede. Este endereço está embutido no programa Android.

Ao iniciar o aplicativo no aparelho Android é aberta a tela inicial. Até então, não existe conexão entre a interface e a aplicação. Ao tentar optar por uma das opções de controle o usuário é informado da necessidade de uso do *bluetooth* no aparelho, as opções de controle só podem ser acessadas com a opção de *bluetooth* ligada no aparelho.

O módulo *bluetooth* é passivo, ou seja, apenas aguarda que um dispositivo se conecte a ele. Para o usuário, a informação de conexão bem sucedida é informada de forma textual. No canto superior direito das telas de controle existe um texto informativo indicando “Conectado” na cor verde, caso o pareamento tenha sido bem sucedido, ou “Desconectado” na cor branca, caso contrário (ver figura 16). A Conexão é de forma automática e acontece enquanto o aplicativo está abrindo alguma das telas de controle (tela de comando de voz ou tela de controle por acelerômetro). Ao retornar para o menu do aplicativo a conexão é desfeita.

Figura 16: UI com informação de conexão

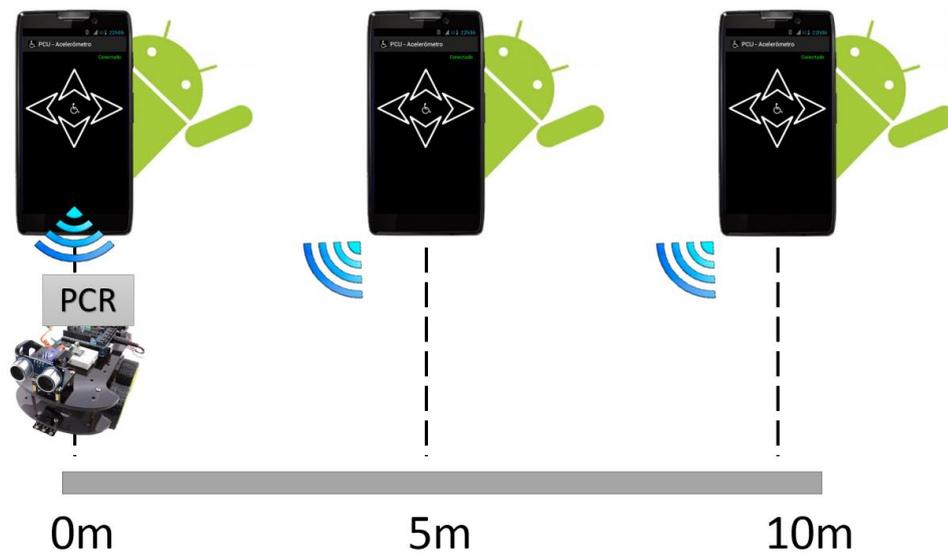


Fonte: Rodrigo Silva (2014)

Pode-se constatar o status da conexão no próprio módulo *bluetooth*. O módulo possui um LED indicador de conectividade, caso este LED esteja piscando, o módulo não está conectado, caso esteja aceso permanentemente, o módulo está conectado.

Os testes foram realizados com a CCR sobre uma superfície lisa, testando a comunicação entre interface e aplicação, considerando 3 distâncias de separação entre eles: 0m, 5m e 10m. A figura 17 esquematiza o experimento.

Figura 17: Teste de conexão



Fonte: Rodrigo Silva (2014)

De acordo com os resultados, o módulo *bluetooth* adotado se mostrou eficaz ao reconhecer bem os sinais do aplicativo Android, mesmo à distância máxima de reconhecimento do sinal para a categoria dele, 10m. Os testes foram feitos em um ambiente que não apresentava interferências para o sinal *bluetooth*. A conexão via *bluetooth* se mostrou bastante eficiente em enviar ao as direções de controle.

5.2 TESTE DE CONTROLE POR ACELERÔMETRO

A primeira opção de controle do aplicativo é o controle por acelerômetro. Para acessar esta opção é necessário que o aplicativo esteja em um dispositivo móvel que possua esta tecnologia, além de estar com o *bluetooth* do aparelho ligado.

Esta opção de controle permite que o usuário controle os movimentos da CCR apenas inclinando o dispositivo móvel na direção desejada (ver figura 18).

Este teste visa analisar a capacidade do controle por acelerômetro.

Figura 18: Teste de controle por acelerômetro



Fonte: Rodrigo Silva (2014)

O acelerômetro possui uma margem de movimento para não interpretar erroneamente uma indicação de direção, isto é, é preciso inclinar o dispositivo móvel em um valor maior que um grau determinado para o comando ser reconhecido, caso não haja uma inclinação maior que o grau determinado, é enviado o comando para parar os motores da CCR.

O acelerômetro indica o ângulo de inclinação do dispositivo em 3 eixos mas para o sistema foi explorado apenas 2 eixos. Quando o dispositivo se inclina para um dos quatro sentidos dos dois eixos, o comando é enviado à CCR fazendo-a se

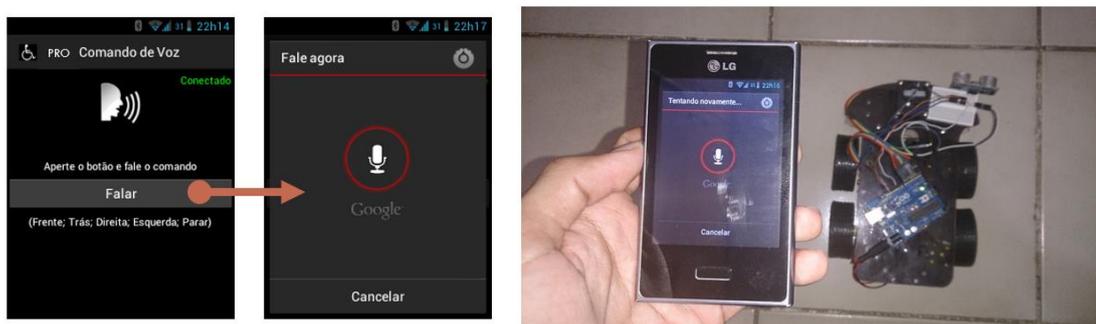
mover. Caso o dispositivo se incline em dois sentidos ao mesmo tempo, como para frente e para direita, o comando enviado é a primeira inclinação que ultrapassar o limite do grau determinado.

Não houveram problemas com esta forma de controle que se mostrou bastante confiável dentro da distância abrangente do *bluetooth*.

5.3 TESTE DE CONTROLE POR COMANDO DE VOZ

A segunda etapa de testes foi a verificação do controle usando o comando de voz. O comando de voz foi programado no aplicativo Android, usando um recurso nativo do Google para reconhecimento de voz, o *Google Speak Now* (GOOGLE, 2014), disponível nos aparelhos celulares com SO Android (ver figura 19). Este recurso usa uma conexão com a Internet para enviar o sinal gravado ao Google e este retorna uma lista de palavras semelhantes ao pronunciado. No aplicativo manipula-se o retorno da função para determinar o que foi dito. A primeira palavra da lista (entende-se que seja a mais parecida com a palavra dita originalmente) é comparada com as palavras que representam os sentidos do movimento, isto é, direita, esquerda, frente, trás e parar.

Figura 19: Teste de controle por comando de voz



Fonte: Rodrigo Silva (2014)

Caso o usuário fale algo que não seja reconhecido pelo aplicativo, é enviado o sinal de “parar” para a CCR e um alerta é exibido para o usuário, informando que a palavra não foi reconhecida.

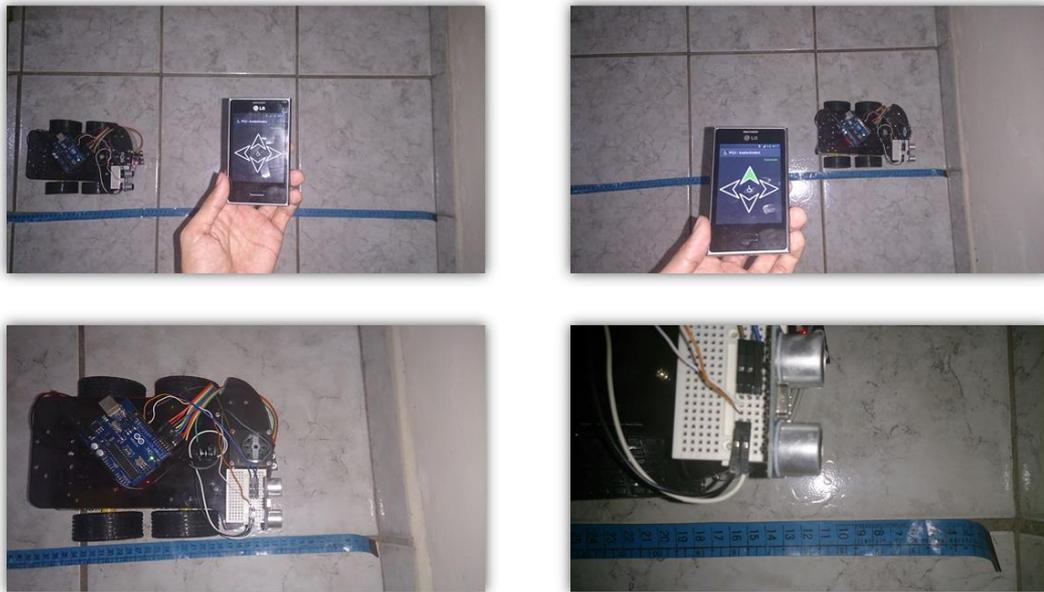
Os testes mostraram que diferentes sotaques, entoações muito diferentes ou qualquer dificuldade em pronunciar o comando resulta no não entendimento do comando. Seria preciso informar no código do aplicativo Android todas as palavras semelhantes aos respectivos comandos, a fim de comparação, só assim, mesmo que dito de forma quase incompreensível, o comando seria reconhecido. Porém existem muitas possibilidades e pronúncias e, portanto, houveram erros de compreensão durante os testes, quando os comandos não foram ditos precisamente.

5.4 TESTE DE DETECÇÃO DE OBSTÁCULOS

O teste final foi a detecção de obstáculos. O sistema detecta obstáculos por meio de um sensor ultrassônico acoplado ao Arduino e que fica posicionado na parte frontal da CCR. O objetivo dele é detectar quando um obstáculo está na sua área de visão, dentro de uma área de segurança de 20cm a 2cm de distância. Quando isto ocorrer, a CCR deve parar, evitando assim, que este colida com o objeto detectado.

O teste foi realizado enviando um comando de movimento à CCR, até que esta se encontrou a 20cm de um anteparo. Enquanto se aproxima o sensor constantemente calcula a distância de objetos a sua frente. Quando o sensor detectou o objeto a uma distância de 20cm, uma ordem de parar foi enviada à CCR. Porém, ainda leva alguns milissegundos para os motores pararem completamente, fazendo o teste variar minimamente a distância de parada. A figura 20 mostra a CCR percorrendo uma curta distância e, logo após parando perante um obstáculo.

Figura 20: Teste de detecção de obstáculos



Fonte: Rodrigo Silva (2014)

O sensor está localizado apenas na região frontal da CCR, isto limita a visão de obstáculos apenas para os que estão à sua frente. Novos sensores ultrassônicos podem ser adicionados aos outros lados da CCR, garantindo uma melhor visão do ambiente ao seu redor, e aprimorando a detecção de colisões.

Os testes mostraram que a inercia e a demora de reação dos motores ao comando do Arduino, fazem com que o sistema acabe parando mais próximo do obstáculo em comparação com os 20cm estabelecidos, ou seja, os milissegundos de tempo de resposta equivalem a centímetros percorridos (de até 8cm). Portanto, é preciso informar uma distância de parada maior que 20cm para garantir que o sistema pare na margem estabelecida.

5.5 COMPARATIVO DE PREÇOS

Com a montagem, programação e testes concluídos, pôde-se constatar que uma cadeira de rodas motorizada no mercado atual brasileiro custa consideravelmente mais caro do que uma cadeira de rodas motorizada construída baseada na CCR.

Ao pesquisar lojas que vendem cadeiras de rodas motorizadas no Brasil, é possível encontrar uma faixa de preço entre R\$ 7.749,00 (MARCAMÉDICA, 2013a) e R\$ 18.100,00 (MARCAMÉDICA, 2013b). Para comparativo, tomou-se o valor mediano barato, uma cadeira simples motorizada. Suas características são: 2 motores de CC com 400W cada, 2 baterias de 12V com 50A, controle por *joystick*, preço R\$ 8.467,00 (MARCAMÉDICA, 2013c).

Fazendo as devidas adaptações para se construir uma cadeira de rodas motorizada baseada no sistema da CCR, verificou-se que o preço dos componentes separadamente torna este sistema mais viável, quanto ao custo:

- Cadeira de rodas - R\$ 349,00 (FIGUEIREDO, 2014)
- 2 motores de CC com 400W - R\$ 145,00 cada
- 2 baterias de 12V com 50A - R\$ 300,00 cada (EMPÓRIO, 2014)
- Arduino UNO - R\$ 50,00
- Demais componentes (controladora de motores, sensor ultrassom, módulo *bluetooth*) - R\$ 200,00

Comparando os preços, a cadeira de rodas construída com base no sistema aqui descrito, custaria no mínimo R\$ 1.489,00. Os preços levantados foram obtidos pela média entre alguns preços. Os baixos custos obtidos com uma cadeira baseada na CCR justificam o motivo desta pesquisa e proporcionam comodidade aos que possuem limitações motoras, e dependem de cadeiras de rodas motorizadas.

6 CONCLUSÕES

Esta pesquisa usou-se da tecnologia de baixo custo proporcionada por componentes baratos e de código aberto (Arduino e Android), para construção de uma central de Controle de Cadeira de Rodas motorizada utilitária, com diferentes formas de controle e capaz de detectar obstáculos.

O sistema foi construído de forma a simular uma situação real de suporte às pessoas com deficiências de locomoção, porém em escala reduzida. Para isto, foram usados motores de menor potência e baterias proporcionais ao tamanho da CCR.

A primeira etapa da pesquisa foi o levantamento das tecnologias e projetos já existentes na área de suporte robótico à cadeirantes. Comparando com os demais projetos, pode-se constatar que o Arduino e o Android são plataformas bastante favoráveis para o desenvolvimento de soluções de baixo custo, no universo da acessibilidade e suporte aos deficientes.

A programação do sistema consistiu no desenvolvimento de uma interface gráfica para dispositivos móveis com o Android, onde o cadeirante pode movimentar sua cadeira através do aparelho via um canal de comunicação *bluetooth*.

Na etapa final da pesquisa, foram realizados os testes necessários para verificação do correto funcionamento do sistema. Metas de funcionamento foram adotadas e testadas. Foram testados a conexão *bluetooth* entre um dispositivo móvel e o sistema, as opções de controle disponíveis (por acelerômetro e por comando de voz), e por fim, a detecção de algum obstáculo à frente.

Pode-se concluir com esta pesquisa, que o sistema pode ser base para uma solução de baixo custo de cadeira de rodas motorizadas, se comparado às cadeiras existentes no mercado atual.

Esta pesquisa deu início à uma central de controle que pode ser ampliado por meio de *software* e *hardware*. Pesquisas futuras podem acrescentar novas formas de controle da CCR e novos recursos de sensores ou atuadores. Isto é possível graças ao núcleo de código aberto em Android e por ser construído com a plataforma Arduino.

7 REFERÊNCIAS

ANDROID. Disponível em <<http://developer.android.com/about/index.html>>. Acesso em: 23/08/2013

ARDUINO. Disponível em <<http://www.arduino.cc/>>. Acesso em: 25/08/2013

EMPÓRIO. Disponível em <http://www.emporiodasbaterias.com.br/ecommerce_site/categoria_98-115_6108_BATERIAS-AUTOMOTIVAS-De-50Ah-Amperes>. Acesso em: 30/07/2014

FIGUEIREDO. Disponível em <<http://www.figueiredohospitalar.com.br/p/cadeira-de-rodas-com-rodas-em-nylon-1009/>>. Acesso em: 28/12/2013

GOOGLE. Disponível em <<https://support.google.com/websearch/answer/2940021?hl=pt-BR>>. Acesso em: 31/07/2014

GOMES Janyne, DUARTE Jhonatan: **Freedom**, Governador Valadares. Disponível em: <<http://www.fandroid.com.br/2011/05/12/o-projeto-freedom/>>. Acesso em: 20 mar. 2013.

HOCKEY A. B.; MILLER D. P.. A demonstration of a conversationally guided smart wheelchair. In: 9^a Conferência Internacional ACM SIGACCESS em Informática e acessibilidade, 2007, Nova Iorque. Anais. Nova Iorque: ACM, 2007. p.243-244.

LECHETA, Ricardo R. Google Android para tablets: aprenda a desenvolver aplicações para o Android – de smartphones a tablets / Ricardo R. Lecheta. São Paulo : Novatec Editora, 2012

MARCAMÉDICA. Disponível em <<http://www.marcamedica.com.br/cadeiras-de-rodas/1058-cadeira-de-rodas-motorizadas-freedom-s.html>>. Acesso em: 27/06/13a

MARCAMÉDICA. Disponível em <<http://www.marcamedica.com.br/cadeiras-de-rodas/1055-cadeira-de-rodas-motorizadas-freedom-millenium-rt.html>>. Acesso em 27/06/2013b

MARCAMÉDICA. Disponível em <<http://www.marcamedica.com.br/cadeiras-de-rodas/1063-cadeira-de-rodas-motorizadas-freedom-sx.html>>. Acesso em: 27/06/2013c

MORAN, T. (1981) "The Command Language Grammars: a representation for the user interface of interactive computer systems. Em International Journal of Man-Machine Studies 15:3-50, Academic Press.

NASCIMENTO, Pablo de Melo. **Consulta de placas de veículos através da plataforma Android/porta serial**. 2012. 63 f. Monografia (Bacharelado) - Departamento de Informática e Matemática Aplicada - Dimap, Universidade Federal do Rio Grande do Norte - Ufrn, Natal, 2012.

REIS, Luis Paulo et al. IntellWheels: Modular development platform for intelligent wheelchairs. **JRRD: Journal of Rehabilitation Research & Development**, Porto, p. 1061-1076. 13 mar. 2012. Disponível em: <<http://www.rehab.research.va.gov/jour/11/489/braga489.html>>. Acesso em: 20 mar. 2013.

RIBEIRO, Maria Isabel: **Sensores em robótica: como é que os robots sentem o ambiente envolvente**. Disponível em: <users.isr.ist.utl.pt/~mir/pub/sensores.pdf>. Acessado em: 18 mar. 2013.