

## Montando um Robô de Baixo Custo

Alexander Hugo Tártari<sup>1</sup>, Adriana Postal<sup>1</sup>, Josué Pereira de Castro<sup>1</sup>

<sup>1</sup>UNIOESTE - Universidade Estadual do Oeste do Paraná

Laboratório de Robótica Inteligente

Rua Universitária, 2069. Jardim Universitário.

Caixa Postal 711 - CEP 85819-110 Cascavel, PR

ugoinfo@gmail.com adriana.postal@unioeste.br  
josue.castro@unioeste.br

**Resumo.** *Este artigo descreve a montagem de um robô com peças de baixo custo para fins didáticos. O robô conta com um conjunto de sensores para obter informações do ambiente, um par de motores para movimentação, e um microcontrolador que lhe permite operar de modo autônomo. O objetivo do trabalho é servir de guia para aqueles que desejam montar um robô, porém contam com pouco ou nenhum conhecimento sobre o assunto.*

### 1. Introdução

Segundo Mataric [1], um robô móvel é um dispositivo mecânico montado sobre uma base não fixa, que age sob o controle de um sistema computacional, equipado com sensores e atuadores que o permitem interagir com o ambiente.

Os robôs provam a cada dia o seu valor para o desenvolvimento da sociedade, tendo influência direta e indireta na maioria de seus aspectos. Com a popularização da robótica, cada vez mais tecnologias e conhecimento sobre o assunto estão disponíveis para quem queira se aprofundar no assunto. O objetivo deste artigo é mostrar que é possível construir, com informações disponíveis em fontes públicas e peças de baixo custo, um robô programável simples, que consegue obter informações pertinentes do mundo à sua volta, interpretá-las, e usá-las para então atuar de alguma forma no ambiente. Para isso, foi construído um robô que se movimenta através de rodas, e capta informações do ambiente através de um sensor ultra-sônico. Foram usadas, quando possível, peças de sucata. Quando a reciclagem de material não foi possível, tentou-se usar materiais facilmente encontrados em lojas, físicas ou virtuais, e cujos custos fossem baixos.

Na seção 2 são descritas as partes que compõem o hardware do robô, e as interligação desses componentes. A seção 3 mostra como são feitas as leituras dos sensores, e a movimentação dos motores. Na seção 4 são relacionados os custos da construção. Na seção 5 algumas aplicações práticas para o uso do robô são apresentadas. A seção 6 traz as conclusões sobre o trabalho, assim como os planos de aplicações futuras do robô construído.

## 2 Hardware

O robô é composto por várias partes fundamentais ao funcionamento do conjunto, como visto na Figura 1. Nesta seção cada uma dessas peças será descrita.

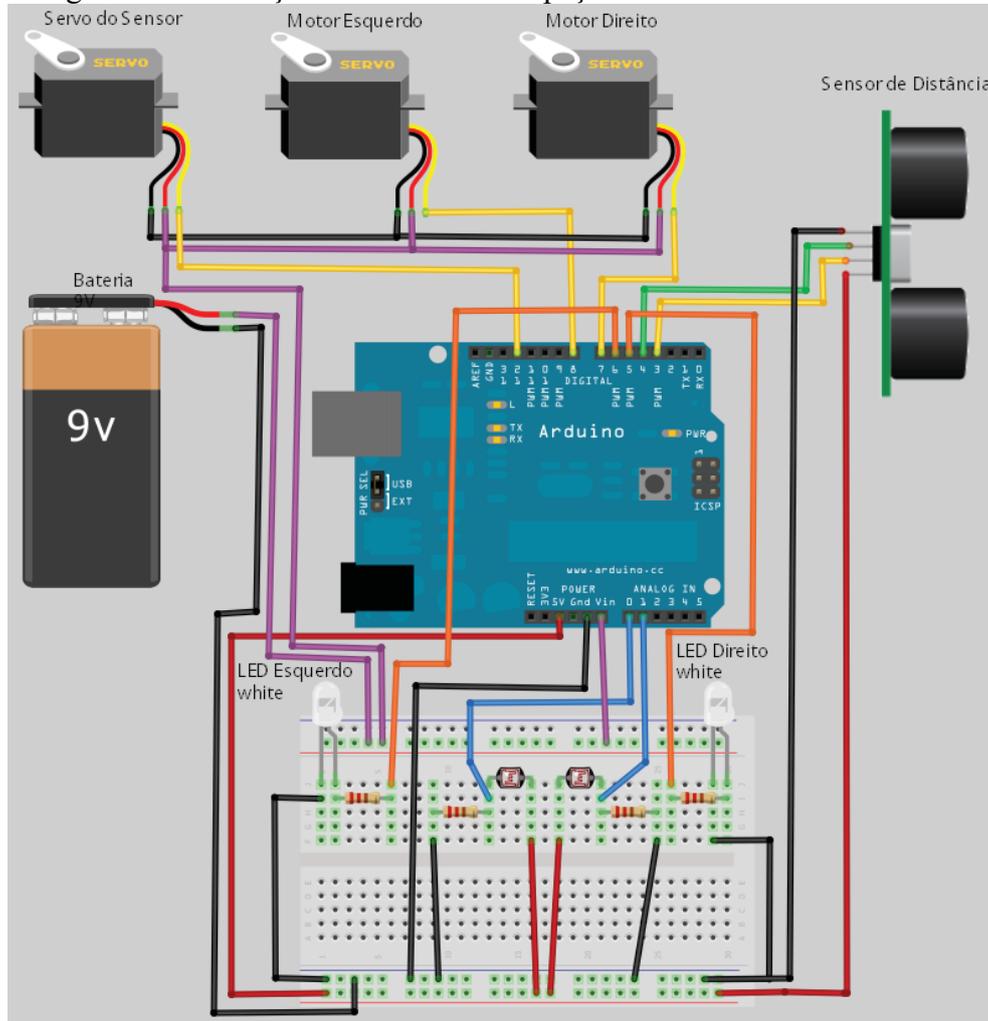


Figura 1. Componentes do Robô e suas Interligações

### 2.1. Arduino

Arduino [2] foi a plataforma escolhida para receber as entradas, interpretá-las, e definir a saída. Esta plataforma foi escolhida pelo baixo custo, facilidade de programação, e facilidade de compra.

O Arduino é uma plataforma de hardware livre. A placa do Arduino consiste em um microcontrolador Atmel AVR de 8 bits, e componentes auxiliares que facilitam a comunicação, programação, e a incorporação de outros componentes [3]. Existem ainda placas compatíveis com o Arduino, porém com microcontroladores alternativos, como a placa Microchip chipKit, que utiliza um microcontrolador de arquitetura MIPS de 32 bits [4]. Sua linguagem de programação é baseada na linguagem Wiring, e é bastante similar às linguagens C e C++ [5].

Projetado por um grupo de pesquisadores da cidade de Ivrea, na Itália [6], o Arduino foi lançado em 2005, e desde então já teve várias versões de hardware [3] e software [7]. Teve grande repercussão, sendo muito usado em projetos eletrônicos recentes, principalmente científicos e artísticos [8].

A grande vantagem desta plataforma é a facilidade de programação e interligação de componentes, permitindo que pessoas com pouco conhecimento em eletrônica e programação possam criar dispositivos complexos, e também que o utilizador possa focar mais em seus objetivos, preocupando-se menos com os meios.

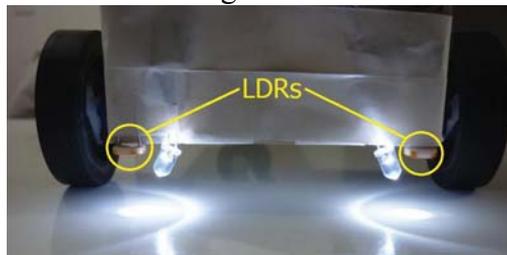
A placa do Arduino tem um conjunto de pinos, divididos em três tipos: alimentação, analógicos e digitais. O número de pinos analógicos e digitais varia de acordo com o modelo, e podem servir tanto como entrada, como saída. Alguns pinos digitais podem trabalhar em modo PWM (no português MLP – Modulação por largura de pulso) [9]. O Arduino conta ainda com uma porta USB, que pode ser usado para fornecimento de energia, e no caso de ser conectado a um dispositivo USB *host*, pode transmitir informações via comunicação serial [3].

## 2.2. Sensor de Distância

Foi usado o sensor ultra-sônico HC-SR04 [10], um sensor de distância de baixo custo encontrado facilmente em lojas virtuais. O sensor emite um som de alta frequência, que reflete em objetos, e retorna ao sensor. O tempo que leva para que o som retorne é usado para estimar a distância entre o sensor e o objeto que fez o som refletir. Observa-se ainda que este sensor pode ser substituído sem grandes problemas por um sensor infravermelho, que tem funcionamento parecido.

## 2.3. Sensores de Luminosidade

Para que o robô possa se orientar através de linhas no chão, foram adicionados dois sensores de luminosidade (LDR - Light Dependent Resistors), posicionados na frente do robô, em lados opostos, como visto na Figura 2.



**Figura 2. Posicionamento dos LDRs**

As linhas usadas para orientar robôs seguidores de linha devem ter um grande contraste em relação ao resto do chão. O robô detecta a presença dessas linhas por ocorrência de variações bruscas na luminosidade.

Os LDRs são sensores baratos, úteis e fáceis de usar. Um LDR é uma resistência que varia de acordo com a luz que recebe. Como mostrado na Figura 1, os LDRs são conectados ao Arduino com o auxílio de um resistor *pull-down* [11], que é ligado entre o

terra e a entrada analógica do LDR a ser lido. Através da entrada analógica, podemos medir a tensão elétrica que passa pelo LDR.

Também foram adicionados dois LEDs brancos para auxiliar na iluminação, visto que os LDRs são dependentes de uma boa iluminação. Os LEDs foram conectados em pinos que permitem o uso de PWM, podendo assim ajustar o seu brilho de acordo com a iluminação disponível no ambiente.

#### 2.4. Motores

A princípio pretendia-se usar motores retirados de sucada para movimentar as rodas. Para isso, motores DC comuns teriam que ser usados. Esta ideia foi descartada pois os motores DC comuns precisam dos seguintes complementos:

- Ponte H para controle do sentido da rotação [12]. Apesar de ser barata e fácil de encontrar, a ponte H deixaria o projeto mais complexo.
- Redutor, para aumento do torque [13]. Sem o redutor o motor provavelmente não teria força suficiente para movimentar as rodas. Estes redutores são relativamente difíceis de encontrar e ajustar, o que os tornam inviáveis.

A solução então foi usar servomotores, os quais já têm redução de rotação, e não necessitam de uma ponte H para controle [14]. Servomotores são facilmente encontrados, pois são muito usados em modelismo, principalmente aéreo.

Para que o motor consiga movimentar seu eixo em rotação contínua, é preciso executar duas modificações: a remoção de um limitador em uma das engrenagens, e a substituição do potenciômetro interno por duas resistências fixas. Este processo pode ser visto em Kurt [15]. Estas modificações são relativamente complexas, mas são mais viáveis que a opção de usar motores DC comuns.

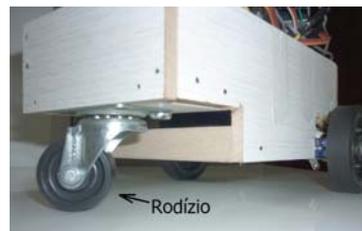
Neste projeto é também utilizado um servomotor sem modificações para movimentar o sensor de distância.

#### 2.5. Rodas

As rodas usadas são parte de uma impressora que já não funcionava mais. Na impressora, elas eram responsáveis por puxar o papel, função que exige um material com boa aderência. Essa qualidade faz com que essas rodas sejam ideais para o uso em locomoção. Um braço do servomotor foi fixado na roda, de maneira centralizada, como pode ser visto na Figura 3(a).



(a) Roda com o braço do servomotor



(b) Rodízio Traseiro

**Figura 3. Rodas do Robô**

Como visto na Figura 3(b), há uma pequena roda (rodízio) na parte traseira do robô, que serve como apoio, não gerando força, e movimenta-se de acordo com o movimento do robô.

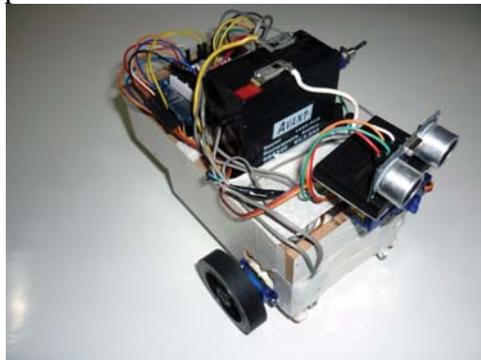
## 2.6. Bateria

Foi usada uma bateria de lâmpada de emergência, ideal ao projeto devido ao seu baixo custo. A tensão da bateria é de 6 volts, conseguindo assim alimentar os servomotores e o Arduino. O sensor de distância é alimentado pela saída de 5 volts do Arduino, a qual é regulada, para evitar problemas com a variação de tensão. Para recarregar a bateria, foi usada a estrutura da própria lâmpada de emergência, que tem um carregador interno.

Como o Arduino tem um regulador de tensão interno, é possível usar baterias entre 6 e 20 volts [3], desde que essa bateria possa fornecer corrente suficiente para alimentar o conjunto todo. Porém para tensões muito acima dos 6 volts é necessário usar um regulador de tensão para os servos, que geralmente operam na faixa de 3 a 5 volts [14].

## 2.7. Estrutura

A estrutura foi feita de madeira, na forma de uma caixa com furos para o posicionamento dos motores, e um recorte para o posicionamento do rodízio, como pode ser visto na Figura 4. Nesta parte do projeto a criatividade vai definir o rumo visual, enquanto deve-se ainda levar em conta o posicionamento de cada parte do robô, de forma que a organização das partes fique clara e fácil.



**Figura 4. Visão Geral do Robô**

## 3. Software

Para que o robô funcione de forma autônoma é preciso programá-lo de modo que o mesmo capte as informações pertinentes, processe-as, e atue no ambiente. Nesta seção descreveremos como são obtidos os dados de entrada e também como controlar a movimentação das rodas.

Vídeos do robô funcionando podem ser encontrados em [16].

### 3.1. Leitura do Sensor de Distância

Para obter a distância detectada pelo sensor de distância, foi utilizada a biblioteca “Ultrasonic.h” [17]. Para usá-la, cria-se uma instância da classe *Ultrasonic*, passando os pinos de *Trigger* e *Echo* correspondentes. Para obter a distância medida, utiliza-se o método “Ranging(CM)”, que retorna um dado do tipo “long” com a distância medida. Um exemplo dessa leitura é mostrado no Algoritmo 1.

---

#### Algoritmo 1 Leitura do Sensor de Distância

---

```
#include "Ultrasonic.h"
Ultrasonic ultrasonico(4,3); // TRIG = pino 4 - ECHO = pino 3
void setup() {}
void loop(){ long distancia = ultrasonico.Ranging(CM); }
```

---

### 3.2. Leitura do Sensor de Luminosidade

A entrada dos LDRs é lida com o comando “analogRead(pinoLDR)”, onde “pinoLDR” é o número do pino em que o sensor está conectado, no caso deste projeto, 0 ou 1.

Para acender os LEDs, devemos primeiro declarar o pino do LED como saída no bloco “setup()”, usando o comando “pinMode(pinoLED,OUTPUT)”, onde “pinoLED” é o número do pino em que o LED está conectado, no caso deste projeto, 5 ou 6. Após feito isso, temos duas opções. A primeira é acendê-lo de forma digital, usando “digitalWrite(pinoLED, HIGH)”. A segunda opção é usar PWM, passando um valor entre 0 e 255, que corresponde à luminosidade do LED. Isso é feito com o comando “analogWrite(pinoLED, valor)”. No algoritmo 2 é mostrado o acendimento dos leds de forma digital, e em seguida a leitura dos sensores de luminosidade.

---

#### Algoritmo 2 Acendimento dos LEDs e Leitura do Sensores de Luminosidade

---

```
int ldrDireito = 1;
int ldrEsquerdo = 0;
int leituraLdrDireito, leituraLdrEsquerdo;
void setup() {
    pinMode(5, OUTPUT); //Define o pino 5 como saída
    pinMode(6, OUTPUT); //Define o pino 6 como saída
    digitalWrite(ledDireito, HIGH); //Acende o led direito
    digitalWrite(ledEsquerdo, HIGH); //Acende o led esquerdo
}
void loop(){
    leituraLdrDireito = analogRead(ldrDireito);
    leituraLdrEsquerdo = analogRead(ldrEsquerdo);
}
```

---

### 3.3. Movimentação dos Motores

Os servos utilizam a biblioteca “Servo.h”, que já vem junto com a IDE do Arduino. Deve-se criar uma instância da classe *Servo*, e no bloco “setup()” atribuir o pino

certo com o método “attach”. Para movimentar o servo usa-se o método “write(ângulo)”, onde ângulo é um inteiro de 0 a 180. No Algoritmo 3 é mostrado um exemplo onde o servomotor do sensor, em loop, movimenta-se de 0° a 180°, e depois volta a 0°.

---

### Algoritmo 3 Movimentação de um Servomotor

---

```
#include <Servo.h>
Servo servoSensor; //cria um objeto da classe Servo
void setup(){ servoSensor.attach(12); } //pino do servo = 12
void loop(){
  for(int pos = 0; pos < 180; pos += 1){ //vai de 0° até 179°
    servoSensor.write(pos);
    delay(15);
  }
  for(int pos = 180; pos>=1; pos--=1){ //vai de 180° até 1°
    servoSensor.write(pos);
    delay(15);
  }
}
```

---

Depois de alguns testes, verificou-se que cada servo modificado comporta-se de maneira diferente. Para que os dois motores girassem na mesma velocidade, foi preciso testar empiricamente. Para isso, foi passado primeiro o valor 0 para os dois servomotores, através do método “write”. Verificou-se então que os dois giravam em sentido horário.

Como estão em posições opostas, para que o robô ande para frente, o motor direito deve girar no sentido horário, enquanto o motor esquerdo deve girar em sentido anti-horário. Foram então testados vários valores para o motor esquerdo, para que ele girasse no sentido anti-horário, com a mesma velocidade do motor direito em sentido horário. O valor encontrado foi 30. Os mesmos testes foram feitos para fazer o robô andar para trás, e também para ficar parado. O resultado pode ser visto na Tabela 1.

Sentido	Motor Esquerdo	Motor Direito
Frente	30	0
Para	14	11
Trás	0	20

**Tabela 1. Valores Aplicados ao Motores**

## 4. Custos

Devido ao foco econômico do projeto, a Tabela 2 foi montada, com os custos dos materiais utilizados. Isto permite uma avaliação da viabilidade econômica.

Nome	Quant.	Custo/Unidade	Custo Total
Arduino Uno	1	R\$ 90,00	R\$ 90,00
Protoshield	1	R\$ 25,00	R\$ 25,00
Sensor de Distância HC-SR04	1	R\$ 45,00	R\$ 45,00
Lâmpada de Emergência Avant	1	R\$ 20,00	R\$ 20,00
Micro Servomotor	3	R\$ 13,00	R\$ 39,00
LDR	2	R\$ 2,00	R\$ 4,00
LED Branco	2	R\$ 0,50	R\$ 1,00
Resistor 330 Ohms	4	R\$ 0,10	R\$ 0,40
Jogo de Jumpers para Protoboard	1	R\$ 12,00	R\$ 12,00
Rodas de Borracha (Sucata)	2	R\$ 0,00	R\$ 0,00
Rodízio	1	R\$ 5,00	R\$ 5,00
Caixa de Madeira (Sucata)	1	R\$ 0,00	R\$ 0,00
Total			R\$ 241,40

**Tabela 2. Custo do Projeto**

Devido ao baixo custo, à possibilidade de customização, e ao aprendizado adquirido no processo, observou-se que a montagem traz muitos benefícios em relação à compra de um kit já pronto, como o Lego® Mindstorms® NXT 2.0, que em sua loja oficial custa USD 279,99 [18], cerca de R\$ 500,00, sem considerar-se o transporte e a taxa de importação.

É importante notar que, como o Arduino é uma plataforma de hardware aberto, é possível encontrar versões mais baratas, algumas delas produzidas no Brasil, que chegam a custar até 30% menos. É possível também fabricar a sua própria placa, como por exemplo o Severino [19], um projeto brasileiro que usa componentes mais simples que o Arduino original, tornando possível sua fabricação caseira.

## 5. Aplicações

Existem várias aplicações práticas de robôs com características comuns ao montado neste trabalho. Os sensores de luz podem ser usados para seguir linhas desenhadas no chão. Este tipo de orientação é bastante utilizado em campeonatos de robótica [20], e principalmente em robôs autônomos que transportam objetos dentro de empresas [21]. Guiando-se através das linhas, é possível ao robô locomover-se por vários lugares sem se perder ou entrar em lugares inapropriados. Já com o sensor de distância, podemos encontrar objetos, evitar obstáculos, assegurar-se que o robô não irá bater em algo, verificar o melhor caminho a seguir, dentre muitas outras possibilidades. É possível também mapear um ambiente estruturado, medindo as distâncias até as paredes ou obstáculos, assunto que será discutido na seção 8.

## 6. Conclusões e trabalhos futuros

A montagem de um robô como este é uma ótima maneira de aprender e ensinar eletrônica e computação, até mesmo para quem não tem nenhum contato com estas áreas. No processo de montagem, consegue-se ter uma visão clara de como o robô funciona, e como suas peças interagem entre si. Além dos fins didáticos, temos também várias aplicações práticas, e diversos tipos de estudos, que não seriam possíveis sem um robô como este.

O robô montado apresentou o desempenho esperado, conseguindo seguir trilhas brancas em um chão escuro, e também desviar de objetos detectados pelo sensor de distância. Com o robô, será possível um estudo mais aprofundado em mapeamento de ambientes estruturados, como o SLAM - Simultaneous Localization And Mapping [22]. Para este estudo, o robô deve medir as distâncias através do sensor, e também medir as distâncias de seus movimentos. Porém, para isso, o robô deve saber com exatidão as distâncias de seus movimentos. O sensor que faz essa medição será o próximo passo no aprimoramento do robô.

## Referências

- [1] Mataric, M. J., “Integration of representation into goal-driven behavior-based robots”. IEEE Transactions on Robotics and Automation 8, 3, Junho 1992, 304–312.
- [2] Arduino, “Arduino – Home Page”, Disponível em: <http://arduino.cc/> , Acesso em: Agosto 2011, 22.
- [3] Arduino, “Arduino – Hardware”, Disponível em: <http://arduino.cc/en/Main/Hardware> , Acesso em: Agosto 2011, 22.
- [4] Microchip, “chipKit Development Platform”, Disponível em: <http://www.microchip.com/pagehandler/en-us/chipKIT-Development-Platform.html> , Acesso em: Setembro 2011, 20.
- [5] Arduino, “Arduino – Language Reference”, Disponível em: <http://arduino.cc/en/Reference/HomePage> , Acesso em: Agosto 2011, 22.
- [6] Lahart, J., “Taking an Open-Source Approach to Hardware”, Disponível em: <http://online.wsj.com/article/SB10001424052748703499404574559960271468066.html>, Acesso em: Agosto 2011, 22.
- [7] Arduino, “Arduino – Software”, Disponível em: <http://arduino.cc/en/Main/Software> , Acesso em: Agosto 2011, 22.
- [8] Hack N Mod, T., Portnoi, M., , “Top 40 Arduino Projects of the Web”, Disponível em: <http://hacknmod.com/hack/top-40-arduino-projects-of-the-web/> , Acesso em: Setembro 2011, 19.
- [9] Moreira, A., Negreiros, T., Portnoi, M., , “Modulação PWM”, Disponível em: <http://www.eecis.udel.edu/~portnoi/academic/academic-files/pwm.html> , Acesso em: Setembro 2011, 19.

- [10] Itead Studio, “HC-SR04 Datasheet”, Disponível em: <http://iteadstudio.com/store/images/produce/Sensor/HCSR04/HC-SR04.pdf> , Acesso em: Agosto 2011, 22.
- [11] Seattle Robotics Society, “Very Basic Circuits”, Disponível em: <http://www.seattlerobotics.org/encoder/mar97/basics.html> , Acesso em: Setembro 2011, 19 .
- [12] Margolis, M., Arduino Cookbook, Vol. 1, cap. 8, O'Reilly Media, Sebastopol CA – Estados Unidos, 1ª ed., março 2011, p. 267-272.
- [13] Lima, A. A. L., “Como Projetar e Construir um Redutor de Velocidade”, Disponível em: <http://www.aviacaoexperimental.pro.br/aero/tecnica/motores/redutor.pdf> , Acesso em: Agosto 2011, 22.
- [14] Francisco, A. M. S., “Servomotores”, Disponível em: <http://amsfrancisco.planetaclix.pt/download/Robotica/Servomotores.pdf> , Acesso em: Agosto 2011, 22.
- [15] Kurt , T. E., “Tiny Servos as Continuous Rotation Gearmotors”, Disponível em: <http://todbot.com/blog/2009/04/11/tiny-servos-as-continuous-rotation-garmotors/>, Acesso em: Agosto 2011, 22.
- [16] Tartari, A. H., “Blog Robot Stuff”, Disponível em: <http://robotstuff.wordpress.com> , Acesso em: Agosto 2011, 22.
- [17] Itead Studio, “Arduino Library For Ultrasonic Ranging Module HC-SR04”, Disponível em: <http://iteadstudio.com/application-note/arduino-library-for-ultrasonic-ranging-module-hc-sr04/> , Acesso em: Agosto 2011, 22.
- [18] Lego Shop, “LEGO® MINDSTORMS® NXT 2.0”, Disponível em: <http://shop.lego.com/en-US/LEGO-MINDSTORMS-NXT-2-0-8547> , Acesso em: Setembro 2011, 20.
- [19] Akashi , A., “Arduino Single-Sided Serial Board (version 3)”, Disponível em: <http://www.arduino.cc/en/Main/ArduinoBoardSerialSingleSided3> , Acesso em: Agosto 2011, 22.
- [20] Latinoware, “Regulamento – Olimpíada Latino Americana de Robótica Livre - Latinoware 2010”, Disponível em: [http://www.robotizando.com.br/cp2011/regulamento\\_2010\\_olimpiada\\_RPL\(1\).pdf](http://www.robotizando.com.br/cp2011/regulamento_2010_olimpiada_RPL(1).pdf) , Acesso em: Agosto 2011, 22.
- [21] Magalhães, L. S., Arantes, J. C. S., Teixeira, I., “Abastecimento de Estoque por Robô Microcontrolado”, Anuário da Produção de Iniciação Científica Discente, Vol. XI, No. 12, 2008, pp. 695-705.
- [22] Riisgaard, S., Blas M. R., "SLAM for Dummies A Tutorial Approach to Simultaneous Localization and Mapping", Disponível em: [http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam\\_blas\\_repo.pdf](http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_blas_repo.pdf) , Acesso em: Agosto 2011, 22.