

ERUS - Team Description Paper - IEEE Open 2012

Roberta Lima Gomes¹, André Georghthon Cardoso Pacheco¹, Lucas Catabriga Rocha¹,
Juan França Muniz de Souza¹, Ebenézer Nogueira da Silva¹,
Guilherme Artém dos Santos¹, Jackson Willian Brito¹, Ivan de Oliveira Nunes¹ e Vitor Buback Covre¹

Centro Tecnológico, UFES
Av. Fernando Ferrari 514 - Goiabeiras, Vitória-ES

Abstract—This article reports the work of the team ERUS building a robot to solve the challenge posed in IEEE Open 2012 category. The robot collects empty cans on an irregular terrain made of sand and deposits them in a trash bin. The system built uses an Android phone connected to an Arduino board.

Abstract—Este artigo relata o trabalho desenvolvido pela equipe ERUS na construção de um robô para solucionar o desafio proposto na categoria IEEE Open 2012. O robô construído coleta latas de refrigerante em um terreno irregular feito de areia e as deposita em uma lixeira. O sistema construído é composto por um celular Android conectado a uma placa Arduino.

I. INTRODUÇÃO

O desafio IEEE OPEN 2012 proposto na Competição Latino Americana de Robótica (Latin American Robotics Competition - LARC) tem como temática a limpeza do lixo em uma praia. O terreno da competição consiste de um círculo de areia cercado por uma região azul que representa o mar. Neste terreno um robô deve coletar latas de alumínio espalhadas e deposita-las em uma lixeira. O robô não pode tocar nos obstáculos presentes, compostos por uma cadeira, uma sombrinha de praia e um manequim.

A Equipe ERUS (Equipe de Robótica da UFES) construiu um robô para solucionar este desafio. O robô possui esteiras para navegar no ambiente irregular e sua estrutura física foi construída com peças de plástico, acrílico e peças de LEGO. A eletrônica do robô tem como base uma placa Arduino. Um celular Android é usado como computador central e para captura e processamento de imagens. As linguagens de programação utilizadas foram C, C++ e Java.

A equipe é formada por alunos de vários períodos diferentes do curso de Engenharia de Computação da Universidade Federal do Espírito Santo.

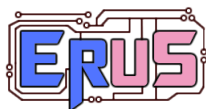


Fig. 1. Logotipo da ERUS

II. OBJETIVO

Este desafio tem como objetivo a construção e programação de um robô capaz de trabalhar de forma autônoma. O robô deve ser capaz de identificar onde estão localizados a cadeira, o manequim, o guarda sol, a lixeira e as latas de alumínio. Além de identificar, é necessário desviar dos obstáculos, recolher as latas e depositá-las no lugar especificado para o lixo.

III. CONSIDERAÇÕES DO AMBIENTE

O cenário proposto é simulado por uma arena com areia e demarcada pelo mar, identificado pela cor azul. A arena total possui 6 metros quadrados, destes 5,5 metros quadrados são compostos de areia e o mar é representado pela cor azul no restante da arena.

Na arena são colocadas latas de alumínio pintadas de preto, um manequim, uma cadeira, um guarda sol e um depósito de lixo. As latas de alumínio devem ser recolhidas e depositadas no depósito de lixo pelo robô.

As latas estão dispostas de forma aleatória, como mostrado na Figura 2.



Figure 1. Scenario of the Open category – Beach Cleaner Robot

Fig. 2. Representação da arena

Outro fator importante é a luminosidade, a arena pode estar em um ginásio fechado quanto em um espaço aberto, exposta a luz do sol.

IV. PROCEDIMENTOS

O robô construído possui dois sistemas principais, um composto pela placa Arduino e outro pelo celular Android

¹ Departamento de Informática da UFES

[1]. Estes dois sistemas se comunicam através de um cabo USB. É possível ainda conectar um computador ao celular através de uma rede sem fio, porém este é um modo apenas para testes e não é utilizado quando o robô opera de forma completamente autônoma.



Fig. 3. ErusBot

A. Estrutura do robô

A estrutura física do robô possui 3 partes importantes: garra, caçamba e a base. Cada uma destas será detalhada a seguir.

- 1) Garra: Construída com o kit Lego Mindstorms RCX [2].
- 2) Caçamba: Formada por acrílico e possui uma comporta na parte traseira para despejar as latas.
- 3) Base: Utilizado o protótipo Rover 5 [3] como principal. Esta parte é responsável pela sustentação do robô. Onde fica localizado o celular e também os sensores de ultrassom.

B. Navegação

A navegação dos robôs é realizada com auxílio de quatro tipos de sensores: ultrassom, câmera, bússola e acelerômetro (os três últimos fornecidos pelo celular Android). Os sensores de ultrassom identificam os objetos próximos ao robô, a bússola retorna as posições geográficas norte, sul, leste e oeste, o acelerômetro retorna . Juntos, os dois sensores fazem com que os robôs andem reto na arena.

C. Arduino

Uma placa Arduino Mega 2560 [4] foi utilizada para realizar as operações de baixo nível necessárias para o funcionamento do robô. Esta placa controla os motores ligados às esteiras do robô, os motores ligados à garra que captura as latas e a comporta na caçamba do robô. Além disso a placa Arduino lê os encoders ligados aos motores das esteiras e os sensores de ultrassom montados no robô. O código responsável pela leitura dos sensores de ultrassom foi baseado na biblioteca NewPing [5].

A placa Arduino foi programada utilizando C e C++.

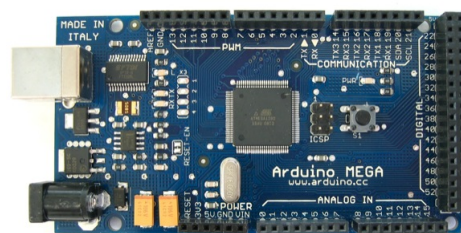


Fig. 4. Arduino Mega

D. Android

Aparelhos celulares modernos possuem grande poder de processamento, além disso possuem vários sensores embutidos, como uma câmera de vídeo, bússola e acelerômetro. Isto cria um ambiente poderoso para desenvolvimento de robôs móveis e por isso foi escolhido pela ERUS como sistema principal a ser utilizado no robô. O celular utilizado foi um Motorola Milestone I, rodando Android v2.2. A linguagem de programação utilizada dentro do celular Android foi Java.

O código no celular recebe periodicamente valores dos sensores de bússola e acelerômetro, uma imagem da câmera de vídeo que deve ser processada, informações capturadas pelo Arduino e sob algumas condições de operação troca informações com um programa rodando em um computador.



Fig. 5. Celular utilizado pela equipe

E. Comunicação Arduino-Android

A comunicação entre a placa Arduino e o celular Android ocorre através de um cabo USB. Para isto é usado um hardware adicional, o USB Host Shield [6], que adiciona ao Arduino uma interface USB.

O celular Android suporta o ADB (Android Debug Bridge [7]), uma ferramenta capaz de se comunicar com o Android através da interface USB.

Para que o Arduino seja capaz de se comunicar com o ADB, é preciso implementar o protocolo necessário. A biblioteca MicroBridge [8] implementa este protocolo e foi usada no código do robô.

Nesta configuração a conexão é vista como uma conexão padrão através de um Socket pelo código que roda no

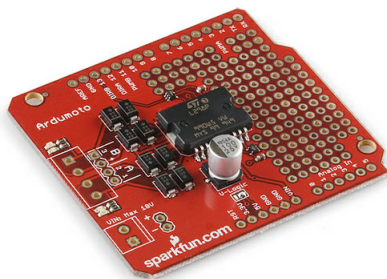


Fig. 6. Arduimoto - Utilizado para a fornecer corrente necessária aos motores

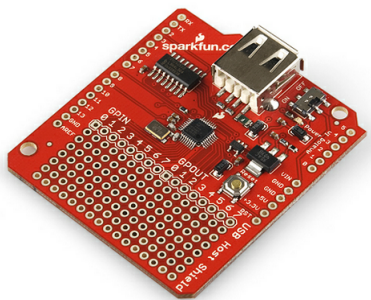


Fig. 7. USB Host Shield - Comunicação Arduino-Android

Android. No Arduino, a conexão é realizada através de funções da MicroBridge. Um protocolo de comunicação básico foi elaborado para que mensagens fossem trocadas entre o Android e o Arduino.

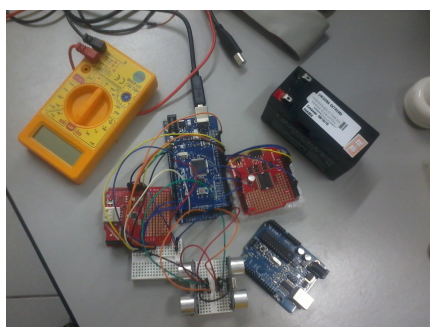


Fig. 8. Teste da comunicação realizado no laboratório

F. Comunicação Android-Computador

Esta conexão não é utilizada no momento da competição, é utilizada na depuração de erros, monitoramento dos sensores do robô durante os treinos e calibração de valores. Decidiu-se implementar esta capacidade para facilitar o desenvolvimento e acelerar a calibração de valores. É possível gravar dados de calibração em um cartão SD no celular através da conexão com o computador, quando o robô estiver operando autonomamente o cartão SD é lido e os últimos dados enviados pelo computador são utilizados. Também é possível rodar o robô em um modo

de monitoramento, no qual todos os valores de sensores e variáveis importantes são enviadas ao computador, sendo possível observar possíveis falhas em tempo real. Para esta finalidade foi desenvolvida, em Java, uma interface gráfica de monitoramento. O meio utilizado para esta comunicação é a interface Wireless disponível no celular e em computadores. O protocolo utilizado é o mesmo que entre o Arduino e o Android, sendo que algumas mensagens podem ser repassadas desde o Arduino até o computador e vice-versa.

G. Visão Computacional

A câmera do Android não possui alguma biblioteca própria para tratar as imagens. A equipe utilizou a biblioteca OpenCV [9] para Android [1]. É necessário implementar algoritmos que identifiquem os objetos que estão na arena. Pensando nisso, a equipe utilizou um detector de blobs para cores. Inicialmente, ao receber a imagem RGB é feita a conversão para HSV [10] para tratamento. São especificadas as cores a serem detectadas e os raios mínimos em que aparecem as cores (isto, já no ambiente HSV). É feito o processamento e retornado os contornos do que foi identificado.

V. PRINCIPAIS DESAFIOS ENCONTRADOS NO DESENVOLVIMENTO DOS PROJETO

Os principais desafios encontrados foram a comunicação Android com Arduino, a estrutura da garra e a visão computacional.

A comunicação Android com Arduino foi problemática até certo ponto. O firmware do Android estava desatualizado para a aplicação ADK [11]. Para solucionar isso, foi utilizada a biblioteca MicroBridge [8].

Outro problema, foi a montagem da garra, pois precisava de movimentos precisos para capturar a lata. A lata deslizava pela garra. Para solucionar isso, a equipe fixou um material aderente na garra.

E por último, o problema da visão computacional. Esta parte é utilizada para detecção e desvio dos objetos na arena e para navegação. Foi realizado muitos testes até chegar em um ponto aceitável.

VI. CONCLUSÃO

Com o avanço da tecnologia, os celulares possuem cada vez mais sensores, mais opções e mais aplicativos a baixo custo. O uso destes na robótica é bastante eficiente, um celular comporta mais de quatro sensores que são frequentemente utilizados em robôs autônomos e o custo de um celular é bem inferior aos sensores que seriam utilizados. Podemos analisar então, que o custo para construir um robô em casa foi reduzido e está acessível a maioria das pessoas. O sucesso do desafio mostra a possibilidade de robôs trabalharem em situações de ambiente pouco controlados.

REFERENCES

- [1] <http://developer.android.com/reference/java/lang/ref/Reference.html>
- [2] <http://mindstorms.lego.com/en-us/Default.aspx>
- [3] <https://www.sparkfun.com/products/10336>
- [4] <http://arduino.cc/en/Main/ArduinoBoardMega2560>
- [5] <http://arduino.cc/playground/Code/NewPing>
- [6] http://www.circuitsathome.com/arduino_usb_host_shield_projects
- [7] <http://developer.android.com/tools/help/adb.html>
- [8] <http://code.google.com/p/microbridge/>
- [9] <http://code.opencv.org/projects/opencv/wiki/OpenCV4Android>
- [10] http://mkweb.bcgsc.ca/color_summarizer/
- [11] <http://developer.android.com/tools/adk/index.html>