Team Description Paper: Team Emerotecos

André Seidel Oliveira[1], Gabriel Lima Guimarães[2], Ivan Seidel Gomes[3], Matheus Pimentel Canejo Pinheiro da Cunha[4], Hudson Cássio Gomes Oliveira[5], Felipe Nascimento Martins[6]

***Abstract*— This paper presents a description of an omnidirectional robot to be used on the Rescue B competition. The main robot controller was written in Java and runs in an Android-based mobile phone. The phone is connected to an IOIO board that serves as an interface between the main controller and the robot's motors and sensors. The IOIO board connects to an Arduino board, that is responsible for the distance and temperature sensors readings, and to an MBed board that controls the motor drivers. There are four infra-red distance sensors mounted on a rotary base on the top of the robot. The base angle is constantly changed by a servomotor, which allows a 360º distance measuring. The robot mechanical structure was designed using a 3D modeling software and was built using acrylic. It has four omnidirectional wheels that allow the robot to move in any direction without the need of turning itself. The robot was entirely designed, built and programmed by the team students.**

I. Introduction

This is to show team Emerotecos' strategies to solve the challenge proposed to the Rescue B Competition.

To build the robot, we didn't use any building kit, as we designed the whole robot from sketch, using Dassault Systemes' Solid Works software [1]. The robot was built basically with 5mm thick acrylic pieces, whose are tough enough for the application.

As the main processor, we use an Android phone, which is programmed using Java. This is a very interesting platform to work with since nowadays Android is the world's most popular mobile platform [2].

To send the signals from the Android to the motors (that are controlled by an MBED board [3]), we use a board called IOIO, which is responsible for reading all the sensors as well [4].

II. Objective

The objective of this project is to build a smart robot that can cross a maze build with wood walls, and identify the electrically heated victims that are placed along the maze's walls.

III. Ambient

The Challenge happens in a modular arena, made with wood, which has 2 floors, and 4 main rooms. The location of the walls are always unknown by the time the robot starts running, so that it runs in a real maze. The only constant information about the arena, is its total size, what makes it easier to the robot to find its way on the place. An illustration of the maze is shown in Figure 1.

Some heated "victims" are randomly positioned on some walls in the arena, and the robot has to identify them to get points.

In some places, there are some "dead ends", which are a black mat on the ground. The robot can run over these black mats, but it has to leave it on the same side it came from, it can't cross the black area.

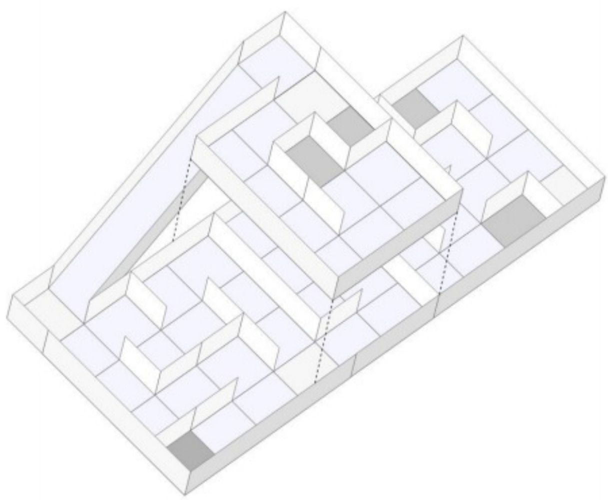

Figure 1. Rescue B Arena illustration.

IV. Strategy

A. Robot Structure

Our robot is built basically with 2 acrylic layers, and some acrylic beams. We use a Mecanum omnidirectional wheels system, which makes our robot able to drive to every direction. The Mecanum system works as a regular omni system, but it uses 4 tractioned wheels, instead of three, normally used. Figure 2 shows the robot design and Figure 3 shows the actual robot, without the control boards.

We use 4 very strong motors, with encoders, so that we can precisely move our robot everywhere. We rely on 4 motors, one for each wheel, because the MECANUM omnidirectional system need dierent control for each wheel. Each motor can spin giving a 6kgm torque, and its encoder can read up to 3592 pulses per turn because we have a 75:1 speed reduction.

The authors are with the Federal Institute of Education, Science and Technology of Espírito Santo (IFES). Rodovia ES-010, km 6,5 - Manguinhos, Serra, ES.
andre_oliveira@live.com[1], gablg@hotmail.com[2], ivanseidel@gmail.com[3], pimentelcanejo@hotmail.com[4], hudsoncassio@ifes.edu.br[5], felipemartins@ifes.edu.br[6].

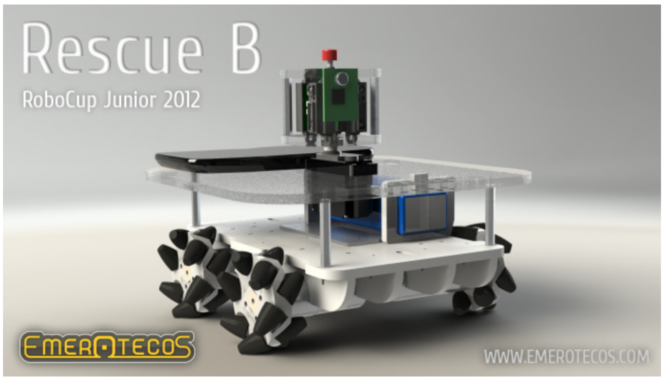

Figure 2. Robot design.

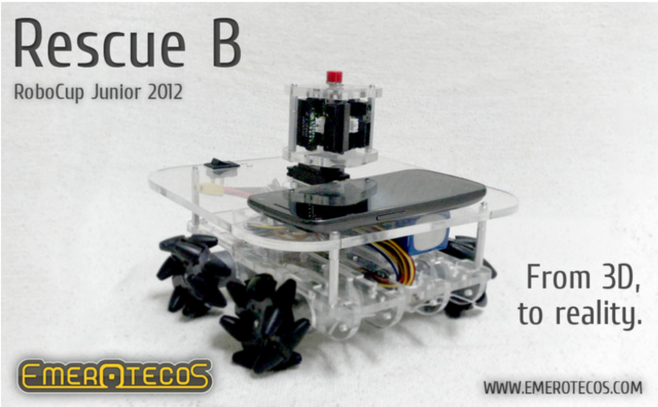

Figure 3. Robot picture(without control boards).

To process the encoder pulses, we have a MBED board, using a ARM Cortex M3, at 100MHz. This board is also responsible for controlling the engines speed. The communication protocol used to read the encoders from the interface, and to set the motors speeds, is the I^2C. The MBED board only has 3 encoder inputs. So, we had to program another encoder counter.

To make good measurements with the infrared distance sensors, we've put them in a turnable head over the robot, so that we can make meausures to every direction. To identify the victims, we have 4 thermal infrared sensors strategically positioned around the robot, so that we can measure temperatures in every wall.

As we use complex algorithms (described latter), we need a very complex processing unit. For this reason we have decided to use an Android phone on our robot. Android because we can program using Java. Also, the phone has a touchscreen LCD screen and several sensors we can access. For instance, we make use of the 3 axles gyroscope and 3 axles magnetometer that are present on the Android phone. This gives the robot a very precise way to know it's position and angle, on all axles.

As you can see in the Figures 2 and 3, the distance sensors are positioned on rotary base on the top of the robot. This base is driven by a servomotor so that the sensors can be aimed to virtually any direction. Its visible too how are the Mecanum wheels mounted on the robot, similar to a Four by Four car, but with the capacity of moving to any direction without the need to turn.

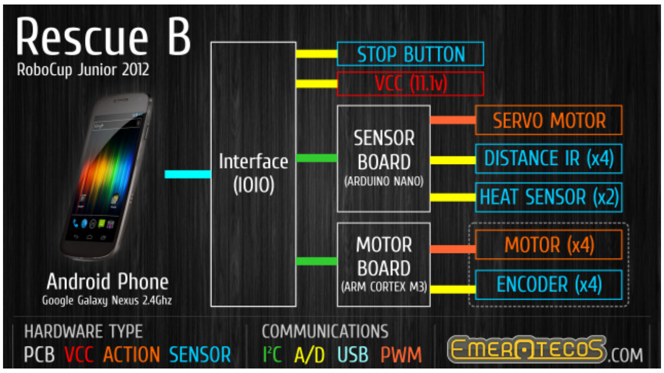

Figure 4. Control System overview.

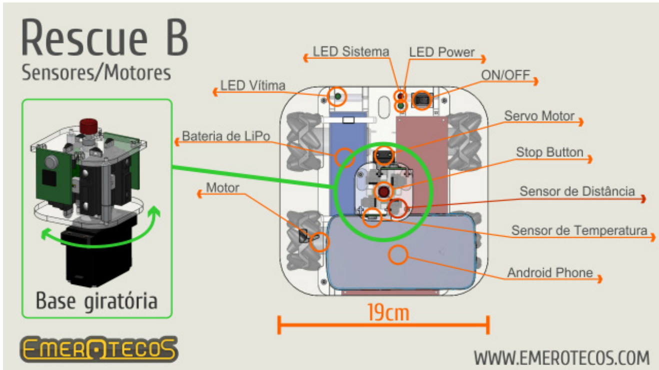

Figure 5. Rotary base (left) and top view of the robot.

B. Programming

The main controller of our robot is an Android cellphone. We choose Android, because it's an open platform [2], and it's possible to connect the cellphone with the sensors easily, using the IOIO board [4].

The Android phone sends and receives data to/from the IOIO, which reads the sensors, and communicates with the MBED board, the one responsible for controlling the motors. Figure 4 shows the control system overview.

Our robot uses 2 types of sensors: infrared distance sensor (by Sharp) and heat sensor. The heat sensor is used to detect the victims and are placed on the robot borders. As explained on the previous section, the infrared sensors are mounted on a rotary base. Each one of infrared sensors are pointed 90º from each other in a way that the 4 distance measures can be made at once. The 4 distance measures are repeated several times during a head rotation (one measure per degree) in order to detect the maze walls and obstacles. Figure 5 shows the rotary base with the infrared sensors placed on it, and presents a top view of the robot.

The distance measures are also used to correct the robot alignment in order to maintain its correct orientation parallel to the maze walls. This is done by comparing the measures at 20º and -20º from the central line, and calculating the necessary robot turning to make the robot alignment. Figure 6 illustrates the described idea.

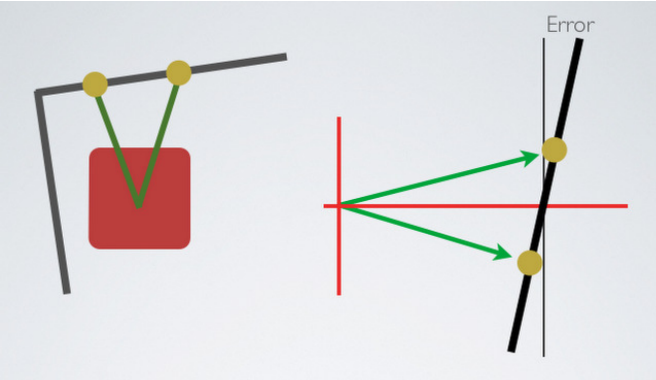

Figure 6. Robot design.

To move the robot, we use a simplified kind of SLAM (Simultaneous Localization And Mapping) algorithm, so that we can map the arena. We know that the walls are oriented in only 2 directions (0º and 90º), and we also know that each part of the maze is formed by a 30 x 30 cm square. With this knowledge and using the distance measures obtained via the infrared sensors, we can find out the walls positions and build a map of the maze. The SLAM was programmed in JAVA, and it runs on the Android while the robot is moving in the arena.

While the robot walks around the arena, we can see the map being formed on the cellphone's screen. Figure 6 shows an example of the mapping screen.

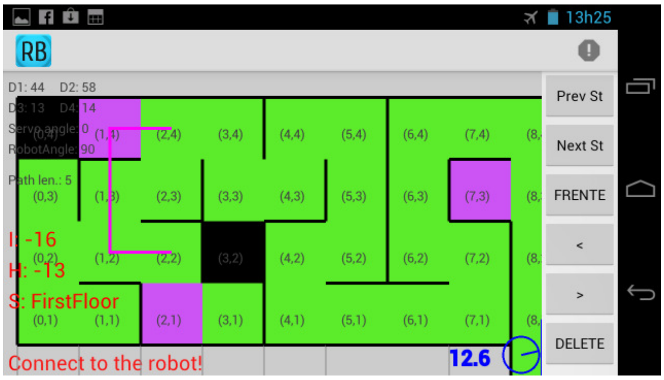

Figure 7. On-line mapping.

V. Conclusion

Mapping is the best way to solve this challenge, because it makes it easy to visit every module of the arena, and consequently, find all victims. We have successfully used this robot in the RoboCupJunior 2012 Rescue B competition. A presentation video about the robot is available in [5].

VI. Acknowledgement

The authors thank the following companies for the financial support that allowed the robot construction and participation at the 2012 RoboCup and CBR competitions: Proesi Componentes Eletrônicos, Nova Didacta, InTechno Desenvolvimento e Capacitação, Copy Express, Produtiva Hailtools, Use Móveis para Escritórios and Qualidata. They also thank IFES for the support and FAPES (a foundation of the Secretary of Science and Technology of the State of Espirito Santo, Brazil) for the financial support to pay for the air tickets and travel expenses to Fortaleza – LARC 2012 (Project 57768773/2012).

References

[1] 3D CAD Design Software SolidWorks. Available at: http://www.solidworks.com/. Access: Sep. 5, 2012.

[2] Andoid, the world's most popular mobile platform. Available at: http://developer.android.com/about/index.html. Access: Sep. 5, 2012.

[3] Mbed - Rapid Prototyping for Microcontrollers. Available at: http://mbed.org/. Access: Sep. 5, 2012.

[4] IOIO Documentation. Available at: https://github.com/ytai/ioio/wiki. Access: Sep 5, 2012.

[5] NERA. Prêmio de Melhor Robô do mundo é de alunos do IFES! (com vídeo). Available at http://nera.sr.ifes.edu.br/?p=143. Access: Sep. 5, 2012.