An Omnidirectional Robot for the RoboCup Junior Rescue B Competition

Abílio Marcos Coelho de Azevedo, André Seidel Oliveira, Ivan Seidel Gomes, Yan Victor Ribeiro Marim, Matheus Pimentel Canejo Pinheiro da Cunha, Hudson Cássio Gomes Oliveira, Felipe Nascimento Martins

IFES – Federal Institute of Education, Science and Technology of Espirito Santo, Brazil
NERA – Núcleo de Estudos em Robótica e Automação

***Abstract*— This paper presents a description of an omnidirectional robot to be used on the RoboCupJunior Rescue B competition. The main robot controller was written in C++ and runs in an Arduino DUE board. The Arduino board is responsible for the distance and temperature sensors readings, and is connected to an MBed board that controls the motor drivers. There are eight infra-red distance sensors mounted on the top of the robot, four of them measuring the distance from the robot to the side walls, two measuring distance to obstacles or walls in front of the robot and the other two measuring distance to obstacles or walls in the back. 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.**

1. Introduction

This is to show team Emerotecos' strategies to solve the challenge proposed by the RoboCupJunior Rescue B competition.

Instead of using a building kit, our robot was designed from a sketch using Dassault Systemes' Solid Works software [1]. The robot was built with 5mm thick acrylic and carbon fiber pieces. These materials are strong enough for the application.

As the main processor, we use an Arduino DUE board [2] and programmed it using C++. We choose the Arduino DUE board because, comparing to the Android mobile phone that we used last year, it communicates faster with the motors and sensors as the board has many analogic and digital inputs. It also has support for I^2C protocol, which we are using.

2. Objective

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

3. Ambient

The Challenge happens in a modular arena, made with wood, with 2 floors, and 4 main rooms. The location of the walls are always unknown at the time the robot begins its run. It navigates a true maze. The only constant information about the arena is its total size, which makes it easier for the robot to navigate the field. An illustration of the arena is shown in Figure 1.

In order to score points the robot must identify various heated "victims" randomly positioned along walls in the arena.

In some places there are "dead ends" (black mats on the ground). The robot may run over these black mats, but must leave the mat on the same side from which it came. It cannot cross the black area.

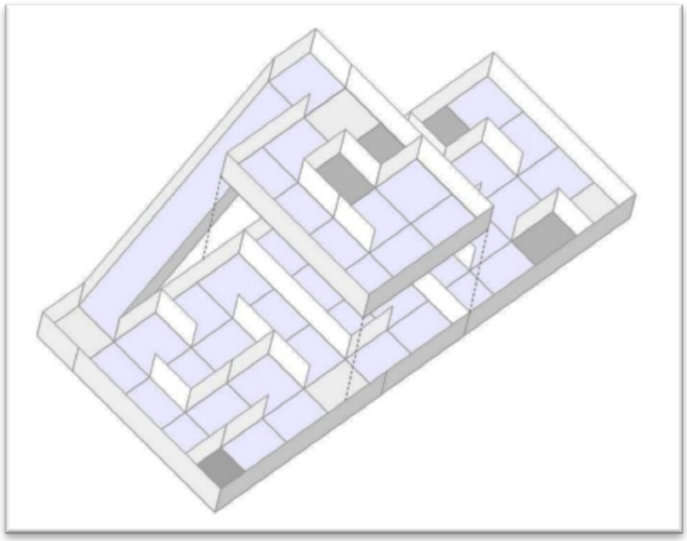

Fig. 1. Rescue B Arena illustration.

4. Strategy

A. Robot Structure

We participated in the Rescue B category last year. Since our robot had a satisfactory performance, we decided to build our new robot in a form similar to our first version [3]. Both robots are shown in Figure 2. Although the structure is the same, we have made some important improvements that we hope will ensure a better performance is this year's competition. The robot's basic structure is built with an acrylic layer and a carbon fiber layer, with acrylic beams between the two layers. We use a Mecanum omnidirectional wheels system, which allows our robot to move in every direction. Four strong motors, with encoders, drive the robot very precisely. We rely on 4 motors, one for each wheel, because the MECANUM omnidirectional system needs independent control for each wheel. Each motor has a torque of 6kgfm, and its encoder can read up to 3,592 pulses per turn because we have a 75:1 speed reduction.

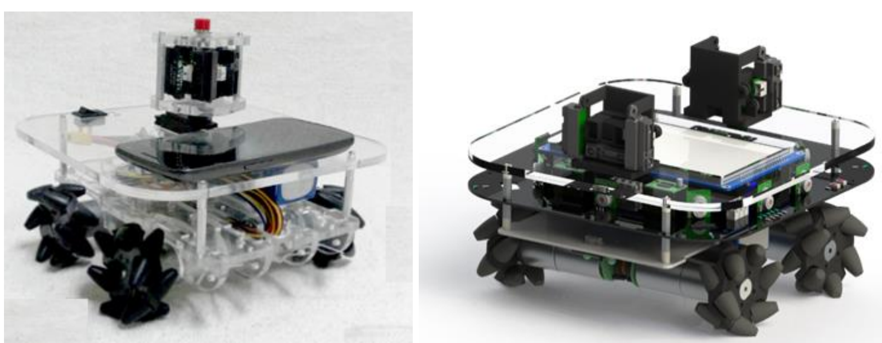

Fig.2. Last year's robot (left) and this year´s robot (right).

The Mecanum system works as a regular omni system. In the new model, we will try a new suspension system to hold the motors. The suspension system is made of Kevlar and its design is illustrated in Figure 3.

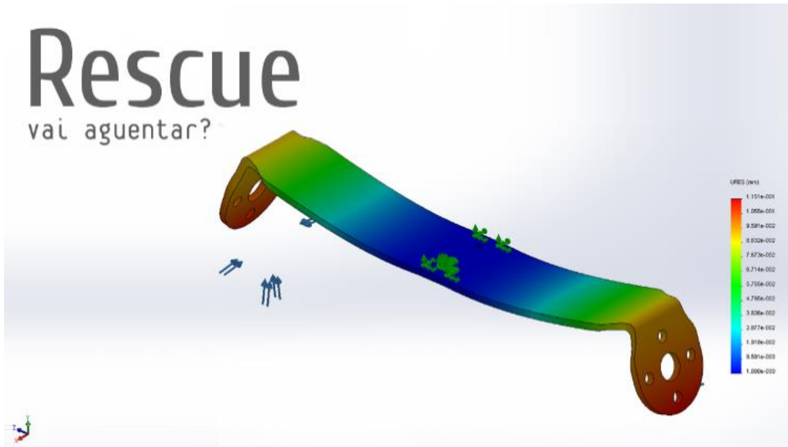

Fig. 3. Suspension bending simulation.

Figure 4 shows the robot electronics Printed Circuit Board (PCB) design. Figure 5 shows the actual board with the LCD placed on top of it. To process the encoder pulses, we have a MBED board [4] with an ARM Cortex M3 processor, at 100MHz. The board is also responsible for controlling the motors speed. The communication protocol used to read the encoders from the interface and set the motors speeds is the I^2C. We also needed to program another encoder counter because the MBED board has only 3 encoder inputs.

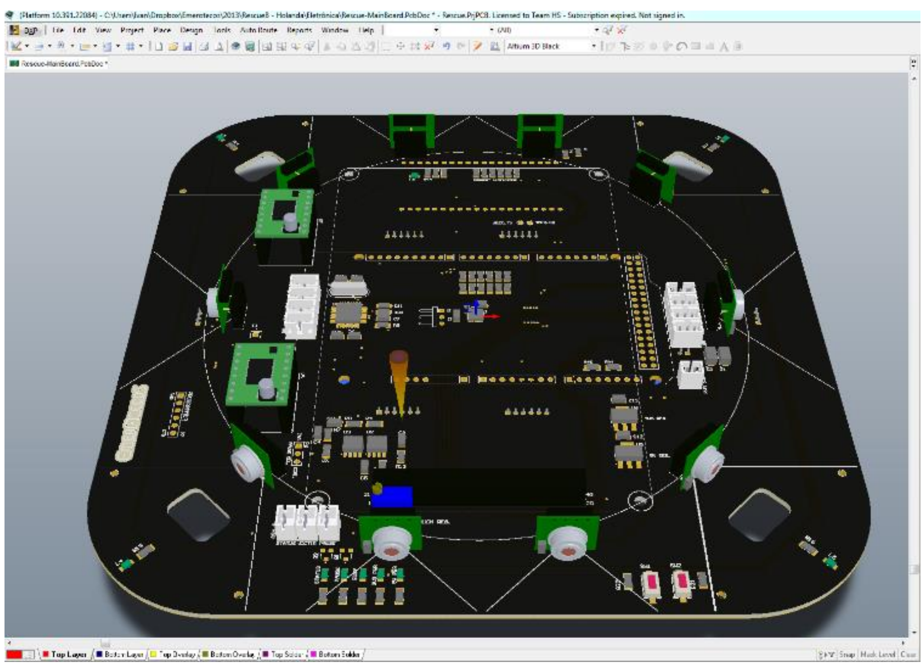

Fig. 4. Eletronics PCB design.

To ensure accurate measurements in every direction with the infrared distance sensors, we have added two supports that will accommodate the sensors in the front and back of the robot. Each support will carry four sensors.

Since we use complex algorithms (described later), we need a very complex processing unit. We decided to use an Arduino Due instead of the Android phone used on our last robot. We chose Arduino because its communication with the other controlling systems will be faster. Figure 6 shows where the Arduino Due and the MBED boards are placed on the bottom of the board.

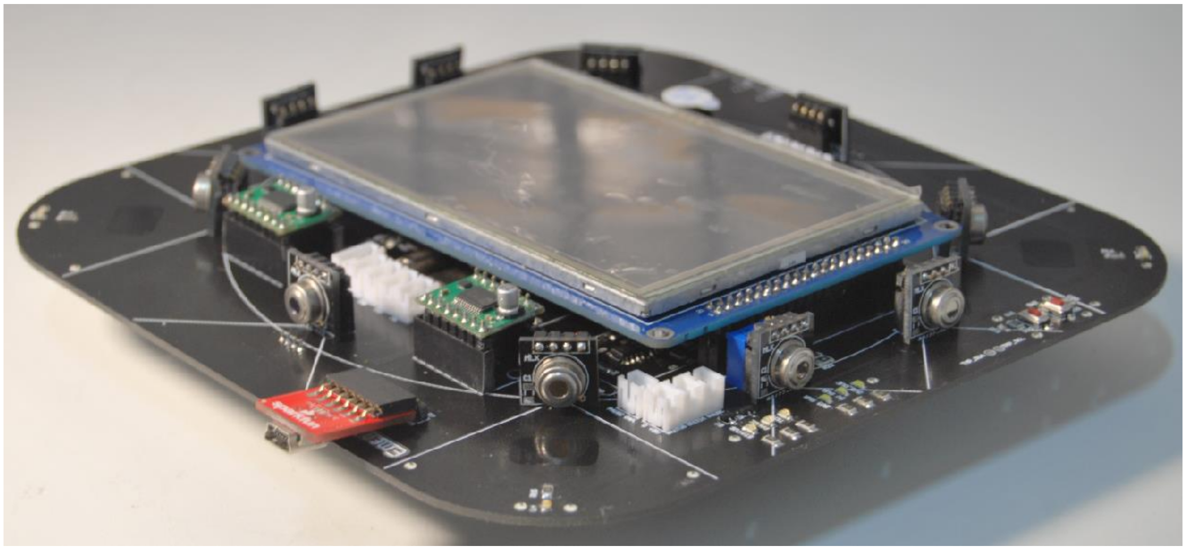

Fig. 5. Actual PCB (top) with heat sensors and LCD.

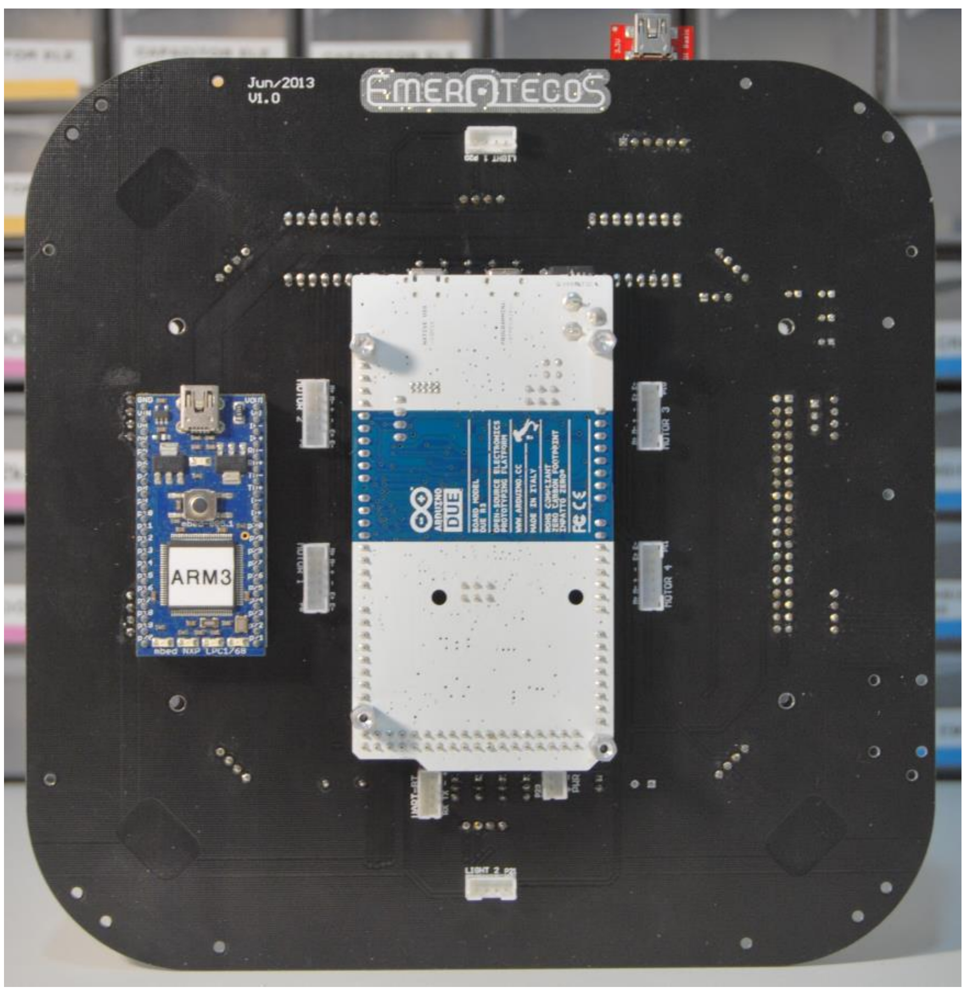

Fig. 6. Actual PCB (bottom) with Arduino Due and MBED boards.

B. Programming

The main controller of our robot, an Arduino Due, processes the main code and receives information from the distance sensors so that it can send commands to the MBED which in turn controls the motors.

Last year, an Android phone was used to send and receive data to/from an IOIO board [5], which reads the sensors and communicates with the MBED board (responsible for controlling the motors). We noticed that the Android phone configuration was overloading the communication system. To overcome this problem we are using an Arduino Due instead of the Android phone. We no longer need the IOIO board, which makes it faster to communicate with sensors and motors.

Our robot uses two types of infrared distance sensors: short distance sensors and long distance ones. Both types are mounted in the front and in back of the robot (pointing in the same direction). This was implemented in order to improve the precision of the distance measured (see Figure 7).

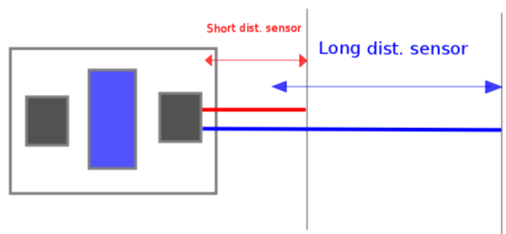

Fig. 7. Front/back sensors displacement

We also have heat sensors to detect victims along the robot borders. The reason for choosing the front/back formation for the sensors supports is so that the robot will know more accurately the end of any set of walls (Figure 8). For this reason they need to be as far from each other as possible.

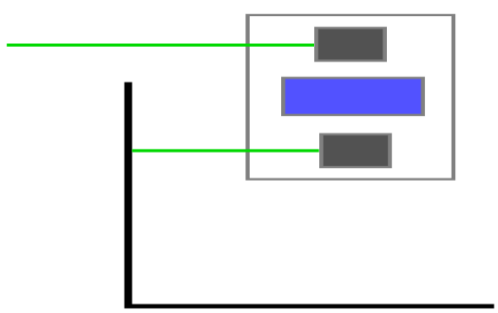

Fig. 8. Lateral sensors displacement

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

We intend to change the algorithm to implement obstacle avoidance and to improve motion control. The idea is to make the robot move continually with no stops. Last year´s robot successfully mapped the arena, as shown in Figure 9, but on each block it had to stop to take distance measurements. We are trying to avoid the robot needing to stop. We also intend to attach more heat sensors. Those improvements will be completed by June 2013 for the RoboCupJunior 2013 competition, in Holland.

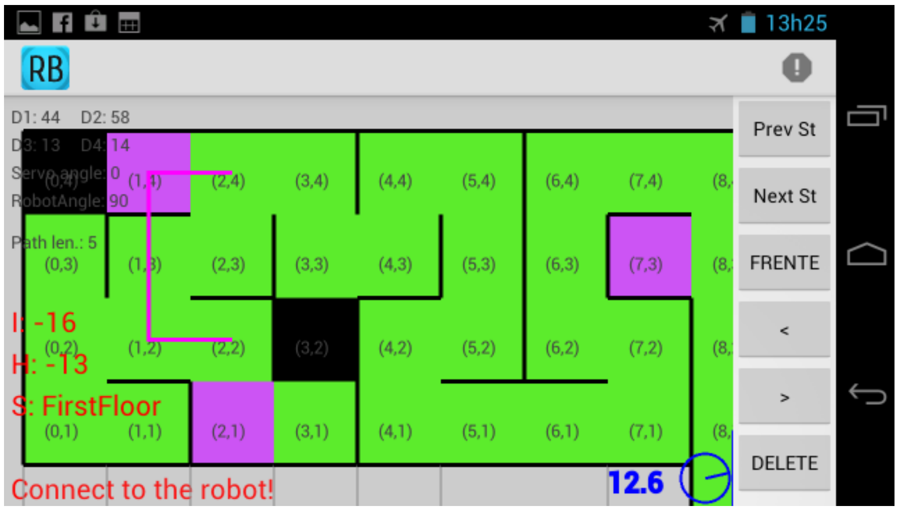

Fig. 9. On-line mapping from last year´s robot.

5. Conclusion

Mapping is the best way to solve the challenge because it makes it easier to visit every module of the arena, and consequently, find all victims. We successfully used this robot in the RoboCupJunior 2012 Rescue B competition. A presentation video about the robot is available [6]. In RoboCupJunior 2013 Rescue B we intend to succeed in mapping the arena using the SLAM algorithm!

6. Acknowledgement

The authors thank the Federal Institute of Education, Science and Technology of Espírito Santo (IFES) for all their support given to our project.

7. References

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

[2] Arduino Due. Available at: http://arduino.cc/en/Main/ArduinoBoardDue . Access: Mar. 29, 2013.

[3] OLIVEIRA, A.S.; GUIMARÃES, G.L.; GOMES, I.S.; CUNHA, M.P.C.P.; OLIVEIRA, H.C.G.; MARTINS, F. N. Design and Implementation of an Android-Based Omnidirectional Robot for the RoboCup Junior Rescue B Competition. In: WEROB 2012 – Workshop on Educational Robotics, part of RoboCup International Symposium, 2012.

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

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

[6] 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.