*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 C++ and runs in an Arduino DUE board. The Arduino board, that is responsible for the distance and temperature sensors readings, 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 form the robot and the side walls, two measuring the distance in front of the robot and the other two measuring the distance from 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.

Team Description Paper: Team Emerotecos

Abílio Marcos Coelho de Azevedo1, André Seidel Oliveira2, Ivan Seidel Gomes3, Yan Victor Ribeiro Marim4 , Matheus Pimentel Canejo Pinheiro da Cunha5, Hudson Cássio Gomes Oliveira6, Felipe Nascimento Martins7

# 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 Arduino DUE boad, which is programmed using C++. We choose the Arduino DUE board because, comparing to the Android mobile phone that we used last year, it is easier to communicate with the motors and sensors as the board has many analogic and digital entries, also using I2C protocol.

# Objective

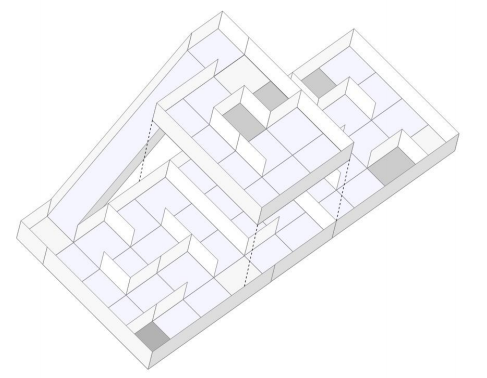
The objective of this project is to build a smart robot that can cross a maze build with wood walls, and identify the electrically heated victims that are placed along the maze’s walls.

# 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.



Rescue B Arena illustration.

# Strategy

## Robot Structure

As we participated in this category last year, we built it basically like our first version(figure 4), but with some important improvements that will ensure a better performance is this competition. The robot´s basic composition are 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, and now we´re going to try new suspensions (figure 2) to hold the motors, made of kevlar. Figure 3 shows the robot eletronics design and Figure 1 shows the actual robot.

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 different 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.

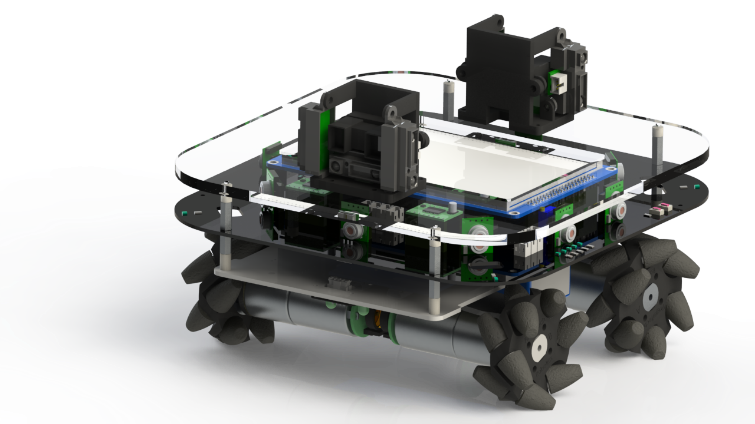


Figure 1 - This year´s Robot

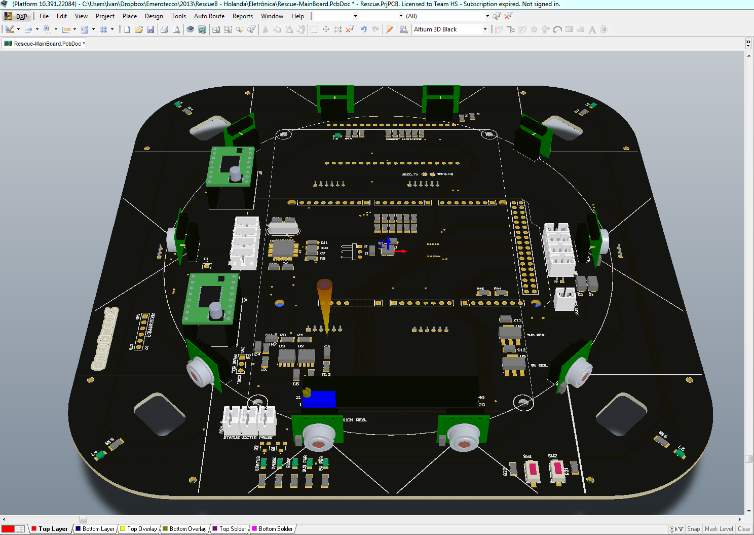


Figure 3 - Eletronics.

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 I2C. 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 made two supports that will accomodade them in the front and back of the robot, to get measurements in every direction. Each support will carry four sensors.

As we use complex algorithms (described latter), we need a very complex processing unit. For this reason we have 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. Also, we have an independent touchscreen LCD screen and several sensors, like IMU, so that we can replace the Android facilities. Our sensors gives the robot a very precise way to know it’s position and angle, on all axles, so that we can execute our mapping device with precision and reliability.

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 needing to turn.

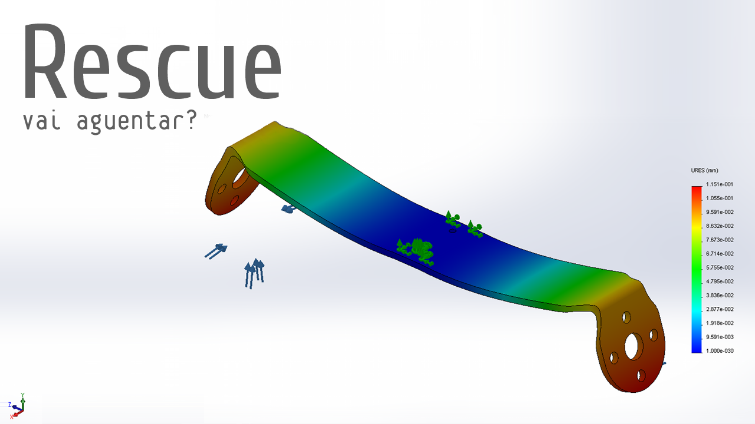


Figure 2 - Suspension pressure test

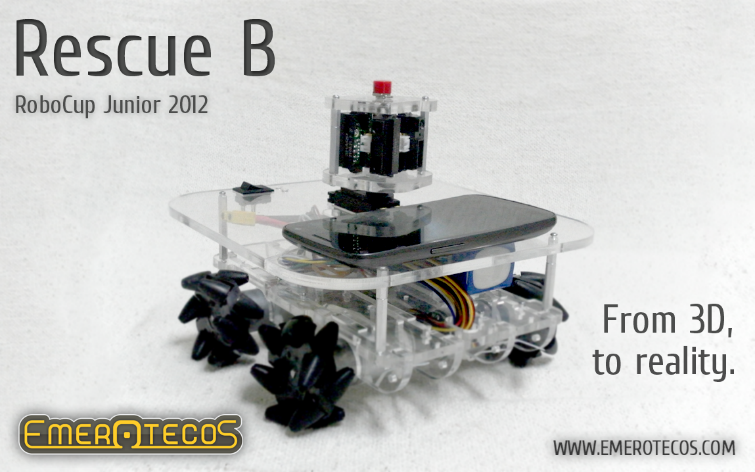


Figure 4 - Last year robot

## Programming

The main controller of our robot is an Arduino Due, witch process the main code and receives information from distance sensors so that it can send the orders to the MBED, that controls the motors.

Last year, the Android phone were used to send and receive data to/from the IOIO, which reads the sensors, and communicates with the MBED board, the one responsible for controlling the motors. Turned out that this configuration was overloading the communication system, so with Arduino we could just take the IOIO board and make a faster contact between sensors and the processing.

Our robot uses 2 types of distance sensors: short distance sensors and long distance ones. That way we can be more precise at taking measurements by the front an back of the robot, as in figure 5.

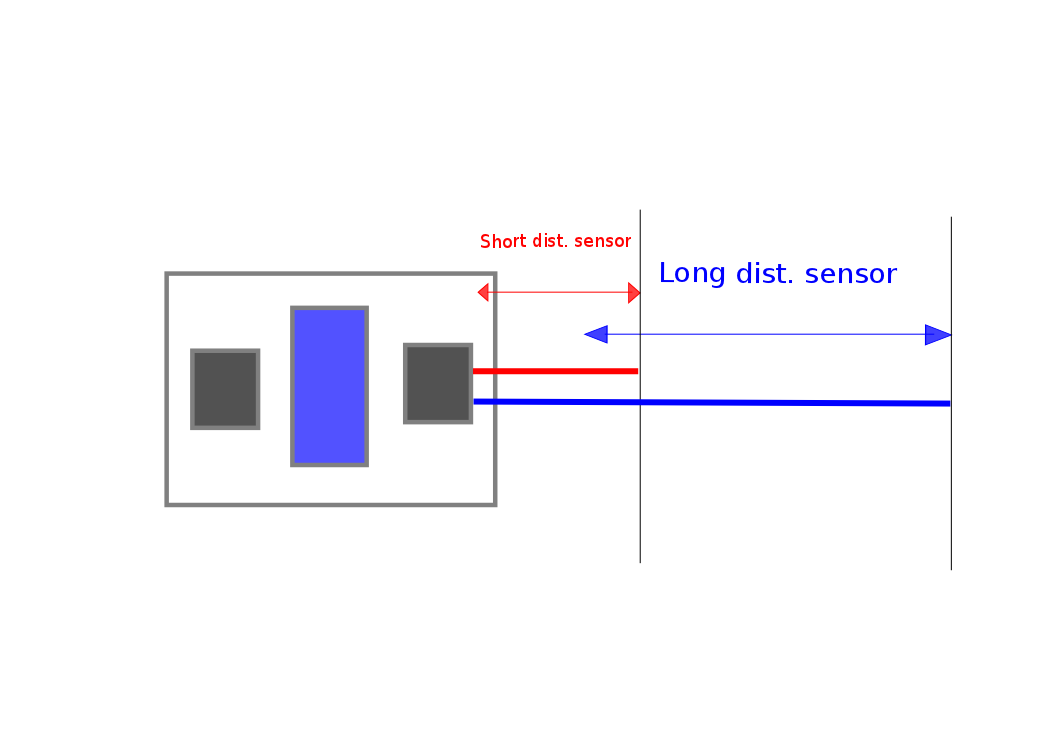


Figure 5 – Front/back sensors displacement

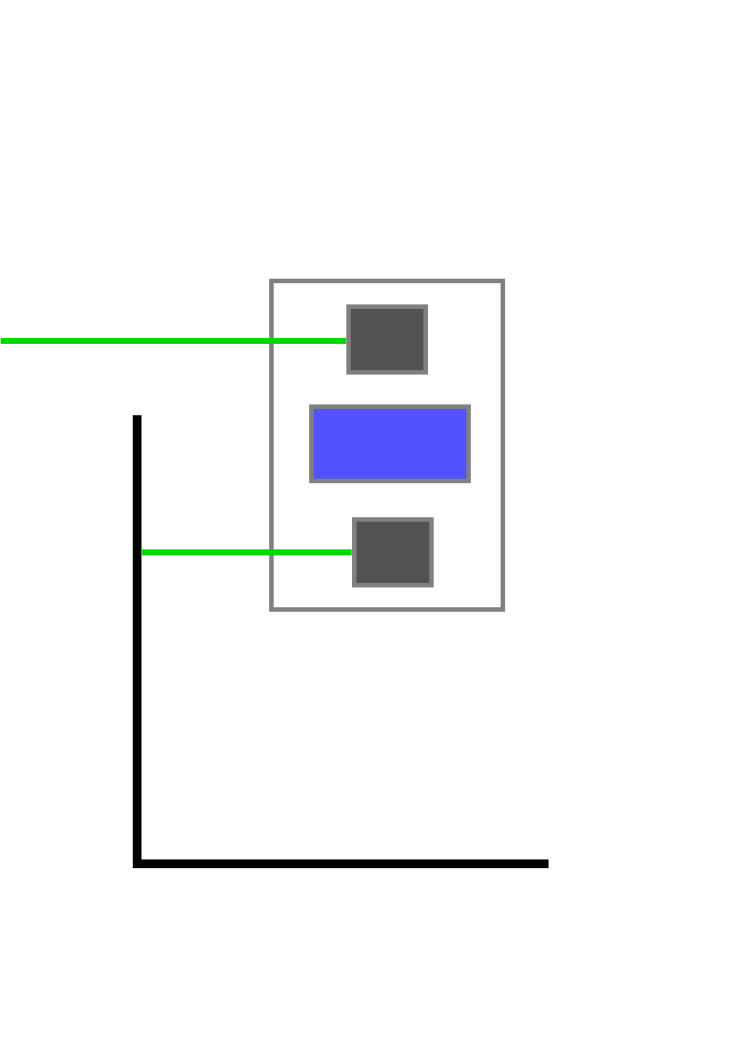
 We also have heat sensors to detect victims all along the robot borders. The reason of choosing the front/back formation for the sensors supports is because that way the robot will know more accurately the end of any set of walls (figure 6) , so they need to be as far as they can be from each other.

Figure 6 – lateral sensors displacement

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.

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 is programmed in C++, and it 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 succeeded to map, as in Figure 7, but on each block it had to stop, so we are trying to avoid it now. We also intend to attach more heat sensors. Those improvements will be completed by June 2013 for the RoboCupJunior 2013, in Holland.

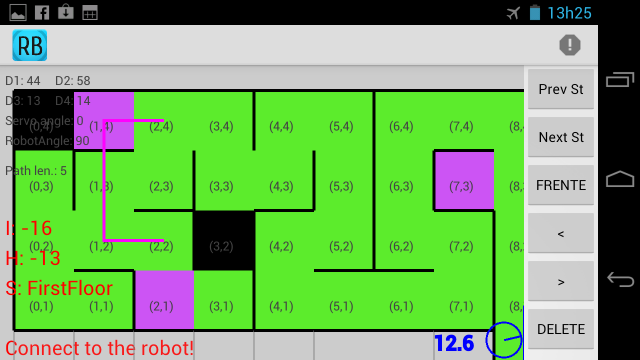


Figure 7 - On-line mapping from last year´s robot.

# 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].

# Acknowledgement

The authors thank the Federal Institute from Espírito Santo (IFES) for all the support given in this project.

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.