The Problem

One of our biggest problems is to find all victims in the maze, the fastest way possible. As we don’t have to find the end of the maze, to find all victims, all we have to do is go everywhere in the maze.

The Robot: Mechanics

This year, we said bye to the NXT brick, thanks to it’s weight, and to it’s engines size. We decided to build from sketch an entire robot, using 2 5mm acrylic layers and some acrylic beams under the structure to support the engines and the electronics.
The MECANUM wheels, are just like an ordinary omnidirectional system, but with the wheel mounted in a different way.
We use a servo in the center of the robot, that is used to move the sensors, giving them a bigger reading area.
Our robot measures 200x200mm, which makes it possible for the robot to spin without touching the walls. We use 5mm acrylic, because we know (by tests) that it is tough enough. We cut it with laser (on CopyExpress) giving us a 0.01cm precision.
All drawing have been made on SolidWorks!
We also used a 3D printer to do some parts on our robot, like the thermopile array`s stand.

The Robot: Electronics

Yes, we use a cellphone! As we use very complex algorithms, we need a very complex processing unit, that’s why there is an Android phone on our robot. Android because we can program

using java, without any problem, and we still have a LCD screen, and a touchscreen panel, with which we can make the initial configurations.
To fire up the engines, read the sensors, and control the engines position, we have, basically 2 PCB’s.
To feed the monster, we use a liPo battery, which gives us 11.1v, 65A, and is very lightweight.

Interface

We have a interface that is responsible to connect the Android (via USB) to the robot. It`s called the IOIO board, which will communicate with the MBED, that will read the sensors and spin the wheels.

Locomotion

We rely on 4 engines, one for each wheel, because the MECANUM omnidirectional system need different control for each wheel. Each engine can spin giving a 6kg torque, and can read up to 3592 pulses per turn of the final axle even though we have a 75:1 speed reduction.
To process all this pulses, we have a MBED board, using a ARM Cortex M3, at 100Mhz speed. That board is also responsible for controlling the engines speed. The protocol used to read the encoders from the interface, and to set the speeds, will be I2C Fast.
The MBED board have an onboard encoder counter for only 3 engines, fact that make us emulate another encoder counter (he hope NXP can read this, so that, on the next time, we won’t need to emulate the counter)

Sensoring

The Algorithm we’re using needs a lot of distance measures around the robot. If one rotates 4 sensors for 90 degrees, one’ve got 360 degrees of pure distance measurement! And as

we use 2 thermopiles arrays, one'll have a 180 degrees of temperature measurement (more than enough).
The MBED is responsible for making the reading, filtering, and sending them via I2C!

IMU

Besides all that sensors, we also have an 9 axles Inertial Measurement Unit (IMU). It's basically an unit that joins a 3 axles gyroscope, a 3 axles accelerometer, and a 3 axles magnetometer, giving the robot a very precise way to know it's position and angle, on all axles (we hope we are in a 3D world).

The Robot: Software

We will do our best to find the victims the fastest we can. We're used to write all the code in C, but this time, we are using C++ for the MBED and Java (and C++) for the Android. This year, we`re implementing a very complex algorithm, usually used in situations involving mazes. It`s called SLAM! (simultaneous localization and mapping). As you can see, the algorithm basically maps the arena, but in the same time, it locates the robot in the arena.