# CHS TechnoBots Team Description Paper

Ciro Salzmann, Davi Nakajima An, Felipe Rodriguez Picchi, Tiago Salzmann, Victor Penteado Rosolem
Comphaus Robotics Laboratory
Rua Aurélia, 1380 – Lapa, São Paulo, SP, Brazil

**Abstract - This document explains the explicit goal of the robot soccer team CHS TechnoBots and methodology applied in its development. The project deals with two robots that have the autonomy to play soccer according to rules of the category of RoboCup Junior International Soccer (Soccer Open Secondary). In this category, teams are to compete each other, where each is allowed to have two robots on a rectangular field composed by a green carpet and measuring approximately 2.5 by 1.8 meters. On the field, the four robots, two from each team, are to play soccer, aiming to score in the opponent's goal with a ball that emits infrared waves at 40kHz. A green carpet composes the field and the ball emits infrared rays that can be located by the robot.**

## I. LOGIC

The robot was designed to compete in Soccer Open Secondary modality and is constructed by a main processor, Arduino Due, which processes the data received by the various sensors strategically positioned on the robots, and finally assigns specific movements. The robots' main logic is to take fast decisions according to the environment's physical characteristics that are interpreted by precise sensors, such as the ultrasonic sensors that are able to determine the robot's position in the field.

The old project used to have an Arduino Mega2560 (a 16MHz microcontroller), but had to be replaced because of the relatively huge amount of time used to complete a full sensor reading. The solution was to use an Arduino Due. He would have enough processing speed (almost 5 times the processing speed of a conventional Arduino Mega). It is identical to the Arduino Mega2560, despite the additional digital ports and processing speed.

## II. SENSORS

### A. Ultrasonic

One of the main aspects that the robot has to possess is the ability to locate its position on the field. By having four ultrasonic sensors (front, back, right and left) positioned on top of the robot, it is able to determine its distance from the walls and therefore its location. Ultrasonic sensors are capable of calculating its distance apart from a certain object by sending a wave of specific frequency, which is then reflected back by an object in front of it and the receptor, present in the sensor itself, identifies the reflected wave. During this process, the time is recorded and the distance between the object and the sensor is calculated, with the knowledge of the speed of the wave (Mach).

One big problem we have experienced since we started using this sensor (Parallax PING))) ultrasonic distance sensor) was the waves being reflected by the floor and the goal. The reason behind this problem is that the waves sent by the sensor do not leave exactly perpendicular to the "transmitter". There is an opening angle, as the wave diffracts when leaving the end of the transmitter. As a result, when the robot is more than one meter away from any object, the distance received by the robot is actually the reflected wave from the floor. To avoid this type of reflections we tried to install a case/filter made of foam. The special feature about foams is that since their structures are not concrete, with holes rather, they are able to absorb, or decrease, the air vibration, that is, including the sound waves emitted by the sensors. The case will then be designed to absorb the waves that are emitted at a relatively high angle.

Another solution to this problem would be turn the sensors up or even place them in a higher place. In our current robot, this is not possible because of the parts that hold the robot. However, for the next version, that is already on the list.

Similarly, one big problem of the older robots is the time taken to do one single loop of the program. The main reason is not the complexity of the program, but the time it takes to read the four ultrasonic sensors. To deal with this problem, the team decided to separate the ultrasonic sensors from the main microcontroller (Arduino Due). An ATMEGA 328P-PU was added, which now reads all ultrasonic sensors and sends all the data to the Arduino Due through a serial communication

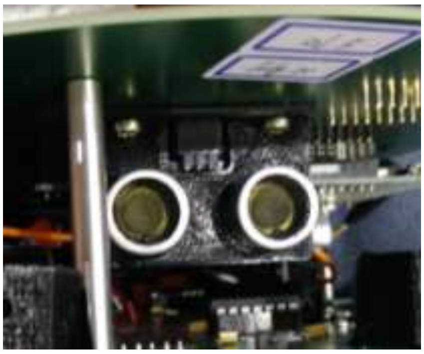

Figure 1. Right ultrasonic sensor

### B. Compass

The task of this specific sensor is to analyze the robot's direction by using the magnetic field of our own planet. Consequently, by calculating the robot's angle according to the calibration given by the human determined initial angle, the robot is able to know its direction on the field and therefore the degree of error to which it is facing the opponent's goal. The compass is also essential to determine the location of the robots. In order for the robot to know its location on the field, it would need the compass to adjust its direction so that the ultrasonic sensors would be able to calculate the correct distances from the walls, as the sensors are exactly directed to the right, left, front and back of the robot. The strategy used is always facing the side in which the robot was initially powered (hence the direction of the opponent's goal). This effect is achieved by using the PID (Proportional-Integral-Derivative) controller.

Because of the strong magnetic field created by the motors (which eventually changes the values of the compass), the sensor had to be positioned on top of the tower (far away from the motors).

Figure 2. Compass sensor on top a tower

### C. *Infra-Red*

Another fundamental task in the modality of soccer is to locate the ball, which is easily identified for its emission of infrared waves in a specific known frequency (40 kHz). The sensors use the same system as an ordinary television, which receives waves sent from a remote control. The 16 sensors placed around the circumference of the robot are able to know a relation between its own location and the angular position of the ball. The sensors are directed radially out of the robot, so that when the ball is within a certain angular range from its actual direction, the readings are stronger. The direction of the ball relative to the robot is calculated by the average values from all the sensors that are reading any source of infrared wave. It is important to note that a single sensor can only identify if it is or not "seeing" the ball in front of it. It does not determine the relative angular position of the ball. That is the reason for all the 16 sensors around the robot. There is a great variety of infrared sensors in the market. In order for us to find the best sensor possible for the purposes required on the soccer modality, the group had to buy and test various kinds, including the following models: KSM603-LM, IRM2636, KSM-16038, IRM3638, SL-1738, TSOP2240, and TSOP1140. After testing all of these sensors, the group has concluded that the TSOP31140 demonstrated the best performance and it is therefore the model that we are currently using in our robots.

Last RoboCup the team noted that the digital sensors were not precise enough when the ball is close to the robot. Therefore, the actual robot uses a combination of two different infrared sensors: eight TSOP31140 (digital) and eight TP (analogic) sensors. When the ball is far, the digital sensors are used, and when the ball is close to the robot, only the analogic sensors are used.

Besides the sensor itself, 3D printed cases and insulating tape were used to cover the sides and back in order to decrease the angle and distance at which the sensor can see the ball. Without the tape, all of the sensors would see the ball.

It is important to point out that the ball emits a frequency with different wave strengths on top of the 40 kHz (so one can estimate the distance between the robot and the ball by the amount of signals received). We decided not to use this changing frequency and installed a capacitor of $4.7\mu F$ to the digital sensors to smooth out the signal (so we receive only one or only zero instead of a frequency).

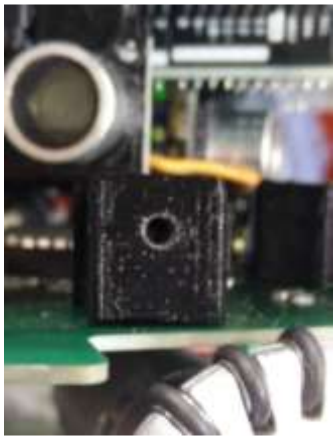

Figure 3. TSOP31140 with 3D printed case

### D. *LCD (Liquid Crystal Display)*

In order to simplify the process of testing the performance of the robots and eliminate bugs in the program, a 2.4" TFT LCD Touch Shield (an Arduino compatible shield) was installed. It gives instant feedback about the sensors readings, such as the inputs of the compass and infrared sensors, thus giving information about the robot's angular and linear position. This piece of tool, although not a sensor, is essential as to save time when improving the robot's program and therefore overall performance.

Figure 4. 2.4" LCD with custom interface

### E. *XBee*

The importance of this tool is establish a communication between the two robots on the field to create a teamwork. Thus, radio module allows both robots to share information about their position relative to the ball and finally decide their best tasks (goalkeeper or attacker).

For instance, if the ball is more favorable to the current goalkeeper than the attacker is, both robots will switch roles, allowing the original goalkeeper to take advantage of its possession of the ball. To establish this communication, the team decided to use the module named XBee, which uses a wireless communication language called ZigBee.

## III. MECHANICS

### A. *Main Motors*

The motors used were 25mm Gear motors from Pololu (with a reduction of 24.1:1) and they were shipped from the USA. Each robot would have three motors angularly positioned 120° apart from each other, where one is directed exactly to the back of the robot. The reason for using three motors instead of four is that, with three motors, we guarantee that all wheels are touching the ground. The three motors are attached with Omni wheels designed by the team, for 2-dimensional movement purposes. The electrical motors are powered by one set of 7.4 V LiPo batteries.

### B. *Motor controllers*

The complexity of the motors requires motor controllers, which would control the velocity and precision of the mechanical piece. Two MC33926 modules were used (one single version and one dual version).

### C. *Omni wheels*

Another very essential part for the movement of the robot is the actual wheel, which is uncommon, in terms of everyday technology, however extremely useful. Its difference from the common wheels is that it is capable of displacing itself in two perpendicular axes. The actual motor attached to it would provide the movement for one axis (tangential), and the net velocity of the other two motors would provide the other (radial) movement. This second axis of movement is possible by the implementation of various small wheels, made of rubber, that are placed perpendicular to and uniformly around the circumference of the original wheel. The wheels were designed and manufactured by the team and the rubber parts were bought in a specialist shop.

### D. *Batteries*

The power source that enables the robot to move and process information is given by one 2S LiPo battery (7.4 V), with 2100 mAh. We have in total four 7.4V LiPo batteries and each robot has only one while playing. The other two that remain are used to replace both batteries that are being used in the first interval of the game. When not totally charged, the robot's performance may not be as great. To avoid this problem we replace the batteries before entering the second half of the game. The iMAX B6 Balance Charger is used to charge all batteries.

## IV. STRUCTURE

### A. *Body Layers*

Three layers that hold the entire robot as a whole mainly structure the robot. The lowest layer contains the three equally separated motors and four LDRs (Light Dependent Resistor). The middle layer is connected with the two other layers and contains all infrared sensors, all motor controllers, and the battery. The last layer, roughly the same size as the others, contains the Compass sensor, the controllers (Arduino Due and ATMEGA), and the LCD.

### B. *Printed circuit*

The two top layers have a very interesting aspect regarding the "cleanliness" (absence of wires). The purpose of using a printed circuit instead of a board full of wires is to maintain a clean environment but mainly to prevent a

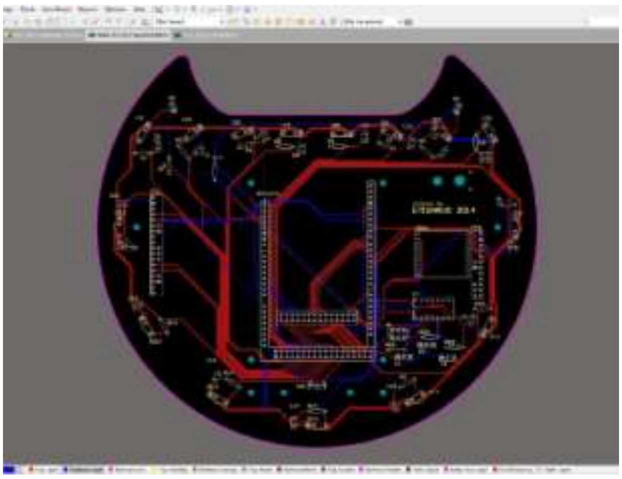

Figure 5. Robot's main circuit in Altium

failure, such as the breaking of poor contact from a wire. The printed circuit was designed on a program called Altium[1] and was fabricated by a Brazilian company. The next picture (Figure 5) shows the middle layer's circuit where all trails ("wire connections") can be seen:

### C. *Design*

The entire project (layers, sensors, motors) was previously graphically analyzed. With the aid of a three-dimensional design program called Autodesk Inventor 2013[2], all pieces and the entire design of the robot in general were built digitally to have an idea of the actual result. This program was also used to design and fabricate the Omni

[1] Access http://www.altium.com/ for more information

[2] Access http://www.autodesk.com/ for more information

wheels as well as the cases for the ultrasonic sensors. The following image (*Figure 6*) shows the design of the robot using the program aforementioned:

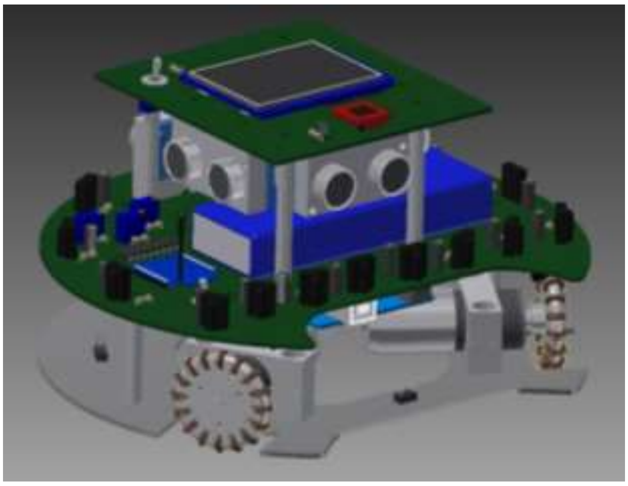

Figure 6 – Robot design created in Autodesk Inventor

## V. PROGRAMMING

The program system used is the Arduino IDE programming environment.

### *A. Classes*

The program, for reasons of organization and simplicity, is divided into several *classes* (C++ data structures), one for each type of sensor. This facilitates when calling the function to read all sensors in different parts of the entire code. This is also used to create an array, since there are many sensors of one just type (this includes the IR sensors, ultrasonic sensors, compass and the motors). In general, each class is located in a different file, helping even more with the organization of the code.

### *B. Headers and Includes*

With the same purpose of using classes, the code is divided into many different files. Each file contains a class or functions related to one sensor or even global variables. This helps when trying to find a certain variable and function, since it will not be all inside one long file. It is worth noting that, although not our priority, the LCD is responsible for almost half the code. This is because of all the communication behind the functioning of the LCD and the interface in general. Even so, the LCD is a very important tool because it enables us to read all the sensors in real time.

### *C. PID Controller*

To join all the needed moves to make the robot actually play we had to install many PID controllers: one for the rotational movement (using the compass values) and one for the horizontal movement (based on the values of the ultrasonic sensors). To implement this in our code we created a new class to make everything easier and created an object for this class in the sensors' structure (in their

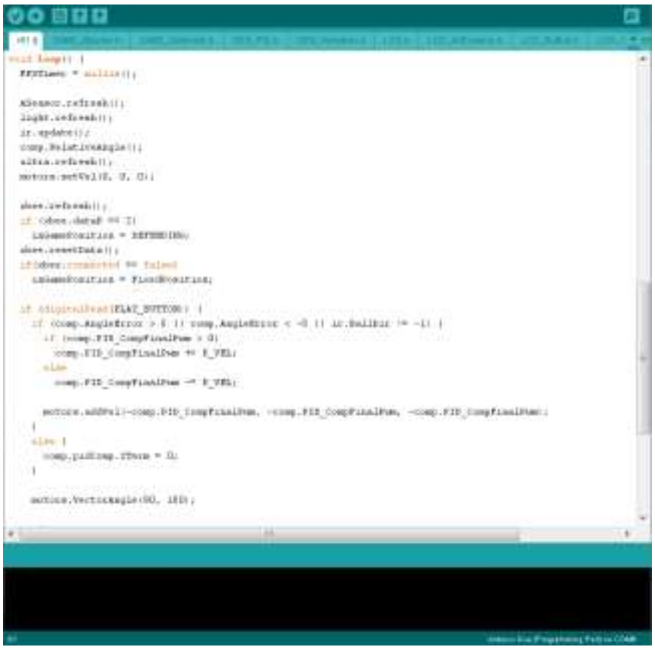

Figure 7 – Arduino IDE with the main code file

files, class etc.). All structures that are related to the PID controller have a delay of approximately 40ms to avoid processing waste, since the ultrasonic sensors read only every 30ms. This happens to all sensors. To have an idea, the infrared sensors read every 30ms and the compass sensor every 5ms. In the case of the motors, which are constantly moving without any type of delay, the logic is to make them move and, while these 30ms of the PID timer have not gone, use the sensor values of the last reading. This makes the robot a little bit unreliable when we think about the time "lost", but we can't forget that we are talking about a thousand of a second.

## VI. THE TEAM

Although all members of the team have plenty understanding in all areas of knowledge involving the construction and programming of a robot, some members focus more on one part of the robot than others. In term of the program, for example, one member writes all the code according to the logic given by the whole team. Similarly, other members focused more on developing the circuit and building the robot virtually before actually constructing it in real life. Following this logic, there is a type of division of labor, which enriches the final product through the cooperativeness. It is important to note that, although there is a division of "chores", no work is seen as "more important" than the other.

## VII. CONCLUSION

This project was not created in just days. Months of dedication and effort were necessary to bring the robots to life. From the design to the sensors used, everything had to be carefully calculated and tested before making part of the robot soccer players. The years of hard work and dedication showed by the team are fully reflected by the quality of work evident on the final products and it is with great pleasure that we can participate in a competition in which we can show our creation and test our skills.

## REFERENCES


[1] "3D CAD Software." *Autodesk*. N.p., n.d. Web. 20 Aug. 2014. <http://www.autodesk.com.br/products/inventor/overview>.

[2] "Altium Designer." *EDA Software*. N.p., n.d. Web. 20 Aug. 2014. <http://www.altium.com/>.

[3] Mizrahi, Victorine Viviane. *Treinamento Em Lingaugem C++, Módulo 1*. 2nd ed. São Paulo: Pearson Prentice Hall, 2006. Print.

[4] "Parallax Inc | Equip Your Genius." *Parallax*. N.p., 2014. Web. 20 Aug. 2014. <http://www.parallax.com/>.

[5] "WHAT IS ARDUINO?" *Arduino*. N.p., n.d. Web. 20 Aug. 2014. <http://www.arduino.cc/>.

[6] "ZigBee Alliance." *ZigBee Alliance*. N.p., n.d. Web. 20 Aug. 2014. <http://www.zigbee.org/Home.aspx>.