ROBOTICS COMPETITION EDITION 2014

Qualification file

UBUNTU

IUT Cherbourg Manche

ARNE Julien, DIELA Afonso and KONATE Ababacar
2nd year DUT GEII students at the IUT of Cherbourg-Manche 05/22/2014

TABLE OF CONTENTS

PREAMBLE

The objective of this document is to describe the Vierzon robotics project to which we We are participating, as part of our 2nd year DUT GEII course, at the IUT of Cherbourg Manche. Our team is made up of three students: Julien Arné, Afonso Diela, and Ababacar Konaté.

We will detail the challenges of the project, how we organized ourselves and the work that we have carried out to meet these objectives.

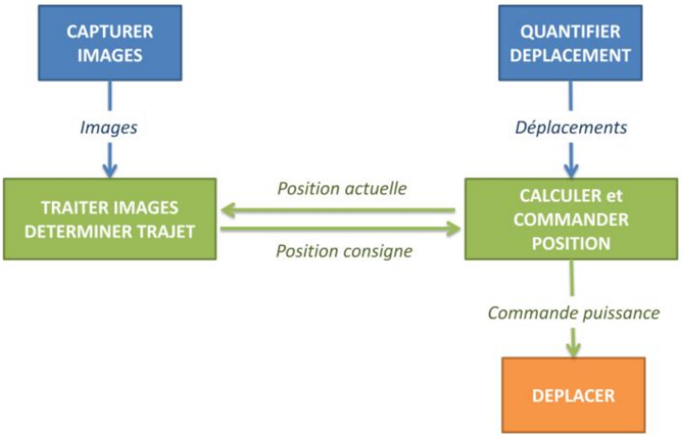
WHY UBUNTU?

"There is a word in South Africa, 'ubuntu', that sums up Mandela's greatest legacy. We are all connected to each other. There is a oneness in humanity.
We fulfill ourselves by giving of ourselves to others and caring for those around us."

Speech by US President Barack Obama
In tribute to Nelson Mandela

ROBOT CONSTITUTION

In an effort to minimize the number of sensors, we have, after careful consideration, established the following diagram: functional below. We have divided this diagram into two parts:



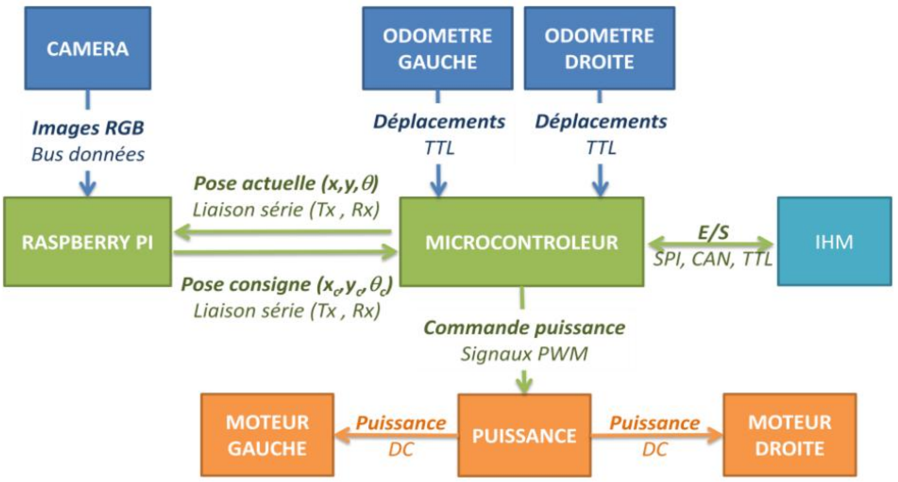
- **Quantify and carry out the movements :**

To calculate in real time the position and orientation of the robot and knowing the position of the arrival zone, control the motors so that the robot moves until its arrival

- **Capture and process images :**

To determine by processing the presence and positions of obstacles and determine an alternative trajectory to get around them

To achieve these different functions, we looked for the most suitable solutions to finally consider the following structural architecture:



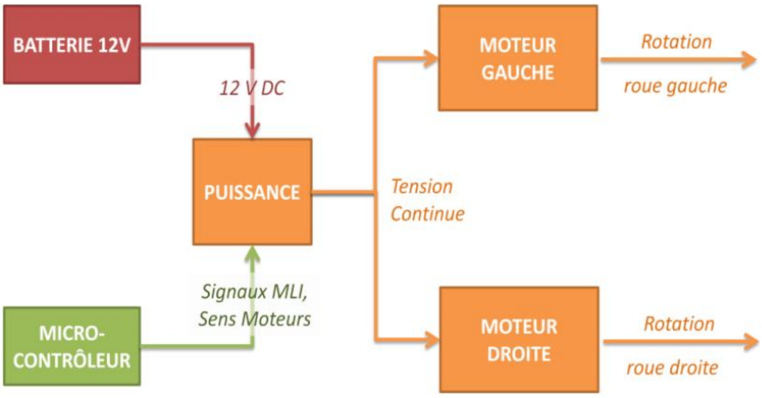
- A *Raspberry Pi card + camera :* to perform obstacle **detection** and **avoidance**
- A *microcontroller-based card + odometers :* to determine and control the **position** in real time
- A *power card :* to control the **motors**
- A *human-machine interface* (HMI) card: to obtain feedback and modify certain robot settings

In terms of interconnections:

- The *Raspberry Pi + camera* solution communicates with the microcontroller board to receive the position in robot course and emit the desired position, by means of a **serial connection.**
- The *microcontroller + odometer board* transmits PWM (pulse width modulation) **power signals** to the *power board,* in order to control the *motors.*
- The *HMI card* exchanges inputs/outputs with the *microcontroller card* using an **SPI link** to drive an **LCD** and analog and logic signals (all or nothing) to drive a **joystick**

POWER AND ENGINES

To move on the track, our robot uses **two motors imposed** by the competition: one driving the left wheel and the other the right wheel. Each of these motors is a **direct current machine** and is therefore controlled by direct voltage: it is the **12V lead battery** imposed by the regulations which must be used to power the motors.



The specifications being again dictated by the competition rules, our work was to **choose the components for the power electronics** controlled by signals from a microcontroller, in order to control the power of the motors and drive the robot's movement. For the microcontroller control strategy, we chose **pulse width modulation** (PWM) which is very widespread for this type of application.

I. The power card and the LM18200

This board's main function is to power the motors that allow the robot to move. In addition to the battery power, the board receives 4 signals from the microcontroller, for each motor:

- 1 signal for the direction
- 1 PWM signal controlling the supply voltage

Power card top view

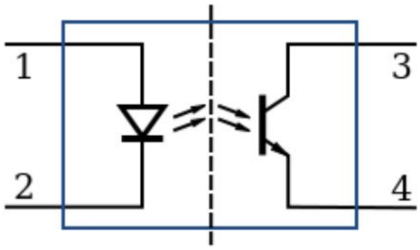
Since the microcontroller outputs cannot supply enough current to power motors, power electronic components (switches) are required, mounted in an "H-bridge" so as to relay the control signals via a power source (here the 12V battery). It is then the application of a PWM signal that will either close (high state of the PWM signal) or open (low state of the PWM signal) these switches, alternately allowing the current to pass from the power source to the motors.

For this we have chosen the LM18200 power component capable of delivering up to 3 amps continuously at the output via an H-bridge from a maximum power supply of 55V.

II. Electromagnetic compatibility issues

To avoid problems with electromagnetic interference when starting the motors (which could be destructive for the control electronics), we use optocouplers in order to have perfect electrical isolation between the two parts (control and power).

The optocouplers used here are HCPL 2200 integrated circuits.

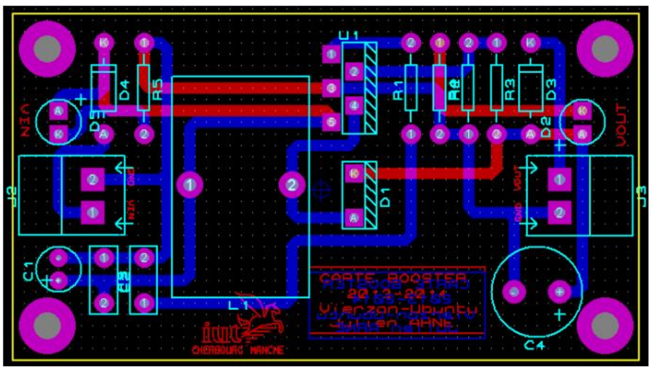


The HCPL 2200 integrated circuit

The insulation is done using photoelectronics with a LED that converts electrical energy into photons. (light) which is sent to a phototransistor and which the latter converts into voltage

III. Boost assembly: Boost

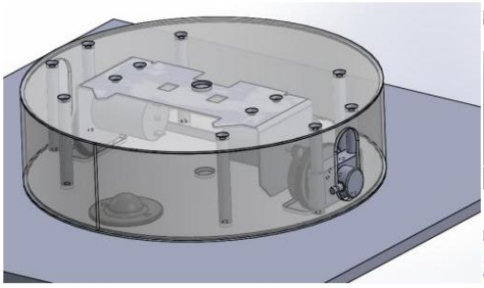
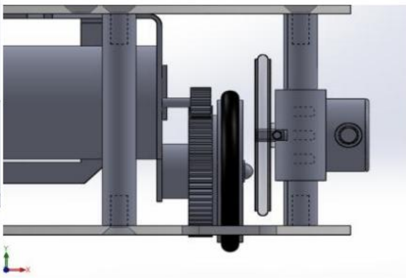
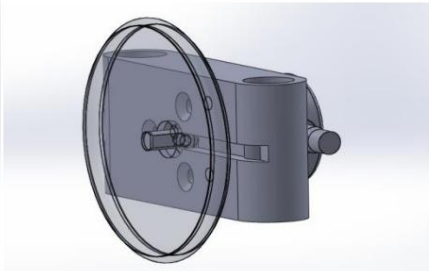
With the aluminum chassis, the weight of the robot did not allow it to move at a sufficient speed. To increase the speed, while using the 12V battery imposed by the regulations, we use a booster assembly, in order to increase the supply voltage to 19V.



MECHANICAL STRUCTURE

As **required by** the **regulations,** the robot must have a **mechanical structure** whose **dimensions** do not exceed those of the template. This structure must be able to contain the metal chassis (including motors, wheels and gears), the 12V lead-acid power battery required by the regulations, and all electronic cards and sensors used.

In addition, as specified in the functional diagram, we wanted to use sensors to quantify the movements, which by the rotation of 2 floating wheels (one left and one right, non-driven) would record the actual movements of the robot.

The challenge for doing **odometry** was therefore to create a **reliable and robust mechanical** structure to **limit inaccuracies** in measuring displacements, while allowing the robot to move quickly on the track.

Vue de dessus

We thought of the simplest shape **to design and make :** it was a tin can that inspired us! The circular base is also very practical because the robot does not bump into corners. It is also the most used shape for robot vacuum cleaners and robot lawnmowers.

Vue de profil

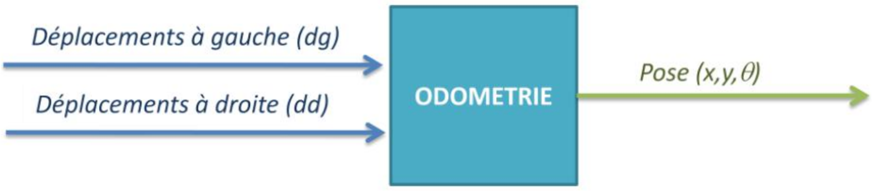
However, with this shape, a shock with an opposing robot could lift the robot and move it without these movements being recorded by the incremental encoders. To limit these consequences, it was therefore necessary to either choose a heavy material which would impact on the speed, or to consider another shape. The latter was therefore made of **aluminum** and **PVC**



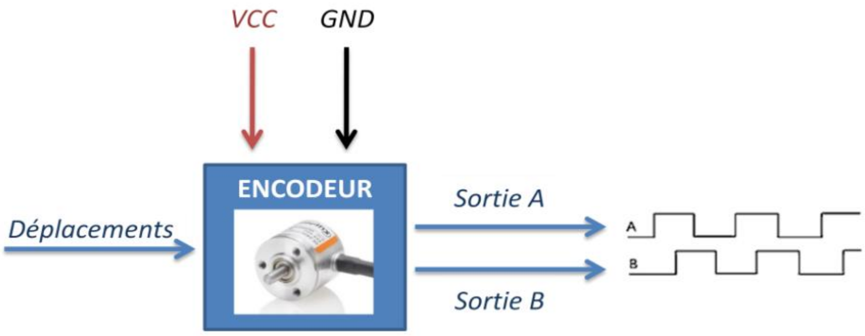
ODOMETRY AND MOVEMENTS

According to the rules of the competition, to win, our robot must be able to move from one corner of the 8x8m square track to the diagonally opposite one, then stop and finally explode the balloon it is carrying. Knowing the **dimensions of the track** and the **initial coordinates** of our robot, the simplest way we thought of was to **calculate** in real time the **position and orientation** of the robot on the track, and to control its parameters in order to guide it towards its arrival and finally stop it.

In order for the robot to know its position on the track in real time, we were interested in odometry as a usage technology, using **incremental optical encoders.** The principle of odometry consists in **determining** the pose of the robot **(position** + **orientation)** at each desired moment, by measuring and integrating small displacements traveled to the left (dg) and to the right (dd).



An incremental encoder is a displacement sensor which, by rotating a mechanical axis, will generate edges on an electrical signal at the output. At the end of **the movements,** we will therefore see a **periodic TTL signal** (0 – 5V) appear at the sensor output, the period of which will be an image of the movement. The 2 outputs of the sensor present 2 identical **signals** but **out of phase** with each other by **+ or – PI/2** depending on whether the axis rotates **clockwise** or **counterclockwise.**



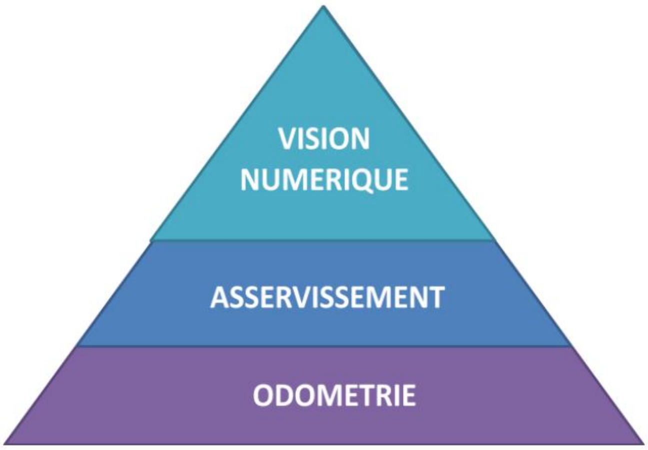
Incremental encoder from the manufacturer
Kübler : 05.2400.1122.1024

For reasons of mechanical precision, we finally chose the incremental encoder from the manufacturer **Kübler, the 05.2400.1122.1024.** It has an **excellent resolution** for the desired precision **(1 axis revolution = 1024 edges),** with a **limiting speed of 12000 rpm.**

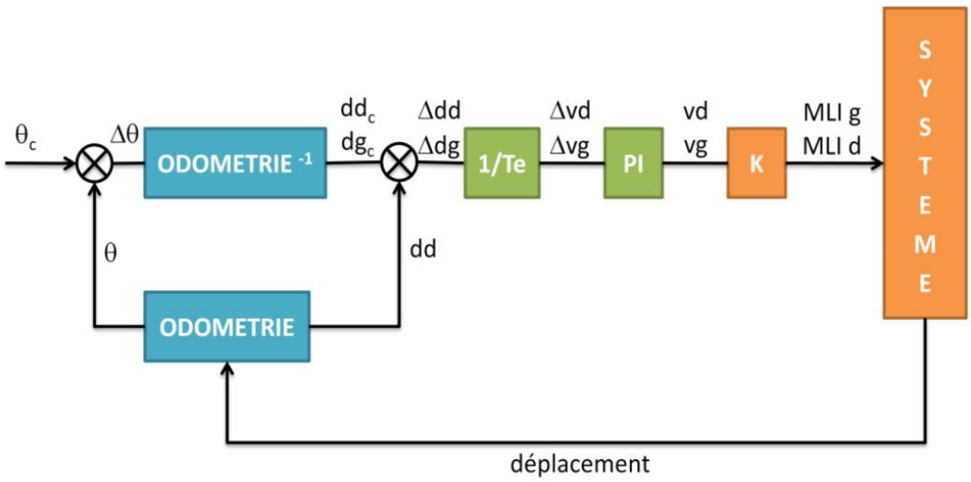
For each encoder, a microcontroller is used to **trigger an interrupt** as soon as a **rising edge** appears on **output** A. The number of edges counted and decounted are then regularly sampled to calculate the new position and the new orientation.

SERVIBILITIES

To get the most out of odometry, we have enslaved our robot so that only one instruction orientation and position is sufficient to simply control the robot's movements.



Here, the system to be controlled corresponds to the robot and its motors. We therefore create an orientation control (diagram above) and a position and speed servo. We use the data returned by the odometry (wheel speeds, robot orientation) to complete the system, then we sample regularly (period Te) and correct the motor control using a proportional integral corrector.



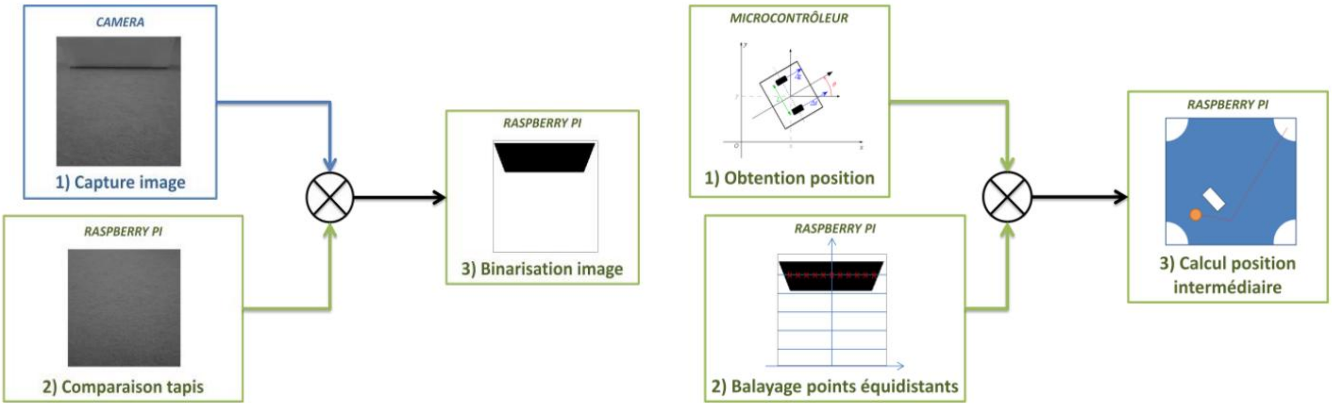
Equivalent servo control is achieved for position and velocity.

DIGITAL VISION

Still with the aim of reducing the number of sensors to a minimum and so that the robot perceives its environment (i.e. fixed and moving obstacles), we decided to use digital vision using a **camera,** the *PiCam,* connected by a bus to its motherboard, the *Raspberry Pi,* a real miniature computer.

The Raspberry Pi card with its PiCam

It is certain aspects of the competition rules that allow us to propose this solution. Indeed, **the homogeneous and sufficient lighting,** the **blue** color of the **track,** and the identical characteristics as well as the **white** color fixed obstacles , greatly simplify the task.



In **C++ language,** we use libraries to capture **30 images per second,** then we perform **processing** above in order to distinguish the obstacles of the carpet. Then, knowing the position of the robot by interrogation via a link microcontroller card serial, we **position the obstacles** in memory. Finally, by an **A* algorithm,** we calculate an intermediate position to avoid the obstacles and reach the arrival position.