Submitted for the Degree of MEng

in Electronic and Electrical Engineering

2009/2010

Eurobot 2010

Sébastien BRULAIS

Supervisor: Dr David Harle

Second assessor: Prof Robert W. Stewart

Acknowledgments

I would like to thank Dr David Harle for his support and for all the time he spent to make the project possible. I would also like to thank the company Schwarzer Precision which sponsors our team, and gave us the possibility to use one of their miniature vacuum pump. An other special thanks to both mechanical and electrical workshops people who made a great job for the robot.

Declaration

I, Sébastien Brulais, declare that this work has not been submitted for any other degree/course at this University or any other institution and that, except where reference is made to the work of other authors, the material presented is original and entirely the result of my own work at the University of Strathclyde under the supervision of Dr David Harle.

Sebastien Brulais

Abstract

Eurobot is a European robotic contest which takes place every year. It regroups more than 400 teams all around Europe which compete with the same rules. All the robots are completely autonomous and face each other on a same table. This year, they have to collect different kinds of objects on the table while they keep avoiding their opponent.

This project was shared with an other EEE student -Paul Monsinjon- and aimed to build a complete robot to take part in this competition.

One of the key success to win matches during the event, is to have a reliable and accurate mechanical solution. This report will detail how the robot parts has been designed, and why the particular solutions have been chosen.

During this project, two main board has also been developed and their complete design will be exposed. The first is the power board which provides several power supply to the other boards of the robot. The second one is the daughter board which interfaces the robot with the intelligence board.

In addition, several other tasks accomplished during the project will also be explained such as the vision system,the way how the robot avoids its opponent and the on-board graphical user interface.

Finally, since this was a huge project including difficulties, the encountered problem will be exposed. The future work will also be discussed, in order to keep the project alive for the next years.

Contents

List of Figures

1 Introduction

1.1 Report outlines

This report will first deal with the project background and motivations

Next, an overview of the project is described, showing how the tasks has been divided and what kind of tools has been used during this project.

It will then show in detail the three major developments of this project which are the mechanical, the power board and the daughter board design.

The additional parts of the project are also explained, including the vision system, the avoidance and the graphical user interface.

Finally, the latest part will draw conclusion and present the encountered problems and what are the future work possibilities.

1.2 Eurobot

1.2.1 Presentation

Eurobot is a european robot contest which takes place every year. Each country has its own qualification phases, which sees the most important number of participant in France: around 250 teams.

It takes place every year in La Ferté Bernard, and the 2010 edition will stand between the 13^{th} and the 15^{th} of may. During these three days, several matches will be organized with two robot competing each other on the same table.

1.2.2 Rules

The main lines of the rules are the following:

- 90 seconds matches
- Completely autonomous system
- Robot dimension: 120 cm non-deployed perimeter, 140 cm deployed, 35 cm height
- Fair play and opponent avoidance
- No more than 48V supply voltage

1.2.3 2010 Theme

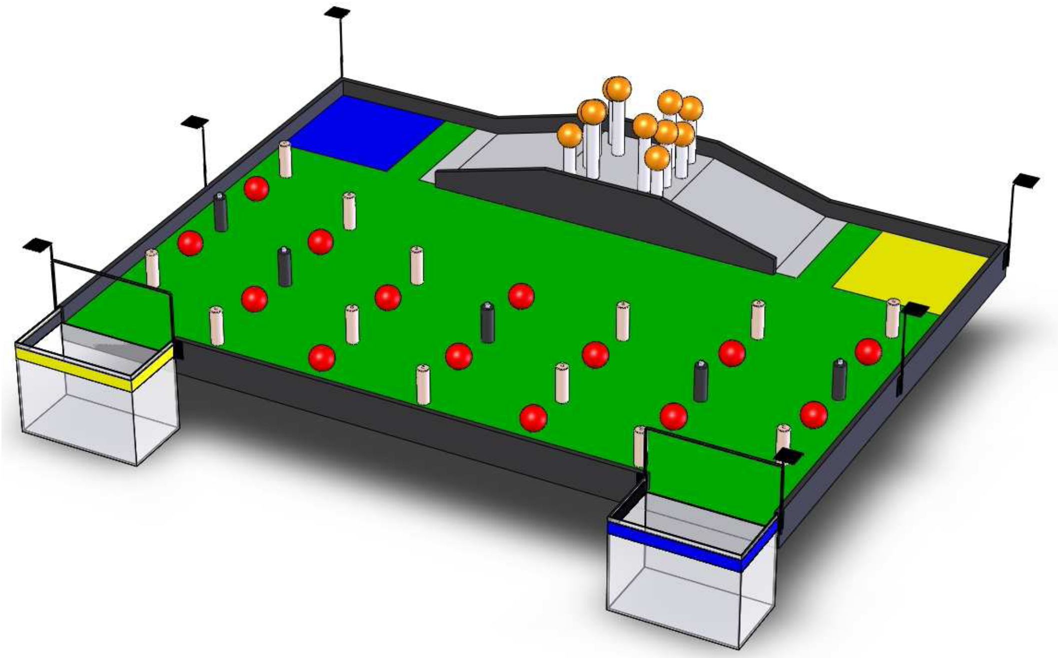


Figure 1.1: 2010 Game table configuration

The theme for this edition is 'Feed the world': the robots will have to gather different kinds of fruit or vegetable, and put it in their associated basket. There is three different types of objects which worth a different amount of points depending on how difficult they are to grab:

- Red balls: tomatoes, 150 points
- Cylinders: corns, 250 points
- Orange balls: orange, 300 points

Note that there is two colors for the cylinders: white which are regular corns, and black which are fake corns. The fake ones are anchored to the table with a screw, so the robots have to avoid them. Moreover initial configuration of the corns is not known in advance, which means the robots have to probably find a way to detect the difference between these.

Each robot starts in its attributed colored square (yellow or blue) and its unload area is in the opposite side of the table. The team which has the more points in his basket at the end of the 90 seconds wins the match.

At least five qualifications matches are performed for each team, and after that, the 16 best teams plays final qualification session (8th quarter semi-final and final). The two firsts teams of the contest are qualified to the Eurobot final which takes place in Switzerland this year, where they will face all the other best European teams.

1.3 Motivation

This project is a student defined subject, which is shared with an other fifth year EEE student : Paul Monsinjon. This proposition has been made because both of us have already been involved in a team in 2009, taking part in the same competition.

The desire of renewing this experience, but this time with a smaller team is one of the key motivation. It also taking up a huge challenge because the other teams are usually composed of more than eight people. Additionally, the other participants can use their previous year robot, board and code; in opposition to this where everything had to be build from nothing.

1.4 Objectives and aim

The aim of the project is straight forward: to build a complete robot in order to take part in the Eurobot qualifications in France. This includes the mechanical design, the development and the realization of the required boards, and the software design of the robot. The next part of the report shows what exactly was needed, and how the tasks has been divided between the two teammates of the project.

2 Project overview

2.1 Robot general organization

Below is a diagram showing the robot is divided in multiple parts as shown on figure 2.1.

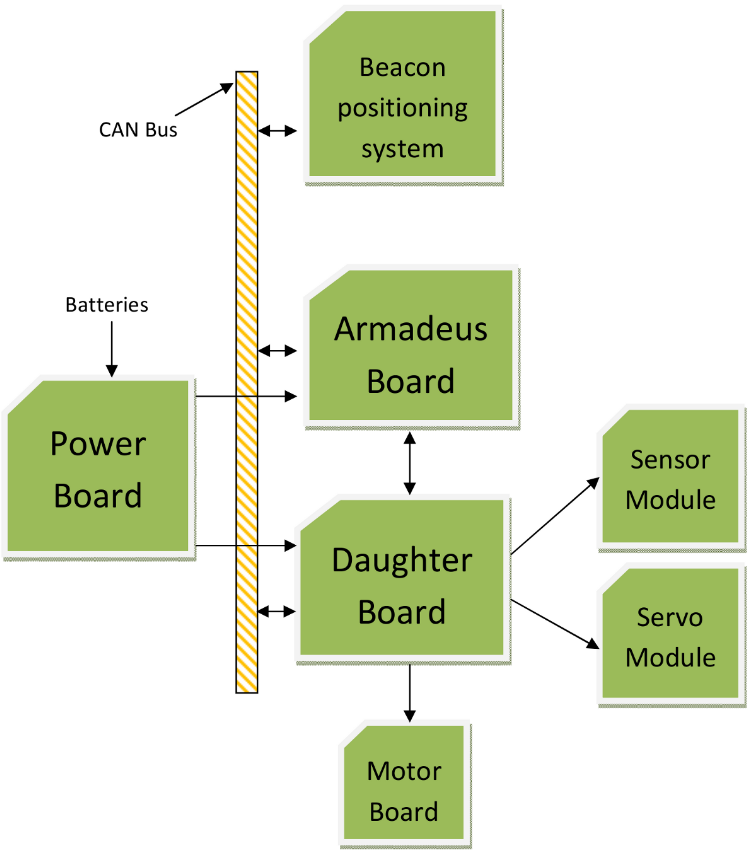


Figure 2.1: The robot general organization

2.1.1 CAN Bus

The CAN [1] bus is an industrial serial communication bus, widely used in new vehicles technologies. For this project, it allows the different systems to communicate with

[1] Stands for Controller Area Network

each others even under harsh environments, such as electromagnetic fields created by the motors.

2.1.2 Armadeus board

The Armadeus board is a board already developed by a company and bought before the project started. This board is the 'brain' of the robot, and is mainly used for the intelligence and the motion control. It has an ARM 9 i.MX27 processor which runs at 400 Mhz with an embedded Linux operating system. It also has a Spartan 3A FPGA, and many other features such as

- 10/100Mbits Ethernet port which was mainly used for an SSH [2] connection to debug the robot remotely.
- USB host connection, which can receive an USB stick, or a Web Cam for instance, and can be treated under the embedded Linux just as if it was on a traditional computer.
- Sound card with input and output lines
- PS2 socket for keyboard and mouse
- RS232 for serial transmission
- a SD card reader
- up to 107 available input/output

2.1.3 Motor board

The motors board is a small board with H bridges which allows the main motors of the robot to be driven from a logical signal.

[2]Secure Shell which is a network protocol that allows to access remotely and securely to a device console

2.1.4 Beacon positioning system

The beacon positioning system, is a system used to ruffly locate the robot's opponent, using triangulation of optical infrared signals. Therefore, the intelligence can takes decision depending on where the other robot is, in order to avoid contacts and to save time.

2.1.5 Servos module

The servos module is a small board capable of controlling up to 8 classical servo motors, and also numerical servo motors via an half duplex serial transmission.

2.2 Tasks repartition

All the parts described above have been developed by the other project teammate, and those which hasn't been mentioned yet will be detailed later in this report. As a result, the tasks repartition of the project was:

Sebastien Brulais	**Paul Monsinjon**
Power board	Motion control
Daughter board	Intelligence
Vision and GUI	Beacons
Sensors module	Servos module
Mechanical Design	

Table 1: Project tasks repartition

2.3 Project tools

2.3.1 Softwares

The softwares used in this project were:

- Cadsoft Eagle as a PCB editor
- Solidworks for all the mechanical design
- Codeblocks for the Armadeus board C coding, with the SDL library for the GUI and OpenCV library for the vision system.
- Mplab IDE for the PIC microcontrollers developments; a user guide for this tool is available in appendix C

2.3.2 SVN

This project, as described before, was shared with another teammate, consequently it required some organization tools.

In order to be able to work on separate station but eventually on the same files, a SVN [3] repository has been opened on a server. Each time a member of the project has finished a part of his job, he 'commits' the files which has been changed; then the other member can 'update' these files on his own computer. Therefore every single file used for the project (code, PCBs, orders, media, datasheets, resources, ...) is stored onto a server, and is available on any computer which can access to the Internet.

There is also the possibility to 'revert' current files from a previous stable version, in case there is a problem.

[3] stands for Subversion, which is a revision control system

2.3.3 Wiki

A wiki website has been created (http://mp-net.fr/elder) in order to share the project work and observation to everyone. It is an open-source philosophy because it can help other teams to take part in the competition for the next year, and also because it is the spirit of the contest: sharing knowledge. Furthermore, it contains the project news and informations for people which are interested to see how the project is going on.

3 Mechanical design

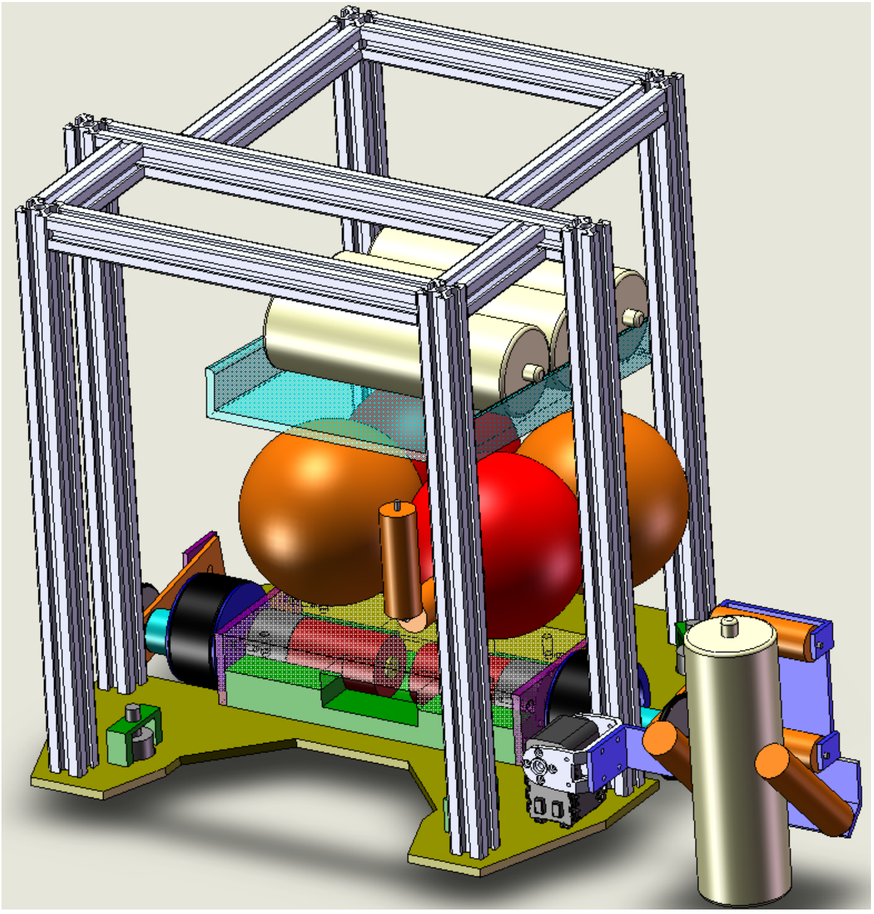


Figure 3.1: The overall robot mechanical design made so far

3.1 Introduction

Even if this project took part in a EEE department, building the robot required a lot of mechanical design, which represent a major part of the overall project success in the competition.

With a small background in this domain and the help of a CAD[4] software, drawings have been made and submitted to the mechanical workshop in order to make all the robot parts.

[4]Computer Aided Design

This part of the report describes why and how the mechanical parts have been made.

3.2 Parts description

3.2.1 Chassis

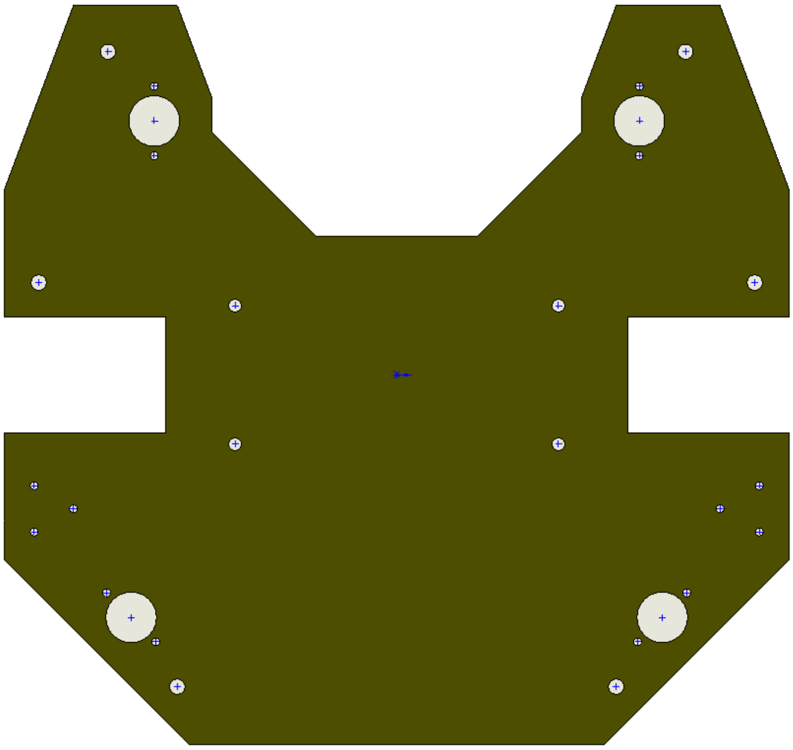


Figure 3.2: Chassis CAD design

The chassis has to be strong enough to carry all the robot system and, more importantly the objects which may weight all together up to 2.2 Kg. To achieve that, a 5 mm thick aluminum sheet is used and milled in such a way that the overall perimeter stays less than 120 cm, according to the contest rules.

The shape has been designed to be able to keep either a ball or a corn at the front of the robot, which can then be grabbed and stocked inside the robot.

3.2.2 Caster balls

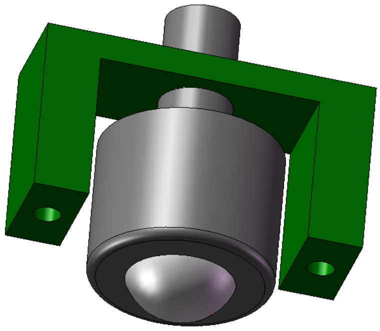


Figure 3.3: Caster ball mount CAD design

Because there is only two propulsion wheels, the robot needs to be equipped with caster balls in order to be stabilized. These allow a smooth movement in all directions, giving 'extra wheels' to the robot.

Four of them are placed in the corners of the robot, and each vertical position can individually be adjusted according to the game area flatness.

3.2.3 Encoder mount

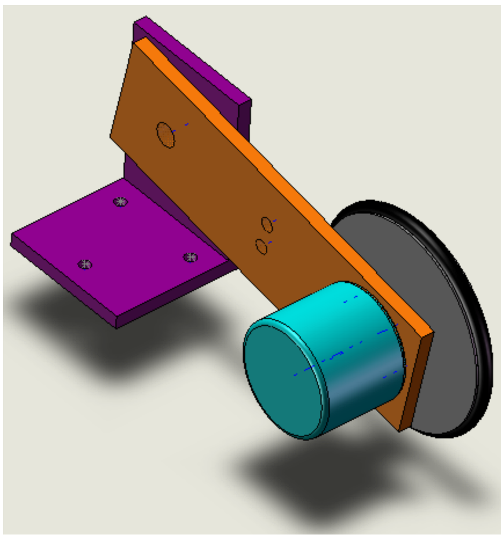


Figure 3.4: Encoder mount CAD design

The encoders are used to measure the robot travel, even if the main wheels slip. The tables during the contest are made of painted wood, which leads to the fact the surface is not perfectly flat. Consequently, they are mounted separately and in such a way that their associated wheel always touch the floor using the encoder's weight and the gravity. The wheel tire is simply an O-ring joint which can be easily changed if it would be worn.

3.2.4 Motor block

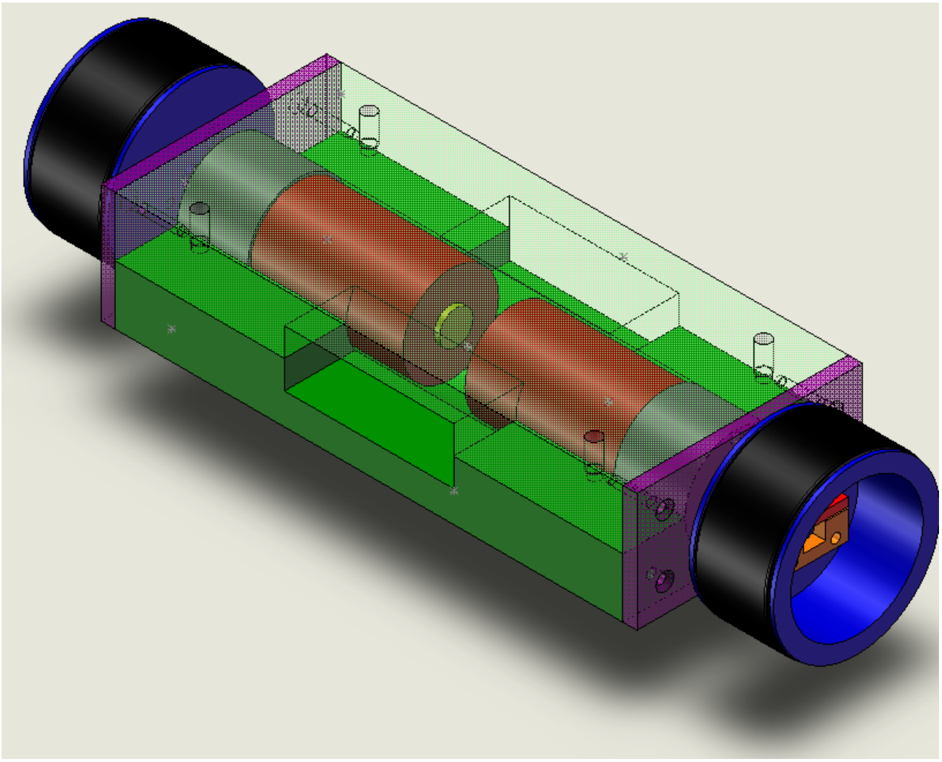


Figure 3.5: Motor block CAD design

The motor block is made in a way that the two motors are as aligned as possible, thus the motion control can benefit a good accuracy. If there were on separate mounts, holes, screws and milling tools may introduce unwanted iprecision in the motion transmission. Each motor is maintained firstly by the cylinder shape inside the motor block, and secondly by three M4 screws. With these condition the motors are strongly anchored and there is no chance they can move even if heavy loads are applied to the robot.

3.2.5 Corn grabbing system

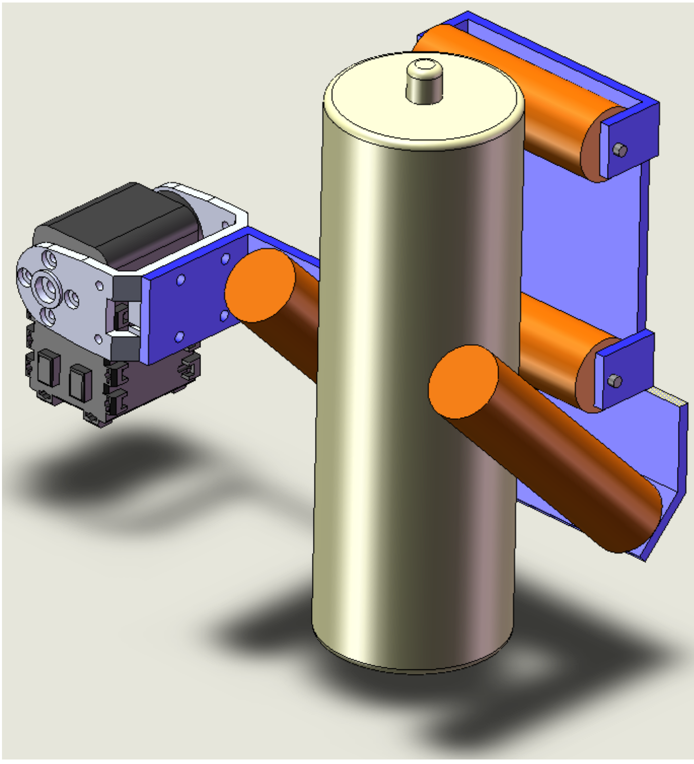


Figure 3.6: Corn grabbing system CAD design

Two grabbing systems are used to gather the corns, one on each side of the robot.

The overall system moves thanks to AX-12 servo motors, which are strong (12 kg/cm at 7V) and adjustable through an half-duplex serial communication.

The corn is lifted from the table with several rollers, then stocked inside the robot, and finally unloaded in the basket next to the table.

3.2.6 Profile framework

At this stage of the project, all the mechanical solutions and implementations are not completely decided.

The use aluminum profile structure allows a great modularity in terms of placement and possible modifications.

By putting nuts in the rail as shown in figure 3.7, almost everything can be anchored everywhere along these profile such as sensors, motors, board mounts, emergency stop button…

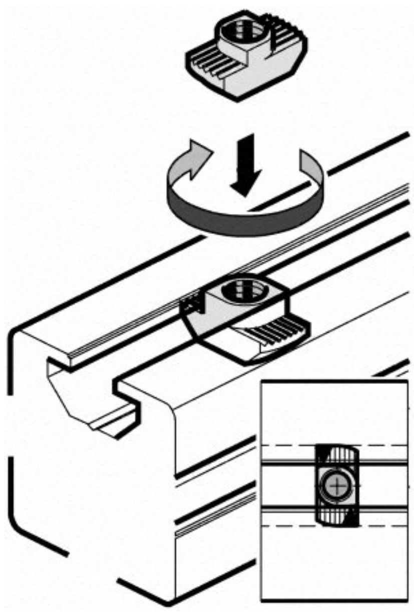


Figure 3.7: Profile and its anchor system

4 Power board

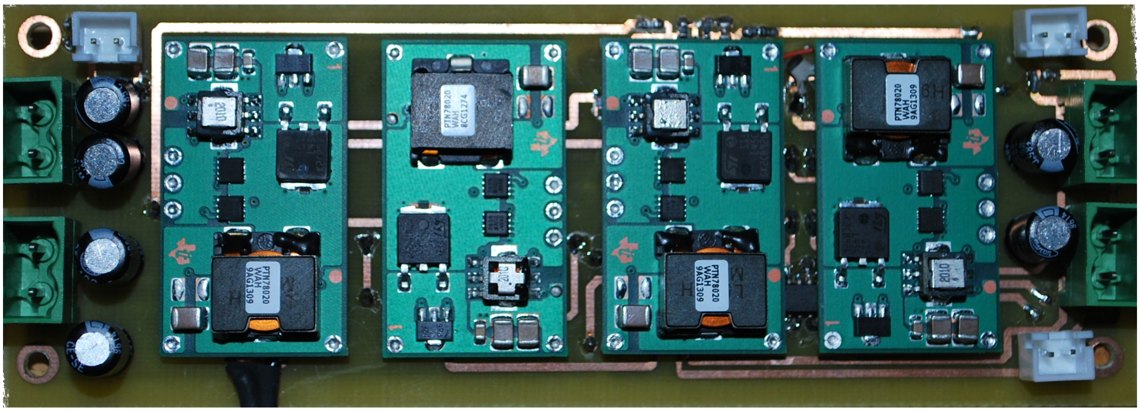


Figure 4.1: The power board

4.1 Introduction

The power board plays a fundamental part in the overall robot: to convert the batteries voltage into several reliable power supply according the other robot elements requirements. It is also a smart board capable of monitoring the outputs and the motors currents.

All these informations are spread into the robot through the CAN bus. It also reports any anomalies by sending warnings, for instance if a measured voltage isn't what it should be. This point is very important because the robot is supplied with Li-Po batteries which requires to not been too much discharged otherwise they can blow up. Thus they need to be monitored, and the robot should be stopped when it's critical.

The CAN bus also allows the intelligence board to send orders to the power board, such as put on or off a power supplies, for instance at the end of a match the intelligence should decide to power off the motors.

4.2 Requirements

The power board had to provide these reliable supply voltage:

- Motors: 12V 3A
- Armadeus board: 10V 2A
- General purpose: 5V 6A (servos, actuators, encoders, ...)
- General purpose: 12V 6A (AX-12 servos, DC motors, ...)

The extra feature requirements were:

- Voltage monitoring
- Motor current monitoring
- CAN bus communication
- notification LEDs

The power board had also a 50x140 mm dimension requirement, in order to be placed in a box with the other boards.

4.3 System design

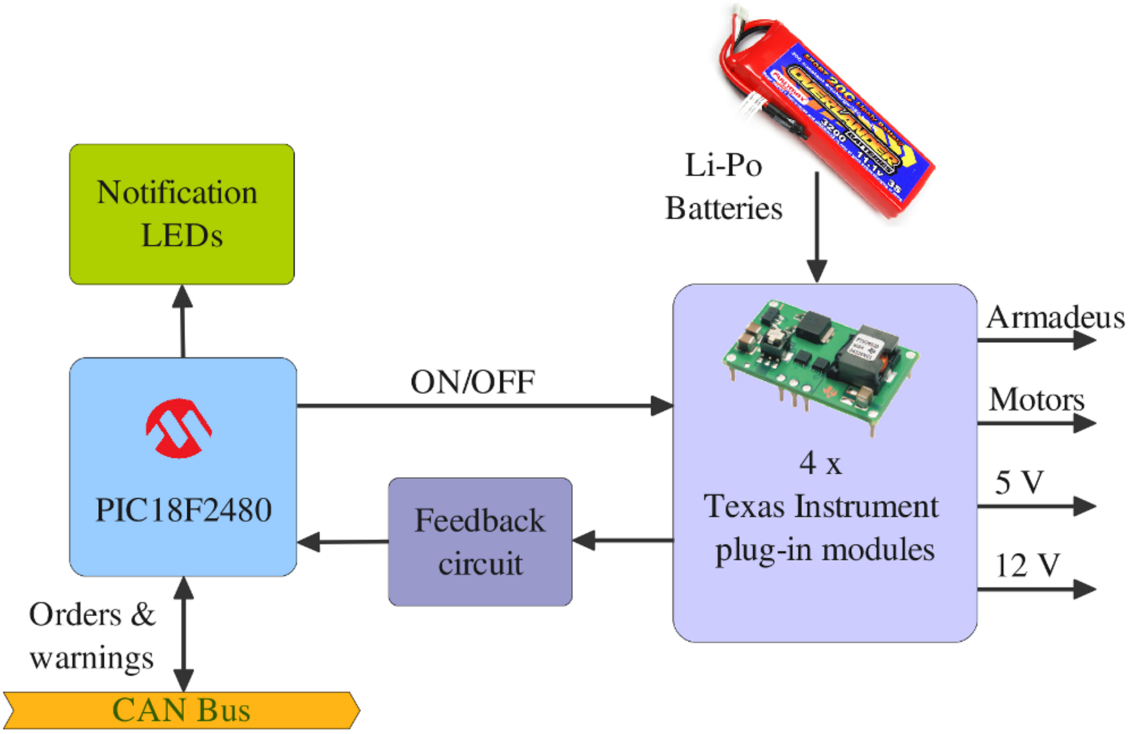


Figure 4.2: Power board system description

4.3.1 DC-DC converters

The most efficient way to convert a DC voltage to another (lower in this case), is to use a switching-mode converter which is more efficient than linear voltage regulator. By measuring the output voltage as a feedback, this conversion method is able to insure a fixed value whatever the current load is. For instance, when the robot is stopped and the motors start to run, the current change from zero to few Amps in a short time which may bring on a voltage drop.

Texas Instrument offers a very wide range of this kind of converter, which also can be switched on/off easily by a microcontroller.

The choice according the previous requirements is the module PTN78020W which has the following characteristics:

- 7 to 36V Input
- 2.5 to 12.6V Output, 6A
- Enable pin

4.3.2 Microcontroller

First of all, a microcontroller family had to be chosen. This choice was the Microchip PIC18F family mainly because the associated programmer was already available. It is a 8 bits microcontroller architecture, easy to develop (a lot of documentation is available on Microchip website), and can be ordered in free samples.

The microcontroller used for the power board is the PIC18F2480, which features fits perfectly with the requirements:

- CAN Bus
- 10 bits ADC, 8 channels
- 28 pins

The micorcontroller runs on this board with a 8 Mhz crystal, which is enough for the application. The supply level is 5V, which is provided by a dedicated regulator.

4.3.3 CAN Bus interface

As seen before, the CAN bus is connected to most of the robot boards. The CAN feature on the microcontroller can't directly be interfaced with the bus because the associated pins of the PIC18F2480 are in TTL levels. The CAN bus works with two differential line CAN H and CAN L in order to be robust under harsh environments. Therefore, to convert the voltage levels from to the microcontroller to the bus, a CAN transceiver is used to provide differential transmit and receive capability at a speed up to 1 Mbps.

Note that at each end of the CAN bus, a terminal resistance of 120Ω should be placed between the two bus wires. The space for this eventual resistance has been allocated on the power board circuit.

4.3.4 Feedback

In order to measure voltages without interfering it, several functions have to be placed between the measure target and the microcontroller's ADC.

- a voltage divider to pull down the voltage to an acceptable level for the microcontroller
- a low pass filter to remove the eventual noise, realized with a RC circuit
- a voltage follower to adapt the impedance and not disturb the input

All these functions are regrouped in a single circuit as shown in figure 4.3

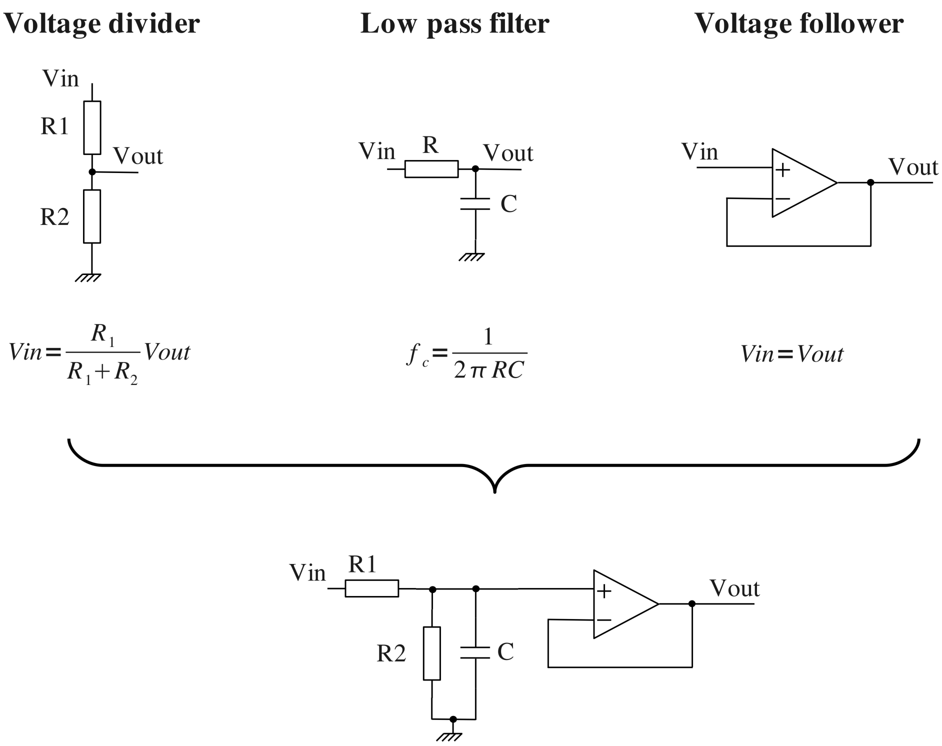


Figure 4.3: Feedback circuit

The filter cutoff frequency can be determined as described bellow.

The impedance of $R_2//C$ is given by:

$$Z_2(j\omega) = \frac{\frac{R_2}{jC\omega}}{R_2 + \frac{1}{jC\omega}}$$

$$Z_2(j\omega) = \frac{R_2}{1 + jR_2C\omega} \tag{1}$$

The transfer function is given by:

$$H(j\omega) = \frac{Z_2}{Z_2 + R_1} \tag{2}$$

Using equation (1):

$$H(j\omega) = \frac{\frac{R_2}{1+jR_2C\omega}}{\frac{R_2}{1+jR_2C\omega} + R_1}$$

$$H(j\omega) = \frac{R_2}{R_2 + R_1} * \frac{1}{1 + j\frac{R_1 R_2 C}{R_1 + R_2}\omega} \tag{3}$$

So the gain, or the division ratio of the circuit is $\frac{R_2}{R_2+R_1}$ and the low pass filter cut-off frequency (in Hz) is given by:

$$f_c = \frac{1}{2\pi\frac{R_1 R_2 C}{R_1 + R_2}} \tag{4}$$

The voltage monitoring is not faster than 20 Hz so the components value can be chosen:

$$R_1 = 220k\Omega$$

$$R_2 = 82k\Omega$$

$$C = 100nF$$

These values give $f_c = 26Hz$ and a 0.27 ratio which is enough to measure voltage level up to $5/0.27 = 18.4V$.

4.3.5 Program structure

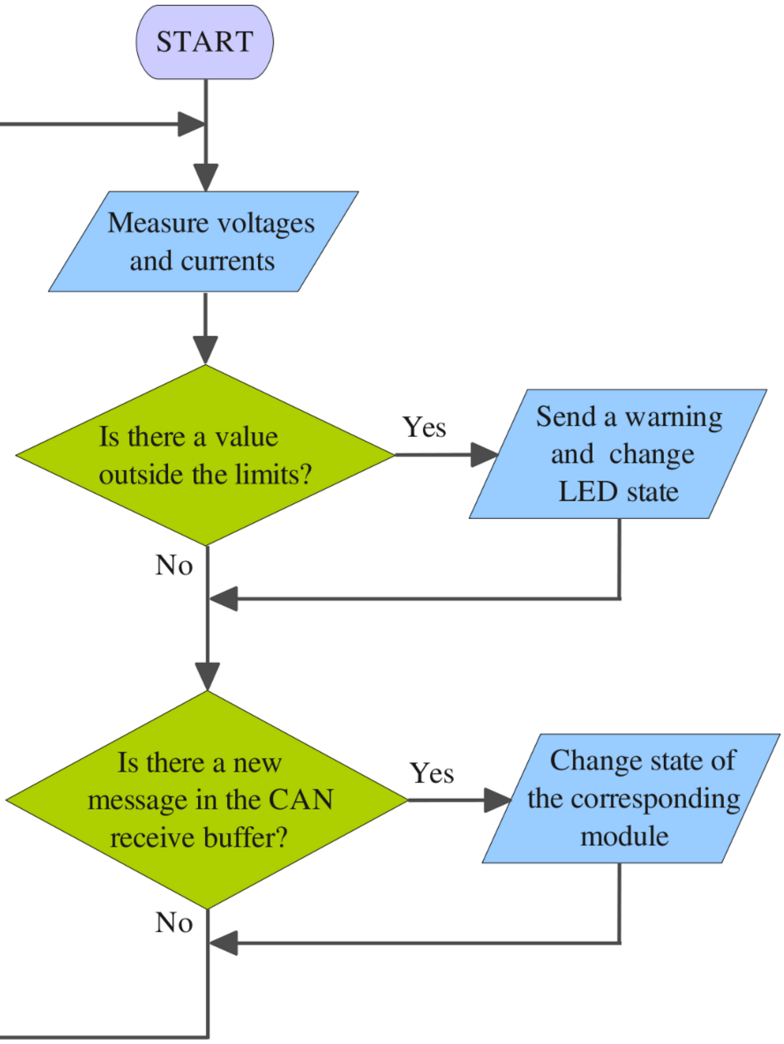


Figure 4.4: Power board program structure

The power board functioning is straight forward.

First, when the microcontroller is powered on, several initialization are made, including the ADC configuration and the CAN parameters settings.

Then in a main loop, the ADCs are read to check if the levels (from the four modules, the Li-Po batteries, and the two motor currents) are correct. If one level doesn't remain inside the defined low and high threshold, a warning message is sent through the CAN bus, with the corresponding source of failure.

Finally, if there is a new message received from the intelligence, such as turning on or off a module, the power board execute the asked order.

Note that the bi-color LED associated with each module are green when it is on, red when it is off and blinking red when there is a failure. There is also a bi-color LED for the batteries, which is green when it is on, and green and red flashing when the batteries level comes bellow 14.8V (3.7V per cell).

5 Daughter board

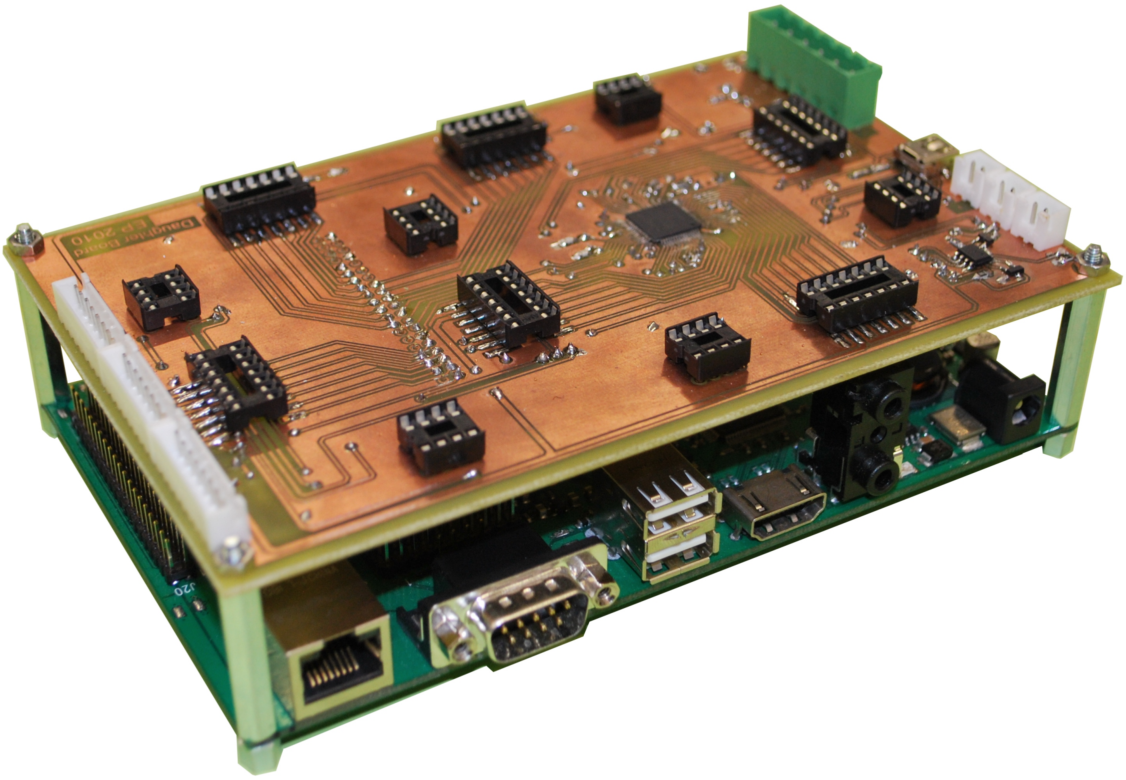


Figure 5.1: The daughter board stacked over the Armadeus board

5.1 Introduction

The aim of the daughter board is to interface the Armadeus board (which is used for the intelligence and the motion control) with everything else on the robot such as sensors, actuators, servos, motors, etc. . . Modules can also be plugged on it according to the robot needs and be interfaced with the Armadeus FPGA or the daughter board's own microcontroller. Thus each year, small boards can be developed in line with the project theme and the chosen mechanical solutions.

5.2 Requirements

Here were the electrical and mechanical requirements before the design started:

- 100 x 160 mm board : the board has to be the same size as Armadeus board in order to be stacked over it.
- 45 x 45 mm plug-in modules (up to 6 can be plugged) with on each:
  - 3.3V logic supply
  - 5V and 12V power supply
  - CAN Bus
  - UART (serial transmission)
  - 8 GPIO [5]
- Encoders interface
- Motors interface (PWM [6] and directions)
- Fan control

Because the Armadeus board belongs to a member of this project, the next years it won't be available. So the daughter board is meant to become a 'mother board' on which can run the motion control (PID) and the intelligence of the robot.

[5] General Purpose Input Output
[6] Pulse Width Modulation

5.3 System design

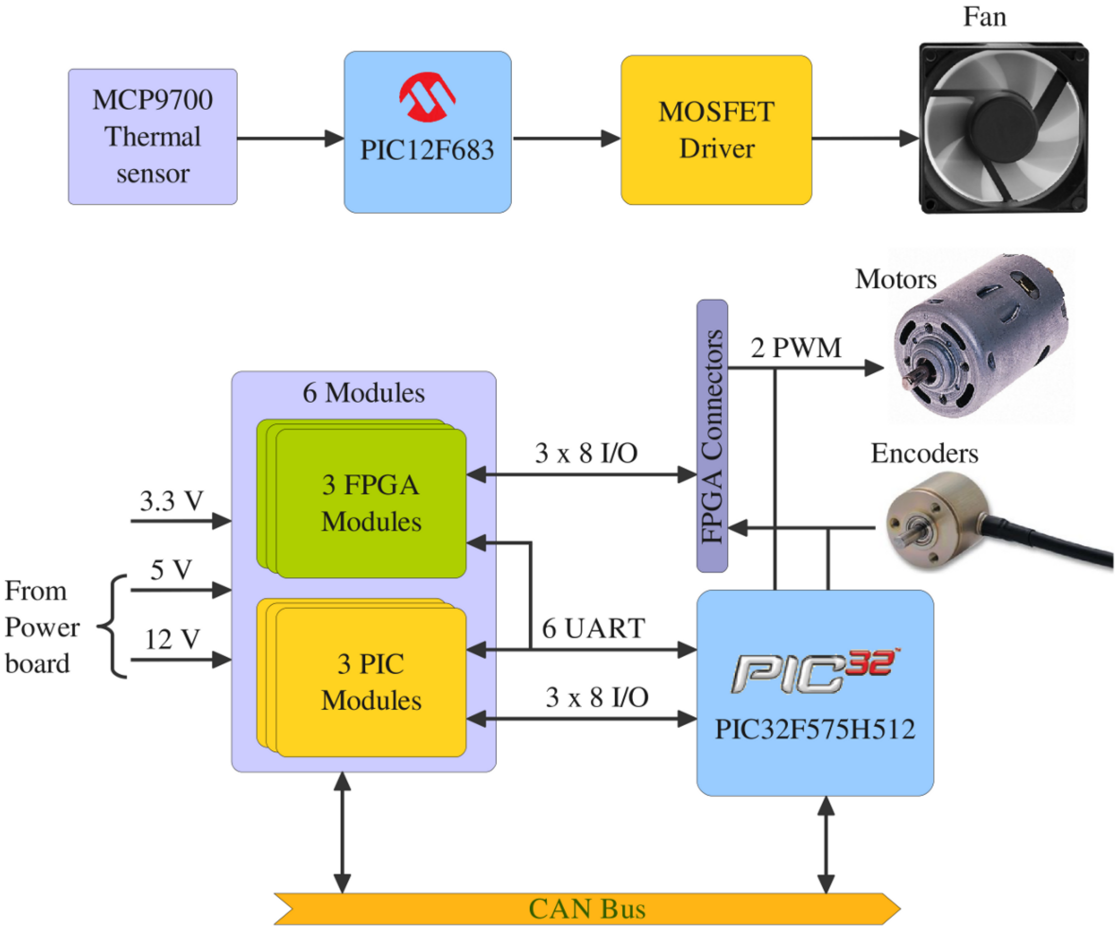


Figure 5.2: Daughter board system description

5.3.1 Microcontroller

The choice of the microcontroller was based on the requirements, and first an MSP430 from Texas Instrument was chosen. But after two weeks of trying to make a programmer for this type of microcontroller without success, a PIC32MX has finally be selected instead.

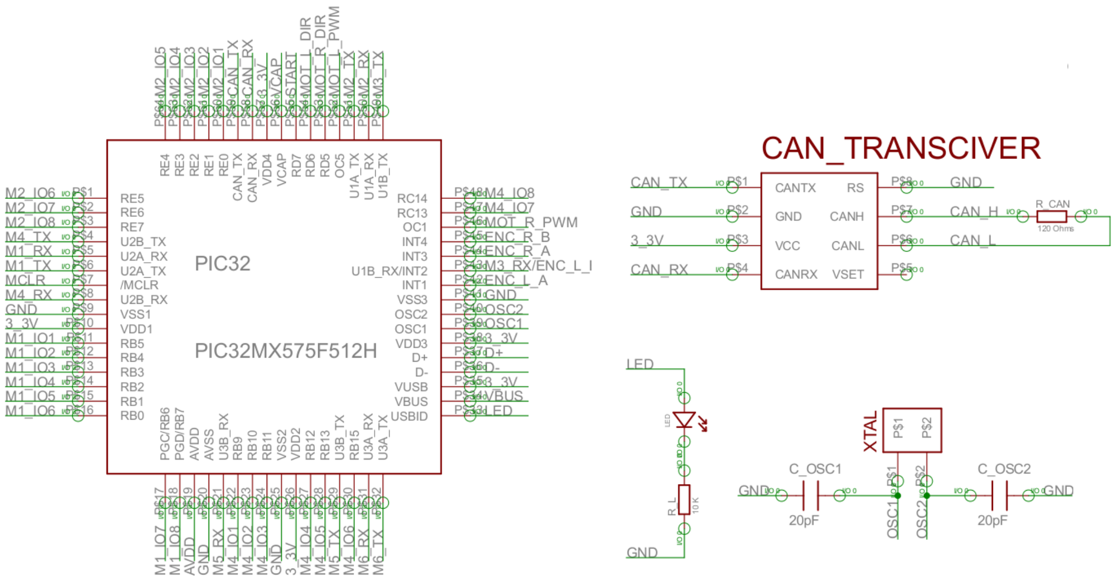


Figure 5.3: Daughter board main microcontroller, with its associated components

The PIC32 is a 32 bits architecture microcontroller family from Microchip, and the PIC32MX575F512H has the following features:

- Max Speed: 80 Mhz
- 10-bits ADC 16 channels
- 5 PWM outputs
- 5 timers
- 53 I/O pins
- 6 UARTS

All this features were enough to fit to the requirements, moreover, this chip could be acquired for free, with the Microchip free samples process.

The programmer used for for this is a PICkit 3; more information on how to use it are available in appendix C.

Note that the microcontroller is also interfaced with the CAN bus, and the implementation is the same as described before for the power board. A LED has been putted on a free pin at the end, which is useful to debug or to make the first 'led blinking test' for example.

5.3.2 Fan control

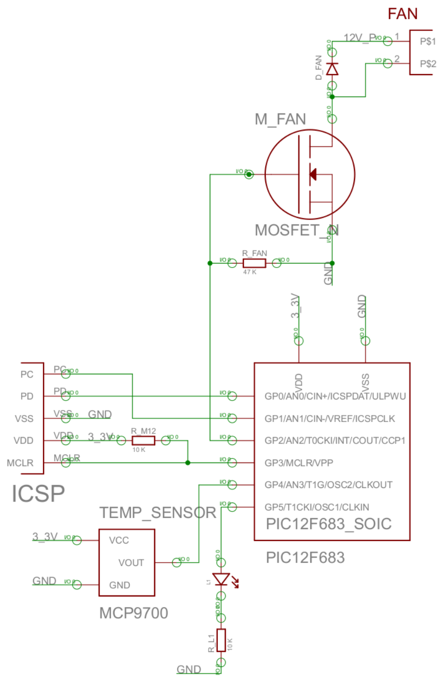


Figure 5.4: Fan control circuit

The Armadeus board, daughter board, power board and plug-in modules are all mounted in a same box. This configuration generate heat which has to be cooled by a fan in order to have correct performances.

A small dedicated microcontroller is used to realize this function, which again has been chosen from the Microchip manufacturer. It is a PIC12F683 which is small (8 pins), has got an ADC for the temperature reading and a PWM output for the fan speed setting.

The fan is driven by a MOSFET which is triggered by the PWM, and gives the possibility to apply a 12V supply voltage directly on it.

A freewheeling diode is also putted between the fan's terminals, in order to protect it from being damaged by instant high voltages generated by the inductive property of the motor.

5.3.3 Encoders interface

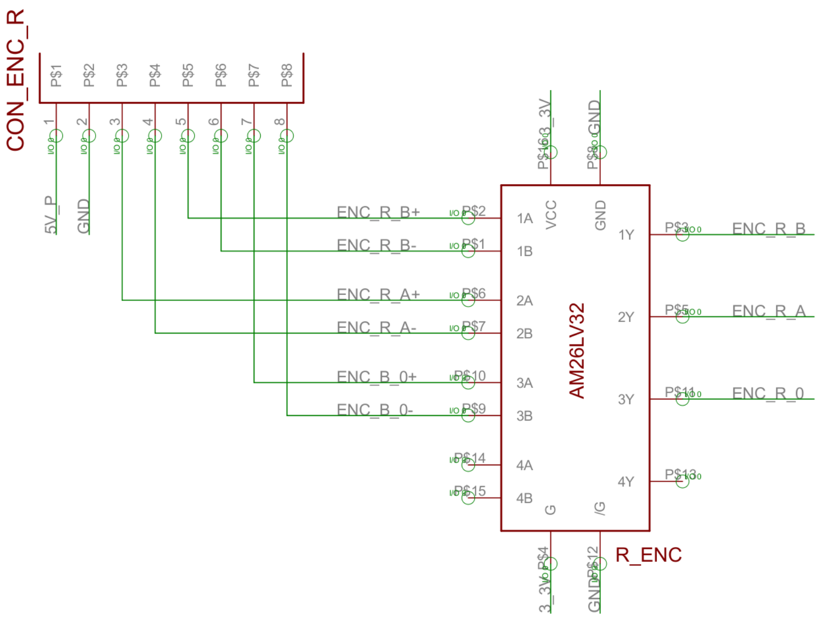


Figure 5.5: Encoders interface circuit

The encoders are used to measure accurately the distance traveled by the robot, it gives 1024 'ticks' for each wheel revolution. These ticks are transmitted via two channels A and B, which both are differential outputs of the encoder.

In order to treat these signals, AM26LV32 differential line receiver are used. This kind of chip can be supplied with only 3.3V however it accepts 5V inputs, which is the minimum supply level for the encoders.

As a result, it plays a double role as converting the differential input in a single output, and as converting a 5V input to a 3.3V output which is acceptable for the FPGA or the microcontroller.

5.3.4 Modules sockets

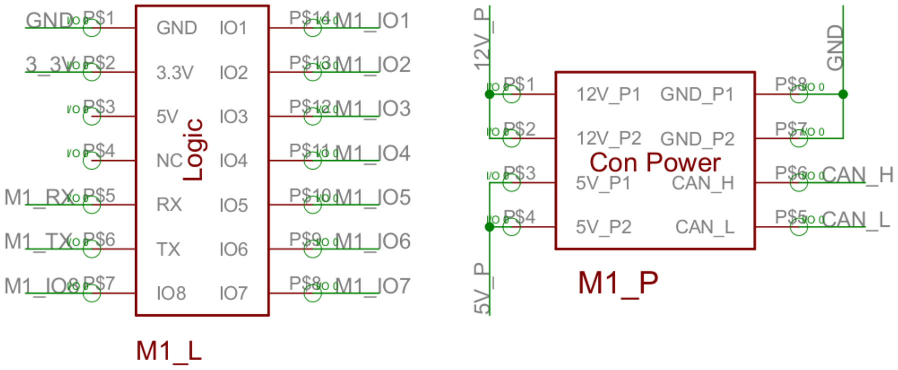


Figure 5.6: Module pin-out

The figure 5.6 shows the module socket pin-out. Each module can access to a 5V and 12V power supply generated from the power board, and a 3.3V logic supply generated by the daughter board. It also has the possibility to use the CAN bus or an UART from the PIC32MX.

Three of six modules have 8 GPIO each directly mapped to the Armadeus board's FPGA, and the 3 others have these available GPIO linked to the PIC32MX micro-controller's ports.

5.3.5 Program

At this stage of the project, there is no module which has been realized, so the only program running on the PIC32MX is blinking led code. This one has been developed in order to test if microcontroller was working, according to the configuration bits. This test has been successful, now need to be debugged the UART and the CAN bus.

For the PIC12F which control the fan, the program is very simple, there is a main loop in which the ADC is read, and this value determines the PWM duty cycle. The relation between the temperature and the duty cycle value is shown in figure 5.7 , where three phases can be seen:

- before 30 °C, the fan is stopped.
- between 30 °C and 60 °C the speed of the fan is linearly increasing according the temperature. An initial offset is required to start the fan's motor.
- after 60 °C, the fan remains at 100% speed

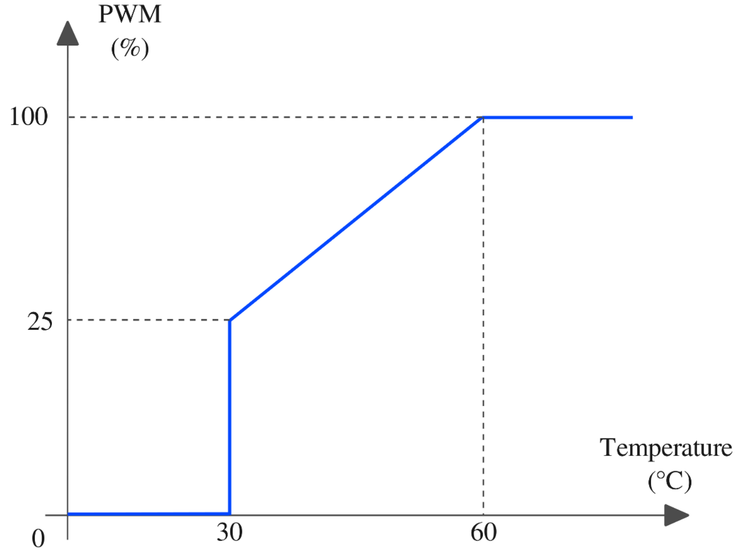


Figure 5.7: Relation between the temperature and the fan's PWM duty cycle

There is also a blinking LED, to be sure the microcontroller is correctly running.

6 Additional parts of the project

6.1 Vision system

As described in the contest presentation, the configuration of the table is not known in advance. It means that the black corn positions has to be determined in order to be able to avoid them.

A vision system has been thought to realize this task as shown in figure 6.1.

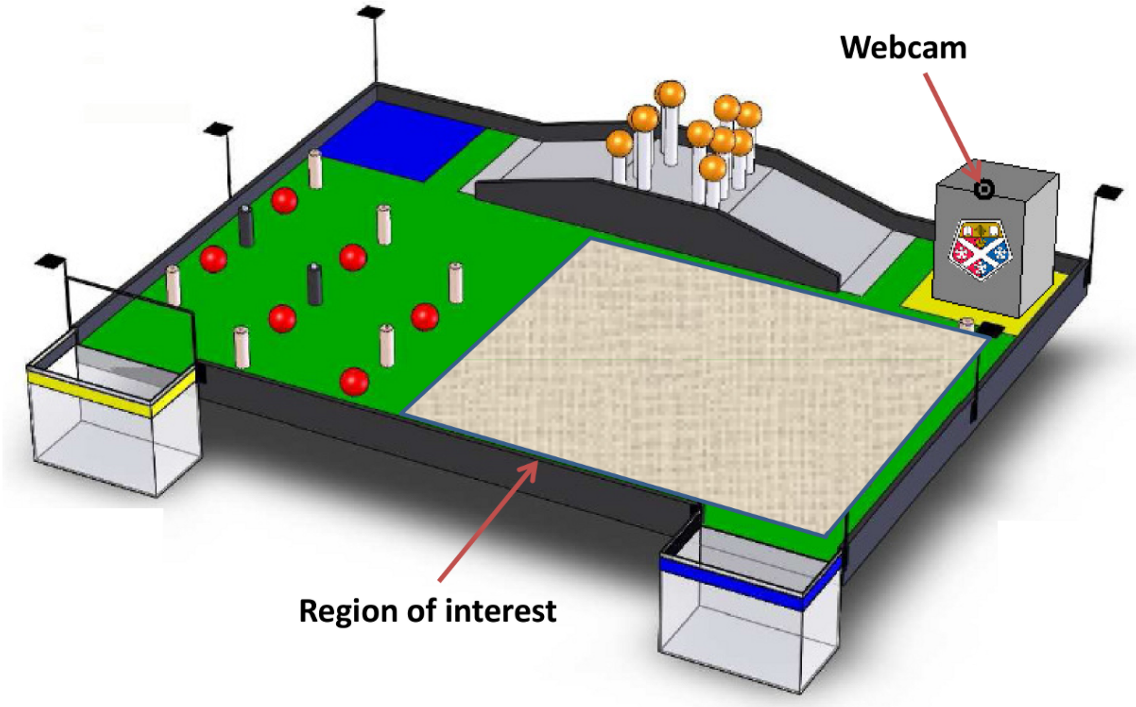


Figure 6.1: Vision system: initial configuration detection

The concept was to take a picture with a web cam at the beginning of the match, and then to correlate it with a set of picture stored on a SD card.

First, the picture has to be processed to remove the background and keep only the region of interest which is the game area. Then it is compared with all the stored images and the highest correlation result determines the game configuration.

An other point where the vision system could be useful is to detect objects during the match. As seen before, several balls are disposed on the table. It is obvious that after few seconds, they will no remain in their initial position because the other robot will probably move them.

A ball in 3D is seen as a circle in 2D from every point of view. This shape detection can be realized with an algorithm based on the Hough transform.

Because of the lack of time due to the complexity of the overall project, the vision system hasn't been implemented yet for the robot. However, the first step of the work has been done and was to start using a library capable of taking pictures from a web cam and process it. This library is OpenCV, which stands for Open source Computer Vision, can be used with C language and is compatible with Linux environment.

6.2 Avoidance

One of the major point to be allowed to participate to the contest is the robot must be able to avoid its opponent. Infrared SHARP sensors are used to insure this function. This kind of sensor can detect if a body is in the 15 cm range as shown on figure 6.2.

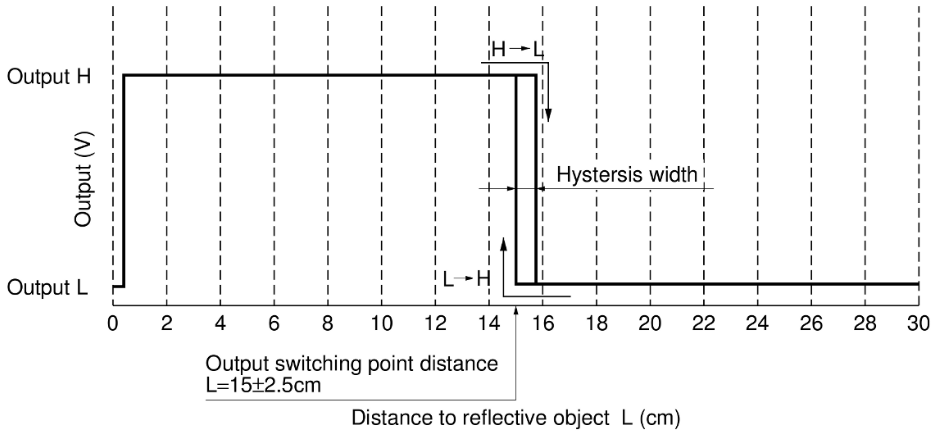


Figure 6.2: Sharp sensor distance characteristics

Eight of them are mounted all around the robot as shown on figure 6.3. Most of the time the robot will go forward, consequently it will meet the opponent from its front part. As a result, the sensors are concentrated in this area.

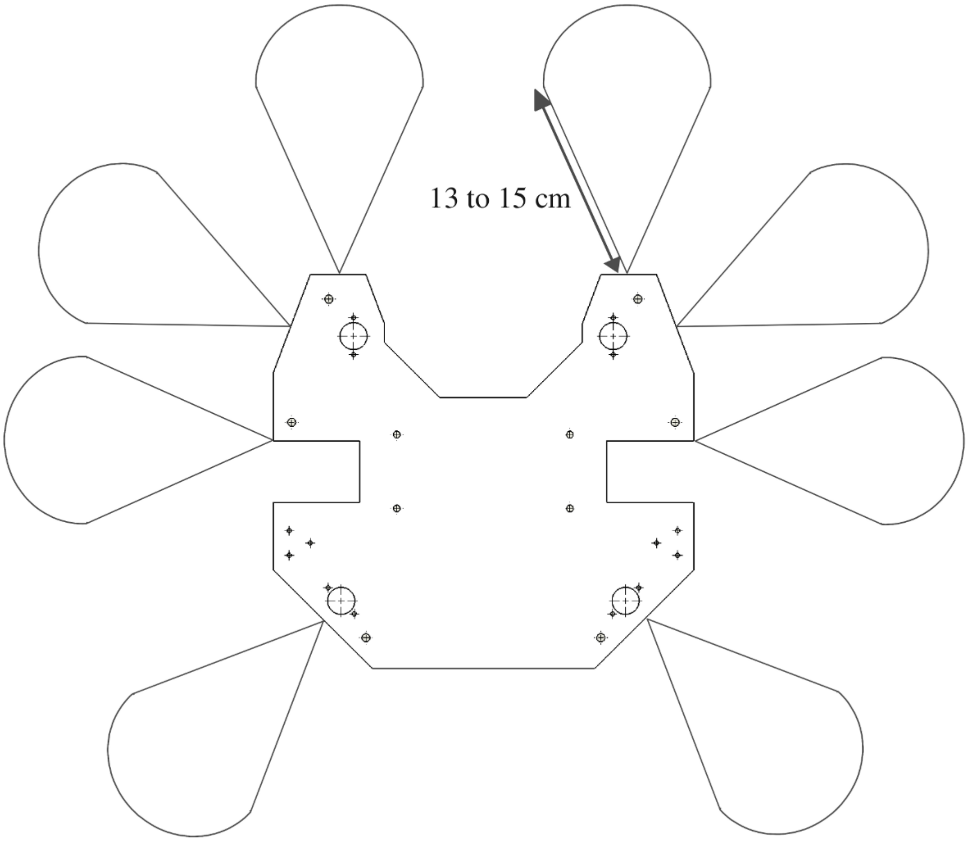


Figure 6.3: Distance sensor positions and detection areas

This sensors has already been tested using a ST Electronics development board, with a simple program sending the states of the sensors on a serial communication. During theses test, the real distance of detection has been measured and was more around 13 cm rather than the 15 cm announced in the datasheet. The good point is that they didn't interfere with each others, which could has been an issue if the robot's opponent has the same sensor (and there will be some).

6.3 Onboard GUI

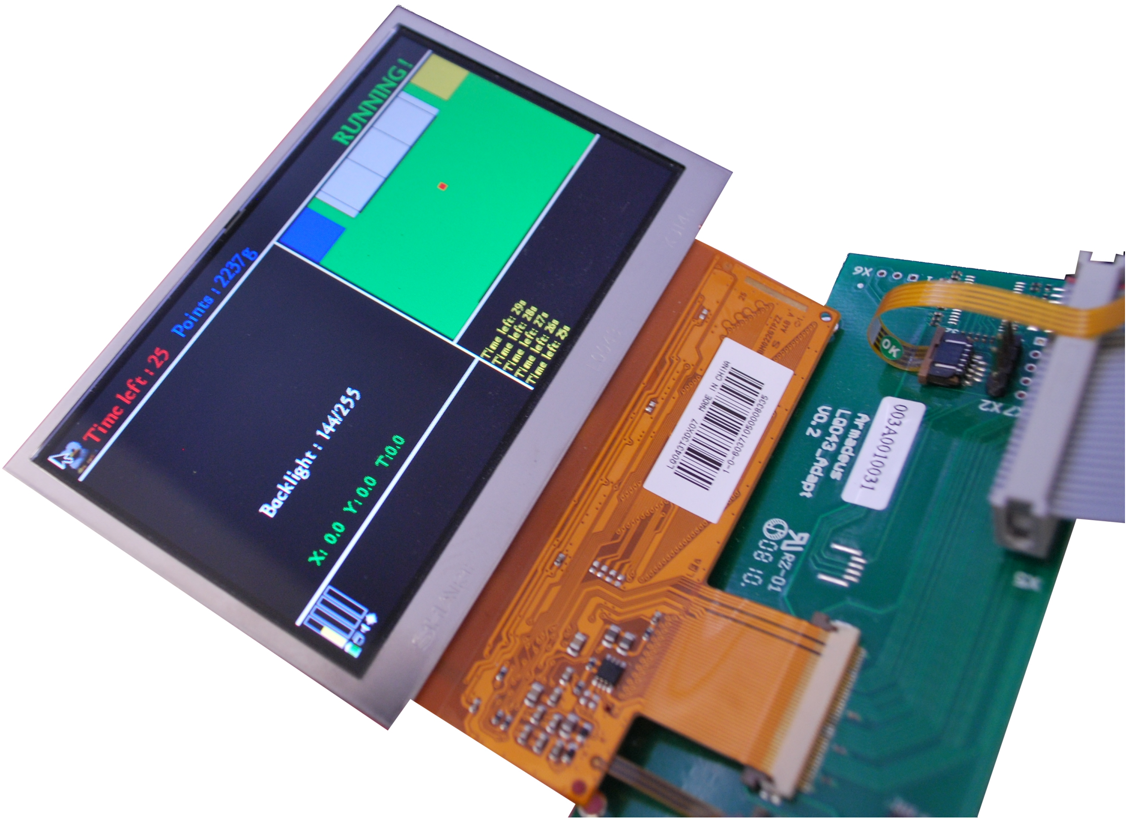


Figure 6.4: GUI on the Armadeus LCD

The Armadeus board can be interfaced with graphical LCD display and its associated touch screen. During the matches, the robot has to be completely autonomous, without a computer plugged to it.

A simple graphical user interface has been developed in order to be able to display any type of useful information on the robot such as:

- The battery level
- The time left during the matches
- The position of the robot on the game table
- The number of points the robot has made

On this screen, there is also a console to display miscellaneous messages.

The touch screen allows the user to be interactive with the robot, to browse menu and ask the robot execute an action. This can be useful just before the matches to check if everything works properly.

The LCD can directly be interfaced with the SDL [7] library. This library can be used in C language to develop multimedia application on various platforms.

[7] Simple DirectMedia Layer

7 Conclusion

The main of this project was to make a robot and work as part of a team. The main boards have successively been developed and tested whereas the vision system and the avoidance still require some work. This section discuss about the encountered problems, and what is planned to do as a future work.

7.1 Encountered problems

7.1.1 Mechanical design issues

The lack of knowledge in mechanics slowed a lot the project, specially for the object grabbing systems. Nevertheless the robot basement was easier to design because of the feedback from the previous participation. All the best teams adopted for the same principle with two propulsion wheels.

Every year the theme change and the difference between good and bad results reside in the capability of the teams to adapt their system.

However, the project is not finished, there is still three more week before the contest in which a major part of the remaining work will be to realize this advanced mechanical design.

7.1.2 MSP430 Programmer

At the beginning of the daughter board design, the first choice of the main micro-controller was a MSP430 from Texas Instrument. A programmer was required to work with it, and has been realized based on the MSP430 Jtag programmer sold by the Olimex company. This is an open source circuit which uses the parallel port of a computer to program and debug the hole family of these chips.

After the device was made, it has been tested but unfortunately didn't work. Some problems have been detected and solved, but despite all the efforts undertaken during two weeks, there were no way of making it working.

Consequently, the PIC32MX has been selected instead, staying in Microchip manufacturer choice which was safer because several projects were already working with the programmer which was used.

7.2 Future work

7.2.1 Before the contest

As said before, a lot of mechanical realization will have to be made, in order to grab corns and tomatoes during the contest.

There is also a small module to design, which will be used as a sensors board, mainly for the avoidance system.

If there will be enough time, the vision system will be designed and implemented on the armadeus board.

7.2.2 After the contest

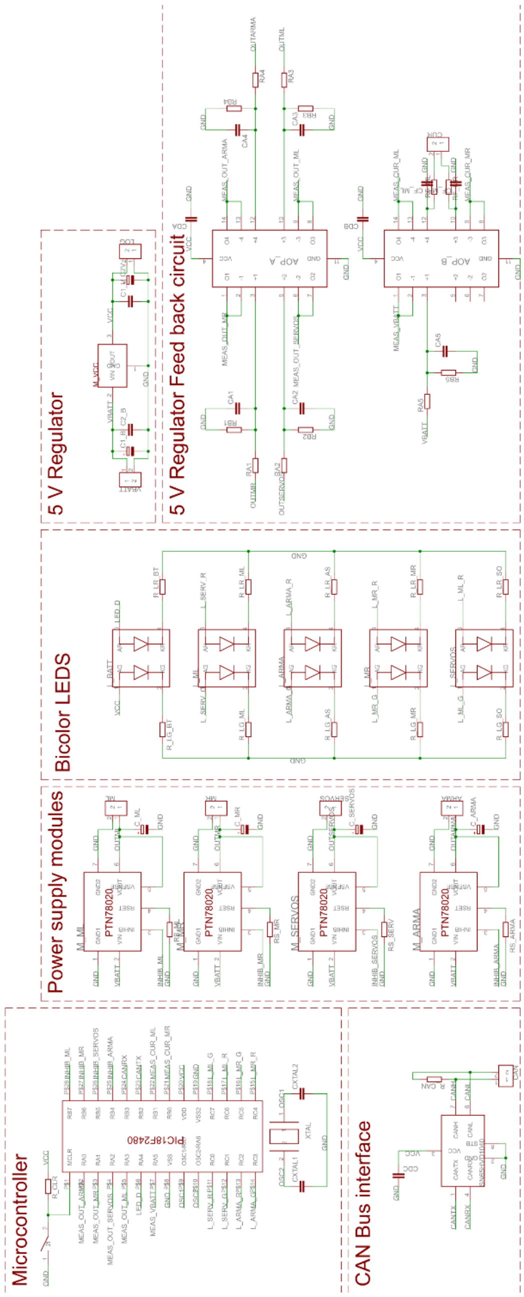
Because this project will also take part of the French Diploma Project Work class, some other development will be performed.

First, the motion control and the intelligence will have to be adapted and implemented on the daughter board, which belongs to Paul Monsinjon and will not be available the next years.

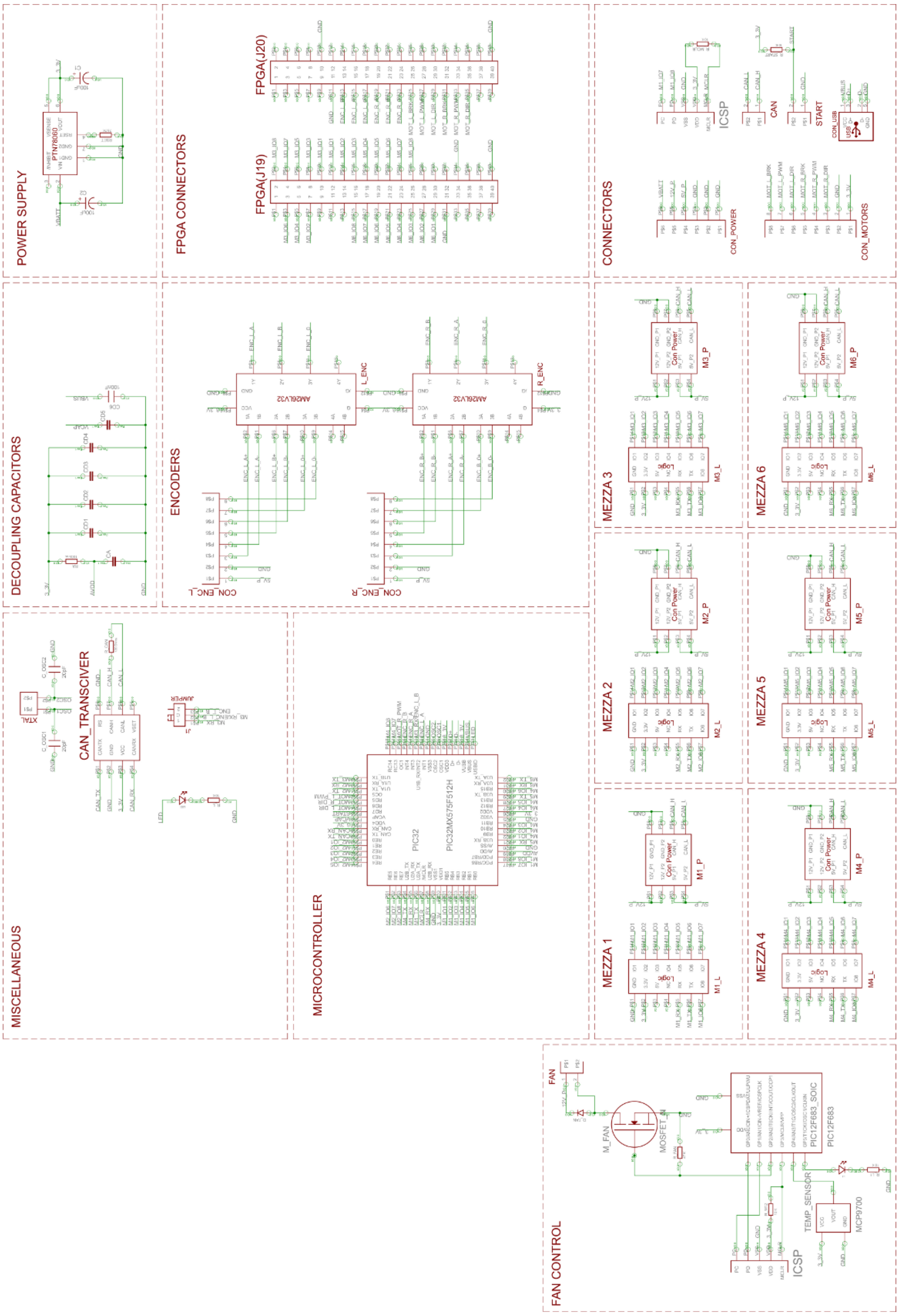
Then a computer GUI interface will be made to be able to communicate with the daughter board via USB. Therefore it will be possible to make configurable demo with the robot, giving it orders to execute depending on the user requirements.

Finally, a set of instruction will be written to make possible the project continuing for the next years.

Appendix A: Power board schematic



Appendix B: Daughter board schematic



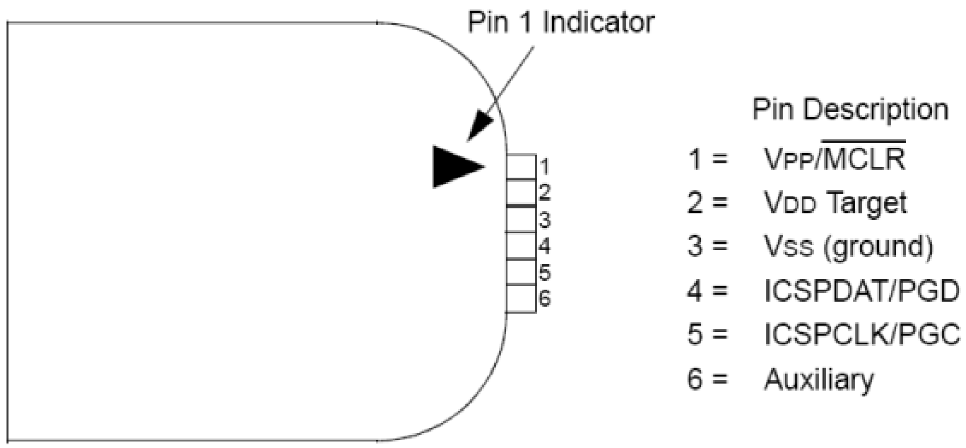
Appendix C: Microchip PIC programming

All the board made in this project are using PIC microcontroller. To program it, MplabIDE is required and can be download at http://www.microchip.com/mplab.

Compiler are also required to be installed, and can also be download from the Microchip website:

- Hi-Tech C compiler for the PIC12F
- C18 compiler for the PIC18F
- Mplab C32 for the PIC23MX

The programmer used for all the microcontroller was a PICkit 3, and the pin out is:



Note that except the pull up resistor between MCLR and VDD, nothing else is required unlike other programmers such as PICkit 2 or ICD2. The auxiliary pin is not used and can be skipped and the header of the programmer as a 0.100“ spacing.

To create a project, use the project wizard manager, select the device and select the appropriate compiler.

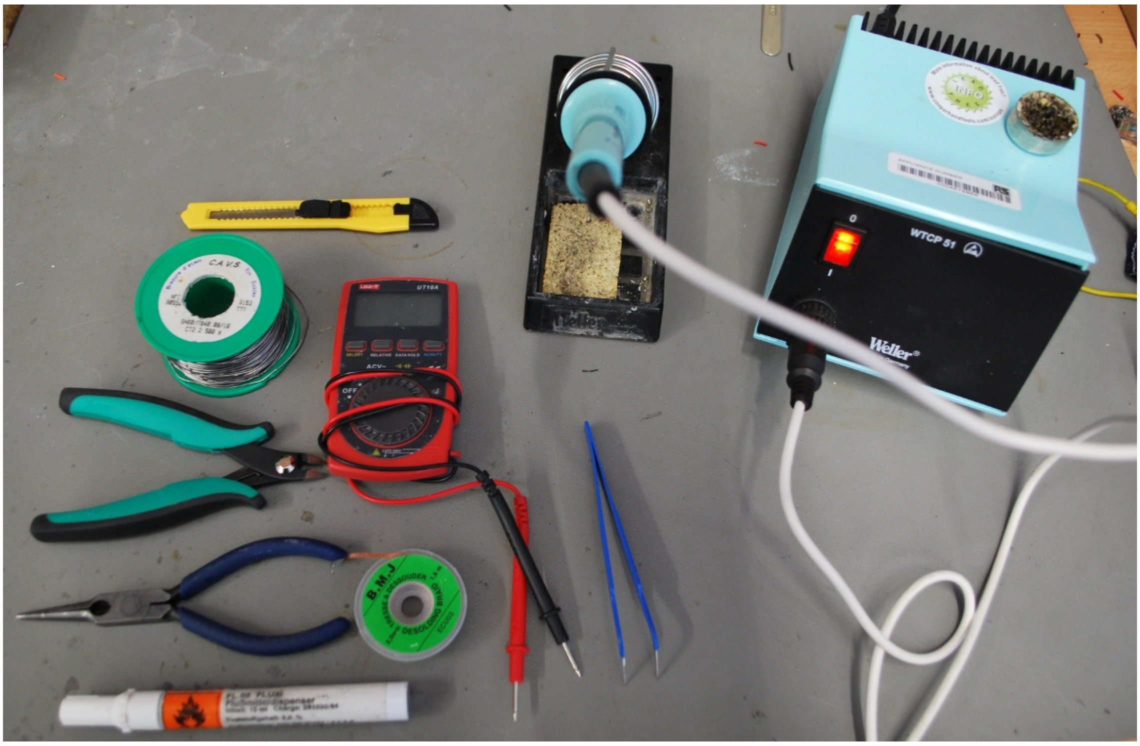
To program the device, connect the PICkit3 to the supplied board, and select in the menu programmer-¿pickit3; this will automatically download the firmware.

The configurations bits have to be set accordingly to the board configuration, specially for the oscillator. Refer to the microcontroller datasheet to see which value correspond to the configuration. There is two ways of setting these:

- Go into the the menu configure-¿configuration bits and tick the appropriate lines
- Include the code '# pragma config' followed by the word and its value in the project source.

Appendix D: Soldering guide for TQFP SMD packages

Several TQFP package has been used for this project, down to a 0.5 mm pitch. This is not impossible to do, there is several steps to follow.



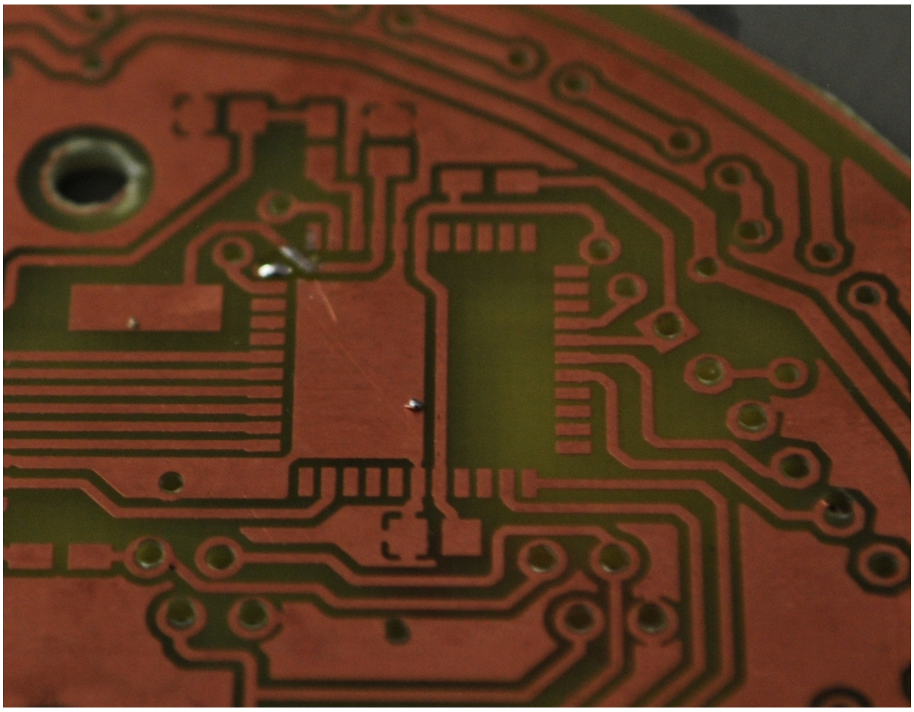
First of all, good tools are required:

- A good soldering station (Weller is a must) with a flat tip if possible
- Cutters, in order to cut wires, separate tracks or remove dust
- Solder wire, with silver if possible
- A multimeter to test track continuity and isolations
- Pliers
- Dessoldering braid
- Flux which is used to clean the tracks before soldering

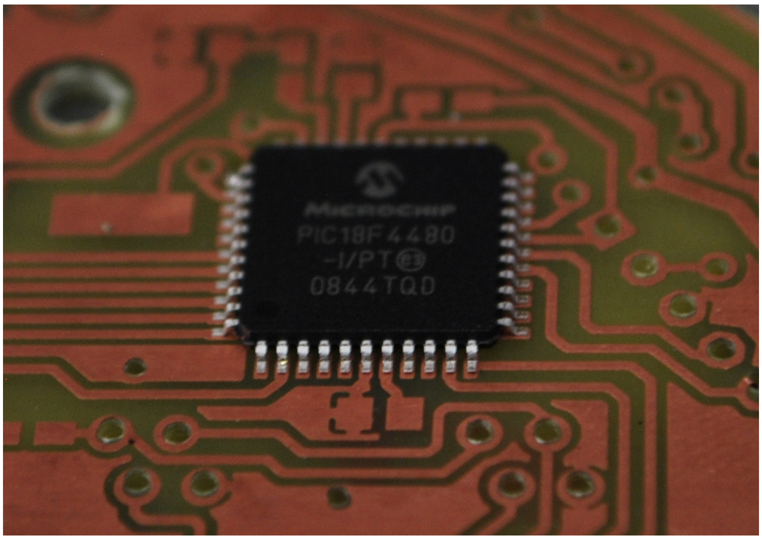
- A good pair of tweezers to handle the components

Make sure the tip is well cleaned, otherwise the solder wire will stay stuck on it:

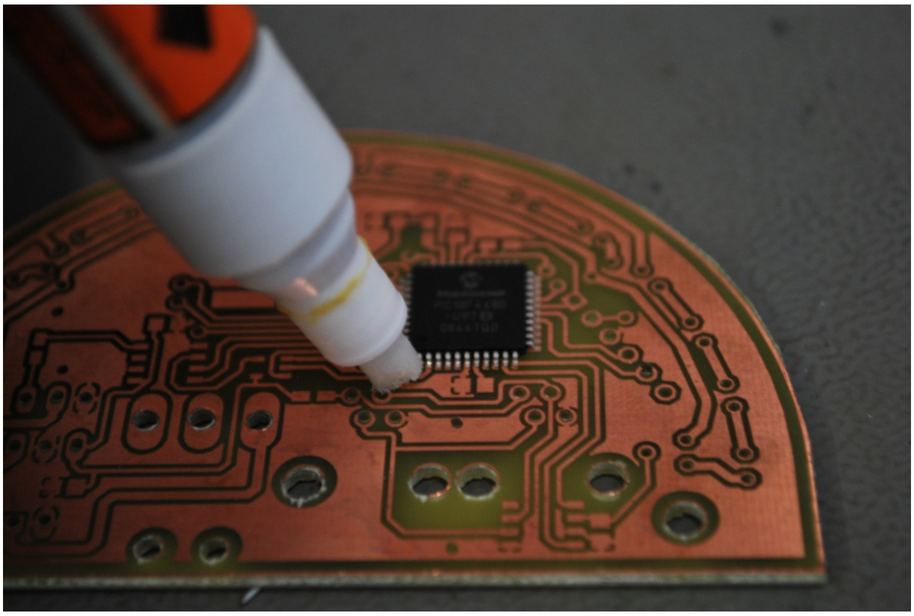
Put a little bit of solder wire in the top left corner:



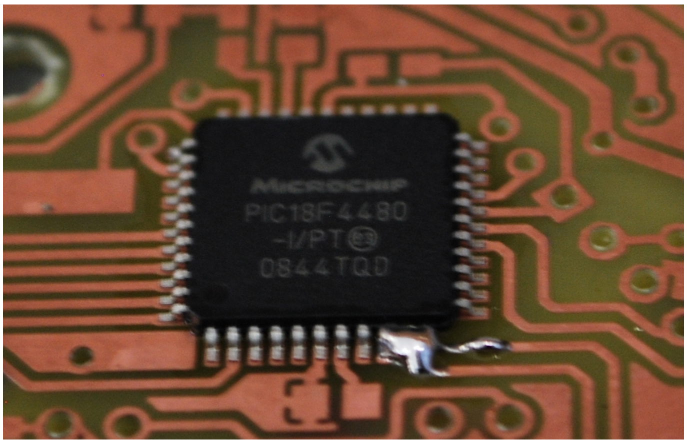
Solder the first pin of the component and make sure it is perfectly aligned on the four sides:



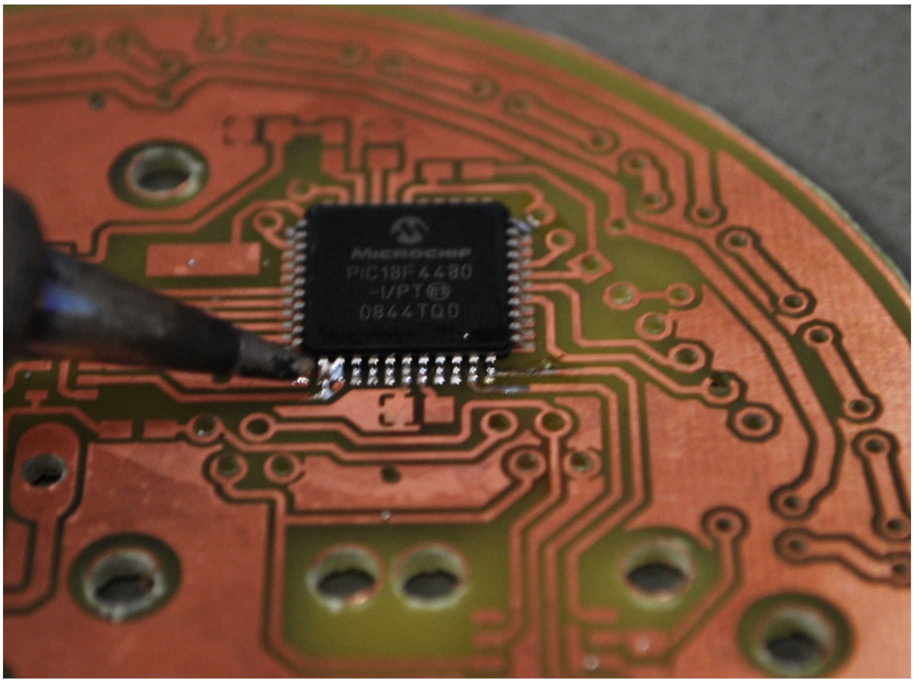
Apply flux on one side of the chip, just before the next step:



Apply a good amount of solder wire has shown:

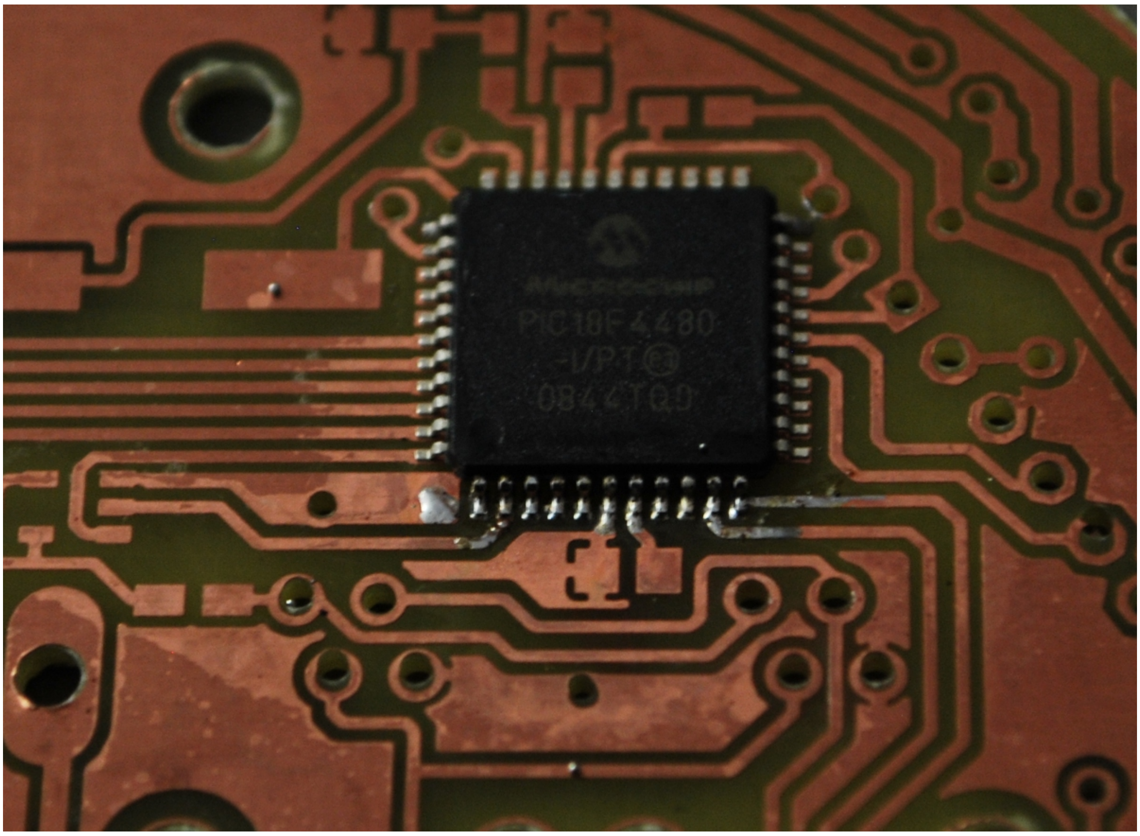


Make sure there is still enough flux and sweep the pins from right to left with the soldering iron. The pins should separate as shown on the picture, if not put some flux again:



Remove the eventual excess with the dessoldering braid and repeat the previous steps for the three other sides.

The result is not as good as if it was with a professional oven, but it works:



Appendix E: Robot maintenance

E.1 Robot parts

The encoder are fragile and must be used carefully, the O-ring joints have to be changed often because they wear down while the robot is moving.

While the robot is not used, it should be stocked in a dark area, or covered by something in order to not be exposed to the sun light. It has been seen that the wheels sticks to the table if it is not respected.

E.2 Batteries

The Li-Po batteries **must remain charged** when they are stocked. Indeed they can blow up if the voltage drops less than 3.7V per cell. To charge them, only use the appropriate Li-Po batteries charger which is made to control individually the cells voltage levels. Always use good connectors with polarizing slot, it is not permitted to make short circuits.