C O V E N T R Y
U N I V E R S I T Y

Faculty of Engineering and Computing

Department of Computing and the Digital Environment

Electronic Engineering

303CDE Individual Project

Swarm Robotics

Author: Andrei Radu Danescu

SID: 2773384

Supervisor: Panos Abatis

Submitted in partial fulfilment of the requirements for the Degree of Bachelor of Electronic Engineering

Academic Year: 2010/2011

Declaration of Originality

This project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet, etc) has been acknowledged within the main report to an entry in the References list.

I agree that an electronic copy of this report may be stored and used for the purposes of plagiarism prevention and detection.

I understand that cheating and plagiarism constitute a breach of University Regulations and will be dealt with accordingly.

Signed: Date:

Copyright

Office Stamp

Abstract

The current paper studies one area of robotics that has been of high interest to research communities around the world, Swarm Robotics, by implementing a modified swarm concept which displays broader communication capacity and the possibility of centralised control.

The main objective of this research is the design and implementation of a robotic system comprised of autonomous robots which communicate between them, forming the swarm. In order to demonstrate and study their behaviour, three complete units have been constructed, following design specifications defined at the start, in order to maintain consistency throughout the whole research.

First a thorough analysis is done to select the best suitable hardware components, followed by detailed, complete instructions for each module (hardware and software) in order to facilitate the reproduction of the project.

All the mechanical design and simulation are done using CATIA. The PSoC1, from Cypress Semiconductor is used as microcontroller, offering the advantage of very low external component count and reconfigurable hardware.

The programming language is C, using the free IDE available, PSoC Designer 5.1.

The final part of the report includes an objective view of the research by looking at the functionality and limitations of the finished product. It also includes ideas for future development as well as a student reflection chapter that gives a general idea of the skills required for such a project.

This paper demonstrates that it is possible to design and implement a system of individual robotic units that cooperate and can be programmed to do different tasks set by the user.

Table of Contents

TABLE OF CONTENTS

Abstract 4

Table of Contents 5

Additional Materials on the Accompanying CD 7

GLOSSARY 8

Acknowledgements 9

1. Introduction 10
2. Literature survey and related work 12
3. Methodology 15
4. Requirements and objectives 18
   - 4.1 Specifications 18
   - 4.2 Possible Implementation 18
   - 4.3 Decisions based on specific project objectives 19
5. Analysis 20
6. System design 22
   - 6.1 – Hardware 22
     - 6.1.1 Chassis 24
     - 6.1.2 Additional materials used 26
     - 6.1.3 Drive-train: motors and gearbox 27
     - 6.1.4 Connectors and cables 31
     - 6.1.5 Motor controller 32
     - 6.1.6 Microcontroller 36
     - 6.1.7 Obstacle avoidance sensors 38
     - 6.1.8 Colour detection 41
     - 6.1.9 Robot and food recognition sensor 44
     - 6.1.10 Food capture 46
     - 6.1.11 Food source 49
     - 6.1.12 Light detection sensor 50
     - 6.1.13 Battery selection and management 51
     - 6.1.14 Power Management 53
     - 6.1.15 Wireless unit 56
     - 6.1.16 Electronic Compass 59

6.2 – Software 60

6.2.1 Basic Behaviour 62

7. Project Management 63

8. Budget and Bill of Materials 64

9. Project Schedule 64

10. Risk Management 66

11. Quality Management 67

12. Conclusions 69

12.1 Critical appraisal and achievements 69

12.2 Future Work 71

12.3 Final discussion 73

13. Student Reflections 75

Bibliography and References 77

APPENDIX A – Project Specifications

APPENDIX B – Interim Progress Report

APPENDIX C -- Software

Additional Materials on the Accompanying CD

- Project Report
- Final logbook
- Preliminary documents
- PSoC software for all the modules developed
- CATIA models
- PCB schematics and artwork
- Datasheet of all the hardware elements

GLOSSARY

IR	Infra-Red
ADC	Analogue to Digital Converter
LED	Light Emitting Diode
LDR	Light dependent resistor
DC	Direct Current
FET	Field Effect Transistor
FM	Frequency Modulation
DSSS	Direct Sequence Spread Spectrum
IC	Integrated Circuit
PWM	Pulse Width Modulation
ISR	Interrupt Service Routine
I/O	Input / Output
WBS	Work breakdown structure
RC	Radio Controlled
PSoC	Programmable System on Chip
SoC	System on Chip
LiPo	Lithium Polymer

Acknowledgements

I would like to thank Mr. Panos Abatis, the project supervisor for his incredible support and the professionalism and promptness with which he patiently guided me whenever I asked for his advice.

A special thank you goes to my family, who supported me throughout my studies, and to my girlfriend for her endless understanding while working on this project.

I would also like to thank Cypress Semiconductor and National Semiconductor for the samples and technical support provided, without which this research would have taken a lot longer to complete and also a higher budget.

1. Introduction

The idea of swarm robotics was derived from the social behaviour of insects and therefore, some of the main aspects of this concept are decentralized control, limited communication abilities amongst units, usage of local information and robustness of the system. It is a relatively new field of study and it is viewed by many as the future of technological development, when large swarms could exhibit social behaviour by working together in perfect harmony.

This research project aims to develop a group of self-coordinated autonomous robots that can work and communicate towards accomplishing a determined final task. These units can be used, among others, in search and rescue operations or for inspection and surveillance where human involvement is not possible (for example in hazardous environments).

The final product consists of three individual, autonomous robotic entities that can navigate a predefined environment, send data from sensors and communicate between them with the possibility of adding as many units as desired. Some of the benefits of such a system include great flexibility and adaptability to most environments, rapid development and robustness to failures.

The project is split into smaller modules, viewed as ”black boxes” that must link together to form the final system, providing: wireless communication, obstacle avoidance, compass orientation, light detection and avoidance, colour detection etc. In addition to the design and construction, software development regards implementation of the behaviour, research into communication and control algorithms. The behaviour studied in this project is not pure swarm behaviour, but a modified version that was adopted to fit the purpose of the research, with the main differences being the communication abilities and the available number of units for demonstration.

The main features of any swarm robotic system are considered to be decentralization, locality, stigmergy, flexibility, robustness and emergence. Some of these features are not directly recognisable in the current research, but were adapted to fit this project as well as possible. The first one, decentralization refers to using a decentralized controller, meaning that each unit is responsible for its own actions, leading to a distributed decision making process amongst all the units of the system. This feature is fully implemented in this research and can be directly observed. Locality refers to a swarm that involves only entities with limited and local sensing and communication ability.

The term “stigmergy” defines a mechanism of coordination between robots and their actions. Basically the interaction between the robot and the environment alters the second one and these impacts on the next action or provides information for the other units of the swarm. This mechanism would lead to spontaneous emergence of behavioural patterns. Both of the two, locality and stigmergy, are implemented in a modified way in the system under discussion: locality is not fully respected because of the global communication available, although it can be restricted trough software to local interaction only, and stigmergy is only viewed from the sensorial point of view, because the robots cannot alter the environment they are studied in. This feature can be implemented easily but was not of interest for the current research.

Flexibility and robustness are two important attributes of any robotic system. Flexibility refers to the system’s ability to adapt to changing environments and robustness regards the system’s ability to continue to function if parts of it stop functioning. Both of these attributes apply well to the system under discussion. The last feature from the list, emergence, regards the global behaviour within the swarm, without any software specific elements. This propriety is not displayed by all the swarm systems and it generally applies to larger swarms where it is easier to observe and quantify. Emergence is strongly related to stigmergy and by implementing properly the stigmergy within a swarm system, certain behaviour patterns will eventually emerge and govern the system. (Vito Trianni, 2008)

2. Literature survey and related work

The I-Swarm (October 2008) project was performed by Universitat Karlsruhe in Germany and combines micro-robotics, distributed and adaptive systems as well as self-organising biological swarm systems. It is a great technological development which facilitates the mass-production of micro-robots, which can then be expected to accomplish a variety of tasks and to perform different strategies. With the realisation of three bio-inspired basic scenarios, the swarm was being able to perform dispersion, aggregation and collective perception. (PHYSORG, 2008)

Southampton University's project **Formica** (2008) is another research which demonstrated various algorithms for distributed problem solving, particularly in simulations. The focus is on complex, emergent behaviour arising from the local interactions of individuals following simple rules. The cost of current robotics platforms prohibits experimentation with swarms numbering more than a few tens of units. As a result, the practicalities of software and hardware maintenance in large swarms are yet to be addressed. (Jeff Gough, 2008)

The **Swarm-bots** (2001-2005) and **Swarmoid** project (2006 – 2010), are two multinational projects funded by the European Union which developed three types of robots: foot-bots – designed for on ground movements, hand-bots – designed for climbing surface and eye-bots – designed for aerial surveillance. The projects are well documented and some of the research papers are available online on the project websites. Relevant to the current research are “New task allocation methods for robotic swarms” by F. Ducatelle et all, and “Interference reduction trough task partitioning in a robotic swarm by Giovanni Pini et all, both papers addressing the problem of task allocation and how the robots choose a certain task in an efficient and distributed way as it would be inefficient to have all the swarm serving the same task at once and leaving all the other announced tasks pending.

The most relevant projects covering a similar subject are the ones mentioned above, adding the MIT "**Ants**" to the list, although this is not so well documented but offers some good videos and pictures as well as hardware details. (MIT, 1994)

Apart from online articles and projects, a good source of information was found trough Servo Magazine and Robot Magazine. Also, although there are only a few available books on this subject, some are worth mentioning for the ideas that can be used in implementing swarm behaviour and social structure:

"Multiple heterogeneous unmanned aerial vehicles'' by A. Ollero offers a very good insight on multi-UAV cooperation, as well as system structure details and case studies relevant to a swarm of ground robots. There is a thorough emphasis on the organisation of hardware and software to enable the units to work together.

Ulrich Nehmzow's *"Mobile Robotics: A practical Introduction''* is also good and the chapters on navigation and modelling robot-environment interaction offer valuable information on the topics. Although modelling robot and environment is not very realistic as it ignores many real life factors, it's a good way to see if a unit behaves as intended under certain conditions.

''*Junkbots, Bugbots & Bots on Wheels*'' by Dave Hrynkiw and Mark W. Tilden is very good for learning how to make unconventional robots and reduce cost and BOM length. The book provides relevant material on autonomous robots and self-sustaining units. It gives very good information on solar robots and how to create robust designs.

"*Ant colony optimization and swarm intelligence*" by Dorigo Marco offers a collection of papers on swarm and ant colony behaviour. The book follows the 6th International Conference ANTS 2008. It is a good starting point for developing and implementing the behaviour of the robotic entities.

"*Springer handbook of robotics*" by Bruno Siciliano covers virtually all major aspects of robot design and it was a very good book to consult in deciding what hardware components are the most appropriate for the current design. The book also has a very good chapter on multi-sensor fusion and robot behaviour modelling.

"*Evolutionary Swarm robotics: evolving self-organising behaviours in groups of autonomous robots*" by Vito Trianni is a conclusion of the Swarmoid project. It is compiled as a collection of research theses, all based on the same model offered by the Swarmoid. It covers all important areas of swarm research up to date, being the most relevant book for this research program and which can also serve for future developments of the system currently under discussion.

3. Methodology

The research method used combines two major models: expanding and refining an earlier model as well as exploratory research.

The project was initially based on the swarm behaviour model and on how this could be implemented for the current research. Previous swarm models and the requirements for such an implementation showed that a large number of hardware units must be produced, but due to limitations of time, resources and workforce that would have not possible. Therefore, a decision was made to develop a modified swarm behaviour that offered the possibility of expansion towards a more accurate implementation of the true model by creating three robotic entities which would demonstrate the concept.

From this point the project moved into the second stage, the exploratory research, because a suitable previous model was not available to develop upon. At this stage an extensive search was conducted to find the best suited hardware to work with and because the specifications were quite open, some constraints had to be placed. From a financial perspective the parts had to offer the best price/quality ratio and fit within the budget. Another important aspect was that the project is an individual research so a large workforce was not available, therefore the functionality, from a behavioural perspective, had to be restricted to something deliverable.

Finally, because the time available was limited, a prototyping and incremental approach was used which meant that simulation time was sacrificed in favour of development and testing. CAD simulation and breadboarding techniques were lightly used; PCB and final schematics were developed from an early stage with further refinements and periodic adjustments to fit better in the final product.

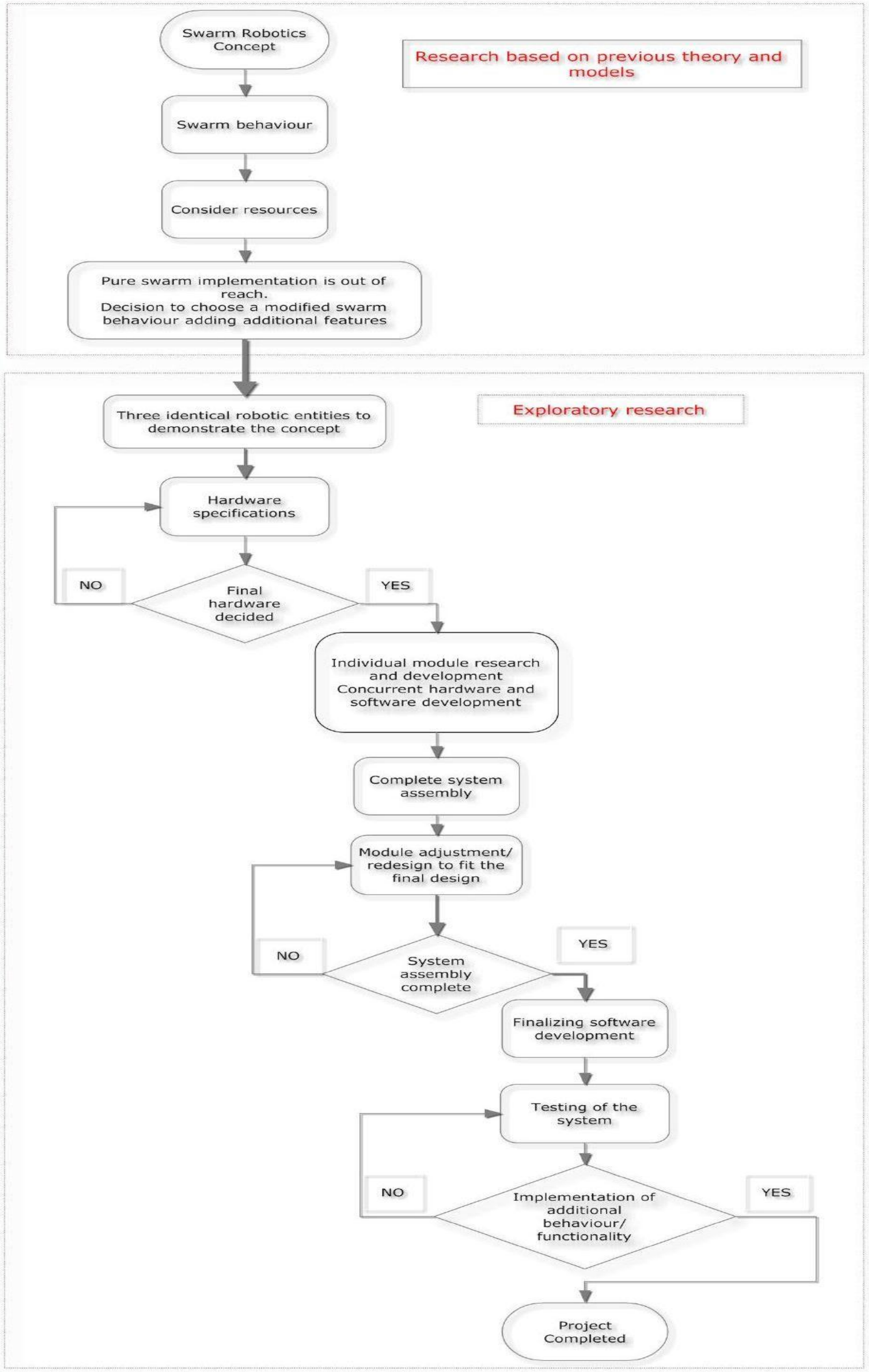
Adopting the “black box” design philosophy meant that all the modules could be viewed, at a system level, as independent entities and they can be replaced with equivalent modules that offer better performance with minimum adjustments. This design method is derived from the “Divide et Impera”, divide and conquer algorithm of problem solving, which involves splitting a complicated problem into small, easy to solve problems and putting them together at the end to solve the original problem.

A concurrent approach to research and development was used throughout the whole project, as software and hardware have been developed alongside, because certain modules could not be finalized without the accompanying software support. This approach led to a very modular design with individual units developed on the "black box" philosophy, offering very easy maintenance and upgradeability. Usual development procedure for the modules implied a design phase, where the schematic of the circuit was created and analysed, a quick breadboarding test and then a PCB design for extensive testing. The PCBs were created by taking the complete system into account so there was little need for redesign or adjustments in the final phases of the project.

No special equipment is needed, except for fabricating the chassis and cutting or drilling different mechanical parts. Because all the manufacturing of parts was done by the author a complete list of the equipment used is given in the table:

Electronic equipment	Mechanical tools	Software
DC power source – used before the power cards were developed.	High speed rotational router for drilling, cutting and sanding.	National Instruments Multisim 11
Measuring equipment: multi-meters, oscilloscope and logic probes.	Precision measurement equipment	Cypress Semiconductor PsoC Designer 5.1
Soldering and de-soldering equipment.	General tools like screwdrivers, different files, etc.	
Optical thermometer for accurate temperature measurement.	Small grip vice	Dassault CATIA
National Instruments DAQ: USB data-logger with the LabView Software.	Metal cutting scissor	Labcenter Proteus

Adopted research path is shown in the figure.



4. Requirements and objectives

All the project requirements were drawn after a thorough analysis of the swarm concept and discussion with the supervisor.

4.1 - Specifications:

Software

- issue and execute commands
- make decisions based on sensor reading
- control motor speed and monitor battery life
- communicate sensorial information to other units

Hardware

- detect obstacles
- communicate with other units
- detect radiation (visible or IR)
- robot recognition
- monitor battery life

4.2 - Possible Implementation:

Software: the only limitations are the processing power requirements and available time. Processing power is not a major concern because the idea behind swarm behaviour is that the units forming the swarm do not possess advanced decision making algorithms or advanced neuronal structure.

Hardware: all the functions must be implemented within both budget and available resources.

Considered hardware options for each function:

- Obstacle detection/ avoidance: mechanical sensors, optical – IR sensors, ultrasonic unit, laser sensors.
- Wireless communication: optical – IR, radio – both FM and DSSS.
- Follow the leader/ robot recognition: RFID, close range IR optical detection, specific sensor development.

- Monitor battery life: individual cell LiPo battery monitor is implemented as a software function but requires an ADC hardware convertor which is available on the microcontroller.
- Chassis: plastic, metal – iron or aluminium, composites – carbon fibre and similar.
- Drive-train: DC motors, stepper motors or modified RC servo units.
- Motor driver: MOSFET H-bridge, IC motor driver, RC auto-model motor driver.

4.3 - Decisions based on specific project objectives:

Following a detailed research on each hardware option and considering the optimal price/functionality ratio the following solutions were chosen:

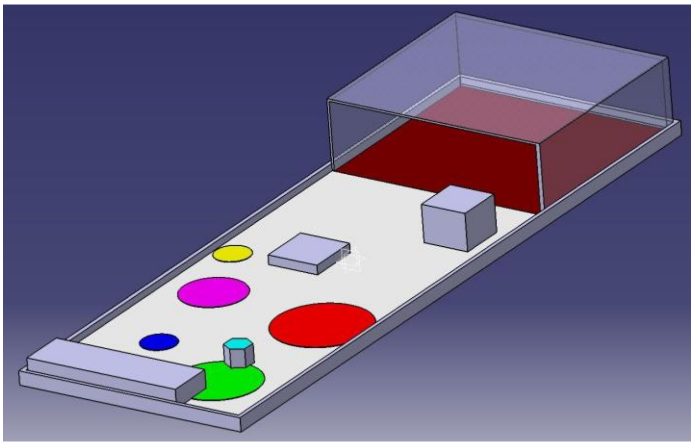
- Obstacle detection: IR distance sensors combined with mechanical switch-type sensors. The mechanical sensors are used as backup/ fail-safe for the optical units.
- Wireless communication: transparent data transmission module on 433MHz frequency.
- Robot recognition: optical sensor for colour detection.
- Battery monitor: software controlled on-chip ADC.
- Chassis: FR4 - glass reinforced epoxy laminate sheet, same material that is used for PCB fabrication with the thickest available option is used.
- Drive-train: 3 V DC motors running individual gearbox. A complete assembly of two motors and two gearboxes in one unit is used.
- Motor driver: 1 A integrated circuit driver with integrated output diodes and over-temperature control.

Overall system complexity had to be limited in order to be completed in the time available:

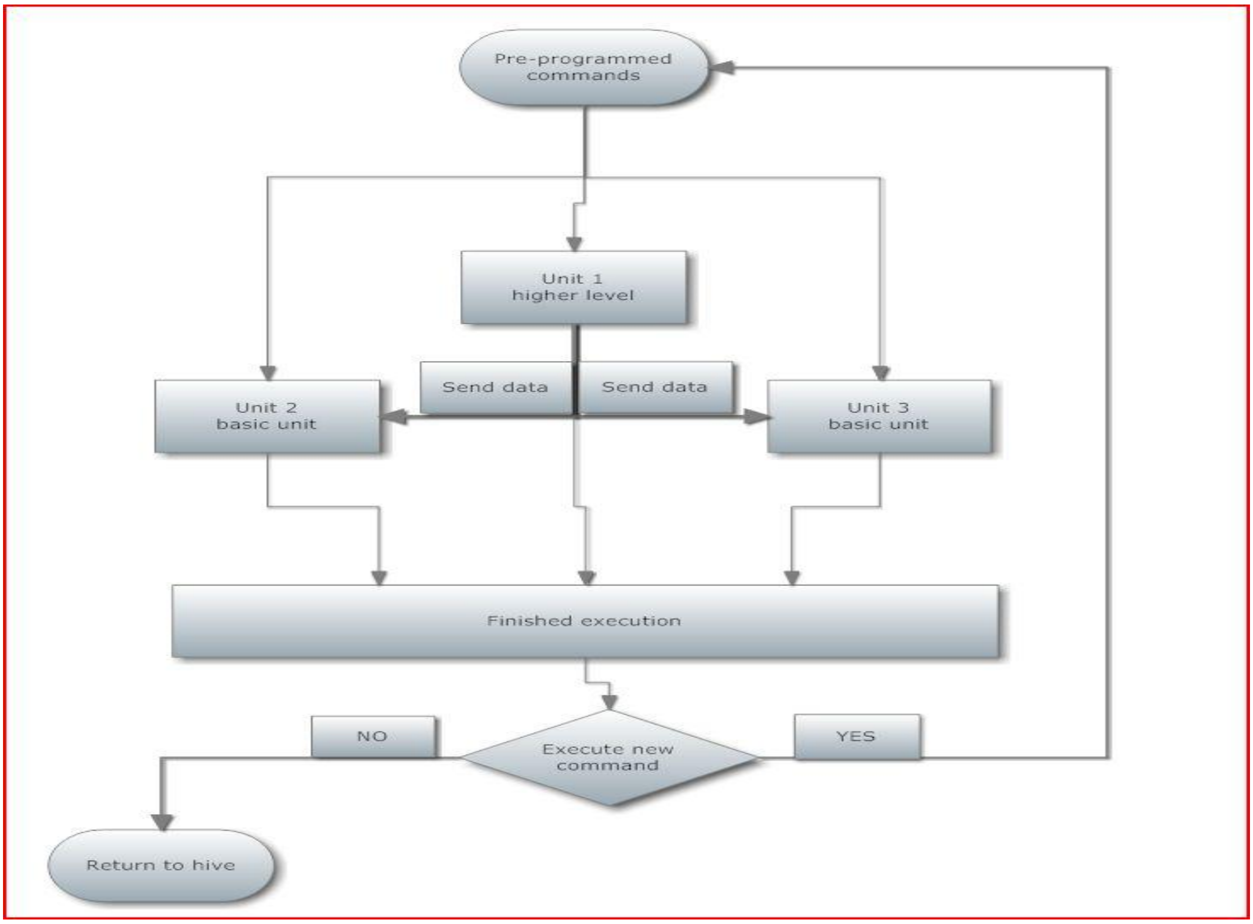
- Software complexity reduced due to limited time available.
- Sensor complexity reduced mainly due to financial considerations.
- Motor selection and power output, as well as chosen platform selection are limited due to financial considerations.
- Hardware material availability and cost are a key factor in deciding the final hardware.

5. Analysis

In order to analyse and demonstrate the concepts researched throughout this project a scenario is setup. A limited environment will be created, consisting of a 150x50 cm arena with different excitation elements.

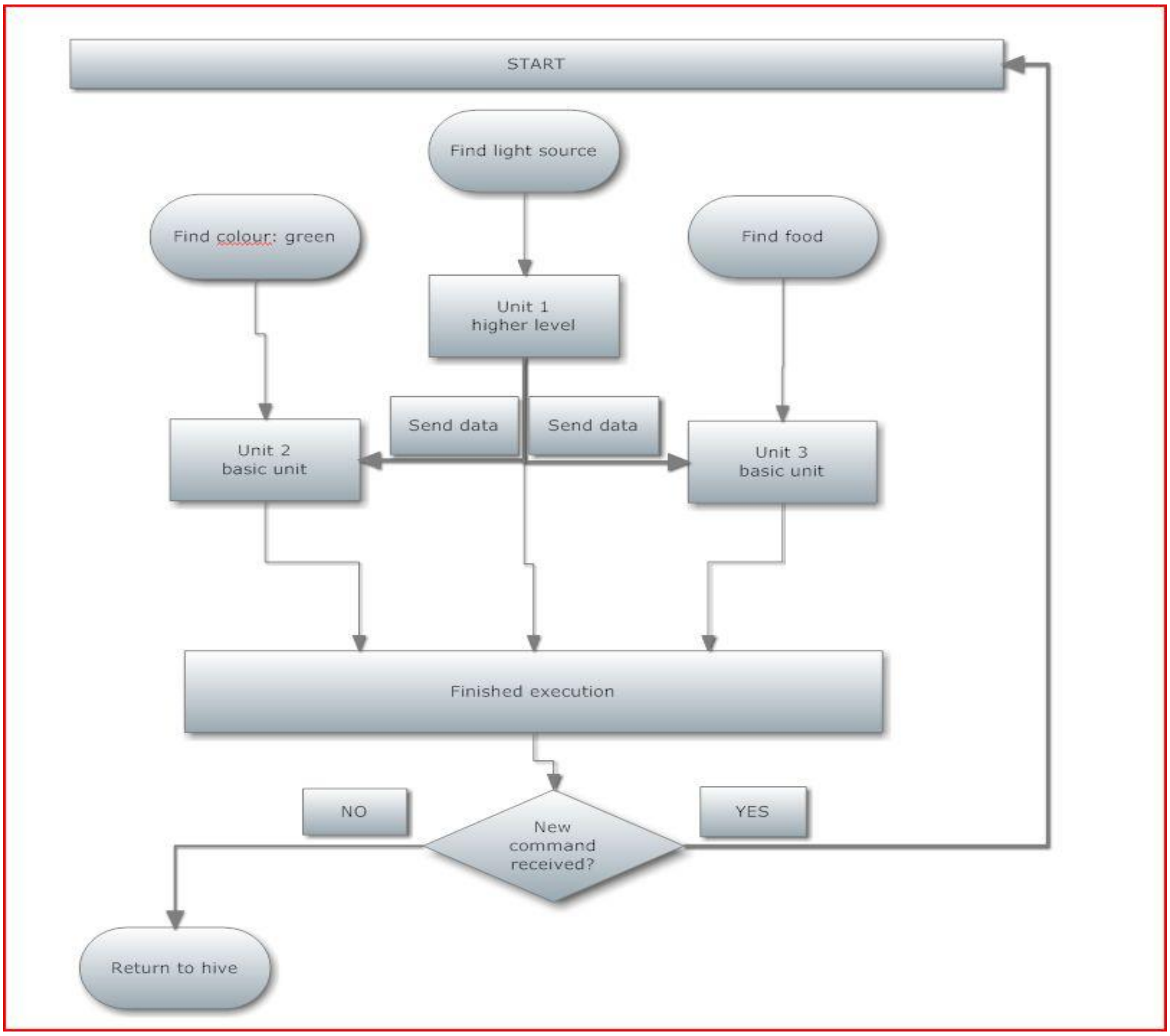


These elements consist of different coloured terrain, available “food” (special coloured LED banks that the robot must find and transport), obstacles (different types that must be avoided) and energy sources. An enclosed part of the environment will function as a “hive”, a place where the robots must return for inspection, battery recharging and others.



The robotic entities must autonomously navigate the environment, collect different sensor data and execute predefined tasks. These tasks can represent “food” hunt and transport, colour search or simply autonomous exploration/ search for energy sources. The tasks are split amongst the units but all are aware of the general mission briefing and if one unit happens to complete a task that was not previously allocated to it, it will send the sensor reading to the other units and pass its task along upon completion of the current one. Following this algorithm, dynamic task allocation is implemented and better efficiency can be achieved, opposed to a single unit working to complete a certain task without passing it along.

A robot recognition system is also implemented and the units are able to recognise and follow each other if it is necessary. This way certain sensors can be implemented only on some units, reducing cost both financially and from the software implementation perspective.



6. **System design**

6.1 – Hardware:

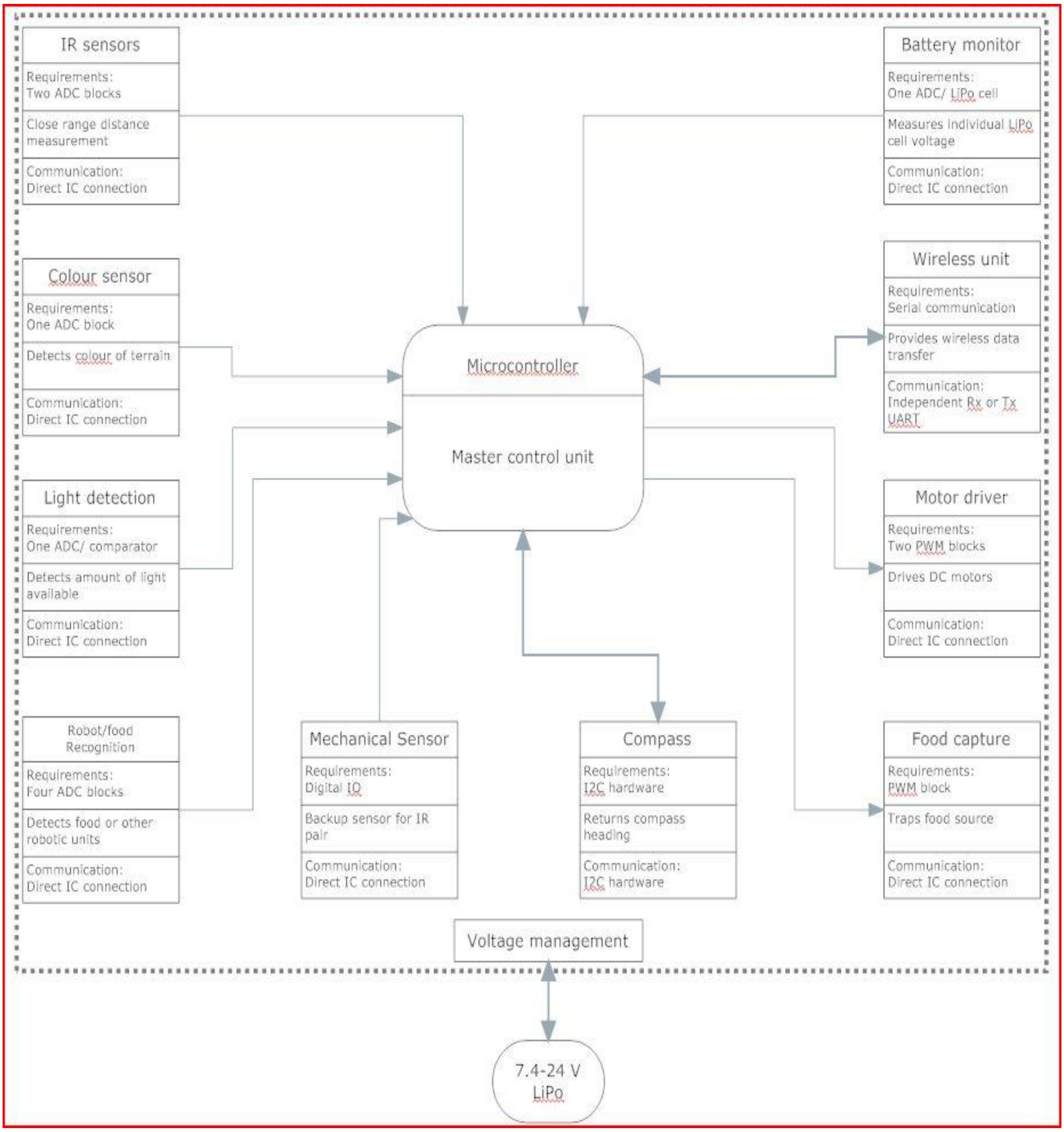
The robotic units of the current project are completely original and specifically designed for the intended research. All the individual modules, apart from the wireless unit, were developed for this project and nothing was bought off the shelf. This approach, while requiring more knowledge and time resources, offered a better solution from a financial perspective and a space saving advantage. Designing the PCBs specifically for the robotic units offered the best performance while maintaining the smallest dimensions possible, all the boards being double sided in order to make the most efficient use of the available copper area.

Both hardware and software developed for this project (except etching the PCBs) were entirely manufactured by the author without using expensive equipments such as CNC machines, laser cutting or specialised welding. This is an important point because it shows that the project can be easily reproduced and developed further.

A major difference from the hardware perspective was offered by the PSoC, with its reconfigurable hardware blocks. The advantages of the PSoC will be described on each module individually, but it is worth mentioning that using a Programmable System on Chip – PSoC - allows for much easier hardware experimentation, fine tuning and analogue to digital convertor options and the chosen unit can implement three typologies of ADC convertors with user selectable number of bits.

Considering the requirements the system has for sensorial readings, it is clear that the accuracy provided by a large number of bits is not needed. Therefore, in order to minimise the number of external components and make use of the advantages offered by the PSoC used, 12bit ADCs are used. The sampled result, with a maximum value of 4096, was divided by 10 in the software, resulting in 409 values for the 0-5 V span. These values can be either further divided if necessary, like in the case of the obstacle avoidance sensors, or enlarged, by dividing the initial 4096 values by a factor of 5 or less, like in the case of the robot recognition senor.

The overall system schematic is shown in the picture. Detailed module information is available in the dedicated section.

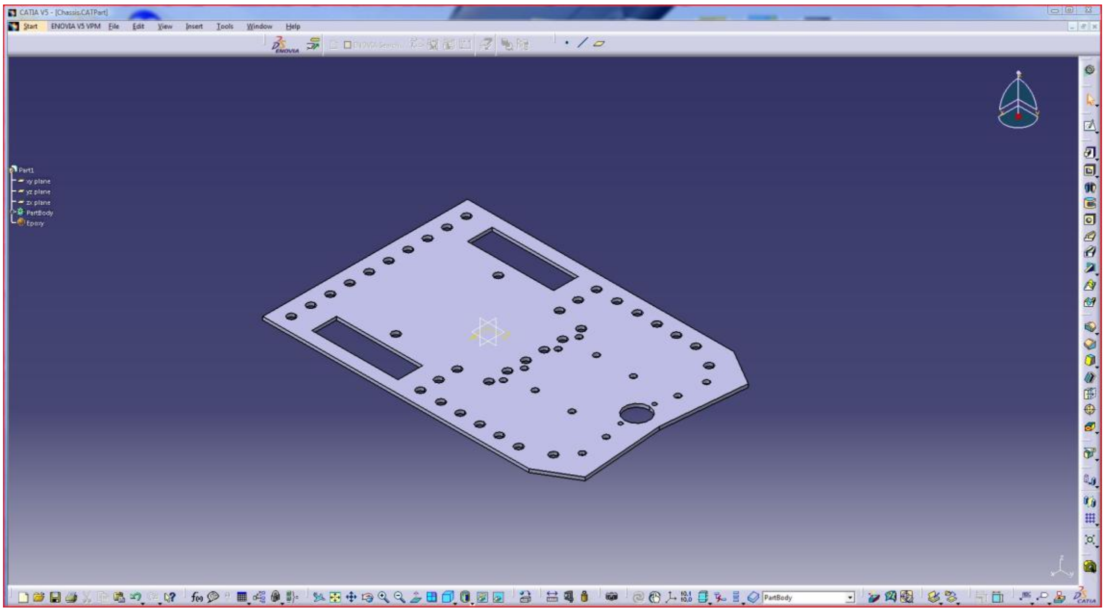


6.1.1 Chassis

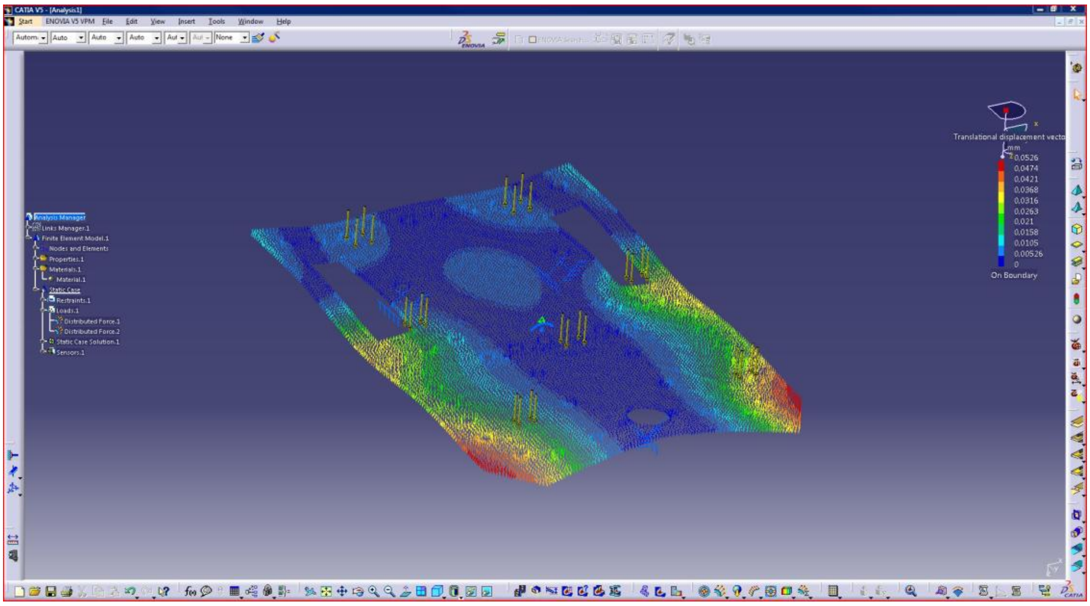
One of the most important elements of the robotic unit is the chassis. This is the basic layer of each unit and a poor design can ruin the whole concept. In the design of the common chassis for the three units, the most important aspects considered were: stiffness, availability and price, upgradeability and weight.

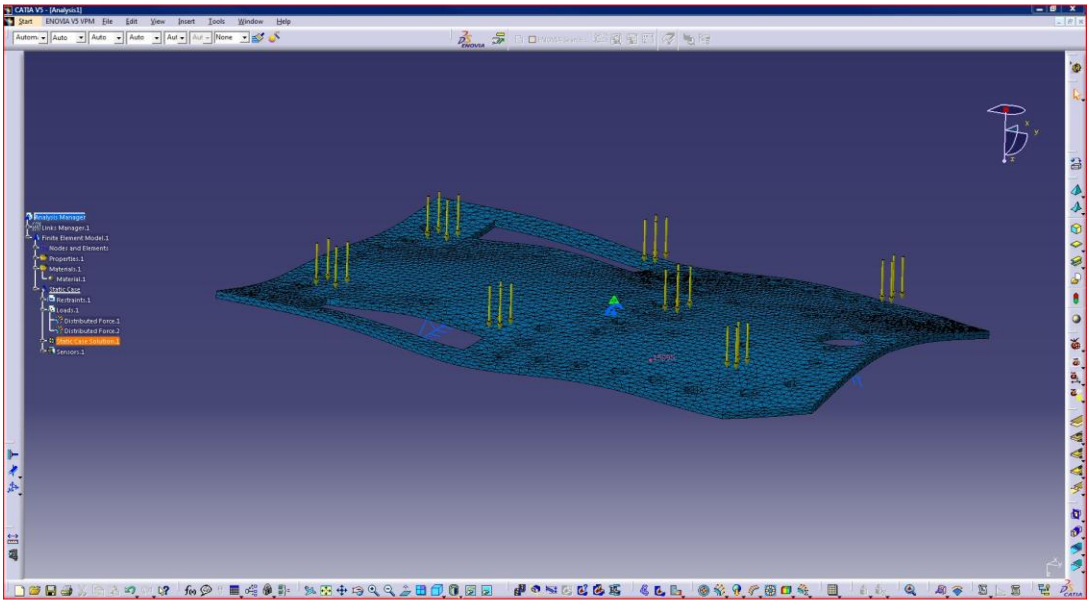
- Stiffness – it was important to choose a material that would not bend, even under heavier loading because some sensors were mounted underneath and bending would lead to these sensors malfunctioning or giving unacceptable errors.
- Upgradeability and weight – the first characteristic, upgradeability refers to the option of fine adjustments or redesigns of some parts, without requiring the whole chassis to be manufactured again. Weight is also an important factor, a heavy chassis leading to a much heavier finished product.
- Availability and cost – the chosen material had to be relatively cheap and easily or immediately available.

Considering the factors above, different materials were simulated: 2mm thick aluminium sheet, 1mm perforated stainless steel sheet, PCB FR4 glass reinforced epoxy laminate sheet, and 1mm carbon fibre. Some of these materials were practically tested: aluminium, steel and FR4 composite. After three prototypes the final version of the chassis was chosen and the 3D model is shown in the picture.

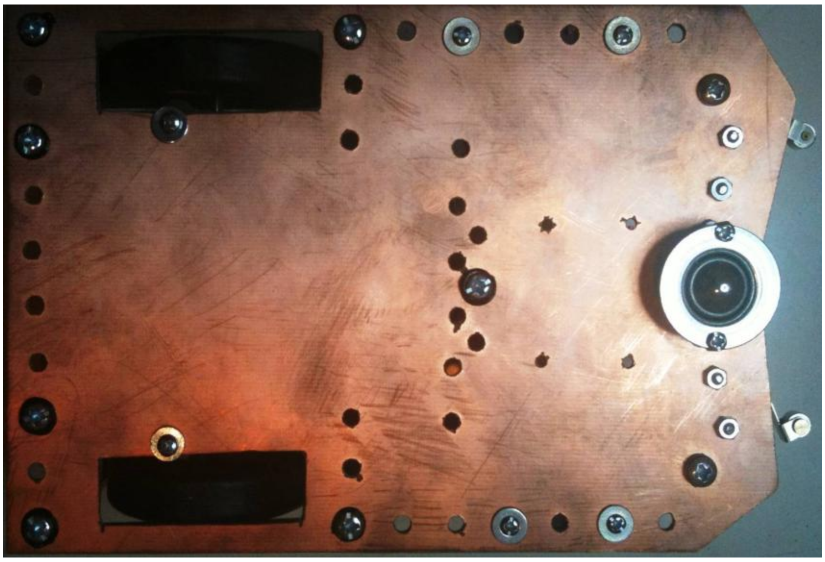


In order to thoroughly analyse the final chassis design, the 3D model is simulated using the CATIA software and FEA analysis. The results showed that aluminium would be a poor choice of material due to large deformation and poor elasticity. Iron gave better results but it proved to be too heavy. The best results were obtained with FR4 composite and this was the eventual design choice. It can be seen in the picture that the material displays little bend and good elastic proprieties (the simulation was done with larger forces acting upon the chassis for a worst case scenario).





The manufacturing process of each chassis is quite straight forward. The 3D model is transformed into a 1:1 scale drawing and this is used to mark each of the FR4 boards. After all the marking is done the holes are drilled and the rest of the cuts are made. All the cutting is done with a composite disc and the high speed router at 5000rpm. After the final cuts are made all the edges are sanded and components are checked to fit. The whole process from drawing to a finished chassis takes around 60 minutes per unit.



Mechanical tests:

Rig testing implied that a simple 3 point rig is made using M3 screws to set the desired height. Load tests are performed with the chassis on the rig using different loads for different parts of the chassis, consisting of calibrated metal weights. All the loads are kept under 1KG because the finished product will not put more than 30% of this loading on the chassis.

6.1.2 Additional materials used throughout the manufacturing and assembly

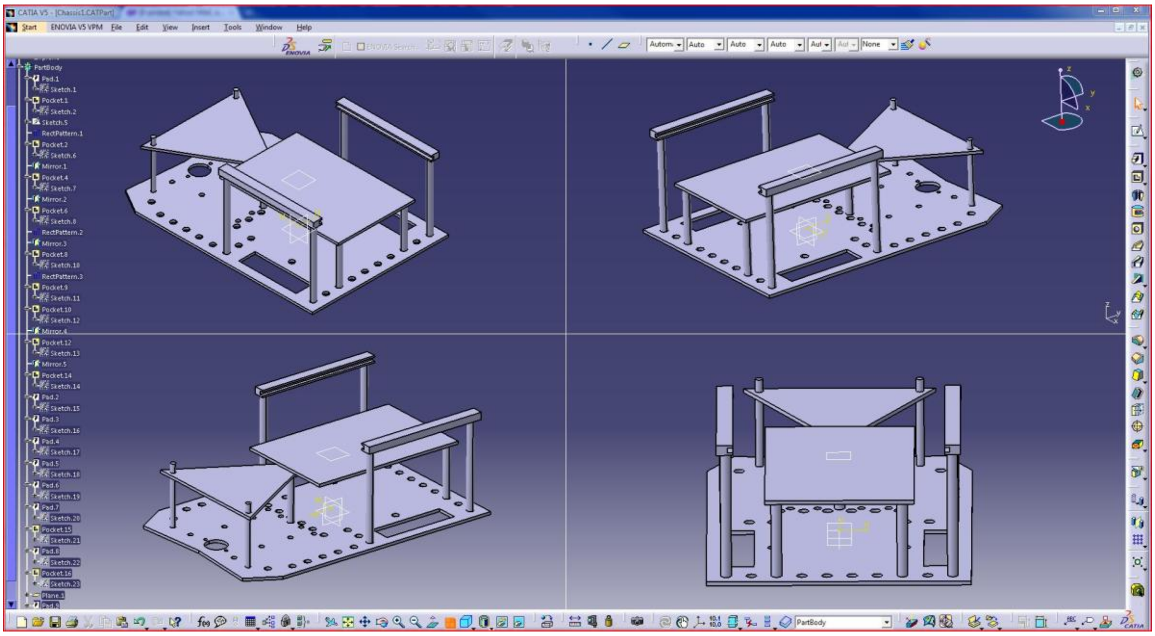
- Polymorph: Polycaprolactone (PCL) is a polyester material with a low melting point. When it cools it becomes very tough and displays good mechanical proprieties. If heated up to around 60 degrees Celsius, the material becomes malleable and it can be used to create any desired shape. It can be painted with water based paint. In this project the servo mounting and the food capturing mechanisms are made from polymorph material.

- Silicone: Industrial grade silicone is used, with good adhesive proprieties. It offers excellent adhesion on plastic, epoxy resin/ composite materials, aluminium and metal.

- Cyanoacrylate adhesive: high viscosity cyano glue is used for permanent bonding of parts. A thin layer is used to bond the aluminium heatsinks to the motor drivers. It has good heat transfer proprieties if used in thin layers and was the best option to expensive IC heatsink mounts.

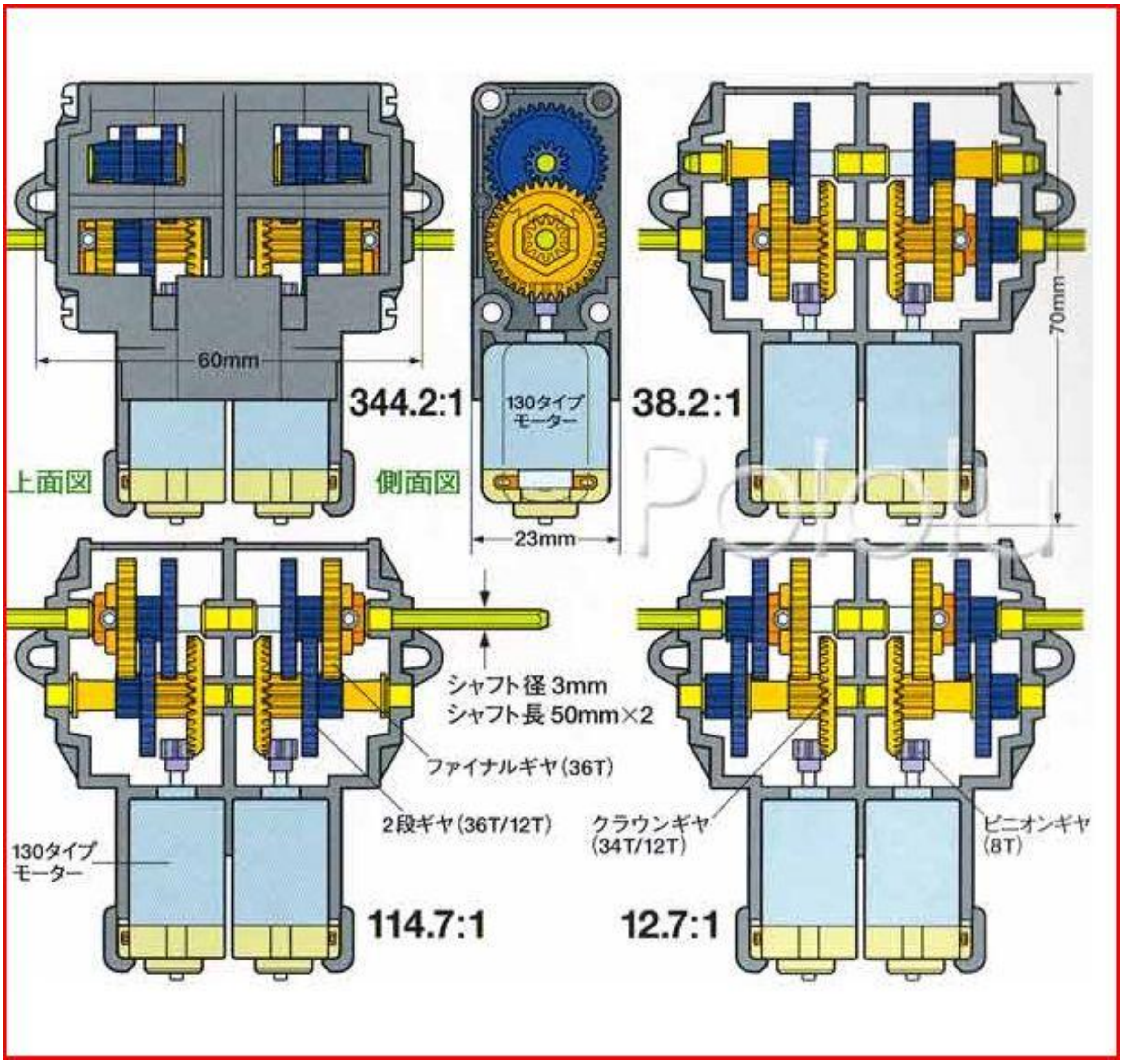
- PCB cooper and solder fixing mechanical parts: this is a simple and very practical solution to mount different metallic parts onto the chassis. It was used to mount the robot recognition sensors at the desired angle. It is a very cheap and easy to use solution, offering good mechanical resistance.

Complete chassis model is shown in the picture:



6.1.3 Drive-train: motors and gearbox

The drive-train was set to be identical on each unit. It had to offer individual motor control, enough speed and torque to carry the finished product and be reliable enough to provide maintenance free operation for the duration of the research. Form the available options the best one was a dual gearbox with individual motors, all packed within the same case. This unit is manufactured by Tamiya: 70168 Double Gearbox Kit. It is supplied as a kit and it offers four gear ratios: 12.7:1, 38:1, 115:1, or 344:1 with plastic spur gears. The gears offer a larger torque capacity than the motors are capable of providing, so there is no problem with deformation of gear teeth breakage.

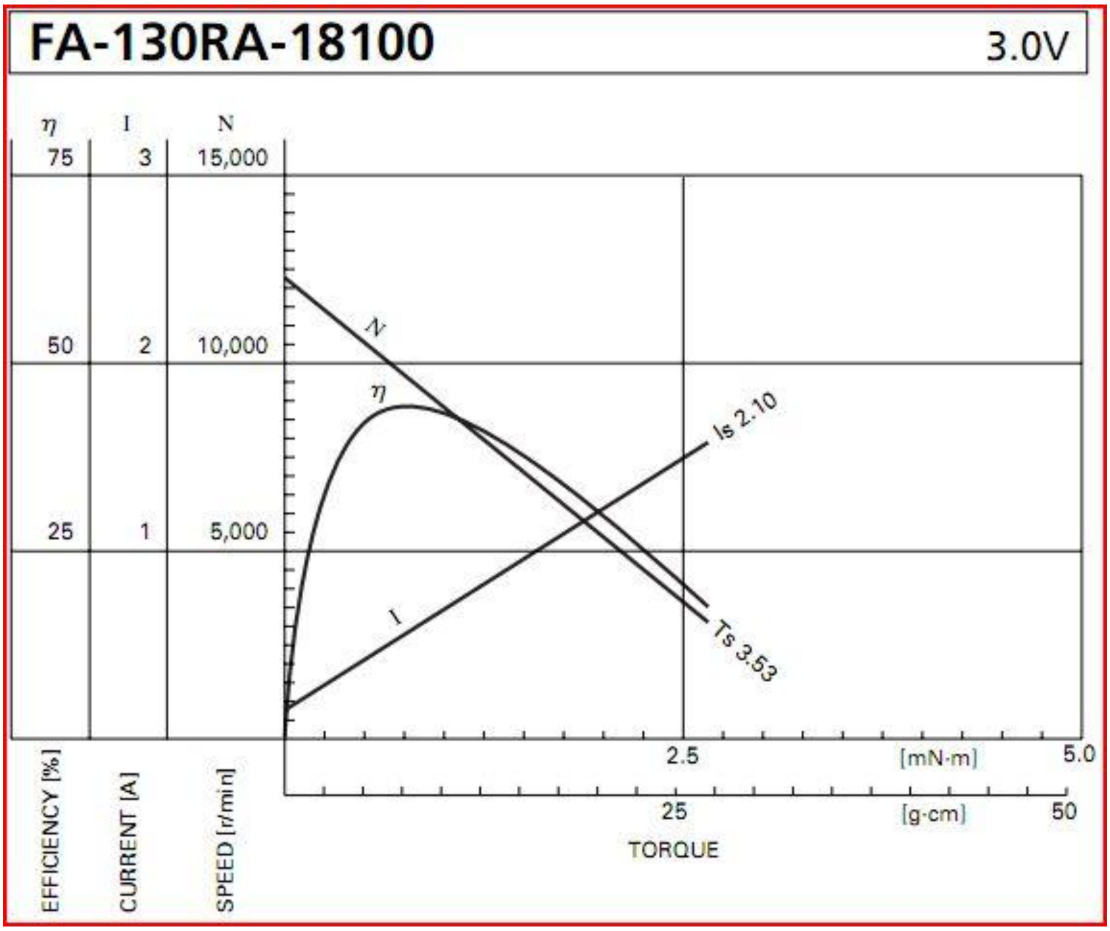


(Pololu Robotics and Electronics)

General specifications

Typical operating voltage:	3 V
Gear ratio options:	12.7:1, 38:1, 115:1, and 344:1
Free-run motor shaft speed @ 3V:	12300 rpm[1]
Free-run current @ 3V:	150 mA[2]
Stall current @ 3V:	2100 mA
Motor shaft stall torque @ 3V:	0.5 oz·in[3]
Color:	gray

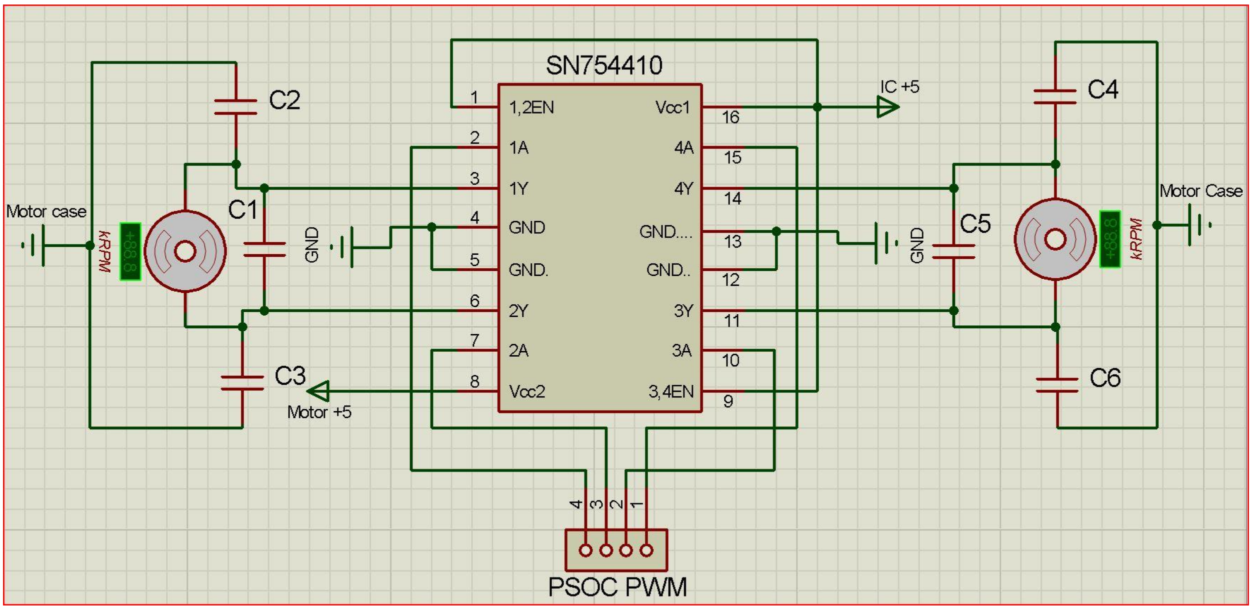
The supplied motors are Mabuchi FA-130RA-18100, rated at 3V. The picture shows the specifications of these motors on a direct drive specification, i.e. without any gearbox.



The drive-train comes as a kit with all the necessary parts included. The assembly instructions are straightforward and do not pose any problem. The only aspect worth pointing out is that enough grease must be used on the gearing system to assure good lubrication.

The first selected gear ratio was 115:1 but the system speed was too high, so 344:1 ratio was eventually chosen, offering a reduction in speed and an increase in torque. The chosen wheels offer enough grip for the current research but do not exceed the torque figure of the gearbox, thus serving as a fail-safe mechanism. If the robot gets stuck, the wheels will keep turning and will not lead to an over-current situation that would be dangerous for both the motors and drivers. The fail-safe mechanism was tested and proved successful.

Completed assembly of chassis and drive-train current draw and noise measurements were done under different loading situation. Because of the chosen gear-ratio there was enough torque available under normal motor operating conditions, preventing currents larger than 0.7 A to be drawn under all tested loads. Noise was not measured but checked to be low enough in order not to interfere with the surrounding electronic modules. All motors were fitted with bypass capacitors as a precautionary measure.

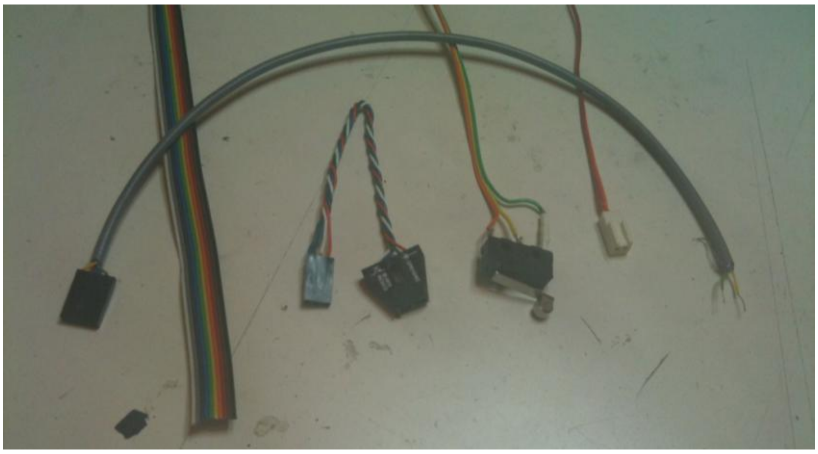


6.1.4 Connectors and cables

Due to the complexity of the system and tight packaging, implying motors, microcontrollers and other noise sources running close to sensors and signal wires, carefully selected cables and connectors must be used. Important aspects when choosing the cables and connectors are size, directionality, colours available and current rating. Size matters both form cable and connector perspective. Directionality refers to the way a connecter is inserted, especially important for power lines, to prevent inversion of polarity.

Available colours are important when the sensors are connected to the microcontroller, to avoid mixing sensors up and wrong connections. The maximum current carrying capacity of a cable must be considered for the power and motor lines. Current measurements must be conducted to ensure that cables are within safety operation area and prevent melting and short-circuits.

Mostly silicon insulated ribbon cables and twisted pairs are used throughout the system. The obstacle avoidance and colour sensors are fitted with screened cables to avoid any interference that might get coupled onto the signal wires.

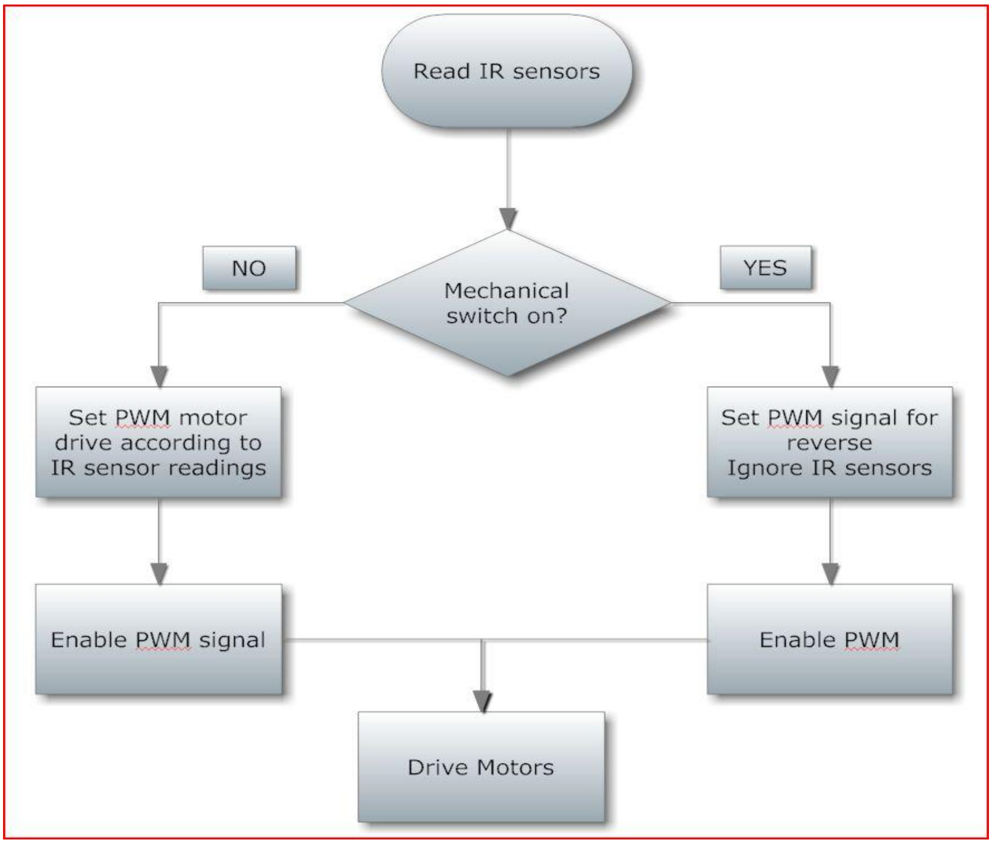


Current carrying tests were done using calibrated 10W ceramic resistors. All the power cables were tested and proved capable of carrying currents in excess of 1.5 A, offering enough over-current margin.

6.1.5 Motor controller

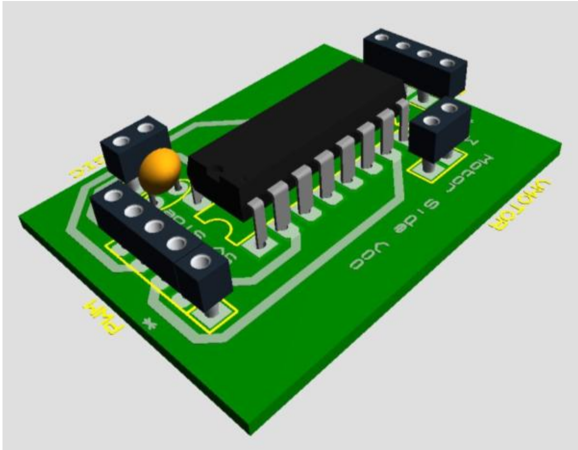
After deciding on which gearbox/ motor combination to use, an appropriate motor driver had to be chosen. The considered options included a discrete H-bridge, a larger current integrated circuit driver– L298 and a smaller current IC driver from Texas Instruments that also offered other features, such as temperature safety shut-off, integrated output diodes etc. All these units were available so practical tests were the best way to determine the most suitable for the project.

Each motor driver is controlled with a mixed PWM signal from the microcontroller. This control method is called locked antiphase and it involves two PWM signals generated in antiphase to each other. The signals are generated using a PWM generator and an inverting gate available within the PSoC (both PWM signals are shown in the picture). For a complete stop, the PWM signal must have a duty of 50. This is the zero value, and a higher duty will drive the motor in one direction (duty cycle close to 100), a lower value will reverse it (duty cycle close to 0).

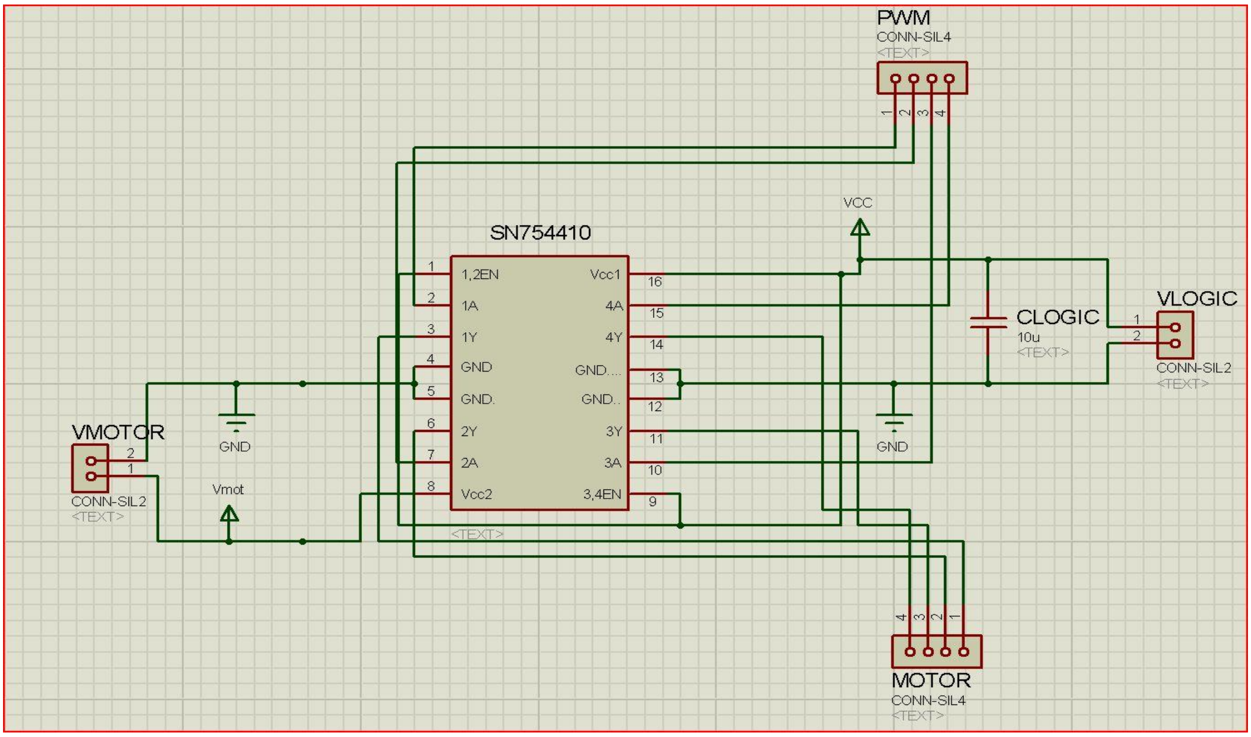


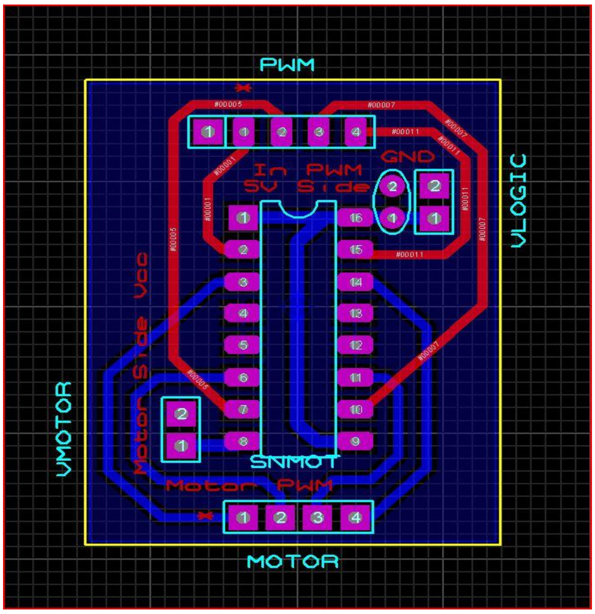
<u>Motor controller breadboard test:</u>

L298 motor driver did not function on the available voltage options, requiring 12 V to run reliably and an H-bridge discrete driver was impractical and expensive, therefore the best option was the TI SN754410 driver. After an initial breadboard test, to prove that it was able to drive the motors reliably with the available resources, a PCB was created and further tests were conducted.

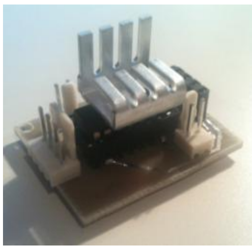


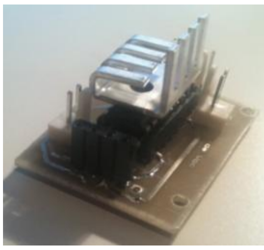
SN754410 driver schematic and PCB:





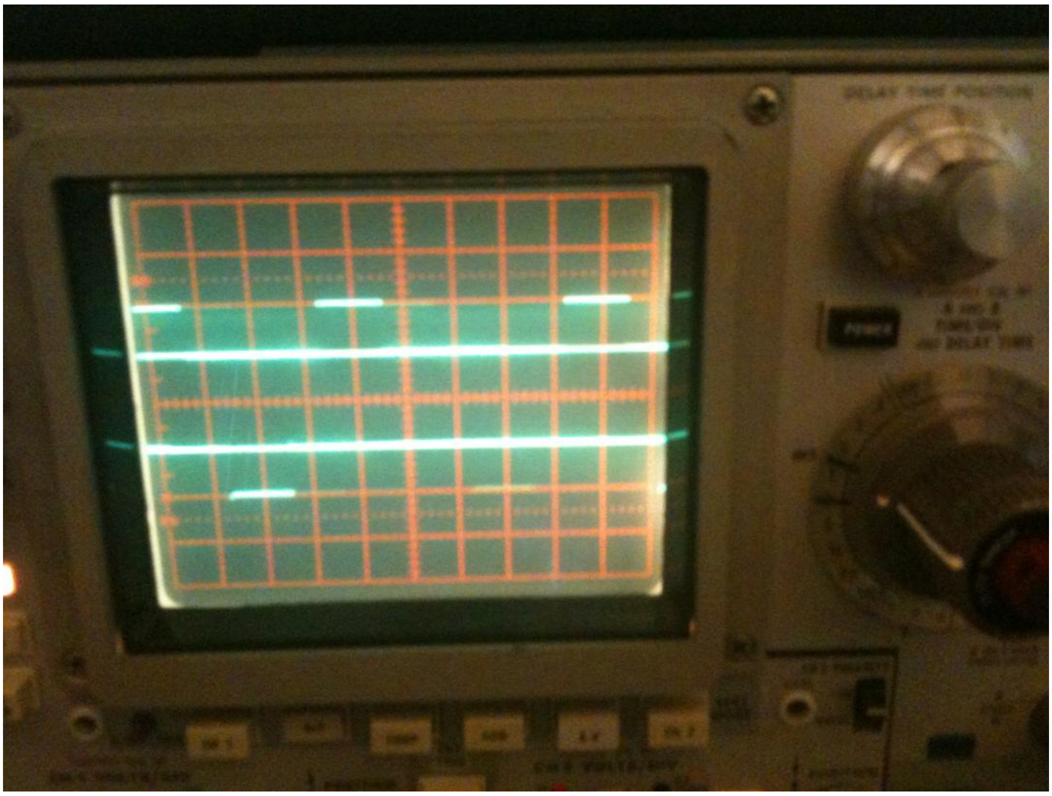
Running tests were done for periods of 30 minutes with 1 hour rest periods and there were no problems. The unit proved to get hot, up to 85 degrees Celsius, but remained reliable and up to the moment no drivers were destroyed, despite the extensive testing some units were subject to. However, small aluminium heatsinks were still fitted as an extra precaution. This proved a very good way of lowering the module temperature, offering a stable 65 degree operating temperature, compared to the initial 85 degree.

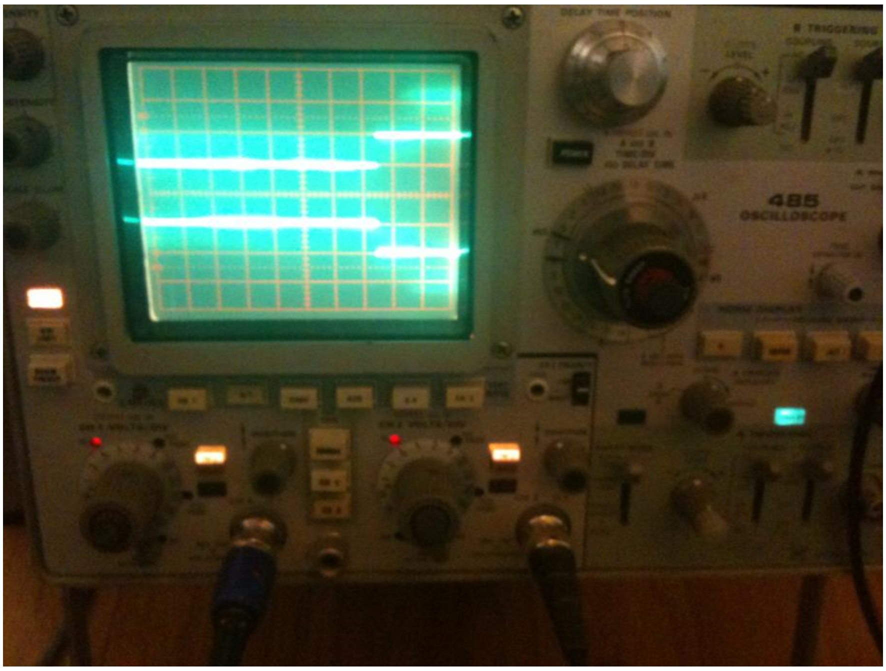




```
speed1= 5 + z*5;     //set speed for motor 1 -- pseudo fuzzy control derived directly from IR
LCD_Position(1,10);
LCD_PrHexInt(speed1);

speed2= 5 + y*5;     //set speed for motor 2 -- pseudo fuzzy control derived directly from IR
LCD_Position(2,10);
LCD_PrHexInt(speed2);

PWM8_1_WritePulseWidth(speed2);
PWM8_2_WritePulseWidth(speed1);
```

The PWM waveforms are shown below:





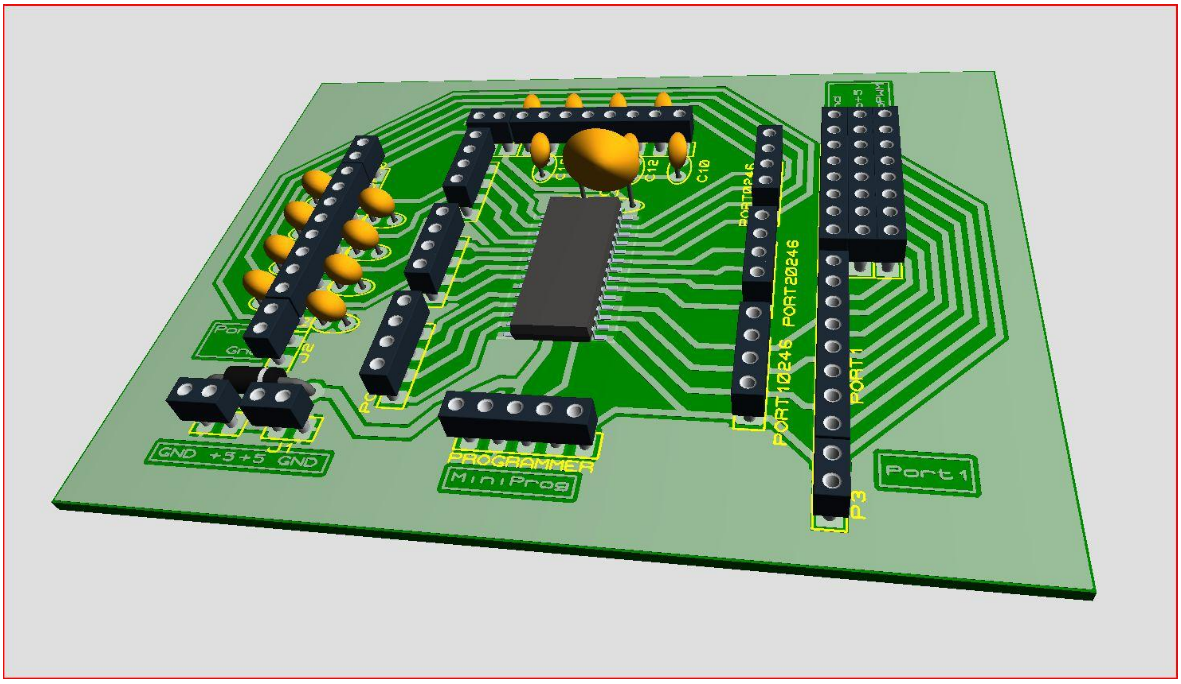
6.1.6 Microcontroller

The processing unit is the single most important element on the whole robot. It is the brain of the system and if it fails the unit will not function any more.

The microcontroller chosen for this project was Cypress Semiconductor's PSoC1. This is a fully programmable system on chip, offering configurable analogue and digital blocks and an 8 bit microprocessor on the same chip. The main reason for choosing this unit is the vast analogue options it offers, such as operational amplifiers, comparators, filters, different topologies of ADC and DAC and hardware support for different communication protocols. The unit does not need any external components like oscillators and capacitors, making it the most appropriate choice for this project.

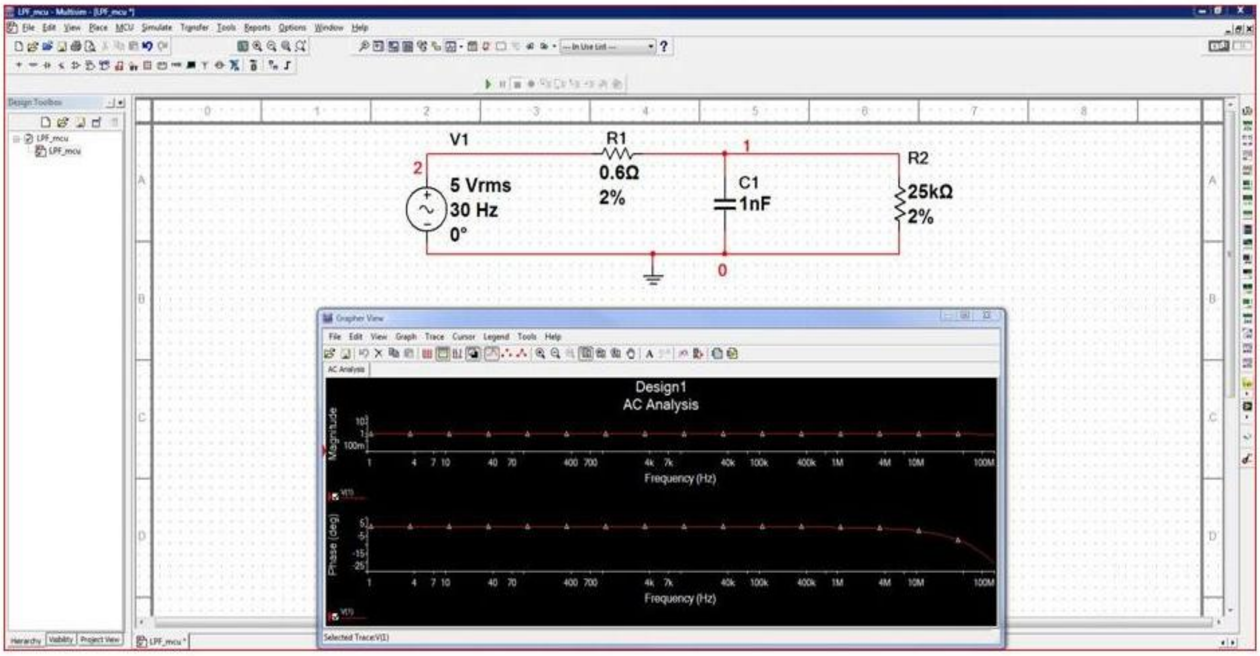
All the hardware afferent to the microcontroller was designed specifically for this project and a complete development board was created. The main features of the board are power supply availability for each individual port, male and female connectors to enhance versatility and serve as test-points for debugging, and a dedicated 3 pin header for RC servo control. There is the option for a separated servo power supply but it was not required.

A 3D model of the final board is shown in the picture:



Measured element:	Value:
Track resistance	0.6 ohm on the longest tracks
Capacitance: track to track	Less than 1 nF
Capacitance: track to ground plane	Less than 1 nF
Inductance	immeasurable with the available equipment

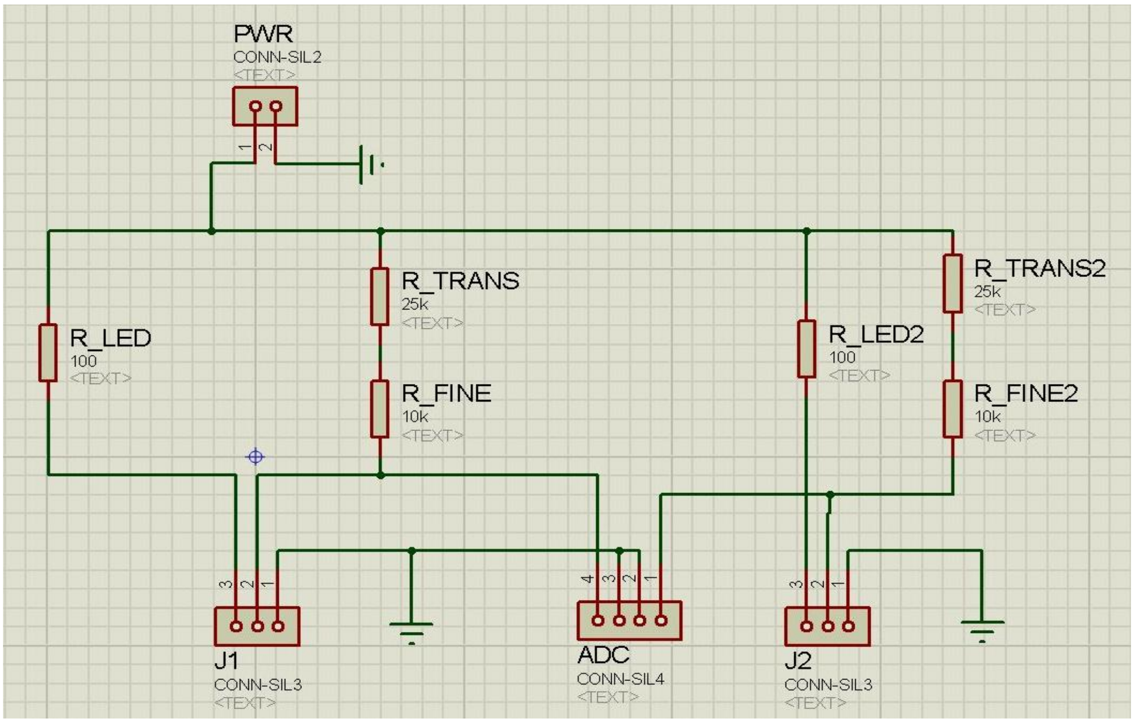
Because of the existing parasitic elements, a low pass filter is created. The filter is simulated in Multisim, to certify it will not interfere with the normal system operation.

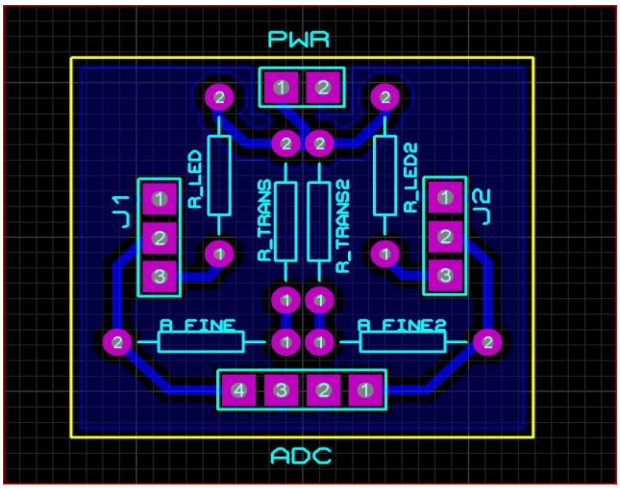


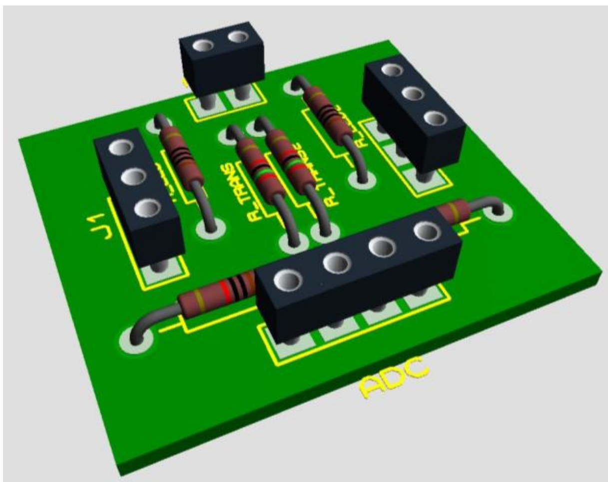
6.1.7 Obstacle avoidance sensors

The main purpose of the obstacle avoidance sensor was to output a logic “high” value when the obstacle was in the preset distance. A better obstacle avoidance sensor would also offer distance measurement and tell the robot precisely how far the obstacle is in order to generate a smoother navigation algorithm.

The obstacle avoidance sensor on each unit is made of a pair of IR sensors and a pair of mechanical contact sensors that must back up the optical ones. The optical sensor diagram is shown in the picture. This diagram is used with different sensing packages available to determine the most suitable unit.



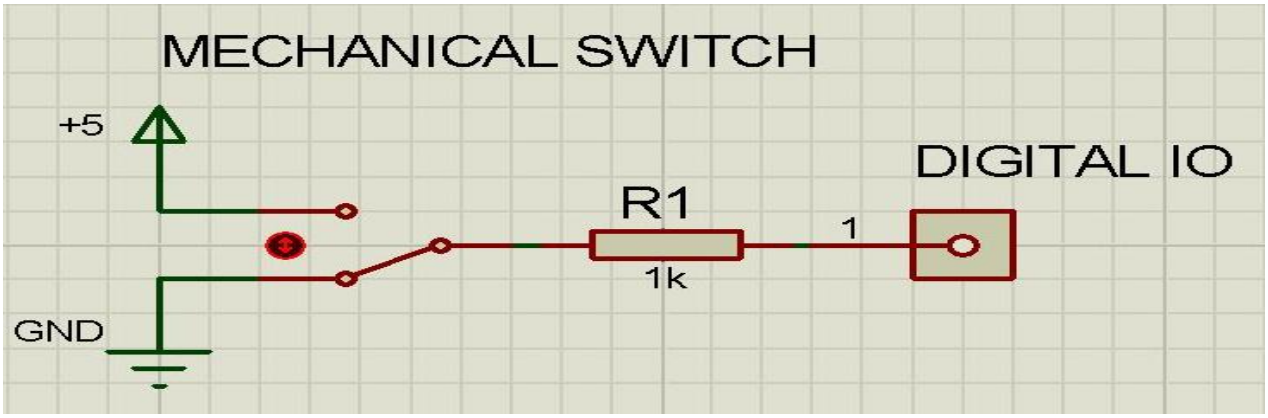




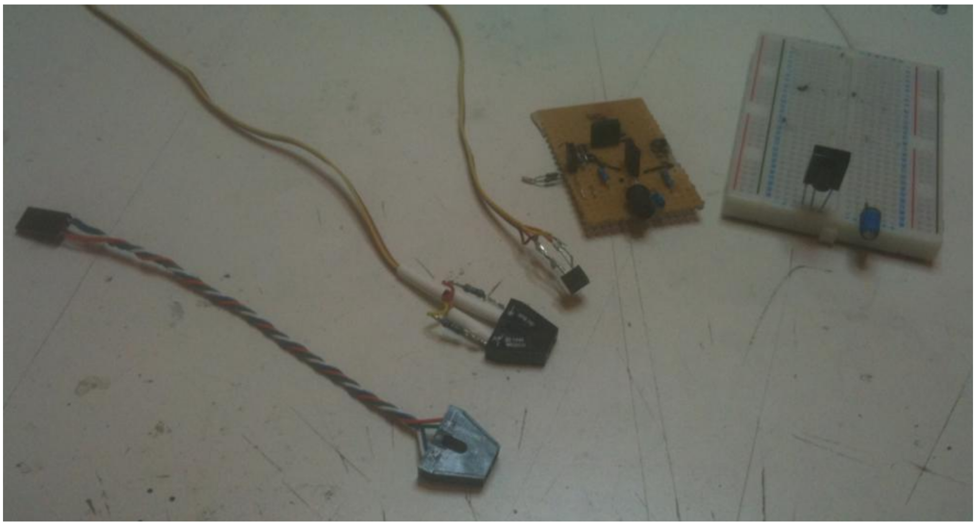
Obstacle avoidance IR – table of distance versus values:

Distance from robot (cm)	Voltage (V)
1	0.2
2	1
3	2.3
4	3.5
5	4.4
6	4.8

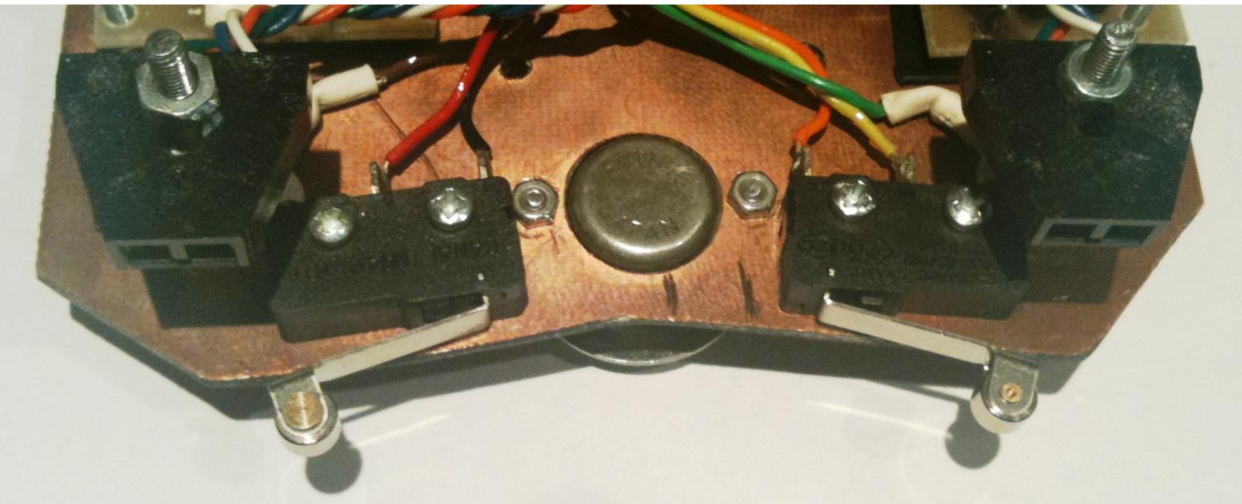
Mechanical sensors are used as three state switches, usually at high logic level, 5V, going low, 0V, when active.



Different IR sensors were available for testing but the best results were obtained with the ones currently on the platform. The tested units are shown in the picture below.



Complete obstacle avoidance assembly with IR sensors and mechanical switches:

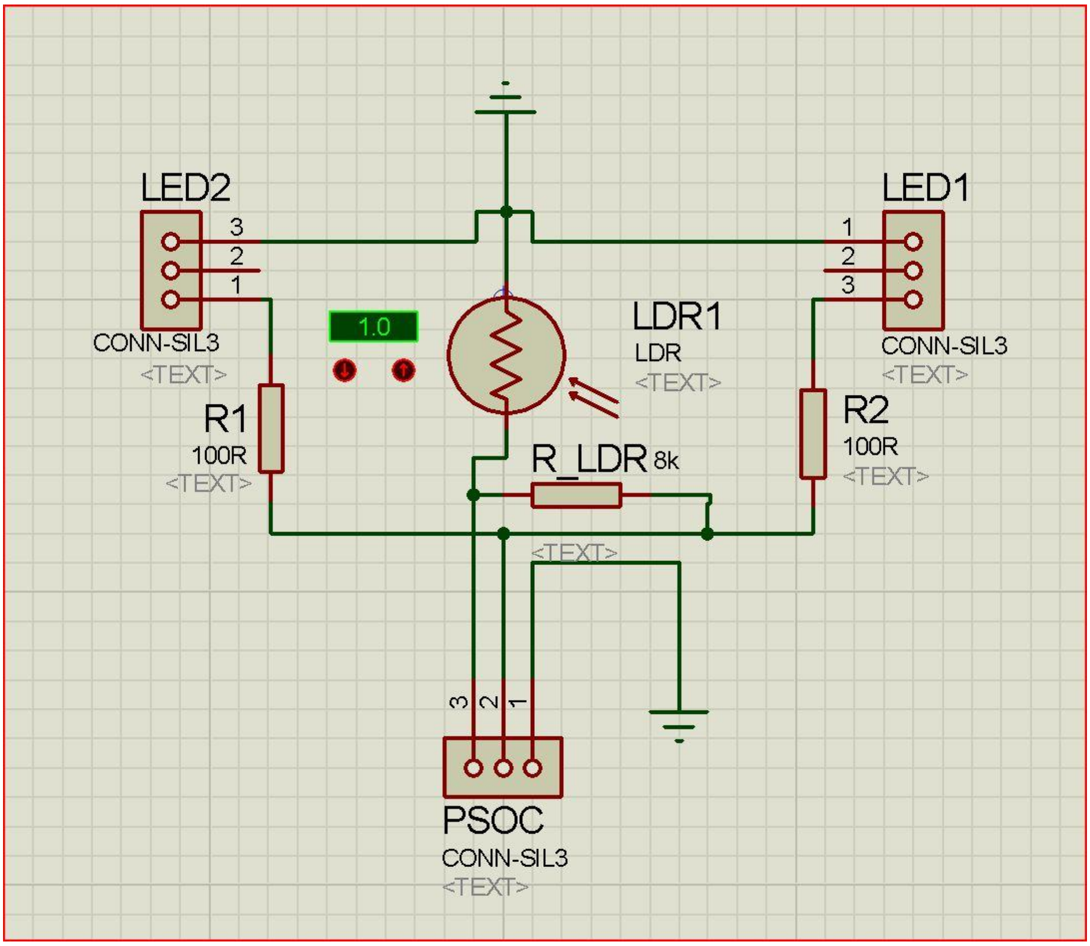


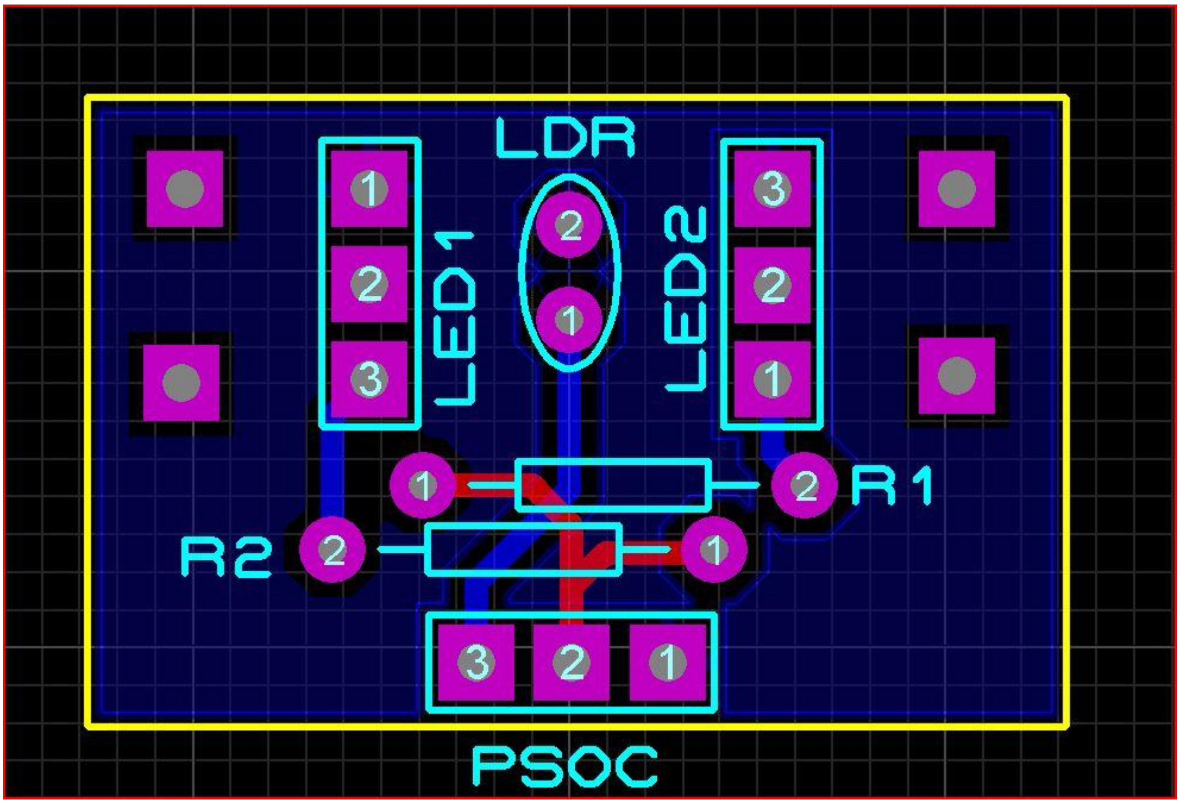
```
//---------------------------------Sample IR1----------------------
for(i=0; i<5; i++){
AMUX8_InputSelect(AMUX8_PORT0_1);
if ( DelSig_fIsDataAvailable() ) {
z = DelSig_iGetDataClearFlag() / 100;
LCD_Position(1,0);
LCD_PrHexInt(z);
}
i++;
}

//---------------------------------Sample IR2---------------------
for(i=0; i<5; i++){
AMUX8_InputSelect(AMUX8_PORT0_2);
if ( DelSig_fIsDataAvailable() ) {
y = DelSig_iGetDataClearFlag() / 100;
LCD_Position(2,0);
LCD_PrHexInt(y);
}
i++;
}
```

6.1.8 Colour detection

The colour detection sensor consisted of a light dependent resistor, LDR, and two white LED units. This is basically a sensor that detects brightness, but with proper placement and considering that different colours absorb or reflect waves more than others, it is possible to transform it into a colour detection sensor.

The sensor will output different voltages for different colours and the software is able to associate each voltage value to a preset colour. In the design of this sensor it is very important to consider shielding the sensing element – LDR from the white light emitted by the LEDs. The sensing element must only be exposed to reflected light from the coloured element. Another available option is an IC sensor from TAOS and this was also tested but it proved too sensible for the current project so the initial custom solution was kept.



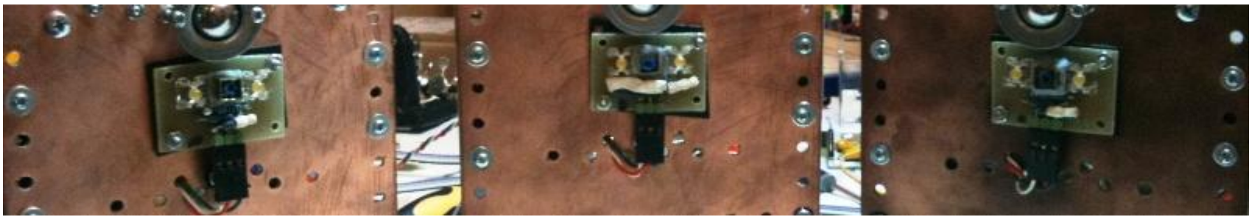


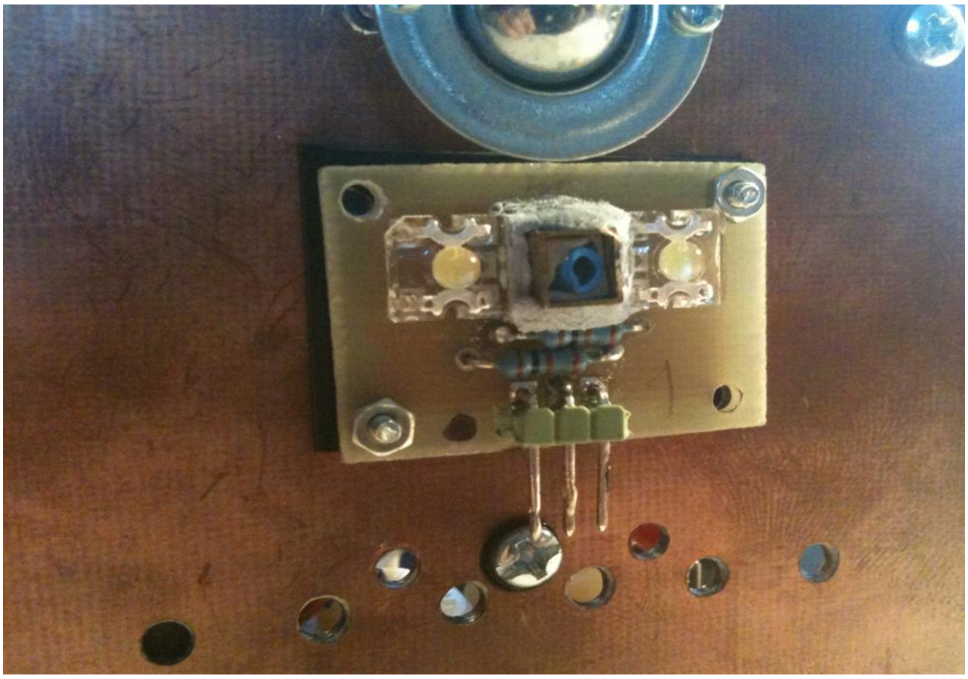
The transducer element, the light dependent resistor is the most important part of this module and selecting a wrong element could lead to an unusable sensor. The key aspect is to select a light dependent resistor that offers a large resistance variation with light. The element used in the current sensor is the NSL-19M51, CdS photoconductive cell from Silonex, offering a measured resistance range from 1k to 300 kOhms. Other sensing elements were tested but the Silonex unit proved to be the best option.

Table of values for different colours:

Colour	Voltage	LDR - Resistance value
Yellow	1 V	2.2 kOhm
Pink	1.13 V	3.4 kOhm
Orange	1.25 V	3 kOhm
Green	1.8 V	6 kOhm
Dark blue	2.8 V	20 kOhm
Black	3.8 V	35 kOhm
White	0.8 V	1 kOhm

Completed colour sensor, mounted underneath the robot:

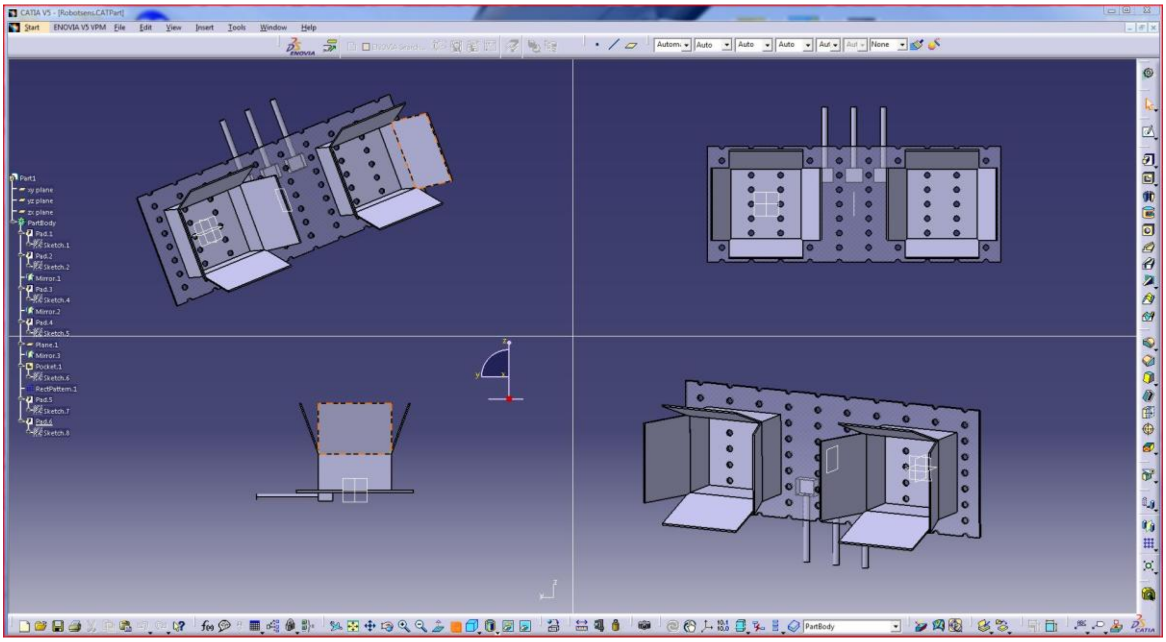




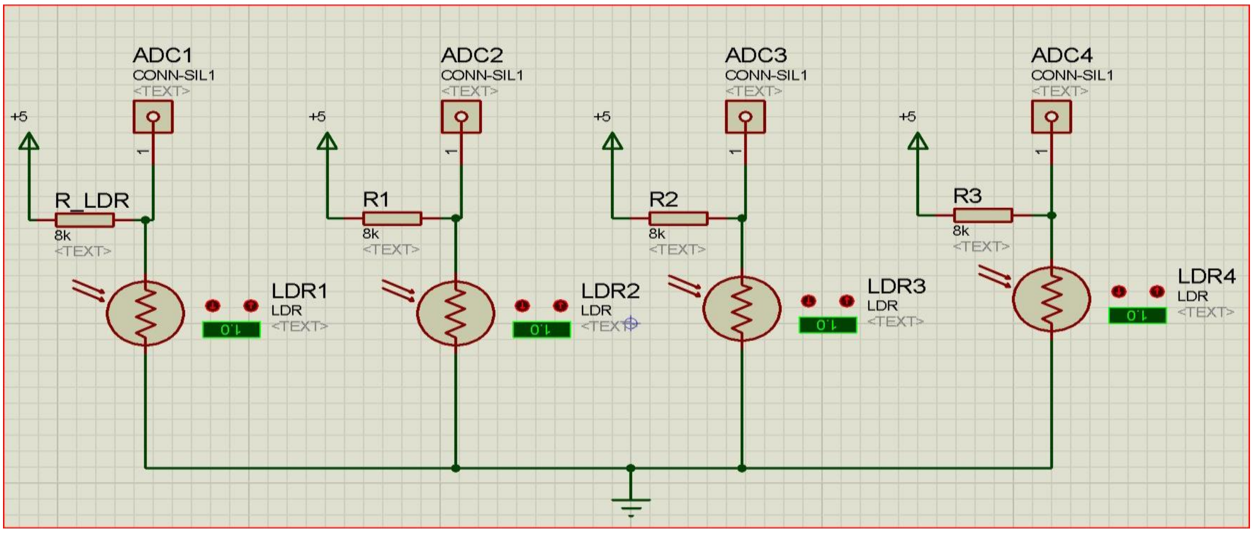
```
while(1){
//-------------------------------Sample colour sensor--------------------
for(i=0; i<5; i++){   //sample each value for 5 times - better accuracy
AMUX8_InputSelect(AMUX8_PORT0_0);
if ( DelSig_fIsDataAvailable() ) {
iSample = DelSig_iGetDataClearFlag() / 100; //colour value
LCD_Position(0,0);
LCD_PrHexInt(iSample);
}
t = iSample ;        //converted to char - will be sent to other units
LCD_Position(0,10);
LCD_PrString( &t );
i++;
}
```

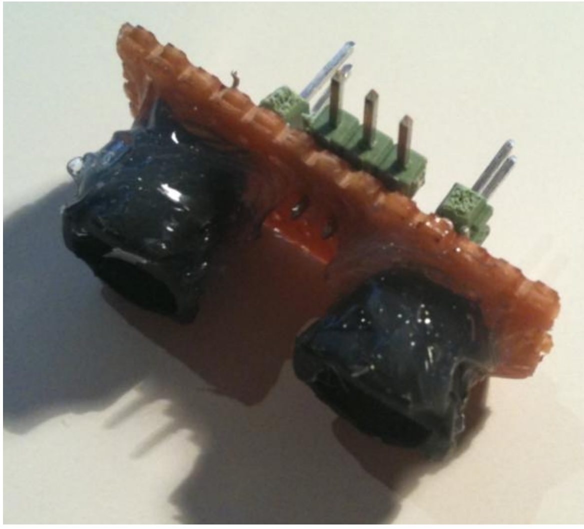
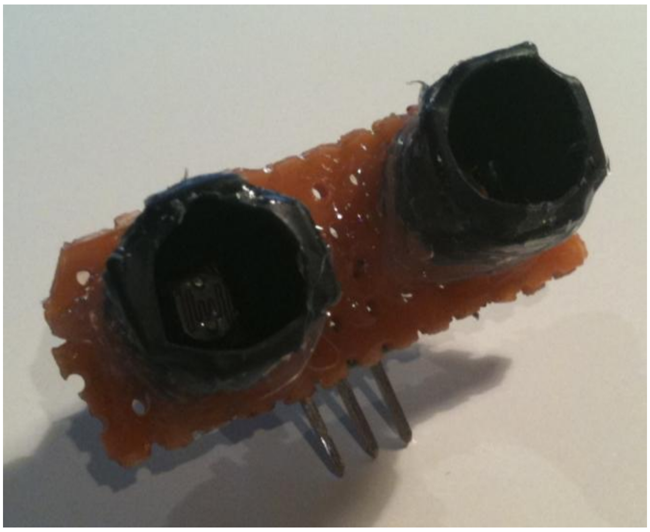
6.1.9 Robot and food recognition sensor

The purpose of this sensor is to interpret received data and decide if the robot detected another robot or a food source it can grab. It is an originally developed sensor for this research and it is also based on colour detection. Each robot will be assigned a colour and based on this colour some LED placed around each unit will be powered. This enables the robots to detect each other and also distinguish between another robot and an available food source.



The mechanical model is shown in the picture and the electrical schematic can be found below.



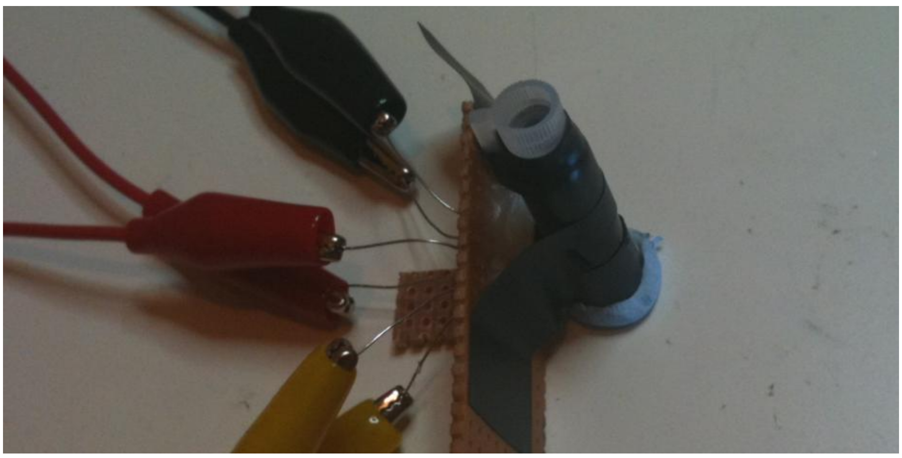



PSoC requirements:

Four 12bit ADC blocks and the required functions for data retrieval and storage are needed. Comparators can be used instead of ADC blocks but the sensorial perception is limited.

The main problem with this sensor was the white, highly reflective surface of the workbench. At the first measurements the sensor was receiving too much external light and the values were very closely grouped. Different solutions were tested but the best results were obtained using a black surface for testing. This is actually closer to a real world scenario, where a white terrain is rarely available.

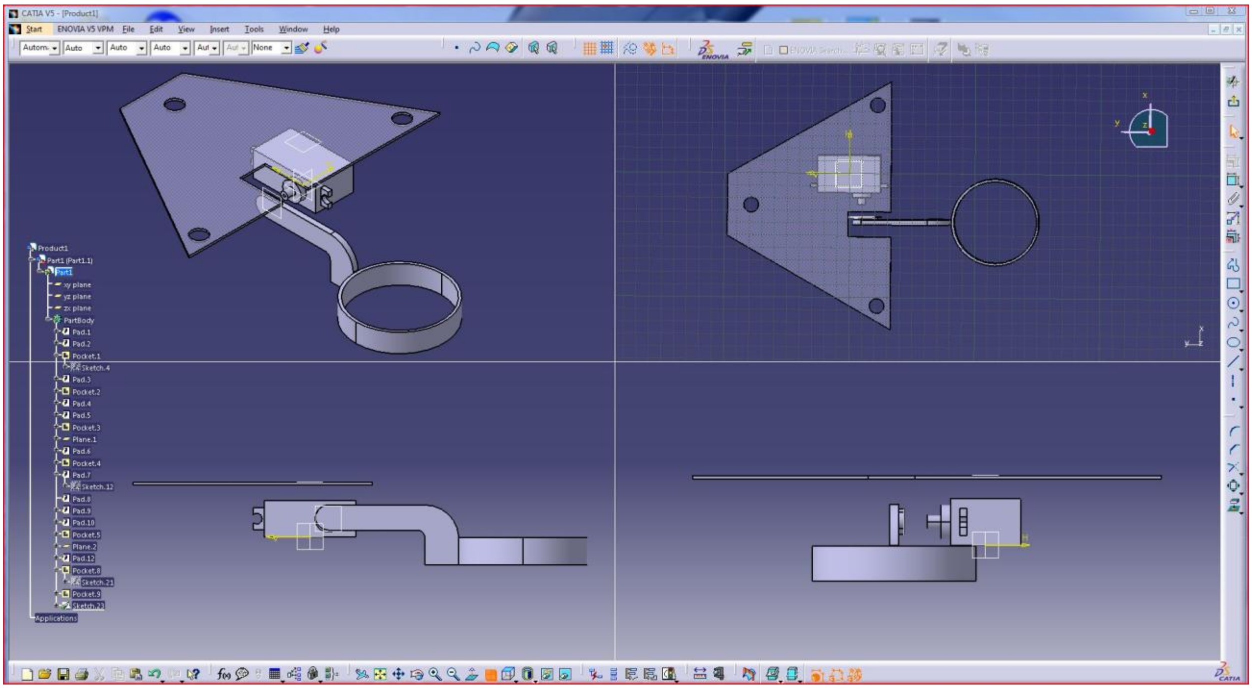
This sensor is the second, refined implementation of the concept. Another unit was designed for the same purposes but it was more complicated, involving a cylindrical lens and an array of three LDR in a line, with good individual light shielding. It worked as intended but it was not as well performing as the current design.



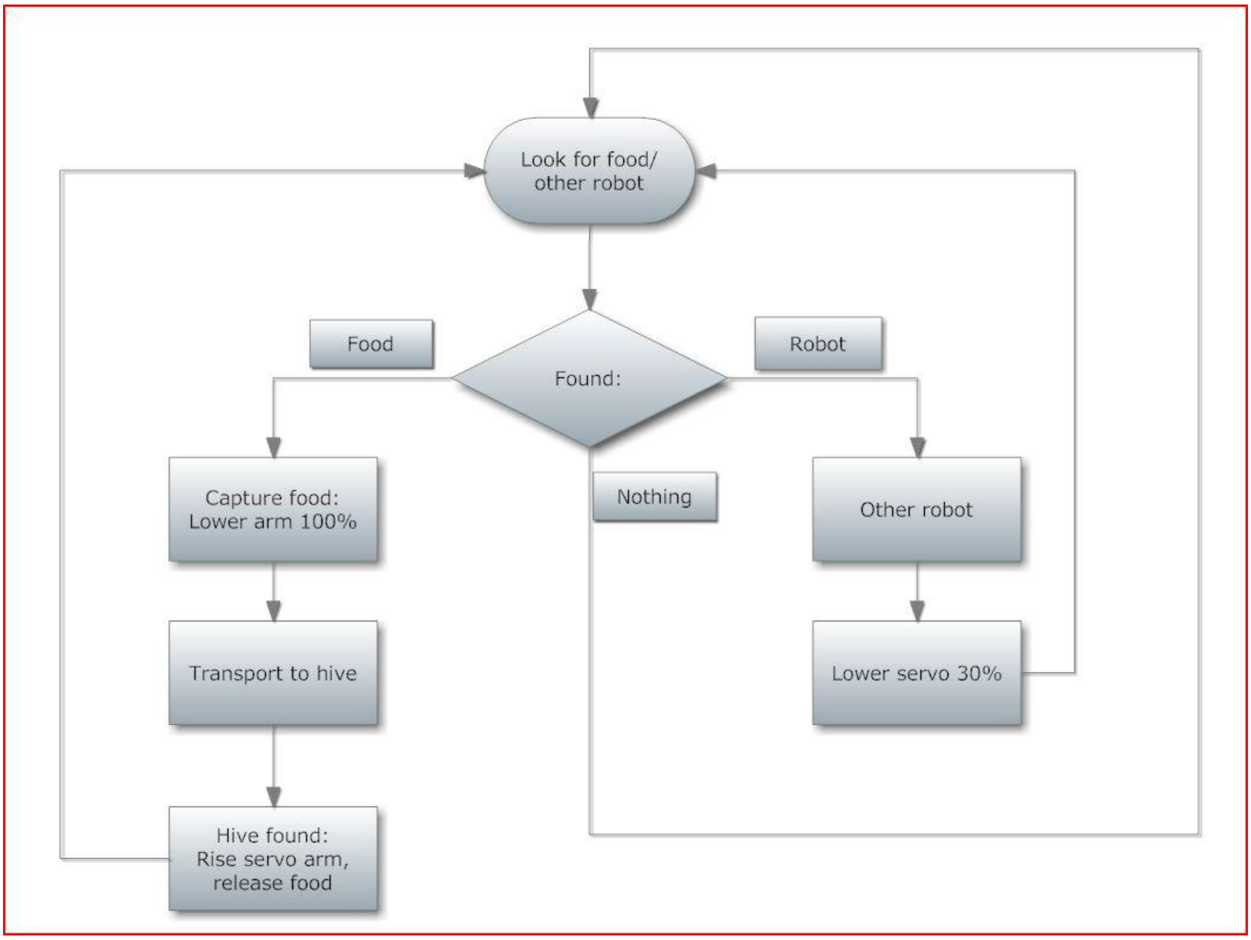
6.1.10 Food capture

The food capture mechanism is designed to be a simple way in which the robot can manipulate available food sources. It was based on the principle of pushing the food around and does not offer any lifting proprieties. The mechanical system consists of a small RC servo connected to a custom made capturing element. This module works in conjunction with the detection sensor previously described. It also serves as a starting point for a robot linking mechanism that can be developed in the future.

The whole system was initially modelled and adjusted in CATIA. After a satisfactory compromise between all dimensions was reached the first prototype was build.



Relationship between the food sensor and capturing mechanism is presented below:



<u>Servo control:</u>

The RC servo unit requires a PWM signal to function. The required duty cycle is derived from the sensors that tell the robot if it is facing a food source or another unit. For a food source the robot must trap it but if it is facing another robot it must only lower the servo arm 30% as an acknowledgement of the other unit. A simple look-up table and comparison tell the robot is it found a food source or another unit.

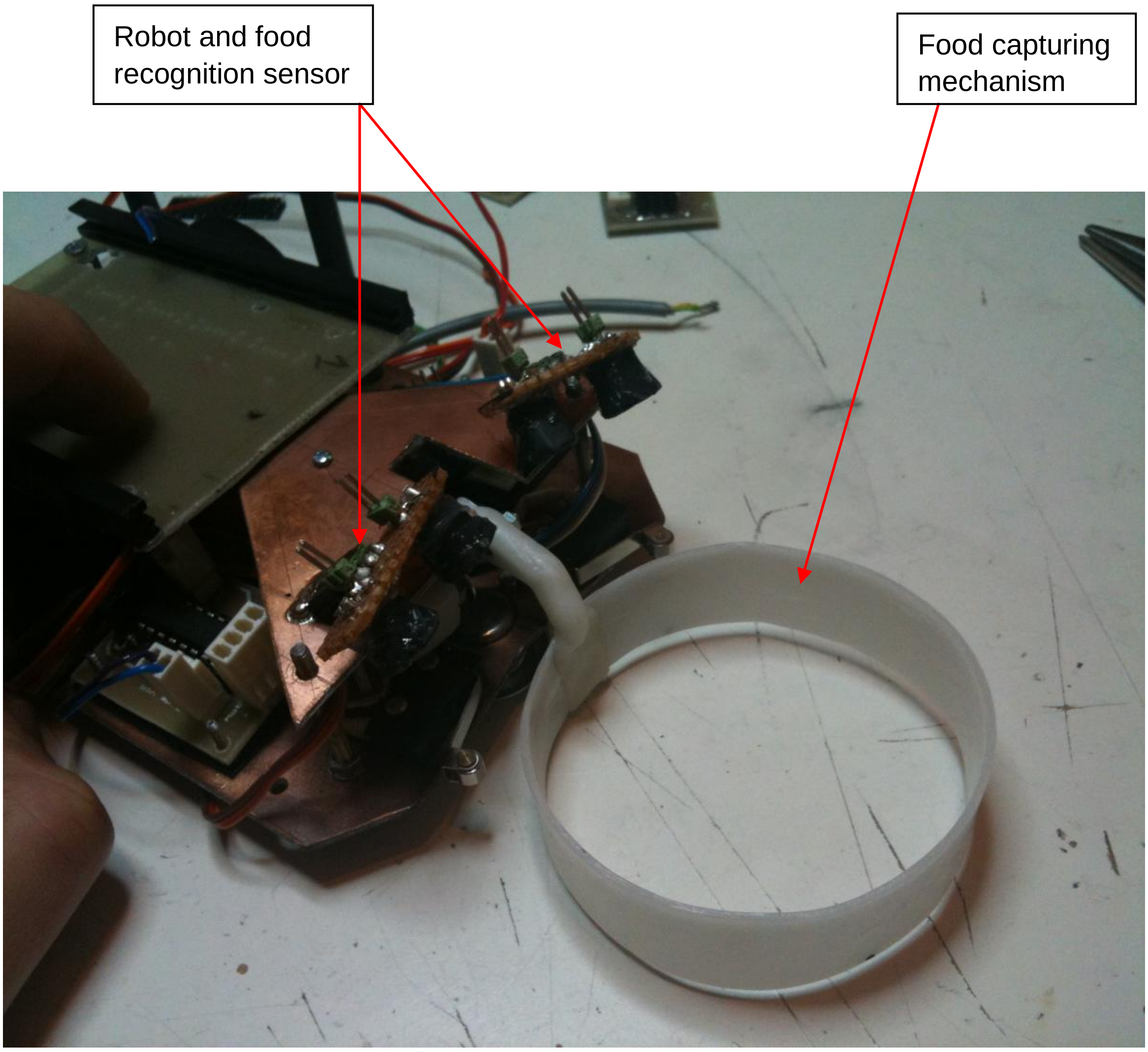
The equations used for determining the PWM signal period and frequency are:

T.out = (PeriodVal + 1) / F.clock

F.out = F.clock / (PeriodVal + 1)

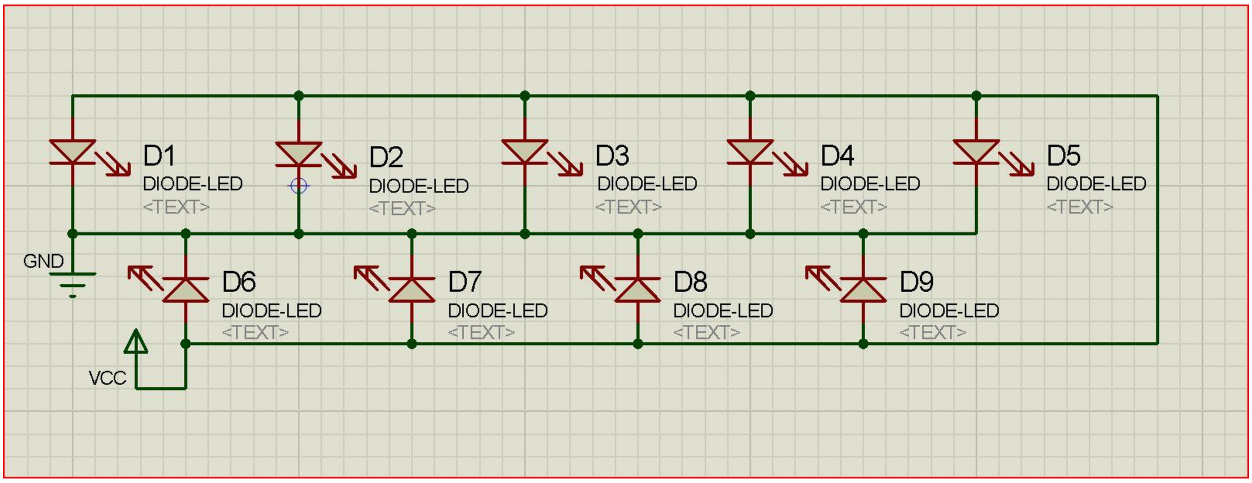
Where T.out is the period, F.out is the frequency of the PWM signal and F.clock is the frequency of the clock input to the PWM module.

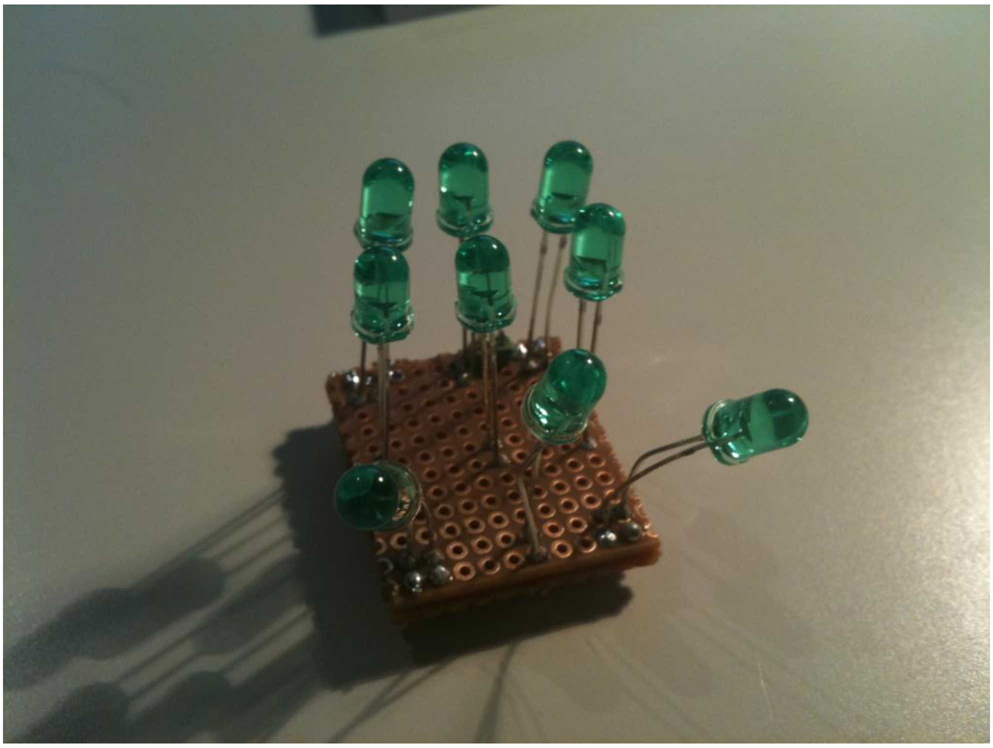
The system proved to be quite successful, identifying the food sources most of the time and showing good robot recognition abilities.



6.1.11 Food source

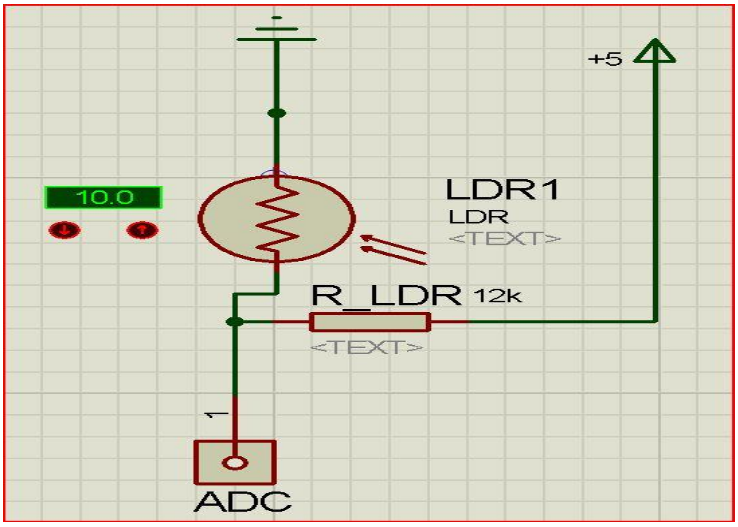
A simple LED array powered by a small 3.2V battery pack is used as food source.





6.1.12 Light detection sensor

The light detection sensor consists of a light dependent resistor that is not completely exposed to the environmental light in order to motivate the robot to search for the most powerful source of light. A simple unit is implemented in the available robots. It would be much more useful to have the sensor used in conjunction with a solar power cell, to determine the best placement for the robot, in order to get the most available sunlight.



```
//--------------------------------Sample light sensor--------------------
for(i=0; i<5; i++){    //sample each value for 5 times - better accuracy
AMUX8_InputSelect(AMUX8_PORT0_3);
if ( DelSig_fIsDataAvailable() ) {
light = DelSig_iGetDataClearFlag() / 100; //light value
LCD_Position(0,0);
LCD_PrHexInt(light);
}
l = light ;      //converted to char - can be sent to other units
LCD_Position(0,20);
LCD_PrString( &l );
i++;
}
```

6.1.13 Battery selection and management

The main power source selection criteria were rated voltage and current, capacity and size. The smallest unit was desired, but the battery was required to provide the necessary running time and current to power up all system. A brief energetic analysis is done to evaluate the general requirements for each robotic unit.

Required_Energy = 2 * Kinetic_Energy + Energy_Losses + Conversion_Losses
Required_Energy = Mass * Velocity^2 + Deceleration_rate * Distance traveled * (1/2 * Mass * Velocity^2) + Energy_Losses + Conversion_Losses

Robot Stats:

Mass	1	kg
Maximum Velocity	0.5	m/s
Travel Distance (per Trip)	3	m
Travel Final Height	0.1	m
Trips	3000	trips
Battery Voltage	8	volts

A Trip is the sum number of times your robot starts and stops divided by 2. Battery Voltage is the voltage your robot motors will be powered at. Travel Final Height is defined as ending_height - starting_height.

Estimate Losses:

De-acceleration Ratio	3				(unitless)
Energy Conversion Efficiencies					
Chemical	80	%	Mechanical	60	%
Electrical	80	%	Thermal	90	%

Remember to check the **robot parts list** to buy your batteries.

Calculated Results:

Total Energy Losses	1.34e+4	joules
Total Efficiency	0.346	%
Required Energy (per Trip)	2.04e+4	joules
Total Required Energy	6.81	joules
Estimated Time till Completion	6.67	hours
Required Battery mAh	710	mAh

(Society of Robots, 2008)

The best choice was to use a LiPo battery, and so a 2 cell unit was chosen. It offered 1000mAh, a maximum current above 10A and small size per unit.

The only problem when using this LiPo is that the battery voltage cannot drop below 3 V/ unit meaning 6V/ package. This is taken care of by software monitoring of each individual cell voltage.

There are two ways to implement the battery management function, either using a dedicated ADC conversion or with a comparator set on a predefined voltage level. The first solution, offers a better control over the available power resources, being able to monitor the exact battery voltage level for each cell. An important detail is that the voltage on each LiPo cell will not exceed 4 – 4.5 V so the integrated ADC can easily handle individual cells, but if the battery pack is to be monitored, a linear voltage divider can be used, calculated accordingly to the maximum ratings of both the LiPo pack and ADC unit.

Battery Setup (pack means in series):
Remember, batteries connected in parallel add in mAh (for higher life and current output) while batteries connected in series add in voltage (for high voltage applications). To increase battery life, get batteries of higher mAh, and/or connect more of the same battery type in parallel.

Voltage per Pack	8	volts
mAh per Pack	1000	mAh
Packs in Parallel	1	packs

Remember to check the **robot parts list** to buy your batteries.

Calculated Results:
Typical battery life states how long your robot will probably run before your battery dies.

Total Continuous Power Draw	12.0	watts
Min Required Battery Voltage	6.50	volts
Idle Motors Battery Life	1.67	hours
Minimum Battery Life	0.556	hours
Typical Battery Life	0.667	hours

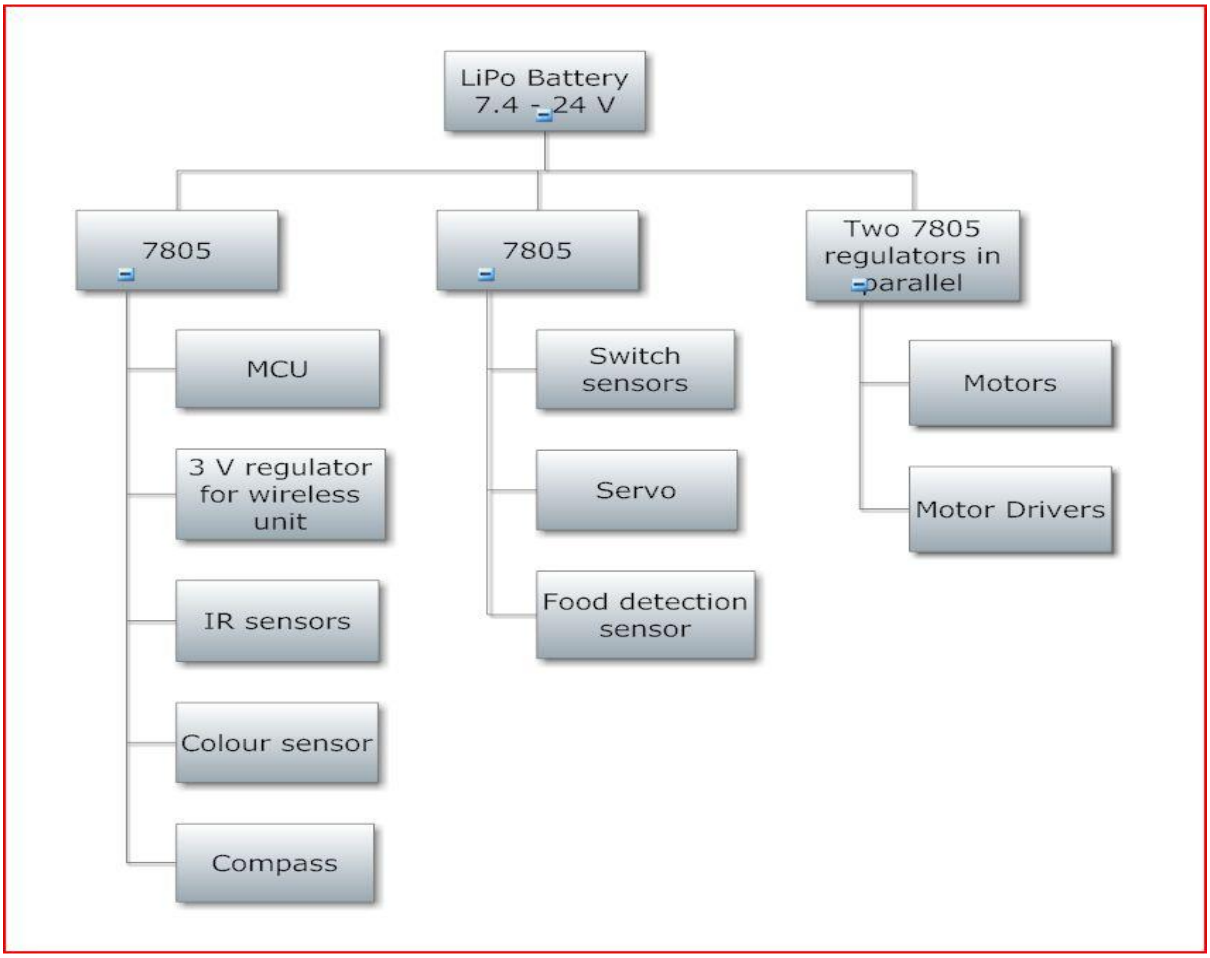
(Society of Robots, 2008)

6.1.14 Power Management

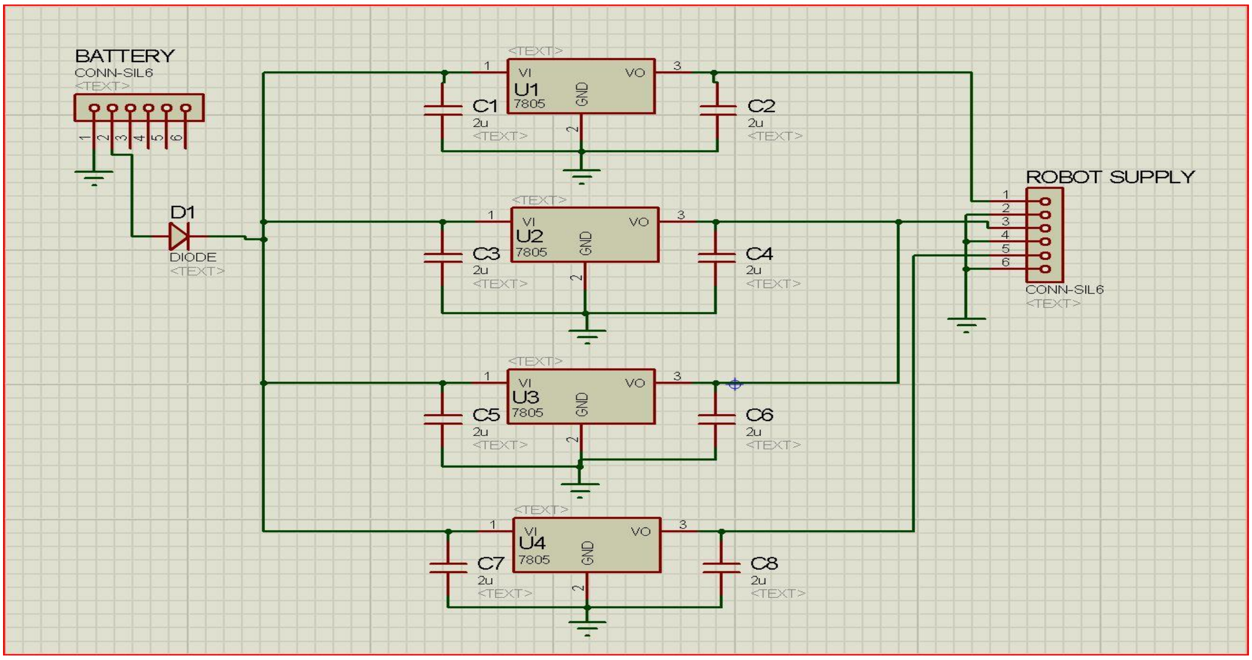
The whole system voltage is regulated at 5 V, using LM7805 linear regulators for each modules. In order to provide a sufficient reserve of current for the motors two units are used in parallel for motor control/ supply. If specific modules (like the wireless one) require a lower/ higher voltage, local, individual regulators are used. The regulators incorporate build in thermal overload protection, short circuit protection and offer good ripple rejection - 78 db.

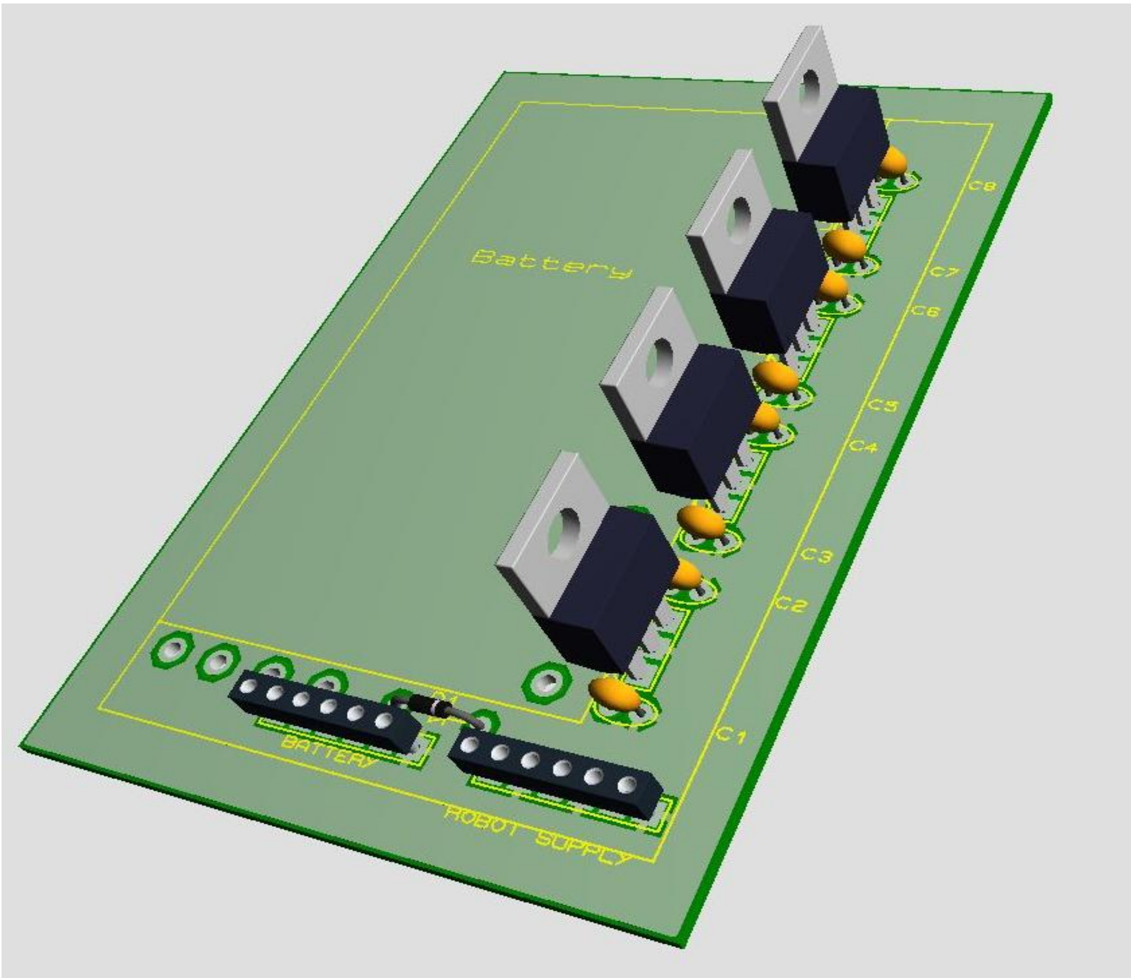
An important aspect considered was the regulator voltage drop. Because the absolute minimum voltage for each LiPo cell is rated around 3V and a 2 cell pack is used, an automatic battery fail-safe is built. When the voltage drops under 6.5 V the 7805 regulators are not supplied with enough voltage and this will lead to a general system stop.

The diagram shows the final power supply layout for each robotic unit.

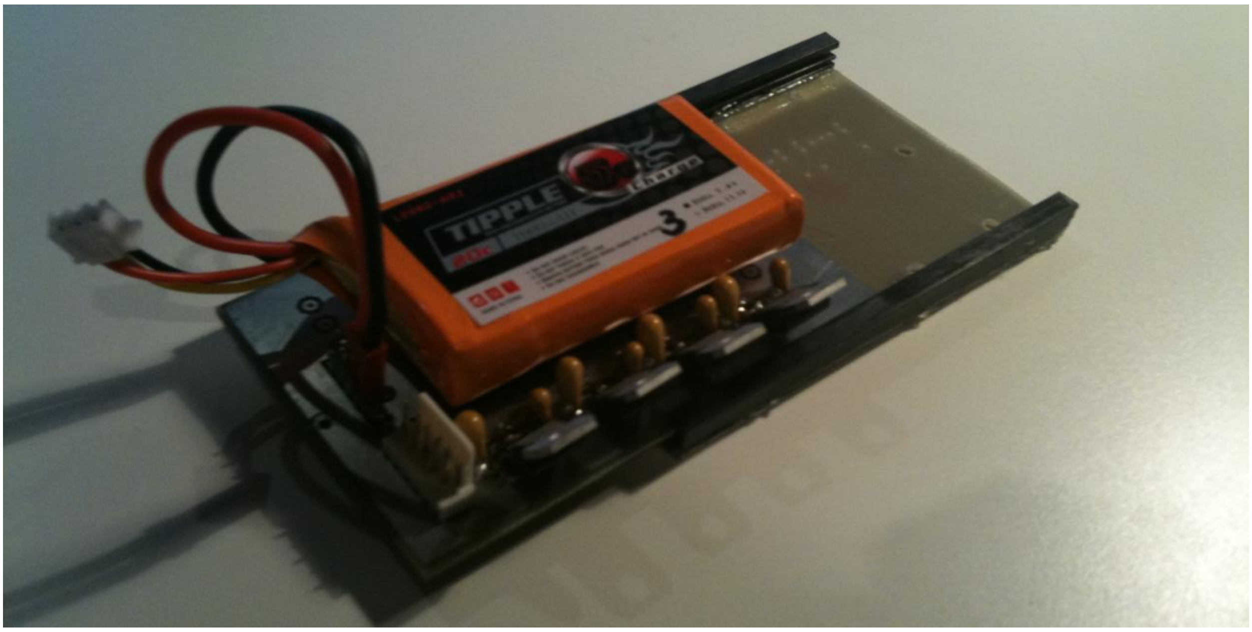


Power card schematic and 3D model:

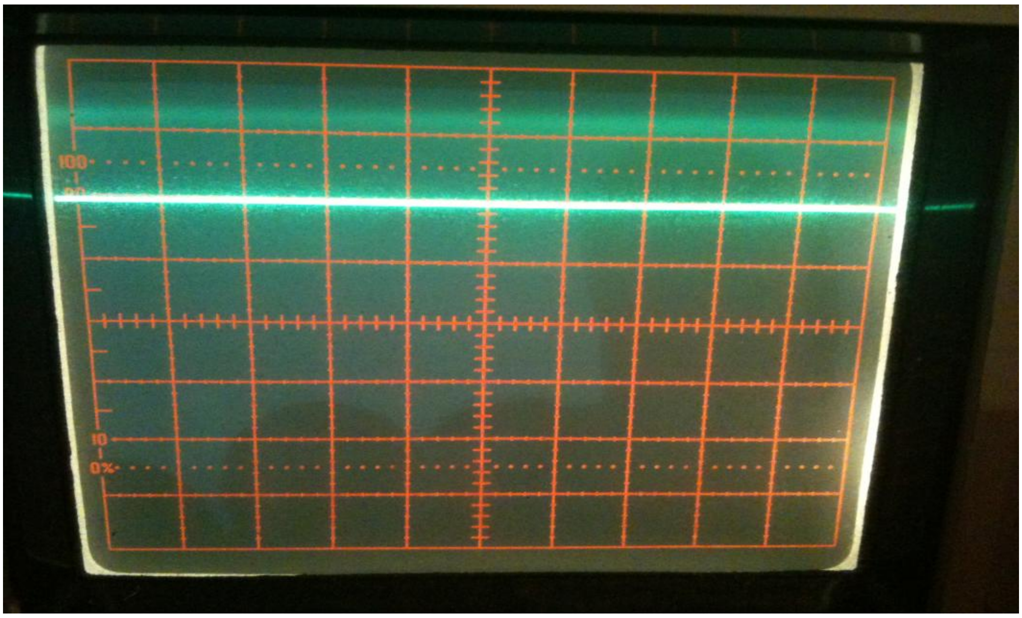




Voltage measurements proved that two 7805 units in parallel provided enough current for the motors and drivers. No voltage drop was recorded and 5.0 V were available throughout the system. The power card was tested on a variable DC source, from 7 to 24 V and provided a stable 5 V output (without any observable noise), offering the option to run the system on any available LiPo battery pack, as long as it would physically fit in the designated space. The use of a large ground plane on the power card provides additional screening between the microcontroller and motors, improving noise immunity.



DC voltage rails observed on the oscilloscope to check the noise level:

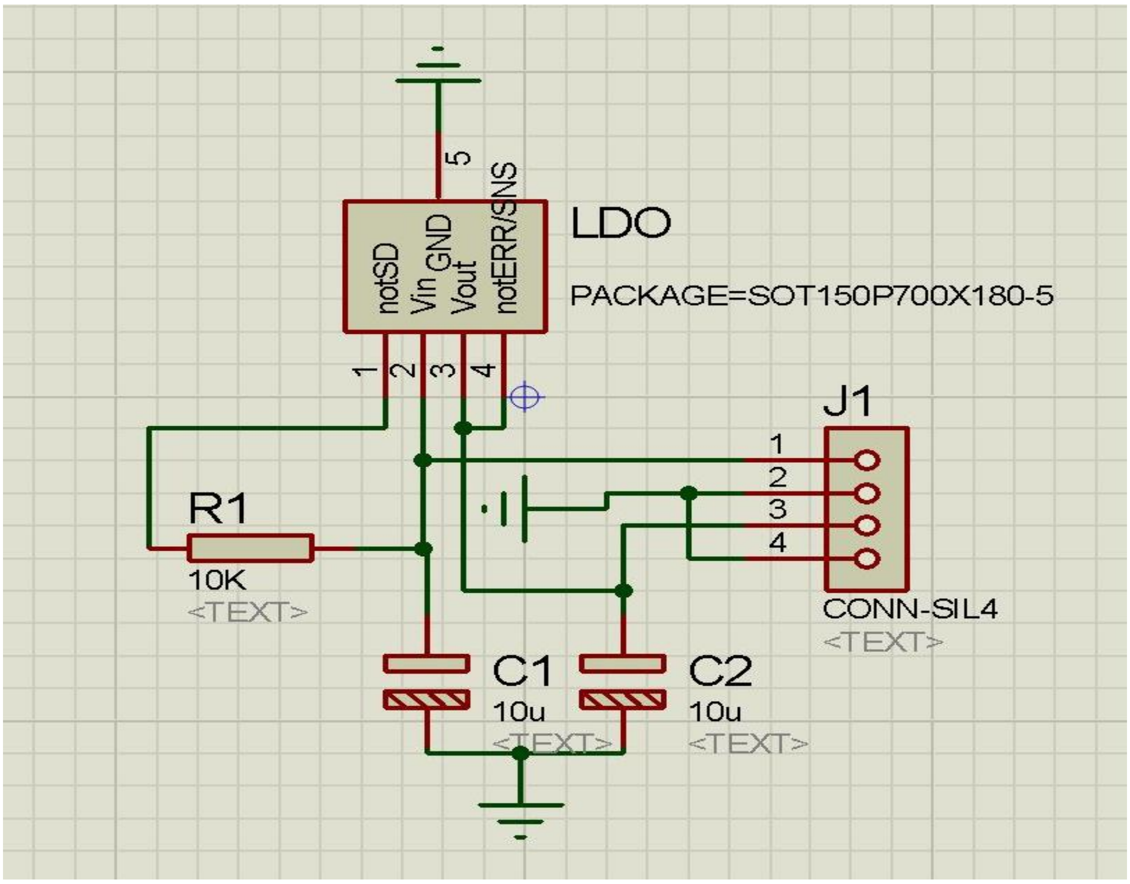


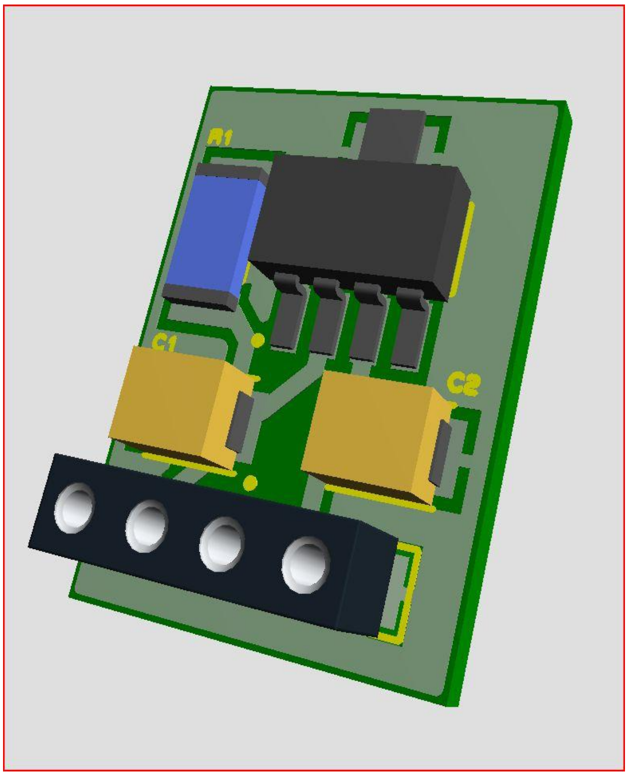
6.1.15 Wireless unit

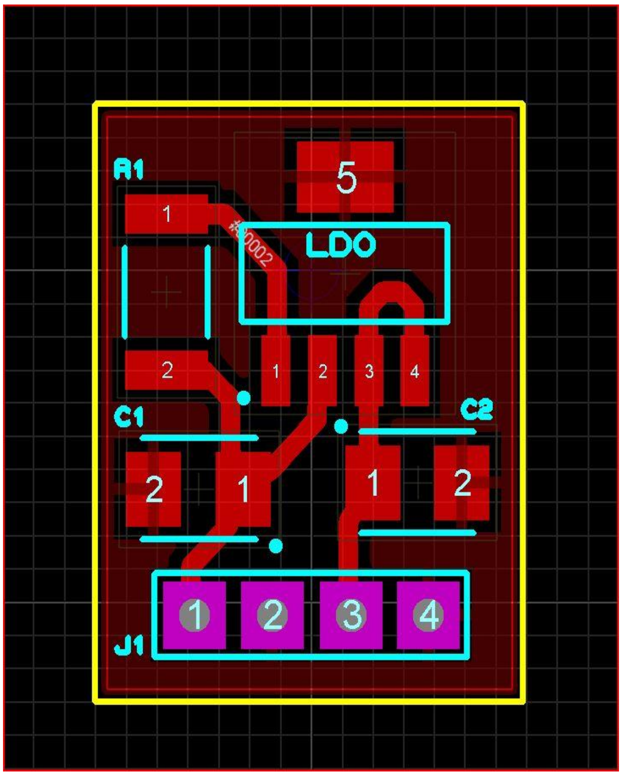
In order to provide communication between units, a long range wireless unit was chosen on the detriment of close range IR communication. There were no specified constrains for the selection of this unit, except the price range as the price per module had to be under £20.

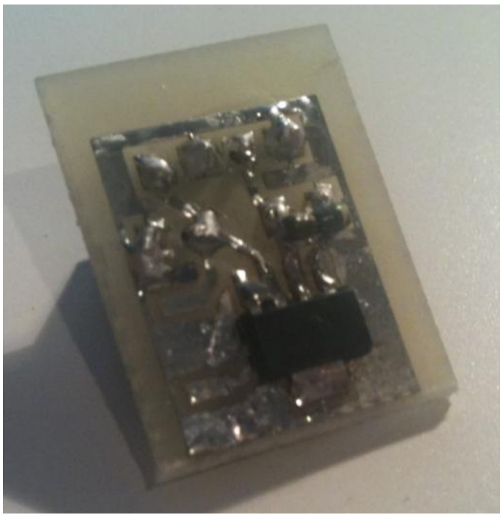
The first option, 2.4GHz DSSS, was a complete radio-on-chip from Cypress Semiconductor. This unit offered very good performance and features for the price. Another option was a transparent data transmission system on 433 MHz from HopeRF. This module was a classic wireless transmission system but had longer distance and a self controlled transmission protocol. Both units were available and were tested.

An important aspect regarding the wireless part of the system was the power required. All the available options worked on voltages up to 3.6 V and this was not readily available within the power rails. A local regulated 3.3 V source was designed. It features a low dropout voltage regulator, L3872 from National Semiconductor, and the afferent connectors.



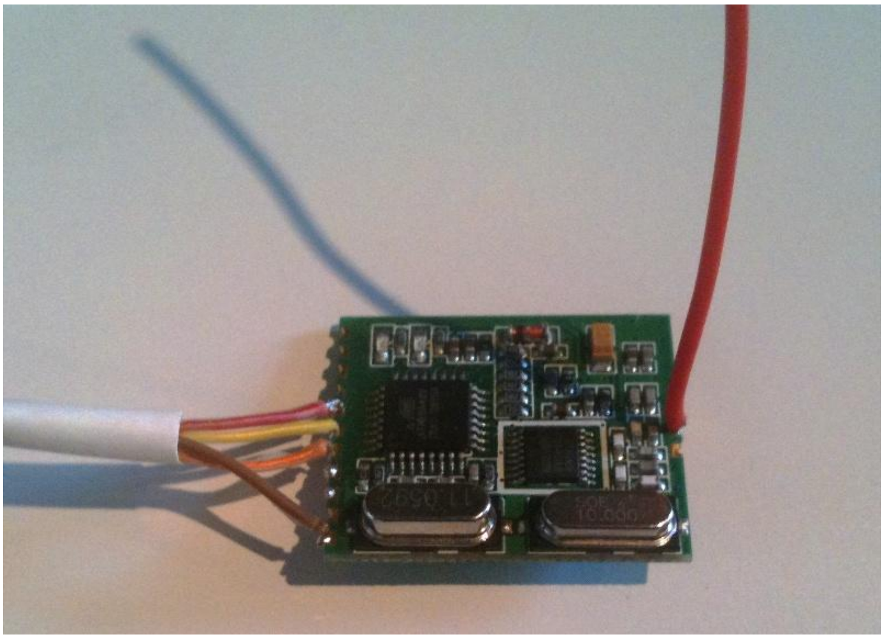






Wireless unit software requirements are greatly reduced by the transparent data transmission module, requiring only a USART module, serial communication interface.

```
void TxZeroTerminatedRamString( BYTE *pbStrPtr ){
while( *pbStrPtr != 0 ){                                   // check for the end condition, before sending the next byte
TX8_SendData( *pbStrPtr );                                 // send the next byte
while( !( TX8_bReadTxStatus() & TX8_TX_BUFFER_EMPTY ) ); // Wait for the data to start transmitting
pbStrPtr++;
}
```



Packet structure: real time communication is available, with 16 bytes of data transmitted and capable of transmitting up to 32 bytes of information. The packet of 16 bytes is made by a header of 3 bytes, consisting of the sending robot ID, the main body, 10 bytes of sensor information, commands and other user-specified data, with the last 3 of the total 16 bytes reserved for emergency signals.

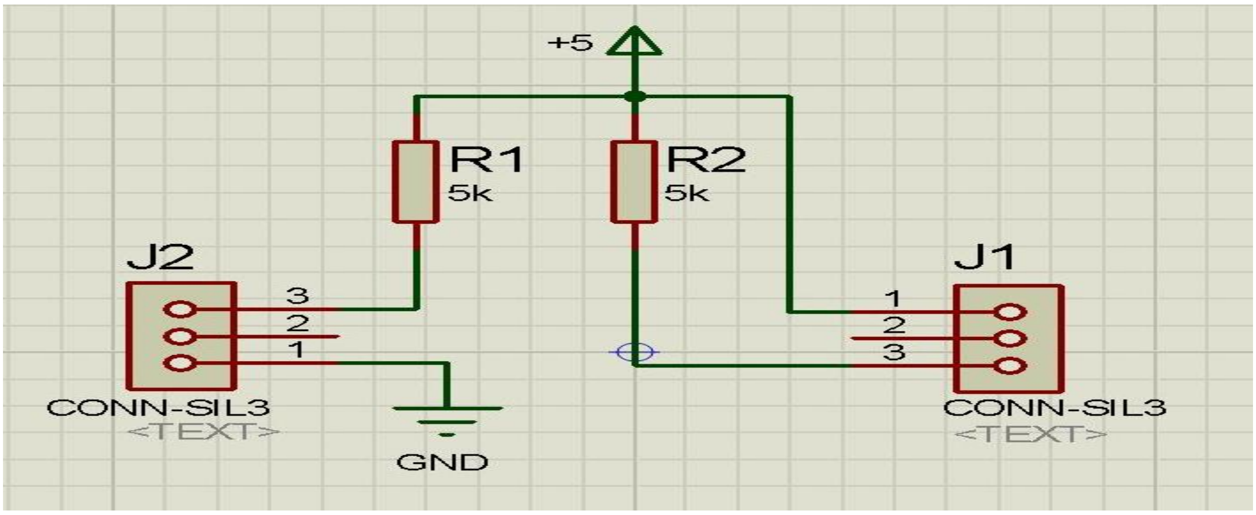
Noise immunity tests were done using the microcontroller at full speed, 24 MHz, and the PWM driven motors. The wireless system proved very reliable, maintaining excellent data integrity.

6.1.16 Electronic Compass

For better environment navigation an electronic compass is implemented. Due to the high cost of this unit a single one is used, but the sensor readings can be passed around the units and eventually each unit can be equipped with one.

The Honeywell HMC6352 was chosen. This compass offers 2-axis magneto-resistive sensors with the afferent analogue/ digital hardware and algorithms required for heading computation, including internal calibration algorithms, stray magnetic field protection and temperature compensation.

The communication protocol required is I2C, with a maximum frequency of 100Kbps.
I2C protocol posed a few problems because the compass IC address is stated wrong in the datasheet and instead of 0x42, 0x21 must be used for communication.




```
BYTE rxBuf[8]={0};
Timer8_Start();
I2Cm_fSendStart(0x21,I2Cm_WRITE);
I2Cm_fWrite(0x41);
I2Cm_SendStop();
Timer8_Start();//timer used to assure reliability - the software works without it
I2Cm_Start();
I2Cm_fSendStart(0x21,I2Cm_READ);
for(i = 0; i < 6; i++) {
rxBuf[i] = I2Cm_bRead(I2Cm_ACKslave); //Read first 6 bytes, and ACK the slave
}
rxBuf[7] = I2Cm_bRead(I2Cm_NAKslave); //Read data byte and NAK the slave to signify end of read.
I2Cm_SendStop();
```

6.2 – Software:

The software was written using the C language. This was the best option because there was not enough time available for a lower level language, like Assembly, and the software complexity was too large and implied too many elements for Assembly to be a viable option. Using Assembly might have led to a more efficient code but it would have made the whole design very hard to follow and develop in the future.

As a software development suite, the PSoC Designer 5.1 is used. This is a free IDE available for download from Cypress Semiconductor’s website. It uses a C compiler from IMAGECRAFT and offers extensive libraries for most of the modules available. The specific functions for each module are described in greater detail in the specific module subchapter. The global settings of the PSoC were maintained the same throughout the whole development process in order to avoid additional problems generated by different option sets.

From the initial specifications what each unit had to be able to do is listed below:

- Autonomously navigate the environment: from the software perspective, each unit had to make decisions based on the reading from the IR obstacle avoidance sensors and the tactile back-up pair.
- Collect different sensor data and share between units: different data from the colour sensor, battery level compass heading had to be shared amongst all the units, making the data available for the whole system.

Additional implemented functionality:

- execute predefined tasks:
    - “Food” hunt and transport: find a certain object regarded as a food source, capture and transport it to a predetermined location.
    - Search for energy sources: find a strong light source (if solar power would be implemented in the future) or a wireless energy charging point (a wireless system was developed but the available time did not allow for a complete implementation)
- algorithm dynamic task allocation: because the information was available to all the units, if one would complete a task that was not initially assigned to it, it

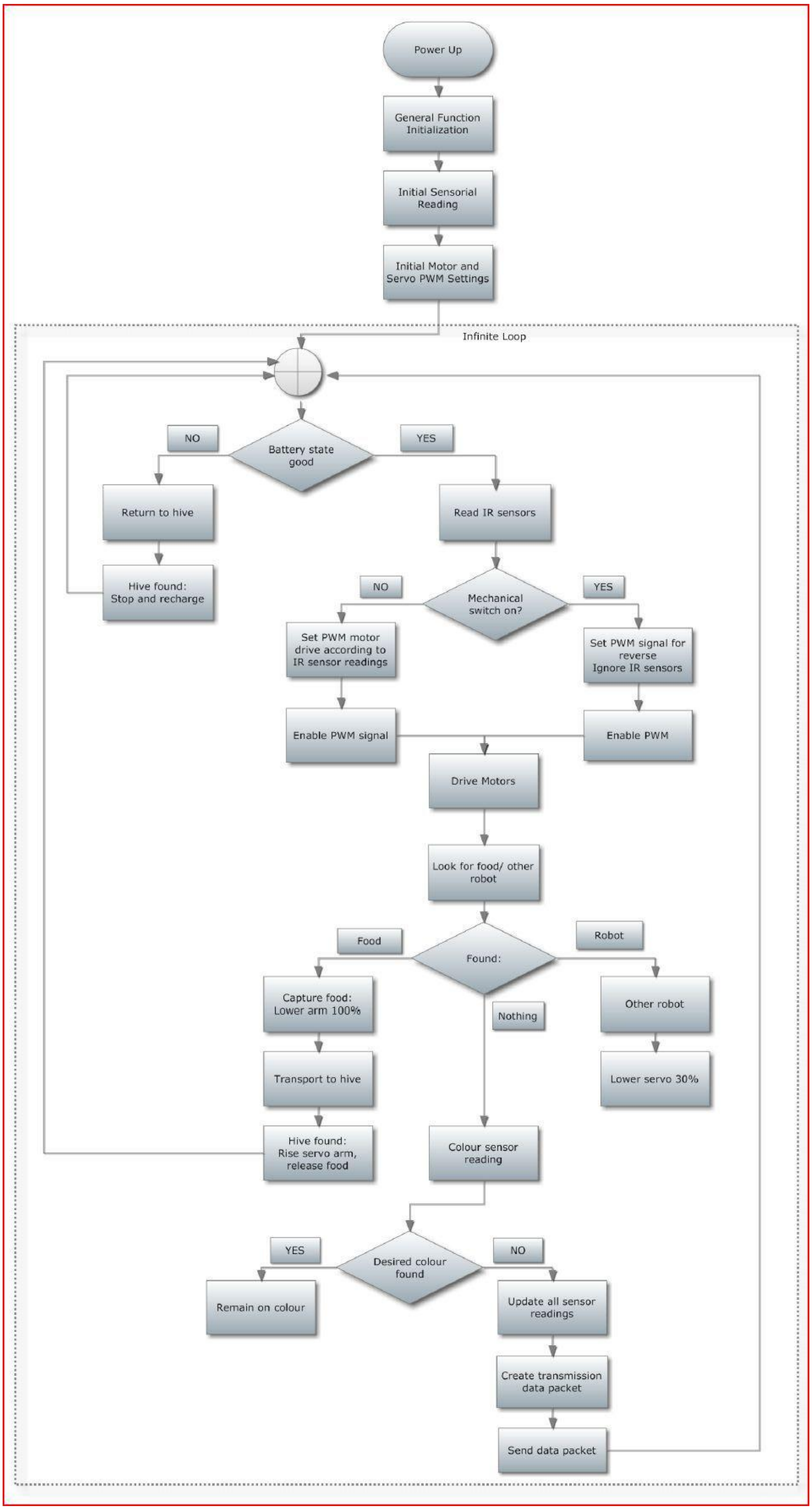
would be able to pass its own task over to other units and take over the task it was about to complete.

- robot recognition system: an array of specially developed sensors would facilitate each robot to detect and recognise food sources or other units of the swarm.

The same modular approach is used in the software development. All the hardware modules were developed with a separate software project for each including specific settings, the functions used and the connections that must be made. This approach offers great versatility and reconfiguration or upgrade capacity. In order to include a new module, the specific project must be consulted and then all the settings and software could be easily transferred to the desired robotic unit.

6.2.1 Basic behaviour

The diagram shows a basic behaviour implementation used to meet the project specifications. This can be further extended with other available modules.



7. Project Management

For the current project a PRINCE2 methodology of management was employed. The seven principles and themes proposed by the PRINCE2 were specifically tailored to suit the research project under discussion and account for the fact that it is an individual research.

PRINCE2 – Seven Principles/ Themes	Adapted principles to "Swarm robotics"
Business justification / Business case	Relevancy of research project
Learn from experience / Organization	Thorough initial research on the subject
Define roles and responsibility / Quality	Work breakdown structure - WBS
Manage by stages / Plans	Time plan with WBS
Manage by exception / Risks	Risk management
Focus on products / Change	Adapt the product to available resources
Tailor to suit project environment / Progress	Limit the overall system complexity without losing functionality

A thorough log-book was kept and consulted for project review and control purposes, and in case something had gone wrong. An updated Gantt chart is shown in the next picture. The first Gantt chart was modified to account for different changes that occurred throughout the project.

Milestones:

- Chassis
- Drive-train and motor drivers
- Obstacle avoidance sensors
- Colour detection
- Wireless communication

Number	Task	Start	End	Duration	% Complete
1	Initial Research	4/10/2010	29/10/2010	20	100.0
2	Chassis design, sensor selection	1/11/2010	5/11/2010	5	100.0
3	Microcontroller selection. Development board design	4/10/2010	10/11/2010	28	100.0
4	Obstacle sensor selection, PCB design and testing	24/11/2010	7/12/2010	10	100.0
5	Drivetrain and wheel selection	3/10/2010	11/10/2010	6	100.0
6	Chassis design, build and test	17/11/2010	1/12/2010	11	100.0
7	WiFi selection, protocol study, test different options	1/12/2010	2/2/2011	46	100.0
8	Motor control with sensor input	2/12/2010	14/12/2010	9	100.0
9	Compass communication	2/12/2010	18/12/2010	12	100.0
10	Obstacle avoidance and motor control. Autonomous behaviour	20/12/2010	15/1/2011	20	100.0
11	Wireless communication and sensorial data exchange	15/1/2011	5/3/2011	35	100.0
12	Colour sensor design and test	6/3/2011	12/3/2011	5	100.0
13	Robot recognition sensor	12/3/2011	18/3/2011	5	100.0
14	Light detection sensor	20/3/2011	22/3/2011	2	100.0
15	Battery monitoring implementation	24/3/2011	26/3/2011	2	100.0
16	Project report and documentation	4/4/2011	22/4/2011	15	80.0
17	Logbook	4/10/2010	22/5/2011	165	100.0
18	Finalizing behaviour for demonstration	8/4/2011	1/6/2011	39	60.0
19	Preparation for poster presentation	28/5/2011	3/6/2011	5	10.0

8. Budget and Bill of Materials

The column "Relevance to project" refers to the exact chosen component and the relationship with other equivalent module.

Quantity (Units)	Price (£)	Component name	Relevance to project 1=low 10=high
3	1.6	Chassis	8
-	6	Additional hardware (mechanical)	5
3	10	Motor and gearbox package	8
3	1.4	Motor driver	9
3	4	Microcontroller	10
6	1.5	Optical sensors (obstacle avoidance)	6
3	2	Colour sensor (terrain colour recognition)	6
3	4	Colour sensor (robot recognition)	5
3	18	Wireless unit	9
3	5.5	Battery pack	10
-	6	Additional electronic parts (connectors, wires, sockets)	7

Total cost/ unit: £60 ex VAT. This is the price without any manufacturing and PCB costs included, as they were done using available university facilities. Price can be lower for a large order of components, especially for the wireless and drive-train units, as these are the most expensive.

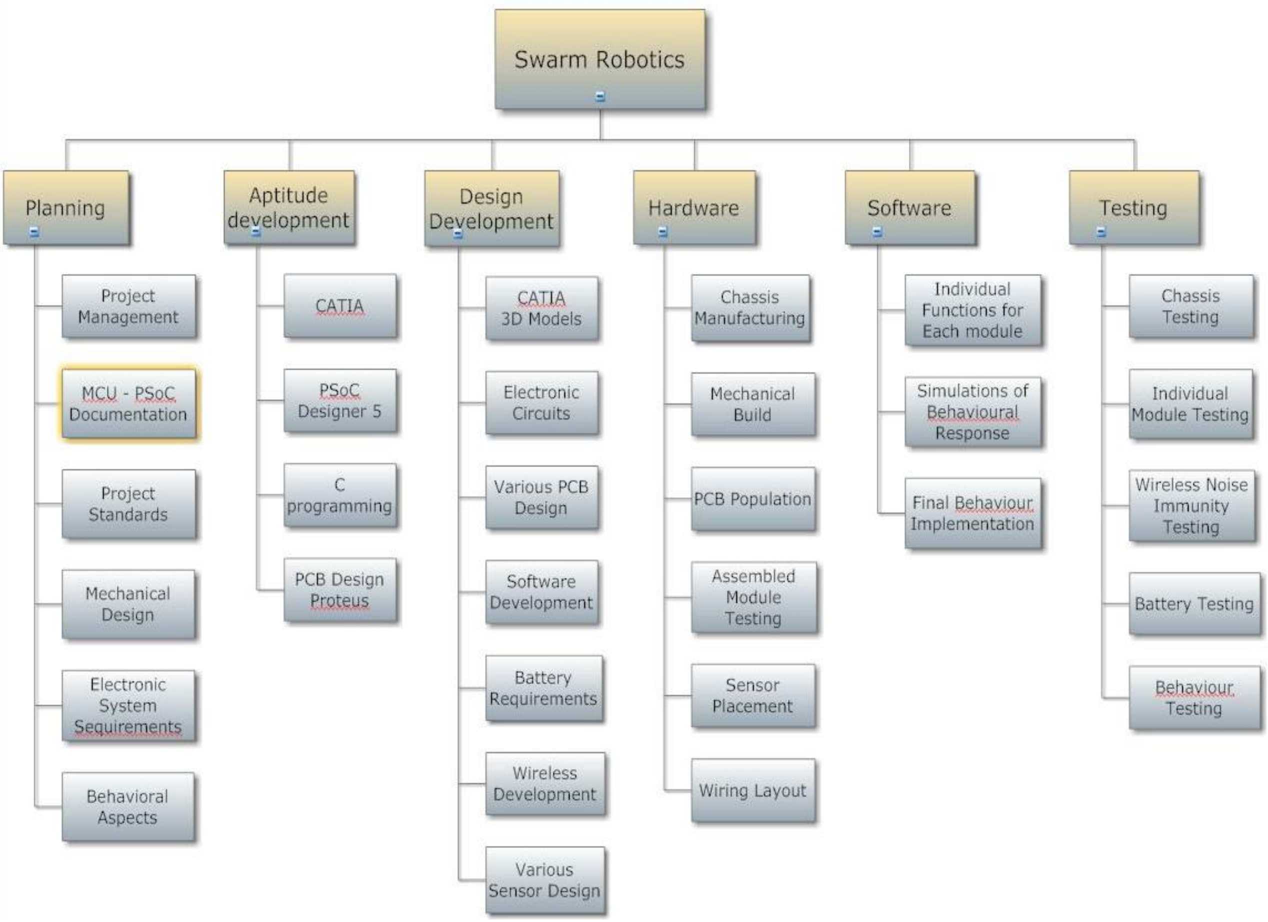
9. Project Schedule

In order to have a clear general picture of the tasks involved and to be able to keep a good record of the completed work and what still needs to be done, a detailed work breakdown structure was generated in the early stages of the project. This was part of the high level planning along with the Gantt chart and proved very useful for managing the project, as well as keeping on schedule. Overall the project was kept well on the initial time plan, although it had to be updated to consider the final report hand-in, demonstration and presentation date.

The work was structured in a concurrent way in order to be able to work on different parts of the project in the same day. This structure and the modular concept of the system proved very useful especially when certain modules were developed but were waiting for PCB manufacture or component delivery.

If any events occurred and certain tasks were delayed, a daily work breakdown structure (consisting of specific daily tasks) was adopted in the form of continuing on schedule as normal but allowing for at least one or two hours of work on the module that was delayed. If the delay happened on a module from the critical path or the development could not continue this became the main priority and all resources were redirected.

The diagram below presents how the work breakdown structure was organized.



10. Risk Management

Considering the general complexity of the project, a risk management strategy was required. Risk identification, quantification and control are presented in the table below. All risks were accepted and a strategy was created to deal with each individual issue if it appears.

Risk probability 1=lowest 100=highest	Risk	Avoidance strategy	Risk Impact on project
75	Data loss	All documents were kept on at least two computers and a memory stick and an automatic online backup was generated each day.	100
40	Components out of stock	The critical components were ordered as soon as the final decision was taken. Each component had at least one back-up unit/ replacement (depending on the price). Different suppliers for the same component were researched.	70
35	Injury risk during manufacturing/ assembly	Minimize using safety equipment and abide safety regulations.	50
45	Electronic problems i.e. circuits malfunction	Early testing and simulation was conducted prior to final design.	60
80	Projects delayed due to other deadlines	The initial time plan was refined and adapted during the project life time. A flexible structure was accounted for since the start.	65
55	Illness/ inability to work	Minimize risk taking necessary precautions and consulting a doctor if the illness was serious.	90
70	Interference due to system complexity	Careful design and relevant precautions were taken.	50
95	Financial issues preventing certain hardware acquisitions	The hardware was chosen considering the price/ functionality ratio and at least one cheaper option was available in all cases. A complete bill of materials was created and the final cost was increased by 20% - to account for possible extra components required.	40
20	Mechanical/ electronic premature failure	Thorough testing and suitable material choices minimize the risk. Highest possible quality components were chosen.	70
40	Inability to meet deadlines/ Report not finished on time	Keep on track with the time plan and allow a certain safety margin for deadline completion. All work finished at least two days before each deadline.	90
45	Supplier sending wrong components	Allowing time for part return. Emergency budget available for new purchase while the wrong parts are refunded.	40
25	Components have bugs/ manufacturer acknowledges reliability issues	Choosing reliable and tested components. Avoid beta versions/ new versions of parts and software.	40
60	Parts sent but not received, or damaged in transit	Consider the necessary delivery time and allowing a safety margin. Contact the supplier if something is delayed.	45
70	PCBs have mistakes, manufacturing problems etc.	Allow enough time for at least another set of boards to be redesigned and manufactured.	55
85	Overall project delay due to various factors	Realistic time plan. Good project tracking and control. Periodic analysis of the results.	60

All the high probability (above 70) risks were closely monitored and reviewed.

Considering the list above, the risks that actually had materialised were regarding component delivery, part acquisition and new tasks that prevented the project to follow the time plan for a short period, less than 10 days. The steps taken were described in the table and proved to be the right way of dealing with the risk situations.

11. Quality Management

BS 8888:2008 – technical product specification standards were considered for technical drawing. Additional hardware components like screws, washers and such were all chosen to be on the metric unit system. Considered standards were ISO 21.040 for screw threads and ISO 21.200 for gears. Also, for ease of assembly, all screws were Philips head type.

The EN 300 220-3 V2.1.2 in class 2, EN 301 489 V1.4.1 in class 1 regulation for EMC compliance was considered due to the wireless communication side of the project.

Regarding electronics components all the chosen components were RoHS compliant, and the standard considered was ISO 31.xxx, especially ISO 31.080 for semiconductor devices.

Extensive testing was carried out through all stages of the project. All aspects of the systems were tested using a large set of specifically designed test, in order to better simulate the future environment.

All the tests were carried in a controlled environment, the conditions were closely monitored and most of tests were repeated with exact methodology over a few days to ensure all measurements were reproducible and the errors are minimised. Initial estimations, assumptions and calculations were also checked during testing.

In order to debug the system and assure that all values are within desired range, an LCD display is the best tool available. All the observed values are displayed during the design and set-up phase. The LCD can be implemented in the final unit but it was not desired in the current stage of the project.

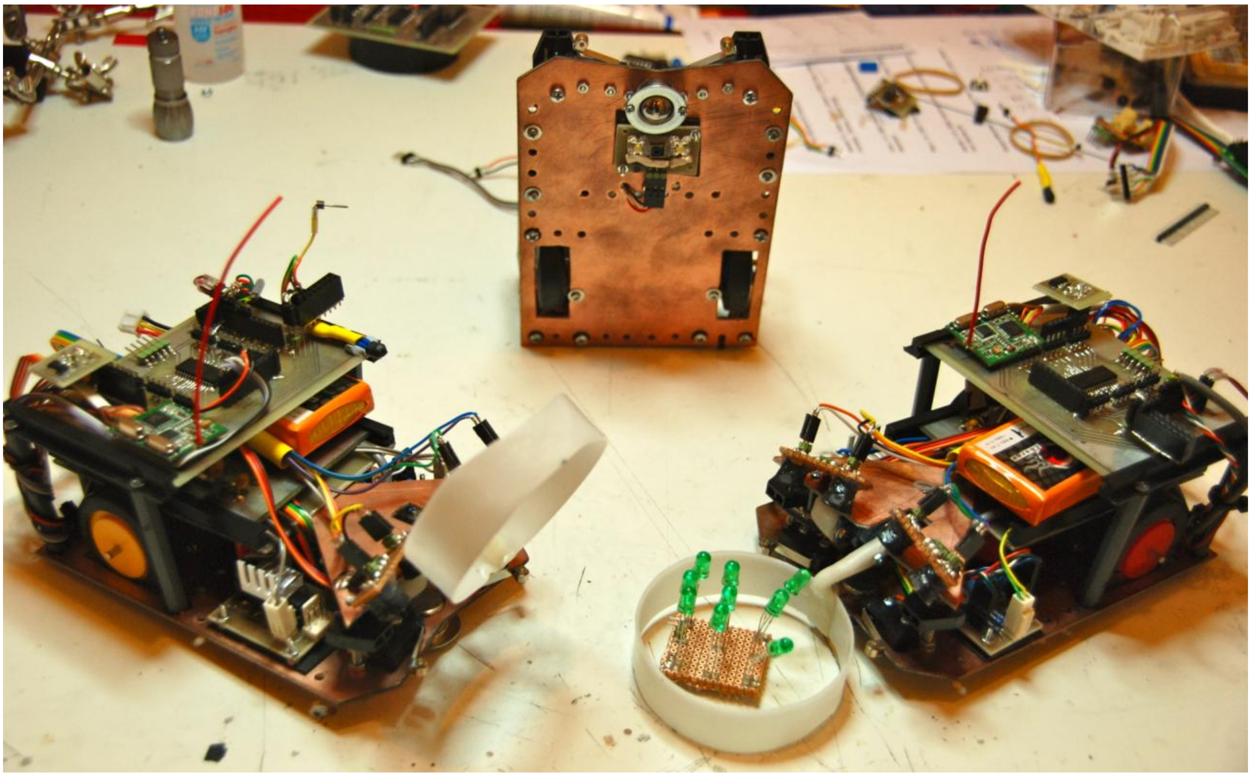
- Mechanical strength tests comprised of different loading conditions of the chassis and gearbox assembly and the finished product as well.
- Endurance/ reliability tests were done in order to assure that the system is capable to attain a running time limited only by the battery life and no other system parameters.
- Temperature testing was done to prevent components from overheating, as this would lead to premature failure.
- Running time – battery life was measured under different loading conditions.
- Software reliability testing was one of the most important parts of the testing phase. The software was checked to perform all the intended functions and offer reliable operation without any endless looping/ system latch-up.
- Noise rejection + transmission distance were done mostly to check if the wireless system was performing up to the intended task.

Results of the tests ran on each module are available with detailed descriptions under the specific module's section of this report.

12.Conclusions

12.1 Critical appraisal and achievements

The task was not easy, but it was done with a lot of time invested and all specified project objectives were met. The robots were able to navigate the environment and avoid obstacles as well as detect “food” sources, capture and transport them to a predefined location. In parallel with these tasks, wireless communication was available, the robots being able to transmit sensorial readings and other information as desired.



The behaviour must be further researched as this was limited due to the time available. However, the versatility of the research platform offers almost unlimited possibilities for behavioural research. Simple neuronal controllers (feed-forward single layer type, with a few input-output neurones) can be implemented successfully with the available processing power of the 8 bit PSoC.

The overall software complexity was kept as small as possible considering how complex the system and of the project are. Keeping the software under control and paying attention to general coding structure and readability also makes the system easier to debug and develop throughout longer periods of time.

Integrating all the modules on a single PCB and replacing most of the elements with SMD equivalents would greatly reduce the size of the unit. Also, some of the cables are worth integrating on the main chassis (if the FR4 material is kept as platform) to reduce the required number of cables. This type of integration is worth implementing for other platforms as well, especially aerial units or smaller designs where space-saving is justified, but for the current research it was not an objective.

Some of the encountered problems worth mentioning are listed below:

- Wireless communication: developed the 2.4GHz link, but in the end it was too complicated from a software perspective. The initial wireless software had around 150 lines of code and required much larger computational resources than the second option – Hoper RF unit, which is the one available on the final product.

- Avoidance sensors: very hard to find the best unit in one package that offered close-range distance measurement as well and not be influenced by external light.

- Motor driver heating up: added heat-sink although the IC provided over-temperature protection.

- PCBs made wrong: because of the SMD technology, the wireless module's 3 V power supply PCB, was mirrored during manufacturing, but it was still usable with a few adjustments and removed traces.

- Compass address: the actual IC address is 0x21 not 0x42 and this meant around fifteen hours of work investigating what was wrong and why the communication was not working. This was one of the biggest problems encountered because the communication had been tested with other peripherals and between two microcontrollers but it was not working on the compass IC.

12.2 Future Work

The following section describes some of the many possible improvements and further implementations that could be added to the current research.

- Enlarge and develop the swarm, integrating different terrain units, aerial, submersible, off-road as well as different functionality elements: soldiers, scouts, workers, large swarm carrying units – transporters, and brains – for data analysis and interpretation.

- Develop software intelligence, employing different neuronal or pseudo-neuronal algorithms and provide the units with ability to evolve by progressively synthesizing better solutions for the problems faced. The evolutionary abilities come at the cost of longer computational times and the necessity for simulation prior to the implementation of the control algorithms.

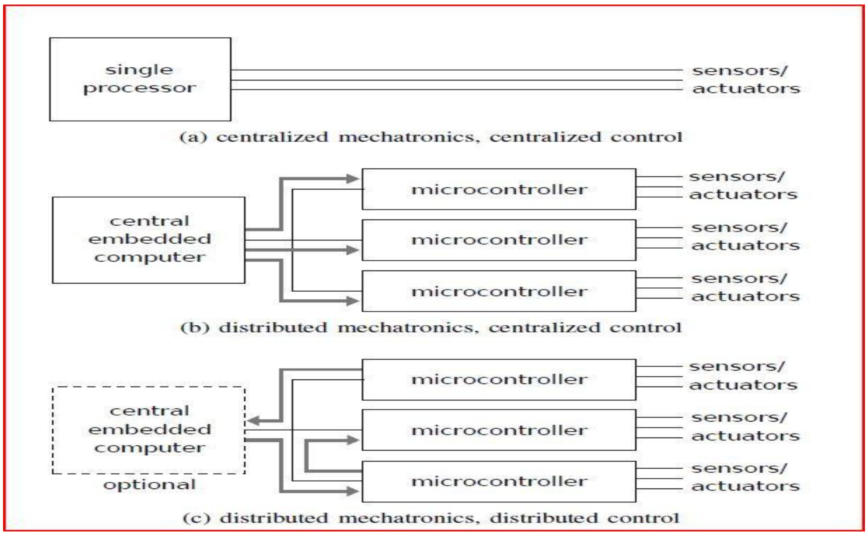
- Better task allocation, cooperative strategies, social behaviour, selective attention and pseudo-language must be researched and tested.

- Local centralization could be implemented, but the ability to detach from a group and join other group if needed, or if centralized controller is lost or damaged, must be maintained to keep the decentralization and flexibility of the swarm.

- 32bit microcontrollers to replace the 8bit currently used, because they are cheapest option available at the moment from a price – performance perspective. The current PSoC1 can be directly replaced with the PSoC5 – ARM core microcontroller, currently available for sampling.

- Important to keep the embedded concept of the project, especially in the case of search and rescue operation where communication infrastructure would be unavailable. Clearly a system based on internet connection or GSM communication would not be feasible.

- ASEBA system architecture: modified and adapted to fit the project best. This concept offers an implementation of modular architecture for event-based control of robot and it seems to be a good option for the management of a complex swarm. This type of modular architecture will prove as an ever better and cost-effective solution as multi-core microcontrollers become largely available. (EPFL 2010)



- Better distance sensors – sensorial array for navigation can be made from one centred ultrasonic sensor and two “Sharp IR Range Finder” modules on the sides of the chassis. These units can be fixed at a desired angle or better, mounted on small servo motors or pan and tilt units, in order to provide a 3D map of the environment.

- Incorporate small camera modules on units. The camera units can be wireless, for sending the images to other units, operator or optional processing units. Also, cameras can be used for image data logging on external memory card or even used in environment navigation via image processing.

- A smart robot recognition sensor must be developed, with dedicated microcontroller for fine colour recognition and better environmental sensing. If a dedicated processing unit is used, the four transducers can be transformed in a sensor capable of colour recognition and distance measurement as well, because of the way they are designed and placed.

- If long range is desired, power regeneration is possible due to motor control method – locked antiphase, but the motor driver must be replaced completely with a suitable unit.

- Solar power is a good option for low power requirements, like sensor reading and data logging. The rest of the system can be completely turned off once the unit has positioned itself in the desired place. Power is managed by the microcontroller and different sensors are employed, depending on the required data.

- Wireless charging station can be created for special cases such as underwater or hazardous environments. Considering current developments in wireless power transfer it will become a feasible and reliable solution, especially in the cases where contacts that might produce sparks or be subject to corrosion are not desired. The wireless power transfer can even be used amongst the swarm units, enabling them to charge other lower power units locally.

- Human control over a certain unit can also be implemented. Switchable control, allowing the user to select different units is a good option. User control could be implemented assigning a special user packet, that once received by the unit it would switch to slave mode and allow full control over the robot functions. Implementing this function can also be used to generate events within the swarm and research different behaviour models under the influence of external stimuli.

12.3 Final discussion

Throughout the current paper a detailed description on how to implement a swarm of robots was presented, from the initial specification stage, through design, implementation and a final demonstration where all functionality is tested and different behaviours are observed.

It was clearly demonstrated that the idea of swarm robotics can be researched using small budget and generally available resources. The “Divide & Conquer” strategy adopted proved a very good method for research and implementation, offering efficiency, parallelism and good overall control of the project. The system demonstrated good reliability and the fail-safe mechanisms integrated throughout proved to function properly and protect the sensitive parts of each robotic entity.

Overall the project was a success, promoted communication and collaborative working between robots and successfully implemented all the specified functionalities. Despite the fact that classical swarm behaviour is based on a low level of intelligence of each member, with the current progress of embedded technology and advancements of the microcontrollers as well as processing power, I believe that it is worth taking the swarm concept one step forward and implement higher and different intelligence levels on each member of the swarm. Together with a more complex communicating system new, better behaviours can arise or be constructed by the researcher. It is a concept with almost unlimited potential for further development and could eventually be employed in life-saving scenarios like environmental/ natural disasters, and maybe even the exploration of other planets.

13. Student Reflections

The “Swarm robotics” project was an immense learning experience for me as it was a very comprehensive and original project, which implied a very broad skill set for successful completion. As the project was based on individual learning I used the available literature and previous work in the field to understand the concepts and development.

Some of the major areas I consider that my skills were improved in are listed below:

- Mechanical: material testing and analysis – rig testing, FEA, material proprieties.
- SMD electronic components soldering skills: although I did not have much experience with surface mounted devices – SMD and soldering them, after a few tutorials everything went well and all components were successfully soldered.
- PCB design and packaging.
- C programming skills.
- Serial (UART, I2C) protocol communication – because I was not experienced in the serial protocols, a deep understanding was necessary for familiarisation with these communication protocols.
- Wireless protocol knowledge – it was the first time a complete wireless system was designed by me so without previous experience a good understanding of the concepts involved by this type of communication was necessary.
- Awareness of standards that must be consulted.
- Project management - risk assessment and good project planning.
- Soft skills – describing my project to professionals and requesting samples, as well as attending events and seminars related to my area of research.

From my point of view this project is feasible and one day large swarms will be successfully employed for multiple coordinated tasks: exploration and search and rescue operations, local data logging and wireless transmission of the logs to CPU, perimeter marking, surveillance etc.

For outdoor data logging and perimeter surveillance the units can be equipped with solar power supply, preserving battery life for extended usage, while offering the advantage of virtually any placement option, depending only on the chosen chassis and size. This is just a simple application for the concept, but with adequate research complicated tasks will be achievable and made simple trough cooperative robotic work. They can lead to the solving of many issues in environments where human activity is restricted, for example in the recent situation in Japan, where radiations didn’t allow direct operation on the systems at the nuclear power plant.

Bibliography and References

Banzhaf, W. (1998) *Genetic programming :an introduction on the automatic evolution of computer programs and its applications*. San Francisco: Morgan Kaufmann Publishers

Bekey, G., A. (2005) *Autonomous robots: from biological inspiration to implementation and control. Cambridge*, MA: MIT Press

Bekey, G. A., Goldberg, Y. (1993) *Neural networks in robotics*. London: Kluwer Academic

Bonabeau, E. (1999) *Swarm intelligence: from natural to artificial systems*. New York: Oxford University Press

Brander, Th., Gerfer, A., Rall, B., Zenker, H. (2009) *Trilogy of Magnetics.* 4th edn. Kunzeslau, Germany: Swiridoff Verlag

Cozzens, R. (2009) *CATIA V5 workbook: release 19.* Mission, KS: Schroff Development

Eckel, B. (2000) *Thinking in C++.Vol. 1, Introduction to standard C++*. London: Prentice Hall

Egan, W. (2003) *Practical RF system design*. Chichester: Wiley

EPFL - Ecole Politechnique Federale De Lausanne (n. d.) *ASEBA - An event-based architecture for distributed control of mobile robots* [online]
available from: <http://mobots.epfl.ch/aseba.php>
[12 April 2011]

Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle - Université Libre de Bruxelles (n. d.) *Swarm-bots* [online]
available from: <http://www.swarm-bots.org/index.php?main=1&sub=11>
[20 April 2011]

Hrynkiw, D. (2002) ed. by Hrynkiw, D. and Tilden. W., M. *Junkbots, bugbots,and bots on wheels: building simple robots with BEAM technology*. London: McGraw-Hill/Osborne

Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle - Université Libre de Bruxelles (n. d.) *Swarmoid [*online]
available from: <http://www.swarmanoid.org/project_description.php>
[15 April 2011]

Jeff Gough (n. d.) *Formica: Affordable, open source swarm robotics* [online]
available from: <http://warrantyvoidifremoved.com/formica>
[13.01.2011]

John, Y., Reza, L., Lotfi, A., and Zadeh (1995) *Industrial applications of fuzzy logic and intelligent systems.* New York: IEEE Press

Jones, J. L., Flynn, A. M. (1993) *Mobile robot: inspiration to implementation*. Wellesley, Massachusetts: A K Peters

Johnson, H., W. (1993) *High speed digital design: a handbook of black magic.* London: Prentice Hall

Kortenkamp, D. et all (1998) *Artificial intelligence and mobile robots: case studies of successful robot systems*. London: AAAI Press, MIT Press

Kruse, R. (1994) *Foundations of fuzzy systems*. Chichester: Wiley

Main, M. (2010) *Data structures & other objects using C++.* Boston: Addison-Wesley

Meyers, S. (1998) *Effective C++: 50 specific ways to improve your programs and designs*. Harlow: Addison-Wesley

Miller, P. (n. d.) *The Genius of Swarms* [online]
available from: <http://ngm.nationalgeographic.com/2007/07/swarms/miller-text>
[05.01.2011]

MIT (n. d.) *MIT Project: ORCA* [online]
available from: <http://web.mit.edu/orca/www/2004_motorController.shtml>
[18 February 2011]

MIT (n. d.) *The Ants: A Community of Microrobots* [online]
available from: <http://http://www.ai.mit.edu/projects/ants/>
[11.01.2011]

Nehmzow, U. (2000) *Mobile robotics: a practical introduction*. London: Springer

Ollero. A. (2007) *Multiple heterogeneous unmanned aerial vehicles.* Berlin: Springer

Peter Miller (n. d.) *The Genius of Swarms* [online]
available from: <http://ngm.nationalgeographic.com/2007/07/swarms/miller-text>
[05.01.2011]

PHYSORG (n. d.) *Robotic ants building homes on Mars?* [online]
available from: <http://http://www.physorg.com/news143815934.html>
[08.01.2011]

Pololu Robotics and Electronics (n. d.) *Tamiya Gearbox* [online]
available from: <http://www.pololu.com/catalog/product/114>
[22 January 2011]

Ross, J. N. (1997) *The essence of power electronics*. London: Prentice Hall

Ross, T. J. (2004) *Fuzzy logic with engineering applications*. Chichester: Wiley

Siciliano, B. (2008) *Springer handbook of robotics.* Berlin: Springer

Smith, P. (2001) *Active sensors for local planning in mobile robotics*. London: World Scientific

Society of Robots (2008) *Battery Calculator* [online]
available from: <http://www.societyofrobots.com/battery_calculator.shtml>
[12 December 2010]

Society of Robots (2008) *Energy Calculator* [online]
available from: <http://www.societyofrobots.com/energy_calculator.shtml>
[12 December 2010]

Trianni, V. (2008) Evolutionary Swarm Robotics. Berlin: Springer

Universitat Karlsruhe (n. d.) *I-SWARM* [online]
available from: <http://www.i-swarm.org/MainPage/Project/P_Partners1.htm>
[10 December 2010]

Weiss, G. (1999) *Multiagent systems: a modern approach to distributed artificial intelligence*. London: MIT Press

Zalzala, A.M.S., Morris, A.S. (1996) *Neural networks for robotic control: theory and applications*. London: Ellis Horwood

Zamani, N., G. (2007) CATIA *V5 FEA tutorials: release 16.* Mission, KS: SDC Publications

Appendix A – Project Specification

Appendix B – Interim Progress Report

Appendix C – Software