

**POSI – Medida 1.3
“Investigação em Consórcio”**

ANEXO 1

**(Relatório de Execução Material)
[RAPOSA]**

(With a CD-ROM)



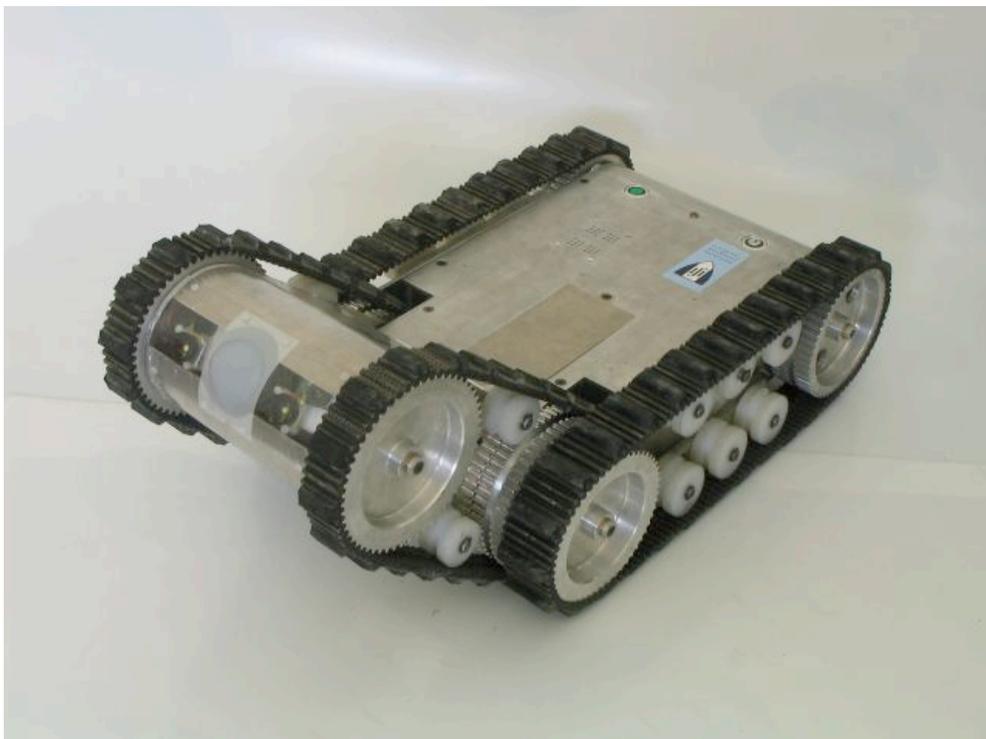
Financiamento no âmbito do Programa Operacional
Sociedade de Informação (POSI) participado pelo
FEDER e por Fundos Nacionais



RAPOSA

Final Report

(01 / 03 / 2003 – 31 / 03 / 2005)



Relatório elaborado por:

João Cristóvão, Carlos Marques
IdMind - Engenharia de Sistemas, Lda

João Frazão, Pedro Lima, Maria Isabel Ribeiro
Instituto Superior Técnico

Table of Contents

1. Summary	7
2. Gantt Diagram	9
3. Summary of the Results per Task	11
T1 – Electronic Hardware Architecture	11
T2 – Software Architecture	11
T3 – Mechanical Structure Project	12
T4 – Equipment Characterization	13
T5 – Software Interface Development	13
T6 – Mechanical Structure Assembly	14
T7 – Electronic Equipment Assembly	14
T8 – Software Modules Integration	15
T9 – Tests on the Lisbon Firefighters Headquarters	15
4. Project Feedback at the Company	17
5. Dissemination Activities	19
6. Task T1 – Electronic Hardware Architecture	21
6.1. Hardware Requirements	21
6.2 Hardware Architecture	21
6.3 High-level Processing	24
6.4 Wireless Communication	25
6.5. Cameras	26
6.5.1. Standard Cameras	26
6.5.2. Thermal Camera	27
6.6 Low-level electronics	28
6.6.1. IC Microcontrollers	28
6.6.2. Linear Accelerometer as an inclinometer	28
6.6.3. Infrared distance measuring sensors	30
6.6.4. Gas Detectors	30
6.6.5. Temperature and Humidity Sensors	30
6.6.6. Servo Motors	31
6.6.7. Lights	31
6.7 Motors and Controllers	31
7. Task T2 – Software Architecture	32
7.1 Introduction	32
7.2 High-Level Software Issues	33
7.3 Detailed Software Architecture	34
7.4 Video Stream	35
7.4.1 Video Bandwidth and Wireless Bandwidth	35
7.4.2 Image Compression	36
7.4.3 Video Compression	36
7.4.4 Video and Image Transmission	37
7.4.5 Bandwidth without Compression	37
7.4.6 Bandwidth Tests with Image Compression	38
7.4.7 Bandwidth Tests with Video Compression	39
7.4.8 Conclusions	39
7.5 Commands and Data stream	40
7.6 User Interface	41
7.6.1 Graphical User Interface	41

7.6.1 Control User Interface	41
8. Task T3 – Mechanical Structure Project.....	42
8.1. Work Scenarios.....	42
8.1.1. Terrain Geometry	42
8.1.2. Environmental Issues	43
8.2. Structural Restrictions	43
8.2.1. Robot Length.....	43
8.2.2. Height and Width.....	45
8.2.3. Orientation.....	45
8.2.4. Weight Distribution	46
8.2.5 Other Requirements / Desired Features	46
8.3. Solutions	46
8.3.1. Frontal Arm.....	46
8.3.2. Autonomy and Communications	46
8.4. Robot Structure Proposal.....	47
8.4.1. General Description.....	47
8.4.2. Motors	48
8.4.3. Frontal Arm.....	53
8.4.5 Motor Controllers.....	53
8.4.6 Tracked Wheels	54
8.4.7 Batteries	54
8.4.8. PC104 and Remaining Electronics.....	56
8.4.9. Docking Mechanism	56
8.4.10. Gas Sensors.....	58
9. Task T4 – Equipment Characterization	59
9.1. PC Hardware Test	59
9.1.1. Introduction.....	59
9.1.2. PC104 Motherboard	59
9.1.2.1 Network Interface.....	59
9.1.2.2 Framegrabber	60
9.1.2.3 Serial (RS232) Interface	60
9.1.3. PC104 Power Supply	61
9.1.4. PC104 four USB2.0 hosts board	62
9.1.5. PC104 PCCard Adapter	62
9.1.6. Laptop Hard Disk.....	62
9.1.7. Compact Flash Hard Disk	62
9.1.8. USB Webcams	63
9.1.9. Thermal Camera	63
9.1.10. Wireless.....	63
9.2. Electronics	64
9.2.1. USB to Serial interface	64
9.2.1.1 USB/Serial FTDI chip.....	64
9.2.1.2 Serial Interface.....	65
9.2.1.3 Serial Interface Block Information.....	65
9.2.2 Motors Control and Monitoring	65
9.2.3 Gas Sensors.....	67
9.2.3.1 Generic Interface (TGS8XX).....	67

9.2.3.2 TGS8XX Preliminary Tests.....	68
9.2.3.3 Carbon Monoxide TGS2442.....	68
9.2.3.4 Carbon Monoxide TGS2442 test circuit.....	69
9.2.4 Temperature and Humidity Sensor.....	69
9.2.5 Lights.....	70
9.2.6 Accelerometers.....	71
9.2.7 Infrared distance measuring sensors.....	72
9.2.8 Batteries.....	72
9.2.9 Robot Power.....	72
9.2.10 Webcam Servos.....	75
10. Task T5 - Interface Software Development.....	76
10.1 Wireless Communications.....	76
10.2 UPS Device Driver.....	76
10.3 Framegrabber Device Driver.....	77
10.4 Commands and Data stream Protocol.....	77
10.5 Video Control Protocol.....	77
10.5.1 Implementation details.....	78
10.5.2 OpenH323 licensing details.....	79
10.6. Low-level PIC Communication Protocol.....	79
10.6.1. Generic PIC<->PC Protocol.....	79
10.6.2. PC->PIC Communication.....	80
10.6.3. PIC->PC Communication.....	81
10.6.4. PIC Memory Tables.....	81
10.7 PIC Software Integration.....	85
10.7.1 Temperature and Humidity Sensor.....	85
10.7.1.1. Communication.....	85
10.7.1.2 Converting the Output to Physical Values.....	86
10.7.2 Motor Board.....	88
10.7.2.1 Microcontroller main software cycle.....	88
10.7.2.2 Motor Interrupt State Machine.....	88
10.7.3 Gas Sensor Board.....	90
10.7.3.1 Microcontroller main software cycle.....	90
10.7.3.2 Interrupt sub-routine.....	90
10.7.4 Accelerometers Board.....	91
10.7.4.1 Microcontroller main software cycle.....	91
10.7.4.2 Interrupt sub-routine.....	92
10.7.5 Frontal Arm Board.....	92
10.7.5.1 Microcontroller main software cycle.....	92
10.7.5.2 Interrupt sub-routine.....	92
11. Task T6 – Mechanical Structure Assembly.....	93
11.1 General Structural Aspects.....	93
11.2 Tracks / Ground Clearance.....	94
11.3 Arm motor.....	95
11.4 Frontal Arm.....	96

11.5 Docking system.....	96
11.6. Web Cams	97
11.7. Weight distribution	98
12. Task T7 – Electronic Equipment Assembly	99
12.1. DC-DC Board	99
12.2. Batteries and Power.....	99
12.3. PC Assembly and WebCams.....	100
12.4. Motors Board	101
12.5. Body PIC #1	101
12.6. Body PIC #2.....	101
12.7. Frontal Arm and associated PIC board	102
12.8. Wireless Antennas	104
13. Task T8 – Software Modules Integration.....	105
13.1 User interface.....	105
13.1.1 Graphical User Interface	105
13.2 Control User Interface	108
14. Task T9 – Tests in the Lisbon Firefighters Headquarters.....	110
14.1 Introduction	110
14.2 Electronics	111
14.3 Mechanics.....	112
14.4 Software.....	114
15. References and Bibliography	116
Appendix 1 – Battery Chemistry Comparison.....	117
Appendix 2 – Gas Sensor Characterization	119
TGS 813 - Detection of Combustible Gases	119
TGS 825 - Sensor for Hydrogen Sulphide.....	120
TGS 842 - Detection of Methane.....	121
Universal Detection Circuit.....	122
Load Resistance.....	123
TGS 2442 – Carbon Monoxide.....	124
Appendix 3 – Docking System Mechanical Drawing	126
Top View	126
Side View	126
Back View	126
Section Views ‘A’ and ‘B’	127
Section View ‘C’	127
Sliding Doors.....	127
Released mechanism on the ground	128
Spring Deformation	128

1. Summary

This report describes all the work done in the RAPOSA project, from 01-March-2003 to 31-March-2005. All nine Tasks are described, giving a perspective of all the work achieved.

Task T1 – Electronic Hardware Architecture: The electronic hardware architecture was defined and its main components chosen. These includes four webcams with one on the back, plus a thermal camera to aid in victim detection even on total darkness; Low-level electronics were necessary to interface with low-level sensors like inclinometers, gas sensors, temperature and humidity sensors; motor control and associated power stages; high-level PC computer, featuring wireless communications. Low-level processing was done by Microchip PICs. Digital communication between low and high-level was made by serial (RS232) and USB links. The robot' power architecture was also projected, including choice of batteries chemistry.

Task T2 – Software Architecture: This task started by the analysis stage where the scope of the problem to be solved was defined. The output was a list of the desirable properties of the system. Based on these properties the overall agents based software architecture was drafted. The final phase was to specify and test software components. Those not only boost the development process but also make the software more steady and reliable.

Task T3 – Mechanical Structure Project: The mechanical requirements for the robot were studied. An aluminum structure was projected, featuring tracked wheels for locomotion in non-structured environments. It has some innovative aspects, like a motorized frontal arm, with variable inclination concerning the main body, also featuring motion tracks and cameras for a wider field of view; a docking mechanism allows the robot to switch from tethered to wireless operation and back in real time. The batteries and motors were chosen at this Task.

Task T4 – Equipment Characterization: The high-level hardware was assembled to form a full featured PC and its characteristics and flaws registered; all different kind of sensors were characterized, such as gas, temperature and humidity sensors, accelerometers configured to measure inclination, variable resistors and encoders for position and speed feedback. The projected electronics were adapted to the experimental sensor results.

Task T5 – Interface Software Development: Linux device drivers for some of the hardware (UPS, framegrabber) had to be developed, since the manufacturers did not provide some of them. A simple but versatile low-level communications protocol was defined, working in either RS232 or USB, thus accommodating future evolutions. Low-level software for the microcontrollers were developed, namely to interface with the sensors. The main cycles of execution on the PICs were defined;

Task T6 – Mechanical Structure Assembly: SetPontes company developed the mechanical chassis for the robot according to what was planned in T3. Since this is a very innovative project to both IdMind and SetPontes, its design was an iterative cooperation between electronics and mechanical. The docking mechanism was only projected at this stage, featuring two sliding doors that grab the cable end; the final weight distribution was studied and the center of mass was shifted forward, on a second iteration of the mechanical project.

Task T7 – Electronic Equipment Assembly: The electronic boards and their associated sensors were connected and assembled in the robot. Experimental characterization on the robot side required some minor electronic adjustments and some interaction with T8 for sensor calibration using the newly developed electronics.

TASK T8 – Software Modules Integration: In this stage all the software components developed on Task T5 were integrated and adapted when needed. During the integration preliminary tests were done. The detected flaws were corrected.

TASK T9 – Tests in the Lisbon Firefighters: The test stage has been reduced to about 3 weeks, since some amount of preliminary testing was done as part of T7 and T8. The robot was tested on the Lisbon Fire-fighters test camp and building. The results were according to expectations and only minor adjustments were required.

At the end of Task 9 the Raposa robot development and testing was concluded. A robot was built, featuring a main aluminum body with an additional arm (also in aluminum) featuring adjustable vertical orientation relative to the main body. Both body and arm feature locomotion tracked wheels. In the arm there are two web cameras with tilting mechanisms plus a thermal camera and artificial lights. There are also infrared distance sensors to evaluate the terrain ahead. The main body holds the processing units, wireless communication modules and antennas, batteries, motors and two other cameras, one at the front, other at the back. It has gas, temperature and humidity sensors. It also features a docking mechanism that allows real-time (de)attachment of a cable.

The solution reached at the end of Task 9 revealed adequate to explore medium complexity terrains (where the irregularities do not exceed the weight of the raised arm), climb stairs of more than 45 degrees inclination, search and find hidden persons on dark environments. All these characteristics met the project goals; the only detected limitation to the robot is its reduced ground clearance and the fact that it is not yet explosion proof.

Future work on the robot concerns mainly software aspects, increasing the robot capability of doing autonomous Tasks freeing or aiding the operator in driving the robot. If more robots were to be developed, the main structural modifications would be reducing its height (to increase ground clearance), weight and using smaller electronics.

2. Gantt Diagram

The following Gantt Diagram shows the predicted time schedule (the darker shaded rectangles above) and the real time schedule followed during the project (the lightly shaded rectangles below):

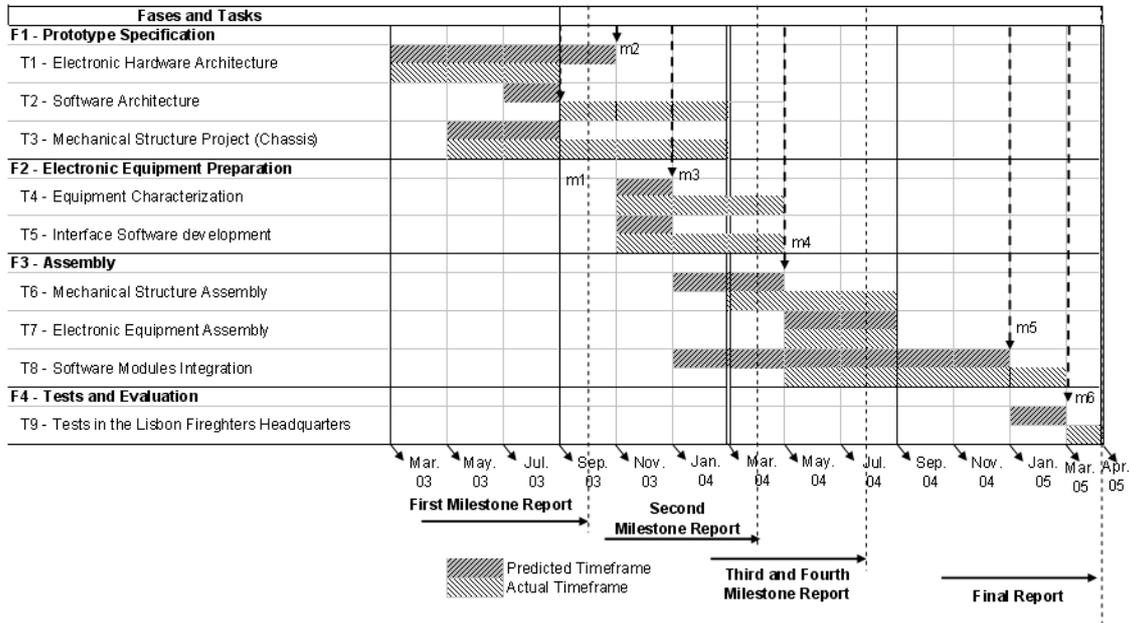


Fig. 1: Gantt Diagram

Task T1 concerns the choice of electronic components to be included in the robot. The sensors, cameras and pre-assembled electronics (PC, motor controllers) were all selected before the predicted time frame. The generic hardware architecture was also fully defined, featuring serial communication between low-level microcontrollers and high-level PC, either by RS232 or USB. The boards feature some flexibility, in such a way that the final sensor architecture could be reshaped, if necessary, in T4.

The software architecture in T2 was started when the IST trainee entered the project. The architecture was defined not only at high-level but also at the low-level.

All low-level software was developed at T5, so that in Task T8 a solid low-level platform was available. This included not only the microcontrollers' code, but also Linux device drivers.

Given some of the very innovative aspects of this robot, Task T3 took longer than expected. The visit to the University of South Florida (USF) was delayed due to the very tight schedule of Prof. Robin Murphy. Finally possible at November 2003, it provided extremely useful feedback of the drawbacks and problems of the existing robotics solutions on search and rescue scenarios. This new information implied some reformulation on RAPOSA, namely abandoning some of the original ideas (loose track wheels – no real advantage) and improving others (the importance of ground clearance, for example). The first drafts done by IdMind were further developed by SetPontes, the company contracted to build the robot.

At the time of Task T4 the acquired material was available (notable exceptions: thermal camera and batteries, the first due to very tight export regulations from the USA, the second due to inappropriate protection circuits) for testing. The sensors were characterized experimentally, so that the electronics design could be finished. At the

end of this Task, all boards projected by IdMind were under construction and some were even finished (e.g.: motor interface board).

Task T5 concerned low-level software, branching into two very distinct activities: microcontroller code, responsible for sensor reading and communications with the PC with a predefined protocol; Linux device drivers development. In Task T2 we defined that the PC on the robot should run Linux. However, not all hardware manufacturers supply Linux / UNIX drivers, leaving that Task to project consortium. The development of a Video4Linux compatible video capture device driver was a lengthy job.

Task T6 was subcontracted by IST to SetPontes, according to IdMind first sketches and experimental feedback. As expected on a prototype, some aspects needed to be revised after the first iteration of the robot. Rollers between wheels to increase ground clearance, rubber teeth on the tracks, reducing weight on the back to shift the center of mass were some of the aspects that required revision. This Task was finished only after all flaws were fully corrected.

Task T7 started as soon as the first version of the Raposa chassis was available. The PC, hard disk, controllers, sensors and custom electronics were assembled in the robot. Their internal localization was already planned on T3, but cabling and interconnection required some extra attention. Having all hardware in place, some extended testing was also conducted.

Task T8 started as soon as Task T2 and T5 were finished, debugged and tested; nevertheless it took longer than expected mainly due to mechanical and electrical work in progress with the robot. All the software work needed to be extensively tested on the laboratory and some hardware problems with the onboard PC have delayed these tests.

Task T9 was delayed since the software was not completely finished and a late solution to increase mobility in the tracked wheels was applied. The tests however revealed that the one month delay was worth it, since the performance of the robot on the Lisbon fire-fighters test camp met all our expectations and the late adjustments were fundamental.

The differences between the predicted and the actual project schedule revealed appropriate, the one month project extension to 31 March 2005 allowed intensive and relevant tests in real scenarios.

All Tasks have been fully completed and the prototype is ready to use.

3. Summary of the Results per Task

This chapter presents a summary of the main results achieved at the end of each Task, thus covering the results of the entire project, from 01-March-2003 to 31-March-2005. The detailed technical report of the work developed on each Task is presented on Chapters 6 to 14.

T1 – Electronic Hardware Architecture

A detailed analysis of the robot objectives and requirements was made in close collaboration with the Lisbon Fire-fighters (RSBL). The robot should be able to explore disaster scenarios, featuring low to no light environments, very irregular terrains, atmospheres where there is risk of explosion and small entrances. These requirements implied the choice of the following robot components:

- Environmental conditions sensors: gas, temperature and humidity sensors;
- Orientation feedback: accelerometers for the three axis to measure gravity, thus ascertaining inclination;
- Distance measurement: infrared distance measurement sensors to evaluate the difficulty of the obstacles ahead;
- 4 web video cameras, one at the rear of the robot and two other assembled in the frontal arm (discussed later), featuring both vertical and horizontal movement (pan & tilt).
- White LED lights, since the robot should operate under collapsed buildings, with no illumination and the illumination should withstand shocks;
- One thermal camera for heat / living person's detection. Even with artificial lights conventional cameras fail to provide any contrast if the environment is covered with dust. The thermal camera provides a valuable feedback on these cases;
- High-level processing unit, an IBM PC compatible computer for on-board data multiplexing, processing, recording and transmission; based on the PC104+ norm that features 96x96mm boards.
- Wireless communication capability;
- Brushless motors that do not generate sparks due to brush commutations.

The batteries are Li-Ion, the commercially available technology that has the best volume and weight energy density.

Low-level sensors and actuators return / receive analogical values, so low-level processing was necessary. This was achieved through Microchip microcontrollers (PICs). Communication between this level and the PC was achieved by a serial link, either by a conventional RS232 connection, or by a modern serial over USB connection. For the high-level software, the difference between the two connections is almost none.

T2 – Software Architecture

Software architecture basics were drafted based on IST recommendations. Their experience on the Rescue - Cooperative Navigation for Rescue Robots project (FCT SRI/32546/99-00) was applied on this project. Therefore, the main execution environment (Linux) and their architecture: software agents, blackboard and control/communication interface was kept. Some innovative aspects unique to this

robot were investigated, including video transmission on irregular quality / variable bandwidth channels. The more adequate protocol / codec was found to be the H26x family and the library used was the OpenH323.

For the console (Human Machine Interface) it was decided to use the Windows operating system and Managed C++ as the development language.

To focus the development of software on the core innovative aspects of our solution, the use of software development libraries was imperative. Namely we have used several libraries from Microsoft, Boost and a library for graphical instrumentation widgets from 9rays Corporation.

T3 – Mechanical Structure Project

The robot objectives and main structural restrictions were discussed on the early stages with the Lisbon Firefighters (RSBL). They provided valuable feedback, namely photos of some of their search and rescue missions and a visit to their training camp. This allowed a categorization of the several types of obstacles to overcome and also to discard scenarios where this robot should not operate, given their extreme complexity (e.g., underwater). The more important restrictions include:

- The robot should fit on a 40cm wide sewer pipe: this does not imply that the robot should operate on water environments, but it is otherwise a good metric for a “medium hole”, frequent on disaster scenarios;
- It should climb standard size steps: 17cm height, 23cm width.

A first mechanical structure was defined in conjunction with the mechanical construction company, SetPontes. Later on the process, IdMind and IST members visited the University of South Florida, having a unique chance to experience some of the available commercial search and rescue robotic solutions. This experience exposed some of the weak points of the original project mechanical design and re-enforced others, namely the necessity of a real time docking mechanism. The final specifications are herein presented:

- Two modules, a main body and a frontal arm, whose relative vertical orientation to the main body is adjustable;
- Both modules have two side tracked wheels to provide locomotion. The frontal arm locomotion is coupled to that of the main body.
- When the robot “flips” upside down, it continues its operation flawlessly. This implies that robot does not have a top or bottom part, so it should detect its orientation by itself in order to exchange the commands to the motors.
- The frontal arm features two webcams, each with an associated light and a thermal camera. These two webcams support has a 30 degrees horizontal pan. Associated with the arms ± 90 degrees tilt possibility, this gives a rather large field of view to the robot.
- The robot can be operated with or without a cable and the switch can be made remotely in real time. The cable supplies power and act as wireless transmitter.
- The 3 Li-Ion batteries location on the chassis was chosen to be on the front part of the main body, as it is crucial that the robot centre of mass is located on its front, so that the robot “falls to its head” when climbing stairs, rather than the opposite (flips or falls);

On the front of the main body of the robot there are two wheels on each side. One connected to the locomotion track on the main body, the other one connected to the

track on the arm. They are attached in such a way that the movement of the main body wheels is transmitted to the arm wheels, both rotating at the same speed.

However, the arm itself must be positioned, without interfering with the locomotion.

The two wheels and the motor to position the arm on the front leave no space to the locomotion motors, which are placed in the back of the robot, driving the back wheels. The tracks transmit the movement to the front. The two motors are placed on top of each other, transmitting the movement to the wheels through unitary gears.

The motors and battery capacities are also studied on this Task. Maxon Motors, gears and controllers were used, 5Ah Li-Ion batteries were found to be the best compromise between size and autonomy,

T4 – Equipment Characterization

The computer boards were analyzed. The network interface of the motherboard was found to be unstable at low voltages and should not be used. The wireless communication was successfully tested, but the final placement of the antennas is discussed on Task 7.

The thermal camera worked as expected, giving a thermal perception of the scenario. The manufacturer failed to provide complete documentation on all features of the camera (white level control, image inversion), which would give extra control over it.

The serial USB chip (emulates RS232 over an USB link) was tested as a perfect alternative to conventional RS232 links. It features an identical interface supporting higher speeds and more accurate error messages.

The boards featuring either RS232 or USB interfaces were devised. The motor board controls the three power stage / motor controllers from Maxon.

The gas, temperature and humidity sensors were studied so that a proper measurement circuit could be designed. As it was the temperature and humidity sensor. Adequate low pass / buffer circuit was also designed for the accelerometers.

Lithium batteries required an additional protection circuit. The first version provided by the manufacturer could not withstand the currents requested. A new circuit was requested and it works as expected protecting the batteries from both short-circuit and low voltage discharge.

The power path was defined and implemented, namely by voltage mixing – through diodes and voltage commutation – through a relay. A switch was associated with each of the batteries: motors and electronics.

T5 – Software Interface Development

Low-level software was developed, both for the microchip microcontrollers and for the PC (Linux device drivers).

The wireless communications were successfully tested (at software level) under their final configuration – Wireless PCCard working as an access point, to which the Ethernet-to-Wireless Bridge connects. However, the appropriate Linux device drivers required an updated kernel (2.4.26).

An UPS device driver was developed, for the test stages, whenever NiMH batteries were used. This driver is not used on the final robot configuration.

The frame grabber device driver was adapted from a similar device found on the internet. It had to be upgraded and adapted since the hardware on the robot features a

different chip. The driver was not video4linux compatible (the standard way Linux applications deal with video). The adaptation and upgrade to video4linux was successful and the video buffer can be accessed by the two standard methods (read and mmap – direct memory access).

A simple communication protocol between the microchip PICs and the PC was developed. Each PIC features a shared memory area where the PC reads or writes on, through a command that specifies the start address, number of bytes to read or write, and the bytes themselves. This protocol is versatile enough to be independent of the sensors or actuators connected to each PIC. This information is directly manipulated by the high-level software, thus allowing greater flexibility.

The protocol between the robot and the console was defined and implemented.

All the software modules defined on Task T2 were also implemented.

Some of the sensors and actuators, namely temperature and humidity, CO gas sensor, LED lights and motors required special low-level coding to produce the timing and/or protocol to achieve correct behavior. The microchip PIC 16F876 microcontroller had enough processing abilities for all these Tasks.

T6 – Mechanical Structure Assembly

The final CAD project and consequent construction were made by a sub-contracted company, SetPontes. The chassis was built in 4 mm aluminum, chosen both by its resistance and lightweight. The 4mm thickness was necessary so that the robot structure does not bend due to the tensions the robot is submitted to.

The robot should be as lightweight as possible, assuring simultaneously that its centre of mass is located at the front of the robot, so that when climbing stairs the robot falls forward. The locomotion motors were placed at the back, since the frontal wheels must transmit locomotion to the frontal arm and the arm orientation motor is also at the front. However, this disposition shifts the centre of mass backwards. To compensate this effect, the back of the robot was lightened, by reducing the back wheels mass (through holes) and removing some of the lateral plates, whenever possible;

Rubber teeth were placed on the track wheels, both to increase the ground clearance and to increase tack of the robot to the ground.

On this Task the docking mechanism was also fully developed, including the detachable pyramidal structure and the cable retention mechanism.

T7 – Electronic Equipment Assembly

During this Task, all electronic was assembled on the robot and tested on its final configuration.

The notebook hard disk was assembled under the PC104 boards. The circular polarization antennas were placed on the robot laterals, between wheels and behind rollers, for best performance at any robot orientation. The communications performance was acceptable, providing a range of about 20 to 30 meters on environments with heavy interference.

Two of the batteries were connected in series, thus giving a medium voltage of about 32V for the motors.

The motor interface board was assembled and tested in conjunction with the motor power controllers. These were properly calibrated for each motor (gain, minimum current, maximum current).

The body PIC #1 handles the data from the two accelerometers that were properly mounted on the chassis on a 90 degrees assembly (one vertical). This PIC is also responsible for driving the docking mechanism, checking if it has reached the desired position or if it is stuck. Connected to this board, we installed a green ON LED for the robot.

The second main body PIC was assembled. The gas sensors that connect to it were assembled on two separate boards and placed next to the PC. These two boards feature sockets for the sensors, so that they can be removed or replaced easily. The battery monitoring circuit needed some adjustments, since the final battery voltage is higher than advertised.

A PIC board was assembled on the frontal arm, responsible for the two camera servos actuation and also the associated white LEDs. The cameras were installed and a polycarbonate window protects them from the outside, namely environment dust and humidity. The thermal camera was successfully placed on the front arm, but the polycarbonate blocks thermal radiation, so a special IR plastic was used instead.

T8 – Software Modules Integration

During this phase the user interface was developed integrating the modules developed so far. During preliminary tests the flaws were corrected. We have taken the opportunity to do major improvements on the usability of the system. A special care was taken to place the human commands in a natural way in the interface to drive the robot. A major redesign was done on the graphical user interface to make it more user friendly. Some of these improvements have implied changes on some of the modules already developed, namely the yaw and roll instruments. Some of the components were only developed during this phases due to its later mechanical implementation namely the docking mechanism.

The video integration took more time to debug and stabilize than predicted, but the final solution has proven to have great video quality and good adaptability to changes on channel bandwidth

T9 – Tests on the Lisbon Firefighters Headquarters

Three test sessions were carried out at the Lisbon Firefighters that allowed extensive analysis of the robot performance. The robot exceeds expectations, being able to operate at irregular terrains and climb stairs even at 45 degrees (better than what was projected).

With the help of the fire-fighters it was finally possible to calibrate the gas sensors. The robot autonomy was about two hours, as expected. The images from the webcams were not always in the best conditions (bad white balance), but the thermal camera fully compensated that gap, allowing to identify people with great ease. A small position adjustment was made in the lights, due to glare effects. The only serious problem detected at electronics is the excessive heating that can limit the robot operation.

From a mechanical point of view the robot was able to operate on all types of scenarios it was designed to. The only exception is moving through grass, or other places that have very small debris that became stuck between wheels and tracks. A possible solution to solve the problem was analyzed. The cable and docking were also successfully tested; the only deviation from the original project is the possibility of suspending the robot from the cable. Some extra weight reduction on the robot should be achieved first.

From a software point of view the interface was easy to operate, as expected, a fire fighter took only about 10 to 15 minutes to get a good control over the robot operating it through the console. The video transmission only has a 1~2 second delay and the possibility of having a variable number of streams was fundamental to get good results under bad scenario conditions. When the robot is almost loosing communication, the single window yet receiving video becomes full of jitter, but the robot is still fully operational. If the robot loses communications, it stops.

4. Project Feedback at the Company

After the two years time period involvement in the Raposa project, IdMind has achieved a higher level of capability in developing medium sized robots. The research performed was valuable not only because of the concrete results achieved, but also on the methodologies that were developed, allowing IdMind to reduce design and production times in other robotic platforms from the company' portfolio.

The main aspects where the acquired know-how revealed important in terms of future company activities are the following:

- USB serial link;
- Generic Serial Protocols;
- Electronic design techniques (e.g.: DC-DC power conversion);
- Li-Ion batteries technology;
- Ultra-bright LEDs;
- Experience with gas sensing technology;
- Linux operating system, including device drivers development;
- Analog video capture;
- Wireless communications, namely antenna placement;
- Mechanical construction solutions and adjustments;
- Motor dimensioning for medium/large robots;
- Electronics and Mechanics integration.

All these technologies and methodologies can now be used not only on future Raposa like robots, but also on other robotic platforms sharing some of the characteristics. The USB communication protocol will be used on all future medium or large size robotic platforms produced and commercialized by IdMind. The generic serial protocol is even being back-ported to already developed products. Since the experience with Li-Ion batteries was not yet trouble free, their use will be restricted to demanding applications.

The dimensioning of motors and controllers constitutes also a valuable know-how, even though IdMind plans to continue studying this topic beyond the scope of the Raposa project. The objective is to develop motor controllers entirely at IdMind.

The project was also fundamental in order to gain a better experience on integrating IdMind electronics with the Linux operating system and even providing some extra high-level functionality to the final client, so that the robotic platform totally fulfills its needs.

All this innovation was only possible due to close collaboration with research and development teams from universities and the fire-fighters, namely:

- IST/ISR: several innovative aspects were included on features in the robot as a result of this collaboration, namely the cable docking mechanism with the possibility to switch between cable and wireless operation in real time and the adjustable autonomy concepts underlying the software design.
- University of South Florida: in their laboratories and test fields it was possible to study currently available robotics solutions to search and rescue projects.

- The Lisbon Fire-fighters shared their experience on the ground, on search and rescue operations. This feedback proved important on the robot development. Once the robot was finished, they also pointed out the main qualities and disadvantages of the final solution.

This prototype can be considered a success, not only because it meets the project objectives, but also because its limitations clearly were identified and are easy to solve in future prototypes.

The know how acquired on building this robot allows IdMind to shorten its development time for building medium size robots, namely sewer inspection robots for fire-fighters and industry, bomb disarming robots for security forces, surveillance robots for police or private security companies. This topic is further detailed on the next chapter.

5. Dissemination Activities

Within the development of the Raposa project, several contacts were established with other companies, in order to search for assistance in areas where IdMind did not have enough expertise. These include:

- SetPontes, the mechanical construction company: the final project of the robot has been developed in close collaboration between the two companies, in order to achieve an optimized prototype;
- SMP - Serviços de Informática: a wireless communications specialized company, that has been providing consultancy on a wireless communication solution;
- PSE Europe: company that supplies the high density Li-Ion batteries;
- Unalboro: provided rubber heights to the tracked wheels.

This project also provided visibility to the company, being an important reference for company' portfolio. The project has been referred in several presentations and media such as:

- Webpage (average monthly hits: ~2000):
 - Portuguese website (<http://www.idmint.pt/raposa/>)
 - English website (<http://www.idmind.pt/raposa/?l=en>).
- Conferences, exhibitions, projects:
 - IAV2004 – 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles;
 - RoboCup 2004 - 8th RoboCup International Symposium;
 - Project Proposal for ESA (European Space Agency);
 - Segurex 2005: an exhibition exclusively dedicated to Security Safety and Protection sectors. The Raposa prototype was showed at the Regimento de Sapadores Bombeiros stand (Pav 1, 1A19). The presentation used is found in the CD-ROM;
 - Robótica 2005, 29/04 to 1/05/2005: Exhibition of the Raposa Robot in action, climbing stairs and going through a pipe.
- Media:
 - Visão Magazine n.600 2/09/2004 page 82;
 - 2010 TV program, n. 12, 19/03/2005 (public television channel 2);
 - Jornal de Notícias newspaper, 18/4/2005: “Pequeno robô para ajudar equipas de salvamento”;
 - Rádio Renascença radio, 20/04/2005 13-14h: public forum about the Raposa robot and search and rescue robots;
 - Diário de Notícias newspaper, 20/04/2005: “Um ‘robot’ para detectar vítimas em catástrofes”
http://dn.sapo.pt/2005/04/20/sociedade/lisboa_robot_para_detectar_vitimas_c.html
 - Público newspaper, 20/04/2005: “Sapadores ganham robot para operações de busca”

- Metro newspaper, 20/04/2005: “Robô ajuda bombeiros a salvar pessoas”
- Jornal da noite TV program, 22/4/2005: 3 minutes news article about Raposa (on SIC channel);
- Correio da Manhã newspaper, 23/4/2005: “Sapadores têm Raposa pronta” <http://www.correiomanha.pt/noticia.asp?id=157702> .

Benefiting from this positive exposure and from the know-how acquired, IdMind is planning into promoting some of the possible robotic platforms mentioned on the previous chapter to potential costumers. These include national public security forces (police and fire-fighters), private security companies, industry and the international market. Concerning the international market, our first target will be European Union countries. The company know-how in the field of mobile robotics allows it to effectively compete in foreign markets.

The docking mechanism is an innovative technology and its registration under a patent is being studied.

RAPOSA project webpage has been kept updated and already features an English version, found at <http://www.idmind.pt/raposa/?l=en>.

6. Task T1 – Electronic Hardware Architecture

This chapter describes in detail the work developed under the Task T1 – Electronic Hardware Architecture. The first section enumerates the robot' requirements, as agreed with the Lisbon fire-fighters. The second section gives a description of the hardware architecture and the remaining sections present each of the chosen components.

6.1. Hardware Requirements

The main hardware requirements defined together with the Lisbon Firefighters are:

a) Sensors and cameras:

- The robot should capture the largest possible field of view. Multiple cameras are a possibility;
- A thermal camera is very useful in detecting living persons under a deep pile of garbage and dust;
- An accelerometer mounted as a tilt sensing unit estimates the orientation of the robot, so that the operator is warned if its moving on dangerous terrains;
- Infrared distance measuring sensors, to avoid holes;
- Microphone(s), so that possible survivors can be heard;
- Gas Detectors: for explosion risk assessment;
- Temperature and Humidity sensors, for atmosphere measurement;

b) High-level hardware:

- A small sized computer, for sensor and image processing, higher level routines and signal multiplexing (prior to wireless transmission);
- Framegrabber(s) for video capture;
- Wireless LAN (or other form of remote communication to the operation console);
- Hard Disk for OS storage and data logging.

c) Low-level electronics:

- Lights: on destroyed buildings, the lighting conditions are normally very bad or just non-existent;

6.2 Hardware Architecture

Given the previous requirements a hardware structure was devised. It is a PC computer based architecture, with multiplexing of data being done preferably at PC level. The use of serial and USB to serial interfaces between high and low-levels allows a modular and expandable design. This structure also gives more control at high-level, increasing the high-level programmer flexibility and simultaneously simplifying the low-level hardware design.

Fig. 2 presents the hardware architecture of the electronics.

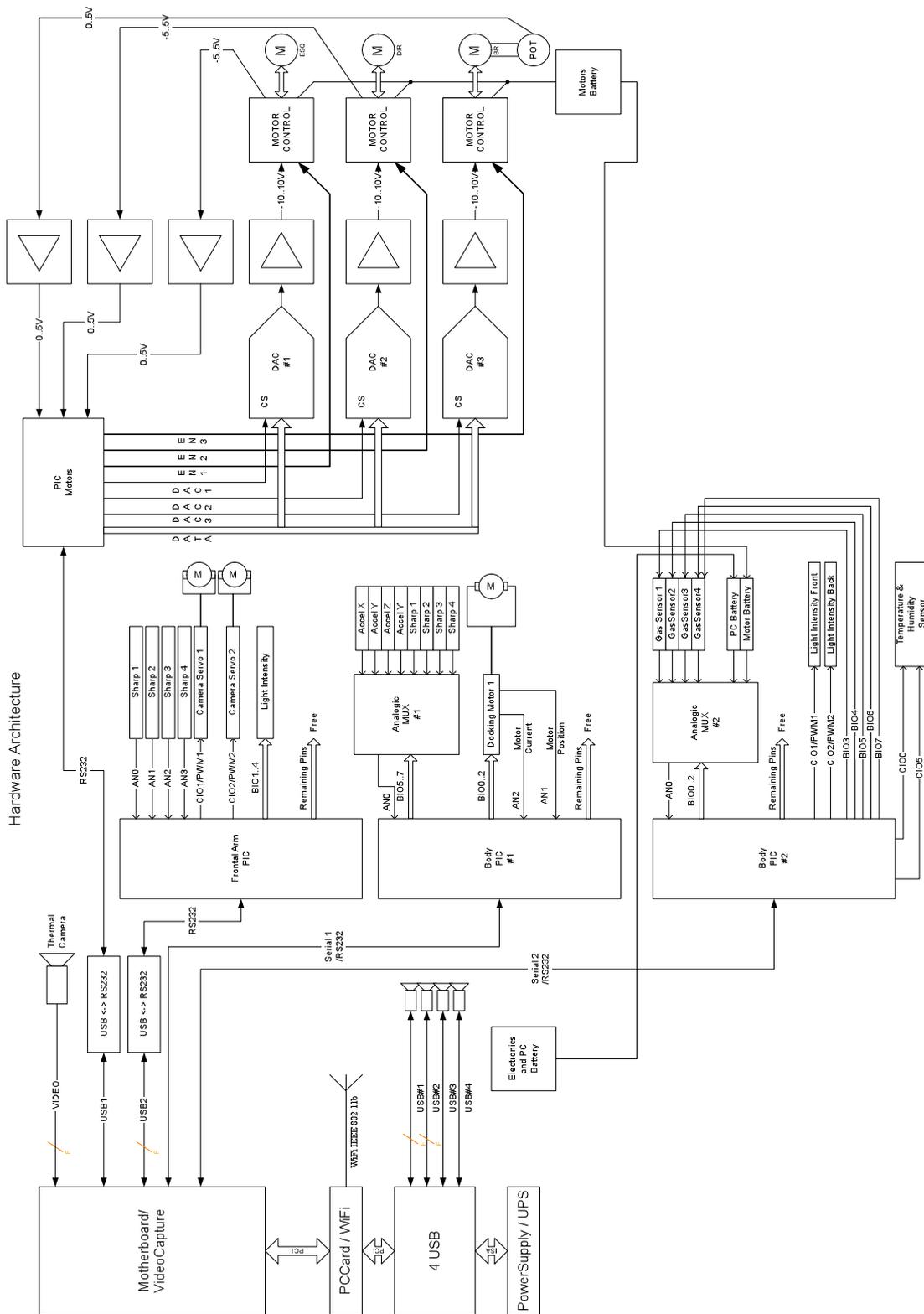


Fig. 2: Hardware Architecture

To understand the necessity of some hardware (or its location), a brief description of the robot mechanical structure follows. The robot has a main body where almost all hardware is located, including two locomotion motors. It also features a mechanical arm at its front that aids both in reaching higher places and in giving a wider range of

view, through the cameras installed there. The angle of this arm relative to the main body can be controlled by the operator, through a third motor on the robots body. The robot also features a docking mechanism, to assist on the real time (de)attachment of a power/communication cable on the robot. The choices that led to this structure are presented on Chapter 8.

The brain of the robot is the PC / motherboard and therefore it was the first component to be selected. Given the tight size restrictions, the smallest available standard board was adopted: PC104(+). The disadvantage of this platform is the somewhat limited processing power that, at the time, did not exceed a Pentium III processor.

To chosen architecture has a strong focus on USB connections. This protocol is modern, robust and easily expansible. This translates into good speeds, good error correction, solid operating system support and easy expansion through inexpensive hubs.

Following this orientation, four conventional imaging USB webcams were used. However, the motherboard only features two USB1.1 plugs that even in conjunction lack the bandwidth for the above cameras. A four way USB 2.0 PC104+ board is used to overcome this limitation. The USB2.0 norm has a 40 times speed increment over normal USB that accommodates the cameras easily.

The use of USB cameras also has the advantage of not requiring dedicated analogue frame grabbers. This way, only one framegrabber input is required to capture the video from the thermal camera, since no thermal cameras with USB interface and adequate size exist on the market. The chosen motherboard features an integrated framegrabber, thus no dedicated framegrabber board is required.

Besides the USB board, two other boards connect directly through the PC104(+) interface to the motherboard: a PCMCIA/PCBoard for connecting a Wireless Ethernet board and a UPS/ Power Supply board.

IdMind has a solid experience working with the serial / RS232 interface (between low-level hardware and high-level PC). However, the board only features two serial links, thus limiting the expandability and flexibility of the low-level interface. So, a brand new USB to serial interface was characterized and the architecture was conceived using USB to serial links in some of the boards. As will be seen in 9.2.1.1 USB/Serial FTDI chip, this interface is even more reliable than RS232, although similar in terms of interface.

Based on the above considerations, low speed sensors were multiplexed through the conventional serial link and the more demanding and critical sensors and actuators through the USB to serial link. In particular:

- The motor boards have a USB connection: the control of the robot' position and speed is critical;
- The frontal arm microcontroller (PIC) should be connected to the motherboard through the USB link, since the cables to the arm pass in the exterior of the robot;
- The remaining sensors and actuators being slow and/or placed on the main body can be multiplexed through one of the two serial links. Two other microchip microcontrollers control this multiplexing and interface with the low-level sensors.

The motor board receives data packets relative to the three motors, separates them and converts each of them into an analog value suitable for controlling the motor power stage.

The frontal arm features the following hardware:

- Two USB webcams;
- A thermal camera;
- One servo motor for each webcam;
- One light for each webcam;
- Two infrared distance sensors.

Since the frontal arm has a variable position relative to the main body, the cable count between the two sections should be as low as possible, both by mobility and noise immunity reasons.

As a result of this, the data associated with the low-level sensors and actuators (servos, lights and infrared sensors) on the frontal arm are processed locally at the frontal arm and only one USB cable connects with the main robot.

On the main body two microcontrollers exist. The first one interfaces with the accelerometers, converting its measurements to inclination, actuates the docking motor and has four other analog inputs available. The second microcontroller reads and enables the gas, temperature and humidity sensors. It also monitors the battery levels and regulates the light intensity of the lights on the main body of the robot.

6.3 High-level Processing

The PC104 is a form factor that defines 96 x 96 mm wide boards, that is easy expandable through a stack connector. PC104 defines an ISA bus stack connector, PC104+ adds a PCI bus additional connector. The motherboard must be PC104+ so that it has both buses / connectors. It is also advantageous that the motherboard includes the maximum features possible, dispensing additional boards. This motherboard includes:

- A Pentium III processor working at 700MHz (actively cooled);
- 128Mb of RAM memory;
- 1 10/100Mbps network interface;
- 2 USB 1.1 plugs;
- 2 serial ports;
- 1 parallel port;
- 1 integrated graphics card;
- 1 framegrabber.

The chosen power supply has 30W and supplies both +5V and +12V given an input voltage from any unregulated voltage from 11V to 30V. It also features a low voltage shutdown, to avoid harming the circuit and the batteries and an UPS feature that would allow battery charging. The board is only PC104, meaning that it has a single ISA bus.

Both the 4 way USB 2.0 and the PCMCIA/PCCard are PC104+ boards.

For data storage the notebook hard disk was used due to its smaller size and increased shock resistance:

Hitachi Travelstar HTS548040M9AT00 : 40Gb 5,400rpm

The maximum allowed values are:

Vibration	
Operating	1G @ 5 - 500Hz (sine wave)
Non-Operating	5G @ 22-500Hz
Shock	
Operating	200G @ 2ms half sine wave
Non-Operating	800G @ 1ms half sine wave

Table 1: Acceleration tolerated by the hard disk

In irregular terrains the vibration values may be exceeded and it is important that the system does not crash. As so, an additional 512Mb compact flash (solid state disk) disk was used. Its shock and vibration resistance is as follows:

- Operating Shock: 1000G
- Vibration Resistance: 15G

The hard disk main limitation is the number of write cycles per sector (300000). Therefore, the operating system and all critical data are stored in the compact flash, as long as it is read only (preferably). All recorded data is stored on the conventional hard drive. If this one stops operating due to vibration, limited functionality can be maintained.

6.4 Wireless Communication

For wireless communications, the first option to be made concerns the type of protocol and frequency to be used. Since the robot features live video transmission, low frequency signals (below 100MHz) are out of consideration. Most live video transmitters operate on the 2.4GHz frequency band, a frequency shared by the Wireless LAN normally used on computers. Since the propagation characteristics are identical, the Wireless LAN has at least one intrinsic advantage: the video can be encapsulated with all the other data (from sensors, to motors, etc) in one single data stream. It is also an inexpensive solution, since there are plenty of boards available in the market.

There are currently three main alternatives of Wireless LAN:

- IEEE 802.11b: the more mature protocol, has a 11Mbits maximum data rate, but commercially available boards reach until 200mW transmit power;
- IEEE 802.11g: an evolution of the previous protocol, allows 54Mbits data rates. However, there is no propagation advantage and the boards available do not reach the power levels of the previous protocol;
- IEEE 802.11a: a slightly different protocol, that also allows 54Mbits data rates, but operates at a higher frequency (5GHz). Although theoretically its range is smaller, experience of some manufacturers [1] shows that the results are better, especially in difficult environments.

The IEEE802.11a is the first option, since a board that supports this protocol also supports the previous two. However, additional considerations must be taken into account: as described on Chapter 8, the robot features an attachable cable that has a wireless transmitter at its end. This must be a very small structure, since it will be

dragged by the robot and it must have a simple electrical interface, thus excluding PCMCIA/PCCard boards.

A small Ethernet to Wireless bridge from SMC - SMC2670W was the smallest equipment found. However, it is only available for the IEEE 802.11b norm and no similar device is found on the other norms.

The wireless PCMCIA/PCCard is "caged" inside the robot structure, a perfect Faraday cage that blocks radiation. Therefore, there is need for external wireless antennas. At the time of specification only boards supporting the first standard allowed external antennas.

Considering these two aspects, the first standard (IEEE 802.11b) was used. The PCCard used was a *Senao* Wireless EL-2511CD PLUS EXT2 featuring two antenna plugs for spatial diversity.

6.5. Cameras

6.5.1. Standard Cameras

The transmitted images are of utmost relevance given that this is a tele-operated robot. The main interpreter of the image is a human operator; therefore the image should be in a familiar format. Precise colour identification is not a requirement, since no on-board shape recognition is done at the RAPOSA robot. Since these images should be transmitted wirelessly in a harsh environment, high frame rates are not sustainable.

High resolution would also be a significant requirement, since it would allow a careful analysis of a given area (with digital zoom). But an alternative set-up was preferred: instead of having a single high resolution camera, always looking front, or with a complicated and heavy pan-tilt unit, several low resolution cameras were used, located at optimized points on the robot. This way, on harsh environments, only the more important cameras will be turned on. If, otherwise, all bandwidth is available, all cameras may be on and since they will be pointing to different directions, a more complete image of the working scenario will be available.

A final requirement is the camera light sensitivity. Most of the times the robot will be sent to dark environments and artificial light will be used.

After an exhaustive search, a Logitech QuickCam Pro 4000 was chosen, mainly due to the following characteristics:

- CCD sensor (more sensible than the more common CMOS);
- Linux Support;
- 215gr;
- 4.5mm f/2.2 glass lens;
- 160x120 / 320x240 video at 30 frames per second (fps), at 640x480, the frame rate drops to 15fps;
- Auto exposure;
- Automatic white balance.

These cameras also have an integrated microphone, thus dispensing a separate component. Four cameras are used; their disposition on the robot is discussed in a following chapter.

The experimental characterization of the cameras in Task T4 was fundamental to determine the exact amount of artificial light required to acquire visible images in total darkness.

6.5.2. Thermal Camera

In a disaster scenario, everything looks grey as the inevitable dust normally covers everything. Therefore, even with artificial illumination, no distinguishable image at all can be retrieved using conventional cameras.

A thermal camera, on the other hand, is sensitive to heat radiation, thus allowing the perception and detection of heat sources. This can prove to be very useful in finding victims under debris and dust. It can also estimate the temperature of a given zone, warning that a fire may hide behind a hidden door or a wall.

Thermal cameras are, however, a rather expensive and bulky component, so a careful market study was made. The following minimal technical specifications were considered as necessary:

- Grayscale;
- Minimum resolution of 160x120;
- Video output compatible with the PC104 frame grabber;
- OEM solution;
- Temperature measurement capabilities (optional).

The chosen camera, Raytheon Control IR300D has the following characteristics:

Parameter	Value
Detector	Hybrid ferroelectric staring focal plane array
Spectral Response	7-14 μm
Image Resolution	320x240
Framerate	30Hz
Field of View	46° H X 35° V
Output	PAL video output
Cooled / Uncooled	Uncooled
Power	9.0-28.0 VDC, <7A (start), <1A (normal)
Weight	0.68 Kg
Size	74 x 102 x 125 [mm]
Operating Temperature	-30° to 66° C
Estimate Price w/ 20 MM Lens	~7500€

Table 2: Raytheon Control IR300D datasheet

The camera was chosen not only by its technical characteristics, but mainly by its size. The market research revealed only two cameras that fit on the robot, this one having the advantage of being the less expensive one.

6.6 Low-level electronics

6.6.1. IC Microcontrollers

Microchip microcontrollers were used for interfacing the analog and other low-level sensors with the high-level computer. The 16F876 chip used features the following pins:

- 5 analog inputs;
- 1 serial interface (RS232);
- 8 bits generic digital input / output;
- 2 PWM configurable outputs,

but any of these pins can be reprogrammed to act as a digital input/output, this giving extra flexibility.

6.6.2. Linear Accelerometer as an inclinometer

The robot should be able to climb 45° degrees inclinations and if the robot flips, the operator should be aware of that fact.

The image acquired by the camera(s) does, sometimes, provide an elusive idea of the correct robot orientation. Furthermore the robot is allowed to operate turned “upside down”. This requires frequent adjustments of the orientation estimates. A particularly difficult situation occurs when the robot is lowered down by a cable attached to its back and it spins around the cable.

The following solutions were studied:

- A simple tilt switch (binary): Gives a rough indication of orientation. Hysteresis must be used near the switching angle to avoid oscillations. At least two sensors would have to be used, in perpendicular assemblage;
- An analogic tilt sensor: these devices are based on analog accelerometers measuring gravity. Currently there are only two axis devices, with a limited range of about 70 to 75 degrees. For complete determination of the robot orientation, two devices assembled in an adequate 90 degrees assembly are required.

Solution a) lacks precision. Solution b), being based on accelerometers, is affected by the robot' accelerations. However, this effect can be minimized by performing a low pass filter on the output of the sensor.

In the next paragraphs we discuss how inclination can be calculated from accelerometer measurements. Given an accelerometer capable of measuring static accelerations, it is sensible to Earth gravity (g). When perfectly aligned with the Earth surface, both x and y axis yield zero and z axis yields g (or $-g$).

Once the robot starts to change its inclination, its tilt and roll angles can be determined. In the sequel we show, as an example, how to determine the tilt angle (associated with the x axis, for example):

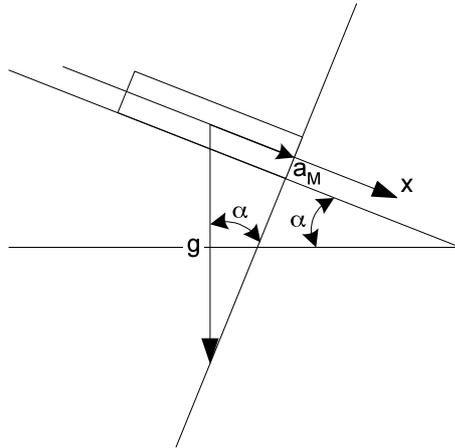


Fig. 3: Finding the inclination angle α by measuring static acceleration

The angle is determined by:

$$\alpha = \arcsin (a_M/g)$$

where a_M is the acceleration measured on that particular axis.

It is clear that for small angles, the relation between α and a_M (and thus, the voltage measured by the sensor) is approximately linear. As the angle grows, this relation no longer applies. The following theoretical graphic was plotted, which allows to establish a relation between the current angle and the available sensitivity.

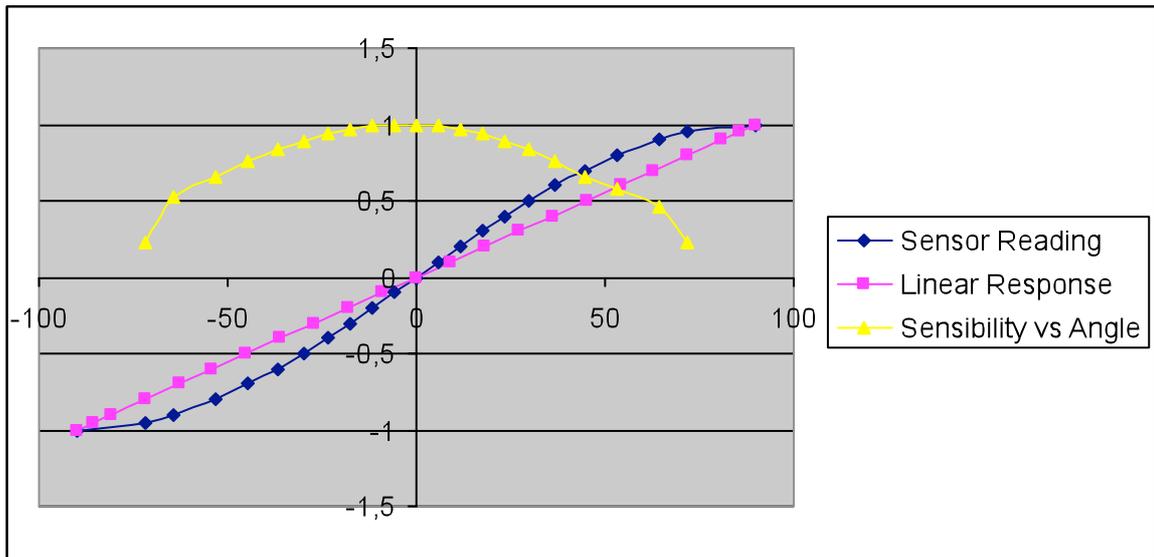


Fig. 4: Sensibility versus actual angle

The blue (diamond) line indicates the sensor response (in Volts) to inclinations ranging from -90 to 90 degrees. The yellow (triangle) line is a differential value, representing the rate of voltage change per degree. An ideal solution would be a straight horizontal (square) line, meaning that the sensibility was independent from the sensor orientation. It can be seen that for angles greater than 60 degrees the sensibility reduces to about half of zero degrees sensibility. However, since this is a “danger zone”, where the robot shall not operate, this limitation is considered acceptable.

Two Analog Devices ADXL311 accelerometers were chosen, with the following characteristics:

ADXL311	Min	Typ	Max
Measurement Range		± 2 g	
Sensitivity at XFILT, YFILT [mV/g]	140	167	195
3 dB Bandwidth At Pins XFILT, YFILT		6 kHz	
Output Resistance		32k Ω	
Zero g Bias Level Each Axis 0g Voltage	$V_D/2-20\%$	$V_D/2$	$V_D/2+20\%$

Table 3: Linear accelerometer short datasheet

6.6.3. Infrared distance measuring sensors

The following Sharp infrared distance measuring sensor was used:

Model	Sensing Distance [cm]	Dimensions L x W x D [mm]	Supply Current [mA]
GP2D120	4-40	13.5x44.5x18.9	50

Table 4: Different infrared distance measuring circuits to be used.

It has analogue outputs, with a characteristic that can be made piecewise linear. These sensors were assembled in the robot exterior, the precise location to be defined in 12.7.

6.6.4. Gas Detectors

According to the Lisbon Fire Fighters, the following toxic gases are to be detected:

- Methane, propane, butane and other gases that indicate high explosive levels;
- Carbon Monoxide;
- Hydrogen Sulphide (organic material decomposition);
- Oxygen.

From the manufacturer Figaro, the following sensors were included:

- TGS842 - High sensitivity to Methane;
- TGS813 - High sensitivity to methane, propane and butane (explosives);
- TGS203 - Carbon Monoxide Sensor;
- TGS825 - Hydrogen Sulphide.

All these sensors have analogue outputs. The last one requires an additional power supply, as the sensor must be heated. Oxygen sensors are vibration sensitive and were not used for now, but the robot hardware is capable of using any additional gas sensor.

6.6.5. Temperature and Humidity Sensors

The temperature and humidity sensors from Sensirion, family SHT1x/7x, gives a measurement of both relative humidity (RH) from 0% to 100% and temperature

measurement from -40° to 120° in a single sensor, using a digital interface. The sensor is very small and it has only four wires - two for power and two for serial communication

6.6.6. Servo Motors

Two Hitec small servo motors are holding the two frontal arm webcams. These motors have a low consumption (less than 200 mA) and thus can be directly driven by a PIC output.

6.6.7. Lights

The conventional solution for artificial lights on the robot is using electric bulbs, similar to the ones found in cars. They have a wide field of view and higher luminosity when compared with white LEDs. However, their casing is bulkier and they are not as resistant to vibration as white LEDs.

Luxeon has a series of white LEDs (Lumiled) with about 18 lumens / 1.43cd with a viewing angle of 110° . For this they require a current up to 350mA. They were acquired and tested with the webcams in total darkness.

6.7 Motors and Controllers

The choice of locomotion and arm motors only makes sense when the robots mass and structure is defined. Thus, this particular electronic aspect was included on Task T3. However, some considerations are possible here. If the robot is to be used on hazardous environments, explosion risks should be reduced at all costs. One usual source for this is the brushes necessary in DC motors for transmitting power to the rotating rotor. This mechanism produces unwanted sparks.

Brushless motors, on the other hand, have the electric windings on the stator, thus no current interruption induced sparks occur. This motor rotates by sequentially activating each of the three windings of the motor. This leads to increased control complexity, thus having a higher cost.

A market researched revealed that Maxon Motors have the best compromise in brushless motors and compatible gear heads (capable of taking the brushless motors highest rotation speeds). Since brushless motors are a very vendor specific product, it is safer to use controllers from the same brand.

The more recent Maxon motors brushless controller takes serial commands through a digital interface. However, it has a rather large size (180x103[mm]), not compatible with the planned robots dimensions. After some research, an older Maxon controller, controlled only by analog voltages was found to be the best solution. The PIC can control digital to analog converters quite easily and this other controller is smaller in size (103x70[mm]).

To control each motor only an analog voltage and a power stage enable bit is necessary. An electronics board featuring a single PIC can handle this task for all three motors. It receives the digital values for each motor through the serial link and the PIC places them on a common 8 bits bus. On this board, to each motor is assigned a digital to analog converter (DAC) that reads the bus at a given moment (commanded by the PIC), holds that value until a new cycle is ordered in its latch and performs a digital to analog conversion whose output commands the motor power stage.

The same PIC uses three of its analog outputs for velocity or position feedback from the controllers.

7. Task T2 – Software Architecture

7.1 Introduction

RAPOSA software architecture relies on the agent-oriented software architecture, which is under development for other projects at ISR/IST.

This architecture enables the concept of *adjustable autonomy*, where the objective is to move towards fully autonomous robots, but keeping humans in the loop for now, however increasing the autonomy level over time. This means that, as a first step, the human operator should not have to be concerned with the robot kinematics, but rather provide motion commands in the form of linear and angular velocities. The next step is to provide the ability for the operator to tell the robot, on a virtual map or on its captured image, where should it move and then let the robot negotiate its path through the environment, possibly cluttered with obstacles.

The agent-oriented software architecture also has the desirable property of *modularity*, allowing for future extensions to robot fleets.

The architecture consists of three basic components:

- Agents, which have their own execution context, its own state and memory and a way of sensing and taking actions on the environment;
- Blackboard, an information structure shared by all agents, possibly implemented in a distributed fashion and through which agents communicate data and more general information to each other;
- Control/communications interface, which handles details of the communications between agents, namely the ports and the communications API.

Each robot or remote operation consoles in the fleet communications network can have more than one agent within its own software architecture.

Five **Execution Modes** for each of the basic components are also defined: the **Control Mode** coordinates the run-time interactions between the basic components, while the **Design, Calibration, Supervisory Control** and **Logging and Data** Modes concern the operator and programmer interface.

The general block diagram of the software architecture is depicted in Fig. 5. The diagram represents the control modes that are available and their location within a communications network with one remote operation console and two robots.

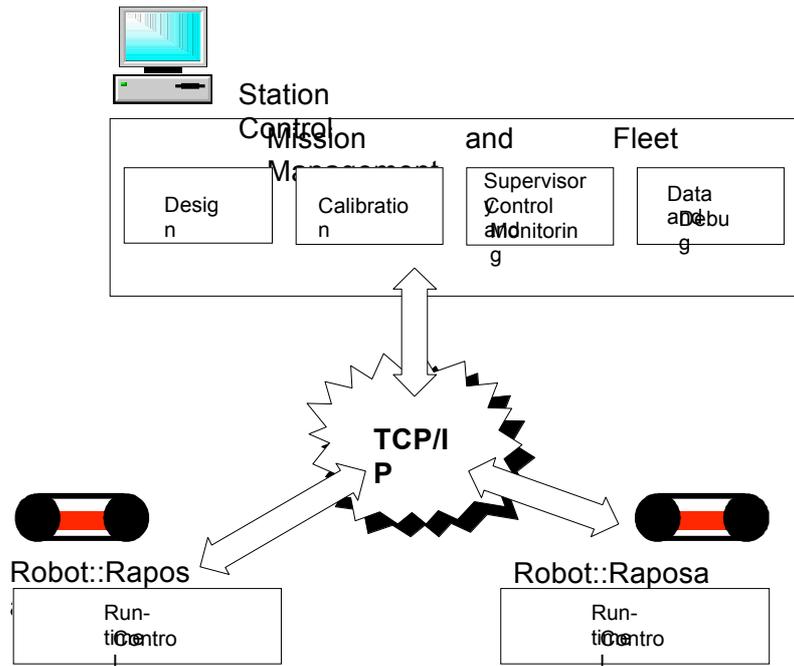


Fig. 5: Control modes of the agent-based software architecture

Agents are very general entities, which can perform a wide range of tasks, from data acquisition to data processing, navigation, motion control, etc. They communicate through ports that live in the blackboard and can be organized in groups. The main advantages of such organization are *modularity* (one agent can be replaced by another that performs the same function with a different algorithm without compromising the entire system) and *scalability* (theoretically, one can keep increasing the number of agents, as they can all get the available data made available in the blackboard by other agents and make their own more or less sophisticated usage of such data – e.g., image data is not property of a visual servoing algorithm or of an object detector algorithm, but rather can be used by separate agents that implement each of those algorithms for given purposes).

7.2 High-Level Software Issues

At RAPOSA project, two main systems must be considered (Fig. 6):

- The robot itself and all associated processing units;
- The remote operation console.

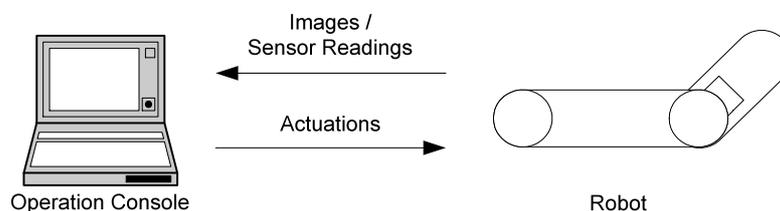


Fig. 6: Flow between the two different systems

Both systems include a high-level processing unit (an Intel compatible PC). The robot has additional low-level processing hardware, which is mentioned in the next sections.

The software running in the high-level units at both systems implement the agent-based software architecture described in the previous section. At the remote operation console, the following operations were implemented:

- Control/Communications interface;
- Supervisory Control and Monitoring execution mode, to translate the operator actuation commands (through keyboard or joystick) into robot motion commands, to display all received video streams and sensor readings in an ergonomic fashion (e.g., allowing focusing on a given camera / sensor at a given time, inverting the image if the robot has flipped, providing shape recognition information and sensor data integration);

At the robot, the following operations were implemented:

- Control/Communications interface, that listens to commands from the remote operation console and sends back image and sensor data ;
- Control execution mode, which includes the wheel motion control, camera activation and multiplexing and interface with the lower level hardware that handles the remaining sensors.

The robot operating system is Linux, due to the following reasons:

- It is freeware;
- It is very modular, allowing a custom configuration;
- It is an open architecture, allowing custom modules to be built;
- It has direct hardware access.

There are however some difficulties:

- Un-official drivers, that do not provide full hardware support;
- Lack of documentation.

The console operating system is Windowstm, due to the following reasons:

- It is more friendly user;
- Extensive support for commercial input devices, like joysticks and game pads;
- It has extensive and very stable software development kits;
- Several commercial libraries for instrumentation widgets.

7.3 Detailed Software Architecture

In Fig. 7 the relations among the agents that support the described operations are depicted. The Controller agents (RobotController and ConsoleController) are the agents responsible for controlling the other parts of the system: they enable, disable and configure the other agents. The logging is done on each controller. Video and data are recorded in different files.

On the robot side, an agent for each of the low-level hardware makes the bridge to the robot motors and robot sensors. The cameras, as special sensors, have an agent for each of them; this agent is responsible for picking the image from the video device. The video is compressed and then multiplexed with audio and sent to the console through the wireless network. The Video Control Agent is responsible for starting and stopping the video Agents.

The Robot Controller Agent controls the other agents, when the communication with the operator fails; he is able to put the robot at a safe state, stopping the motors. At last if the communication fails for too long, he is able to stop the video agents to spare the wireless bandwidth.

Finally, at the console, the video stream is decoded and sent to the corresponding window in the graphical user interface. For the user commands, an input controller agent controls the keyboard and the joystick. For each of the robot functionalities there is a small Robot Logic Micro Agent. The Robot and sensor Data are decoded by the several Robot Logic Micro Agents and presented on the graphical user interface.

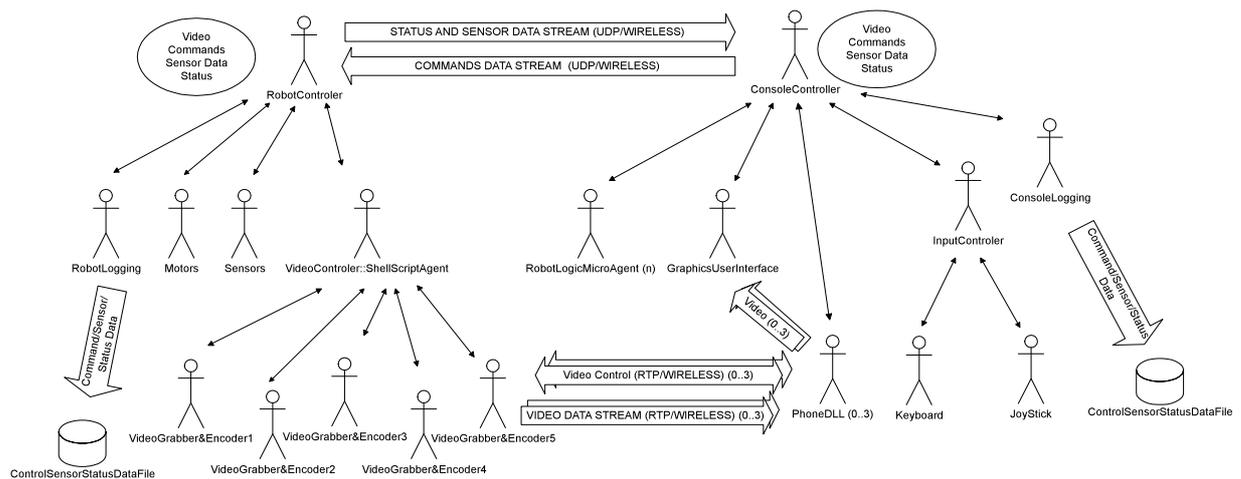


Fig. 7: Software Architecture

7.4 Video Stream

For remotely controlling the robot, the video stream for this application must have the lowest latency possible and the best possible smoothness. Furthermore, since we are using a wireless communication channel that has a low bandwidth (for video), the video stream bandwidth has to be compatible with the wireless channel bandwidth.

7.4.1 Video Bandwidth and Wireless Bandwidth

The video stream bandwidth is characterized by the frame rate (fps) and the size in Kbytes of each image. The size of image in Kbytes depends on the image resolution in pixels (e.g., 640x480, 320x240) and the colour depth resolution of each pixel (16, 24 or 32 bits). For greyscale images the pixel depth is always 8bits. This can be summarized by the following expression:

Video bandwidth [bits/s]=Image width x image height x color depth x frames per second (fps)

The wireless channel is prone to errors due to interference and packet collisions from other sources (normally burst errors). Also, as distance increases between the peers, packet loss due to weak signals also increases, thus reducing the effective bandwidth (see Table 5).

Transmission Speed (Mbps)	Outdoors ¹	Indoors ²
1375 KB/s	128m	27m
687 KB/s	152m	31m
250 KB/s	167m	32.5m
125 KB/s	250m	33m

Table 5: Wireless Bandwidth for IEEE 802.11b

The video stream bandwidth can be reduced in several ways: by reducing the image resolution, by reducing the frame rate or by the use of compression. We can use compression in two ways:

- using image compression (for instance jpeg) in each frame or;
- using video compression (for instance Mpeg).

Compressing the image or the video also consumes computational resources (mainly on the robot side) and inserts additional delays that may perturb the robot operation.

7.4.2 Image Compression

Using image compression reduces not only the image size but also its quality. The compression ratio can be chosen to exchange image quality by image size; the compression (JPEG) intelligently reduces image resolution and colour depth from where it is less noticeable by humans.

Choosing the correct image compression and resolution to meet an adequate image size is not straightforward because it depends on the subject of the image. Also the trade-off between image quality and video smoothness (by reducing the frame rate) is subjective.

7.4.3 Video Compression

Video compression normally has better image quality and smoothness for the same bandwidth than only image compression. On the other hand is more computational greedy, has bigger latency and is more sensitive to transmission burst errors; if an intra-frame is lost due to a communication error the stream is disrupted till a retransmission or till the next intra-frame.

For compressing video there are several encoding schemes and protocols. The best known is the MPEG family (Mpeg, Mpeg2 and Mpeg4). The mpeg family is not particularly suited to real-time low latency applications because it takes time and resources to encode and it also need to buffer images due to encoding scheme. Usually its applications are video on demand, streaming from video files and so on.

For low-latency real time video the H.26X family is better suited, this family was developed for low-latency applications running through variable bandwidth channels, namely for internet video conferencing (MSN Messenger uses H.263).

The video compression identifies blocks between images; it then codes the stream using intra-frames and inter-frames. H.261's intra-coded frames are frames which are fully encoded with no reference. Inter-coded frames are frames whose encoding is based on the previous frames. The receiver picks the intra-frame, displaying it and the

¹ Transmission with line-of-sight without interferences

² Typical home or office environment with floor and ceiling obstructions

n^{th} after inter-frames are decoded using the previous frames. H.261 also has a variable compression scheme that couples with a variable bandwidth channel. H.263 is an evolution of H.261 requiring half the bandwidth for the same video quality.

7.4.4 Video and Image Transmission

After being coded, the video/image is organized in packets and sent through the network. The receiver picks the packets to reassemble the frame info, decodes the frame and finally displays it on the monitor (see figure). On the network we can have several ways of transmitting the video: like RTP/UDP, HTTP/TCP, UDP or TCP.

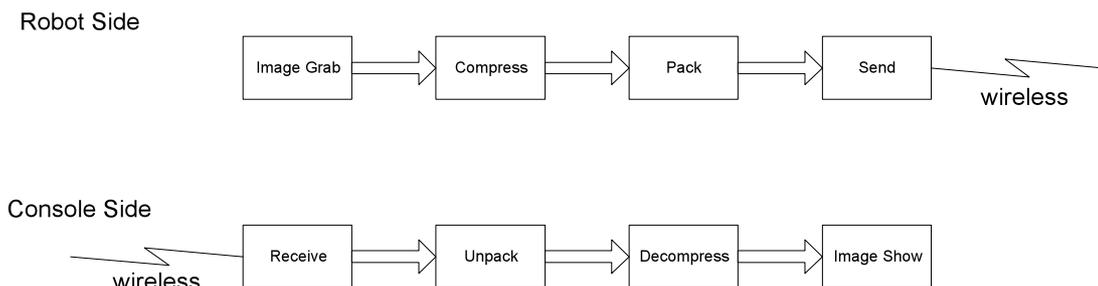


Fig. 8: Video Streaming

7.4.5 Bandwidth without Compression

We have conducted several tests³ in order to study possible freeware solutions for coding, transmitting and displaying the video: Ffmpeg, Mp4ip, Videolan and Palantir. We have tried also several commercial products like MSN Messenger. Several of these solutions provide more than one possibility for coding the video. We have tried the mpeg1, mpeg2 and mpeg4 video codec's and the H261 and H263. We have tried also JPG for compressing the images and to transmit images without compression. To transmit the data we have tried RTP/UDP, TCP, RTS/UDP and HTTP/UDP. Most of the solutions are not compatible amongst them because they use different or proprietary packing schemes.

The major problem encountered was video latency: well over 1s delays which is unacceptable for remotely operating a vehicle of this type.

We could not conclude if this is due mainly to the delays encoding the stream or to the players doing video buffering. In fact almost all players are designed to play a smooth (but delayed) video stream without losing frames, instead of the desired jerky but real-time no-latency video stream. Nevertheless we have some good results using JPG, H261 e H263 with players designed with low-latency concerns.

Table 6 shows the used video bandwidth when sending images without compression. Right away we can notice that for a 320x240@16bpp image the used bandwidth is already superior to the maximum available wireless bandwidth. Even in greyscale the used bandwidth is superior to compressed colour images (See Table 6).

³ On a local network (12,5MB/s) using a similar computer to the RAPOSA onboard CPU.

Image Resolution	Image Size	Used Bandwidth (12fps)
80x60@8bpp (grayscale)	4.8KB	57.6KB
320x240@8bpp (greyscale)	76.8KB/s	921.6KB/s
640x480@8bpp (greyscale)	307.2 KB/s	3686KB/s
80x60@8bpp	9.6KB	115.2KB
320x248@16bpp	153.6KB/s	1843.2KB/s
640x480@16bpp	614.4 KB/s	7372KB/s

Table 6: Used Bandwidth without Image Compression

7.4.6 Bandwidth Tests with Image Compression

With a JPG compression and a 320x240 image with 16bits of colour we have attained frame rates of 12fps with delays below half second. The used channel bandwidth (changing the compression parameter) ranges from 32KB/s (low quality) till 590 KB/s (high quality). On Table 7 we can observe how the compression parameter affects the bandwidth. On Table 8 we show the used bandwidth with for a 640x480 image.

Jpeg compression Quality parameter	Frame rate	Image Size (average)	Used Bandwidth
1 (A) (low quality- high compression)	15	2.1KB	32KB/s
25 (B)	15	4.5KB	68KB/s
50 (C)	15	6.3KB	94KB/s
80 (D)	15	10.3KB	154KB/s
99 (E) (high quality – low Compression)	15	39.3KB	590KB/s

Table 7: Image of 320x240@16bpp with JPEG Compression

A subjective analysis of the image quality was made. For the 32KB/s the image looked like an 80x60 image almost in greyscale. With the parameter set to high quality the image is virtually equivalent in quality to the image before compression. For compression values between high and low compression the final image quality were already very good. As a side note high resolution image with a high compression ratio does not make sense, because in fact the final image has lower quality. Instead we can choose to pick right away a less resolution image from the camera and use a lower compression ratio; having this way the same image quality and size using less computational resources (compare the tables with and without compression).

Jpeg compression Quality parameter	Frame rate	Image Size (average)	Used Bandwidth
1 (F) (low quality- high compression)	12	6.3KB	75KB/s
25 (G)	12	11,7KB	140KB/s
50 (H)	12	18.8KB	226KB/s
80 (I)	11.7	31.2KB	365KB/s
99 (J) (high quality – low Compression)	10	68.2KB	713KB/s

Table 8: Image of 640x480@16bpp with JPEG Compression

7.4.7 Bandwidth Tests with Video Compression

Finally using H261 and H263 the bandwidth used is depicted in the next table. The H261 and H263 impose that the image resolutions be CIF or QCIF (Quarter CIF). CIF is a 352x288 image YCrCb and QCIF as 176x144 also in YCrCb and is used more information on the luminance signal than on the chrominance (4Y, 1Cr, 1Cb). As mentioned these are protocols that can adapt to bandwidth changes on the network, so they are appropriated for this application. They can change the used bandwidth by changing a parameter P, the used bandwidth changes in accordance to $P \times 64\text{KB/s}$ ($P=1, 2, 3 \dots 30$), assuming 30fps. So the target bit rate is from 64KB/s till 1920KB/s, which is compatible with the wireless channel. The frame rate can go from 7.5 till 30 fps which is compatible with our needs.

Parameter P	Frame Rate	Bandwidth H261	Bandwidth H263
Medium-Low	10fps	48KB/s	24KB/s
Medium	15fps	96KB/s	48KB/s
medium-High	15fps CIF	192KB/s	96KB/s

Table 9 A CIF image (352x288) with H261 or H263 video compression

All the images have much better quality than the lowest quality attained by JPG, we can subjectively say that for the same JPG quality the H261 uses 1/2 to 1/3 of the bandwidth and the H263 uses 1/4 to 1/6 of the jpeg bandwidth.

Since we are also limited in terms of computational power for encoding the video stream and since H26X is also more computational greedy than JPG, possibly with bigger latencies, our options are dependent on the type of software libraries that we can use and further tests using these libraries.

7.4.8 Conclusions

Table 10 relates bandwidth and number of video streams to video quality. The strategy is to have only one parameter (the bandwidth) to control the video quality. The operator chooses which cameras he/she wants to view (till a maximum of 3 at the same time) and the systems adapts the video size, compression and frame rate of the selected camera streams given the current wireless bandwidth. We did this for the 3 types of encoding schemes that we might use. We chose 5 or 10 frames per second, because it is enough for our needs. We also tried to have the best quality, with a safety boundary, for each configuration. For lower channel bit rates we chose to drop the image resolution before reducing the video frame rate. For the table places where we did not have a suitable channel bandwidth, using the H261 or H262 we could have at least one

video stream. In the worst case scenario it is expected that the operator only uses one camera at a time.

Bandwidth	JPEG			H261			H263		
	1Cam	2Cam	3Cam	1Cam	2Cam	3Cam	1Cam	2Cam	3Cam
50 KB/s	-	-	-	A	-	-	B	A	-
125 KB/s	B	-	-	B	A	-	C	B	A
250 KB/s	G	C	B	C	B	A	C	C	B
687 KB/s	I	H	H	C	C	C	C	C	C
1375 KB/s	I	I	I	C	C	C	C	C	C

Table 10: Bandwidth Occupancy

Analyzing the Table 10 we can see that the H26x family needs less bandwidth and gives better results.

In our tests the solution using H261 as proven to be the most feasible in terms of computational power, latency and robustness to bandwidth fluctuations. The chosen library was the openH323. This library includes also the audio encoding, packetizer, network and control channels. The more important are the frame rate, the maximum used bandwidth, the compression quality and the image size. Those parameters can be changed with little effort during the mission.

7.5 Commands and Data stream

The operation console periodically sends command data to the robot. The robot periodically sends status and sensor data to the operation console. This data is sent through a UDP connection because is better to loose packets than to waste time waiting for the retransmission of packets already outdated. Immediately afterwards a packet is lost, another one is ready to be sent with updated information.

This implies that it must be possible to recover completely the state of the robot from the packets, even if some of them are lost. To achieve this we periodically send the whole commands from the operation console to the robot in packets, even if a particular command did not change its value. From the robot to the operation console we do exactly the same thing but this time instead of commands we send the sensor and robot status data.

Each PIC in the robot has a table for writing the commands and a table for reading the sensors state. The size of the table is 16 Bytes for writing and 16 Bytes for reading. Since there are four PICs in the robot, this means that each UDP packet for commands can have at top 4x16 Bytes long (64). For the robot status we need also at least 64Bytes for reading all the sensors state in all PICs. So in the worst case scenario we can have 128 Bytes.

The used bandwidth for these streams ranges from 1kB/s to 5kB/s, with a very generous safety margin. We are assuming also a command and sensor rate of 10-15 Hz. For these values we can afford to exchange all this data without having bandwidth concerns and thus gaining robustness to communication failure. For this used bandwidth we do not need to apply data compression to the streams.

7.6 User Interface

7.6.1 Graphical User Interface

A graphics user interface must be simple, easy to understand and easy to control.

Clearly, it is preferable to minimize the control options available to the operator, instead of cluttering the interface with numerous but useless options.

We have several levels of operation, following this concept, by providing “advanced” options that can be used only in a very particular circumstance or by “advanced” users. These options were removed from the basic level of the interface and putted on the “advanced” level. Thus the user can choose from several levels the one suited for her/him or for a particular task.

The same applies to the information that is displayed to the operator. We have several levels of detail. On the basic level we show only the information that is most appropriate for basic robot operations. As the level of detail increases we should show the data from sensors (like gas) that most of the time the operator does not want to analyse.

There are also other ways of easing the operator job. Several tasks are automated:

- Set threshold values on the sensor values - this allows the operator to focus on driving instead of being all the time looking at the sensor readings. If a sensor value goes higher (or lower) than the threshold value, the operator is notified;
- Reverse the motor commands and the cameras image if the robot flips over;
- Select automatically the best video quality in accordance to the available wireless bandwidth;
- Reverse the commands if the operator chooses to drive backwards.

All the automatic tasks are presented as an option to the operator; he/she is able to choose which ones he/she feels comfortable to deal with, from the ones that he/she would like to control.

7.6.1 Control User Interface

A game-pad type of joystick was chosen for driving the robot and for choosing the more common options. The operator can hold this type of joystick in his hands without the need of a stable support, thus being a valuable feature on the field of operations.

8. Task T3 – Mechanical Structure Project

The operation scenarios and requirements of a search robot were studied with the Lisbon Fire-fighters, using their experience on Search and Rescue (S&R) operations. IdMind and IST also visited the University of South Florida, where they experienced with some of the already existing S&R robots. Based mainly on these two sources of information, this chapter presents the projected mechanical structure.

The first step discusses the kind of terrain in which the robot operates. Then, a structure is proposed and implemented in a 3D CAD drawing, used to assist in the building (Task T6).

8.1. Work Scenarios

8.1.1. Terrain Geometry

A Search and Rescue robot main feature is its ability to overcome difficult terrains, while transmitting relevant data to the operation console. The first parameter to be taken into account is the height the robot has to climb and how the surrounding scenario renders difficult or simplifies this task. The four main classes of obstacles are:

A) A simple vertical climb (a.k.a. step) (height h):

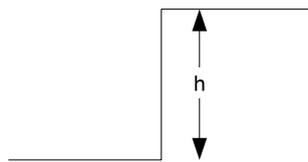


Fig. 9: Vertical height h

B) "Hole" (width w , height h) where the robot might get stuck:



Fig. 10: Hole

C) As a start point we assume a conventional vehicle structure, with its locomotion wheels on the sides of the robot (a rectangular form). Due to terrain changes resulting from the robot movement, a situation may arise where the robot only contact point to the ground is at its centre, leaving its (tracked) wheels on the air. So, the robot cannot get out of such situation because the locomotion parts have nothing to stand upon.

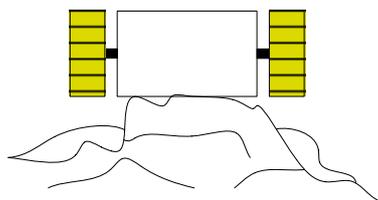


Fig. 11: Robot stuck at its middle, with no contact between the tracked wheels and the ground.

D) An inclination. In this case, it is important to choose the maximum angle at which the robot can still operate.

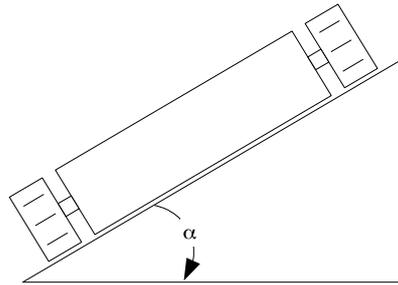


Fig. 12: Maximum operation angle α

Both A) and D) are project variables: maximum height and inclination the robot can climb / withstand. The first obstacle can be repeated, to form stairs. The way the robot transverses it is a major concern: one step at a time, or can it climb more easily once the first step is done?

Point B) has two possibilities: if the hole width is bigger than the robot length, the situation is identical to the one on item A). If however, the hole is smaller, the angle at which the robot approaches the height is also a matter of concern.

Point C) is more difficult to resolve, as it requires ingenious ways to release the robot.

A first approach to solve these problems is to have a high (so that no obstacle can oppose it) and long robot, so that it does not get stuck inside holes. However, the main purpose of this robot, as stated by the Fire Fighters, is the ability to enter small holes. They stated that the robot should be able to enter in sewer of 0.4 meters radius.

8.1.2. Environmental Issues

As the search and rescue operations proceed, a key factor is the evaluation of the risk involved. When the risk is too high, a decision has to be made not to send the teams to the field. Most of the times, however, the correct evaluation is only possible by actually measuring the variables in-situ. Should this be the case, a robot can provide a valuable help. It can travel to the desired place, estimate the environment life support capabilities, without any human risk involved. If properly designed, it can even work in an explosive environment.

So, a major requirement is that the robot does not generate any spark, so that it can travel in gas-leaked environments without igniting them. Also prone to happen in a disaster scenario is water pipes rupture and inherent flooding. However, this implies other design concerns that are outside the scope of this project. For now, the robot must withstand splashes, but it should not be used immersed under water.

8.2. Structural Restrictions

We now proceed to a quantitative analysis of the required parameters.

8.2.1. Robot Length

As described in A), the stairs are particularly unfavourable obstacles. They can be climbed in one of two ways: one step at a time or two or more steps at a time.

The first solution might seem simpler in a first approach, but it requires a lengthy horizontal plane after the step, so that the robot can straighten itself up, before it

approaches the next obstacle. Also, since each step is taken once at a time, the robot orientation is always changing.

The alternative is illustrated in the following picture.

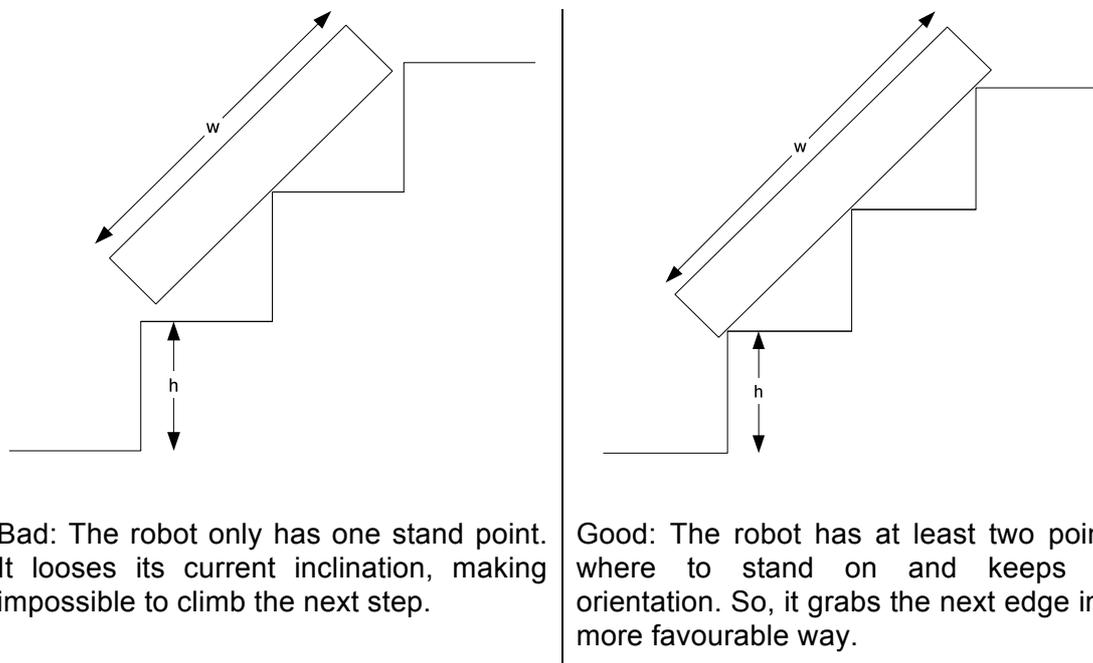
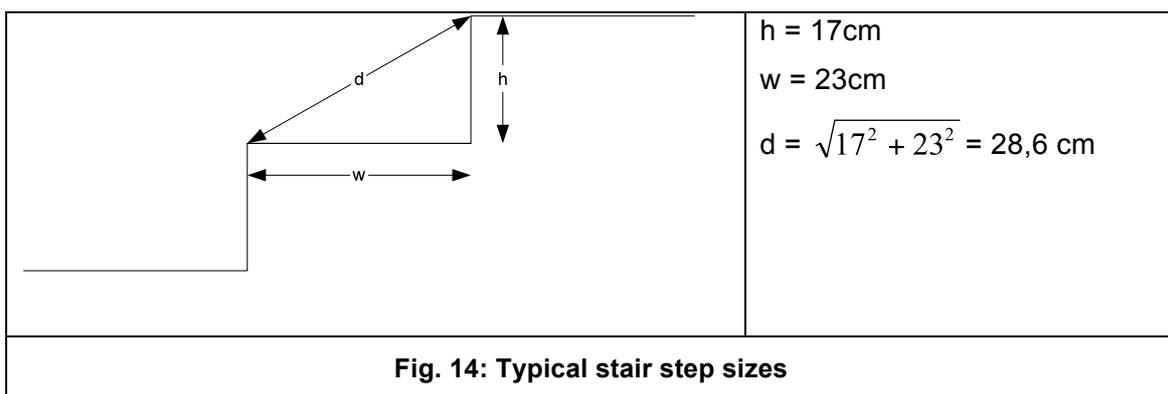


Fig. 13: Robot behavior for different robot lengths.

If the robot length is big enough to support itself in, at least, three stair vertices, it can preserve its inclination and climb the stairs more easily, meaning that once the first step is being traversed, the robot can grasp the next step and from now on the inclination is constant, leading to a simpler kinematics.

Based on these facts the robot length must be greater or equal to the diagonal of two steps. Regular household stairs have the following dimensions,



This allowing us to calculate a minimum length for the robot.

$$L = 2 * d = 2 * 28.6 = 57.2\text{cm}$$

If wheels (even if they are large) are used, we cannot assume that the robot keeps its orientation when it is climbing stairs. With only two or three small contact points, the

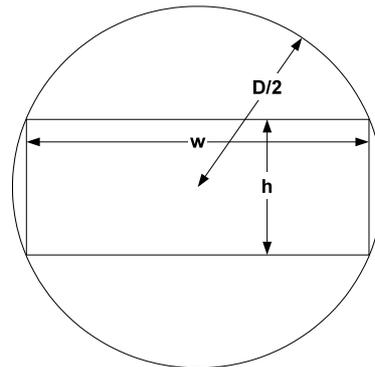
robot angle is largely dependent on its position. If, on the other hand, tracked wheels are to be used, they provide an approximately planar surface, which provides a better support for the robot. The tracked wheels also provide a larger contact surface to the ground, thus improving traction. This was the adopted solution.

8.2.2. Height and Width

After the meeting with the Lisbon Fire Fighters it was agreed that the robot should be capable of entering a hole with a 40cm diameter. Since this corresponds to standard sewer pipes, based on this value, a set of possible Heights vs Width was found:

Hole Diameter [cm]	Height [cm]	Maximum Width [cm]
40	10	38,73
40	12	38,16
40	14	37,47
40	15	37,08
40	16	36,66
40	17	36,21
40	18	35,72

Table 11 : Robot Width



$$\left(\frac{h}{2}\right)^2 + \left(\frac{w}{2}\right)^2 = \left(\frac{D}{2}\right)^2$$

$$\Leftrightarrow w_{\max} = \sqrt{D^2 - h^2}$$

Fig. 15 : Robot Width

Larger wheels are a preferable option, since they allow surpassing larger obstacles. On the other hand, they also are heavier to drive by the motors. This requires larger motors, therefore raising other restrictions, namely in terms of power supply and internal space availability for installation.

Some extra restrictions are found next, so for now only the set of possible values is defined.

8.2.3. Orientation

The bigger the angle the robot must still climb / hold on, the more torque its driving motors require. This, however, can lead to undesired weight increase.

One major problem on very steep slopes is the risk that the robot turns upside down. If the robot is designed not to have a “top” and “bottom” (meaning they are interchangeable), the flipping of the robot is not a severe problem.

Nevertheless, it is very important to know when the robot has flipped, because:

- The wheels rotation has to be reversed;
- The camera images have to be flipped, aiming not to confuse the remote human operator.

The motors must climb an angle of $\alpha = 45^\circ$.

8.2.4. Weight Distribution

The robot must be heavier on its front. This is a critical aspect when the robot is climbing stairs, so that it always falls forward. Although the robot can operate flipped, this should not happen when climbing stairs. The closer the centre of mass is of the front of the robot, the more aligned with the ground it moves.

8.2.5 Other Requirements / Desired Features

The robot may transmit via radio link, but an option for tethered operation is also to be included and not mutually exclusive. The tether would carry both power and communications. It could also be used as a way of dropping the robot onto a deep hole.

8.3. Solutions

In this section a number of solutions for the previous problems and requirements are presented and studied.

8.3.1. Frontal Arm

To overcome problems A) B) and C) a frontal arm with variable orientation concerning the main body and with included locomotion (engaged with the main locomotion) is considered. This arm provides means to grab edges higher than the robot main body height. It may also prove useful to grab the lower ground when only the robot main body has ground contact (C) problem).

This arm is an effective second body, with sensors and cameras included. Its orientation is driven by a DC brushless motor. Additionally, this allows a wider range of view points for the cameras installed on its front.

8.3.2. Autonomy and Communications

Existing robots have one of two main configurations:

- Fully wireless solution. All power is derived from batteries;
- Tethered system. Power is delivered through the cable.

Both solutions have strong and weak points.

The tethered solution provides better autonomy and assured bandwidth. It can also be used to sustain or pull the robot. However, a cable may get stuck, broken, etc, thus limiting the robot' mobility.

The wireless solution, on the other hand, proves itself less dependent on the terrain where it can move and the number of turns it has to take. Its autonomy depends solely on batteries. Wireless communications may prove very unreliable. Standard wireless LAN devices can reach a maximum of 50m indoors, in good conditions. This is not the case on disaster scenarios where twisted metal, big piles of concrete, all kinds of obstacles, edges, electrical wires, etc, block and reflect the signal, making it difficult to communicate at high data rates (or to communicated at all).

The feedback received from other search and rescue teams advises the use of a tether. In many cases the electromagnetic noise is too high and wireless communications may not work at all. A cable, although being a "dead weight", provides stable power and communication.

Traditional solutions allow either configurations, but even if the same robot supports both configurations, the change must be done at the setup stage, being a time consuming job. The solution proposed for this robot goes a step further, allowing the

cable to be attached / detached whenever necessary in real time. To that purpose, a “docking mechanism” was installed on the back of the robot. The robot comes close to the cable, grabs it and attaches it, through a perpendicular lock.

If the cable is not required anymore, the lock is pulled off. As the robot moves, the cable releases itself from the robot. This solution requires an additional camera on the back of the robot (the fourth webcam), to assist on finding and attaching the cable.

8.4. Robot Structure Proposal

8.4.1. General Description

The robot' structure is now presented based on the previously discussed restrictions and solutions.

The robot has two main parts: a main body and a frontal arm. The mobile arm moves around an axis parallel to the front wheels axis, doing a vertical sweep. The arm main purpose, as seen previously, is to grab higher edges and otherwise help the robot to free itself on difficult terrains. It has thicker tracked wheels, connected to the main wheels traction system. Thus, each of these tracked wheels moves at the same speed as the main wheels on that side (discounting transmission losses). This solution avoids two extra motors to drive the arms wheels. Fig. 16 shows the main body tracked wheels (red) and the arm tracked wheels (green).

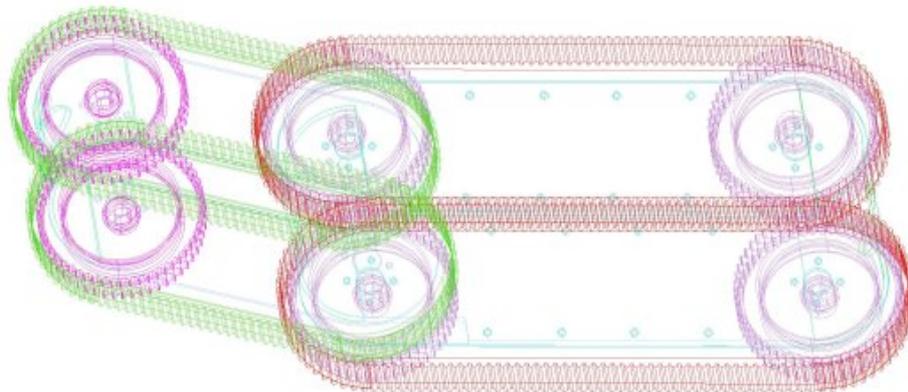


Fig. 16: The main body and arm tracked wheels

The chosen dimensions are: 140 mm height (accommodates for the PC stack, aluminium sheets, some ground clearance and the tracks), 369mm width (from the table above).

The arm has a 217mm length from the exterior of the main body. This allows it to grab edges (with the arm at a 45 degrees angle) with:

$(217 + 140/2 - 140/2) \times \sin(45^\circ) + 140/2 = 223\text{mm} > 170\text{mm}$ of the standard sizes height.

The structure is longer than the minimum required, to be safe. The final projected dimensions are listed in figure Fig. 17, measured in cm.

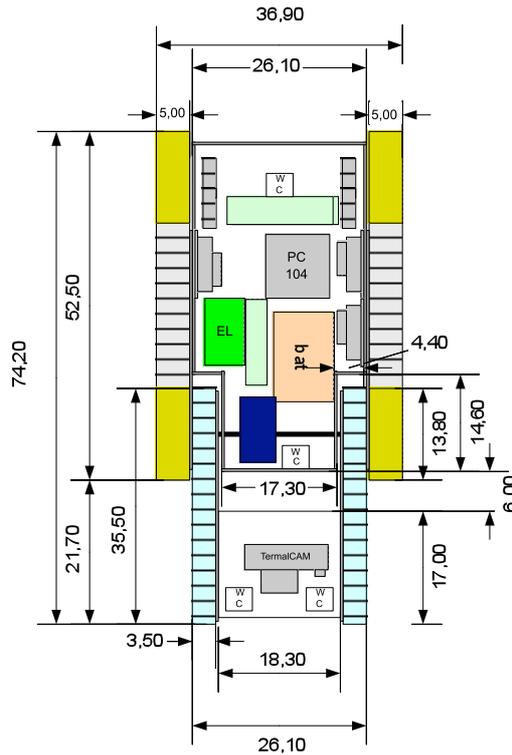


Fig. 17: Robot top view and dimensions in cm.

The mobile arm carries two webcams and the thermal camera, disposed side by side. Due to its size, the thermal camera is at the middle and rotated 90 degrees. The software can rotate the image back 90 degrees, so that the operator does not need to be aware of the real camera orientation. Each webcam has an associated white LED.

Additionally, a set of infrared distance measuring units are also assembled in the arm (not represented on the figure), allowing it to estimate the distance of the surrounding terrain.

The main body also features two webcams. One is used when the arm is at a steeper angle, to see what is straight ahead of the robot. The arm is hollow nearby the main body, thus reducing the minimum angle at which the main body webcam gives usable image. The other camera is at the robot' back to aid in the cable docking.

The robot main body includes 3 motors (light green on the figure): one for each side – differential drive, placed on top of each other and a third one to move the arm up and down. It also carries the batteries, PC104 computer and accessories, low-level electronics, sensors (orientation, gas, temperature and humidity – not in the figure). Finally, the cable attach mechanism is located back of the main body.

On the remaining section, each of the components is explained in more detail.

8.4.2. Motors

The first step in choosing the motors is to estimate the weight of the robot. Although most of the components are not fully defined at this stage and the final weight of the metal structure itself was still unknown, rough estimates were made:

Component	Weight [kg]	(A)curate / (E)stimate
PC104 w/ power supply	0.6	A
Toshiba Portable Hard Disk	0.1	A
2x (Locomotion Motor and Controller)	2.1	E
1x (Arm Motor and Controller)	0.7	E
Low-level electronics	0.5	E
Batteries	1.5	A
Cameras	1	E
Robots Chassis	16	E
Total	22.5	

Given the necessary margin, the robot was dimensioned for a total weight of about 30kg.

To determine the locomotion motors and the gearboxes to be used, three cases were studied.

1 – MIN: Represent the minimum torque needed to make the robot go forward in a horizontal plane without any obstacle, using the two motors at the same time.

2 – MAX/2: Represent the torque needed to make the robot climb a 45 degree obstacle using the two motors at the same time.

3 – MAX: Represent, the worst case scenario, the torque needed to make the robot climb a 45 degree obstacle using only one motor.

	MIN	MAX/2	MAX
Robot mass (Kg)	30	30	30
Wheels used	2	2	1
Radio of the wheel (m)	0,07	0,07	0,07
Acceleration (m/s ²)	1	9,8	9,8
Maximum angle (°)	-----	45	45
Force (N)	30,0	103,9	207,9
Gearbox torque (Nm)	2,1	7,3	14,6

Table 12: Necessary torque at the end of the gearbox

In the MIN scenario, an acceleration of 1 m/s was used to estimate the force that the robot needs to go forward. In the other two situations only the vertical vector of the gravity acceleration at a 45 degree obstacle was used.

Table 12 indicates that the motor and the gearbox should be able to provide a torque, at the end of the gearbox, of at least 15 Nm.

A gearbox of 113:1 was chosen.

Gearbox	MIN	MAX/2	MAX
Reduction X:1	113	113	113
Max cont. torque (Nm)	15	15	15
Interm. Perm. Torque (Nm)	22,5	22,5	22,5
Efficiency (%)	72	72	72
Motor Torque (Nm)	0,026	0,089	0,179

Table 13: Necessary torque at the motor

The parameters of the gearbox were used to calculate the torque of the motor that should be used to make the robot run.

A motor with a maximum continuous torque of 0.108Nm and a stall torque of 0.705 Nm was selected. With these motors the robot are able to climb an obstacle with 45 degree angle, using 1 or 2 motors. When using only one motor, the motor of the robot exceeds the maximum continuous torque but is still able to climb the 45 degree obstacle for a small period of time.

The angular and the linear velocity, plus the electrical current in the robot for these three scenarios, both at 24 and 32 volts, are listed below:

	MIN	MAX/2	MAX
Speed/Torque gradient(rpm/mNm)	8,5	8,5	8,5
Max. Cont. Torque(5000 rpm)	0,108	0,108	0,108
Stall torque	0,705	0,705	0,705
Torque constant (mNm/A)	38,2	38,2	38,2
Speed Constant (rpm/V)	250	250	250
Current at Motor Torque (A)	0,68	2,34	4,68
Speed at Motor Torque - 24V (rpm)	5781	5240	4480
Speed at Motor Torque - 32V (rpm)	7781	7240	6480
Angular speed at the gearbox 24V [rpm]	36,8	33,4	28,5
Angular speed at the gearbox 32V [rpm]	49,6	46,1	41,3
Wheel Linear Velocity at 24V (m/s)	0,27	0,24	0,21
Wheel Linear Velocity at 32V (m/s)	0,36	0,34	0,30

Table 14: Effective velocities

The maximum current needed is about 5A, but we estimate that in a free obstacle terrain the average current consumption is about 1A for each motor, in a very irregular terrain the average current consumption is around 1,5A and 3A.

The maximum velocity of the robot ranges from 20cm/s to 36cm/s, depending on the power supplied to the motor and the angle of the obstacle that the robot has to climb.

To determine the arm motor and the gearboxes to be used we determined the force necessary to move the arm.

This was accomplished by connecting a 13 cm fixation iron system to the axe of the arm. The maximum weight necessary to move the arm at 13cm was found to be around 11.5 Kg. So the force necessary at the axe should be 2930.2 N.

It was necessary to use two gearboxes, one with a high output torque with a 90° axe output and a second gearbox connected between the first gearbox and the motor.

The calculations are shown in the sequel:

Mass	299
Radio of the Axe	0,005
Acceleration (m/s ²)	9,8
Force (N)	2930,2
First Gearbox Torque (Nm)	14,7

Table 15: Estimated force necessary to move the arm

The 90° high torque gearbox was chosen with characteristics:

Reduction X:1	10
Efficiency (%)	60
Second Gearbox Torque (Nm)	2.4

Table 16: First gear box input

The planetary gearbox was chosen:

Reduction X:1	246
Max cont. torque (Nm)	6
Interm. Perm. Torque (Nm)	7,5
Efficiency (%)	60
Motor Torque (Nm)	0,017

Table 17: Second gear box input

Motor:

Speed/Torque gradient(rpm/mNm)	8,5
Max. Cont. Torque(5000 rpm)	0,108
Stall torque	0,705
Torque constant (mNm/A)	38,2
Speed Constant (rpm/V)	250
Current at Motor Torque (A)	0,81
Speed at Motor Torque - 24V (rpm)	10641
Speed at Motor Torque - 32V (rpm)	14361
Angular speed of the motor at 24V (rpm)	2,6
Angular speed of the motor at 32V (rpm)	3,5

Table 18: Motor characteristics

The chosen components are herein resumed:

Locomotion

- 2 x Maxon Brushless DC Motor EC40 120W 24V (ref: 118896)
- 2 x Maxon Planetary Gear head GP42C 113:1 (ref: 203126)
- 2 x Maxon Digital Encoder HEDS 5540 (ref: 110515)
- 2 x Maxon 4-Q-EC Servo amplifier 50VDC (ref: 132368)

Arm

- 1 x Maxon Brushless DC Motor EC32 80W 24V (ref: 118889)
- 1 x Maxon Planetary Gear head GP32C 246:1 (ref: 166949)
- 1 x Maxon Digital Encoder HEDS 5540 (ref: 110513)
- 1 x Maxon 4-Q-EC Servo amplifier 50VDC (ref: 132368)

The two 120W locomotion Maxon motors are rather lengthy (about 15cm), so their placement was not a trivial task. The option was to place them on the back of the robot, where it is wider. The front axis already has the arm motor and the transmission between main body and arm tracked wheels, thus leaving no space for additional motors. Since the frontal wheels are connected to the back wheels through the strap, the rear locomotion possible induced instability does not exist. The two motors are placed on top of each other. The transmission to each wheel axis was done by two gears, with a 1:1 reduction ratio.

Since the two motors are on top of each other, adequate support for both is required, and although their added diameter is smaller than the robots' height, the remaining top and bottom space is used by their supporting pieces.

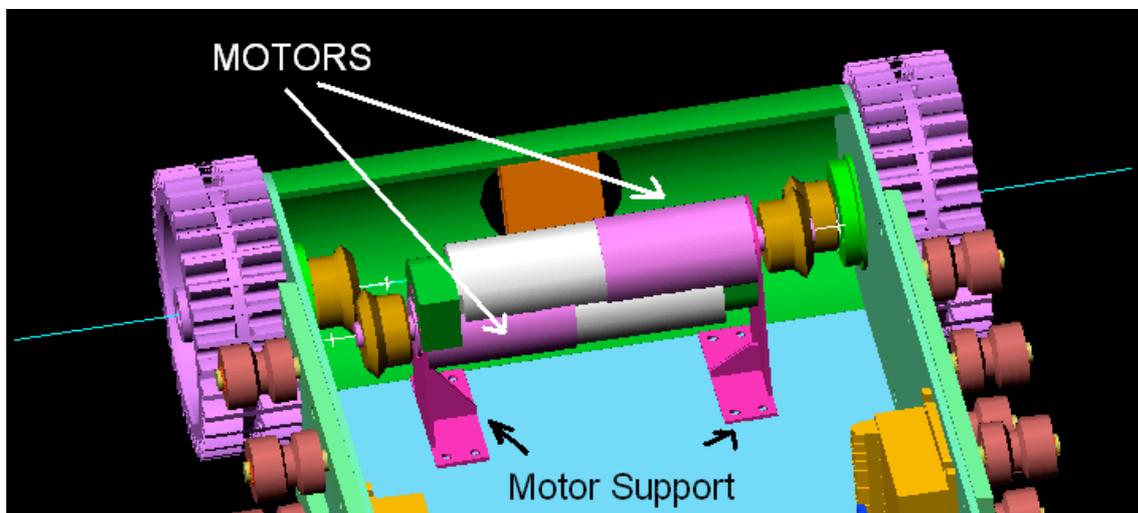


Fig. 18: Locomotion Motors and its support structure

The tracked wheels mechanism is isolated from the robots body, through a thick aluminum sheet.

The frontal arm is driven by a third motor in order to change its orientation, thus requiring a common axis to the frontal wheels. A (45 °) set of gears allows the motor to be placed at a 90° degrees orientation relative to the arm axis; otherwise the motor does not fit.

8.4.3. Frontal Arm

The robot arm is a second body attached to the main in such a way that it can change its vertical orientation. Its width is smaller than the main body, fitting between the two main tracked wheels. It also has tracked wheels (albeit smaller, about 3.5cm wide), each one synchronized with the main body tracked wheel on the respective side.

To move the arm up and down a third motor is used, driving an axis that supports the two internal front wheels. The frontal part of the robot is less wide than the main body, in order to accommodate for the second set of wheels. Therefore, even though this motor is a little shorter than the previous two, it has to be installed perpendicularly relative to the wheels axis.

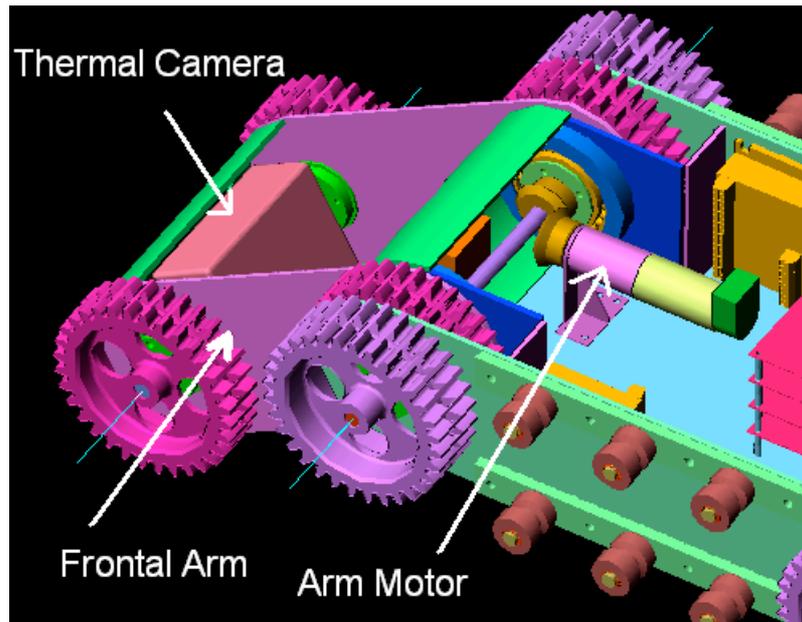


Fig. 19: Frontal Arm

The frontal arm holds the thermal camera and two webcams. The thermal camera is higher than the body, so it has to be installed horizontally in order to fit. The software rotates the image accordingly. Due to the limited angle of vision of the Webcams (about 30°), there is a small area just in front of the thermal camera that where they are not able to see.

In order to overcome this problem and give some extra flexibility to the robot, a very small DC servo is attached to each Webcam, allowing a slight adjustment of the pan angle of each of the cameras (not more than $\pm 10^\circ$). Both servos are coupled in a way that their angle concerning the robot longitudinal axis is always symmetrical. This way, only one software parameter controls the cameras total angle.

The frontal arm also features lights and infrared distance sensors, but their small size does not require any mechanical consideration at this time.

8.4.5 Motor Controllers

All three motor controllers are equal and occupy a rather large area. They feature an aluminum heat spreader of about 103x70 mm. In order to better dissipate their heat they were attached to the aluminum plate that separates the body from the tracked wheels. This is the most favorable placement, both in terms of occupied space as well as heat spreading. Fig. 20 shows the location of the controllers in an empty robot.

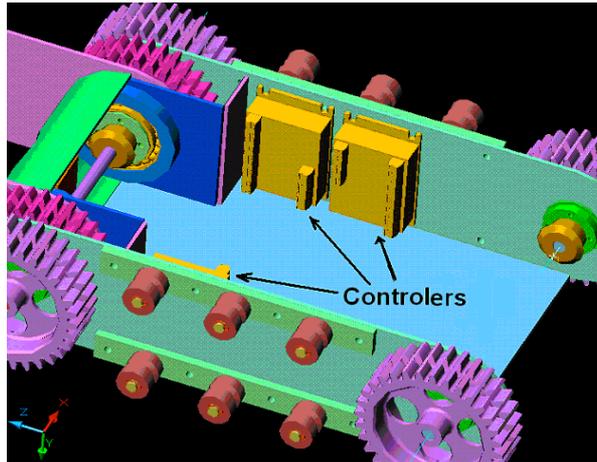


Fig. 20: Motor controllers

8.4.6 Tracked Wheels

If the tracked wheel strap was only supported by the two opposed wheels, there would be little grabbing force between strap and ground, as the irregularities of the terrain would deform the strap, with no force feedback other than the strap itself. If a set of passive rollers is placed along the strap, the irregularities force the strap against the rollers, but no farther from there, thus achieving a better contact to the ground. These rollers are represented on dark red in Fig. 20.

8.4.7 Batteries

IdMind current expertise shows that having separate batteries for the motors and electronics is generally a good idea, since the noise induced by the motors does not affect the remaining electronics.

Appendix 1 evaluates different battery chemistries. The main conclusion concerns the energy density of the chemistries: Li-Ion gives better results both on energy volume and weight density. Since these are two critical aspects of our robot (size and weight), the choice was clear. There are, of course, some disadvantages:

- Risk of explosion: if allowed to exceed the nominal ratings, there is a real risk of battery explosion. A protection circuit is a mandatory part of these batteries.
- Higher cost: since this is a newer technology, its costs are yet higher than other conventional chemistries.

Li-Ion cells also feature the higher cycle life and their best operating current is equal (or lower) than their rated capacity. This means that, contrary to the lead acid and NiMH solutions, their use at rated capacity does not shorten their cycle life. On the other hand, their peak current should not exceed 2 times the rated capacity.

The first step onto choosing the batteries is to estimate the power consumption.

- PC with all 4 Webcams turned on: $12V \cdot 2.5A = 30W$ (current: 2.5A)
- 3 Motors with no load operation: $24V \cdot 0.6A \cdot 3 = 43.2W$ (current: 1.8A)
- 3 Motors with full load operation: $120W \cdot 2 + 80W = 320W$ (current: $320/24 = 13.3A$)

The first two estimates are in fact done in a real situation, since the material was already available. The PC consumption can be assumed constant over time. The

motors consumption, on the other hand, shall be inferior to the full load operation value (the arm would not be always up and down on normal operation).

Since space and weight are very tight restrictions on the design of this robot, some research has been done with the PSE company who have suggested 5Ah/16V packs with about 122x33x66 [mm] lithium batteries.

With a target value of 5Ah, the following autonomy can be estimated:

- PC: $5/2.5 = 2$ hours
- Motors under medium load: $5 / ((120 * 2 * 0.6 + 80 * 0.3)/24) = 5/7 = 0.7$ hours
- Motors under low load: $5 / ((120 * 2 * 0.4 + 80 * 0.2)/24) = 5/7 = 1.1$ hours

The 0.6, 0.3, 0.4 and 0.2 multiplication factors represent the estimate percentage of usage of each motor on normal operation. 120 and 80 is the power consumption of the motor and 24V the approximated voltage. A search operation of about half an hour covers a large enough area, accordingly to the fire-fighters, so these results are considered acceptable.

As it can be seen on section 6.8, separate motors and electronics batteries were used. Accordingly, the final configuration is:

- 1 x 14.4V/5Ah Rechargeable Lithium battery for the PC and electronics;
- 2 x 14.4V/5Ah Rechargeable Lithium battery for the three motors (associated in series, giving about 28.8V);

Two packs in series instead of a bigger pack is a preferable solution, because all packs are equal and easily interchangeable.

Each cell has the following characteristics:

Reference:	ICR33600	Unit
Nominal Voltage:	3.6V	[V]
Rated Capacity	5Ah	[Ah]
Height	60.1mm	[mm]
Diameter	33.0mm	[mm]
Weight	0.125	[Kg]

Table 19: Battery Cell Characteristics

Each pack consists of four cells associated in series. The following figure illustrates the placement of the packs.

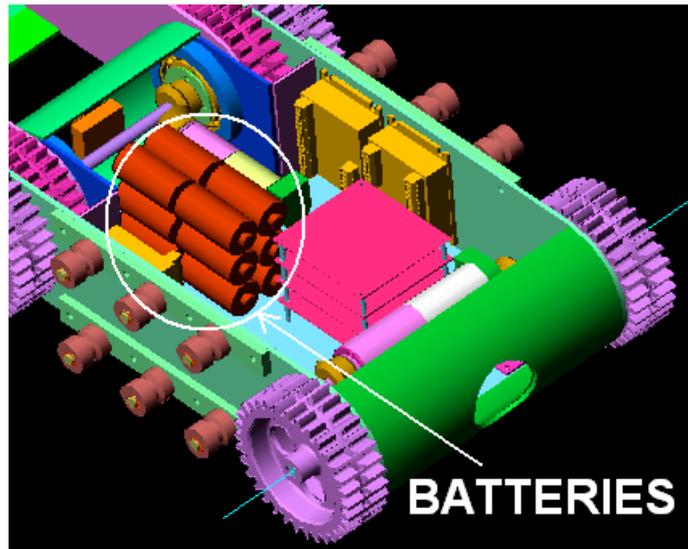


Fig. 21: Localization of the battery packs

8.4.8. PC104 and Remaining Electronics

The previous choices left only one possible location for the PC 104 and remaining electronics: between the arm motor / batteries and locomotion motors. The available surface is about 220x120 [mm], full robot height. This space shall accommodate the PC104 (Fig. 22, pink square), disks and remaining electronics.

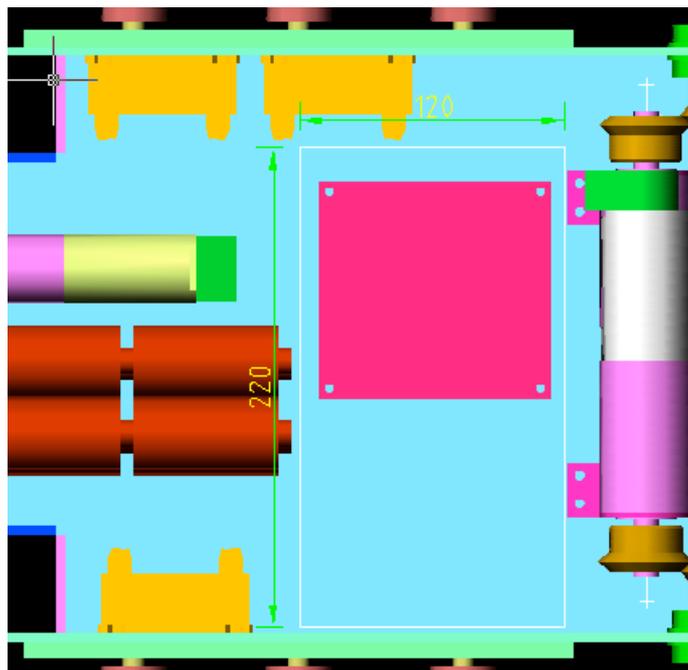


Fig. 22: Available space for the PC and remaining electronics

8.4.9. Docking Mechanism

The robot features an opening in the back, where a cable shall enter and be locked. The lock shall be strong enough to hold the robots weight, so that it can be lowered by the cable into a hole. The docking mechanism allows real-time docking / undocking of the cable, anywhere on its course.

The cable is flexible, but ends in a solid structure. This allows both unrestricted movement and a way to raise the cable so that the robot can grab it. Fig. 23 presents a possible structure that has the following advantages:

- Raise the cable (white conductor);
- Does not rotate easily, even if dropped in a non-horizontal plane;
- The power plug (represented in white) maintains its orientation approximately, independently of the way the structure is dropped on the ground;

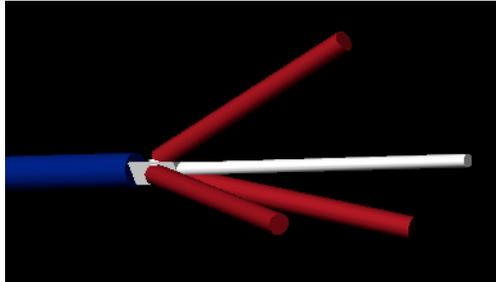


Fig. 23: A possible cable ending structure

The average distance from the ground to the end of the white conductor is still less than half the robots height. However, there is a clever way to overcome this limitation: if the frontal arm is pushed down, the main body rear comes closer to the ground, as represented in Fig. 24.

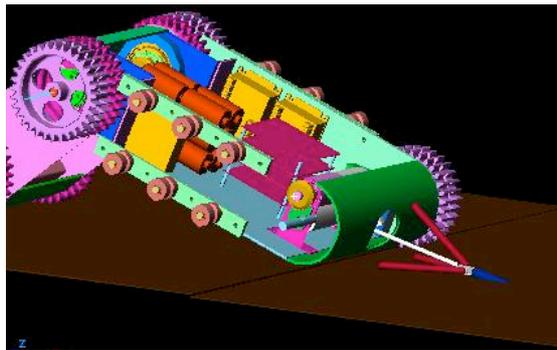


Fig. 24: Robot lowering its back to facilitate docking

The need of assuring physical contact of six Ethernet terminals once the cable is locked, for communications purposes, is quite demanding. We avoided the problem by using a wireless bridge / antenna at the end of the cable to communicate wirelessly with the robot. Although there is no physical contact, transmission is assured in the best possible conditions, since the distance between antennas is very small. The power transmission (DC voltage to power the robot) needs to have physical contact, however.

The type of antennas used in both the robot and the end of the cable is a key subject to the success of this solution. Since the robot is up-down invertible, an antenna cannot be placed perpendicular with the robots body, since it would be smashed once the robot flips. The use of two circular polarization antennas in the back of the robot, adjacent to the docking hole solves this problem, since they do not exceed 10cm height.

It is fundamental to have a rear camera to aid the docking process. The camera was placed at the robot centre, vertically aligned with the insertion hole, so that it is useful either when the robot is bottom-top or top-bottom. When the docking mechanism is not

attached and the operation scenario has no dust, the docking hole can be open and this camera used to have a view of the environment on the back of the robot.

The remaining details of the docking mechanism final structure are detailed on the construction Task, T6.

8.4.10. Gas Sensors

Since some of these sensors are very similar (TGS842/813), they share the same pins. Also, their time span is not very long, so it is pertinent that they are easily replaceable or removable, because it may not be desirable to use them at all times. For that purpose the board that holds them has sockets, so that they can be easily removable.

9. Task T4 – Equipment Characterization

On this chapter the acquired equipment is fully characterized. Since some of the devices characteristics were already presented on Task T1 that data is only presented if necessary in the context. The emphasis is on the experimental characterization of the sensors and other equipment and consequent circuit development. The boards are referred individually here. Some aspects concerning board interaction, calibration and experimental results once the sensors are assembled on their final configuration belongs to Task T7, electronic equipment assembly.

This chapter is divided onto two main parts. Subsection 9.1 details the computer (high-level) components and subsection 9.2 concerns the low-level electronics, namely a careful analysis of each of the sensors.

9.1. PC Hardware Test

9.1.1. Introduction

The computer was built using the following PC104(+) boards:

- A PC104 power-supply including battery level detection and UPS;
- A PC104+ Motherboard with soldered 128Ram, a PIII700, integrated graphics card, video grabber and network interface;
- A PC104+ four USB2.0 hosts board;
- A PC104+ PCCard/PCMCIA adapter, for use with the wireless board.

And the following additional hardware:

- A laptop hard disk with 40Gb capacity;
- A 512Mb compact flash memory to be used as an auxiliary hard disk.

The system was assembled and Red Hat Linux 9.0 was installed. This is a 2.4 kernel based Linux release. No problems were encountered to bring the basic system up and running. In the following chapters a detailed analysis of the setup of each individual board / feature is presented.

9.1.2. PC104 Motherboard

The following motherboard features have been tested and required no special care:

- Embedded graphics card: the requirement for a graphics card was merely to assist in debugging at development stage, as no display will be plugged to the board on normal situations. When developing the video capture driver, it was found that the video capture chip was tightly coupled with the graphics card. This is discussed in section 9.1.2.2.
- USB 1.1 Interface: our preliminary tests with one Webcam in each input showed no problems and Linux support is well established.

On the next paragraphs the remaining motherboard features are described in more detail.

9.1.2.1 Network Interface

The network interface has revealed itself as one of the weak points of the motherboard. Sometimes, after a reboot, the network adapter would stop being recognized by the OS, while the other features remained functional.

The problem arises when the battery is low (but not low enough for the UPS to notice, raising a shutdown). The network adapter is the first component to fail, probably due to the fact that it has to drive long cables (in our test assembly, the cable is 15 meters long). This is a rather unpleasant feature that must not be ignored. Nevertheless, since the robot always operates wireless, this problem does not affect it during normal operation, only when debugging with a conventional network link.

9.1.2.2 Framegrabber

The included framegrabber was based on a SA7111A chipset that performs the analog to digital conversion and feeds the resulting data to the graphics card memory, interacting directly with the CT69030 graphics card chip.

Two distinct sets of Linux drivers were found, but none of them were functional with our board. The work performed in order to achieve working drivers is described on Chapter 10.3. It was possible, at this stage, to ensure the hardware correct functioning in Microsoft Windows.

At hardware level, the circuit accepts up to three distinct composed video analog inputs, which are multiplexed to the video grabber. Only one input is used (input 0) to receive signals from the thermal camera. The composite video signal requires only two wires, ground and signal, that are shared with the monitor connector on the PC104.

J25 VGA monitor (CRT-Signals)

J25 Header			15 pins HiDensity DSUB	
	10 Pin -M	Signal	Pin	Signal
	Pin 2	VGA red	Pin 1	Red
	Pin 4	VGA green	Pin 2	Green
	Pin 6	VGA blue	Pin 3	Blue
	Pin 8	Horizontal Synch	Pin 13	H-Synch
	Pin 9	Vertical Synch	Pin 14	V-Synch
			Pin 5 + 11	Bridged
	Pin 1	Ground	Pin 5, 6, 7, 8	Ground
	Pin 3	Video Input 3		
	Pin 5	Video Input 2		
	Pin 7	Video Input 1		
	Pin 10	Video Clock In		

Fig. 25: Video Capture Motherboard Pins

9.1.2.3 Serial (RS232) Interface

The first step was to test and characterize the serial port available in the RAPOSA computer. The serial interface is a piece of well-proven hardware that has not changed for several years. So, the tests carried out were expected to give the same results in any computer.

Experimental tests have revealed that each read operation takes at least 10ms, even if the received number of bytes is a multiple of the USART buffer size.

This is believed to be a Linux serial driver limitation (or, more generally, a Linux I/O read operation limitation). Each time a blocking read operation is done (one that waits until some data is available) the system sends the current process to sleep. The Linux scheduler awakes this process in at least 10ms (depending on the number of other process running, it may be more).

Supposing that the read operation just followed a write operation, the following calculations estimate the number of bytes transmittable on that time frame:

Data format: 1 start bit, 8 data bits, no parity, and 1 stop bit: 10 bits per byte.

At 115200 bauds ~ bits/s we have:

$(115200\text{bauds} / 10\text{bits}) * 10\text{ms} = 115,2$ bytes per read.

For some applications (modems) this might be considered a small amount of data, but one must consider the type of operations/data being exchanged:

- transmit commands to the motor controllers;
- Receive feedback from the motor controllers;
- Receive data from the sensors;
- Activate lights.

As none of the motor controllers or sensors sends/receives floating point values (in fact, they are all 1 byte values), the 115 limit exceeds largely our requirements.

Waiting periods of 10ms gives about 100 send/receive data pairs per second. This is small enough time to accommodate even the fastest motor response and no other sensor has such a high response rate.

9.1.3. PC104 Power Supply

The acquired board, besides doing the necessary DC – DC conversion (12...30V to 12V, 5V, -12V), also provides an UPS / Battery monitoring feature, using the battery to power the PC instead of mains power if this fails, charging the battery otherwise. If the power is coming from a 12V battery, the board automatically monitors its level and issues a timed shutdown of 127 seconds once it goes under a predefined value (10.6V). Fig. 26 illustrates a typical work case.

This is a fully automatic process, with no user intervention. It is however possible to monitor the battery status through a series of I/O ports and even program the board to raise an interrupt on one (or both) of the following conditions:

- Battery low;
- Power mains low.

This allows the operating system or user program to be notified immediately of the power breakdown. Being a very specific board, there were no drivers available for Linux to take a given action once one of the previous conditions aroused.

A driver was developed as part of Task T7, so further details can be found in section 10.2. It should be noted however that this driver is only useful in preliminary stages of the robot development, when it is being powered by conventional NiMH 12V batteries. Lithium batteries, having a higher voltage range, never trigger the low voltage warnings when going low, since the electronics of the batteries shuts them down and their nominal voltage is at least 2V higher.

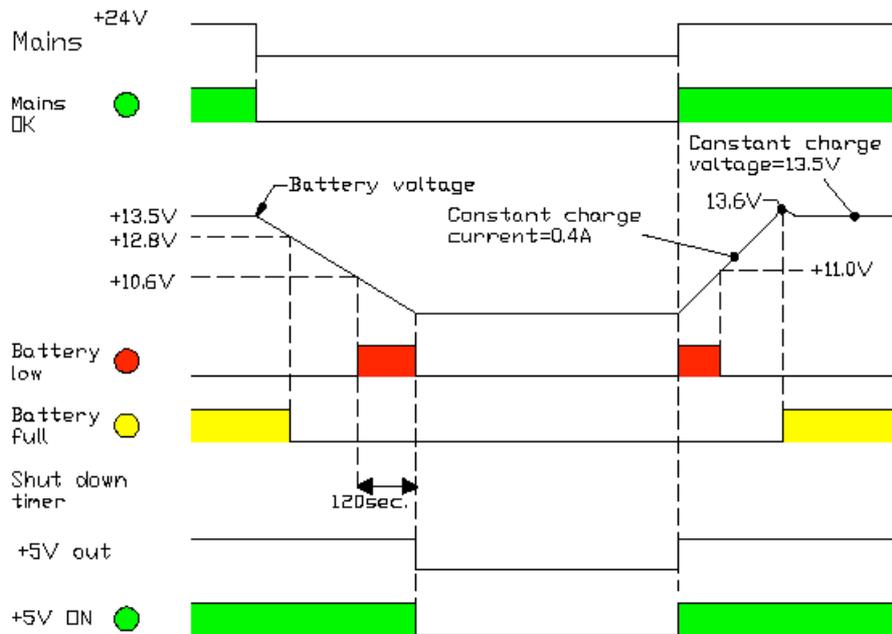


Fig. 26: Battery Status

9.1.4. PC104 four USB2.0 hosts board

The main purpose of this board is to allow the use of up to 4 USB Web Cam's, without them mutually interfering / using each other's bandwidth. Whereas in a desktop Pentium IV with two USB jacks per channel mutual interference between those two cameras was a problem (in Microsoft Windows), that problem did not happen with this board.

Using a GPL software named *palantir*[2] the following setup was successfully tested:

4 WebCams transmitting through the network interface to a remote PC simultaneously, each at 320x240x10 fps. The frames per second were reduced to about 7fps if all cameras were recording agitated scenes.

The framerate depending on the type of image is clearly a limitation of the processor and the network bandwidth and not the USB2.0 board, since the processor has to re-encode the images prior to sending them through the network interface.

9.1.5. PC104 PCCard Adapter

Linux recognizes this adapter. The wireless PCCard was used without problem, so this component was completely transparent to the user, requiring no special configuration.

9.1.6. Laptop Hard Disk

The Hard Disk has presented no problems so far and it has delivered good performance, due to its 5400 rpm speed, instead of 4200rpm normally found on this kind of hard disks. In this test stage, the disk was mounted physically in a diagonal position without any sort of problem.

A partition for Microsoft Windows was created, in order to test the framegrabber and the remaining disk space was separated on Linux partitions.

9.1.7. Compact Flash Hard Disk

Using a Compact Flash to IDE converter, it is possible to use this 512Mb non-volatile memory as an auxiliary hard disk. Our primary intent is for it to hold the boot code for

the operating system, so that even if the primary hard disk fails due to vibration, the system can still operate.

The first practical tests featured the following configuration:

- IDE Master: Laptop Hard Disk;
- IDE Slave: Compact Flash Memory;

This configuration did not function properly, as the compact flash memory data became corrupted. The alternative configuration:

- IDE Master: Compact Flash Memory;
- IDE Slave: Laptop Hard Disk;

works as expected.

The /boot partition was successfully mounted on it. Further advances may include a stripped /root mounting, in order to achieve full conventional hard disk independency.

However, after some use, the compact flash started to behave erratically, giving lots of errors and effectively slowing down the system. At this point it was preferred not to use it, so that the remaining development was not compromised.

9.1.8. USB Webcams

The web cams worked as expected. Linux support was available through a partial GPL license. The non GPL part was required to use higher resolutions and framerates. Its use, however, was still free, only the source code was not available.

9.1.9. Thermal Camera

The thermal camera has a four wire interface. Two for power, ground and 12~14.4V (electronics battery) and two for signal, ground and composite video output. These connect on the mainboard framegrabber, as seen on the 9.1.2.2. This assembly was successfully tested. There is an extra set of wires coming out of the camera, allowing to control features such as gain, level and polarity control and such. But the manufacturer failed to provide enough documentation, so those features are not used for now.

9.1.10. Wireless

The wireless link was tested. The communications data path is represented in Fig. 27: the laptop where the operation console runs connects to the docking cable, which comprises an Ethernet crossed cable, connecting to the SMC2670W Ethernet-to-Wireless bridge adapter [3]. This very simple device can only operate in infrastructure mode, meaning that it can only connect to an access point. Consequently, the *Senao* Wireless PCCard [4] has to operate in access point mode. This topic is further developed in section 10.1, concerning low-level drivers.

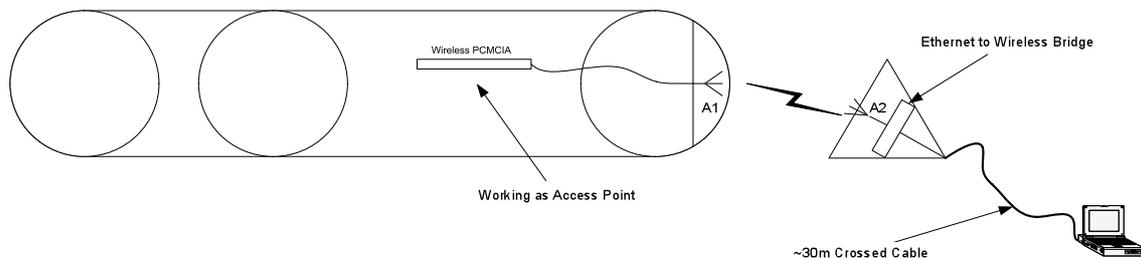


Fig. 27: Wireless Communications

Given the space restrictions of the robot and since it can operate flipped, it is impossible to have regular lengthy point up antennas. A D-Link DWL-R60AT microstrip circular polarization antenna [5] was used. The circular polarization reduces the interference from multipath transmission. The antenna features a 6dBi signal gain and has a very convenient size of 58.4 x 58.4 [mm]. This way, it can be easily assembled parallel to the robot sides, independent of its vertical orientation.

9.2. Electronics

9.2.1. USB to Serial interface

Although the RAPOSA PC still features two serial ports, a future upgrade to the PC may use a motherboard without them. But every modern computer features USB (Universal Serial Bus) ports. The USB protocol is more advanced and flexible, allowing an increase on the number of ports through the use of inexpensive hubs. On the other hand, it also introduces unnecessary complications, like a more complex protocol, the need of special drivers' development for every new device, etc.

There is a compromise solution: a USB device that emulates a serial port. With the adequate set of drivers (freely available for Linux) this USB device is similar to any other serial port. At low-level, it provides the RS232 5V signals that can be directly interfaced with the PIC.

9.2.1.1 USB/Serial FTDI chip

The USB/Serial chip used was a FTDI FT232BM [6]. This chip emulates a serial port featuring 384 bytes receive / 128 bytes transmit buffer memory and a 16 ms timeout. In order to differentiate more than one of those devices connected to the same bus, an EEPROM (Electrical Erasable Programmable Read Only Memory) is attached to each of them allowing assigning it a unique serial number and saving individual chip settings.

A commercial module (DLP-USB232M[7]) was acquired and tested. It features:

- FT232BM;
- EEPROM;
- 6MHz Oscillator.

It is connected to the remaining circuit by means of a conventional DIP socket simplifying the assembly considerably.

Experimental tests revealed that the 16ms timeout have a predominant effect over the 10ms process pre-emption time of Linux, meaning that a blocking read always waits a minimum of 16ms after the command has been sent. With the aid of an oscilloscope, we can see the PIC responding immediately, but the PC only sends a new command 16ms after. So, the effective number of write / read transactions per second is around: $1s / 16ms = 62.5$. This value is still enough for our applications, as it provides a higher frequency than the maximum camera framerate (30fps).

A block schematic is presented in Fig. 28:

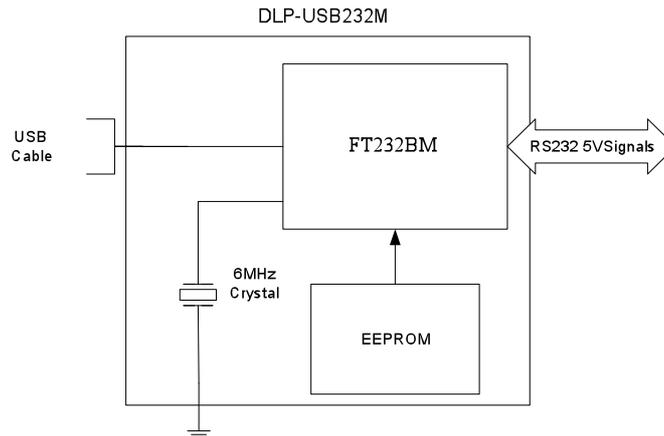


Fig. 28: DLP-USB232M USB to RS232 Bridge

9.2.1.2 Serial Interface

Since the motherboard still featured a serial interface, it was used. The adaptation circuit is fairly simple, only requiring a voltage level adaptation, between the 12V serial port values available at the PC serial port and the 5V required by the PIC. A MAX232 chip features a maximum baud rate of 115200 bauds, enough for our requirements.

The block is represented in Fig. 29.

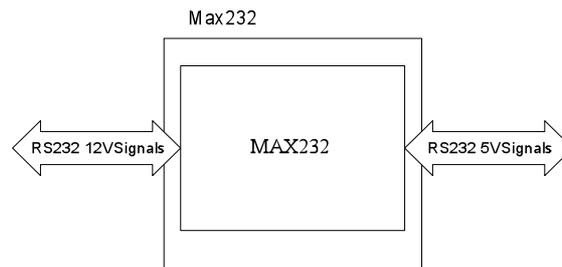


Fig. 29: RS232 level conversion

9.2.1.3 Serial Interface Block Information

To conclude this section, two distinct ways to achieve the same functionality were presented. Both have the same interface concerning the PC and the only noticeable differences are:

- The standard serial ports are referred as /dev/ttyS0, 1, 2; the USB serial ports are referred as /dev/ttyUSB0, 1, 2.
- The time required for a read operation is a maximum of 16ms on a USB serial device and 10ms on a standard serial device.

To the PIC, both blocks behave identically, providing a RS232 5 V signal. In the following text, the term “serial interface” (or link) is used to identify this generic block, independently of its implementation.

9.2.2 Motors Control and Monitoring

The brushless motors require adequate controllers. The Maxon controllers used have the following set of relevant characteristics [8]:

- Speed control through an analogue voltage from -10 to 10V;
- Current limiting through an analogue voltage from 0 to 10V;

- Power stage enable / disable through a TTL compatible bit;
- Speed feedback through an analogue voltage from -5V through 5V.

The current limiting through the analog voltage input was not used, since no dynamic control over this parameter is needed. This way, the current limit is adjusted through the internal potentiometer on the controller, at the maximum possible within the motors thermal range.

Although all the motors give a velocity feedback, this value is only valuable in the two locomotion motors, since they are speed controlled. The arm motor, on the other hand, is position controlled. An analog potentiometer connected to the arms axis translates its position into an analog voltage. So, at low-level, all three motors return an analog voltage that is feed to the PIC ADCs (analog to digital converters) representing the measurement of interest. The PIC converts this voltage to a digital value that is returned to the PC and only then, the significance of the value is taken into account (meaning that the PIC does not use these values).

Since the speed feedback does not provide voltage levels compatible with the PIC, additional OPAMPs are required for the conversion.

There are also individual enables to each of the controllers, so that each power stage can be turned off, for power saving / guaranteed robot immobilization.

The block schematic of this circuit is present in Fig. 30:

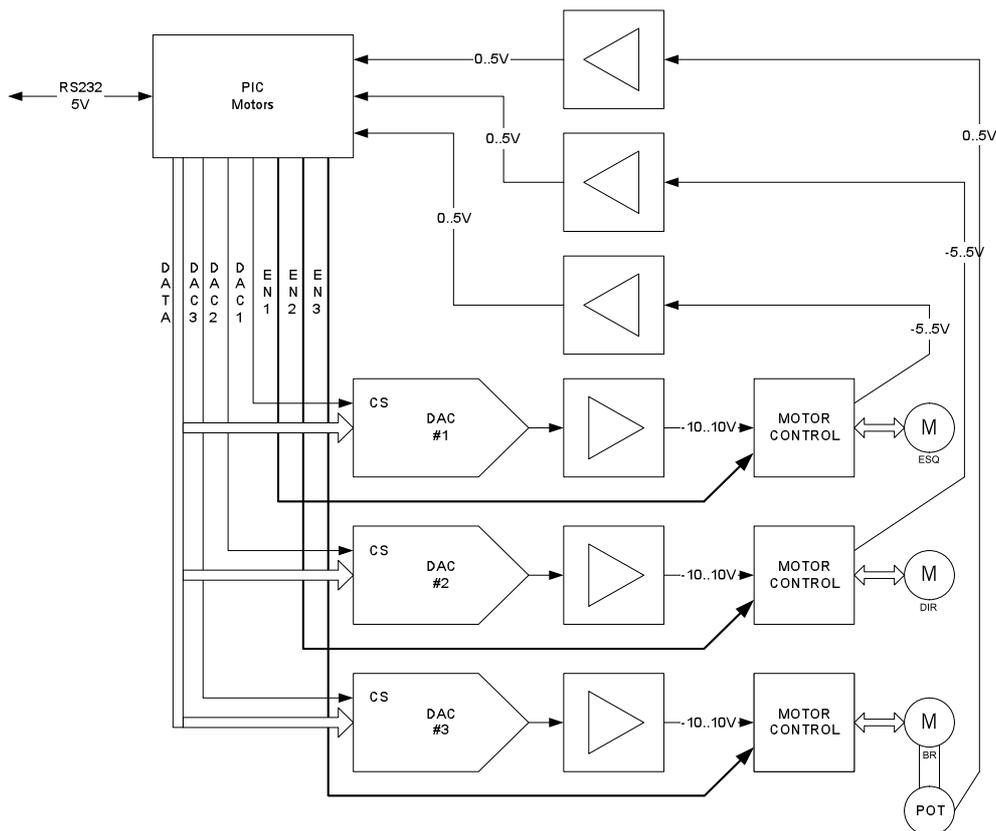


Fig. 30: Motors Control Architecture

A circuit was built featuring all functionality mentioned above and it worked flawlessly.

9.2.3 Gas Sensors

As mentioned previously (section 8.4.10), modularity is a key concept to the gas sensors, since it may not be possible / desirable to use them at all times. So, it is preferable that the interface circuits be sensor independent, leaving the differentiation to the software level.

All of these gas sensors require heating.

9.2.3.1 Generic Interface (TGS8XX)

From the four gas sensors selected, three of them are very similar both in mechanical and electrical interfaces: TGS813, TGS825 and TGS842. They are analyzed first, the remaining Carbon Monoxide sensor being analyzed later.

A summary of the datasheets and a detailed analysis of the most relevant parameters to all sensors are showed on Appendix 2. The main choices and results are presented here.

The sensing element on each of these sensors operates at high temperatures, hence the need for a heater circuit. The current required to excite the heater varies according to the sensor, but it never exceeds 200mA.

However, if the three sensors are to be used at once, the total current may reach about 600mA, a value large enough to affect the robots autonomy. To reduce power consumption each of the sensors may be individually activated. The following circuit delivers the necessary current without lowering the voltage below the 4.8V, as required:

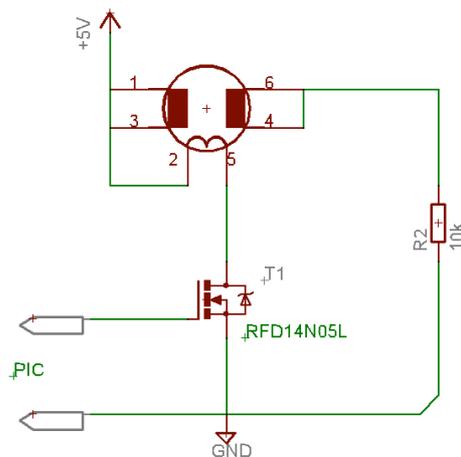


Fig. 31: Generic Gas Sensor Circuit

The RFD14N05L N-Channel Power Mosfet has a very low on resistance: 0.1Ω and it is specially designed to be driven by a 5V logic level signal, coming right from the PIC.

In Appendix 3, it is proven that an 8 bits resolution is enough, based on the maximum achievable R_S for each of the sensors. R_L was chosen to be equal to $10k\Omega$, a value that provides better sensibility on lower gas concentrations.

The values of these sensors are sent to the PIC through an analog multiplexer (CD4051B) that features a maximum on resistance of 1050Ω . This gives a worst case scenario output impedance of about $8.4k\Omega$, still below the PIC maximum input impedance ($10k\Omega$).

9.2.3.2 TGS8XX Preliminary Tests

All these three sensors have a minimum preheating period of at least two days before use, if previously stored for a long time. This procedure is necessary in order to get a stable and predictable response. After this period a calibration, using reference gases at the specified concentrations, must be performed in order to fully characterize the sensor response. At IdMind there are no facilities to perform such calibration that is only possible at the RSBL headquarters during Task T9. Since the electronics can process all possible values the gas sensor can produce, the calibration can be achieved at high-level, by software.

Nevertheless, a rough testing was performed. Both TGS813 – Detection of combustible gases and TGS842 – Detection of Methane (mostly sensible to this gas, but detects other combustible gases too) showed differences on V_{R2} over 2V when exposed to the gas of a cigarette lighter.

The TGS825 – Hydrogen Sulphide also showed a very slight variation when exposed to commercial available alcohol.

Unfortunately the MosFet had a higher resistance than expected at such a small current (<200mA), so the measured heater voltage was around 4.7V, below the recommended value. This means that the sensor predicted characteristic may be a little shifted, but then again, only after experimental characterization and calibration it is possible to measure reliable values.

9.2.3.3 Carbon Monoxide TGS2442

The carbon monoxide sensor to be used is a TGS2442, with an estimate 1 second response time. The analysis on Appendix 3 shows the following extreme resistance values:

Sensor	Gas	Min R_S	Max R_S
TGS2442	Carbon Monoxide	900 Ω	158 k Ω

Table 20: Carbon Monoxide Range of Resistances

The recommended measurement circuit is represented on Fig. 32, along with the required timings. Some other restrains include:

Load Resistance : $R_L \geq 10k\Omega$.

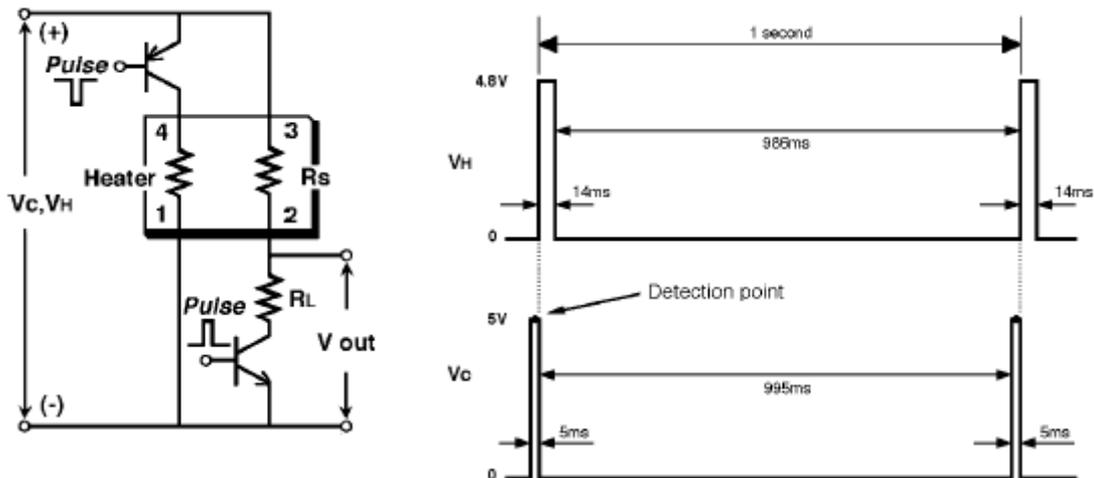


Fig. 32: Measurement circuit and timings

The circuit has three distinct stages:

- During 14ms the heater transistor receives a negative pulse on its base, applying around 4.8V to the heater;
- The heater is shutdown (heater transistor cut-off by a positive voltage) for 986ms;
- 951ms after the beginning of the heater cycle (at this time the heater is off), the load transistor receives a 5ms positive pulse, thus applying circuit voltage to the sensor resistance. At half of this 5ms time frame is the adequate instant to sample V_{Out} .

By the end of the 5ms, the cycle repeats.

Given that the output resistance of this exceeds $10k\Omega$, an OPAMP buffer circuit is used to drive the PIC analog input.

9.2.3.4 Carbon Monoxide TGS2442 test circuit

A dedicated electronic sensing interface was developed (see details in Fig. 33) and worked as expected.

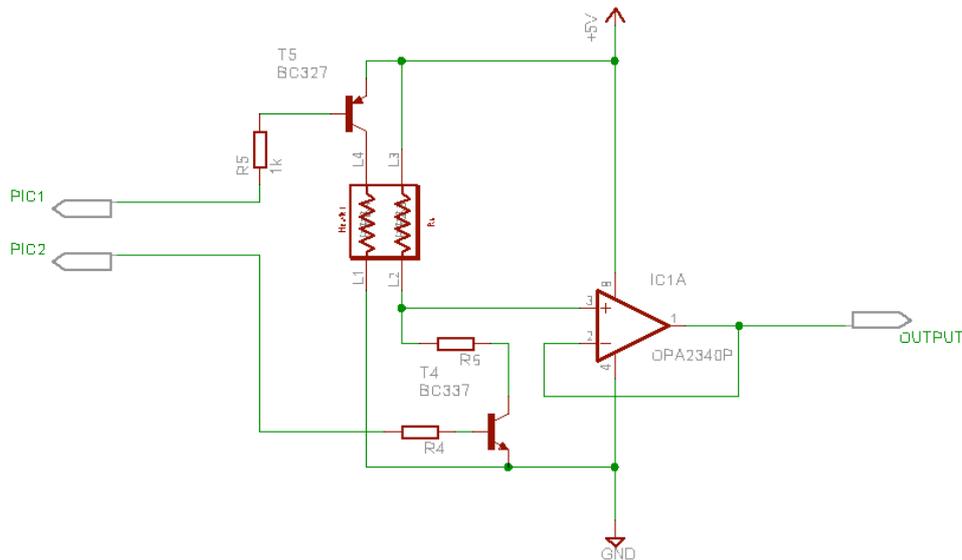


Fig. 33: Carbon Monoxide Sensing Circuit

As previously analyzed, the heater and load transistor have different timings, this justifying the two distinct control inputs. Since the recommended load resistance ($R5$) is rather high ($> 10k\Omega$), a buffer OPAMP is used. The final calibration is done at the fire fighter headquarters, on Task T9.

9.2.4 Temperature and Humidity Sensor

The temperature and humidity sensors from Sensirion, family SHT1x/7x, gives a measurement of both relative humidity (RH) from 0% to 100% and temperature measurement from -40° to 120° in a single sensor, using a digital interface. The sensor is very small and it has only four wires - two for power and two for serial communication.

The sensor datasheet can be found on [4]. The most relevant data is on the following table:

Temperature	Min	Typ	Max
Range	-40 °C		123.8°C
Resolution		0.04°C	
Accuracy	± 0.5°C		± 3°C
Response Time	5s		30s
Relative Humidity	Min	Typ	Max
Range	0 %RH		100%RH
Resolution		0.5%RH	
Accuracy	± 3.5%RH		± 5%RH
Response Time		4s	

Table 21: Sensor characteristics

A few remarks:

- Over the -10°C up to 60°C the temperature accuracy maintains itself under ± 1.5°C;
- The accuracy of the humidity value is at its best between 20% and 80% RH.

This sensor should be positioned near the gas sensors. Given its small size, this is not a problem. This way, the values read may be used to do the temperature and humidity compensation for the gas sensors, giving more accurate results (on very dry environments, for example, the gas sensor sensibility is severely reduced, what would lead to a dangerous underestimation of the gas concentration).

Although not featuring a remarkable response time or accuracy, this should be enough for our application. The gas sensors have a much slower response time (a maximum of 30s).

The sensor digital interface resembles I2C, in the fact that is a two wire serial communication, differing mainly in the start sequence. The data signal requires a 10k pull-up resistor. The study of its communication protocol belongs in section 10.7.1, since it involves PIC programming.

9.2.5 Lights

A single Luxeon 1W white LED was tested with a webcam on total darkness, the results were quite good, the camera got perceptible images. No more than a LED is required for each webcam.

These LEDs feature a V_{ON} between 3.42V and 3.99V, the maximum driving current is around 350mA. The following circuit has been tested:

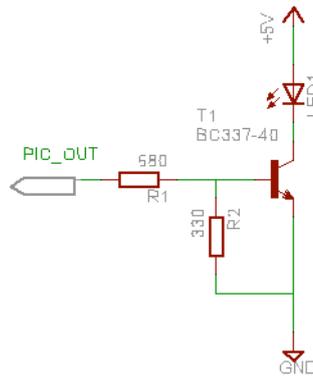


Fig. 34: Led Driver

The circuit shall be driven directly from a PIC output and if a PWM signal is applied, the led intensity can be regulated.

9.2.6 Accelerometers

Having presented the ADXL311 datasheet on 6.6.2. Linear Accelerometer as , the analysis of this sensor is herein concluded, so that a measurement circuit can be devised.

The zero G voltage uncertainty of this sensor, due to the manufacture process, has the same order of magnitude of its dynamic measurement range. In order to avoid a broader range of detection (having the cost on resolution, given the fixed 8 bits discretization), a local potentiometer can provide a zero adjustment for each axis (zero acceleration exactly $V_D/2 = 2.5V$).

The variation on the sensor sensibility could also be compensated with the use of another potentiometer, but the extra space required makes it an undesirable option.

The sensors are calibrated for their nominal sensitivity value. If the experimental sensibility exceeds the nominal value, we loose the ability to measure 2G, as the measurement saturates on 5V (or 0V) at a slightly lower acceleration. But since our goal is to measure only gravity, this limitation is not serious.

The following calculations present the effective resolution of the sensor, with 8 bits.

Voltage interval to measure:

$$\Delta V = (2 - (-2)) \cdot 0.167 = 0.668 \text{ V}$$

$$\text{Amplified } \Delta V = 0.668 \cdot (220/32) = 4.59 \text{ V}$$

$$8 \text{ bits acceleration distinct values: } 4.59 / (5/256) = 235$$

Since there is no need to measure the robot instant accelerations and vibrations, the bandwidth of the sensor shall be as small as possible, so that preferably only gravity (a constant acceleration) is measured.

$$f_{-3dB} = \frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 32000 \cdot 1 \cdot 10^{-6}} = 4.97 \text{ Hz}$$

With a $1\mu\text{F}$ capacitor in each output a $\sim 5\text{Hz}$ bandwidth is obtained.

If this value proves insufficient, the capacitor can be adjusted accordingly. The final circuit is present in Fig. 35:

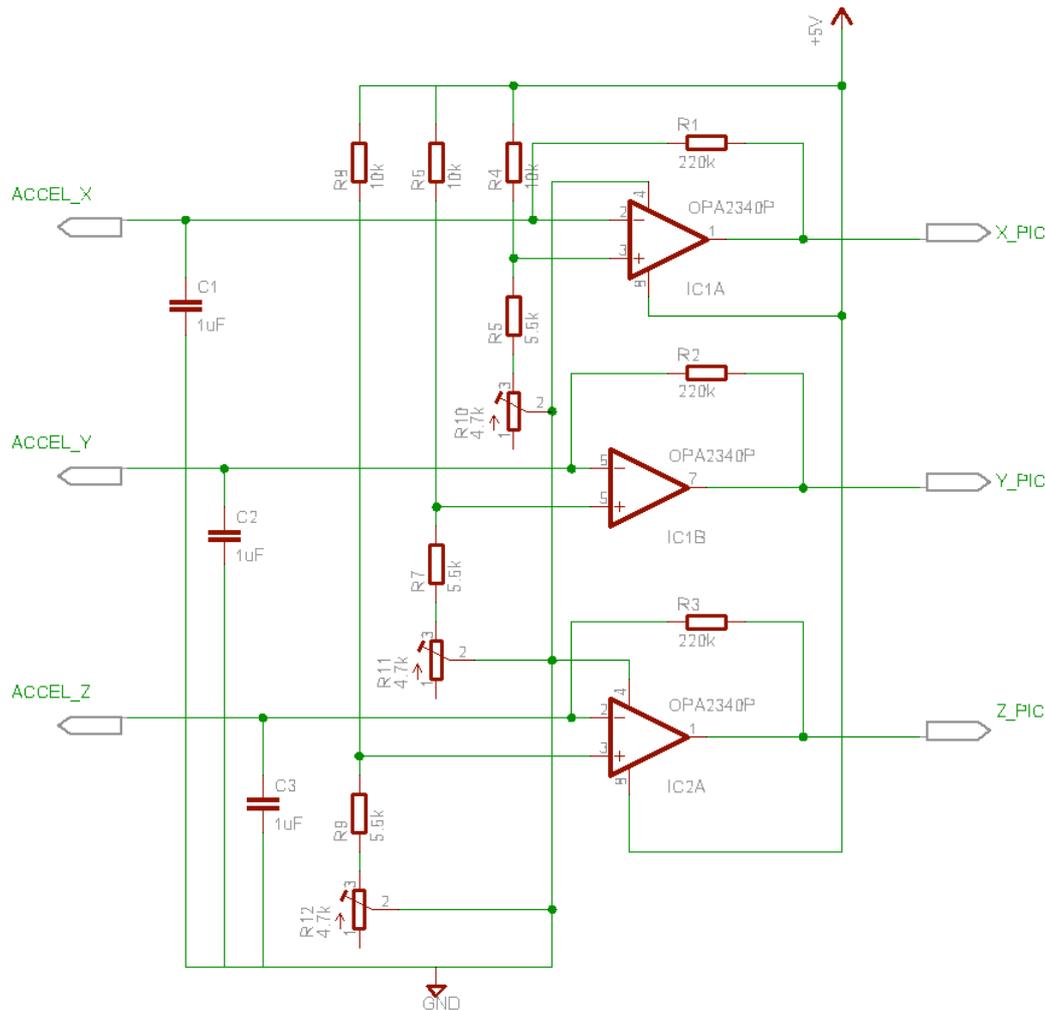


Fig. 35: Accelerometer Adaptation Sub-Circuit

9.2.7 Infrared distance measuring sensors

IdMind has a long experience with these Sharp sensors and has determined that they do not require any signal processing at electronics level, besides two polarization resistances. This simple assembly connects directly to the PIC analog input or an analog multiplexer.

9.2.8 Batteries

The Li-Ion batteries were delivered in May 2004. Their energy density fulfils expectations. Full load operation could not be tested at this stage, but there were satisfactory results on horizontal robot movement. The PC and electronics consumption estimates proved good. The motor batteries lasted for at least two hours.

9.2.9 Robot Power

The robot features two independent sets of batteries and it may also receive power from the cable. Commutation between both is performed by a simple diode scheme, for the PC and electronics power (Fig. 36). If mains power is present (through the cable) and the battery is below the threshold level, the UPS board tries to feed a 400mA constant current to the battery in order to charge it. This would be a good feature for lead acid batteries, but Li-Ion batteries cannot be charged using this simple method.

Since this current cannot be injected onto the battery, the following simple circuit shall protect the battery:

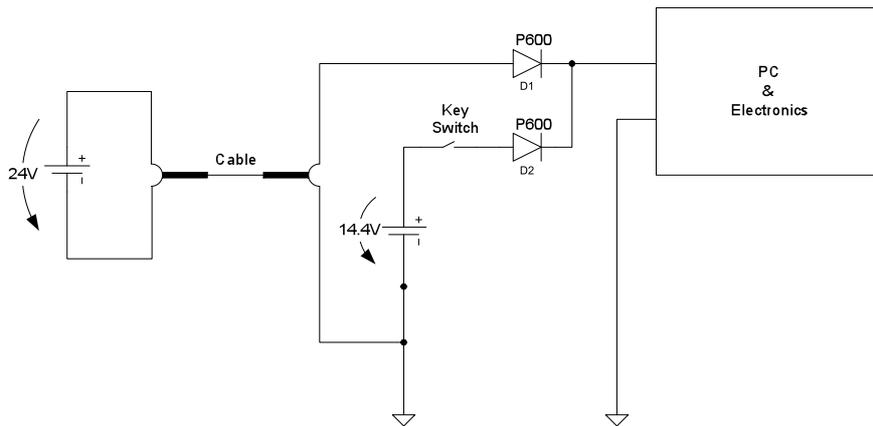


Fig. 36: PC Power Supply

Diode D2, beyond aiding in power mixing, protects the battery from the UPS charging current.

As it is implicit on the figure, the UPS features two distinct power inputs, one for the battery and other for the main power. This way, no DC-DC conversion from the 24V cable voltage is required, since the UPS board has this feature.

For the motors a more conventional solution, based on a relay (Fig. 37) is used. If neither the manual enable nor the console batteries enable are active, the robot gets its locomotion power from the external cable. When operating with batteries, if the operator wishes to enable the motors, it must turn on the manual enable button on the robot and switch to batteries power on the console.

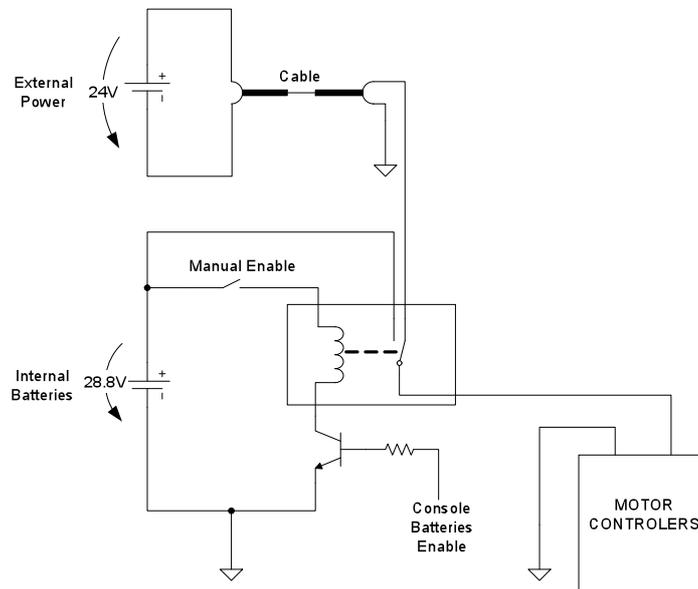


Fig. 37: Motors Power Supply

At any moment, in remote operation, can the operator flip between cable or battery power, as long as the manual enable is active. The manual enable is an extra mechanical check (and visual, since this switch has an enabled light) that the batteries are not being inadvertently connected, thus depleting their charge.

An optional capacitor may be used at the exit of the relay in order to smooth the transition between different sources of power.

As long as the motor batteries have enough voltage it is possible to activate the relay. If that is not the case, then it makes no sense to use the batteries.

The remaining electronics requires 5V for the PIC microcontrollers and 9 to 30V for the thermal camera. In the following table the expected power consumptions on the 5V rail are listed:

	Current[mA]	Current[A]
Gas Sensors	3 x 200mA + 250mA	0.85
Lights	3 x 350mA	1.05A
Docking Motor	~100mA	0.1
Webcam Servos	2 x 250mA	0.5
PICs	4 x 25mA	0.1
Total		2.6A

Table 22 : Electronics current

Given these requirements a 3A/5V output DC/DC voltage converter is used, PT6302[9]. The following circuit is used:

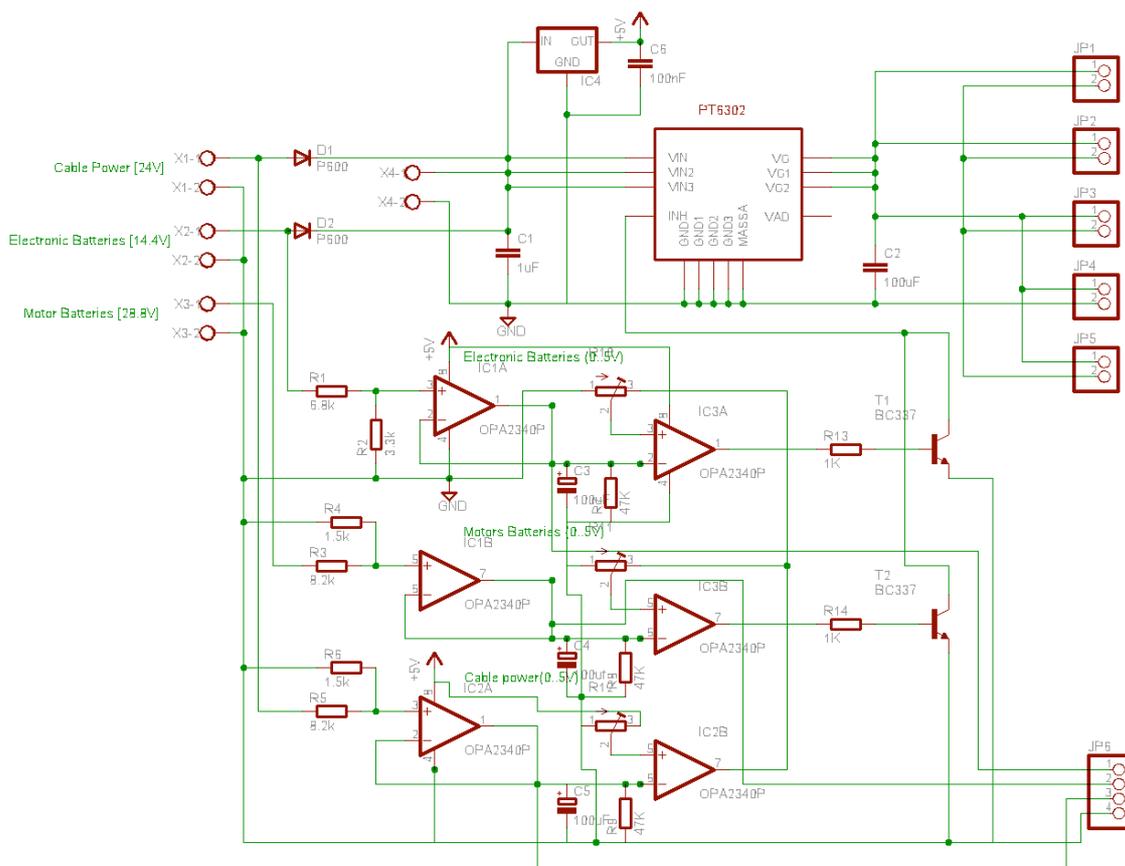


Fig. 38: DC-DC and voltage monitoring electrical circuit

D1 and D2 diodes on Fig. 38 are the same that on Fig. 36.

The battery voltage monitoring circuit is a simple set of two amplifiers, with the necessary voltage dividers and a separate 5 volts regulator to power it. This circuit current is negligibly small (less than 5mA). That sub-circuit is shown in the following figure:

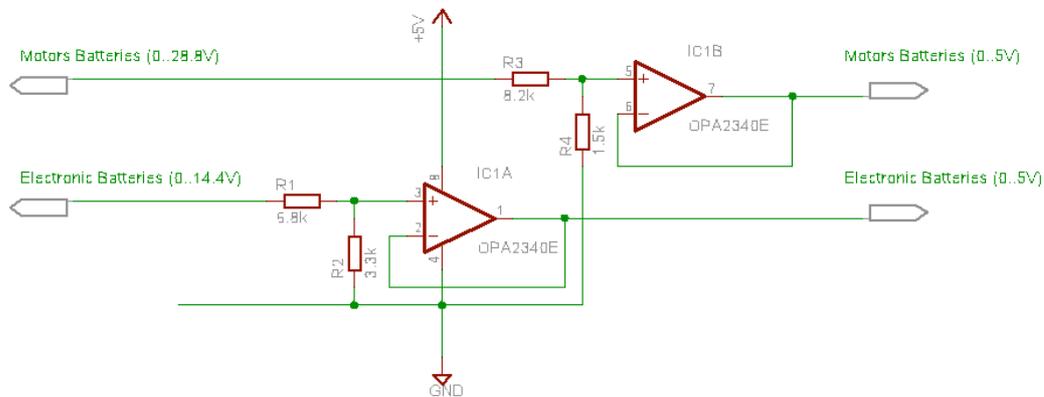


Fig. 39: Voltage Monitoring

The maximum voltage measurable is about 30V for the motors and 15V for the batteries. This allows estimating over voltage. A value representing the battery voltage scaled down to 5V is sent to the PIC, for relaying to the PC. An OPAMP is used in order to protect the PIC against over-voltages, because the OPAMP does not exceed the 5V output.

The second level of OPAMPs is used to define the level at which the batteries are considered low, using a potentiometer. In order to also shut off the power circuits and to provide voltage feedback for the PIC, a 4 pin socket is available to connect to the PIC board. These OPAMPs outputs connect to the DC-DC converter inhibit function that allows it to be shut off. This input is activated on the following condition:

$$\text{Shutdown} = \overline{\text{CablePower}} \cdot (\overline{\text{ElectronicBatteries}} + \overline{\text{MotorBatteries}})$$

If there is no power connected through the cable and one of the batteries fails, the whole DC-DC converter is shut off, to protect the Li-Ion batteries. The OPAMPs however need to keep their power to sustain this condition, so a secondary low power regulator (LM7805) is used. This low power circuit has an irrelevant current that does not affect the battery autonomy at a small and medium term (some hours).

9.2.10 Webcam Servos

The webcams servos have three pins: ground, 5V power input, from where they drain their energy and the PWM input. Since this PWM signal is just for control, it can be taken directly from the PIC, from one of its PWM outputs.

10. Task T5 - Interface Software Development

On this chapter the low-level software, both at electronics level (PIC programming) and at PC level (drivers) is discussed.

10.1 Wireless Communications

As seen on 9.1.10, the wireless PCBoard should work in access point mode. This could be a problem, but the chosen wireless PCBoard is based on a Prism 2.5 chipset that can work as a limited access point with the appropriate Linux device drivers.

This option was carefully investigated. It required a Linux Kernel update, since the available driver did not compile under the current installed version. In order to avoid further incompatibilities, the latest 2.4 (2.4.26) kernel was installed. The functions performed as an access point proved to be enough for successful communications between the two devices.

After the installation of the new kernel, the driver performed as expected and only some manual initialization was required.

10.2 UPS Device Driver

The UPS device driver is only useful at the test stage, when NiMH batteries are used to power the robot. Nevertheless, it protects the PC shutting it down prior to battery dropout and thus maximizes its lifetime.

The following I/O address range allows the supervision of the battery and main power status and to program the triggering of interrupts:

- I/O Port Range: 250h-257h.

The following useful operations are available:

- Monitor battery and mains power status (252h);
- Program IRQ number and the condition(s) that raise it (253h);
- Monitor seconds to shutdown (254h).

The installation routine of the driver searches one interrupt that it is not yet occupied with another peripheral, from the set of seven possible interrupt lines the UPS may trigger. This way the driver is simplified, since once the interrupt is issued the driver knows that only the UPS has triggered it. An analysis of the current hardware connected to the main board shows that there are plenty of available interrupts.

The driver monitors both the level of the battery and mains power. If they both fail, the driver initiates a controlled shutdown in one minute. The user is informed immediately, so he/she can save his work safely.

For some unknown reason the Linux shutdown command does not power off the computer (probably the motherboard does not conform to the ATX norm). But at the end of the 127 seconds UPS automated shutdown the computer is already on a "ready to shutdown" state.

The device driver was developed and it is targeted to the version 2.4 kernel.

There is interest in shutting down the PC even if the power status is OK. By programming an adequate value (about 20 seconds) on the shutdown script as its last action, the UPS becomes responsible for effectively shutting down the PC in all cases.

10.3 Framegrabber Device Driver

The acquired motherboard features a video capture chip, a Philips SAA7111A, responsible for the analog to digital video conversion. This chip interfaces directly with the graphics card chipset, a Chips and Technologies CT69030, through which the capture chip is controlled by I2C. The CT69030 receives the digital video and stores it directly into the video memory, at a programmable address.

An adequate driver provides a linear access to the captured video (by memory mapping) and allows changing some of the capture parameters. A driver from Alessandro Rubini [10] that meets these requirements was found and after some tweaking it compiled ok on the version 2.4 kernel.

The next step was to adjust:

- The analog channel used by our test camera (three were available);
- The RGB15 colour mode;
- The interlaced image feature.

All these settings were different in our hardware, so a complete driver analysis was required. Since the motherboard manual is rather poor, some of the settings were determined by trial and error. At this stage, a functional driver was adapted from the previous code.

The next step was to add Video4Linux functionality. This includes access to the memory poll, both by read operations (slow) and direct memory mapping – mmap (fast). Also, double buffering was enabled to assure a more smooth video capture.

The video driver actually supports a minimum subset of the Video4Linux interface that allows it to be used by the more common Linux applications (xawtv, ffmpeg) just like another video source, without specific initialization.

10.4 Commands and Data stream Protocol

As previously defined the communication between the console and the robot is made through a UDP socket using the following packet style.

Agent ID	:	Agent Data	:
----------	---	------------	---

Each entry can be a string, representing integers, floats, or even strings, tables, etc. The Agent ID is an integer while the data is Agent dependent, meanwhile we only control the PIC Agents. The Agent Data is very similar to the low-level PIC Protocol.

This protocol is symmetrical, meaning that the frames are the same in both communication directions. Most of the controlled agents onboard the robot are for controlling the PICs, they have similar AgentData, namely in the agent data usually are requests for sensors readings and commands to write values on the PIC tables. The operation Console uses the PIC protocol facility to write or requests more than a value in each packet; improving the throughput in all levels of communication, reducing both the latency and the used bandwidth.

10.5 Video Control Protocol

In addition to the PICS, we have also to control the cameras' resolutions and activation, for that we use the facilities of the H323 protocol. On the robot side this is made by the

five H323 Video Agents (one for each of the cameras) onboard the Robot. All five Video Agents wait for a connection and as soon as this connection is made they automatically answer with the corresponding video stream and, as soon as the connection is stopped, by the console, they return to the waiting state. On the console side we have three Video control Agents, one for each of the console video windows. Those three agents can connect to a chosen Video Agent onboard the robot. So we have at most three H323 video streams at a time. As explained the H323 protocol uses the H261 video protocol that is able to adapt the transmission bit rate to the available channel bandwidth.

10.5.1 Implementation details

On the robot side we have used the Ohphone application controlled by a shell scrip Agent. The options for video quality, microphone device and video device are in a shell script file executed by the shell script agent. In the operation console we have wrapped the OpenH323 library inside a DLL that exports all the needed functionality to the main program, it was necessary to do extensive changes to the OpenH323 library, using the Microsoft Visual Studio 6.0. This library is then imported by the main program Developed in managed C++ using Visual Studio .net.

The DLL exports an object called PhoneDLL. In the console operation we have three of those objects and each corresponds to a Video window on the operation console. This object takes the role of a video phone and is able to make connections (Call(address) method) to the several video phones pairs onboard the robot. After the connection is made it is possible to start the video streaming with startVideo() method. The user may choose, to break the connection (as well as the video streaming) with Hangup().

The interface DLL is presented next; most of the methods of this interface are self explanatory. The notable exceptions are the SetHWnd(int) and the several Paint Methods. The SetHWnd(int) propose is to give an Window Handler to the DLL, the DLL in turn invalidates the window when it has a new video frame ready to Paint on the window. The operation console detects the window invalidation and Calls one of the Paint Methods to tell to the DLL to draw the new image. A double image buffering scheme is used on the DLL to minimize video flickering. Thus the DLL plays the role of a drawing function.

The several Paint methods differ on the image manipulation, the first two only draw an image on a given graphic context; the third is able to crop the source image and to scale the cropped image to fit the destination window. The last Paint Method on top of cropping and scaling is capable of doing the rotation of the image. All of them follow the GDI convention for parameters.

```
class PHONEDLL_API PhoneDLL {
public:
    PhoneDLL();
    ~PhoneDLL();
    void SetHWnd(int);

    void Call(const char* address);
    void Hangup();
    void AcceptCall();
    void RefuseCall();
};
```

```

void Transfer(const char* address);
void Hold();
void Intrude(const char* address, unsigned capability);

void StartVideo();

void Paint(int hdc);
void Paint2(int hdc);
void Paint(int hdc, int sx, int sy, int cx, int cy, int dx0, int dy0, int dx2, int dy2);
void Paint(int hdc, int sx, int sy, int cx, int cy, int dx0, int dy0, int dx1, int dy1, int dx2, int dy2);
private:
    DLLH323* dll;
};

```

10.5.2 OpenH323 licensing details

In both sides the OpenH323 library is not part of the solution and thus can be changed later if needed. For that the DLL interface needs to be respected. In the robot any program that is able to run in a shell is a candidate to transmit the video.

The licensing agreement states that the source code of the DLL is made available for anyone who wishes it. The licensing policy does not imply the public release of the robot and/or operation console source code.

10.6. Low-level PIC Communication Protocol

On the robot side the PC is connected to the low-level electronics composed of four PICs; one for the motors and another three for all the other sensors/servos. The total number of sensors and actuators connected to the PICs is subject to changes, due to the fact that the robot can be adapted to other types of missions.

In spite that the PICs are connected to the USB, the device is viewed like a serial RS232 interface by the CPU because we are using a USB to RS232 Bridge.

A special communication protocol was developed on top of the serial protocol. This legacy protocol defines the data frames exchanged between the PC and the PIC. The PIC has a low processing capability therefore the communication protocol is simple, namely the values exchanged are already in binary form. This protocol has also some simple communication failure/recovering protection.

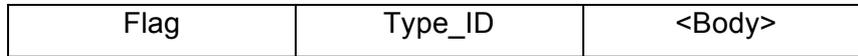
We have decided to have the same serial communication protocols to all PICs. Therefore the protocol is able to couple the different number of sensors/actuators of each PIC and also adapts easily to changes in number or type of sensors/actuators in each of the PICs giving the required flexibility to the RAPOSA final arrangement.

10.6.1. Generic PIC<->PC Protocol

The generic protocol is based on frames. All the frames of the protocol start with a flag; the flag is the special value (FFh) that can not appear in any of the other fields. This flag makes possible to reacquire the frame synchronization with the PICs in case of bad initialization, data communication error and buffer overflow.

The second field is the type identifier for the frame. The subsequent fields are parameters/data. The Body field is of variable length. The size and data of the body field depends of the frame type and is explained in the next section. Since we are using “FF” as the Flag value this means that this value cannot appear in the Body field.

The frames follow the same structure regardless of the direction of communication; that is to say from the PC to PIC or the PIC to the PC.



Flag = A byte having the “FF” value (hexadecimal).

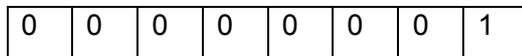
We have also a special reset frame that allows the PC to request a PIC reset; it is composed of two flags in a row. The PIC is always expecting this frame, in all possible states. Whenever this frame is detected the PIC should do a reset. This frame is also used to advertise the PC that the PIC successfully started his operation; so this frame is transmitted by the PIC whenever it starts.



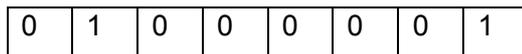
10.6.2. PC->PIC Communication

The protocol from the PC to the PIC is based in different types of operations: *Get* and *Set*, *Get_Size* and *Set_Size* . The *Set* is used when the PC wants to change a value in the PIC. The *Get* is used to get a value from the PIC. The PIC has one memory Buffer; for writing the commands and for reading the data. This buffer address space is 32 positions of one Byte. The buffer starts at 0x20Hex.

Set:



Get:



The *Set*, in the simple case, works as follows: the <address> is the PIC memory position that we want to change with the new <value>. Each memory position corresponds to a command to a device connected to the PIC; So the value is, for instance, a velocity command for one of the motors, a bit mask for enabling/disabling sensors and so on.



The *Get* operation, in the simple case, works in a similar way to the *Set* operation. The <address> corresponds to the memory position that we want to read. The PIC then returns the position value; this can be a sensor reading, a status data and so on.

Since we are changing one frame for each individual `Get` or `Set` this could pose several communication delays and difficulties. For these reason we have envisaged a type of `Set` and `Get` that can exchange several values in the same frame.

For setting/getting several values the operation Byte (the first Byte following the flag Byte) is the following:

`Set_Size` byte:

0	0	0	S4	S3	S2	S1	S0
---	---	---	----	----	----	----	----

`Get_Size` byte:

0	1	0	S4	S3	S2	S1	S0
---	---	---	----	----	----	----	----

The Bits S4..S0 denote the size (in Bytes) of the data that will be transferred. In the case that the Bits are S4..S0=00001 this means that we are in the simple case where only one Byte is exchanged.

The frame for the `Set_Size` is the following.

Flag	<Set_Size>	<Address>	<Value_0>	...	<Value_n>
------	------------	-----------	-----------	-----	-----------

The `Set_Size`, works as follows: the Bytes are changed from the <Address> memory position till <Address+Size> memory position. The Size as explained is specified by the Size Bytes (S4..S0) of the `Set_Size` Byte. The number of Value Fields must be equal to the specified size.

Flag	<Get_Size>	<Address>
------	------------	-----------

The `Get_Size`, works in a similar way to the `Set_Size`: We can read “n” values starting at the <Address> memory position. The number of Bytes for reading is specified by the Size Bits (S4..S0) of the `Get_Size` Byte.

10.6.3. PIC->PC Communication

When the PIC receives a `Get` or `Get_Size` frame he should reply with a frame containing the requested data, this frame also works as acknowledge:

Flag	<AckGet>	<Data_0>	<...>	<Data_n>
------	----------	----------	-------	----------

The `AckGet` Byte is specified as follows:

0	1	0	S4	S3	S2	S1	S0
---	---	---	----	----	----	----	----

Where S4..S0 denote the number of bytes that where requested, the number of data fields in the data frame must be the same as de number denoted by S4..S0.

10.6.4. PIC Memory Tables

The tables that are presented in this section show for each of the PICs the purpose of each memory position. We have decided to split the PICs memory tables in two parts,

the first part is for writing the commands arriving from the PC whilst the second one is where the PICs records the data to send to the PC. Each position is one Byte long.

When we refer to a mask, this means that only the seven least significant bits are used. The mask purpose is for enabling or disabling sensors or actuators. Each bit position corresponds to a single device. If the bit is 0 the device is to be disabled, otherwise the device is to be enabled. The tables are self explanatory.

Address (Hex)	Purpose	Operation Type (Set or Get)																		
0x20	Desired Velocity Motor 1	Set																		
0x21	Desired Velocity Motor 2	Set																		
0x22	Desired Position Motor 3	Set																		
0x23	<table border="1"> <tr> <td colspan="9">Motor Mask Enable/Disable</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Arm</td> <td>M2</td> <td>M1</td> <td></td> </tr> </table>	Motor Mask Enable/Disable									0	0	0	0	0	Arm	M2	M1		Set
Motor Mask Enable/Disable																				
0	0	0	0	0	Arm	M2	M1													
0x24	Max Velocity Motor 3	Set																		
0x25	Motor 3 Hysteresis	Set																		
0x26	Arm Inferior Limit	Set																		
0x27	Arm Superior Limit	Set																		
0x28	<i>Kp gain</i> Arm	Set																		
0x29	<i>Kp1 gain</i> Arm	Set																		
(...)	<i>Reserved for Future Use</i>	Set																		
0x2F	<i>Reserved for Future Use</i>	Set																		
0x30	Effective Motor 1 Velocity	Get																		
0x31	Effective Motor 2 Velocity	Get																		
0x32	Effective Motor 3 Position	Get																		
0x33	<i>Reserved for Future Use</i>	Get																		
(...)	<i>Reserved for Future Use</i>	Get																		
0x3F	<i>Reserved for Future Use</i>	Get																		

Table 23: Memory table for the Motor PIC.

Address (Hex)	Purpose	Operation Type (Set or Get)								
0x20	Camera Servo 1 (0-120)	Set								
0x21	Camera Servo 2 (0-120)	Set								
0x22	Light Intensity (0-140)	Set								
0x23	Light Mask <table border="1" style="margin-left: 20px;"> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>L3</td> <td>L2</td> <td>L1</td> <td>L0</td> </tr> </table>	0	0	0	0	L3	L2	L1	L0	Set
0	0	0	0	L3	L2	L1	L0			
(...)	<i>Reserved for Future Use</i>	Set								
0x2F	<i>Reserved for Future Use</i>	Set								
0x30	Sharp 1 Distance	Get								
0x31	Sharp 2 Distance	Get								
0x32	Sharp 3 Distance	Get								
0x33	Sharp 4 Distance	Get								
0x34	<i>Reserved for Future Use</i>	Get								
(...)	<i>Reserved for Future Use</i>	Get								
0x3F	<i>Reserved for Future Use</i>	Get								

Table 24: Memory table for the Frontal Arm PIC

Address (Hex)	Purpose	Operation Type (Set or Get)
0x20	Docking Motor Position	Set
0x21	Motor Hysteresis	Set
0x22	Max current	Set
0x23	Activate Motor 0->1	Set
0x24	<i>Reserved for Future Use</i>	Set
(...)	<i>Reserved for Future Use</i>	Set
0x2F	<i>Reserved for Future Use</i>	Set
0x30	Sharp 1 Distance	Get
0x31	Sharp 2 Distance	Get
0x32	Sharp 3 Distance	Get
0x33	Sharp 4 Distance	Get
0x34	Sharp 5 Distance	Get
0x35	Sharp 6 Distance	Get

0x36	Sharp 7 Distance	Get
0x37	Sharp 8 Distance	Get
0x38	Accelerometer X	Get
0x39	Accelerometer Y	Get
0x3A	Accelerometer Z	Get
0x3B	Accelerometer W	Get
0x3C	Real Door Position	Get
0x3D	Motor Current	Get
0x3E	Motor State: 0 – Moving 1 – Finished w/ Max Current Error (probably stuck) 2 – Finished Ok	Get
0x3F	<i>Reserved for Future Use</i>	Get

Table 25: Memory table for the Body PIC #1

Address (Hex)	Purpose	Operation Type (Set or Get)								
0x20	Light Intensity Front (0-140)	Set								
0x21	Light Intensity Back (0-140)	Set								
0x22	Enable/Disable Lights and Battery test <table border="1" style="margin-left: 20px;"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>PWM1</td><td>PWM2</td><td>Batt</td> </tr> </table>	0	0	0	0	0	PWM1	PWM2	Batt	Set
0	0	0	0	0	PWM1	PWM2	Batt			
0x23	Enable/Disable Gas Sensor Mask Bit <table border="1" style="margin-left: 20px;"> <tr> <td>0</td><td>0</td><td>0</td><td>S1</td><td>S2</td><td>S3</td><td>S4</td><td>T/H</td> </tr> </table>	0	0	0	S1	S2	S3	S4	T/H	Set
0	0	0	S1	S2	S3	S4	T/H			
(...)	<i>Reserved for Future Use</i>	Set								
0x2F	<i>Reserved for Future Use</i>	Set								
0x30	Gas Sensor 1 (CO)	Get								
0x31	Gas Sensor 2 (Hydrogen Sulfide)	Get								
0x32	Gas Sensor 3 (Combustible Gas1)	Get								
0x33	Gas Sensor 4 (Combustible Gas2)	Get								
0x34	PC Battery	Get								
0x35	Motor Battery	Get								
0x36	Temperature Byte 0	Get								
0x37	Temperature Byte 1	Get								
0x38	Humidity Byte 0	Get								
0x39	Humidity Byte 1	Get								

0x3A	<i>Reserved for Future Use</i>	Get
(...)	<i>Reserved for Future Use</i>	Get
0x3F	<i>Reserved for Future Use</i>	Get

Table 26: Memory table for the Body PIC #2

10.7 PIC Software Integration

In this section the low-level programming of the PICs for sensor communication and PC interaction is detailed. The temperature and humidity sensor has a very specific communication protocol that is discussed first. Next, the PIC<->PC communication protocol is presented. The next sub-sections show how this protocol is used on each of the boards.

10.7.1 Temperature and Humidity Sensor

The sensor SHT71 communication uses a serial Interface communication not compatible with the standard I2C communication. For that reason it was necessary to create a compatible communication between the sensor and the PIC microcontroller.

This has two power lines that are connected to +5V and Ground and two communication lines, the Serial Data (DATA) Line and the Serial Clock Input Line (SCK).

The SCK is used to synchronize the communication between the PIC and the sensor. The SCK was connected to the PORTB 1 pin and configured as an output.

The DATA line is used to transfer data in and out of the device. The DATA line was connected to the PORTB 2 of the micro and must be configured as an output when the PIC is sending data and input when the PIC is receiving data from the sensor.

10.7.1.1. Communication

Connection Reset Sequence:

A Connection Reset Sequence is required when the sensor and Microcontroller are powered or if the communication with the device is lost.

The Data line is pulled high and the SCK line is toggle 9 times followed by a "Transmission Start"(see Transmission Start).

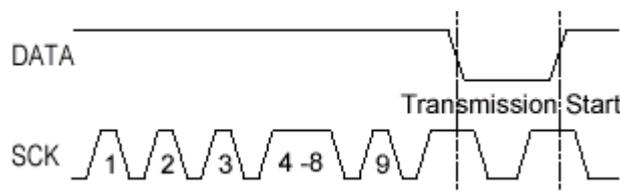


Fig. 40: Connection Reset Sequence

Transmission Start:

To initialise the communication sequence the PIC must send a "Transmission Start". The sequence start with the Data line high and SCK low. The SCK is pulled high. Then the Data line must be pulled low. The SCK shall toggle from high to low and high again. When SCK is high the DATA Line is pulled to low. Pulling the SCK low ends the "Transmission Start".

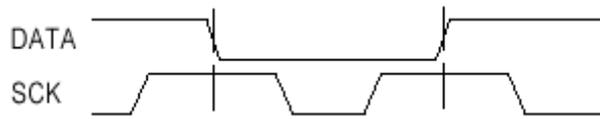


Fig. 41: Transmission Start Sequence

Measurement Sequence:

The measurement sequence is started by sending a "Transmission Start" followed by a byte composed of 3 address bits (sensor address 000) and 5 bits of Command.

Command	Code
Measure Temperature	00011
Measure Humidity	00101
Read Status Register	00111
Write Status Register	00110

Table 27: Sensor operation codes

After sending the last bit of control a NoACK must be sent by the PIC. The PIC releases the data line and the sensor takes control of it, pulling the line high.

The sensor measurement takes about 11/55/210 ms (For 8/12/14 bits respectively). During this period the data line stays high and the SCK line stays low. When the measurement ends the sensor pulls down the data line.

Then the PIC toggles the SCK line 16 times and reads the data line every time the SCK come high composing a 2 bytes word corresponding to measurement of the humidity or temperature. In the last bit the PIC gets control of the DATA line and sends an ACK in the DATA line and Toggle the SCK line one time.

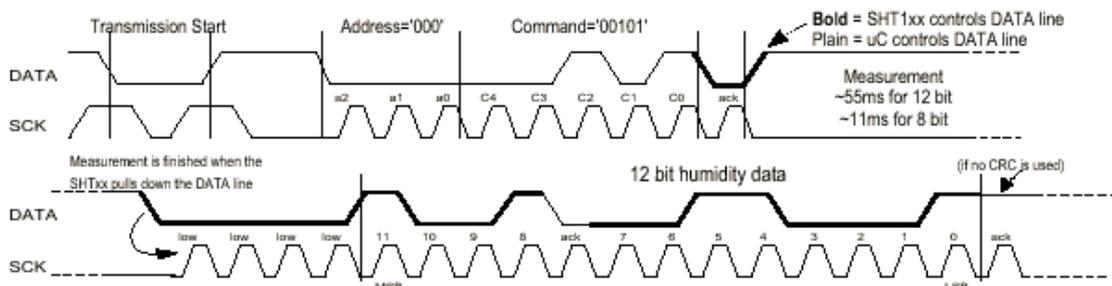


Fig. 42: Transmission Start Sequence

To read the other sensor or to repeat the measurement the PIC must repeat the same process choosing the command to be sent.

10.7.1.2 Converting the Output to Physical Values

Temperature

For temperature the sensor gives the possibility to choose between 2 resolutions: one with 12 bits and another with 14 bits resolution. With 12 bits the sensor can measure the temperature with a resolution of 0.04°C and with 14 bits the resolution is 0.01°C. The associated error for both resolutions can be found in the temperature accuracy figure.

To convert the output to physical values the following calculations must be performed:

$$T_{\text{C}} = -40.00 + d_2 \times \text{SO}_T$$

Where SO_T is the 2 bytes temperature read from the sensor. And $d_2 = 0.01$ (14 bits) or $d_2 = 0.04$ (12 bits).

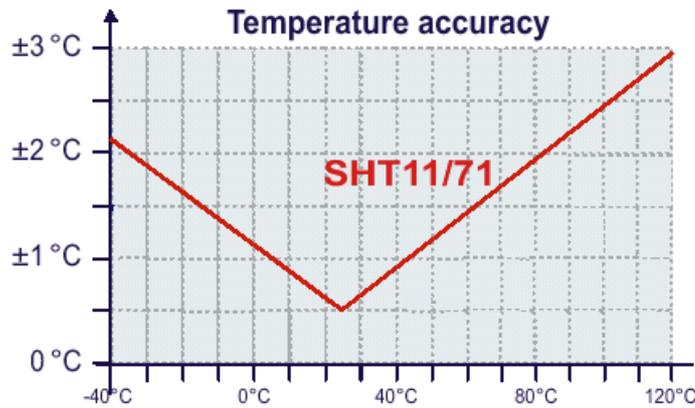


Fig. 43: Temperature Accuracy

Relative Humidity

For Humidity the sensor gives the possibility to choose between 2 resolutions: one with 8 bits and the second one with 12 bits resolution. With 8 bits, the sensor can measure the temperature with a resolution of 0.5%RH; with 12 bits, the resolution is about 0.03%RH. The associated error for both resolutions can be found in the Relative Humidity absolute accuracy figure.

To convert the output to physical values the following calculations must be performed:

$$\text{RH}_{\text{linear}} = C_1 + C_2 \times \text{SO}_{\text{RH}} + C_3 \times \text{SO}_{\text{RH}}^2$$

Where SO_{RH} is the 2 Bytes humidity read from the sensor. $C_1 = -4$, $C_2 = 0.648$ and $C_3 = -7.2 \times 10^{-4}$ for a resolution of 8 bits or $C_1 = -4$, $C_2 = 0.0405$ and $C_3 = 2.8 \times 10^{-6}$ for a resolution of 12 bits.

Compensation of RH/Temperature dependency

For temperatures significantly different from 25°C the temperature coefficient of RH sensor should be considered.

$$\text{RH}_{\text{true}} = (T_{\text{C}} - 25) \times (t_1 + t_2 \times \text{SO}_{\text{RH}}) + \text{RH}_{\text{linear}}$$

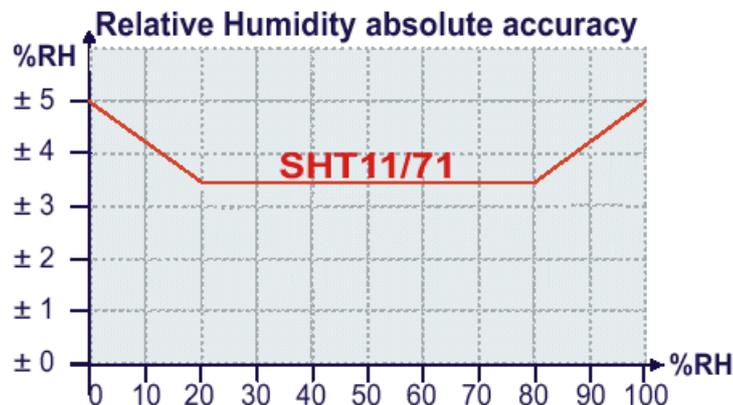


Fig. 44: Relative Humidity absolute accuracy

Where $t_1 = 0.01$ and $t_2 = 0.00128$ when using 8 bit resolution or $t_1 = 0.01$ and $t_2 = 0.00008$ when using 12 bit resolution.

10.7.2 Motor Board

The PIC Motor software was implemented and tested on the real robot. The RAPOSA computer can communicate with this board by opening a serial channel with a baud rate of 115200 bits/s. The board allows separate enabling, disabling and actuation of the three motors. Actuation is achieved by sending a value between 0 and 254 to each of the locomotion motors that translates into speed. The frontal arm motor also receives a value between 0 and 254 that translates into the desired elevation with the help of the feedback from the potentiometer attached to the frontal arm.

The computer communicates with this board by using the software protocol defined in the previous sub-section.

The onboard computer sends 8 bits of data for each motor corresponding to the desired angular velocity or position of the motors to the microcontroller. The microcontroller saves these values internally and if the corresponding motor is enabled, the microcontroller loads the Digital to Analogue (DAC) converter with 8 bits value through the 8 pins of PortB. Then the ADC generates a voltage between -10V and 10V. The value 0 corresponds to -10V(backward), 128 to 0V(stop) and 254 to 10V(forward).

As long as the computer does not send new data, the old data is maintained. Each time new data arrives from the PC an interrupt routine is invoked in the PIC, thus assuring that the values take immediate effect. This sub-routine assures all communication with the PC and since it is interrupt driven, it may be called anywhere on the main software cycle.

The main software cycle is described in the next sub-section.

10.7.2.1 Microcontroller main software cycle

The microcontroller main software cycle reads the feedback velocity from the two traction motors and the position of the frontal arm using the potentiometer attached to it, using this last value to control the arm motor. Since the DAC features an internal latch, there is no need to keep sending the same values over and over. So, the value on the microcontroller register is only written to the DAC if the respective motor is enabled and the Valid Data Flag is set (in the case of the traction motors).

After actuating the motors, the Valid Data Flag is cleared to indicate that the value on the DAC has been updated.

The microcontroller program reads the enable/disable flag:

```
[0,0,0,0,0,M3,M2,M1]
```

where **M1**, **M2** and **M3** are enable/disable bits to enable or disable the controllers that control the motors. By setting one of these bit flags, the corresponding controller becomes enabled and generates the proper control for the associated. If the bit is clear the controller ignores the velocity/position control values and disconnects the power supply of the motors.

10.7.2.2 Motor Interrupt State Machine

Whenever the robot computer sends information to the motor board, the microcontroller generates an interruption on the PIC software main cycle, jumping to an interrupt sub-routine, where the flow of the data is controlled by a state-machine represented in Fig. 45.

Each state has a relation with the data sequence protocol defined in [5]. The first state (STATE 0) is the reset state. Communication is initiated with a start byte, so the PIC jumps to STATE 1 to signal its reception. The next transmitted byte is the operation (read or write) and number of bytes to process: STATE 2 indicates the correct reception of this indication. The final byte of data indicates the starting address of the operation and thus STATE 3 is achieved. Then, the PIC has all the information needed and it begins either the data transmission or reception, repeating respectively

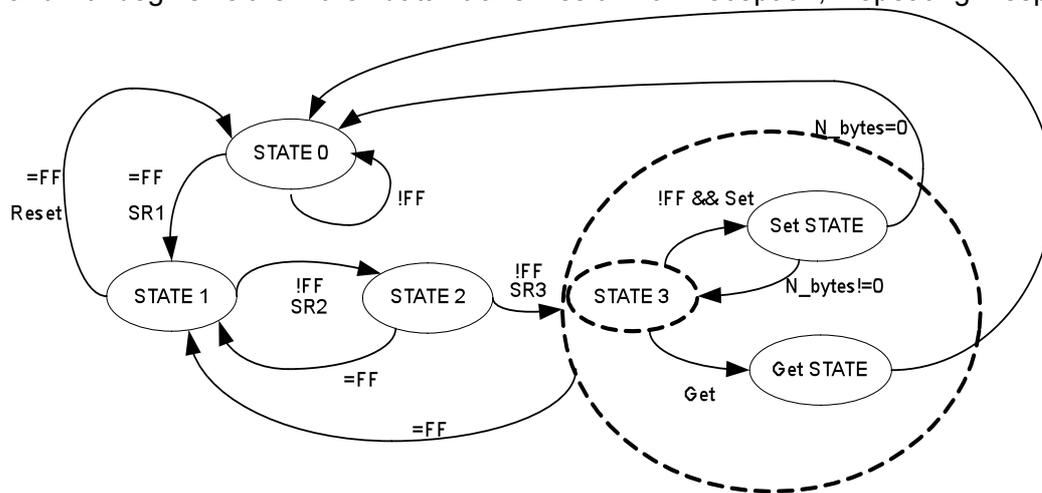


Fig. 45: Motor State Machine

Set STATE or Get STATE as many times as the number of bytes to process. The following describes the state diagram in more detail:

When the microcontroller is waiting for a new package of information the state machine is in a reset state, **STATE 0**, waiting for the start byte: "FF". If the received byte is the starting byte, the microprocessor runs the Sub Routine SR1, where the STATE counter is increased, and the state machine waits for the next byte in STATE 1. If the byte received is not the start byte then the state machine stays on STATE 0.

In STATE 1, when the microcontroller receives a 2nd byte it compares it with "FF". If equal, the microcontroller has received a reset command, thus resets the state Machine and disables all the devices. If different, SR2 is run. This Task separates the 8 bits of the byte received into two types of information. The 7th bit (OP) indicates if the computer wants to send (0) or to receive (1) data. The least significant 4 bits (S3-S0) indicate the number of bytes to send or receive. The number of bytes is stored in a variable called **n_bytes**. The state counter is increased by one and the state machine waits for a new byte in STATE 2.

0	OP	0	0	S3	S2	S1	S0
---	----	---	---	----	----	----	----

In STATE 2, when the microcontroller receives a 3rd byte, compares it with "FF", if equal return to STATE 1. If different, it runs SR3. The byte corresponds to the starting address where the data to be read or write is stored or will be stored. The state counter is increased by one and the state machine waits for a new byte in STATE 3.

If, in the 2nd byte received, the 7th bit is "1" the microcontroller sends to the computer the **n_bytes** starting from received address. After sending all the data requested, the state machine resets to STATE 0.

If, in the 2nd byte received, the 7th bit is "0" the microcontroller waits for **n_bytes** from the computer different from "FF". When all the bytes are received the state machine is

resettled to STATE 0 and the Valid Data Flag is set. If one of the bytes is equal to "FF" the state machine resets to STATE 1.

10.7.3 Gas Sensor Board

The computer onboard can communicate with this board by opening a serial channel, communicating with a baud rate of 19200 bits/s. The board allows enabling and disabling the different sensors onboard and also four possible LEDs used to illuminate the environment.

The user can enable the 4 gas sensors, connected in the board, by controlling the connection of the sensor to the power supply.

The light emitters (LEDS) can also be enabled or disabled, or set to a specific intensity by the user.

All the other sensors or measures (temperature & humidity, battery values) are read only as if they were always enabled. The temperature & humidity sensor does not consume a high current but takes about 2 seconds in communication to update its values. This means that it takes a lot of time in microcontroller cycles.

The microcontroller main software cycle only reads the sensors that are enabled or actuates the lights if they are enabled. The communication with the computer is assured by a sub-routine raised each time an interrupt occurs (the interrupt is triggered by new data arriving to the serial link).

10.7.3.1 Microcontroller main software cycle

The microcontroller program reads the enable/disable flag:

[S1,S2,S3,S4,T&H,PWM1,PWM2,Bateries]

where **S1**, **S2**, **S3** and **S4** are the gas sensors. By setting the corresponding bit, each sensor is connected to the power supply and the microcontroller periodically reads the value of the sensor. By placing the bit at state 0, the sensor is disconnected from the power supply and the microcontroller does not to read it. S2, S3 and S4 are simple gas sensors that must be connected for about 1 minute to have true values. The controller only converts the voltage at the output of the sensor to digital values. S1 (monoxide carbon sensor) has a more complex operation method like it was previously described in Section 8.2.5.1.4 of [5] and Chapter 6.2 of the current one.

T&H is the Temperature and Humidity sensor. By setting the T&H bit the microcontroller periodically reads the value of the sensor by using the communication described in 9.2.1 of [5]. By clearing this bit the microcontroller ignores the sensor.

PWM1 and **PWM2** correspond to the light emitters. By setting the corresponding bit the user can turn "ON" the power supply for this module. When this bit is activated the microcontroller generates a PWM wave form with an intensity that is given by the user.

Batteries correspond to the three voltages of the batteries in the robot and also the value of the cable power supply. By setting this bit the microcontroller periodically reads the voltages of the power supplies of the robot. By clearing this bit the robot ignores this measurement.

10.7.3.2 Interrupt sub-routine

The communications interrupt sub-routine is similar to the motor interrupt subroutine. The only noticeable difference is the lack of the Valid_Data_Flag. That particular flag

only applies to the motors and since the DAC's has an internal latch, it makes no sense in this case.

10.7.4 Accelerometers Board

The computer onboard can communicate with this board by opening a serial channel, communicating with a baud rate of 19200 bits/s. This board is used to control the following features:

- Determine the inclination of the robot;
- Control the docking mechanism system;
- Determine the current of the system;
- Switch the main power from batteries to cable supply;
- Inform the user of the robot state by activate/deactivate a green LED.

The microcontroller reads the 4 accelerometers and collects this data in 4 independent registers. This feature is always active.

It also controls the docking mechanism by reading the current position of the mechanism and enabling it, according to user input and the position hysteresis of the docking system. With these instructions the microcontroller verifies if the enable of the docking system is set and makes the motor of the system run to the desire position. The position is sensed by a linear potentiometer that is read by the microcontroller and the value is used to close the control loop of this system. This value is saved in a register and is used by the main computer unit to show the evolution of the docking system, if is opening or closing.

A second feature was included in the system, the possibility to determine the current consumption. This is important to determine if the system is running or stuck due to mechanical cause. The user can set the maximum current value allowed. Below this value the system is considered to be running normally or stopped. Beyond this value the system is considered stuck. If the maximum value is exceeded the motor is stopped and an error flag is activated. The microcontroller saves the value of the current and the error flag in two independent registers that can be used by the main computer unit to show the evolution of the current of the docking system.

The possibility of switching the main power from batteries to cable was implemented on this board. One digital output pin of the microcontroller is used for this propose. A second digital output pin is used to activate/deactivate the external robot state green LED of the robot. These two features are controlled only by the user by writing the proper configuration on a flag register.

The communication with the computer is assured by a sub-routine raised each time an interrupt occurs (the interrupt is triggered by new data arriving to the serial link).

10.7.4.1 Microcontroller main software cycle

The microcontroller checks if the docking system is active. If active it sends a motion command that leads the system to the desired position (reactive control). If the position is reached the system is made inactive. Also, if the maximum current is exceeded the system is made inactive and an error flag register is set.

Else the microcontroller tests the flag register that controls the power switch and the green light. If the green light bit register is activated, the microcontroller sets the digital output PORTC 3, else clears the digital output. After this procedure the same register is checked but for the power mode.

If the bit of the register is clear the digital output PORTC 4 is cleared, making the power to be supplied by the cable. If it is set, PORTC 3 is set and the power is supplied by the batteries.

The end of the cycle is used to determine the inclination of the robot. For that 4 accelerometers are read one by one. After analyzing the reads from those sensors we verified that there was a high error rate associated, due to white noise. To reduce this error, 100 samples from each sensor are used to determine the average of the signal. The average of each signal is saved in 4 independent registers that can be used to determine the evolution of the inclination of the robot.

10.7.4.2 Interrupt sub-routine

The communications interrupt sub-routine is similar to the gas sensor board subroutine.

10.7.5 Frontal Arm Board

The PIC Motor software was implemented and tested on the real robot. The RAPOSA computer can communicate with this board by opening a serial channel with a baud rate of 115200 bits/s.

The user can activate/deactivate 4 high intensity lights, control the servos connected to the cameras and read the distance of the floor to the arm of the robot.

The microcontroller main software cycle reads the sensors, actuates the lights if they are enabled and controls the position of the servo motors. The communication with the computer is assured by a sub-routine raised each time an interrupt occurs (the interrupt is triggered by new data arriving to the serial link).

10.7.5.1 Microcontroller main software cycle

The microcontroller main software cycle starts by reading all the sensors connected to the analogue inputs, namely distance sensors that are used to measure the distance of the arm to the floor. A voltage square wave is used to control the intensity of the lights. This is made by the use of one timer (Timer1) that generates a square wave with a fixed period but with a duty-cycle that must be defined by the user. The control of the servo motors uses the 2 PWM outputs of the microcontroller to generate the position actuations for the servo motors. The position of the servos must be defined by the user.

10.7.5.2 Interrupt sub-routine

The communications interrupt sub-routine is similar to the gas sensor board subroutine.

11. Task T6 – Mechanical Structure Assembly

11.1 General Structural Aspects

On this task the robot chassis was built by the SetPontes Company, mainly using 4 mm aluminum plates. A number of small adjustments relative to the initial CAD were done. The main steps and topics of the mechanical construction are explained on this chapter.

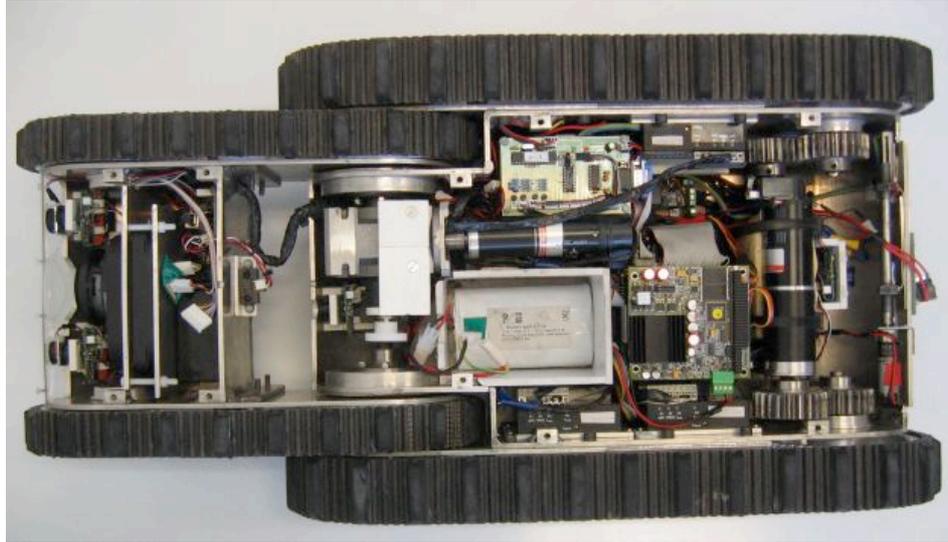


Fig. 46: Robot Inside View

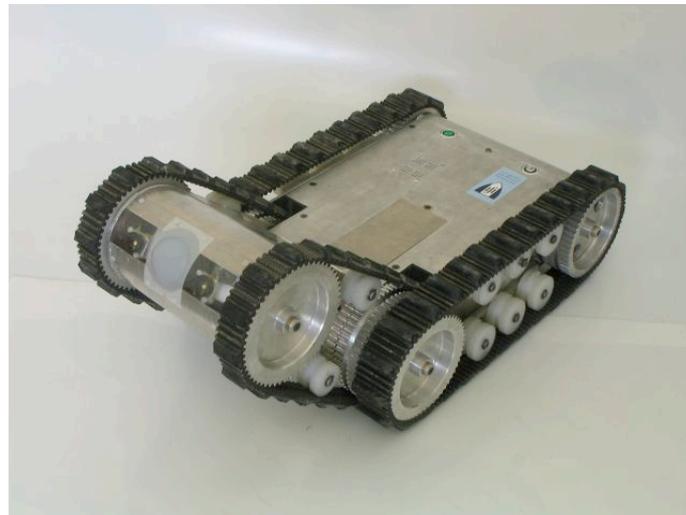


Fig. 47: Robot External View

The previous figures show the robot inside and outside views. It was all build on aluminum, a light yet resistant material. The aluminum plates were soldered together for better mechanical resistance, except the top cover, which is secured by M5 Alan screws, to allow access to the robots interior.

The rear locomotion motors are fixed with metal supports at the gear heads end.

The hard disk is assembled below the PC, between the PC104/+ connectors of the bottom PC board. Special care is taken on its housing, to isolate it electrically from the surrounding 104/+ pins.

The batteries are confined by a PVC box at the position defined in the initial CAD. There is no need to use aluminum there, since inside the robot there are no structural stresses. The box has a hole for the wires and is aligned with the battery cover on the top cover.

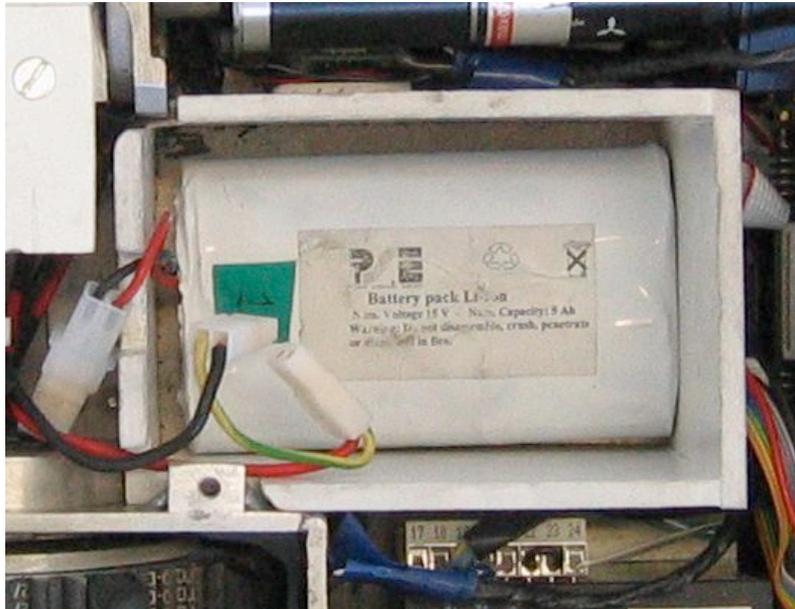


Fig. 48: Battery box

Fig. 48 shows the box with one battery inside. The remaining two batteries are placed side by side on top of this one, since the original configuration slightly exceeds the total height, due to the protection circuit on each battery.

The motor controllers are attached to the two lateral aluminum plates, horizontally (instead of vertically, as represented on the CAD). This was found more convenient for fixations purposes.

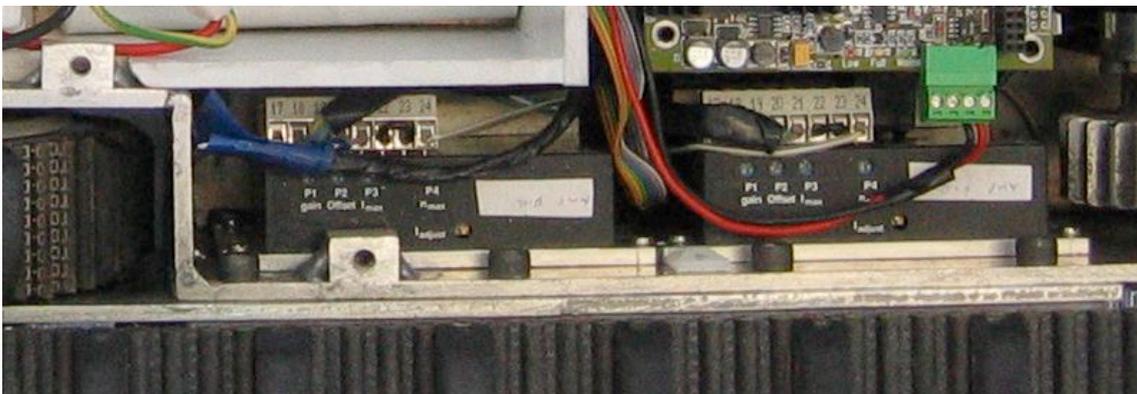


Fig. 49: Controllers attached to the aluminum plate

Other aspects of the mechanical construction require a more detailed analysis, thus are explained on one of the following dedicated sub-sections.

11.2 Tracks / Ground Clearance

The robot's tracks were built using automobile transmission belts. With adequate wheels and the same stepping, full traction could be assured. Fig. 50 shows this aspect in detail.



Fig. 50: Tracked wheels sulks

To avoid that the tracks slide out of the wheels, the wheels have a guide along its middle and the tracks have a corresponding central elevation that fits on that guide. This way, once fitted, the tracks do not slide out so easily.

From the preliminary tests done to the structure, it was obvious that the wheels were not enough to maintain the tracks next to the ground. Rollers were placed between the two wheels, both on the main body (three rollers top and three rollers down, on each side) and arm (on roller top and one bottom, on each side). The rollers were placed so that the track became a little higher than the wheels. This way full contact to the ground all along the track is achieved.

Nevertheless, although good contact is made to the ground by the tracks, this does not mean good adherence. Besides that, the robot has a small ground clearance and the main body is in contact with the ground more often than desirable. To solve this problem soft rubber teeth of about 12.5mm height were placed all along both tracks, as it can be seen on Fig. 51:



Fig. 51: Black rubber heights on the tracks

These rubber teeth increase the tracks adherence to almost all kind of materials and increases the main body distance to the ground, thus lightening both problems.

11.3 Arm motor

The arm motor was assembled with a support similar to the locomotion motors, fixing it at the gear head end. The output axis of the gear head is then connected to the additional 10:1 90 degrees gear head that drives the axis. Fig. 52 shows how it was all assembled.

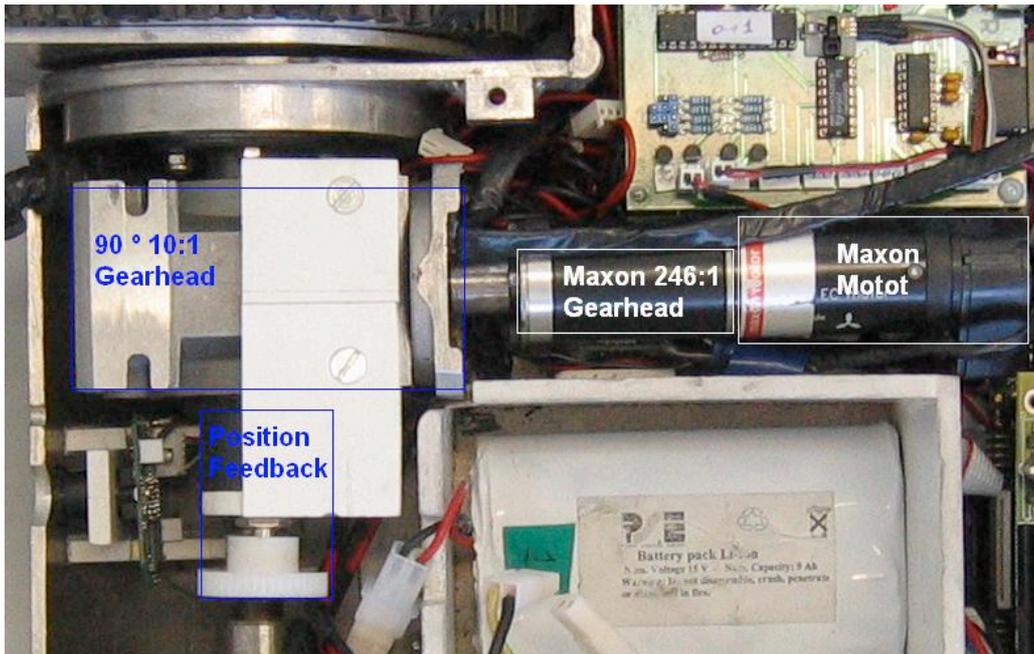


Fig. 52: Arm motor

The position feedback potentiometer was secured to the 90 degrees gear, given its steady assembly and was connected to the axis through plastic gears.

11.4 Frontal Arm

The frontal arm features both thermal camera and two webcams with associated servos, besides the associated electronic board. To secure the thermal camera two aluminum plates were used. This may seem a too heavy solution, unnecessary on the interior of the robot, but as previously stated, the center of mass of the robot shall be the more frontal possible. Thus, this heavy support is indeed convenient. In the same way, the rollers on the frontal arm are also secured with steel pieces (and not aluminum). The extra weight being placed in the arm and not in the body has the advantages of being both farther from the center of the robot and being mobile, thus allowing dynamic shifts on the center of mass of the robot. The webcams fixation is detailed on another sub-section.

Instead of opening a viewing hole for each of the cameras, a window was opened with the width of the arm. This way, the cameras and lights can be placed anywhere. The drawback of this approach is that some transparent material must be used to close the window. This is explained in section 12.7.

11.5 Docking system

The docking system is composed by two parts, the cable part, that is released on the ground and the grabbing mechanism on the robot back.

The released part has a pyramidal structure by the reasons explained on Task T3. But the final structure aspect is more elaborate, as displayed in Fig. 53:

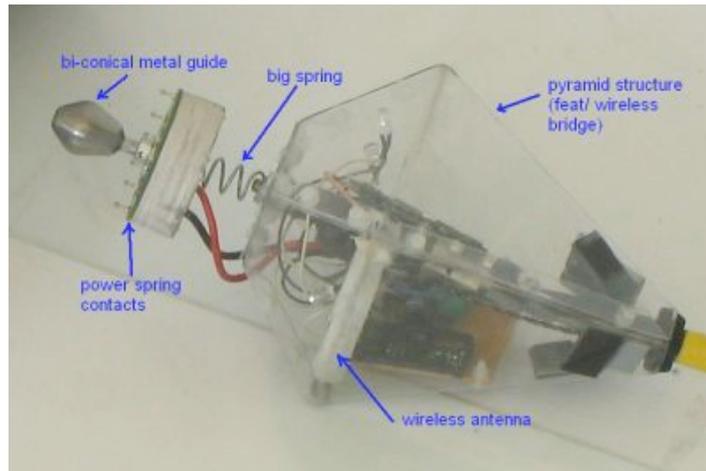


Fig. 53: Docking pyramid

On the robot side, two sliding doors are able to release or grab the cable part. When the doors are closed the cable part is pulled to the robot inside due to its bi-conical metal guide. Meanwhile, electrical power spring contacts are pressed against two concentrically arranged rings on the back of the robot. The rings are made of conductive material. Each ring is connected to a voltage pole. Since the spring contacts are also concentrically arranged, the poles are never inverted. Before attaching and prior to detachment of the docking system the power cable must be turned off to prevent electrical glitches. Appendix 3 presents illustrative figures that help understanding the process.

When the mechanism is released the doors are closed to prevent debris and water to enter the robot.

The power contacts and bi-conical metal guide are separated from the pyramid body structure through a large spring, whose purpose is to avoid breaking the docking mechanism when it is dragged to unfavorable positions.

The pyramidal structure is built using 1mm thick polycarbonate. This is a lightweight yet rigid material that allows the structure to be solid while not becoming too heavy for the robot to drag.

Besides the Ethernet to wireless adaptor and corresponding antenna (fixed to one of the pyramid edges), the pyramid also features:

- A small DC-DC 2A board: the power traveling by the cable is prone to noise and instability. If the power is regulated at the end of it, these effects can be minimized;
- 4 green LEDs. If the pyramid is dropped on a dark environment, this aids in finding it again. The fact that the LEDs are also at an equal distance from the bi-conical guide helps precise maneuvering to grab the cable end.

11.6. Web Cams

The chosen Webcams are rather bulky, so the first step taken was to strip them up, leaving only the circuit board and lens. The circuit measures about 45x38 [mm], the lens features a height of 25 mm, giving the set a total height of 40mm.

The circuit boards do not feature adequate mounting points, so a PVC structure was devised. The webcams on the frontal arm have an associated servo motor, so their support required some extra care.

11.7. Weight distribution

The condition that enables the robot to climb all types of legal stairs was determined, as represented in Fig. 54.

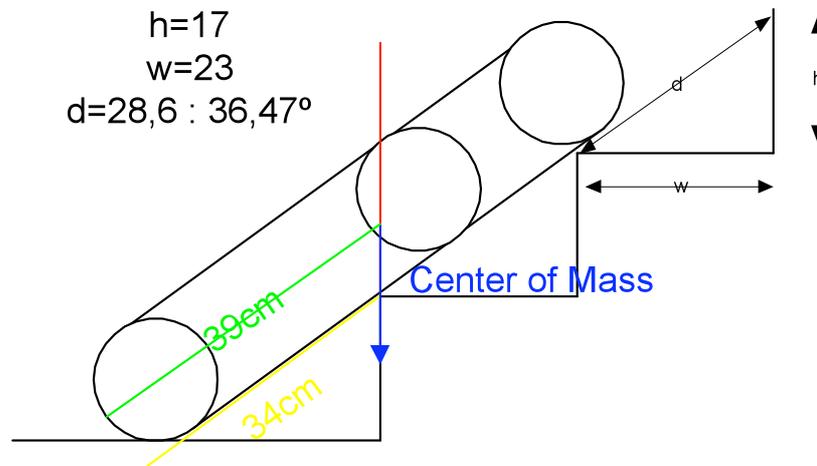


Fig. 54: Center of Mass

This condition imposes that the robot center of mass is located at 39cm from the back of the robot. Some care was taken to achieve this situation, namely:

- The back wheels were drilled to alleviate some weight;
- Not all lateral aluminum plates stretch to the back of the robot.

Nevertheless, this proved insufficient. So, a 0.5Kg weight was placed on the arm, behind the thermal camera. This was enough to make the robot able to climb stairs.

12. Task T7 – Electronic Equipment Assembly

Task T7 is not always discernible from Task T4, given that for an adequate test of most of the sensors an electronic circuit assembly is required. So, much of the work that could be showed here was already discussed previously.

Nonetheless, some extra details concerning electronics can be more explicit, at this stage not centered on individual sensors, but from a system point of view.

All the electronic boards developed by IdMind are assembled on a stack parallel to the arms motor.

12.1. DC-DC Board

The inhibit function of the DC-DC was not used. This feature was necessary at the time when the protection circuit of the lithium batteries was not working correctly and thus could not be trusted. At this time, with a proper protection circuit on the batteries, the inhibit function is disabled because it is not necessary and it could turn off the DC-DC boards while the batteries were still operational.

This board also has the output voltage for the thermal camera, but no regulation is made, only the diode mixing from different voltage sources, since the camera has a wide range of input voltages, from 9 to 28VDC.

This board is the first one (lowest) of the electronics board stack. The power goes to the other board through small Molex terminals.

12.2. Batteries and Power

The batteries original connector was replaced by other with golden contacts. This gives less oxidation problems and the plug itself is a more close fit. The batteries voltage was found to be higher than what was advertised. At full load the batteries have 16.3V. Two batteries in series have 32.6V.

The relay board was attached to the bottom of the robot, between the PC and one of the motor controllers. A suitable switch was found for the motor enable. It must fulfill the following requirements:

- Have a visible indication of the motor power state (light);
- It should be flat, meaning than when it is assembled on the robot it does not increase its height, neither is accidentally activated by irregularities on the ground.

The following was selected and assembled on the robot top, behind the motors, just before the non planar part of the robot's back.



Fig. 55: Motor enable

For the PC a key switch was used. This has even a lower risk of accidental turn-off and also features a low profile:



Fig. 56: PC Start switch

The following board was assembled on the back of the robot, centered on the docking mechanism hole. It features two concentric circles, the exterior for the negative voltage terminal and the interior for the positive voltage terminal. From here two wires go to the interior of the robot, for the relay and diodes inputs. The wires used have a 1.5'' section, to assure they can carry the current required by the robot (maximum of 15A). After the DC-DC and motor controllers the wires are thinner, since the current splits to each of the components.

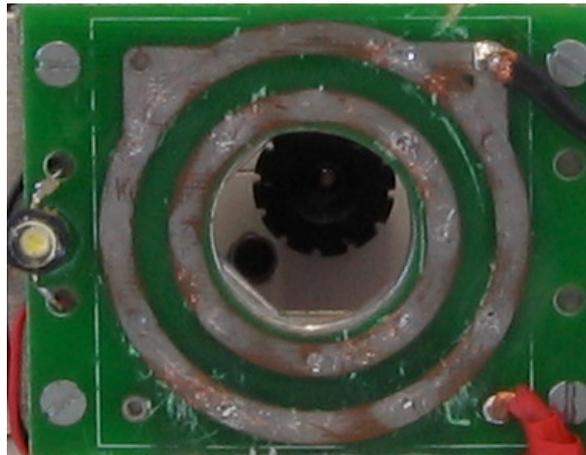


Fig. 57: Concentric voltage connectors

12.3. PC Assembly and WebCams

The notebook hard disk was assembled under the PC104 boards, in the middle of the ISA and PCI connectors. In order to avoid short circuiting these pins, the disk was laterally isolated.

The four webcams were stripped from their casing and connected to the four inputs USB 2.0 board. The original cables were replaced by adjustable ones, so that only the exact length of wire is occupying space and also featuring smaller plugs. No noticeable data degradation was observed.

The remaining PC was assembled according to the CAD, but this was only possible due to the smaller USB plugs, otherwise it would not fit.

12.4. Motors Board

The final motor board was successfully tested under its final environment. The arms position feedback potentiometer was also connected to this board.

Disabling the power stage did not cut as much current as expected, the three controllers drained about 600mA on that state. This is a rather high current, which may reduce the robot' autonomy, thus fully justifying the motor enable switched discussed previously.

The motor controllers were also connected and all wires were blinded to avoid interference.

The motor controllers were adjusted according to the supplier manual, the gain at its maximum while being stable, the offset adjusted with a zero velocity applied at its input until the motor fully stops. The current limiting is adjusted at its maximum.

12.5. Body PIC #1

On this board the following sensors/actuators connect two accelerometers (2 axis), the docking motor, current and position feedback of the docking motor. There are still available terminals for additional infrared Sharp distance measurement sensors that may be used in the future.

The first accelerometer was assembled to the bottom of the robot, giving the first two axes (the two parallel to the bottom of the robot). The second accelerometer was connected to the batteries box, being perpendicular with the robot bottom, returning the remaining axis. This way we have acceleration values in all three axes. The redundant axis in the second accelerometer was also connected to the PIC, as it may prove of some use in the future. The sensors themselves are quite small, so they were assembled on a little board just for fixation purposes.

The docking mechanism features a conventional motor driving a worm gear box that translates the rotating movement to a linear displacement of the two plates (one against (or apart from) each other). There are two unknowns to be dealt with: where are the plates at a given moment and if they become stuck (the docking pyramid edge did not enter correctly at the hole). To monitor the first issue a linear potentiometer is attached to one of the plates, thus giving a proportional value to the position. To monitor if the motor gets stuck, a low ohmic resistance (33Ω) is placed in series with it and its voltage is monitored. This gives a direct ratio of the current being drawn in normal operation: if it exceeds a given limit, the motor is probably stuck.

Once a docking mechanism is activated, it proceeds to the other end until it has reached, or the current limit defined on the operation console is exceeded.

An additional green LED is used, placed between the two robot switches. This led turns ON once the PC has terminated is booting, thus indicating it is ready for operation.

12.6. Body PIC #2

This board features the gas sensors, batteries level, lights, temperature and humidity sensors.

For the gas sensors themselves two other smaller boards were made, for extra flexibility in their assembly. They feature adequate sockets for each sensor. These two boards are assembled fixed to the top cover, on the free space between the PC and the motor controller. These boards connect to the PIC board through 3 or 4 wire Molex terminated cables. On the cover, 9 holes were drilled for each sensor (therefore 9x4

holes). A future evolution for the robot is to make a stanch housing for the gas sensors, avoiding water sprinkles from reaching the robot interior.

The temperature and humidity sensor was assembled in one of the gas sensor boards. Given its small size, it fits almost everywhere.

On the final robot assembly only one lighting LED was assembled on the main body, on the back, to assist on the docking. The board provided enough light intensity dynamic range; there is still a terminal available for a LED to the front of the main body.

The battery level monitoring circuit did not work as expected. Given the higher than expected nominal voltage of the lithium batteries (as noted on the previous subsection), the circuit output saturates at 5V even when the batteries are at medium charge. The following changes are recommended on the circuit of Fig. 38:

- R2 changes from 3.3k to 2.7k
- R3 changes from 8.2k to 12k;
- R4 changes from 1.5k to 1.8k;

These three changes make that a 16.3V voltage of a single pack translates into 4.35V and that the 32.6V of two packs in series translates into 4.25V. These are very similar values, thus allowing the same scale to be used on the console PC.

12.7. Frontal Arm and associated PIC board

The PIC of the frontal arm is responsible for commanding the two webcam servos. They were assembled on a PVC structure that gives an approximate 10 degrees for each side positioning flexibility. The white LED associated with each camera was attached to this pack, so that it can rotate with the camera.

Two Sharp infrared sensors were also assembled on the back of the arm, one on the top and other on the bottom. This way, if the robot is moving into a deep hole, once the arm has passed it – but the robot main body continues secure on the ground, it is possible to have a measurable indication of the deepness of the hole ahead.

As mentioned previously, a full width window is opened on the arm aluminum chassis for the cameras and lights. This hole is covered by a 1mm polycarbonate transparent material, fixed with screws to the chassis.

However, this cannot be used in front of the thermal camera. In fact, neither could glass, acrylic, or any other of the conventional materials be used, since they are all very good thermal insulators. An IR material window, molded in a flexible 0.457mm thick, milky white plastic, from Edmund Optics, was used instead. Fig. 59 shows its transmission on the thermal spectrum (from 7 to 14um). Experimental testing shows no significant difference on the captured image with or without the plastic window.

A circular opening was opened on the polycarbonate window and the IR plastic was fixed there. Fig. 58 shows the final assembly on the arm.

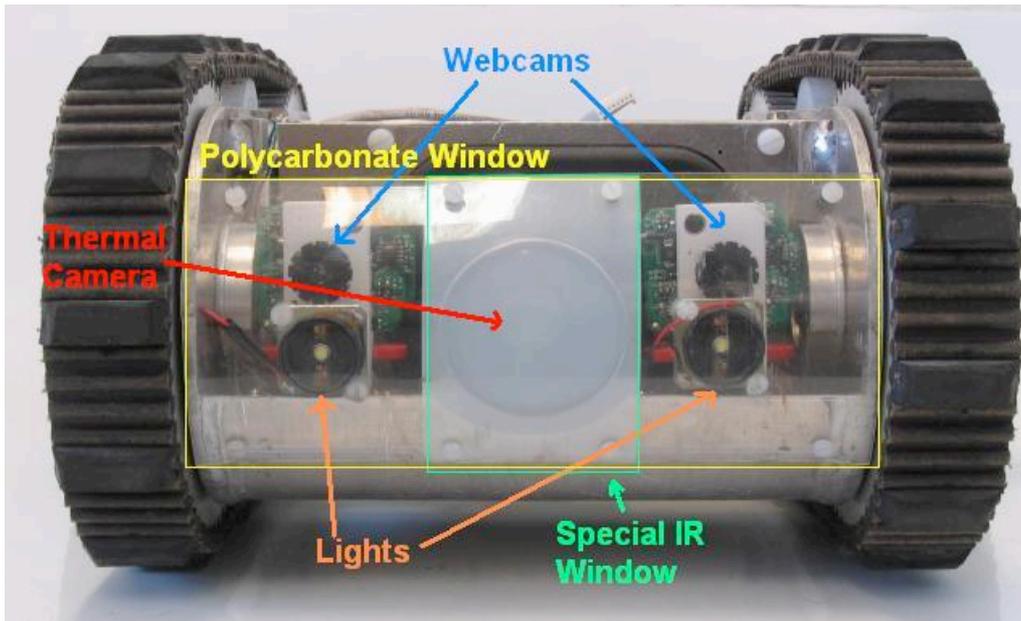


Fig. 58: Frontal Arm

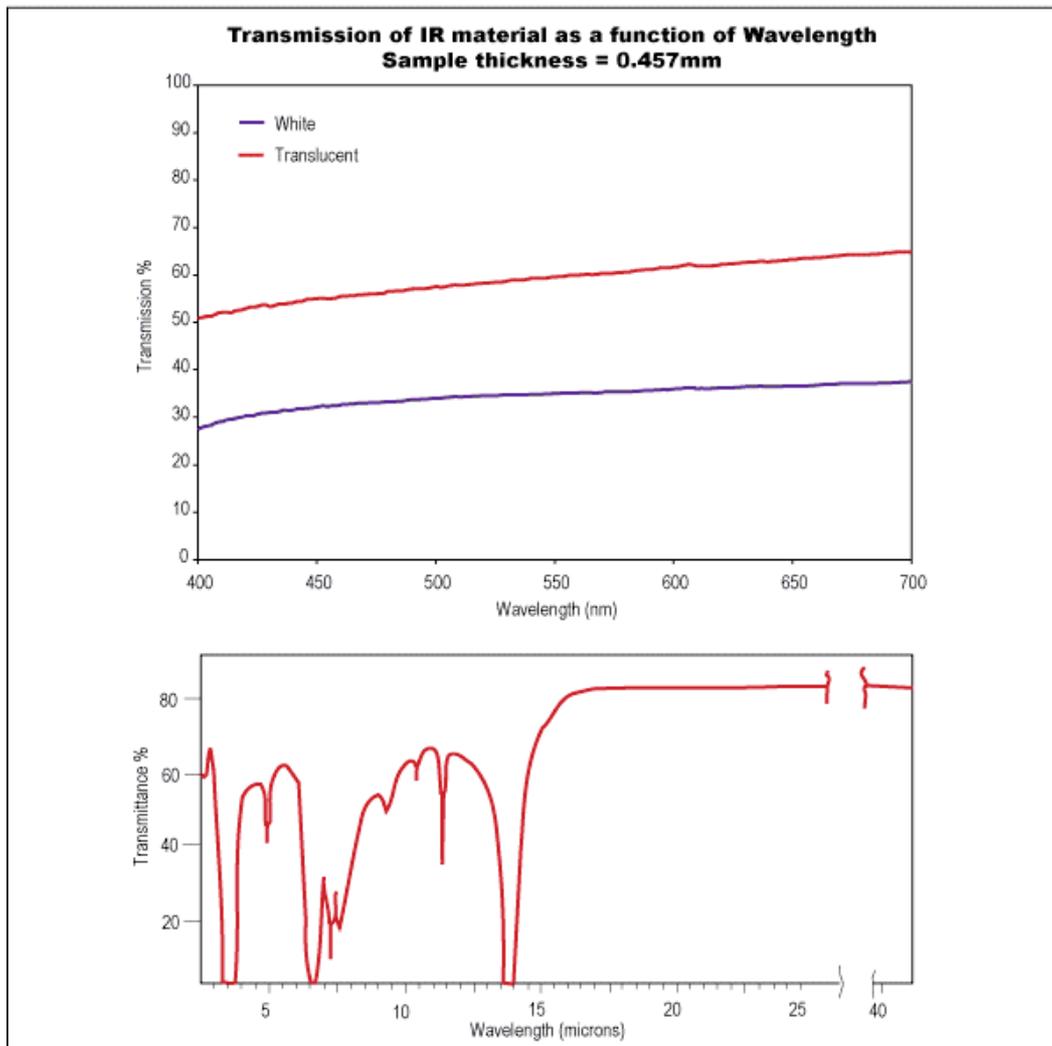


Fig. 59: Transmission of IR plastic

12.8. Wireless Antennas

The two circular polarization antennas were placed first at the rear of the robot. The real advantage of using two antennas instead of one was ascertained, with the doubling (approximately) of the wireless distance at which communication is still possible.

However, the fact that the antennas are both at the robot' rear implied that once the robot was facing forward the cable wireless antenna, the full body of the robot became an obstacle to wireless transmission and communications were lost.

If, on the other hand, the antennas were placed in the space between the tracked wheels near the roller, one on each side (right and left), the communication performance became more independent of the robot current orientation. This proved to be the best assembly. Besides that, the antennas were fixed in a more secure manner.

13. Task T8 – Software Modules Integration

To achieve a smooth hardware and software integration, it was important to define how these modules interface to each other and with the human machine interface.

The software was developed and tested in an incremental way according to the software architecture presented in Chapter 7. In the robot, all hardware modules were finished and integrated. The communication protocol between the robot and the console was defined, implemented and tested.

Meanwhile, at the console, the system was developed, integrated and tested with the robot. As stated, in the console it was decided to use the MS Windows™ operating system due to its greater and stable support for graphical user interfaces and input interfaces, for example the use of joysticks. Also the easy of use and better spread of the MS Windows™ operating system seems more suited for future robot operators.

13.1 User interface

13.1.1 Graphical User Interface

The Graphical user interface is composed of three views: the first view (see Fig. 60) is the Operation console for driving the robot; the second view (see Fig. 61, Fig. 62) is the setup view for robot and sensor thresholds configuration, the third view (see Fig. 63) is an advanced debug view, where we can watch and control in real time the low level micro-controllers data tables.



Fig. 60: Operation View

On the operation view (Fig. 60) the user has the camera feedback images as well all the data from the robot sensors and the sensor and actuator commands. Namely, on top left, the pitch and roll angles, on the top right the arm position and the lights intensity. On the bottom right all the sensor values, as well as the network state. On the bottom left the battery state, the motors velocity and the docking mechanism state.

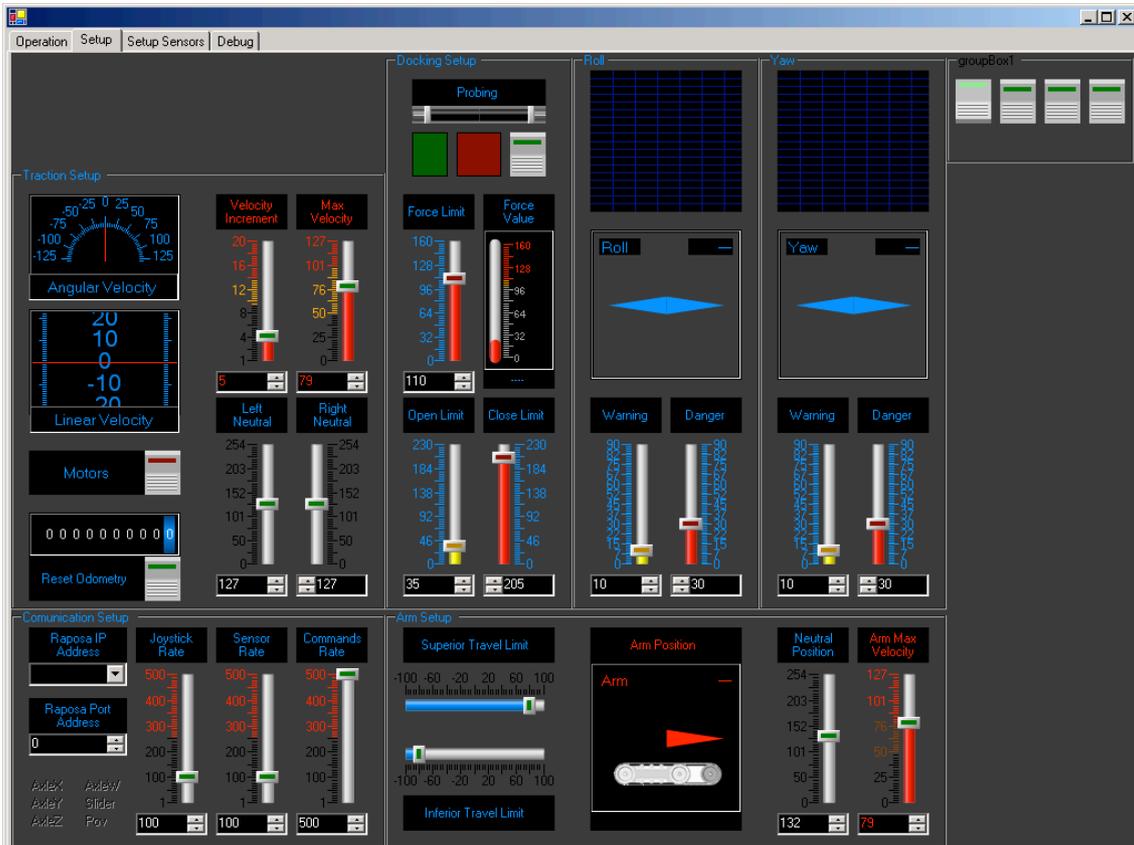


Fig. 61: First Setup View

On the first Setup View (Fig. 61) the user can adjust the sensors and command rates, as well as to establish limits on the front-arm movement and to define limits on the motors speeds. The human operator can also set the Warning and Danger values of the Roll and Yaw sensors. More configuration settings can be included on this view if necessary.

The commutation between views is possible selecting each of the corresponding tabs on the top of the interface.

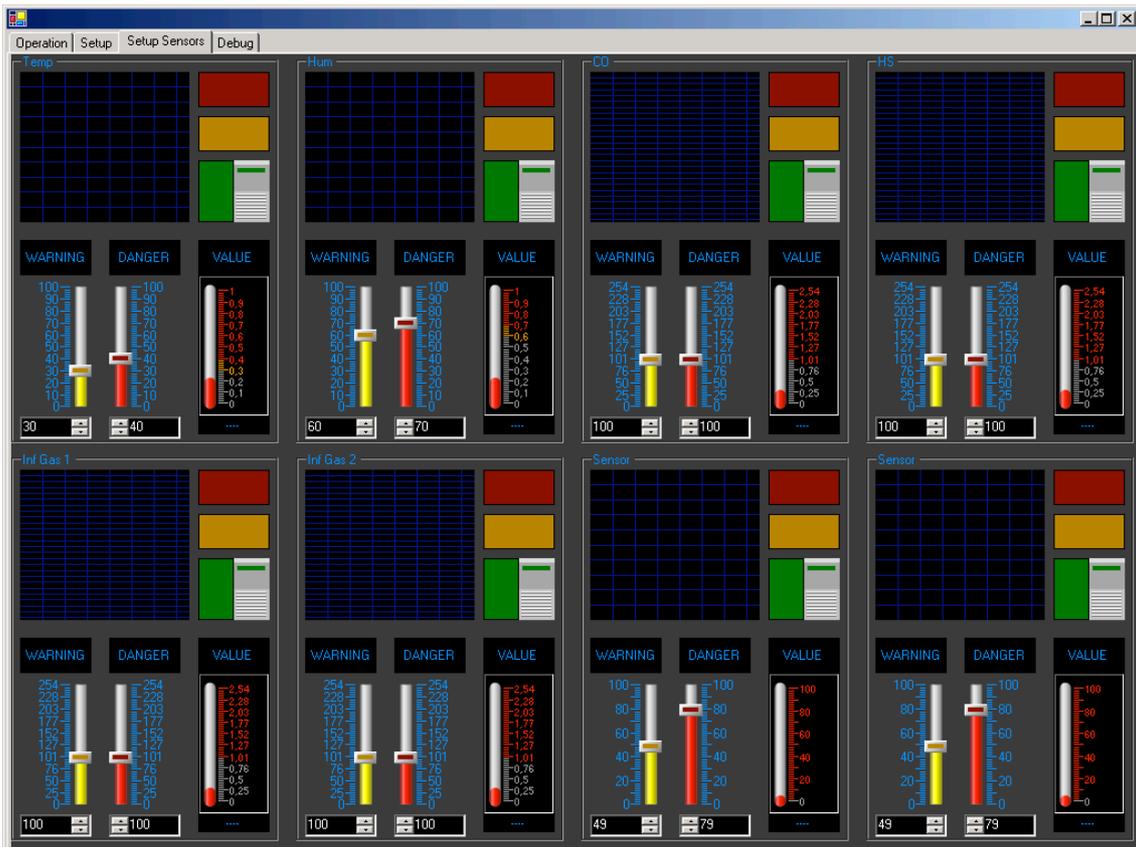


Fig. 62: Second Setup View

On the Second Setup screen (Fig. 62) the user can change the sensors thresholds for *Warning* and *Danger*. In this view it is also possible to turn on or off a specific sensor. For each sensor there is also a history of the last values.



Fig. 64: Game pad used to control the robot

Microsoft Windows operating system has a mature software API for input controllers that does not impose a particular brand or input device on the software. Any input controller compliant with the API can be used. The API allows mapping different functions to each of the buttons and joysticks; this allows the user to customize the behavior of the input controller.

14. Task T9 – Tests in the Lisbon Firefighters Headquarters

14.1 Introduction

Three tests were performed at the Lisbon Fire Fighters test camp. The robot was used in realistic scenarios, namely inside a wrecked pipe (Fig. 65) and through a set of stairs (Fig. 66):



Fig. 65: Robot going through a wrecked pipe to find a person



Fig. 66: Robot going downwards through stairs

In the sequel, rather than focusing on each test performed, we describe the main conclusions reached with this prototype, on three main areas: electronics, mechanics and software. Whenever appropriate, the circumstances at which problems revealed during the experiments are detailed.

14.2 Electronics

The LEDs on the frontal arm obfuscate the cameras, because they reflect on the polycarbonate. A solution to this problem is to separate the lights from the servo mechanism, remove their lens and attach them directly to the polycarbonate, while maintaining the plastic material that surround them. This reduced the glaring on the cameras to usable levels and it still produces enough light even when the cameras are pointing outwards.

The gas sensors were calibrated, using a bottle featuring a mixture of each of the reference gases, namely:

Reference GAS	Percentage available in the bottle
H ₂ S	25ppm
CO	50ppm
Methane	50% LEL

Table 28: Percentages of each gas on the bottle

These values represent the warning percentage, that is to say, the value at which the measuring equipment should give a warning on the presence of any of these gases. Applying the bottle output to each of the sensors, the measured circuit at the interface is registered and was used to calibrate the gas sensors at the interface.

A future evolution of the robot should use a stronger docking mechanism motor, since the current limit currently has to be pulled to 90% of its maximum for satisfactory results, giving small margin to detect jamming.

The robot power autonomy was according to what was expected, from 1 hour in continuous operation to about 2 hours in intermittent or low speed operation.

When operated through the cable, using two NiMH 9Ah batteries in series (around 25V total nominal voltage), the robot behaved correctly and no instabilities occurred due to the long cable.

The main problem detected at hardware level was the general heating of the robot. The most sensible part was the arm motor that reached temperatures well beyond 70 degrees Celsius. There are a number of reasons for this:

- The real voltage of the batteries is 32V, a value 8V superior to the nominal voltage of the motor;
- The speed limit was taken to its maximum at the interface;
- The robot was operated more than two hours continuously, with no time for natural cooling.

The arm motor did stopped working and needed to be replaced. The new motor is similar, but features a 36V working voltage. A future evolution of the robot shall have a metal structure that embraces the motor and gearbox, thus connecting it thermally to the chassis in a more efficient way. This should be sufficient to avoid the arm motor thermal breakdown. Nevertheless, as soon as the new motor arrives, the temperature

and humidity sensor will be attached to the motor to better characterize its thermal response.

Another possible solution to the thermal problem is to attach a fan close to the docking mechanism hole, on a non – intrusive way (and thus less efficient), to use it when open for cooling purposes, forcing the air to circulate inside the robot.

The webcams gave usable images, but their automatic white balance feature is less than uniform between different cams (same model). The thermal camera helps finding people even when the webcams fail to give a clear image, as it can be seen on Fig. 67, where a person can easily be detected behind the pipe.

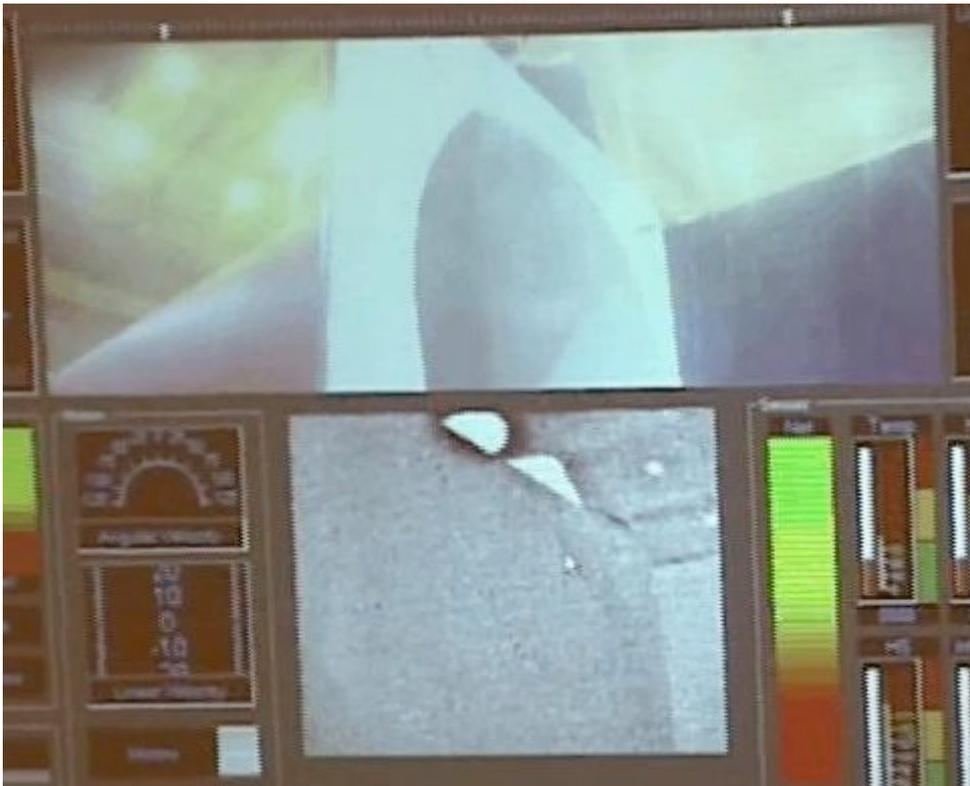


Fig. 67: Thermal camera image

The hard disk gained some defective sectors. It is unclear if this is due to vibration or excessive heating. It was replaced by a Samsung hard disk, of lower velocity - 4200rpm and lower power consumption and this translated on less heating. This replacement hard disk is working properly. The compact flash also gained bad sectors and no replacement was bought until now.

The remaining sensors and actuators behaved according to what was expected and required no special care during the tests.

Future evolutions of the robot may use, besides the extra cooling and more powerful docking motor, smaller electronics boards. These will be possible using SMD technology that IdMind is currently developing.

14.3 Mechanics

The robot, in spite of being a prototype, is very solid and robust. It has withstood all sorts of mechanical tests, namely small impacts, small falls and vibration. It was tested in various types of terrains, namely concrete, grass, carpet, asphalt, soil, debris and inside empty water pipes. It was tested also against several types of obstacles like

stairs, rocks, 45° slopes, boxes, tires and wood bars. All tests in the Lisbon fire fighters department and in two other events have totalized more than 40 hours of robot operation. The robot mechanics proved to be of excellent durability, nevertheless there are some improvements that should be implemented in future robot evolutions.



Fig. 68: Raposa at Segurex

The robot has good traction and the motor power is impressive, the robot velocity is very good given the robot dimensions and mass.

The moving capabilities are very good in the majority of surfaces, but in grass and in terrains cluttered with big debris the robot maneuverability is poor. In the grass the major problem is the accumulation of grass material between the tracks and the motorized pulley wheels. This material accumulation forces the tracks to slide out of the pulley wheels. In debris the robot has the problem of low ground clearance, being stuck too easily in the middle. This problem can be attenuated by using the frontal arm to unstuck the robot and thus continue the mission.

In stairs and 45° slopes the robot not only moved without difficulty but also maneuvered with ease.

The robot was also tested with the cable attached, moving without problems. In excess of 15 meters of cable, the robot starts to have problems in moving, mainly due to the cable being stuck around obstacles and building corners. We didn't do a test of suspending the robot by the cable due to the fact that the cable and docking mechanism were not well prepared for that.

We have conducted several tests for remotely grabbing the cable which have been successful, the camera has proven to be in the right position and the docking is not difficult.

In the tests we were able to remotely operate the robot inside empty water pipes filled with obstacles, like tires and debris. The robot transposed more than 20 meters inside the pipe which was an impressive achievement. Inside the test building we remotely operated the robot with and without cable in distances of more than 30 meters.



Fig. 69: Raposa on Empty water pipe

We didn't try to run the robot on wet or water surfaces as the current chassis is not water proof. This can be done easily with silicone based glue.

After the intensive tests the tracks have lost their tension and now, in surfaces with more traction and when the robot squids steer, the tracks fall off often.

In future versions we can improve some aspects of the mechanical construction, namely the weight of the robot, what can be done using lighter materials on the robot structure. Also the motor mounts should be improved and included a track tensioning system. For the grass, a protective track shield should be applied on the robot sides. Some sort of impact suspension on the frontal arm gearbox should be used. Finally the ground clearance of the robot must be slightly increased.

14.4 Software

The usability of the system proved to be very good; it was operated by several firefighters and after a short briefing of 5-10 minutes followed by a small conduction test, they were able to drive it. The game-pad style joystick is easy to grab on the field and operators learn fast how to use it. The graphical user interface is easy to understand, the value indication, the warning and the danger feedback from the various sensors are easy to interpret even for first time lookers.

The robot responsiveness to commands is very fast and can be considered to be real time, even with bad communications. When the communication is lost the robot safely stops and as soon as communications is reestablished the operator regains immediate control over the robot.

After some tuning on the video parameters, the video feedback turned to be very good, smooth and with a low delay. Nevertheless, the color is not so good due to the compression and sometimes the web cams do not perform proper white balance.

When confronted with wireless communications difficulties the video adapts well until a certain point, after which it is better to have only one video stream open. It was

observed that even on the worst possible conditions the video is still usable, but the frame rate and latency are bad; 1 fps or lower with 1s, or more video latency.

The video normally is slow to start (1s), so when changing/choosing cameras, the operator has an uncomfortably sense of non-responsiveness by the system. This is a limitation of the codec not easily solvable, but in normal operation it poses no real problem. The exception being when the communication is very intermittent: the video transmission stops and it is not restarted unless explicitly commanded by the operator, so that it does not affect the robot responsiveness in such hard communication conditions.

Some occasional PC reboot were observed during the tests that are not acceptable in a real situation. Among the possible causes for this we can account for the hard disk problem mentioned earlier, the lack of thermal dissipation on the robot and possibly a bad electrical contact that reveals itself when the robot falls. Nerveless this is a sensible problem and should be addressed carefully in future robot evolutions, further evaluating alternative solid state disks.

15. References and Bibliography

- [1] Atheros Communications, "802.11 Wireless LAN Performance White Paper".
- [2] Palantir : a multichannel interactive streaming solution
<http://www.fastpath.it/products/palantir/>
- [3] EZ Connect™ Wireless Ethernet Adapter SMC2670W
http://www.smc.com/drivers_downloads/library/SMC2670W_DS.pdf
- [4] EL-2511CD PLUS EXT2 Long Range Wireless PC Card with Two Antenna Jacks
<http://www.netgate.com/Library/Spec2511CD+.pdf>
- [5] D-Link DWL-R60AT Indoor 6 dBi Microstrip Antenna
http://www.dlink.com.au/products/wireless/antenna/DSr60_at.pdf
- [6] FT232BM USB UART (USB - Serial) I.C.
<http://www.ftdichip.com/Documents/DataSheets/ds232b17.pdf>
- [7] DLP-USB232M USB – SERIAL UART Interface Module
<http://www.dlpdesign.com/usb/dlp-usb232m13.pdf>
- [8] Maxon Motor 4-Q-EC Servoamplifier 50 VDC
<http://ftp.maxonmotor.com/Public/Download/Product/Pdf/132368-e.pdf>
- [9] PT6302 2A adjustable positive step-down integrated switching regulator
<http://www-s.ti.com/sc/ds/pt6302.pdf>
- [10] Alessandro Rubini Evaquo Framegrabber Driver (discontinued)
<ftp://ar.linux.it/pub/evacquo/>

Appendix 1 – Battery Chemistry Comparison

The first step for a correct choice is to evaluate the different battery chemistries commonly available today for new projects:

- Lead Acid:
 - Inexpensive;
 - Well established technology;
 - No memory effect;
 - Low weight and volume energy density (they became heavy and large for high capacities);
- NiCd:
 - Hazardous to health and being replaced by NiMH. Discarded from now on.
- NiMH:
 - Middle priced;
 - Almost no memory effect;
 - Medium weight (and volume) energy density;
- Li-Ion:
 - highest weight energy density;
 - no memory effect;
 - Expensive;
 - Risk of explosion if allowed to operate off limits;
 - No memory effect;
 - High weight (and volume) energy density (means lighter and smaller batteries);

The next table gives a more detailed generic comparison of the different types of chemistries (even the innovative Fuel Cells).

	NiCd	NiMH	Lead Acid	Li-ion	Li-ion polymer	Reusable Alkaline
Gravimetric Energy Density (Wh/kg)	45-80	60-120	30-50	110-160	100-130	80 (initial)
Internal Resistance (includes peripheral circuits) in mW	100 to 2001 6V pack	200 to 3001 6V pack	<1001 12V pack	150 to 2501 7.2V pack	200 to 3001 7.2V pack	200 to 20001 6V pack
Cycle Life (to 80% of initial capacity)	15002	300 to 5002,3	200 to 3002	500 to 10003	300 to 500	503 (to 50%)
Fast Charge Time	1h typical	2-4h	8-16h	2-4h	2-4h	2-3h
Overcharge Tolerance	moderate	low	high	very low	low	moderate
Self-discharge / Month (room temperature)	20%4	30%4	5%	10%5	~10%5	0.3%

Cell Voltage (nominal)	1.25V6	1.25V6	2V	3.6V	3.6V	1.5V
Load Current - peak - best result	20C 1C	5C 0.5C or lower	5C7 0.2C	>2C 1C or lower	>2C 1C or lower	0.5C 0.2C or lower
Operating Temperature (discharge only)	-40 to 60°C	-20 to 60°C	-20 to 60°C	-20 to 60°C	0 to 60°C	0 to 65°C
Maintenance Requirement	30 to 60 days	60 to 90 days	3 to 6 months ⁹	not req.	not req.	not req.
Typical Battery Cost (US\$, reference only)	\$50 (7.2V)	\$60 (7.2V)	\$25 (6V)	\$100 (7.2V)	\$100 (7.2V)	\$5 (9V)
Cost per Cycle (US\$) ¹¹	\$0.04	\$0.12	\$0.10	\$0.14	\$0.29	\$0.10-0.50
Commercial use since	1950	1990	1970	1991	1999	1992

Source: <http://www.buchmann.ca/chap2-page2.asp>

The table on the next page compares a series of commercial available solutions.

Battery	Capacity [Ah]	Max Current [A]	Voltage [V]	Dimensions WxLxH [mm]			Weight [kg]	Price	Cells /Pack	Pack Price	Pack Voltage [V]	Packs	Total Price	Total Weight [kg]	Cell Volume	Total Volume	Volume Density	Weight Density
NiCd Mono	5	1,2	35	35	61	0,2	10,95	10	109,5	12	2	219	4	74725	1494500	1,61E-05	6,00	
NiMH Panasonic	6,8	1,2	35	35	61	0,17	17,95	10	179,5	12	2	359	3,4	74725	1494500	1,61E-05	7,06	
NiMH Conrad	8	1,2	35	35	61	0,16	14,22	10	142,2	12	2	284,4	3,2	74725	1494500	1,61E-05	7,50	
NiMH GP	7	1,2	35	35	61	0,17	14,95	10	149,5	12	2	299	3,4	74725	1494500	1,61E-05	7,06	
NiMH Panasonic II	7	1,2	35	35	61	0,157	12,95	10	129,5	12	2	259	3,14	74725	1494500	1,61E-05	7,64	
NiMH PSE	8,4	1,2	32	32	61	0,15	9,988	10	99,88	12	2	199,8	3	63558,3	1271166	1,89E-05	8,00	
LeadAcid Panasonic	7,2	12	151	102	94	3,9	49,95	1	49,95	12	2	99,9	7,8	1447788	2895576	8,29E-06	3,08	
LeadAcid Voltcraft	7	12	151	65	95	2,66	21,95	1	21,95	12	2	43,9	5,32	932425	1864850	1,29E-05	4,51	
LeadAcid Sonnenschein	6,5	12	152	65	94	2,6	38,95	1	38,95	12	2	77,9	5,2	928720	1857440	1,29E-05	4,62	
Li-Ion 5Ah PSE	5	10	3,6	33	33	61	0,125	54	4	216	14,4	2	432	1	66429	531432	5,42E-05	28,80

Li-Ion batteries are still an under development technology, subsequently the energy density of different brands can still vary, contrary to lead acid, where they all feature the same values. Several commercial solutions were considered. The first choices were laptop batteries, the cheaper Li-Ion batteries available today. Those batteries, however, have standardization on dimensions that proved incompatible with our available free space. After a careful market research, Power Supply Europe (PSE) very high density Li-Ion cells, packed in groups of four are the best solution.

Appendix 2 – Gas Sensor Characterization

A general characterization of the gas sensors is presented. The first set of sensors analyzed shall be the TGS8XX family. They all share one aspect in common: before the sensor reading can be considered stable in order to perform a calibration, they require a pre-heating period of two to four days.

The other sensor analyzed shall be the TGS2442, designed for carbon monoxide detection.

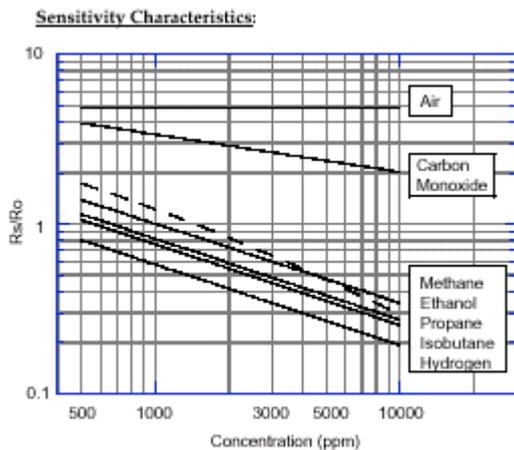
The sensed resistance varies not only with the gas concentration, but also with temperature and humidity. In this appendix worst case scenarios are considered, to determine the full range of values possible and to determine the number of bits necessary to represent the measured values.

TGS 813 - Detection of Combustible Gases

This first sensor is a generic combustible gases detector.

Item	Symbol	Condition	Specification
Sensor Resistance	R_S	Methane at 1000ppm/air	5k Ω ~ 15k Ω
Change Ratio of Sensor Resistance	R_S/R_0	R_S (Methane at 3000ppm/air)	0.60 \pm 0.05
		R_S (Methane at 1000ppm/air)	
Heater Resistance	R_H	Room temperature	30.0 \pm 3.0 Ω
Heater Power Consumption	P_H	$V_H=5.0V$	835mW (typical)

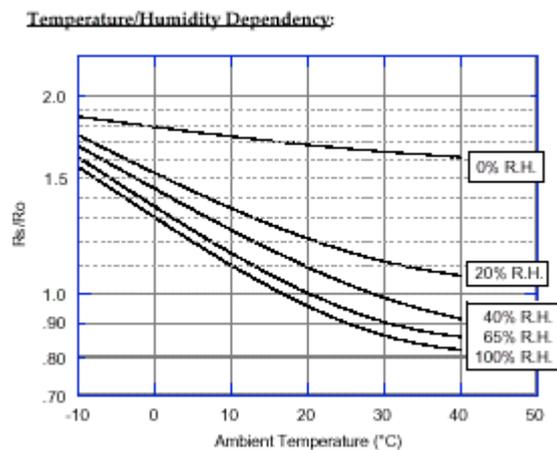
Table 29: TGS 813 Sensor Characteristics



The Y-axis is indicated as sensor resistance ratio (R_S/R_0) which is defined as follows:

R_S = Sensor resistance of displayed gases at Various concentrations

R_0 = Sensor resistance in 1000ppm methane



The Y-axis is indicated as sensor resistance ratio (R_S/R_0) defined as follows:

R_S = Sensor resistance at 1000ppm of methane at various temperatures/humidities

R_0 = Sensor resistance at 1000ppm of methane at 20°C and 65% R.H.

The first graphic shows that the sensor is also capable of detecting carbon monoxide. But since there is already a specific sensor for this type of gas and to detect it using this one would require a wider sensibility, values larger than $2R_S/R_O$ are discarded.

The worst case scenarios are:

- Ethanol 500ppm : $R_S/R_O \sim 1.8$
- Hydrogen 10000ppm: $R_S/R_O \sim 0.2$

The value given on Table 29 gives us the maximum variation for R_S at a given concentration (the range takes in account temperature and relative humidity variations). To give the total range of values of interest, we multiply those values by the constants taken from the graphic above:

- Ethanol 500ppm : $15k\Omega \times 1.8 = 27k\Omega$
- Hydrogen 10000ppm: $5k\Omega \times 0.2 = 2.5k\Omega$

The dynamic range necessary is $27k\Omega / 2.5k\Omega = 10.8$. So, if the smallest achievable resistance is used as a unit only 4 bits are necessary to cover the full scale. The analysis of the resolution needed to measure $27k\Omega$ is done latter.

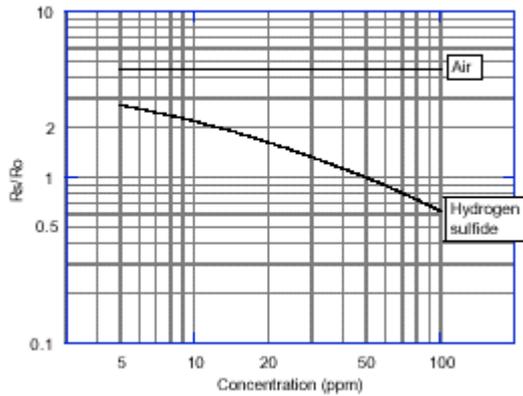
TGS 825 - Sensor for Hydrogen Sulphide

This second sensor is a Hydrogen Sulphide detector. This gas is generated by the decomposition of organic matter.

Item	Symbol	Condition	Specification
Sensor Resistance	R_S	Hydrogen sulphide at 50 ppm/air	$3k\Omega \sim 30k\Omega$
Change Ratio of Sensor Resistance	R_S/R_O	$R_S(H_2S \text{ at } 50\text{ppm/air})$ $R_S(H_2S \text{ at } 10\text{ppm/air})$	0.45 ± 0.15
Heater Resistance	R_H	Room temperature	$38.0 \pm 3.0\Omega$
Heater Power Consumption	P_H	$V_H = 5.0V$	660mW (typical)

Table 30: TGS825 Sensor Characteristics

Sensitivity Characteristics:

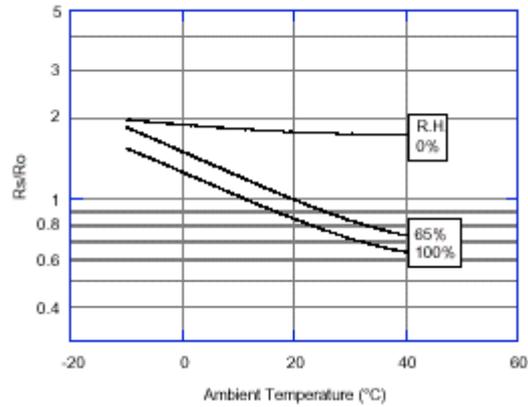


The Y-axis is indicated as sensor resistance ratio (R_s/R_o) which is defined as follows:

R_s = Sensor resistance of displayed gases at various concentrations

R_o = Sensor resistance at 50ppm of hydrogen sulfide at 20°C and 65% R.H.

Temperature/Humidity Dependency:



The Y-axis is indicated as sensor resistance ratio (R_s/R_o), defined as follows:

R_s = Sensor resistance at 50ppm of hydrogen Sulfide at various temp./humidity's

R_o = Sensor resistance at 50ppm of hydrogen Sulfide at 20°C and 65% R.H.

Once again we assume the Sensor Resistance indicated in Table 30 includes all humidity and temperature variations and uses those values to calculate the sensor resistance full range based on graphic 1.

- 5ppm : $3k\Omega \times 0.6 = 1.8k\Omega$
- 100ppm: $30k\Omega \times 2.8 = 84k\Omega$

The dynamic range necessary is $84k\Omega / 1.8k\Omega = 46 : 6$ bits.

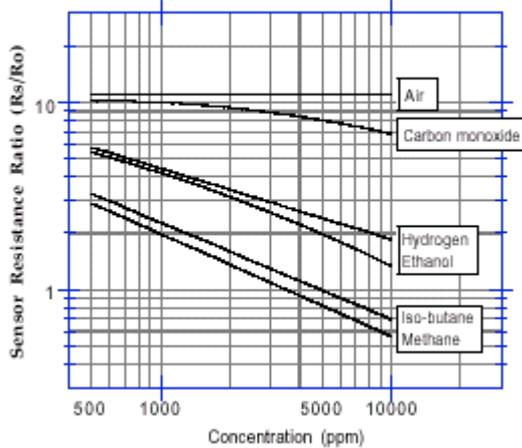
TGS 842 - Detection of Methane

This third sensor is a similar to the first, since it detects the same family of gases. But its response is different, showing bigger sensibility to methane. So, all calculations on this sensor are done having methane detection in mind.

Item	Symbol	Condition	Specification
Sensor Resistance	R_s	Methane at 1000ppm/Air	$3k\Omega \sim 15k\Omega$
Change Ratio of Sensor Resistance	R_s/R_o	R_s (Methane at 3000ppm/air) R_s (Methane at 1000ppm/air)	0.55 ± 0.05
Heater Resistance	RH	Room temperature	$30.0 \pm 3.0\Omega$
Heater Power Consumption	PH	$V_H=5.0V$	835mW (typical)

Table 31: TGS842 Sensor Characteristics

Sensitivity Characteristics:

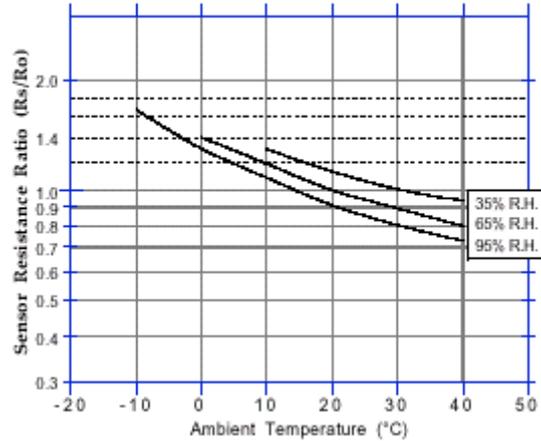


The Y-axis is indicated as sensor resistance ratio (R_S/R_O) which is defined as follows:

R_S = Sensor resistance of displayed gases at Various concentrations

R_O = Sensor resistance in 3500ppm methane

Temperature/Humidity Dependency:



Again, the Y-axis is indicated as sensor resistance ratio (R_S/R_O), defined as follows:

R_S = Sensor resistance at 3500ppm of methane

at various temperatures/humidity's

R_O = Sensor resistance at 3500ppm of methane

at 20°C and 65% R.H.

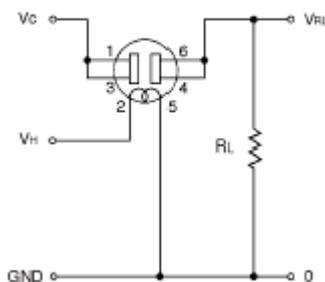
Doing the same calculations as above (optimizing for methane).

- 500ppm : $3k\Omega \times 0.55 = 1.65k\Omega$
- 100ppm: $15k\Omega \times 3 = 45k\Omega$

The dynamic range necessary is $45k\Omega / 1.65k\Omega = 27 : 5$ bits.

Universal Detection Circuit

Basic Measuring Circuit:



Sensor Resistance (R_S) is calculated by the following formula:

$$R_S = \left(\frac{V_C}{V_{RL}} - 1 \right) \times R_L$$

This is the standard measuring circuit. In order to simplify matters, V_C is also 5VDC. This way there is a guaranty that the resulting voltage V_{RL} (the value effectively measured) is always inferior to 5V and thus may be directly measured by the PIC, without any voltage shifting.

Our previous calculations estimated that the shorter resistance to measure is: 1.65kΩ.

With $R_L = 10k$, we obtain a $V_{RL\ Max} = (10) / (10 + 1.65) * 5 = 4.29V$

The largest resistance available is: $84k\Omega$.

With $R_L = 10k$, we obtain a $V_{RL\ Min} = (10) / (10 + 84) * 5 = 0.53V$

The PIC features an 8 bit resolution over 5V scale, which leads to a quantum voltage of:

$$5/256 = 0.02V$$

This value is much smaller than the minimum achievable V_{RL} , so even the smaller gas concentrations can be measured. The useful range of values is:

$$(4.29 - 0.53) / 0.02 = 138 \text{ values.}$$

The conclusion reached with this analysis is that the 8 bit available in the PIC are enough for this application.

By using a $10k\Omega$ R_L we assure the output impedance is always inferior to $10k\Omega$, thus allowing direct connection to the PIC analog inputs.

Load Resistance

The following aspects influence the choice of a R_L resistance:

- The higher R_S values correspond to lower gas concentrations;
- The graph that plots the relation between R_S/R_L versus V_{RL}/V_C clearly indicates that the V_{RL}/V_C slope is at its maximum when $R_S = R_L$, thus giving more resolution around those values; this graph is presented in Fig. 70 and it is sensor independent;
- Using a 8 bit data sampling, from 0 to 5V, the graphic in Fig. 71 represents the number of different samples expectable with each sensor full range (later it is seen that 8 bits is sufficient to measure the smallest values).

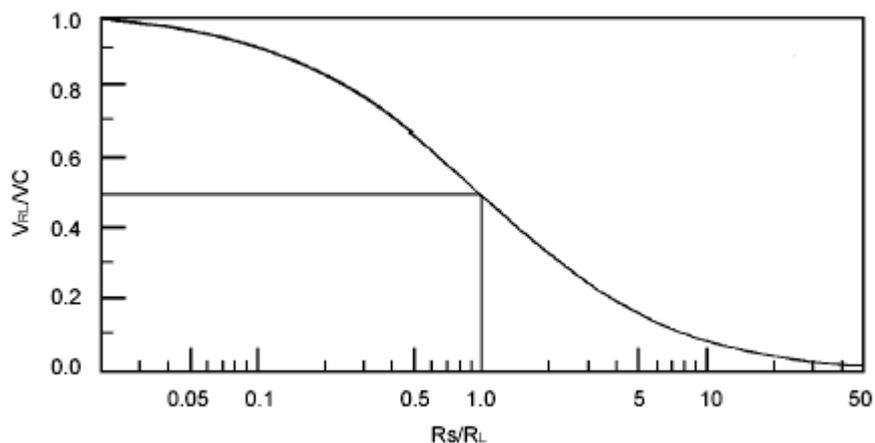


Fig. 70: Resolution versus Load Resistance

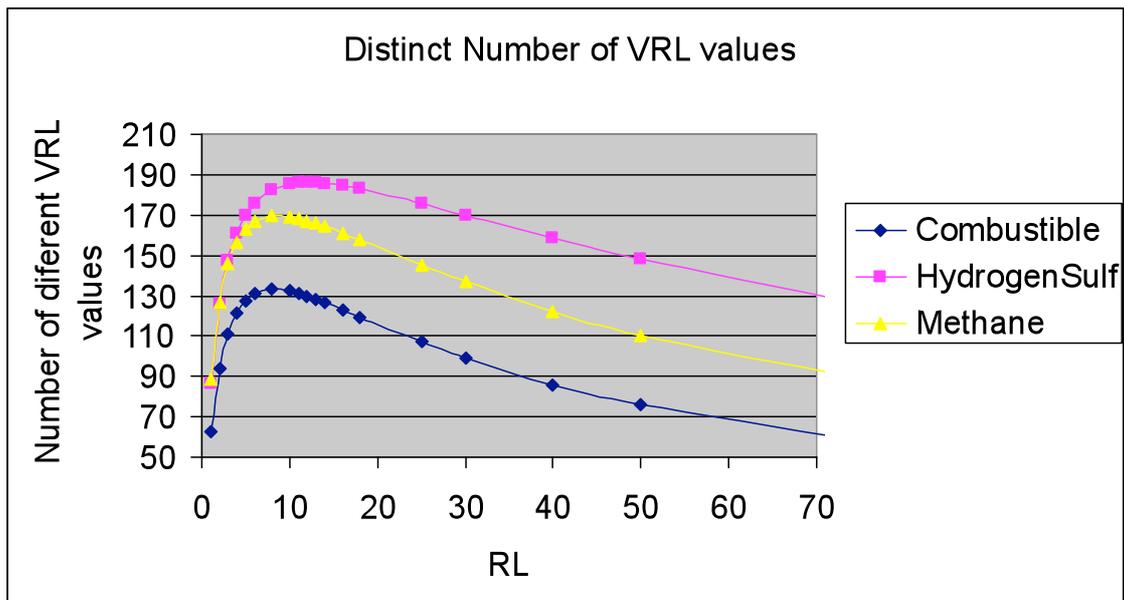


Fig. 71: Number of distinct samples at 8 bits versus Load Resistance

Having these three aspects in consideration, we conclude that any value of R_L between $3k\Omega$ and $30k\Omega$ gives more than one hundred distinct values for each of the sensors. Since this robot profile concerns human rescue, it makes sense to optimize it for low gas concentrations, at values already considered dangerous. Until some feedback about the proper value is received from the RSBL, the $R_L = 10k$ value is used.

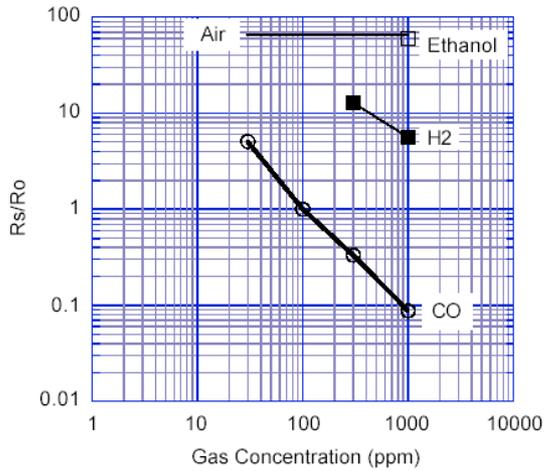
TGS 2442 – Carbon Monoxide

A different class of sensor, optimized for carbon monoxide detection.

Item	Symbol	Condition	Specification
Sensor Resistance	R_s	CO at 100ppm/Air	$10k\Omega \sim 31.6k\Omega$
Change Ratio of Sensor Resistance	β	$\beta = \frac{R_s(\text{CO}, 300\text{ppm})}{R_s(\text{CO}, 100\text{ppm})}$	0.29~0.40
Heater Resistance	R_H	Room temperature	$17 \pm 2.5\Omega$
Heater Power Consumption	P_H		14mW (aver.)

Table 32: TGS2442 Sensor Characteristics

Sensitivity Characteristics:

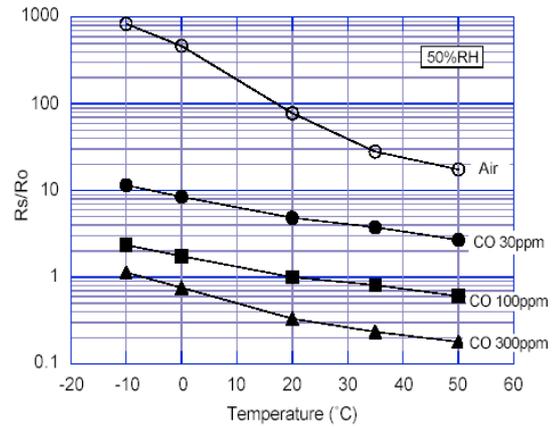


The Y-axis is indicated as sensor resistance ratio (Rs/Ro) which is defined as follows:

Rs = Sensor resistance of displayed gases at various concentrations

Ro = Sensor resistance in 100ppm CO
(estimated $\beta \sim 0.33$)

Temperature/Humidity Dependency:



The Y-axis is indicated as sensor resistance ratio (Rs/Ro), defined as follows:

Rs = Sensor resistance at 30ppm, 100ppm and 300ppm of CO at various temperatures and 50%R.H.

Ro = Sensor resistance at 300ppm of CO at 25°C and 50% R.H.

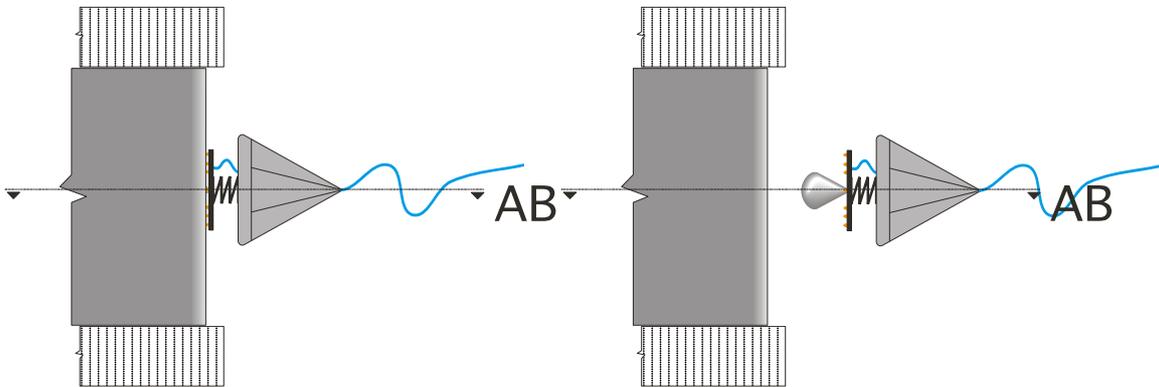
Maximum and minimum expectable resistance values concerning concentrations (relative to the sensitivity graphics):

- 30ppm : $31.6k\Omega * 5 = 158k\Omega$
- 1000ppm: $10\Omega * 0.09 = 900\Omega$

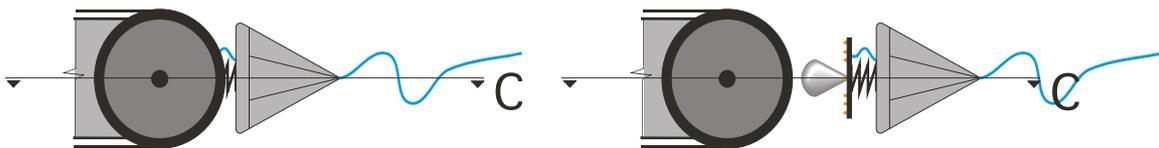
The dynamic range necessary is $158k\Omega / 900\Omega = 175$: 8 bits.

Appendix 3 – Docking System Mechanical Drawing

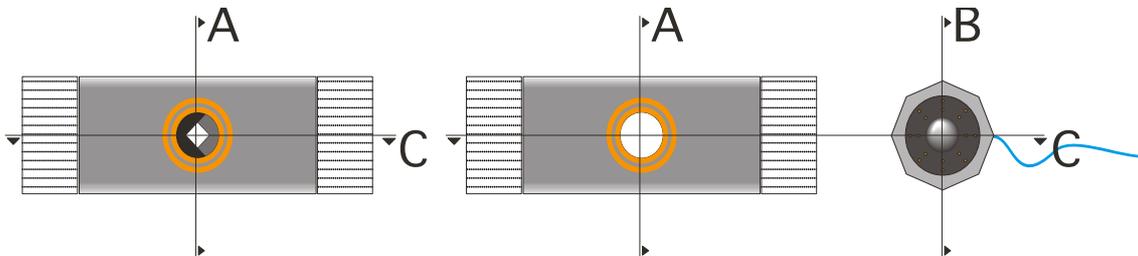
Top View



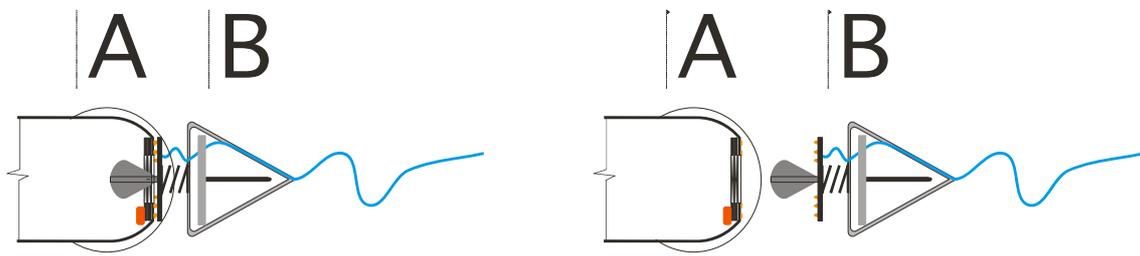
Side View



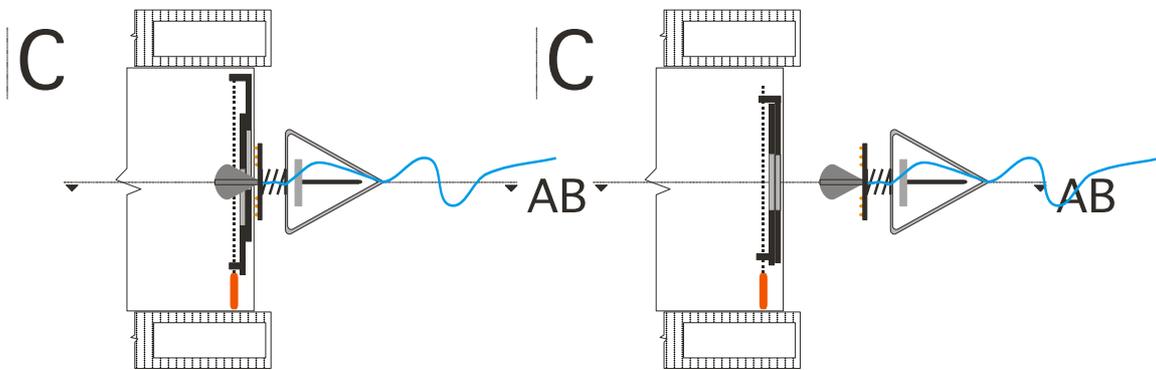
Back View



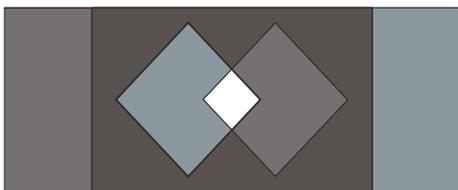
Section Views 'A' and 'B'



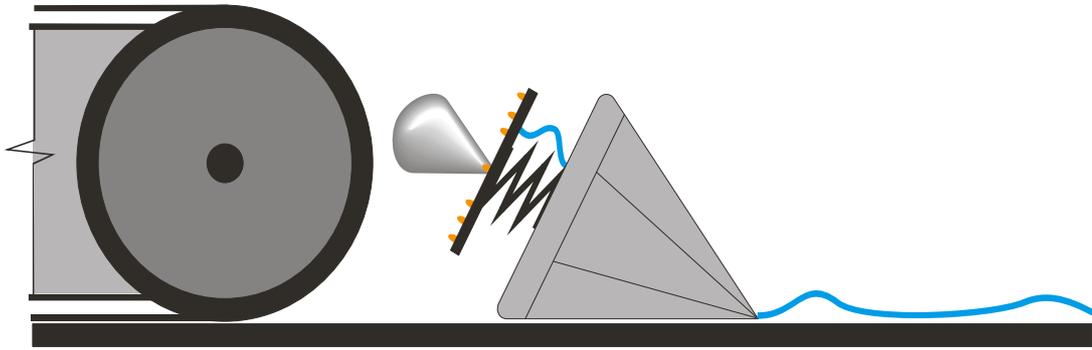
Section View 'C'



Sliding Doors



Released mechanism on the ground



Spring Deformation

