



FROG

FUN ROBOTIC
OUTDOOR GUIDE

Deliverable: D1.4.1

Platform User and Developer Manual (update)

Consortium

UNIVERSITEIT VAN AMSTERDAM (UvA)

YDREAMS - INFORMATICA S.A. (YD)

IDMIND - ENGENHARIA DE SISTEMAS LDA (IDM)

UNIVERSIDAD PABLO DE OLAVIDE (UPO)

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE (ICL)

UNIVERSITEIT TWENTE (UT)

Grant agreement no. **288235**

Funding scheme **STREP**



DOCUMENT INFORMATION

Project	
Project acronym:	FROG
Project full title:	Fun Robotic Outdoor Guide
Grant agreement no.:	288235
Funding scheme:	STREP
Project start date:	1 October 2011
Project duration:	36 Months
Call topic:	ICT-2011.2.1 Cognitive Systems and Robotics (a), (d)
Project web-site:	www.frogrobot.eu

Document	
Deliverable number:	D1.4.1
Deliverable title:	Platform User and Developer Manual (Update)
Due date of deliverable:	M30 - 15 March 2014
Actual submission date:	M30 - 23 March 2014
Editors:	IDM
Authors:	IDM, UPO
Reviewers:	IDM, UT
Participating beneficiaries:	IDM, UPO
Work Package no.:	1
Work Package title:	Robot Specification, Design and Construction
Work Package leader:	IDM
Work Package participants:	IDM, YD, UPO, ICL, UT
Estimated person-months for deliverable:	1
Dissemination level:	Public
Nature:	Report
Version:	2
Draft/Final:	Final
No of pages (including cover):	70
Keywords:	Robot description, electronics, architecture, operation, low-level control loops, safety

Table of Contents

1. Summary.....	5
2. FROG Robot Features and Components.....	6
2.1. FROG Robot Base Platform.....	6
2.2. FROG Upper Structure.....	7
2.3. Sensors.....	8
2.3.1. Perception Sensors	9
2.3.2. Navigation Sensors	9
2.3.3. Interaction Sensors	10
2.3.4. Environment Sensors.....	10
2.3.5. Low-level Safety Sensors.....	10
2.4. Actuators.....	10
2.4.1. Locomotion Motors.....	10
2.4.2. Interaction Actuators	10
3. FROG Robot Architecture.....	12
3.1. FROG Electronic Power Architecture.....	12
3.2. Low-level Communication Architecture.....	14
3.3. High-level Communication Architecture.....	14
4. FROG Robot Electronics.....	16
4.1. Motor Controller Board.....	16
4.2. Sensor and Charger Board.....	18
4.2.1. Sensor&Management Board	19
4.2.2. Power Switch Board	19
4.2.3. Charger Board.....	20
4.3. Sonar Board.....	22
4.4. Charger Docking Station.....	23
4.5. Joystick Wireless Controller.....	24
4.6. Interaction Board.....	26
4.6.1. 3DOF Arm with Laser Pointer.....	27
4.6.2. FROG Eyes.....	28
5. Low-level Control Loops.....	29
5.1. Sensor&Management Low-level Control Loops and Interrupt Driven Events.....	29
5.1.1 Sensor&Management Low-Level Control Loop.....	29
5.1.2. Sensor&Management Interrupt Driven Events.....	33
5.2. Motor Board Low-level Control Loops and Interrupt Driven Events.....	33
5.2.1 Motor board Low-level Control Loop.....	34
5.2.2. Motor Board Interrupt Driven Events.....	36
6. Safety.....	39
6.1. Failure Mode and Effect Analysis (FMEA).....	39
6.2. Safety Measures.....	41
7. Working with the Robot.....	43
7.1. Setting up the Hardware	43
7.2. Wireless Joystick Control.....	43
7.3. Shutting Down the FROG Platform.....	44
7.4. Charging the Batteries.....	44
8. FROG Integration Meetings and Second Year Review Meeting.....	46
9. Tests and Results.....	47
9.1. Detected Problems and Solutions	47
9.2. Analysis of Performance of Low-level Control Loops	48
9.2.1 Sensor&Management Board Low-Level Control Loops and Interrupt Driven Events.....	48
9.2.2 Motor Board Low-Level Control Loops and Interrupt driven Events	49

9.3. Performance Analysis of Robot Odometry and Velocity Control.....	49
9.3.1. Experimental Setup.....	50
9.3.2. Linear and Angular Velocity Commands Calibration.....	52
Annex A. Board Connections.....	54
Annex B. Board Controllers Software Protocol.....	60
B.1. Motor Board Communication Protocol.....	60
B.2. Sensor&Management communication protocol	64
B.3. Interaction Communication Protocol.....	67
Annex C. Robot Platform CAD Measurements.....	69

1. Summary

This deliverable is the robot platform user and developer manual. It explains the way the different components are integrated, where they are connected and how they communicate.

The structure of the document is as follows:

- FROG Robot Features and Components - This chapter describes the main robot features that have been specified in the previous Work Package deliverables. It gives an overview of the different components that have been integrated, their function, placement on the robot, and internet links for the component manuals.
- FROG Robot Architecture - The adopted power and communication architecture is explained. The chapter begins with a description of how the robot components will be powered. A table with the component energy distribution per battery shows an even distribution of the battery power consumption. The last two sub-chapters are dedicated to explaining the low-level communication, showing the way the developed electronics communicate with each sensor and actuator, and the high-level communication between the onboard computers, interaction and navigation sensors and actuators.
- FROG Robot Electronics - All of the electronics that have been projected, designed, produced and programmed within the project are explained.
- Low-level Control Loops - The electronics control boards that are controlling the motors and onboard power all run low-level control loops and interrupt driven events that control the system and also check for faults in the normal operation of the platform. This chapter explains the low-level software running inside the microcontrollers that control the FROG base platform.
- Safety - During the first review meeting, issues of human-robot interaction safety were raised. This chapter shows the safety procedures that have been implemented.
- Working with the Robot - This is a manual with simple instructions for running the robot.
- FROG Integration Meetings - In the first two years of project six integration meetings were conducted, two in the Royal Alcázar in Seville and four in the Lisbon Zoo. The chapter describes the work of each partner during those events and also the setup of the demonstration for the second year review meeting.
- Tests and Results - This chapter describes some of the tests performed with the robot and their results.

Additionally to the manual, three technical annexes have been added:

- Board Connections - This is the manual of connections between each board connector and each cable inside the robot.
- Board Controllers Software Protocol - This section explains the low-level control protocol implemented between the onboard navigation computer and the Sensor&Management and Motor Board and also between the interaction computer and the Interaction Board.
- Robot Platform CAD Measurements - This section shows CAD representations of the dimensions of the robot.

2. FROG Robot Features and Components

In terms of kinematics the platform is a four-wheeled differential drive, where on each side of the robot two wheels are connected to a single motor. This type of kinematics is adequate for an uneven outdoor (or indoor) terrain, and the robot will deal easily with small obstacles.

The following sub-chapters describe the main features and components integrated into the base platform and upper structure, the sensors and finally the actuators. Some of this information has been also included in the previous deliverable D1.3 and some values were updated based on new measurements.

2.1. FROG Robot Base Platform

Robot Base Platform Main Features

- Robot Kinematics: 4-wheeled differential drive (pneumatic tires for uneven terrain)
- Weight without batteries: 39 Kg
- Weight with batteries
 - When using 8 12V/17Ah lead free batteries: 87 Kg
 - When using 8 12V/17Ah LiFePO4 batteries: 56 Kg
- Payload Capacity: 30 Kg
- Height: 1.3m
- Maximum Velocity: 1.6 m/s (low-level software limited to 1.0 m/s)
- Acceleration: 1 m/s² (low-level programmed)
- Emergency Stop Acceleration: 3.3 m/s² (low-level programmed)
- Batteries Autonomy: 4 to 6 hours
- Installed components sensors and actuators:
 - Sensor&Management Boards
 - Motor Control Board
 - Power Switch Board
 - Charger Board
 - Two Hokuyo UTM-30LX lasers
 - Ring of bumpers (with 12 switches)
 - 4 Temperature sensors (two to measure the motors temperature and two to measure the drivers temperature)
 - 1 Temperature and humidity sensor to measure the environment conditions
 - 1 Digital Rain sensor

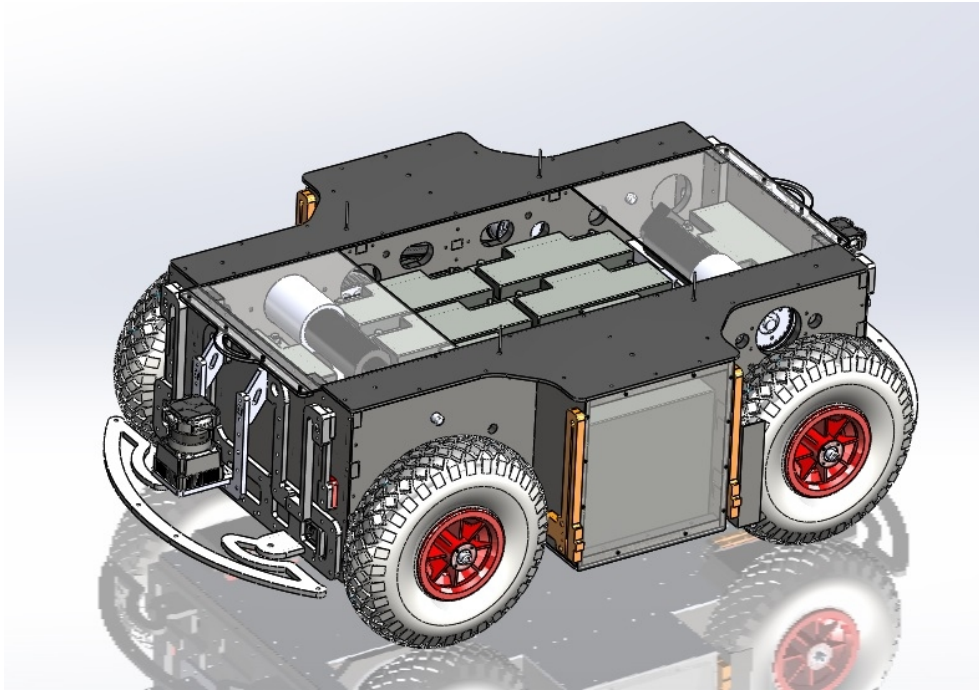


Figure 1: picture of the platform base structure

2.2. FROG Upper Structure

Upper Structure Platform Components

- One 17" touchscreen Monitor
- One Kinetic Xbox camera
- One set of two stereo cameras
- Two stand-alone cameras
- One inertial sensor XSens MTi-G
- One Hokuyo UTM-30LX laser
- One 3DOF motorized arm with a laser pointer
- One projector
- One Ethernet Wireless router
- One Ethernet Switch
- Three mini-ITX computers
- One laptop

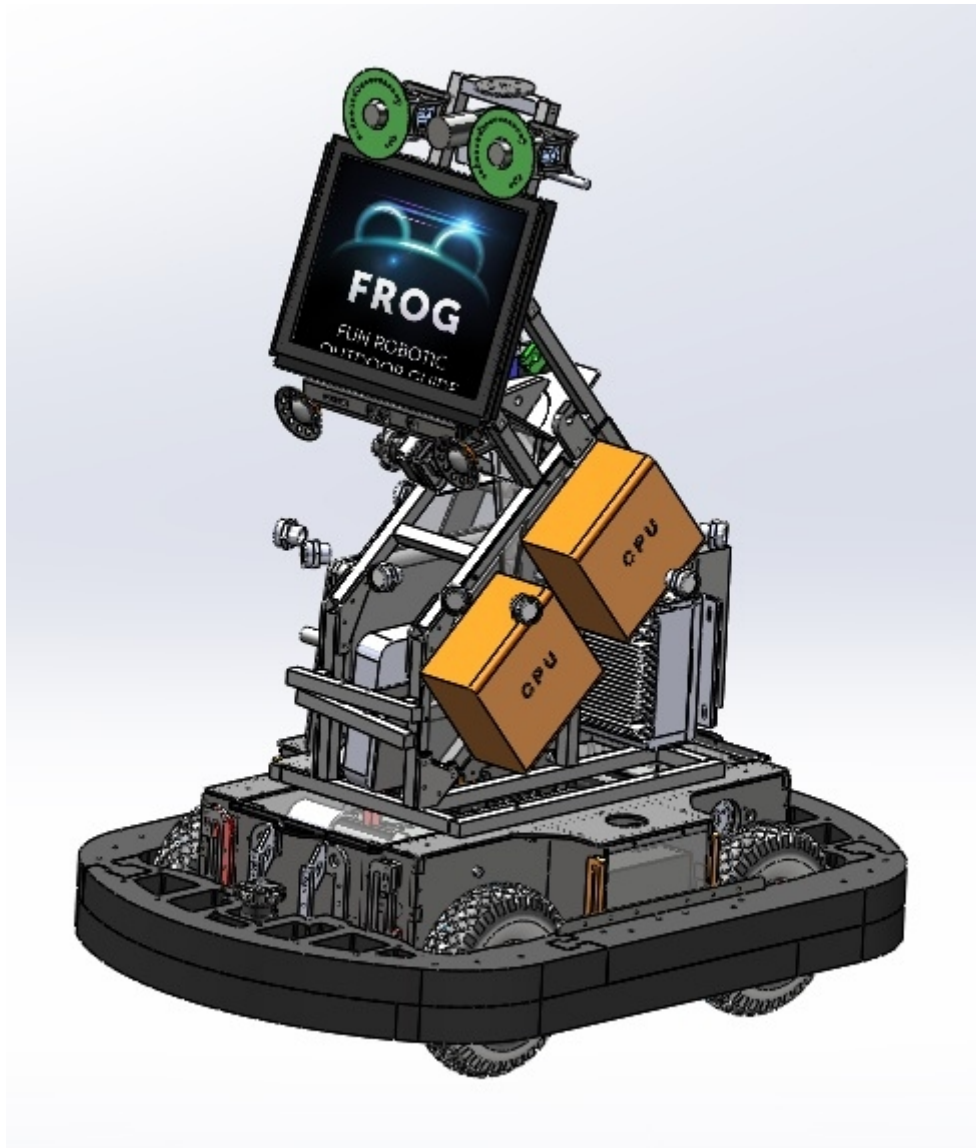


Figure 2: picture of the upper structure

2.3. Sensors

The robot is equipped with perception, navigation, interaction, environment and low-level safety sensors. For **perception** the robot is equipped with cameras, providing a middle range perception (<25m) and short-range perception (1 to 2m). For **navigation** the robot uses encoders to control the velocity of the motors, an inertial sensor with GPS and a stereo camera to determine the position/orientation in the environment, lasers to detect obstacles and for mapping, and a standalone camera for guidance missions. For **interaction** the robot uses a standalone camera for people tracking, face analysis and body gesture recognition, and also microphones. For **environmental** sensing the robot will be equipped with temperature, humidity and rain sensors. Finally, the bumpers and sonar sensors will provide **low-level** safety sensing.

Here follows a list of sensors that are being used on board.

2.3.1. Perception Sensors

The robot makes use of different cameras for navigation of the environment, feature identification, pedestrian detection and tracking, body orientation estimation, face analysis and body gesture recognition. They can also be used to detect changes in the surrounding environment.

- Front Stereo Vision Cameras: Dalsa Genie-HM1400 XDR¹ (2x)
 - Function: localization, obstacle detection, pedestrian detection and body orientation estimation;
 - Position on Robot Platform: looking ahead; 1.2m high
- Front Vision Camera: Dalsa Genie-HM1400 XDR (1x)
 - Function: face analysis and fine body gesture recognition
 - Position on Robot Platform: looking ahead; 1.2m high
- Rear Vision Camera: IDS UI-5240 colour camera
 - Function: AR and people tracking for guidance mission
 - Position on Robot Platform: looking back; 1.2m high

2.3.2. Navigation Sensors

The robot will navigate in the environment while making a fusion of measures provided by different sensors. Outdoors, the robot will be able to use the stereo pair, lasers, GPS (where available), encoders odometry and the inertial sensor to estimate its position and orientation. For obstacle avoidance, mapping and localization it uses the lasers and sonar sensors. The front stereo vision camera will also help in the navigation by detecting obstacles and persons.

- Inertial Sensor IMU with GPS: Xsens MTI-G
 - Function: localization estimation (position and orientation)
 - Position on Robot Platform: in the robot's centre of rotation
- Front 2D laser rangefinder: Hokuyo's UTM-30LX
 - Function: mapping, localization and obstacle avoidance
 - Position on Robot Platform: frontal and horizontal
- Front 2D laser rangefinder: Hokuyo's UTM-30LX
 - Function: obstacle avoidance (3D perception of ramps, slopes, etc)
 - Position on Robot Platform: frontal, tilted or vertical orientation
- Rear 2D laser rangefinder: Hokuyo's UTM-30LX
 - Function: mapping, localization and obstacle avoidance
 - Position on Robot Platform: on the rear and horizontal
- Sonar Sensors: SRM400
 - Function: obstacle detection (e.g.: glass wall or objects)
 - Position on Robot Platform: ring of sonars around the robot

¹ <http://www.teledynedalsa.com/imaging/products/cameras/area-scan/genie/CR-GM0X-H140X/>

2.3.3. Interaction Sensors

The front vision stand-alone camera is used to sense visual user feedback for natural user interaction. There will also be the following sensors.

- Microphone array and 3D: Microsoft Kinect
 - Function: user sound feedback for natural user interaction
 - Position on Robot Platform: turned to the users
- Sunlight Readable 17" Touchscreen
 - Function: user feedback on specific contents
 - Position on Robot Platform: turned to the user

2.3.4. Environment Sensors

The environment sensors are used to detect variations in the weather conditions that can affect the normal operation of the robot. These sensors are: temperature sensor; humidity sensor; and rain sensor.

2.3.5. Low-level Safety Sensors

The fundamental sensors for low-level safety are the sonar sensors, internal temperature sensors, motor current sensing and the bumper ring switches.

2.4. Actuators

The robot will be equipped with locomotion and "interaction" actuators.

For locomotion the robot uses two motors to actuate the 4 wheels of the robot. Each motor will actuate one pair of wheels on one side of the robot (differential drive).

For user interaction there is one touchscreen, stereo speakers, an image projector and a laser pointer actuated by a small 3DOF arm.

2.4.1. Locomotion Motors

The robot locomotion uses two motors to drive the four-wheeled differential drive platform. Each side of the robot has two wheels that are connected to a single motor.

- Two Maxon RE 50 200W 24V motor with a Maxon GP 62A 19:1 Gearbox and encoder HEDS 5540 with 500 pulses
 - Function: provide a locomotion system to the robot
 - Position on Robot Platform: Inside the platform base, connected to the drive system

2.4.2. Interaction Actuators

Here follows the list of interaction devices. The robot is able to display the contents on the interaction monitor or project them over a surface.

- 17" Sunlight Readable Monitor with Touchscreen²
 - Function: interaction with displayed contents (e.g., AR contents)
 - Position on Robot Platform: front of the robot

² http://www.litemax.com/Download/Datasheet//Platfrom/SLO/DS_SLO1768_20130418_V1.pdf

- Video Projector³
 - Function: projection of contents on points of interest (POIs)
 - Position on Robot Platform: projecting to the rear and/or front of the robot
- 3-degree of freedom arm with a laser pointer
 - Function: point to POIs
 - Position on Robot Platform: will be mounted on the top of the robot
- Eyes green LED lights
 - Function: show robot expressions
 - Position on Robot Platform: mounted around the stereo camera lenses
- Stereo Speakers
 - Function: sound exhibition of contents; robot communication
 - Position on Robot Platform: turned to the user

3 http://www.casio.com/products/Projectors/Slim_Projectors/XJ-A251/content/Technical_Specs/

3. FROG Robot Architecture

This chapter describes the power and communication architecture implemented in the FROG robot.

The FROG robot is an autonomous robot able to navigate in outdoor and indoor environments, with no need for human intervention. To power all the components, it will use up to 8 12V (17-20Ah) batteries. The first subchapter explains the power distribution inside the FROG robot.

The second subchapter shows the low-level communication architecture. The electronics that will control the motors, read sensors and actuate motors are considered to be low-level electronics.

Finally the third subchapter presents the high-level communication architecture. A Ethernet router and a switch allow the creation of a local network that connects all the computers in the platform. Each computer will connect to different devices, such as monitors, cameras, projectors or electronic boards, by the use of different technologies such as USB, HDMI or Ethernet connections.

3.1. FROG Electronic Power Architecture

The robot can be powered by several 12V 17-20AH batteries. It uses one pair of 12V batteries connected in series to deliver 24V to the motor drivers. Up to 6 other batteries to provide energy to all the other computers and electronic components. An individual charging unit is used inside the robot to charge each battery. The batteries and the power in the robot will be managed by a Sensor&Charger Board that measures the battery levels, battery charge, and also controls the units (motors, sensors, actuators and inverters) powered by the batteries.

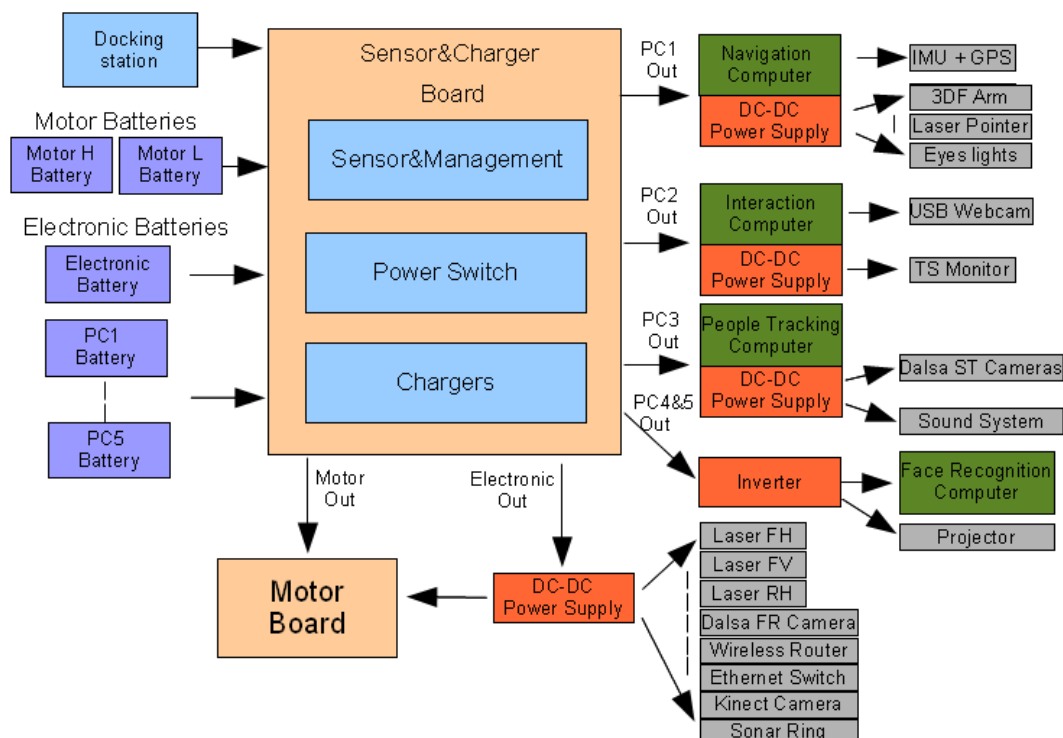


Figure 3: Onboard FROG Robot Power Architecture

All of the onboard electronic systems can be powered by the battery system. The ATX computer power supply will provide regulated voltages (from 5V to 12V) and it will be also included an inverter able to provide 600W 230V AC to any device requiring AC power supply (e.g. video projector) will also be included. Figure 3 depicts the onboard power architecture. Several DC-DC converters will also be used to provide the necessary regulated power for the lasers, cameras and other DC-DC powered devices.

Table 1 shows the equipment that is connected to each battery and the nominal current and power consumption.

In charging mode all the batteries will be put on charge and some of the devices will be powered by the 18V from the charging station. The motor drivers and all the equipment powered by the inverter will not be connected to the 18V of the power station.

Battery Name	Powered item	Nominal Current	Power
Motor H (4Ah)	24V High Side Motor Driver Power	4Ah	48W
Motor L (4Ah)	24V Low Side Motor Driver Power	4Ah	48W
Electronics (4.2Ah)	Management Board	0.45Ah	5.4W
	Frontal Horizontal Hokuyo UTM-30LXLaser	0.7Ah	8.4W
	Frontal Vertical Hokuyo UTM-30LX Laser	0.7Ah	8.4W
	Rear Horizontal Hokuyo UTM-30LX Laser	0.7Ah	8.4W
	Dalsa Genie-HM1400 XDR front face recognition camera	0.35Ah	4W
	Wireless Ethernet Router	0.5Ah	6W
	Ethernet Switch	0.3Ah	3.6W
	Xbox380 Kinect Camera	0.4Ah	4.8W
	Sonar ring	0.1Ah	12W
PC1(3.15Ah)	Navigation Computer	2.0Ah	24W
	Inertial sensor XSens MTi-G (by the Computer USB port)	0.1Ah	1.2W
	3-Degree of freedom arm	0.5Ah	6W
	Laser Pointer	0.15Ah	1.8W
	Eyes LED lights	0.4Ah	4.8W
PC2 (4.3Ah)	Interaction Computer	1.1Ah	13.2W
	HD Webcam	0.3Ah	3.6W
	Touchscreen Monitor	2.9Ah	35W
PC3 (4.8Ah)	People tracking computer	2.5Ah	30W
	Dalsa Genie-HM1400 XDR stereo cameras	0.7Ah	8.4W
	Sound System	1.6Ah	20W
PC4 and PC5 (41Ah)	Front face recognition computer (through the 600W Inverter)	16Ah	200W
	Video Projector	25Ah	270W (Eco Off) / 190W(Eco On Level 1) / 130W(Eco On Level 2)

Table 1: the 8 FROG batteries and the systems they each power

3.2. Low-level Communication Architecture

The onboard robot navigation computer communicates with the two boards (Sensor&Charger Board and the Motor Controller Board) using 2 USB ports. In each board there are USB-to-RS232 converters that convert the USB data packages to serial RS232 packages for the boards' controllers.

Each board controller will communicate with the other allowing the exchange of information between them. This communication channel will allow the execution of low-level behaviours, for example, react against an imminent collision, enter into charging mode with motors shut down, reduce the motors' velocity when the batteries are low, or react to changes that can affect the robot's operation, which is fundamental to the improvement of the overall system dependability.

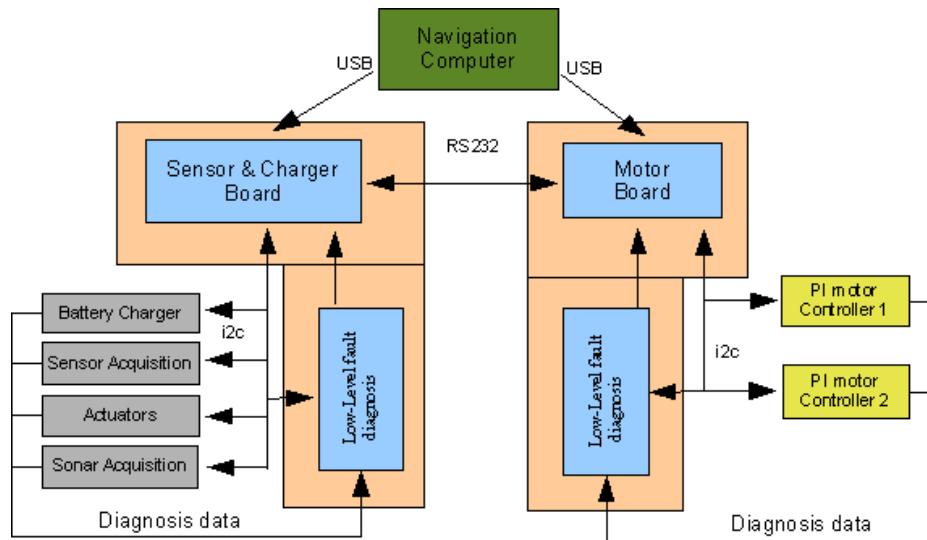


Figure 4: Low-level Communication Architecture

The main controller from the Sensor&Charger Board will communicate with other microcontrollers using Inter-Integrated Circuit (I2C) communication ports. The main controller will act as the master and the other microcontrollers will behave like slaves. The Sensor&Charger Board will connect to the battery charger, sensor, devices actuators and sonar acquisition boards. The Motor Controller Board will connect to the PI Motor controllers and also to temperature sensors.

Each controller will have a low-level fault diagnosis that will check the operation state of each microcontroller and also monitor all the communication between the devices. The low-level communication architecture is depicted in Figure 4.

3.3. High-level Communication Architecture

The FROG Robot has a local Ethernet network installed. A Wireless Ethernet router provides the IP address to all of the computers and a 1Gbit Ethernet switch will allow the exchange of messages between all the computers. The Navigation, Interaction and People Tracking computers connect to the Ethernet switch using Ethernet cables. The Face Recognition must use a wireless connection to the router to perform the data exchange.

External computers can also be connected to the FROG robot by using the wireless Ethernet connection with the Router.

The Navigation Computer is connected to the navigation sensors and to the platform board controllers using six USB ports.

The Interaction computer will connect to the Touchscreen Monitor using the DVI output, to the Projector using the HDMI output, to the Sound System using the audio line out, to the AR camera using an ethernet connection, and will use 2 USB connections to connect to the 3DOF Arm with the laser pointer and the eyes' lights.

The People Tracking computer is using 2 additional high-speed Ethernet connections to sample and control the pair of Stereo Dalsa Cameras.

The Face Recognition computer uses a high speed Ethernet connection to connect to the Face Recognition Dalsa Camera and a USB port for the Kinect Camera.

The high-level communication architecture is depicted in Figure 5.

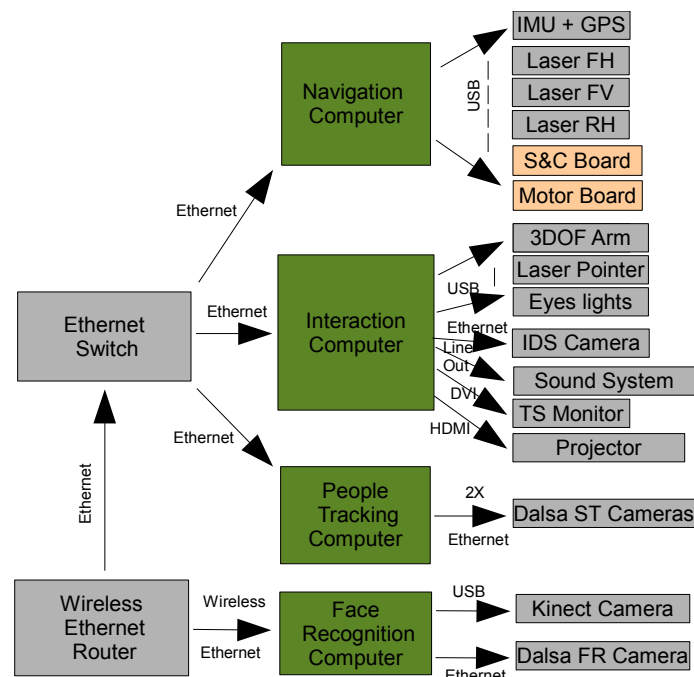


Figure 5: High-level Communication Architecture

4. FROG Robot Electronics

Several electronic boards have been designed and added to the FROG hardware. The following subchapters show and describe the boards created for the project.

The following electronic systems have been included on the FROG robot:

- **Motor Controller Board** - this board makes the movement of the robot possible. By the use of one microcontroller that communicates with the Navigation Computer and sends commands to two PI controllers that control two high power H-Bridges, it is possible to control the high power motors installed on the platform.
- **Sensor&Charger Board** - This board is divided into 3 different boards:
 - **Sensor&Management Board** - that communicates with the Navigation Computer and controls all the other boards that compose the Sensor&Charger Board;
 - **Power Switch Board** - that allows the robot to connect, disconnect and charge the auxiliary batteries;
 - **Charger Board** - is used to charge the batteries.
- **Sonar Board** - allows the use of sonars in the robot. By sending sonic pulses and by capturing the obstacles' echoes it is possible to detect and measure the distance between the robot and the near obstacles.
- **Interaction Board** - allows the Interaction Computer to control the 3DOF antenna with the laser pointer, enable or disable the laser pointer and finally control the LED lights around the robot eyes;
- **Wireless Joystick Controller** - The wireless joystick controller was an add-on introduced to be able to move the robot from one place to another without having to send direct commands to the Navigation Computer. It is also one important safety measure that allows the robot to be moved or stopped in case of danger.
- **Charger Docking station** - This is an external device that facilitates the autonomous docking of the robot and the charge of the batteries without human intervention.

4.1. Motor Controller Board

The Motor Controller Board manages the robot locomotion. The Master Motor Controller uses a **PIC18F6527** microcontroller to control and manage all the communication between the high-level robot Navigation Computer and the Slave PI Motor Controllers. The Slave PI Motor Controller uses a **PIC18F2431** microcontroller to provide all of the necessary control signals to the motor drivers. The overall architecture of the Motor Controller Board is depicted in Figure 6.

The Master Motor Controller will:

- Run low-level control loops to check for critical changes in the motor system that can affect the robot operation.
- Provide a I2C bus Observer that checks the information received from the I2C Slave devices to understand faults in the communication or on the devices.
- Check the good function of each Slave PI Motor Controller by changing status information between them.
- Control the motor and drivers' temperature and control the power of fan devices to cool them.

- Monitors the electronics, motors and the drivers' power supply.
- Measure the motors' current.

The Slaves PI Motor Controller will:

- Receive velocity references from the Master Motor Controller;
- Periodically read the encoder pulses;
- Send the read encoder pulses to the Master Motor Controller;
- Calculate the error between the reference velocity and the read velocity;
- Implement a PI controller to calculate the motor actuation;
- Implement an anti-windup error limit;
- Implement a reference velocity acceleration/deceleration profile;
- Remove the dead zone of the motor;
- Limit the maximum velocity of the motors.

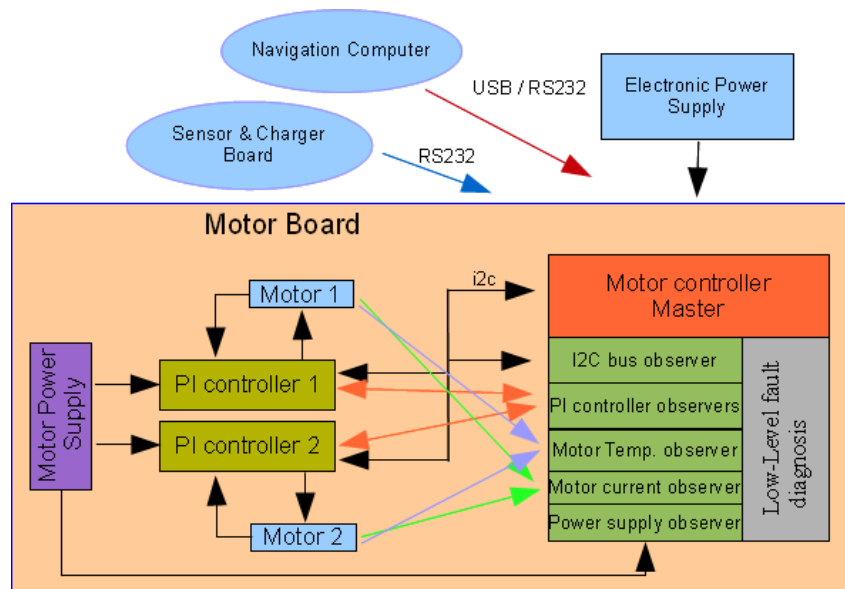


Figure 6: Motor Controller Board Architecture

The Master Motor Controller receives velocity commands from the computer and returns the encoder pulses. The Controller connects to two PI microcontrollers that generate the control actuations to follow velocity references. Each microcontroller connects to the motor using a 1000W H-bridge, and provides the pulses measured by the encoder.

Each microcontroller is optically isolated from the motor driver using a high-speed optocoupler for the control actuation signals and an optical amplifier for the current measurements. It is also optically isolated from the computer communication port, again using a high-speed bidirectional optocoupler. Several low-level fault diagnostics have been implemented to detect problems in the normal working of each component or lack of communication.

Figure 7 shows the assembled Motor Controller Board. The board has several information LED lights allowing a visual check of the state of each component previously described.



Figure 7: developed Motor Controller Board

The Motor Board communicates with the Sensor&Management Board exchanging information about the system status, diagnostics and environment condition, allowing low-level robot reactions to changes that can affect the robot's operation.

It is able to communicate with the computer using a USB-to-RS232 Converter and with the Sensor&Management Board through a RS232 bidirectional communication.

4.2. Sensor and Charger Board

The Sensor&Charger Board is responsible for the power management and also the sensor acquisition. It receives orders from the onboard robot Navigation Computer and returns information about the batteries, sensors and actuators. The Sensor&Charger Board Architecture is depicted in Figure 8.

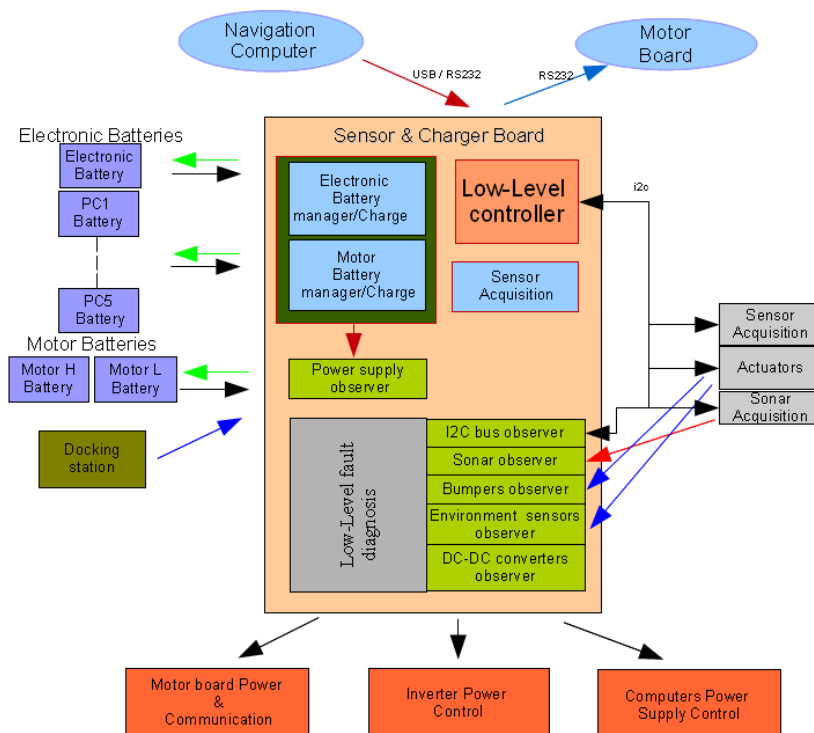


Figure 8: Sensor&Charger Board Architecture

Due to its complexity, it was decided to separate the Sensor&Charger Board into three individual boards: Sensor&Management, Power Switch and Charger Boards.

The Sensor&Management Board controller will be responsible for all the communications, control signals and sensor acquisitions. The Power Switch Board is controlled by the Sensor&Management Board controller allowing 5 auxiliary batteries (that can be used to power robot computers or devices) to be activated, deactivated or put on charge. The Charger Board will be responsible for the charge of each individual battery.

4.2.1. Sensor&Management Board

The Sensor&Management Controller, depicted in Figure 9, uses a **PIC18F6527** microcontroller to control and manage all the communication between the high-level robot Navigation Computer and to control all the actuators and sensors devices connected to the Sensor&Management Board.

The Sensor&Management Board is responsible for managing all of the power system by:

- Measuring the energy level in each battery;
- Connecting/disconnecting the power of the devices;
- Managing the connection to the Charge Docking station;
- Controlling the charge of each battery.

It is also responsible for connecting a set of sensors and actuators that will be used in the project. Several low-level fault diagnostics have been implemented to detect problems in the normal work of each component and communication. The Sensor&Management Controller will analyse the information gathered from the sensors and will run low-level control loops that will check for critical changes in the environment or system that can affect the robot operation.

The Controller has one dedicated channel to communicate with the Motor Controller, allowing it to send direct (and fast) commands to the motors and also to get information from them.



Figure 9: developed Sensor&Management Board

4.2.2. Power Switch Board

The Power Switch Board, depicted in Figure 10, has the necessary hardware to activate, deactivate or put in charge mode up to 5 batteries that can be used to power different components of the robot. It is possible to connect several batteries in parallel to power equipment that demands more energy. It has also the ability to charge each battery independently.



Figure 10: developed Power Switch Board

This board will be controlled by the Sensor&Management Board which will provide all of the necessary signals to activate, deactivate or charge each of the batteries. The Power Switch Board also has the necessary electronics to provide the measure of voltage of each of the batteries. All the control signals and voltage measurement circuits are completely decoupled by the use of optocouplers and opto-amplifiers.

Finally this board connects to the Charger Board that will allow the connection and charge of up to 8 batteries.

4.2.3. Charger Board

The Charger Board is used to charge up to 8 batteries independently. Each charging circuit uses a 3-phase lead acid battery charger already developed, tested and used in IDM's Projects.

The charger uses the following phases:

1. **Bulk Phase.** In this charging phase the charger will supply a constant 2A current to the battery that is charging. This will allow the battery to charge to about 70% of its capability.
2. **Absorption Phase.** After leaving the Bulk Phase the charger will enter the Absorption Phase where the charger will supply a constant voltage to the battery until it reaches about 100 to 120% of its capability.
3. **Maintaining Phase.** When the battery is fully charged the charger will leave the absorption phase and enter the Maintenance Phase where a 13.4V supply is supplied to the battery until the battery is disconnected from the charger.

The charging process for the 12V 17-20Ah batteries used in the FROG robot is the following. If the battery is dried (9.5V - 0%) the battery will start to charge in Bulk Phase for about 7 hours at 2A. At the end of the 7 hours the level of the battery is about 70%, that means that the charge level will raise about 10% each hour. After +/- 7 hours the charger will enter into the Absorption Phase where there will be an increase of about 5% of level of battery each hour. After 5 to 6 hours the battery level will reach 100%-120% (depending on the type of battery, number of cycles and correct maintenance). After this period the system will enter into the Maintaining Phase, and the battery will maintain the charge level.

Figure 11 shows the battery charge characteristics using the developed charger. Only the Bulk and Absorption Phases are displayed. The Maintaining Phase start can be set to any point of the Absorption Phase by changing the value of one resistor component.

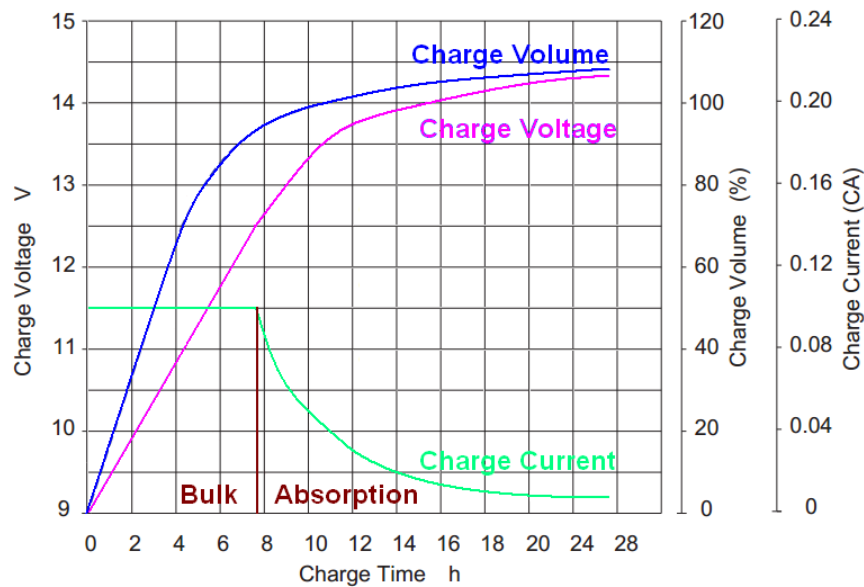


Figure 11: battery charge characteristics

The Charger Board is currently being used to charge the robot batteries. Figure 12 shows the FROG Battery Charger board.

The Charger Board was designed to charge Lead Acid batteries, but at this moment there is the possibility of changing the Lead Acid to LiFePO₄ batteries with PCM protection circuits. These batteries have the same size and capabilities as the ones that we are using at this moment, but they weigh $\frac{1}{3}$ of the weight of the Lead Acid batteries. The LiFePO₄ battery manufacturer states that the users can swap the lead batteries with the new ones without having to change anything, and also will be able to charge the batteries with the normal battery chargers. At this moment the charger circuit is being tested for charging these new batteries.

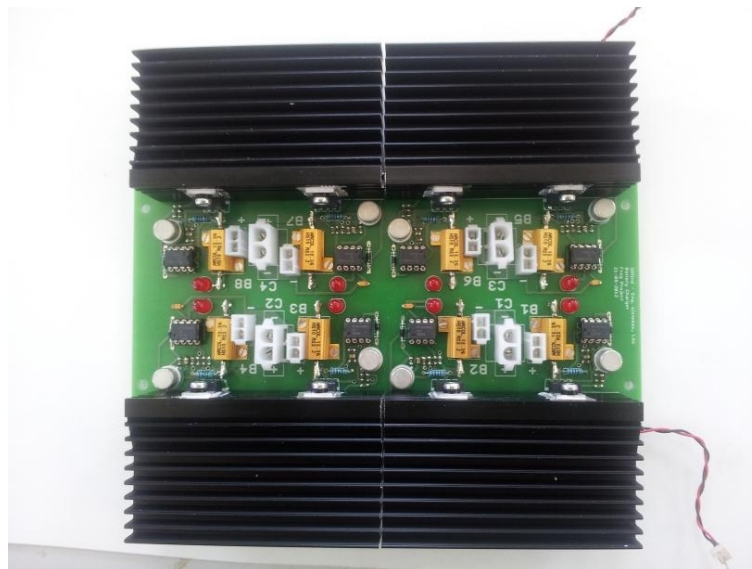


Figure 12: FROG Battery Charger Board

4.3. Sonar Board

The Sonar Board, depicted in Figure 13, will be used to control the firing of up to 10 sonars. This board is connected to the Sensor&Management Board through an I2C connection. The information provided from this board can be used to create low-level behaviours, for example reducing the velocity of the robot in the presence of an obstacle.

The system uses the SRM400 sonar transducers that are able to detect an obstacle at a distance of up to 2 m.

The board design allows the users to configure the sonar sampling and also to connect or disconnect the sonar readings or power.

Some of the modes that can be used allow the individual fire and reading of all or some of the sonars or the fire of all the sonars at the same time and the individual reading making a radar-like sensor.

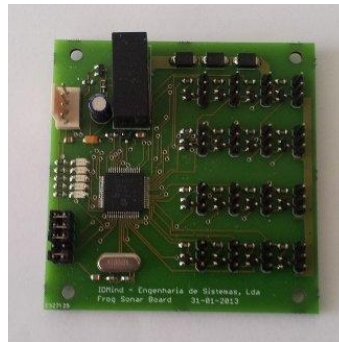


Figure 13: Sonar Processor Board

The Frog robot uses a ring of 10 sonars. They are disposed as depicted in Figure 14.

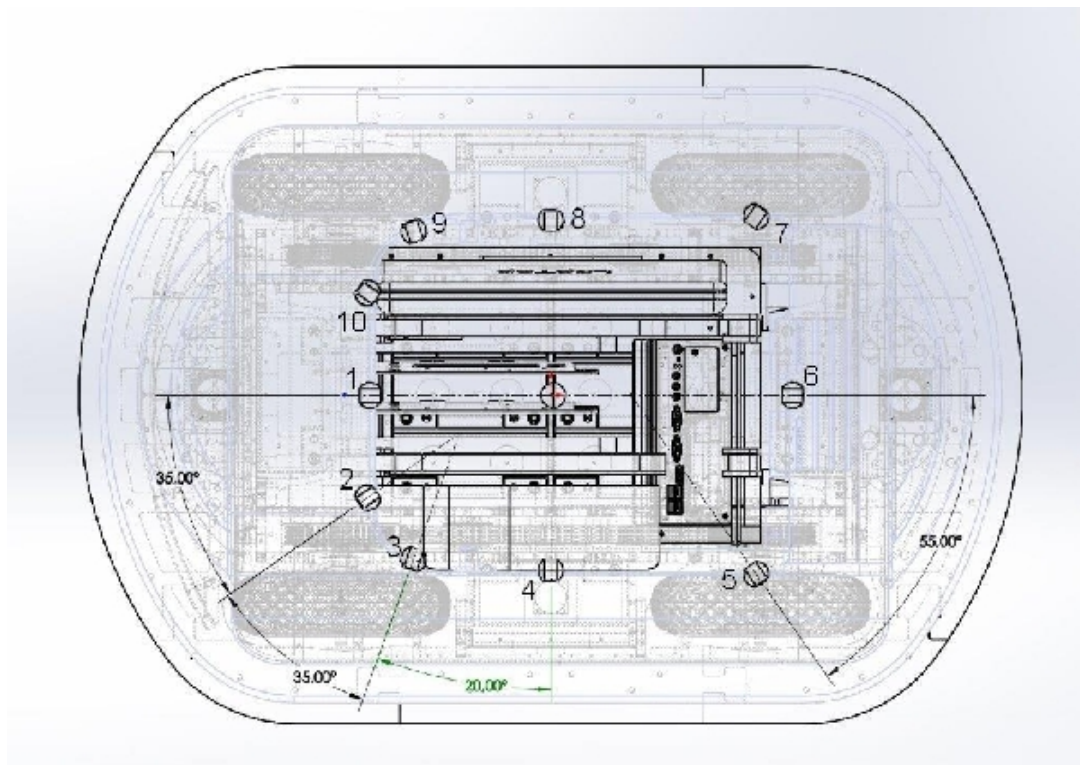


Figure 14: Sonar Processor Board

Three groups of sonars were created:

- Group A – includes sonars 1, 4, 6 and 8
- Group B – includes sonars 2, 5, 7 and 10
- Group C – includes sonars 3 and 9

The sonars in the same group are fired at the same time and the distance from the obstacles to each of the sonars is measured. To reduce the crosstalk effect, all the sonars in each group are disposed in, at least, 90° multiples.

Each group takes about 100ms to fire and process the echo. The processing of a new group is only started after the end of the processing of the last group (A->B->C->A->B->...).

4.4. Charger Docking Station

One important capability of the robot is the possibility to work without human intervention. To achieve this point the robot must be able to manage the onboard power and autonomously charge itself. One charger docking station is being developed and to be installed in a service area, where the robot can enter and plug itself in. This docking station will provide the necessary power that the onboard battery charger needs to charge the batteries. The Docking Station is a passive power station: control of the charging process is managed by the onboard Sensor&Charger Board. The Docking Station will be equipped with a 480W switch mode power supply able to power all the onboard equipment and at the same time provide about 2,5A of current for each of the 8 battery chargers.

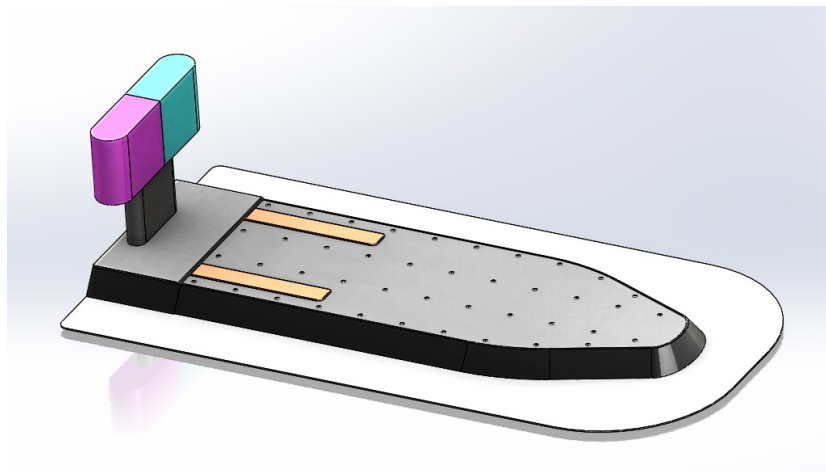


Figure 15: FROG Docking Station

The mechanics to guide the robot to dock and the Docking Station are being produced. Figure 15 shows the current design of the Docking Station. From the image it is possible to see the Docking Station's three main features:

1. **The Docking Station guiding shape.** The V-U design is about 5 cm high and will mechanically guide the robot to the power bar. This design has been tested with the real robot allowing the perfect end alignment, even when the robot enters the docking station with a high rotation error.
2. **The two guiding colour marks.** This will allow the robot to adjust its position to the Docking Station, the robot system will be able to measure the distance to the Docking Station, by calculating the relation between the real size of the colour marks and the size of the image

and also will be able to adjust it's position by trying to centre the common edge of the two colours in the centre of the image.

3. **The long and wide power bars.** They will provide enough surface area to provide all the power that the robot needs to charge the batteries and power all the components.

The following figures depict the robot in Docking Station. Figure 16 shows the robot aligned and parked in the station. Figure 17a) shows the two contact springs of the FROG robot connecting to the power bars of the docking station. One of the bars is the negative pole and the other is the positive pole of the 18V charger power supply. Figure 17b) shows a zoom image of the contact between the power springs of the robot and the power bar of the Docking Station.

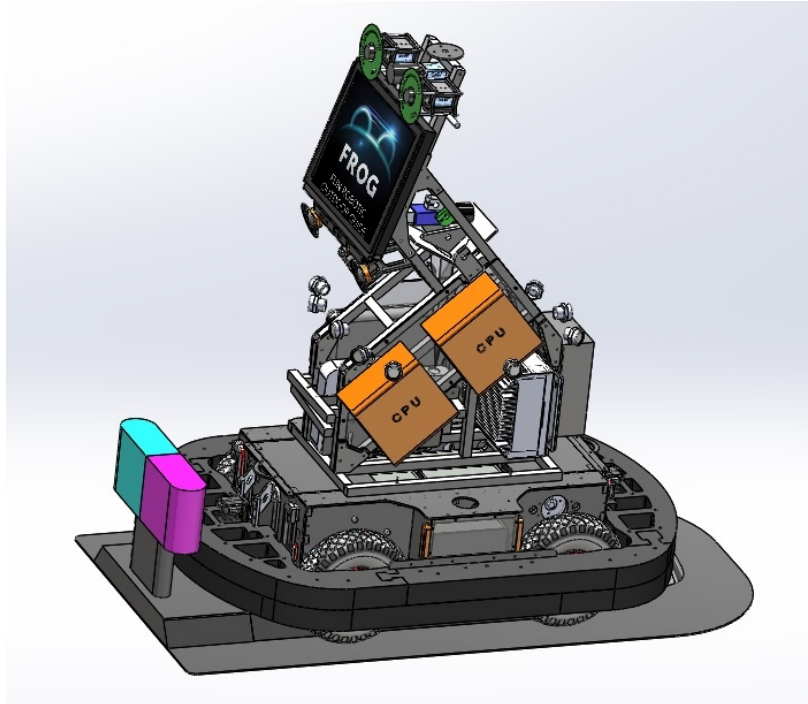


Figure 16: FROG Robot in the Docking Station

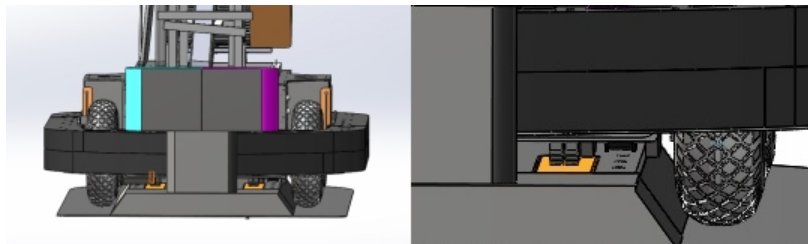


Figure 17: a) robot connecting to Docking Station; b) detail of the robot contact spring

4.5. Joystick Wireless Controller

The Joystick Wireless Controller is composed of two different circuit boards (depicted in Figure 18):

- **Wireless Joystick Emitter Controller Board** - This board is controlled by a PIC18F2520 that reads the information from an Analog Joystick and converts it to command references that are sent to the FROG robot platform. An Enable push button and an Emergency button are also connected to the board, allowing the user to take control of the platform or send a

Stop Emergency command to the platform. To transmit the data, a Wireless 19200 bits of baud connection is used. For that purpose an ERA900TS wireless emitter has been installed. The wireless Joystick system is powered by 4xAAA rechargeable batteries, and an external plug allows the system to be recharged, without having the need to open the controller and remove the batteries.

- **Wireless Receiver Controller Board** - This board is controlled by a PIC18F2431 that receives the wireless data transmission from the **Wireless Joystick Emitter Controller**, processes the received data and sends control commands to the **Motor Board**. To receive the data the system uses an ERA900TRS transducer. The board uses 2 different ways to send and alert the **Motor Board** of the existence of Joystick commands. Two digital ports indicate if the Joystick is taking control or if the Emergency button has been pressed. If the Joystick is taking control then the Motor board will use the I2C port to get information about the values to send as motor references.

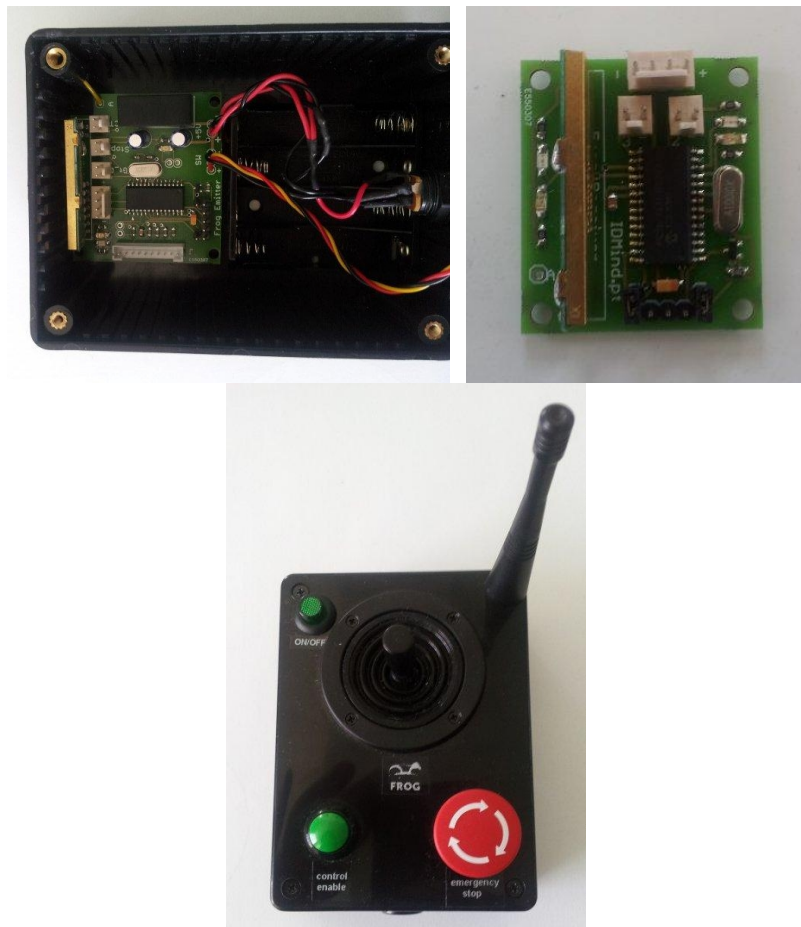


Figure 19: Wireless Joystick Emitter command

One Joystick controller box has been created to control the platform. On that controller box, depicted in Figure 19, there is an analog auto centered joystick that will be used to give the velocity and rotation commands to the platform, one ON/OFF press button to turn on and off the controller box, one Enable press button that will indicate that the Joystick will take control of the robot platform and finally one Emergency button, that when pressed will stop the platform.

On the upper right corner is possible to see the antenna that will be used to send the data.

4.6. Interaction Board

The Interaction Board allows the Interaction Computer to control the arm with the laser pointer, enable or disable the laser pointer and finally to control the LED lights around the robot eyes.

Figure 20 shows how the Interaction Board is connected to the Interaction Computer and the different communication and power types that are implemented on the board.

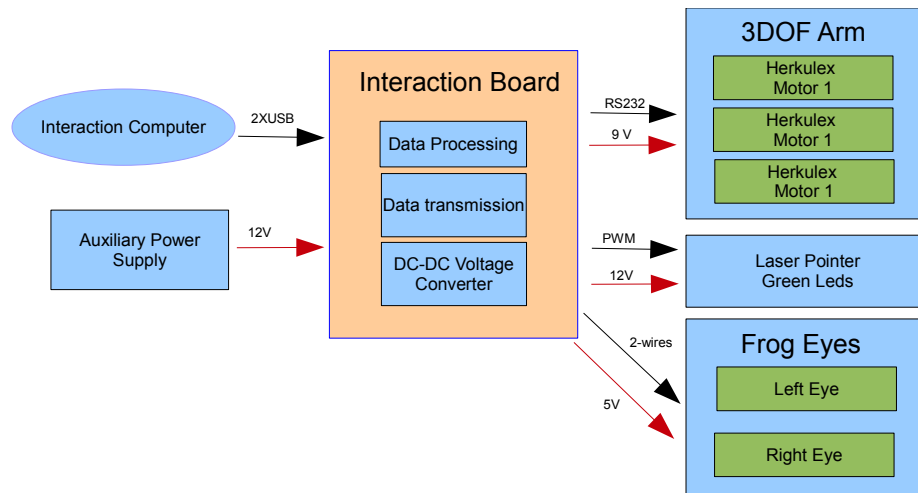


Figure 20: Interaction Board Communication and Power connections

The Interaction Computer will communicate with the Interaction Board using two distinct USB ports. One port will communicate directly through an USB to RS232 converter with the three Herkulex motors that control the 3DOF arm. The other USB port will connect to the PIC18F2520 microcontroller and will be used to send commands to control the green eyes of the FROG Robot, to enable and disable the Laser pointer and finally to control the intensity of the green LED lights that surround the laser pointer. Figure 21 depicts the Interaction Board layout.

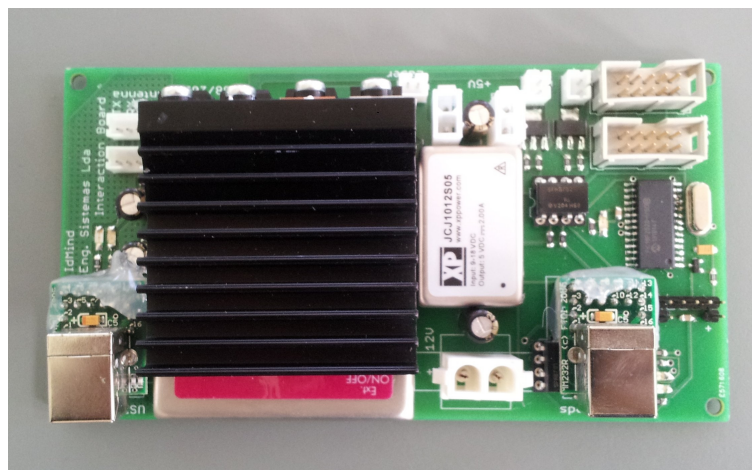


Figure 21: Interaction Board layout

4.6.1. 3DOF Arm with Laser Pointer

The 3DOF Arm is able to move by the use of three Herculex motors that are placed in each joint of the arm. The Herculex motors⁴, depicted in Figure 22, are state-of-the-art motors that incorporate motor, gearbox, control circuitry and communications capability in one single package.



Figure 22: Herculex Motors

The motors are connected in a daisy chain way and each motor has its own address. The Interaction Board connects to the base motor by the use of a four-wire cable, then this motor connects to the second and the second connects to the third motor.

To control each motor the controller has to send the motor address, the position command and the time that the motor has to perform that operation.

In terms of control algorithm, the motor uses a Proportional-Integral-Derivative (PID) controller with a trapezoidal velocity profile.

It is able to give the feedback of the position, speed, temperature, load and other internal values.

The final version of the arm is depicted in Figure 23. The arm is composed by 3 Herculex servos. One in the base that allows a rotation of 300 degrees and two others in the upper joints.



Figure 23: 3DOF arm with laser pointer

The end effector of the arm houses a green laser pointer and green LED lights that mirror those around the eyes.

4 <http://www.robotshop.com/content/PDF/manual-drs-0101.pdf>

The laser pointer can be used to point towards specific points of interest and the green LEDs are used to draw people's attention.

4.6.2. FROG Eyes

One of the ways that FROG has to interact with people is by the use of the eyes. They will have an important role in drawing people's attention to look in a certain direction or exposing the robot "mood".

Each eye will be mounted around one of the stereo Dalsa cameras. The lens of the camera will be placed in the middle and the lights will show where the robot is looking.

Each eye has 96 independent green LEDs, divided over 3 rows of 32 LEDs. The internal circle has a diameter of 3.6 cm to allow the camera lens to be mounted and also the lens protection to be fitted. The external circle has a diameter of 12 cm. The right image of Figure 24 shows the final placement where the LEDs will be soldered.

Each eye is composed of two boards: the board with the LEDs and a second board with the LED controller. They are depicted in Figures 24 and 25.

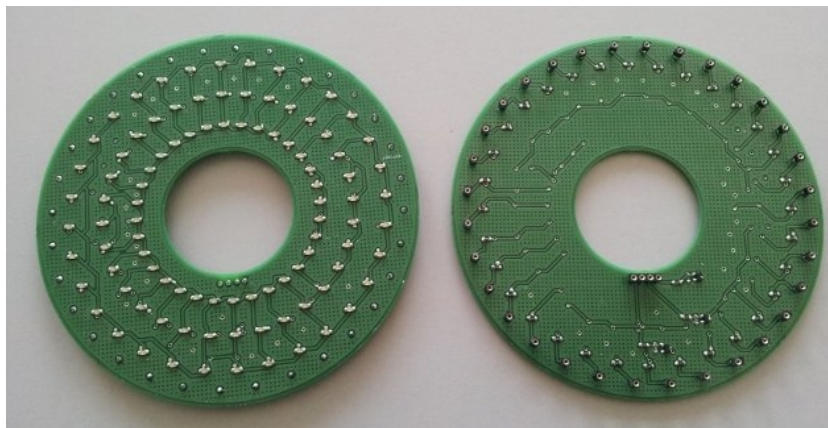


Figure 24: LED eye board (front and rear face)

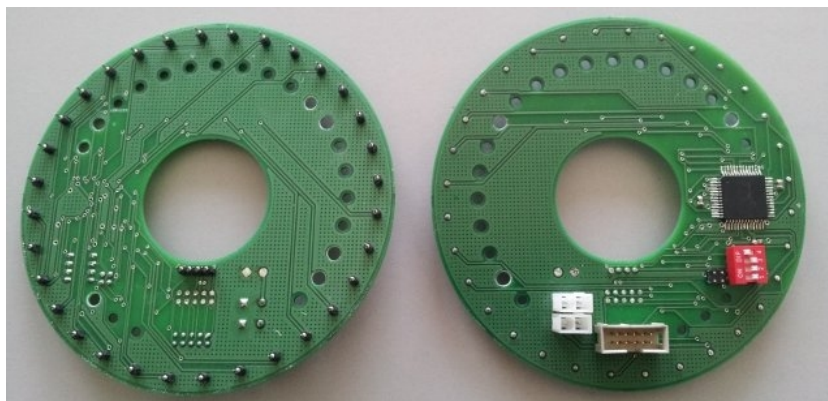


Figure 25: LED eye's controller (front and rear face)

The board with the LED controller will have all the power connections and communication ports to connect to the interaction board. This board has the same size as the board with the LED eyes and will connect to it using sockets placed around the exterior of the board.

To fix the eyes to the structure with the Dalsa camera a set of holds have been introduced in the LED control board.

5. Low-level Control Loops

The Sensor&Management Board and the Motor Board control the low-level hardware. Both run low-level control loops and interrupt driven events that control the system and also check for faults in the normal operation of the platform.

In the following subchapters the low-level control loops and interrupt driven events for both boards are displayed and explained.

5.1. Sensor&Management Low-level Control Loops and Interrupt Driven Events

The **Sensor&Management Board** runs the control loop displayed in Figure 26. To ensure the correct processing of the loop, a watchdog timer is running and if the system blocks in some part, the system will force a reset of the board.

The **Sensor&Management Board** uses two different methods to control the robot:

1. **Low-level Control Loop** - The loop will check for USART communication problems, communication losses, check for collision, obstacle detections, manage the power supplies and finally sample the environment sensors.
2. **Interrupt Driven Events** - Whenever the Navigation Computer sends a command to the Sensor&Management Board an interrupt event is generated. This will allow priority to be given to the commands from the Navigation Computer. When an interrupt event is activated the system processing jumps from the low-level control loop to the interrupt routines. When it finishes processing the interrupt the system processing will be restored to the low-level control loop.

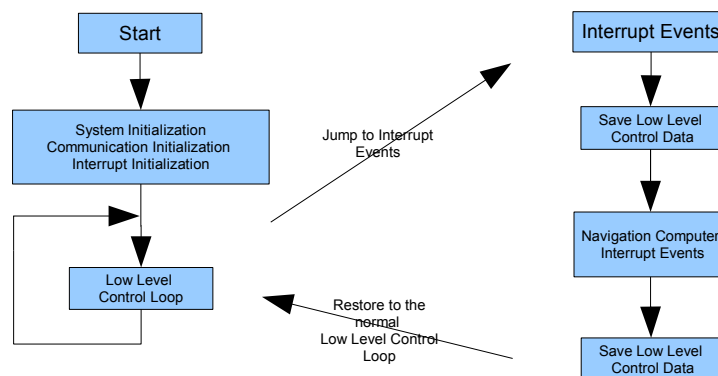


Figure 26: Sensor&Management Control Loop and Event Management

The following sub-section will explain the way that each of the methods is implemented.

5.1.1 Sensor&Management Low-Level Control Loop

The Sensor&Management Board runs the control loop displayed in Figure 27. To ensure the correct processing of the loop, a watchdog timer is running and if the system blocks in some part the system will force a reset of the board.

The cycle starts by checking:

- **USART Communication** - the system will check for problems in the configuration between the Navigation computer and the board. It will also detect if errors in the data transmission have occurred. When a problem is detected the system will:
 - Update the **Error Status**;
 - Send the **Error Status** to the Motor Board;
 - Re-initialize the USART communication ports.

Then the system will verify whether the USART problem has been solved. If not, it will try to correct the problem until it has been corrected or the watchdog reset will be activated. In this case the board will make a complete reset.

If the problem is solved the system will:

- Clear the **Error Status**;
 - Send the **Error Status** to the Motor Board.
- **Collision Detection** - The system will detect if the bumper switches have been activated. In case of collision detection the system will:
 - Update the **Collision Status**;
 - Send the **Collision Status** to the Motor Board.

If no collision is detected the system will:

- Clear the **Collision Status**;
 - Send the **Collision Status** to the Motor Board (only after a collision is detected).
- **Obstacle Detection** - The system will get the sonar information from the Sonar Board, and check for possible obstacles. If an obstacle is detected inside the safety area the system will:
 - Update the **Obstacle Status** and the obstacle's distance;
 - Send the **Obstacle Status** and the obstacle's distance to the Motor Board.

if no Obstacle is detected the system will:

- Clear the **Obstacle Status**;
 - Send the **Obstacle Status** to the Motor Board (only after an obstacle is detected).
- **Motor Board Status** - The System will get status information from the Motor Board and verify if problems have occurred. The **Motor Board Status** will give the information about the communication between the two PI controllers, communication errors, Motor, Drivers and Electronic power supply problems. If a problem is detected the system will:
 - Force the Motor Board Reset;
 - Update **Motor Board Error Status**;
 - Disable the Power from the Motor Drivers.

If no error is detected the system will:

- Update **Motor Board Error Status**.
- **Battery and Charger Management** - The system will measure all the batteries' voltages, cable and manage the power connection and charges. The **Battery and Charger Management** will be explained later.
 - **Sample Environment Sensors** - Check all the environment sensors.

The I2C communication between the device that is connected to the Sensor&Management is always checked by the use of checksums inside the sent and received data packages. The I2C control

registers of the microcontrollers also provide error status information that is used to correct any abnormal situation.

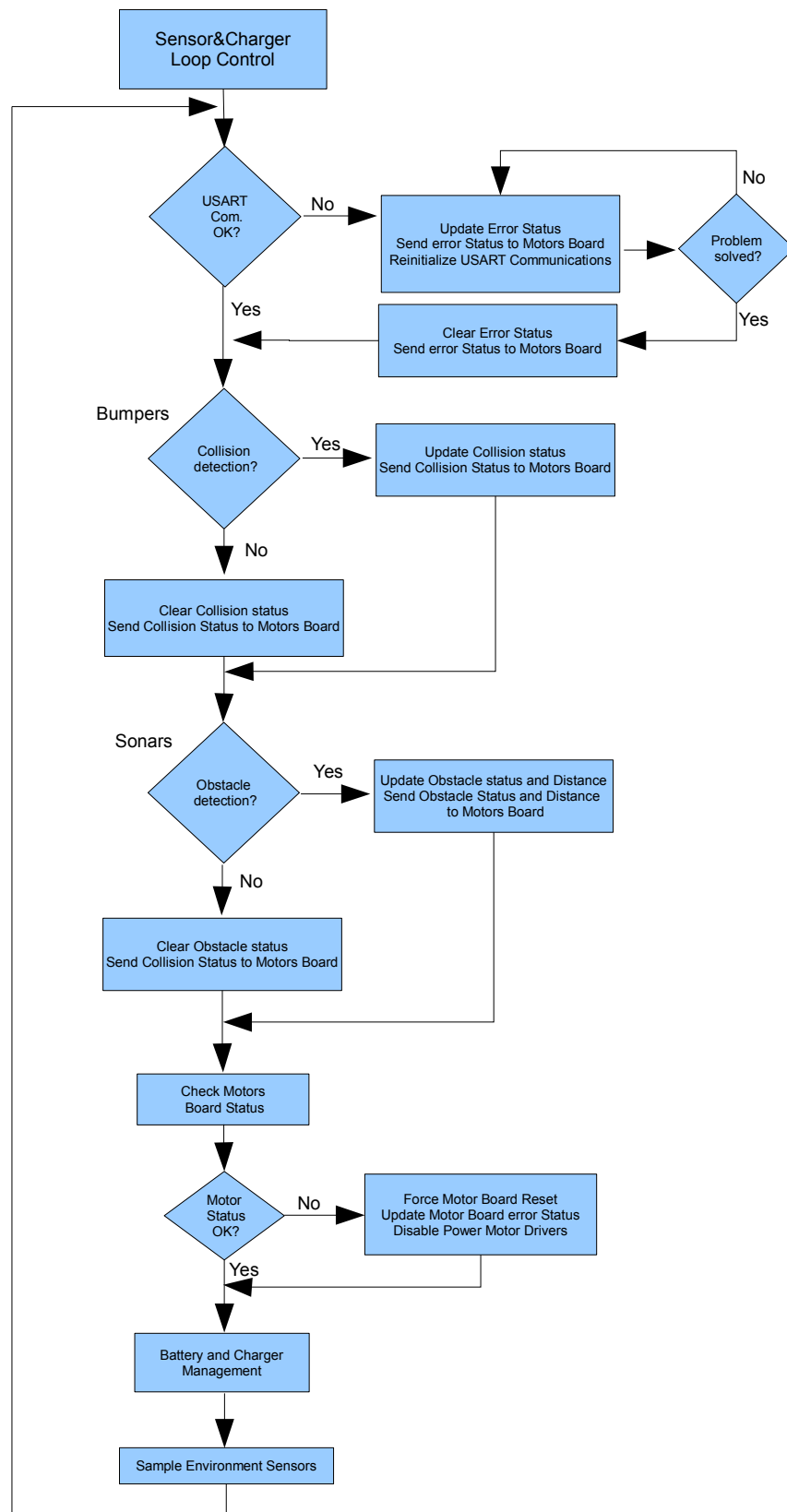


Figure 27: Sensor&Management Board Control Loop

5.1.1.1. Battery&Charger Management Algorithm

The Battery&Charger Management algorithm will deal with voltage measurement of the different batteries and also the charging cable and perform the management of the charging of the batteries. Figure 28 depicts the Battery&Charger Management algorithm.

The algorithm starts by performing an analog to digital reading of the voltage of all the batteries and charging cable. Then it will check whether the robot is on the docking station by checking if the charging cable has a value higher than 16V.

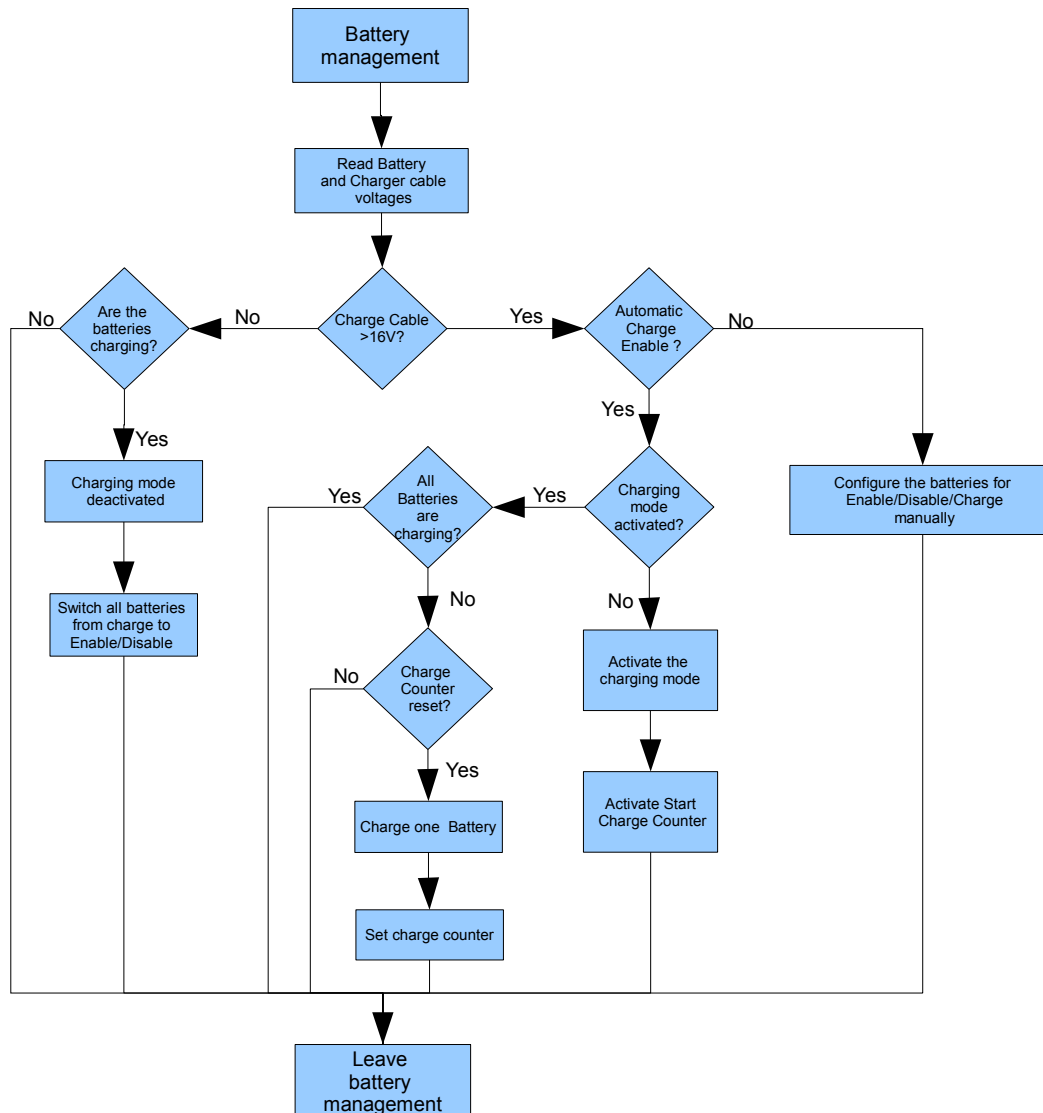


Figure 28: Battery&Charger Management Algorithm

If the docking station is detected then the algorithm will check if the charging is on manual or automatic charging mode.

Manual Mode - In this mode the navigation computer will fully control the Enable/Disable and charge of each battery. This will allow the robot to reach the docking station and only charge the batteries that are needed. It also prepares the robot to leave the docking station, by enabling all the electronic and motor batteries.

Automatic Mode - In this mode, after detecting that the robot has reached the docking station, the system will start to charge the batteries. To reduce the spikes, a charger counter is used. Whenever it reaches zero one battery is put on charge. It will take 8 charge counter cycles to put all the batteries on charge. The automatic mode does not allow the motor batteries to switch back. To leave the docking station the Navigation Computer must send a command to the Sensor&Management Board to enter into the Manual Mode and enable the Motor Batteries.

If the robot was connected to the docking station and for some reason it is disconnected from it, the system will automatically change the battery connections from charging to Enable/Disable.

5.1.2. Sensor&Management Interrupt Driven Events

When the Navigation Computer sends set or get commands to the Sensor&Management Board, a serial communication interrupt event is initiated/triggered. The Sensor&Management Board processing jumps from the **Low-Level Control Loop** to the **Navigation Computer Interrupt Events**. All the data that was being used in the Low-Level Control Loop is saved and will be restored when the processing of the interrupt event finished.

The **Navigation Computer** is able to force the following events:

- **Set Events**
 - Set Enable/Disable sonars;
 - Set Enable/Disable/Charge main batteries;
 - Set Enable/Disable/Charge auxiliary batteries.
- **Get Events**
 - Get main batteries and docking station power and system status;
 - Get auxiliary batteries;
 - Get sonar readings;
 - Get Bumpers state;
 - Get Temperature/Humidity/Rain sensor readings;
 - Get communication status.

5.2. Motor Board Low-level Control Loops and Interrupt Driven Events

The **Motor Board** runs the control loop displayed in Figure 29. To ensure the correct processing of the loop, a watchdog timer is running and if the system blocks in some part, the system will force a reset of the board.

The **Motor Board** uses two different methods to control the robot:

1. **Low-level Control Loop** - The loop will check for USART communication problems or communication losses, manage the joystick actuations, check the motor board power supplies and motor and drivers' temperatures.
2. **Interrupt Driven Events** - there are 3 different events that can interrupt the microcontroller:
 - Navigation Computer Serial Commands;
 - Sensor&Management Serial Commands;
 - Timer event commands.

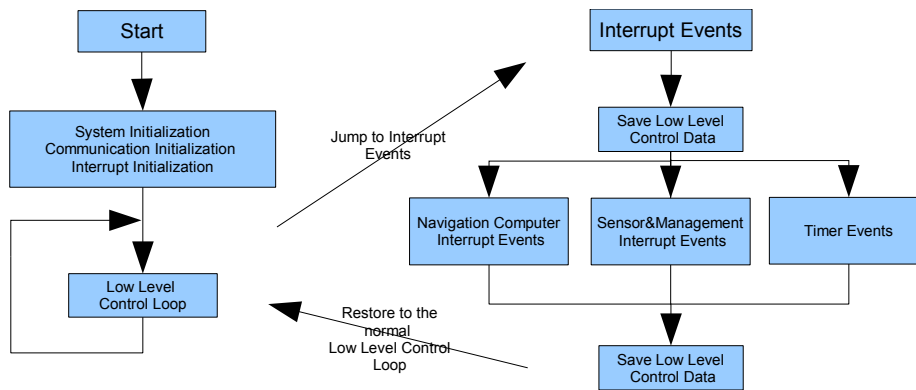


Figure 29: Motor Board Low-level Control Loop and Event Management

The following sub-section will explain the way that each of the methods is implemented.

5.2.1 Motor board Low-level Control Loop

The **Motor Board** runs the control loop displayed in Figure 30. To ensure the correct processing of the loop, a watchdog timer is running and if the system blocks in some part the system will force a reset of the board.

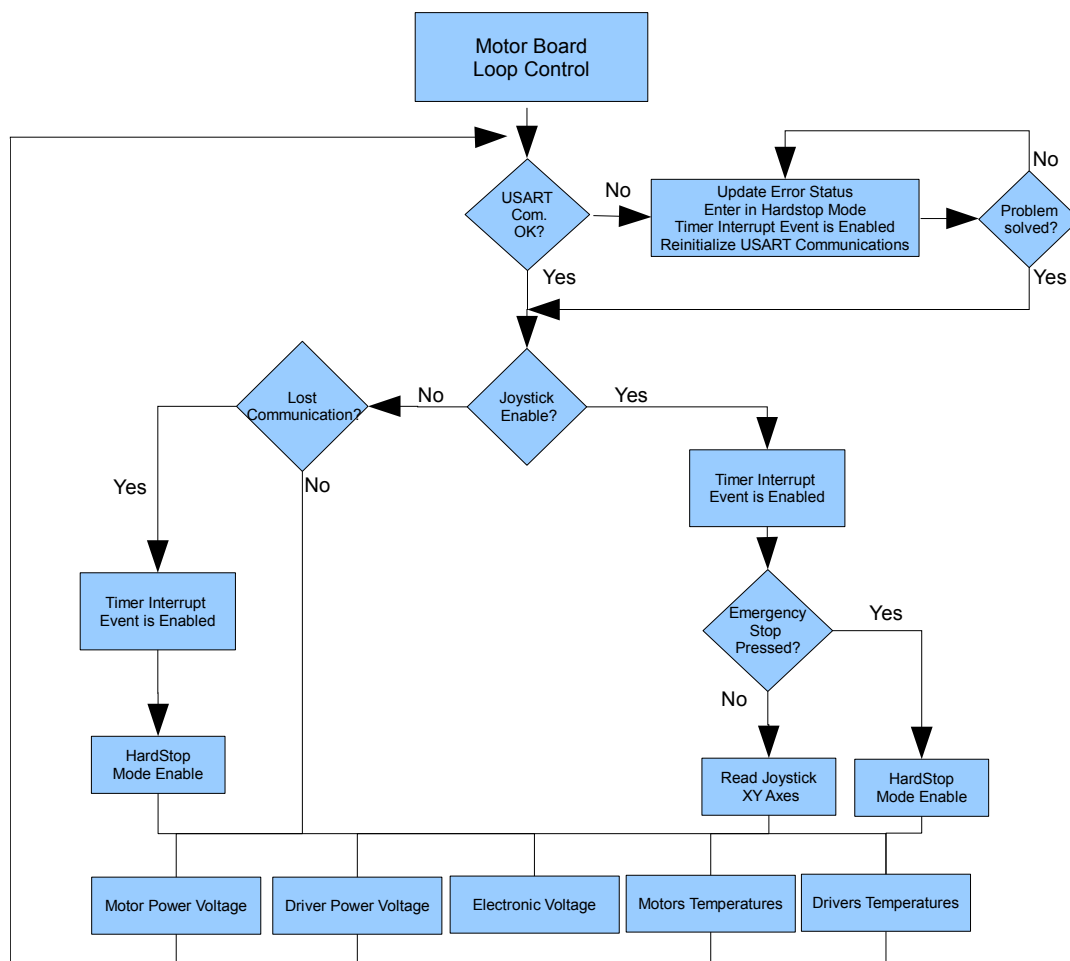


Figure 30: Motor Board Low-level Control Loop

- **USART Communication** - the system will check for problems in the configuration between the Navigation computer and the board. It will also detect whether errors in the data transmission have occurred. When a problem is detected the system will:
 - Update the **Error Status**;
 - Activate the **Hardstop Mode**;
 - Activate the **Timer Interrupt Event**;
 - Re-initialize the USART communication ports.

Then the system will verify if the USART problem has been solved. If not, it will try to correct the problem until it has been corrected or the watchdog reset will be activated. In this case the board will make a complete reset.

If the problem is solved the system will:

- Clear the **Error Status**;
- **Joystick Control** - the system will check whether the Joystick is enabled or not. There are 3 different possible states:
 - Joystick Control Enable OFF;
 - Joystick Control Enable ON;
 - Joystick Control Emergency Stop pressed.

If the **Emergency Stop** button is pressed the system will enter the **Hardstop Mode** and activate the Timer Interrupt event. The **Timer Interrupt Event** will send stop commands to the motor Controllers and all the velocity commands from the Navigation Computer and from the Joystick will be ignored.

If the **Joystick Control Enable** button is activated all the velocity commands from the Navigation Computer are ignored. The Timer Interrupt Event is Enabled and the platform starts to be controlled by the Wireless Joystick.

If the **Joystick Control Enable** button is disabled only the commands from the Navigation computer will be accepted and will control the motor Controllers. The **Timer Interrupt Event** will become disabled.

The **Low-level Control Loop** will only configure the method used to send commands to the motors, the actual actuation will be performed by **Interrupt Driven Events**.

- **Motor System Measures** - The Motor Board Loop will sample the following sensors:
 - Motor Power Voltage;
 - Driver Power Voltage;
 - Electronic Voltage;
 - Motor Temperatures;
 - Driver Temperatures.

Due to the fact that all the sensors displayed have a low change latency, we are able to read only one of them each time that the cycle is performed. This will increase the frequency of the Control Loop.

The results from these measures will be compared with the normal voltages for each power supply and normal temperatures for each of the devices. If one or more of the values are out of the normal scale then the corresponding error bit will be activated on the **Error Status**.

The temperature values will also activate the cooling fans on the robot.

The I2C communication between all the devices that are connected to the Motor Board is always checked by the use of checksums inside the sent and received data packages. The I2C control registers of the microcontrollers also provide error status information that is used to correct any abnormal situation.

5.2.2. Motor Board Interrupt Driven Events

As was mentioned before there are 3 different events that can interrupt the microcontroller. They are described below.

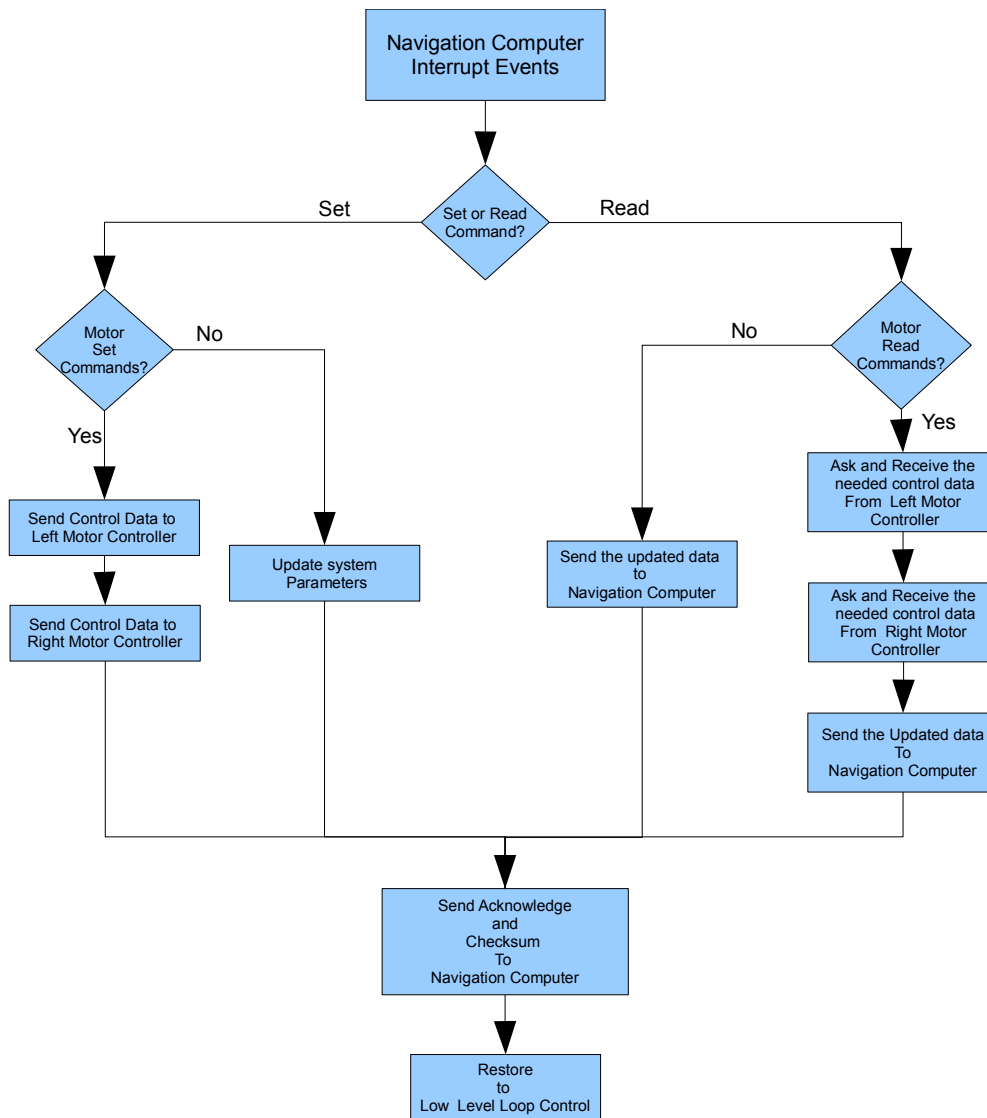


Figure 31: how the system processes Navigation Computer Interrupt Events

1. **Navigation Computer Interrupt Events** - When the Navigation Computer sends set or get commands to the Motor Board (see Figure 31), a serial communication interrupt event is triggered/initiated. The Motor Board processing jumps from the **Low-level Control Loop** to the **Navigation Computer Interrupt Events**. All the data that was being used in the Low-level Control Loop is saved and will be restored when the processing of the interrupt event has been finished.

The **Navigation Computer** is able to force the following events:

- **Set Events**
 - Set Velocity;
 - Set cooling fans.
- **Get Events**
 - Get Encoders Ticks;
 - Get Motor Board Voltage;
 - Get temperature;
 - Get cooling fans;
 - Get communication status.

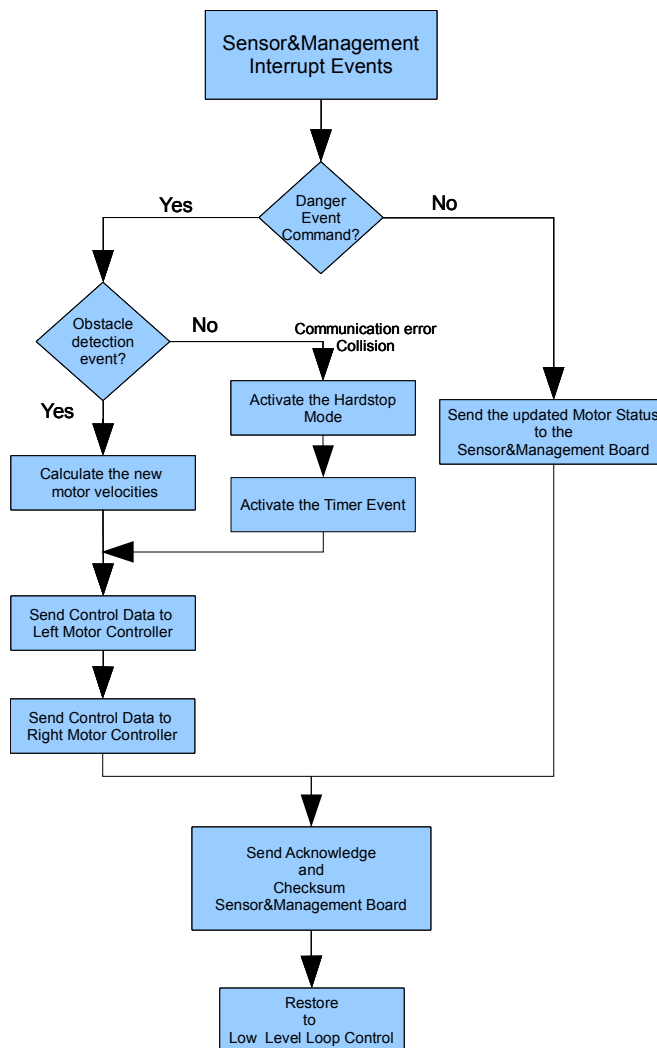


Figure 32: how the system processes Sensor&Management Interrupt Events

2. **Sensor&Management Interrupt Events** - When a communication error, collision or an obstacle is detected by the Sensor&Management Board, it sends an error message to the Motor Board. By doing that it will generate a Sensor&Management Interrupt event (see Figure 32). The **Sensor&Management Board** is able to send status information, but is also able to use this channel to get the Motor Board status.

The **Sensor&Management** is able to force the following events:

- **Set Events**
 - Set Collision Event;
 - Set Obstacle Detection Event;
 - Set Communication Error Event.
- **Get Events**
 - Get Motor Status.

3. Timer Interrupt Events - The Timer Interrupt Enabled can be enabled by the activation of the Hard-stop Mode (see Figure 33).

There are several ways to enter into the **Hard-stop Mode**:

- **Communication Lost** with the **Navigation Computer**;
- **Status Error** from the **Sensor&Management board**;
- **Collision** detection from the **Sensor&Management board**.

All these events will enable the **timer interrupt** that will periodically check if the Motors board can get out of the **Hard-stop Mode**. In this mode the system will ignore all the actuation commands. All the other functions will continue to run. If the system does not detect a Hard-stop event for more than 5 seconds it will leave the **Hard-stop Mode**.

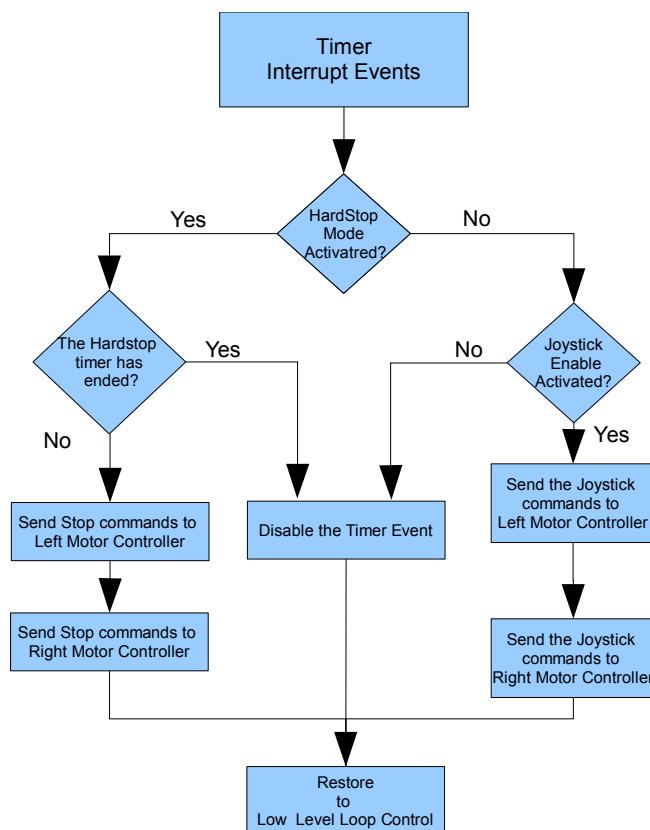


Figure 33: how the system processes Timer Interrupt Events

6. Safety

During the first year review meeting held in Amsterdam in December 2012, questions on Human-Robot safety interaction were raised. It was important to review the mechanics, electronics and the low-level software to be able to navigate safely in crowded places.

For that a Failure Mode and Effect Analysis (FMEA) has been performed and actions have already been introduced.

Subchapter 6.1. shows a preliminary FMEA study and subchapter 6.2. will present some of the actions that have been introduced to prevent or deal with cases of emergency.

This section will be updated with relevant information as the robot is being continuously evolving to its final state and the partners are developing their work.

6.1. Failure Mode and Effect Analysis (FMEA)

Component: Robot

Sub Component	Failure	Local Effect	End Effect	Detection Method	Corrective Action
Shell	Kids try to climb over the robot	Damage to the shell and kid injuries	Robot shell needs to be repaired	Visual	Redesign of shell to reduce the possibility of kids climbing over
Shell	Obstacle Detection	Robot hits an obstacles	Robot shell needs to be repaired	Visual	Add sensors or increase system performance
Bumper	Robot-Human collision	The robot does not stop in contact with a human	Human injuries	Collision Sensor	Introduction of collision sensors around the robot
Bumper	Robot-Human collision	The robot stops but it is not instantaneous	Human injuries	Collision Sensor	Introduction of soft materials in the construction of the bumper and increase the stop acceleration
Laser Range Finders	Human Eyes exposures	-----	Eye injuries	Visual	Use of eye safe Laser Range Finders. Class I Laser
Laser pointer	Human Eyes exposures	-----	Eye injuries	Visual	Use of Class II Laser pointers and never point to the people.
Projector	Human Eyes exposures	-----	Eye injuries	Visual	Use a low power projector and never use it to project towards people.
Platform	Is lost and can put people in danger	Robot is trying to find its way without any localization sensor	Robot shell needs to be repaired, human injuries, damage to the workspace	Visual	Visible Emergency stop button to cut motor power and to allow moving the robot to other places by hand
Platform	Should be placed in another location	Robot is moving and there is no possibility to move it to another location	Different problems	Visual	Introduction of a Wireless joystick to take control of the platform
Platform	Needs to be immediately	Robot is moving out of control	Robot shell needs to be	Visual	Introduction of a wireless emergency stop button

	stopped and there is no way of reaching it in time		repaired, human injuries, damage to the workspace		
Platform	The robot is moving at its full speed and is taking too long to stop	The robot is running at about 1.6 m/s and needs about 30 to 45cm to stop	Collision with obstacles, human injuries	Visual	Introduction of a velocity limit that allows the robot to safely stop in case of emergency

Component: Sensor&Management Board

Sub Component	Failure	Local Effect	End Effect	Detection Method	Corrective Action
Batteries	Does not detect one or more batteries	Some or all parts of the robot are not powered	The robot does not perform as expected	Voltage converters on the board	Check the battery voltage, fuses and connectors
Navigation Computer	Does not communicate	Lost communication	Robot stops	No Serial COM detection	Check/replace: USB cable, usb-to-serial converter and/or communication optocouplers
Navigation Computer	Loses communication	Intermittent communication	Abnormal operation	Visual and losses of communication with the navigation computer	Check/replace: USB cable, usb-to-serial converter and/or communication optocouplers
Bumpers	Obstacle detections/collisions	Collision with obstacles	Robot shell needs to be repaired, human injuries, damage to the workspace	Distance and collision sensors	Communication between the Sensor&Management board with the motors board to transmit obstacles/collision detection
Motors Board	Loss of communication with the motors board	Impossibility to communicate the presence of an obstacle	Robot shell needs to be repaired, human injuries, damage to the workspace	Communication errors	Cut the power from the motors

Component: Motor Board

Sub Component	Failure	Local Effect	End Effect	Detection Method	Corrective Action
Motor	Overheat	Reduction of motor performance	Robot needs to stop to cool down the motors	Temperature sensors	Introduction of heat sinks with fans to reduce the heat
Motor	Everything is Enabled but the motors do not rotate	The motor does not rotate	Robot is stopped	Visual	Check/replace optocouplers
Motor Drivers	Not powered	The motors are stopped	The robot does not move	Power not detected. Drivers power LED off	Check/Replace the motor, batteries and fuses
Motors Drivers	Fans not working	Overheat of the drivers, reduction of motor performance	Robot needs to stop to cool down the motor drivers	Temperature sensors	Replace fans and check the actuation pins

Navigation computer	Does not communicate	Motor Board sent a hardstop to the motor controllers	Robot is stopped	No Serial COM detection	Check/replace the: USB cable, USB-to-serial converter and communication optocouplers
Navigation computer	Loses communication with the Motor Board	The last drive command will continue to be executed	Robot will continue to run with the same actuation	Visual and loss of communication with the navigation computer	Check for communication losses
Temperature sensors	Do not work	Overheating of the component it is attached to	Burning of a component, reduction of the robot capabilities	Temperature detection error and visual	Replace the temperature sensor

6.2. Safety Measures

To deal with some of the safety issues identified, some measurements were introduced.

In terms of platform base **Mechanics** the following actions were performed:

- Complete redesign of the bumper system;
- Introduction of 12 switch sensors to detect collisions in any direction;
- The bumper was increased all around the FROG platform structure;
- Lateral detection was introduced;
- Introduction of soft materials;
- There was a redesign of the laser supports.

In terms of the **Robot Shell**:

- Redesign of the robot shell, avoiding lower horizontal flat surfaces that can be used to climb on to.

In terms of **Electronics** the following actions were performed:

- Redesign of the sonars board allowing for:
 - Activate/deactivate sonars;
 - Change of sonar fire method - It is possible to select the sonars that are going to be fired at any time, and it is possible to fire all the sonars at the same time to create a radar detection.
- Wireless console Joystick with stop button. Allows operators to stop or drive the robot using a wireless console. When enabled, this wireless control will override any motor control command given by the navigation computer;
- Direct communication link between the Sensor&Management Board and the Motor Board;
- Detection with 12 collision detection switches placed around the robot.

In terms of the **Motor PI controllers software** the following actions were performed:

- Velocity and acceleration control.
 - Velocity limitation of 1m/s;
 - Different acceleration/deceleration ramps for normal operation and for sonar obstacle detection/bumper detection. For normal operation the acceleration is 1m/s^2 and in

the presence of an obstacle the acceleration will be increased to 3.3m/s^2 . Allowing to Stop the robot, fully loaded on a horizontal surface, at a 1m/s velocity in just 15 cm.

- Implementation of a watchdog timer to verify the correct processing of the control loop.

In terms of the **low-level safety control loops and interrupt events software** the following actions were performed:

- Watchdog timers in all microcontrollers;;
- Error status messages exchanged between the Sensor&Management and Motor Board;
- All the boards must be in normal operation to Enable the motor drivers;
- The minimum obstacle distance is sent to the Motor Board, if necessary the Motor Board will reduce the maximum velocity to a safer velocity;
- If a bump is detected, a stop message is sent to the Motor Board to stop the motors immediately;
- Detection of communication errors between the boards and the Navigation Computer. If an error is detected the FROG robot will stop immediately;
- Wireless Joystick controller with Emergency stop button control override;

7. Working with the Robot

This chapter describes the quick steps that allow the user to move the FROG robot. There are two ways to drive the platform. The first one is by using the Wireless Joystick controller and the second is by using the navigation computer for that purpose.

The following subchapters explain the quick steps to set up the robot to a quick and to a normal operation.

7.1. Setting up the Hardware

To be able to control the FROG base platform, the user must select the components that it will use. The minimum configuration that can be selected is using the FROG Base Platform with the Wireless Joystick controller. In this configuration there is no need to power any computer and only three batteries are necessary to control the robot.

To work in the minimum configuration the user must perform the following steps:

1. Connect three charged batteries to the Motor L, Motor H and Electronic Batteries Connectors;
2. Press the Electronics power button;
3. Press the Motor power button;
4. Activate the motor drivers by releasing the emergency button.

At this moment the robot platform can be controlled by the wireless Joystick controller or by an external computer/laptop by connecting to the Motor and Sensor&Management Boards' USBs.

To work with the Navigation Computer the user must use a fourth battery, or reorganize the power to use the electronic battery to power the navigation computer. This can be done quickly by connecting the power cable connector of the PC directly to the non-regulated output of the Sensor&Management Board.

To work in the minimum configuration with the navigation computer, with 4 batteries, the user must perform the following steps:

1. Connect four charged batteries to the Motor L, Motor H, Electronic and PC1 Batteries Connectors;
2. Press the Electronics power button. By turning on the Electronic power button all the auxiliary batteries will be Enabled, providing power to all the components aboard. This means that the Navigation computer that is plugged in the PC1 power connector will be powered.
3. If the Navigation computer power button is OFF then the user must turn it ON. But if the switch is maintained in ON state the computer will boot normally.
4. Press the Motor power button;
5. Activate the motor drivers by releasing the Emergency button.

At this point the navigation computer is ready to control the system, the user can use the wireless network connection to communicate with the system.

7.2. Wireless Joystick Control

The FROG platform is equipped with a wireless joystick controller (see Figure 34) that can be used to control the robot. It can also be used in case of emergency. It incorporates an Emergency button that

when activated will send a Hardstop command to the robot. The robot will only be able to navigate after the release of the Emergency button. By locking the Emergency button the Wireless controller is sending a stop command to the wireless receiver installed in the FROG platform that connects directly to the Motor Board. The Wireless Receiver will pull High the Emergency Stop of the Motor Board. This will make the Motor Board ignore all the navigation commands, and send a Hardstop command to the motor controllers.

Another feature of this command is the capability to override the motor velocity commands sent by the Navigation Computer. When the Control Enable of the command is pressed, the wireless receiver pulls the Joystick Command Enable of the Motor Board to a high level. From this moment the Motor Board will only receive commands from the Wireless Joystick control. The user only has to keep the Control Enable pressed and use the Joystick to drive the robot. By releasing the Control Enable button or by switching OFF the wireless controller the Joystick Command Enable of the Motor Board is pulled to a low level and only the commands from the Navigation Computer will control the platform.



Figure 34: Wireless Joystick Emitter command

7.3. Shutting Down the FROG Platform

There are different ways to shut down the FROG platform. The first thing to do is to disable the motors and remove the power from the motor drivers. To do that the user should:

1. Push the Emergency button on the back of the robot.
2. Switch OFF the Motor Power Switch.

If the user is only using the Joystick to control platform and all of the computers are disconnected, the user only needs to disconnect the Electronics power button.

If the computers are switch ON, then the user should send a shutdown command to all the computers and wait until they are all OFF before disconnecting the Electronics power button.

7.4. Charging the Batteries

The quick way to charge the batteries is to place the robot in the charging Docking Station. The Docking Station should be powered and all the batteries and power buttons should be ON.

The Docking Station will power the electronics and all the computers.

The final charging method depends on the way that the user wants to charge the robot. Two different approaches can be used:

- **Manual** - Before entering the docking station switch OFF the Motor Drivers Power Button. Place by hand the robot on the Docking Station and switch ON the Motor Driver Power Button. At this moment the Frog robot will start to place all the batteries to Charge.
- **Automatic** - Enter the Docking Station with all the buttons ON. After one second the Frog robot will start to place all the batteries to Charge.

The charging procedure is fully automatic. When the batteries are fully charged the charger will stop the charge (the charging limit is set to 14.6V). When the charge voltage reaches the charging limit the Sensor&Management board will remove the battery from charge.

To return from the charge mode two approaches can be used:

- Switch OFF the Motor Drivers Power Button and remove the robot from the charger by hand. By using this procedure the Sensor&Management Board will disconnect all the batteries from the charger and put them all in enable mode.
- Use the Navigation Computer to switch back all the batteries from charge mode to enable/disable mode. With this procedure the user can avoid current peaks that can reset the computers and also allow the platform to autonomously enter and leave the Docking Station.

8. FROG Integration Meetings and Second Year Review Meeting

The FROG robot has been used in the six integration/data collection meetings held in the Royal Alcázar in Seville and in the Lisbon Zoo.

The integration meeting in the Royal Alcázar in Seville was the first opportunity for all the partners to be in touch with FROG's robot platform. Each partner was able to integrate their own components on the robot structure. In the end of this first integration meeting a map of the Alcázar path was created using the robot lasers and it was possible to perform point-to-point navigation with obstacle avoidance. The robot was able to navigate safely inside the crowded Royal Alcázar. In the same meeting it was possible to collect data for face tracking and sound acquisition. It was also possible to get data for people tracking and also to test developed tracking algorithms. The existing interaction devices, the projector, the touch screen monitor and the laser pointer were tested in the real environment.

The following integration meetings were held in the Lisbon Zoo. The first meeting allowed to test the platform in a completely different environment, with different types of ground and high inclination routes. A new map was built using the frog platform lasers and some fiducial marks were placed in the zoo to be used in the robot localization. The face and people tracking algorithms were tested and data acquisition was performed. The arm and projector were evaluated in this new environment. This opportunity was also used to create a storyboard for the Zoo tour.

The second Zoo integration meeting was a opportunity to try to navigate in the Zoo using the map created in the previous meeting and detect the possible obstacles/problems that the robot could encounter in the Zoo. In this meeting the robot was able to navigate from point-to-point using the behaviour tree to set the next POI (Point-Of-Interest).

The next two meetings were used to prepare the second review meeting demonstration. The partners wanted to show a first version of the complete system in operation while providing a real Zoo tour for the reviewers. This included navigation in all the tour path, play videos and sounds and use the interaction elements already available in the robot.

In this second review meeting FROG was already able to show the following features:

1. Detect and approach people, present himself and start the tour. For this it was using the stereo cameras to detect groups of people, approach them by locating himself on the map and navigate to their location. Then it was able to display a welcome message on the monitor and speakers while showing different patterns on the LED eyes. The behaviour tree was managing all these events.
2. After starting the tour the robot navigated to the first POI. There, it was able to reach the exact posture, start a video and give information about the POI. The projector was tested to show how it will be used in the future. After ending the content presentation about the POI, the robot entered a wait state inviting the users to touch the screen when they are ready to proceed to the next POI. Once they touch the screen, the robot invited them to follow him and moved to the next POI.
3. After visiting all the POIs the robot reaches the end of the tour. Here it says goodbye to the visitors and returns to the starting point.

During the navigation the robot was subject to some disturbances. The robot had to avoid collisions with the people, detect bumper collisions and had to recalculate the trajectory to a POI due to impossibility to follow its normal path. The zoo tour was about 1.8Km and it lasted for about 45 minutes.

9. Tests and Results

The FROG robot is currently fully mounted with all partners components. Several integration weeks were performed mainly in the Royal Alcázar in Seville and in the Lisbon Zoo.

The UPO team has successfully integrated the navigation and mapping software into the robot. After mapping both testbeds (Alcázar and the ZOO) the FROG robot was able to autonomously navigate in these two environments..

The UVA software was also completely tested aboard the FROG robot. Using the stereo camera setup installed in the eyes of the robot, it is possible to detect nearby groups of people while detecting the distance and orientation of each individual person.

The ICL hardware and software is been installed and integrated into the platform.

YD have already some simple behaviour trees that allow the robot to perform simple tasks. It is able to send orders to visit the different POI's while controlling the eyes, arm, monitor and projector.

UT and YD are creating the contents for projection, displaying on the monitor and eyes and arm actions.

During the integration weeks some problems were detected. The next chapter shows some of the detected problems and the solutions found.

9.1. Detected Problems and Solutions

UPO and IDM have detected some problems. They are now discussed.

1. **Motor, drivers and board lines overheat** - UPO and IDM have detected some problems between the control algorithms and the motor controllers that were producing quick overheating of the motors, board lines and motor drivers. 1. Due to the Zoo ground structure and high inclination of the paths, the robot had to stop from time to time due to overheating of the motors. In the Lisbon Zoo this problem was studied and UPO introduced smoother acceleration ramps in the navigation control algorithms. IDM introduced some changes in the hardware with direct impact in the motor power supply, namely, the motor power lines volumes were increased, the motor power cables were changed for high current silicone wires, heatsinks were introduced in both motors and finally use of light weight LiFePO4 batteries. With these changes it was possible to proceed with the data collection and ZOO navigation without the increase of the motors and drivers temperature.
2. **Laser Range Finders Position** - During the UPO tests, it was detected that the lasers mounted on the robot platform were too low. The lasers were detecting the floor. This is being corrected and the lasers will be raised 10 cm from the original position.
3. **Vertical Range Finder Orientation** – A vertical laser was mounted under the head of the robot, this was used to detect ramps and stairs. This solution worked and was able to detect ramps and stairs, but it could not detect small obstacles in the robot path. For that reason this laser was rotated to an horizontal position but with a 45 degree pointing down angle. This will allow the robot to identify stairs, ramps and also detect small obstacles in the front of the robot.
4. **People tracking computer autonomy** - It was detected that the People Tracking computer autonomy, when running the code at full speed, was too low, about 1:30 hours of battery. The problem was that the battery that was powering this computer was also powering all the

electronics aboard the robot. The solution was to use the PC3 Battery to power only the People Tracking computer, the Dalsa stereo cameras and the sound system.

5. **Sound system volume** - During the Zoo meeting it was determined that the sound system installed on the robot was too weak and it was impossible to hear the sounds produced by the interaction system when the robot was outdoors. The solution was to buy a new 30W power amplifier.
6. **FROG robot platform weight** - The weight of the robot is a very important factor that should be taken into account when designing the robot. It will affect the autonomy, the center of mass of the robot, the way the robot can travel on different surfaces, the way it can rotate, the maximum inclination slopes that the robot is allowed to pass or the brake capability. The FROG robot is a very complex machine with lots of components, computers, sensors and actuators; to power all the equipment aboard it will need a total of 8 batteries. The normal 12V 17Ah lead acid batteries, that are reliable for this kind of robots, weight about 5.5 Kg to 6 Kg each, depending on the model, which means in the worst case the robot will carry about 48 Kg of batteries just to power all the components. After working in the Royal Alcázar and in the Lisbon Zoo the group realized that it was crucial to reduce the weight of the robot. The LiFePO4 batteries, with the PCM protection circuit, were the solution. They have the same size, same power, and same voltage and can be charged in the same way as the lead acid batteries. These batteries weigh 2.1 Kg that is $\frac{1}{3}$ of weight of the lead acid batteries. This reduced the total battery weight to 24,6 Kg reducing the total weight of the robot by about 23,4 Kg. The LiFePO4 batteries are compatible with the FROG battery charging system. Since FROG is using 24V motors, they can not be used to power the motors because their specifications do not allow to connect them in series to make 24V and the PCM protection board will cut the power if the maximum current is exceeded

9.2. Analysis of Performance of Low-level Control Loops

The analysis of the performance of Low-level Control Loops and the interrupt driven was performed. This allows to understand the hardware reaction to internal and external events.

The following subsections shows the reaction time of the Low-level Control Loops and interrupt driven events.

9.2.1 Sensor&Management Board Low-Level Control Loops and Interrupt Driven Events

The Sensor&Management board low-level control loop takes:

- 11ms with the sonars OFF;
- 13ms with sonars ON.

The Sonar board low-level control loop takes :

- 300ms to process the 3 groups of a total of 10 sonars.

The Sensor&Management board interrupt driven set commands take:

- 1ms for each set command.

The Sensor&Management board interrupt driven get commands take:

- 6ms Get Batteries Voltages command;
- 3ms Get Auxiliary Voltage command;

- 2ms Get Bumpers;
- 6ms Get Sonars.

9.2.2 Motor Board Low-Level Control Loops and Interrupt driven Events

The Motor board low-level control loop takes:

- 6,5ms to perform a complete loop cycle.

The Motor board interrupt driven set commands take:

- 5ms Set Motor Velocity command;
- 2ms Set Hard-stop timer command.

The Motor board interrupt driven get commands take:

- 8ms Get Encoder ticks command;
- 2ms Get Battery voltage command;
- 8ms Get Temperature command;
- 1ms Get Hard-stop Flag command.

Motor Board Communication from the Sensor&Actuation board

- 2ms to send status (includes bumper information)

From these results it is possible to set a communication cycle with Set Velocity and Get Encoder Ticks of 13 to 15ms, which is about 60Hz. That let us understand that it is possible to get an update of the odometry every 1.5cm when the robot is running at 1m/s.

It is also possible to understand that the Sensor&Management board will take a maximum of 15ms to communicate to the Motors board that a bumper has been activated and will completely stop in the next 15,5cm (with 1m/s of velocity and Hard-stop acceleration of $3,3\text{m/s}^2$), that means that while running at a velocity of 1m/s the robot is able to stop in a distance of 17 cm.

Has it was said the robot motor/wheels will stop in less that 17cm, but depending on the surface/inclination the tires can slip and this distance may increase.

9.3. Performance Analysis of Robot Odometry and Velocity Control

Odometry is a key aspect for ground robots; it allows short-term estimation of the robot position based on wheel encoders, enabling a simple yet effective method for integrating the sensors on-board the robot with the environment.

On the other hand, the ability of the robot to follow the speed commands given by the robot planner is another important aspect for a smooth and natural robot motion. The planner will take into account the robot limitations in speed and acceleration in order to draw trajectories affordable for the system. However, the planner will probably generate not feasible trajectories if the acceleration or speed limits are not well parametrized, with the consequent impact in the robot motion.

This sections analyses both robot odometry and velocity commands, and proposes a calibration method whether needed. Experiments will be carried out in order to identify the odometry errors and how the robot follows the velocity commands, and a calibration method will be proposed in order to improve the velocity commands.

9.3.1. Experimental Setup

In order to test the precision of the odometry provided by the FROG platform a second sensor for distance estimation was used, a laser scanner placed in front of the vehicle. The robot was commanded to follow a straight line in a long corridor (25 meters) ended with a closed door, so the scanner can estimate the distance from the robot to the end of the corridor during the experiment (see Figure 35).

The purpose of the experiment is threefold:

- Measuring the distance travelled by the robot using the laser scanner in order to have a ground-truth for the computed odometry during the experiment.
- Measuring the angle deviation that the robot undergoes at the end of the trajectory.
- Analysing how the robot follows the speed commands, we can compute the speed ground-truth also from the laser scanner.

Thus, four tests have been performed (commanded velocities 0.5, 0.7, 1.0 and 1.3 m/s). The platform was placed at the start point in the beginning of the corridor and the front laser measures the distance to the end of the hallway. The velocities are commanded to the platform (in a step-command) during a certain time.

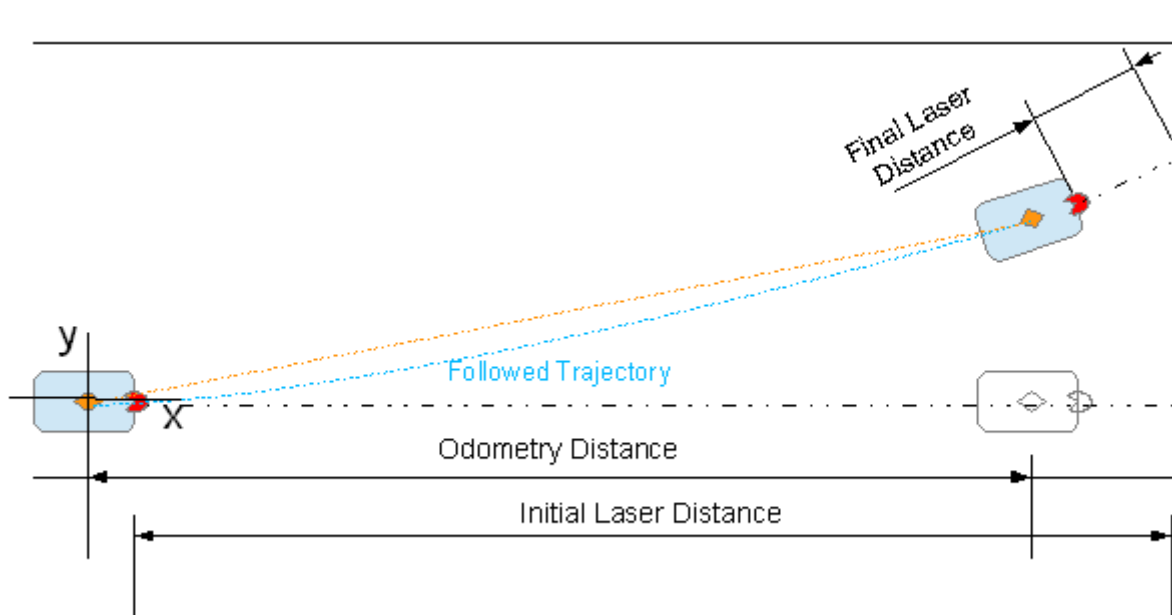


Figure 35: Experimental setup for odometry and velocity identification

The results of the experiments for odometry can be seen in Figure 36. It can be seen how the errors in the traversed distance is very small in all experiments, with a mean error at the end of the trajectory of 10cm approximately, which is about 0.5% of the total distance travelled (17 meters). It is worth to mention that this error is computed in an almost ideal scenario with good wheel gripping and without turning, but it is definitely a good estimation and calibration process will probably not provide significant improvement in the measurement.

The deviation angle with respect the straight trajectory was also computed in the odometry experiments. This angle was from 10° to 15° , which is also a not significant distortion after travelling 17 meters.

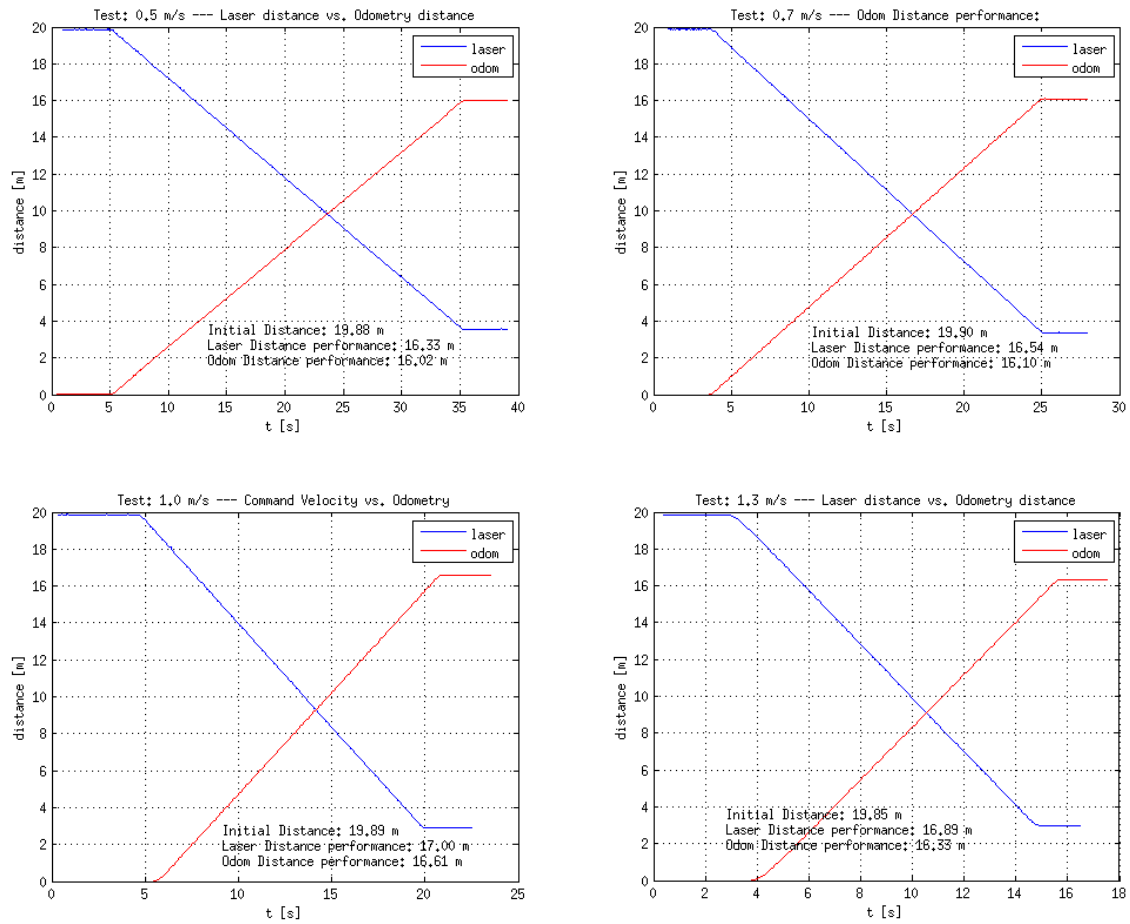


Figure 36: Results of the experiments for odometry

The speed was also measured during the experiment and compared with the commanded speed; the results are shown in Figure 37. It can be seen how the robot quickly follows the commanded speed, introducing a significant acceleration to reach the commanded reference, which points to a very responsive system and decrease the planner constraints regarding accelerations. However, the speed is followed with a small permanent error. This error could lead to approach obstacles faster than planned, making the navigation less smooth and forcing local replanning more frequently.

Finally, the robot was commanded to follow angular speeds. The result of the angular velocity command was compared with the speed provided by the yaw gyro once BIAS is remove. The results are presented in the following table:

Angular speed commanded	Angular speed measured
0.2 rad/s	0.10 to 0.12 rad/s
0.3 rad/s	0.21 to 0.23 rad/s

It can be seen how the angular speed has some errors following the reference, producing differences between real angular velocity and commanded velocity in the order of 30% of the reference.

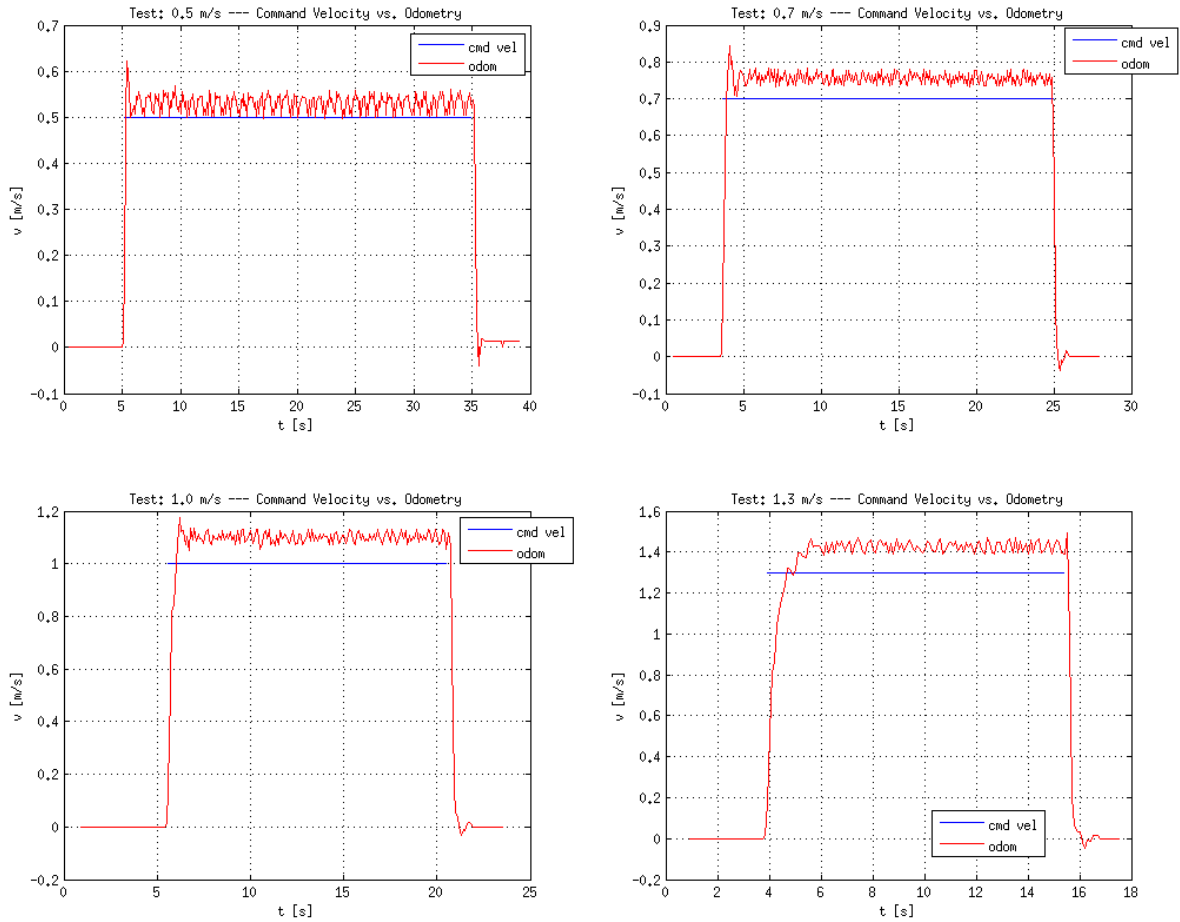


Figure 37: Results of the experiments for commanded speed

9.3.2. Linear and Angular Velocity Commands Calibration

The experimental setup showed that FROG platform provides a very stable odometry, with errors in the order of 0.5% of the distance travelled by the robot. Nevertheless, the velocity (linear and angular) shown to have small permanent errors to follow the reference. This section proposes a recalibration of the linear and angular velocity commands in order to adapt the reference commands to the real speed reached by the robot.

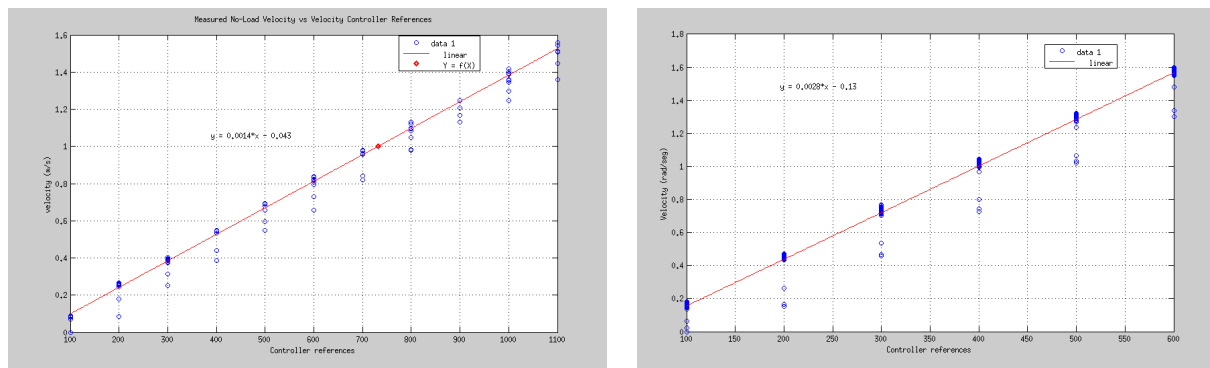


Figure 38: Calibration experiments for linear and angular velocities

Then, assuming the odometry as being very precise (experimentally demonstrated above), the robot was commanded at different linear and angular velocities and the response was stored in a log file. This information was later on used to calibrate the relationship between servo reference and speed. The results of such experiments are shown in Figure 38.

As a result, the equation that relates servo reference was pre-computed, obtaining the following results:

1. Linear velocity. Where X stands for the servo reference and Y for the desired linear velocity.

$$Y = 0.0014 \cdot X - 0.043$$

2. Angular velocity. Where X stands for the servo reference and Y for the desired angular velocity.

$$Y = 0.0028 \cdot X - 0.13$$

With these new velocity/servo equations, the velocity experiments were carried out again in order to validate the results. Figure 39 shows the linear speed experiments, it can be seen how the robot follows the speed commands with very small errors. The angular velocity experiments were also re-evaluated and showed very small errors.

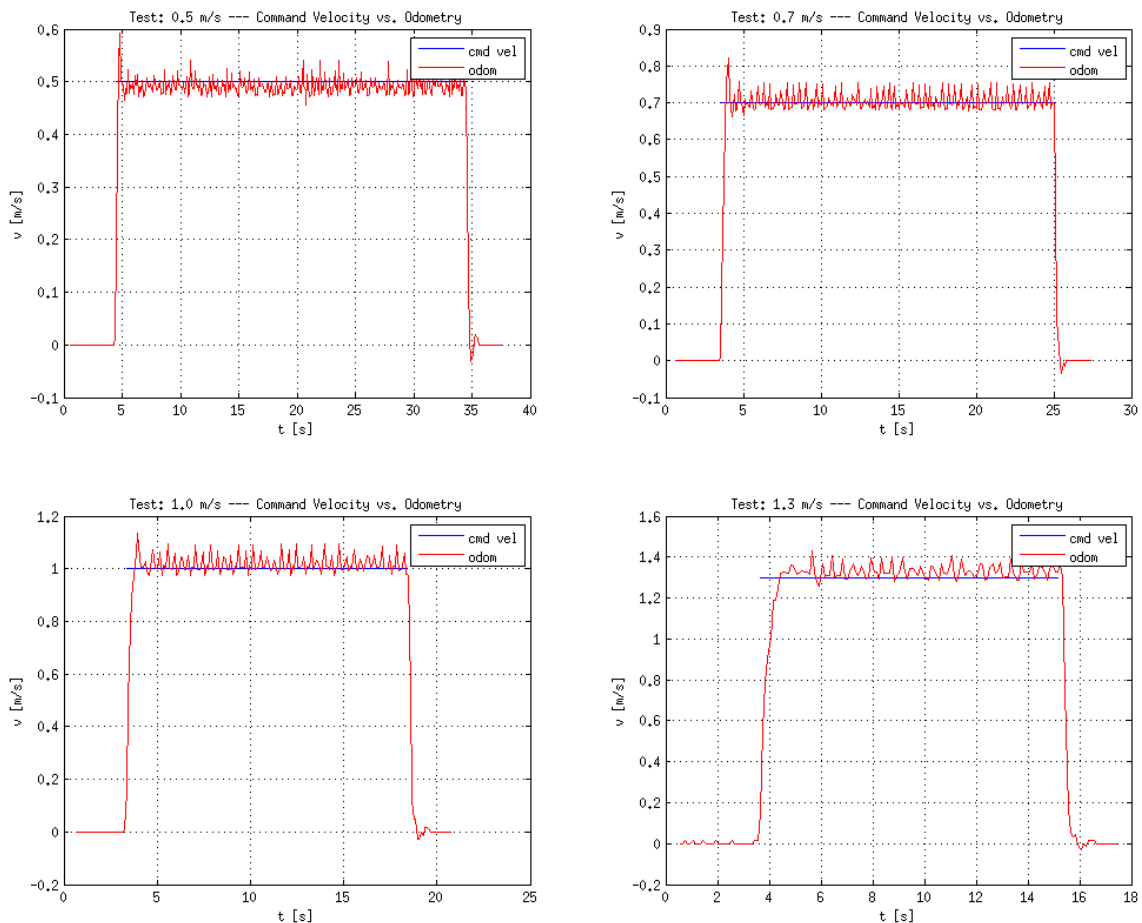


Figure 39: Linear velocity experiments after calibration

Annex A. Board Connections

The following chapter shows the cable connections between each board and each component. Each connector in a board is represented by a letter or a letter and a number. Each cable is represented by a number. The following figures show the letters that correspond to each connector.

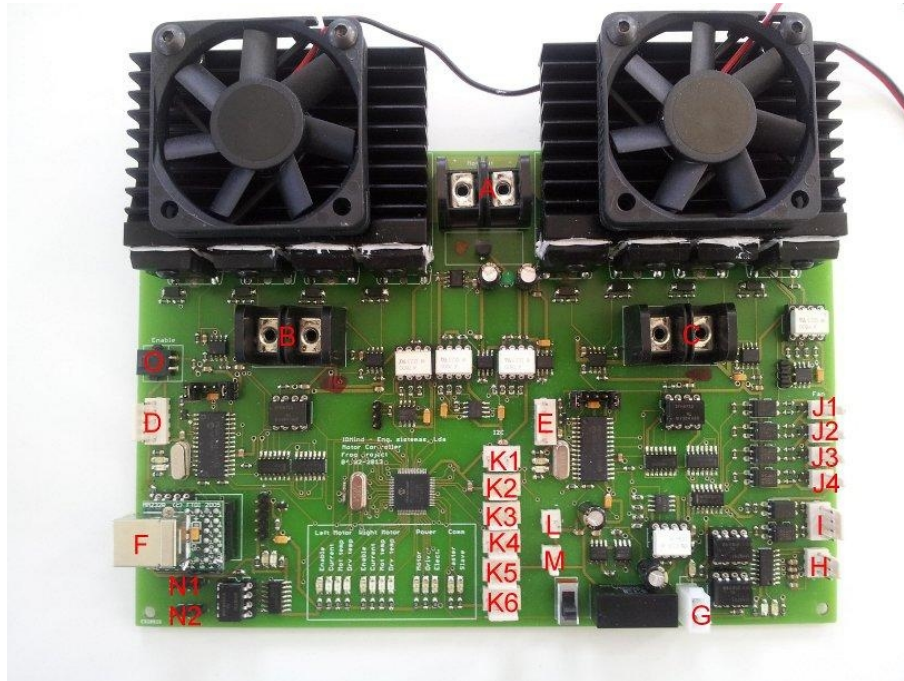


Figure 40: Motor Board connectors

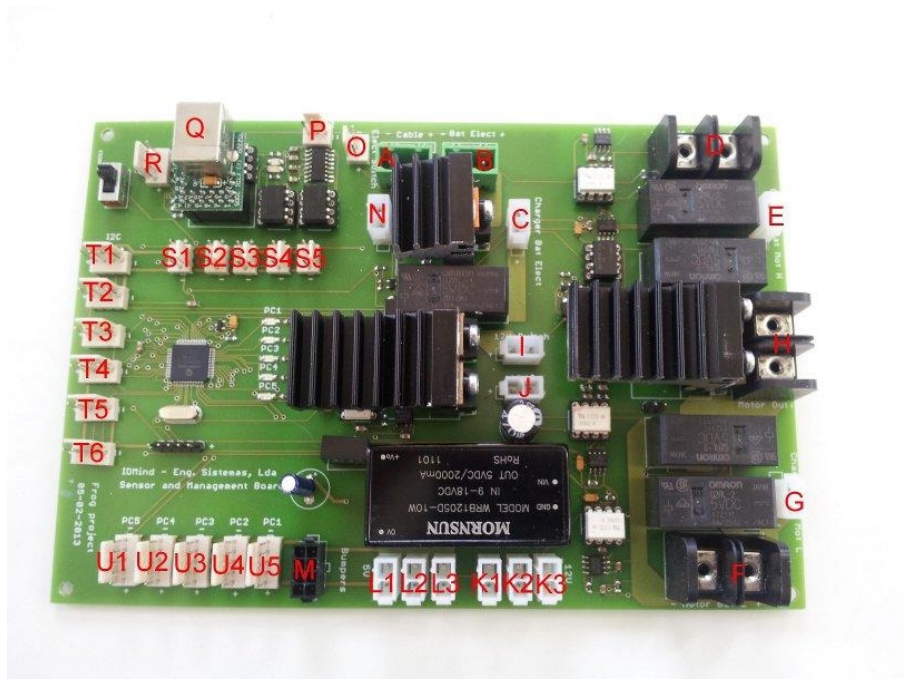


Figure 41: Sensor&Management Board connectors

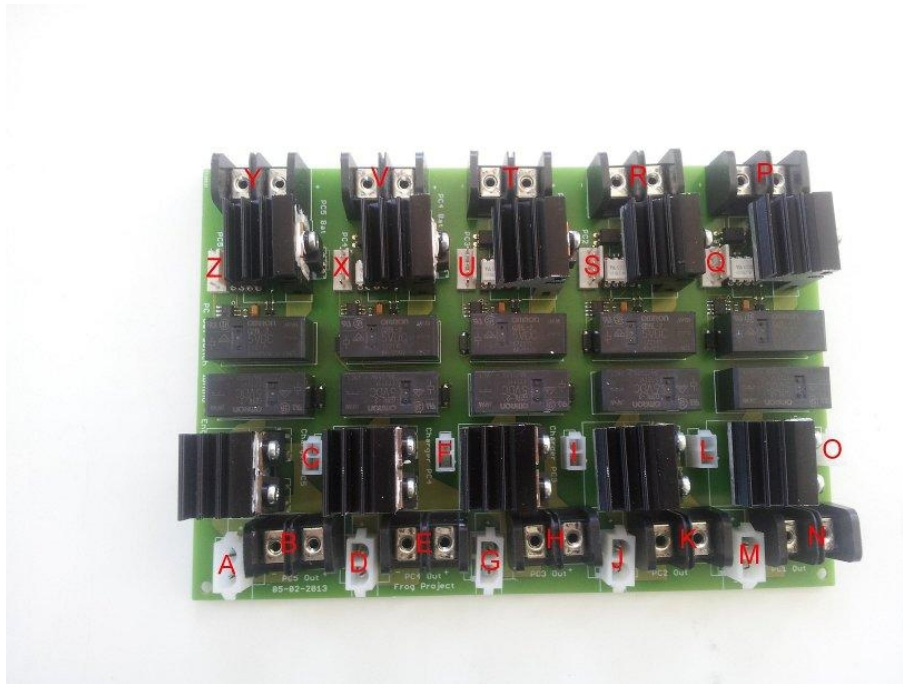


Figure 42: Switch Board connectors



Figure 43: Charger Board connectors

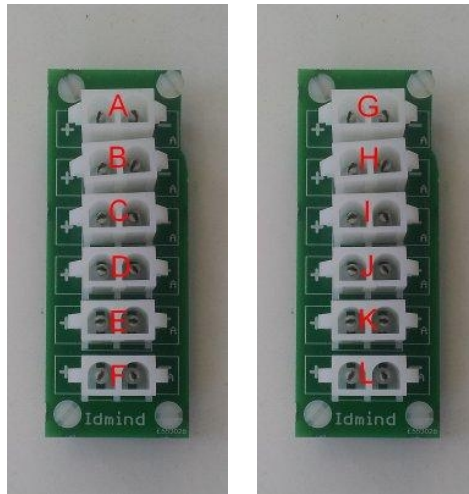


Figure 44: Power Extension Boards

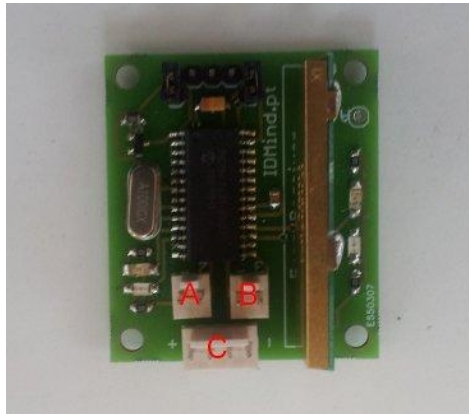


Figure 45: Wireless Receiver Board

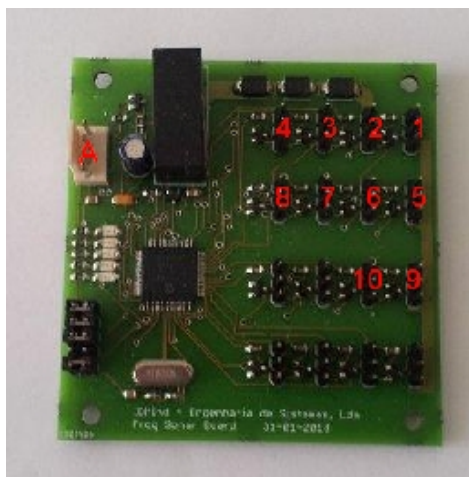


Figure 46: Sonars Board

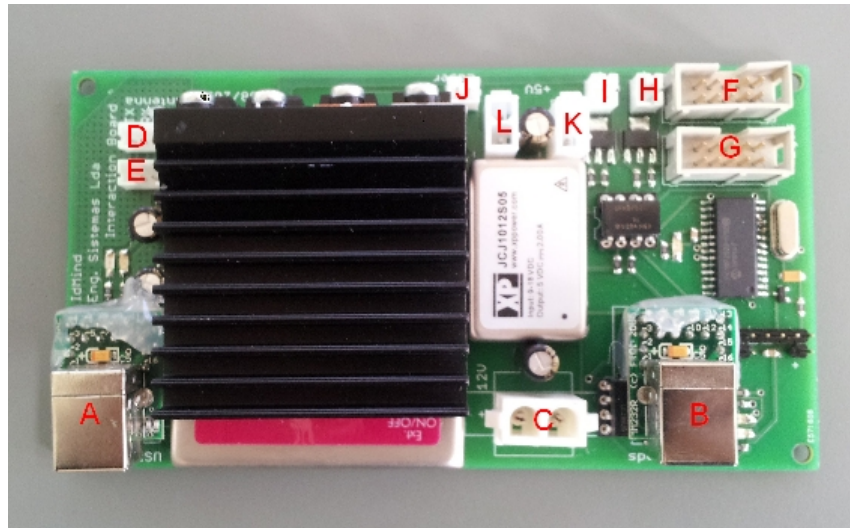


Figure 47: Interaction Board

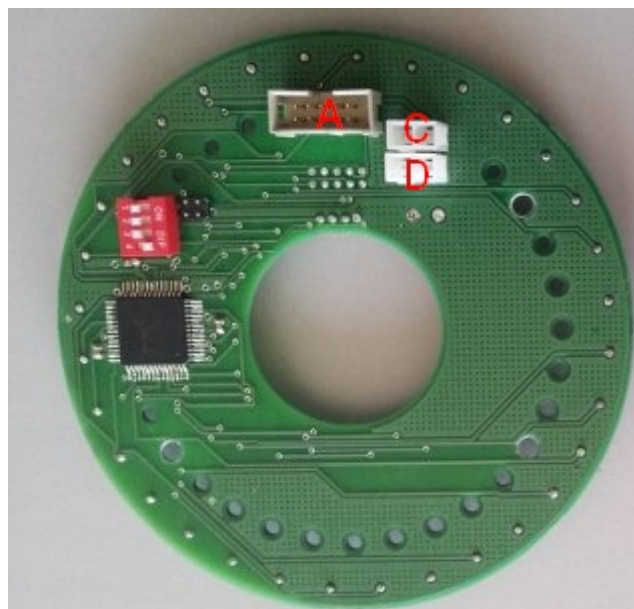


Figure 48: Eyes controller Board

Table 2 gives the connection between each connector and each cable.

Cable	From:	Port	To:	Port
00	Motor Board	B	Right Motor	in
01	Motor Board	C	Left Motor	in
10	Motor Board	E	Right Encoder	out
11	Motor Board	D	Left Encoder	out
12	Motor Board	A	Sensor&Management Board	H
13	Motor Board	G	Sensor&Management Board	K3

14	Motor Board	H	Sensor&Management Board	P
15	Motor Board	O	Navigation Computer USB port	out
16	Motor Board	O	Motor Drive Enable Switch	out
17	Sensor&Management Board	O	Electronic Switch	out
18	Sensor&Management Board	S5	Motor Power Switch	out
19	Motor Board	J1	Motor Drivers Fans	in
22	Motor Board	L	Wireless Receiver Board	A
23	Motor Board	M	Wireless Receiver Board	B
24	Motor Board	K6	Wireless Receiver Board	C
25	Sensor&Management Board	Q	Navigation Computer USB port	out
26	Sonar Board	A	Sensor&Management Board	T1
27	Interaction Board – Arm	A	Interaction Computer	USB
28	Interaction Board – Eyes	B	Interaction Computer	USB
29	Interaction Board	C	12V no regulated input	in
30	Interaction Board	D, E	Arm connectors	out
31	Interaction Board	L	Left eye	C
32	Interaction Board	K	Right eye	C
33	Interaction Board	F	Left eye	A
34	Interaction Board	K	Right eye	A
35	Interaction Board	I, H	Arm green LEDs	out
36	Interaction Board	J	Laser pointer	out
50	Sensor&Management Board	M	Bumper	out
51	Sensor&Management Board	T1	Sonar Board	A
52	Sensor&Management Board	T2	Temperature and humidity sensor	out
55	Power Switch Board	M	Power Extension	B
56	Power Switch Board	J	Power Extension	D
57	Power Switch Board	G	Power Extension	H
58	Power Switch Board	D	Power Extension	J
59	Power Switch Board	A	Power Extension	K
61	Battery Charger Board	B	Battery Charger fan	in
62	Battery Charger Board	B	Battery Charger fan	in
63	Battery Charger Board	H	Power Extension	F
64	Battery Charger Board	K	Power Extension	L

65	Battery Charger Board	F	Power Switch Board	C
66	Battery Charger Board	G	Power Switch Board	F
67	Battery Charger Board	I	Power Switch Board	I
68	Battery Charger Board	J	Power Switch Board	L
69	Battery Charger Board	L	Power Switch Board	O
71	Sensor&Management Board	G	Battery Charger Board	A
72	Sensor&Management Board	C	Battery Charger Board	C
73	Sensor&Management Board	E	Battery Charger Board	D
74	Power Switch Board	N	PC1 Auxiliary 12V power supply	in
75	Power Switch Board	K	PC2 Auxiliary 12V power supply	in
76	Power Switch Board	H	PC3 Auxiliary 12V power supply	in
77	Power Switch Board	E	PC4 Auxiliary 12V power supply	in
78	Power Switch Board	B	PC5 Auxiliary 12V power supply	in
80	Sensor&Management Board	U5	Power Switch Board	Q
81	Sensor&Management Board	U4	Power Switch Board	S
82	Sensor&Management Board	U3	Power Switch Board	U
83	Sensor&Management Board	U2	Power Switch Board	X
84	Sensor&Management Board	U1	Power Switch Board	Z
85	Sensor&Management Board	L3	Auxiliary 5V Power Supply	in
86	Sensor&Management Board	K1	Auxiliary 12V Power Supply	in
87	Battery Charger Board / Power Extension	B/G	Charging Station Cable	out
88	Power Extension	A	Charging Station Cable	out
89	Battery Charger Board	E	Charging Station Cable	out
90	Sensor&Management Board	A	Charging Station Cable	out
91	Sensor&Management Board	F	Motor L Battery	out
92	Sensor&Management Board	D	Motor H Battery	out
93	Sensor&Management Board	B	Electronic Battery	out
95	Power Switch Board	P	PC1 Battery	out
96	Power Switch Board	R	PC2 Battery	out
97	Power Switch Board	T	PC3 Battery	out
98	Power Switch Board	V	PC4 Battery	out
99	Power Switch Board	Y	PC5 Battery	out

Table 2: cable connections between boards

Annex B. Board Controllers Software Protocol

The Onboard Navigation Computer is able to communicate with the Motor and Sensor&Management Boards using two independent USB Ports. In each board a FTDI USB to RS232 converter converts the USB data connection to a TTL Serial PORT data connection for the microcontrollers on the board.

The Interaction Computer will communicate with the Interaction Board through 2 USB communication ports. The first one will control the Eyes and the Laser pointer and a second USB Port will control the 3DOF arm. The first will use the same communication protocol used by the Motor and Sensor&Management Boards and the second will use the Herkulex protocol⁵.

All the USB board connectors can be connected using the following UART settings:

Baud rate: 115200bps
Data bits: 8
Stop bits: 1
Parity bit: No parity
HW Flow Control: Disable

To communicate with the robot boards the following protocol is used (except the 3D arm):

Write -> [Command][First Byte][Second Byte][...]

Read -> [Command][First Byte][Second Byte] [...][Send Number][Checksum_H][Checksum_L]

Where:

[**Command**] is the set or get command;

[**Send Number**] is an incremental number that counts all the sends that each microcontroller has already performed;

[**Checksum_H:Checksum_L**] is the checksum of the [Command]+[First Byte]+[Second Byte]+ [...]+[Send Number].

B.1. Motor Board Communication Protocol

Set Velocity command:

[Command] = 0x56

[Left wheel Velocity] = [Left wheel Velocity High][Left wheel Velocity Low]

[Right wheel Velocity] = [Right wheel Velocity High][Right wheel Velocity Low]

Write -> [0x56][Left wheel Velocity High][Left wheel Velocity Low][Right wheel Velocity High][Right wheel Velocity Low]

Read -> [0x56][Send Number][Checksum_H][Checksum_L]

To obtain the Left and Right Wheel Velocities, use the following known equations:

Left wheel Velocity = -(Linear Velocity + Rotation Velocity);

Right wheel Velocity = (Linear Velocity - Rotation Velocity);

To obtain the High and Low bytes use:

5 <http://www.robotshop.com/content/PDF/manual-drs-0101.pdf>

Left wheel Velocity High = (byte) (Left wheel Velocity >> 8);
Left wheel Velocity Low = (byte) (Left wheel Velocity & 0xFF);
Right wheel Velocity High = (byte) (Right wheel Velocity >> 8);
Right wheel Velocity Low = (byte) (Right wheel Velocity & 0xFF);

Get Encoders Ticks command:

[Command] = 0x4A

Write -> [0x4A]

Read -> [0x4A][Left Motor Ticks High][Left Motor Ticks Low][Right Motor Ticks High][Right Motor Ticks Low][Send Number][Checksum_H][Checksum_L]

To get the Left Motor Ticks:

Left Motor Ticks = (int)Left Motor Ticks High*256 + Left Motor Ticks Low;

To get the Right Motor Ticks:

Right Motor Ticks = (int)Right Motor Ticks High*256 + Right Motor Ticks Low;

To calculate the CheckSum use the following formula:

Checksum = 0x4A + Left Motor Ticks High + Left Motor Ticks Low + Right Motor Ticks High + Right Motor Ticks Low + Send Number;

Get Motor Current command:

[Command] = 0x50

Write -> [0x50]

Read -> [0x50] [Left Motor current] [Right Motor current] [Send Number] [Checksum_H] [Checksum_L]

To get the real measured current do:

Left Motor measured current = (double)((double)[Left Motor current]/10.0);

Right Motor measured current = (double)((double)[Right Motor current]/10.0);

Get Motor Board Voltage command:

[Command] = 0x51

Write -> [0x51]

Read -> [0x51] [Motor Voltage] [Driver Voltage] [Electronics Voltage] [Power Status] [Send Number] [Checksum_H] [Checksum_L]

Where:

[Motor Voltage] is the voltage measure of the power supply of the motors. In normal operation will have a value between 21V and 26 V.

Motor voltage power = (double)((double)[Motor Voltage]*2.0 / 10.0);

[Driver Voltage] is the voltage measured of the power supply of the drivers for the motors. In normal operation the value should be between 12 to 15V.

Driver voltage power = (double)((double)[Driver Voltage] / 10.0);

[Electronics Voltage] is the voltage measured of the power supply of the control electronics. In normal operation the value should be between 9.5 to 13V.

Electronics voltage power = (double)((double)[Electronics Voltage] / 10.0);

[Power Status] is the voltage status [AXXXBCD].

where:

- A =0 : Motor Drivers disabled;
- A =1 : Motor Drivers enabled;
- B =0 : Motor Power not OK;
- B =1 : Motor Power OK;
- C =0 : Driver Power not OK;
- D =1 : Driver Power OK;
- D =0 : Electronic Power not OK;
- D =1 : Electronic Power OK.

Get temperature command:

[Command] = 0x52

Write -> [0x52]

Read -> [0x52] [Left Motor Temperature] [Right Motor Temperature] [Left Motor Driver Temperature] [Right Motor Driver Temperature] [Temperature Status] [Send Number] [Checksum_H] [Checksum_L]

where:

[Left Motor Temperature] is the temperature of the left motor in °C;

[Right Motor Temperature] is the temperature of the right motor in °C;

[Left Motor Driver Temperature] is the temperature of the Left Motor Driver in °C;

[Right Motor Driver Temperature] is the temperature of the Right Motor Driver in °C;

[Temperature Status] is the temperature status [XXXXABCD]

where:

- A=0 : Left Motor Temperature less than 40°C;
- A=1 : Left Motor Temperature more than 40°C;
- B=0 : Right Motor Temperature less than 40°C;
- B=1 : Right Motor Temperature more than 40°C;
- C=0 : Left Motor Driver Temperature less than 40°C;
- C=1 : Left Motor Driver Temperature more than 40°C;
- D=0 : Right Motor Driver Temperature less than 40°C;
- D=1 : Right Motor Driver Temperature more than 40°C.

note: The 40°C value can be changed after a better understanding of the normal operation of the robot.

Set cooling fans command:

[Command] = 0x53

Write -> [0x53][Fan Control]

Read -> [0x53][Send Number] [Checksum_H] [Checksum_L]

where:

[Fan Control] allow automatic or manual control of the fans to cool the motors and the drivers. [AXXXBCDE]

[A]=0 manual control of the fans

[A]=1 automatic control of the fans. One or more fans will start to cool a motor or a motor driver when its temperature reaches the 40°C and stop to work when the temperature is reduced to about 25°. It will not be possible to control the fans manually.

In manual control mode:

- [B]=0 : turn OFF the left motor fan;
- [B]=1 : turn ON the left the motor fan;
- [C]=0 : turn OFF the right motor fan;
- [C]=1 : turn ON the right the motor fan;
- [D]=0 : turn OFF the left motor driver fan;
- [D]=1 : turn ON the left motor driver fan;
- [E]=0 : turn OFF the right motor driver fan;
- [E]=1 : turn ON the right motor driver fan.

In automatic control the controller will update automatically the [B] to [E] bits. Using 0 when the fans are OFF and 1 when the fans are ON.

note: The 40°C and 25°C values can be changed after a better understanding of the normal operation of the robot.

Get cooling fans command:

[Command] = 0x54

Write -> [0x54]

Read -> [0x54][Fan Control][Send Number] [Checksum_H] [Checksum_L]

where:

[Fan Control] reads the Fan Control status byte described previously on the “Set cooling fans command”

Get communication status command:

[Command] = 0x55

Write -> [0x55]

Read -> [0x55][Communication Status][Send Number] [Checksum_H] [Checksum_L]

where:

[Communication Status] is the communication status between the Motor board and the navigation computer, the PI motor controller slaves and the Sensor&Management Board.
[AXXXBXCD]

- [A]=0 : Normal communication with the navigation computer;
- [A]=1 : No communication or communication error with the navigation computer;
- [B]=0 : Normal communication with the Sensor&Management Board;
- [B]=1 : No communication or communication error with the Sensor&Management Board;
- [C]=0 : Normal communication with the Left PI motor controller Slave;
- [C]=1 : No communication or communication error the Left PI motor controller Slave;
- [D]=0 : Normal communication with the Right PI motor controller Slave;
- [D]=1 : No communication or communication error the Right PI motor controller Slave.

Set Hardstop timer command:

[Command] = 0x58

Write -> [0x58][Hardstop Timer]

Read -> [0x58][Send Number] [Checksum_H] [Checksum_L]

where:

[Hardstop Timer] allow to set the time that the robot will stay in Hardstop Mode. It is possible to select between 0s to 255s.

Get Hardstop Status:

[Command] = 0x59

Write -> [0x59]

Read -> [0x59][Hardstop Status][Send Number] [Checksum_H] [Checksum_L]

where:

[Hardstop Status] is the Hardstop status [XXXXXXXXA].

where:

A =0 : Hardstop Enabled;

A =1 : Hardstop Disabled;

Get Firmware version number command:

[Command] = 0x20

Write -> [0x20]

Read -> [0x20] [Firmware information (25bytes)] [Send Number] [Checksum_H]
[Checksum_L]

where:

[Firmware information] is a set of 25 bytes in ascii with the following information:

“Board2 fw A.BC YYYY/MM/DD” for example “Board2 fw 1.00 2014/02/14”

A is the version and BC the revision number.

B.2. Sensor&Management communication protocol

Enable/Disable/Charge Motors Power:

[Command] = 0x45

Write -> [0x45][Control]

Read -> [0x45] [Send Number] [Checksum_H] [Checksum_L]

where:

[Control] = 0 : Motor power Disabled;

[Control] =1 : Charge motor power enable Mode;

[Control] = 2 : Motor power Enable.

Enable/Disable/Charge Auxiliary Power battery 1 to 5:

[Command] = 0x40 to 0x44

Write -> [0x40 to 0x44][Control]

Read -> [0x40 to 0x44] [Send Number] [Checksum_H] [Checksum_L]

where:

- [Control]** = 0 : Auxiliary power battery (1 to 5) Disabled;
- [Control]** = 1 : Charge auxiliary power battery (1 to 5) enable Mode;
- [Control]** = 2 : Auxiliary power battery (1 to 5) Enable.

Enable/Charge Electronic Power battery:

[Command] = 0x46

Write -> [0x46][Control]

Read -> [0x46] [Send Number] [Checksum_H] [Checksum_L]

where:

- [Control]** =1 : Charge electronic battery enable Mode;
- [Control]** =2 : Electronic power battery Enable.

Get Batteries Voltage command:

[Command] = 0x51

Write -> [0x51]

Read -> [0x51] [24V Motor Battery] [Motor High Battery] [Motor Low Battery] [Electronics Battery] [Charger] [Send Number] [Checksum_H] [Checksum_L]

Where:

[24V Motor Battery] is the sum of the Motor High and Motor Low battery voltages. When in charging mode the value is +/- 0V.

To obtain the real measured voltage of this battery the following formula must be used:

$$\text{Motor battery voltage} = (\text{double})((\text{double})[24V \text{ Motor Battery}] * 2.0 / 10.0);$$

[Motor High Battery] is the battery that will connect to the (+) of the Motor Drivers Power supply and the (+) of the [Motor Low Battery]. It will only be different from 0V when in charging mode.

To obtain the real measured voltage of this battery the following formula must be used:

$$\text{Motor High battery voltage} = (\text{double})((\text{double})[Motor \text{ High Battery}] / 10.0);$$

[Motor Low Battery] is the battery that will connect to the (-) of the Motor Drivers Power supply and the (-) of the [Motor High Battery]. It will be only different from 0V when in charging mode.

To obtain the real measured voltage of this battery the following formula must be used:

$$\text{Motor Low battery voltage} = (\text{double})((\text{double})[Motor \text{ High Battery}] / 10.0);$$

[Electronics Battery] is the battery that powers all the electronics and the onboard Navigation Computer.

To obtain the real measured voltage of this battery the following formula must be used:

$$\text{Electronic battery voltage} = (\text{double})((\text{double})[Electronics \text{ Battery}] / 10.0);$$

[Charger] is the power from the docking station.

To obtain the real measured voltage in the charger the following formula must be used:

$$\text{Charger voltage} = (\text{double})((\text{double})[Charger] / 10.0);$$

Get Auxiliary Batteries Voltage command:

[Command] = 0x52

Write -> [0x52]

Read -> [0x52] [PC1 Battery] [PC2 Battery] [PC3 Battery] [PC4 Battery] [PC5 Battery] [Send Number] [Checksum_H] [Checksum_L]

Where:

[PC1-PC5 Battery] are auxiliary batteries that can be used to power computers or other electronic devices that the robot can carry.

To obtain the real measured voltage of this battery the following formula must be used:

$$\text{PC1-PC5 battery voltage} = (\text{double})((\text{double})[\text{PC1-PC5}] / 10.0);$$

Get Bumpers command:

[Command] = 0x53

Write -> [0x53]

Read -> [0x53] [Bumpers][Send Number] [Checksum_H] [Checksum_L]

Where:

[Bumpers] allows the detection of collisions in different parts of the robot. The received [bumpers] byte indicates if a bump is being detected. Bumper byte has the following composition [0000ABCD] where:

[A] is the front left bump;

[B] is the front right bump;

[C] is the rear left bump;

[D] is the rear right bump.

Each bit can assume two possible values: 0 if no bump is being detected or 1 if a bump is being detected.

Get Sonars command:

[Command] = 0x54

Write -> [0x54]

Read -> [0x54] [Sonar0_High_Byte] [Sonar0_High_Byte] ---- [Sonar9_High_Byte] [Sonar9_High_Byte] [Send Number][Checksum_H] [Checksum_L]

The get sonars command will retrieve the distance information of 10 sonar sensor in cm. To get the distance value in cm the following calculation must be performed:

$$\text{Sonar 0..9} = (\text{int})((\text{int})[\text{Sonar0..9_High_Byte}] * 256 + [\text{Sonar0..9_Low_Byte}]) \times 0.0138 + 20.07;$$

note: the number of sonar sensors to be used will depend on future tests.

Enable/Disable Sonars command:

[Command] = 0x55

Write -> [0x55][Control]

Read -> [0x55] [Send Number] [Checksum_H] [Checksum_L]

where:

[Control] = 0 : Disable sonar readings;

[Control] = 1 : Enable sonar readings.

Get Environment sensors command:

[Command] = 0x56

Write -> [0x56]

Read -> [0x56] [Temperature] [Humidity] [Rain] [Send Number][Checksum_H] [Checksum_L]

The Environment sensor command retrieves the information of the environment temperature, humidity and rain where:

[Temperature] retrieves a temperature between 0°C to 125°C;

[Humidity] retrieves the relative humidity between 0% and 100%;

[Rain] Detects whether it is or is not raining.

Get communication status command:

[Command] = 0x57

Write -> [0x57]

Read -> [0x57] [Communication Status] [Send Number] [Checksum_H] [Checksum_L]

The communication status command retrieves the information about the communication status where:

[Communication Status] is the communication status between the Sensor&Management Board and the navigation computer, the Sonar Controller board, the Motor Controller Board and other slave devices. [AXBXCXEF]

[A]=0 : Normal communication with the navigation computer;

[A]=1 : No communication or communication error with the navigation computer;

[B]=0 : Normal communication with the Motor Controller Board;

[B]=1 : No communication or communication error with the Motor Controller Board;

[C]=0 : Normal communication with the Sonar Controller Board;

[C]=1 : No communication or communication error the Sonar Controller Board;

[E]=0 : Normal communication with the Slave device 1;

[E]=1 : No communication or communication error the Slave device 1;

[F]=0 : Normal communication with the Slave device 2;

[F]=1 : No communication or communication error the Slave device 2.

Get Firmware version number command:

[Command] = 0x20

Write -> [0x20]

Read -> [0x20] [Firmware information (25bytes)] [Send Number] [Checksum_H]
[Checksum_L]

where:

[Firmware information] is a set of 25 bytes in ascii with the following information:

“Board2 fw A.BC YYYY/MM/DD” for example “Board1 fw 1.00 2014/02/14”

A is the version and BC the revision number.

B.3. Interaction Communication Protocol

LED Eyes Control

[Command] = 0x40

Write -> [0x40][Eye][32 bytes of 8 LEDs]

Read -> [0x40] [Send Number] [Checksum_H] [Checksum_L]

where:

[Eye] = 0 : Left Eye;

[Eye] = 1 : Right Eye.

[32 bytes of 8 LEDs] - the LED controller is able to control a matrix of 32 LED columns (32 bytes) by 8 LED rows (8 bits). The Eyes are organized in 32 LED columns by 3 LED rows. Each column is represented by a single 8 bits byte. Where the least significant bit will activate the external LED circle of the eye, the second bit will activate the middle line and the third will activate the inner LED circle. The other 5 bits are not used. To enable a LED the bit must be changed to '1' and to disable the bit must be changed to '0'.

LED Eyes intensity Control

[Command] = 0x41

Write -> [0x41][Eye][Intensity]

Read -> [0x41] [Send Number] [Checksum_H] [Checksum_L]

where:

[Eye] = 0 : Left Eye;

[Eye] = 1 : Right Eye;

[Eye] = 2 : Both Eyes.

[Intensity] = [0-15] is the intensity level for the each eye. The maximum level is 15 and the minimum is 0.

Laser Pointer Control

[Command] = 0x42

Write -> [0x42][Enable Control]

Read -> [0x42] [Send Number] [Checksum_H] [Checksum_L]

where:

[Enable Control] = 1 : Laser pointer ON;

[Enable Control] = 0 : Laser Pointer OFF;

Laser Pointer green LEDs control

[Command] = 0x43

Write -> [0x43][Intensity][time]

Read -> [0x43] [Send Number] [Checksum_H] [Checksum_L]

where:

[Intensity] = [0-100] is the intensity percentage of the signal that will drive the green LEDs.

[time] = [0 -255] is the time that will take to reach the new intensity level. Where [0] is 0 seconds [1] is 0.01s and [255] is 2,55s.

Annex C. Robot Platform CAD Measurements

The drawings presented in this section should be used as a reference. As soon as the production of the shell is concluded this section will be updated with the final CAD drawings.

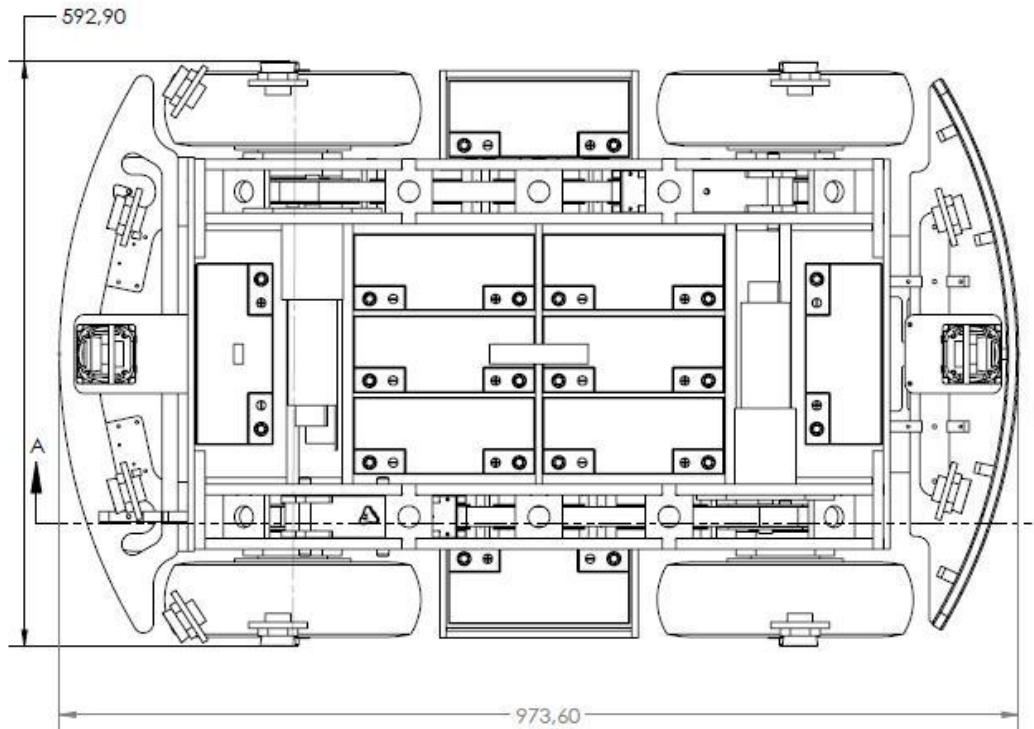


Figure 49: FROG platform. Top view

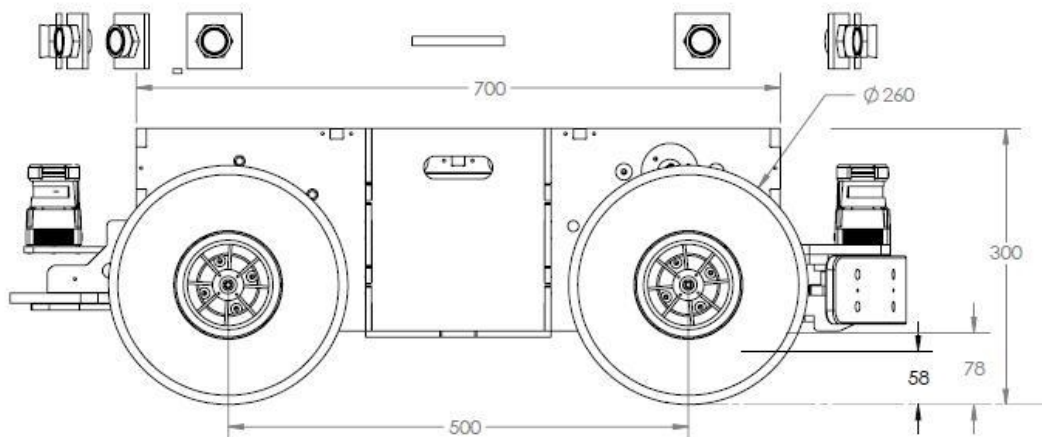


Figure 50: FROG platform. Left view.

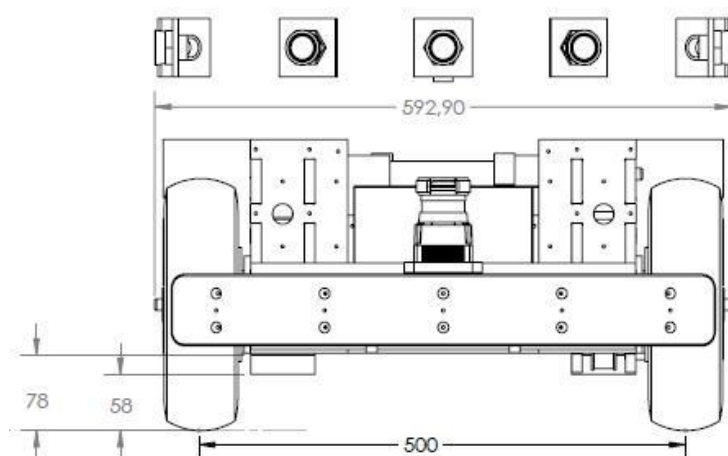


Figure 51: FROG platform. Front view.

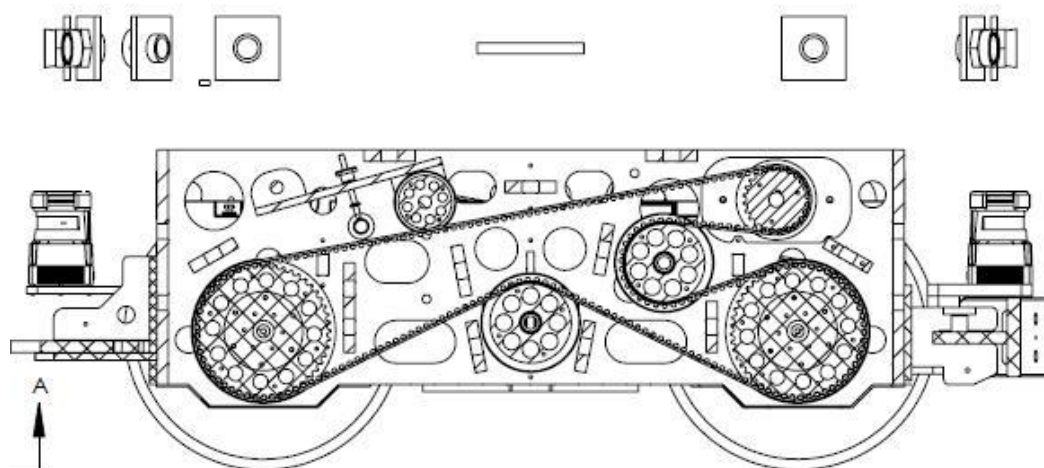


Figure 52: FROG Platform. Pulley system left view.