Editor: Raúl Sebastiá
Sub-Editor: Rafa López

GUARDIANS

D5.1 Guardians prototype

CogSys
Cognitive Systems

Contents

1 Introduction

This deliverable consists of the following parts:

- Physical prototype of the Guardians robot.
- Complete list of the robot hardware and software manuals.

The robot is already mounted and working properly, and we present in this document the complete description of this robot from the following points of view:

- System Architecture.
- System Installation and Configuration.
- Power and Control.
- System Start-Up.
- Mechanical Components and Maintenance.
- Electrical Drawings.

The final goal of this document is to give all the needed robot information to a user that decides to use a Guardians mobile robot.

2 Motivation

Nowadays there are a lot of different types of robots for a lot of different types of applications. Particularly, for fire assisting tasks robots usually have a heavy weight, slow speed and reduced mobility. The main motivation for the development of the Guardians robot within the Guardians project was to overcome these limitations so as to provide the consortium with a robot platform where to test the project results in a realistic environment.

In a very first stage, Erratic and Khepera robot platforms were selected to test the project developments, while a new Guardians robot was thought to provide the consortium partners the needed features to test them in real world.

Some of the new features that the Guardians robot offers to the project are listed below:

- High mobility, in order to avoid obstacles.
- High speed.
- High power.
- Climbing stairs capability.
- Dedicated space for special firefighting tools carriage.
- Open source software.
- Player/Stage software architecture.

These capabilities provide a large number of extra experiment possibilities, as:

- Following a firefighter in debris.
- Carrying an additional breathing apparatus for a trapped civilian.
- Taking thermal imaging equipment.
- Offering a controllable water jet and monitor.

The Annex of this deliverable shows some robots that are similar to the Guardians robot and that are used for similar application concepts.

3 System Architecture

3.1 Software Architecture

This part presents the description of the software architecture, as well as some ideas about its behavior.

First of all this part shows an introduction of the Player/Stage architecture (open source), the architectures used for the robot start up. The second part describes the different robot software components.

3.2 Player/Stage Architecture

Player is a network server for robot and sensor control

Some features of Player:

- Permits access to robots and sensors via a network.
- Includes device developed drivers used in the robot.
- Libraries for C, C+, Java, Lisp and Python.
- Free software.

Player provides a clean and simple interface to the robot's sensors and actuators over the TCP/IP network. The client program talks to Player over a TCP socket, reading data from sensors, writing commands to actuators, and configuring devices on the fly.

Stage is a 2D multiple-robot simulator. Stage simulates a population of mobile robots, sensors and objects in a two-dimensional bitmapped environment. Stage is designed to support research into multi-agent autonomous systems, so it provides fairly simple, computationally cheap models of lots of devices rather than attempting to emulate any device with great fidelity.

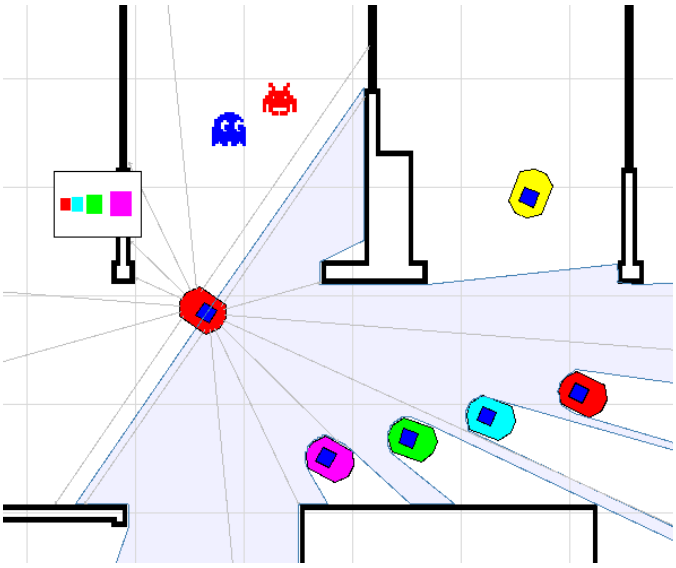


Figure 1: STAGE Interface example

The main features of Stage are:

- Is used as a Player plug-in module, in other words, as a device driver.
- Uses robot and sensor drivers as clients of Player.
- It provides a big variety of robots, sensors and different actuators.
- Allows the importing of complex simulation environments from image files.
- Quite good real simulator.

3.2.1 Player Devices

Player defines a set of standard interfaces, each of which is a specification of the ways that you can interact with some class of devices. For example the position2d interface covers ground-based mobile robots, allowing them to accept commands to make them move (either velocity or position targets) and to report their state (current velocity and position). Many drivers support the position2d interface, including p2os, obot, and rflex, each of which controls a different kind of robot. The job of the driver is to make the robot support the standard interface. This way, Player control code that works with one robot will work on another robot.

```
Player is similar to the way standard files are accessed with
functions"
```

The way of accessing devices in Player is similar to the way standard files are accessed with functions like open, close, read or write.

Guardians features:

- Formed by several devices.
- It is permitted to have multiple requests for the same device.
- There isn't any locking between the different requests.

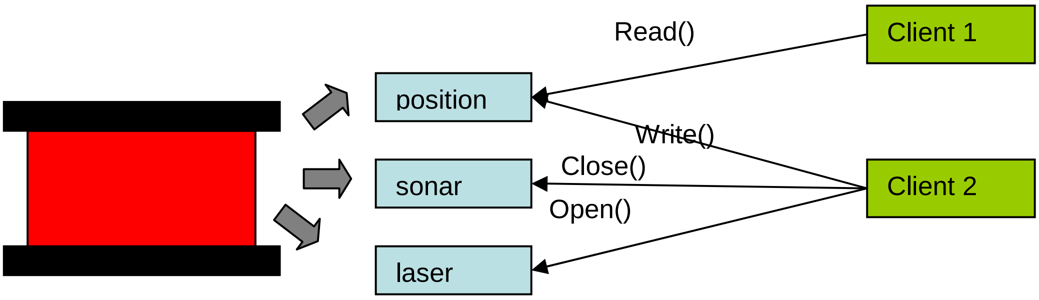


Figure 2: Player device access example

3.2.2 Interfaces vs. Drivers

The Player interfaces specify what is or is not possible to do in a driver developed for a device, defining common syntax and semantics for the exchange of messages.

Features:

- Offer support for different types of hardware. There is also the possibility to define custom interfaces, although it is not recommended unless it is important to keep the cohesion and the standardization of the interfaces.
- Specify:
    - Data format.
    - Commands.
    - Configuration.
- Define how to interact with the driver.

Example:

As commented before, the position2d interface covers ground-based mobile robots, allowing them to accept commands to make them move (either velocity or position targets) and to report their state (current velocity and position)

On the other hand, a driver in Player specifies and implements at a low level, how the interface functions perform.

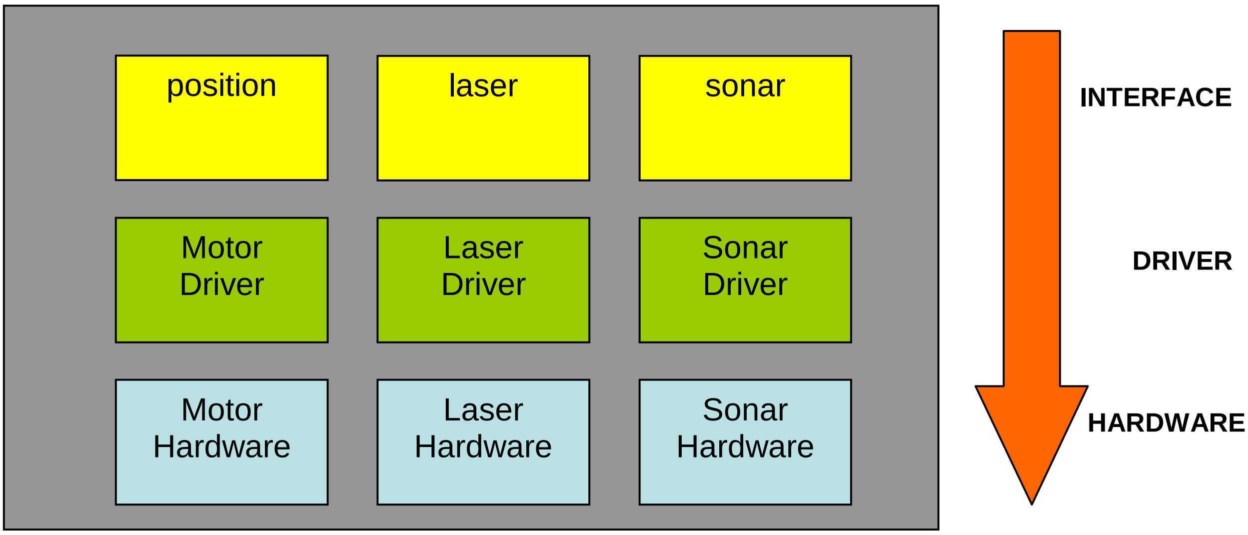


Figure 3: Player driver general scheme

There is more information about the Player/Stage architecture in two deliverables of the workpackage 4 (D4.1-2: "Preliminary System Architectural Design and Software Architecture deliverable" and D4.3: "Software architecture design for a linux environment")

3.3 Guardians Architecture

As commented above, the control software of the Guardians robot is based in the Player control architecture, so it is possible to access and modify its state, using the existing Player libraries.

3.3.1 Player Interfaces Implemented

The Guardians robot implements the following interfaces:

- **Position2d**: is used to control mobile robot bases in 2D.
- **Laser**: provides access to laser range sensor, such as a SICK or Hokuyo laser range-finder.
- **Imu**: provides access to an Inertial Measurement Unit sensor.
- **Power**: provides access to the robot power subsystem.
- **Aio**: provides access to an analog I/O device.

- **Dio**: provides access to a digital I/O device.
- **Gps**: provides access to an absolute position system, such as GPS.
- **Opaque**: allows you to send user-specified messages. With this interface a user can send custom commands to their drivers/plugins. See examples/ plugins/ opaquedriver for an example of using this interface in a plug-in.

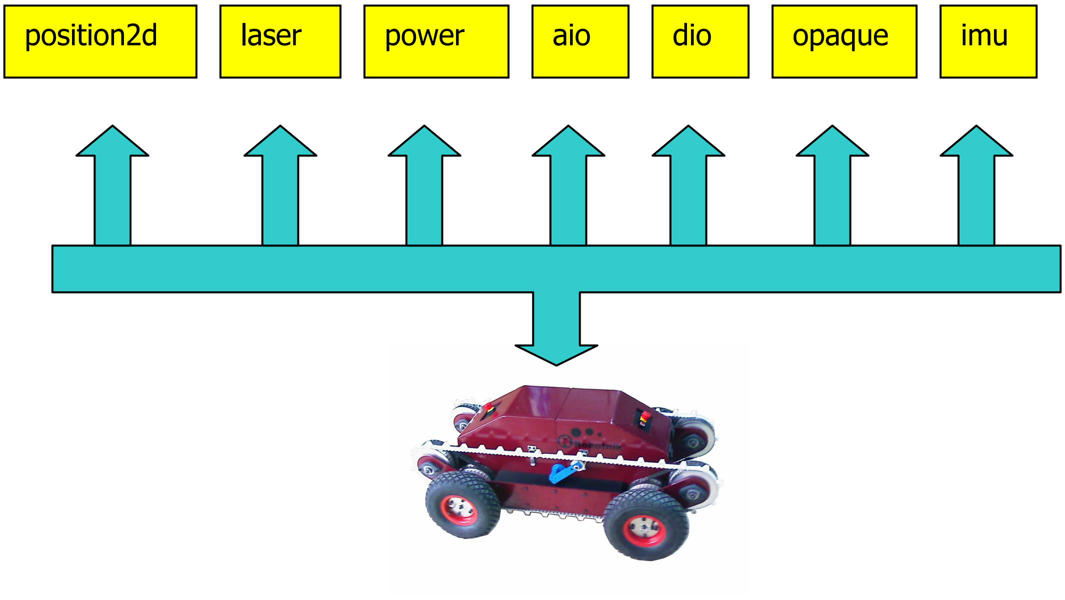


Figure 4: Guardians implemented Player Interfaces

3.3.2 Software Components

In what follows there is a description of the components that are part of the robot control software.

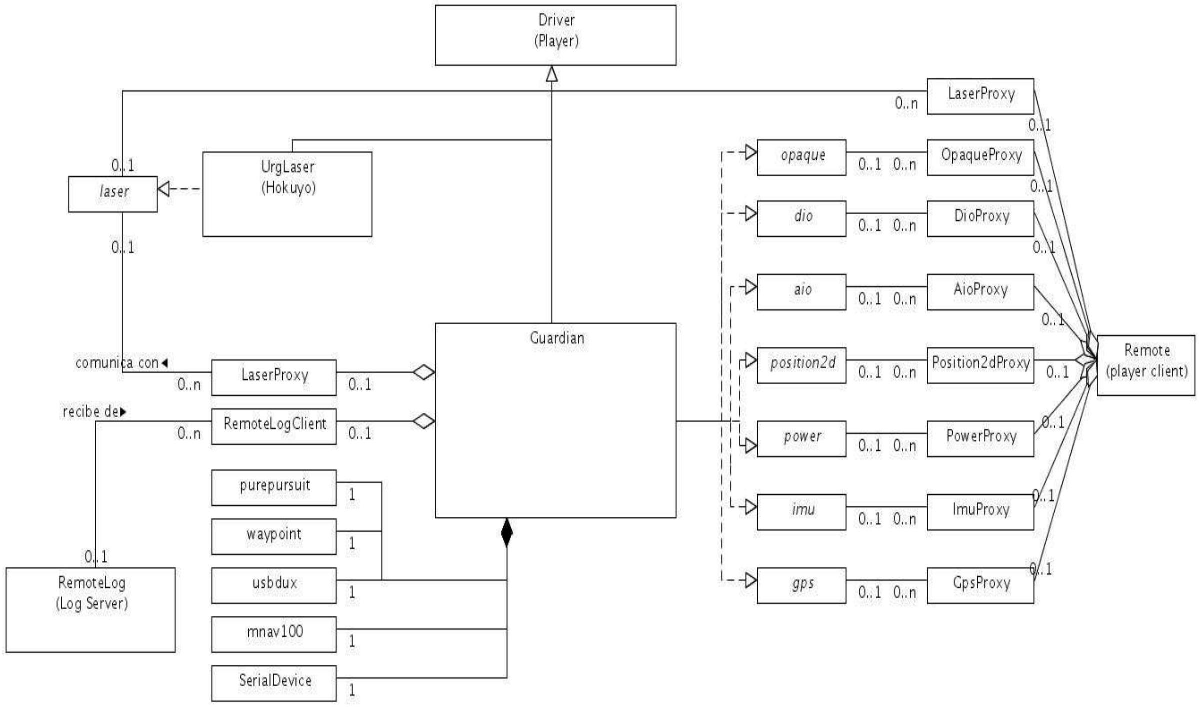


Figure 5: Guardians robot software components

- Guardians:
    - Robot principal class.
    - Forms the Guardians driver/plug-in for Player.
    - Manage the correct system operation.
    - Implements the interfaces and manages the commands and requests that arrives via Player.
- UrgLaser:
    - Player driver for the Hokuyo laser rangefinder.
    - Implements the laser interface.
- Remote:
    - This client connects to the robot and/or the laser, from the same machine or from a remote one.
    - It can be connected to one of the different interfaces implemented in the robot and the laser.
- LaserProxy:
    - Object for the communication to the laser sensor.
    - It can be used from the Guardian driver, as well as from a remote client.
- OpaqueProxy:

- Object for the communication with the opaque interface of the robot.

- DioProxy:
  - Object for the communication with the dio interface of the robot.
- AioProxy:
  - Object for the communication with the aio interface of the robot.
- Position2dProxy:
  - Object for the communication with the position2d interface of the robot.
- PowerProxy:
  - Object for the communication with the power interface of the robot.
- ImuProxy:
  - Object for the communication with the imu interface of the robot.
- GpsProxy:
  - Object for the communication with the gps interface of the robot.
- RemoteLog:
  - TCP/IP Server that saves the robot and the client logs.
  - This command is executed in the machine by default, but it can be executed from other machines in the same network.
- RemoteLogClient:
  - Object to register events in the log server.
- SerialDevice:
  - Object that allows opening and configuring a serial device.
  - It is used for the communication with the AX3500 control card, which is connected to the motors.
- Chemical:
  - Object that controls the chemical sensor
- Waypoint:
  - Object that stores the point that the robot needs to follow in a route.
- Purepursuit:
  - Object that implements the pure pursuit algorithm for following a trajectory.

3.3.3 Files Structure

The files and programs of the Guardians robot software are organized inside "usr/local/src/Guardians" in the following way:

- Src:
  - Control program source files
  - Guardians.cc:
    - *Robot main class*
  - remote.cc:
    - *Example program that acts like a robot client.*
  - joystick.cc:
    - *Class for reading usb joystick.*
  - SerialDevice.cc:
    - *Class that manages the communication through a TCP/IP socket.*
  - SocketClient.cc:
    - *Class for the communication through a TCP/IP socket.*
  - waypoint.cc:
    - *Class for the management and the operation of route points.*
  - purepursuit.cc:
    - *Class that implements the pure pursuit algorithm to follow trajectories.*
  - RemoteLogClient.cc:
    - *Communication interface with the robot log Server.*
- Include:
  - Control program headline file.
- Bin:
  - Executables.
- Build:
  - Program compiled files.
- Docs:
  - Generated documentation.
- Lib:
  - Used libraries.
- Remotelog:
  - Source and programs of the remote log server executed in the robot.

- Makefile:
  - Configuration files for the program compilation.
- Guardians.cfg:
  - Player configuration file, in which robot parameters and available interfaces are declared.
- doxy_Guardians:
  - Doxygen documentation automatic generation program configuration file.

4 System Installation and Configuration

4.1 Guardians Robot control (Control PC)

4.1.1 System Hardware and Requirements

For the Guardians software installation you need the following elements:

- Xubuntu-8.01.4-i386 adjustable distribution.

The Guardians control software is designed to be updated and migrated to the latest Pc processors. The installation process that is detailed below is designed for the hardware supplied with the platform.

The supplied PC hardware includes the following devices:

- CPU:
    - Main board: CV700A-3R10C.
    - Chipset: VIA CN700 / VT8237R Plus.
    - Processor: VIA C7 1GHz.
    - Memory: 1 bank DDR2 1GB.
    - Graphic card: VGA CN700 1600x1200. 64MB shared. Sub-D 15pin.
    - Network card: RJ45 Realtek 8100C 10/100Mb - 3 ports RJ 45.
    - Ports:
        - *1 port IDE for HD 2.5".*
        - *1 port SATAI.*
        - *1 port 50pin for Compact Flash E/S - 3 USB Tipo-A.*
        - *1 parallel EPP/ECP.*
        - *2 Serial 9 pin RS-232.*
        - *2 PS/2 Keyboard + Mouse.*
        - *Audio AC97 2.1, Full Duplex. Output and microphone (Jack 3.5mm).*

Figure 6: Computer

- Joystick: Logitech Extreme 3D Pro.

Figure 7: Joystick

4.1.2 Operative System Installation

For a correct installation of the software next steps must be followed:

- Install the xubutun-Guardians-8.01.4 distribution following the steps of the software.
- Necessary extra package for the robot (included in the distribution by default):
  - Ssh.
  - c++.
  - nfs-common.
  - nfs-kernel-server.
  - pkg-config.
  - Doxygen.
  - doxygen-gui.
  - Flex.
  - Bison.
  - libncurses5-dev.
  - libltdl3-dev.

4.1.3 Network Configuration

4.1.3.1 Ethernet

Configure the network device connected to the wifi router. It is initially configured with the fixed IP 192.168.2.11

- Guardians > ifconfig eth0 up 192.168.2.11.

4.1.3.2 Wireless

The wifi router configuration by default is the following:

- IP in 192.168.2.1.
- User/password: admin/Robotnik2008.
- Channel 9.
- Encrypt WEP 128 bits (abf35678c1abf35678c165da12).
- DHCP unavailable.

4.1.3.3 IP Camera

The configuration by default of the IP camera is the following:

- IP fixed in 192.168.2.33.
- User/password:
  - root: Robotnik2008
  - Guardians: Guardians

4.1.4 Player Installation

In the installed distribution the default version of Player is 2.1.2, that is by default in the directory /usr/local/bin

In case you need to recompile, the best is the following way:

- ./configure –disable-alldriver –enable-urglaser (only install the laser Hokuyo driver).

- make install

- Add to file .bashrc of /root the following line:
    - export LD_LIBRARY_PATH=/usr/local/lib

4.1.5 Guardians Driver Compilation and Execution

When the player libraries have been installed, the robot driver can be compiled and executed. To do this, the "make" script should be executed from the project directory. For running the control program, use "player guardian.cfg".

4.1.6 NFS Configuration

For modifying the source file of the robot driver in a remote way you need to enable access to the project folder through the NFS.

Steps to follow to configure and access through NFS, the Guardians robot directory.

Necessary packages:

- nfs-common.
- nfs-kernel-server.
- Portmap.

In the configuration folder */etc/exports* add the directory to share:

- /usr/local/src/Guardians/ 192.168.2.0/24 (rw).

In the configuration folder */etc/hosts.allow*, add the IP direction of the remote stations that will connect to the robot:

- ALL: 192.168.2.0/255.255.255.0

4.1.7 Scripts and Start Configuration

For the start up and the execution of the robot control programs, you need to configure and add some scripts:

4.1.7.1 autologin

Script that orders to automatic logging in a computer terminal:

- #!/bin/sh
- exec 0</dev/$1 1>/dev/$1 2>&1
- cat /etc/issue
- shift
- exec $*

Copy this script in the directory /sbin/, with execution permission (*chmod+x*)

4.1.7.2 event.d/ttyX

In the directory /etc/event.d/ there is a configuration script for each of the computer terminals. You must modify the scripts corresponding to tty2 and tty3 adding/changing the following lines:

- exec /sbin/getty 38400 tty2

Change for

- exec /sbin/autologin tty2 login -f user

In which *user* is the user name, with which you would log in later.

4.1.7.3 bashrc

This is a start script, which every user has in their root directory. Edit this file for the user used in the previous script *ttyX.* So, for example, if you have a user Guardians, you need to edit the file /home/Guardians/.bashrc.

You need to add the following code at the end of the file:

- echo "ROBOTNIK GUARDIANS"
- Terminal=`tty`
- case $Terminal in
- "/dev/tty3") cd /usr/local/src/Guardians/RemoteLog;
- ./bin/RemoteLog 15560 ./log Guardians.log overwrite;; #Log server
- "/dev/tty2") cd /usr/local/src/Guardians/;
- sleep 2;

- player ./Guardians.cfg;; #runs player Server in terminal 2
- esac

4.2 Remote PC

4.2.1 Necessary Software

For executing a robot client program from a remote pc, it is needed to have installed Player and configure a cordless connection with the parameters of the previous section.

4.2.2 NFS

Necessary packages:

- nfs-common.

Remote unit montage:

Example*: mount -t nfs 192.168.2.11:/usr/local/src/Guardians /mnt/nfs*

5 Power and Control

5.1 Introduction

This section shows information about the internal robot communication lines, the power supply drawings, connections on the terminal board and main components used.

5.2 Communication Diagram

The following figure shows the communication diagram existing inside the robot.

The functionality of the system can be further extended by using the two free Ethernet ports and the existing free USB port.

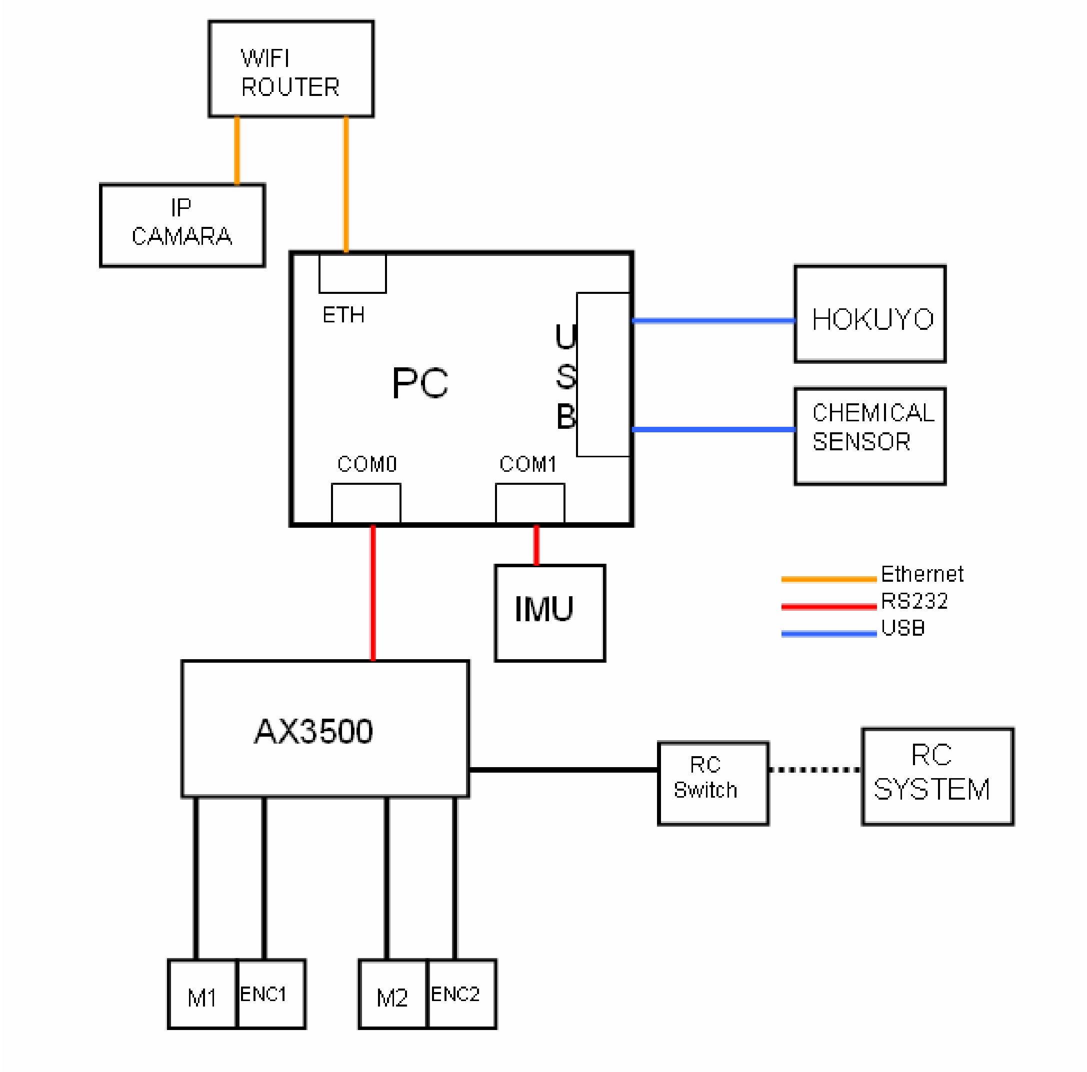


Figure 8: Communications diagram

5.3 Power Supply Drawing

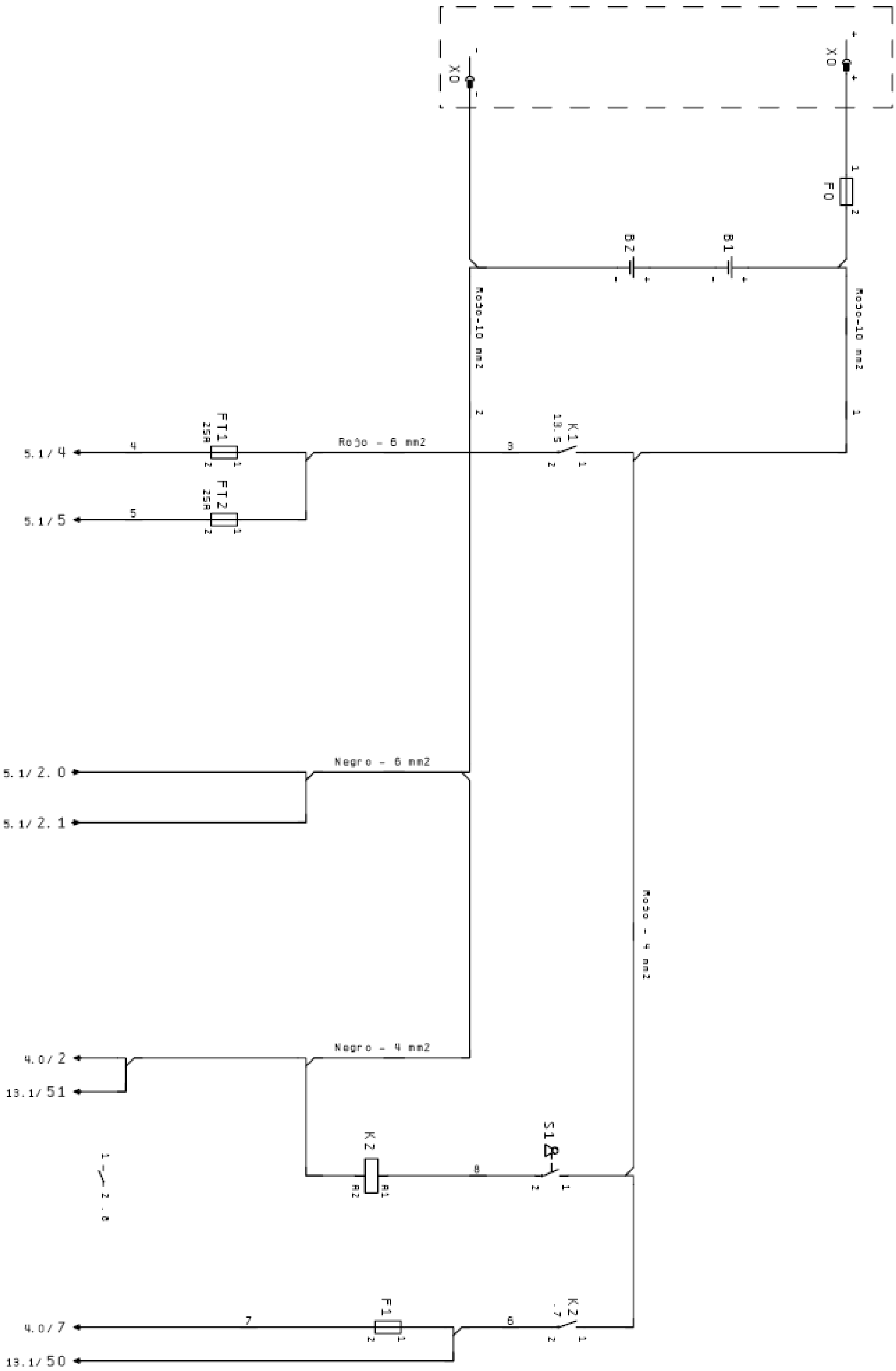


Figure 9: Power supply drawing

The robot receives the power supply from two batteries (B1 and B2) of 12V each. They are placed in serial connection (24V) and have a full capacity of 33Ah that supply all the components.

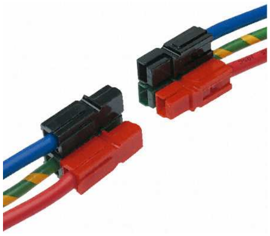


Figure 10: Power supply connector

Figure 11: General ON/OFF switch

Figure12: Emergency button

Behind the supply connector the current is divided into two circuits: direct supply and supervised supply. The supervised supply goes through the security contactor K1 that opens the circuit and cuts the supply when the emergency button is pushed. This circuit supplies power to the motor driver.

The direct supply circuit is open when the general switch S1 is ON. Through this circuit, the control DC/DC converter that powers the different control devices is supplied. For security reasons, when the S1 is OFF, the supervised supply is disconnected too.

5.4 Internal Terminal Block

The following figures show the internal terminal board.

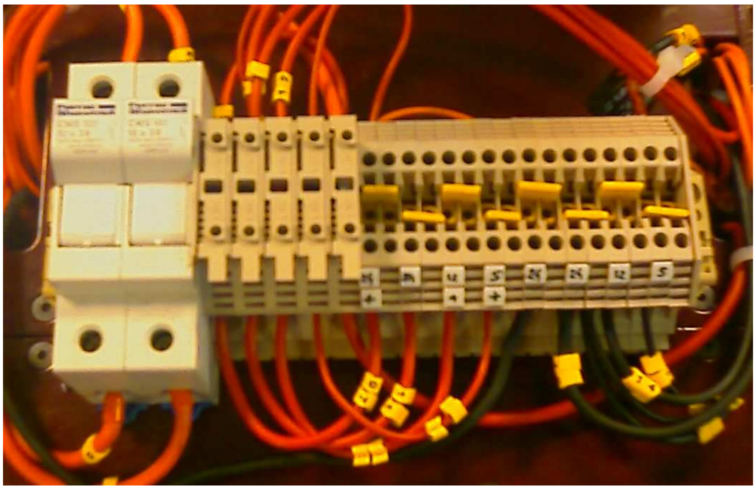


Figura 13.Direct Power Terminals

To the right of the picture you can see the direct supply terminal board, to which you can connect the elements that run with 24V direct from the batteries, 24V from the dc/dc converter, 12V and 5V in the following way:

	+24V	+24V	+12V	+5V
Origin of Power	Direct from Batteries	From DC/DC Converter	From DC/DC Converter	From DC/DC Converter
Components			Computer WiFi	Laser

Table 1: Component connection to the direct power supply terminal board

At the left of the Picture you can see a terminal board with the following fusels:

- Fuse F1: protection for the input circuit components without the possibility to run when the batteries are charged. This voltage is direct from the batteries.
- Fuse F2: protection for the output circuit of the DC/DC converters (+24V).
- Fuse F3: protection for the output circuit of the DC/DC converters (+12V).
- Fuse F4: protection for the output circuit of the DC/DC converters (+5V).
- Fuse F5: protection for the input circuit of the main DC/DC converters (+24V).

The supervised supply circuit (to the left of the figure 13) consists of two 20 A fuses, one for each motor of the robot.

The following picture shows the security circuit, which is composed of the auxiliary contact of the security contactor, the contacts of the emergency button and a button for the circuit reset. When the emergency button is pushed the power is cut and you need to pull the mushroom and push the reset button to re-establish the power.

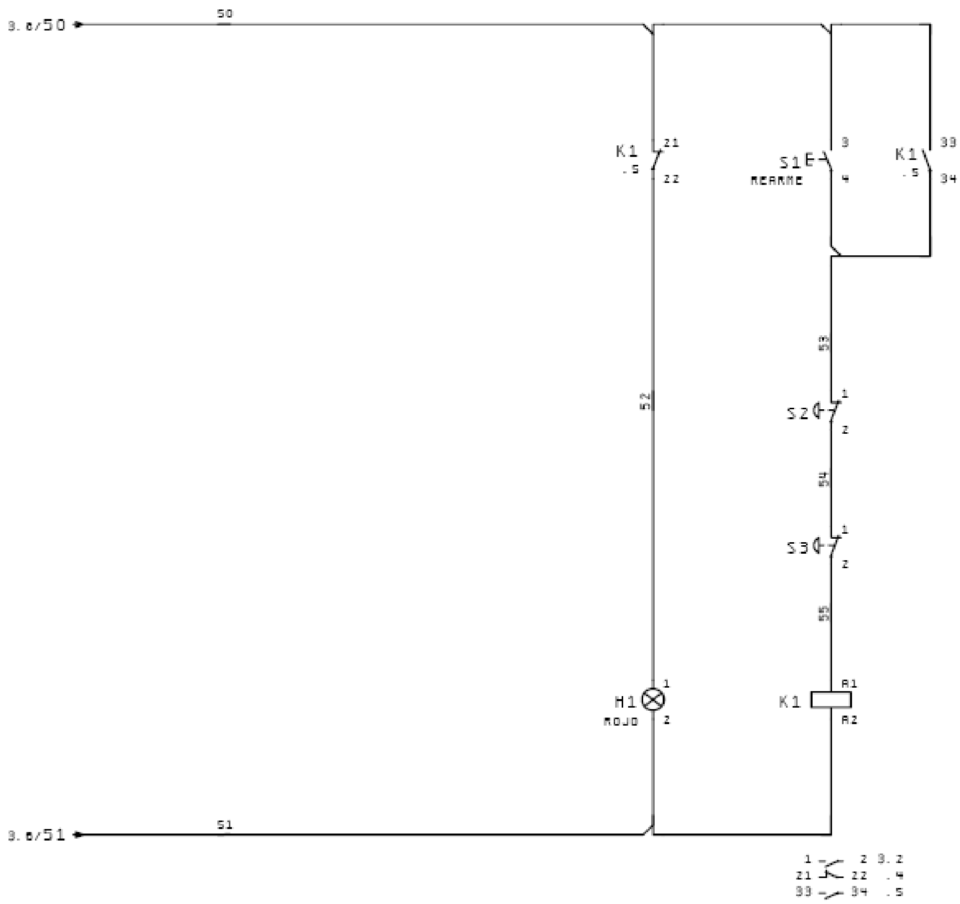


Figure 14: Security drawing

5.5 Guardians Robot Components

A list of the main components of the robot is show in the following table:

COMPONENT	UNIT	COMPANY
DC/DC Converter (24V-24V)	1	Sevcon
DC-DC Converter (24V-12V)	1	Alfatronix
DC-DC Converter (24V-5V)	1	RS
CPU	1	Lex
Batteries 12 V	2	LIFELINE
Inertial Measurement Unit	1	Crossbow
Laser Range Finder	1	HOKUYO
Camera IP	1	Logitech
Encoder	2	Leine & Linde
Motor Control	1	Roboteq
RC transmitter	1	Futaba
RC receiver	1	Futaba
RC Switch	1	Team Delta Engineering

Table 2: Main components of the Guardians robot

6 System Start-Up

6.1 Hardware

6.1.1 Electrical System

6.1.1.1 Main Power Supply

The robot receives the main power supply from two 12V batteries in series that give a complete power supply voltage of 24V to the system. With this set of batteries the robot is able to operate up to 8 hours, depending on the robot movements.

The batteries are connected directly to the robot through the security contactor (K1) and the control contactor (K2). For charging the batteries there is a connector in the back of the robot where the charger can be connected. It is possible to charge the robot with the control connector open, but it's recommendable to cut the power to the motor pushing the security button.

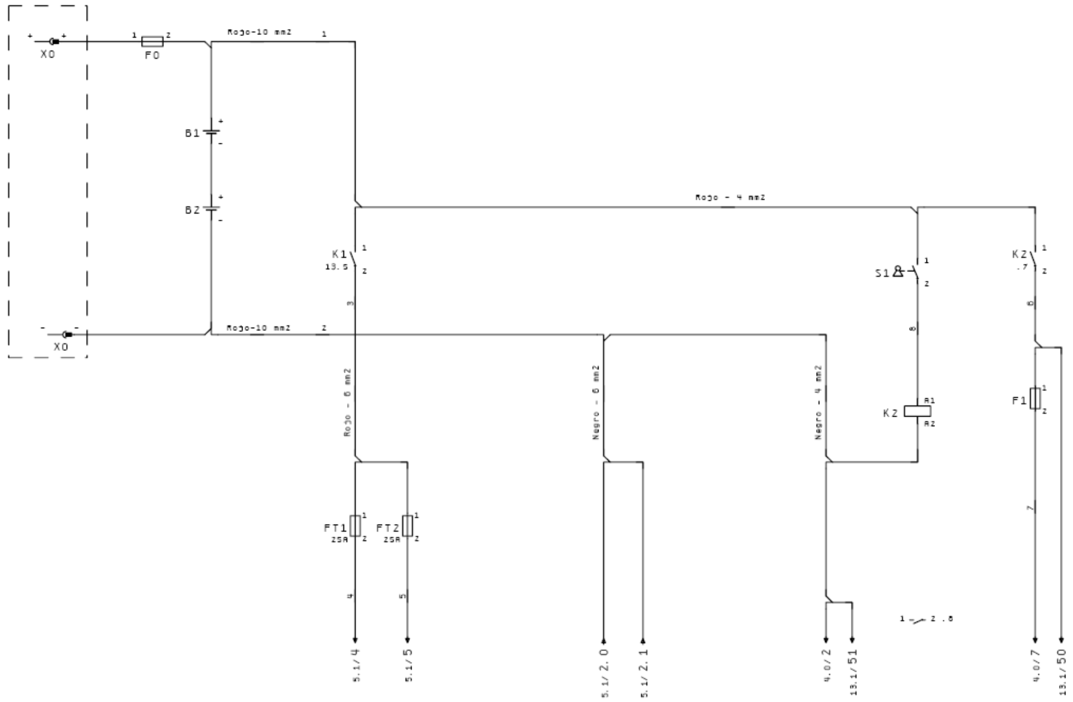


Figure 15: Power supply drawing

Charging time is around 3 hours.

6.1.1.2 Control Panel

The robot presents several buttons and indicators on its back cover with the following functions:

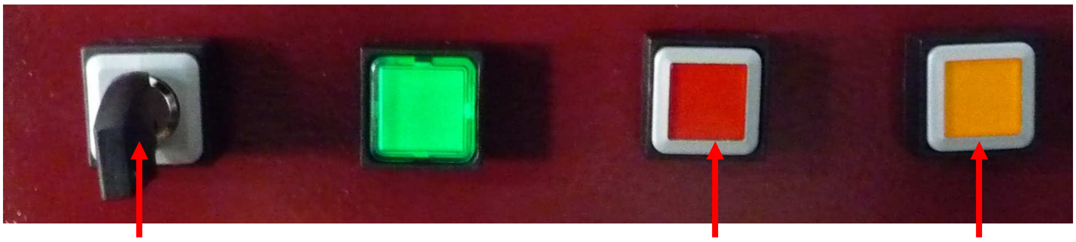


Figure 16: Control panel

- The general ON/OFF switch cuts the power of the whole robot.
- The On indicator/switch indicates that the security system has been activated, so motors can not be moved again until it is restarted. It is also the restarting switch.
- The control indicator/switch indicates that the system is being operated under control (manual or remote). It also switches on the control PC.

6.1.1.3 Start-up Sequence

The general ON/OFF switch must be activated for giving energy to all the elements of the system. When ON, the light of the stop indicator will illuminate, which means that it is necessary to push the restart button. Once pushed, it is necessary to turn ON the computer, so the Control button will be pushed. At this moment the PC starts-up and loads all the necessary files for booting.

After booting, it is possible to connect to the system in a remote way or connect to the robot manually.

6.1.1.4 Emergency Stop

The Guardians robot has an emergency stop system that cuts the power of every mobile element that might cause any damage to a user or to the robot environment. To activate the security system it is needed to push the emergency button located in the back cover.

Figure 17: Emergency stop

It is recommended to keep the emergency stop pressed when any maintenance operation is to be done on the robot (camera adjusting, new component installing, etc) or when there is the risk of any danger from a non controlled robot movement.

For exiting from the emergency stop mode, the following steps have to be carried out:

- Check if the danger situation that caused the emergency stop has disappeared.
- Pull the emergency stop button.
- Push the restart button situated in the back panel.
- Push the PC start button to start-up the control PC.

7 Mechanical Components and Maintenance

7.1 Introduction

This section describes the main parts of the Guardians mobile robot, as well as how they are assembled. Every main piece includes a little description of the mechanical component that composes it, emphasising the elements that need a special periodical control and maintenance.

The location and maintenance of other non mechanical components is also described below.

Finally, a summary of the basic drawings of the vehicle and the main components has been included.

7.2 Components Representation

The next figure shows the four main parts of the robot: lateral drive sides, main chassis and the upper cover.

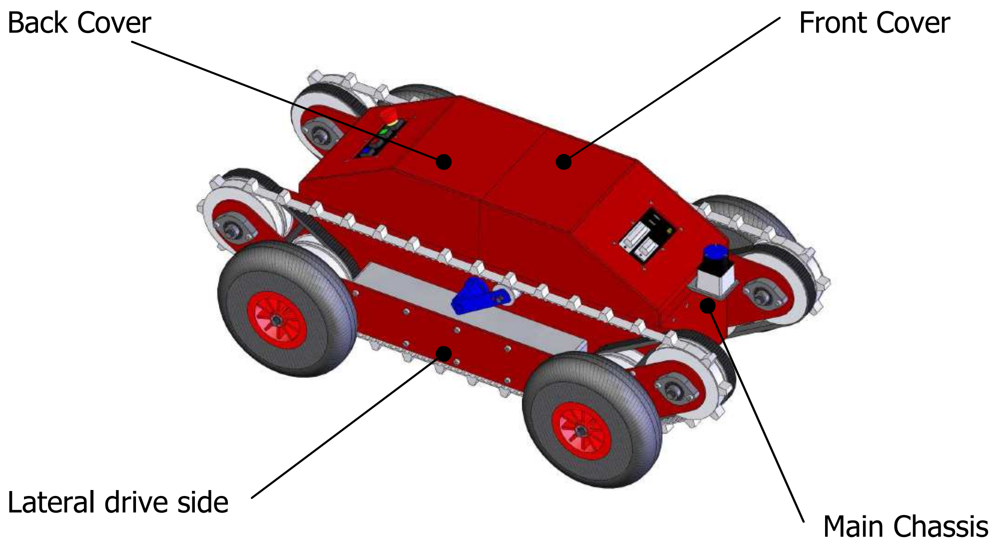


Figure 18: Main parts of Guardians robot

The four main parts that form the robot are:

- Main Chassis: Inside it are placed the electrical components.
- Laterals drive sides: where components for driving the robot are placed.
- Covers: Allow access to the interior of the robot where some of the control components like the driver AX3500 or the control computer are placed.

The lateral drive sides are the parts of the robot that contain the largest quantity of mechanical components that might need some maintenance.

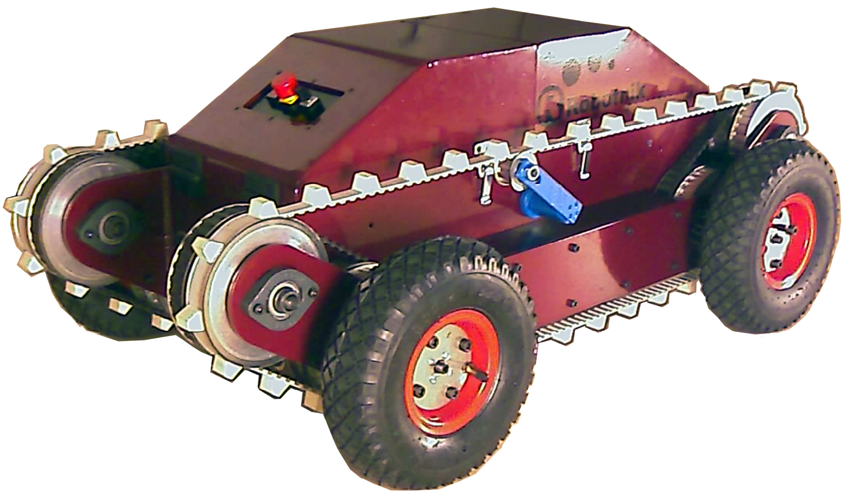


Figure 19: Guardians robot

7.2.1 Traction Laterals

The main components that form the lateral drive sides are shown in the following figure.

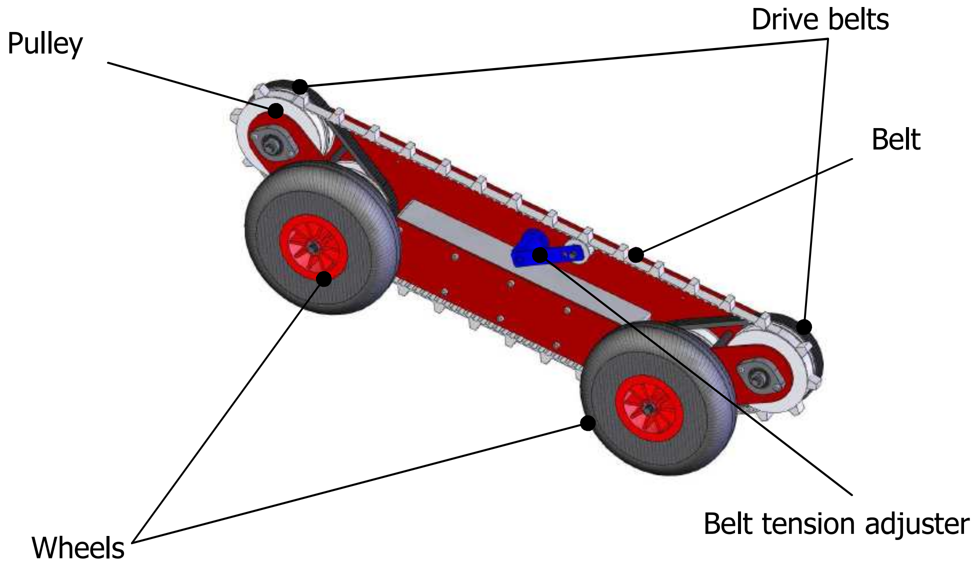


Figure 20: Lateral Drive side

- Pulley: transmits the movement of the belts to the axis where they are fixed. It is necessary to check the state of the pulleys once a year.
- Drive Belts: transmit drive from the motors. It is necessary to check the state of wear once a year.
- Belts: These belts have two functions. First they transmit drive between the front and rear pulleys. Second they make use of their external cleats to help to avoid obstacles and climb stairs. It is necessary to check the level of debilitation and the consistency of the cleats at a minimum once a year.
- Belt tension adjusters: These elements keep the tension of the belts. It is necessary to check the belt tension every 6 month.
- Wheels: the wheels make the transmission of the robot. It is necessary to check the pressure of the wheel once a month.

Figure 21: Lateral view of the Guardians robot

7.3 Other Non-Mechanical Components

The following figure shows other non-mechanical components to be considered for maintenance tasks.

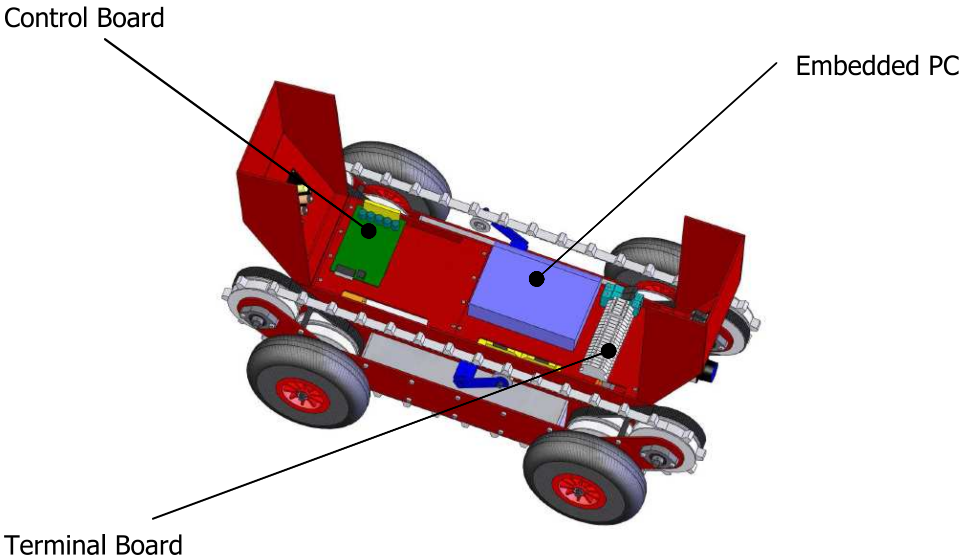


Figure 22: Non-mechanical Guardians components

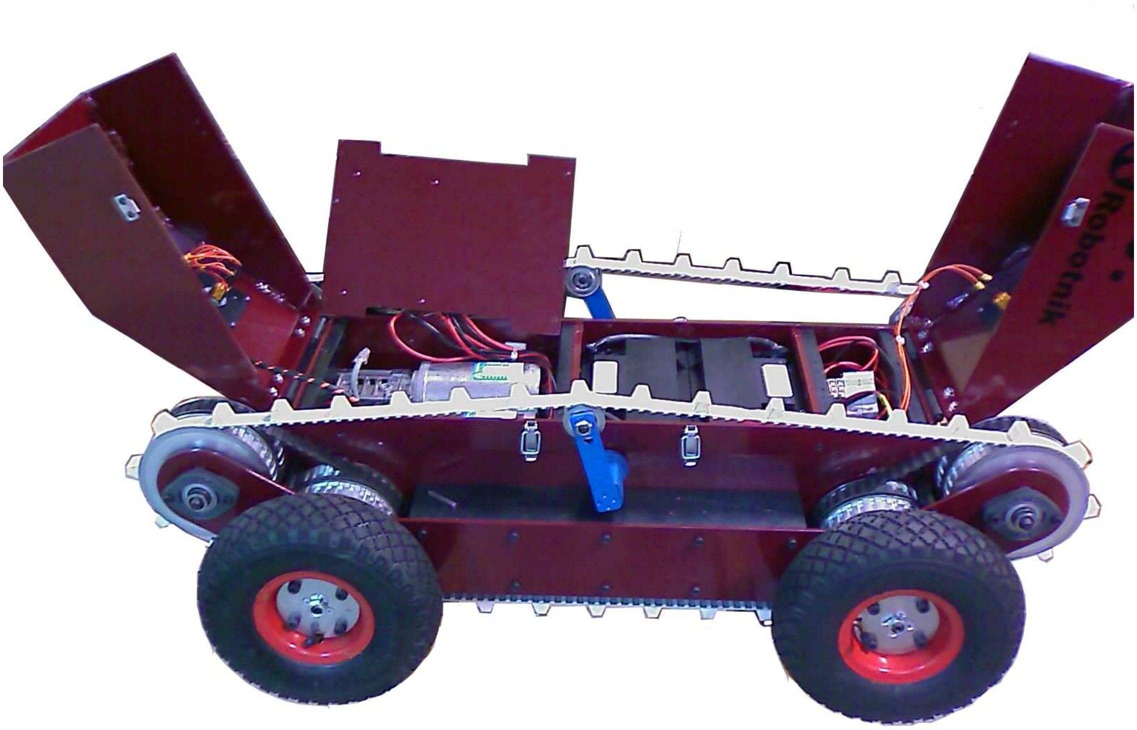


Figure 23: Guardians inside view

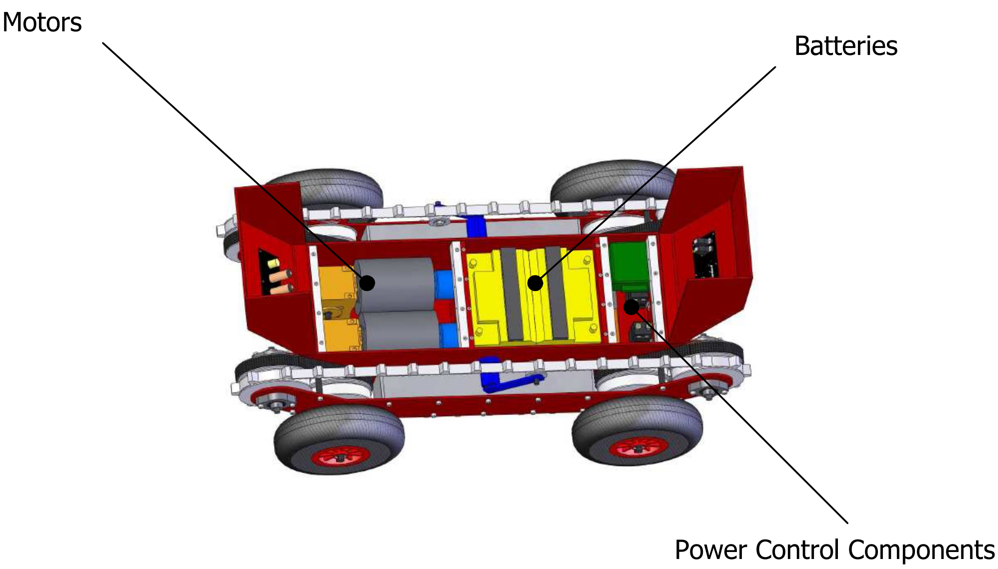


Figure 24: Other Non-mechanical components inside Guardians

7.3.1 Embedded PC

The embedded control PC is located in the middle of the robot and is fixed to the battery cover by means of four silent-blocks. Its maintenance is equivalent to a standard PC station.

The main problem can be due to the accumulation of dust in the internal components, so it acts as thermal insulator. The heat generated by the components cannot be well dissipated because it is trapped in the dust layer.

The oil and grease particles contained in the environmental air mix with the dust, Guardians thus creating a big insulation layer that reflects the heat to other components. This effect causes a reduction of the system useful life. On the other hand, the dust contains conductive particles that can generate short-circuits throughout the circuit boards or the peripheral cards.

The best way to extend the life of the equipment and make it free of reparations for many years is to clean it and remove the dust frequently.

7.3.2 Batteries

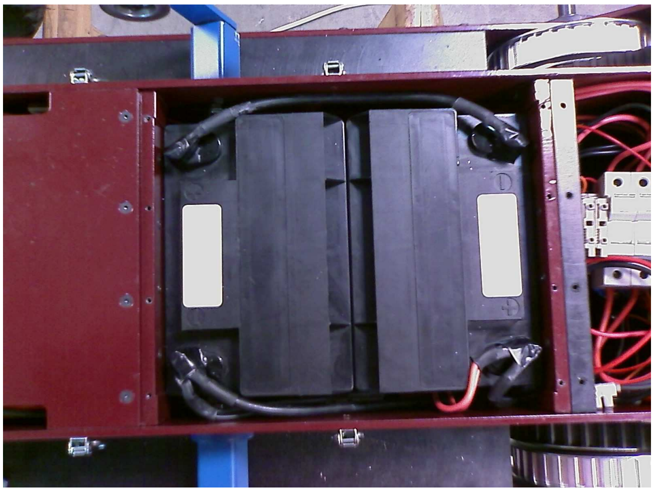


Figure 25: Batteries location

The batteries must be kept clean and dry in order to avoid escape currents. The cleaning of the batteries must be done following the instructions of the datasheet ZVEI “Cleaning Batteries”. The plastic parts of the batteries, especially the cell container, must be cleaned only with water, without additives.

At least once every 6 month it is needed to measure and register:

- Batteries voltage.
- Batteries location temperature.

Once a year it is needed to measure and register:

- Batteries voltage.
- Surface batteries temperature.
- Batteries location temperature.

Once a year it is needed to make visual controls of:

- Screw connections.
- Batteries location.
- Air flow, ventilation.

7.4 Summary

In the previous parts some main parts of the vehicle and components that need maintenance or only supervision have been mentioned. The following table summarises all the elements that need maintenance and the periodicity of this maintenance.

	Every 6 months	Every 12 months	Observations
Bearings			If needed, it is better to replace it with a new one
Screws		Ensure are not loose	
Drive Belts		Visual control of the debilitation level	In case of cracks, replace it with a new one.
Belts (With cleats)		Visual control of the debilitation level	In case of cracks, replace it with a new one.
Wheels	Pressure control		
Belt tension adjuster	Check the tension of the belt		
PC		Clean the interior and visual control of the correct work of the fan	
Batteries	Batteries Voltage Surface Temperature Location Temperature	Clean outside with water. Visual control of the screw connections	

Table 3: Maintenance summary

7.5 Basic Drawing

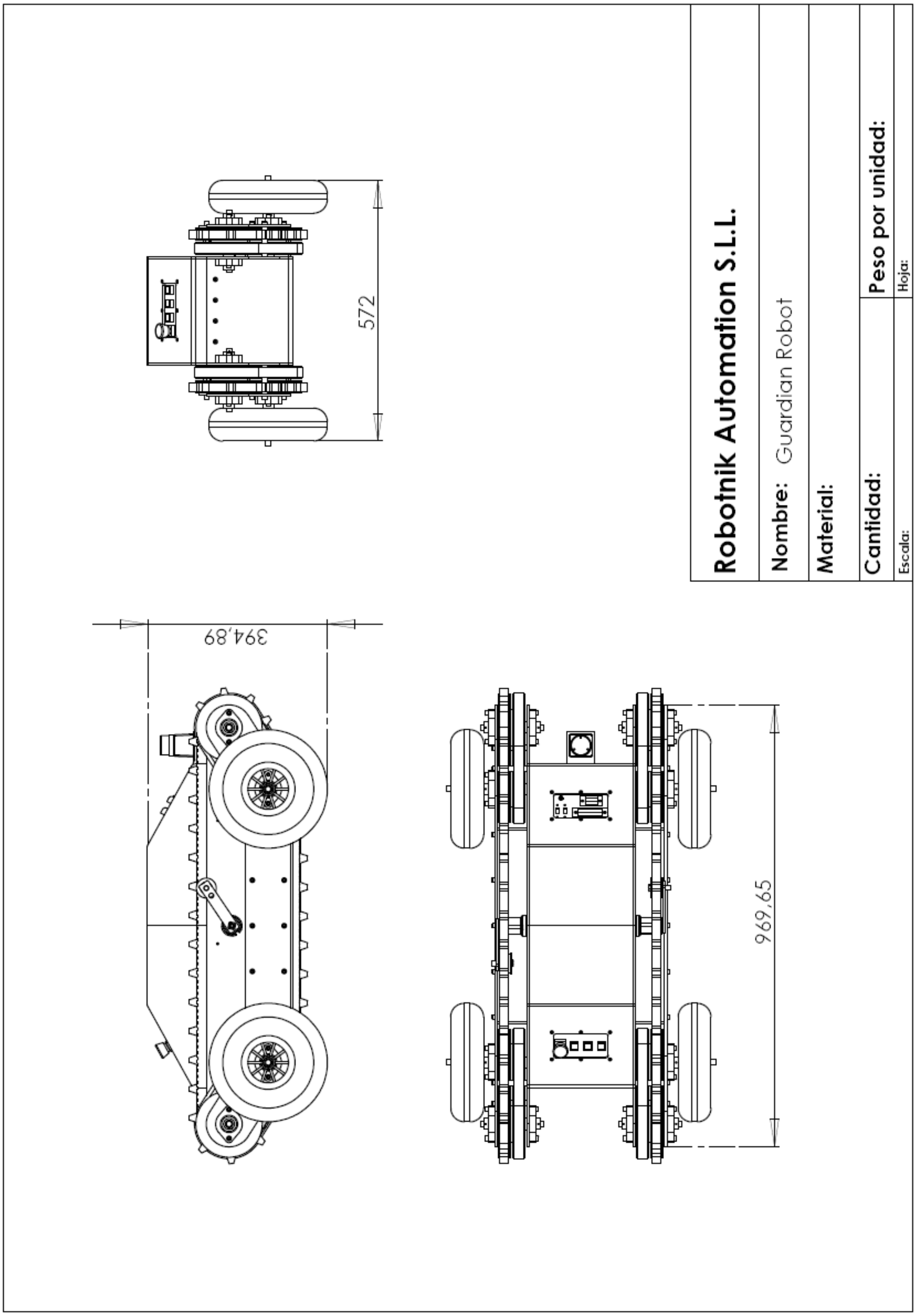


Figure 26: Basic Drawing

8 Electrical Drawings

8.1 Main Supply

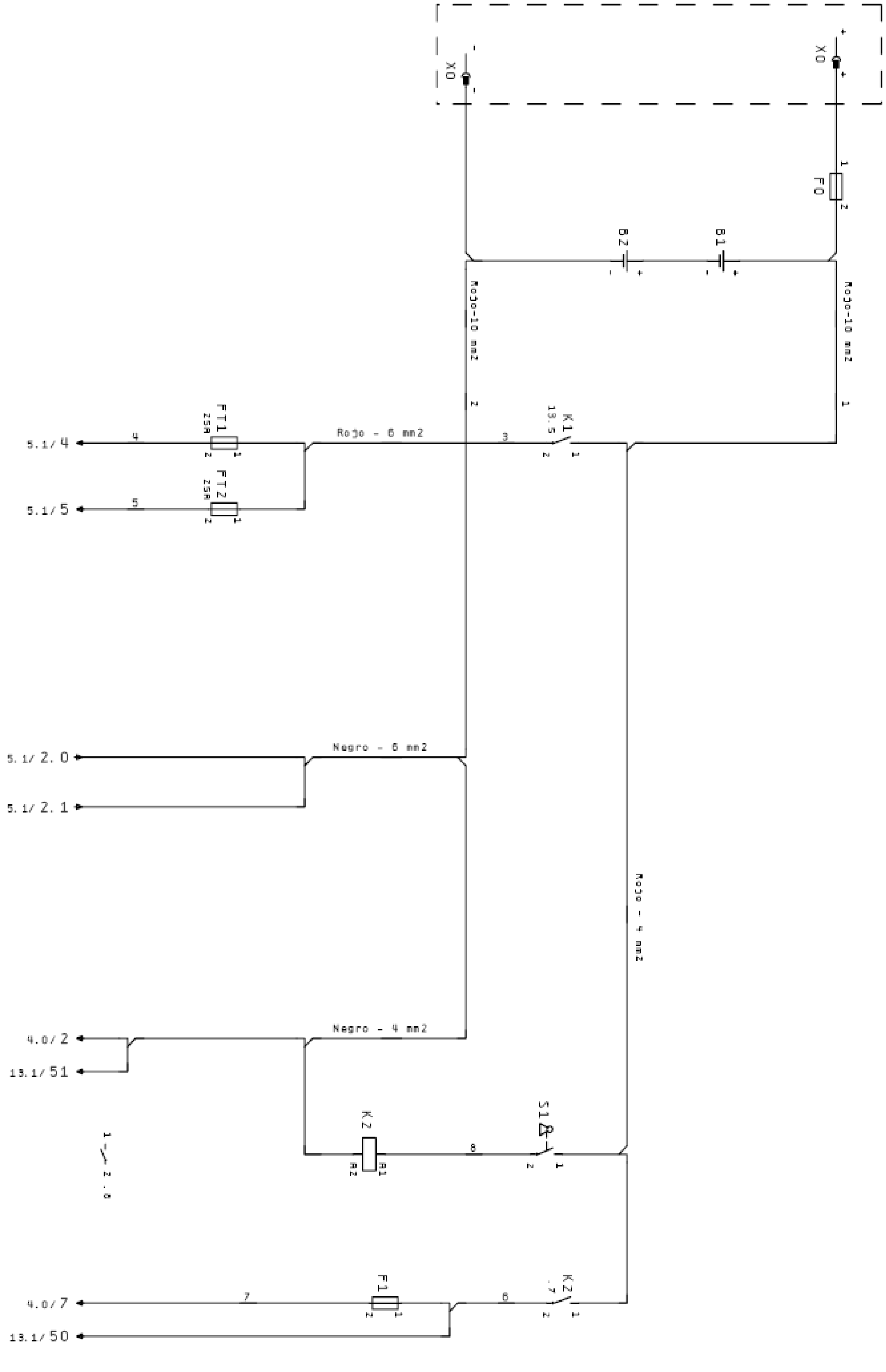


Figure 27: Main Supply Drawing

8.2 DC/DC Converters

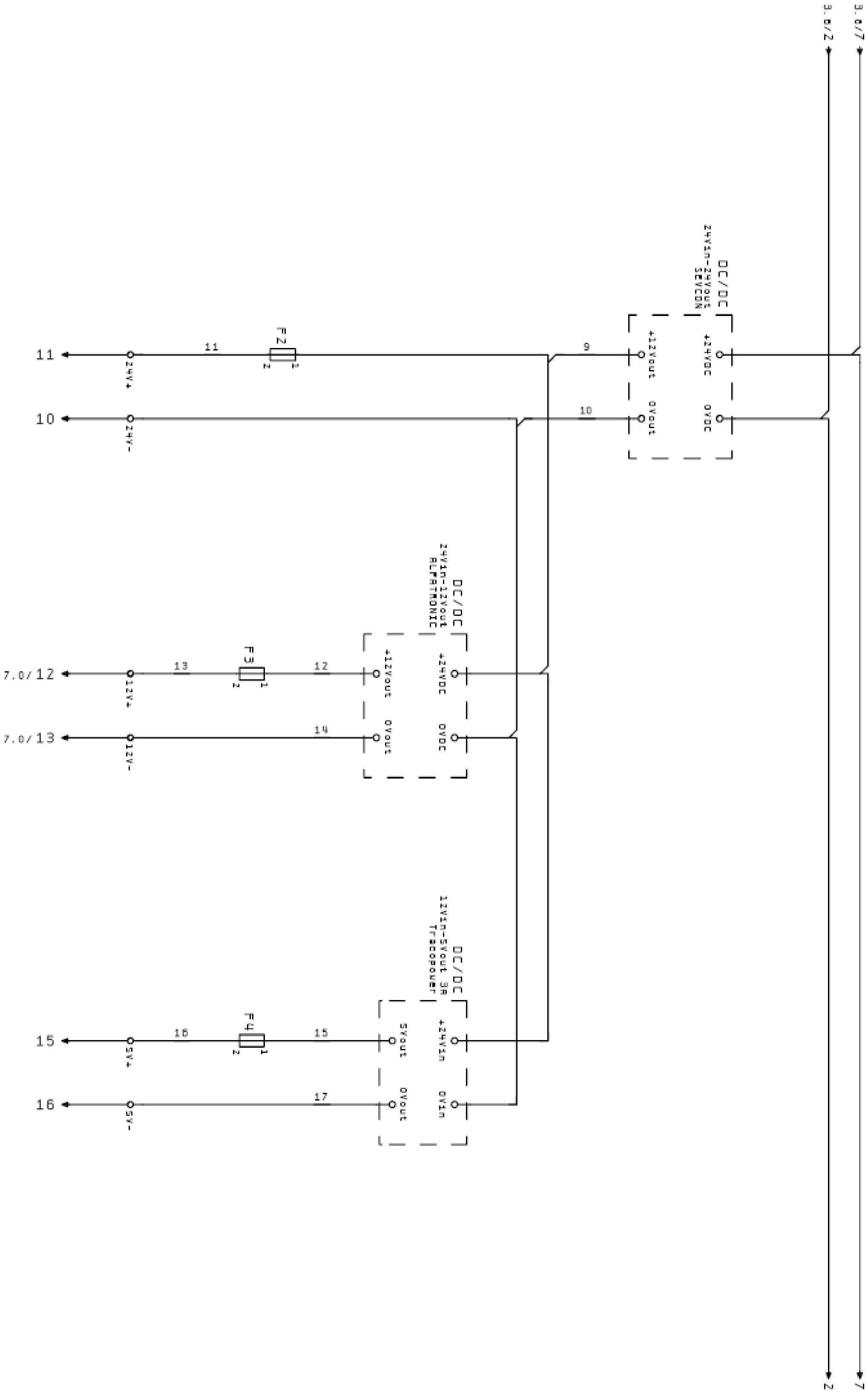


Figure 28: DC/DC Converter Supply Drawing

8.3 Driver Power

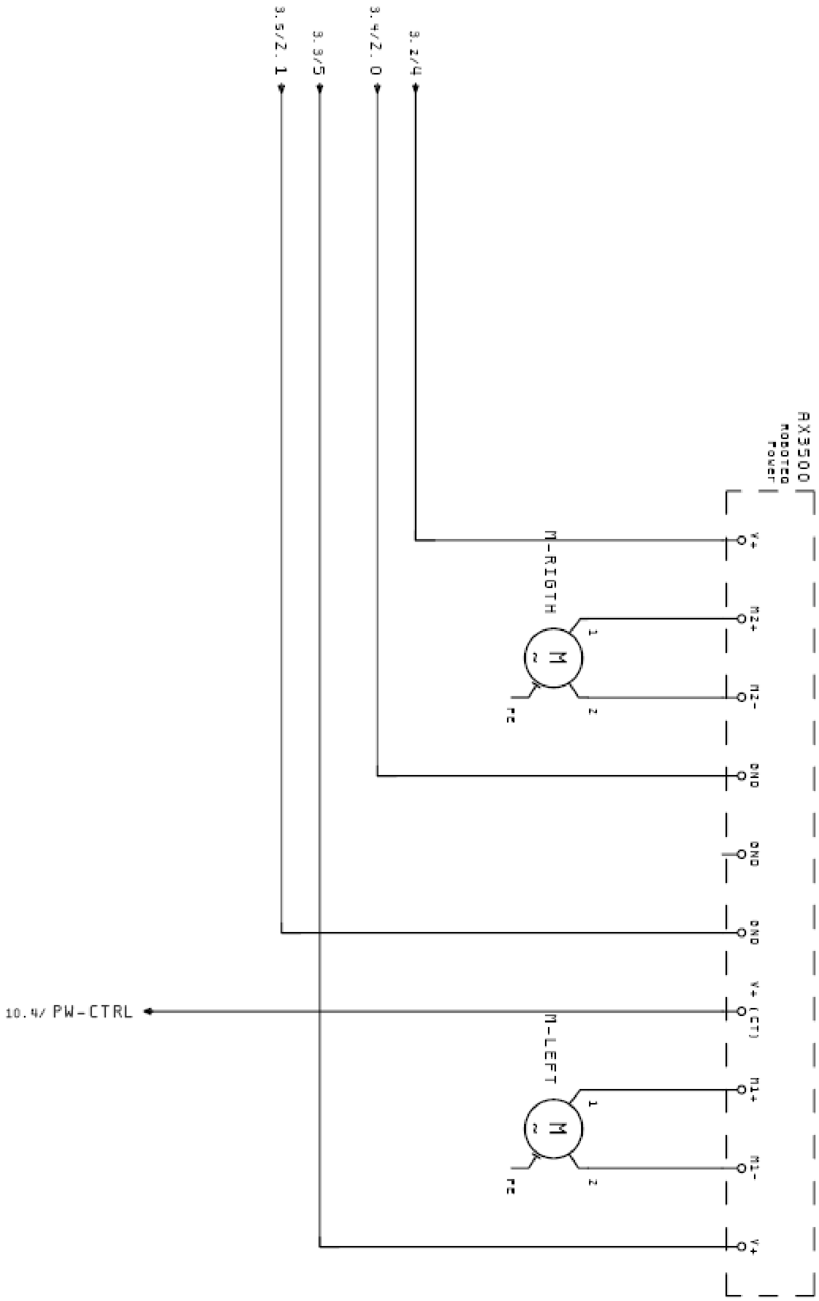


Figure 29: Driver Power Drawing

8.4 Driver - Encoders

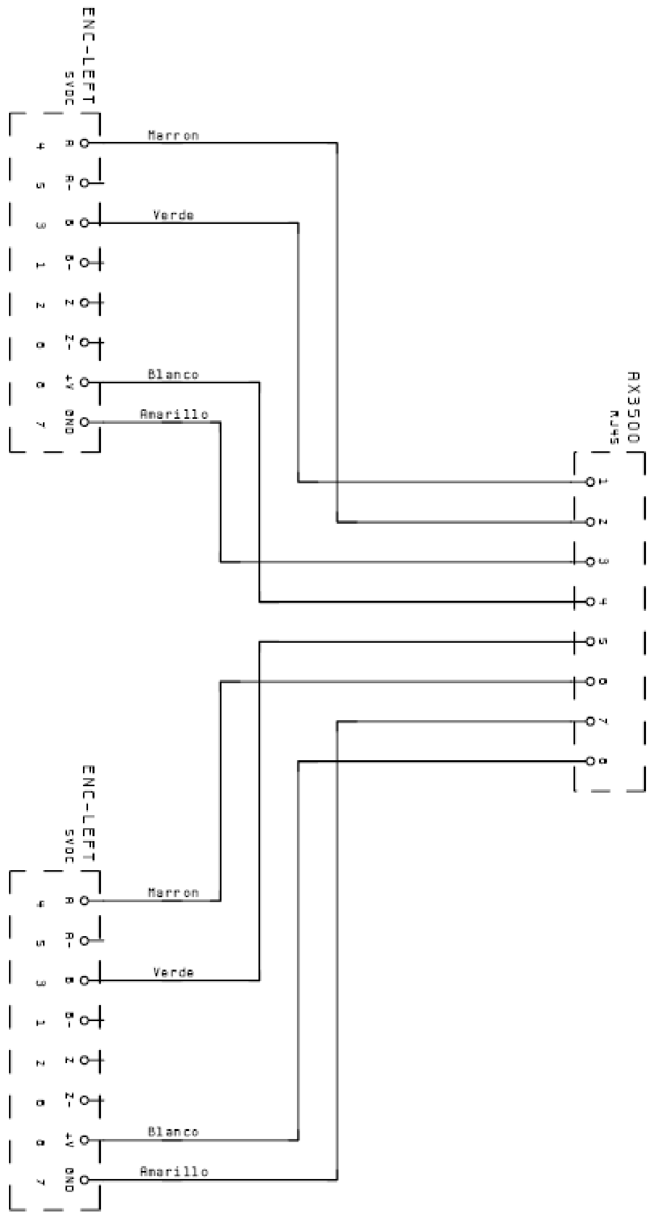


Figure 30: Driver - Encoder Drawing

8.5 Components Power Supply

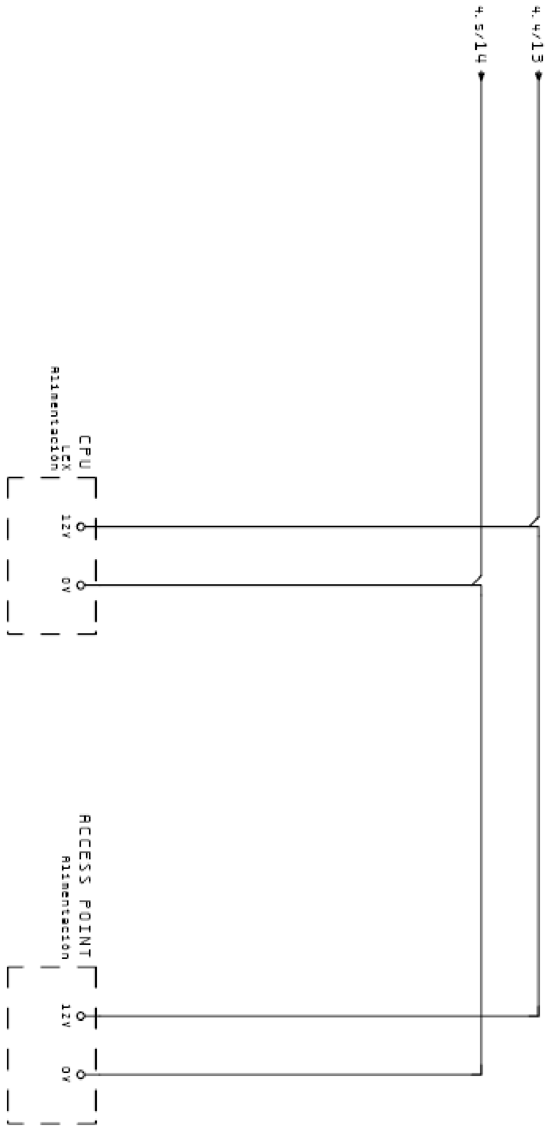


Figure 31: Components Power Supply Drawing

8.6 Laser Connections

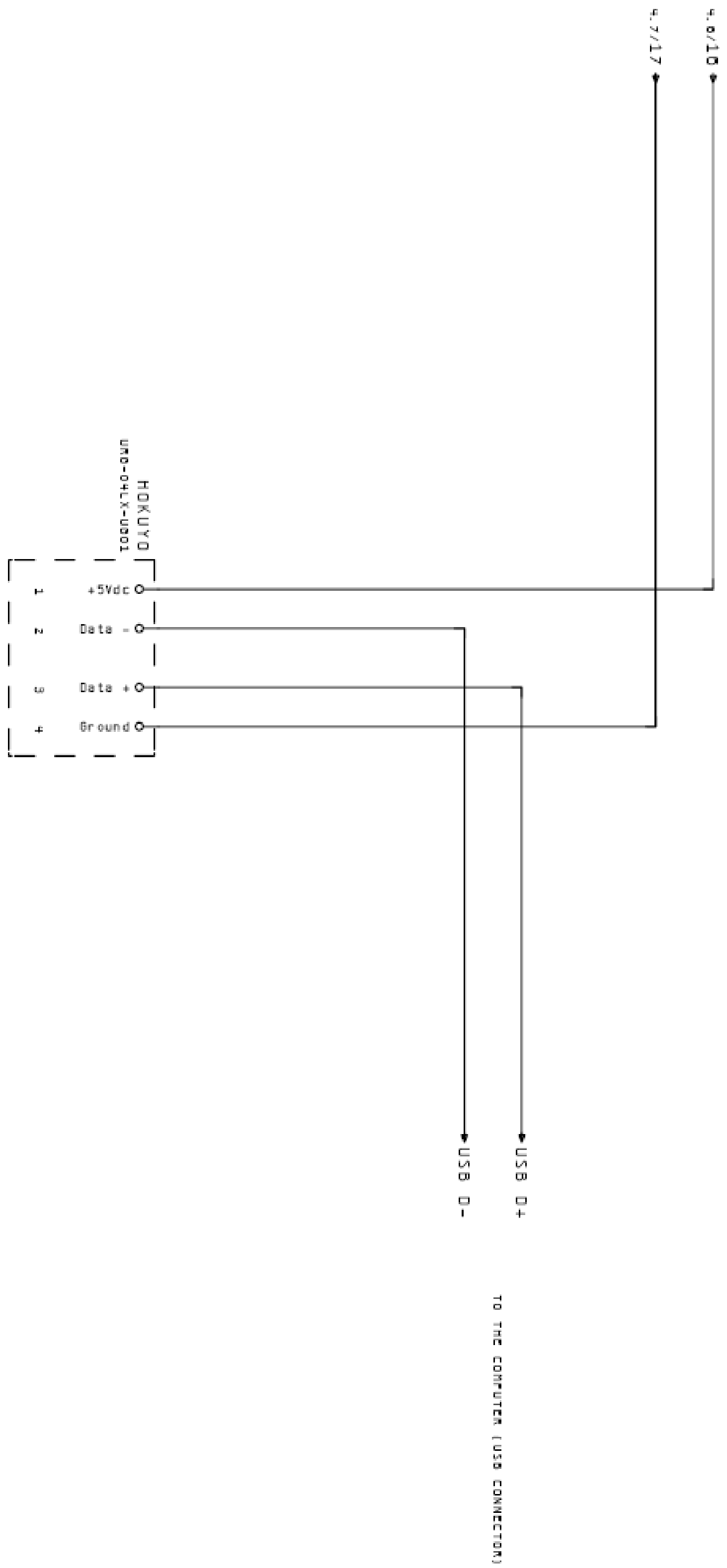


Figure 32: Laser Connections Drawing

8.7 Driver Signals

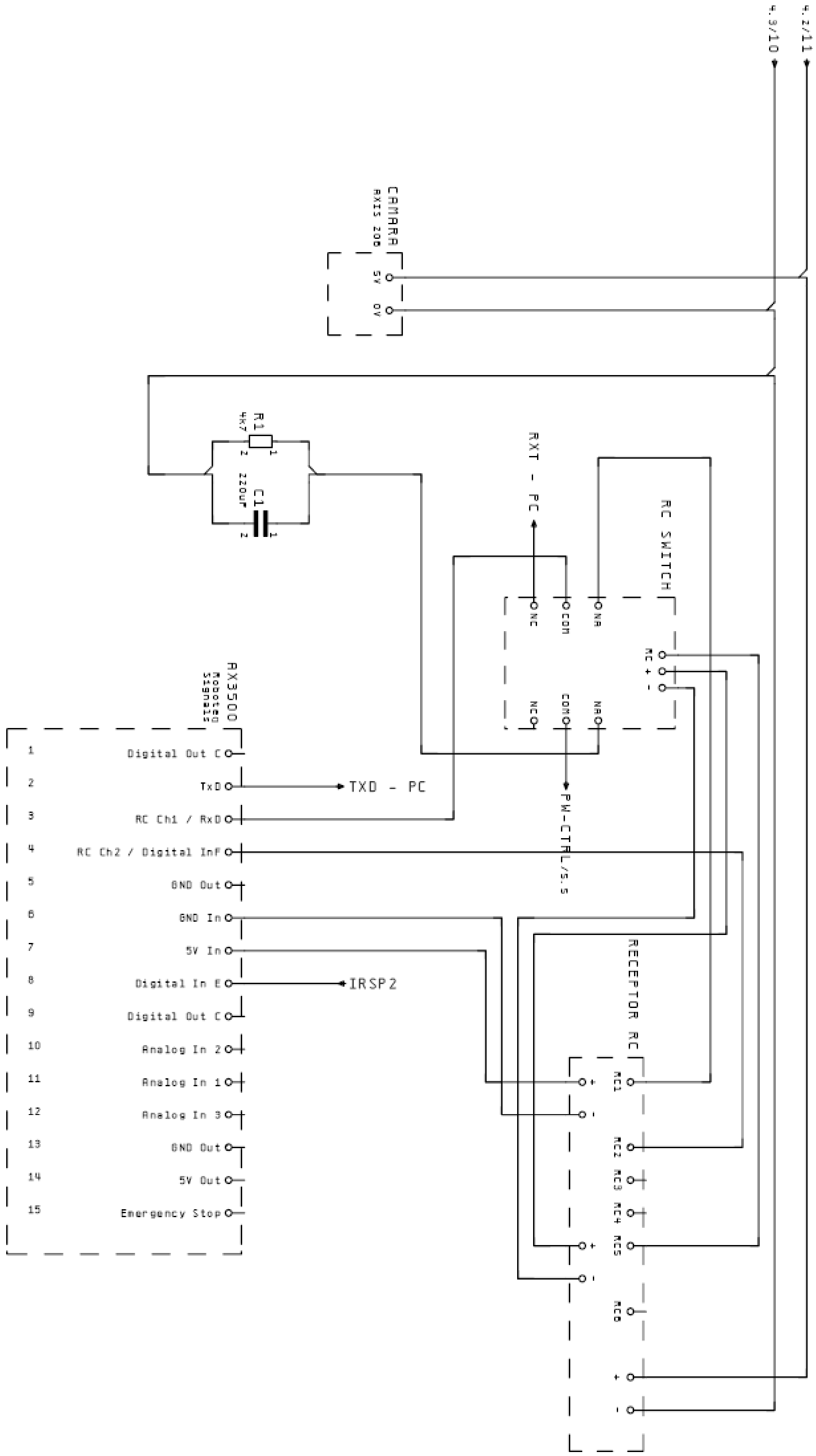


Figure 33: Driver Signals Drawing

8.8 Security Drawing

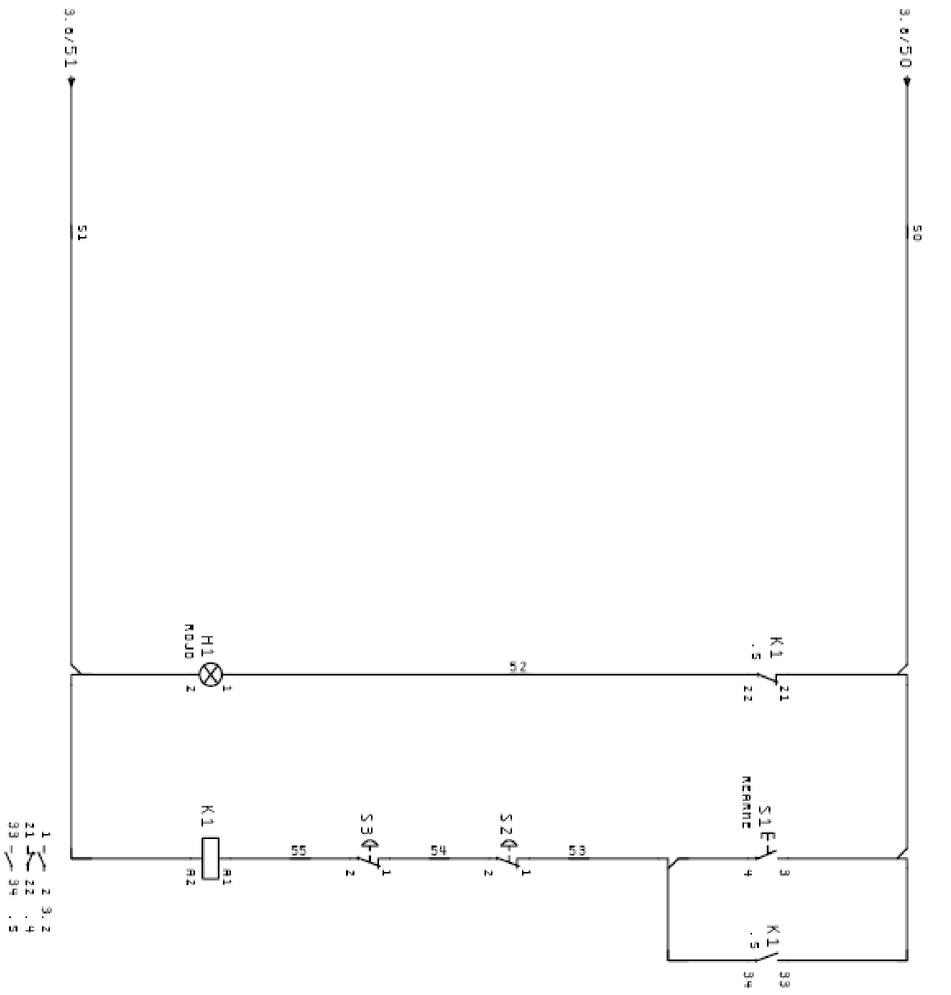


Figure 34: Security Circuit Drawing

9 Conclusion/Summary

Summarized below are the improvements or new features that the Guardians robot offers to the consortium in comparison with the Erratic and Khepera robots:

- High mobility.
- High speed.
- High power
- Obstacles avoidance.
- Possibility to climb stairs.
- Possibility to carry fire fighting special tools.

9.1 Future Work

Next list summarizes several improvements that can be made on the Guardians robot in order to provide more capabilities for fire-fighting applications:

- Foam making possibilities
- Enhanced heat resistance
- Long distance water spraying
- Water pump
- Fire sources discovery capabilities

Finally, the link below shows a video in youtube in which a first version of the Guardians robot can be seen. Moving capabilities are shown as is climbing stairs capabilities inside a building:

http://www.youtube.com/watch?v=IEbsWYB3j60

10 Glossary

CPU	Central Processing Unit
GPS	Global Positioning System
DDR	Double Data Rate
DHCP	Dynamic Host Configuration Protocol
EPP/ECP	Enhanced Parallel Port/Enhanced Capability Port
HD	Hard Disc
I/O	Input/Output
IDE	Integrated Driver Electronics
IP	Internet Protocol
NFS	Network File System
PC	Personal Computer
RC	Radio Control
RJ	Registered Jack
TCP	Transmission Control Protocol
USB	Universal Serial Bus
WEP	Wired Equivalent Privacy

11 List of Figures and Tables

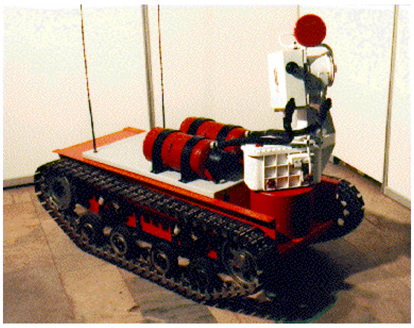
12 References

http://www.inrobtech.com/ProductsSolutions_Cat1_Prd4.asp
http://www.neva.ru/CNII-RTC/fireman.html
http://www.securobot.co.uk/
http://www.research-in-germany.de/coremedia/generator/dachportal/en/04__Research_20Areas/Security_20Technologies/Lab_20Ticker__Security_20research.html
http://www.americancrane.com/Telerob/Firerob.htm
http://www.armtecbrasil.com/ingles/index.php?pagina=area&area=Defsecu&produto=saci
http://www.luf60.com/

Annex. Similar robots

The aim of this annex is to show some robots similar to the Guardians that have been already developed by other companies or universities. Every robot has a brief description of its functionalities.

- **CD&RI RTC**



Assigned for equipping fire-fighting municipal services, airport security services, special-purpose technical centres of the State Committee for Extraordinary Situations. Some of the robot equipment includes: set of television, infrared sensors or a small-size foam-making machine.

Specifications:

- Motion Speed: 1.0 m/s.
- Carrying capacity: 100 Kg.
- Working area: control via radio 1000 m / control via cable 200 m.
- Dimensions: 1.2 x 0.7 x 0.6 m.
- Tele-operated.

- **FIREMOTE**

The Firemote™ is a remote vehicle fitted with a monitor that can be sent into dangerous areas to fight fires. Fire fighting procedures increasingly prevent fire crews from entering buildings on fire, forcing them to fight it less effectively from outside.

Specifications:

- Motion Speed: 1.67 m/s.
- Weight: 200 Kg.
- Working area: control via WiFi, control via umbilical.
- Stair and obstacle climbing ability.
- Dimensions: 0.99 x 0.685 x 1.03 m.
- Tele-operated.

- **OLE**

It is destined to extinguish fires before they can turn into disasters. Is supposed to monitor large areas of forest with the help of infrared and biosensors, discover fire sources and immediately report and fight them. Still under development.

- **FireRob**

The remote-control FIREROB robot can be deployed to fight fires in factories and tunnels, as well as in searching areas where there is the risk of explosion or structural collapse.

Specifications:

- Heat Resistance: 1250 ºC.
- Tele-operated.

➢ **FFR-1**

FFR-1 is a robot designed for high-risk fire fighting missions, allowing fire-fighter to pursue operations until now impossible due to physical constraints or risk to human life.

Specifications:

- Heat Resistance: 400 ºC (1000 ºC sort exposures).
- Motion Speed: 3 – 4 km/h.
- Weight: 940 Kg.
- Dimensions: 1.62 x 1.14 x 1.38 m.
- Tele-operated.

➢ **SACI (Incidents Support Combat System)**

The SACI (Incidents Support Combat System) robot brings the firehose directly into the fire or sprays water or foam from 60 meters. The robot can work for 3 hours before it needs a recharge.

Specifications:

- Modular Construction.
- Distance of the operator to the robot: 120 m.
- Dimensions: 1.60x1.80x1.50 m.

➢ **LUF60**

The LUF60, a wireless remote controlled mobile firefighting support unit, clears the path for advancement of up to a distance of 1000 feet by incorporating a high capacity positive pressure ventilator and a "water beam" fog. This combination clears away smoke, heat, toxic gases and reduces the intensity of the fire, allowing firefighting and rescue teams to more safely approach the incident. The diesel powered LUF60 is a rugged machine that can withstand the rigors of severe operating conditions and confined spaces.

Specifications:

- Size 7'7" L x 4'6" W x 6'7" H.
- Weight 4,410 lbs.
- Turbo charged 95 HP 4-cylinder diesel engine.
- 13 gallon fuel tank.
- 100+ GPM water pump.