

Exploroid

Jonathan Mallia M00518013

Melvyn Bonello M00509680

Owen Muscat M00475032

Julian Camilleri M00555302

Table of Contents

[Objectives 5](#_Toc437553867)

[Design 6](#_Toc437553868)

[Logical System Diagram 6](#_Toc437553869)

[Logical System Flow 8](#_Toc437553870)

[Flow chart - Autonomous mode 8](#_Toc437553871)

[Flow chart - Locomotion in Manual Override Mode 9](#_Toc437553872)

[Laser Pointer Target Mode 10](#_Toc437553873)

[Communication Diagram 11](#_Toc437553874)

[Physical construction 12](#_Toc437553875)

[Functionality 14](#_Toc437553876)

[Move on rough terrain 14](#_Toc437553877)

[Wireless Communication 14](#_Toc437553878)

[Measuring atmospheric variables such as pressure, distance, CO2 Levels, temperature, humidity. 15](#_Toc437553879)

[Motor Drivers 18](#_Toc437553880)

[Laser for targets 18](#_Toc437553881)

[Testing 20](#_Toc437553882)

[Project Management 24](#_Toc437553883)

[Bill of Materials 26](#_Toc437553884)

[Conclusion 27](#_Toc437553885)

[References 28](#_Toc437553886)

Table of Figures

[Figure 1. Hardware Block Diagram 6](#_Toc437549729)

[Figure 2. The Remote Control Block Diagram 7](#_Toc437549730)

[Figure 3. Communication Diagram for the Exploroid Remote control 11](#_Toc437549731)

[Figure 4.Robot base for Chassis 12](#_Toc437549732)

[Figure 5. The Layout of Exploroid major components. 13](#_Toc437549733)

[Figure 6. Wireless Communication 14](#_Toc437549734)

Table of Tables

[Table 1. Pins used to connect to the Arduino 15](#_Toc437412237)

[Table 2.Temperature and Humidity Sensor Data 15](#_Toc437412238)

[Table 3. Pin Connection for DHT22 with Arduino 16](#_Toc437412239)

[Table 4.Ultrasonic Sensor Data 16](#_Toc437412240)

[Table 5.Pin Connection for Ultrasonic Sensor with Arduino 16](#_Toc437412241)

[Table 6.Pressure Sensor Data 16](#_Toc437412242)

[Table 7.Pin Connection for Ultrasonic with Arduino 16](#_Toc437412243)

[Table 8.LDR Data 17](#_Toc437412244)

[Table 9.Pin Connection for LDR with Arduino 17](#_Toc437412245)

[Table 10. CO2 Level Sensor Data 17](#_Toc437412246)

[Table 11. Pins to connect motor driver 18](#_Toc437412247)

[Table 12. Laser Diode Specification 19](#_Toc437412248)

# Objectives

The following are the objectives we have set in our project proposal:

* Move on rough terrain
* Perform Wireless Communication
* Avoid obstacles
* Measure surrounding atmospheric variables such as;
  + Pressure
  + Distance
  + CO2 levels
  + Temperature
  + Humidity
* Autonomous or manual control
* Path memory (remember the paths taken via GPS)
* Transmit crucial data to base
* Drill and clear debris
* Defend itself using a laser pointer

# Design

## Logical System Diagram

Figure 1 shows the composition of the Hardware to be used as a block diagram. The connectivity of each part will be documented below, when describing the functionality of the system. At that stage, the components are defined by part.

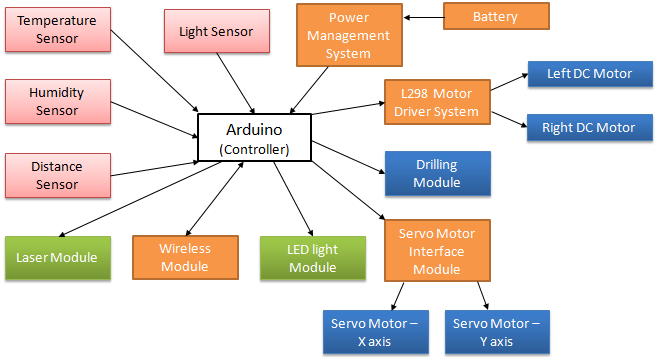


Figure 1. Hardware Block Diagram

The block diagram shows how each component revolves around the Arduino Mega. The following shows a list of blocks connecting to the Arduino.

* The Arduino is the controller which will communicate with all the surrounding blocks such as to drive motors and collect data.
* The Temperature sensor will be connected to the controller so that it provides an analogue voltage which will be converted to a digital value.
* The Humidity sensor is the sensor which will provide the level of the surrounding humidity.
* The Distance sensor will provide feedback to the controller so that the Robot will not collide with walls or other obstacles.
* The Light sensor will be used to measure the surrounding light intensity so that when it gets dark, the Robot switches on the LED Lights.
* The Power Management system will monitor the battery level so that the Robot avoids running out of Power.
* The L298 Motor Driver system will provide the interface between the Arduino controller and the DC motors, so that the controller can provide the signals to drive the motors to Forward, Reverse, Left and Right position.
* The Drilling system will consist of a buffer to drive a DC motor which is coupled with a Drill bit. The Drill bit can be used to collect samples.
* The Servo Motor Interface System is used to drive the servo motors. In this case, two servo motors will be driven from which they will represent the ‘X’ and ‘Y’ axis movements.
* The Laser Module will be mounted on the Servo motors so that it can point to targets.
* The Wireless module is also connected to the controller to act as an interface between the remote device sending commands to Exploroid. The wireless module will then convert the data to serial data and sends it to the Arduino on the serial port.
* The LED light Module is connected to the controller so that when Exploroid senses very low light intensity, the LED lights are switched on.

To remotely control Exploroid, the Arduino will be built with a wireless transmitter to send commands. The Block Diagram shows the Arduino Esplora which consists of all the switches so that they will be used to send the commands to the Robot. The Wireless module will be used as the block to send the commands via RF.

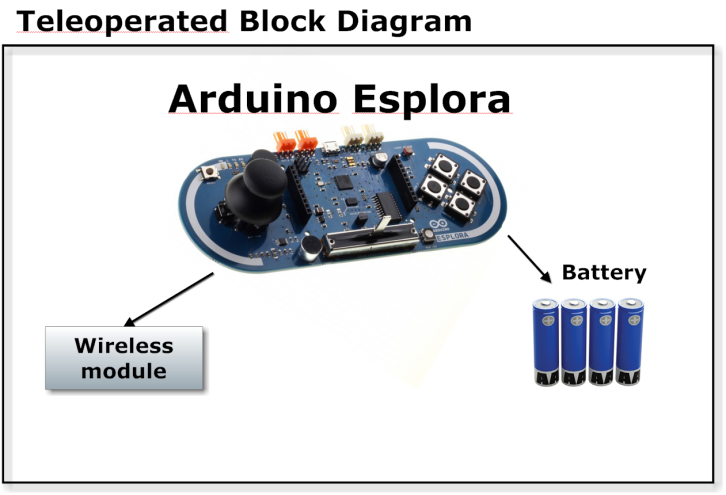


Figure 2. The Remote Control Block Diagram

## Logical System Flow

### Flow chart - Autonomous mode

The following is the flow of how will Exploroid work when moving around. Due to the limitation in battery, the diagram omits functionality to read data from respective sensors such as temperature, humidity or CO2 levels. These will be dealt with in a different flowchart. What is important during locomotion is the detection of obstacles via the ultra-sonic sensors attached to the machine.

Left or Right turn possible?

Rotate Left

Rotate Right

Reverse

YES – Turn Left

NO

Move Forward

Is obstacle detected?

NO

YES

Start

Initialize Hardware

YES – Turn Right

### Flow chart - Locomotion in Manual Override Mode

The following shows the functions called from the received commands which are then executed by Exploroid.

Start

Command Received?

Command == F / R

Move Forward / Reverse

Command == L / R

Move Left / Right

Command == S

Read data from sensors

Command == M1

Change Mode to Automatic

Command == M2

Change Mode to Servo Target Device

End of Manual Mode

Yes

No

No

No

Yes

Yes

No

No

Yes

Yes

Yes

### Laser Pointer Target Mode

The following shows the commands sent to Exploroid remotely to operate the servo module to Drill and aim the laser mechanism to targets.

Yes

No

No

No

No

Start

Command Received?

Command == X / Y / Z

Rotate X/Y Servo Motor/s

Command == L

Switch ON / OFF Laser

Command == S

Rotate Spindle to Drill

Command == M1

Change Mode to Automatic

Command == M3

Change Mode to Manual Override

End of Laser Mode

No

Yes

Yes

Yes

Yes

Yes

### Communication Diagram

Figure 3shows how Exploroid will communicate with home base station. Also remote controller is shown where it can be used to override and control Exploroid directly. The system is pretty much like a Client-Server application, from which the Client (either the Home Base Station or Remote Control) send commands to the Server (Exploroid). A protocol will be established so that both ends can communicate effectively. Both ends will communicate with each other via Radio Frequency (RF) transmission.

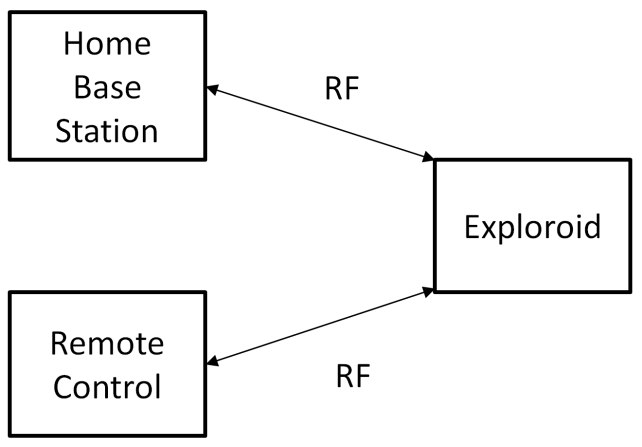


Figure 3. Communication Diagram for the Exploroid Remote control

The next block diagram shows the basic components connected to Exploroid where an Arduino will be used to control all peripherals.

# Physical construction

As defined in the objectives, Exploroid will be able to move on different types of terrain. - In order to achieve this, tracks were chosen as the ideal type of wheels. A differential drive will be used in order to control each track separately.

The figure below shows a similar model of the physical construction of the chassis.



Figure 4.Robot base for Chassis

The advantages of using these tracks are:

* Higher traction base area due to their design
* Have a higher performance and much more power output
* Move on different types of terrain or uneven terrain, such as:
  + Rocks
  + Snow
  + Mud

DC motors will be coupled to the tracks therefore moving the tracks to perform locomotive motion. A more detailed illustration in top view, of how the components will be embedded on the Exploroid chassis is shown below.

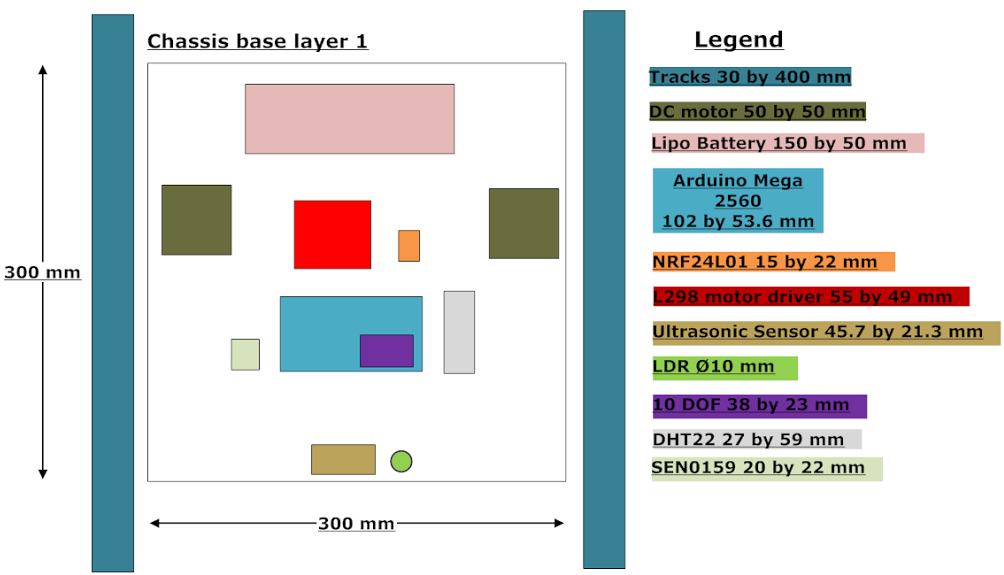


Figure . The Layout of Exploroid major components.

Figure 5 shows the layout in block form, of how the sensors and other major components such as the Arduino, motor driver and sensors will be hosted and embedded on the Exploroid chasis. The diagram is omitting the required wires to connect all the modules together. However, the wires will be passed from underneath the chassis and they will still be covered via a protection layer material. The following table shows the technical specifications of the chassis chosen for Exploroid.

|  |  |
| --- | --- |
| **Specification** | **Values** |
| Material | Aluminium Alloy |
| Weight | 3.25Kg |
| Working Voltage | 7 – 12V |
| Supply Current in TX mode @2000kbps | 12.3mA |
| Dimensions (L x W x H) | 300m x 300m x 500mm |
| DC Motors | High Torque DC Motors |

Table . Chassis Technical Specifications

# Functionality

This section will show the main components which are to be used by Exploroid, so that the Robot will perform several functions, such as, to avoid obstacles, sense high temperatures and other environmental variables.

## Move on rough terrain

This feature will enable Exploroid to explore its surroundings with least possible effort. There will be a lot of challenges to move across terrain such as mud, sand, snow, rocks, etc. Exploroid will be able to move on top of obstacles or slopes to climb to higher grounds. In order to successfully move on such terrain, the robot would need to have locomotive motion.

## Wireless Communication

The best way to communicate with home base station is by wireless communication, this function will help to relay any information gathered through exploration to the base station.

Also this communication will help the user to intervene at any moment and control Exploroid remotely.

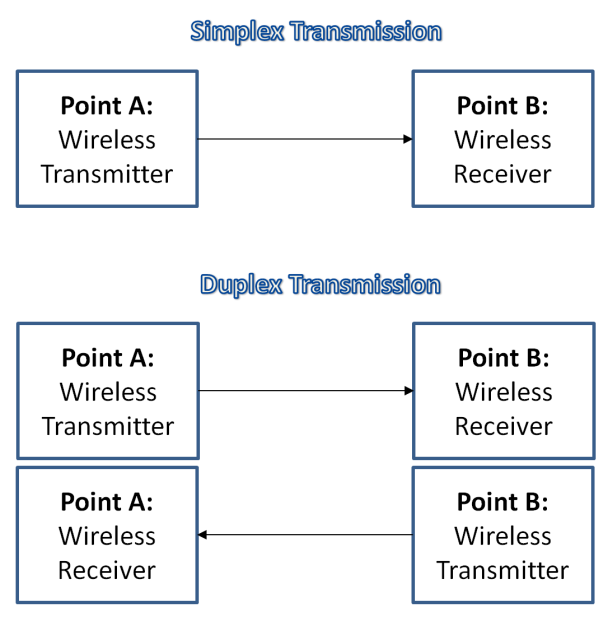


Figure 6. Wireless Communication

There are several wireless technology standards which can be used to communicate between sensors and also home base; each communication standard has different features. We will use Radio Frequency (RF) transmission.Exploroid will use Simplex and Duplex transmission. Simplex will be used when the Robot will be controlled via the Arduino Esplora, which will just send commands to control the robot locomotion.Exploroid will use Duplex since it will be able to receive commands from both the base station and the Esplora Remote control. It must also send data back to the station, thus it must both send and receive.

The module which will be used to transmit and receive data wirelessly will be the NRF24L01. The following table shows how it will be connected to the Arduino.

|  |  |
| --- | --- |
| **Specification** | **Values** |
| Minimum Supply Voltage | 1.9V |
| Maximum data rate | 2000kbps |
| Supply Current in TX mode | 11.3mA |
| Supply Current in TX mode @2000kbps | 12.3mA |
| Temperature Range | -40°C - +85°C |
| Supply Current in sleep mode | 900nA |
| Dimensions | 33mm x 15mm |

Table . Wireless module Data Specifications



|  |  |
| --- | --- |
| **Arduino Pin Name** | **Device pin to be connected** |
| Digital pin 9 (PWM) | NRF24L01 pin 3(CE) |
| Digital pin 53 (SS) | NRF24L01 pin 4(CSN) |
| Digital pin 52 (SCK) | NRF24L01 pin 5(SCK) |
| Digital pin 51 (MOSI) | NRF24L01 pin 6(MOSI) |
| Digital pin 50 (MISO) | NRF24L01 pin 7(MISO) |

Table . Pins used to connect to the Arduino

## Measuring atmospheric variables such as pressure, distance, CO2 Levels, temperature, humidity.

Exploroid will be able to explore its surrounding by collecting feedback via sensors. The following are the sensor types which will be used by the Robot.

* Temperature & Humidity Sensor:
  + The DHT22 will be the sensor to use in order to take temperature and humidity measurements. The advantage of using this sensor is that both temperature and humidity can be read from the same sensor, thus saving space and extra components. The following are the main specifications of the sensor.

|  |  |
| --- | --- |
| **Specification** | **Values** |
| Power | 3V to 5V |
| Current Consumption | 2mA (During conversion) |
| Humidity Readings | 2-5% accuracy |
| Temperature Readings | ±0.5°C |
| Sampling Rate | 0.5Hz |
| Dimensions | 27mm x 59mm x 13.5mm |

Table .Temperature and Humidity Sensor Data

|  |  |
| --- | --- |
| **Arduino Pin Name** | **Device pin to be connected** |
| Analog pin 0 | DHT22 pin 2 |

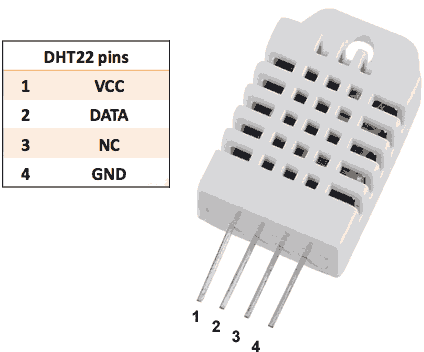
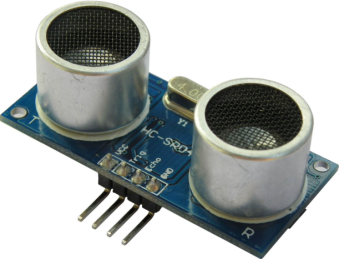


Table .Pin Connection for DHT22 with Arduino

* Ultrasonic Sensor:
  + Ultrasonic Sensors will be used to determine the actual distance from an object. This type of sensor was chosen over infra-red sensors since these are not affected by sunlight or other UV sources which could distort the actual reading.

|  |  |
| --- | --- |
| **Specification** | **Values** |
| Working Voltage | 5V |
| Working Voltage | 15mA |
| Working Frequency | 40Hz |
| Maximum Range | 400cm |
| Minimum Range | 2cm |
| Measuring Angle | 15° |
| Trigger Input Signal | 10uS TTL Logic pulse |
| Echo Output Signal | Input TTL lever signal and the range in proportion |
| Dimensions | 45mm x 20mm x 15mm |

Table .Ultrasonic Sensor Data



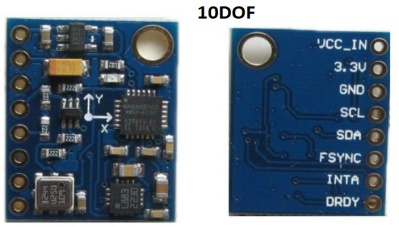
|  |  |
| --- | --- |
| **Arduino Pin Name** | **Device pin to be connected** |
| Digital pin 7 (PWM) | Ultrasonic sensor |

Table .Pin Connection for Ultrasonic Sensor with Arduino

* Pressure Sensor:
  + A pressure sensor will be used to measure atmospheric conditions from the area under exploration. This will help understand more about the area being explored.

|  |  |
| --- | --- |
| **Specification** | **Values** |
| Resolution | 0.17m |
| Pressure Range | 300 - 1100hPa |
| Weight | 2.8g |
| Dimensions | 38mm x 23mm x 3mm |

Table .Pressure Sensor Data



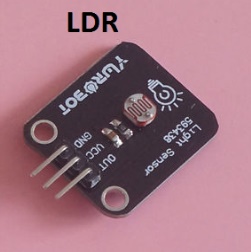
|  |  |
| --- | --- |
| **Arduino Pin Name** | **Device pin to be connected** |
| Digital pin 21 (SCL) | 10 DOF SCL pin |
| Digital pin 20 (SDA) | 10 DOF SDA pin |

Table .Pin Connection for Ultrasonic with Arduino

* Light Dependent Resistor:
  + A Light dependent resistor will be used to detect whether Exploroid has travelled into a cave or areas which barely have light. This could be used also to enable or disable functionality from the Robot.

|  |  |
| --- | --- |
| **Specification** | **Values** |
| Power Dissipation | 80mW |
| Operating Temperature | -40°C to + 75°C |

Table .LDR Data



|  |  |
| --- | --- |
| **Arduino Pin Name** | **Device pin to be connected** |
| Analog pin 1 | LDR pin 3 (out) |

Table .Pin Connection for LDR with Arduino

* CO2 Level Sensor:
  + These sensors will be used to measure the air quality of the environment Exploroid will be exploring. This will be done using the SEN0159 sensor.

|  |  |
| --- | --- |
| **Specification** | **Values** |
| Working Voltage | 5V |
| Interface (Output Voltage) | 30mV to 50mV |
| Operating Temperature | -20°C to +50°C |
| Current Dissipated | 210mA |
| Dimensions | 32mm x 42mm |

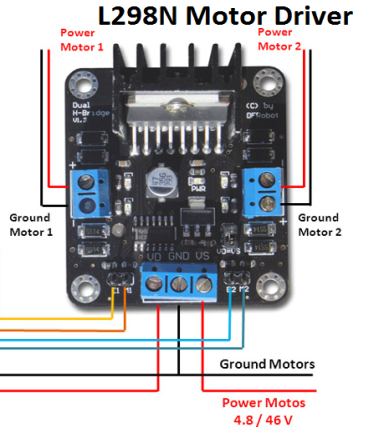
Table . CO2 Level Sensor Data

## Motor Drivers

The motor drivers (L298N) will be used as an interface to drive the motors used for locomotion. The two motors are DC motors and will be utilised to drive the Robot in differential drive. The driver will be used since the Arduino controller is not capable of sourcing enough current to drive the motors. In addition, the motor driver will protect the Arduino controller from a back electro motive force build up in the motor coil. The following table shows the pins used by the Motor driver. The motor driver shield is shown in a separate figure. Another motor driver is also used to driver the motor which will be used to perform drilling.

|  |  |
| --- | --- |
| **Specification** | **Values** |
| Operating supply voltage | Up to 46V |
| Logic supply voltage | 4.5-7V |
| Total DC current | Up to 4A |
| Low saturation voltage | N/A |
| Over-temperature protection | N/A |
| Logical ‘0’ input (high-noise immunity) | Up to 1.5V |
| Dimensions | 43mm x 43mm x 27mm |

Table . L298N H-Bridge Motor Driver Specifications



|  |  |
| --- | --- |
| **Arduino Pin Name** | **Device pin to be connected** |
| Digital pin 10 (PWM) | L298N C1 pin |
| Digital pin 11 (PWM) | L298N M1 pin |
| Digital pin 12 (PWM) | L298N B2 pin |
| Digital pin 13 (PWM) | L298N M2 pin |

Table . Pins to connect motor driver

## Laser for targets

A Laser Diode will be used to point to targets. The Laser diode can be used as a defence mechanism to fire shots to targets which pose as threats. The following shows the specifications of the laser diode.

|  |  |
| --- | --- |
| **Specification** | **Values** |
| Weight | 6.3g |
| Diameter | 10mm |
| Length | 31mm |
| Wavelength | 650nm |
| Voltage Input | 2.8-5.2V |
| Current | 25mA |
| Operating Temperature | -10°C - 40°C |

Table . Laser Diode Specification



|  |  |
| --- | --- |
| **Arduino Pin Name** | **Device pin to be connected** |
| Digital pin 8 (PWM) | Anode (Red) wire |

# Testing

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Expected | Actual |
| 1 | This test case will cover the forward movement of the robot.  **Instructions:**  Pass ‘move forward’ command to robot. | The motors of the robot will both rotate in a clockwise motion moving the robot forward |  |
| 2 | This test case will cover the backward movement of the robot.  **Instructions:**  Pass ‘move backward’ command to robot. | The motors of the robot will both rotate in an anticlockwise motion moving the robot backward |  |
| 3 | This test case will cover the turning left motion of the robot.  **Instructions:**  Pass ‘move left’ Command to robot. | The left motor of the robot will decrease in speed while keeping the right motor at the same speed moving the robot forward or backwards in the left direction |  |
| 4 | This test case will cover the turning right motion of the robot.  **Instructions:**  Pass ‘move right’ Command to robot. | The right motor of the robot will decrease in speed while keeping the left motor at the same speed moving the robot forward or backwards in the right direction |  |
| 5 | This test case will cover the arm rotating on its own axis.  **Instructions:**  Pass ‘arm rotate right/left’ command to robot. | The servo on the base of the arm will turn anticlockwise or clockwise depending on the command passed rotating the arm on its own axis. |  |
| 6 | This test case will cover the arm raising and lowering.  **Instructions:**  Pass ‘arm elevate’ command to robot | The servo on the arm will turn anticlockwise or clockwise depending on the command passed raising or lowering the arm. |  |
| 7 | This test case will cover the arm opening its gripper.  **Instructions:**  Pass ‘arm open gripper’ command to robot | The servo on the gripper will turn anticlockwise until the gripper is open. |  |
| 8 | This test case will cover the arm closing its gripper.  **Instructions:**  Pass ‘arm close gripper’ command to robot | The servo on the gripper will turn clockwise until the gripper is closed. |  |
| 9 | This test case will cover the laser switching on.  **Instructions:**  Pass ‘laser on’ command to robot | The laser on the robotic arm will switch on. |  |
| 10 | This test case will cover the laser switching off.  **Instructions:**  Pass ‘laser off’ command to robot | The laser on the robotic arm will switch off. |  |
| 11 | This test case will cover the laser’s up movement.  **Instructions:**  Pass ‘laser up’ command to robot | The laser should point further up. |  |
| 12 | This test case will cover the laser’s down movement.  **Instructions:**  Pass ‘laser down’ command to robot | The laser should point further down. |  |
| 13 | This test case will cover the laser’s left movement.  **Instructions:**  Pass ‘laser left’ command to robot | The laser should point further to the left. |  |
| 14 | This test case will cover the laser’s right movement.  **Instructions:**  Pass ‘laser right’ command to robot | The laser should point further right. |  |
| 15 | This test case will cover the drill functionality.  **Instructions:**  Pass ‘drill’ command to robot | The drill on the robotic arm will rotate. |  |
| 16 | This test case will cover the temperature sensor.  **Instructions:**  Pass ‘temperature’ command to robot; and log the temperature value.  Place more heat on the robot and run the ‘temperature’ command again. | The temperature sensor should log two values; the first lower than the second value. |  |
| 17 | This test case will cover the humidity sensor.  **Instructions:**  Pass ‘humidity’ command to robot; and log the humidity value.  Place the humidity sensor in a higher humidity area (such as; close to boiling water) and run the ‘humidity’ command again. | The humidity sensor should log two values; the first lower than the second. |  |
| 18 | This test case will cover the light dependent resistor.  **Instructions:**  Pass ‘light status’ command to robot in a lit room; and log the value.  Place the light dependent resistor in a dark room and run the ‘light status’ command again. | The light dependent resistor should log two values; resistance should vary between the two. |  |
| 19 | This test case will cover the CO2 sensor.  **Instructions:**  Pass ‘co2’ command to robot; and log the value.  Place CO2 near sensor and pass the ‘co2’ command again. | The CO2 sensor should log two values, one negative and one positive. |  |
| 20 | This test case will cover the pressure sensor.  **Instructions:**  Pass ‘pressure’ command to robot; and log the value.  Place pressure on the sensor and run the ‘pressure’ command again. | The pressure sensor should’ve logged two values; the first lower than the second. |  |
| 21 | This test case will cover the ultrasonic sensor.  **Instructions:**  Place a box in front of the ultrasonic sensor at around 50cm away; pass the ‘ultrasonic’ command to robot and log the values. | The ultrasonic sensor should log values of ~50cm. |  |

# Project Management

A project timeline was created so that every member knows which task needs to be done in the allowed time frame to succeed in completing this project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task ID** | **Task Description** | **Start Date** | **NO. days to complete** | **Stop Date** | **Assigned to** |
| EXP-1 | Order all parts and materials | 09/12/2015 | 5 | 13/12/2015 | Julian & Owen |
| EXP-2 | Design electronic schematic | 11/12/2015 | 13 | 23/12/2015 | Melvyn & Jonathan |
| EXP-3 | Assemble hardware chassis | 02/01/2016 | 4 | 05/01/2016 | Melvyn & Jonathan |
| EXP-4 | Design software structure | 13/12/2015 | 4 | 16/12/2015 | Jonathan & Julian |
| EXP-5 | Code, assemble and test DHT22 sensor | 14/12/2015 | 6 | 19/12/2015 | Owen |
| EXP-6 | Code, assemble and test L298N driver | 14/12/2015 | 5 | 18/12/2015 | Julian |
| EXP-7 | Code, assemble and test LDR | 14/12/2015 | 3 | 16/12/2015 | Owen |
| EXP-8 | Code, assemble and test 10DOF sensor | 06/01/2016 | 5 | 10/01/2016 | Melvyn & Jonathan |
| EXP-9 | Build remote control | 17/12/2015 | 10 | 26/12/2015 | Jonathan |
| EXP-10 | Design software UI interface | 17/12/2015 | 30 | 15/01/2016 | Owen & Julian |
| EXP-11 | Code, assemble and test NRF2401 module | 18/12/2015 | 16 | 02/01/2016 | Melvyn |
| EXP-12 | Write firmware integration with hardware | 04/01/2016 | 16 | 19/01/2016 | Melvyn & Jonathan |
| EXP-13 | Interface UI software with hardware | 05/01/2016 | 18 | 22/01/2016 | Owen & Julian |
| EXP-14 | Initial hardware integration tests | 23/01/2016 | 2 | 24/01/2016 | Jonathan & Owen |
| EXP-15 | Update software & hardware according to test results | 24/01/2016 | 15 | 07/02/2016 | Melvyn, Jonathan, Owen & Julian |
| EXP-16 | Write first draft of Project Report | 20/01/2016 | 10 | 29/01/2016 | Melvyn, Jonathan, Owen & Julian |
| EXP-17 | Re-test hardware and software functionality | 29/01/2016 | 3 | 31/01/2016 | Melvyn & Julian |
| EXP-18 | Update software & hardware according to test results | 31/01/2016 | 12 | 11/02/2016 | Melvyn, Jonathan, Owen & Julian |
| EXP-19 | Calibrate hardware | 11/02/2016 | 3 | 13/02/2016 | Jonathan & Julian |
| EXP-20 | Write report | 15/02/2016 | 18 | 03/03/2016 | Melvyn & Owen |
| EXP-21 | Write presentation | 24/02/2016 | 6 | 29/02/2016 | Jonathan & Julian |
| EXP-22 | Review write up | 03/03/2016 | 2 | 04/03/2016 | Jonathan & Julian |
| EXP-23 | Final software and tweaks | 21/02/2016 | 6 | 26/02/2016 | Melvyn, Jonathan, Owen & Julian |
| EXP-24 | Submit robot and presentation | 06/03/2016 | 2 | 07/03/2016 | Melvyn, Jonathan, Owen & Julian |

Table . The tasks identified and their respective human resources being allocated

# Bill of Materials

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Components | Quantity | Price | Supplier | Part Number |
| Arduino Mega 2560 R3 | 1 | € 87.98c | RS Components | 769-7418 |
| Arduino Esplora | 1 | € 68.31c | RS Components | 774-8150 |
| LV-MaxSonar-Ez (MB1000) | 1 | €24.95 | Adafruit | 172 |
| LDR-VT90N2 | 2 | €5.58 | RS Components | 234-1050 |
| Led strip | 1 | €19.95c | Adafruit | 2237 |
| NRF24l01 Transceiver RF | 1 | €5.32 | Ebayuk | 131298391032 |
| 10DOF Breakout Board | 1 | €27.54 | Adafruit | 1604 |
| SEN0159 (CO2 Sensor) | 1 | €55.04 | Jameco | 2213270 |
| Lipo Battery | 1 | €10.67 | HobbyKing | T2200.3S.25 |
| Buzzer | 1 | €1.85 | RS Components | 535-8253 |
| Capacitor Pack | 1 |  |  |  |
| Resistor Pack | 1 |  |  |  |
| Switches | 2 | €2.70 | RS Components | 734-7154 |
| Laser Diode | 1 | €1.76 | Ebayuk | 251809256411 |
| DHT22 Temperature and Humidity sensor | 1 | €9.13 | Adafruit | 385 |
| L298 Motor Shield | 2 | €4.88 | Ebayuk | 371284780041 |
| DC motors | 3 | €24.93 | Ebayuk | 150530611989 |
| Servo Motors | 4 | €49.88 | Ebayuk | 380754442695 |
| Wires (Mixed Colours) | 2 | €8 | Ebayuk | 331134442188 |
| Chassis | 1 |  |  |  |
| Breadboard | 1 | €9.42 | RS Components | 102-9147 |
| Vero Board | 1 | €12.82 | RS Components | 433-826 |
| Printed Circuit Board | 1 | €2.96 | Ebayuk | 261199157440 |
| Drill Bit | 1 | €6.63 | RS Components | 213-922 |

# Conclusion

In conclusion to the complete design report; Exploroid will be able to;

* Explore its surroundings
* Gather information through sensors and link the information to the current GPS location; such as:
  + CO2
  + Temperature
  + Humidity
  + Pressure
* Visualization of the surroundings through an attached camera
* Carry out tasks with an articulated arm
  + Drill
  + Pick up objects
* Navigate through different types of terrain

Raspberry Pi will be used to carry out two crucial features;

* Give visualization of the surroundings through a camera
* Host a web server
* Communicate commands to Arduino

The reason why we chose to host a web server on Exploroid is to be able to;

* Remotely access the robot through web and carry out tasks
* Lower risk of data corruption

*Why is there a lower risk of data corruption?*

There’s a lower risk of data corruption because the data is stored directly onto a local database; that database will be synced (according to a timeframe) to a remote database.

# References

Google Docs,.'HC-SR04 User's\_Manual'.N.p., 2015. Web. 16 Nov. 2015.Calin, Dragos et al. 'Wheels Vs Continuous Tracks: Advantages And Disadvantages | Into Robotics'. Into Robotics.N.p., 2013. Web. 12 Nov. 2015.

Arduino.cc,.'Arduino - Home'.N.p., 2015. Web. 11 Nov. 2015.

Instructables.com,.'Arduino + 2 Servos + Thumbstick (Joystick)'.N.p., 2015. Web. 14 Nov. 2015.

Wikipedia,.'Robotic Arm'.N.p., 2015. Web. 14 Nov. 2015.

Adafruit.com,. 'DHT22 Temperature-Humidity Sensor + Extras ID: 385 - $9.95 :Adafruit Industries, Unique & Fun DIY Electronics And Kits'. N.p., 2015. Web. 15 Nov. 2015.

Dfrobot.com,.'CO2 Sensor (Arduino Compatible)'.N.p., 2015. Web. 15 Nov. 2015.

http://maxbotix.com/documents/LV-MaxSonar-EZ\_Datasheet.pdf