2023 - 2024

Prepared by
Mohamed Adam Jemal

RoverBot

ARDUINO

WELCOME TO “RoverBot” PRESENTATION

PLAN

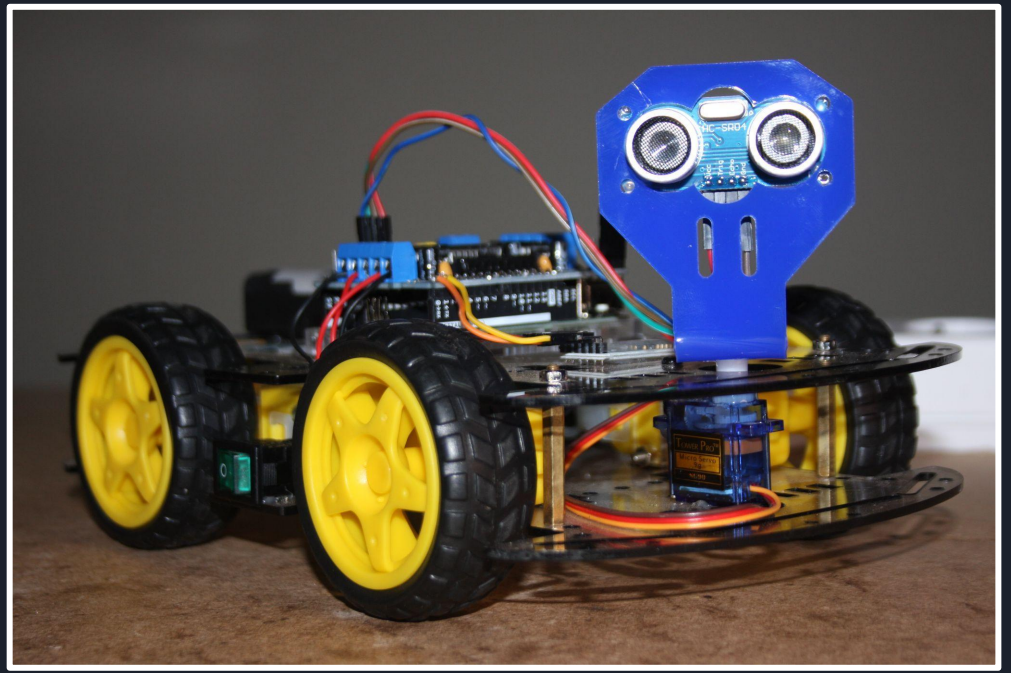
I. HARDWARE
1. PIECES USED
2. TOOLS USED
3. HARDWARE ASSEMBLY PROCESS
4. FINAL APPEARANCE

II. SOFTWARE

III. ACQUIRED SKILLS
1. HARD SKILLS
2. SOFT SKILLS

IV. PRESENTATION VIDEO OF “RoverBot”

V. ACKNOWLEDGEMENT



HARDWARE

Pieces used

Piece(s)	Datasheets Links
Arduino Uno Rev3	Arduino Uno Rev3
L293D motor driver shield	L293D Based Arduino Motor Shield
Robot Car Kit 4WD	Robot Car Kit 4WD | Joy-IT
4X gear motors and 4 Wheels	Gear Motor
Servo Motor (Tower Pro SG90)	SG90 Servo Datasheet pdf - Micro Servo. Equivalent. Catalog

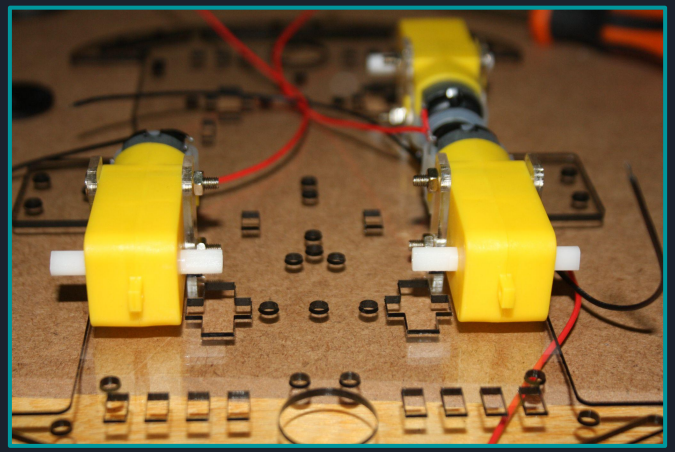
HARDWARE

Bluetooth Module (HM-10 BLE 4.0)	HM-10 BLE 4.0 Bluetooth Module
Ultrasonic sensor (HC-SR04)	Ultrasonic Ranging Module HC - SR04
2X 18650 Battery (3200mAh) with holder	Panasonic NCR18650BD 3200mAh 3C Battery
ON/OFF Switch	ON-OFF switch black 16A-250V TES-13 ADAJUSA
Jumper wires	PRODUCT DATASHEET

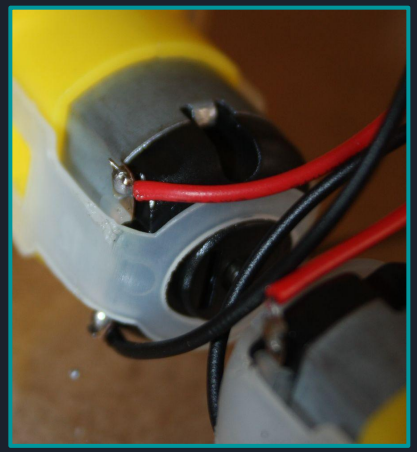
Tools used

Tools	Links
Multimeter	Multimeter - Wikipedia
Soldering Iron	Soldering iron - Wikipedia
Glue Gun	Hot-melt adhesive - Wikipedia
Nut and bolts	Nut (hardware) - Wikipedia Bolt (fastener) - Wikipedia
Screwdriver	Screwdriver - Wikipedia
Pliers	Pliers - Wikipedia
Adhesive tape	Adhesive tape - Wikipedia

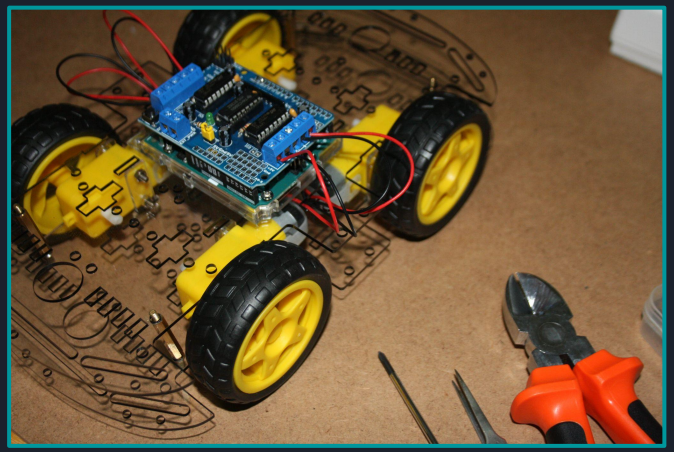
Hardware Assembly Process



Mounting of the motors on the chassis

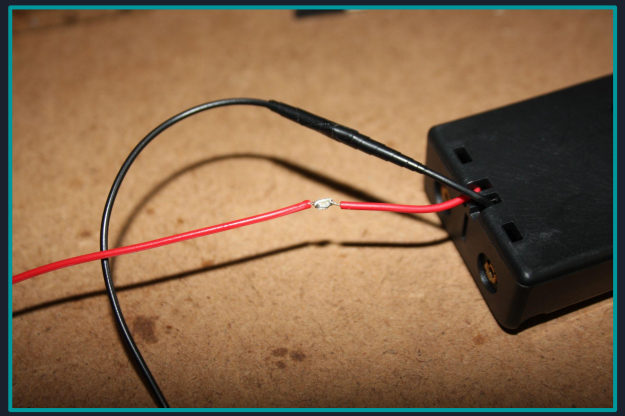


Soldering the wires to the motors

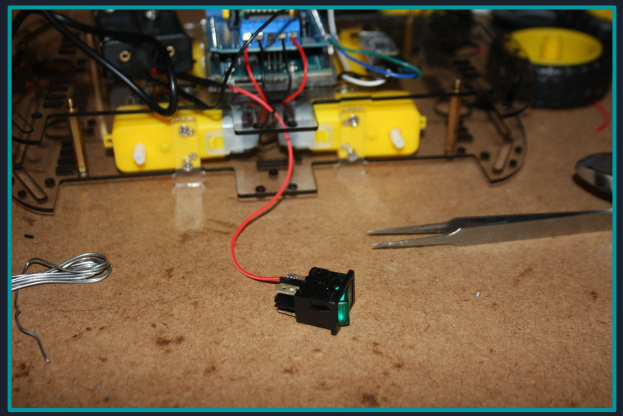


Mounting of the Arduino Uno board (with L293D motor driver shield) on the chassis

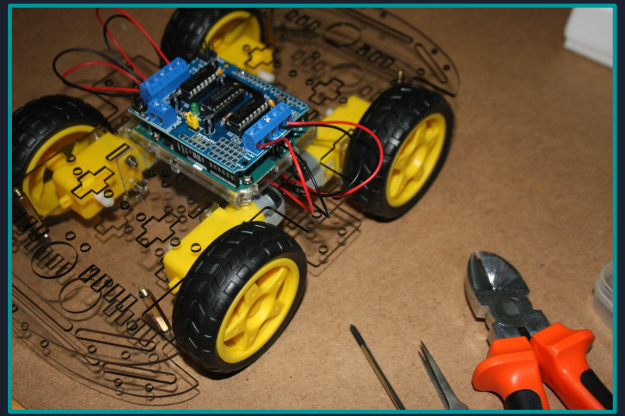
Hardware Assembly Process



Soldering the battery holder wires

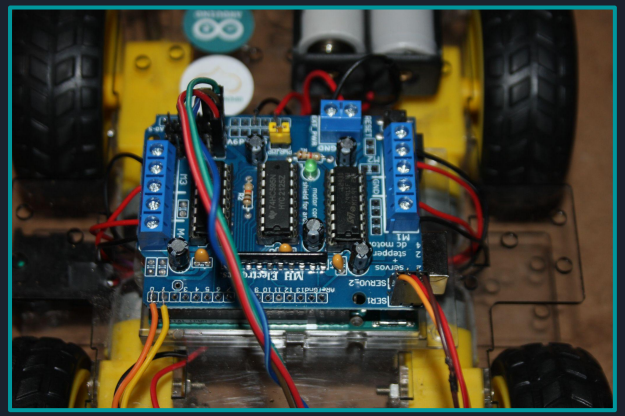


Soldering the wires to the switcher

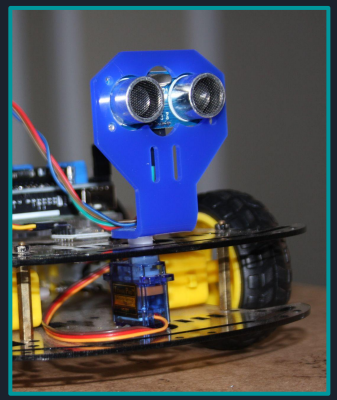


Mounting of the Arduino Uno board (with L293D motor driver shield) on the chassis

Hardware Assembly Process



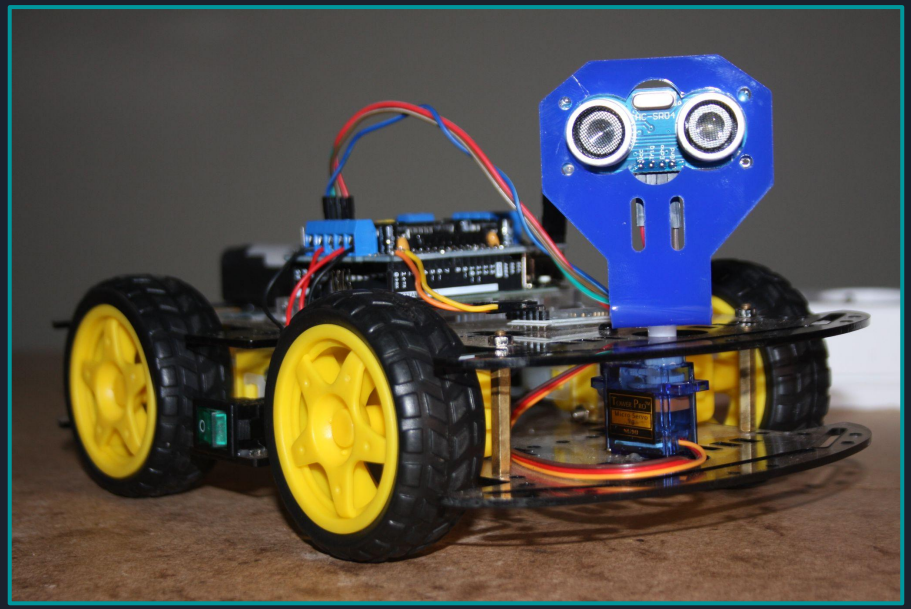
Connecting all wires to the L293D shield



Assembling the ultrasonic sensor and the servo motor

Setting up the HM-10 Bluetooth module

Final Appearance



Hi I'm "RoverBot"

SOFTWARE

Code Link : Arduino IDE (INO FILE) Or TXT FILE

Code explanation

Inclusion of libraries

```
#include <AFMotor.h>    // Include the AFMotor library for motor control
#define CUSTOM_SETTINGS
#define INCLUDE_GAMEPAD_MODULE
#include <Dabble.h>    /* Include the Dabble library for Bluetooth
Communication */
#include <Servo.h>    /* Include the Servo library for servo motor
control */
#include <NewPing.h> /* Include the NewPing library for the ultrasonic
sensor */
```

Creating Objects

```
NewPing uls_sensor1(A0, A1, 400);   /* Create an object of the
NewPing class for ultrasonic sensor with the AnalogPin 0 as Trig
pin and the AnalogPin1 as an EchoPin and a max distance of 400cm*/

Servo servo1;  // Create a Servo object for the servo motor

AF_DCMotor motor1(1);   // Create AF_DCMotor objects for motor1
AF_DCMotor motor2(2);   // Create AF_DCMotor objects for motor2
AF_DCMotor motor3(3);   // Create AF_DCMotor objects for motor3
AF_DCMotor motor4(4);   // Create AF_DCMotor objects for motor4
```

SOFTWARE

Movement functions

```
/* Function to make RoverBot move
forward */
void forward() {
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}
```

```
/* Function to make RoverBot move
backward */
void backward() {
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
}
```

```
// Function to make RoverBot turn right
void right() {
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
}
```

```
// Function to make RoverBot turn left
void left() {
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}
```

Movement functions

```
// Function to stop all RoverBot motors

void stop() {
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}
```

SOFTWARE

Distance measurement functions

```
// Function to measure the forward distance using the ultrasonic sensor
unsigned int mes_dist_forward() {
  unsigned int distance = uls_sensor1.ping_cm();
  return distance;
}
```

```
/* Function to measure the distance to the right using the servo and ultrasonic sensor */
int mes_dist_right() {
  servo1.write(10);
  delay(1000);
  unsigned int dist_right = uls_sensor1.ping_cm();
  delay(300);
  servo1.write(90);
  return dist_right; }
```

```
/* Function to measure the distance to the left using the servo and ultrasonic sensor */
int mes_dist_left() {
  servo1.write(170);
  delay(1000);
  unsigned int dist_left = uls_sensor1.ping_cm();
  delay(300);
  servo1.write(90);
  return dist_left; }
```

SOFTWARE

Distance measurement functions

```
/* Function to choose between turning to the right or the left
based on the longest distance captured by the ultrasonic
Sensor */
void search() {
  if (mes_dist_right() >= mes_dist_left()) {
    right();
    delay(350);
    forward();
  } else {
    left();
    delay(350);
    forward();
  }
}
```

Setup function

```
void setup() {
  Dabble.begin(9600, 0, 1);  // Initialize Dabble for Bluetooth communication
  motor1.setSpeed(255);  // Set motor speeds
  motor2.setSpeed(255);
  motor3.setSpeed(255);
  motor4.setSpeed(255);
  servo1.attach(10);  // Attach the servo to pin 10
  servo1.write(90);   // Set the initial position of the servo (rotation degree)
}
```

Loop function

```
bool obs_mode = true; /* Initialize RoverBot to the
Autonomous Mode (Obstacles avoiding Mode) */

void loop() {
  Dabble.processInput();  // Process Bluetooth input from
Dabble app
  if (GamePad.isCrossPressed()) {
    obs_mode = false; // deactivate the Autonomous Mode
  } else if (GamePad.isTrianglePressed() || (obs_mode)) {
    obs_mode = true; // activate the Autonomous Mode
    if (mes_dist_forward() >= 33) {
      forward();
    } else {
      stop();
      delay(500);
      search();
      forward();
    }
```

```
} else {
    if ((GamePad.isUpPressed()) && (mes_dist_forward() >= 33))
{
      forward();
    } else if (GamePad.isDownPressed()) {
      backward();
    } else if (GamePad.isRightPressed()) {
      right();
    } else if (GamePad.isLeftPressed()) {
      left();
    } else {
      stop();
    }
  }
}
```

ACQUIRED SKILLS

Hard Skills

1. Programming :

Writing code for “Roverbot” enhances my coding skills, particularly in C++, which includes several libraries and Object-Oriented Programming.

1. Electronics Knowledge :

Building and assembling “RoverBot” requires an understanding of electronic components, circuits, and wiring. I learned how to use Arduino, L293 motor driver shield, Bluetooth module, sensors, and motors, and how to connect and control them.

1. Soldering :

There is no doubt that soldering is necessary in the electronics field. That’s offered me the opportunity to acquire basic soldering skills.

4. Troubleshooting :

While I was working on the Robot Car , I encountered several issues and bugs. Troubleshooting and debugging RoverBot behavior helped me develop my precise observation and fixing skills.

5. Data Analysis:

Since Roverbot uses sensors to collect data (e.g., distance measurements), it is necessary to analyze this data to make decisions. This improved my data analysis skills.

Soft Skills

1. Stress management :

Managing stress and pressure correctly can bring out the best in individuals, leading to heightened performance and the ability to excel under challenging conditions.

1. Problem Solving :

Building and programming a Robot involves solving various technical problems and challenges, enhancing your problem-solving and persistence abilities. As well as my focusing ability because dealing with electronics components is extremely sensitive in terms of voltage, amperage and more.

1. Creativity :

Designing my Roverbot's functionality and appearance allows me to exercise my creativity in engineering and design choices.

1. Project Management :

Planning and executing a robotics project teaches project management skills such as time management, task prioritization, and goal setting.

1. Presentation Skills:

Sharing my project with others, whether in a presentation, report or video, improves my ability to convey complex technical information to a general audience.

1. Resourcefulness :

Limited resources or unexpected challenges may require being resourceful in finding creative solutions and workarounds.

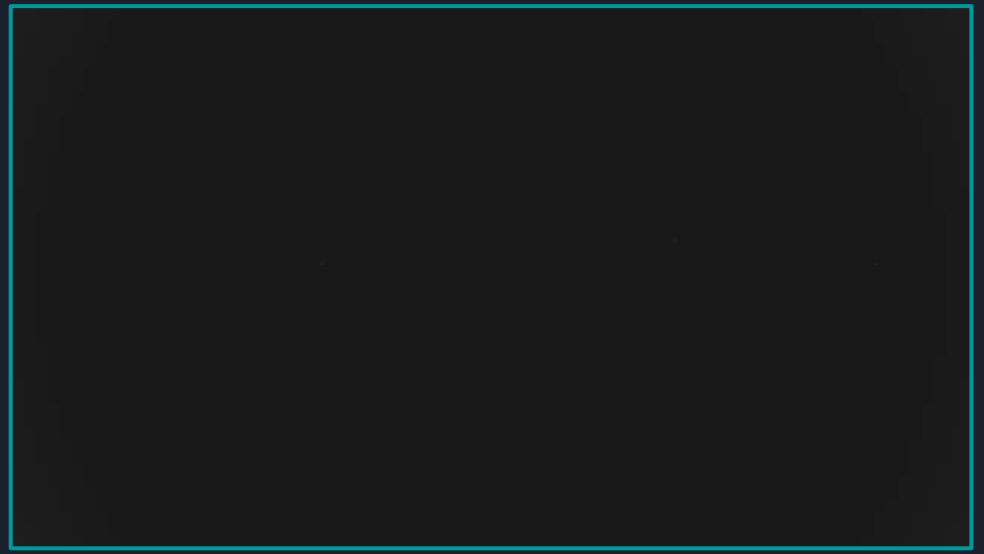
PRESENTATION VIDEO OF “RoverBot”

PRESENTATION VIDEO OF “RoverBot”

Video Purpose :

This video was created to explain how “RoverBot” actually acts in real world.

The capabilities and control of “RoverBot” are thoroughly explained in the video.



Video Link (For PDF Version) :
https://drive.google.com/file/d/1pXUqYOhvQ6C1ip4mzgITkAk81Kv5wtOX/view?usp=sharing

ACKNOWLEDGEMENT

ACKNOWLEDGEMENTS

Thank you for your Attention. It’s an honor to share this project with you. I hope that you enjoy it.

Special Thanks to

My parents
Dear aunt, Dr. Sana Belguith
Dear brothers, Youcef Barraj and Yacine Barraj