People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
M'hamed Bougara University - Boumerdès

Faculty of Engineering Technology

Department of mechanics
engineering

Field : Science and Technologies
Specialty Electromechanics

End of cycle memory in view of obtaining the Bachelor degree

Theme

Wireless controlled car

Bouabdallah Khaled Kamel

Morsli Mohamed

Benlalam Abd Erezak

2019/2020

Abstract

Environment pollutions is due to several causes, one of which is the cars emissions.

To avoid this many solutions have been proposed such as using better fuel or changing the air filter. But those and such helps only to decrease it and not get ride of it completely.

From this comes the idea of the electric cars (also called DC cars). Those of which are easier to control and would not have polluted emissions at all.

In this light we opt to realize a miniature design using Arduino which is wirelessly controllable.

Keywords Arduino, smart car, automatic.

Acknowledgment

In the name of Allah the most powerful the most just the most lenient, we begin our acknowledgments.

We would first like to thank the Almighty and Merciful God, who gave the strength and patience to accomplish this modest work.

In the second place we would like to thank the LMSS team, especially Mr.Touzout Walid for providing the help we needed.

Our sincere thanks also go to the members of the jury for their interest in our research by agreeing to examine our work and enrich it with their proposals.

Finally, we would also like to thank all the people who participated in near or far to the accomplishment of this work.

Hope you enjoy your reading.

Boumerdès,2020.

Contents

List of Figures

Introduction

Nowadays electrics are taking an important place in our daily lives, from coffee machines and dish washer to drills and other.

Smart cars have made appearance since an early time (the first ideas goes back to the 80s) and have been further developed since.

Arduino is a computing platform that provides a micro-controller board for building, it is easy to be connected with electronic components. The code which controls it is C++ based which make it easy to understand and implement.

In this project we have developed a DC car using Arduino for hardware and C++ based code for software.

Our project report is structured as follow:

- **Chapter 1** To introduce the parts used in building the car,
- **Chapter 2** Explains the code that makes the car function,
- **Chapter 3** The diffrents view of the final product,
- **A general conclusion** with future works propositions.

Chapter 1

Hardware system design

1.1 Introduction

Hardware represents the physical part of the products, for our project we have choose to use Arduino and related material to build the physical layer of the car.

1.2 Hardware description

1.2.1 Arduino Uno

The Arduino Uno is an open source microcontroller boar based on the microchip ATmega328P.

The board is equipped with 14 digital input/output (I/O) pins that may be interfaced to various expansion boards and other circuits: 6 analog input pins, a USB connection, a power barred jack, an ICSP header and a reset button.

(The ATmega328P on the Arduino Uno comes preprogramed with a boot loader that allows uploading new codes to it without the use on an external hardware programmer)

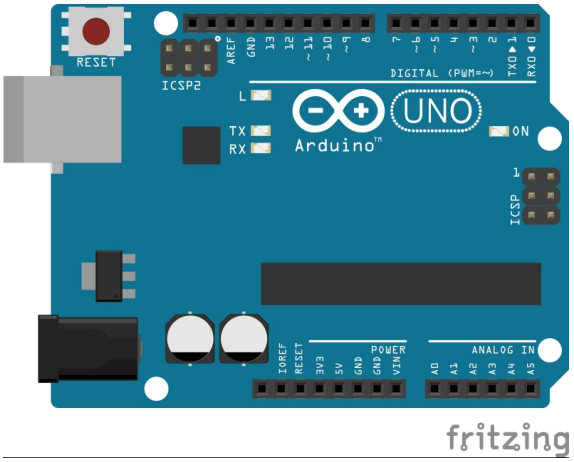

Figure 1.1: Arduino Uno

1.2.2 Arduino Nano

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328. It has more or less the same functionality of the Arduino Uno, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one.

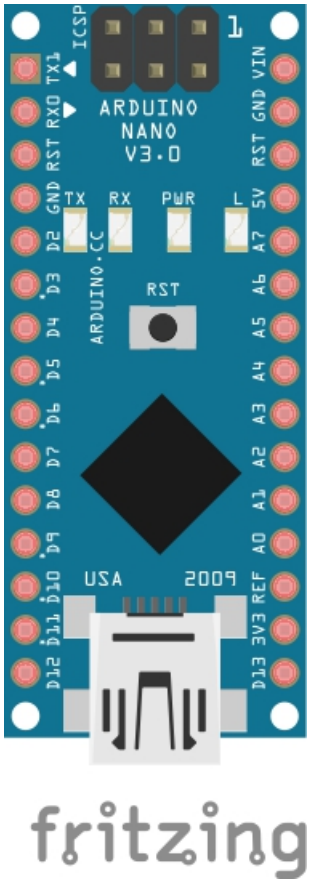

Figure 1.2: Arduino Nano

1.2.3 ADXL335

The ADXL335 is a small, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

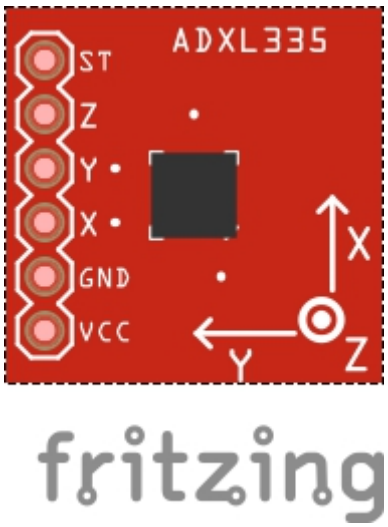

Figure 1.3: ADXL335

1.2.4 L298N driver

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two (or more) DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.

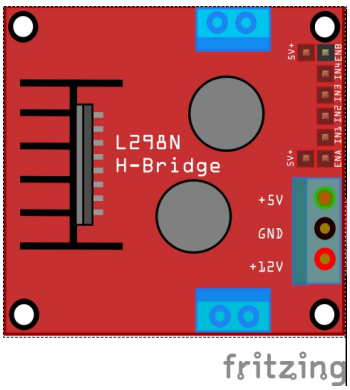

Figure 1.4: L298N driver

1.2.5 NRF24L01

The nRF24L01 transceiver module is designed to operate in 2.4 GHz worldwide ISM frequency band and uses GFSK modulation for data transmission.
The data transfer rate can be one of 250kbps, 1Mbps and 2Mbps.

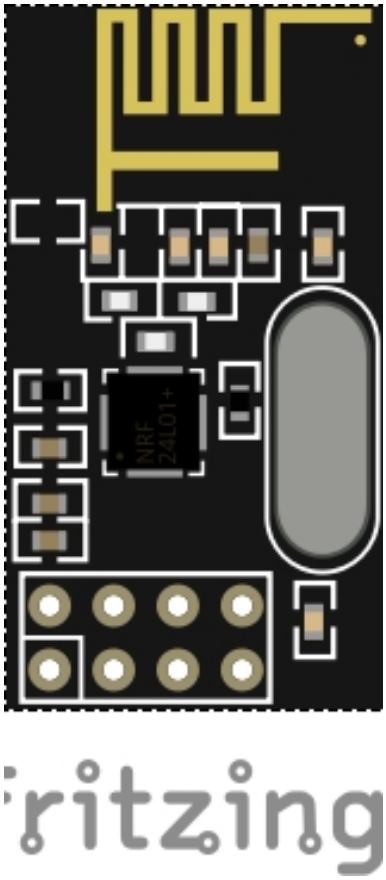

Figure 1.5: NRF24L01

1.2.6 DC motor

DC motor is any of a class of rotary electrical motors that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields.

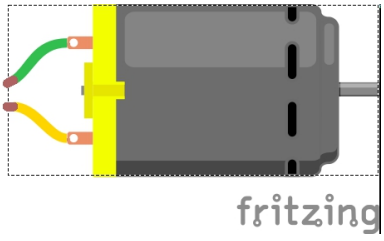

Figure 1.6: DC motor

1.2.7 Joystic

The Analog Joystick is similar to two potentiometers connected together, one for the vertical movement (Y-axis) and others for the horizontal movement (X-axis). The joystick also comes with a Select switch. It can be very handy for retro gaming, robot control, or RC cars.

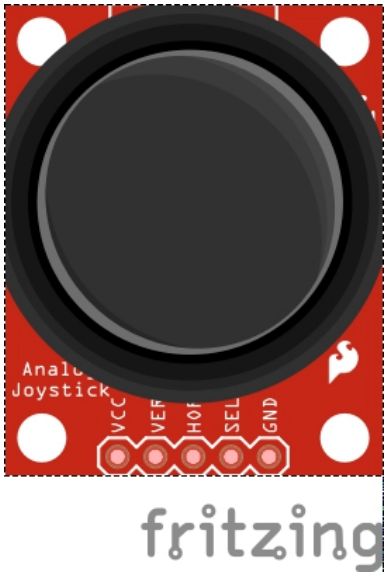

Figure 1.7: Analog Joystick

1.3 The general block diagram circuit

1.4 The overall circuit

1.4.1 Microcontroller 1 (The transmeter)

The first circuit consists of:

- Arduino Nano
- NRF24L01
- ADXL335
- 9v Battery

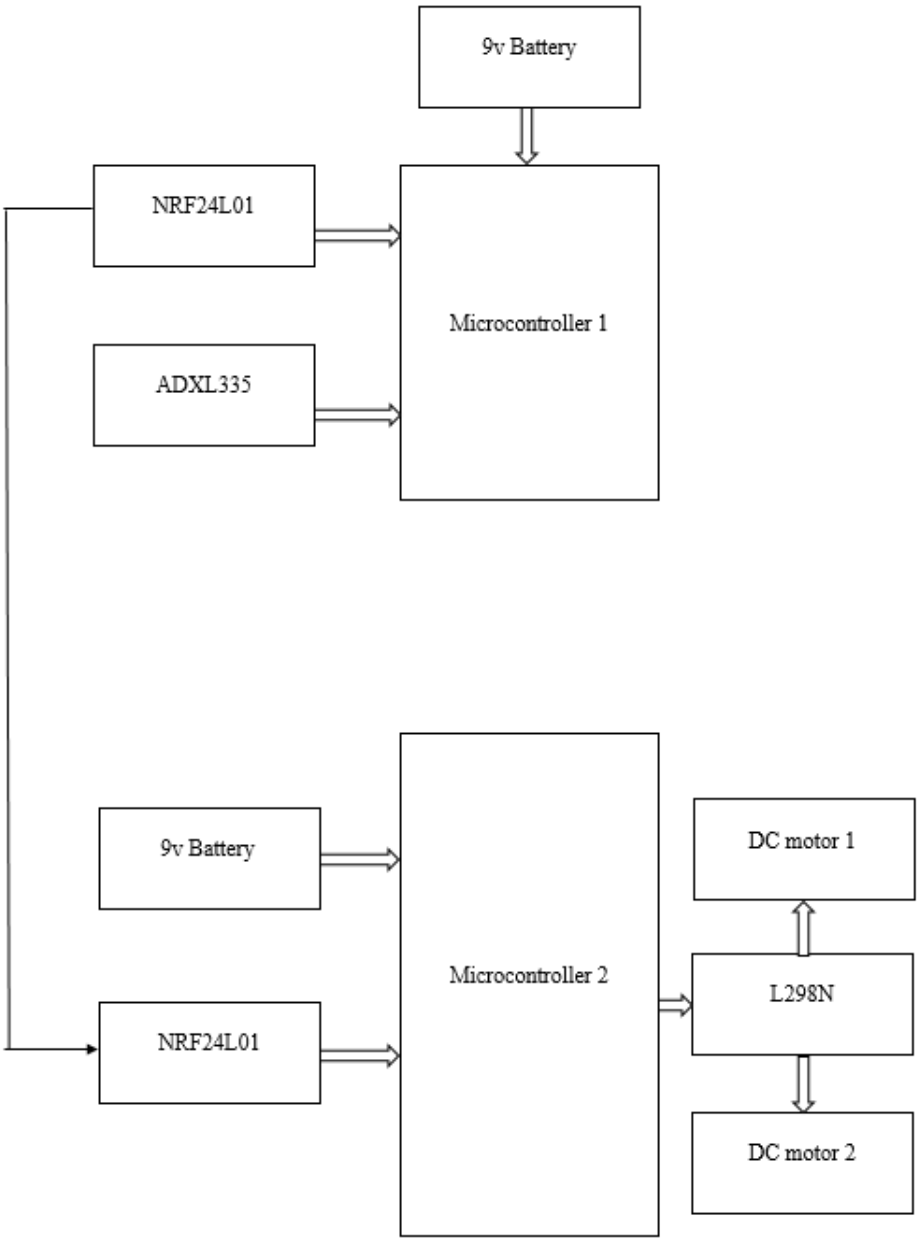

Figure 1.8: The general block diagram

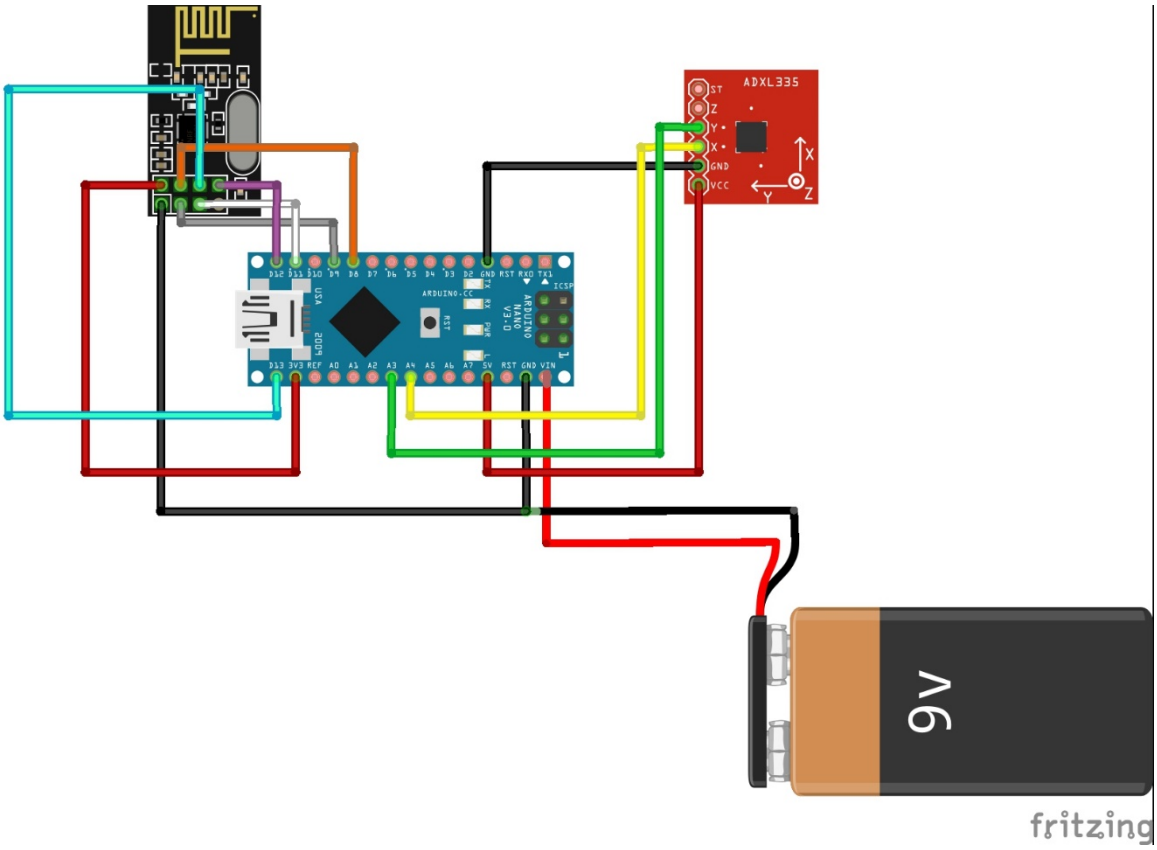

Figure 1.9: The transmeter circuit

1.4.2 Microcontroller 2 (The receiver)

The second circuit consists of almost the same parts with some changes to enable receiving:

- Arduino Nano
- L298N driver
- NRF24L01
- 4 DC motors
- 9v Battery

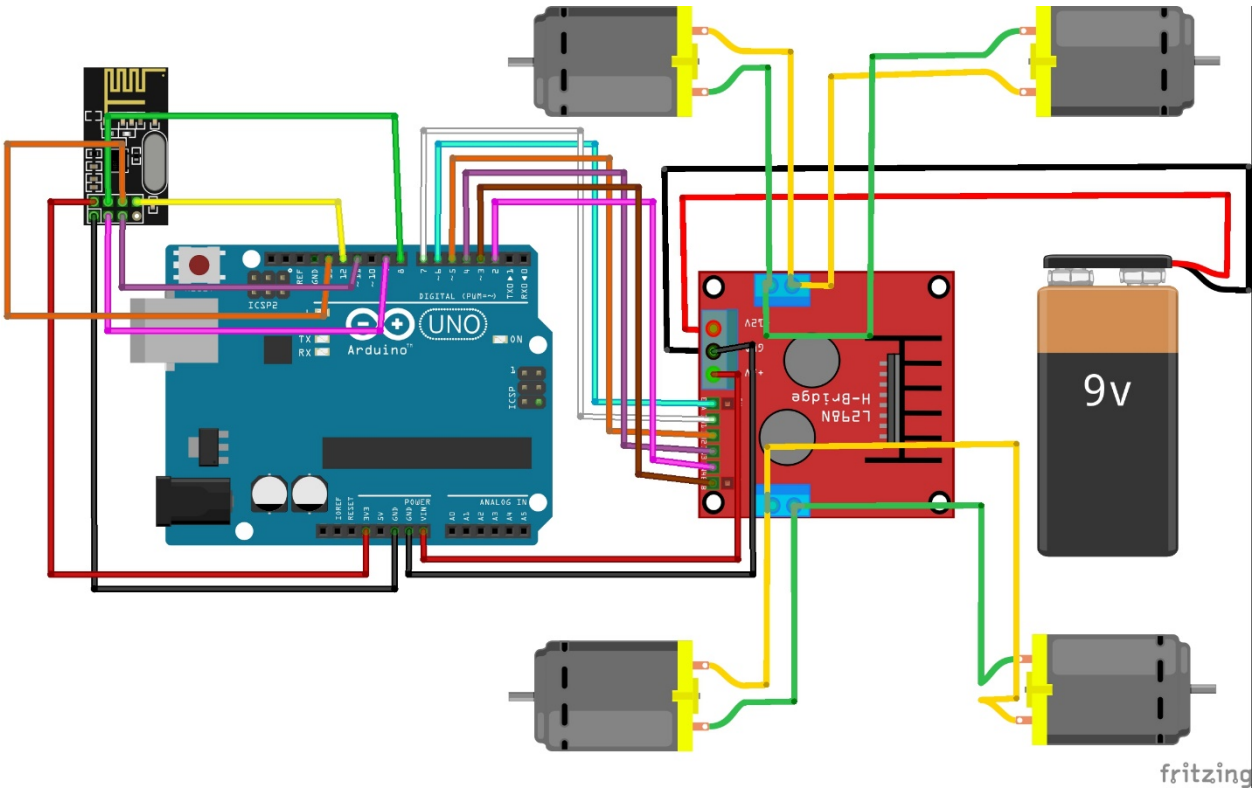

Figure 1.10: The receiver circuit

1.5 Conclusion

In this chapter we've presented the parts we have opt to work with, along-with their description and use. We have chosen those exact part for they respond to our needs and connect better together.
The pictures included of the parts were captured using **FRITZING** program.

The next chapter will cover the software we've used to build the source code.

Chapter 2

Software system design

2.1 Introduction

This chapter is dedicated to the software, we explain the execution steps, how the programs are connected and the different libraries used to build up the code.

To ease understanding the work we've presented it in a flowchart which explains the sequence of functions.

2.2 System overview

Initially, when the hand moves left, right, top or bottom, the accelerometer (ADXL335) measure the coordinates that the hand has moved, those coordinates will be transferred and then will be sent RF NRF24 module, when the receiver (other NRF24) get the information needed, it will send the signals so that the L298n will be able to take action and control which end will receive the current and voltage to the DC motors so the car can move. In order to control the different elements, like the accelerometer or the L298n, two Arduino (an Uno one and a Nano one) microcontrollers are used. The Arduino IDE is an open-source software where the code is written and uploaded to the Arduino boards.

2.3 The used libraries

2.3.1 Serial peripheral interface SPI

SPI is a synchronous serial data protocol used by microcontrollers for the communication with one or more peripheral devices quickly over short distance. It can also be used for communication between two microcontrollers. With an SPI connection there is always one master device which controls the peripheral devices.

2.3.2 RF24.h

It ease the communication between the pins (Miso, Mosi, SCK) and the SPI Arduino pins (uno and nano).

2.4 Software design

2.4.1 Microcontroller 1

Top level flowchart in figure 2.4.1

The program contains two main function: the setup which is executed once for initializing the different sensors, input and output ports and the loop function which executed repeatedly.

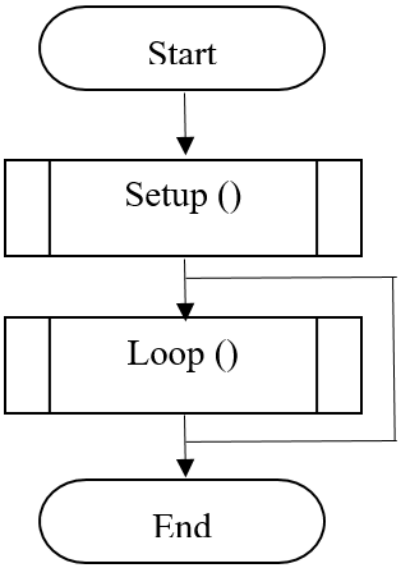

Figure 2.1: Program Flowchart

2.4.1.1 The setup()

The nRF transmeter is initialized, after declaring the pins, if the initialization fails, an error message is display to allow the programmer to fix it.

2.4.1.2 The loop()

The loop subroutine contains the call of the different needed subroutines to read the sensed values, process the data. First the Arduino gets the data from the accelerometer to calculate the coordinates. The calculated value is sent to the nRF so it can be sent the other nRF receiver.

2.4.2 Microcontroller 2

Since we are using basically the same microcontroller, we have the same structure as the first Arduino code (Setup() executed once, loop() executed repeatedly).

2.4.2.1 The setup()

Similarly, to the nRF transemeter, the nRF receiver is initialized after declaring the pins and initializing the serial communication, if the initialization fails, an error message is displayed to allow the programmer to fix it.

2.4.2.2 The loop()

When the serial communication is available, the Arduino is ready to receive data from the module.

2.5 Conclusion

This chapter was dedicated to explaining the structure of the source code and the main functions and libraries used in both programs. The next chapter will present the final product.

Chapter 3

Realization

3.1 Introduction

In this chapter we will present the realized car from different angles to show the used components and how are they connected.

3.2 The car

The figure 3.1 shows a top view of the car components and the wires used to connect them

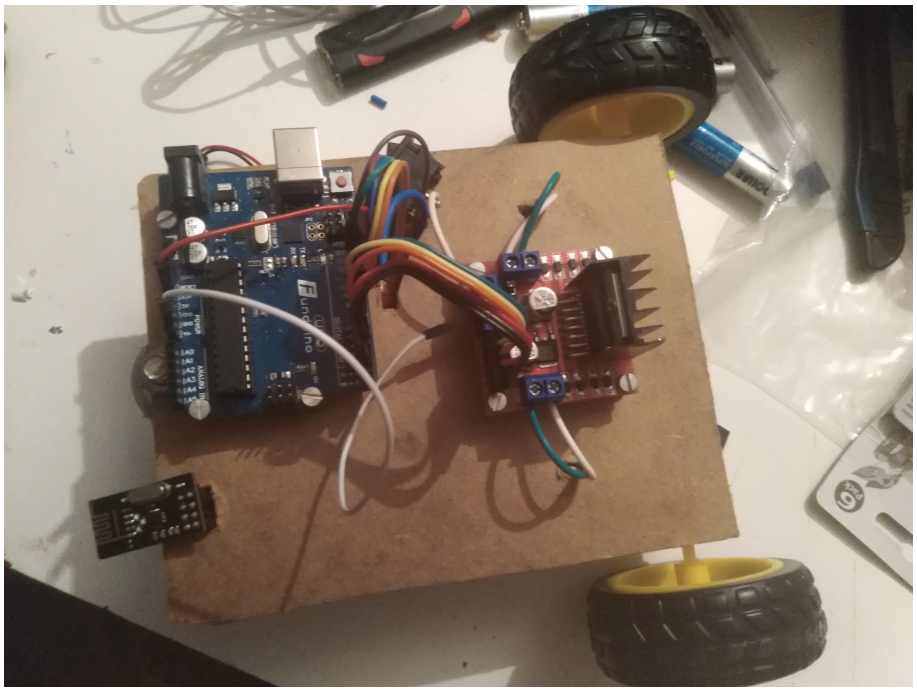

Figure 3.1: Top view of the car

Figure 3.2 shows the batteries insertion.

Figure 3.2: Car batteries

Figure3.3 show the Arduino Uno and its connections

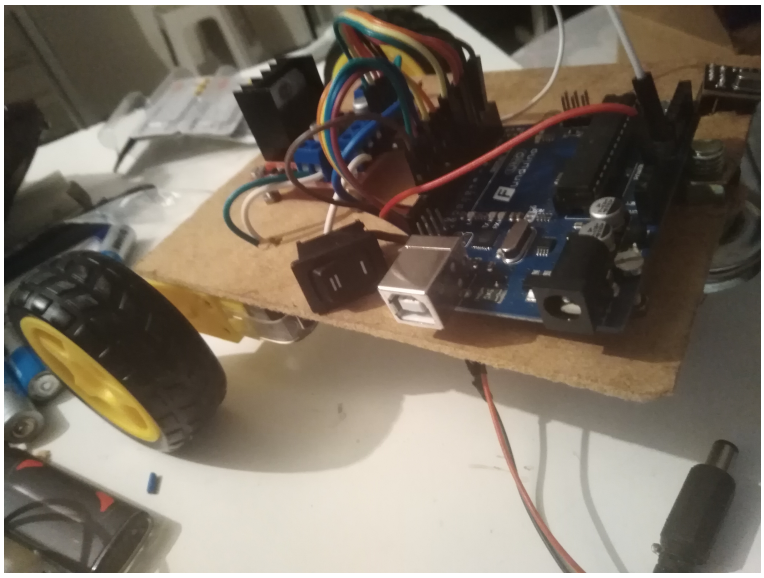

Figure 3.3: Side view of the car

3.3 The controller

Figure 3.4: Transmeter

3.4 Conclusion

We presented this realized product from different views, due to the lack of components we've replaced the ADXL-335 by a joystick.

General Conclusion

Automating has now covered a lot of fields to ease human intervention and provide safety.

Controllable electrical cars uses rechargeable batteries and are controlled from distance.

In this project we have built an electric car from scratches using Arduino related components. The car is controlled either by a joy stick or ADXL-335 by answering to the hand movement. It can move in all directions and uses only 9V battery.

Structuring the report we started by presenting and describing the Arduino parts and other used material, then we defined the instructions which enable the car to move, and finally presented our work from different views.

For further improvements we would:

1. Add a sensor so that the car can detect and halt at any obstacle,
2. Add a lithium battery to the receiver part to increase speed.