People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Tissemsilt

Faculty of Science and Technology
Department of Science and Technology

Graduation thesis for graduation
of an academic Master's degree in

Sector: Electronics
Specialty: Instrumentation

Presented by: **Teiri Khelifa & Belkhir Bilal**

Theme

Remote control of vehicle using atmega328p µc and NRF24L01 module

---

Defended on, /12 /06/ 2023

Before the Jury:

Marich Mohamed	President	MCB	Univ-Tissemsilt
Hamdani Mostefa	Examiner	MCA	Univ-Tissemsilt
Nail Bachir	Supervisor	MCB	Univ-Tissemsilt
Benkheira Ghalem Turki	Co-Supervisor	Doctorant	Univ-Oran

Academic year:2022-2023

Abstract

In this project, we will build a explorer vehicle that can controlled over a distance without the need for physical wires or cables, which consists of two main elements are: the remote control and a vehicle to move in various terrains. At first, we will make printed circuits board using Easy Eda, secondly, we will remotely control the car by wireless communication. To do this we used the most popular wireless communication module the NRF24L01, to connect it to the ATmega328P microcontroller using the SPI interface. We would also need to connect the L298 motor driver to the microcontroller and the DC motors. Once everything is connected, we write a code to send commands from a remote device to the NRF24L01 module, which will then be received by the microcontroller and used to control the motors.

At the end, we will control the vehicle by sending data from the remote that include the speed and the direction which it gets from the Joystick and the MPU6050 and the buttons.

Key word:

Vehicle, Remote Control, NRF24L01, L298, MPU6050, Joystick, Dc motor

Résumé

Dans ce projet, nous construirons un véhicule explorateur qui peut être contrôlé à distance sans avoir besoin de fils ou de câbles physiques, qui se compose de deux éléments principaux : la télécommande et un véhicule pour se déplacer sur divers terrains. Dans un premier temps, nous fabriquerons des circuits imprimés en utilisant EasyEda, deuxièmement, nous contrôlerons à distance la voiture par communication sans fil. Pour ce faire, nous avons utilisé le module de communication sans fil le plus populaire, le NRF24L01, pour le connecter au microcontrôleur ATmega328P à l'aide de l'interface SPI. Nous aurions également besoin de connecter le pilote de moteur L298 au microcontrôleur et aux moteurs à courant continu. Une fois que tout est connecté, nous écrivons un code pour envoyer des commandes d'un appareil distant au module NRF24L01, qui seront ensuite reçues par le microcontrôleur et utilisées pour contrôler les moteurs.

À la fin, nous contrôlerons le véhicule en envoyant des données à partir de la télécommande qui incluent la vitesse et la direction qu'il obtient du joystick et du MPU6050 et des boutons.

Mot-clé :

Véhicule, télécommande, NRF24L01, l298, mpu6050, joystick, moteur à courant continu

ملخص

في هذا المشروع، سنقوم ببناء سيارة يمكن التحكم فيها عن بعد دون الحاجة إلى أسلاك أو كابلات، والتي تتكون من عنصرين رئيسيين هما: جهاز التحكم عن بعد وسيارة للتحرك في تضاريس مختلفة. في البداية، سنصنع لوحة الدوائر المطبوعة باستخدام Easy Eda ثانيا، سنتحكم في السيارة عن بعد عن طريق الاتصال اللاسلكي. للقيام بذلك استخدمنا وحدة الاتصالات اللاسلكية الأكثر شعبيةNRF24L01 ، لتوصيله إلى المتحكم الدقيق ATMEGA.328P سنحتاج أيضا إلى توصيل L298 بالمتحكم الدقيق ومحركات التيار المستمر. بمجرد توصيل كل شيء، نكتب برنامجا لإرسال الأوامر من جهاز بعيد إلى NRF24L01، والتي سيتم استلامها بعد ذلك بواسطة المتحكم الدقيق واستخدامها للتحكم في المحركات.

في النهاية، سوف نتحكم في السيارة عن طريق إرسال البيانات من جهاز التحكم عن بعد والتي تتضمن السرعة والاتجاه الذي تحصل عليه من عصا التحكم وMPU 6050و الأزرار.

الكلمات المفتاحية :

السيارة، جهاز التحكم عن بعد، تردد الراديو، L 298وحساس الحركة MPU6050، عصا التحكم، محرك تيار مستمر

Appreciation

First of all, we would like to thank God the Almighty for giving us courage, will and patience throughout our studies

Our special thanks go to our supervisor Dr. NAIL BACHIR for his great availability and wise advice during our preparation of this project.

Many thanks to all members of the jury for agreeing to review and evaluate our work.

We express our deep gratitude and thanks to our professors Mrs. CHABAH, Mr. BERBARA, Mr. MEHRAR to provide assistance.

And all our teachers for the quality of

teaching they have given us during these five years spent at the university of Tissemsilt.

Thank you to all my colleagues from the electronic section.

Finally, we send our heartfelt thanks to our families, our friends for their support and encouragement throughout the realization of this work.

Dedication

Praise be to Allah who commands goodness and mutual assistance in piety. May His peace and blessings be upon the last of the prophets.

I dedicate this modest work to:
My dear parents, who have made numerous sacrifices and continue to do so to see me succeed. They have ensured my education and supported me throughout these years. This work is the result of their strict upbringing, and I hope they find in it the expression of my deep love and gratitude.

To my brother Sid Ali and my sisters Zohra, Naima, and Fatiha. Sara

To all my friends Ghani, Radouane, Fodhil, Lotfi, Sid Ahmed, Youcef, Bilal.

I would also like to express my warmest thanks to my teammate Bilal, who shared this humble work with me. Without him, I would not have been able to accomplish this task.

And to the entire Instrumentation 2019-2023 class.

Khelifa

Dedication

To my dear parents, for all their sacrifices, their love, their tenderness, their support and their prayers throughout my studies,

To my dear brothers, for their support and encouragement,

To all my family for their support throughout my university journey,

May this work be the fulfillment of your long-alleged wishes, and the flight of your infallible support,

to my friend khelifa with whom I shared pleasant moments

To all my friends Houcine, Khaled , Abderazzek ,Riadh , Houari Abdelkader , Nourdine , Mesbah , Arafet and Khelifa

Thank you for always being there for me.

Bilal

Contents

Chapter 2 Wireless Communication

Chapter 3 Simulation and Realization

General Conclusion

Annex

Bibliography

List of Tables

List of Figures

Glossary

WI-FI: Wireless Fidelity

WLAN: Wireless Local Area Network

GPS: Global Positioning System

RF: Radio Frequency

PA : Power Amplifier

PLA : Low Noise Amplifier

RX: Receiver Module

TX: Transmitter Module

SPI: Serial Peripheral Interface

CSK: Serial Clock

SCL: Serial Clock Line

IRQ: Interrupt Request

CE: Chip Enable

CSN: Chip Select Not

MOSI: Master Output Slave Input

MISO: Master Input Slave Output

PSK: Phase-Shift Keying

FSK: Frequency-Shift Keying

QPSK: Quadrature Phase-Shift Keying

TDM: Time Division Multiplexing

FDM: Frequency Division Multiplexing

SDA: Serial Data Line

VCC: Positive Power Supply

VDD: Voltage Drain Drain

VIN: Voltage Input

GND: Ground

DPS: Digital Power Supply

DC: Direct Current

IDE: Integrated Development Environment

ICSP: In-Circuit Serial Programming

USB: Universal Serial Bus

I2C: Inter-Integrated-Circuit

SRAM: Static Random Acces Memory

EEPROM: Electrically Erasable Programmable Read Only Memory

PWM: Pulse Width Modulation

USART: Universal Synchronous/Asynchronous Receiver -Transmitter

PCB: Printed Circuit Board

EN: Enable Pin

EDA: Electronic Design Automation

CAD: Computer Aided Design

Kbps: Kilobits Per Second

Mbps: Mégabits Per Second

SMPS: Switched Mode Power Supply

HVAC: Heating Ventilation and Air conditioning

SCSI: Small Computer System Interface

SOIC: Small Out Line Integrated Circuit

General introduction

In recent years, robots have experienced a remarkable development, which has made them an absolute necessity in everyday and practical life, where we find it today in various fields such as health, the armed, the farmer, exploration...etc., This is to replace humans in operations that are usually repetitive and dangerous. Today researchers are much more interested in mobile exploration robots, because they are widely used especially in terrestrial explorations.

the tendency to use mobile robots in various applications is increasing daily. One of the challenges we need to face when using a robot mobile is navigation on different terrains. There the concept of multi-terrain robots appeared. The objective of this work is to build a mobile multi-terrain explorer vehicle, whose design is based on a car controlled by a radio frequency control, the remote is composed of four buttons, a joystick, a potentiometer, a gyroscopic sensor MPU6050, a microcontroller and a radio frequency transmitter NRF24L01+ PA+ LNA. The robot consists of an Arduino nano board, a two DC motor and an L298N module and a receiver NRF24L01+ PA+ LNA.

Since Arduino is an open source, The Uses of this car are unlimited, depending on the program applied to it, as it is programmed as needed, and in the field to be worked on , and among the things that motivated us to make this car is to help lay internet wires through narrow ceilings, as well as explore the places of blockages in the ground pipes .

This memoir contains three chapters:

Chapter 01:

The first chapter contains general information about Arduino and components suitable for our application.

Chapter 02:

In this chapter, we dealt with the aspect of wireless communications, in particular the NRF24L01.

Chapter 03:

in the last chapter we dealt with how to design the electronic board for controlling the vehicle, and then we tested the various components used, then we assembling the vehicle and writing the various program related to controlling the vehicle.

Chapter 1

Arduino and components used

1 Introduction

Programming was a very difficult thing due to the lack of capabilities and the difficulty of the matter. But the recent emergence of Arduino has led to a major revolution in the world of programming due to its ease of us. The Arduino IDE is very simple and this simplicity is probably one of the main reason Arduino became so popular. We can certainly state that being compatible with the Arduino IDE is now one of the main requirements for a new microcontroller board. Over the years, many useful features have been added to the Arduino IDE and you can now manage third-party libraries and boards from the IDE, and still keep the simplicity of programming the board.

2 Definition

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message and turn it into an output - activating a motor, turning on an LED, publishing something online.

You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing [1].

The Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program [2].

As Arduino is open source, there are a large number of compatible Arduino boards, just as there are many official Arduino boards with special functions as shown below [3.4] . Arduino Uno is the most popular

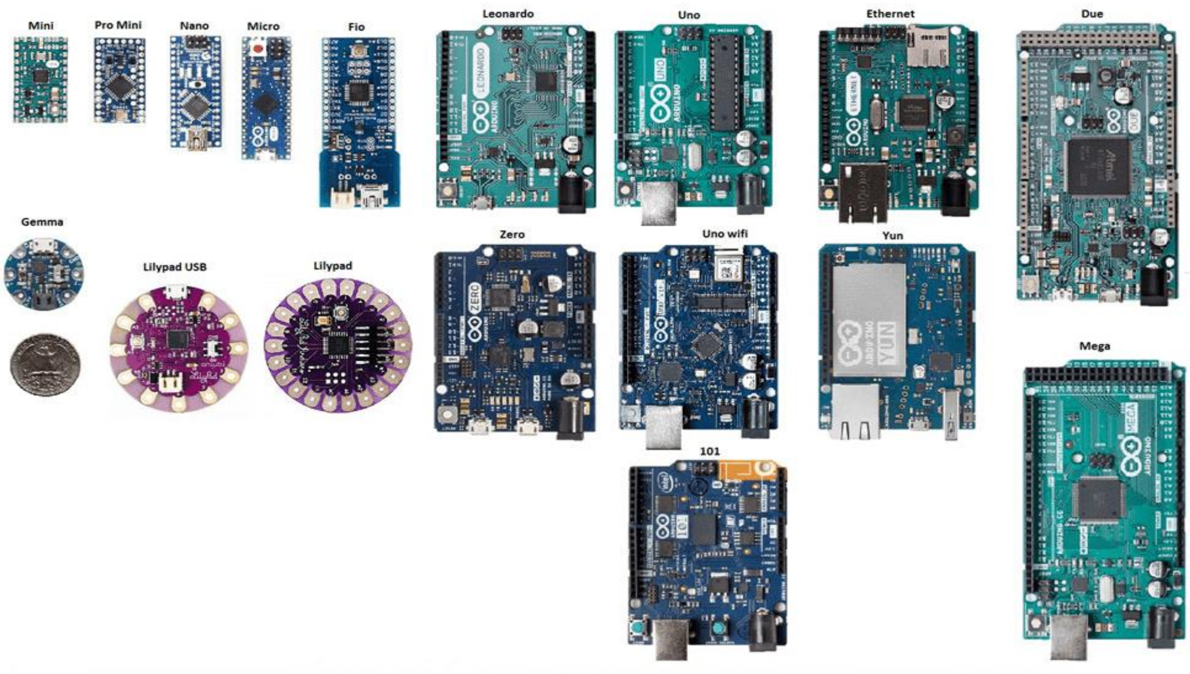

Figure 1.1: Types of Arduino Boards [3.4]

In our project we used two types of Arduinos Uno and nano:

2.1 Arduino Uno

The Arduino Uno board is composed of 14 digital input/output pins, in general six pins are available in PWM (3,5,6,9,10,11), and 6 pins of analog inputs (A0, A1, A2, A3, A4, A5), a USB connection, a power supply connection, a ICSP port and a RESET button [5].

The description of all the pins of the Arduino Uno board is presented in the picture below:

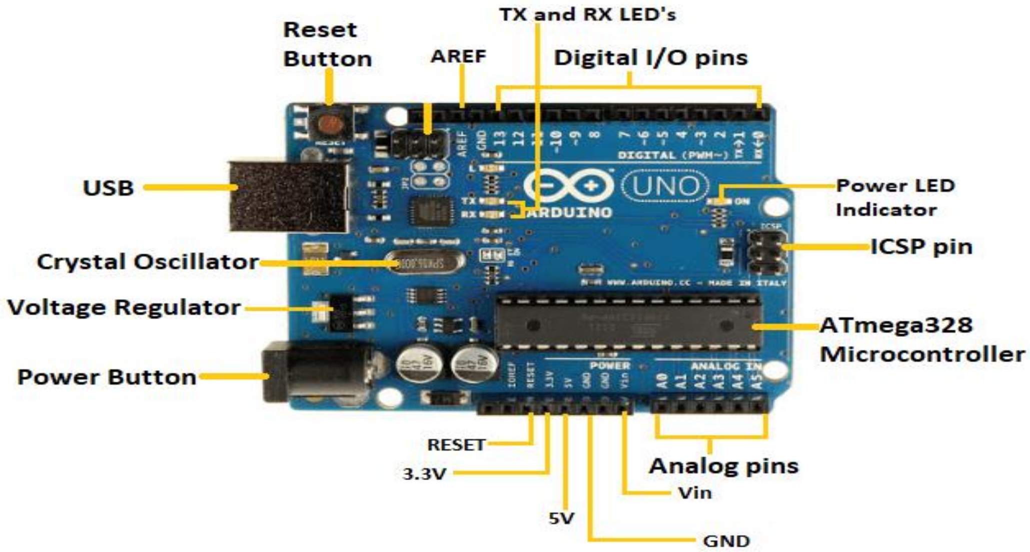

Figure 1.2: Arduino uno [5]

2.1.1 Specification

Table 1.1: Arduino Uno specifications [5]:

Microcontroller	Atmega 168
Operating Voltage (logic level)	5V
Input Voltage(recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14
Analog Input Pins	6
DC Current per I/O Pin	40 mA
Dc Current for 3.3V pin	50 mA
Flash Memory	32 KB (Atmega 328)
SRAM	2 KB (Atmega 328)
EEPROM	1KB (Atmega 328)
Clock Speed	16MHz

2.2 Arduino Nano

Arduino Nano is a small complete chip board based on Atmega 328 (v3.0) or Atmega 168 (v2.0). Every Arduino has the same functionality and the same features except the number of pins and size. One of the major flaws of this board is that it doesn't have any power jack. So, you can't supply power from any external power source like a battery. This board is quite similar to any Arduino board [6] developed by Arduino.cc in Italy in 2008 and contains 30 male I/O headers, configured in a DIP30 style [7].

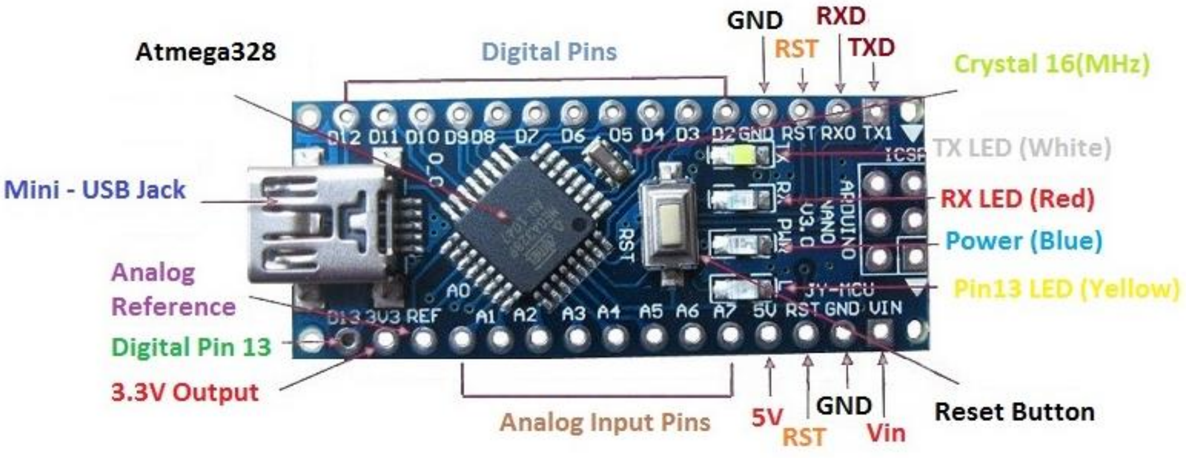

Figure 1.3.1: Layout of Arduino Nano Board [6]

you can use a power source using the Vin Pin, but you need to take a few precautions. Otherwise, it will burn the board [8] ,This Nano board is different in packaging. The power supply can be give using a small USB port otherwise straight connecting to the pins like VCC and GND [9].

2.2.1 Arduino Nano Pinout

- contains 14 digital pins, 8 analog Pins, 2 Reset Pins & 6 Power Pins.
- The Nano has 8 analog inputs, each of which provide 10 bits of resolution (1024 different values). By default, they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analog Reference () function. Additionally, some pins have specialized functionality.
- It comes with an operating voltage of 5V; however, the input voltage can vary from 7 to 12V.
- Arduino Nano's maximum current rating is 40mA, so the load attached to its pins shouldn't draw current more than that.
- Each of these Digital & Analog Pins is assigned with multiple functions but their main function is to be configured as Input/Output.
- Arduino Pins are acted as Input Pins when they are interfaced with sensors, but if you are driving some load then we need to use them as an Output Pin.
- Functions like pinMode() and digitalWrite() are used to control the operations of digital pins while analogRead() is used to control analog pins.
- Arduino Nano comes with a crystal oscillator of frequency 16 MHz It is used to produce a clock of precise frequency using constant voltage.

- This board doesn't use standard USB for connection with a computer, instead, it comes with Type-B Micro USB [10].

The following figure shows the pinout diagram of the Arduino Nano board [11].

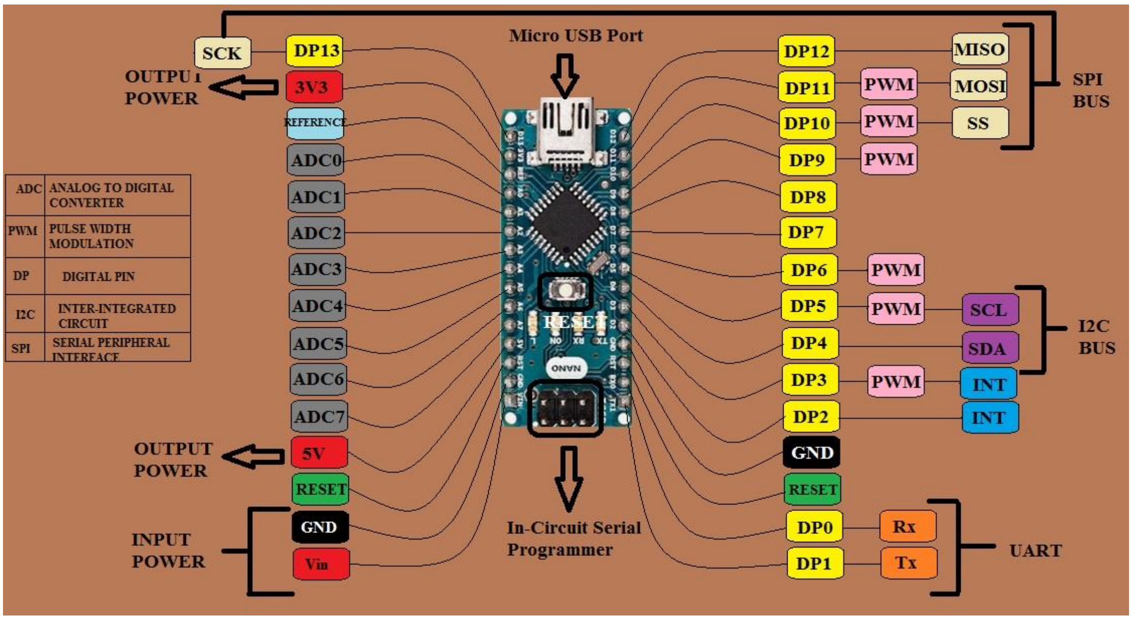

Figure 1.3.2: Arduino NANO Pinout Diagram [11]

2.2.2 Specifications

Table 1.2: Arduino Nano specifications [12]:

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage(recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 MA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70". Length: 45 mm Width: 18 mm

2.2.3 Arduino Nano Pin Description

In this section, we'll cover the Arduino Nano Pinout, we will discuss pin description of each pin integrated on the board[12].

2.2.3.1 Arduino Nano Power Pins

VIN: This is an input voltage to the Arduino board when using an external power source (6-12V).

3.3V: It is a minimum voltage produced by the voltage regulator on the board.

5V: Regulated power supply used to power up the controller and other components on board.

GND: Two ground pins are available on the board.

2.2.3.2 Arduino Nano Function Pins

LED: The unit comes with a built-in LED connected to pin 13 on the board. This LED is used to check the board if it's working fine or not. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

AREF: It is an Analog Reference that is applied to the unit as a reference voltage from an external power supply.

Reset: Two reset pins are integrated on the board. These pins are used to reset the controller internally through software [12].

2.2.3.3 Arduino Nano I/O Pins

Digital Pins: There are 14 digital pins on board which is used to connect external component.

Analog Pins: 6 analog pins on board that is used to measure voltage in a range from 0 to 5V.

External Interrupts: Pin 2 and 3 are used to trigger external interrupts. These pins are used in case of emergency, when we need to stop the main program and call important instructions. The main program resumes once interrupt instruction is called and executed.

PWM Pins: Arduino Nano has 6 PWM pins, which are Pin#3, 5, 6, 9, 10 and 11. (All are digital pins)

These pins are used to generate an 8-bit PWM (Pulse Width Modulation) signal [12].

2.3 Arduino Pinout for Communication Protocols

USART: The board supports USART serial communication that carries two pins Rx which is used for receiving the serial data and Tx which is a transmission pin used to transmit serial data.

I2C: The unit comes with an I2C communication protocol where two pins SDA and SCL are used to support this communication. It’s developed using A4 and A5 pins, where A4 represents the serial data line (SDA) which carries the data and A5 represents the serial clock line (SCL) used for data synchronization between the devices on the I2C bus. The Wire Library of Arduino Software can be accessed to use the I2C bus.

SPI Protocol: Four pins 10(SS=Slave Select), 11(MOSI =Master Out Slave In), 12(MISO =Master In Slave Out) and 13(SCK = Serial Clock) are used for SPI (Serial Peripheral Interface) Protocol.

SPI is an interface bus and is mainly used to transfer data between microcontrollers and other peripherals like sensors, registers, and SD cards [12].

2.4 Arduino Programming & Communication

- The Nano board comes with the ability to set up communication with other controllers and computers.
- The serial monitor is added to the Arduino IDE, which is used to transmit textual data to or from the board.
- FTDI drivers are also included in the software which behaves as a virtual Com port to the software. The Tx and Rx pins come with an LED which blinks as the data is transmitted between FTDI and USB connection to the computer.
- Arduino Software Serial Library is used for carrying out serial communication between the board and the computer.
- The Arduino Nano is programmed by Arduino Software called IDE which is a common software used for almost all types of board available. Simply download the software and select the board you are using.
- Uploading code to Arduino Nano is quite simple, as there’s no need to use any external burner to compile and burn the program into the controller and you can also upload code by using ICSP (In-circuit serial programming header).
- Arduino board software is equally compatible with Windows, Linux or MAC; however, Windows are preferred to use [13].

2.5 Arduino Coding Environment and basic tools

2.5.1 What language is Arduino?

Arduino code is written in C++ with an addition of special methods and functions, which we'll mention later on. C++ is a human-readable programming language. When you create a 'sketch' (the name given to Arduino code files), it is processed and compiled to machine language.

2.5.2 Arduino IDE

The Arduino Integrated Development Environment (IDE) is the main text editing program used for Arduino programming. It is where you'll be typing up your code before uploading it to the board you want to program. Arduino code is referred to as sketches [14].

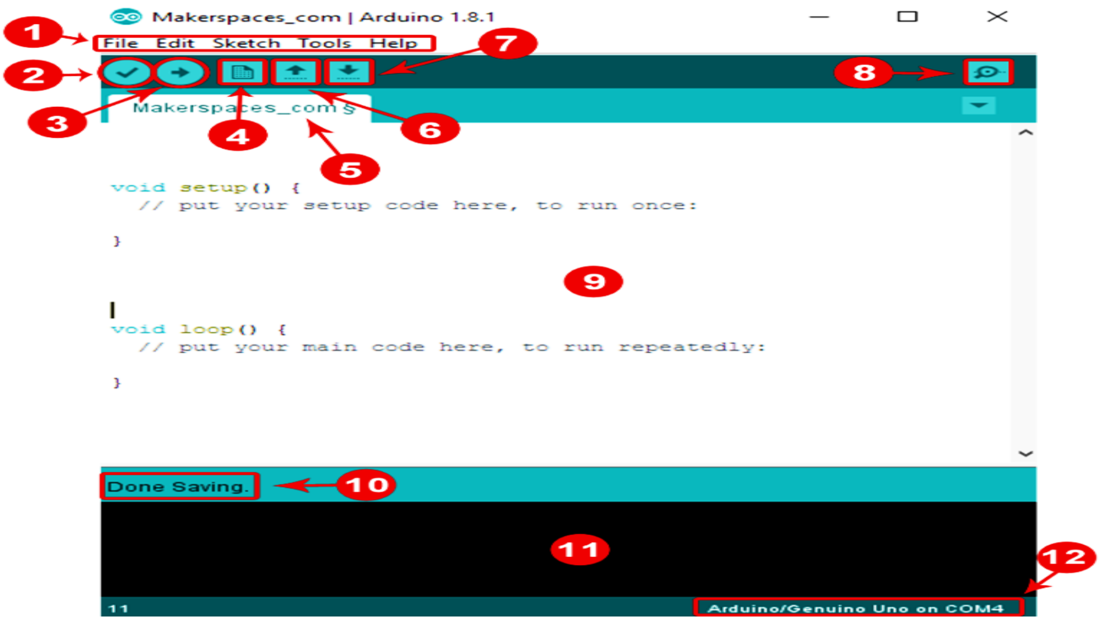

Figure 1.4.1: Arduino IDE [14]

1. **Menu Bar:** Gives you access to the tools needed for creating and saving Arduino sketches.
2. **Verify Button:** Compiles your code and checks for errors in spelling or syntax.
3. **Upload Button:** Sends the code to the board that's connected such as Arduino Nano in this case. Lights on the board will blink rapidly when uploading.
4. **New Sketch:** Opens up a new window containing a blank sketch.
5. **Sketch Name:** When the sketch is saved, the name of the sketch is displayed here.

6. **Open Existing Sketch:** Allows you to open a saved sketch or one from the stored examples.
7. **Save Sketch:** This saves the sketch you currently have open.
8. **Serial Monitor:** When the board is connected, this will display the serial information of your Arduino
9. **Code Area:** This area is where you compose the code of the sketch that tells the board what to do.
10. **Message Area:** This area tells you the status on saving, code compiling, errors and more.
11. **Text Console:** Shows the details of an error messages, size of the program that was compiled and additional info.
12. **Board and Serial Port:** Tells you what board is being used and what serial port it's connected to [15].

2.5.3 The Programing Structure of an Arduino Sketch

The structure of an Arduino sketch consists of 3 main sections as explained below:

2.5.3.1 Declaration Section:

This section used for declare variables, constants, Pin Numbers and to include Libraries (C++ header files).

2.5.3.2 Setup Section:

In this section, the setup () function is used to initialize serials, configure Pin mode, Pin Numbers, using Libraries …etc. This function will only run once after each power up or reset of the Arduino board.

2.5.3.3 Loop Section:

The main program of the sketch will run in this section. The Loop () Function will run infinitely [16].

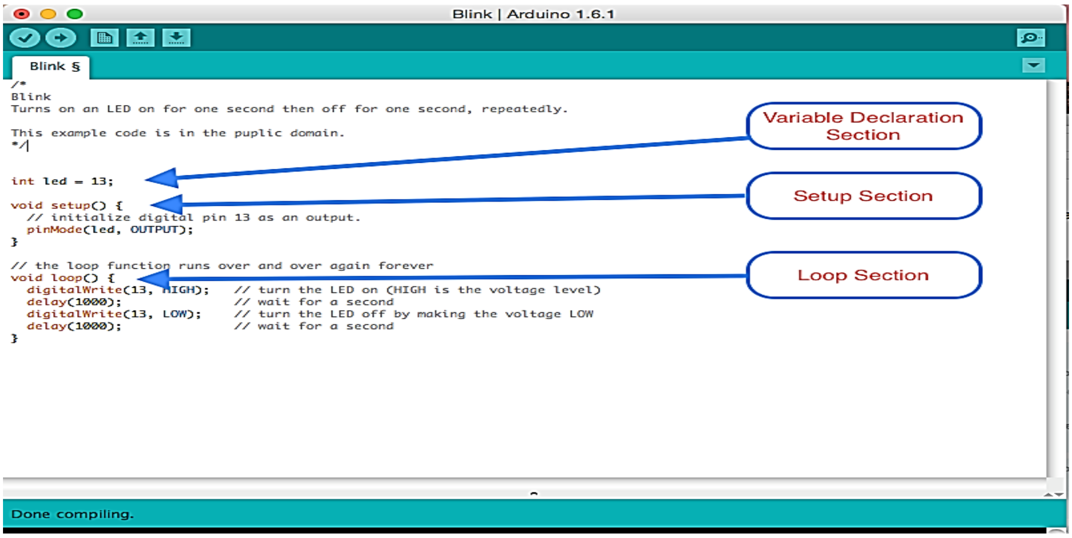

Figure 1.4.2: Arduino Program structure [16]

2.5.4 Choose the Correct Arduino Board

If this is the first time you've used Arduino IDE or the first time you've built code for a Nano, we need to set the Arduino environment so it knows to build the code for the specific board Just go to the **Tools** menu at the top of the IDE and then slide down to the **[Board >]** menu item, when you slide over that item a menu will pop out with a large number of choices. Slide down the Arduino Nano and click it to select it. [17]

The IDE should automatically select the "Processor: ATMega328P" for that board, because it

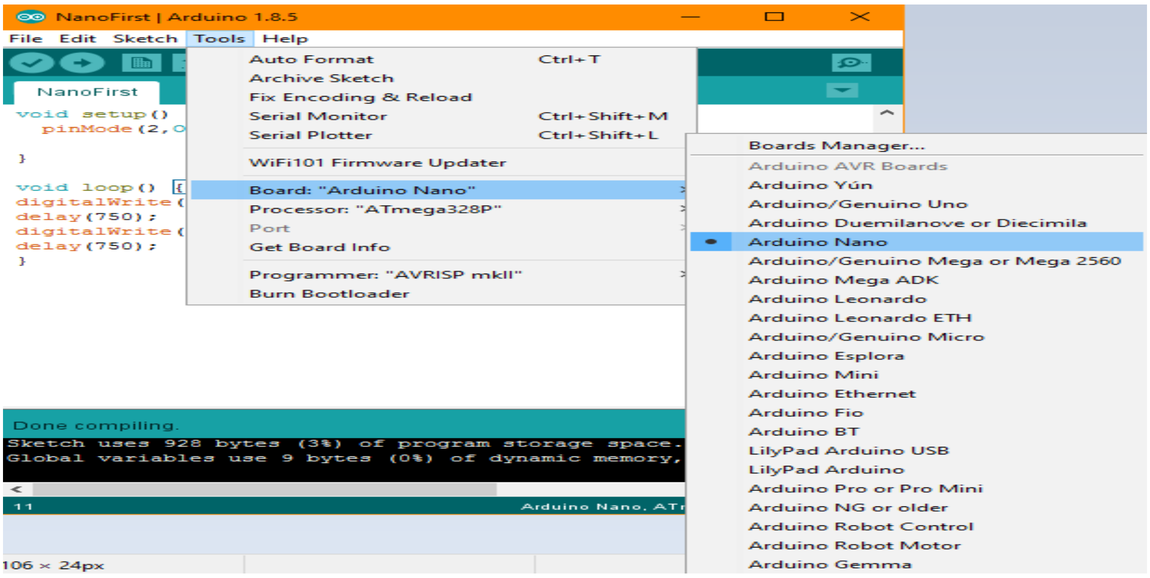

Figure 1.4.3: Choose the Correct Arduino Board [17]

is the newer processor used on these boards. However, you want to make sure it is selected properly.

The next image shows you how to select the proper processor.

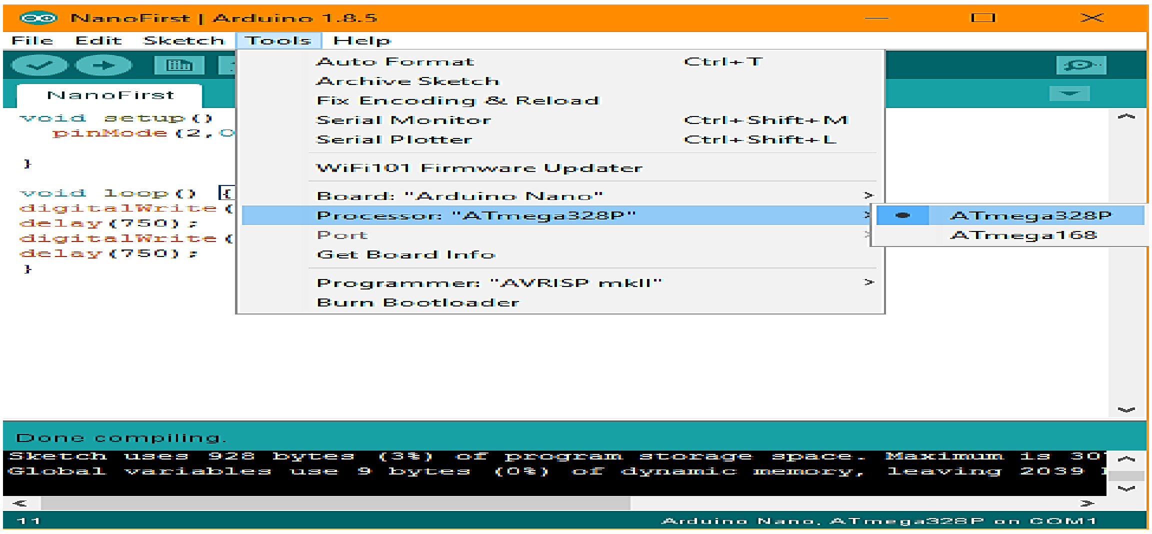

Figure 1.4.4: Select the right processor for Arduino Nano [17]

These two options ensure that the IDE will build the code that is correct for your board. Notice that the bottom status area of the IDE (shown in previous image) confirms that you are building for the Arduino Nano with an ATMega328P and the device is connected to COM1 [17]

2.6 Difference between Arduino UNO and Arduino Nano

The main difference between these two is the size. Because Arduino Uno size is double to nano board. So, Uno boards use more space on the system. The programming of UNO can be done with a USB cable where as Nano uses the mini-USB cable. The main differences are listed in the following table [18,19].

Table 1.3: Difference between Arduino UNO and Arduino Nano [18.19]:

Name	Arduino nano	Arduino uno
MCU	Atmega329p/atmega168	Atmega328p
Power	5V	5V
Input voltage	7V-12V	7V-12V
Maximum current rating	40 mA	40 mA
Clock frequency	16 MHz	16MHz
Flash memory	16 KB / 32KB	32KB
USB	Mini	standard
USART	Yes	yes
SRAM	1KB / 2KB	2kB
PWM	6 out of 14 digital pins	6 out of 14 digital pins
GPIO	14	14
Analog pins	8	6
EEPROM	512bytes / 1KB	1KB

3 General Description of The Components Used

3.1 The MPU6050 Accelerometer

The IMU (Inertial Measurement Unit), MPU-6050 is a printed circuit board contains a three-axis MEMS gyroscope and a three-axis MEMS accelerometer, which allows it to measure linear and angular accelerations, it is often used a standard I2C bus to the transmission of data [20].

3.1.1 MPU6050 technical specifications

- Support for I2C communications: up to 400 kHz
- User programmable gyroscope, full scale range of ±250, ±500, ±1,000 and ±2,000°/s (dps)
- User programmable accelerometer, full scale range of ±2 g, ± 4 g, ±8 g, and ±16 g
- VLOGIC/VDD: 1.8 V ±5 %
- Current of use of gyroscope and accelerometer: 3.8 mA.
- power supply current in full standby mode: 5 mA [21].

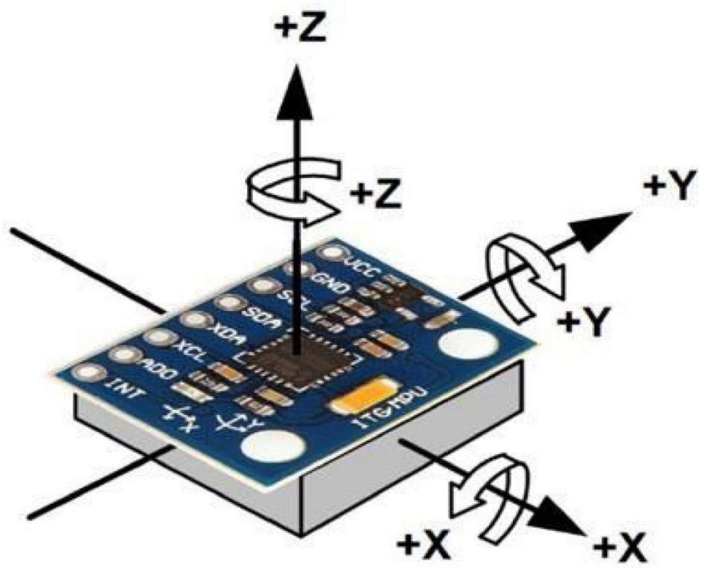

Figure 1.5: MPU6050 sensor [21]

3.2 Joystick

The joystick looks like two potentiometers connected together, one for the movement on the X axis and the other on the Y axis, it is often used to control vehicles RC.

The joystick arm is positioned in the center thanks to a spring at the point (x, y) =(512,512), if the arm moves in the positive direction of the X and Y axes the torque value (x, y)tends to (1023,1023), and if it moves in the negative direction the value of (x, y) tends to (0,0).

3.2.1 Pin Configuration

- GND: ground
- +5V: 5V DC
- VRx: voltage proportional to x position
- VRy: voltage proportional to y position
- SW: switch pushbutton [22].

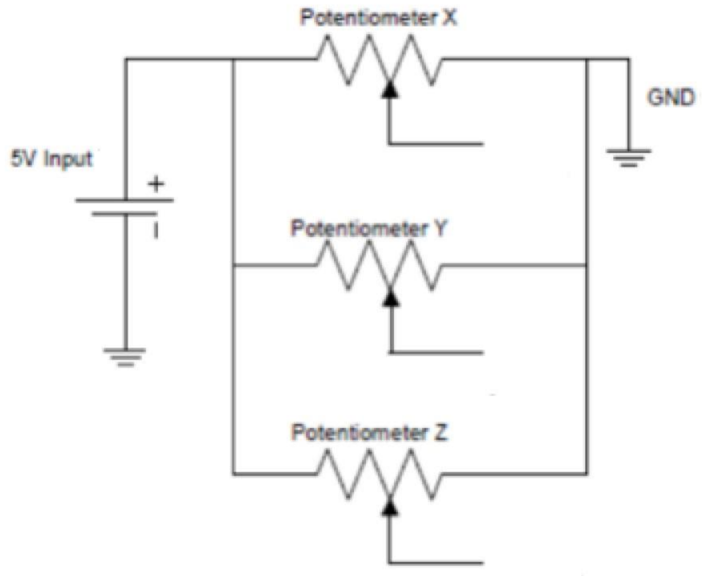

Figure 1.6: Schematic Diagram of joystick [22]

3.3 LM7805

The LM340 and LM7805 Family monolithic 3-terminal positive voltage regulators employ internal current- limiting, thermal shutdown and safe-area compensation, making them essentially indestructible. If adequate heat sinking is provided, they can deliver over 1.5-A output current. They are intended as fixed voltage regulators in a wide range of applications including local (on-card) regulation for elimination of noise and distribution problems associated with single-point regulation. In addition to use as fixed voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents. Considerable effort was expended to make the entire series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the output, this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply [23].

3.3.1 Applications

- Industrial Power Supplies
- SMPS Post Regulation
- HVAC Systems
- AC Inventors
- Test and Measurement Equipment
- Brushed and Brushless DC Motor Drivers
- Solar Energy String Inventors [23].

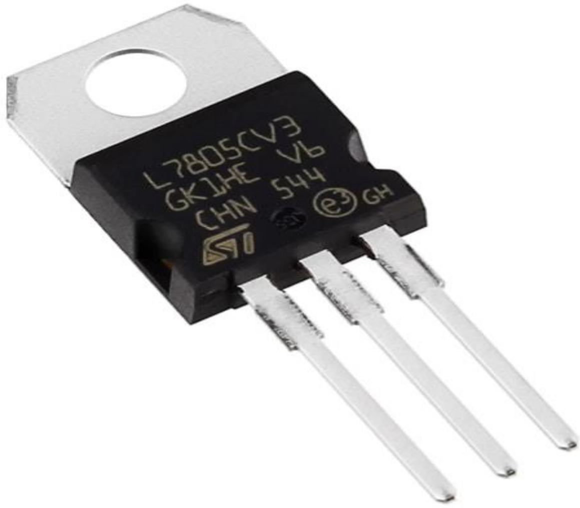

Figure 1.7: lm7805 [24]

3.4 AMS1117

the AMS1117 series of adjustable and fixed voltage regulators are designed to provide up to1A output current and to operate down to 1V input-to-output differential. The dropout voltage of the device is guaranteed maximum 1.3V, decreasing at lower load currents.

On-chip trimming adjusts the reference voltage to 1.5%. Current limit is set to minimize the stress under overload conditions on both the regulator and power source circuitry.

The AMS1117 devices are pin compatible with other three-terminal SCSI regulators and are offered in the low-profile surface mount SOT-223 package, in the 8L SOIC package and in the TO-252 (DPAK) plastic package [25].

3.4.1 Application

- High Efficiency Linear Regulators
- Post Regulators for Switching Supplies
- 5V to 3.3V Linear Regulator
- Battery Chargers
- Active SCSI Terminator
- Power Management for Notebook
- Battery Powered Instrumentation [25]

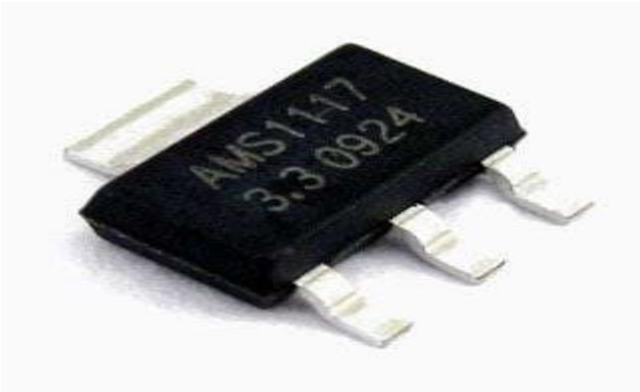

Figure 1.8: AMS1117 [25]

3.5 Dc motor

The DC motor is a direct current electric motor that transforms the energyelectrical in mechanical energy, constitutes of two parts the stator and the rotor, when the motor is powered the stator provides a constant magnetic field that sets the motor in motion in one direction, in order to make it rotate in the opposite direction, the supply poles are reversed [26].

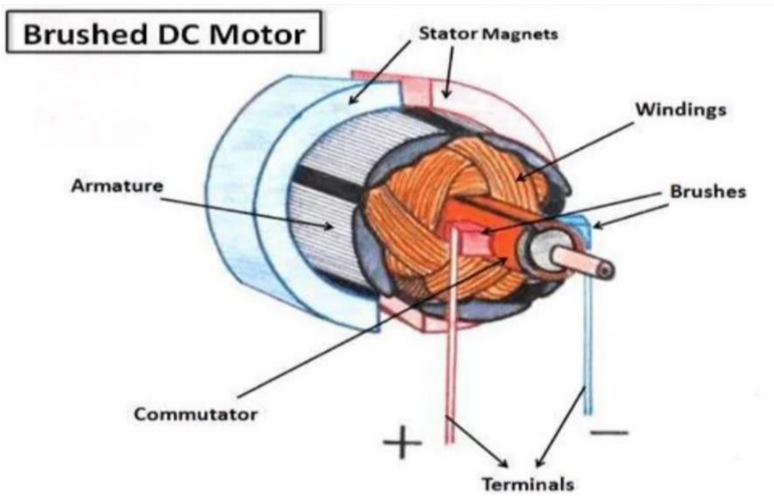

Figure 1.9: Dc motor instruction parts [26]

3.5.1 Applications of DC Motor

Owing to the fairly constant speed and medium starting torque of shunt DC motors, they are used in the following applications:

- Centrifugal and reciprocating pumps
- Lathe machines
- Blowers and Fans
- Drilling machines
- Milling machines
- Machine Tools [26]

3.6 The L298N module

The L298N module is an electronic circuit allowing the control of motor to direct current or a stepper motor [27].

It is capable of controlling DC motors from 5 to 12V, and we can control their speed using the pins ENA for the motor 1 and ENB for the motor 2 thus their direction of rotation using the pins IN1 and IN2 for the motor1 and IN3 and IN4 for the motor 2, The power supply of this module is varied between 3 and 35 V.

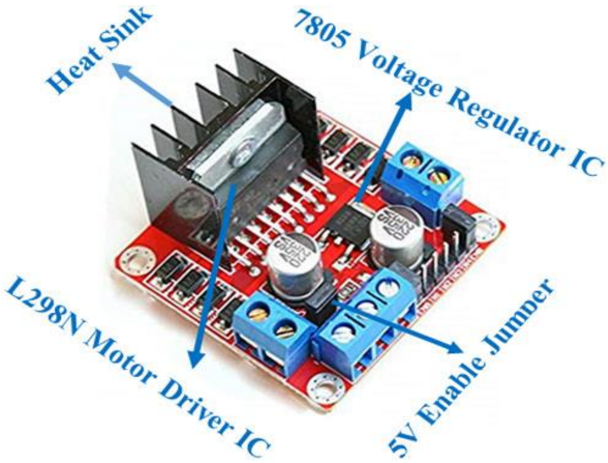

Figure 1.10: L298 motor driver [27]

3.7 The Battery

It is a rechargeable energy source necessary in order to power the devices.

in our project we used a Li-ion type screwdriver battery characterized by a voltage of 20 V and a current of 1.7 A and capacity of 1500mAh

Figure1.11: Lithium battery

4 Conclusion

Arduino has transformed the world of electronics by making it more accessible to people with varying levels of knowledge and experience. Its simplicity and versatility have allowed people to explore and create innovative projects, which would have been difficult or impossible to achieve with traditional electronics.

Chapter 2

Wireless Communication

1 Introduction

Wireless Communication is the most dynamic technological field in the field of communication. It is an integral part of our daily activities. It is a method of transmitting information from one point to another without using a connection such as wires, cables or any other physical medium. It is a mode of remote communication using electromagnetic waves. The different types of wireless communication include radio broadcasting (RF), infrared (IR), satellite, micro-waves, Bluetooth, mobile phones, GPS, and Wi-Fi.

2 Wireless Communication

The communication systems can be wired or wireless and the medium used for communication can be guided or unguided. In wired communication, the support is a physical path such as coaxial cables, twisted pair cables and the optical fiber links that guide the signal to propagate from one point to another. In wireless communications, the transmission and reception of signals are made with antennas.

Antennas are electrical devices that transform electrical signals into radio signals and vice versa. Radio signals are waves electromagnetic (EM) waves that propagate in space and are adapted to the transport of the voice and image [28].

2.1 The electromagnetic wave

Electromagnetic waves carry electromagnetic energy from the electromagnetic field through space. Electromagnetic waves include gamma rays (γ-rays), X-rays, ultraviolet rays, visible light, infrared rays, microwave rays and radio waves.

Radio waves are used in wireless communications to carry signals. In the electromagnetic spectrum, these are between 10 kHz and 20 GHz.

An electromagnetic wave is the propagation of a variation of an electric field $\bar{E}$ and a magnetic field $\bar{B}$ at the speed of light (300,000 Km/s). These two fields are perpendicular to each other and to the direction of propagation as illustrated in the (Figure 2.1). From a mathematical point of view, an electromagnetic wave can be described by Maxwell's equations [28].

The characteristics of an electromagnetic wave are:

- Its frequency v which is expressed in hertz (Hz), with $v = \frac{1}{T}$ (T: period(s)).
- its wavelength λ which is expressed in meters (m) .

- Its celerity c, or speed, which is expressed in m/s. It is 3x10 8 m/s in the empty.

These characteristics are related to each other by the equation:

$\mathbf{c = \lambda \cdot v}$

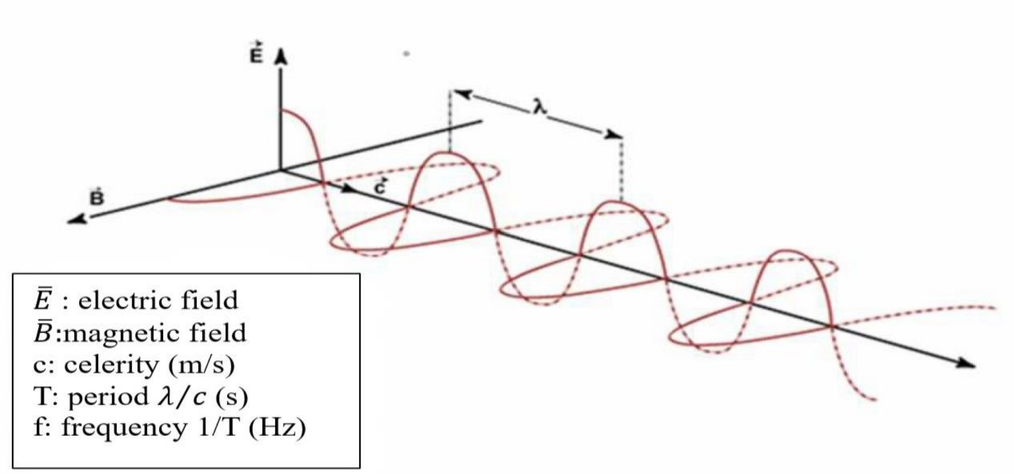

Figure 2.1: propagation of an electromagnetic wave [28]

2.2 Advantages of wireless communication

- Freedom from wires
- Easy to setup
- Better or global coverage
- Flexibility
- Cost-effectiveness
- Mobile and portable
- Mobility
- Data transmission is fast
- Network planning
- expandable

2.3 Disadvantage of wireless communication

- it is less secure
- Unreliability More open to interference
- Increased chance of jamming
- Wireless networks can be easily hacked
- Wireless networks require a careful radio frequency when they are installed
- setting up a wireless network is very costly
- Difficult to set up little experience people

2.4 Basic elements of a wireless communication system

A typical wireless communication system can be divided into three elements: the transmitter, the channel and the receiver. **Figure 2.2** shows the functional diagram of the system of wireless communication [28].

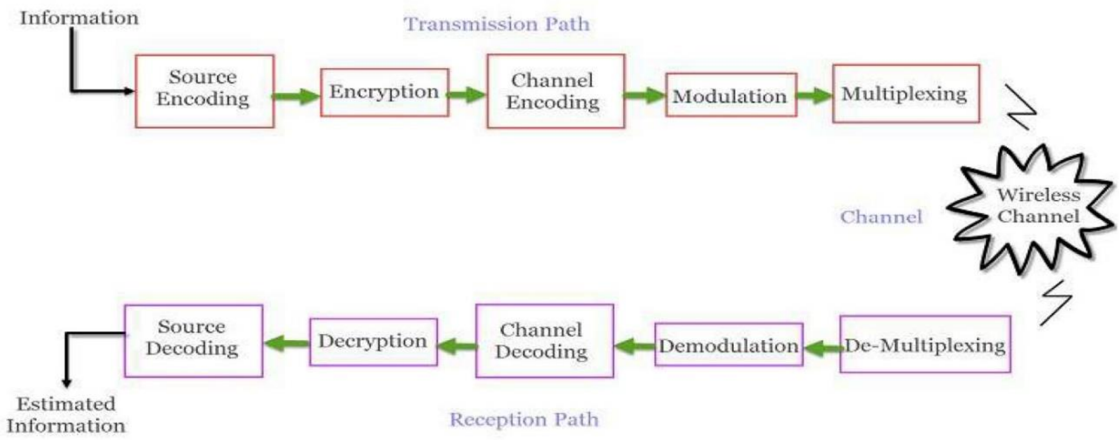

Figure 2.2: Wireless transmission chain [28]

2.4.1 transmission part

The typical transmission path of a wireless communication system includes the source encoder, the encryption, the channel encoder, the modulation and the multiplexing.

The signal from the source passes through the source encoder, which converts the signal into a appropriate form for applying signal processing techniques. Information redundant parts of the signal are removed in this process in order to maximize the use of resource.

This signal is then encrypted using an encryption standard so that the signal and the information is secure and do not allow any unauthorized access.

Channel coding is a technique applied to the signal to reduce the degradations such as noise, interference, etc. During this process, a small amount of redundancy is introduced into the signal so that it becomes robust against noise.

Then the signal is modulated using an appropriate modulation technique (PSK, FSK and QPSK, etc.), so that the signal can be easily transmitted using a antenna.

The modulated signal is then multiplexed with other signals using different multiplexing techniques such as time division multiplexing (TDM) or frequency division multiplexing (FDM) to share the precious bandwidth [28].

2.4.2 Channel

The wireless communication channel refers to the means of transmitting the signal in an open environment. A wireless channel is unpredictable and also very variable in nature and random. A channel can be subject to interference, distortion, noise, broadcast and consequently, the received signal may be marred by errors [28].

2.4.3 Reception part

The receiver aims to restore the source signal from the received signal. The reception path of a wireless communication system comprises demultiplexing, the demodulation, channel decoding, decryption and source decoding. According to the components of the receiving path, it is clear that the receiver's task is just the opposite from that of the issuer.

The channel signal is received by the demultiplexer and is separated from the other signals. They individual signals are demodulated using appropriate demodulation techniques and the original message signal is recovered. The redundant bits of the message are deleted using the channel decoder.

Since the message is encrypted, the decryption of the signal removes the security and the transforms into a simple sequence of bits. Finally, this signal is transmitted to the source decoder to recover the original transmitted message or signal [28].

2.5 Types of wireless communication systems

Wireless communication systems provide different services such as the video conferencing, cell phone, paging, television, radio, etc. In reason for the need for various communication services, different types of wireless communication are developed. The different types of technology of the most common wireless communication today are:

- Television and radio broadcasting
- Satellite communication
- Radar communication
- Mobile phone system (cellular communication)
- Global Positioning system (GPS)
- Infrared communication
- WLAN connection (Wi-Fi)
- Bluetooth
- Zigbee Connection
- Two Transmission Modules (NRF24L01)

- Cordless phones

Wireless communication systems can also be classified as Simplex, Half Duplex and Full Duplex. Simplex communication is a one-way communication unique. An example is the radio broadcasting system. The Half Duplex is a communication bidirectional but not simultaneous. An example is the walkie-talkie (radio of a group civil). Full Duplex is also a two-way and simultaneous communication. The best example of full duplex is that of mobile phones. The devices used for wireless communication may vary from service to the other and may have a different size, shape, data rate and cost. There area covered by a wireless communication system is also an important factor.

Wireless networks can be limited to a building, an office campus, a city, a small regional area or may have global coverage [28].

3 Study of the NRF24L01 wireless communication module and its interfacing with an Arduino board

3.1 Introduction

Our project implements RF wireless communication using module NRF24L01. Here we will present its different characteristics, describe its principle of operation and explain its interfacing with the Arduino board.

3.2 frequency modulation

Frequency shift modulation, better known by its English-language name of frequency-shift keying (FSK) is a frequency modulation mode used for digital communications, in which the modulated signal varies between two frequencies predetermined f_0 and f_1 as a function of the value of the bits to be transmitted.

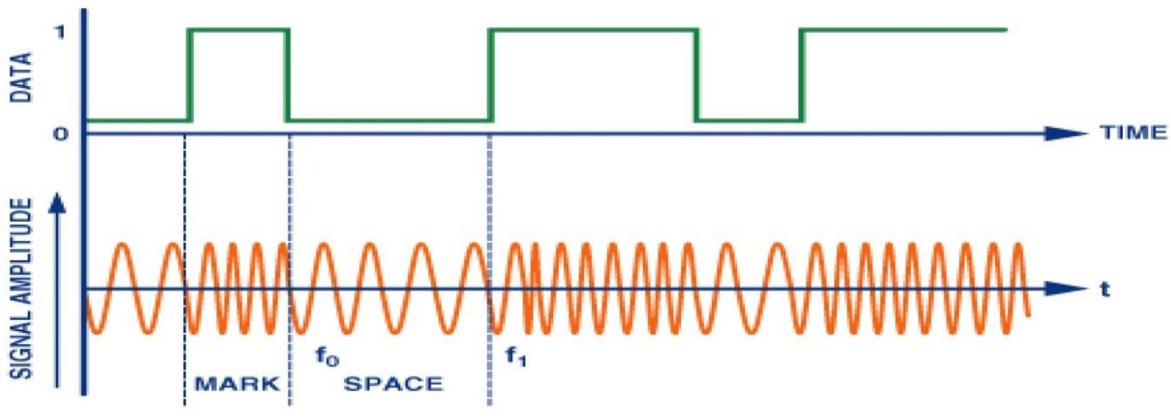

Figure 2.3: frequency modulation [29]

The digital frequency modulation GFSK (Gaussian Frequency Shift Keying) is a variety of FSK modulation. The GFSK modulation allows the reduction of the spectral occupancy of a signal modulated in FSK. GFSK modulation is used in Bluetooth communications. It is based on the same principle as the FSK but with the addition of a Gaussian low-pass filter.

For a GMSK modulation, the expression of the transmitted signal is:

$$M(t) = a_0.\cos(2\pi.(f_p \pm \Delta F/2).t)$$

Where:

a_0 and f_p represent the amplitude and the frequency of the carrier

and:

ΔF the frequency difference between the transmission of a logic state 0 and 1:

$f_0 = f_p - \Delta F/2$ et $f_1 = f_p + \Delta F/2$ [29].

3.3 The NRF24L01 radio module

The NRF24L01 is an RF module from Nordic Semiconductor. This is a low transmission power transmitter/receiver that communicates via the ISM band (Industrial, Scientific and Medical) of 2.4GHz and uses GFSK modulation (Gaussian Frequency-Shift Keying) for data transmission. The communication protocol called "Shock Burst" allows two or more modules to exchange data, with addressing, management of transmission errors and automatic retransmission in case of no-recipient's response. The data transfer rate can be 250kbps, 1Mbps and 2Mbps with low power consumption. It incorporates a 16MHz quartz [29].

3.3.1 The NRF24L01 is available in two versions

The classic NRF24L01 module with an antenna integrated into the PCB, illustrated in Figure 2.4(a), which can reach a range of 25m in an enclosed environment and 50m in open middle.

The NRF24L01 PA LNA module, illustrated in Figure 2.4(b), has:

- a PA amplifier (Power Amplifier) on the transmitter side and a LNA amplifier (Low Noise Amplifier) on the receiver side.
- an external antenna allowing to reach a much longer range large of about 1000m [30].

3.3.2 Pinout of the NRF24L01 module

The NRF24L01 and NRF24L01+PA/LNA modules are both equipped with an 8-pin connector as shown in Figure 2.4 As we can see, this is the same pinout [31].

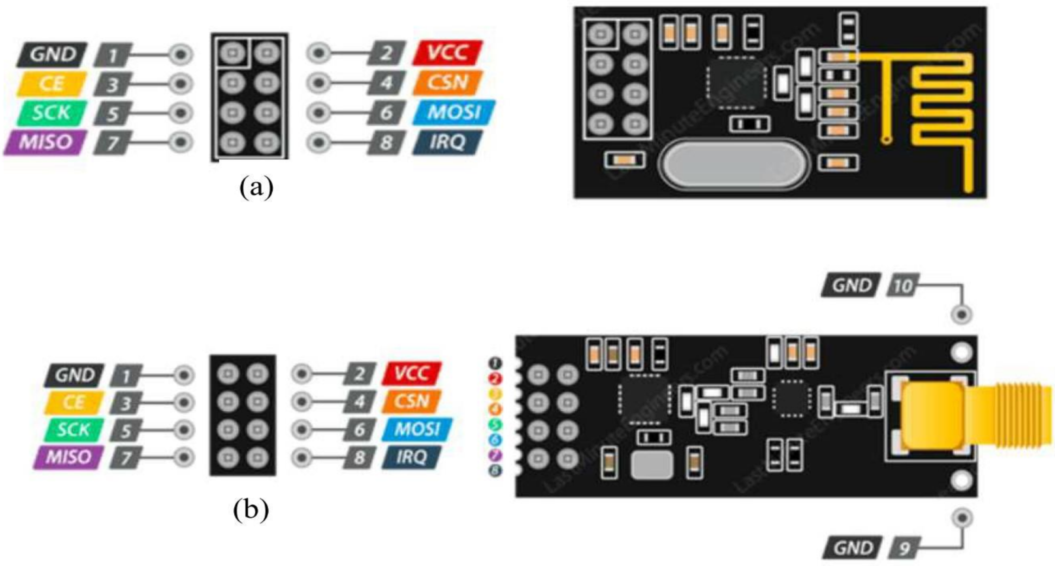

Figure 2.4 : Brochure: (a) NRF24L01 module (b) NRF24L01 module + PA amplifier / PA amplifier

The Table gives the meaning and the role of each pin.

Table 2.1: Pinout of the NRF24L01 module [32]:

Pin	Signification	Role
GND	Ground	Mass
VCC	Power	Power Supply (1.9V to 3,6V)
CE	Chip (RX/TX) Enable	Active in the high state (RADIO) if active, the module sends or receives
CSN	Chip Select Not	Active in the low state (SPI Select) If active, the circuit communicates in SPI
SCK	Serial Clock	SPI communication clock
MOSI	Master Out Slave In	Master to slave data
MISO	Master In Slave Out	Slave to master data
IRQ	Interrupt Request	Active interruption in the low state, signals that a package has been received, issued or failed.

3.3.3 NRF24L01 module power options

The NRF24L01 module operates at supply voltages V DC ranging from 1.9 to 3.6V. In practice, this one is most often powered by 3.3V. As for the other pins, they are 5V compatible, which simplifies its connection with an Arduino board.

If the NRF24L01 module draws its 3.3V power supply directly from a card Arduino, it may happen that this voltage drops due to current peaks that can occur during the transmission of data resulting in malfunctions. By therefore, it is then essential to solder an electrolytic capacitor of 10 µF between the power supply pins GND and V DC, as illustrated in Figure 2.3(a), to ensure a stable operation and correct emission.

Another solution is to opt for a power supply module specifically designed for the NRF24L01, illustrated in Figure 2.3(b), whose role is to provide a stabilized power supply to it, in particular when the transmission power becomes too important. The power supply module is a 5V/3.3V voltage converter [32].

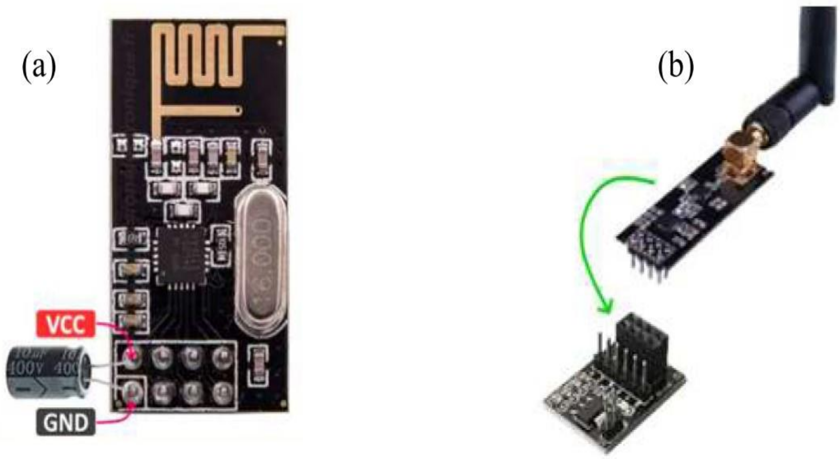

Figure 2.5: Power options for the NRF24L01 module [32]

3.3.4 RF channels of the NRF24L01 module

The transmission and reception of data by the NRF24L01 module is done on a certain frequency called the RF channel. The NRF24L01 has 125 RF channels spaced from 1MHz between them as illustrated in Figure 2.6.1. Therefore, the NRF24L01 works on the range from 2.4 GHz to 2.525 GHz. To communicate with each other, the NRF24L01 modules use the same frequency channel [31].

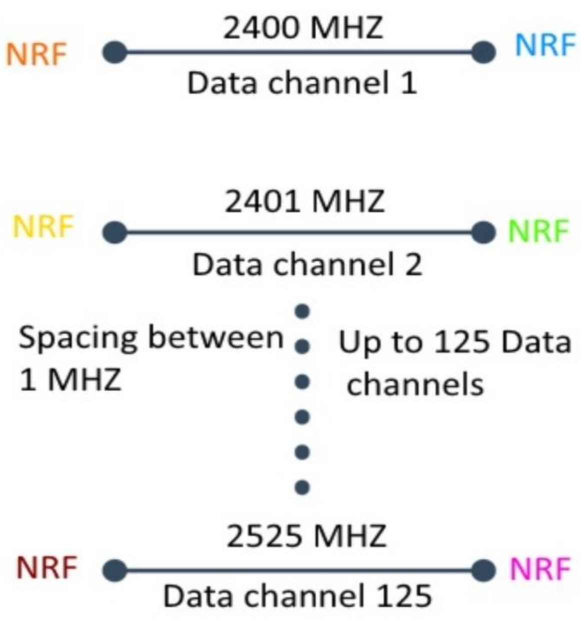

Figure 2.6.1 Different channels of the NRF24L01 module [31]

In addition, an NRF24L01 module has six communication pipes or tunnels, as illustrated in Figure 2.6.2. Therefore, an NRF24L01 module can communicate simultaneously with 6 other modules. However, only one pipe can transmit and receive then that others can only receive.

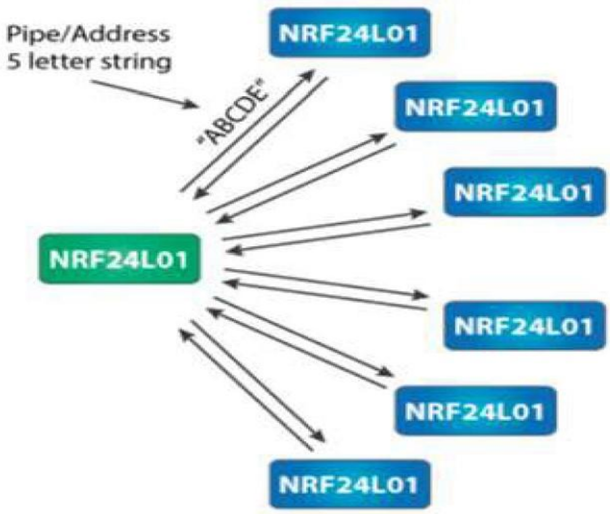

Figure 2.6.2: Communication of a NRF24L01 module with 6 Pipes [33]

Each pipe is assigned an address (see Table). Only 2 pipes have an address encoded on five bytes. For the other 4 pipes, the address is coded in on only one byte and completed with the first 4 bytes of the second pipe.

Table 2.2: Pipes and addressing [32]:

N° of the Pipe	Name of the pipe	Can issue	Can receive	Address size
1	Pipe0	X	X	5 octets
2	Pipe1		X	5 octets
3	Pipe2		X	1octet
4	Pipe3		X	1octet
5	Pipe4		X	1octet
6	Pipe5		X	1octet

3.3.5 The SPI link

The NRF24L01 module communicates with an Arduino-type microcontroller or compatible via the SPI bus (Serial Peripheral Interface). The SPI link is a serial link synchronous which operates in full-duplex mode.

We find at the level of the pinout:

- Connect the three pins common to the SPI bus, namely:
  - MOSI: via which the master transmits data to the slave.
  - MISO: via which the slave transmits data to the master.
  - SCK: clock signal, generated by the master, synchronizes the transmission.
- The two pins:
  - CE: in TX mode, active in the low state to transmit and in RX mode, active at the high state waiting for data.
  - CSN: still in the high state. Set to the low state to activate the SPI bus.
- An IRQ interrupt request pin: active in the low state when an event occurs [32].

3.4 Mounting the NRF24L01 module with the Arduino board

To connect the NRF24L01 module to the Arduino board, it must be powered by 3.3 V and connect its SPI interface to the digital pins of the card. To do this, it is necessary to respect the correspondence given in Table 2.3. As for the CE and CSN pins, there is no constraints, they can be connected on the digital pins that remain available. In general, the IRQ pin is not necessary and therefore is not used. This wiring must be the same for the transmitter and receiver, as shown in the example of Figure (2.7) [34].

Table 2.3: Connection of the NRF24L01 module to the SPI bus of the microcontroller [33]

	MOSI	**MISO**	**SCK**
Arduino Uno	11	12	13
Arduino Nano	11	12	13
Arduino Mega	51	50	52

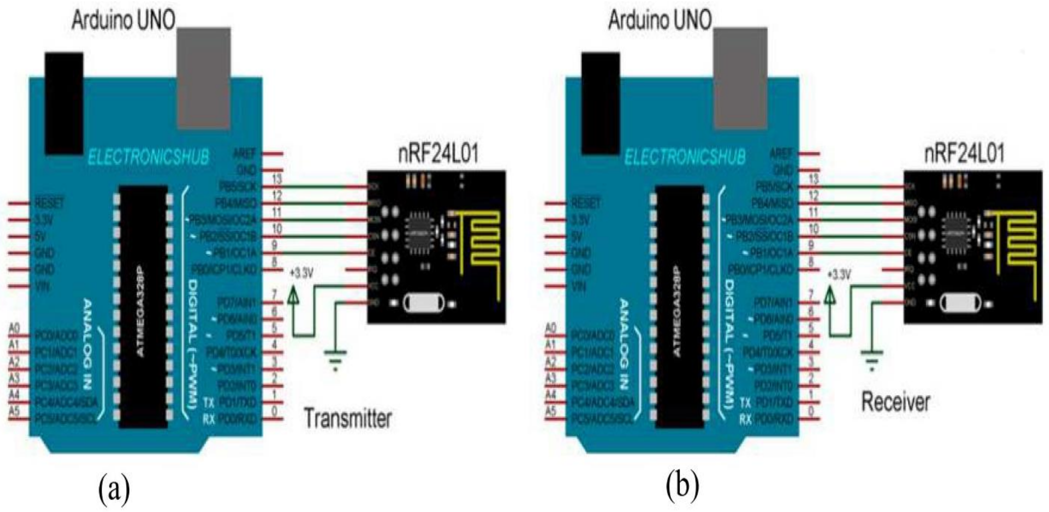

Figure 2.7: Example of connecting the NRF24L01 module with the Arduino Uno board [33]

In the sketch, it is necessary to install the NRF24L01 library beforehand, available in the library manager of the Arduino IDE software. In the area of search for the "Library Manager" window, just type the keyword RF24 then install the TMRh20 package. In the header, we start by including:

- the SPI library, required for exchanges between Arduino and NRF24L01
- the RF24 library, which includes a main file and a file specific to modules of the NRF24L01 family.

  #include <SPI.h>

  #include <NRF24L01.h>

  #include <RF24.h> [32]

We then declare:

- a radio variable of type RF24, specifying the numbers of the pins on which are connected, in order, the CE and the CSN: RF24 radio (CE, CSN);
- a function used to set the address of the NRF24L01:

  example: const byte address [6] = "00001";

In the Setup part, we have:

- **radio.begin():** initializes the radio module:
- **radio.openWritingPipe(address):** where "address" represents the address of the communication channel on 5 bytes. This function allows the transmitter to open the communication channel with the receiver module.
- **radio. openReadingPipe (0, address):** this function allows the receiver to to accept the communication channel proposed by the transmitter with which it will communicate.
- **radio.stopListening():** this function allows to put the module in mode transmitter.
- **radio.startListening():** this function allows to put the module in mode receiver.

In the loop part, we have:

- **radio.available():** this function makes it possible to continuously test whether the

  data have been received.
- **radio.write(&text, sizeof(text)):** This function allows the transmitter module to send the data. "text" represents the data to be transmitted and sizeof(text) the length of the data.
- **radio.read(&text, sizeof(text)):** This function allows the receiver module to recover the data. "text" represents the data to be received and sizeof(text) the length of the data [32].

4 Conclusion

In this chapter, we have discussed the study of the NRF24L01 module. This is a 8-pin wireless communication module from Nordic Semiconductors using the SPI communication protocol. It comes in two versions: the classic NRF24L01 and the NRF24L01 PA/LNA. In our project, our choice fell on the nRFL01 version classic. It offers data rates of 250kbps, 1Mbps and 2Mbps and it is characterized by a low consumption power. It has 125 channels and each module can communicate with 6 spades at the same time. The power supply of the module is 3.3V while its pins support 5V voltages, which simplifies its interfacing with the Arduino board.

However, its SPI bus connects to the Arduino board according to a very precise correspondence which must be the same for both the transmitter and the receiver.

Chapter 3

Simulation and Realization

1 Introduction

In this chapter, we focus on the realization of our vehicle, this project is composed of two parts which are: the vehicle part and the programming part in the first part we will do some tests on the components, then we move on to remotely control the direction and the speed of a DC motor, at the end of this part we will do the final assembly.

The second part we will calibrate and then control the gyroscope, at the end we will try to control the vehicle by a single RF command.

For the realization of our project, the project consists of two parts:

- The vehicle part
- The programming part

2 PCB

A printed circuit board is an electrical circuit whose electronic and conductive components are integrated into a mechanical structure. A PCB is composed of several layers of copper covered with an insulating material. In addition to many components, printed circuits require connectors such as copper tracks, pads, and heat sinks [34].

2.1 Composition

A PCB is sort of like a layer cake or lasagna- there are alternating layers of different materials which are laminated together with heat and adhesive such that the result is a single object.

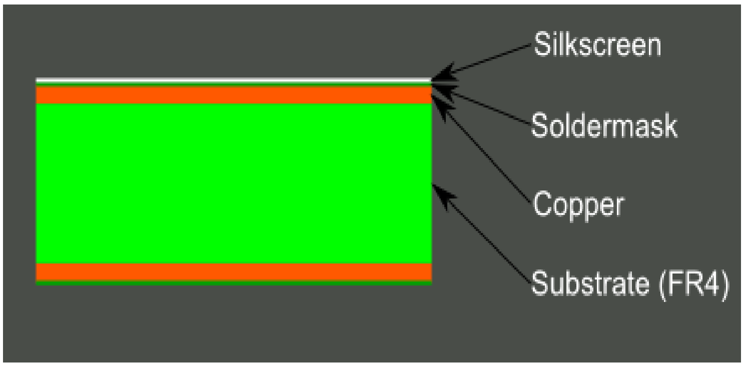

Figure 3.1: PCB compositions [35]

2.1.1 FR4(substrate)

The base material, or substrate, is usually fiberglass. Historically, the most common designator for this fiberglass is "FR4". This solid core gives the PCB its rigidity and thickness. There are also flexible PCBs built on flexible high-temperature plastic (Kapton or the equivalent).

2.1.2 Copper

The next layer is a thin copper foil, which is laminated to the board with heat and adhesive. On common, double-sided PCBs, copper is applied to both sides of the substrate. In lower cost electronic gadgets, the PCB may have copper on only one side. When we refer to a double sided or 2-layer board we are referring to the number of copper layers (2) in our lasagna. This can be as few as 1 layer or as many as 16 layers or more.

2.1.3 Soldermask

The layer on top of the copper foil is called the soldermask layer. This layer gives the PCB its green (or, at Spark Fun, red) color. It is overlaid onto the copper layer to insulate the copper traces from accidental contact with other metal, solder, or conductive bits. This layer helps the user to solder to the correct places and prevent solder jumpers.

2.1.4 Silkscreen

The white silkscreen layer is applied on top of the soldermask layer. The silkscreen adds letters, numbers, and symbols to the PCB that allow for easier assembly and indicators for humans to better understand the board. We often use silkscreen labels to indicate what the function of each pin or LED.

Silkscreen is most commonly white but any ink color can be used. Black, gray, red, and even yellow silkscreen colors are widely available; it is, however, uncommon to see more than one color on a single board [35].

3 Easy Eda

EasyEDA is a free, no-install, Web, and cloud-based EDA CAD suite, featuring powerful schematic capture, mixed-mode circuit simulation, and PCB layout in a user-friendly cross-platform browser environment, for engineers, educators, students, and hobbyists. It's compatible with windows, mac, and Linux, as long as there is one browser on the OS and connecting to the network. You can draw schematics quickly using the available libraries on the browser. It offers Simple, Easier, Friendly, and Powerful general drawing capabilities along with Real-time Team Cooperation & Integrated PCB Fabrication services with the collaboration of JLCPCB [36].

featuring a large production scale, great manufacturing & production facilities, and superior PCB quality

3.1 Complete PCB Design Tutorial Using EasyEDA

Follow each step as shown below:

3.1.1 Start Making Your PCB Design Using EasyEDA

Start by drawing the initial schematic of your PCB using the EasyEDA design tool. For this, go to the EasyEDA website and create a free account. Registering using a Google Account is also possible Once you are logged in, go to "Editor" and create a new project. Now start drawing your circuit diagram. Make sure you add the components from JLCPCB/LCSC library while drawing the schematics in EasyEDA.

3.1.2 Allocate & Route the Circuit Components

Now, pick the correct parts from the LCSC library & place them on the canvas in EasyEDA using the "wire" tool. After the design is complete, verify & save the circuit.

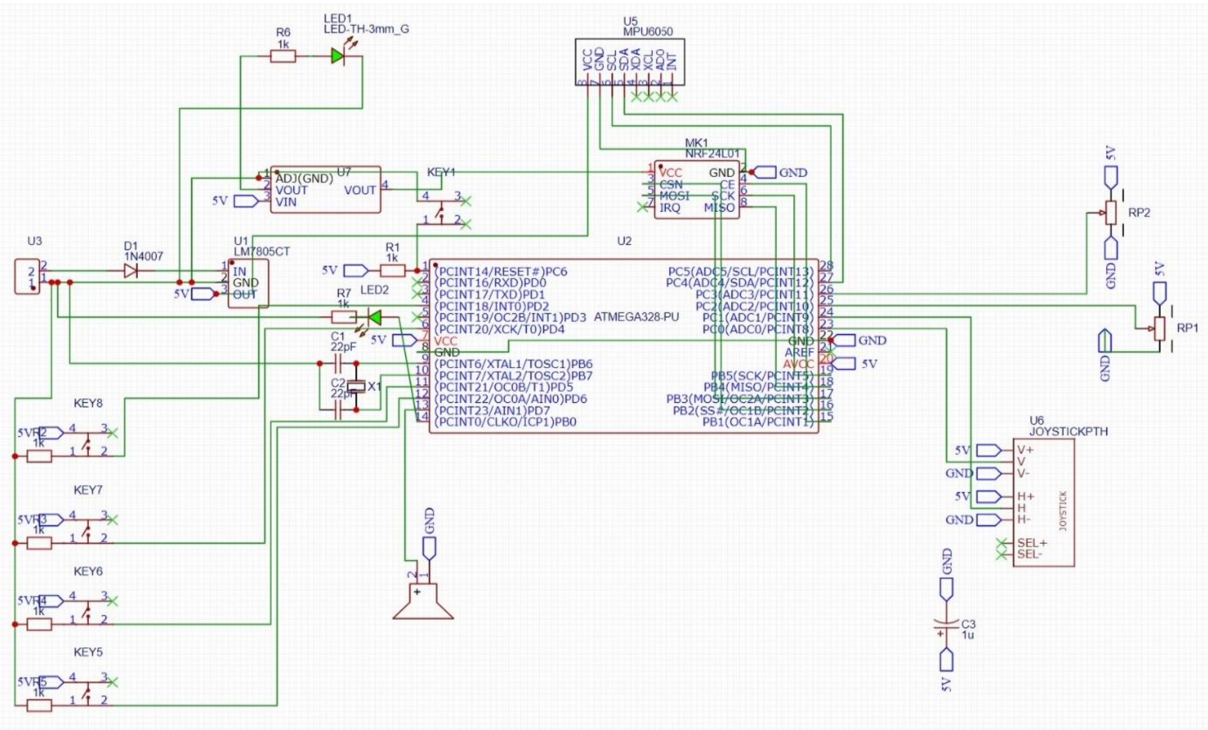

Figure 3.2.1: PCB schematic for the remote control

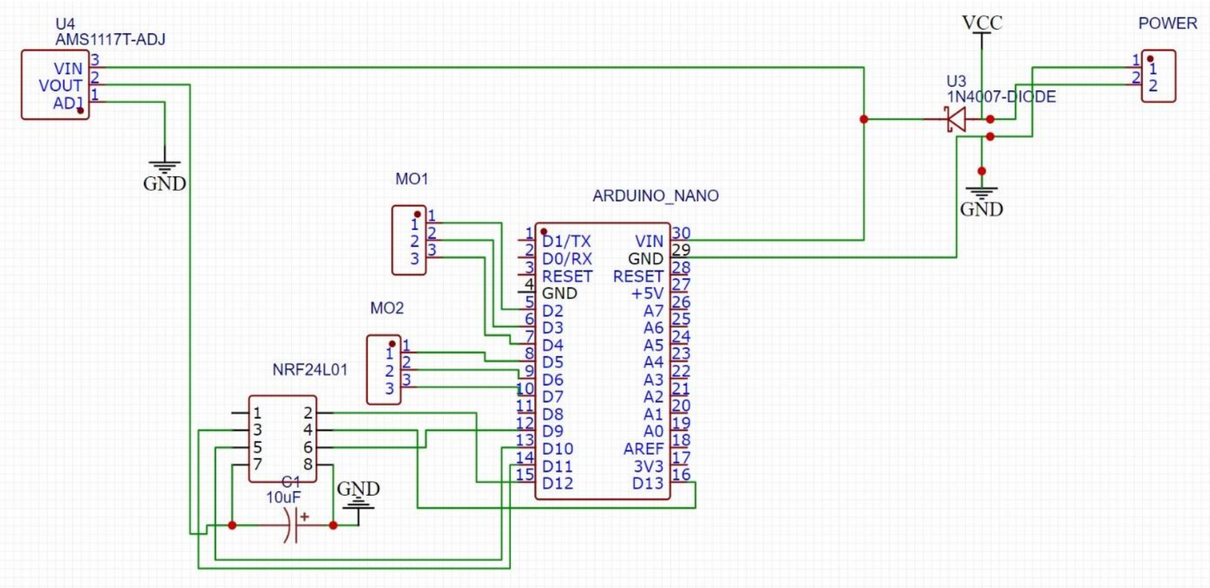

Figure 3.2.2:PCB schematic for the vehicle

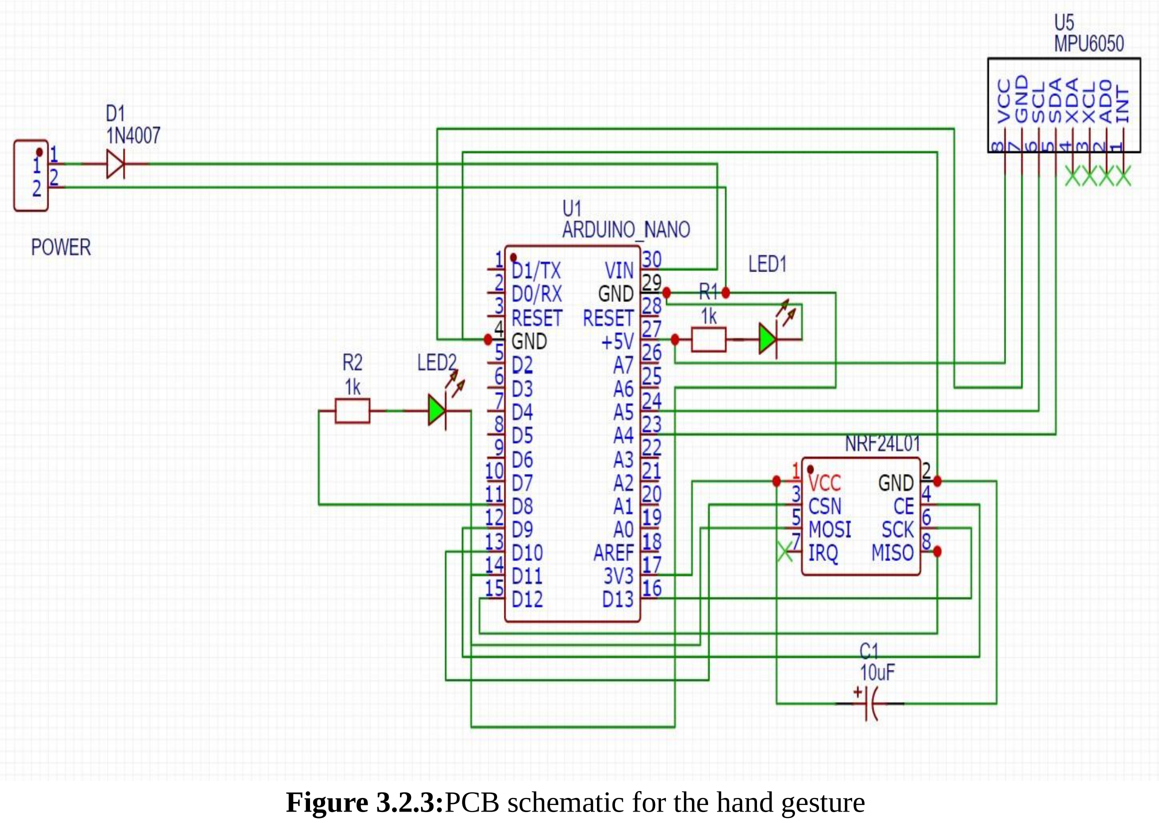

Figure 3.2.3:PCB schematic for the hand gesture

3.1.3 Build the PCB Layout

After that, go to the top tool list, click on the cover and select "Convert to PCB". A new window with the layout will open, here you can place the components inside the boundary and rearrange them any way you want. In order to get the best possible look for your PCB, place & route parts with the same functions close to each other.

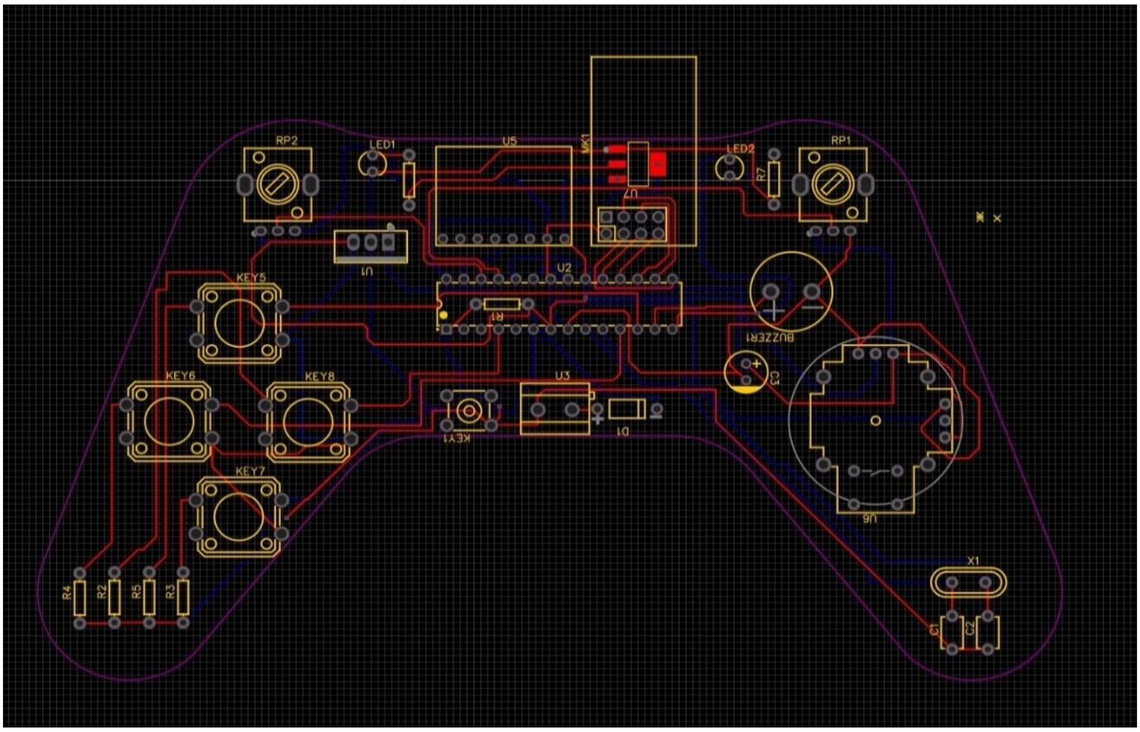

Figure 3.3.1: PCB layout for the remote control

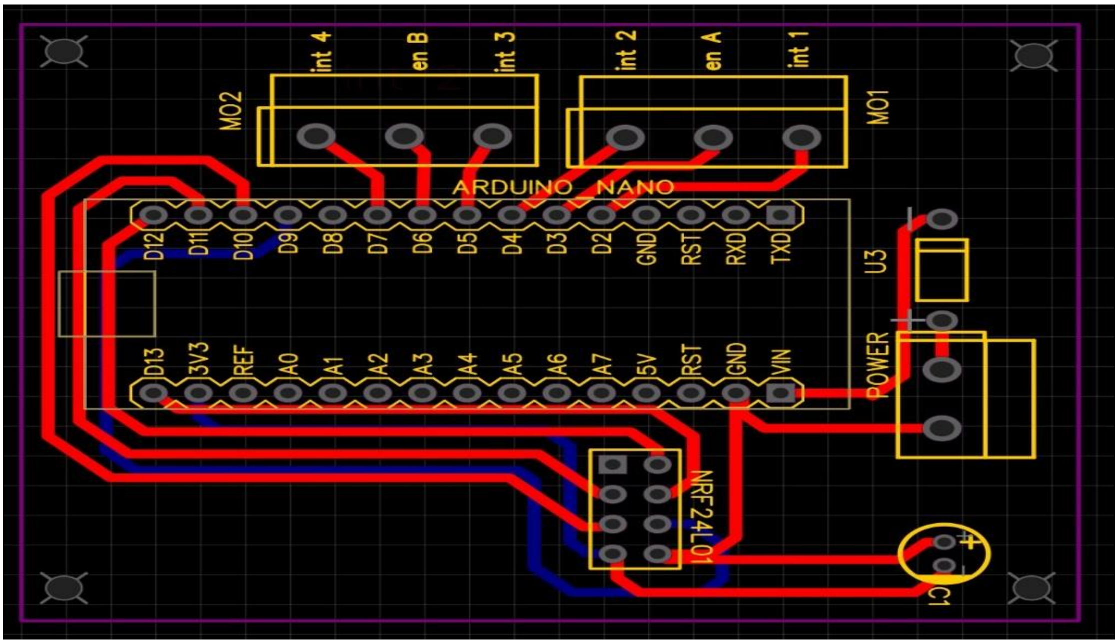

Figure 3.3.2:PCB layout for the vehicle

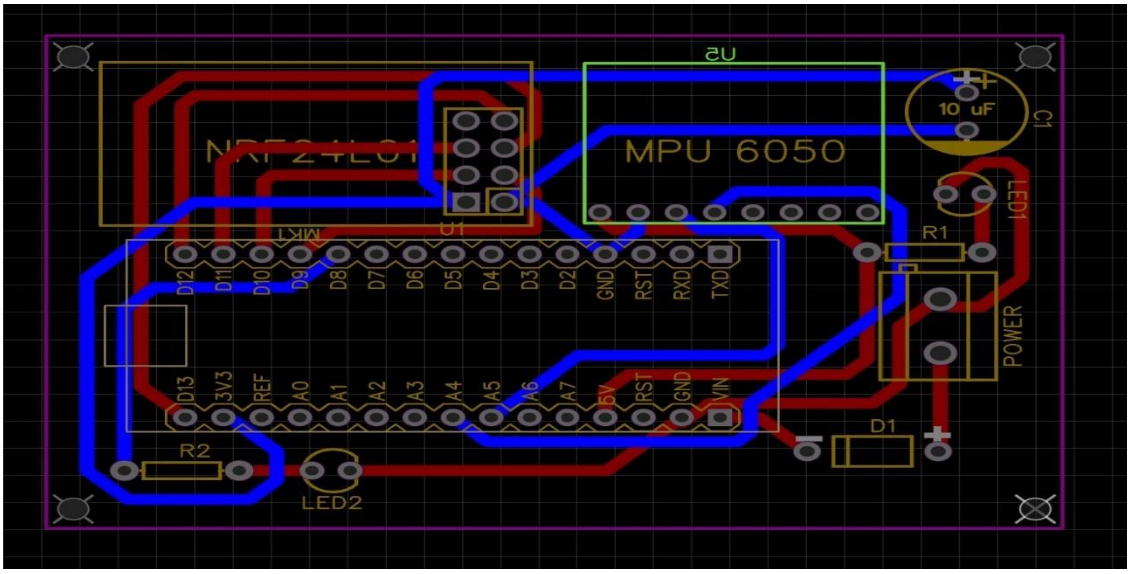

Figure 3.3.3: PCB layout for the hand gesture

Now, save the PCB layout. After saving, start the routing of your PCB components. You can use the manual routing tool to do this. But an easier approach is to use the built-in autoroute feature. To use that, click on the Route Tool and Select "Auto Router". This will open an Auto Router Config Page where you can provide details such as clearance, track width, layer information, etc. After that, click on "Run"

3.1.4 Import Gerber, BoM & CPL/PNP Files

After your layout is completed, import the Gerber, BoM & CPL files, which you can use to manufacture your PCB from JLCPCB.

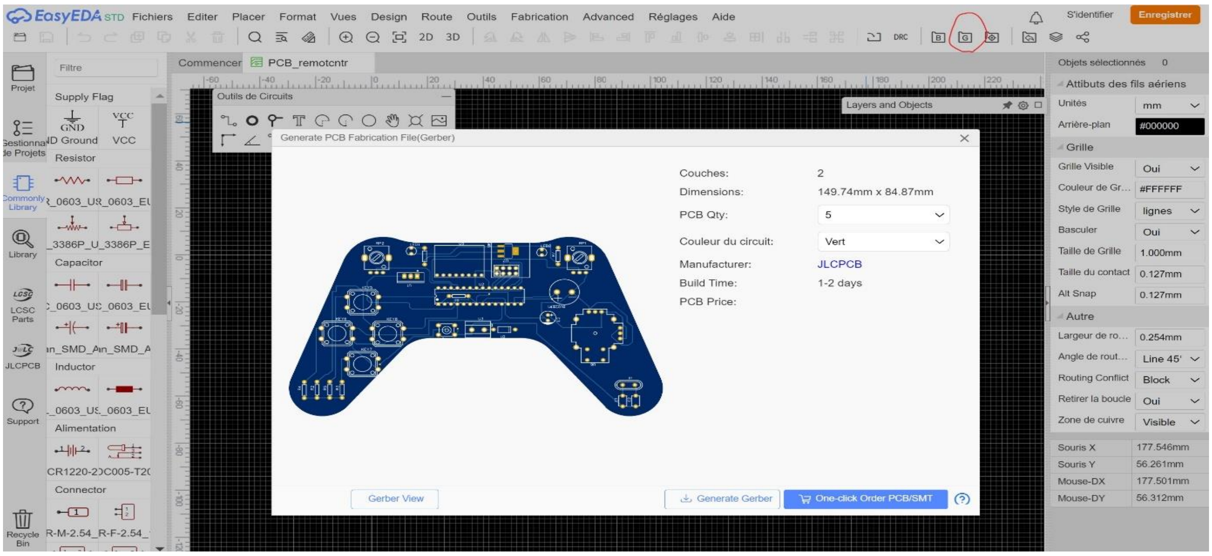

Figure 3.4: generate PCB fabrication file

- **Gerber Files**

  The Gerber File has info on your PCB such as PCB layout information, Layer information, spacing information, and trace/track parametric to name a few.

- **CPL/PNP File**

  CPL/PNP file (Component Placement List/Pick & Place File). SMT Assembly machines need this file to determine where each part should be placed on the board.

- **BoM File**

The Bill of Material (BoM) contains the list of all components in the PCB Layout.

3.1.5 Setup & Sign-in to your JLCPCB Account

Now, Visit the JLCPCBs website and create a free account, you can also use your google/Facebook account to register for a new account.

3.1.6 Export the Gerber File

After signing into your JLCPCB account, click on Quote Now and upload your Gerber File.

3.1.7 Add SMT Assembly

After that, the user interface will show a preview of your PCB board. Verify that the PCB layout is as per your desire. Below the PCB preview, there are various custom design options like PCB

Quantity, Texture, Thickness, Color, etc. Tick the options as per your design needs. After that, choose "Assemble your PCB Boards".

3.1.8 Export BoM & CPL/PNP (Pick-Place) File

After that, Upload the BoM & CPL file that you downloaded earlier.

3.1.9 Choose Essential Parts

Now, choose all the components you want JLCPCB to assemble in your PCB.

3.1.10 Evaluate Assembly Order

Here, you can review & assess your PCB design order. Check & verify the PCB layout & components. If you see any problem, click on Go Back to edit your order.

3.1.11 Checkout

After that, click on Save To Cart. On the next page, you can choose a shipping & payment option and Check Out Securely. You can either use PayPal or Credit/Debit Card to pay for your PCB.

4 The components used

4.1 For the vehicle

- Chassis: It is a plate that represents the frame of the vehicle, on which it supports and assembles all the components of the vehicle.
- Arduino NANO
- DC Motor
- The L298N module
- NRF24L01 Receiver
- Battery

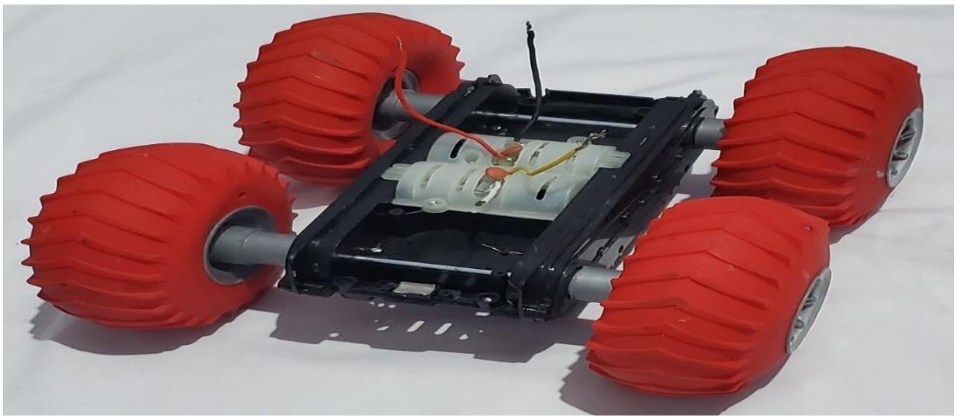

Figure 3.5.1: vehicle chassis

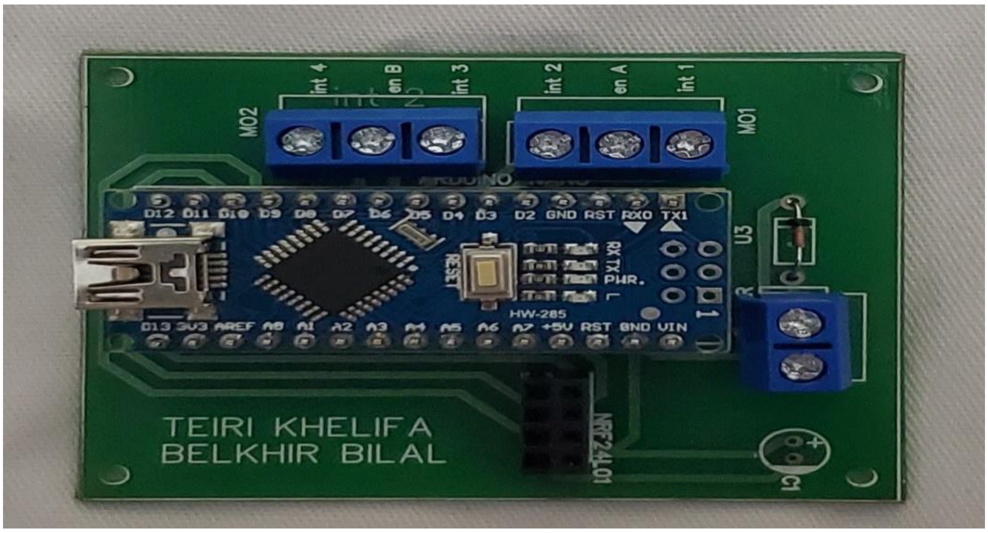

Figure 3.5.2: vehicle circuit

4.2 For the remote control

- atmega 328p
- Transmitter NRF24L01
- A joystick
- Gyroscope MPU6050
- buttons
- potentiometer
- buzzer
- battery

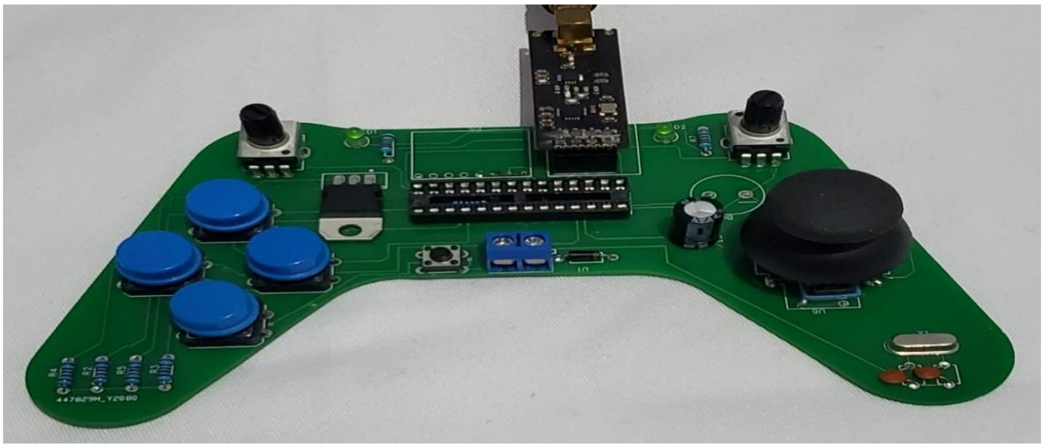

Figure 3.6: remote control circuit

4.3 For the Hand Gesture

- Arduino Nano
- Transmitter NRF24L01
- Gyroscope MPU6050
- Led
- Battery

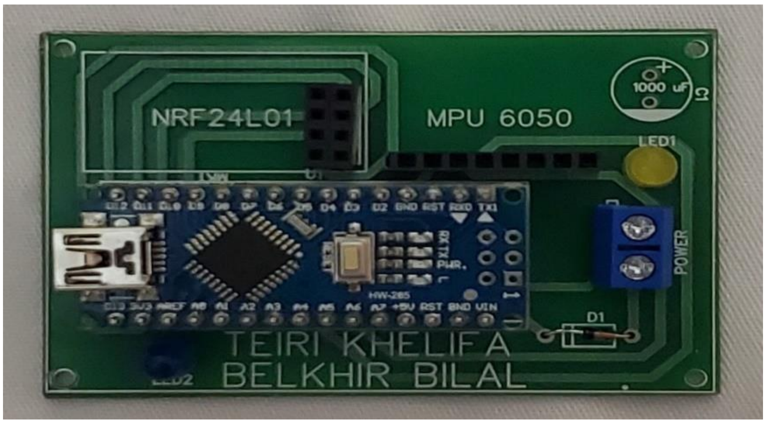

Figure 3.7: hand gesture circuit

5 Components testing

To begin, we will test the NRF24L01 transmission and reception module, so we're going to make a simple program that sends the values of a joystick. The figures the following show the transmitter and receiver assembly as well as the test results:

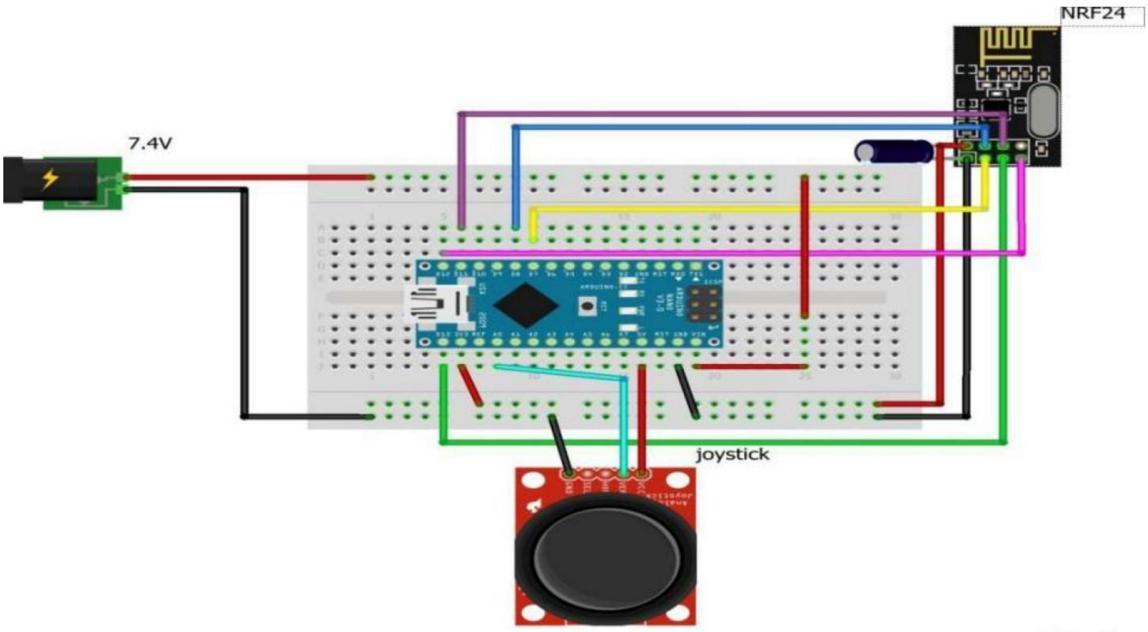

Figure3.8.1: the mounting of joystick

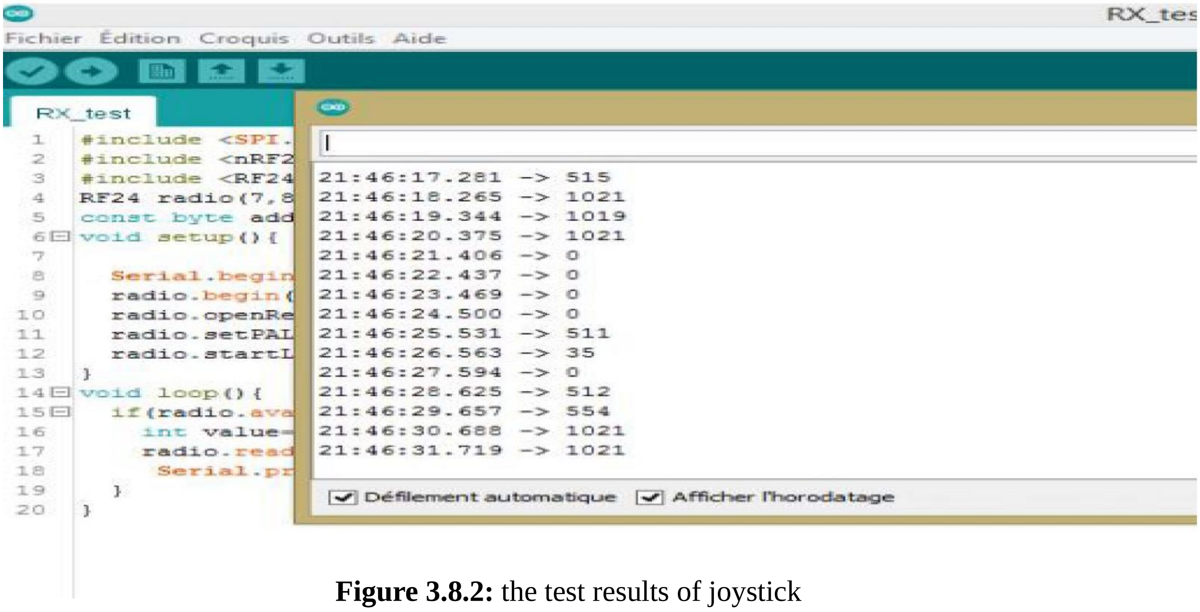

Figure 3.8.2: the test results of joystick

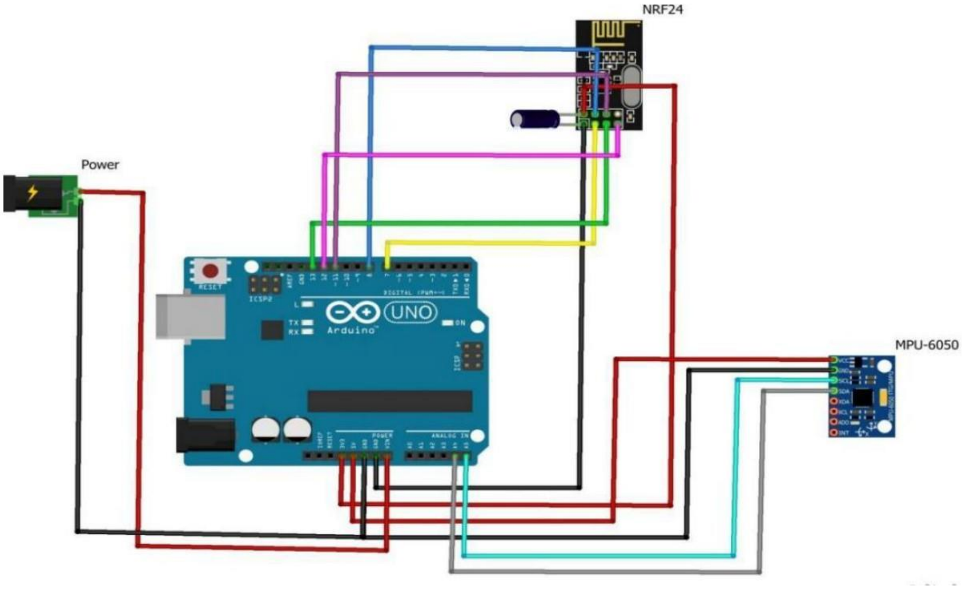

Figure3.8.3: the mounting of the MPU6050

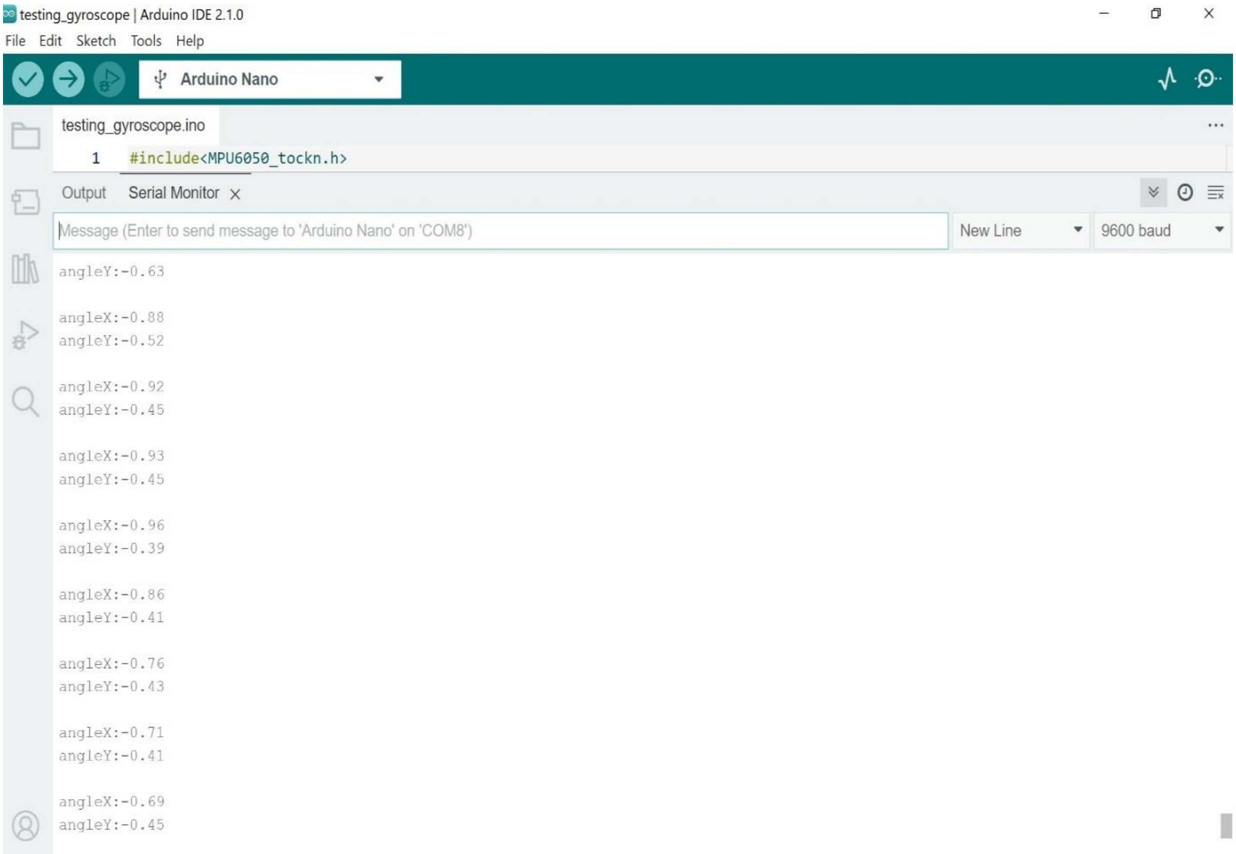

Figure 3.8.4: The Result Test of mpu6050

Now, when everything is working well, we will take our first step in our project, which consists in remotely controlling the direction and speed of a current motor continuous using Arduino and the L298N module and the NRF24L01 module.

5.1 Connecting the components

After testing all the components for our vehicle we connect each element in its place.

5.1.1 For the transmitter

The connection of NRF24L01 to Arduino is as follows:

- THIS at PIN 7
- CSN at PIN 8
- SCK at PIN 13
- MOSI at PIN 11
- MISO with PINE 12
- VCC at 3.3v
- GND to GND

The joystick connection to the Arduino is as follows:

- VCC at 5v
- GND to GND
- VRX not connect
- VRY with PIN A0

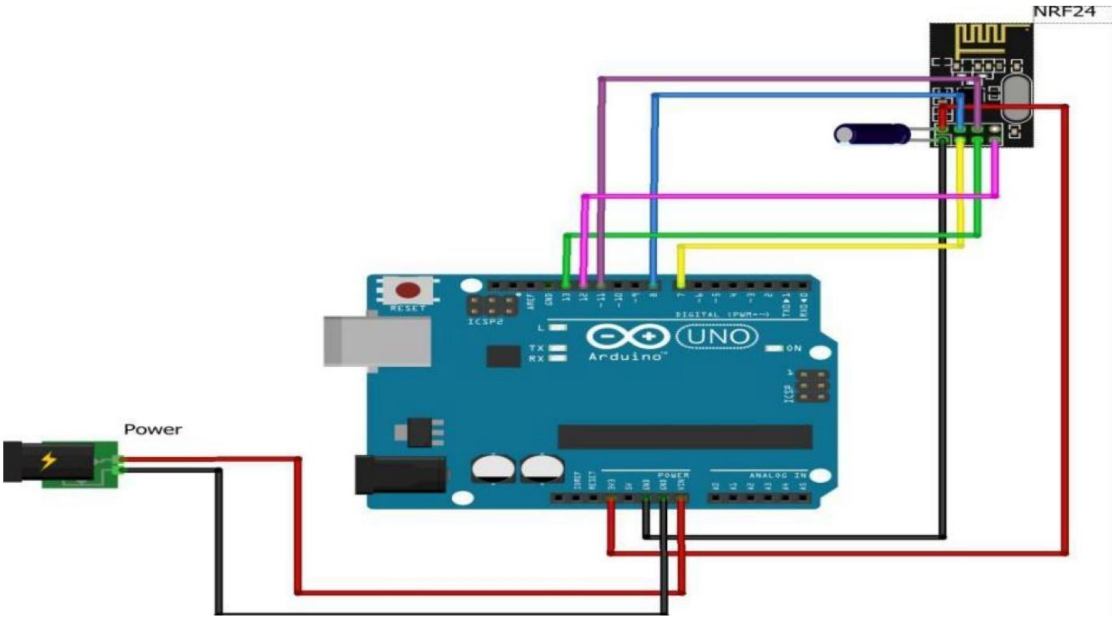

Figure 3.8.5: the transmitter assembly

We used VRY to control the DC motor, and VRX to control the servo motor, The power supply of the Arduino is done by a 7.4v li-ion battery.

5.1.2 For the receiver

The connection of NRF24L01 to Arduino is the same as the transmitter.

The connection of the L298N module to Arduino is:

- ENA at PIN 10 (where ENA activates the PWM signal for motor A)
- IN1 to PIN 4
- IN2 to PIN 5
- GND to GND

The L298N module is powered by an 11.1v li-ion battery, and the Arduino by 5v of L298N, the DC motor is connected to the A bridge of L298N, the following figure shows the assembly receiver plus the DC motor:

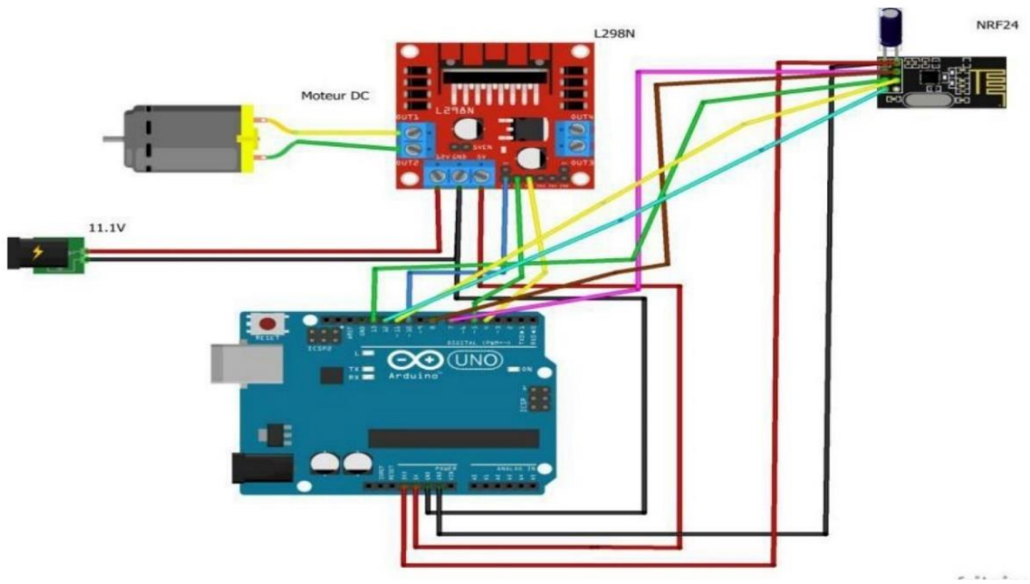

Figure 3.8.6: the receiver mounting plus DC motor

6 Conclusion

While large parts of the PCB manufacturing have stayed relatively stable over the last few decades, online CAD library services are making big impacts on the design process. By reducing the amount of time waste on CAD modeling and improving the reliability of designs these services are dramatically improving the engineer's time. Additionally, they are revealing for the first time the potential for a closer relationship between component manufacture and the individual engineer. By feeding insights of engineers' behavior within the library back to major comments manufacturers, these services anticipate and lay the foundation for innovation.[37]

General conclusion

The goal of this project is to create a mini explorer robot functional in all terrains, it can used in Barbershops to collect hair and clean the place, as well as in car parks where the distance is small and also to clean the sidewalk, this mobile robot based on Arduino, controlled remotely using radio frequency control. In the first chapter, we talked about the Arduino in general, and define the components to use in practical realization, applications and some meanings to help us in the following parts. Secondly, in chapter two we talked about wireless communication, and explains the radio frequency NRF24L01. Finally, in chapter three we have the PCB manufacturer and after doing the tests, assembling and controlling the car by the remote control, The car can be equipped with a camera and a position sensor, but unfortunately, we were not able to achieve this due to the difficulty and high cost.

Annex

Arduino using code :
For the Remote Control:

```
#include <SPI.h>
#include "NRF24L01.h"
#include "RF24.h"
#include "Wire.h"
#include "I2Cdev.h"
#include "MPU6050.h"
const uint64_t pipe = 0xE8E8F0F0E1LL;
MPU6050 mpu;
int16_t ax, ay, az ;
int16_t gx, gy, gz ;
RF24 radio(9,10);
int data[10];
int A,B,C,D,I,J,M,N;
const int button_left=5; const int button_up =6;const
int button_right = 2; const int button_down = 4 ;
int led = 8;
bool E = false; bool F = false; bool G = false;
bool H = false;
void setup()
{
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();
  radio.begin();
  radio.openWritingPipe(pipe);
    pinMode(button_left,INPUT);
    pinMode(button_up,INPUT);
    pinMode(button_right,INPUT);
    pinMode(button_down,INPUT);
    pinMode(led,OUTPUT);
}
```

```
void loop()
{
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
    A = analogRead(A0); B = analogRead(A1);
    C = analogRead(A2);
    D = analogRead(A3);
    E = digitalRead(button_left);
    F = digitalRead(button_up);
    G = digitalRead(button_right);
    H = digitalRead(button_down);
    I = map(C, 0, 1023, 0, 255);
    J = map(D ,0, 1023, 0,255);
    data [0] = A;
    data [1] = B;
    data [2] = E;
    data [3] = F;
    data [4] = G;
    data [5] = H;
    data [6] = I;
    data [7] = J;
    data [8]= M=map(ax, -17000, 17000, 300, 400);
    data [9]= N=map(ay, -17000, 17000, 100, 200);
  radio.write( data, sizeof(data) );
   if (E == HIGH || F == HIGH || G == HIGH || H ==
HIGH ||A>=700||A<=25|| B>=700||B<=25 || M<320||
M>380 || N<130|| N>170)
    {digitalWrite(led,HIGH);}
    else {digitalWrite(led,LOW);}
}
```

For Vehicle with Remote Control:

```
#include <SPI.h>
#include "NRF24L01.h"
#include "RF24.h"
#include <digitalWriteFast.h>
const uint64_t pipe = 0xE8E8F0F0E1LL;
RF24 radio(9,10);
int data[10];
int a; int b; int c; int d;int i;int J;int M; int N;
bool e; bool f; bool g; bool h;
int mo0 = 2;
int mo1 = 4;
int mo2 = 5;
int mo3 = 7;
int pwm1 = 3;
int pwm2= 6;
  void setup(){
    Serial.begin(9600);
    Serial.println("Receiver");
    radio.begin();
    radio.openReadingPipe(1, pipe);
    radio.startListening();
    pinMode(mo0,OUTPUT);
    pinMode(mo1,OUTPUT);
    pinMode(mo2,OUTPUT);
    pinMode(mo3,OUTPUT);
    pinMode(pwm1,OUTPUT);
    pinMode(pwm2,OUTPUT);}
void loop(){
    if ( radio.available() ) {
     radio.read( data, sizeof(data));
```

```
a = data[0];
b = data[1];
e = data[2];
f = data[3];
g = data[4];
h = data[5];
i = data[6];
J = data[7];
M = data[8];
N = data[9];
if (J>125){
if (a >= 750||h==HIGH) {
  analogWrite(pwm1,i);
  analogWrite(pwm2,i);
  digitalWriteFast(mo0,LOW);
  digitalWriteFast(mo3,HIGH);
  digitalWriteFast(mo1,HIGH);
  digitalWriteFast(mo2,LOW);}
 else {
   analogWrite(pwm1,0);
   analogWrite(pwm2,0);}
 if(a<=25||f==HIGH){
  analogWrite(pwm1,i);
  analogWrite(pwm2,i);
  digitalWriteFast(mo0,HIGH);
  digitalWriteFast(mo3,LOW);
  digitalWriteFast(mo1,LOW);
  digitalWriteFast(mo2,HIGH);}
 if(b>=750||g==HIGH){
  analogWrite(pwm1,i);
  analogWrite(pwm2,i);
  digitalWriteFast(mo0,LOW);
  digitalWriteFast(mo3,LOW);
  digitalWriteFast(mo1,HIGH);
  digitalWriteFast(mo2,HIGH);}
```

```
if(b<=25||e==HIGH){
      analogWrite(pwm1,i);
      analogWrite(pwm2,i);
      digitalWriteFast(mo0,HIGH);
      digitalWriteFast(mo3,HIGH);
      digitalWriteFast(mo1,LOW);
      digitalWriteFast(mo2,LOW);}}
    else if(J<120){
    if (M>380) {  //backward
      analogWrite(pwm1,i);
      analogWrite(pwm2,i);
      digitalWriteFast(mo0,LOW);
      digitalWriteFast(mo3,HIGH);
      digitalWriteFast(mo1,HIGH);
      digitalWriteFast(mo2,LOW);}
      else {
       analogWrite(pwm1,0);
       analogWrite(pwm2,0);}
   if(M<320){  //forward
      analogWrite(pwm1,i);
      analogWrite(pwm2,i);
      digitalWriteFast(mo0,HIGH);
      digitalWriteFast(mo3,LOW);
      digitalWriteFast(mo1,LOW);
      digitalWriteFast(mo2,HIGH);}
    if(N>180){  //right
      analogWrite(pwm1,i);
      analogWrite(pwm2,i);
      digitalWriteFast(mo0,HIGH);
      digitalWriteFast(mo3,HIGH);
      digitalWriteFast(mo1,LOW);
      digitalWriteFast(mo2,LOW);}
    if(N<130){//left
      analogWrite(pwm1,i);
      analogWrite(pwm2,i);
      digitalWriteFast(mo0,LOW);
      digitalWriteFast(mo3,LOW);
      digitalWriteFast(mo1,HIGH);
              digitalWriteFast(mo2,HIGH);} } }}
```

For the Hand Gesture:

```
#include <SPI.h>
#include "NRF24L01.h"
#include "RF24.h"
#include "Wire.h"
#include "I2Cdev.h"
#include "MPU6050.h"
MPU6050 mpu;
int16_t ax, ay, az;
int16_t gx, gy, gz;
int led =8;
int data[2];
RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL;
void setup(void){
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();
  radio.begin();
  radio.openWritingPipe(pipe);   }
void loop(void){
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  data[0] = map(ax, -17000, 17000, 300, 400 ); //Send X axis data
  data[1] = map(ay, -17000, 17000, 100, 200);  //Send Y axis data
  radio.write(data, sizeof(data));
if(data[0]>380||data[0]<320||data[1]>180||data[1]<130){
      digitalWrite(led,HIGH);}
  else{
      digitalWrite(led,LOW);}
  }
```

For Vehicle with Hand Gesture:

```
#include <SPI.h>
#include "NRF24L01.h"
#include "RF24.h"
const int enbA = 3;
const int enbB = 6;
const int IN1 = 2;
const int IN2 = 4;
const int IN3 = 5;
const int IN4 = 7;
int RightSpd = 100;
int LeftSpd = 100;
int RightSpd1= 120;
int LeftSpd1 = 120;
int data[2];
RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL;
void setup(){
  pinMode(enbA, OUTPUT);
  pinMode(enbB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(1, pipe);
  radio.startListening(); }
void loop(){
  if (radio.available()){
    radio.read(data, sizeof(data));
    if( data[0] > 380){ //forward
      analogWrite(enbA, RightSpd);
      analogWrite(enbB, LeftSpd);
      digitalWrite(IN1, LOW);
```

```
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);}
  else{
    analogWrite(enbA, 0);
  analogWrite(enbB, 0);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);}
if( data[0] < 320){//backward
  analogWrite(enbA, RightSpd);
  analogWrite(enbB, LeftSpd);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);}
if( data[1] > 180){  //left
  analogWrite(enbA, RightSpd1);
  analogWrite(enbB, LeftSpd1);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);}
if(data[1] < 130){ //right
  analogWrite(enbA, RightSpd1);
  analogWrite(enbB, LeftSpd1);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);}
  }}
```

Bibliography

Bibliography

[1] https://www.arduino.cc/en/Guide/Introduction (01 february 2023)

[2] https://learn.sparkfun.com/tutorials/what-is-an-arduino/all (07 february 2023)

[3] https://www.elprocus.com/an-overview-of-arduino-nano-board/ (05 march 2023)

[4] https://predictabledesigns.com/how-to-choose-the-best-development-kit-the-ultimate-guide-for- beginners/ (02 april 2023)

[5] Bellili, S. Conception Et Réalisation D'un Robot D'inspection Des Pipelines À Base De Carte Arduino Uno [Mémoire de Master, Université Mohamed Khider - Biskra. (2019)] (09 april 2023)

[6] https://www.c-sharpcorner.com/UploadFile/167ad2/introducing-arduino-nano/ (13 march 2023)

[7] https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html (01 march 2023)

[8] https://www.c-sharpcorner.com/UploadFile/167ad2/introducing-arduino-nano/ (14 february 2023)

[9] https://microdigisoft.com/introduction-of-arduino-nano-board/ (20 february 2023)

[10] https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html (10 may 2023)

[11] http://www.farnell.com/datasheets/1682238.pdf (03 may 2023)

[12] https://www.rs-online.com/designspark/what-is-arduino-nano-a-getting-started-guide (15 march 2023)

[13] https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html (06 february 2023)

[14] https://www.circuito.io/blog/arduino-code (08 february 2023)

[15] https://www.makerspaces.com/simple-arduino-projects-beginners/ 16february2023

[16] https://stemify.weebly.com/programming-structure.html (01 march 2023)

[17] https://www.codeproject.com/Articles/1247684/Getting-Started-With-Arduino-Using-the-Small-Inexp (09 march 2023)

[18] https://www.elprocus.com/an-overview-of-arduino-nano-board/ (03 march 2023)

[19] https://projectiot123.com/2019/04/08/arduino-nano-for-beginners/ (08 march 2023)

[20] Benhamou, O., & Boudebza, B. Intitulé du sujet Réalisation d'un drone Quad-copter.(2019) http://e-biblio.univ-mosta.dz/handle/123456789/13139 (12 february 2023)

[21] http://domochris.canalblog.com/archives/2015/09/11/32611773.html (15february2023)
[22] http://www.energiazero.org/arduino_sensori/joystick_module.pdf (11 february 2023)
[23] fiche technique TEXAS INSTRUMENTS, LM340, LM340A, LM7805, LM7812, LM7815 SNOSBT0L –FEBRUARY 2000–REVISED SEPTEMBER 2016 (05 february 2023)
[24] https://www.majju.pk/product/12v-to-5v-lm7805-voltage-regulator-ic-5v-1a/(01 march 2023)
[25] Advanced Monolithic Systems AMS1117, 1A LOW DROPOUT VOLTAGE REGULATOR (05 march 2023)
[26] https://byjus.com/physics/dc-motor/ (13 february 2023)
[27] Baddou Mohamed. Robot superviseur [Projet fin d'étude, Université Mohamed V de Rabat. (2016)] (06 february 2023)
[28] https://www.electronicshub.org/wireless-communication-introduction-types-applications/ (01 may 2023)
[29] https://www.carnetdumaker.net/articles/communiquer-sans-fil-avec-un-module-NRF24L01-la-bibliotheque-mirf-et-une-carte-arduino-genuino/#le-montage (15 march 2023)
[30] http://pignol.univ-tln.fr/DUT_GEII/DevCompProj/RF24/FS_RF24.pdf (06 march 2023)
[31] https://lastminuteengineers.com/NRF24L01-arduino-wireless-communication/ (04 february 2023)
[32] https://passionelectronique.fr/tutorial-NRF24L01/ (16 march 2023)
[33] https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-NRF24L01-tutorial/ (13 february 2023)
[34] https://resources.altium.com/fr/p/what-is-a-pcb (14 march 2023)
[35] https://learn.sparkfun.com/tutorials/pcb-basics/all (10 february 2023)
[36] https://maker.pro/pcb/tutorial/pcb-design-tutorial-using-easyeda-jlcpcb (03 march 2023)
[37] https://www.wevolver.com/article/the-printed-circuit-board-design-and-manufacturing-cycle-symbiotic-relationships-engage-innovation (04 may 2023)