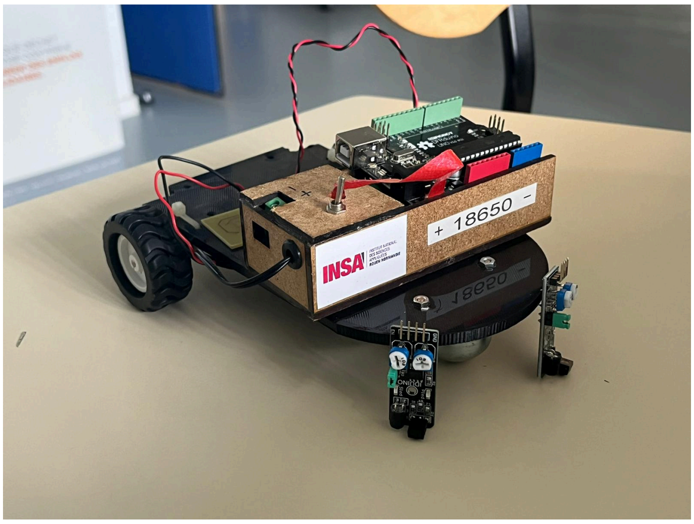
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN

Projet de Physique P6

STPI2/P6/2024 – 43

Programming the Arduino for a line-following robot



Nb:The project not being finished, we disposed the different components to have an idea on how it should look

Etudiants :

Yasser HINANE **Ali IKCHED**

Remi ROZIER **Mathilde TANQUEREL**

Enseignant-responsable du projet :

Simon BUSBRIDGE

Cette page est laissée intentionnellement vierge.

Date de remise du rapport : ***14/06/2024***

Référence du projet : ***STPI/P6/2024 – 043***

Intitulé du projet : ***Programming the Arduino for a line-following robot***

Type de projet : ***Programming and matériel***

Objectifs du projet :

- ***Learn how to use the Arduino***
- ***Learn how to use FreePCB***
- ***Learn how to use LTspice***
- ***Learn how to modelize with CATIA***
- ***Assemble a line-following robot***
- ***Use the Arduino for the robot***

Mots-clefs du projet :

- Arduino
- Robot
- Programming
- Assembly

Dear Mr. Busbridge, Mr. Williams, and Mr. Michael,

We would like to express our deepest gratitude for the invaluable support you provided us during the completion of our line-following robot project, as part of our second-year engineering studies.

Mr. Busbridge, your technical expertise and insightful advice were crucial to the success of this project. Your guidance, patience, and clarity helped us overcome the numerous challenges we faced throughout this scientific endeavor. Your passion for robotics has been a true inspiration for each of us.

Mr. Williams and Mr. Michael, your rigorous supervision and valuable feedback allowed us to deepen our knowledge and develop essential engineering skills. Your availability and kindness created an environment conducive to learning and innovation. Thanks to you, we were able to successfully complete this ambitious and rewarding project.

This line-following robot project not only enabled us to acquire technical skills but also strengthened our teamwork and problem-solving abilities. We are convinced that the lessons learned from this experience will be a valuable asset in our future careers.

Once again, we sincerely thank you for your support, guidance, and dedication. We hope to have the opportunity to work with you again in the future.

With our deepest appreciation,

Yasser Hinane, Ali Ikched, Rémi Rozier and Mathilde Tanquerel.

Table of contents

1. Introduction

In the unexpectedly advancing subject of robotics, the improvement of independent structures able to appear unique obligations with minimum human intervention is of paramount importance. Our task specializes in designing and constructing a line-following robot using the Arduino platform. This undertaking now no longer most effectively showcases the sensible packages of robotics however additionally serves as a hands-on getting to know enjoyment in electronics, programming, and automation.

The line-following robot is a form of independent cell robotic designed to comply with a predetermined path, normally marked by means of a high-evaluation line at the floor. This capability has tremendous real-global packages, which includes automatic shipping structures in warehouses, clever manufacturing, and carrier robots in diverse industries.

Our task goals to attain the subsequent objectives:

1- Design and Construction: To layout and assemble a strong mechanical framework for the robot, making sure it can navigate diverse music situations effectively.

2-Sensor Integration: To put into effect sensor generation that lets the robot locate and comply with a line. Typically, this entails the usage of infrared sensors or cameras which can differentiate among the road and the heritage surface.

3 -Programming and Control: To expand the software program the usage of the Arduino platform, allowing the robotic to method sensor data, make decisions, and manipulate vehicles to comply with the road accurately.

4- Optimization and Testing: To take a look at and refine the robotic`s overall performance thru iterative improvement, making sure it may deal with unique music layouts and barriers efficiently.

This task is a mix of hardware and software program components, requiring understanding in regions which includes circuit layout, programming, and gadget integration. By leveraging the flexibility and simplicity of use of the Arduino platform, we aim to create a powerful and dependable line-following robot that demonstrates the ideas of independent navigation and manipulate structures.

Ultimately, this task now no longer most effectively complements our expertise of robot structures however additionally prepares us for greater complicated demanding situations withinside the realm of independent technologies. Through this line-following robotics, we aspire to make a contribution to the developing subject of robotics and encourage destiny improvements in automatic structures.

2. Methodology / Work Organization

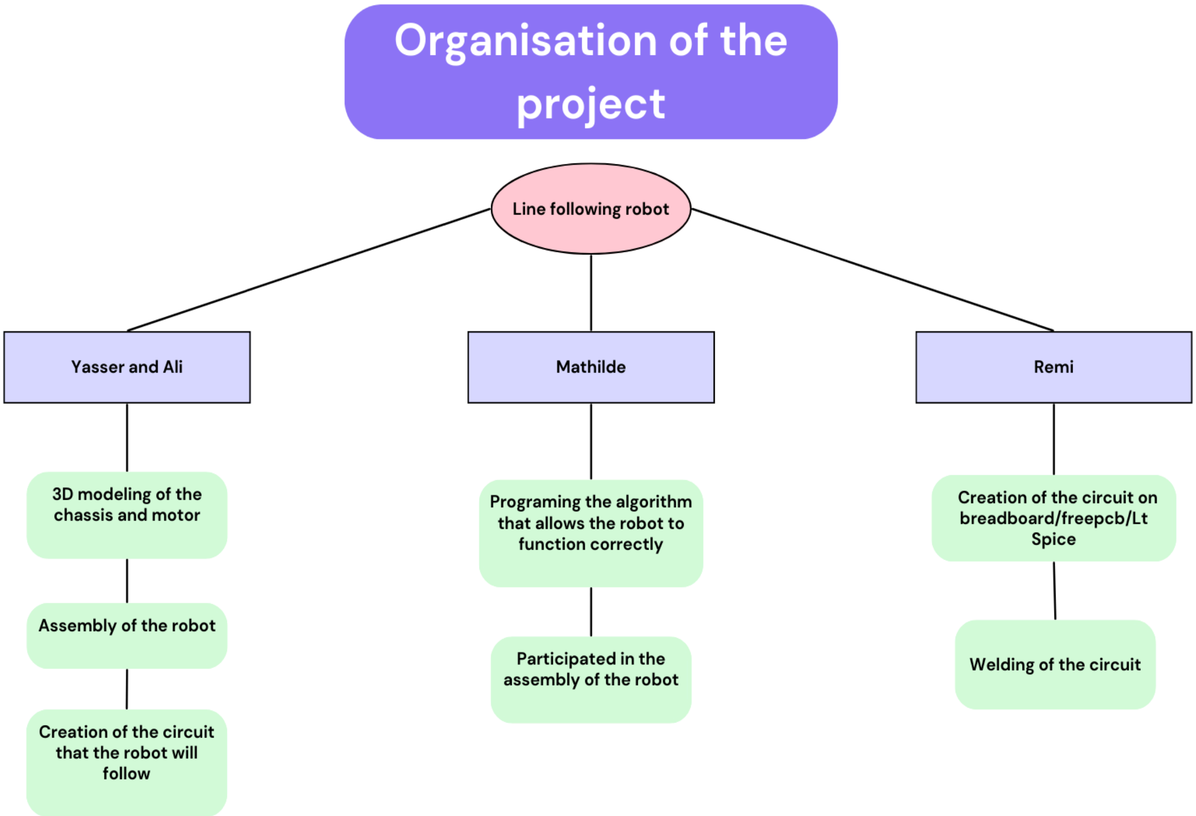
To carry out this work, we divided the different parts of the project and the writing of the report among us, while pooling our research to ensure better coherence in writing. Here is the distribution of tasks among the team members:

1- Yasser and Ali took care of the 3D modeling of the chassis and motor, as well as the assembly of the robot and the creation of the circuit that the robot will follow. Their contribution includes the detailed design of the robot.

2- Mathilde was responsible for the programming part. She programmed the algorithm that allows the robot to function correctly. Additionally, she participated in the assembly of the robot.

3- Remi handled the electrical circuit design, which is essential for powering the motors and is a crucial part of the project.

Thanks to this organization, we were able to delve deeply into each aspect of the project while ensuring coherence and uniformity in the final report. This collaborative approach allowed us to make the most of each team member's skills and knowledge.



3. Work completed and Results:

3.1. Robot:

First of all, Ali and Yasser were in charge of getting the main components required for the robot. For that purpose, they went to M.PASCAL to know if it was possible to borrow some of the components. After having discussed it with the other members of the group, they finally kept this list of components:

- **Two rear-driven wheels :** Essential for the propulsion, these wheels are coupled to 2 different engines in order to steer the robot by adjusting the rotation speed of each wheel (See figure 1)

- **A 360-degree omnidirectional front wheel :** This piece was important for the different turns that the robot had to perform. At first, we thought of using pivot wheels that you can find in strollers for example but ended up choosing this omnidirectional wheel as it was easier to set up (the piece being handed by M.PASCAL).This front wheel allows the robot to pivot in place and perform tight turns thus offering great maneuverability essential for following curved lines or making quick trajectory corrections.(See figure 1)

Figure 1. THE WHEELS

- **2 Obstacle Avoidance Sensors (ST1081) :**Positioned at the front of the robot, they transmit data to the Arduino board for guidance. This infrared sensor is adaptable and highly precise, equipped with a pair of infrared transmitters and receivers. The transmitter tubes emit a specific frequency of infrared light. When detecting the presence of an obstacle (reflector), the infrared receiver tube detects the reflected infrared light. When the sensor detects the reflection, it outputs a digital signal through its signal output interface. This signal can be adjusted using the potentiometer knob.

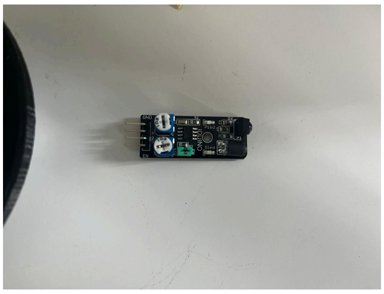


Figure 2. The sensors

- **Electrical wires :** Electrical wires that connect the different electronic components of the robot and assure the energy and data transfer between the sensors, the engines and the Arduino Card. (voir figure 3)

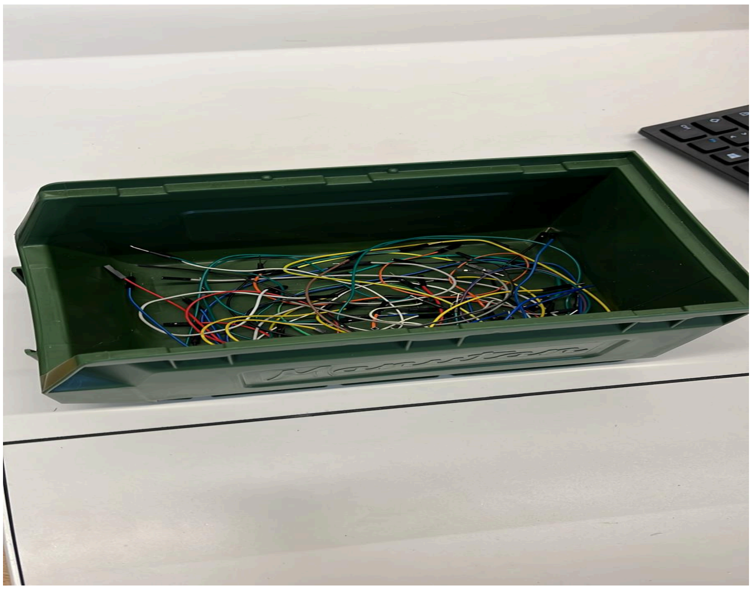


Figure 3. Electricals wires

- **An Arduino Card for the control :** This whole project relies on the Arduino card that is the “brain’’ of the robot. It is a programmable microcontroller that receives data from the sensors and sends commands to the motors to steer the robot. This will then allow us to program the “line-following” code for the robot to navigate the course.

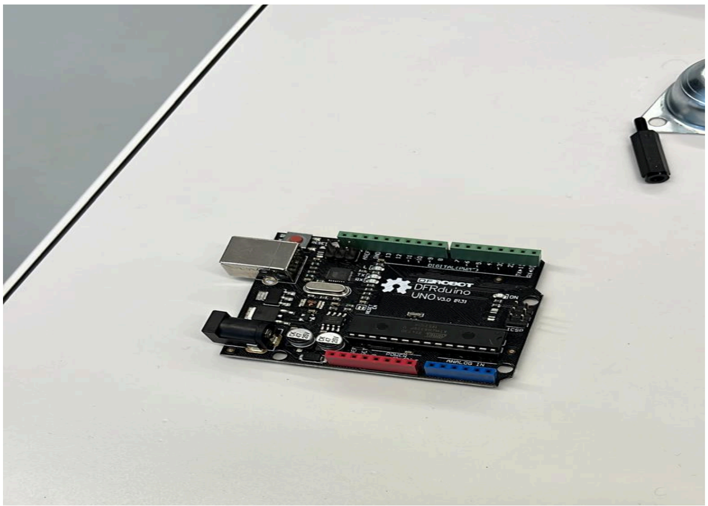


Figure 4. The arduino card

- **A battery-based generator :**The generator provides the energy required for the main components.

Figure 5. The generator

Ali and Yasser then tried to make the chassis. They first sketched a draft and computed the correct dimensions to know where to place each component.

In order to make this chassis, they had 2 different solutions:

1. **Take a wooden board and cut it using a laser :** This method, even if it is really fast and effective, doesn't allow as much precision and flexibility in the conception of complex forms
2. **Make a 3D model and print it :** This solution allows a conception very accurate and personalized. Although more costly in terms of time and resources, it provides a better final result.

They finally chose the second option and therefore had to get used to SolidWorks for this task. However, having difficulties finding an activation key, they preferred to use the software CATIA instead.

3.1.2 Robot Model Design using CATIA

The chassis was conceived as a rectangle with an arc at the front to fix the front wheel and the captors easier. The dimensions were 160 cm in length, 100 cm in width and 5 mm in height. This conception was suited to offer stability to the robot without overloading it in terms of weight.

1. **Basic sketch :**
   - We began by creating a 2D sketch representing the top view of the chassis using tools like the rectangle and the arc to draw the wanted forms.
   - The dimensions were specified according to our needs (160x100cm)
2. **Extrusion of the sketch :**
   - We then extruded the croquis to give depth and create volume to our chassis. The height has been set at 5mm.

Verifications and adjustments :

1. **Analysis of the conception :**
    - We used the analysis tools of CATIA to check the strength and stability of the conception by making sure it fits the mechanical restraints.
2. **Eventual adjustments :**
    - We finished with the final adjustments to optimize the structure, the balance of masses and the accessibility of the components

Preparation for the 3D printing :

1. **Exportation of the file :**
    - Once the conception was finished, we exported the model in a format compatible with the 3D printer at the INSA (here we used STL)
2. **Preparation of the model for the printing:**
    - We verified and prepared the model for the 3D printing using cutting softwares such as Cura or Simplify3D. This includes the orientation of the model, the management of the supports and the cutting in slices for the printing.

Advantages of CATIA :

- **Precision :** CATIA offers precise modeling tools allowing to create complex forms with high precision
- **Visualization:** The capacity to visualize the 3D model helping to anticipate potential issues and optimize the conception before printing anything.
- **Analysis:** The analysis tools integrated help check the strength and stability of the conception prior to its creation, saving a lot of time.

By using CATIA, we managed to create a strong and functional chassis (see figure .) ready to be 3D printed. This approach allowed us to have precise and compliant results according to our needs for this line-following robot.

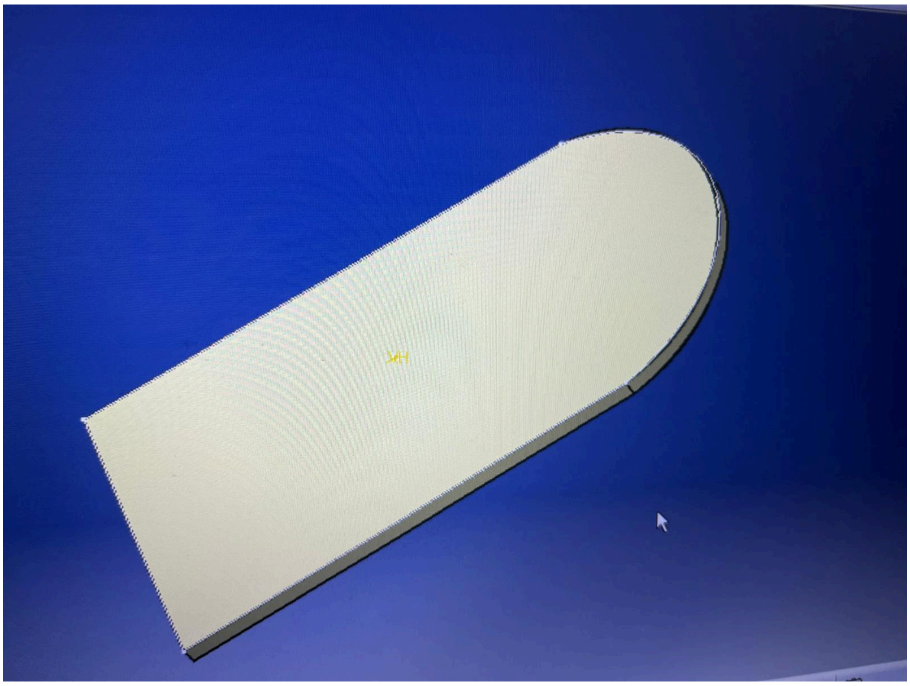


Figure 6. Model of the chassis in CATIA

After carrying out the modeling on CATIA, we used a 3D printer to produce the physical chassis of the robot. 3D printing allowed us to materialize our design precisely and quickly.

Once the chassis was printed, we proceeded with assembling the components. The motors were attached to the rear of the chassis, the line sensors were positioned at the front, and the Arduino board was installed in the center of the chassis for easy wiring and port access.

The robot thus assembled is ready to be programmed to follow a line. The line sensors will detect the contrast between the line and the surrounding surface, allowing the robot to follow the traced path. Based on the information received, the Arduino board will send instructions to the motors to adjust the robot's trajectory and keep it on the line.

This integrated approach, from design to 3D printing and final assembly, allowed us to create a custom-made line follower robot, tailored to our specific needs. This project combines skills in mechanical design, additive manufacturing and programming to create an autonomous and functional robot.

3.1.3 Assemblage du robot

To finally assemble the robot, Ali and Yasser had to wait until all the components were ready. Once everything was prepared, they began connecting all the parts. However, they encountered several issues during the assembly:

Alignment of Components:

They found that some holes for the screws were not properly aligned with the component brackets. They had to adjust the position of the holes by re-drilling the chassis to ensure precise and secure mounting of the components.

Wire Length:

The wires provided were too short to reach some connections. They had to solder extension wires to make reliable connections between the sensors, motors and the Arduino.

Fixing the Motors:

The engine mounts were not strong enough to hold the engines in place during operation. They reinforced the brackets with additional screws and zip ties to ensure stability.

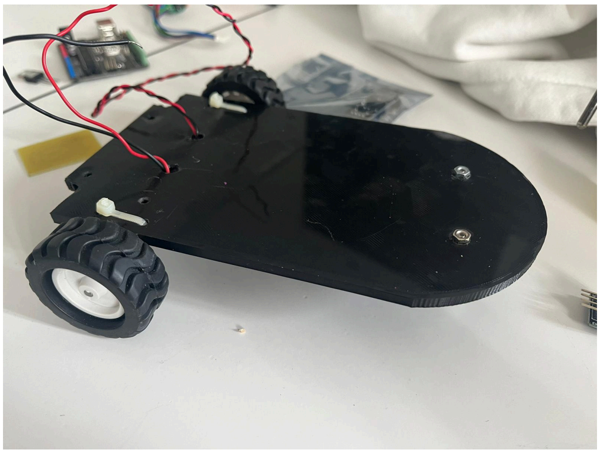


Figure 7. The chassis

3.2 Informatique:

3.2.1 The Arduino IDE.

Arduino IDE (Integrated Development Environment) is an open source development platform designed for programming Arduino microcontrollers. The Arduino IDE allows you to write programs, using a programming language based on C/C++. These programs are transcribed into the Arduino board and then uploaded to the Arduino board to perform particular tasks. The software offers a simple user interface, which includes a text editor for writing code, a button for compiling the code and another for adding it to the associated Arduino board.

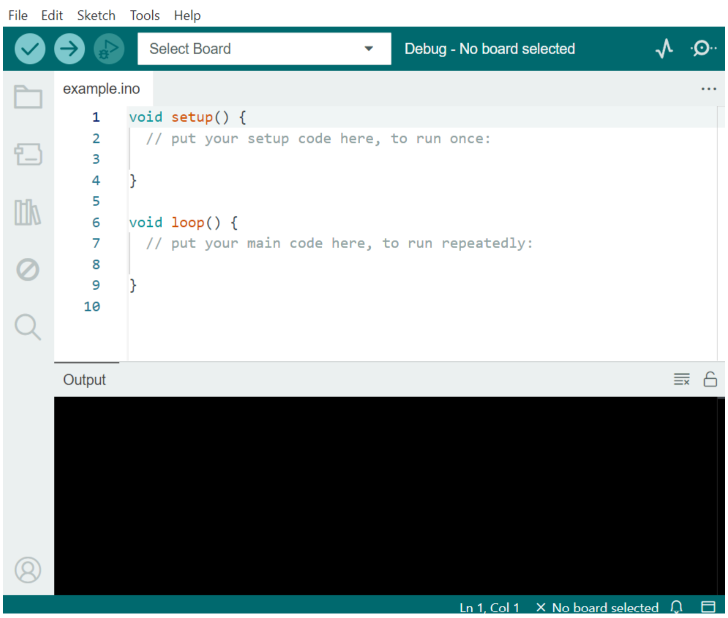


Figure 8. The arduino board, PWM pins are framed in red

3.2.2 The programmation

We never learned the Arduino language or C/C++. We therefore learned thanks to resources on the internet, in particular the basics thanks to the “developez.com” site [1]. In addition, we conducted research on specific points that we wanted to explore further.

The Arduino program can be divided into three parts: the first allows the declaration of variables, the second is intended for the initialization of the inputs/outputs, and finally the main part which executes in a loop.

In the first part of the program, each variable is associated with a specific pin number on the Arduino board. The variables corresponding to the motors are connected to pin numbers that correspond to the PWM pins on the Arduino, it's indicated by the ~.

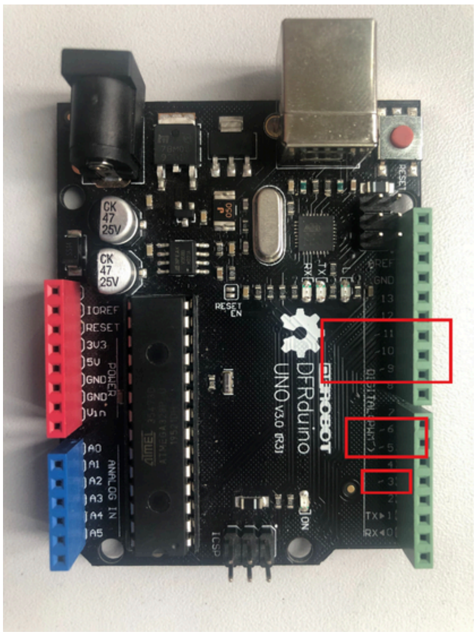


Figure 9. The arduino Card, PWM pins are framed in red

Pulse Width Modulation (PWM) is a method that adjusts the power sent to the motors by changing the time during which the electrical signal is on compared to when it is off. It allows us to change the speed of the motors.

The initialization part, indicated by the setup() function, configures the pins of the left and right sensors as INPUT, allowing the Arduino to read information from the sensors. The motor pins are configured as OUTPUT, allowing the Arduino to send signals to make the robot move.

In the last part, the loop() function runs continuously after setup(). It reads the states of the sensors and determines the robot's action based on these states. The actions are represented by four functions:

- The d() function makes the robot turn right by activating the right motor at a certain power.
- The g() function makes the robot turn left by activating the left motor in the same way.
- The av() function makes the robot move straight ahead by activating both motors at the same power.
- The arr() function stops all the motors by setting all controls to LOW, which turns off the motors.

The `analogWrite()` function, which is used in these functions, allows us to control the speed of the robot. Its second parameter corresponds to the duty cycle value of the PWM signal, which ranges from 0 (a signal that is always off) to 255 (a signal that is always on). The value 20 corresponds to a duty cycle of 20/255, which is approximately 7.8%. This means that the PWM signal is at a high level 7.8% of the time and at a low level 92.2% of the time.

We decide to take a line whose size is larger than the robot.

3.2.3 Improvement clue:

Our code is configured such that the robot is only capable of making sharp turns. In fact, we have programmed the system so that when the robot turns, one wheel stops while the other continues to turn. However, this method is not optimal for wider turns. It would have been beneficial to develop different turning functions suited to varying degrees of curvature.

For this, it would have been necessary to install more sensors to better detect the severity of the turn. If the sensors detect a shallow turn, instead of completely stopping a wheel to turn, it should have simply reduced its speed. This would have required carrying out tests to determine the optimal speed of the wheel to follow less tight turns

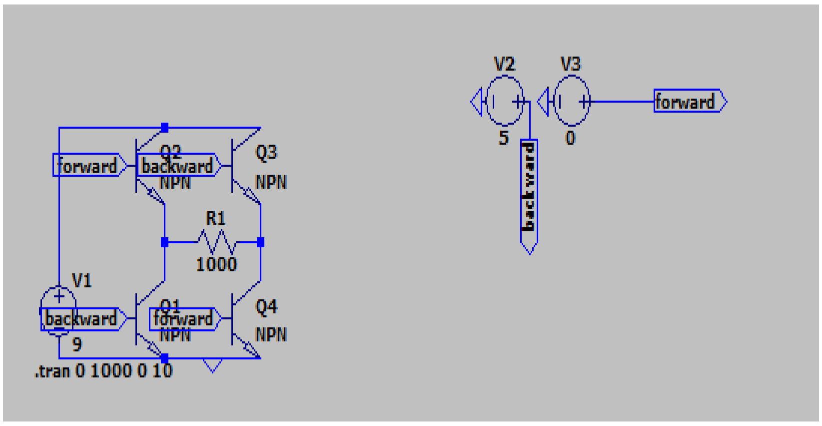
3.2.4 Difficulties

We didn't know C++, so we had to take help from online resources to learn it. Additionally, we were unable to test the program. Indeed, for the electrical part we encountered certain problems that we managed to resolve only very late. The remaining time did not allow us to test the program, which would have been useful for adjusting certain aspects, such as motor speeds.

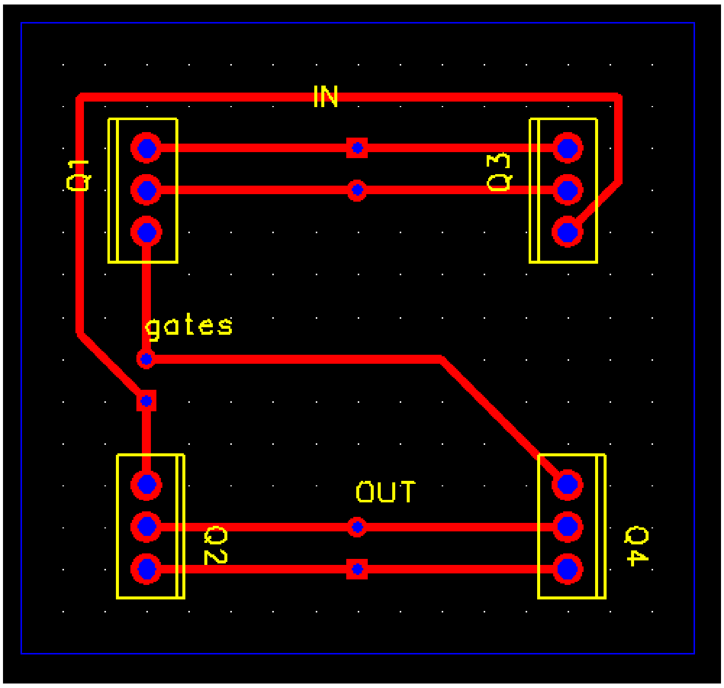
3.3. Electricity:

3.3.1 the first circuit:

Once we had decided to make a line-following robot using Arduino, Remi was the one in charge of making the required circuit, that is to say 2 h-bridges circuits using the transistors M.BUSBRIDGE gave us. Being completely new with the process of making circuits, Remi had to learn about lt spice and how to use it and we encountered loads of issues. The main issue was that he forgot the first crucial step:making the circuit on a breadboard first and testing it. This therefore led to the printing of the circuit without a prior test on the breadboard. First, he thought of making a version with a basic NPN transistor fyi a h-bridge is an electrical component with the ability but our supervising professor informed him that he couldn’t work with that. In fact, at that time, M.BUSBRIDGE came to INSA and gave him some transistors to work with. He told him these were pmos and nmos mosfets transistors and, at the time, he did not think they might have different pinout, he then made another version and completely forgot there ever had been 2 different transistors types while making the circuits. The circuits were made using FreePCB, so Rémi had to learn how to use it, add components to the circuits, and connect to external parts. For the first circuits, headers were used to facilitate connections.



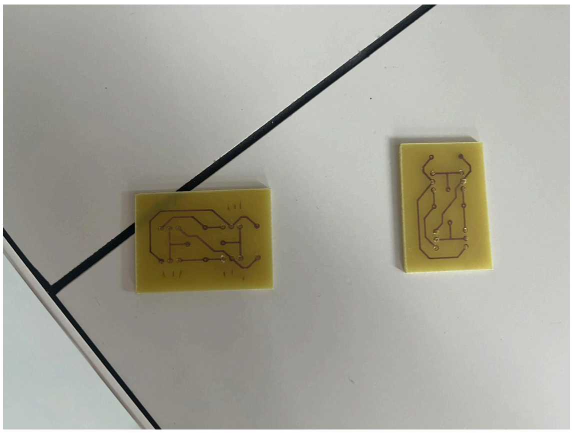
this circuit simulate the motor with a resistor and the arduino control with generators than be switched on and off manually



Once printed there were a few issues, first some of the holes had not been fully dug through the plate and a copper width of 20 mil turned out to be too thin, making it very easy to damage, on the hole had been corrected, came the issue of soldering, as it was remi's first time soldering there were a few troubles as for how to do it and how much tin to put, the soldering process was realized with tin at 445°C, all of this further increased the time it took to realize there was something wrong with the circuit.

3.3.2 the second circuit:

The first circuits were made and failed. After investigating why, he finally figured out that the shape of the transistors were different for a reason. He then found the datasheet for the pmos, it was actually an irf9630 but he read the other as irf690 while it was actually irf630 which led to a lot of trouble as he didn't find anything corresponding. After numerous checks, he finally found out the correct reference and its corresponding datasheet. This failure made him realize that the testing phase on the breadboard was crucial. He therefore decided to test the set up on a breadboard, and got very weird readings for the irf9630 as it was a pmos and thus worked the opposite of what he was expecting. Knowing that he was testing the circuit with a stronger current than the one we would be using on the arduino he then thought that he might have overloaded the transistor and the current was just forcing his way through. Further testing proved him wrong, and got him even more confused. As time was running out, he made another version of the circuits(annex 3.4-3.6), sent it for printing and asked M.BUSBRIDGE to check it. Unfortunately, what he had forgotten to mention was that he was using resistors as input, meaning the resistor was on the circuits but would never actually be added and would be replaced by cables connected to the arduino and the motor. Since Remi did not mention that, when M.BUSBRIDGE talked about the resistor in his answer he misunderstood the rest of his response. M.BUSBRIDGE then explained to him in a way he understood what was wrong with the circuit. As a last resort, he tried moving the cable and resistor in every way possible, managing to make it work in one direction, but not both at the same time, he then realized that was actually not a concern as the robot had no need to move backward.



4. Conclusions and Future Perspectives:

4.1 Conclusions on the Work Accomplished:

The project of designing and developing a line-following robot during our second year of engineering studies has been a significant and educational experience. It enabled us to merge theoretical concepts with practical skills across various domains including electronics, programming, mechanical design, and additive manufacturing.

Component Procurement: Initially, we identified and procured essential components such as the rear drive wheels, a 360-degree omnidirectional front wheel, line sensors, connecting wires, an Arduino board, and a power generator. This phase taught us the importance of meticulous planning and effective resource management for a technical project.

Chassis Modeling: We use CATIA for the 3D modeling of the robot chassis. This software allowed us to create a detailed and precise structure that could effectively accommodate all the components. The modeling process involved optimizing the dimensions and placements of the components to ensure stability and functionality, along with making necessary adjustments based on our design requirements.

3D Printing and Assembly: Post-design, we proceeded with 3D printing the chassis. This approach provided us with a custom, accurately manufactured piece. During the assembly, we encountered and overcame various technical challenges, such as aligning the mounting holes, extending the wires for proper connections, securing the motors, and managing electronic interference. These challenges enhanced our problem-solving abilities and adaptability.

Programming: Programming the robot was a crucial step, where we developed code that enabled the robot to follow a line by interpreting sensor data and controlling the motors accordingly. We learned how to code with Arduino.

Learning Outcomes: This project offered a comprehensive practical application of our theoretical knowledge and helped us develop new skills in several key areas:

- **Electronics and Programming:** We learned to interface sensors with the Arduino board and to write control programs for autonomous robots.
- **Mechanical Design:** We gained insights into the constraints and possibilities of 3D modeling and additive manufacturing.
- **Project Management:** We improved our skills in planning, resource management, teamwork, and problem-solving throughout the project.

4.2 Personal Contributions and Benefits from this Project Course:

Participating in the line-following robot project has significantly contributed to my personal and professional growth. Here are the key areas where this project has impacted my development:

Enhanced Technical Skills: This project has provided me with hands-on experience and enhanced my technical skills in several crucial areas

Project Management: In fact, our project management was a bit neglected which caused a lot of issues, like the project that didn't completely work on time and made us realize how important it really was. Working on this project required effective planning, time management, and resource allocation. Through this, we developed skills in:

- **Planning and Organization:** Setting milestones, scheduling tasks, and ensuring timely completion of each project phase.
- **Problem-Solving:** Addressing and overcoming unexpected challenges, such as technical glitches during assembly and programming errors.
- **Team Collaboration:** Working closely with my teammates, Ali and Yasser, taught me the importance of communication, coordination, and leveraging each other's strengths to achieve a common goal.

Practical Application of Theoretical Knowledge: The project served as a bridge between theoretical concepts learned in the classroom and their practical application. We applied principles from:

- **Control Systems:** Understanding and implementing feedback loops and control algorithms to keep the robot on track.
- **Electronics and Circuit Design:** Designing and assembling circuits that connect sensors, motors, and the Arduino board.

Innovation and Creativity: Designing and developing a custom robot chassis from scratch required creativity and innovation. We learned to:

- **Design Custom Solutions:** Creating a unique chassis design that meets the specific requirements of the project.
- **Optimize Design:** Iterating on the design to improve stability, functionality, and ease of assembly.

Confidence and Independence: Successfully completing this project has boosted our confidence in our engineering abilities. It demonstrated that we can independently handle complex tasks from concept to execution. We Feel more prepared and self-assured in tackling future engineering challenges.

Career Readiness: The skills and experiences gained from this project are directly applicable to our future career in engineering. We now have practical experience in several key areas of robotics and automation, which are highly valuable in the industry.

4.3 Future Perspectives for the Continuation of this Project:

Continuing the line-following robot project offers several exciting opportunities for improvement and expansion. Here are some simple perspectives for future development:

Advanced Sensor Integration:

- Adding more advanced sensors to improve line detection accuracy and obstacle avoidance.

Improved Control Algorithms:

- Implementing more sophisticated algorithms for better path planning and navigation.

Enhanced Mobility:

- Upgrading the wheels and motors for smoother and more efficient movement on various surfaces.

Wireless Communication:

- Incorporating Bluetooth or Wi-Fi modules for remote control and monitoring via a mobile app.

Energy Efficiency:

- Optimizing the power supply to extend battery life and reduce energy consumption.

Additional Functionalities:

- Adding features like object recognition and environmental interaction for broader applications.

Educational Use:

- Using the improved robot as a teaching tool to help students learn about robotics and programming.

Collaboration and Open Source:

- Sharing the project as an open-source initiative to encourage community contributions and enhancements.

These improvements can make the robot more versatile, efficient, and useful for educational and practical applications.

5.Bibliographie et annexes

5.1Bibliographie

[1]https://arduino.developpez.com/tutoriels/cours-complet-arduino/?page=programmer-arduino

5.2 Annexes :

5.2.1 Codes:

Annexe 1: The Arduino code

```
// Code written by Mathilde Tanquerel, for the project.
// Declaration of variables
int CG = 1; // Pin for the left sensor
int CD = 2; // Pin for the right sensor
int MGA = 11; // Pin for left motor forward
int MGR = 10; // Pin for left motor reverse
int MDA = 5; // Pin for right motor forward
int MDR = 6; // Pin for right motor reverse
// Initialization
void setup() {
    pinMode(CG, INPUT); // Arduino receives information
    pinMode(CD, INPUT);
    pinMode(MGA, OUTPUT); // Arduino sends information
    pinMode(MGR, OUTPUT);
    pinMode(MDA, OUTPUT);
    pinMode(MDR, OUTPUT);
}
// Functions
void d() { // Turn right
    analogWrite(MDA, 20);
    digitalWrite(MDR, LOW);
    digitalWrite(MGA, LOW);
    digitalWrite(MGR, LOW);
```

```
}
void g() { // Turn left
    digitalWrite(MDA, LOW);
    digitalWrite(MDR, LOW);
    analogWrite(MGA, 20);
    digitalWrite(MGR, LOW);
}
void av() { // Move forward
    analogWrite(MDA, 20);
    digitalWrite(MDR, LOW);
    analogWrite(MGA, 20);
    digitalWrite(MGR, LOW);
}
void arr() { // Stop
    digitalWrite(MDA, LOW);
    digitalWrite(MDR, LOW);
    digitalWrite(MGA, LOW);
    digitalWrite(MGR, LOW);
}
// Main loop
void loop() {
    int capteurDroit = digitalRead(CG);
    int capteurGauche = digitalRead(CD);
     if (capteurGauche == HIGH && capteurDroit == HIGH) { //the left and right sensor detect the black line
        av(); //the robot moves forward
    }
     else if (capteurGauche == HIGH && capteurDroit == LOW) { //the left sensor detects the black line
        g(); //the robot turns left
    }
```

```
    else if (capteurGauche == LOW && capteurDroit == HIGH) { //the right sensor detects the black line
        d(); //the robot turns right
    }
    else {
        arr(); //the robot stops
    }
}
```

5.1.2. Technical Documentation

Datasheet for the sensors:

https://fr.scribd.com/document/696677169/fiche-technique-1485307-iduino-1485307-capteur-infrarouge-adapte-pour-ordinateur-monocarte-arduino-1-pcs

Datasheet for the arduino:

https://www.application-datasheet.com/pdf/dfrobot/dfr0216.pdf