Stepper Motor

Control & Drive

Ayman Miri & Christian Steffen
LYCÉE GUILLAUME KROLL 32, rue Henri Koch

Table des matières

1 Introduction

1.1 Describe the mission

- Stepper motor project : We got this project in our second semester, our task was controlling the motor. As we know, controlling a stepper motor needs a driver and a microcontroller, now we can say it's easy until you hear that you are not allowed to use library and a driver who has a translation module inside of it and the motor you are going to drive is a bipolar motor, we all know that this type of motor requires a lot of work to be driven like software and hardware.
- The main task was driving the motor with four buttons and each button got his own task like first button will move the motor 10 steps forward and second one will move it just one step for the other two button it's just the opposite.

1.2 How it goes

- We started first with the time planning and dived the mission between us, then searching about the materials that we are going to use like the type of driver and the microcontroller also, and of course looking how the bipolar motor works.
- We decided to use the driver l298n because it has two full **H bridge circuits**, and he got no translation module.
- Then the choice of the microcontroller was of course the (Arduino UNO) not because it has more Ram and speed then other, but it's just because we had it already.
- After the decision on the time planning, and the search that we effected on the materials, we started the planning process that you will see coming on the chapter of time planning.
- We found a lot of sources on internet how to drive the stepper motor "Bipolar", it was easy once you understood the logic behind it, but here we faced a problem, to us it was a problem but to our teacher it wasn't, the problem is that when we programed the microcontroller we used the function **delay** between each step, and we wanted to use an oled in addition to the project, the oled wasn't in the task but we added to just to make it more interactive. And of course, when you use **delay**, you block the whole program until this delay is done. So, we decided to use **millis()** function, but also, we faced a problem in controlling the output of Arduino, because **millis()** It is like a process aside of your main process.
- We also decided to build box to contain the project and it wasn't required to.

2 Material List

Equipment	Type	Nr.
Arduino	Uno (SMD)	1
Stepper Motor (Bipolar)	SM-17HS4023	1
Driver	L298N	1
Power-Supply	MW MB10EU Variant output DC voltage: 3V > 12V Output current : 1000mA	1
Jumper cable	Male/Female , Male/Male	x
Button	Pushbutton	5
Resistance	10k ohm	5
OLED	ADA938 OLED 128x64	1
3D Printer	Artillery	
Filament	PETG	

3 Software

Name	Description
IDLE Arduino	Used to write the program
Autodesk Eagle	Used to draw the circuits
Autodesk Fusion 360	Used to draw the 3D design
Ultimaker Cura	Used to generate the (.gcode) extension to use on the 3D printer
Draw.io	Used to draw the flowchart

4 Time planning

- For time planning it was changed one time, I'm going to show you the first one then the modified one.

5 Theory

5.1 Driver

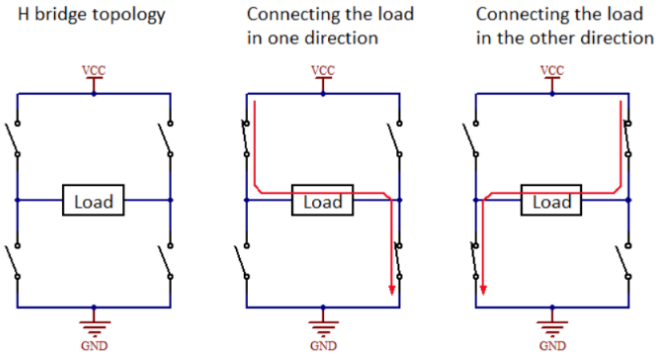
- To have a complete control over a motor you must control his speed and rotation direction. This can be achieved by combining these two techniques :
  - **PWM** – to control speed.
  - **H-Bridge** – to control the rotation direction.

- **Control Speed** : On the Arduino we didn't use the PWM Pins, but we wrote a program to generate digital pulses in controlling the time (frequency). As the digital pulses increase in frequency, the step movement changes into continuous rotation, with the speed of rotation directly proportional to the frequency of the pulses.

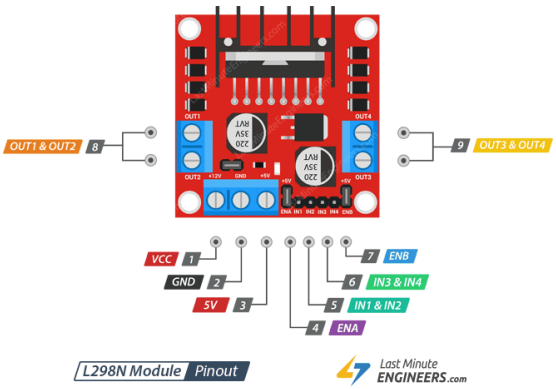
- **Rotation Direction** : we just reversed how the Pins were activated on the coils of the motor that connected to two H-Bridge circuits.

- **How H-Bridge circuit works** : The spinning direction of any DC motor can be controlled by changing the polarity of its input voltage. A common technique for doing this is to use an H-bridge. Closing two specific switches at a time reverses the polarity of the voltage applied to the motor. This causes a change in the spinning direction of the motor.

  The following images show the working of the H-bridge circuit.



- Here also an image of the L298n Describe his inputs and outputs connections.



5.2 Stepper Motor

5.2.1 What is a Stepper motor?

- A stepper motor is a brushless, synchronous electric motor that converts digital pulses into mechanical shaft rotation. Its normal shaft motion consists of discrete angular movements of essentially uniform magnitude when driven from sequentially switched DC power supply.

5.2.2 How does a stepper motor work?

- Every revolution of the stepper motor is divided into a discrete number of steps, in many cases 200 steps, and the motor must be sent a separate pulse for each step. The stepper motor can only take one step at a time and each step is the same size.

- Since each pulse causes the motor to rotate a precise angle, typically 1.8°, the motor's position can be controlled without any feedback mechanism. As the digital pulses increase in frequency, the step movement changes into continuous rotation, with the speed of rotation directly proportional to the frequency of the pulses.

5.2.3 Types of Stepper motor

- There are two types of Steppers motor, Unipolar and Bipolar, in our project we used a Bipolar one. As we know the bipolar requires a lot of software and hardware, and we are going about the software that we used and the hardware.
- We start with the hardware since we know it's a little bit confusing for some of people :
  - In this motor we got 4 wires connected to two coils, first to of them go to first coil and call (A- & A+), and the other two go to second coil and call (B- & B+).
  - We connect these 4 wires to the driver l298n how has two built-in H-Bridge circuits, each coil connected to an H-Bridge circuit. See in **Figure 1**.

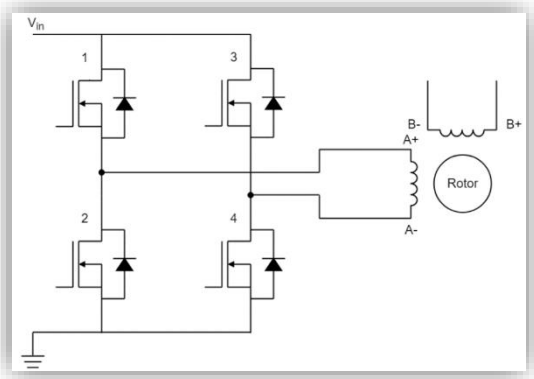
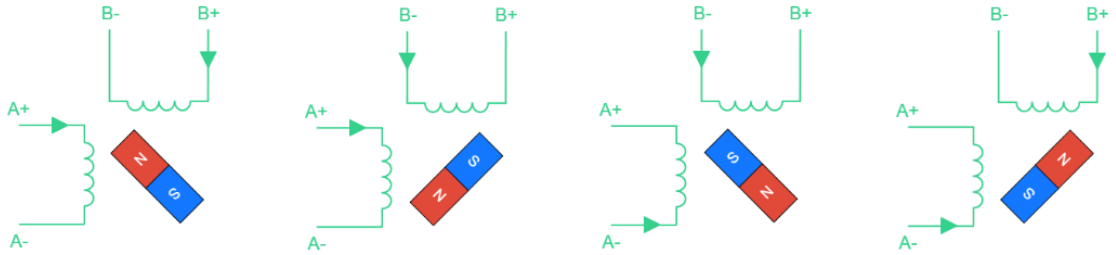


Figure 1: example of coil connected to H-Bridge circuit.

5.2.4 Stepper motor driving techniques

- There are 4 different driving techniques for a stepper motor :
  1. Wave mode : only one phase at a time is energized.
  2. Full-step mode : two phases are always energized at the same time.
  3. Half-step mode : is a combination of wave and full-step modes.
  4. Microstepping mode : can be seen as a further enhancement of half-step mode because it allows to reduce even further the step size and to have a constant torque output.
- For us we used the full-step mode.



- Here also you have the truth table that we used in our code to represent the situation in this image.

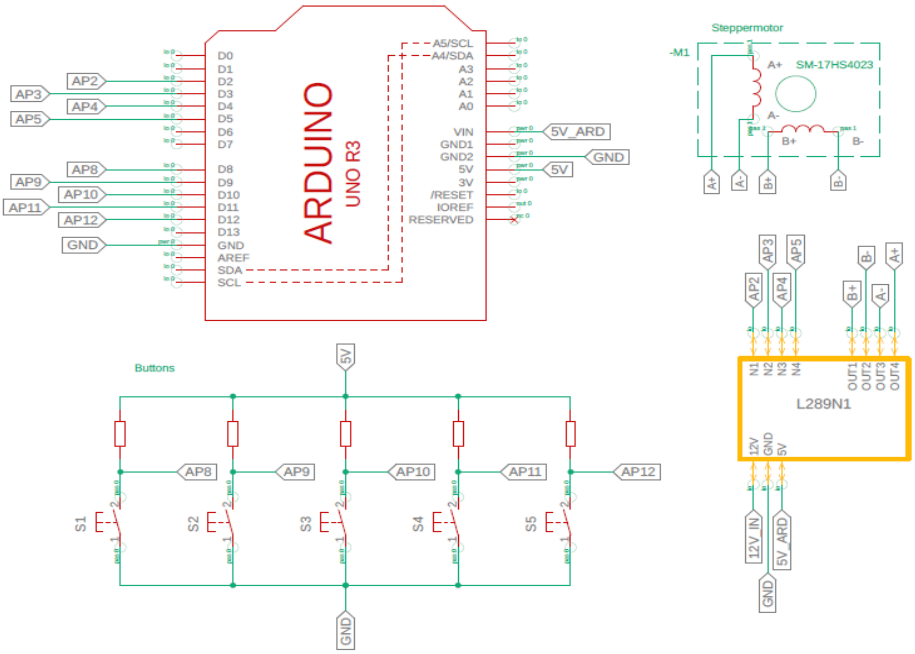
		1	2	3	4
coil	A	HIGH	LOW	LOW	HIGH
	$\overline{A}$	LOW	HIGH	HIGH	LOW
coil	B	HIGH	HIGH	LOW	LOW
	$\overline{B}$	LOW	LOW	HIGH	HIGH

- To understand more about these four different modes, we invite you to have a look on this website in the section of stepper motor driving techniques, Here.

6 Schematic

6.1 General schematic

- Picture of the plan. The plan will be printed and provided with the appendices .



6.1.1 Arduino UNO SMD connecting plan

- Explanation of the connected Pins :

Pins name	Implementation
AP 2/3/4/5	Stepper Motor coils
AP 8/9/10/11/12	Pushbuttons
SDA/SCL	OLED 128x64 (SDA -> Data & SCL -> CLK)
5V_ARD	5V Provided From the regulator in the driver L298N
5V	(Internal 5V Arduino) OLED / pushbuttons circuit
GND	Ground between driver/pushbuttons circuit/ Arduino (Ground common).

6.1.2 Buttons connecting plan

- Explanation of the connected Pins :

Pins name	Implementation
5V	From Arduino connecting directly to the resistances then from the resistances to the Arduino Pins (Pullup)
GND	Ground common. Short Arduino Pins to ground to detect changes
AP#	Inputs Arduino

6.1.3 L298N connecting plan

- Explanation of the connected Pins :

Pins name	Implementation
12V_IN	External supply
5V_ARD	Supply to the Arduino
GND	External supply (Ground common)
AP#	Arduino outputs Pins
A+	Motor coil A+
A-	Motor coil A-
B+	Motor coil B+
B-	Motor coil B-

6.1.4 OLED display connecting plan

- Explanation of the connected Pins :

Pins name	Implementation
5V	Power the display from Arduino
GND	Ground common
SCL	I2c from Arduino
SDA	I2c from Arduino

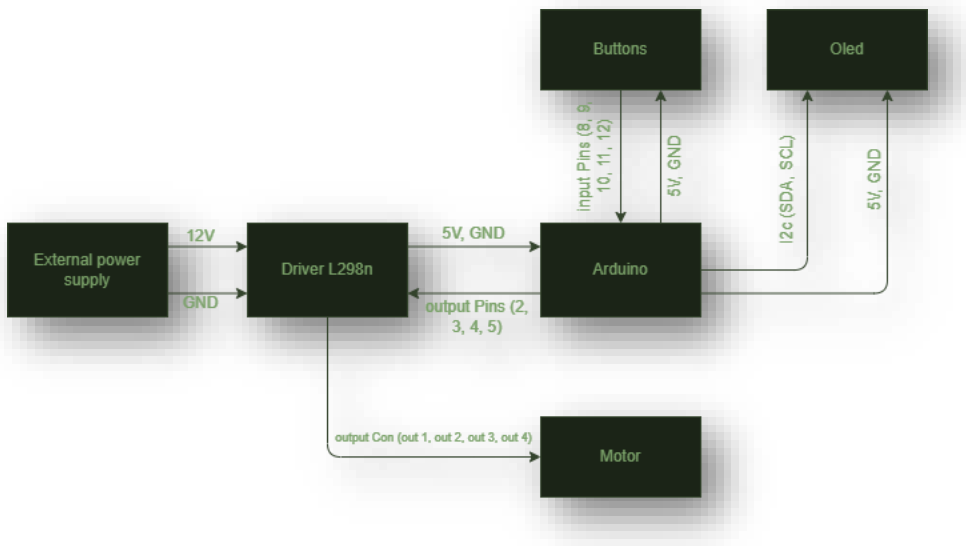
6.1.5 Stepper motor connecting plan

- Explanation of the connected Pins :

Pins name	Implementation
A-	Motor coil A- connected to output 2 (driver)
A+	Motor coil A+ connected to output 1 (driver)
B+	Motor coil B+ connected to output 3 (driver)
B-	Motor coil B- connected to output 4 (driver)

6.2 Wiring

- Here in this Diagram, you see a general view of the wiring circuit.



7 Program :

8 3D Design and Printing :

9 Product usage

10 Evaluation

11 Appendices

- Datasheet of the stepper motor.
- Datasheet of the driver.
- Schematic printed.
- Program printed.