Faculty of
Computer and Communication Sciences

Mathieu Faure

Building and operating a robotic arm

Supervisor:
Prof. Christoph Koch

Laboratory:
Data Analysis Theory and Applications Laboratory

09.06.2023

Contents

1 Introduction **2**

2 Problems with the existing project **3**

2.1 Structural Issues 5

2.1.1 Problematic weight distribution 5

2.1.2 Weak components 5

2.1.3 Cable management and carrying 5

2.2 Functional Issues 6

2.2.1 Limit switches setup 6

3 Implemented solutions **8**

3.1 Robotic arm base and mechanical parts 8

3.1.1 New base 8

3.1.2 Motor case 10

3.1.3 Tapered Roller bearings turntable 12

3.2 Electronics and wiring 12

3.3 Testing code and libraries 15

4 Observed Results and future development paths **19**

4.1 Results 19

4.2 Future development paths 20

4.2.1 A4988 drivers reorganisation 20
4.2.2 Limit switches replacement 20
4.2.3 Upper joints optimisation 21
4.2.4 Motor case upgrades 21
4.2.5 Last touch to the software part 21

5 Links and references **22**

Chapter 1

Introduction

This formal report presents the culmination of a bachelor semester project focused on the development and implementation of a robotic arm. The project at hand, which focuses on the development and improvement of a robotic arm, emerges as a natural progression following the course CS-358, "Making Intelligent Things." The objective of this research is to improve the capabilities, robustness, modularity and ergonomy of the robotic arm, building upon the groundwork laid by Léo Wolff in the previous semester. This report provides a detailed account of the methodology, findings, and outcomes achieved throughout the project, as well as future development paths that could be followed to further improve this project.

Chapter 2

Problems with the existing project

As mentioned earlier, this project was built on top of Léo Wolff's work performed during the previous semester. This consisted of a 6 degrees of freedom, stepper motors robotic arm manufactured using MDF laser cut parts, 3D printed PETG parts and custom designed machined aluminium parts, as well as a controller application, consisting of a client running on the user's laptop and a server running on a Raspberry Pi and sending serial instructions to the Arduino Mega 2560 used to control the arm. This software allows the user to control the arm either by inputting precise movements or by running preregistered movements. All the code, 3D models, hardware specification files of this project can be found in the GitHub repository that is made available by the link above.

The software side of this project is functional and executable on a laptop (both the server and the client) as it is detailed in the project's repository. However, the server side still needs to be transposed on a Raspberry Pi to allow the user to control the arm via the application.

Concerning the hardware side of the project, several issues avoiding an optimal functioning and operating of the robotic arm were noticeable and had to be dealt with. This section aims at detailing these issues and how they affected the proper functioning of the arm.

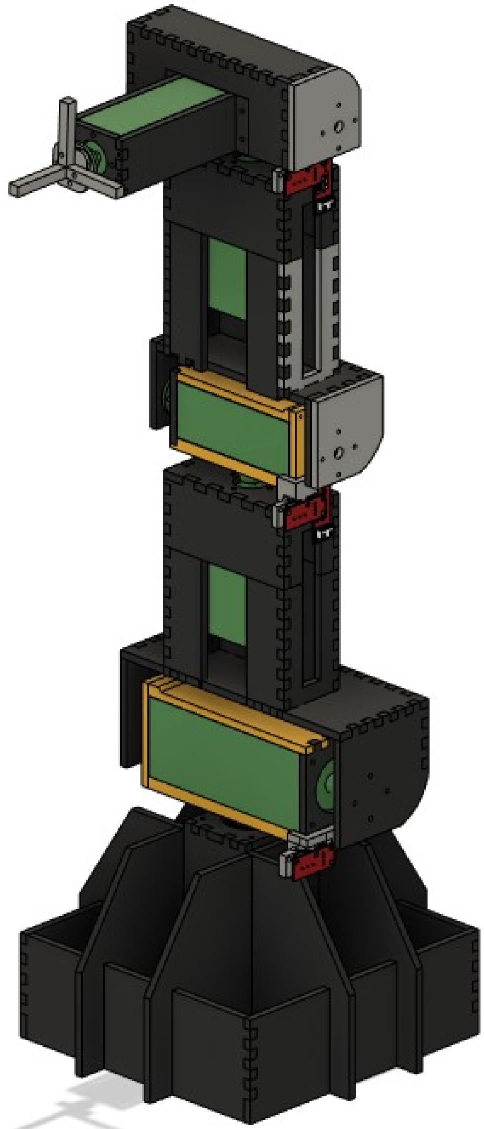

Figure 2.1: Overview of the first version of the robotic arm

2.1 Structural Issues

2.1.1 Problematic weight distribution

The first issue with the initial iteration of the robotic arm is the distribution of the weight. As the arm does not rely on any transmission system, the motors are placed on top of another and have a significant mass. This implies that there is a non-negligible weight in the wrist of the arm. When it is in a fully extended position, this weight needs to be counterbalanced by the base of the arm, which was not the case. During tests conducted at the beginning of the semester, when the arm was extended in some measure, the base of the arm was lifted off the ground and the arm threatened to fall, which of course leads to both safety and material integrity issues.

2.1.2 Weak components

The functionality of the robot arm was significantly impeded by the presence of weakly designed components. For instance, the turntable on which the entire arm rotated proved to be wobbly, unstable, and lacking the necessary rigidity. This compromised the arm's precision and overall performance. Additionally, the motor case, constructed using a combination of MDF wood and 3D printed PETG, proved to be insufficient to withstand the weight of the arm and the powerful forces exerted by the motors. As a result, the motor case frequently experienced damage, with parts being ripped off, further hampering the arm's reliability and longevity. Addressing these design weaknesses became crucial in order to enhance the arm's stability, durability, and operational capabilities.

2.1.3 Cable management and carrying

The arm, brought to life by seven stepper motors controlled by two TB6600 Microstep and five A4988 motor drivers, in addition to an Arduino Mega 2560, requires a substantial number of jumper wires to establish the necessary connections. If these elements are not securely attached to a fixed position and the associated wires are not properly managed, it can quickly become a complex and troublesome task. The movement and transportation of the arm can result in tangled wires, knots, and inadvertently unplugged connections. This scenario poses a significant challenge and can turn into a nightmare when attempting to debug and troubleshoot the system. Therefore, meticulous attention to wire organization and secure attachment of the components is essential to ensure smooth operation and ease of maintenance. The positioning of these control units outside of the arm's base, which was housed within a cardboard box, further compounded the transportation challenge. This setup not only made it difficult to transport the arm effectively but also resulted in significant time loss whenever the arm needed to be operated or tested. The need to assemble and disassemble the arm, ensuring proper connections and securing the control units, added unnecessary complexity and delayed the operational readiness of the system. Finding a more streamlined and convenient arrangement for the control units and addressing the transportation issue became imperative to optimize efficiency and reduce time wastage during operation and testing.

2.2 Functional Issues

2.2.1 Limit switches setup

The main functional issue arose during the setup of the initial position of the arm, where the implementation of endstop limit switches was crucial. These switches played a vital role in enabling the stepper motors to determine their absolute positioning accurately. However, a significant problem emerged due to the design of the limit switches. These switches used a metal presser that, unfortunately, got bent by the joint's movement. Consequently, the motors continued to rotate even after reaching the limit, leading to unintended and damaging movements. This issue not only jeopardized the arm's structural integrity but also raised concerns regarding the safety and precision of its operation. It became imperative to address this problem promptly to prevent further damage and ensure reliable and accurate positioning of the arm. As this setup is necessary, an alternative to these basic limit switches needs to be found.

The means that have been brought to solve the majority of the exposed issues will be detailed in the next section.

Chapter 3

Implemented solutions

In order to address the aforementioned challenges and improve the overall performance of the robot arm, a comprehensive redesign of its base was undertaken. Using Autodesk Fusion 360, several mechanical parts were redesigned, taking into account factors such as stability, rigidity, and durability. The new design aimed to provide a solid foundation for the arm, ensuring smoother and more precise movements.

In the following sections, this report will delve into further detail regarding the mechanical components that were designed, the wiring scheme employed for the electronic components, and the specific code and libraries used to test and validate the upgrades made to the robot arm.

3.1 Robotic arm base and mechanical parts

3.1.1 New base

In order to address the various challenges faced in the development of the robotic arm, several improvements were implemented, primarily centered around the design of a larger and more functional base. This new base not only serves as a platform for the arm but also offers solutions to critical issues related to weight distribution, cable management, portability, and adaptability.

One key problem that was encountered in the earlier iterations of the robotic arm was the presence of excessive weight at the extremity of the arm. This imbalance caused stability issues and hindered the arm's overall performance. To mitigate this problem, a larger base was designed, which provides a solid foundation and improved weight distribution. By expanding the base's size, the center of gravity of the arm is shifted towards the lower region, resulting in a better stability.

Furthermore, the increased size of the base allowed for the inclusion of additional features to address cable management and portability concerns. Within the newly designed base, ample space was allocated to accommodate the electronic components of the robotic arm. The centralized placement of electronic components also facilitates ease of access for maintenance and troubleshooting.

Moreover, a dedicated space was incorporated into the base design to securely hold the power supply cable, which is held in place by a 3D printed piece.

To enhance adaptability and modularity, the new base design includes strategically placed holes. These holes allow for the attachment of additional modules, should the need arise in the future. This flexibility enables the integration of supplementary components or upgrades, thereby ensuring scalability for potential future developments or modifications.

In summary, the implementation of a larger base in the robotic arm project has successfully addressed multiple challenges. The enhanced weight distribution has improved stability, while the provision of space for electronic components has resolved cable management and portability issues. Additionally, the inclusion of a dedicated area for the power supply and strategically placed holes for attaching modules enhances the adaptability and expandability of the arm. These improvements collectively contribute to a more efficient and robust robotic arm system.

Figure 3.1: Overview of the new base

3.1.2 Motor case

The new motor case I designed for the first X-axis joint addresses the previous issues and significantly improves its functionality. The case has been designed to perfectly accommodate the HT 23HS5628 motor and PLE23-G50-D8 gearbox, ensuring a secure and precise fit. Unlike the previous design, this new case is constructed in two pieces that fit together seamlessly, providing enhanced robustness and eliminating the risk of it being ripped off. By tightly locking the motor in place, any potential wobbling or instability is avoided, resulting in increased precision during operation. The improved motor case serves as a crucial component in ensuring the overall stability and accuracy of the lowest X-axis joint, contributing to the enhanced performance of the entire system. In the bottom part, six holes have been incorporated to fit M6 80mm screws attaching it to the turntable and the other motor below. A picture of the newly designed motor case can be found below.

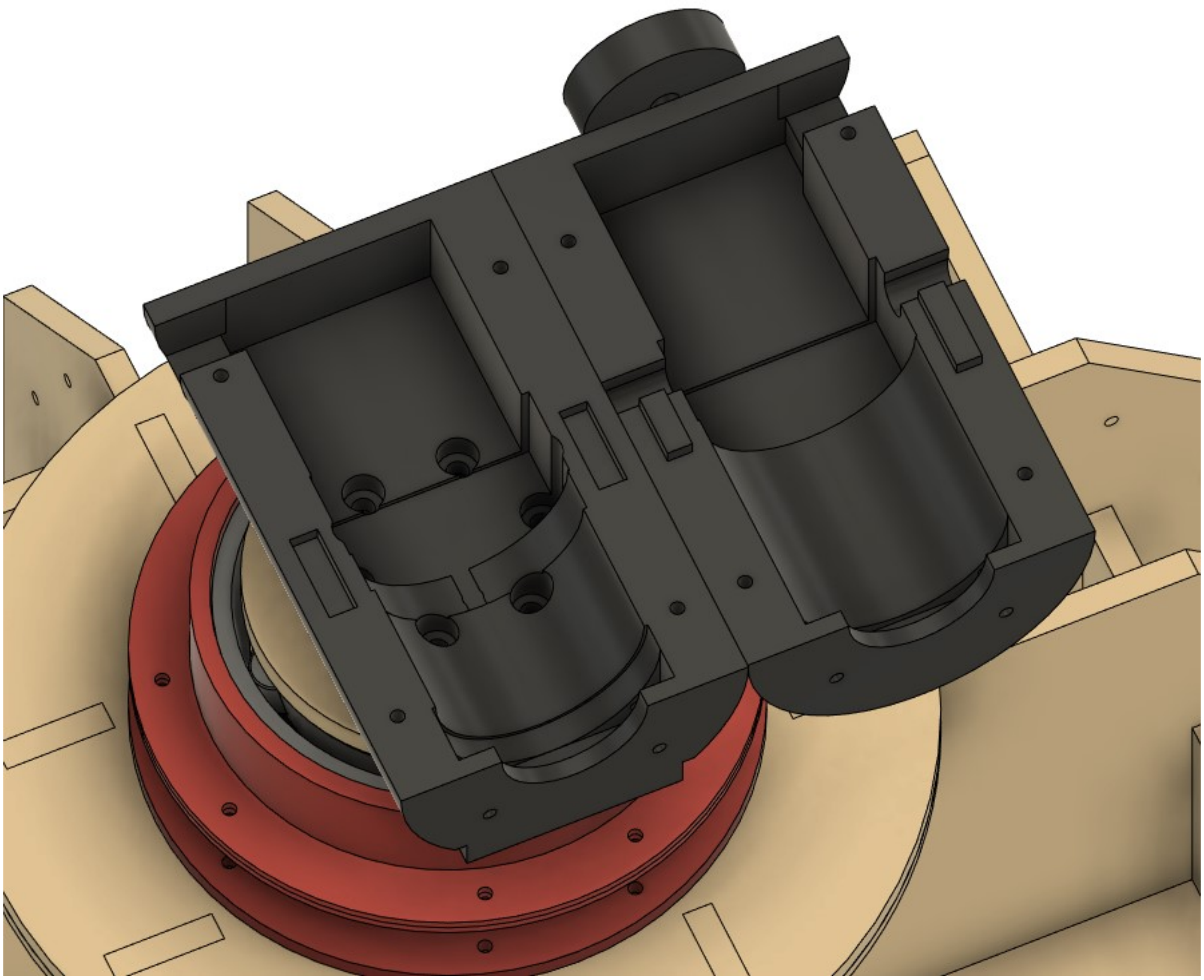

Figure 3.2: Overview of the new motor case

3.1.3 Tapered Roller bearings turntable

To replace the problematic turntable on which the original iteration of the robotic arm relied, I chose to design a turntable using two tapered roller bearings facing each other and rotating together, as suggested by the DLLEL prototyping coach Simon Lütolf. The reference of the bearings that were used for this purpose can be found in Chapter 5. This configuration, heavily used in the automobile industry, offers more rigidity, resistance to constraints, and smoothness in the rotation. A cross-sectional image of this part can be found below. Both bearings' non rotative parts are held by the red 3D printed components. The rotative parts are pressed together by the two laser cut slices. 80mm long M6 screws go first through the holes that are located at the bottom of the case holding the motor situated on top of the turntable, then through the top slice, through the PETG part at the center of the bearings keeping them in the right axis, and then through the bottom wood slice. The motor's coupling flange positioned vertically and hidden in the arm's base is attached to this bottom slice and allows the whole top part of the arm to rotate on this turntable. For a deeper structural insight, you can refer to the 3D files that are provided in the GitHub repository linked at the bottom of the report.

3.2 Electronics and wiring

The two following diagrams describe the wiring between the stepper motors and the stepper motor drivers. The first was done by myself using Wokwi and describes the wiring of the five smaller motors 17HS4401S and the A4988 drivers, while the second one was found on the internet and describes the wiring of the Microstep TB6600 driver and the two bigger HT 23HS5628 motors.

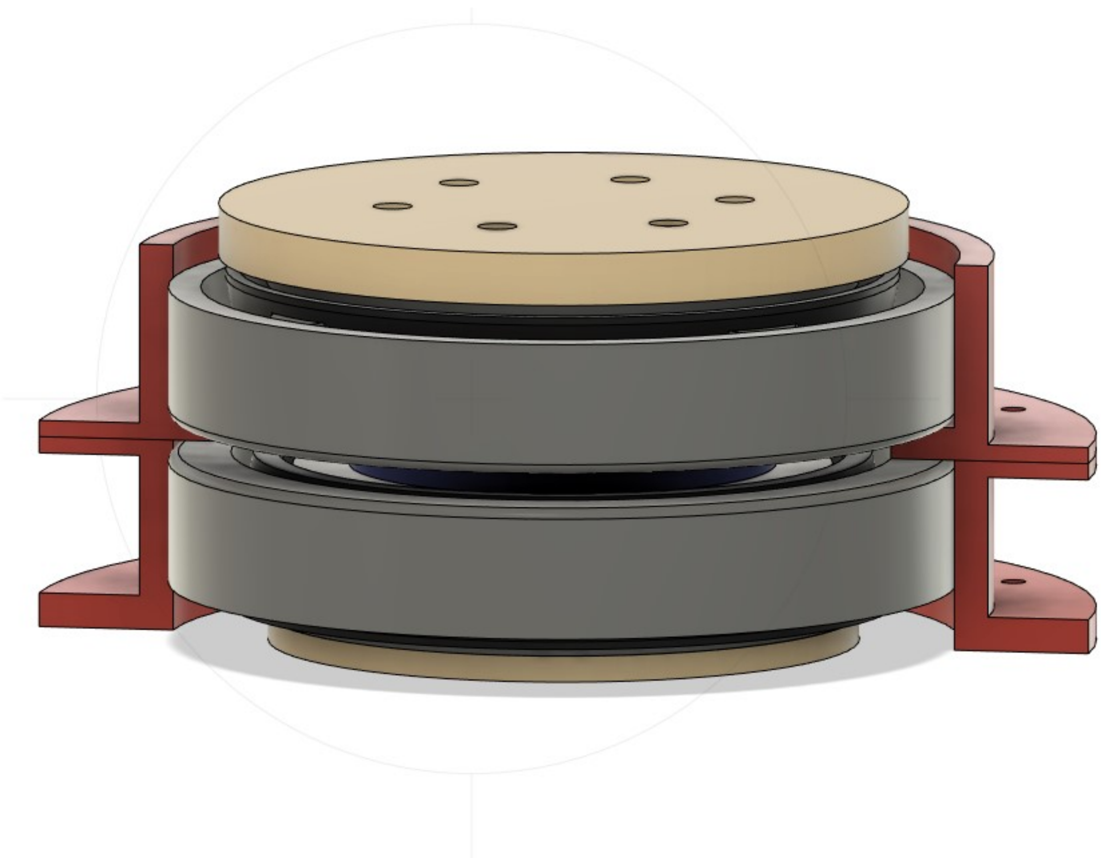

Figure 3.3: Cross-sectional image of the turntable system

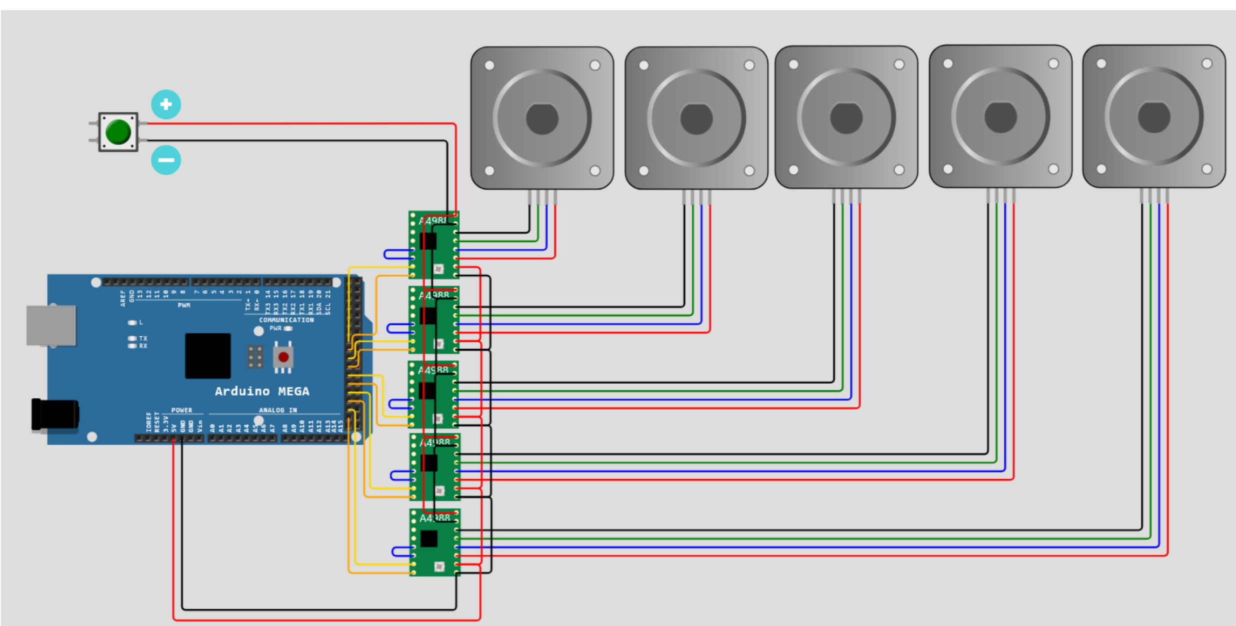

Figure 3.4: Wiring for the five 17HS4401S stepper motors

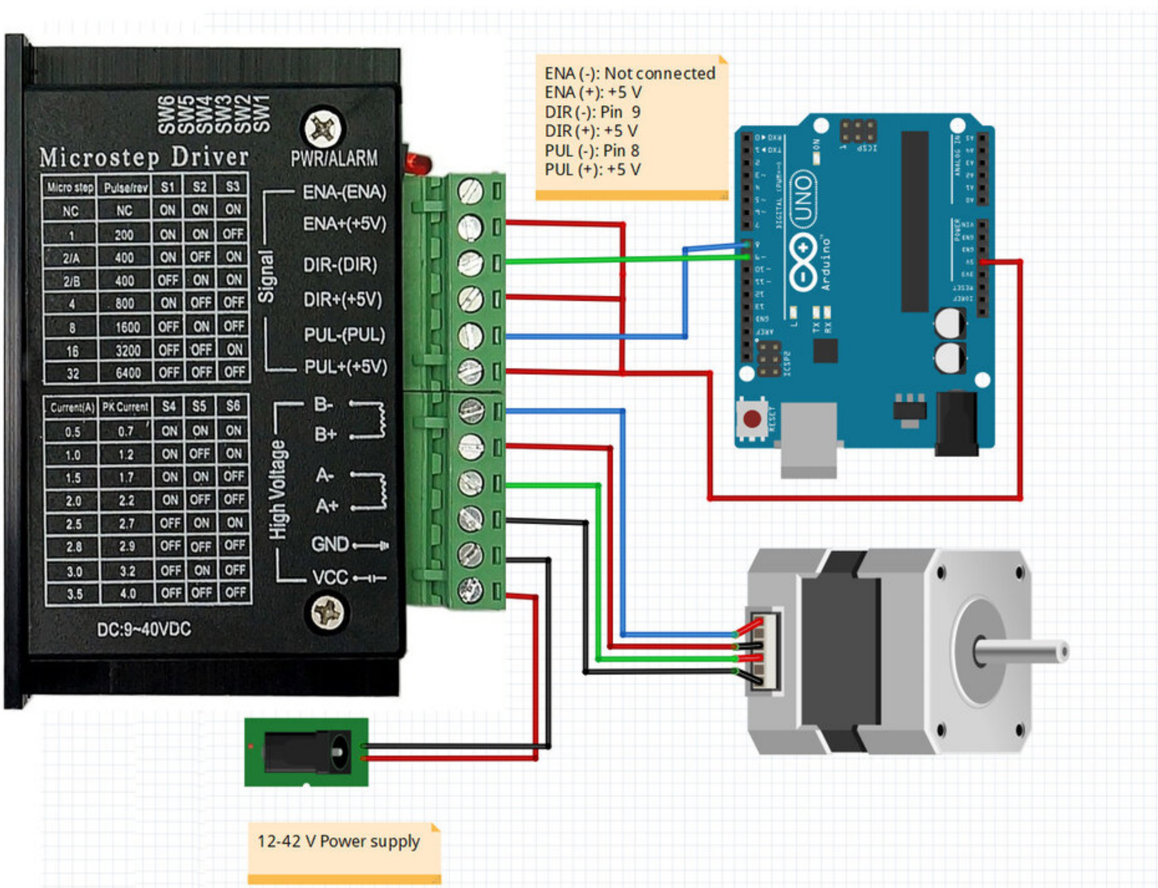

Figure 3.5: Wiring for the two HT 23HS5628 motors

3.3 Testing code and libraries

To evaluate the effectiveness of the upgrades made to the arm and verify its improved functionality, the Arduino IDE was employed alongside the AccelStepper library.

The AccelStepper library offers several advantages that make it a popular choice for controlling stepper motors in projects like the robot arm. Some of the key advantages of the AccelStepper library are:

1. **Advanced Motion Control**: The library provides advanced motion control features, including acceleration, deceleration, and precise positioning of stepper motors. It allows for smooth and controlled movements, ensuring accurate positioning and reducing the risk of overshooting or missed steps.

2. **Flexible Speed Profiles**: With the 'AccelStepper' library, it is easy to define custom speed profiles for stepper motors. This flexibility allows for dynamic adjustments in speed, enabling gradual acceleration or deceleration, which is crucial for achieving smooth and controlled movements in applications such as the robot arm.

3. **Multiple Motor Support**: The library supports simultaneous control of multiple stepper motors. This feature is especially beneficial for complex projects that require coordinated movements of multiple motors like this one, as it simplifies the programming and synchronization of the motor actions.

4. **Step and Direction Control**: The 'AccelStepper' library provides convenient step and direction control interfaces, making it straightforward to integrate with various microcontrollers and driver modules. This compatibility allows for seamless integration into existing projects and eases the implementation of control signals.

5. **Comprehensive Documentation**: The 'AccelStepper' library is well-documented, providing clear explanations, usage examples, and troubleshooting guidance. This

extensive documentation assists developers in quickly understanding and implementing the library in their projects, reducing development time and enhancing productivity.

6. **Active Community and Support**: The ‘AccelStepper‘ library has a thriving community of users and developers, making it easier to seek help, share experiences, and find code examples or solutions to common challenges. The active community support enhances the reliability and usability of the library.

Overall, the AccelStepper library simplifies the control and operation of stepper motors, offering advanced motion control features, flexibility, and reliable performance. These advantages make it an excellent choice for testing and implementing upgrades in projects like the robot arm, where precise motor control and smooth movements are essential.

Here is a simple code example using the AccelStepper library to test the two lower degrees of freedom of the arm:

```
#include <AccelStepper.h>

#define m1Pulse  24
#define m1Dir  26
#define m1Reduc 10
#define m1MicStep 4

#define m2Pulse  28
#define m2Dir  30
#define m2Reduc 50
#define m2MicStep 4

#define degPerStep 1.8

#define MotorInterface 1
```

```
AccelStepper stepper1(MotorInterface, m1Pulse, m1Dir);

AccelStepper stepper2(MotorInterface, m2Pulse, m2Dir);



void setup() {
  // put your setup code here, to run once:
    stepper1.setMaxSpeed(1000);
    stepper2.setMaxSpeed(1000);
    stepper1.setAcceleration(500);
    stepper2.setAcceleration(500);

}

void loop() {
  // put your main code here, to run repeatedly:
  moveTo(1, 105);
  delay(2000);
  moveTo(2, 70);
  delay(2000);
  moveTo(1, 0);
  delay(2000);
  moveTo(2, 0);

}


```

```
void moveTo(int mIndex, float deg) {
  if(mIndex == 1){
    stepper1.moveTo(deg/degPerStep * m1Reduc * m1MicStep);
    stepper1.runToPosition();
  }
  if(mIndex == 2){
    stepper2.moveTo(deg/degPerStep * m2Reduc * m2MicStep);
    stepper2.runToPosition();
  }
}
```

Listing 3.1: Arduino code example

Chapter 4

Observed Results and future development paths

4.1 Results

The implemented improvements have yielded significant positive results, effectively addressing the majority of the challenges outlined earlier. The newly designed base has played a key role in resolving these issues. Firstly, the redesigned motor case now perfectly fits the motor, providing a robust and custom-designed enclosure that meets the arm's requirements. This improved motor case ensures enhanced protection and stability, minimizing the risk of damage caused by the weight and force exerted by the rest of the arm.

Moreover, the integration of a new double tapered roller bearing turntable has brought substantial improvements. The new turntable exhibits better robustness, offering a smooth and stable rotation. It is highly resistant to torque, eliminating the wobbling and instability that were previously encountered. These enhancements significantly enhance the arm's overall performance and precision.

The redesigned base not only facilitates the attachment of electronic components but also addresses cable management concerns. It reduces the risk of having unplugged connections in transport. Additionally, the base incorporates modularity, providing the flexibility to accommodate outriggers for improved stability or the integration of additional components as required.

The arm's compactness has been significantly improved, rendering it easily pluggable and more convenient to carry. These improvements contribute to the arm's overall portability and user-friendliness.

In summary, the implemented improvements, including the redesigned base, motor case, turntable, and cable management system, have effectively mitigated the earlier challenges. The arm now boasts enhanced robustness, stability, and torque resistance. Its compact and easily pluggable design, coupled with improved cable management, facilitates transportation and usability. The modularity of the base allows for further enhancements and customization, ensuring the arm's adaptability to future requirements.

4.2 Future development paths

4.2.1 A4988 drivers reorganisation

Currently, the five A4988 stepper drivers are mounted on copper breadboards. This setup could be improved. When testing the arm after finalising the mechanical improvements, I noticed some inconsistency in the functioning of some of the motors controlled by these drivers, which could be explained by not clean enough soldering on these copper boards. Designing a custom fitting part to hold these drivers and cleaning the wired connections could help in getting rid of these inconsistencies.

4.2.2 Limit switches replacement

Relying on limit switches for the arm's initialisation is a problem as they can be inconsistent and their inconsistency can cause heavy damages. However, their role is essential, so an alternative has to be found. Several options are available : an alternative sensor such as an incremental encoder CUI AMT103-V, or having a support on which the arm relies when it is not used, and use this precise position as a 0 position for the stepper motors. Such a platform could be easily added thanks to the outriggers attaches that are available on the arm's base.

4.2.3 Upper joints optimisation

The actuation of the upper joints could be optimised using a system similar to the one used for the shoulder joint, but with smaller bearings and/or motor cases.

4.2.4 Motor case upgrades

The newly designed motor case only has two small holes on the sides allowing it to cool down, which has not appeared to be a problem in the tests that have been run until now. However, it would be advisable to conduct tests to monitor the motor's temperature after significant usage and evaluate its performance under varying conditions. If the temperature measurements indicate a need for additional cooling, the motor case should be slightly modified to fit these requirements.

4.2.5 Last touch to the software part

This task should be realised once the mechanical side is fully optimised, as the software relies on reverse kinematics and on the precise dimensions of the several parts and joints composing the arm. It would consist in having the controller's server running on a raspberry pi and communicating both with the client on the user's laptop and the Arduino Mega via serial connection.

Chapter 5

Links and references

- All 3D, Arduino code and other files available in this repository
- Original project repository
- Tapered Roller Bearing