People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Echahid Hamma Lakhdar University of El-Oued
FACULTY OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING

Master's Thesis

Presented for obtaining the degree of

ACADEMIC MASTER

Field: Sciences and Technologies

Program: Electromechanics

Specialty: Electromechanics

Topic

Design and Implementation of a Robotic Arm Controlled By an Arduino

Before the jury composed of:

Dr	President
Dr	Examiner
Dr Hamza Mesai Ahmed	Supervisor

Presented by:

- Bekakra Mohmmed elBachir
- Gaid Oussama
- Radouni Abderraouf

2024/2023

Acknowledgments

First and foremost, we would like to express our gratitude to Allah, the Almighty, for granting us the courage, patience, and determination to achieve this goal.

We would like to extend our sincere thanks to our supervisor, Dr. HAMZA MESAI, for agreeing to oversee this thesis and for his patience during the mentoring period.

Our heartfelt thanks also go to the members of the jury for honoring us with their evaluation of this modest work.

Additionally, we express our gratitude to our friend Ayman Ammari for allowing us to work in his private workshop and for providing us with advice and guidance.

Finally, thanks to all the individuals who supported and assisted us, directly or indirectly, in the completion of this work.

Dedications

I dedicate this work primarily to my father and mother, who have always been my constant supporters, as well as to all my siblings, relatives, and friends.

Bachir

I dedicate this success to myself first, and then to those who accompanied me on this academic journey. I offer the fruit of this humble effort to my mother and father, who showed great patience for my success. To the cherished soul of my grandmother, who always wished to witness my graduation. And to my entire dear family.

Oussama

I dedicate this work to my mother, my father, my grandmother, and all my brothers and sisters.

Abdel Raouf

Index

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations

3D - Three dimensions

AFRI - French Association for Industrial Robotics

AI - Artificial Intelligence

CPU - Central Processing Unit

DLR - German Aerospace Center

DOF - Degrees of Freedom

DSS - Design Study SolidWorks

EEPROM - Electrically Erasable Programmable Read-Only Memory

GUI - Graphical User Interface

I/O - Input/Output

IDE - Integrated Development Environment

PLC - Programmable Logic Controller

PPP - Prismatic Prismatic Prismatic (Cartesian structure - three prismatic joints)

PRP - Prismatic Rotational Prismatic (Variant of cylindrical structure)

PWM - Pulse Width Modulation

RAM - Random Access Memory

ROM - Read-Only Memory

RRP - Rotational Rotational Prismatic (Spherical/Polar structure)

RRR - Rotational Rotational Rotational (Anthropomorphic structure)

RPP - Rotational Prismatic Prismatic (Cylindrical structure - one rotational and two prismatic joints)

SCARA - Selective Compliance Articulated Robot for Assembly

SRAM - Static Random Access Memory

TQFP - Thin Quad Flat Pack (Type of microcontroller package used for surface mounting)

USB - Universal Serial Bus

GENERAL INTRODUCTION

In the world of robotics, from ancient automatons to modern myths, the word "robot" resonates through time. Distinguishing fact from fiction is a challenge as we aim to make fantasy a reality. Arduino-controlled robotic arms represent a new frontier, blending precision with accessibility, sparking innovation. My passion for this field stems from a love of discovery, exploring the fusion of hardware, software, and imagination. Like the industrial robotics "boom" of the past, Arduino integration promises to reshape automation, inspiring creativity and shaping our technological future.

A passion for Arduino, robotics, and programming drives a joy in discovery and invention. Also, the use of computer tools to bring ideas to life and the constant motivation to learn more and develop new ideas through real-world applications are the reasons for choosing this project.to learn more and develop new ideas through real-world applications.

How to Design and Build a Functional Robotic Manipulator Arm Using an Arduino Development Board for Control?

We structured our study into three chapters to tackle this issue comprehensively. The first chapter offered a comprehensive overview of robots , which we named "State of the Art on the Robotic System".

Programming and simulation were covered in detail in the second chapter, which also introduced and explained the software used in each case. Additionally, we create flowchart to help us write our code and produce a simulation for the robot's control circuit.

The third chapter was hands-on, starting with modeling the robot's dimensions, detailing the components utilized, and concluding with the assembly process.

Finally, the thesis concludes with a conclusion that summarizes the knowledge acquired during this project, as well as prospects for enhancing our manipulator arm and advancing our work in the future.

During the completion of our thesis, we encountered several challenges. The most significant of them were financial difficulties due to the high cost of electronic components, which limited our creativity and forced us to work with basic resources only. Also faced numerous technical issues during the robot assembly, such as motor failures and programming problems. Additionally, we dealt with some personal and familial issues that affected our focus and progress.

CHAPTER I:

STATE OF THE ART ON THE ROBOTIC SYSTEM

I.1.Introduction

The world of robotics is vast and dynamic in terms of the technologies used and the great benefits that manifest in their efficiency and uses. In this chapter, we will address the field of robots in general, mentioning their types, characteristics, and areas of use, as well as their advantages in facilitating human life.

I.2.Definitions

I.2.1. Robotics

Robotics is a multidisciplinary field that encompasses the design, construction, programming, and use of robots. A robot is an automated system capable of performing physical or logical tasks, often in response to environmental stimuli. Robotics integrates concepts from computer science, electronics, mechanics, kinematics, sensory perception, and artificial intelligence to create machines capable of performing complex actions autonomously or semi-autonomously. The goal of robotics is to improve the efficiency, accuracy, and versatility of automated systems in various fields such as industry, medicine, space exploration, and many others [1].

I.2.2. Robot

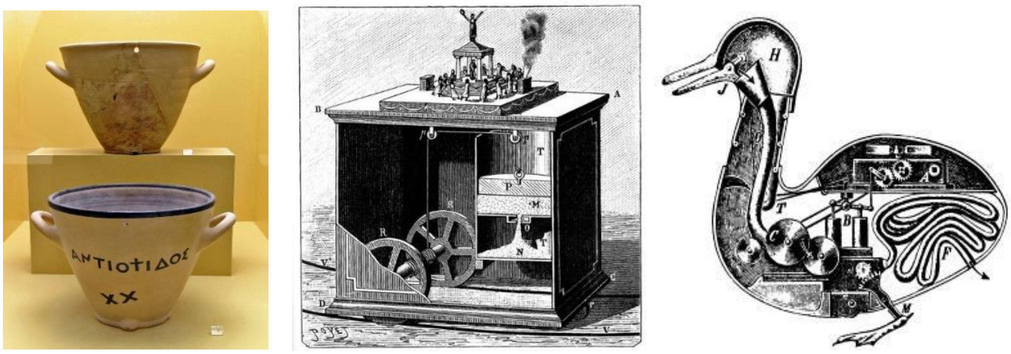
A robot is an automatic device with multiple degrees of freedom capable of manipulating objects or performing operations according to a fixed, modifiable, or adaptable program [2].

I.3.The history of robotics

Throughout history, humans have sought to replace themselves for specific tasks that could be dangerous, tedious, or time-consuming. Therefore, they have attempted to design machines capable of exhibiting similar or lifelike abilities. The concept of the robot has been established through numerous historical creative achievements, such as the water clock introduced by the Babylonians (1400 BC), the automaton theater of Heron of Alexandria (around 100 AD), and the automata of the Arab Al-Jazari (1200 AD) [3].

One of these highly evolved specimens was presented by Jacques de Vaucanson in 1738: it depicted a man playing a wind instrument. Jacques de Vaucanson also created an automaton representing a duck eating and digesting its food after ingestion.

However, the emergence of physical robots had to wait for the arrival of advanced and more suitable technologies during the 20th century.



(a) (b) (c)

Fig I.1. From left to right: (a) Water clock of Clepsydra, (b) Mobile theater of Heron of Alexandria, (c) Digesting duck by Jacques de Vaucanson. [4]

- 1920: The origin of the word "robot" comes from the Czech language, where its ancestor "robota" means forced labor. It was introduced in 1920 by the Czech writer Karel Čapek in the play "Rossum's Universal Robots" [5].

- In 1959: The robot called Unimate, fig. I.2, is considered the first industrial robot. Unimate was a manipulator with hydraulic pivot drives, its control was achieved using a magnetic drum memory in joint coordinates [6].

Fig I.2. Unimate, the first industrial robot. [7]

- In 1961: The industrial robot Unimate began to be used in automobile production at General Motors [8].

- 1973: KUKA, a German company, created the first industrial robot with 6 electromechanically controlled axes: the Famulus [5].

- 1974: ASEA delivered the first all-electric industrial robots controlled by a microprocessor [6].

- 1978: Unimation developed the Programmable Universal Machine for Assembly (PUMA) robot, depicted in the figure below; it was one of the most widely used robots.

Fig I.3. PUMA Robot. [9]

- In 1983: The number of robots worldwide reached 66,000 [6].
- 1998: ABB, Sweden, developed the FlexPicker, the world's fastest order picking robot based on the delta robot developed by Reymond Clavel, Swiss Federal Institute of Technology in Lausanne (EPFL) [5].
- 2003: The first exploration mission to the planet Mars involving two mobile robots, Spirit and Opportunity.

Fig I.4. OPPORTUNITY Robot. [5]

- En 2004: Motoman introduced a robot controller capable of synchronizing the movement of four robots with up to 38 axes [6].

- In 2006: With KUKA, the German Aerospace Center (DLR) developed the third generation of its Lightweight Robot (LBR). This is an anthropomorphic arm with 7 joints, each joint equipped with a drive sensor and an output angle sensor, as well as a torque sensor. The LBR is now commercially available through KUKA [6].

- 2011: Robonaut (R2B), the first humanoid robot sent into space, built and designed by NASA at the Johnson Space Center (JSC) in Houston, Texas, in collaboration with General Motors (GM) and Oceaneering.

Fig I.5. Robonaut, the first humanoid robot. [5]

- In 2015: Robot sales increased to 253,748 units [10].

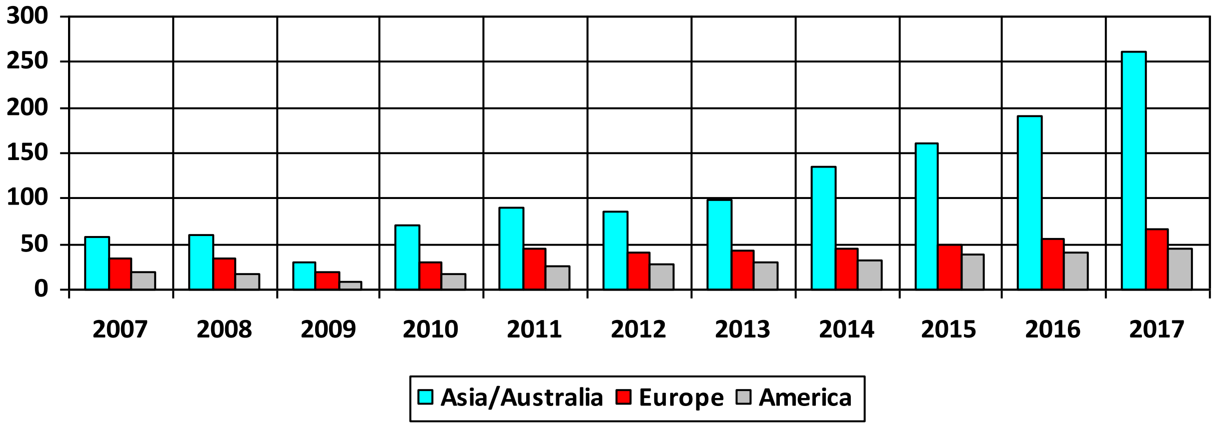


Fig I.6. Graph representing the estimated annual global shipments of industrial robots by regions. [5]

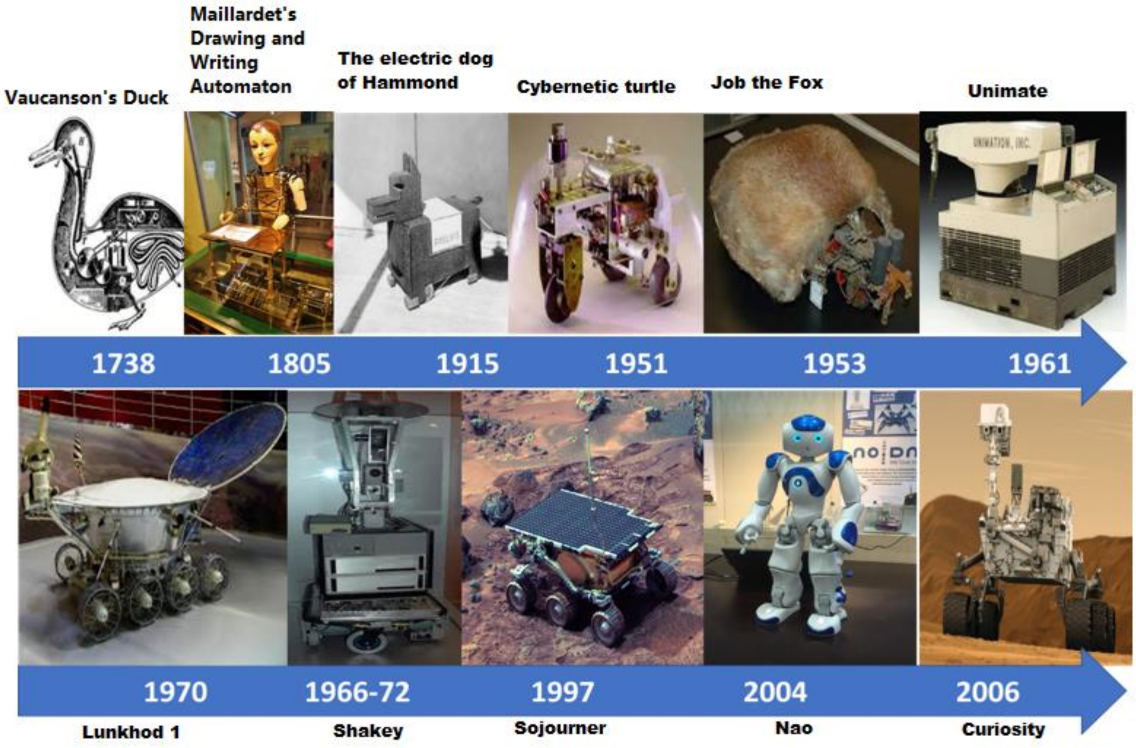


Fig.I.7: The evolution of robotics.

I.4. Types of robots:

The robots are divided into two types in terms of structure: there are mobile robots and manipulator robots. We mention some of them here.

I.4.1. Mobile Robots :

In general, mobile robots refer to all robots with a mobile base. However, conventionally, this term often specifically denotes wheeled mobile robots. Other mobile robots are typically designated by their mode of locomotion, whether they are walkers, underwater, or aerial robots [11].

Fig I.8. Mobile robot.

I.4.2.Manipulator Robots:

Manipulator robots are robots integrated into a dedicated space for transferring semi-finished products between production lines. The robot is typically suspended due to space constraints and installation requirements [12].

Fig 1.9. Manipulator Arm.

I.5.Components of a robotic system:

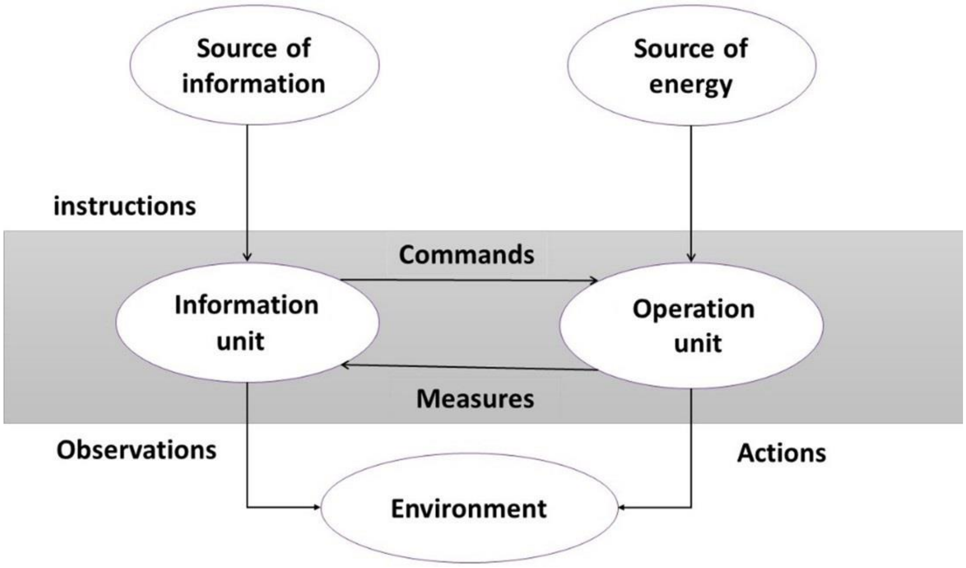


Fig I.10. The functional structure of a robot.

I.5.1. Information Unit:

It receives instructions describing the task to be accomplished, measurements related to the internal state of the mechanical structure that constitutes the manipulator arm, and observations concerning its environment. It consequently develops the commands for its various joints to execute these tasks. Current systems operate in permanent interaction according to the information-decision-action cycle [13].

I.5.2. Operation unit:

It executes the commanded actions by drawing the necessary power from the energy source. This part, which constitutes the physical robot, integrates the mechanical structure (segments, joints, architecture...), energy modules (amplifiers, converters, servo valves...), energy converters (motors, cylinders...), mechanical transmission kinematic chains (reducers, ball screws, toothed belts...), proprioceptive sensors placed on each axis to continuously measure their position and speed, and finally the effector, or end effector, which interacts with the environment [13].

I.5.3. Articulated mechanical structure:

A manipulator robot generally consists of two distinct sub-assemblies: an end effector, which is the device intended for manipulating objects, and an articulated mechanical structure (AMS), consisting of a set of solids connected to each other, typically in sequence, where each solid is movable relative to the previous one. This mobility is expressed in terms of degrees of freedom (DOF), which is defined as the number of independent movements possible for a solid C1 relative to the solid directly connected to it, C2.

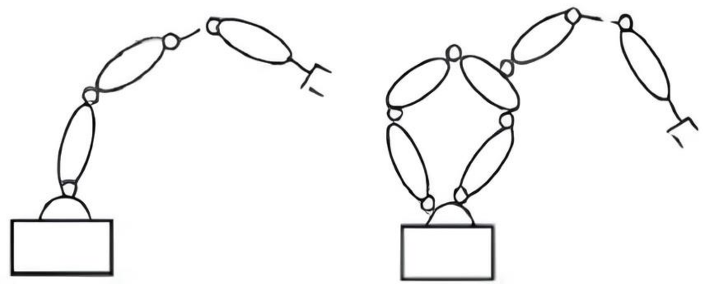
An articulated mechanical structure can be represented by an architecture composed of several chains of rigid bodies assembled by connections called joints. The chains can be either open or in series, in which all bodies have at most two connections, or they can be branched, where at least one of the bodies has more than two connections. The chains can also be closed, in which the end effector is connected to the base of the mechanism through multiple chains [14].

I.5.3.1. Mechanical structure articulated with a simple kinematic chain.I.6.

It's a kinematic chain where each member has a connection degree (number of mechanical linkages) less than or equal to two. A serial robot is formed by a simple kinematic chain where the base and the end effector have a connection degree of one (meaning they are connected to only one body), and the other elements have a connection degree of two.

I.5.3.2. Mechanical structure articulated with a closed kinematic chain.

It's a kinematic chain where there is mechanical feedback from one or more segments to another within the chain.



a) open and series structure b) closed structure

Fig I.11.SMA with a closed and series kinematic chain. [15]

I.6.Robot Architecture:

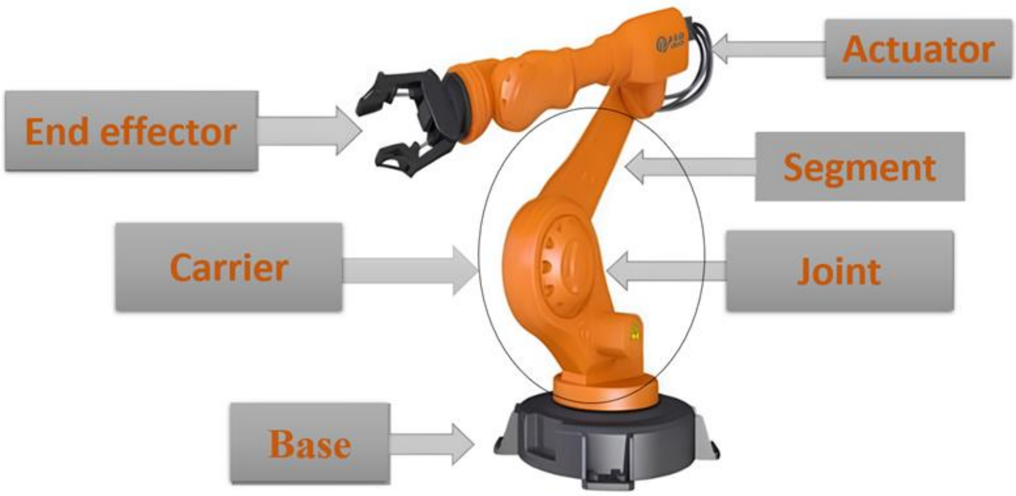


Fig I.12. The architecture of a robotic manipulator. [16].

I.6.1. Base

The base of the manipulator is fixed at the workplace. This is the case for the vast majority of industrial robots [16].

I.6.2. Carrier

The carrier represents the core of the articulated mechanical system. Its role is to position the end effector in a specific situation dictated by the task (the situation of a body can be defined as the position and orientation of a reference frame attached to this body relative to a reference frame). It is composed of (Segment_ Joint) [16].

I.6.3. Actuator

The actuators commonly use permanent magnet DC motors with armature control. Additionally, there is an increasing use of electronically commutated (brushless) motors or, for small robots, stepper motors. For robots required to handle very heavy loads (for example, an excavator), the actuators are most often hydraulic, acting in translation (hydraulic cylinder) or rotation (hydraulic motor) [16].

I.6.4. End effector:

This encompasses all devices intended to manipulate objects (clamping devices, magnetic devices, vacuum devices, etc.), or to transform them (tools, welding torches, paint guns, etc.). In other words, it is an interface that allows the robot to interact with its environment [16].

I.7. Classification of robots

The classification of robotic systems is challenging because there are numerous criteria for their descriptions. However, we will classify them according to two structures, which are as follows:

I.7.1. Functional classification

The number of classes and the distinctions between them vary from country to country (6 classes in Japan, 4 in France). The A.F.R.I. distinguishes 4 classes illustrated below:

I.7.1.1. Manual control manipulator

Figure 1.13 represents manual control manipulators.

Fig I.13. Manual control manipulators.

I.7.1.2. Automatic manipulator with pre-set cycles

The adjustment is mechanically done by cams, stops; control can be done by a programmable logic controller (PLC); we can distinguish between fixed-cycle manipulators and programmable-cycle manipulators.

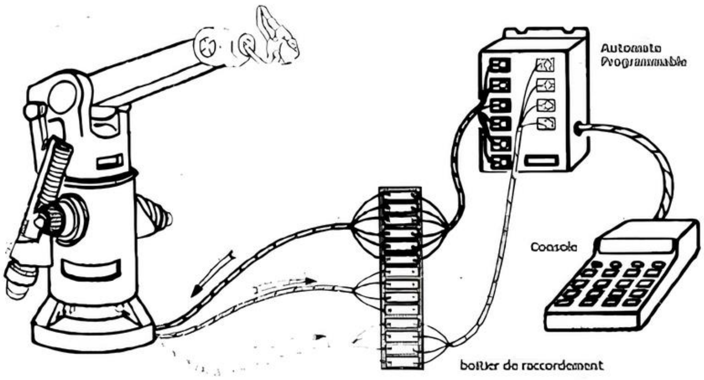


Fig I.14. Pre-set cycle manipulator. [17]

1.7.1.3. Programmable robots

They repeat movements that have been taught or programmed to them, without information about the environment or the task being performed. One can also distinguish between "play-back" robots that reproduce the learned task and numerical control robots that can be programmed offline [18].

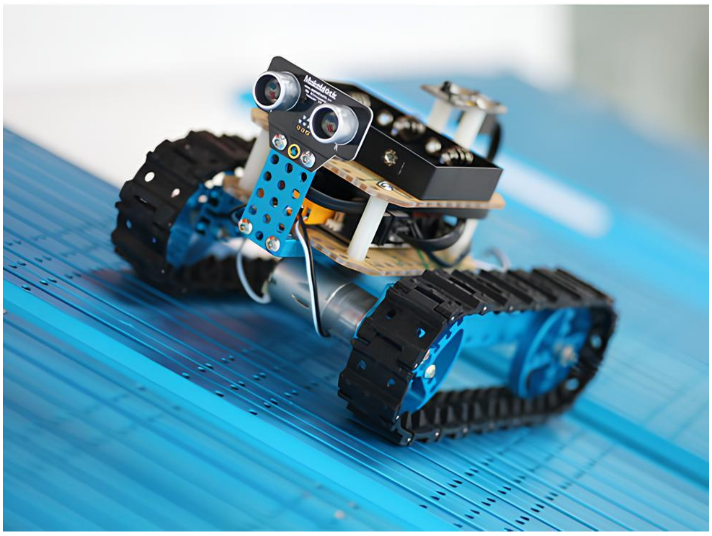


Fig I.15. Programmable robot.

1.7.1.4. Intelligent robots

Currently, second-generation robots are capable of acquiring and utilizing certain information about their environment (vision systems, proximity sensors, force sensors, etc.). Third-generation robots are being studied, which are capable of understanding a spoken language close to natural language and of autonomously navigating in a complex environment, thanks to the use of artificial intelligence.

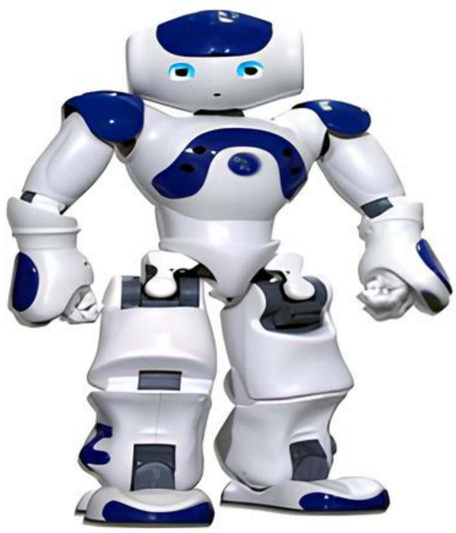


Fig I.16. Intelligent robot. [19]

I.7.2. Geometric classification

There are different architectures of the manipulator: The Cartesian structure which consists of three consecutive prismatic joints (PPP), the cylindrical structure (RPP or PRP) this means one rotary joint and two prismatic joints, the spherical or polar structure contains two rotational joints and one prismatic joint (RRP), the so-called SCARA structure (RRP), and finally the anthropomorphic structure which consists of three consecutive rotational joints (RRR).

1.7.2.1. Cartesian Structure (PPP)

The characteristics of a Cartesian structure (PPP) are as follows: three axes, taken two by two in series with three degrees of freedom, very good precision, and slow speed. The working volume is a parallelepiped whose dimensions are the translations permitted by the three prismatic joints [20].

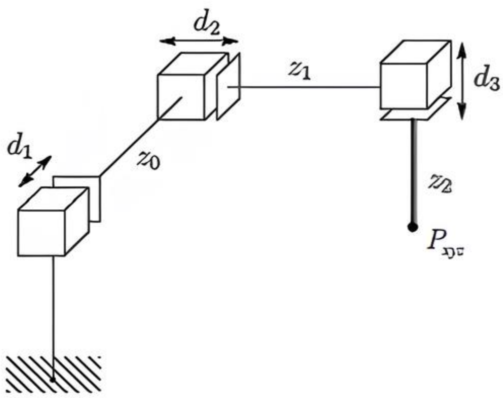


Fig I.17. PPP Structure. [21]

1.7.2.2. Cylindrical Structure (RPP) or (PRP)

The characteristics of a cylindrical structure (RPP or PRP) are as follows: three non-perpendicular axes in series with three degrees of freedom and high speed. The working volume is a solid or hollow cylinder, in other words, a torus with a rectangular section, where the height L is the translation permitted by a prismatic joint [21].

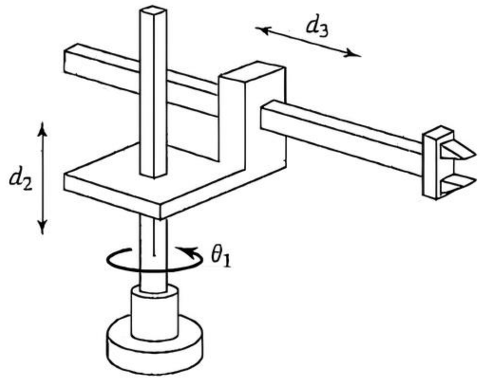


Fig I.18. RPP Structure. [22]

1.7.2.3. Spherical or Polar Structure with Orthogonal Rotation Axis (RRP)

The working volume is a hollow sphere, with the inner and outer radii determined either by the arrangement of the prismatic joint and the translation it allows, or by the lengths of the two parts of the arm [23].

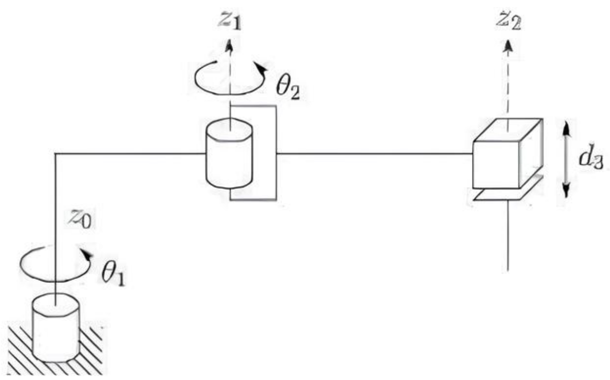


Fig I.19. Structure RRP. [24]

1.7.2.4. SCARA Structure

The characteristics of a SCARA structure (Selective Compliance Articulated Robot for Assembly) are: 3 cylindrical axes in series (RRP) with three degrees of freedom. The particularities of this structure are that they are precise and very fast [23].

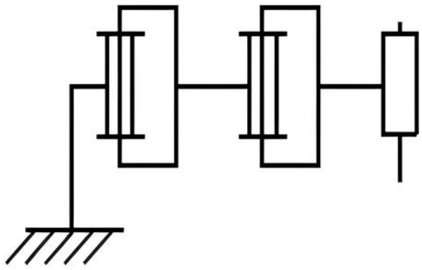


Fig I.20. SCARA Structure. [23]

1.7.2.5. Anthropomorphic Structure (RRR)

This structure has a more generalized architecture resembling the human arm. Its workspace has complex kinematics and dynamics, but its configuration is more flexible.

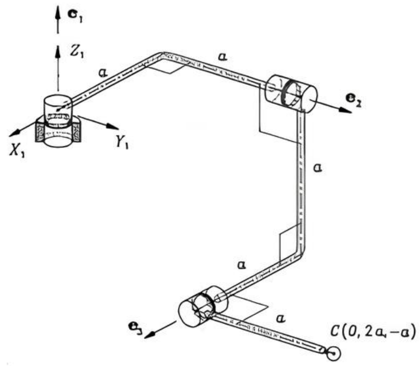


Fig I.21. RRR Structure. [25]

I.8. Characteristics of a robot

A robot should be chosen based on the application it is intended for. Here are some parameters to consider, if necessary:

- Payload capacity of the robot.
- Workspace: it is the set of space situations that the end effector of the robot can reach.
- Maximum speeds and accelerations, which determine the cycle time.
- Performance (the difference between a commanded situation and the average of the situations achieved) [26].

I.9. Application areas:

I.9.1. Industry

Factory robots have taken over most tasks that require precision, speed, and patience. Automobile factories use robots for cutting and assembling parts. On the assembly line, there are welding robots, manipulators, and painters.

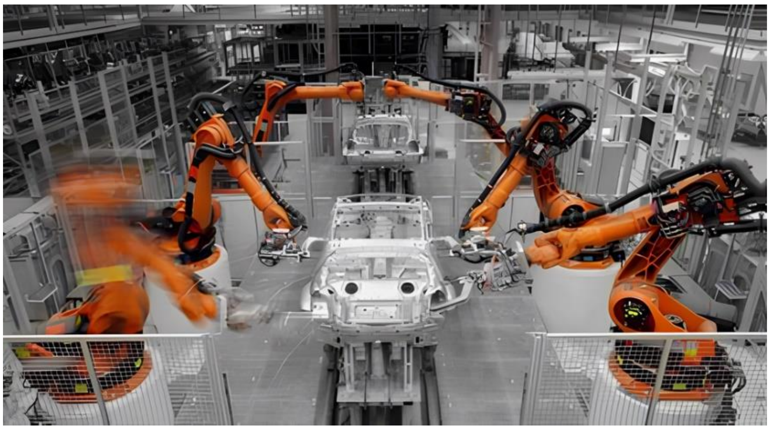


Fig I.22. Industrial robots.

I.9.2. Medicine

In the medical field, robots can be used to perform delicate operations that surgeons cannot perform or as aids in routine surgery. This is called surgery, which includes the introduction of the latest computer and robotic tools into medical and surgical practice. As a result, the practice of this "assisted surgery" is increasing, although still uncommon. These robots are found in surgery, tele-surgery, and macro-surgery.

Fig I.23. Surgical robots.

I.9.3. Agriculture

Robots can now increase the efficiency and productivity of farmers in crop management, as they can perform repetitive human tasks more effectively and efficiently.

Fig I.24. agricultural robotics.

I.9.4. Military

Robots are increasingly used in the military domain. Indeed, miniaturization now allows the creation of discreet robots rich in sensors, ideal for espionage or reconnaissance missions.

Fig I.25. Robot in the service of military forces.

I.9.5. Machining

The principle of machining involves using machine tools to remove material so that the original workpiece has the desired shape and size. With this technique, we obtain parts of high precision. Machining is widely used in aerospace, automotive, railway, nuclear, and even prototyping. The robot is capable of processing a wide variety of materials.

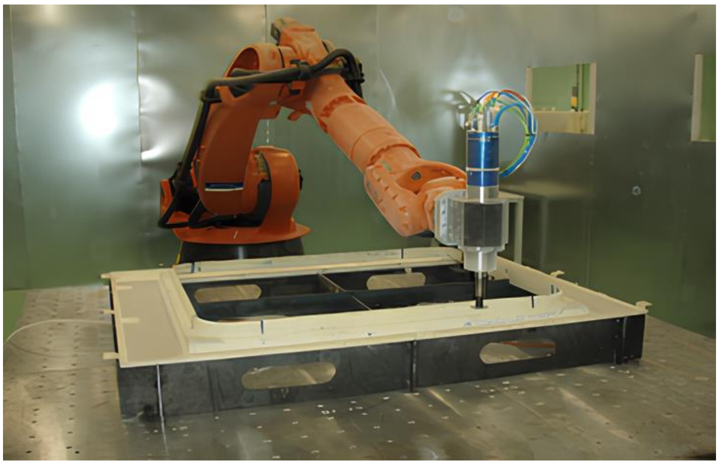


Fig I.26. Robotics in machining.

I.10. Advantages and Disadvantages of Robots:

I.10.1. Advantages:

- Increased production rates.
- Improved product quality.
- Reduction in material waste.
- Decrease in accidents.
- Low employment rates (reduction in labor).
- Greater flexibility and programmability (scalability).

I.10.2. Drawbacks:

- Replacement of human labor.
- Robots represent an expensive initial cost.
- Can only do what they are programmed to do; nothing more, nothing less.
- Increased unemployment.

I.11. Conclusion

In this chapter, a general overview of robotics was provided, including the history of robots, their structures, uses, different types, classifications, and fields of application. This information will assist in the construction of the manipulator arm.

CHAPTER II:

PROGRAMMING AND SIMULATING

II.1. Introduction

After providing a brief overview of robotics in the previous chapter, the focus now shifts to programming and simulation. Programming and the Integrated Development Environment (IDE) Arduino will be explored, followed by the development of the program for the system. Afterwards, Understanding the simulation program and its processes will allow for system simulation and improvement, code application, and reviewing the results

II.2. Programming the Arduino Uno Board

The Arduino Uno board can be programmed directly using the "Arduino software". The Atmega328 on the Uno board comes pre-programmed with a bootloader that allows loading code for a new application without using an external hardware programmer. It communicates with a computer using the ATMEL STK500 protocol. However, you can also program the controller on the board using the ICSP (In-Circuit Serial Programming) port. The source code for the firmware of the auxiliary Atmega8U2 controller is available.

II.2.1. Automatic Software Reset

It is possible to perform a reset via the Arduino software. Indeed, the DTR line on the Atmega8U2 is connected to the Reset line of the Atmega328 through a capacitor. When this line is brought to logic state 0, a Reset signal is sent to the controller.

II.2.2.USB Overcurrent Protection

The Arduino Uno board has a fuse protection for the USB port if a current of more than 500mA is requested. The disconnection will last until the source of excessive consumption has stopped.

II.2.3. Dimensions

The maximum lengths and widths of the PCB are 6.9 and 5.3 cm respectively.

- **Structural Diagram:**

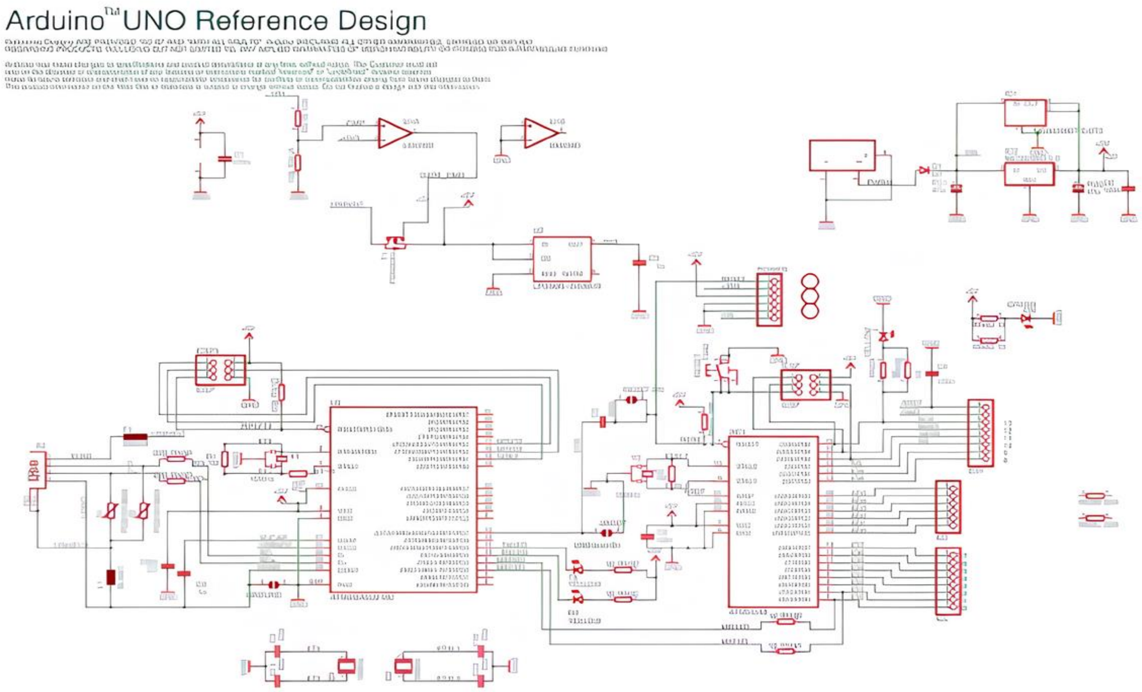


Fig II.1. Structural Diagram of the ARDUINO UNO.

II.3. Presentation of the Integrated Development Environment (IDE) Arduino

II.3.1. Description of the interface

The Arduino Integrated Development Environment (IDE), shown in Figure II.2, is a multi-platform application written in the Java programming language. The Arduino IDE is derived from Processing and Wiring projects for software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation. Additionally, it is capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is referred to as a "sketch" [27].

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring," named after the original Wiring project. This greatly simplifies common input/output operations. Users only need to define two functions (`void setup()` and `void loop()`) to create an executable program [27].

The Arduino software has the following main functions:

- To write and compile programs for the Arduino board
- To connect with the Arduino board to transfer programs to it
- To communicate with the Arduino board

This integrated development environment (IDE) dedicated to the Arduino language and programming of Arduino boards includes:

• A MENU BAR like any graphical user interface (GUI) software.

• A TOOLBAR providing direct access to the software's essential functions, simplifying its usability.

• A SYNTAX HIGHLIGHTING EDITOR for writing program code, with navigation tabs.

• A MESSAGE AREA displaying the status of ongoing actions.

• A TEXT CONSOLE showing messages regarding the compilation result of the program [28].

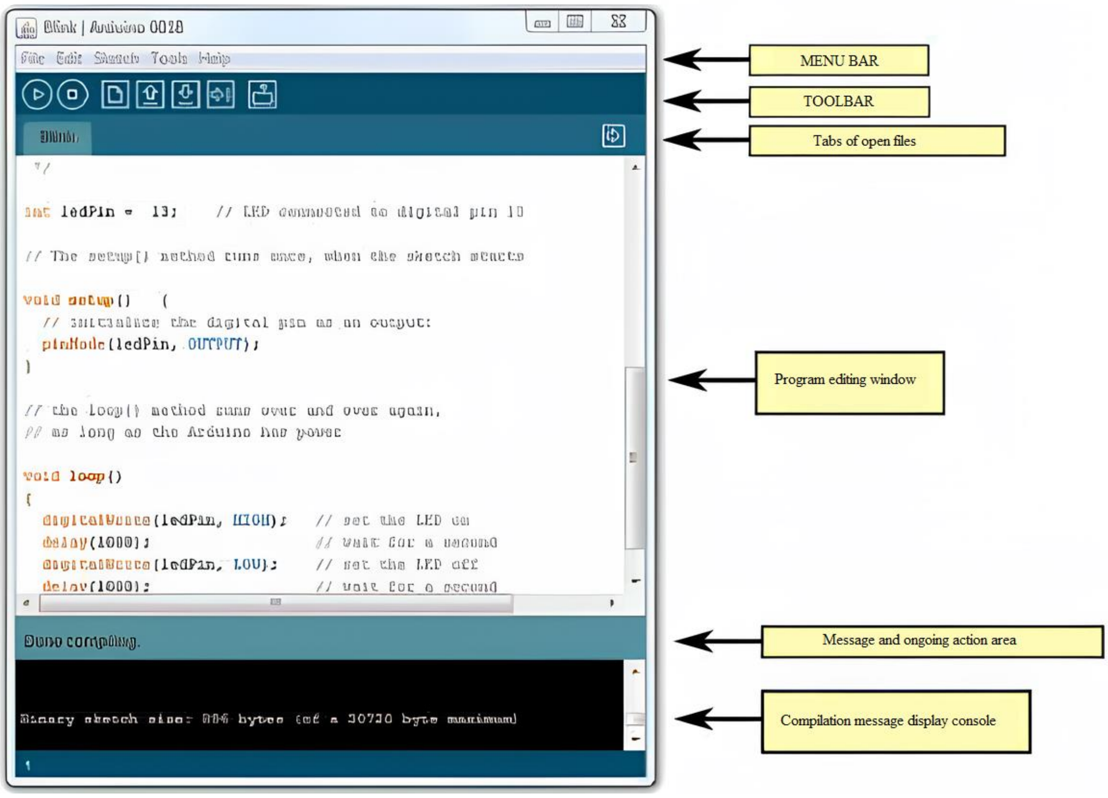


Fig II.2. Arduino IDE interface.

The Arduino software also includes a SERIAL TERMINAL (separate window) that allows displaying text messages received from the Arduino board and sending characters to the Arduino board. This functionality greatly facilitates program debugging, enabling the display of variable states, calculation results, or analog-to-digital conversions on the computer: an essential element for improving, testing, and correcting programs. [28].

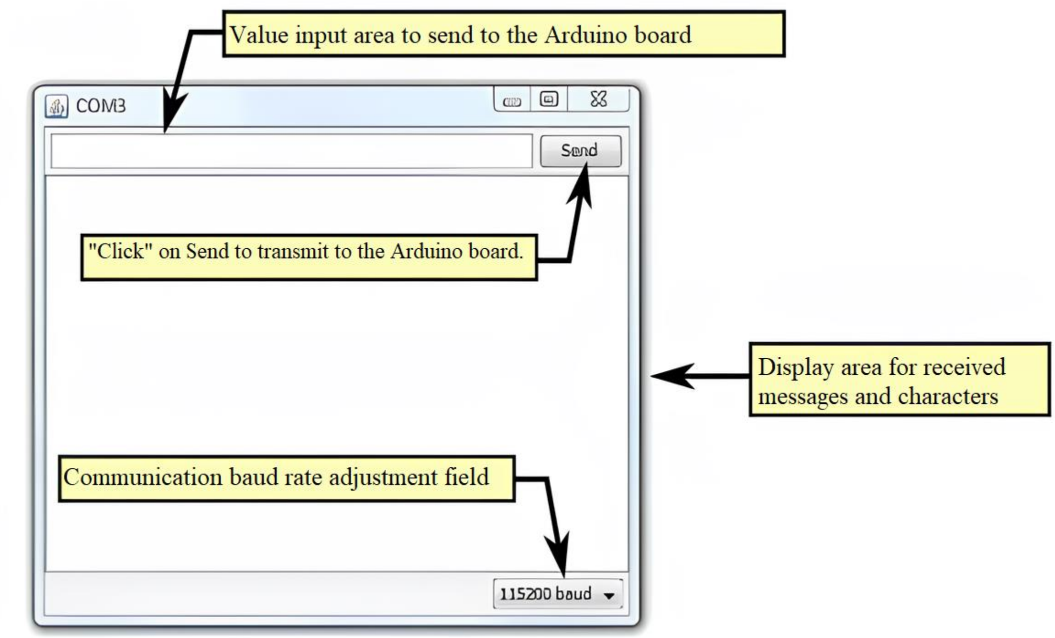


Fig II.3. Command Interface.

II.3.2. General Principle of Use

Code written with the Arduino software is called a sketch (or a program in French).

- These programs are written in the text editor. It has usual functionalities like copy/paste and find/replace.
- The message area provides the status of ongoing operations during saves, exports, and also displays errors.
- The text console displays messages produced by the Arduino software, including detailed error messages and other useful information.
- The toolbar allows for syntax checking and program transfer, creating, opening, and saving code, and opening the serial monitor.

- The menu bar provides access to all the features of the Arduino software.

II.3.3. Description of the Toolbar

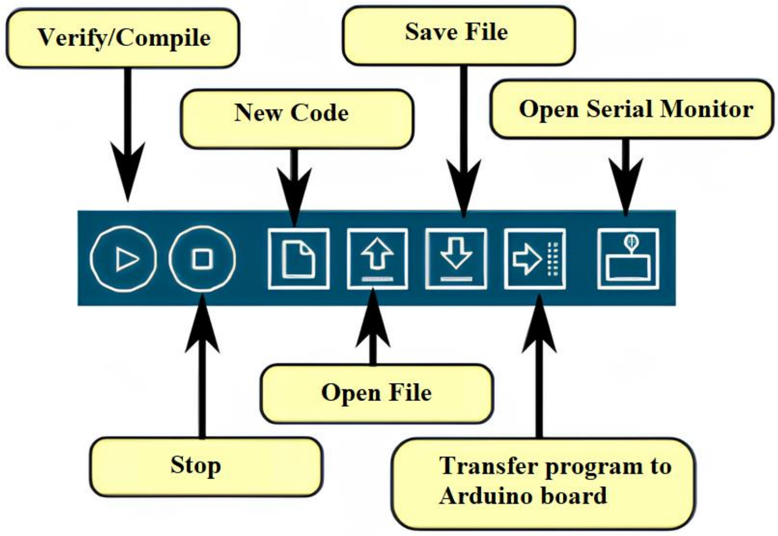


Fig.II 4. The Toolbar elements.

	Verify/Compile: Checks the code for errors
	Stop: Stops the serial monitor or other active buttons.
	New: Creates a new code (opens a blank editing window)
	Open: Opens the list of all programs in the "program book". Clicking on one of the programs opens it in the current window. Note: Due to a bug in Java, this menu
	Save: Saves the program
	Transfer to board: Compiles the code and transfers it to the Arduino board. See below "Transfer programs" for details
	Serial Monitor: Opens the serial monitor (or terminal) window.

Table II.1. Tools Definition.

II.4. Description of Program Structure

II.4.1. General Description of Program Parts

A user's Arduino program is a sequence of elementary instructions in text form, line by line. The board reads and executes the instructions one after the other, in the order defined by the lines of code, like in classic programming. This structure is divided into three parts:

- Description of constants and variables of the program.
- Main function: configuration of inputs/outputs and elements to be configured (this part will only be executed once) in the VOID SETUP() part.
- Loop function: description of the general operation of the program (management of interactions between inputs/outputs) in the VOID LOOP() part.

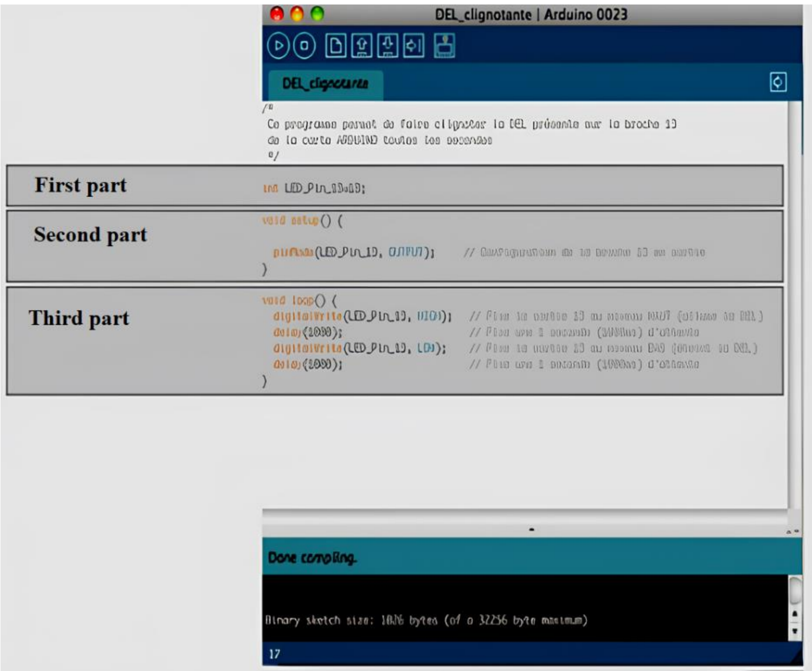


Fig II.5. IDE's graphical screen.

✓ **Note**:

It is possible to add comments to the program. This can be done in two ways:

At the end of the line by adding "//".

By enclosing comments between "/" and "/".

II.4.2. Detailed Description of Parts

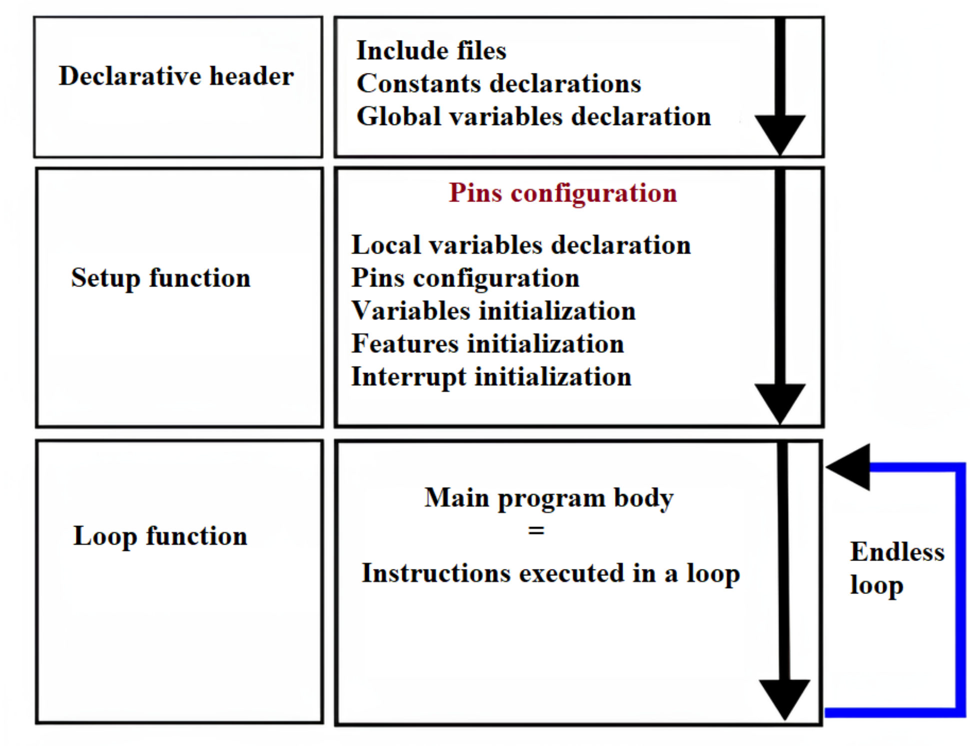


Fig II.6. Programming Structure.

II.4.2.1. Definition of Variables and Constants

In this part, we declare elements used throughout the program: constants (static) and variables (dynamic). These are memory locations used to store data (values) that can be used in the rest of the program.

II.4.2.2. Main Function: void setup()

This function is executed only once at the start of the program. It allows the configuration of the inputs and outputs of the board. The digital pins of the Arduino can be configured as both inputs and outputs.

PinMode (name, state) is one of the four functions related to digital inputs and outputs that we will see below.

II.4.2.2. Loop Function: void loop()

The loop() function does exactly what its name suggests: it runs endlessly in a loop, allowing your program to execute. In this loop, we define the operations to be performed. The loop() function is mandatory, even if empty, in every program.

II.5. Programming Our Robot's Arduino

Writing a program requires various steps.

II.5.1 Writing the Algorithm

An algorithm is a method for solving a problem. It is a way for the programmer to present their approach to the problem to others. In fact, an algorithm is a statement in a well-defined language of a sequence of operations to solve the given problem. An algorithm must therefore be understandable even by a non-informatician. Before writing a program, it is necessary to have an algorithm.

- Our algorithm's structure is as follows:

Control Robotic Arm()

1. Initialize the robotic arm components:

   1.1. Initialize servo objects for grip, wrist, and arm.

   1.2. Attach servo pins: grip to pin 10, wrist to pin 11, arm to pin 5.

   1.3. Set pin modes for additional components: pin 3 and pin 9 as OUTPUT.

   1.4. Set initial positions for servos:

      1.4.1. Set grip servo position to 90 degrees.

      1.4.2. Set wrist servo position to 110 degrees.

      1.4.3. Set arm servo position to 180 degrees.

   1.5. Delay for initialization (500 milliseconds).

2. Execute loop indefinitely:

   2.1. Move the arm downwards:

      2.1.1. Call function dwn().

      2.1.2. Delay for a specified duration (di).

   2.2. Close the grip:

      2.2.1. Call function closeg().

2.2.2. Delay for a specified duration (di).

2.3. Move the arm upwards:

2.3.1. Call function up().

2.3.2. Delay for a specified duration (di).

2.4. Move the arm to position 1:

2.4.1. Call function arm1().

2.4.2. Delay for a specified duration (di).

2.5. Move the arm to the right:

2.5.1. Call function right().

2.5.2. Delay for a specified duration (di).

2.6. Move the wrist to position 1:

2.6.1. Call function wrist1().

2.6.2. Delay for a specified duration (di).

2.7. Move the arm downwards:

2.7.1. Call function dwn().

2.7.2. Delay for a specified duration (di).

2.8. Open the grip:

2.8.1. Call function openg().

2.8.2. Delay for a specified duration (di).

2.9. Move the arm upwards:

2.9.1. Call function up().

2.9.2. Delay for a specified duration (di).

2.10. Move the arm to position 2:

2.10.1. Call function arm2().

2.10.2. Delay for a specified duration (di).

2.11. Move the wrist to position 2:

2.11.1. Call function wrist2().

2.11.2. Delay for a specified duration (di).

2.12. Move the arm to the left:

2.12.1. Call function left().

2.12.2. Delay for a specified duration (3000 milliseconds).

End of Algorithm.

II.5.2. Flowchart of the proposed Algorithm

A flowchart is a graphic representation of a process or algorithm that is commonly used in programming to demonstrate the progression of control inside a program.

- The foolwing Figure repraesenting Our flowchart:

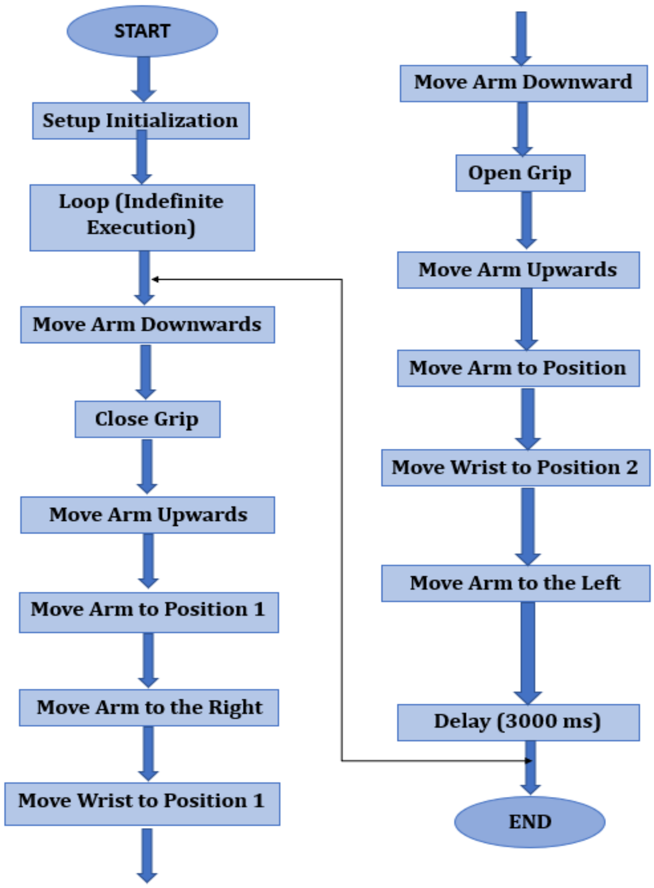


Fig II.7. The flowchart is a graphic representation.

II.5.3. Program development process

The program is written directly in line with the algorithm above. It's crucial to include comprehensive comments for non-programmers. Detail the program and partition it into logical blocks.

- The program is written below:

```
#include <Servo.h> // Including the Servo library for controlling servo motors
Servo grip;  // Creating a servo object for the grip
Servo wrist;  // Creating a servo object for the wrist
Servo arm;   // Creating a servo object for the arm
int di =500; // Setting a delay interval
void setup() {
   grip.attach (10); // Attaching the grip servo to pin 10
   wrist.attach (11); // Attaching the wrist servo to pin 11
   arm.attach (5); // Attaching the arm servo to pin 5
   pinMode(3, OUTPUT);  // Setting pin 3 as output
   pinMode(9, OUTPUT);  // Setting pin 9 as output
   grip.write(90); // Setting the initial position of the grip servo
   wrist.write(110); // Setting the initial position of the wrist servo
   arm.write(180); // Setting the initial position of the arm servo
   delay (500); // Adding a delay for initialization
}
void loop() {
   dwn (); // Calling the function to move down
   delay (di); // Adding a delay
   closeg (); // Calling the function to close the grip
   delay (di); // Adding a delay
   up (); // Calling the function to move up
   delay (di); // Adding a delay
   arm1 (); // Calling the function to move arm to position 1
   delay (di); // Adding a delay
   right (); // Calling the function to move right
   delay (di); // Adding a delay
   wrist1(); // Calling the function to move wrist to position 1
   delay (di); // Adding a delay
```

```
  dwn (); // Calling the function to move down
  delay (di); // Adding a delay
  openg(); // Calling the function to open the grip
  delay (di); // Adding a delay
  up (); // Calling the function to move up
  delay (di); // Adding a delay
  arm2 (); // Calling the function to move arm to position 2
  delay (di); // Adding a delay
  wrist2(); // Calling the function to move wrist to position 2
  delay (di); // Adding a delay
  left (); // Calling the function to move left
  delay (3000); // Adding a delay
}
void dwn () {
  analogWrite (9, 50); // Moving down by setting a PWM value to pin 9
  delay(400); // Adding a delay
  analogWrite (9, 0); // Stopping the movement
  delay(1000); // Adding a delay before next action
}
void up () {
  analogWrite (9, 240); // Moving up by setting a PWM value to pin 9
  delay(444); // Adding a delay
  analogWrite (9, 0); // Stopping the movement
  delay(1000); // Adding a delay before next action
}
void right (){
  analogWrite (3, 50); // Moving right by setting a PWM value to pin 3
  delay(850); // Adding a delay
  analogWrite (3, 0); // Stopping the movement
  delay(1000); // Adding a delay before next action
}
void left () {
  analogWrite (3, 240); // Moving left by setting a PWM value to pin 3
  delay(810); // Adding a delay
  analogWrite (3, 0); // Stopping the movement
```

```
    delay(1000); // Adding a delay before next action
}
void openg(){
    grip.write (90); // Opening the grip by setting a specific angle
}
void closeg (){
    grip.write(0); // Closing the grip by setting a specific angle
}
void wrist1(){
    wrist.write (70); // Moving the wrist to position 1 by setting a specific angle
}
void wrist2(){
    wrist.write (110); // Moving the wrist to position 2 by setting a specific angle
}
void arm1 (){
    arm.write (0); // Moving the arm to position 1 by setting a specific angle
}
void arm2 (){
    arm.write (180); // Moving the arm to position 2 by setting a specific angle
}
```

II.6. Simulation of the proposed System

We may think of online simulation as an innovative and beneficial way to study and build abilities in a variety of professions. Simulation uses software or tools to depict a realistic environment or specialized system, enabling users to interact with it and test a concept or design before putting it into action. Simulation has several advantages, including providing a safe setting for testing ideas and projects, giving opportunities for hands-on and experience learning, and allowing for creativity and experimentation without requiring major physical resources.

In this sense, Tinkercad is a well-known online simulation program that allows users to develop and test 3D designs and electronic circuits graphically and interactively. Tinkercad provides a safe and realistic simulation environment for exploration and experimentation, which helps users improve their creative and technical abilities in a variety of sectors.

II.6.1. Introducing Tinkercad.

Tinkercad was created in 2011 by Kai Backman and Mikko Mononen. Firstly, the information product was positioned as the first Web-platform for 3D-projecting, in which users could share results with each other. In 2013 the service was purchased by Autodesk company and complemented the product family 123D. Over the whole period, with the help of this service, users have created and published 4 million projects (3D models). In June 2017 Autodesk made a decision to move a part of its functional from another service Electroinics Lab Circuits.io, after that Tinkercad got really important instruments, which were able to significantly facilitate Arduino studying of projecting and programming new circuits for newbie developers. All old Circuits.io projects could be easily exported to Tinkercad [29].

II.6.2. Tinkercad Control Circuit Simulation.

To use Tinkercad, you must first register with Autodesk. Tinkercad registration is completely free and all you have to do is visit the site and complete the regular steps.

Then, to execute the simulation, we follow these steps:

II.6.2.1. Creating project

After loading the main page of Tinkercad, the project gallery is displayed and there is an option to call up an existing project or create a new one. As a result, a project with the default name will be created. By clicking on it, you can go to the viewing mode of the list of schemes included in this project, where you can change the properties of the project (indicating their name) by clicking on the corresponding button [29].

II.6.2.2. We add a new Circuits scheme

You can create a new scheme in Tinkercad in two ways - develop your own electronic scheme or use one of the ready-made electronic schemes available in the Tinkercad starter kits. To create a new scheme in Tinkercad, select Circuits from the menu on the left and select the Create new Circuit command [29].

II.6.2.3. Using a Tinkercad circuit in edit mode

By clicking on the «Change» command, the transition to edit mode is carried out. With the help of user-friendly and simple graphic interface, an electric circuit can be designed. It is also possible to highlight, move objects and delete them using «Drag & Drop» method with the help of a mouse [29].

In edit mode, the service working window is separated into two sections: the bookmark panel at the bottom, which is the component library. Above it, there is an area for visual alteration of the circuit using the toolbar, as well as an area where the circuit will

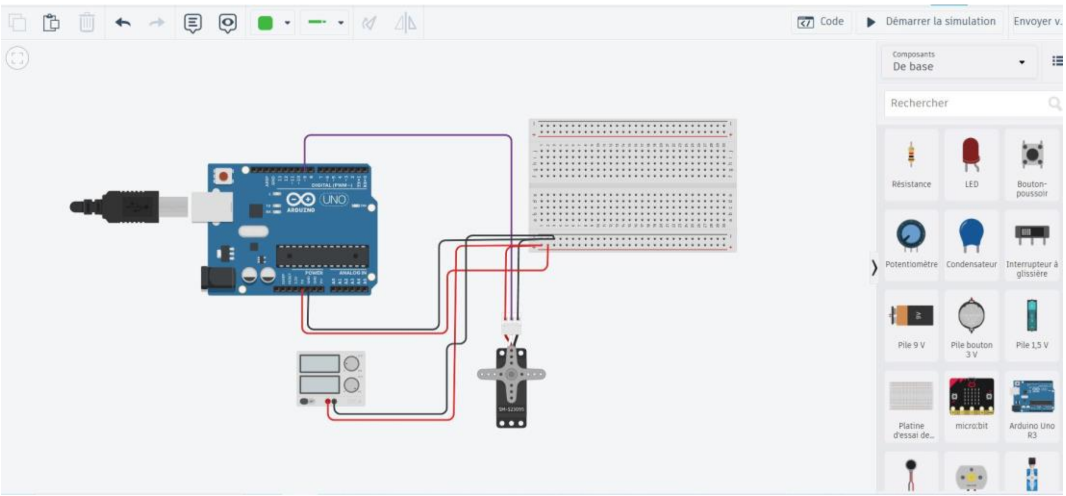


Fig II.8. The Control Circuit (Scheme editing process).

II.6.2.4. Control Circuit Creation step by step

1. Adding Components:
   - Drag and drop an Arduino Uno from the component library to the workplane.
   - In a similar way include a breadboard and power supply module.
2. Connecting Components:
   - Connect the Arduino Uno to the breadboard using jumper wires from the Arduino's digital pins.
   - Connect a power supply module to the breadboard to give electricity.
   - Ensure that the power and ground connections are properly wired.
3. Add Servo Motors: Connect five servo motor components to the breadboard. Arrange them in a proper configuration to accommodate wiring.
4. Wiring Servo Motors:
   - Attach each servo motor's power (typically red wire) and ground (usually black wire) to the breadboard's corresponding rails.

- Connect the signal wire (typically yellow or white) of each servo motor to separate digital pins on the Arduino Uno.

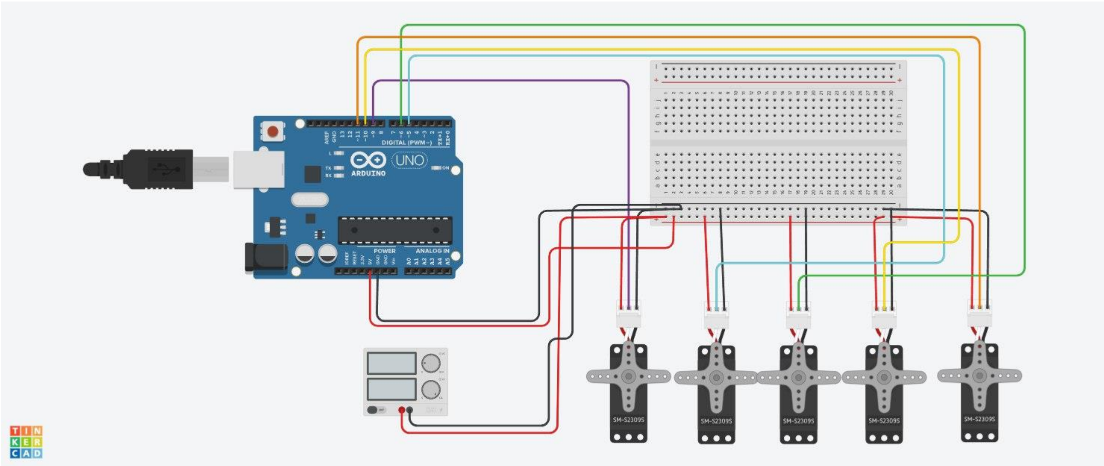


Fig II.9. The Control Circuit (Scheme After creation).

The process of picking from the component library is quite simple. The list of components appears on the right side of the screen. After choosing an element, click on it and then drag it to the correct location on the diagram before clicking again. The component list window can be hidden or displayed by clicking the «Components» button on the toolbar.

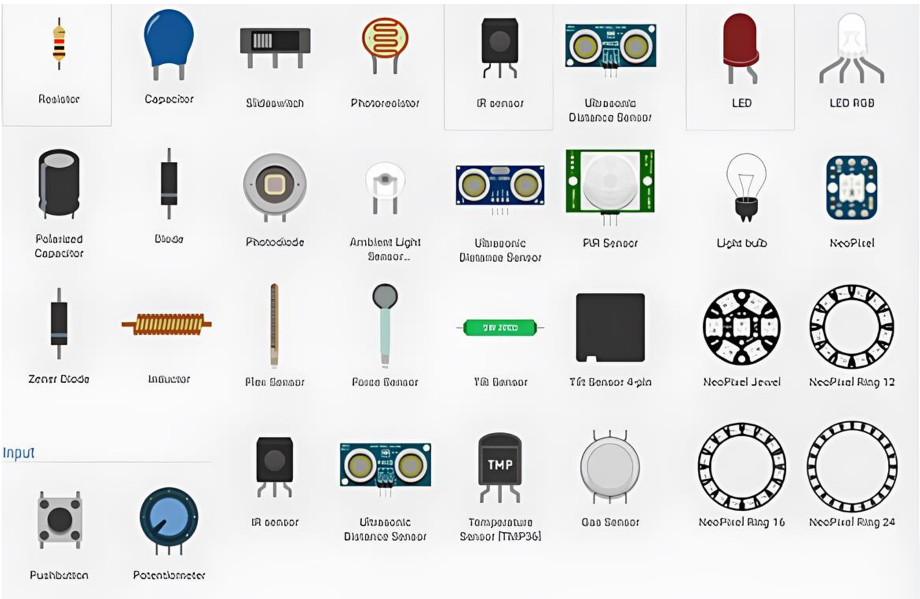


Fig II.10. Tinkercad's component list. [29]

II.6.2.5. Programming the sketch of virtual Arduino

After switching to the appropriate mode by selecting the "Code Editor" button in the upper panel all code editing tools become available.

We load the sketch that we previously wrote into the 'virtual controller' and run the simulator.

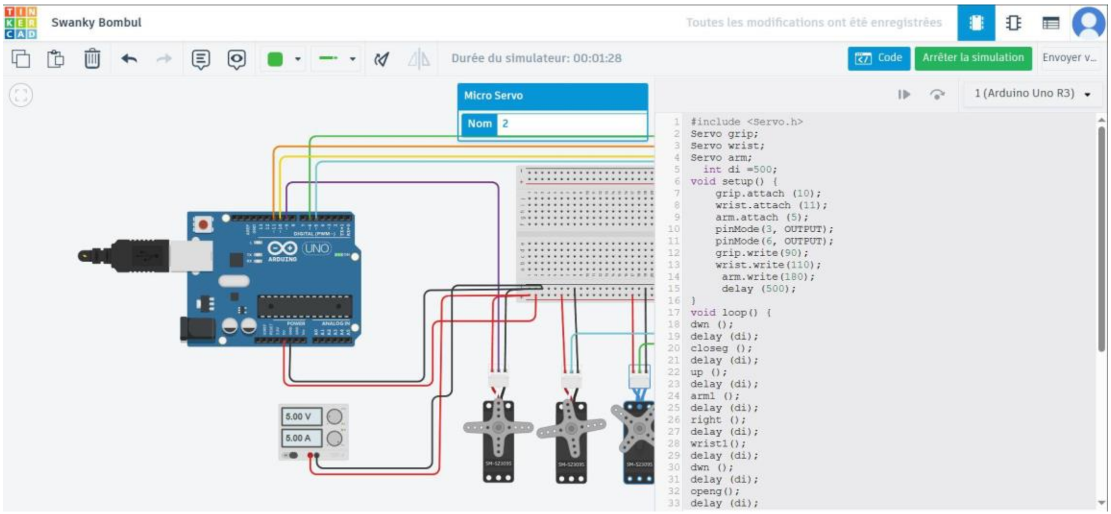


Fig II.11. the control circuit with the sketch loaded.

II.6.2.6. Programming the sketch of virtual Arduino

Running the Arduino simulator to start the simulator, click on the «Start Simulation» button in the upper panel. After starting the simulator, the ability to edit the electronic circuit of the system is blocked. The simulator starts with program compilation. If errors are found in the program, a corresponding message is issued. If there are no errors, the circuit is connected to the power source and the program commands begin. If necessary, in this mode it is possible to enable «Serial Interface Monitor» to provide I/O operations to the serial port [29].

II.6.3. The results of the Simulation

After running the simulation, we checked the accuracy of the assembly and discovered that all components are functioning as expected. The motion of the motors was consistent with the algorithm we implemented, indicating that the program we uploaded to the Arduino is working correctly. This success in the simulation phase is crucial because it assures us of the reliability and effectiveness of our design.

II.6. Conclusion

In this chapter, the IDE software was utilized to develop the program following the development of the algorithm and its graphical representation (flowchart). Furthermore, simulation using the Tinkercad application was explored, understanding its characteristics, simulating the controlled circuit, and applying the system's code to it. Finally, assurance of the specific code's validity and the construction and connection of the controlled circuit was ensured. This guarantee offers confidence that the robot will be able to carry out its specified tasks with accuracy and efficiency once it is physically built and installed

CHAPTER III :

EXPERIMENTAL

VALIDATION

III.1. Introduction

Programming and simulation were discussed in the previous chapter. Building on that, the transition from simulation to reality will occur in this chapter. It will involve steps such as modeling robot parts using SolidWorks and using the control circuit obtained, also discuss the components used in it, assemble and gather all the components, and install all the parts needed to obtain our robot.

III.2. Solidworks

III.2.1. Generality

Solidworks is also known as "DSS Solidworks". DSS stands for Dassault Systèmes, which is the developer of this CAD software. It is a 2D or 3D modeling software that uses parametric design. It generates 3 types of files related to three basic concepts: part, assembly, and drawing. These files are interconnected. Any modification at any level is reflected in all related files [30].

Fig III.1. Software Interface.

What is SolidWorks used for:

It is used to design 3D mechanical parts or assembled objects (e.g., bolts, motors, vehicles, etc.), as well as furniture and frameworks through the carpentry module. It allows for generating the machining sequence with tools available in the library [30].

Once the machining sequence is created, it can generate the numerical control file.

III.2.2. Robotic Arm Designe

III.2.2.1. The Base

The following figure represents the dimensions of the base in SolidWorks:

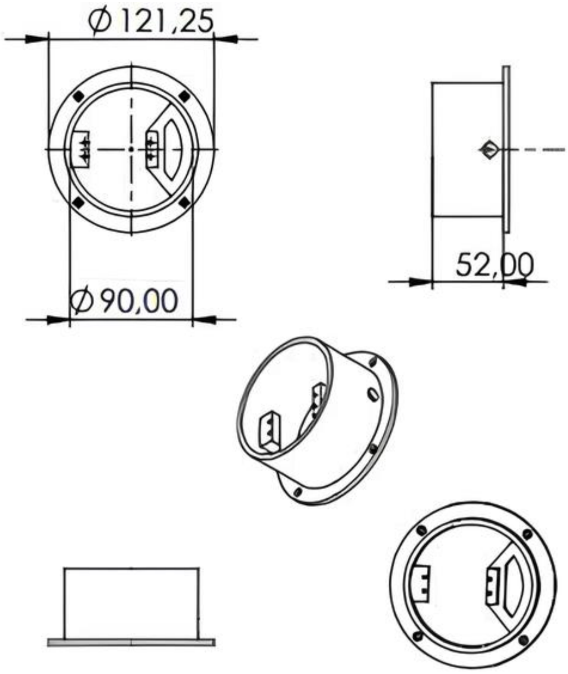


Fig III.2. The dimensions of the base in SolidWorks.

In the above figure, there is a 3D representation of the base:

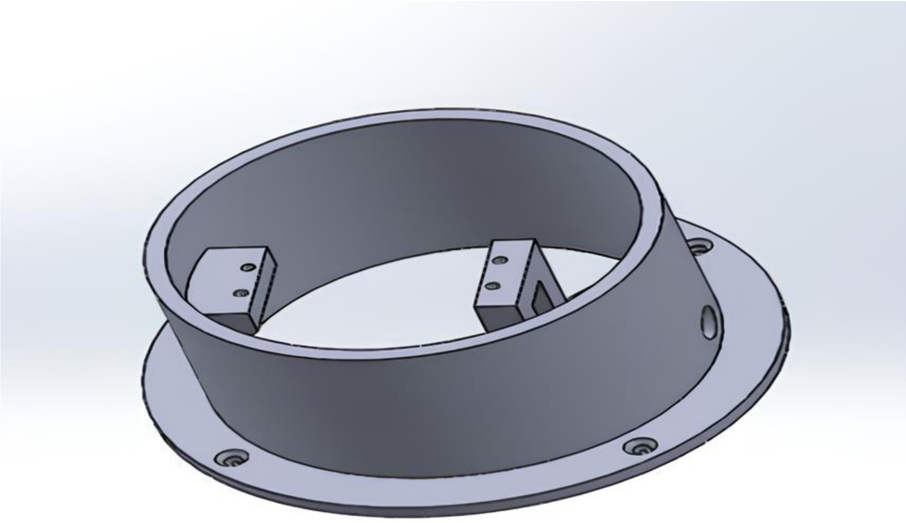


Fig III.3. The Base 3D representation .

III.2.2.2. The Base

We represent the dimensions of the first arm as follows:

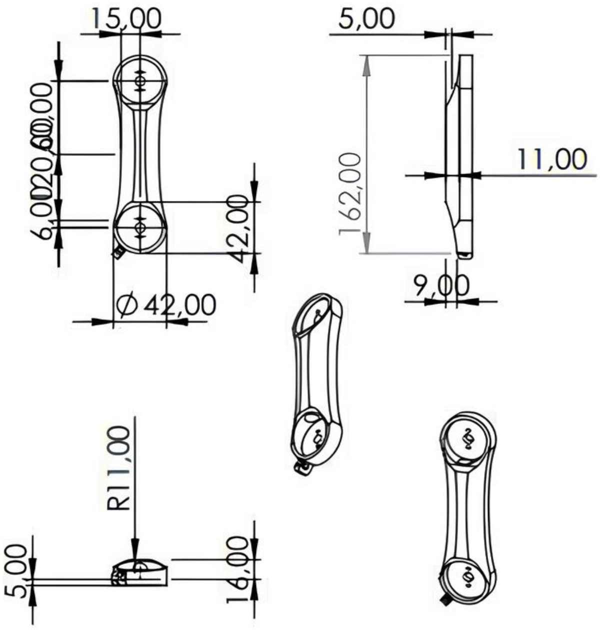


Fig III.4. The dimensions of the first arm.

The first arm represented in 3D:

Fig III.5. The first arm.

III.2.2.3. The Second arm

The figure above represents the dimensions of the second arm:

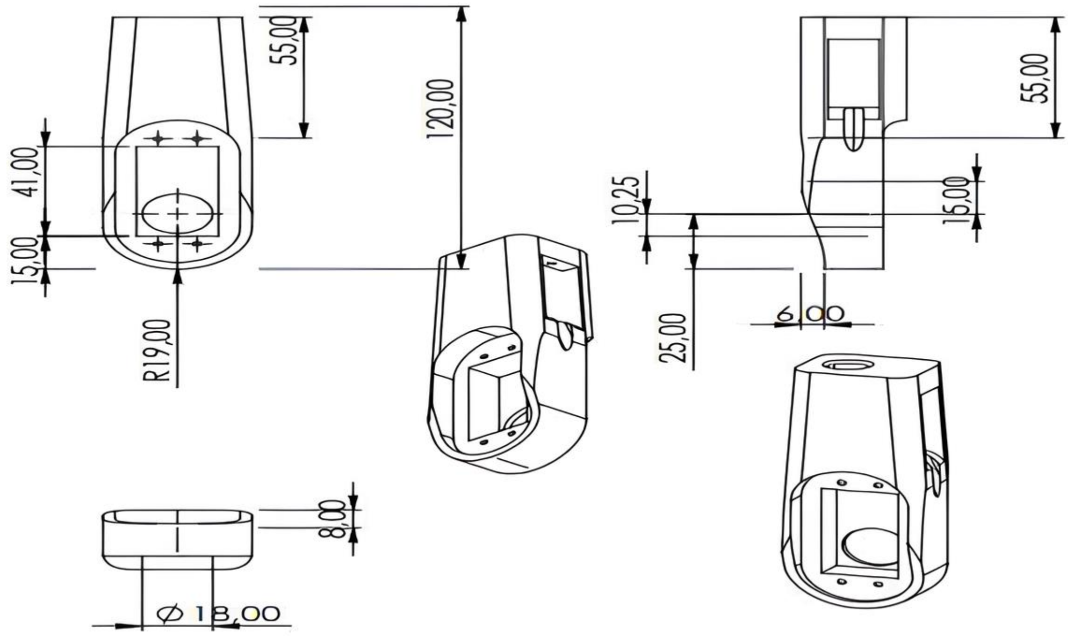


Fig III.6. The dimensions of the second arm**.**

The 3D design process goes as follows:

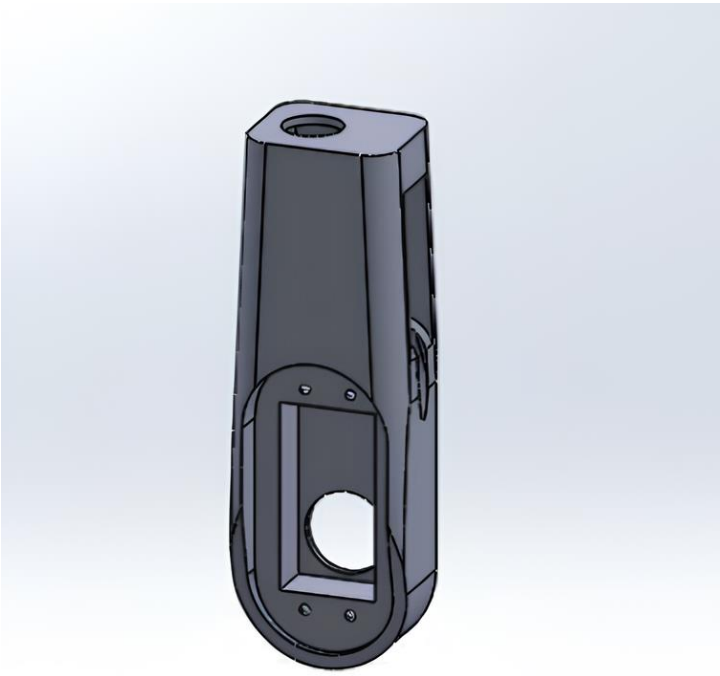


Fig III.7.the second arm**.**

III.2.2.4. The Third Arm(Wrist)

The following figure represents the dimensions of the third arm:

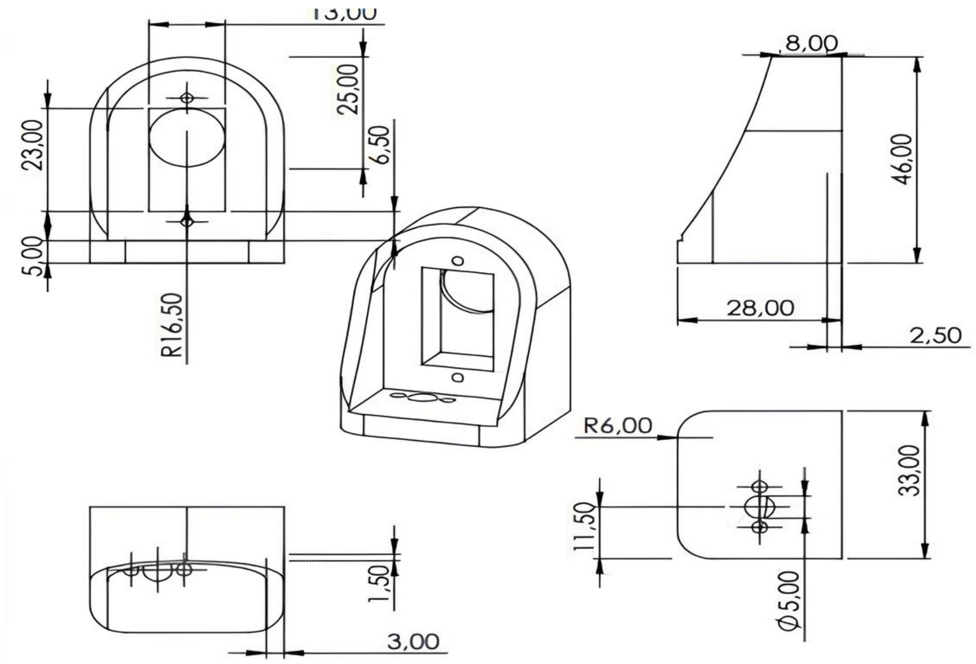


Fig III.8. The dimensions of the Third Arm.

The 3D model:

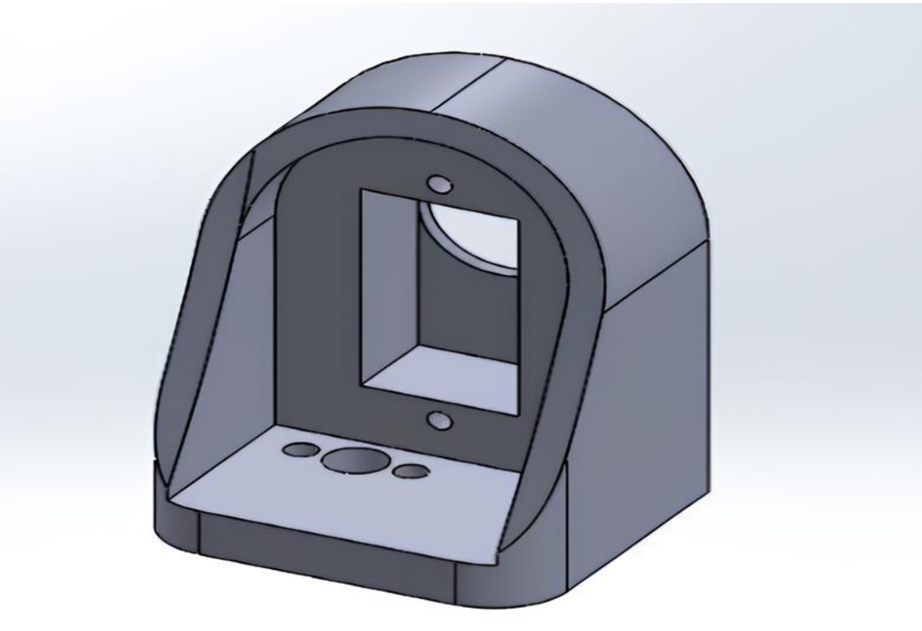


Fig III.9. The Third Arm

III.2.2.5. The Third Arm The Final Tool (gripper)

The following figure represents The gripper :

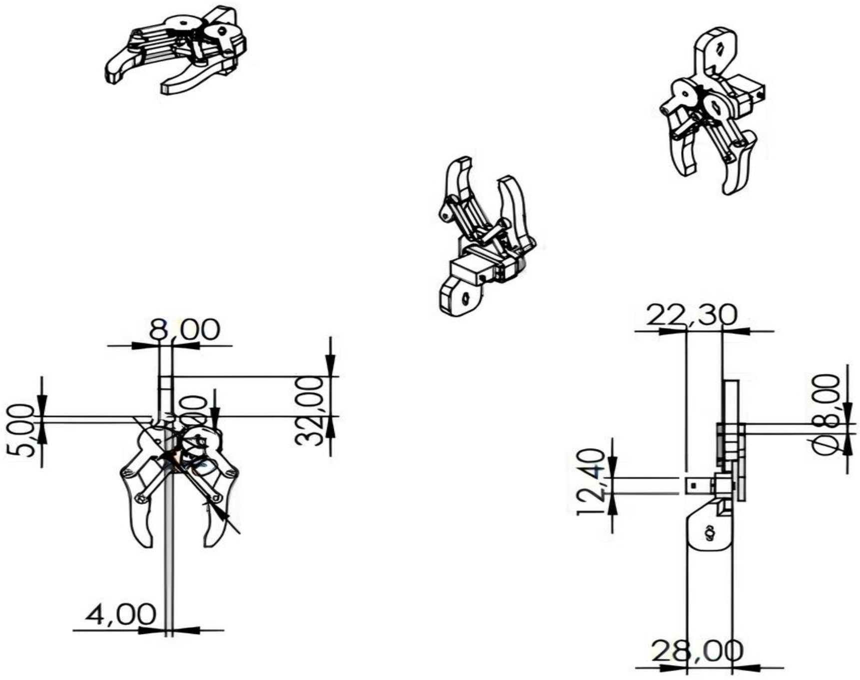


Fig III.10. The dimensions The gripper.

The 3D model of the gripper:

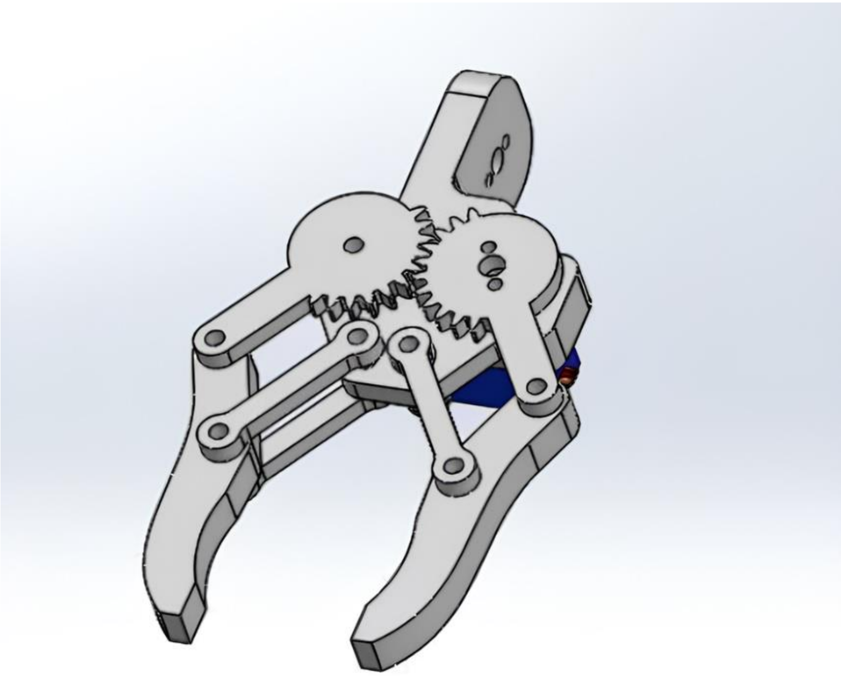


Fig III.11. The gripper.

III.3. Control circuit components

Based on the control circuit obtained in the simulation(**FigII.9**), we installed the control circuit for the arm, which consists of the following components

III.3.1. Arduino board:

The Arduino board is an open-source microcontroller, which has been embraced by the designer community. It is a digital system with a processor surrounded by memory and interfaces with the outside world.

These microcontrollers have limited performance but are small in size and consume little power, making them indispensable in any embedded electronics solution (cars, garage doors, robots, etc.). The Arduino board is not the most powerful microcontroller, but its architecture has been published as open-source [31].

In our project, we have chosen the Arduino UNO board.

III.3.1.1 .Arduino UNO board

The Arduino UNO is a development board featuring an Atmel ATMEGA328P microcontroller, commonly used by the general public.

The advantage of this board is to avoid the use of bare components. The necessary peripheral components are already present, so there is no need to solder them. Additionally, the ATMEGA is quite comprehensive and easy to use. Finally, it is also inexpensive and very popular.

The microcontroller contains, notably [32]:

- A CPU (Central Processing Unit)
- RAM memory
- ROM memory
- Input-output ports

The pinout of the Atmega 328 P 32-pin TQFP is provided in Appendix A.

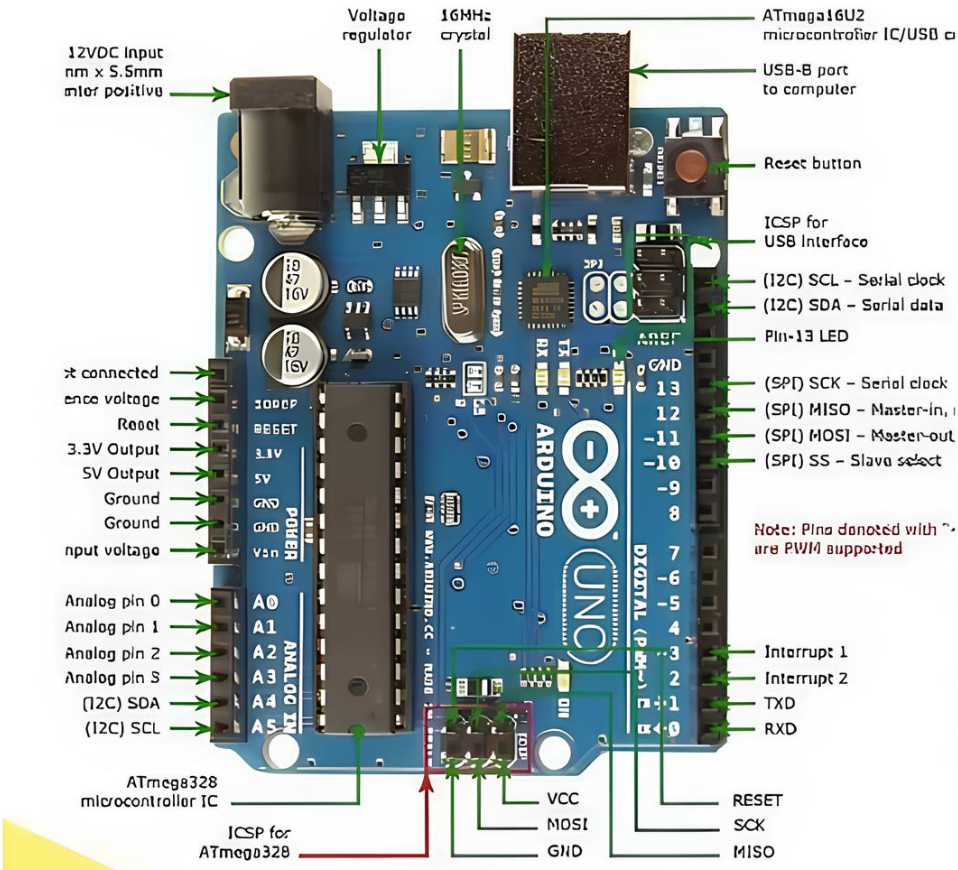


Fig.III.13. The Arduino UNO board.

III.3.1.2. Characteristics of the Arduino UNO board

Microcontroller	ATmega328
Operating Voltage	5V
Supply Voltage	7-12V
Supply Voltage (limits)	6-20V
Digital I/O Pins	14 (including 6 providing PWM output)
Analog Input Pins	6
Max Available Current for 3.3V Output	50 mA
Max Available Current for 5V Output	40 mA
Flash Program Memory	32 KB (0.5 KB used by bootloader)
SRAM Memory (volatile memory)	2 KB
EEPROM Memory (non-volatile memory)	1 KB
Clock Speed	16 MHz

Table III.1. Characteristics of the Arduino UNO [32].

III.3.2. Actuators (servo motors):

A servo motor consists of a DC motor, a rotating shaft, a position sensor for the shaft's orientation angle, and an electronic board. This board is used to control the shaft's position and motor operation. The servo motor is characterized by higher precision and increased power. It operates on a closed-loop system. A servo motor can maintain a predetermined position as instructed.

III.3.2.1. Advantages of servo motors:

One of the main advantages is that it can rotate at high speed while maintaining the nominal torque. They can also withstand nearly twice their nominal torque for a short period of time. However, be cautious as this leads to heating and therefore requires the ability to dissipate this heat afterwards. These motors are equipped with encoders, making them precise and reliable.

III.3.2.2. Drawbacks of servo motors:

These motors can be very expensive. They require complex encoders. If the encoder is not configured correctly, it may require more maintenance. Servo motors may also require the addition of a speed reducer to use low rotation speeds or to increase available torque. This obviously adds an element of complexity to the system and therefore an additional cost.

In our project, we have chosen:

Fig III.14. SG90 Micro Servomotor.

Fig III.15. MG996R Servomotor.

III.3.2.3.Technical Parameters of Servo Motors

References / Specifications	SG90	MG996R
Weight	9g	55 g
Stall torque	1.8 kgf ·cm	9.4 kgf ·cm (4.8 V), 11 kgf ·cm (6 V)
Operating voltage	4.8 V (~5V)	4.8 V à 7.2 V
Speed	0.1 s/60°	0.17 s/60o (4.8 V) ,0.14 s/60o (6V).
Rotation angle	180°	180°

Table III.2. Characteristics of Servos.

III.3.3. Breadboard

A breadboard is a plastic block with an array of electrical sockets large enough to accept thin connecting wires, component wires, and transistor and integrated circuit (IC) pins. Sockets are normally linked in rows of five on the board.

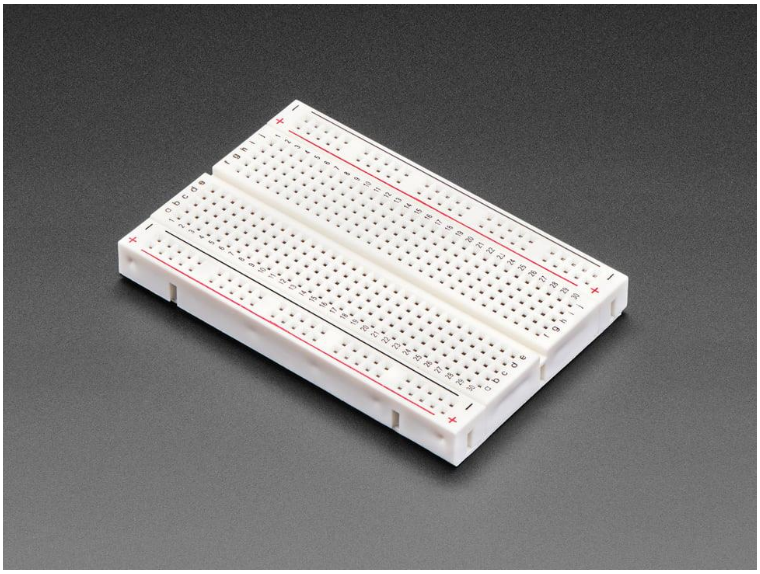


Fig III.16. The Breadboard.

- ✓ Note: The Tinkercad program does not include the perfboard used in our setup, so we replaced it with a breadboard.

III.3.4. The Alimentation

We used the following Power supply adapter charger for the alimentation:

Fig III.17. Power supply adapter charger.

We selected it after performing the following calculations:

Three Micro Servomoto SG90 consumes 0.55 amps for each

And

Two Servomoto MG996R consumes 0.9 amps for each

So

$$(0.55 \text{ amps x } 3) + (0.9 \text{ amps x } 2) = 3.45 \text{ amps}$$

The power supply adapter charger we chose produces 4 amps, which is more than 3.45 amps. Therefore, it is suitable for our robot.

III.4. Assembly of the robot's parts

By using a 3D printer, we obtained the parts for our robot.

Fig III.18. The robot's parts.

Firstly, we assembled MG996R Servomotor with the base due to its high torque and its ability to move and support the total weight of the arm.

Fig III.19. Servomotor assembled on the base.

After that, we installed the same motor, but this time on the shoulder, for lifting and lowering the arm due to its strength to support the total weight of the arm.

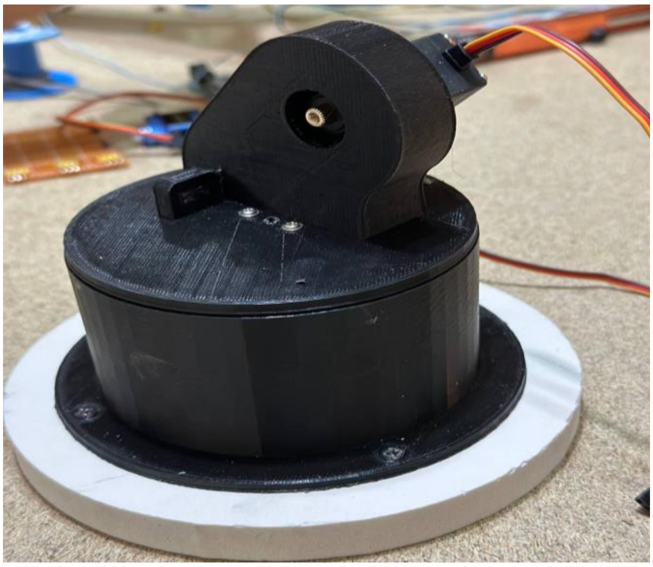


Fig III.20. Servomotor installed on the shoulder.

Next, we installed all the arms in their respective positions.

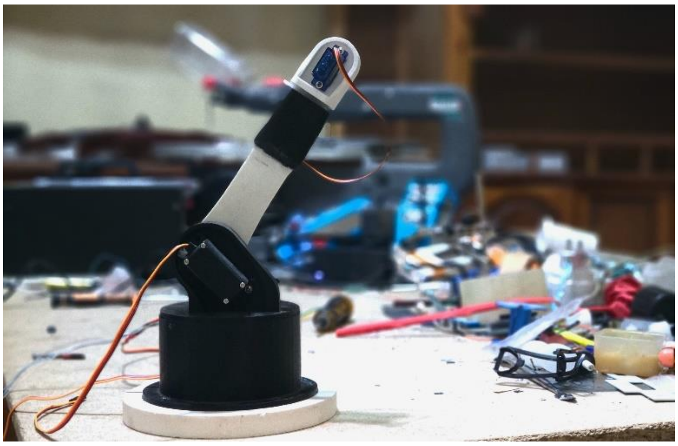


Fig III.21. All arms in their positions.

Finally, we assembled the gripper and installed it on the robot , also we installed 3 Micro Servomotors SG90 in their positions: one for rotating the wrist, another for lifting and lowering it, and the last one for the gripper.

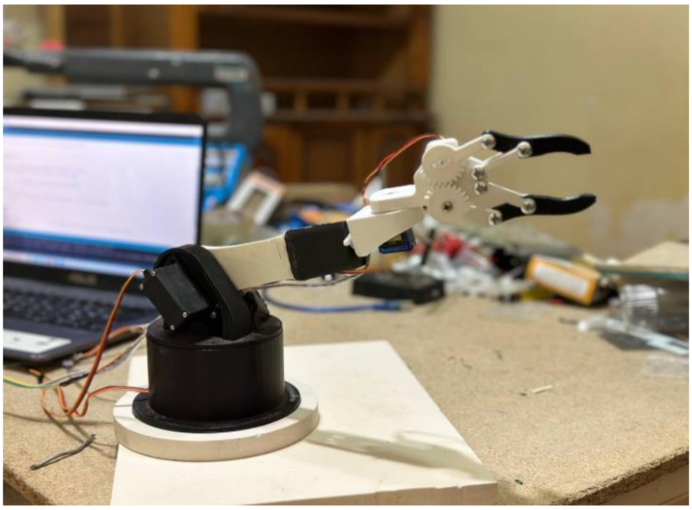


Fig III.22. The robot with the gripper installed.

The following pictures illustrate the final shape of the robot with some movements and they're organized as follows:

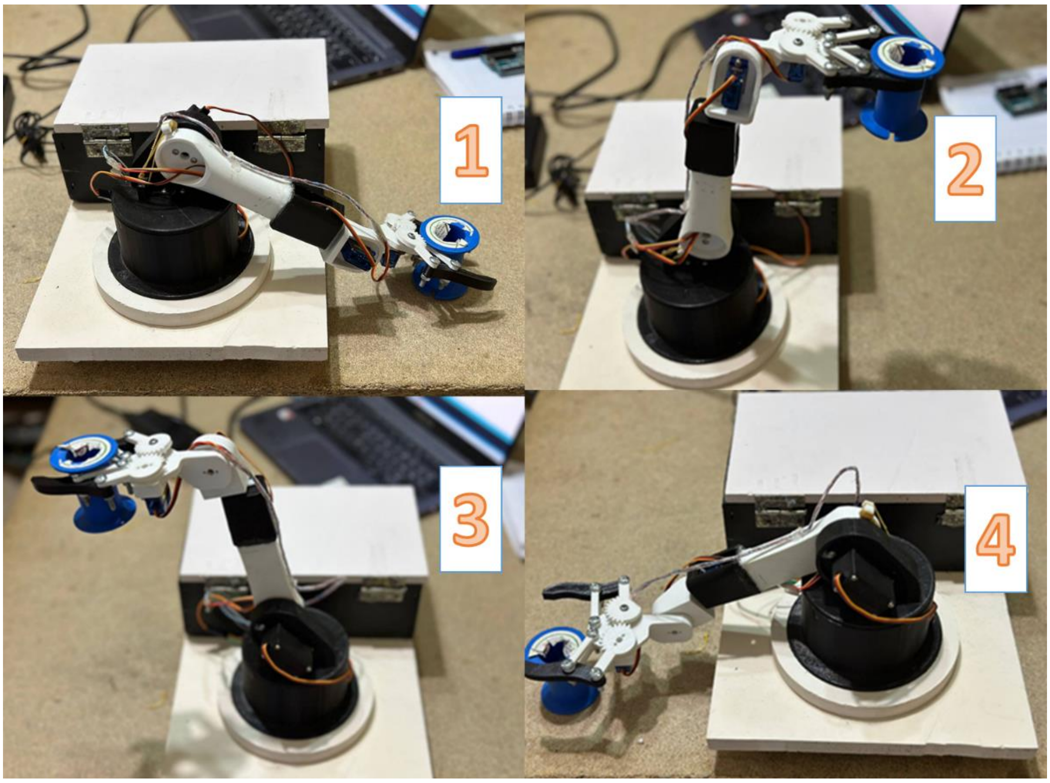


Fig III.23. The final assembly includes some movements.

III.5. Conclusion

In this chapter, several components of 3D design and hardware for executing the idea were discussed. Also, control circuit design and assembly stages were explored, which aided in reaching the goal of building a robotic arm. The information supplied in the previous two chapters was relied regarding this matter.

GENERAL CONCLUSION

Robotic arms are considered one of the most important applications in the field of robotics due to their ability to perform multiple tasks that require high precision and speed. Therefore, the importance of designing and implementing these systems in an efficient and reliable manner.

As conclusion this project successfully designed and implemented a robotic manipulator arm controlled by an Arduino Uno board. After studying the state of the art of robotic systems, their components, architectures, and applications, the process continued with programming the arm's control in the Arduino IDE environment. The algorithm, flowchart, and source code were developed, and the control circuit was simulated on Tinkercad to validate its functionality. The experimental validation phase then involved modeling the mechanical parts of the arm using SolidWorks, assembling the electronic circuit with servomotors, and physically constructing the prototype by 3D printing the components and assembling them with the electronic elements. Thus, the final robotic manipulator arm, controlled by the Arduino board running the developed program, was successfully realized.

This multidisciplinary project, combining theory, programming, simulation, and practical implementation, led to the successful design and fabrication of a functional robotic system, demonstrating the skills acquired in various aspects of robotics.

For future work, with the advancement of artificial intelligence (AI), the suggestion is to incorporate it into the project by integrating sensors that relay real-time changes. Consequently, the robot's movements can become instantaneous, with programming adapting dynamically based on the data available at each moment.

REFERENCES

[1] Moulières-Seban, T. (2017). Conception de systèmes cobotiques industriels. Approche cogni-tique Application à la production pyrotechnique au sein d'Ariane Group (Doctoral dissertation, ED SPI 209).

[2] BOUZIANE Fatima Zohra *“Rétroconception du bras horizontal de robot manipulateur de la cellule flexible”* UNIVERSITE ABOU BEKR BELKAID-TLEMCEN Option Ingénierie des systèmes mécaniques productiques, (2013).

[3] Composition d’un robot , UQAM/faculté des sciences , disponible en ligne sur : www.eu-reka.uquam.ca.

[4] EYRAUD, Charles-Henri. « Horloges astronomiques au tournant du XVIIIe siècle: de l'àpeu-près à la précision ». Doctoral thesis, Lyon 2, 2004.

[5] Official website IFR INTERNATIONAL FEDERATION OF ROBOTICS.

[6] Alexander Winkler, Sensorgeführte Bewegungen Stationärer Roboter. Münster:Monsenstein et Vannerdat OHG, Universitätsverlag Chemnitz 2016.

[7] Industrie and Technologies. Unimate le doyen de l'industrie. [En ligne]. Disponible sur :https://www.industrie-techno.com/unimate-le-doyen-de-l-industrie.32925.

[8] A. A. Korinski et A. E. Korinski, Bras Manipulateurs des Architectures et théories. MIR,Moscou, 1985.

[9] Polytechnique Montréal. Kinematics of the PUMA 560. [Online]. [Accessed on:September09,2019].
Available at: http://www.professeurs.polymtl.ca/luc.baron/index.php?id=210&lg=f.

[10] Daniel Ichbiah, genèse d'un peuple artificiel. Minerva, Mars, 2005. ISBN 2830707893.

[11] EL HAMZAOUI Oussama « Localisation et Cartographie Simultanées pour un robot mobile équipé d’un laser à balayage : CoreSLAM », Doctoral thesis, École Nationale Supérieure des Mines de Paris, September 25, 2012.

[12] BOSTELMAN Roger « PERFORMANCE MEASUREMENT OF MOBILE MANIPULATORS », Doctoral thesis, University of Burgundy, March 16, 2018.

[13] E. Dombre,Analysis and Modeling of Manipulator Robots, 2nd ed. Hermès Sciences, France, 2001.

[14] A. Cherfia, “Analyse Cinématique des Robots Parallèles”, Doctor of State thesis, Department of Mechanical Engineering. University of Frères Mentouri, Constantine (Algeria), 2008.

[15] Robotics Course (ensta-bretagne.fr).

[16] J.L. Boimond, « Robotique », « http://laris.univ-angers.fr/fr/vie-unite/liste-du-personnel1/jean-louis.html » Université ANGERS.

[17] Boukli Hacene Lotfi Fazil « Commande et supervision d’un ensemble de robots via internet », Master's Thesis, University of Sciences and Technology of Oran, April 26, 2012.

[18] Le Bars Fabrice « Analyse par intervalles pour la localisation et la cartographie simultanées ; Application à la robotique sous-marine », Doctoral Thesis, University of Western Brittany, November 23, 2011.

[19] FATOUX Julien « Mise en oeuvre d’un robot humanoïde et contribution à la génération de marches dynamiques optimales », Doctoral Thesis, University of Poitiers, August 7, 2006.

[20] Fisette P., Buyse H., Samin J.C. " Introduction à la robotique " Course, February 19, 2004.

[21] Rashidifar, Mohammad Amin, Rashidifar, Ali Amin, and Ahmadi, Darvish. " Modeling and Control of 5DOF Robot Arm Using Fuzzy Logic Supervisory Control », Article, vol.." Article, vol. 2, no. 2, 2089-4856, January 2013, International Journal of Robotics and Automation (IJRA).

[22] CRAIG, John J. « Introduction to robotics: mechanics and control », Pearson Education International, Book, 2009.

[23] GANGLOFF Jacques « Cours de Robotique », Course: ENSPS 3A Master ISTI, 2018.

[24] EL-HUSSIENY Haitham « Robotics Engineering », Course, Benha University, 2017.

[25] ANGELES Jorge « Fundamentals of robotic mechanical systems », Springer, Book, 2002.

[26] W. Khalil et E. Dombre “Modélisation identification et commande des robots”, 2nd edition. Hermès Sciences, Paris, 1999.

[27] BOUDHAR Lies « Conception d’un système de capteur pour l’analyse des propriétés physico-mécanique du sol en temps réel», Doctoral thesis, National Higher School of Agronomy El-Harrach -Algeria-, 2018.

[28] A. Shared, « Cours d’Arduino Chapitre 1 : Découverte de l’Arduino », University of Sousse, Tunisia.

[29] Leontiy Petrovich, Golubev Mykhailo Martynovych Tkach, Dmytro Anatoliyovych Makatyora, « Using tinkercad to support online the laboratory work on the design of microprocessor systems at technical university Information Technologies and Learning Tools », 2023, Vol 93, № 1. (2023).

[30] OUAKLI Hanane « CONCEPTION SOUS SOLIDWORKS ET COMMANDE DE LA MACHINE D'ENROULEMENT FILAMENTAIRE REALISEE » Master Thesis, National Polytechnic School, Mechanical Engineering Department, 2015

[31] BOUQUET Frédéric, BOBROFF Julien « Microcontroleur arduino », Course, University of Paris-Sud, Spring 2015.

[32] RAO Michael « Introduction à l'Arduino », Cours, Université de Lyon,2016.

Abstract

The use of manipulator robots is an industry standard for automating production lines. These robots are precise, robust and fast. To achieve their performance, they are designed with powerful actuators and they are made of massive parts. Several strategies such as the use of control algorithms and mechanical designs are used to reduce the danger of a robot manipulator and as well as improving its performance for users. This dissertation aims to study, through simulation and practical implementation, the design of a robotic arm manipulator with three degrees of freedom.

Key words: Robotics, Robots, Manipulator robots, Design, Simulation.

Résumé

L'utilisation de robots manipulateurs est un standard en industrie pour automatiser des chaînes de production. Ces robots sont précis, robustes et rapides. Pour atteindre leur performance, ils sont conçus avec des actionneurs puissants et ils sont faits de pièces massives. Plusieurs stratégies comme l'utilisation d'algorithmes de contrôle et des designs mécaniques sont utilisés pour réduire le danger d'un robot manipulateur et améliorer sa performance pour un utilisateur. Cette mémoire vise à étudier par simulation et réalisation pratique la conception bras robotique manipulateur à trois degrés de liberté.

Mots clés: Robotique , Robots ,Robots manipulateurs , Conception ,Simulation.

الملخص

يعد استخدام الروبوتات المناولة معيارًا صناعيًا لأتمتة خطوط الإنتاج. هذه الروبوتات دقيقة وقوية وسريعة. لتحقيق أدائها، تم تصميمها بمحركات قوية وهي مصنوعة من أجزاء ضخمة. يتم استخدام العديد من الاستراتيجيات مثل استخدام خوارزميات التحكم والتصميمات الميكانيكية لتقليل خطر مناور الروبوت وتحسين أدائه للمستخدم. تهدف هذه الأطروحة إلى دراسة تصميم مناور ذراع آلي بثلاث درجات من الحرية من خلال المحاكاة والتنفيذ العملي.

الكلمات المفتاحية: الروبوتات، الروبوتات، الروبوتات المتلاعبة، التصميم، المحاكاة.