KHAZAR UNIVERSITY

Faculty: Graduate School of Science, Art and Technology
Department: Physics and Electronics
Specialty: Electronics Devices

MASTER THESIS

Topic: Designing industrial pick and place robotic arm with AVR microcontroller

Master Student: Nicat Hajiyev Azar
Supervisor: Ph.D. in physics, assoc. prof.
Sevda Garibova Nizami

Baku - 2024

XƏZƏR UNİVERSİTETİ

Fakültə: Təbiət elmləri, Sənət və Texnologiya yüksək təhsil
Departament: Fizika və Elektronika
İxtisas: Elektron cihazlar və qurğular

MAGİSTR DİSSERTASİYASI

Mövzu: AVR mikrokontrolleri ilə sənaye “seç və yerləşdir” robot qolunun layihələndirilməsi

Tələbə: Nicat Azər oğlu Hacıyev
Elmi rəhbər: f.ü.f.d., dos. Sevda Nizami qızı Qəribova

Bakı - 2024

TABLE OF CONTENTS

CHAPTER I. LITERATURE REVIEW

CHAPTER II. MATERIAL AND METHODICS OF THE RESEARCH

CHAPTER III. THEORETICAL PART OF THE PROJECT

INTRODUCTION

In the rapidly evolving landscape of industrial automation, the demand for efficient and precise pick-and-place operations has become paramount. This abstract outlines the comprehensive development and optimization process of an industrial robotic arm designed specifically for the intricate task of picking and placing objects in a manufacturing environment. The objective is to enhance productivity, reduce labor costs, and ensure a high level of precision in handling diverse materials.

The robotic arm is designed with a modular and adaptive structure, allowing for easy integration into existing manufacturing setups. The kinematics and dynamics of the robotic arm are meticulously modeled and simulated to optimize its performance for various tasks, including speed, payload capacity, and reach. Advanced control algorithms, such as PID and machine learning-based approaches, are employed to ensure real-time adaptability and responsiveness to changing production requirements.

Sensor integration is a crucial aspect of the system, enabling the robotic arm to perceive and respond to its environment with high accuracy. Vision systems, tactile sensors, and proximity detectors are strategically incorporated to enhance the arm's perception capabilities. This sensory feedback loop is vital for the arm to navigate and interact with its surroundings, avoiding collisions and ensuring a reliable pick-and-place operation.

The end-effector, or gripper, is designed with versatility in mind, accommodating a wide range of shapes, sizes, and materials. Utilizing a combination of suction, mechanical gripping, and intelligent feedback, the gripper is optimized to securely handle delicate and irregularly shaped objects, contributing to the overall flexibility of the robotic arm.

The development process includes a rigorous testing phase, where the robotic arm undergoes extensive trials to validate its performance under various conditions. The optimization involves fine-tuning control parameters, enhancing sensor calibration, and refining the mechanical design to achieve optimal efficiency and reliability.

The economic feasibility and return on investment (ROI) are considered throughout the development process, ensuring that the implementation of the robotic arm aligns with industry standards and provides a tangible competitive advantage. The potential impact on job displacement is also analyzed, with a focus on retraining and up skilling the workforce to adapt to the changing industrial landscape.

Revelance of the topic: In the realm of industrial automation, the quest for enhanced efficiency, precision, and cost-effectiveness has spurred the evolution of robotic technologies. One pivotal application within this domain is the pick-and-place operation, a fundamental task in manufacturing that involves the precise manipulation and relocation of objects. As industries strive for increased productivity and adaptability in the face of dynamic market demands, the development of a pick-and-place industrial robotic arm becomes a focal point for transformative innovation.

Actuality of the project is to control a robotic arm is still hassle and time consuming in many industrial section. Many universities and researchers works in this field to make its simple and smarter (or skilled). The unique phase is human-robot interaction. This interaction mode varies in different purposes. Potential application range from the pure athletic application of Real Steel (Hollywood robotic boxing movie) or the sci-fi television channels Robot Combat League to augmenting traditional robotic surgery (Da Vinci system) etc (F. Pugin, 2011).

This thesis embarks on a comprehensive exploration of the design, development, and implementation of an industrial robotic arm specifically tailored for pick-and-place tasks. What sets this endeavor apart is the integration of an AVR (Alf and Vegard's RISC processor) microcontroller as the central nervous system of the robotic arm, introducing a novel dimension to the control architecture. The utilization of AVR technology brings forth a unique set of advantages, including real-time processing capabilities, energy efficiency, and a compact footprint, making it an ideal candidate for the intricacies of industrial automation (Morteza Hadipour, 2020).

The purpose of this research stems from the need to create a robotic arm that not only excels in precision and adaptability but also adheres to the principles of cost-effectiveness and simplicity in control systems. The AVR microcontroller, renowned for its reliability and versatility in embedded systems, is harnessed to orchestrate the intricate dance of motors, sensors, and actuators that characterize the movements of the robotic arm.

The evolution of industrial robotics has witnessed a paradigm shift towards the integration of microcontrollers, ushering in an era of intelligent automation (Balkeshwar Singh, 2013). The AVR microcontroller, known for its role in diverse applications, from consumer electronics to automotive control systems, is strategically chosen as the cornerstone of this robotic arm's control architecture. This choice is driven by the desire to strike a balance between computational power and efficiency, ensuring that the robotic arm meets the stringent requirements of industrial applications while remaining accessible and adaptable.

The object of this thesis is multi-faceted, encompassing the mechanical design of the robotic arm and the implementation of control algorithms on the AVR microcontroller. The aim is to develop a robotic system that not only navigates the complexities of pick-and-place operations but also does so with a level of precision that exceeds current industry standards. Additionally, the incorporation of the AVR microcontroller introduces an element of scalability, enabling seamless integration into existing manufacturing setups without imposing undue complexities.

As industries continue to embrace the era of Industry 4.0, characterized by smart factories and interconnected systems, the role of innovative robotic solutions becomes indispensable (Niko Sudibjo, 2019). This thesis seeks to contribute to this transformative wave by presenting a pick-and-place industrial robotic arm that combines the precision required for modern manufacturing with the intelligent control facilitated by the AVR microcontroller. Through a detailed exploration of the design process, control strategies, and experimental validation, this research aims to offer a blueprint for the next generation of cost-effective and efficient industrial automation solutions.

Pick and place robotic arms heavily depend on joints, which are used to join or connect the two consecutive bodies in the robot and can be rotary joint. Joints define the movement of the arm. Arms decide the degree of freedom of the components. Consequently, all robotic arm consists of following basic components (H. A. F. Almurib, A review of application industrial robotic design, 2012): controller, manipulator, grippers and power source.

A robotic arm has a mechanical structure that alters its form using a group of electric motors that behave like servo motors, pneumatic, or hydraulic actuators. They are usually programmable, with similar functions to a human arm; the arm may be the sum total of the mechanism or may be part of a more complex robot. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement. The links of the manipulator can be considered to form a kinematic chain. The terminus of the kinematic chain of the manipulator is called the end effector and it is analogous to the human hand.

In conclusion, this abstract provides an overview of a meticulously designed and optimized pick-and-place industrial robotic arm, emphasizing its adaptability, precision, and efficiency in modern manufacturing environments. The integration of advanced control algorithms, sensor technologies, and versatile end-effectors positions this robotic arm as a valuable asset for industries seeking to enhance their automation capabilities and stay competitive in a rapidly evolving market.

CHAPTER I. LITERATURE REVIEW

1.1. Robotic arms description and types of robotic arms

Robotic arms come in various types, each designed to fulfill specific functions and cater to diverse applications. The classification of robotic arms is often based on their structure, degree of freedom, and application. Here is an overview of different types of robotic arms.

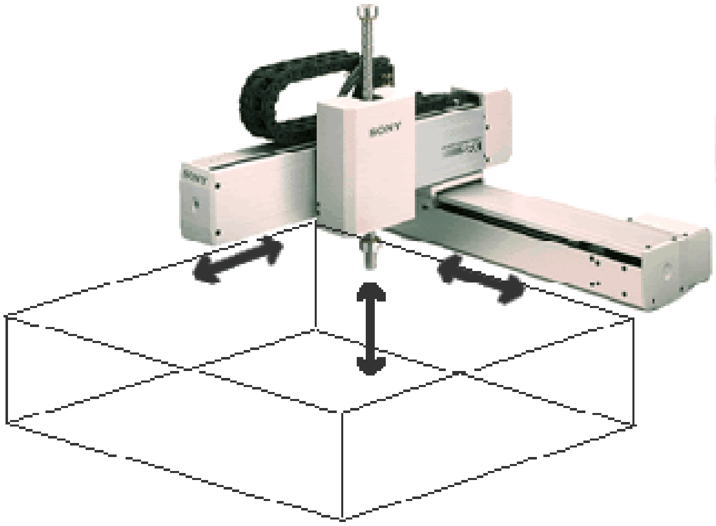


Figure 1.0. Cartesian robot

Figure 1.1. Cylindrical robot

Cartesian or rectangular robots (Figure 1.0) (COLLINS): Cartesian robots have three linear joints that move along the X, Y, and Z axes. The arm's movements resemble those of a gantry system, providing precise and controlled motion in a specific coordinate system (Jisu Elsa Jacob, 2022). Commonly used in tasks requiring precise linear movements, such as pick-and-place operations, CNC machining, and 3D printing.

Cylindrical robots (Figure 1.1): Cylindrical robots feature a rotating base, a prismatic joint for vertical motion, and a wrist joint for rotational movement. This structure allows them to operate in a cylindrical workspace (Cenk Baykal, 2017). Suitable for tasks like assembly operations, arc welding, and handling objects along a circular path.

Spherical or polar robots (Figure 1.2) (Types of Robots): Spherical robots have a jointed structure that allows them to move in a spherical workspace. They typically consist of a shoulder joint, an elbow joint, and a wrist joint**)** (Marek Bujňák, 2022). Ideal for tasks requiring a wide range of motion, such as painting, welding, and assembly in curved or spherical spaces.

Articulated robots (Figure 1.3): Articulated robots have multiple rotary joints, typically resembling a human arm. This design provides a high degree of freedom, enabling complex movements and versatile applications (Bradley Saund, 2013). Commonly used in tasks that require flexibility and dexterity, including assembly, material handling, and tasks in confined spaces.

Figure 1.2. Polar robot

Figure 1.3. Articulated robot

Selective compliance assembly robot arm - SCARA robots ((Figure 1.4): SCARA robots feature two parallel rotary joints for planar movements and a prismatic joint for vertical motion. They are known for their speed and precision in horizontal planes (S. Suri, 2018). Well-suited for applications like assembly, pick-and-place, and material handling in a planar workspace.

Parallel robots (Figure 1.5): Parallel robots have multiple interconnected arms with both revolute and prismatic joints. The end effector is connected to the base by multiple limbs, providing enhanced stability and precision (Taghirad, 2013). SCARA is sed in high-precision applications such as flight simulators, machine tools, and applications requiring rapid and precise movements.

Delta robots (Figure 1.6): Delta robots have three arms connected to a common platform. Each arm features parallel kinematics, providing fast and precise movements in a three-dimensional workspace (Poppeova, Uriček, Bulej, & Šindler, 2011). Widely used in industries like packaging, food processing, and high-speed pick-and-place operations due to their rapid and accurate motion capabilities.

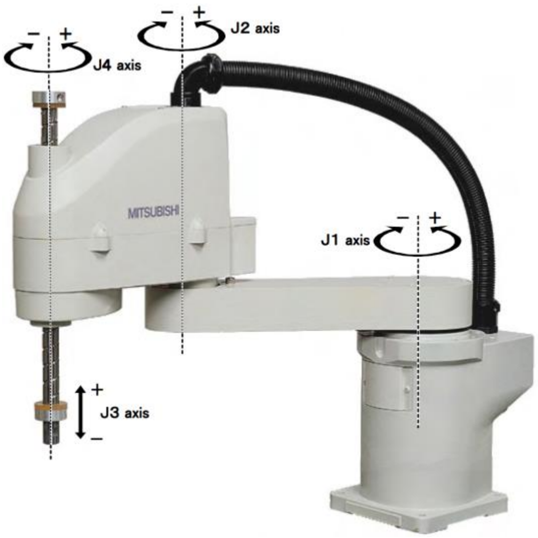


Figure 1.4. SCARA robot

Figure 1.5. Parallel robot

Collaborative robots (Cobots) (Figure 1.7): Collaborative robots are designed to work alongside humans in a shared workspace. They are equipped with sensors and safety features to ensure safe interaction with human operators (Danica Kragic, 2018). Suited for tasks are requiring human-robot collaboration, including assembly, inspection, and material handling in environments where close interaction with humans is necessary.

Figure 1.6. Delta Robot

Figure 1.7. Collaborative Robot

Soft Robots (Figure 1.8): Soft robots use flexible materials and pneumatics or hydraulics to achieve movement. They differ from traditional rigid robotic arms and are often designed to mimic the flexibility of biological organisms (Daniela Rus, 2015). Emerging in fields such as medical surgery, delicate material handling, and tasks that require interaction with fragile objects.

Industrial robot arms (Figure 1.9): Industrial robot arms are designed for various manufacturing applications and are often programmable for tasks such as welding, painting, assembly, and material handling (H. A. F. Almurib, A review of application industrial robotic design, 2012). Widespread use in automotive assembly lines, electronics manufacturing, and other industrial processes.

Figure 1.8. Soft Robot

Figure 1.9. Industrial Robot

Figure 1.10. Underwater Robot

Underwater robotic arms (Figure 1.10): Designed for subsea operations, these robotic arms are equipped to withstand underwater conditions and perform tasks such as maintenance, inspection, and manipulation in offshore environments. Deep-sea exploration, underwater construction, and offshore oil and gas industry tasks. The choice of a robotic arm type depends on the specific application, workspace requirements, precision needs, and environmental conditions in which it will operate. The continual advancement of robotic technology continues to introduce new types of robotic arms with enhanced capabilities and broader applications (Loris Barbieri, 2018).

1.2.Project aim and objective

The aim of this project is design an autonomous robot with complete system allow the robot wander about its environment and to interact with certain object that its encounter. The importance of automatic robotic arms in various industries is profound, driven by their ability to revolutionize manufacturing processes, enhance efficiency, and address complex challenges. The significance of this technology can be understood through several key perspectives:

Increased efficiency and productivity - automatic robotic arms excel in performing repetitive tasks with a level of precision and speed that is often unattainable for human workers. This translates to increased production rates and improved overall efficiency in manufacturing processes. Unlike human workers, robotic arms do not require breaks or shifts. They can operate continuously, contributing to a significant increase in overall production output and reducing the time required to complete tasks (C. Schmidbauer, 2023).

Enhanced safety - robotic arms are adept at handling tasks that involve hazardous materials, extreme temperatures, or environments unsafe for human workers. By taking on these tasks, they contribute to a safer work environment and help reduce workplace injuries. Consistency in Safety Protocols: Robotic arms consistently adhere to safety protocols, minimizing the likelihood of accidents caused by human error. This reliability is especially crucial in industries where strict safety standards must be maintained (Achim Buerkle, 2021).

Precision and accuracy: Robotic arms can execute tasks with a high degree of accuracy and repeatability. This consistency ensures that products meet stringent quality standards, reducing defects and improving overall product quality (China, 2014).

In industries such as electronics and pharmaceuticals, where micro-precision is paramount, robotic arms can perform intricate tasks with precision that surpasses human capabilities.

Flexibility and adaptability: Automatic robotic arms can be programmed to perform a wide range of tasks, making them highly versatile. This adaptability is especially beneficial in industries with diverse product lines and changing production requirements (Gordon Cheng, 2001). Robotic arms can be easily reprogrammed for different tasks, allowing for quick changeovers in production processes. This agility is essential for industries that need to respond rapidly to shifts in market demands.

Cost-effectiveness: While the initial investment in robotic arms can be substantial, the long-term cost savings in labor expenses, especially for repetitive and labor-intensive tasks, are significant. This cost-effectiveness contributes to the overall economic viability of automated systems (Yechu Hua, 2022). Automation minimizes the risk of errors and defects, resulting in lower production costs associated with rework, waste, and quality control measures.

Competitive Advantage: Companies that embrace automated technologies, including automatic robotic arms, position themselves at the forefront of innovation and modernization. This can lead to a competitive advantage in the marketplace, attracting customers and investors alike. The agility and efficiency brought about by automated systems enable companies to respond quickly to market demands, launch new products faster, and adapt to changes in consumer preferences (Nidhi Chahal, 2023).

In conclusion, the importance of automatic robotic arms lies in their transformative impact on industrial processes. They contribute to a safer, more efficient, and cost-effective manufacturing environment while positioning industries at the forefront of technological innovation and competitiveness. As automation technologies continue to advance, the role of robotic arms in shaping the future of manufacturing and other industries becomes increasingly pivotal.

1.3. Scope of the project

The scope of a project involving automatic robotic arms, specifically those controlled by an AVR microcontroller for pick-and-place operations, is broad and encompasses various dimensions. Here are key aspects to consider when defining the scope of such a project:

System Design and Architecture: All the specifications of the robotic arm, including its size, payload capacity, range of motion, and end-effector characteristics should be defined. Considerations for modularity and adaptability to different manufacturing environments should be explored (Jadeja, 2019).

We should specify the components of the control system, emphasizing the integration of the AVR microcontroller and outline the communication protocols between the microcontroller and peripheral devices, such as sensors and actuators.

Control algorithms and programming: We should detail the control algorithms governing the movements of the robotic arm. This involves kinematics and dynamics modeling, as well as the implementation of control strategies (e.g., PID controllers) on the AVR microcontroller (Sutyasadi, 2022). The programming language to be used for the AVR microcontroller should be specified. Common languages for AVR programming include C and assembly language.

Experimental validation: Quantitative performance metrics for the robotic arm, such as precision, speed, and payload capacity should be established. Develop a testing protocol to evaluate the system's capabilities under different conditions (J.A. Somolinos, 2002). Simulations and practical experiments that mimic real-world manufacturing scenarios should be considered. This includes diverse object shapes, sizes, and weights to validate the robotic arm's versatility.

Economic feasibility and integration: We should conduct a thorough economic analysis and consider both the initial investment and the long-term operational costs. The return on investment (ROI) and potential cost savings associated with automation should be evaluated (Abdel-Malek, 1987). The compatibility of the robotic arm with existing manufacturing setups should be addressed and strategies for seamless integration, minimizing downtime during implementation should be developed.

Documentation and reporting: All comprehensive technical documentations covering the design specifications, control algorithms, programming code should be created (Ilo, 2018).

We should generate a detailed project report that encompasses the entire development process, from conceptualization to implementation. Include insights gained from experimental validation, challenges encountered, and lessons learned.

Future directions: Finally, the potential for scaling up the project, whether in terms of incorporating multiple robotic arms or expanding the scope of automated tasks should be discussed (Pérez, Gutiérrez, & Zotovic, 2018).

Future enhancements, such as the integration of machine learning algorithms for adaptive control, enhanced sensor technologies, or collaboration with other automated systems should be considered.

Defining the scope with clarity ensures that the project remains focused and achievable within a defined timeframe. It also serves as a guide for project stakeholders, including developers, engineers, and decision-makers, to understand the project's objectives and deliverables.

1.4. Purpose of the robotic arm

This robot is a mechanical arm, a manipulator designed to perform many different tasks and capable of repeated, variable programming. To perform its assigned tasks, the robot moves parts, objects, tools, and special devices by means of programmed motions and points. The robotic arm performs motions in space. Its function is to transfer objects or tools from point to point, as instructed by the controller. The purpose of a robotic arm in industrial and manufacturing settings is multifaceted, encompassing a range of objectives that contribute to increased efficiency, precision, and safety. Here are several key purposes of a robotic arm:

Automation of repetitive tasks: Robotic arms excel at performing repetitive tasks with a high level of speed and precision. Automation of such tasks reduces the workload on human operators and enhances overall process efficiency (Dimeas, 2019).

High precision handling: Robotic arms are capable of executing movements with extreme precision, ensuring accurate positioning and manipulation of objects. This level of accuracy is essential in industries where even minor errors can have significant consequences (Jason D. McNulty, 2014).

Increased productivity: Unlike human workers who require breaks and rest, robotic arms can operate continuously, leading to increased productivity. This continuous operation is particularly valuable in industries with high-volume production requirements (M. B. Jamshidi, 2018).

Workforce safety: Robotic arms are well-suited for tasks involving hazardous materials or environments that pose risks to human safety. By taking on these tasks, robotic arms contribute to creating a safer work environment (Helander, 1990).

In environments with extreme temperatures, high radiation, or other dangerous conditions, robotic arms can operate with precision and consistency, mitigating the risks associated with human exposure.

Flexibility and adaptability: Robotic arms can be programmed and configured to perform a wide variety of tasks, making them versatile across different industries and adaptable to changing production requirements (al., 2011).

The ability to reprogram robotic arms for different tasks allows for quick changeovers in production processes, facilitating agility in responding to shifts in market demands.

Quality control: Robotic arms contribute to consistent product quality by minimizing variations in manufacturing processes. This is especially important in industries where product quality directly influences customer satisfaction and market competitiveness (Herakovic, 2010).

The precision and repeatability of robotic arm movements reduce the likelihood of defects in manufactured products, leading to lower rates of rework and waste.

Cost savings: While there is an initial investment in acquiring and setting up robotic arms, the long-term cost savings in terms of reduced labor costs, increased production efficiency, and minimized errors contribute to overall cost-effectiveness (James Pierce, Keith Needham, & Christopher Adams, 2020).

Robotic arms contribute to operational efficiency by streamlining processes, reducing cycle times, and optimizing resource utilization.

Competitive advantage: Companies that incorporate robotic arms into their manufacturing processes gain a competitive edge by demonstrating innovation and modernization. This can attract customers, improve market positioning, and enhance the company's reputation within the industry (Mel Wolfgang, 2017).

In summary, the purpose of a robotic arm is to serve as a versatile and efficient tool in industrial and manufacturing contexts, addressing challenges associated with repetitive tasks, precision, safety, adaptability, and overall process optimization.

1.5. Problem Statement

Robotic arms are of particular importance as they aid or replace humans in difficult possibly dangerous extravehicular activities. However, robot intelligence and autonomy are still limited. Therefore, robots need to be supervised or directly teleported in order to accomplish complex tasks in diverse environments (Mr.M.V.N.R.P.Kumar, 2015).

This ‘‘pick and place’’ robot being designed to ease the sorting process of heavy materials. Usually the transfer process of the heavy materials is being carried out using man power and if the transfer process is repeated for a period of time, it can cause injuries to the operator. By using this particular robot, the operator will no longer have to bent and lift up heavy loads thus preventing injuries and increasing the efficiency of the work. Operator will make mistakes whether small or big everyone in a while .In the industrial world, the industry cannot afford to take kind of mistakes. Every mistake is costly whether in time, money, and material. In manufacturing industry and nuclear industry, a large fraction of the work is repetitive and judicious application of automation will most certainly result in optimum utilization of machine and manpower (Billard, 2013). A pneumatic `Pick and Place' Robot has been developed to achieve automation in applications where great sophistication is not needed and simple tasks like picking up of small parts at one location and placing them at another location can be done with great ease.

In contemporary manufacturing environments, the demand for increased efficiency, precision, and automation has become paramount. Traditional manual pick-and-place operations are often associated with limitations in speed, consistency, and adaptability, leading to challenges in meeting the dynamic demands of modern production facilities. This project aims to address these limitations by developing an automatic pick-and-place robotic arm controlled by an AVR microcontroller, providing a solution that enhances productivity, accuracy, and versatility in manufacturing processes.

Manual Labor Limitations: Manual pick-and-place operations are inherently limited by human factors, such as fatigue, speed, and consistency. This results in suboptimal performance and can be a bottleneck in high-throughput manufacturing environments (AUDE BILLARD, 2019).

Precision and consistency: Achieving a high level of precision and consistency in pick-and-place tasks is challenging with manual operations. This is particularly crucial in industries where tight tolerances and quality standards must be maintained (Eljamel, 2006).

Adaptability to varied tasks: Traditional manufacturing setups often struggle with the adaptability required to handle a diverse range of products, each with unique shapes, sizes, and handling requirements. This lack of adaptability can hinder the overall efficiency of the production line (Andriella, 2020).

Labor-intensive processes: Manual pick-and-place processes are labor-intensive, leading to increased operational costs, particularly in industries where repetitive tasks are prevalent. Automating these processes can significantly reduce labor costs and improve overall cost-effectiveness (R. Lankin, 2020).

Real-time responsiveness: Manual operations may struggle to respond in real-time to changes in production requirements, leading to delays and inefficiencies. An automated system with an AVR microcontroller can provide the necessary responsiveness and adaptability to dynamic manufacturing needs (Hauser, 2012).

1.6. Function of the robotic arm

The scope of the automatic pick-and-place robotic arm, centered on the integration of an AVR microcontroller, is expansive and encompasses various dimensions, ranging from technical aspects to economic feasibility. The project's scope is defined by the overarching goal of revolutionizing traditional manufacturing processes by introducing a versatile, automated solution. Here is a detailed breakdown of the scope.

Mechanical design and kinematics: A robust mechanical design for the robotic arm, considering factors such as size, payload capacity, and reach should be designed (Ç. Ersin, 2020). Kinematic models to ensure precise and coordinated movements of the robotic arm, incorporating mathematical models to achieve optimal performance should be implemented.

Control system development: The AVR microcontroller such as Attiny or Atmega models for implementing control algorithms that govern the motion and actions of the robotic arm should be utilized (Kumar, 2017). Advanced control strategies, such as PID controllers and adaptive algorithms should be explored to enhance real-time responsiveness and adaptability.

Programming and software development: Software interfaces for programming and controlling the robotic arm, ensuring user-friendly interaction should be developed (K. Kruthika, 2016). The software to facilitate easy reprogramming for different tasks and quick adaptability to changing production requirements should be optimized.

Simulation and modeling: Simulation tools to model and simulate the behavior of the robotic arm in various scenarios should be utilized (GORGHIU, 2009). Virtual testing to validate the performance of the control algorithms and refine the design before physical implementation should be conducted.

Hardware implementation: The mechanical components, control electronics should be assembled and integrated to create a functional prototype of the automatic pick-and-place robotic arm (Qaryouti, 2019). Iterative testing and refinement to ensure seamless hardware integration and reliable operation should be conducted.

Experimental validation: The robotic arm should be tested rigorously, under diverse conditions, including variations in payload, environmental factors, and object shapes (M. Taylan Das, 2005). Data on key performance metrics, such as precision, speed, and reliability should be collected to validate the effectiveness of the automated system.

Economic feasibility analysis: A comprehensive economic analysis to assess the financial viability of implementing the automated system should be performed (C. Blanes, 2011). The potential cost savings in labor, improvements in production efficiency, and the return on investment over time should be evaluated.

Integration into existing manufacturing setups: Guidelines and protocols for the seamless integration of the robotic arm should be developing into existing manufacturing processes. Downtime during implementation should be minimized and support for adapting the system should be provided to diverse production environments (Al Mamun, 2023).

Documentation and knowledge transfer: Detailed technical documentation covering the design specifications, control algorithms, and implementation details should be created (Andrew

Lobbezoo, 2021). Knowledge transfer by providing documentation that serves as a resource should be facilitated for future maintenance, upgrades, and modifications.

Scalability and future enhancements: The scalability of the robotic arm system, considering the potential should be explored for incorporating multiple robotic arms into a coordinated system (Rawat, 2016). Avenues for future enhancements should be identified such as the integration of advanced control algorithms, additional sensors, or machine learning capabilities.

Ethical considerations and workforce impact: The ethical implications of introducing automation in the workforce should be evaluated considering potential job displacement and the need for reskilling and upskilling programs (Patrick Lin, 2012). Concerns related to job safety and human-robot collaboration should be addressed in manufacturing environments.

In conclusion, the scope of the automatic pick-and-place robotic arm project with an AVR microcontroller is comprehensive, covering technical intricacies, economic considerations, and ethical dimensions. The project aims to not only develop a cutting-edge automated solution but also contribute valuable insights to the ongoing discourse on the role of robotics in modern manufacturing.

CHAPTER II. MATERIAL AND METHODICS OF THE RESEARCH

2.1. Practical part and main components of the robot

In the practical part of the project we are talking about main component of the robot such as giving deep information about their appearance and working principle. Generally, the working principle of the robotic arm consists of only two steps such as, firstly, we place each payload cubes on their own coordinates and secondly, we turn on the robot by plugging in the power supply in a energy socket. Finally, the robot starts to operate automatically and take every payloads from their coordinates and place them on destinated coordinates (Sharath Surati, 2021). Let's take a look every main component of the robotic arm.

This robotic arm is based on Atmega328p microcontroller and consists of 4 pieces of SG90 servo motors. As using 4 pieces of Servo motors it results 3 DOF (Degrees of Freedom) and easy to move. The robotic arm does all tasks automatically. It uses inverse kinematics to calculate the degrees of servo motors by given coordinates on its system. With giving certain coordinates to the robot, we make it moving the first destination and grabbing the cubes, then moving second destination and putting its cubes top of each other. The platform is made from wood and cardboard by sticking with each other. Cubes are also made from cardboard by sticking with glue. The robotic arm can easily grab and carry these cubes. Here are the hardware appearances of the platform, cubes and robotic arm (Figure 2.1 & Figure 2.2):

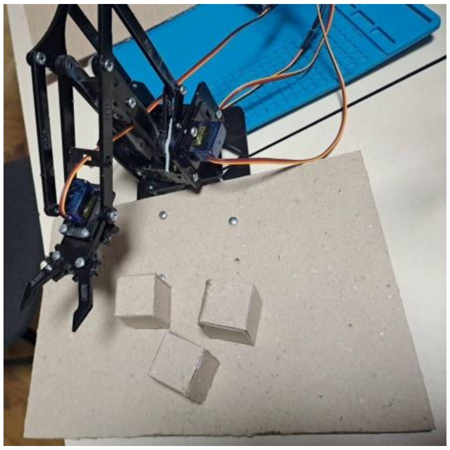


Figure 2.1. Left side of the robot

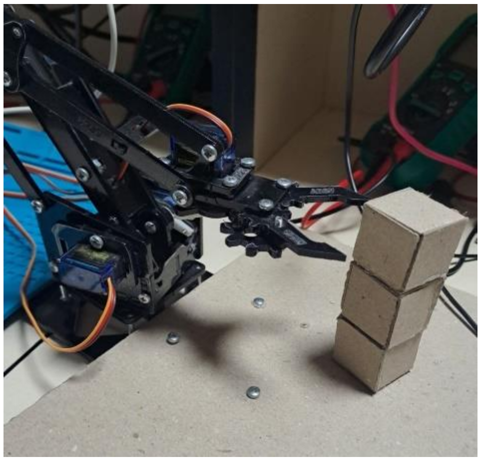


Figure 2.2. Right side of the robot

The work of industrial Pick and Place robotic arms can be vary for its aims for their own specific aims. For our robotic arm, the main goal is to describe and show that how industrial robots take coordinates of payloads, grab them and carry to specific coordinates. We use inverse kinematics formulas in mathematical side of our project (Serdar Kucuk, 2006). Firstly, we give the instant coordinates payloads to the main system of robot, secondly we give target coordinates of payloads which we want the robot carry them on. And it automatically calculates all angles of motors and grabs them after each other then place on target coordinates. We can see below (Figure 2.3) a photo of working of the robotic arm.

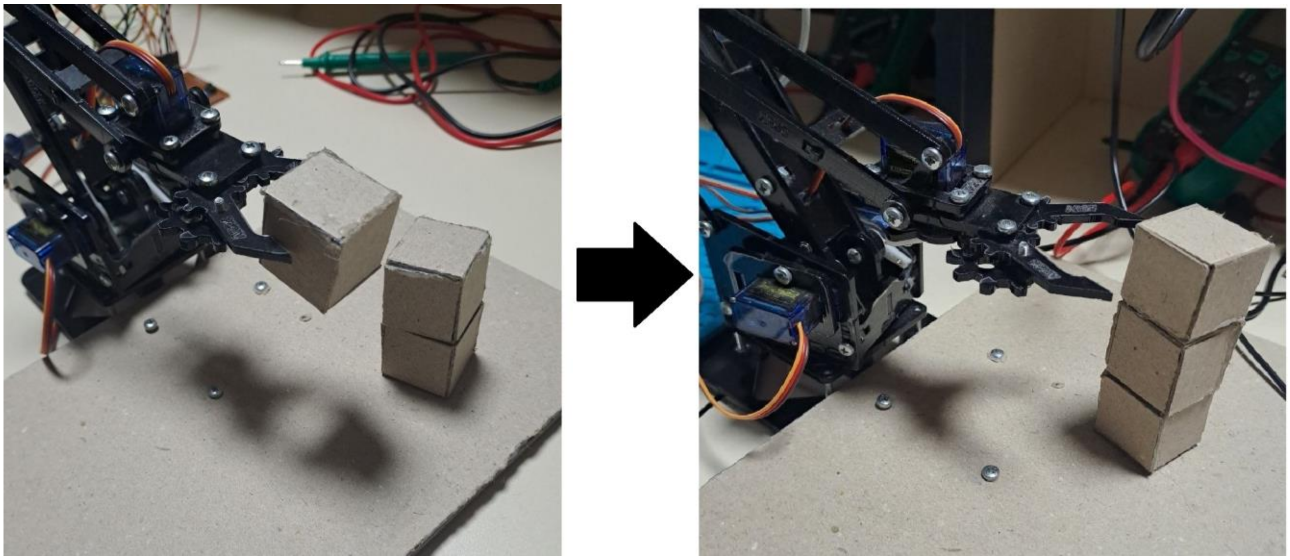


Figure 2.3. The robot appearance with payload and without payload

Totally the robotic arm consists of 6 main parts:

1) A platform (Wooden surface)
2) A payload (Three cubes)
3) Frame and Mechanical Structure (Plastic body)
4) Four Servo motors(SG90)
5) The Atmega328p AVR microcontroller (Arduino)
6) A power supply (5V/3A and 18W)

In conclusion, the ATmega328P-based robotic arm with 3 degrees of freedom (DOF) and 4xSG90 servo motors represents a versatile and accessible platform for academics and enthusiasts interested in robotics. The ATmega328P microcontroller serves as the central intelligence, orchestrating the precise control of the servo motors to achieve coordinated and programmable movements (ARM, 2020).

The hardware components, including the SG90 servo motors, power supply and the mechanical structure, collectively contribute to the overall functionality of the robotic arm. The use of PWM signals to control the servo motors allows for accurate positioning of each joint, enabling the creation of various motions and maneuvers.

For mathematical side of this robot we use inverse kinematics which we will talk about it later chapters. Inverse kinematics enhances the robotic arm's capabilities by providing feedback and simplifying the control process, respectively. This flexibility allows users to adapt the robotic arm to different applications and tasks.

As with any robotic system, proper testing and calibration are crucial to ensure reliable and accurate performance. Careful consideration of the power supply and electrical connections is necessary to prevent potential issues and ensure the longevity of the components.

Overall, the ATmega328P-based robotic arm serves as an educational, innovational and practical platform for individuals looking to explore and learn the fundamentals of robotics, programming, and mechatronics. Its scalability and open-source nature make it an excellent starting point for those interested in further customization and expansion of robotic systems. Let’s take a look at each main parts of the robotic arm individually.

2.1.1. A platform

The dimensions of the width and length of a robotic arm's platform are critical considerations in its design, influencing its stability, reach, and workspace. The specific dimensions can vary widely depending on the intended application, payload capacity, and operational requirements. There are some key factors to consider when determining the width and length of a robotic arm's platform:

Stability - a wider base generally contributes to greater stability, reducing the risk of tipping during movements. The width of the platform should be designed to provide adequate stability for the robotic arm, especially when carrying heavier payloads or executing dynamic motions.

Workspace and reach - the length of the platform affects the overall reach of the robotic arm. A longer platform allows the arm to access a larger workspace, enabling it to manipulate objects over a more extensive area. We should consider the operational requirements and the specific tasks the robotic arm will perform when determining the ideal length.

Payload capacity - the width and length of the platform are directly linked to the robotic arm's payload capacity. A larger platform can accommodate larger and heavier payloads, providing more versatility in terms of the objects the arm can manipulate. Ensure that the platform dimensions align with the intended payload capacity.

Operational environment - consider the physical constraints of the environment in which the robotic arm will operate. If space is limited, a more compact platform may be necessary to navigate confined areas. On the other hand, larger platforms may be suitable for applications where a broad range of motion is required.

Dynamic movements - if the robotic arm is designed to execute dynamic movements or maneuvers, such as rapid pick-and-place actions, the platform's dimensions should be configured to support these motions without compromising stability. The width and length play a role in the arm's ability to maintain balance during dynamic operations.

Integration with workspace - ensure that the dimensions of the platform align with the workspace in which the robotic arm will be deployed. This is crucial for applications where the arm needs to interact with specific machinery, workstations, or tools.

Compactness and maneuverability - in some applications, a more compact and maneuverable robotic arm may be preferred (A. Mazumdar, 2012). A smaller platform can be advantageous in situations where precise positioning and navigation are critical.

Aesthetic and ergonomic considerations - depending on the application and setting, aesthetic and ergonomic considerations may come into play. The dimensions of the platform should complement the overall design and functionality of the robotic arm, especially in environments where aesthetics matter.

Ultimately, the width and length of the robotic arm's platform should be tailored to meet the specific needs of the intended application, taking into account factors such as stability, reach, payload capacity, operational environment, and maneuverability. The optimal dimensions strike a balance between these considerations to ensure the robotic arm performs effectively and safely in its designated tasks.

In our project the platform has been designed from sticked wood and cardboard. Dimensions its width and length are accordingly 16.2 cm and 23.2 cm. Additionally, a dimension of its thickness is 0.8 mm. The thickness of a platform increases its payload capacity. The mass and dynamic acceleration of the robotic arm can cause the platform breaking or injuring. Therefore the platform in our project has been chosen by considering these probabilities and results. Below we can see the platform of our robotic arm (Figure 2.4 and 2.5).

In conclusion, the design and construction of the platform for a robotic arm represent a pivotal phase in the development of a functional and versatile mechatronic system. The platform serves as the backbone, providing structural integrity, support for crucial components, and a framework for precise and coordinated movement. As we delve into the myriad considerations surrounding the platform, it becomes evident that its design plays a central role in defining the capabilities and characteristics of the robotic arm.

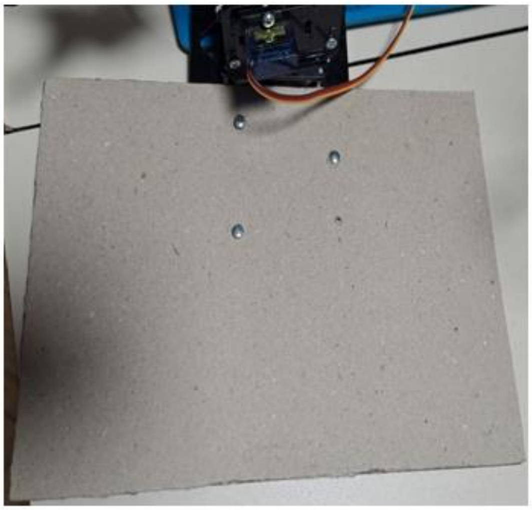


Figure 2.4. Top view of the platform

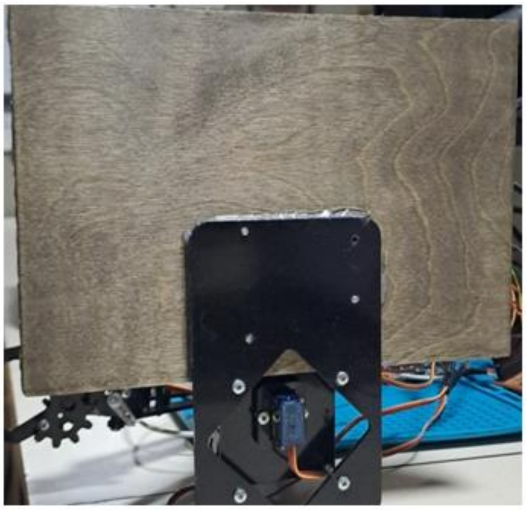


Figure 2.5. Bottom view of the platform

The choice of materials for the platform is a fundamental decision, impacting the overall strength, weight, and resilience of the robotic arm. Common materials such as aluminum, steel, wood and various plastics each bring their unique set of attributes, influencing factors such as cost, ease of manufacturing, and the arm's capacity to withstand operational stresses. A thoughtful consideration of these material properties is crucial for achieving the desired balance between durability and weight efficiency.

Mechanical design emerges as another critical facet, encompassing the arrangement of joints, links, and actuators on the platform. This design determines the degrees of freedom (DOFs) of the robotic arm, directly affecting its range of motion and versatility. In the context of a 3-DOF robotic arm, the design should harmonize these axes of movement to avoid interference while ensuring smooth and controlled articulation. Achieving this delicate balance demands meticulous attention to detail in the mechanical design.

End-effector mounting on the platform introduces another layer of complexity. The secure attachment of grippers, tools, or specialized end-effectors is paramount for the effective functioning of the robotic arm. The platform must offer a stable and rigid connection to the end-effector to guarantee precise control and accurate manipulation of objects or tools.

Compatibility with other hardware components, including the microcontroller, servo motors, power supply is a foundational consideration. The platform should be conceived with strategically placed mounting points, cable management solutions, and spaces for electronic components to ensure an organized and efficient integration of the various elements. This integration is crucial for the seamless communication and synchronized operation of the entire robotic system.

Ease of assembly and maintenance further underscores the importance of a well-designed platform. A modular and accessible design not only expedites the initial assembly process but also facilitates troubleshooting and maintenance tasks throughout the robotic arm's operational lifespan. This consideration is particularly significant for users seeking to experiment, modify, or upgrade their robotic arms over time.

Base stability, where the robotic arm connects to the supporting surface, emerges as a key determinant of overall stability during operation. The size and shape of the base contribute directly to the prevention of tipping or undesired movements, safeguarding the integrity of the robotic arm's maneuvers.

Lastly, the scalability of the platform opens up possibilities for future enhancements or modifications. A platform that accommodates expansion, be it additional DOFs, supplementary sensors, or upgraded components, provides users with the flexibility to adapt their robotic arms to evolving needs or applications.

In essence, the platform of a robotic arm is the canvas upon which the intricate dance of engineering, mechanics, and electronics unfolds. A well-conceived and executed platform lays the foundation for a robotic arm that is not merely a mechanical contraption but a dynamic and adaptable system, poised to navigate the complexities of diverse tasks and challenges. As the field of robotics continues to evolve, the platform remains a crucial terrain for innovation and exploration, offering endless possibilities for those who seek to push the boundaries of mechatronic engineering.

2.2.2. A payload

In the context of robotics and automation, the term "payload" refers to the maximum weight or mass that a robotic system, such as a robotic arm or robot, is designed to carry, manipulate, or support. Payload capacity is a crucial specification that directly impacts the robot's application capabilities, performance, and overall functionality. There are some key considerations related to the payload of a robotic system:

Payload Capacity - Payload capacity is the maximum weight that a robotic system can handle while maintaining its intended performance and safety (Lim TG, 1990). It is a critical specification that influences the design, selection of components, and overall capabilities of the robot.

End-Effector and Tool Weight - The weight of the end-effector or tool attached to the robot contributes to the overall payload. Different end-effectors, such as grippers, welding tools, or sensors, have varying weights, and these must be considered when determining the payload capacity (Yoram Koren, 1987).

Dynamic Loads - In certain applications, the payload may experience dynamic loads, such as acceleration, deceleration, or sudden changes in direction. Dynamic loads can impact the robot's performance and require additional considerations for control and stability.

Center of Gravity - Understanding the center of gravity of the payload is essential for maintaining the stability of the robotic system. Proper weight distribution and awareness of the center of gravity help prevent tipping or imbalance during movements (D. G. Chung, 2016).

Torque and Motor Power - Carrying a heavier payload requires more torque from the robot's motors to achieve the desired movements. The robot's motor power and torque specifications must be sufficient to handle the payload effectively (K. Bodie, 2016).

Material Handling and Manipulation - In industrial settings, robots are often used for material handling and manipulation tasks. Payload capacity is a crucial factor in determining the robot's suitability for lifting, transporting, or manipulating specific materials or products (Zain Ali, 2023).

Safety Considerations - Exceeding the specified payload capacity can compromise the safety and stability of the robotic system. It's important to adhere to recommended payload limits to prevent equipment damage, malfunctions, or accidents. Safety features, such as load monitoring systems, may be implemented to ensure safe operation (M. Hamad, 2019).

Application-Specific Requirements - The payload requirements vary based on the specific application. For example, in assembly tasks, a robot may handle lighter components, while in material handling tasks, it may need to lift and transport heavier loads. Understanding the application's demands is crucial in determining the appropriate payload capacity (Tzafestas, 1991).

Compliance and Force Control - Some robotic systems are designed with compliance or force control features, allowing them to interact with the environment or respond to external forces. Payload considerations play a role in implementing these features to ensure appropriate force sensing and response.

Understanding and specifying the payload capacity of a robotic system is essential for successful deployment in diverse applications. It ensures that the robot can effectively perform its tasks while maintaining stability, precision, and safety. Manufacturers provide payload specifications for their robotic systems, and these should be carefully considered when selecting or designing a robot for a particular application.

Payloads are three cubes in our project. They are made by sticked cardboard. Each of these cubes have sizes of 2.5x2.5x2.5 cm^3. These sizes allow the grabber of robotic arm to hold the payloads for a long time. These payloads are only for prototyping. When we give their coordinates into system, the system automatically calculates all angles of servo motors sequently. The advantage of using cardboard cubes as payloads is making the robot more fast and work easily for payload’s light weight. The main reason of using equal sided cubes is keeping the center of gravity stable. Using link system in our project increases the payload capacity of it. As all servo motor has 9g/cm torgue, the maximum torgue of the arm is equal approximately 15-18gram. The mass of our cardboard cubes are 3.7 grams (AVCalc LLC, 2024). As result, our robotic arm has absolute ability of grabbing, holding and carrying its payloads easily and performs tasks clearly. We can see a cube grabbed by robotic arm and not grabbed by robotic arm (Figure 2.6).

In conclusion, the concept of payload in robotics represents a pivotal determinant that profoundly influences the design, capabilities, and practical utility of robotic systems. Payload capacity is the epitome of a robotic system's strength, defining the maximum weight it can proficiently carry, manipulate, or support within its operational context. As we navigate the nuanced landscape of payload considerations, it becomes evident that this parameter is not merely a technical specification but a guiding principle that shapes the entire architecture and functionality of robotic platforms.

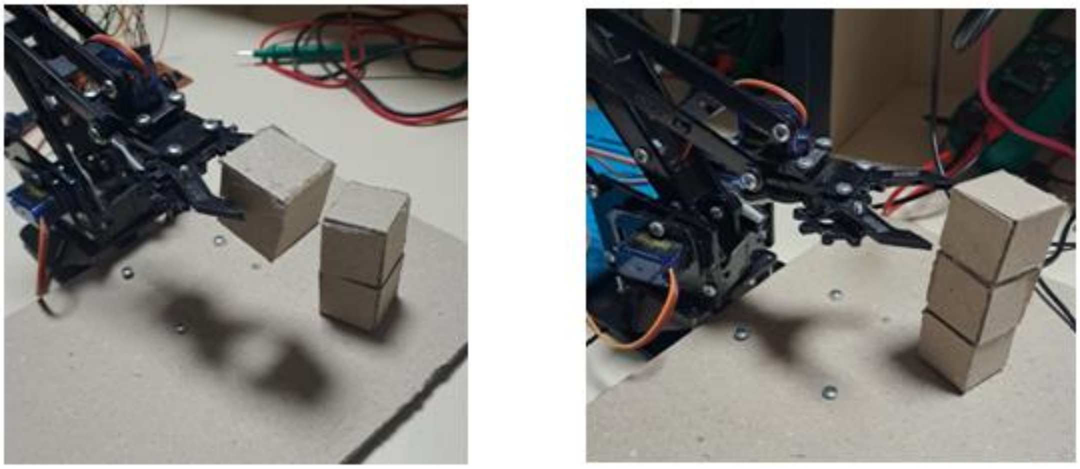


Figure 2.6. A cube grabbed mode and free mode of the robot

The payload capacity of a robot is akin to its physical prowess, setting the boundaries within which it can exert its mechanical prowess and fulfill its designated tasks. It serves as a fundamental metric, dictating the extent of the external forces a robot can endure and manage without compromising performance, stability, or safety. The implications of payload extend beyond mere weight-bearing, permeating every facet of robotic operation, from motor torque requirements to end-effector design and the intricacies of dynamic load management.

One of the critical factors influencing payload is the inherent nature of the application for which a robot is intended. Whether it's in the precision assembly of delicate components or the robust material handling of heavier loads, understanding the specific demands of the task at hand is paramount. The payload requirement becomes a blueprint, guiding engineers in selecting or designing a robotic system that aligns seamlessly with the intricacies and challenges of the application, ensuring optimal efficiency and reliability.

Dynamic loads introduce an additional layer of complexity, necessitating a holistic approach to payload considerations. The ability of a robot to navigate sudden accelerations, decelerations, or changes in direction without compromising stability or precision is directly tied to its payload handling capabilities. This dynamic interplay between the robot and its payload underscores the importance of a sophisticated control system and responsive actuators.

Moreover, the concept of payload transcends the mere lifting capacity of a robot; it extends to the strategic management of forces within its workspace. Understanding the center of gravity of the payload and ensuring proper weight distribution are integral to preventing unintended tilting or tipping, safeguarding both the robot and its surroundings. This awareness of spatial dynamics further underscores the sophistication required in the mechanical design and control algorithms of robotic systems.

In the pursuit of enhanced robotic capabilities, the relationship between payload and torque becomes a central consideration. The motors driving the robotic joints must deliver sufficient torque to not only support the static weight of the payload but also accommodate the dynamic forces associated with its movement. The synergy between payload and torque serves as the nexus of robotic performance, dictating the precision, speed, and efficiency with which tasks are executed.

Safety considerations loom large in the discourse on payload. Exceeding specified payload limits can compromise the integrity of the robotic system, potentially leading to malfunctions, equipment damage, or, in extreme cases, safety hazards. Adherence to recommended payload guidelines and the incorporation of safety features, such as load monitoring systems, are essential safeguards that ensure the secure and responsible deployment of robotic technology.

In the ever-evolving landscape of robotics, where applications span industries and the boundaries of innovation are constantly pushed, the understanding and meticulous management of payload remain at the forefront. As we continue to witness the integration of robotic systems into diverse sectors, from manufacturing and logistics to healthcare and beyond, the importance of payload considerations will only intensify. It is not merely a static parameter but a dynamic force shaping the evolution of robotics, steering it towards greater precision, adaptability, and safety in the face of complex and varied tasks. In essence, the payload is not just a weight limit; it is the threshold beyond which the future capabilities of robotics unfold.

2.1.3. Frame and mechanical structure

The frame and mechanical structure of a robotic arm constitute the physical framework upon which the entire system is built. This structure plays a crucial role in determining the arm's stability, strength, range of motion, and overall performance. There are some key aspects for choosing frame and mechanical structure of a robotic arm:

2.1.3.1. Material Selection

The choice of materials for the frame is a critical decision that directly influences the arm's strength, weight, and durability. Common materials can be Aluminum, Steel, plastic or

Composite materials. Aluminum materials have advantage of lightweight and corrosion-resistant, therefore they are often chosen for its balance between strength and weight (AlloyMuhammad Nor Aiman Abd Razak, 2023). Steel materials offers high strength but is heavier compared to aluminum. Some applications use composite materials to achieve a balance between strength, weight, and specific application requirements. Plastic materials can offer specific advantages and challenges. As we also use plastic structure, let’s take a look into some key points to consider when incorporating plastic into the design.

There are many advantages of plastic in Robotic Arm construction:

Weight - plastic materials are generally lightweight compared to metals like aluminum or steel. This can be advantageous in applications where minimizing the overall weight of the robotic arm is important, such as in mobile or collaborative robotic systems.

Cost-effectiveness - plastic materials often have a lower cost compared to metals. This can contribute to cost-effectiveness in the manufacturing and assembly of robotic arms, making them more accessible for certain applications.

Corrosion resistance - unlike metals, plastics are generally resistant to corrosion. This property is beneficial in environments where exposure to moisture or harsh chemicals may be a concern, as plastic materials are less likely to degrade over time.

Electrical insulation - plastic is an insulating material, which can be advantageous in situations where electrical components or wiring need to be isolated from the mechanical structure. This can contribute to enhanced safety and prevent electrical interference.

Design flexibility - plastic materials can be molded into various shapes and configurations, providing designers with greater flexibility in creating complex and intricate structures. This can be beneficial for achieving specific ergonomic or aesthetic requirements.

Introducing plastic materials into the construction of robotic arms introduces a set of challenges and considerations that engineers must carefully navigate. While plastics offer distinct advantages, such as lightweight characteristics and cost-effectiveness, they also present unique limitations that impact strength, durability, and overall performance and don’t match the strength and rigidity of metals. These limitations can impact the overall stability and load-bearing capacity of the robotic arm. Let’s have a look into them for clearness.

Temperature sensitivity - some plastic materials may have limitations in terms of temperature resistance. In high-temperature environments or applications that involve exposure to heat sources, the mechanical properties of certain plastics may be compromised.

Wear and tear - plastics may exhibit higher wear rates compared to metals, especially in applications with repetitive or high-impact movements. This could affect the longevity and durability of the robotic arm, requiring careful material selection and design considerations.

Precision machining - precision machining of plastics may require different techniques compared to metals. Some plastic materials may have lower tolerances for certain manufacturing processes, impacting the precision of the robotic arm's components.

Limited load capacity - the load-bearing capacity of a plastic-based robotic arm may be limited compared to metal counterparts. This can influence the types of applications for which the robotic arm is suitable.

In summary, the incorporation of plastic materials in the frame and mechanical structure of a robotic arm involves a trade-off between benefits such as weight reduction, cost-effectiveness, and specific challenges related to strength and durability. The choice of materials should be driven by the specific requirements of the application, taking into account factors such as load capacity, environmental conditions, and overall system performance.

2.1.4. Mechanical Design

The mechanical design of a robotic arm is a critical aspect that lays the foundation for its structural integrity, functionality, and overall performance. This intricate process involves the thoughtful consideration of various elements to ensure that the robotic arm meets the requirements of its intended application. The mechanical design includes the arrangement of joints, links, and other structural components. There are some considerations that are included in mechanical design. Here is an in-depth introduction to the mechanical design of robotic arms:

Number of degrees of freedom (DOF) - The mechanical structure is designed to provide the necessary degrees of freedom, allowing the robotic arm to move in specific ways. The number and arrangement of joints contribute to the overall DOF of the arm.

Joints and links - joints are points of articulation that enable movement. Common types include revolute joints (rotational) and prismatic joints (linear). Links are rigid segments connecting joints, forming the arm's structure. Links can vary in length and may be composed of multiple materials.

End-effector mounting - the end-effector, such as a gripper or tool, needs to be securely mounted to the robotic arm's structure. Proper mounting ensures stability and accuracy in manipulating objects.

Safety features and human-robot interaction - safety considerations and the design for human-robot interaction are paramount. The conclusion emphasizes that incorporating safety features, such as emergency stop mechanisms and collision detection systems, is crucial to prevent accidents and facilitate safe collaboration between robots and humans.

Precision machining and tolerance - precision machining and tolerance considerations are critical for achieving high levels of accuracy. The conclusion drawn is that the mechanical design process must account for the specific requirements of machining plastics or metals, ensuring that components fit together precisely and function seamlessly.

Base and support structure - the base provides the foundation for the robotic arm. A stable and well-designed base is essential for preventing tipping or excessive vibrations during operation. The support structure includes elements such as the arm's columns or pillars.

Compactness and reach - the mechanical structure should be designed to balance between compactness and reach. A compact design is beneficial in confined spaces, while an optimized reach allows the arm to cover a larger workspace.

Ease of assembly and maintenance - a modular and accessible design simplifies the assembly process and facilitates maintenance. Easy access to components ensures that troubleshooting and repairs can be carried out efficiently.

Scalability - scalability allows for modifications or upgrades to the robotic arm. A scalable design can accommodate additional joints, links, or other components as needed, enabling customization for evolving requirements.

Aesthetic and ergonomic considerations - depending on the application and environment, aesthetic and ergonomic factors may be important. In collaborative settings, for instance, an ergonomic design ensures safe interaction with human operators.

Kinematics and inverse kinematics - kinematics defines how the joints and links move in relation to each other. Inverse kinematics, a critical aspect of the mechanical design, involves calculating the joint angles required to position the end-effector at a specific location. This contributes to the arm's precision and accuracy.

Drive mechanisms - the mechanical design incorporates drive mechanisms to actuate the joints. This may involve electric motors, pneumatic systems, or hydraulics, depending on the application requirements. The choice of drive mechanisms influences factors such as speed, torque, and energy efficiency.

Weight distribution - distributing the weight of the robotic arm evenly is crucial for maintaining balance and stability. Proper weight distribution contributes to smooth and controlled movements.

Manufacturability and cost - considerations for manufacturability and cost are integral to the mechanical design process. Design choices should align with practical manufacturing methods, and material selections should balance performance with cost-effectiveness.

The mechanical design of our robotic arm involves 3 DOF, 3 Joints and links and stable end effector grabber. All these well designed mechanical parts of the robot give us its structural integrity, functionality and high overall performance.

3 Degrees of freedom is allowing the robotic arm to move in specific ways. The number and arrangement of joints contribute to the overall DOF of the arm. It is also eligible for power consumption by motors for performing tasks more clearly. When we choose Degrees of Freedom of the robotic arm, we should also consider dimensions of length of links. They should be suitable for each other to decrease power consumption and increase precision, speed and overall performance.

Our robotic arm uses three joints and links which are points of articulation that enable movement for grabbing, moving payloads and perfoming its task. The type of its joints is revolute joints (rotational) which allows to rotate up to 90 degrees. Length of the link connected to shoulder joint is 8.5 cm. Length of the link connected elbow joint is 8 cm.

We use a gripper as an end-effector which securely mounted to the robotic arm's structure. Proper mounting ensures stability and accuracy in manipulating objects. Opened gripper has 4cm space while cubes have 2.5cm of side sizes. It allows to grab payload objects easily and move them to the target coordinates.

Drive mechanisms consist of 4 servo motors which allows for precise control of multiple joints or components. Each servo motors have 9g/cm of torgue. First servo motor controls the base of frame. Second servo motor controls shoulder joint of the arm. Third servo motor controls elbow joint of the robotic arm. Fourth servo motor controls the gripper to grab objects. Here is size dimensions overview of mechanical structure of the robotic arm: (Figure 2.7)

The frame and mechanical structure of a robotic arm represent the backbone of the entire system. A well-designed structure not only provides the necessary support for the arm's components but also determines its overall capabilities and adaptability to different tasks and environments. The mechanical design is an integral part of the robotic arm's functionality, influencing its performance, efficiency, and safety.

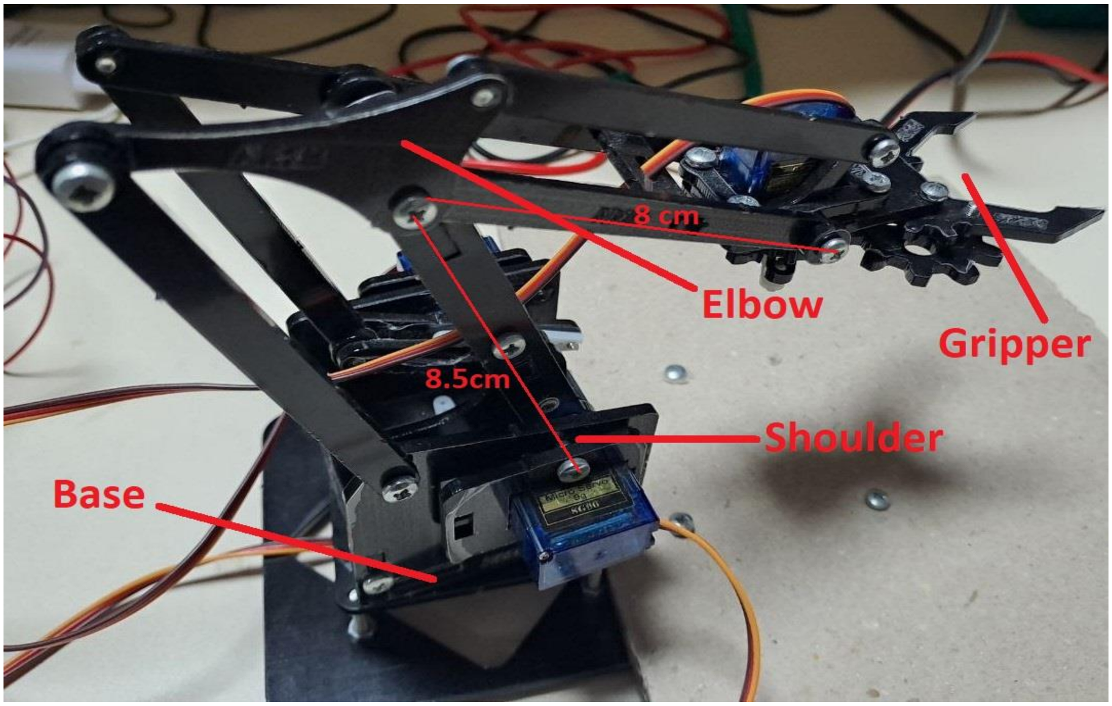


Figure 2.7. Dimensions of the robot

The mechanical design of a robotic arm is a multidimensional process that involves a delicate interplay of engineering principles, material science, and application-specific requirements. It lays the groundwork for the arm's functionality, precision, and adaptability, making it a pivotal stage in the overall development of robotic systems. A well-executed mechanical design not only ensures the structural robustness of the robotic arm but also determines its capacity to excel in diverse and demanding operational scenarios.

In conclusion, the frame and mechanical structure of a robotic arm represent the foundational elements that determine its overall performance, stability, and adaptability. The intricacies of the mechanical design process and material selection significantly impact the arm's functionality across various applications.

2.1.5. Four servo motors (SG90)

The SG90 is a popular and widely used micro servo motor in the robotics and electronics fields. Let’s have some information about the SG90 servo motor (Pro, 2014):

Size and weight - the SG90 is a small-sized servo motor, commonly referred to as a "micro" servo. Its dimensions are approximately 23mm x 12.2mm x 29mm. It is relatively lightweight, making it suitable for various applications.

Torque and speed - the SG90 typically has a torque rating around 1.5 to 2.5 kg/cm. The speed of the SG90 is usually in the range of 0.1 to 0.12 seconds per 60 degrees.

Voltage and power - the SG90 servo motor is designed to operate at 4.8V to 6V. It is powered by a direct current (DC) supply.

Construction - the SG90 motor is a 3-wire device with power (VCC), ground (GND), and control (PWM) wires. It often comes with plastic gears and a plastic case. (Figure 2.8)

Control signal - the SG90 servo motor is controlled using pulse width modulation (PWM). A 50 Hz PWM signal is commonly used, with a pulse width ranging from approximately 1 ms to 2 ms.

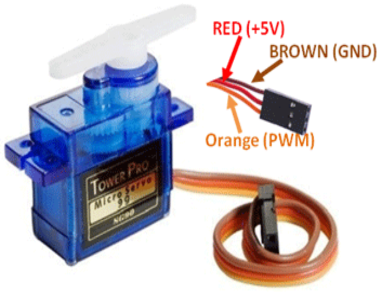


Figure 2.8 Servo motor

Applications - due to its small size and reasonable performance, the SG90 servo motor is frequently used in hobbyist projects, robotics, remote-controlled vehicles, and other applications where precise control of small mechanical movements is required.

Limitations - the SG90 is not suitable for heavy-duty applications due to its limited torque. It may exhibit some backlash, which is the amount by which the output shaft can rotate without corresponding input.

Additionally, when using an SG90 servo motor, it's important to check its specifications and datasheet for precise details. Additionally, be aware of the voltage and current requirements to ensure proper operation and prevent damage. It's a versatile and affordable choice for many projects, especially those involving small-scale robotics and electronics.

Utilizing four servo motors in the mechanical design of a robotic system, such as a simple robotic arm, allows for precise control of multiple joints or components. Servo motors are widely used in robotics due to their ability to provide accurate angular control and position feedback. Here's an overview of how four servo motors might be employed in a robotic system:

Joint Rotation and Degrees of Freedom (DOF) - Each servo motor can be dedicated to controlling the rotation of a specific joint, contributing to the overall degrees of freedom of the robotic system. For example, in a robotic arm, these servo motors might be assigned to control the movements of the shoulder, elbow, base, and gripper.

Shoulder Joint Servo Motor - The servo motor controlling the shoulder joint allows the robotic arm to move in an up-and-down or side-to-side motion, contributing to the overall reach and positioning of the arm.

Elbow Joint Servo Motor - The servo motor assigned to the elbow joint controls the bending and straightening of the arm. This movement is crucial for reaching objects at different heights or adjusting the arm's orientation.

Base Joint Servo Motor - The servo motor associated with the base joint allows the robotic arm to rotate its frame, facilitating additional flexibility in positioning and manipulation. This joint allows the arm to rotate horizontally, providing a way to position the entire arm in different directions. The choice of a servo motor for the base joint is important to achieve the desired range of motion and precision.

Gripper or End-Effector Servo Motor - The fourth servo motor can be dedicated to controlling the gripper or end-effector. Depending on the design, this servo motor enables the opening and closing of the gripper, allowing the robotic arm to grasp, hold, and release objects.

Coordination for Smooth Movements - Coordinating the movements of these four servo motors ensures smooth and accurate control of the robotic arm. The integration of a microcontroller or a dedicated control system facilitates the synchronized operation of the servo motors.

Feedback Systems for Precision - Each servo motor typically incorporates feedback systems, such as encoders, to provide precise position information. This feedback allows the control system to adjust and maintain the desired positions, enhancing the overall precision of the robotic arm.

Programming and Control - Programming the servo motors involves defining specific angles, speeds, and sequences of movements. This flexibility in control enables the robotic arm to perform a variety of tasks based on the programmed instructions.

Power and Energy Efficiency - Servo motors are generally known for their energy efficiency, and the use of four motors allows for optimized power consumption. This is especially beneficial in scenarios where battery-powered operation or energy conservation is a consideration.

In conclusion, the incorporation of four servo motors in the mechanical design of a robotic system, such as a robotic arm, offers precise control over multiple joints or components. The coordinated movement of these motors, along with feedback systems and programming capabilities, enhances the versatility and functionality of the robotic arm across various applications. This modular and scalable approach allows for adaptability and customization, making it suitable for a wide range of tasks in diverse environments.

2.1.6. The Atmega 328p AVR microcontroller (Arduino)

The ATmega328P microcontroller is a key component in many electronic devices and embedded systems. It is widely used for its versatility, low power consumption, and extensive features. Let’s know some main features of the ATmega328P microcontroller (Atmel, ATmega328P Datasheet, 2015).

Microcontroller Architecture - The ATmega328P is based on the Modified Harvard Architecture, featuring 8-bit RISC (Reduced Instruction Set Computing) CPU. It combines a rich instruction set with low power consumption, making it suitable for a wide range of applications.

Memory - It has 32 KB of Flash memory for program storage, which allows for the storage of the firmware or code that runs on the microcontroller. The microcontroller also includes 1 KB of EEPROM (Electrically Erasable Programmable Read-Only Memory) for non-volatile data storage. Additionally, it has 2 KB of SRAM (Static Random Access Memory) for temporary data storage during program execution. With 32 KB of Flash memory, 1 KB of EEPROM, and 2 KB of SRAM, the microcontroller provides ample resources for storing program code, non-volatile data, and temporary variables. This makes it suitable for projects with diverse memory requirements.

Peripheral Features - The ATmega328P comes with a variety of built-in peripherals, including timers/counters, USART (Universal Synchronous and Asynchronous Receiver and Transmitter), SPI (Serial Peripheral Interface), and I2C (Inter-Integrated Circuit) communication interfaces.It has multiple GPIO (General-Purpose Input/Output) pins, enabling digital and analog input/output capabilities. The presence of timers/counters, communication interfaces (UART, SPI, I2C), and an ADC enhances its capability to interface with sensors, actuators, and other devices. These peripherals contribute to the microcontroller's adaptability in various applications.

Clock Speed - The microcontroller typically operates at a clock speed of 16 MHz. It has also 8 MHz internal oscillator. It can be configured to run at lower speeds to reduce power consumption in battery-powered applications. Operating at a clock speed of 16 MHz, the ATmega328P provides a balance between performance and power consumption. Its ability to operate at lower clock speeds makes it energy-efficient, a crucial feature for battery-powered applications.

Analog-to-Digital Converter (ADC) - The ATmega328P features a 10-bit ADC, allowing the microcontroller to convert analog signals into digital values. This is crucial for interfacing with analog sensors or reading analog signals from the environment. The inclusion of a 10-bit ADC facilitates the conversion of analog signals to digital, enabling the microcontroller to interface with a wide range of analog sensors and respond to real-world conditions.

Communication Protocols - It supports popular communication protocols such as UART (Universal Asynchronous Receiver/Transmitter), SPI, and I2C, making it suitable for communication with other devices, sensors, or peripherals. Support for popular communication protocols makes the ATmega328P well-suited for projects requiring interaction with other devices or communication across a network. Its compatibility with UART, SPI, and I2C simplifies connectivity.

Low Power Consumption - The microcontroller has power-saving modes and features that enable it to operate in low-power scenarios, making it ideal for battery-powered applications or systems where energy efficiency is crucial. The microcontroller's low-power modes enhance its suitability for battery-operated systems, where power efficiency is critical. These modes contribute to extending the battery life in applications with stringent power constraints.

Bootloader Support - The ATmega328P can be programmed using various programming methods, including in-circuit programming. It supports the use of a bootloader, allowing firmware updates to be loaded via serial communication.

Development Environment - It is commonly used with the Arduino development platform, and there are various development tools and libraries available to simplify the programming process. The ATmega328P has gained widespread adoption, particularly in the Arduino ecosystem. This compatibility simplifies development, making it accessible to a broad audience, including hobbyists, students, and professionals.

Applications - The ATmega328P is widely used in various applications, including robotics, home automation, consumer electronics, and educational projects. Its popularity in the Arduino ecosystem has contributed to its widespread adoption in the maker and hobbyist communities. The availability of development tools, resources, and a supportive community makes the ATmega328P an excellent choice for educational purposes. It serves as a stepping stone for learning embedded systems and microcontroller programming.

The ATmega328P microcontroller's combination of features, performance, and ease of use makes it a popular choice for a broad range of embedded systems and DIY projects. Its flexibility and extensive capabilities contribute to its enduring popularity in the world of microcontroller-based development.

The ATmega328P's 8-bit RISC architecture, ample program memory, and various built-in peripherals contribute to its versatility. It can cater to a wide array of applications, ranging from simple DIY projects to more complex embedded systems.

In essence, the ATmega328P microcontroller remains a cornerstone in the world of embedded systems, offering a balance of features, ease of use, and widespread support. Its enduring popularity showcases its adaptability to a myriad of applications, making it a go-to choice for engineers, hobbyists, and educators alike.

In conclusion, the ATmega328P microcontroller stands as a versatile and widely utilized component in the realm of embedded systems and electronics. Its significance lies in its adaptable architecture, robust feature set, and broad applicability across diverse projects.

2.1.6.1. The Arduino

Arduino refers to both a hardware platform and a software ecosystem designed for simplified and accessible microcontroller programming and development. It has gained immense popularity in the world of electronics, maker communities, and education due to its user-friendly approach. Here's an overview of Arduino:

Arduino Microcontroller Boards - Arduino boards are equipped with microcontrollers, with the ATmega series being one of the most common. Examples include Arduino Uno, Arduino Mega, and Arduino Nano.

Variety of Shields - Arduino supports a range of shields—add-on boards that provide extra functionalities. Shields can include sensors, communication modules (Wi-Fi, Bluetooth), motor controllers, and more.

Open-Source Design - Arduino's hardware design is open-source, allowing users to study, modify, and create their own versions of the boards.

Integrated Development Environment (IDE) - The Arduino IDE is a user-friendly software environment for writing, compiling, and uploading code to Arduino boards. It supports the C++ programming language.

Arduino libraries simplify complex tasks by providing pre-written code for various functions. Users can easily incorporate these libraries into their projects.

Open-Source Software: The Arduino IDE is open-source, and its source code is available for modification and improvement.

Programming Language - Arduino programming is based on a simplified version of C++, making it accessible even to beginners. The simplified syntax and extensive libraries ease the learning curve.

Community and Collaboration - The Arduino community is vast and active. Users can find support, share projects, and seek advice on forums and online platforms. The open-source nature of Arduino encourages collaboration, allowing users to share and build upon each other's work.

Rapid Prototyping - Arduino facilitates rapid prototyping, allowing users to quickly implement and test ideas. Its plug-and-play nature, combined with a variety of available shields, accelerates the development process.

Extensibility and Customization - Arduino libraries cover a wide range of functions, from basic input/output to complex sensor interfaces. Users can also create and share their libraries. Users can modify Arduino boards and design custom shields to tailor the platform to their specific project requirements.

Commercial and Industrial Applications - Arduino is often used for prototyping in commercial and industrial settings. Once a prototype is validated, the design can be scaled up to custom-designed boards.

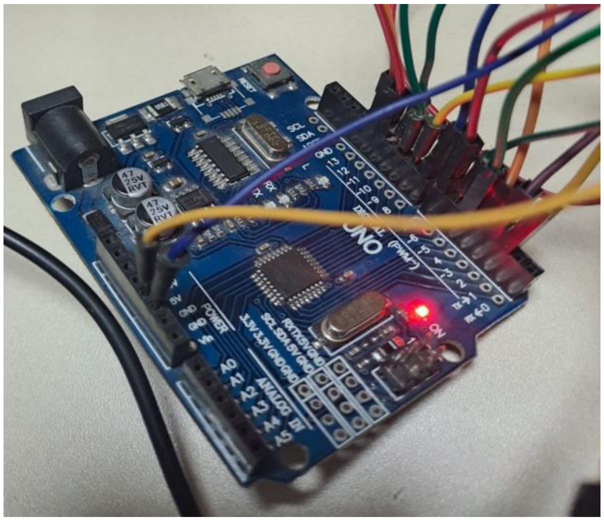


Figure 2.9. The Arduino UNO

In our project we use Arduino Uno board which has great and suitable features for controlling the servo motors of the robotic arm. Let's have a look at some main characteristics of Arduino Uno (Figure 2.9). The Arduino Uno is a popular microcontroller board that is widely used in the maker and hobbyist communities. Here are some main characteristics of the Arduino Uno:

Microcontroller - The Arduino Uno is based on the Atmel ATmega328P microcontroller.

Clock Speed - The ATmega328P on the Arduino Uno runs at a clock speed of 16 MHz.

Flash Memory - It has 32 KB of flash memory, where the user can upload their program (Arduino sketch).

SRAM - Arduino Uno has 2 KB of SRAM (Static Random Access Memory) for variables and runtime data storage.

EEPROM - There is 1 KB of EEPROM (Electrically Erasable Programmable Read-Only Memory) for non-volatile storage of data.

Digital I/O Pins - The board has 14 digital input/output pins, where some can be used for PWM (Pulse Width Modulation) output.

Analog Input Pins - The Arduino Uno has 6 analog input pins, allowing for analog sensor connections.

Voltage Regulator - It includes a voltage regulator that allows it to be powered with an external power supply or through the USB connection.

USB Interface - The board has a USB interface for programming and serial communication with a computer.

Power Supply - The Arduino Uno can be powered using an external DC power supply (7-12V) or through the USB connection.

Operating Voltage - The operating voltage of the Arduino Uno is 5V.

Clock Source - The ATmega328P can be clocked from an external crystal or resonator.

Reset Button - A reset button allows you to restart the microcontroller.

LEDs - The board includes a built-in LED connected to digital pin 13, commonly used for simple testing and debugging.

We can see all these components in arduino board with explanation(Figure 2.10) and its principal circuit diagram(Figure 2.11).

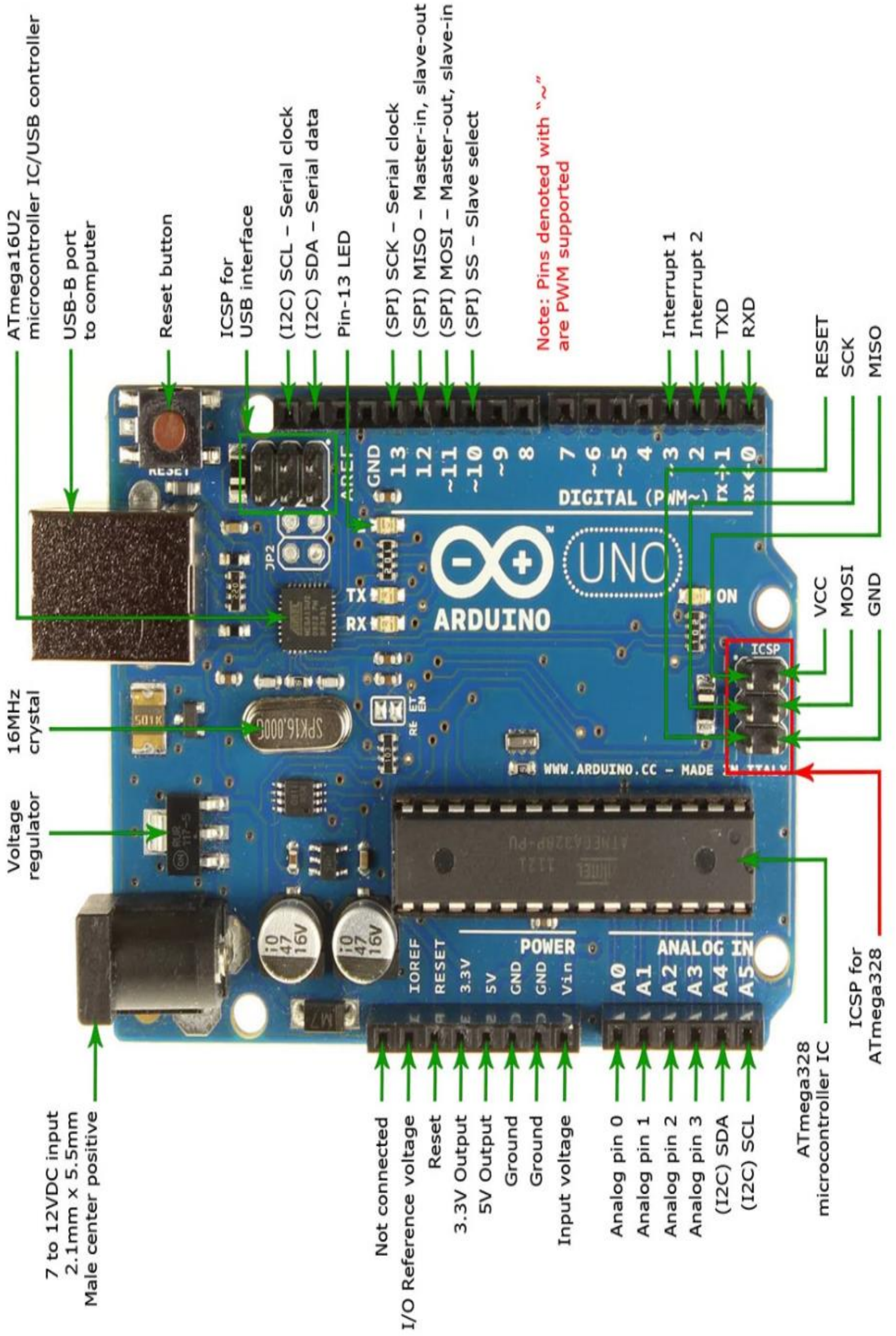


Figure 2.10. Explanation of the Arduino UNO board

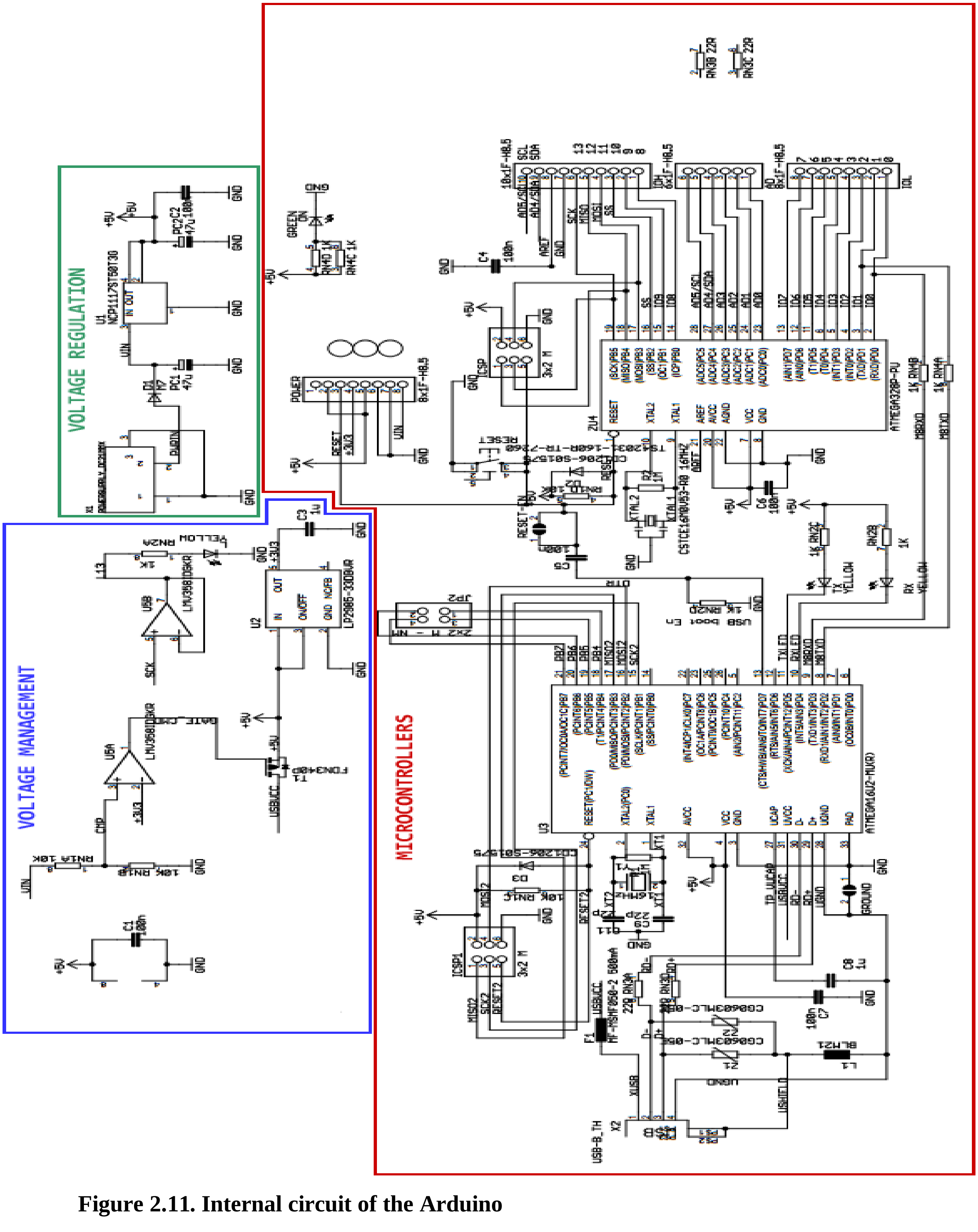


Figure 2.11. Internal circuit of the Arduino

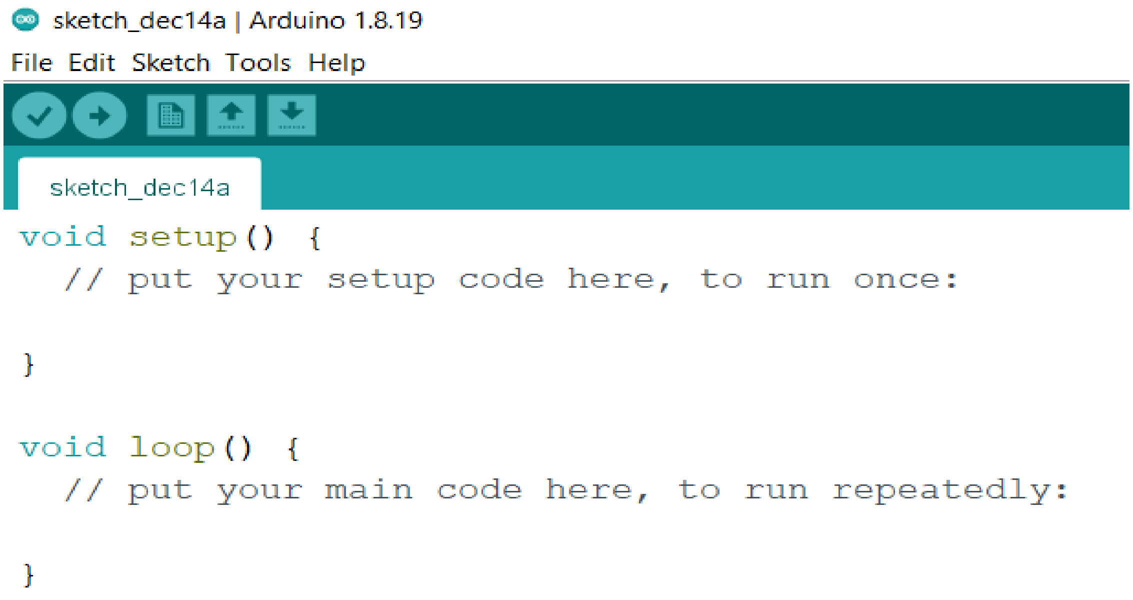
Open Source - Arduino Uno, like other Arduino boards, is open-source hardware and software. The design files and source code are freely available for modification and distribution.

Integrated Development Environment (IDE) - Arduino Uno is programmed using the Arduino IDE, a user-friendly environment for writing, compiling, and uploading code to the board. We can see the appearance of IDE window on Windows 10 in Figure 2.12:

The Arduino Uno is known for its simplicity and ease of use, making it an excellent choice for beginners and experienced developers alike for various projects, from simple LED blinking exercises to more complex robotics and automation applications.

Arduino's success lies in its commitment to simplicity, accessibility, and community collaboration. Whether you're a beginner exploring electronics or an experienced developer prototyping a complex project, Arduino provides a versatile and inclusive platform for a wide range of applications.

In essence, Arduino has transformed how people engage with technology, fostering a culture of exploration, learning, and collaboration. As it continues to evolve and adapt to emerging trends, Arduino remains a cornerstone in the world of microcontroller development, leaving a lasting legacy of empowerment and innovation. Arduino has become a revolutionary force in the world of electronics, programming, and maker communities, bringing accessibility and simplicity to microcontroller development.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 2.12. Main structure of the code on the Arduino IDE

2.1.7 A power supply (5V/3A/18W)

The power supply for our ATmega328P-based robotic arm with 4x SG90 servo motors is a critical aspect of our system. There are some considerations to choose effective power supply for our project:

Voltage Requirements - The voltage supplied to the ATmega328P is within its specified operating range, typically 1.8V to 5.5V. Most Arduino boards, including those based on the ATmega328P, can be powered via the USB port or an external power source. The recommended voltage for most Arduino boards is in the range of 7-12V. SG90 servos typically operate within a range of 4.8V to 6V. Exceeding the specified voltage can damage the servos.

Stability and Regulation - We should use a stable power supply with minimal voltage fluctuations. Voltage regulators may be necessary to ensure a consistent voltage level, especially if we are using batteries or an external power source with variable voltage.

Protection Features - The power supply for our robotic arm should include protection features such as overcurrent protection and reverse polarity protection to prevent damage to our components in case of power-related issues.

Power Distribution - If using multiple servos, we try to distribute the power to the servo motors efficiently to avoid voltage drops or uneven performance. It is good to consider using separate power lines for each servo or a power distribution board.

We use 18W SMPS power supply which is using as phone charger. It has 5V and up to maximum 3A in its output. A power value of the charger is 18W. It is enough for working of the robot. On calculation part we will write more clearly about voltage and current requirements of the robotic arm components. So, because of providing high current at its output, it is also called quick charge adapter. We can see its circuit diagram below in Figure 2.13.

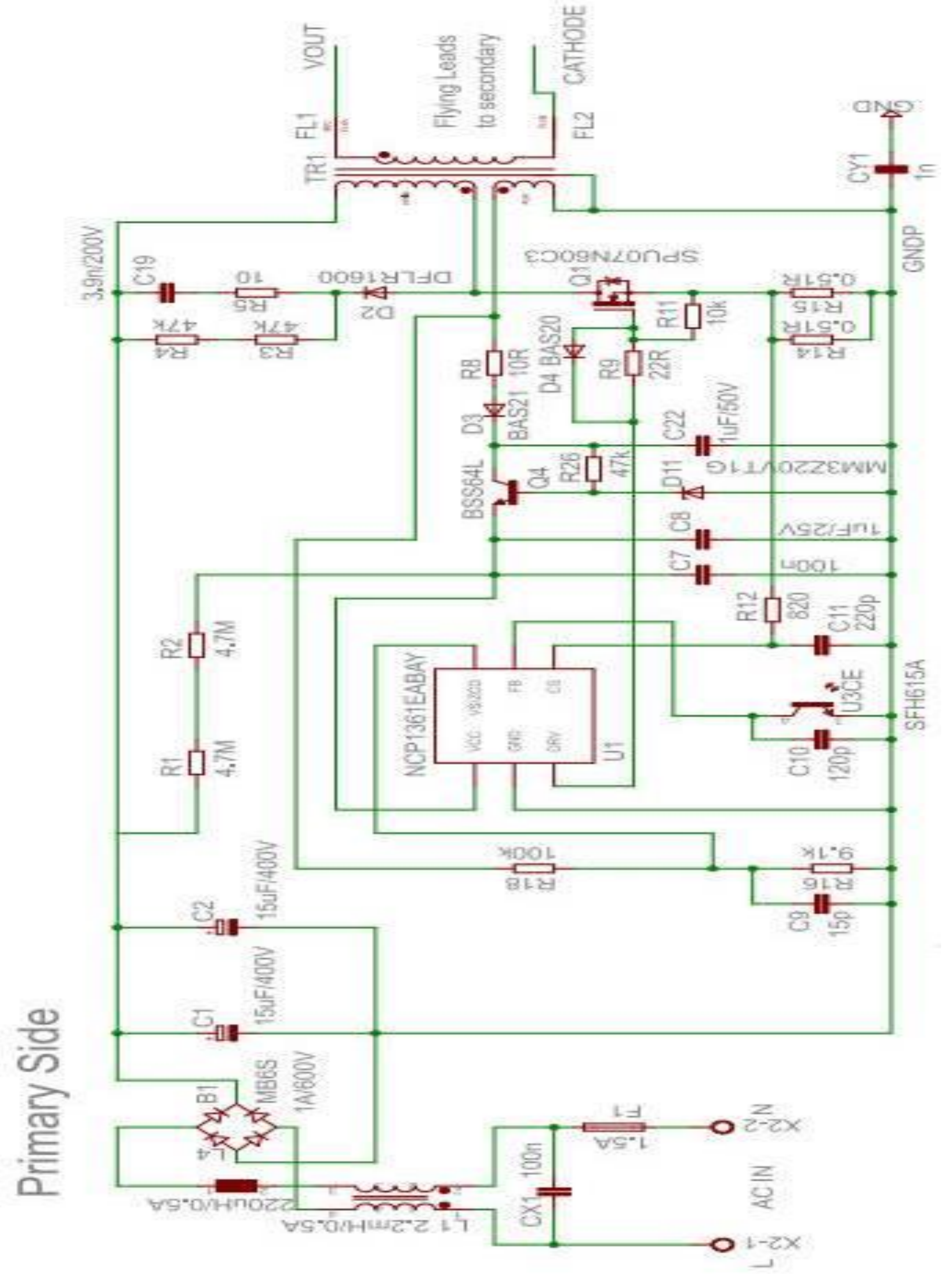


Figure 2.13. Internal circuit of the Power Supply

2.2. Block diagram and purpose of components

Defining the block diagram, a powerful visual representation that simplifies the intricacies of systems and aids in comprehending their functionality.

A block diagram is a schematic representation that employs simple geometric shapes or "blocks" to illustrate components, their connections, and the flow of signals or information within a system. These diagrams serve as a bridge between intricate technical details and a more digestible overview, allowing engineers, designers, and enthusiasts to grasp the essence of a system at a glance.

The main block diagram of our project consists of 4 main parts and 8 sub-parts (Figure 2.14). These 4 main parts and 8 sub-parts are below:

1) Power Supply
2) Arduino UNO
    2.1-Reset circuit
    2.2-AVR microcontroller
    2.3-Resonator circuit
    2.4-USB to TTL converter
3) Servo Motors
    3.1-DC motor
    3.2-Positional potentiometer
    3.3-Gears
    3.4-Controller board
4) Frame

In later chapters we will give more information about all parts. Significance of introducing the block diagram of our project is mainly for understanding the interaction between different components and time and cost efficiency. Therefore, by offering a visual roadmap, teams can work more efficiently, identify potential issues early on, and avoid costly errors. Let’s get deeper about each part of the robotic arm.

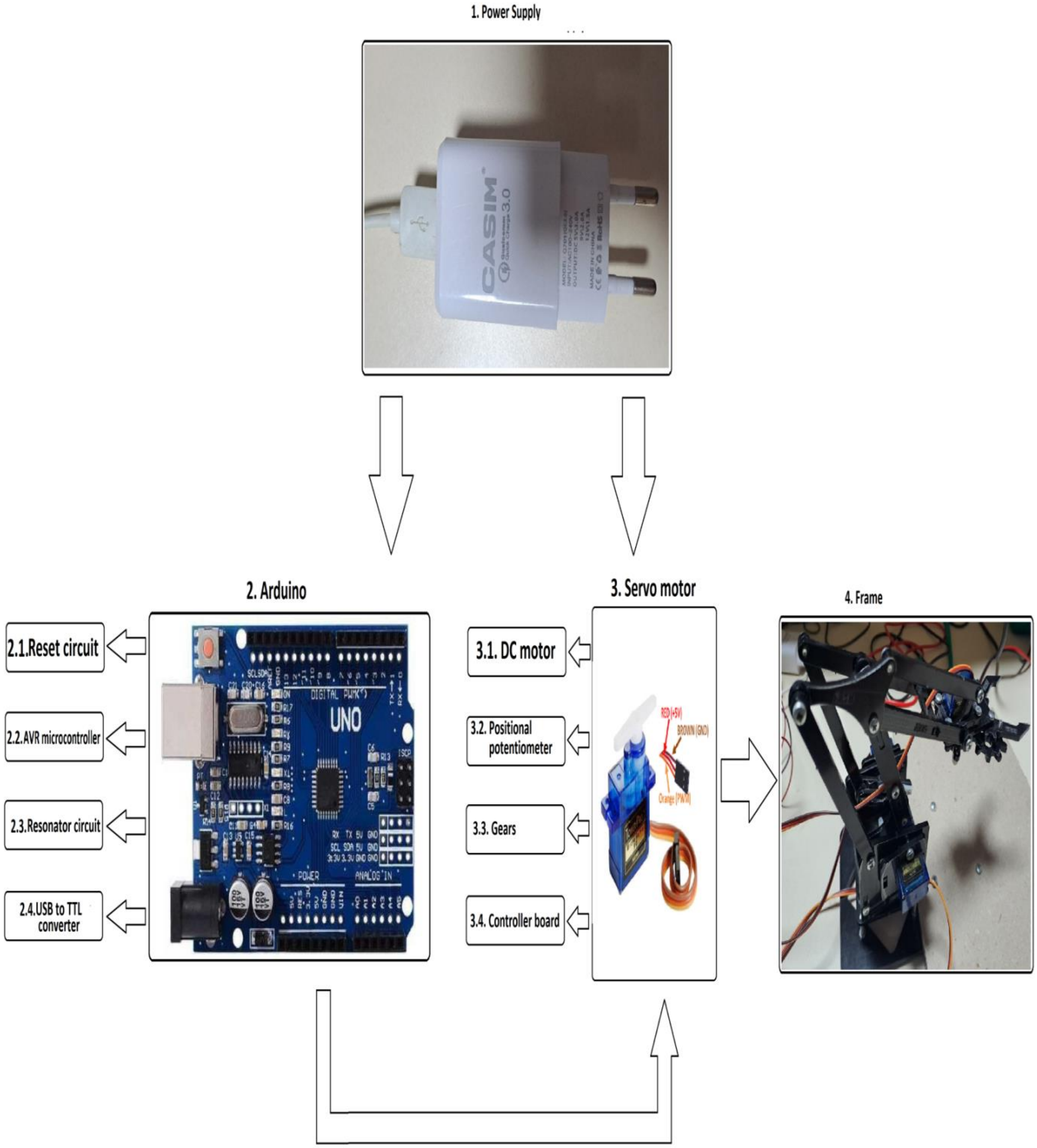


Figure 2.14. The main block diagram of the project

2.2.1. Power Supply

We use 18W SMPS power supply which is using as phone charger. It has DC 5V and up to maximum 3A in its output. Because of providing high current at its output, it is also called quick charge adapter. This Power Supply provides Arduino and Servo motors with DC 5 volt as maximal value. Required current by load can vary for its conditions such as at motion mode, stop mode, load mode and free mode. DC voltage is suitable for stable operation of microcontroller and servo motors.

2.2.2. Arduino UNO

Arduino UNO is a circuit board that is based on Atmega328p microcontroller chip. We use Arduino Uno smd version as main board for the project. Our Arduino UNO board uses SMD Atmega328p chip as microcontroller. There are 4 sub-parts of the Arduino Uno:

- Reset Circuit
- AVR microcontroller(Atmega328p)
- Resonator circuit
- USB to TTL converter

The reset circuit in the Arduino plays a crucial role in managing the microcontroller's startup and reset operations. The Arduino board typically includes a physical reset button. Pressing this button manually triggers a reset of the microcontroller, restarting the program execution from the beginning. The RESET pin on the microcontroller (e.g., ATmega328P in Arduino Uno) is an active-low pin, meaning that pulling it LOW triggers a reset. In normal operation, this pin is held HIGH through a pull-up resistor. To introduce a slight delay during power-up and ensure a stable reset, a capacitor is often connected between the RESET pin and ground (GND). This capacitor, known as the reset capacitor, forms an RC (resistor-capacitor) time constant, and its value determines the delay. The reset capacitor and pull-up resistor determine the duration of the reset pulse. The duration should be sufficient for the microcontroller to complete its reset and stabilize before starting program execution.

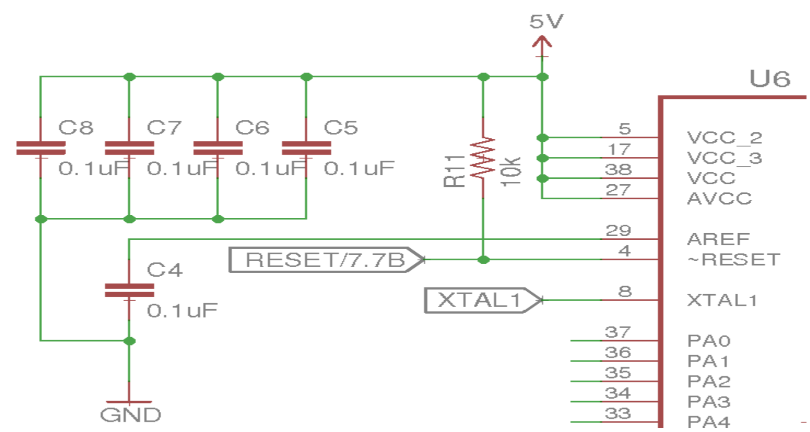


Figure 2.15 Reset circuit of the Arduino

The ATmega328P is an 8-bit microcontroller based on the Modified Harvard Architecture. It belongs to the AVR family of microcontrollers developed by Atmel (now a part of Microchip Technology). It typically operates at a maximum clock speed of 16 MHz. The clock speed can be configured using an external crystal or resonator connected to the microcontroller. It has 32 KB of Flash memory for storing the user's program (Arduino sketch). This memory is non-volatile, meaning the program remains stored even when power is removed. The ATmega328P is equipped with 2 KB of SRAM (Static Random-Access Memory), used for runtime data storage during program execution.

The most common resonator used in Arduino boards is a two-pin crystal oscillator that simplifies the oscillator circuit. The crystal oscillator provides a very precise clock signal. This oscillator is commonly used with a pair of load capacitors to stabilize the oscillation frequency. The crystal oscillator's frequency used in the Arduino Uno is typically 16 MHz. This frequency is the standard clock speed for the ATmega328P microcontroller on the board. Crystal oscillators usually have two pins: one for connection to the XTAL1 (crystal oscillator input) pin and the other to the XTAL2 (crystal oscillator output) pin of the microcontroller. It requires external load capacitors connected to the XTAL1 and XTAL2 pins for stabilization. These capacitors help control the oscillation frequency. The specific manufacturer and model of the crystal oscillator may vary between different batches or revisions of the Arduino Uno. Common manufacturers include Epson, Murata, and others.

The USB to TTL (Transistor-Transistor Logic) conversion is facilitated by the ATmega16U2 or ATmega8U2 microcontroller, which serves as the USB-to-Serial bridge. The

ATmega16U2/ATmega8U2 is responsible for converting USB communication to TTL-level serial communication, allowing the Arduino Uno to communicate with a computer or other devices via USB (Atmel, ATmega16U2 Datasheet, 2015). The USB connection is primarily used for programming the Arduino and for serial communication between the Arduino and the computer. When you connect the Arduino Uno to a computer, the computer typically recognizes it as a USB serial device. To facilitate this, you need to install the appropriate USB driver for the ATmega16U2/ATmega8U2. The driver establishes communication between the computer's USB port and the virtual serial port provided by the Arduino Uno. The ATmega16U2 or ATmega8U2 converts USB signals to TTL-level serial signals. The TTL-level serial signals are then transmitted to the main ATmega328P microcontroller via the TX (transmit) and RX (receive) pins. The TX and RX pins on the ATmega16U2/ATmega8U2 are connected to the RX and TX pins, respectively, of the main ATmega328P microcontroller on the Arduino Uno. This allows bidirectional communication between the USB port and the main microcontroller.

2.2.3. Servo Motor

Servo motors are a type of rotary actuator that allows for precise control of angular position. Servo motors typically consist of 4 main parts (Figure 2.16):

- small DC motor
- gears
- a potentiometer
- and a control circuit.

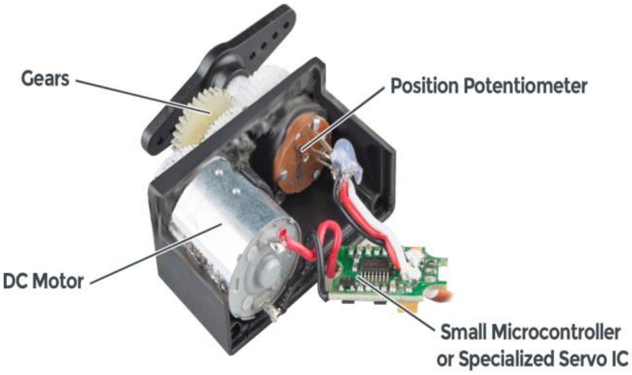


Figure 2.16 Internal parts of the Servo motor

Servo motors are known for their ability to rotate to a specific angle, typically within the range of 0 to 180 degrees. The specific range of motion depends on the design of the servo and its intended application. Servo motors receive control signals in the form of pulses. The width of the pulse determines the position to which the servo motor should move. Commonly, a pulse width modulation (PWM) signal is used, where the duty cycle of the signal corresponds to the desired position. The speed of a servo motor is influenced by factors such as voltage, load, and design. Servos are capable of relatively precise and controlled movements, making them suitable for applications where accuracy is crucial.

At the heart of a servo motor is a DC motor. The DC motor provides the rotational motion necessary for the servo's operation. The speed and direction of the DC motor are controlled by the servo's electronics.

Servo motors often include a gear train that connects the output shaft of the DC motor to the output shaft of the servo. The gear train serves to reduce the speed of the motor while increasing torque. This gearing allows for higher precision in positioning.

One of the critical components inside a servo motor is a potentiometer, which acts as a feedback device. The potentiometer is mechanically linked to the output shaft and provides feedback on the current position of the servo motor. This feedback is crucial for the closed-loop control system.

The control circuitry is responsible for interpreting the input signals and adjusting the position of the motor based on the feedback received from the potentiometer with PID technology (Maheshappa, 1989). The control circuitry typically includes a microcontroller and associated electronics.

2.2.4. Frame

The frame consists of a robotic arm and a platform. Robotic arm is made from the material - Plexiglas. Plexiglas is a brand name for a type of transparent acrylic plastic. The generic term for this material is polymethyl methacrylate (PMMA). PMMA is a synthetic polymer of methyl methacrylate, a monomer. It is often used as a lightweight and shatter-resistant alternative to glass. The arm has 4 DOF(Degrees of Freedom). The mechanical parts are laser cut to precision and are versatile to use. Its dimensions are 20 × 14 × 5.5 cm. The weight is approximately 150-200 grams.

2.3. Working Principle of the Robotic Arm

The working principle of a robotic arm with servo motors involves using these motors to control the movement of the arm's joints. Each servo motor corresponds to a joint in the robotic arm, and by adjusting the angle of each servo, you can control the position and orientation of the arm in three-dimensional space. Each part of the arm has own duty in the project.

Servo Motors: Servo motors are rotary actuators that allow for precise control of angular position. They consist of a DC motor, a set of gears, and a feedback control system. The feedback system typically includes a potentiometer or an encoder that provides information about the current position of the motor.

Arduino Control: The Arduino Uno is a microcontroller that serves as the brain of the robotic arm. It sends signals to the servo motors to set their angles. The Arduino runs a program (as shown in the code example) that defines the desired positions for each servo motor.

Degrees of Freedom (DOF): The number of degrees of freedom refers to the number of independent movements a robotic arm can make. In this case, the robotic arm has three degrees of freedom, meaning it can move in three directions: pitch, roll, and yaw.

Movement Commands: The program running on the Arduino defines specific angles for each servo motor to achieve the desired position and orientation of the robotic arm. By sending these commands to the servo motors, the arm can move to different positions in its workspace.

Mechanical Design: The physical structure of the robotic arm is designed to mimic the human arm or any other structure based on the application. The servos are attached at the joints of the arm, and the mechanical design determines how these joints move relative to each other.

Feedback Mechanism: Some advanced robotic arms incorporate feedback mechanisms to ensure accurate positioning. The feedback system allows the controller (Arduino) to receive information about the actual position of the arm and make adjustments if necessary.

Power Supply: Servo motors usually require an external power supply to operate. Ensure that the power supply provides enough voltage and current to drive all the servo motors in the robotic arm.

By coordinating the movements of the servo motors based on the programmed instructions, the robotic arm can perform tasks such as picking and placing objects, drawing, or any other application that involves precise positioning in space.

CHAPTER III. THEORETICAL PART

The theoretical part of this project contains some calculations with formulas, measurements of angles of arms, angular and linear speed of arm movements. This chapter is divided to 3 sub-chapters:

1) Modes of operation – we will analyze every conditions of robot interacting with the payloads, measure motors data parameters, voltage and current values. Modes of operation contain 4 main conditions: silent mode without payload; silent mode with payload; motion mode without payload; motion mode with payload. We will use table method to visualize each measurement of the modes.

2) Calculation of arm angles and angular/linear speed values – we will use many formulas to calculate all angles of arms and use these answers to calculate angular speed for each joint motors. Linear speed formulas are only for grabber of the robot to know the total time for carrying the payloads.

3) Calculation of power dissipation – we will calculate power dissipation values for each component of the robot by using current and voltage values which we measured in the first sub-chapter. Comparing these calculated power dissipation values with each component's nominal power values will show us how to use this robot at safe conditions.

3.1. Modes of operation

There are 4 operation modes of the robotic arm. All data signal values and oscilloscope graphes will be explained at this part. Additionally, we will use multimeter tool for measuring current and voltages values also for calculating total power dissipation at the third part. Voltage value will be same at each modes for all conditions. Because of we have connected each parts parallel to each other. Voltage remains a same value in parallel circuits. Figure 3.1 shows the circuit which has been built by author of this project on KiCAD PCB (Printed Circuit Board) design application.

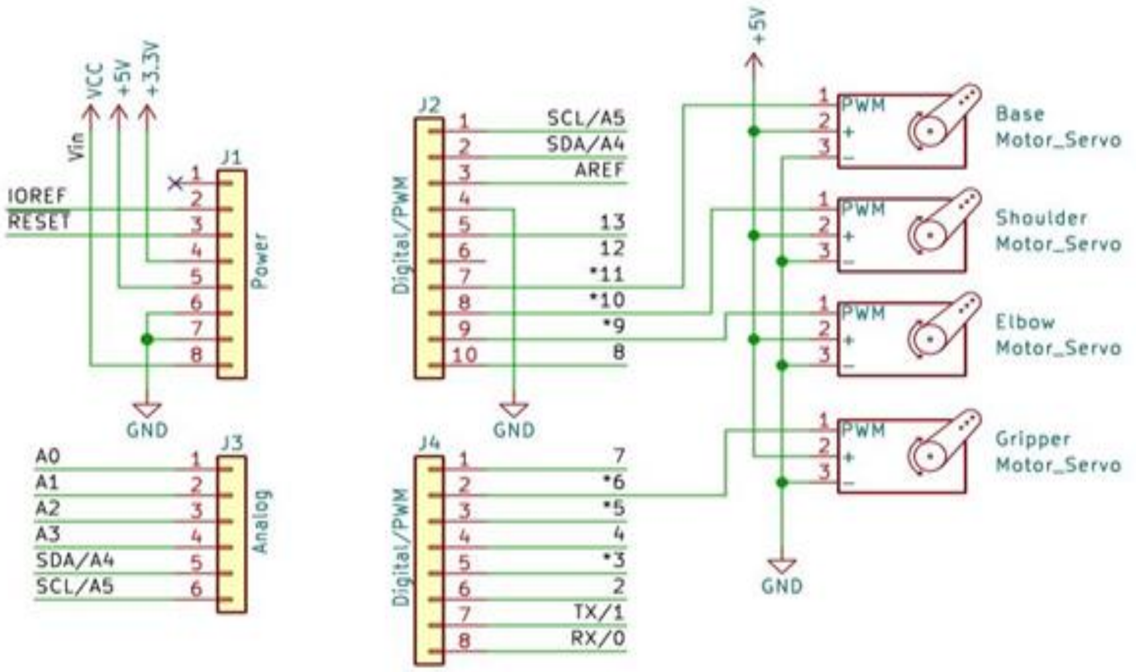


Figure 3.1. KiCAD circuit of the Robot controller

In this circuit, we can see all Arduino pins and Servo motors joined each others. The supply voltage and ground are common for all components. Supply voltage is 5v and maximum 3A that is enough for proper operation of all components. We can see each motor has 3 pins. “+” pins are supply voltage pins that is parallel connected with other motors and Arduino. “-” is ground pin that is common for all circuit. “PWM” pins of servo motor are signal pins that is connected according Arduino pins.

“PWM” pin of Base Servo motor is connected to digital pin 11 of the Arduino. “PWM” pin of Shoulder Servo motor is connected to digital pin 10 of the Arduino. “PWM” pin of Elbow Servo motor is connected to digital pin 9 of the Arduino. “PWM” pin of Gripper Servo motor is connected to digital pin 6 of the Arduino.

We measured all parameters in three operation modes. In each operation mode we measured degrees of elbow, shoulder, base and gripper motor, separately. We should know that these values are measured approximately. Considering additional parameters in real life may affect these values and cause some differences.

Measured parameters are illustrated in table 1 above. We measured all signal that comes into Servo motors with UTD2152 osilloscope. These type Servo motor work with PWM (Pulse

Width Modulation) signals. In PWM signal main parameters are duty cycle, high level width duration and period. On Table 3.2 we demonstrated high level width duration values of PWM signals that come into Shoulder motor.

Table 3.1. Measured parameters during 4 operation modes

Operation modes Motor Parameters		Silent mode without payload	Silent mode with payload	Motion mode without payload	Motion mode with payload
Gripper Motor	Current	3.2mA at 90^0 3.2mA at 48.7^0	0.46A at 90^0 0.46A at 48.7^0	3.2mA	0.35A
	Voltage	5.19V at 90^0 5.23V at 48.7^0	4.98V at 90^0 5.04V at 48.7^0	5.2V	5.04V
Shoulder Motor	Current	2.5mA at 90^0 2.5mA at 48.7^0	35mA at 90^0 2.5mA at 48.7^0	2.7mA	3.3mA
	Voltage	5.19V at 90^0 5.23V at 48.7^0	4.98V at 90^0 5.04V at 48.7^0	5.2V	5.04V
	Angular speed	-	-	$90\frac{degree}{s}$	$90\frac{degree}{s}$
	Linear speed	-	-	$765\frac{cm}{s}$	$765\frac{cm}{s}$
Elbow Motor	Current	1.2mA at 90^0 34mA at 48.7^0	1.5mA at 90^0 33.3mA at 48.7^0	2.3mA	2.8mA
	Voltage	5.19V at 90^0 5.23V at 48.7^0	4.98V at 90^0 5.04V at 48.7^0	5.2V	5.02V
	Angular speed	-	-	$90\frac{degree}{s}$	$90\frac{degree}{s}$
	Linear speed	-	-	$720\frac{cm}{s}$	$720\frac{cm}{s}$
Base Motor	Current	2.6mA at 90^0 2.6mA at 48.7^0	2.6mA at 90^0 2.6mA at 48.7^0	64mA	75mA
	Voltage	5.19V at 90^0 5.23V at 48.7^0	4.98V at 90^0 5.04V at 48.7^0	5.2V	5.02V
	Angular speed	-	-	$90\frac{degree}{s}$	$90\frac{degree}{s}$
	Linear speed	-	-	$405\frac{cm}{s}$	$405\frac{cm}{s}$

Table 3.2. PWM values of Shoulder motor measured on Oscilloscope

Angle	PWM value(+width duration)
48.7^0	2.4 ms
90^0	1.88 ms

Figure 3.2 shows PWM signal graph of shoulder motor in 2 degrees on the UTD2152 oscilloscope screen. As seen, each rectangle is 5ms on the screen of the oscilloscope. It means

the period of PWM is 20ms. The high level duration time of PWM signal for Shoulder motor on 48.7 is 2.4ms. The high level duration time of PWM signal for Shoulder motor on 90 is 1.88. But period values are same for both conditions.

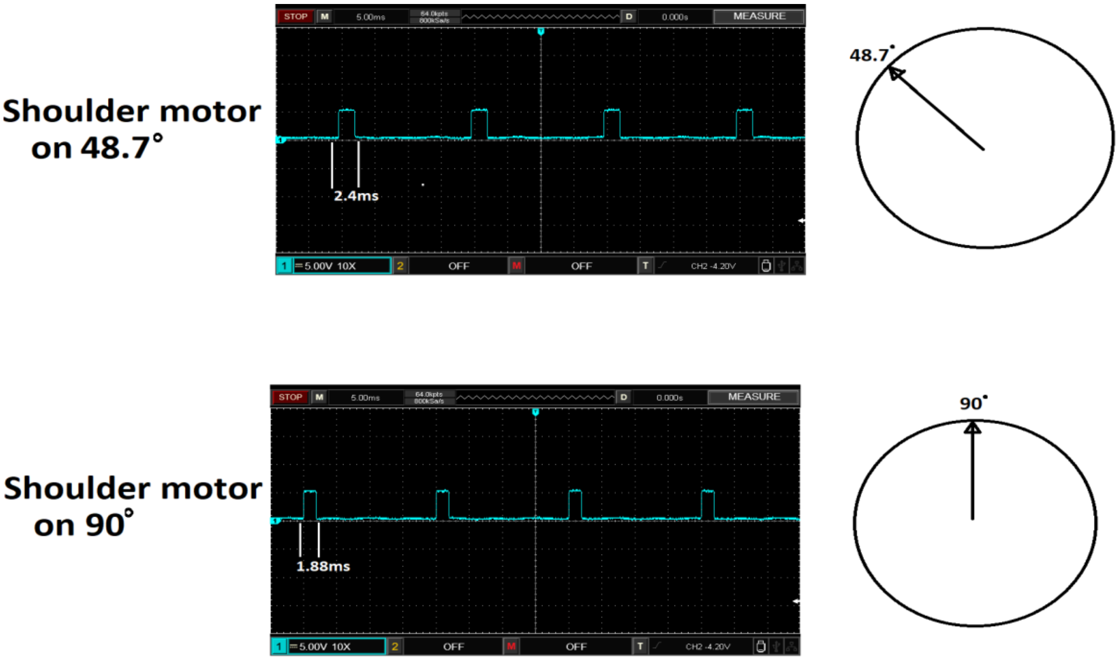


Figure 3.2. Oscilloscope screenshot of the Servo motor signal graph

On the motion operation mode motors have total 5 main steps. First step is initial position. That means robot plugged in and started now. It takes its initial position.

Step 1 means robot takes the payload on the left side. Step 2 means robot places the left payload on the center payload. Step 3 means robot takes the right payload. Step 4 means robot places the right payload on the center payload.

Table 3.3. Calculated arm angles for 5 steps of the robot

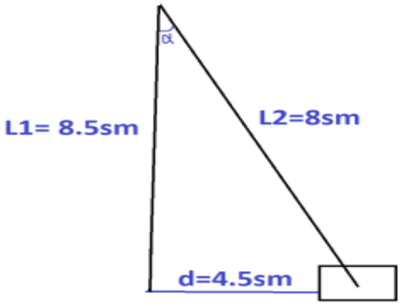
№	Motors	First position	Step 1	Step 2	Step 3	Step 4
1	Elbow	34^0	78.46^0	67.6^0	78.46^0	86.5^0
2	Shoulder	90^0	48.7^0	82.3^0	48.7^0	90.57^0
3	Base	Center	Left	Center	Right	Center

3.2. Calculations of arm angles

First Position:

$$sin\alpha = \frac{d}{L2} = \frac{4.5}{8} = 0.56 \quad \arcsin[0.56] = 34^0$$

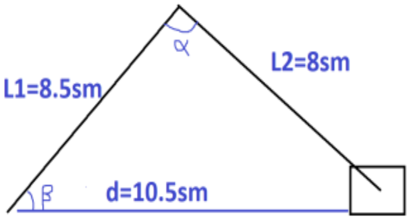
$$cos\beta = 90^0$$



Step 1:

$$cos\alpha = \frac{L1^2 + L2^2 - d^2}{2 * L1 * L2}$$

$$cos\beta = \frac{L1^2 + d^2 - L2^2}{2 * L1 * d}$$



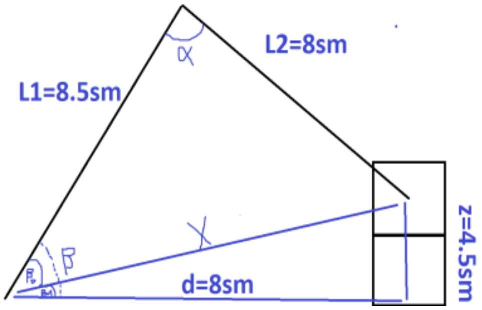
Step 2:

$$x^2 = d^2 + z^2 \quad \beta = \beta_0 + \beta_1$$

$$cos\alpha = \frac{L1^2 + L2^2 - x^2}{2 * L1 * L2}$$

$$cos\beta_0 = \frac{x^2 + L1^2 - L2^2}{2 * L1 * x}$$

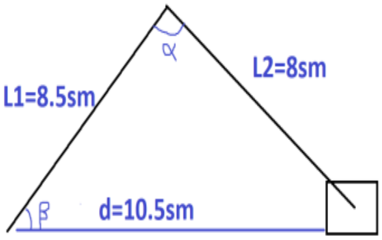
$$tan\beta_1 = \frac{z}{d} = \frac{4.5}{8} = 0.56 \quad \arctan[0.56] = 29.2^0$$



Step 3:

$$cos\alpha = \frac{L1^2 + L2^2 - d^2}{2 * L1 * L2}$$

$$cos\beta = \frac{L1^2 + d^2 - L2^2}{2 * L1 * d}$$

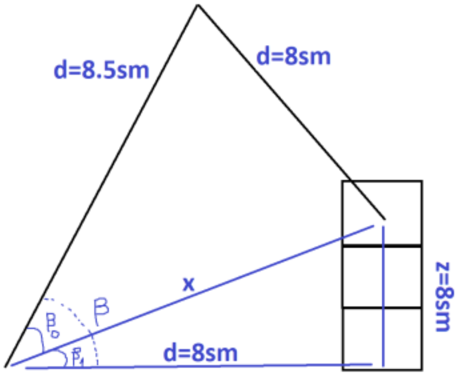


Step 4:

$$x^2 = d^2 + z^2 \quad \beta = \beta_0 + \beta_1$$

$$cos\alpha = \frac{L1^2 + L2^2 - x^2}{2 * L1 * L2}$$

$$cos\beta_0 = \frac{x^2 + L1^2 - L2^2}{2 * L1 * x}$$



$$tan\beta_1 = \frac{z}{d} = \frac{4.5}{8} = 0.56 \quad \arctan[0.56] = 29.2^0$$

3.2.1. Calculation of angular and linear speed

For angular speed by manufacturers datasheet we know these servo motors spends 1ms for rotation of each degree [32]. Hence, we can calculate all motors angular speed as below (Formula 1):

$$\omega = \frac{\theta}{t} = \frac{1^0}{1ms} = 1000\frac{degree}{s} \tag{1}$$

As we see 1000 degree/s angular speed value is so fast for a robotic arm to move. Thus, we have used slow movement system for each motor. We set additional 10ms interval for each degree rotation. It means each servo motor will spend 11ms for each degree rotation. So calculation of angular speed value for each motor is shown below (Formula 2):

$$\omega = \frac{\theta}{t} = \frac{1^0}{11ms} = 0.09\frac{degree}{ms} = 90\frac{degree}{s} \tag{2}$$

As we know length of the shoulder frame is r=8.5cm we can calculate its linear speed as below(Formula 3):

$$\upsilon = \omega * r = 90 * 8.5 = 765\frac{cm}{s} \tag{3}$$

For the length of elbow frame(r=8cm) linear speed for the elbow frame will be as below (Formula 4):

$$\upsilon = \omega * r = 90 * 8 = 720\frac{cm}{s} \tag{4}$$

Calculating linear speed for the base frame requires some asumement for the motion. As we know when gripper motor grabes the payload and pull it towards to the base frame, the distance will become 4.5cm between payload and base motor on movement shown as first position on the calculation of angles part. Thus, we can take the r value as r=4.5cm. Then, let's calculate linear speed for the base frame (Formula 5):

$$\upsilon = \omega * r = 90 * 4.5 = 405\frac{cm}{s} \tag{5}$$

3.2.2. Matlab simulation of slowed motion

On matlab we plotted ω(t) graph to visualize how much ω values differs from each others. We gave only two values to ω. First $1000\frac{degree}{s}$ which we calculated above, second

$90\frac{degree}{s}$ which we calculated for slowed motion. Time also has two values – 1ms and 11ms. We can show the matlab graph below in Figure 3.3.

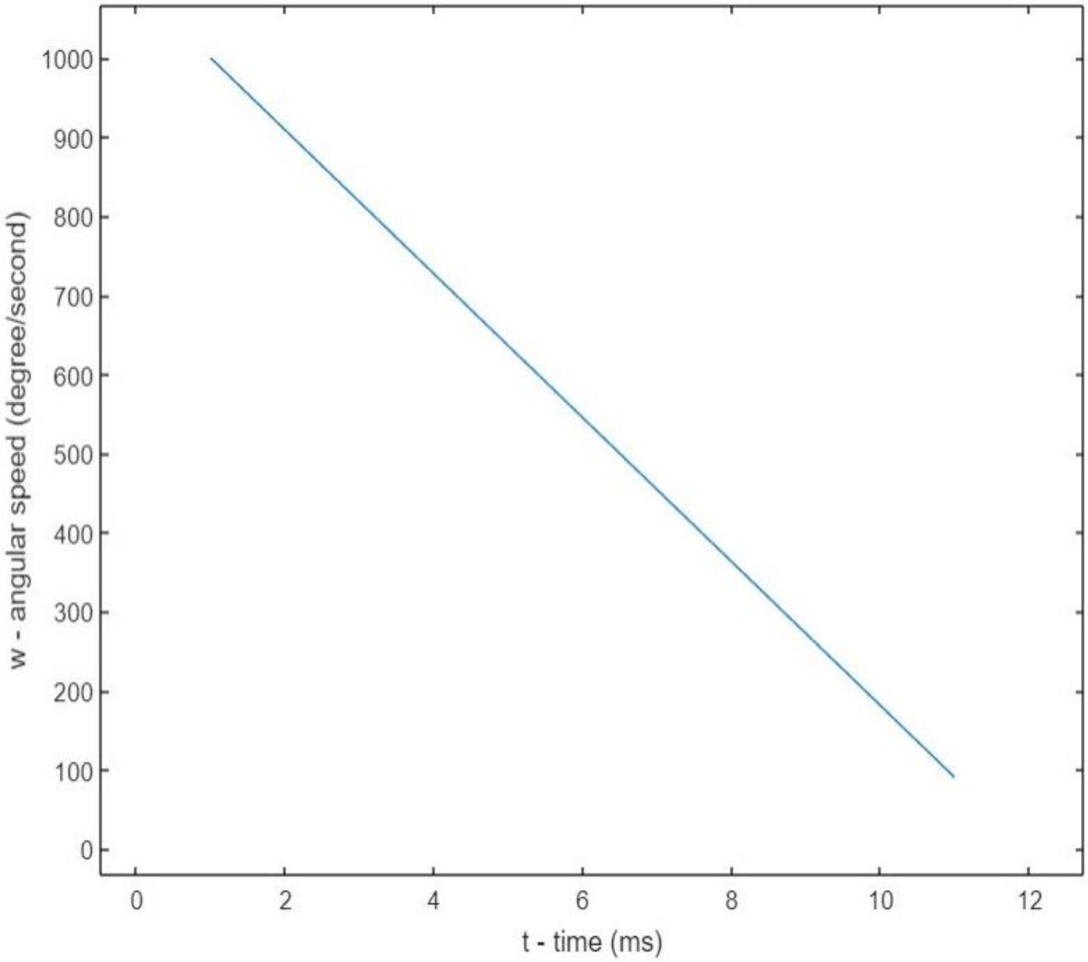


Figure 3.3. Graph of slowed motion on Matlab

3.3 Calculation of power dissipation

To estimate the power dissipation of a the robotic arm built using 4 servo motors controlled by an Arduino Uno, we will need to consider a few factors such as the current draw of each servo motor and the overall power requirements of our setup. The purpose of this chapter involves showing that how much calculated power dissipation requires from the 25W adapter. It will give us the effectiveness and power loss of the project components.

Firstly, we need to identify Servo Motor current and voltage parameters for accurate calculation. It can be get from its table 1. Then, we will calculate power consumption value for the total project. This formula can be used for power dissipation of each motor.

$$P = I * U \ (3)$$

Below, you can see total calculated power dissipation values of all Servo Motors for each operation mode:

Silent mode without payload: 0,2 W

Silent mode with payload: 2.6 W

Motion mode without payload: 0.4 W

Motion mode with payload: 0.4 W

According to the datasheet of Arduino UNO (Arduino, 2023), we can see current consumption of Arduino UNO is approximately ~0.7A. If we apply formula 3 for Arduino voltage 5V and Current 0.7A, we can find power dissipation of Arduino. So, it gives us 3.5W. It seems total power consumption of Arduino UNO and 4 Servo motors is approximately 6.1W. As our adapter is 25W, 6,1W consumes so little power from the adapter. It makes our project much useful and unique.

As conclusion, we identified the operating voltage and stall current of each servo motor and calculated the total power dissipation of the project. Arduino Uno has current limitations on its 5V output pins and overall current supply capability. Calculations showed us the total current draw of the servos exceeds what the Arduino can safely provide (typically up to 200 mA from the 5V pin). Thus we used external power supply for the project. After experiments we ensured that the power supply and wiring can handle the total power dissipation without overheating. Once we have estimated the power dissipation and ensured that our power supply setup is adequate, then we implemented and tested our robotic arm system to verify its performance and stability under different operating conditions. By carefully analyzing and addressing the power dissipation requirements of our 3DOF(Degrees Of Freedom) robotic arm with 4 servo motors, we can design a reliable and efficient control system using an Arduino Uno while ensuring safe and optimal operation of the servos.

CONCLUSION

The result involves the summary of potential outcomes and achievements from this project. Successfully building and programming a 3DOF robotic arm needs practical knowledge in robotics, electronics, and programming. In this thesis we demonstrated theoretical and practical side of a robotic arm that can perform basic tasks like picking and placing objects within its range of motion automatically. Engaging in this project provides valuable hands-on experience in robotics and embedded systems development, skills in Arduino programming, including interfacing with servo motors and implementing control algorithms and understand how to utilize Arduino's capabilities for real-world applications like robotics, learn to troubleshoot issues related to servo motor control, wiring, power management, and mechanical design.

Implementing and using a robotic arm based on 4 SG90 Servo motors and Arduino Uno successfully, shows that low power consumption of SG90 Servo motors and some technical parameters of Arduino Uno such as Flash memory size, operating frequency etc. are absolutely suitable for this kind of projects.

Using an SG90 servo motor in a robotic arm based on an Arduino Uno offers several advantages, too. It is small and lightweight, making it suitable for applications where space is limited, such as robotic arms. SG90 servos are inexpensive, making them a cost-effective choice for hobbyist and educational projects. This model servo motors can be easily interfaced with Arduino boards like the Uno, thanks to built-in libraries and example code available for servo control. They provide precise angular control over its rotation (0-180 degrees), allowing for accurate positioning of robotic arm joints. Their low power consumption feature is also suitable for our project. They operate at a low voltage (typically 5V) and consume relatively low power as we calculated before. And finally, while not as powerful as larger servo motors, the SG90 still offers enough torque for small-scale robotic arm movements.

In programming part of the thesis we discussed how to make Servo motor shaft rotate more slower than standart. Implementing such a feature on any robotic arm, makes it more accurate and safe. The main advantages and good differences of our robotic arm is slow-motion mode to pick and place payloads to targeted places on the board and operating automatically, not controlled by human.

Autonomous robotic arms offer several significant advantages over manually operated or semi-autonomous robotic arms. Autonomous robotic arms can operate continuously and perform tasks with high precision and speed, leading to increased productivity and efficiency in manufacturing or industrial settings. Thus, by replacing human labor with autonomous robotic arms, businesses can save on labor costs associated with repetitive or tedious tasks. Other advantages of autonomous robotic arms are unlike human operators, autonomous robotic arms can work around the clock without the need for breaks, reducing downtime and maximizing operational uptime and on other hand by automating tasks that are hazardous or ergonomically challenging for humans, autonomous robotic arms can enhance workplace safety and reduce the risk of workplace injuries.

In conclusion, the result of building a 3DOF robotic arm with servo motors controlled by an Arduino Uno extends beyond the physical device itself. It encompasses skills development, problem-solving, creativity, and potential contributions to the maker community. Such projects serve as excellent learning experiences and can pave the way for future endeavors in robotics and technology. The development and utilization of Arduino-based robotic arms offer a promising avenue for addressing a myriad of challenges across diverse fields. This thesis has explored the applications, advantages, and programming considerations associated with Arduino-based robotic arms, highlighting their versatility, affordability, and accessibility.

Through a comprehensive review of literature and practical experimentation, it has been demonstrated that Arduino-based robotic arms find application in education, prototyping, home automation, assistive technology, small-scale manufacturing, art and entertainment, STEM outreach programs, remote operation, agriculture, and environmental monitoring. These applications underscore the adaptability and potential impact of Arduino-based robotic arms across various industries and domains.

Furthermore, the thesis has emphasized the advantages of Arduino-based platforms, including ease of programming, cost-effectiveness, availability of support and resources, low power consumption, and integration capabilities. These advantages make Arduino-based robotic arms an attractive choice for beginners, hobbyists, educators, researchers, and professionals seeking to develop and deploy robotic systems for a wide range of purposes.

Programming considerations, such as pin mapping, sensor integration, motor control, algorithm implementation, error handling, and communication protocols, have been addressed to provide a comprehensive understanding of the technical aspects involved in developing Arduino-based robotic arms. Additionally, safety measures, testing procedures, and documentation practices have been emphasized to ensure the reliable and efficient operation of Arduino-based robotic arms in real-world scenarios.

In summary, Arduino-based robotic arms represent a valuable platform for innovation, learning, and problem-solving in robotics and related fields. By leveraging the capabilities of Arduino microcontrollers and open-source development environments, individuals and organizations can explore new applications, iterate on designs, and contribute to the advancement of robotics technology in a collaborative and accessible manner. As the field of robotics continues to evolve, Arduino-based platforms are poised to play a pivotal role in shaping the future of automation, creativity, and human-machine interaction.

REFERENCES

1. AVCalc LLC. (2024). Retrieved from AVCalc LLC: https://www.aqua-calc.com/calculate/volume-to-weight
2. A. Mazumdar, M. L. (2012). A compact, maneuverable, underwater robot for direct inspection of nuclear power piping systems. 2012 IEEE International Conference on Robotics and Automation (pp. pp. 2818-2823). Saint Paul, MN, USA: IEEE.
3. Abdel-Malek, L. (1987). Robot's economic repeatability. Engineering Costs and Production Economics, 12(1-4), pp. 93-97.
4. Achim Buerkle, W. E. (2021). EEG based arm movement intention recognition towards enhanced safety in symbiotic Human-Robot Collaboration. Robotics and Computer-Integrated Manufacturing, 70, p9.
5. Al Mamun, M. A. (2023). Design and Fabrication of a Magnet Based Pick and Place Mechanism for Industrial Applications. Dhaka, Bangladesh: Sonargaon University, Dhaka.
6. al., S. K. (2011). Robot concept for scalable, flexible assembly automation: A technology study on a harmless dual-armed robot. IEEE International Symposium on Assembly and Manufacturing (ISAM), (pp. pp. 1-5). ampere, Finland: IEEE.
7. AlloyMuhammad Nor Aiman Abd Razak, M. S. (2023). Design andSimulateRobotic Arm Structures from Dissimilar MaterialsBetween T6061 Aluminum and Cast-Iron Alloy. Research and Innovation in Technical and Vocational Education and Training , pp. 80-86.
8. Andrew Lobbezoo, Y. Q.-J. (2021). Reinforcement Learning for Pick and Place Operations in Robotics: A Survey. 10th Anniversary of Robotics—Feature Papers in Intelligent Robots and Mechatronics, p27.
9. Andriella, A. T. (2020). Short-Term Human–Robot Interaction Adaptability in Real-World Environments. Int J of Soc Robotics, pp. 639–657.
10. Arduino. (2023, 11, 10). Arduino R3. Retrieved from Datasheet: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf
11. ARM, R. H. (2020). A.C. GHEORGHE. Sciendo, pp. 37-39.
12. Atmel. (2015). ATmega16U2 Datasheet. Retrieved from ATmega16U2 Datasheet: https://ww1.microchip.com/downloads/en/DeviceDoc/doc7799.pdf

13. Atmel. (2015). ATmega328P Datasheet. Retrieved from ATmega328P Datasheet: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
14. Aude Billard, D. K. (2019). Trends and challenges in robot manipulation. Science, p8.
15. Balkeshwar Singh, N. S. (2013, May). Evolution of Industrial Robots and their Applications. International Journal of Emerging Technology and Advanced Engineering, 3(5), p763-768.
16. Billard, M. V. (2013). Safety issues in human-robot interactions. International Conference on Robotics and Automation (pp. 197-204). Karlsruhe, Germany: IEEE.
17. Bradley Saund, R. D. (2013). High Accuracy Articulated Robots with CNC Control Systems. SAE International Journal of Aerospace-V122-1, SAE International Journal of Aerospace-V122-1EJ, p780-784.
18. C. Blanes, M. M. (2011). Technologies for robot grippers in pick and place operations for fresh fruits and vegetables. Spanish Journal of Agricultural Research 2011, pp. 1130-1141.
19. Ç. Ersin, M. Y. (2020). Upper Limb Robot Arm System Design and Kinematic Analysis. ECJSE, pp. 1320–1331.
20. C. Schmidbauer, S. Z. (2023, April). An Empirical Study on Workers' Preferences in Human–Robot Task Assignment in Industrial Assembly Systems. Transactions on Human-Machine Systems, 53, pp. 293-302.
21. Cenk Baykal, R. A. (2017). Asymptotically Optimal Design of Piecewise Cylindrical Robots using Motion Planning. Robotics: Science and Systems 2017 (pp. 1-3). USA: Massachusetts Institute of Technology.
22. China, R. (2014). Standardization in Education versus Education about Standardization. Paradigm Shift in the Education of XXI-th Century? THE INTERNATIONAL CONFERENCE “Quality and dependability (pp. pp. 192-196). Sinaia, Romania: CCF.
23. Collins, D. (n.d.). Cartesian robot. Retrieved from Linear Motion Tips: https://www.linearmotiontips.com/what-is-a-cartesian-robot/
24. D. G. Chung, M. H.-S. (2016). Gravity compensation mechanism for roll-pitch rotation of a robotic arm. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. pp. 338-343). Daejeon, Korea (South): IEEE.

25. Danica Kragic, J. G. (2018). Interactive, Collaborative Robots: Challenges and Opportunities. Twenty-Seventh International Joint Conference on Artificial Intelligence (pp. pp. 18-25). Stockholm, Sweden: IJCAI.
26. Daniela Rus, M. T. (2015, November 27). Design, fabrication and control of soft robots. Nature 521, pp. 467–475.
27. Dimeas, F. F. (2019). Towards Progressive Automation of Repetitive Tasks Through Physical Human-Robot Interaction. Human Friendly Robotics (p. p151). Springer, Cham.
28. Eljamel, M. S. (2006). Robotic application in epilepsy surgery. The International Journal of Medical Robotics and Computer Assisted Surgery, pp. 233-237.
29. F. Pugin, P. B. (2011, October). History of robotic surgery : From AESOP® and ZEUS® to da Vinci®. Journal of Visceral Surgery, Pages e3-e8.
30. Gordon Cheng, A. N. (2001). Continuous humanoid interaction:: An integrated perspective — gaining adaptivity, redundancy, flexibility — in one. Robotics and Autonomous Systems, pp. 161-183.
31. Gorghiu, P. C. (2009). THE MODELING AND SIMULATION OF A ROBOTIC ARM. Bucharest, Romania: “Valahia” University of Târgovişte,.
32. H. A. F. Almurib, H. F.-Q. (2012). A review of application industrial robotic design. 2011 Ninth International Conference on ICT and Knowledge Engineering (pp. p105-112). Bangkok, Thailand: IEEE.
33. H. A. F. Almurib, H. F.-Q. (2012). A review of application industrial robotic design. 2011 Ninth International Conference on ICT and Knowledge Engineering (pp. pp. 105-112). Bangkok, Thailand: IEEE.
34. Hauser, K. (2012). On responsiveness, safety, and completeness in real-time motion planning. Auton Robot, pp. 35–48.
35. Helander, M. G. (1990). Ergonomics and safety considerations in the design of robotics workplaces: A review and some priorities for research. International Journal of Industrial Ergonomics, pp. 127-149.
36. Herakovic, N. (2010). Robot Vision in Industrial Assembly and Quality Control Processes. Ljubljana, Slovenia: University of Ljubljana.
37. Ilo, N. (2018). Robotic Process Automation Implementation in Record-to-Report process – Case Company X Oy. Hameenlinna, Finland: Master’s programme in business management and entrepreneurship.

38. J.A. Somolinos, V. F. (2002). Design, dynamic modelling and experimental validation of a new three-degree-of-freedom flexible arm. Mechatronics, 12(7), pp. 919-948.
39. Jadeja, Y. (2019). Design And Development Of 5-DOF Robotic Arm Manipulators. International Journal of Scientific & Technology Research, pp. 2158-2167.
40. James Pierce, P. M., Keith Needham, B., & Christopher Adams, A. C. (2020, July). Robotic Arm–Assisted Knee Surgery: An Economic Analysis. The American Journal of managed care, 26.
41. Jason D. McNulty, T. K. (2014). High-precision robotic microcontact printing (R-μCP) utilizing a vision guided selectively compliant articulated robotic arm. Lab on a Chip, pp. 1923-1930.
42. Jisu Elsa Jacob, M. N. (2022). Robotics Simplified. India: BPB Publications.
43. K. Bodie, C. D. (2016). ANYpulator: Design and control of a safe robotic arm. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. pp. 1119-1125). Daejeon, Korea (South): IEEE.
44. K. Kruthika, B. M. (2016). Design and development of a robotic arm. 2016 International Conference on Circuits, Controls, Communications and Computing (I4C) (pp. pp. 1-4). Bangalore, India: IEEE.
45. Kumar, A. B. (2017). Arduino controlled robotic arm. 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA) (pp. pp. 376-380). Coimbatore, India: IEEE.
46. Lim TG, C. H. (1990). Payload capacity of balanced robotic manipulators. Robotica, pp. 117-123.
47. Loris Barbieri, F. B. (2018). Design, prototyping and testing of a modular small-sized underwater robotic arm controlled through a Master-Slave approach,. Ocean Engineering, 158, Pp. 253-262.
48. M. B. Jamshidi, A. L. (2018). Socialization of Industrial Robots: An Innovative Solution to improve Productivity. 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, (pp. pp. 832-837). Canada.
49. M. Hamad, N. M. (2019). The Role of Robot Payload in the Safety Map Framework. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. pp. 195-200). Macau, China: IEEE.

50. M. Taylan Das, L. C. (2005). Mathematical modelling, simulation and experimental verification of a scara robot. Simulation Modelling Practice and Theory, pp. 257-271.
51. Maheshappa, H. D. (1989). Digital PID Controller for Speed Control of DC Motors. IETE Technical Review, pp. 171–176.
52. Marek Bujňák, R. P. (2022, February 12). Spherical Robots for Special Purposes: A Review on Current Possibilities. Sensors 2022, p1-36.
53. Mel Wolfgang, V. L. (2017). Gaining Robotics Advantage. bcg.perspectives by The Boston consulting group, p6.
54. Morteza Hadipour, J. F. (2020, June 15). Fully automatic cleaning system of smart street lights: a new design. SN Appl. Sci., 2, p1-3.
55. Mr.M.V.N.R.P.Kumar, M. K. (2015). AVR Microcontroller Based Moving Arm System. International Journal of Research in Advent Technology, pp. 30-33.
56. Nidhi Chahal, R. B. (2023). Robotic Arm Impact on Industrial and Domestic Applications. India: Handbook of Computational Sciences: A Multi and Interdisciplinary Approach.
57. Niko Sudibjo, L. I. (2019). Characteristics of Learning in the Era of Industry 4.0 and Society 5.0. Advances in Social Science, 372, p276-278.
58. Patrick Lin, K. A. (2012). Robot Ethics: The Ethical and Social Implications of Robotics. London, England: The MIT Press.
59. Pérez, R., Gutiérrez, S. C., & Zotovic, R. (2018). A Study on robot arm machining: Advance and future challenges. Annals of DAAAM & Proceedings, 2018 (p. p0931). Spain: EBSCO.
60. Poppeova, V., Uriček, J., Bulej, V., & Šindler, P. (2011). Delta robots - Robots for high speed manipulation (Vol. 18). Tehnicki vjesnik / Technical Gazette.
61. Pro, T. (2014, April 22). SG90. Retrieved from SG90: https://datasheetspdf.com/mobile/791970/TowerPro/SG90/1
62. Qaryouti, G. &.-M. (2019). Design and implementation of a three dimensions (3D) printer for modeling and pre-manufacturing applications. International Journal of Electrical and Computer Engineering (IJECE), pp. 4749-4757.
63. R. Lankin, K. K.-C. (2020). ROS-Based Robot Simulation for Repetitive Labor-Intensive Construction Tasks. 8th International Conference on Industrial Informatics (pp. pp. 206-213). Warwick, United Kingdom: IEEE.

64. Rawat, P. A. (2016). Pick and place industrial robot controller with computer vision. 2016 International Conference on Computing Communication Control and automation (ICCUBEA) (pp. pp. 1-4). Pune, India: IEEE.
65. S. Suri, A. J. (2018). SCARA Industrial Automation Robot. 2018 International Conference on Power Energy, Environment and Intelligent Control, (pp. pp. 173-177). Greater Noida, India.
66. Serdar Kucuk, Z. B. (2006). Robot Kinematics: Forward and Inverse Kinematics. In S. Cubero, Industrial Robotics: Theory, Modelling and Control (p. p964). Austria, Germany: Pro Literatur Verlag.
67. Sharath Surati, S. H. (2021, February). Pick and Place Robotic Arm: A Review Paper. International Research Journal of Engineering and Technology (IRJET), pp. 2121-2129.
68. Sutyasadi, P. (2022). Control Improvement of Low-Cost Cast Aluminium Robotic Arm Using Arduino Based Computed Torque Control. pp. 650-659.
69. Taghirad, H. D. (2013). Parallel Robots: Mechanics and Control. Tehran: CRC Press.
70. Types of Robots. (n.d.). Retrieved from EVS TECH CO., LTD: https://www.evsint.com/industrial-robots-types/
71. Tzafestas, S. G. (1991). Intelligent Robotic Systems. Boca Raton, Florida, USA: CRC Press.
72. Yechu Hua, J. S. (2022). Cost-effectiveness analysis of robotic-arm assisted total knee arthroplasty. PLoS ONE 17, p14.
73. Yoram Koren, M. S. (1987). End-Effector Guidance of Robot Arms. CIRP Annals, pp. 289-292.
74. Zain Ali, M. F. (2023). Design and development of a low-cost 5-DOF robotic arm for lightweight material handling and sorting applications: A case study for small manufacturing industries of Pakistan. Results in Engineering, p9.