



AN-NAJAH NATIONAL UNIVERSITY
FACULTY OF ENGINEERING AND INFORMATION
TECHNOLOGY
COMPUTER ENGINEERING DEPARTMENT

SmartStorage

Prepared By:
Osama Dweikat
Osama Khatib

Supervisor:
Eng.Muhannad Al-Jabi

January, 2025

Acknowledgment

First and foremost, we express our gratitude to God for the blessings bestowed upon us and for guiding us in this meaningful endeavor. We sincerely thank our parents and family for the support and encouragement necessary to achieve success and distinction in life.

We also wish to express our sincere appreciation to Eng. Muhannad Al-Jabi, our supervisor for his unwavering support and assistance throughout this project. His guidance and insights have been instrumental in helping us navigate the challenges and enabling us to accomplish this project.

Finally, we would like to thank the instructors of the Computer Engineering Department at An Najah National University for creating a positive learning atmosphere and sharing insightful information. Additionally, We would also like to extend our appreciation to all our colleagues in the workshop who worked alongside us, contributing to a collaborative and motivating environment. Their shared knowledge and teamwork made this project a truly enriching experience.

Dedication

This project is dedicated to the martyrs of Gaza and the West Bank whose memory and steadfast spirit are a source of strength and resilience. It is dedicated to our families for their continuous support and belief in us, and to our friends and colleagues for their encouragement and sharing in our moments of learning and growth. This work is a small effort towards much bigger battles and it is done with all sincerity, admiration and appreciation.

Disclaimer

This report was prepared by Osama Dweikat and Osama Khatib, both affiliated with the Computer Engineering Department within the Faculty of Engineering at An-Najah National University. It has not been modified or revised beyond editorial adjustments required for assessment, and it may include errors in language and content.

Table of Contents

Acknowledgment	1
Table of Contents	4
List of Figures	6
Abstract	7
1 Introduction	8
1.1 Problem Statement	8
1.2 Objectives	8
1.3 Significance	8
1.4 Report Organization	9
2 Constraints and Earlier coursework	10
2.1 Limitations & Constraints.....	10
2.1.1 Limited Time.....	10
2.1.2 Limited Resources	10
2.1.3 Lake of experience & Handling with components.....	10
2.2 Earlier coursework.....	11
3 Literature Review	12
4 Methodology	13
4.1 Standards and specifications.....	13
4.2 Hardware Components.....	13
5 Implementation	23
5.1 Overview of the system	23
5.2 Web & Mobile Page	23
5.3 Block Diagram	25
6 Results and Analysis	24
6.1 Storage Accuracy.....	24
6.2 Communication Efficiency and Reliability.....	24
6.2.1 Serial Communication Between ESP32 and Arduino Mega.....	24
6.3 Determine type of product by color sensor.....	25
6.4 Step Control and Movement Accuracy	25

6.5	Getting to the right place	25
6.6	Gripper Control	25
7	Conclusion and Future Work	26
7.1	Conclusion	26
7.2	Future Work.....	26
8	References	27

List of Figures

4.1	Arduino Mega 2560	14
4.2	ESP 32.....	14
4.3	COLOR SENSOR	15
4.4	LIMIT SWITCH	16
4.5	Mg996R METAL GEAR SERVO MOTOR.....	16
4.6	Stepper Motor Nima 17	17
4.7	Stepper Motor Nima 23.....	17
4.8	TB6600 Stepper Motor Driver	18
4.9	AC/DC Power Supply	18
4.10	Jumper Wires.....	19
4.11	Final Design.....	20
5.1	Control Page	22
5.2	Block Diagram.....	23

Abstract

Our graduation project focuses on developing an intelligent robot designed to enhance the process of arranging and storing products in warehouses and large spaces. This robot helps in accelerating the arranging process, utilizing the maximum possible space, reducing the need for human labor, and retrieving any product from any place in the shortest possible time.

The robot is a compact storage unit suspended at the highest possible height in the storage space and mounted on moving rails equipped with a gripper to help it pick up products. There is a limit switch opposite the specified storage areas to help the robot reach the specified location. The robot relies on information from the user via the ESP channel so that the user determines the location to be reached whether to arrange or retrieve a product.

1 Introduction

1.1 Problem Statement

Storing and arranging products is important for project management. Space may not be utilized properly, which leads to spaces being left unused, and arranging at high distances may endanger the lives of workers. Sometimes, not handling products properly and carefully leads to damage to these products and financial losses. Technological progress is very important to preserve these products and arrange them in an ideal way, which speeds up the process of retrieving them and knowing the available places for storage and all the details about the products such as their types, numbers and location without the need to search for them, which saves time and effort.

1.2 Objectives

The main objective of this project is to develop a smart robot capable of moving independently in storage spaces to arrange products in the appropriate places and retrieve them quickly and efficiently and utilize the largest possible space, which leads to reducing human labor and saving time and effort. It also provides all information related to storage spaces and alerts the user to allocate additional spaces for items in the event that the spaces allocated to them are fully filled.

1.3 Significance

The main objective of this project is to develop a robot capable of performing basic warehouse tasks independently with accuracy and efficiency. The project contributes to saving time and effort for project owners by automating activities such as sorting and retrieval. In addition, this robot aims not only to enhance efficiency but also to contribute to providing the necessary information about storage locations and the status of these locations, whether they are full or empty, and the possibility of adding products to them and arranging them in the appropriate manner.

1.4 Report Organization

This report provides a comprehensive summary of the Smart Storage project, covering various aspects. We begin by identifying the constraints encountered in our work, including equipment limitations, tools used, and time constraints. Following this, we review existing storage robotics systems to provide background on the topic and highlight the unique characteristics we have incorporated compared to those systems. The methodology adopted in our work is thoroughly explained, offering insight into the methods and strategies utilized. Subsequently, we present our findings and engage in a discussion to analyze and compare the results obtained. In conclusion, we offer a summary of the entire project and discuss our plans for future enhancements aimed at further improving our work.

2 Constraints and Earlier coursework

2.1 Limitations & Constraints

2.1.1 Limited Time

Working on a smart storage project was challenging because we had difficulty coordinating the project, concurrent materials, and labs, including time constraints and limited hours at the College of Engineering.

2.1.2 Limited Resources

We faced several challenges in terms of resources for a smart storage project. First, designing the project for 3D printing was a daunting task to find the right design for our idea. While working on the project, we discovered that the design was not suitable enough for our project idea. Second, finding the right steel bars for the design was difficult as we could not easily find the right thickness for the project, which slowed us down. The main challenge was finding the right handle for the project and trying to balance it on the designated places and work on lowering it in the right way. These resource constraints not only slowed down the progress of the project, but also required us to make adjustments to our design and budget, which ultimately affected the overall result.

2.1.3 Lack of experience & Handling with components

Working with new components, the ESP32 and the color sensor, was a challenge, especially when trying to integrate them to work together at the same time. In addition, we lacked practical experience. We encountered some hardware issues throughout our studies, and this was our first time in practical application, so we made some mistakes that cost us extra effort, money and time.

2.2 Earlier coursework

Our university coursework provided a solid foundation of knowledge crucial for the development of the SmartStorage robot. Courses in hardware engineering, such as Digital Circuits Design I and II, Microcontroller Programming, Electronics, Microprocessor Lab, and Critical Thinking and Scientific Research, equipped us with the necessary tools and expertise. These courses enabled us to tackle the challenges of designing and integrating hardware for the SmartStorage project and preparing a compelling presentation.

3 Literature Review

Here are some examples of agricultural robotics that we found while searching for some features to add to our project.

- Mobile Robot

Mobile robots, including Automated Guided Vehicles (AGVs) and Autonomous Mobile Robots (AMRs), are designed to transport goods within warehouses. These versatile machines can replace conveyors by following predetermined paths or navigate autonomously using advanced sensors and software, making them valuable for material handling and inventory management.

- Robotic Arms

Robotic arms, also referred to as robotic manipulators, pick and place robots or piece-picking robots, are equipped with multiple joints and end-effectors to handle a wide range of tasks. In logistics, they are used for picking, packing, and palletizing products. These precision instruments improve order fulfillment speed and accuracy. Robotic arms can replace humans at AS/RS workstations for picking, packing, and preparing orders.

- Automated Forklifts

Automated forklifts are designed to load and unload pallets without human intervention. They enhance warehouse efficiency by navigating predefined paths or employing advanced vision systems to operate autonomously.

- Automated Cranes

Automated cranes are employed for heavy-duty material handling and storage tasks. These towering machines can efficiently move and stack goods in high-bay warehouses, optimizing vertical space and minimizing manual labor.

4 Methodology

In this chapter, we provide a detailed account of the materials, methods, and standards applied in the development of our SmartStorage robot. This methodology encompasses the design, integration, and testing of various components, each crucial for the effective execution of storage tasks.

4.1 Standards and specifications

Our project adheres to engineering standards, ensuring reliability and compatibility. Specifically, we use the IEEE 802.11 standard for communication protocols, emphasizing seamless connectivity and data exchange within the system. We have written 3 main pieces of code which are for the Arduino Mega, the main design ESP32, and one for the ESP32, which handles the wireless communication. The codes were implemented using the Arduino IDE in C++ language which includes several libraries such as Servo.h, Wire.h, and many other libraries that helped to implement the required functionalities.

4.2 Hardware Components

In this section, we will discuss the hardware components we utilized and the reasons for their selection:

- **Arduino Mega**

The Arduino Mega 2560 is a flexible and powerful microcontroller board, an extended version of the popular Arduino Uno. It features the ATmega2560 microcontroller, providing 54 digital I/O pins, 16 analog inputs, and sufficient memory. We used it for the large number of inputs/outputs we needed to complete this project.

We used the Arduino Mega 2560 to connect with several key components to ensure smooth operation. We have attached a color sensor to distinguish between products during sorting, in addition to several limit switches to help the robot reach the required places.



Figure 4.1: Arduino Mega 2560

- **ESP32**

The ESP32 is a versatile microcontroller designed for IoT projects, featuring dual-core processing, built-in Wi-Fi and Bluetooth, large memory, GPIO pins, and communication interfaces. It is widely used in wireless communication, sensor integration, and IoT applications. In our project, we used a single ESP32 module that communicates with the robot and sends it information about where to perform operations, such as ordering or retrieving. The ESP32 sends a signal using the ESP-NOW protocol to the Arduino Mega, which prompts the robot to initiate the necessary operations. This setup allows for efficient and responsive automation.



Figure 4.2: ESP 32

- **Color Sensor**

The Color Sensor Module is a complete color detector, including a 4 white LEDs and TAOS TCS3200 RGB sensor chip. The four white LEDs provide a broad-spectrum light source. The TCS230 has an 8 x 8 array of photodiodes with color filters (16 Red, 16 Blue, 16 Green, 16 Clear). A light to frequency converter generates a 50% duty cycle square wave on the output pin. Frequency is directly proportional to light intensity. Output scaling 100%, 20% and 2% allows for wide dynamic range of intensities. The color selects (S2, S3), frequency scale (S0, S1) and output are TTL logic level and can be connected directly to the Arduino.



Figure 4.3: color sensor

- **Limit Switch**

The limit switch is also known as a micro switch. In CNC, 3D Printing, and Rep Rap Printers these switches are used as End Stops and to detect the max and min limits of the axis.

Mechanical switches are simple to use and cheaper than optical end-stops because they do not require a circuit board and only use 2 wires to connect the switch.

Pull-up and down resistors can be put close to the mainboard.

Limit switch with lever ended with a roller.

In our project, we used seven limit switches distributed over nine storage areas and a starting area, through which the robot's movement is monitored and it is helped to reach the specified location. When the robot reaches the limit switch of the area, it gives a signal to the robot to stop, and then the robot moves the grabber to place the product in its designated area or take a product from the area and place it in the starting area.



Figure 4.4: Limit switch

- **Mg996R Metal Gear Servo Motor (360 Degree Rotation)**

The TowerPro MG995 – Continuous Rotation 360° Servo is a standard high-speed servo motor that can rotate 360 degrees. You can use any servo code, hardware or library to control this servo. It is a digital servo motor that receives and processes PWM signal faster and better. This servo motor can be controlled using an Arduino that is connected with jumper wires. It has advanced internal circuitry that provides good torque, holding force and faster updates in response to external forces. This TowerPro SG90 Continuous Rotation 360° Servo Motor is unique among all the servo motors available because its operation is completely different from that of a standard servo. Instead of going to a specific angle, this servo will be fixed on a pulse of 1.5 milliseconds, with the longer pulse giving a forward rotation and the shorter pulse giving a backward rotation.

In our project we used the MG996R Metal Gear Servo Motor to manage the gripper. Its ability to control the angles precisely ensures a good grip on products of different sizes for transportation to storage areas.



Figure 4.5: The MG 996R Metal Gear Servo Motor

- **Stepper Motor Nima 17**

The NEMA 17 stepper motor is a basic stepper motor for 3D printers and CNC applications. Stepper motor is very useful in various applications, especially those that require low speed with high precision. This stepper motor is a high precision four-wire bipolar stepper motor, NEMA 17 size, 1.8 degree (200) step angle is one of the most parameters of stepper motor

In our project we used four NEMA 17 stepper motors. Two to move the robot left and right(Y-axis). And two to move the robot's gripper up and down(Z-axis).



Figure 4.6: Stepper Motor Nima 17

- **Bread Board**

This breadboard used for prototyping electronic circuits. It has two power rails (top and bottom) for distributing VCC (red) and GND (blue) and a central grid where components are inserted and connected. The middle gap separates two sections, making it ideal for placing microcontrollers. We used the breadboard to be able to distribute the VCC and GND to the parts used in the project and to connect the ESP32 to the Arduino.

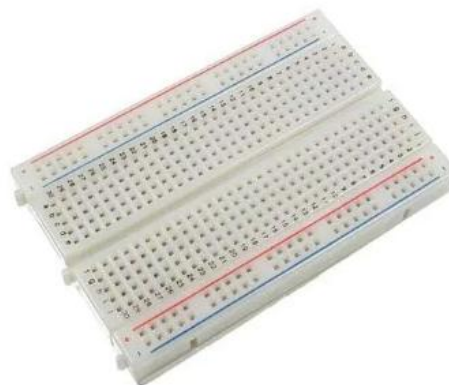


Figure 4.7: Bread Board

- **Stepper Motor Nima 23**

The Nema23 stepper motor is a high-torque electromechanical device that converts electrical pulses into discrete mechanical movements. Unlike traditional motors, which rotate continuously, stepper motors move in precise steps, allowing for accurate control of position and speed. In our project, we used one stepper motors to move the robot forward and backward(X-axis).

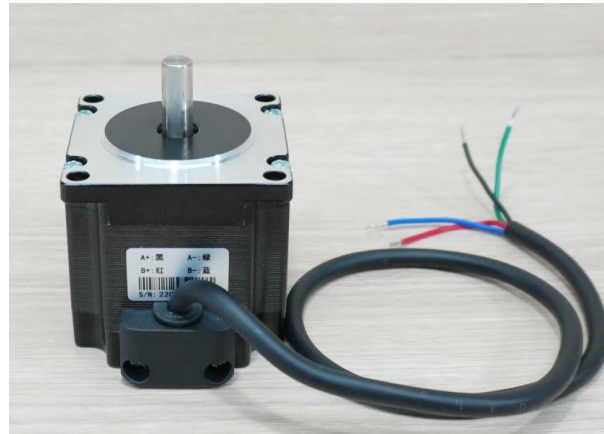


Figure 4.8: Stepper Motor Nima 23

- **TB6600 Stepper Motor Driver**

The TB6600 Stepper Motor Driver is a robust and reliable device designed to control stepper motors in high-performance applications that require high precision and torque. The driver is Known for its ability to handle up to 4A of current and operate within a voltage range of 9V to 42V. In our project, we utilized three TB6600 drivers to manage the stepper motors that control the movement of robot, ensuring accurate and reliable operation.

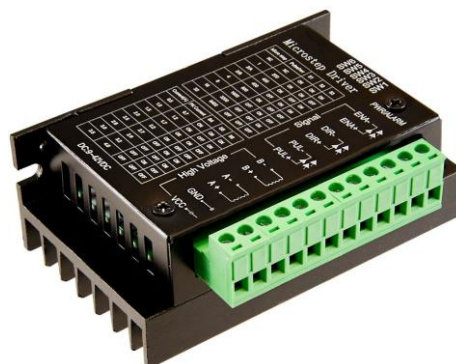


Figure 4.9: TB6600 Stepper Motor Driver

- **AC/DC Power Supply Adapter**

This 12V 5A 60W Power Supply, in our project, this power supply is used to power the entire system, including motors and controllers. It delivers constant power to all components, ensuring reliable performance throughout the robot's operation.



Figure 4.10: Power Supply

- **Jumper wires**

Jumper wires are flexible wires with connectors on each end used for making temporary electrical connections. They are commonly used in prototyping and circuit building to establish connections between various electronic components. In this project, we utilized jumper wires to establish the necessary connections between different components. We connected the pins of the microcontroller, sensors, modules, and other electronic components by simply plugging the connectors of the jumper wires into the appropriate slots.



Figure 4.11: Jumper wires

- **Final Design**

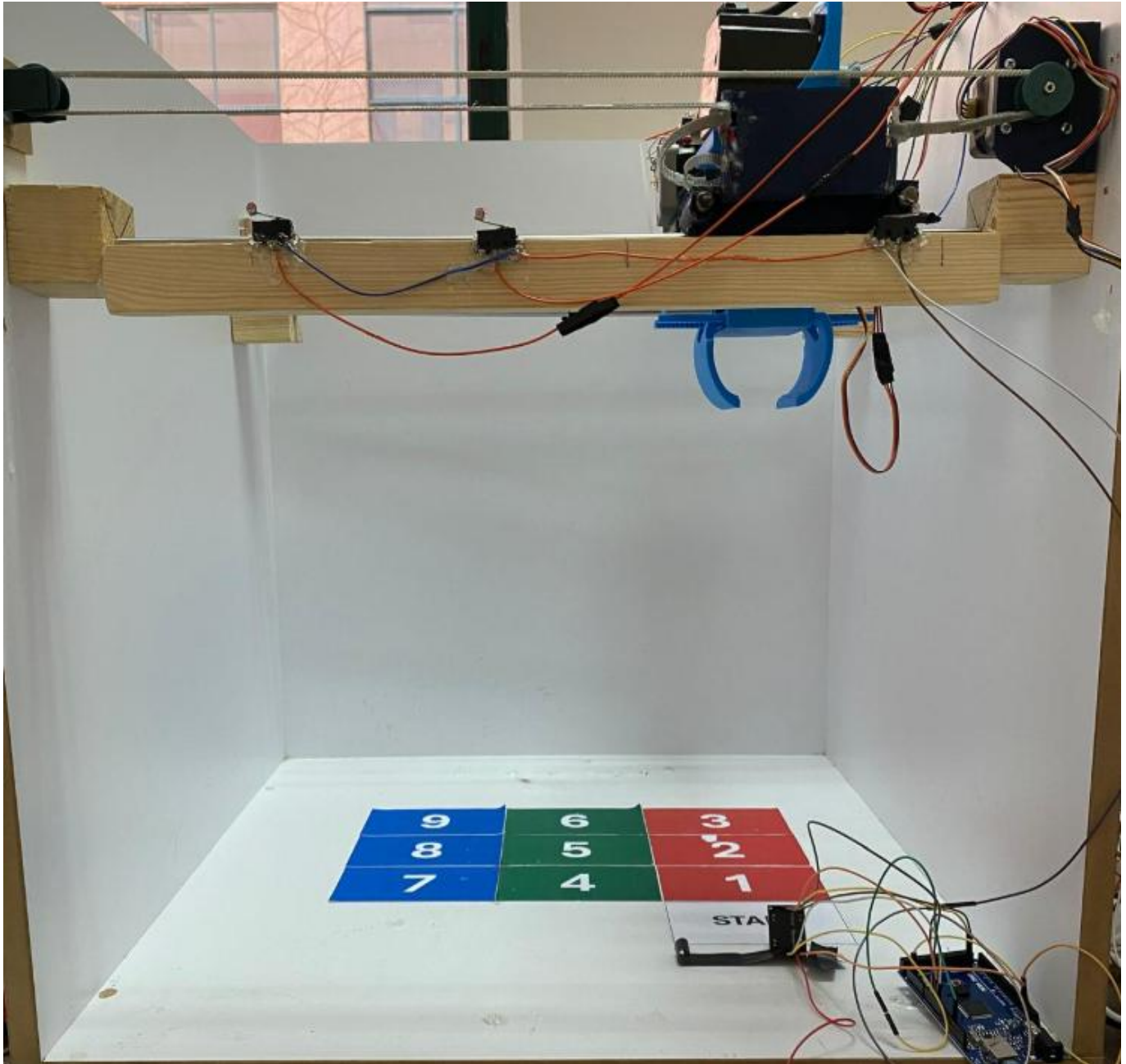


Figure 4.12: Final Design

5 Implementation

5.1 Overview of the system

The Smart Warehousing System is designed to automate storage tasks, providing a seamless user experience from start-up to operation. Upon activation, the system initializes all connected components, including sensors, actuators, and communication modules, ensuring they are ready for immediate use. The Smart Warehousing's core control interface is managed through an HTML page hosted on an ESP32 module. This interface allows users to control the robot directly, sending commands to perform various tasks such as sorting and retrieving products at any time. Users can monitor the status of the warehouse and information about the products inside such as their type and number, and monitor the available storage spaces and the spaces filled with products.

5.2 Web & Mobile Page

The Smart Storage System page is designed to be a user-friendly control interface, allowing for seamless management of the robot's functions. A dedicated button is placed at each location, with a dedicated button for automatic operation, providing intuitive access to the robot's various operations. Here is a detailed overview of the available controls:

- Automatic Button: When press this button, the product color is sent to the robot via the color sensor, and then the robot places the product in the closest place that matches the product and contains the least number of products.
- Numbered Buttons (1-9): These buttons allow the user to choose the place where he wants to arrange the product. When he clicks on them, a message appears for the user to specify the type of operation he wants to apply, arrange the product or retrieve it. If the place is full, the user cannot place products in it, and if it is empty, he cannot retrieve a product from the place.

Smart Storage

9	6	3
8	5	2
7	4	1

Automatic Storage Retrieve

Figure 5.1: Control Page

5.3 Block Diagram

In the smart storage system, the block diagram shows the interconnections between the various components to facilitate smooth operation. Sensors, including a color sensor that detects the type of product to be stored.

The Arduino Mega 2560 acts as the main controller, interacting with the ESP32 and managing the system outputs. These outputs include stepper motors, which control the movement of the robot, and limit switches to track the movement, enhancing the operational efficiency of the robot.

In this setup, the ESP32 is used to send necessary information about the storage locations such as the storage number and the type of operation to be performed on the products.

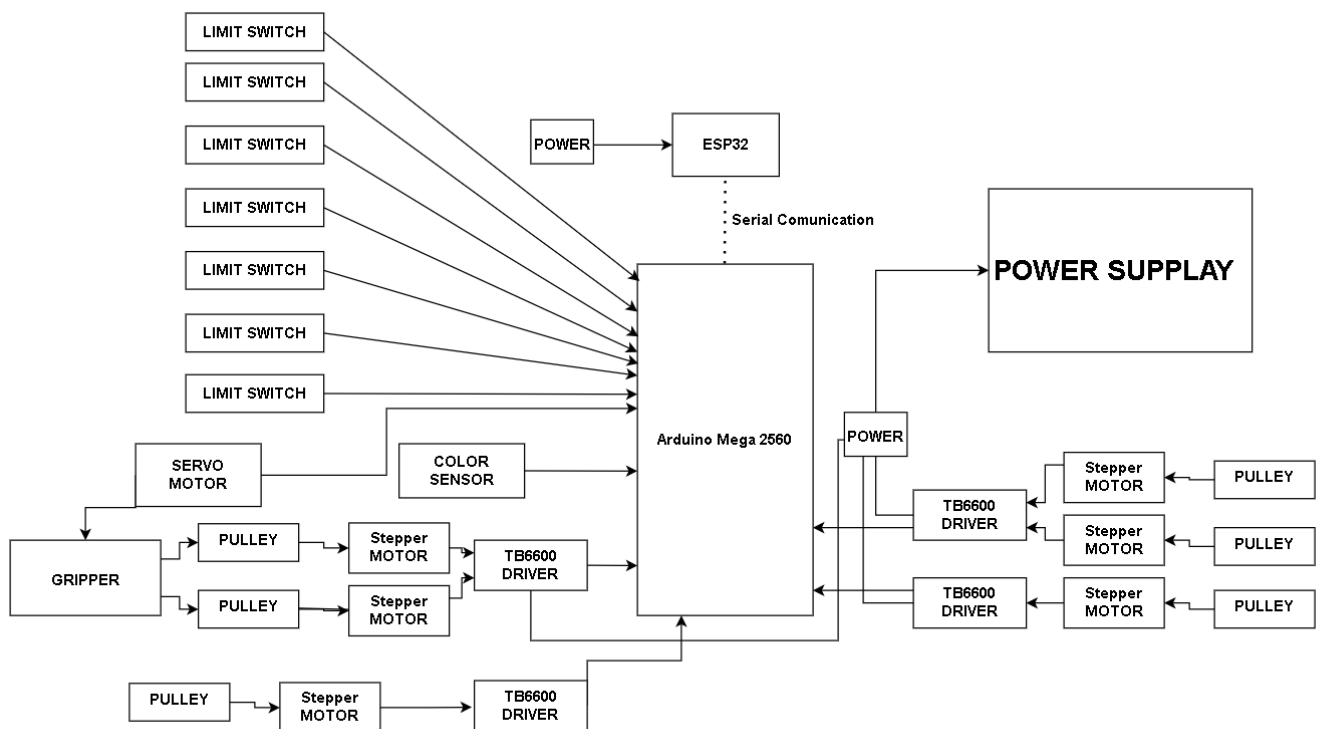


Figure 5.2: Block Diagram

6 Results and Analysis

In this section, we present a detailed summary of the data collected during the experimental trials of the SmartStorage system. The results are analyzed, and statistical treatment is applied to derive meaningful insights into the performance and efficiency of the system.

6.1 Storage Accuracy

The SmartStorage robot places the product in the right place with an accuracy 85%. We did not reach the ideal condition. This is due to many reasons, including the effect of the servo motor wires on the movement of the handle during descent and the light weight of the handle, which makes it easy to affect its movement.

6.2 Communication Efficiency and Reliability

The ESP32 module successfully sends the necessary information about storage locations and alerts the user if the product does not fit the chosen location or if additional locations for the products are needed.

6.2.1 Serial Communication Between ESP32 and Arduino Mega

The serial communication between the ESP32 and the Arduino Mega was successful, with data transfer remaining stable and consistent. The data transfer rate of 115200 allowed for fast and reliable data transfer, including the requested place number, product type, and the type of operation to be performed on it. This reliable serial communication supported real-time decision making and control, which is critical for robot operations within the warehouse.

6.3 Determine type of product by color sensor

The use of a color sensor has restricted us from using specific colors to distinguish between products. The reason for this is that the color sensor works with RGB model technology. Another note is that the color sensor does not measure long distances, which led to placing it close to the product to give the correct reading about it.

6.4 Step Control and Movement Accuracy

The Nema23 and Nema17 stepper motors, controlled by the TB6600 drivers, exhibited high precision in movement. Allowing the robot to move smoothly and accurately through different areas of the warehouse. This precision is essential to ensure that products are arranged and managed correctly.

6.5 Getting to the right place

The Limit switch was used to track the movement of the robot while arranging or retrieving products, which made the robot move correctly to reach the location specified by the user via the web page.

6.6 Griper Control

The servo motor is used to control the gripper with high precision. A 360-degree servo is used to give the gripper the freedom to hold different sized products.

7 Conclusion and Future Work

7.1 Conclusion

Smart Storage is a major advancement in warehouse management and organization, providing an effective integration of hardware and software to enhance product management for entrepreneurs. The project demonstrates how technology can improve and organize warehouses by automating product ordering and movement in storage space with precision.

The successful deployment of the Limit Switch in storage areas ensures that the robot moves correctly and facilitates access to these areas. Precise control of the robot's movement and reliable communication between components allows for real-time and accurate operation, reducing manual labor and minimizing errors.

The ability of Smart Storage to automate basic tasks and provide real-time data makes it an easy-to-use and effective solution for entrepreneurs looking to improve their product management. Overall, Smart Storage highlights the thoughtful integration of advanced technologies and how it can lead to significant improvements in reducing and optimizing storage space.

7.2 Future Work

For future enhancements, SmartStorage can benefit from several improvements.

1. Use QR reader instead color sensor: Use QR code instead of color sensor to know the type of product and not to be restricted by colors, which helps to know the largest number of products.
2. Upgrade Gripper: Using modern gripper can be able to gripe more than one box in the same process and also make it gripe boxes of different sizes.
3. Use sensors on places: Which facilitates the process of knowing the number of products available in places without the need to store them using array.
4. Multitasking: Enhance the robot to handle more tasks, such as Sort and retrieve products in the same process, speeding up robot process.
5. Add new process to the robot: Such as adding the process of rearranging products of the same type, where the robot distributes the products to their designated places equally.

8 References

- 1 Arduino Mega 2560 Rev3. <https://docs.arduino.cc/hardware/mega-2560>.
- 2 Stepper motor Tb6600 and tutorial. <https://www.instructables.com/Nema-Stepper-Motor-23-With-Tb6600-Driver-With-Ardu/>
- 3 ESP32 Tutorials. <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/?fbclid=IwAR1QOf-rmBSVUqzWyaFbDvnrZV3kf3JnphQMYq6riaj3L479rdc>
- 4 [7 Types of Robots in Warehousing: AutoStore & Beyond](#)
- 5 YouTube. Various channels like, <https://www.youtube.com/@nikodembartnik>