An-Najah National University

Faculty Of Engineering and Information Technology

Computer Engineering Department

<u>Storage Machine System</u>

PREPARED BY:
Tasneem Abu Rajab
Heba Damide

SUPERVISED BY:
Dr. Hanal Abu Zant

January 29, 2024

Acknowledgements

We want to extend our deepest thanks to Dr. Hanal Abu Zant, our supervisor, whose unwavering commitment and ongoing assistance have been instrumental in the completion of our project. His invaluable scientific mentorship has significantly contributed to our achievements. Additionally, we extend our appreciation to the academic professionals in the Computer Engineering Department for their guidance and support, along with our heartfelt gratitude to our friends and family for their unwavering belief in our potential. Their collective contributions, encouragement, and confidence in our abilities mean the world to us.

Disclaimer

The subsequent report has been written by students belonging to the Computer Engineering Department, Faculty of Engineering, at An-Najah National University. Minor modifications have been applied to the report, primarily focusing on editorial improvements, yet there might persist inaccuracies in both language and content. It's crucial to emphasize that all viewpoints articulated in this report, encompassing conclusions and suggestions, are the sole perspectives of the students. An-Najah National University disclaims any responsibility or legal obligations for potential repercussions resulting from the use of this report for objectives divergent from its originally intended purpose.

Contents

Abstract

A storage machine is a system that automates the organization and sorting of goods. The robot operates by using a barcode scanner to read the barcode on each box. The worker places the boxes on the robot arm, and it automatically deposits them in the correct cell within a cabinet. Moreover, the robot can deliver the box to the worker.

The robot employs a push mechanism, allowing the buyer to select the desired box. This integrated system, combining robot and retrieval mechanisms, enhances the overall management process. It optimizes the storage and dispensing of goods, reducing the likelihood of human errors, and providing convenient and efficient access for users to obtain the goods they need.

1 Introduction

Our graduation project evolved from real-world storage challenges, where existing methods struggled with inefficiencies, causing delays and errors. These obstacles inspired our creation of an innovative automated storage system. Powered by advanced robotics and barcode scanning, our solution tackles these issues by simplifying sorting, reducing errors, and enabling direct item delivery. This endeavor aims to close the gap between traditional storage limitations and the demand for a more effective, error-free, and user-friendly system.

1.1 Problem Statement

The main problem is the drawbacks inherent in typical storage systems that prompted the inception of our project. These traditional systems encounter issues like sluggish sorting, manual errors, and challenges in organizing inventory, causing delays, higher expenses, and operational inefficiencies. The lack of automation leads to slower operations and increased chances of errors. Moreover, restricted user-friendly interfaces hinder accessibility for end-users. Our initiative aims to tackle these limitations by introducing an advanced automated storage solution, aiming to streamline processes, reduce errors, and improve accessibility compared to conventional methods.

1.2 Objectives

Our aim is to create a groundbreaking automated storage system that transforms traditional methods. Our main targets involve improving efficiency, speeding up sorting, minimizing errors, and enhancing user access. This will be achieved by integrating advanced technologies like robotics and barcode scanning, prioritizing user-friendly interfaces, and showcasing cost-effectiveness compared to standard methods. Additionally, we aim to ensure the system's adaptability for future expansions. These objectives serve as a roadmap for evaluating the impact and effectiveness of our system.

1.3 Scope of work

1. **Automated Organization System:** A system designed to automate the sorting and arrangement of goods using barcode scanning. It involves a robotic arm to handle the placement of items into specific storage compartments within the cabinet.

2. **Barcode Recognition Technology:** Integration of advanced barcode scanning features for swift and accurate item identification within the system.

3. **User-Friendly Interface:** Creation of an intuitive interface enabling users to access easily. This interface includes a selection mechanism for users to choose desired items, and add new ones.

4. **Robotics and Control Mechanism:** Incorporation of robotics for precise movement and placement of items. Additionally, a control system manages the automated processes, potentially featuring a touch-screen interface for operation.

5. **Enhanced User Accessibility:** Ensuring easy access for end-users to retrieve items efficiently either via the system interface or by enabling direct item delivery.

1.4 Significance

Our automated storage system marks a significant stride in efficiency by considerably diminishing errors and time consumption in contrast to conventional methods. It not only streamlines processes but also presents a cost-effective solution, ensuring straightforward user access to stored items, highlighting a considerable advancement in storage management with its simplicity. Furthermore, integrating cutting-edge technologies such as robotics and barcode scanning sets a pioneering benchmark, potentially revolutionizing storage methodologies across diverse sectors.

2 Constraints and Earlier Coursework

2.1 Constraints & Limitations

1. Some components were hard to find with the right dimensions for the project. Therefore, we had them customized and ordered upon a specific request.
2. Accuracy was crucial in the project. An error of just one millimeter during testing led to collisions and damage to the components.
3. Finding suitable power supplies for the project.
4. Meeting in one place for project work was tough due to occupation barriers limiting our mobility and time.

2.2 Earlier Coursework

1. Coursework in electronics encompasses diverse elements of electronic systems and technologies.
2. A course dedicated to microcontrollers delves into their fundamental principles and real-world applications.
3. The Microcontrollers Lab offers practical exposure, involving hands-on sessions with Arduino and its features, exploring subjects like stepper motor control.
4. The curriculum includes critical thinking and scientific research, imparting skills like analyzing scientific literature and employing contemporary tools such as LaTeX for crafting research documents.

3 Literature Review

The evolution of storage solutions has witnessed substantial progress over time. Initial concepts centered on manual organization, gradually transitioning to automated methodologies that transformed storage management. Innovations like barcode scanning, integration of robotics, and the advent of IoT (Internet of Things) have expanded storage system capabilities.

The demand for effective storage solutions has been emphasized across various industries and sectors. Research highlights the hurdles users face, including accessibility issues and limitations in traditional storage methods. For example, deficiencies in manual inventory management systems have led to inefficiencies in item retrieval and inventory control.

Our project addresses these issues by implementing an advanced automated storage system. By integrating cutting-edge technologies like robotics and barcode scanning, our system aims to offer a cost-efficient and effective solution, overcoming the constraints of conventional storage approaches. This comprehensive approach is poised to revolutionize storage practices, providing streamlined operations and improved accessibility to stored items.

4 Methodology

4.1 System Architecture

4.1.1 Wooden Body

We designed a wooden box, which will be used later to attach the screen.

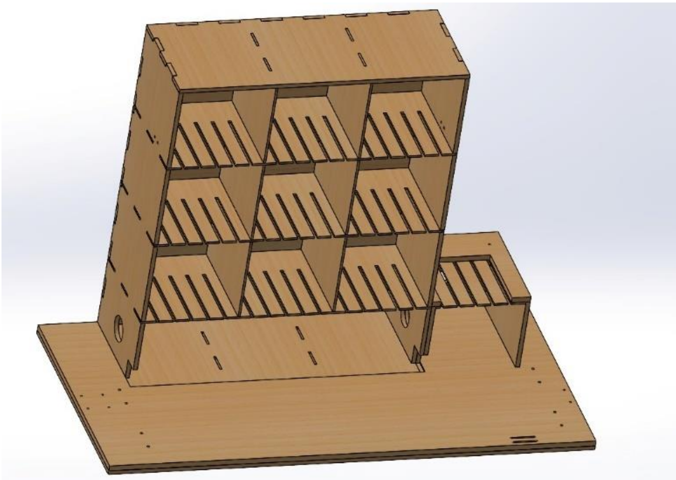
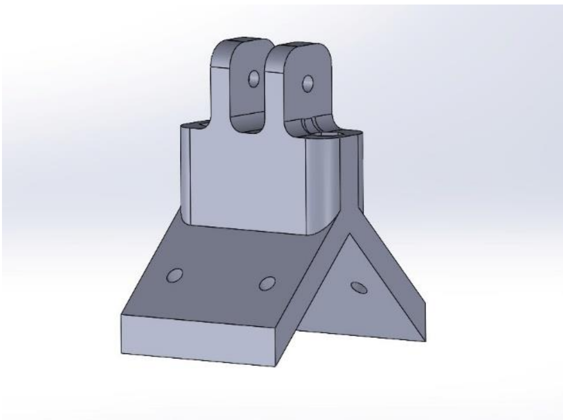
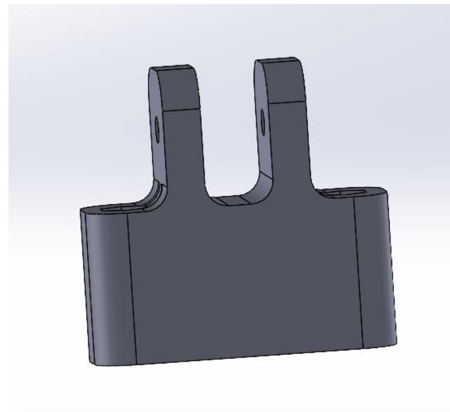
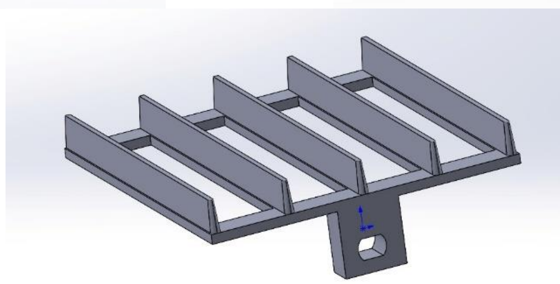


Figure: Initial Design for wooden box

These components were used in the process of installing the robot and providing it with weight. Through them, different parts of the system were interconnected to ensure proper functioning. These parts were manufactured using 3D printing technology.

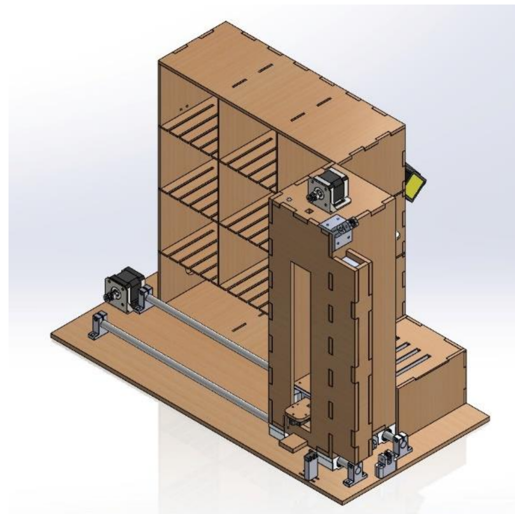


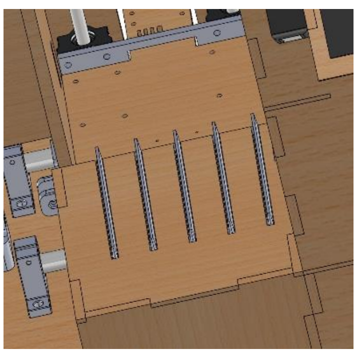




This forklift was utilized for the pulling and proper storage of

Wooden Box Design After Connecting to Other Components





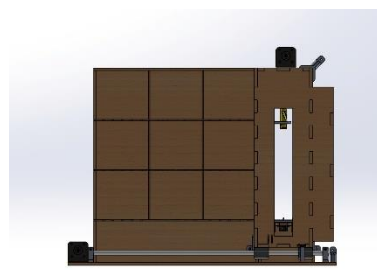
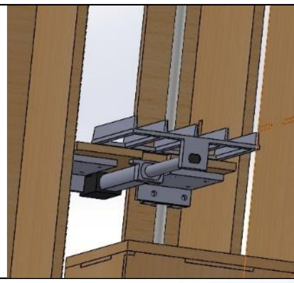
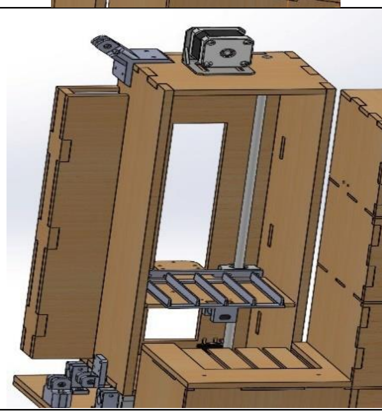
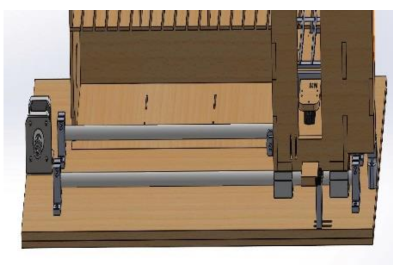


Figure 1: Final Result for wooden box

4.1.2 Movement Mechanism

	the movement of z-axis using DC motor
	the movement of y-axis using stepper motor
	the movement of x-axis using stepper motor

Processing Units and Used Devices

4.2.1 Arduino MEGA 2560 Chinese CH340 (Without Cable)

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560[1]. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller.

We utilized one Arduino MEGA which was responsible for driving the Nextion Screen, barcode scanner, motors., and was driven by the ESP8266.

.

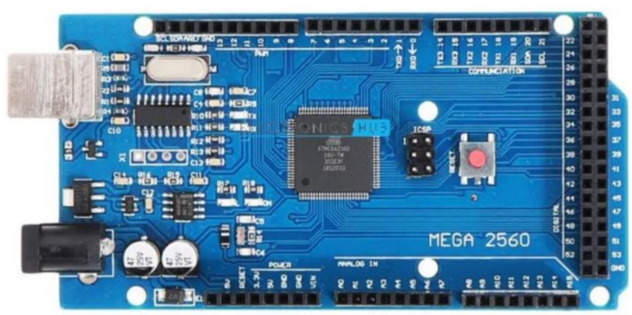


Figure: Arduino MEGA

4.2.2 ESP8266 NodeMCU

The NodeMCU (Node Microcontroller Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (Wi-Fi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things(IoT) projects of all kinds.
We utilized ESP8266 as the main chip to communicate with the internet, and to control screen, barcode scanner and motors with controlling the Arduino MEGA.

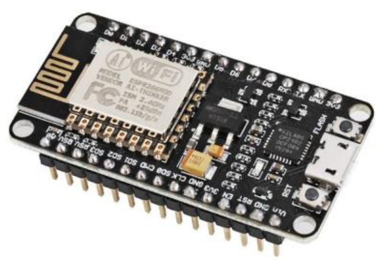


Figure: ESP8266 NodeMCU

4.2.3 Power Supply

We used a DC 12V 4.16A 45W power supply to obtain 12 volts for operating two stepper motors and one DC motor.

Figure: Power Supply

4.2.1 Buck Converter

The LM2596 DC-DC Buck Converter is a popular choice, offering a versatile solution with an input voltage range of 4V-35V and an adjustable output voltage between 1.23V and 30V. With a maximum input current of 3A and an impressive 92% conversion efficiency, this compact and lightweight module (45*20*12mm *(about 0.47 in)*, 10g) ensures reliable performance. Operating at a switching frequency of 150KHz, it minimizes output ripple (up to 30mA). The device excels in load regulation (±0.5%) and voltage regulation (±2.5%), making it suitable for various applications in temperatures ranging from -40 to +85 degrees Celsius. The package includes 1 x Buck Converter, providing a comprehensive solution for voltage regulation needs.
Both the display and the controller receive a 5-volt power supply, making it the chosen voltage for our project.

Figure: Buck Converter

4.2.2 NEMA 17 Stepper Motor

The NEMA 17 stepper motor is highly praised for its small size and remarkable torque performance, making it a commonly chosen option across diverse applications. It functions with an exact step angle of 1.8 degrees per step, requiring 200 steps for a complete revolution. Equipped with two windings, it can manage currents of up to 3.5 A and supports voltage inputs in the range of 3 to 12 volts.

The NEMA17 Stepper Motor model 17HS8401, featuring a 1.8A rated current and a 4.5x23mm shaft size, demonstrates specific attributes. It maintains a step angle accuracy of approximately ±5% in full step mode under no-load conditions. The motor's resistance accuracy falls within the range of ±10%, and its inductance accuracy is around ±20%. The temperature rise is limited to a maximum of 80 degrees Celsius, determined by the higher value between the rated current and both phases being on. The motor operates within an ambient temperature range of -20 to +50 degrees Celsius. Electrically, it exhibits an insulation resistance of at least 100MΩ at 500VDC and can withstand an insulation strength test of 500VAC for one minute.

In our project, we employed two of them to manage the robot's movement, with one dedicated to controlling motion along the Y-axis and the other for movement along the X-axis. Each unit is linked to an A4988 Stepper Motor Microstep driver, establishing a connection with the controller.

Figure: NEMA17 Stepper Motor

4.2.6 A4988 Stepper Motor Microstep driver

We incorporated this driver into our system as an extra component layer positioned between the Arduino Microcontroller and the Motor, offering a range of functionalities such as current supply and regulation, overheating protection, finer steps for enhanced precision, and the required torque. We configured the driver to operate at a current of 3.3 A and with a step size of 1/8. The motor is linked and managed via three pins: Enable, direction, and step pin (PUL). The driver receives power from our power supply. The A4988 Stepper Motor Microstep Driver employs a method known as micro stepping to achieve precision within this system, which is a vital aspect for its effective performance. The original NEMA17 Stepper Motor has a basic resolution of 200 steps per revolution, corresponding to a movement of 1.8 degree. This inherent resolution is relatively modest. The driver intervenes by subdividing the pulses, allowing the generation of 16 pulses for each motor step. Consequently, instead of providing 200 revolutions, each revolution is now divided into 200 * 16, resulting in 3200 micro steps. This proves extremely advantageous for improving precision and ensuring smooth operation, we used 2.

- **Steps per Revolution:**

  $\text{Steps per Revolution} = \frac{\text{360 degrees}}{\text{Step Angle}}$

  Where the step angle (Step Angle) for a NEMA17 motor is typically 1.8 degrees.

- **Microsteps per Revolution:**

  $\text{Microsteps per Revolution} = \text{Steps per Revolution} \times \text{Microstepping Factor}$

  Where the microstepping factor is 16 in this case.

Therefore, these can be expressed with the following equations:

$$\text{Steps per Revolution} = \frac{360}{1.8} = 200\,\text{steps}$$

$$\text{Microsteps per Revolution} = 200 \times 16 = 3200\,\text{microsteps}$$

Figure: A4988 Stepper Motor Microstep Driver

4.2.7 12V DC Linear Actuator 100mm Stroke 15mm/S 60N Force LA-YR Type

The JS35A series is composed of compact DC motor-driven linear actuators designed to push or pull loads along their stroke. The DCL4013 model, with a 100mm stroke length, is capable of moving at speeds reaching 15mm/s and exerting a force of 60N at lower speeds. It is equipped with built-in limit switches, enabling safe operation within the actuator's full stroke range without causing damage. Included in the package are 2 mounting brackets, 2 M4 screws (12mm), and 2 M4 locking nuts, providing convenience for easy installation and versatility across different applications.

We utilized this DC motor for the movement of the robot along the Z-axis. Alongside it, we employed the L298N Dual H-Bridge Motor Driver Module Board to establish the connection with the controller.

4.2.8Nextion Touch Screen

The Nextion Touch Screen is positioned at the front of our system. By utilizing this intelligent display, the login procedure is initiated, followed by the capability to add, retrieve, or modify goods. Alerts are generated in case the location is empty upon request, full when an item is placed, or if the product is unrecognized in the database. The touchscreen interface allows for easy adjustment and interaction.

Nextion is a Human Machine Interface （HMI） solution combining an onboard processor and memory touch display with Nextion Editor software for HMI GUI project development. Using the NEXTION Editor software, you can quickly develop the HMI GUI by drag-and-drop components (graphics, text, button, slider etc.) and ASCII text-based instructions for coding how components interact at the display side. Nextion HMI display connects to peripheral MCU via TTL Serial (5V, TX, RX, GND) to provide event notifications that peripheral MCU can act on, the peripheral MCU can easily update progress and status back to Nextion display utilizing simple ASCII text-based instructions.

The Intelligent Series products have more powerful hardware in terms of MCU, Flash storage and SRAM compared with Basic Series and Enhanced Series. What’s more, the audio, video, and animation play functions enrich the user’s project HMI interaction. The Intelligent Series supports advanced software features and functions such as transparent components, page loading effect, component Move and Drag.

4.2.9 L298N Dual H-Bridge Motor Driver Module Board

The L298N Dual H-Bridge Motor Controller, a popular choice for motor speed and direction control, exhibits versatility in applications such as LED arrays, relays, and solenoids. Featuring a sturdy design with a robust heat sink, it can effectively handle motors within the 5-35V power supply range, supporting a maximum current of 2A. With the L298N chip enabling a double H-bridge drive, precise motor control is ensured. Operating at a logical voltage of 5V, the controller has a logical current range from 0mA to 36mA. Its compact size of 43 x 43 x 27mm and lightweight of 30g, coupled with a storage temperature range of -20 to +135 degrees Celsius, make it a versatile and powerful motor driver suitable for various applications.
Utilizing this driver allows us to control the motor, enabling movement in both directions and implementing a safety feature to restrict motion.

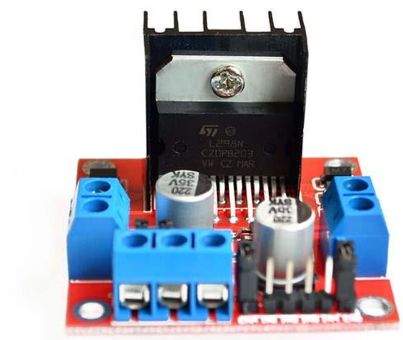


Figure: L298N Dual H-Bridge Motor Driver Module Board

4.2.10 42CH Stepper Motor Driver Shield Expansion Board for A4988

The 42-stepper motor driver expansion board uses on-board DIP switch, and we can easily adjust the drive segments, which is so convenient for us to operate. It is a terminal power connector, and we can easily connect the drive power supply, which provides convenience for our larger, it’s compatible with 12/24V drive scheme, we used 2.

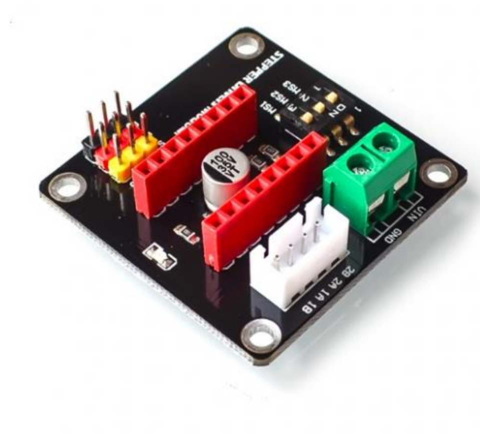


Figure: 42CH Stepper Motor Driver Shield Expansion Board for A4988

4.2.11 Limit Switch KW11-N KW12 5A/250V 3 Port Roller

The application of the KW11-N or KW12 5A/250V 3 Port Roller Limit Switch typically involves its use as an endpoint or sensor in various electromechanical systems. These limit switches are commonly employed to identify the physical boundaries of motion in machinery or devices. Equipped with a roller for smoother contact with moving parts, the switch ensures reliability and precision in the detection process. With a current rating of 5A and a voltage rating of 250V, it can manage moderate electrical loads. Integration of the KW11-N or KW12 limit switch is frequently observed in automation, robotics, and machinery applications where accurate positioning or endpoint detection is crucial for the correct operation of the system. In our system, our dependence is on time, specifically, a timeframe of 8 seconds.

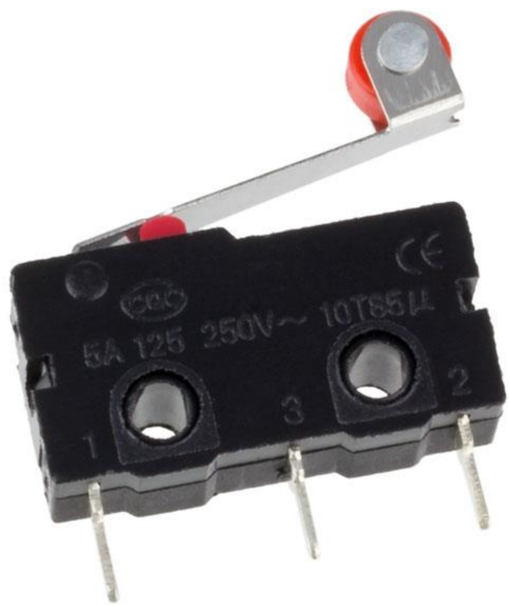


Figure: Limit Switch KW11-N KW12 5A/250V 3 Port Roller

4.2.12 SH-400 TTL 2D Barcode And QR Scanner

A Barcode and QR Scanner is a tool created to capture and interpret data embedded in barcodes and Quick Response (QR) codes

Figure: Barcode And QR Scanner

4.2.13 GT2 Timing Pulley 16 Teeth Bore 5mm

GT2 Timing Pulley 16teeth (16 teeth) Aluminum Bore 5mm fit for GT2 belt Width 6mm.

The GT2 pulley with 16 teeth or grooves is one of the best choices for 3d printer construction. Compared with the T2.5 or MXL timing pulley, the GT2 pulley tooth profile is anti-backlash. Why? The rounded teeth of this tooth and belt are engaged together longer providing a smoother transition between teeth, 3D printer accessories 2GT 16 teeth synchronous wheel pulley wheel Perlin passive idler pulley wheel bore 3mm, we used two of them.

Figure: GT2 Timing Pulley 16 Teeth Bore 5mm

4.2.14 GT2 16 Teeth Idler Pulley Bore 3mm

3D printer accessories 2GT 16 teeth synchronous wheel pulley wheel Perlin passive idler pulley wheel bore 3mm, we used two of them.

Figure: GT2 16 Teeth Idler Pulley Bore 3mm

4.2.15 IR Infrared Adjustable Proximity Sensor 3-80cm E18-D80NK 5V

This photoelectric sensor comprises a transmitter and receiver set within one unit, allowing for adjustable detection distances based on specific needs. The sensor is characterized by its features, including a detection range, resilience to visible light interference, cost-effectiveness, ease of assembly, and user-friendly design. It is particularly suitable for applications involving robot obstacle avoidance and assembly line processes. In terms of electrical specifications, the sensor operates at a voltage of 5VDC, draws a current of 100mA, and has a detection distance ranging from 3 to 80 centimeters. These specifications make it versatile for deployment in various situations.

4.2.17 1meter Rubber GT2-6mm Open Timing Belt ,6mm Width

Figure: 1meter Rubber GT2-6mm Open Timing Belt ,6mm Width

How The System Works

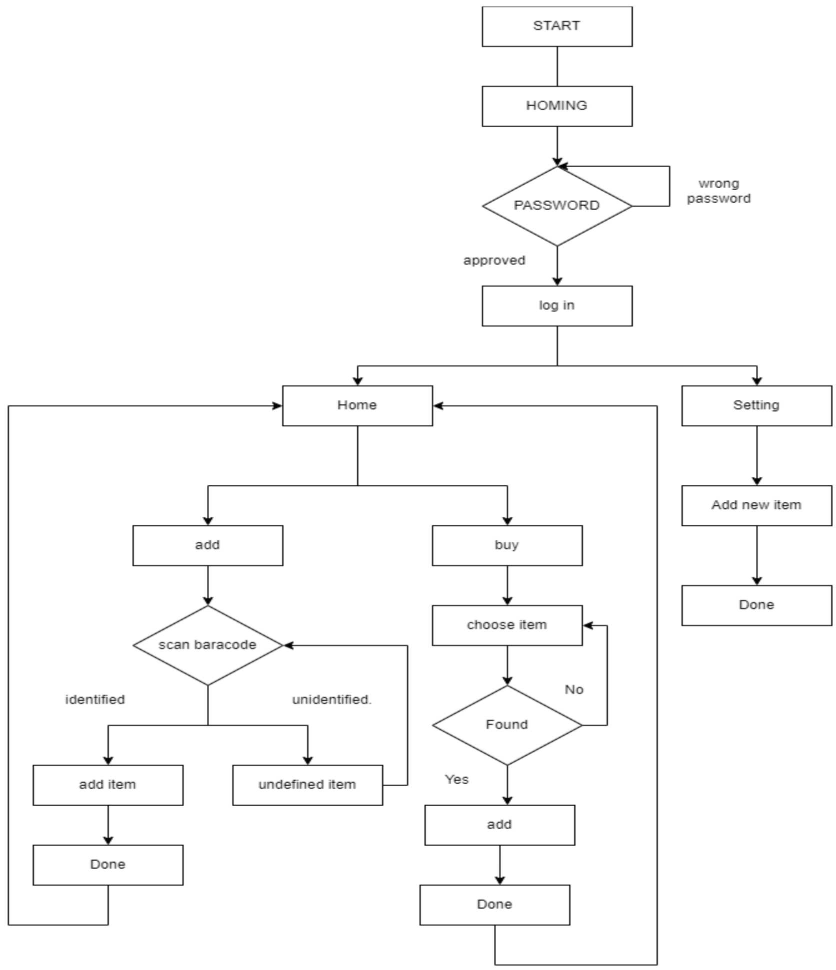


Figure 22: System flowchart

5 Results & Discussion

5.1 Results

The system in place functions as an exceptional automated storage solution, showcasing heightened efficiency in organization and streamlined operations. It provides users with seamless organization and effortless modifications, catering to both automated and manual control preferences. Its adaptability to diverse user requirements offers a flexible method for handling stored items. With its intuitive design and user-friendly features, it ensures an optimized and efficient storage experience without the necessity for a mobile application.

5.2 Discussion

The system introduces fresh and innovative elements when juxtaposed with traditional concepts and methods. While currently in the deployment phase, it offers a solution that holds the promise of substantially improving storage efficiency and accessibility. Looking forward, this pioneering system holds the potential to transform storage practices, potentially benefiting various industries and sectors by optimizing inventory management and simplifying access to stored items.

6 Conclusion & Future Work

6.1 Conclusion

We foresee that this concept harbors the capacity to significantly impact storage practices and may emerge as a highly sought-after product down the line. Its uncomplicated design and ease of use are positioned to cater to a broad spectrum of users. This overarching aim has been the foundational aspiration motivating our project right from its inception.

6.2 Future Work

- Enhanced Accessibility Features: Introduce user-friendly functionalities ensuring the system accommodates diverse users, including those with specific accessibility requirements.

- Exploration of Advanced Security Measures: Explore and integrate state-of-the-art security protocols to maintain data integrity and strengthen system safety against potential risks.

- Customization Options: Create adaptable features empowering users to personalize the storage system, enhancing user experience and flexibility.

- Energy Efficiency Enhancements: Investigate and implement energy-conserving attributes to reduce power usage, promoting sustainability in system operations.

- Robustness Testing with Alternative Materials: Conduct thorough examinations using different materials to ensure durability, portability, and resilience across various situations.

- Partnerships for Implementation: Establish collaborations with relevant industries to facilitate the practical application and rollout of the storage system.

- Security Integration: Enhance security measures by incorporating advanced AI algorithms for early detection and response to potential security breaches, securing stored data.

- Mobile Application Development: Create a mobile app to streamline system control and accessibility, enabling remote inventory management and enhancing operational efficiency and convenience.

A Main Code for Arduino Mega

```
#include <EEPROM.h>

#define NXSerial Serial1
#define BCSerial Serial2

#define ProxSensorPin 6
#define floarHightMM 120
#define stepPerMMx 100
#define stepPerMMy 100
#define speedDelayX 100
#define speedDelayY 100

#define PULx 2 //define Pulse pin
#define DIRx 3 //define Direction pin

#define PULy 4 //define Pulse pin
#define DIRy 5 //define Direction pin

#define IN1 9
#define IN2 10
#define limitSwitchPinX 7
#define limitSwitchPinY 8

#define ReceiveShelfX 0
#define ReceiveShelfY 1

#define UnderSelfMM 1
#define aboveShelfMM 11
#define srokeDelay 8000

#define homeingSpeedDelay 100
```

```
String BCdataIn = "";
String NXdataIn = "";
bool ProxSensorReading;
const int maxBarcodeNamesQty = 20;
String BCnames[maxBarcodeNamesQty] = {"001", "002", "003", "004", "005", "006", "007", "008", "009"};
int PsitionsOfBCnames[maxBarcodeNamesQty][maxBarcodeNamesQty]  = {{135, 1} , {268, 1}, {404, 1}, {134, 125}, {268, 125}, {404, 125},
{134, 250}, {268, 250}, {404, 250}};
bool addFlag = 0;

unsigned long lastXmm;
unsigned long lastYmm;

unsigned long Xtimer1 = 0;
unsigned long Ytimer1 = 0;

void setup() {

  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode (PULx, OUTPUT);
  pinMode (DIRx, OUTPUT);
  pinMode (PULy, OUTPUT);
  pinMode (DIRy, OUTPUT);
  pinMode (limitSwitchPinX, INPUT_PULLUP);
  pinMode (limitSwitchPinY, INPUT_PULLUP);
  Serial.begin(115200);
  BCSerial.begin(115200);
  NXSerial.begin(9600);
  BCSerial.setTimeout(50);
  NXSerial.setTimeout(50);
  goToPageNX("8");
  homeingMachine();
  //extendForkLift();
  //reduceForkLift();
  resetMem();
  readMem();
```

```
  goToPageNX("0");

}

void loop() {

  if (Serial.available()) {
    String desMM = Serial.readString();
    int xmm = desMM.toInt();
    Serial.print("x:");
    Serial.println(xmm);
    while (Serial.available() == 0) {
      delay(10);
    }
    desMM = Serial.readString();
    int ymm = desMM.toInt();
    Serial.print("goTo:");
    Serial.print(xmm);
    Serial.print(",");
    Serial.println(ymm);
    goToRelativeMM(xmm, ymm);
  }

  if (NXSerial.available()) {
    NXdataIn = NXSerial.readString();
    Serial.println("NXdataIn>>" + NXdataIn);
    if (NXdataIn.indexOf("add") >= 0) {
      addFlag = 1;
      goToPageNX("5");
    }
    else if (NXdataIn.indexOf("sell") >= 0) {
      processingProductFromStoreShelf(NXdataIn[NXdataIn.indexOf("sell") + 4] - '0');
      goToPageNX("7");

    }
  }
```

```
  if (addFlag == 1 ) {
    if (BCSerial.available()) {
      BCdataIn = BCSerial.readString();
      Serial.println("BCdataIn>" + BCdataIn);
      BCdataIn.trim();
      sendStrToNX("page5.barcode", BCdataIn);
      goToPageNX("12");


      while (NXSerial.available() == 0) {

      }
      NXdataIn = NXSerial.readString();
      Serial.println("NXdataIn>>" + NXdataIn);
      if (NXdataIn.indexOf("exit") >= 0) {
        addFlag = 0;
      }
      else if (NXdataIn.indexOf("ok") >= 0) {
        processingProductFromReceiveShelf(NXdataIn[NXdataIn.indexOf("ok") + 2] - '0');
        goToPageNX("7");
        addFlag = 0;
      }


    }
  }
}
void processingProductFromReceiveShelf(int i) {
  goToRelativeMM(ReceiveShelfX, ReceiveShelfY - UnderSelfMM);
  extendForkLift();
  goToRelativeMM(ReceiveShelfX, ReceiveShelfY + aboveShelfMM);
  reduceForkLift();
  goToRelativeMM(PsitionsOfBCnames[i][0] , PsitionsOfBCnames[i][1] + aboveShelfMM);
  extendForkLift();
  goToRelativeMM(PsitionsOfBCnames[i][0] , PsitionsOfBCnames[i][1] - UnderSelfMM);
```

```
  EEPROM.put(i, 1);
  reduceForkLift();
  goToRelativeMM(ReceiveShelfX, ReceiveShelfY - UnderSelfMM);
}
void processingProductFromStoreShelf(int i) {
  goToRelativeMM(PsitionsOfBCnames[i][0], PsitionsOfBCnames[i][1] - UnderSelfMM);
  extendForkLift();
  goToRelativeMM(PsitionsOfBCnames[i][0], PsitionsOfBCnames[i][1] + aboveShelfMM);
  reduceForkLift();
  EEPROM.put(i, 0);
  goToRelativeMM(ReceiveShelfX, ReceiveShelfY + aboveShelfMM);
  extendForkLift();
  goToRelativeMM(ReceiveShelfX, ReceiveShelfY - UnderSelfMM);
  reduceForkLift();
}

void extendForkLift() {
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
  delay(srokeDelay);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 0);
}
void reduceForkLift() {
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 0);
  delay(srokeDelay);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 0);
}

void goToRelativeMM(int Xmm, int Ymm) {
  bool xMotorDirction;
  bool yMotorDirction;
  Serial.print("goToRelativeMM(");
  Serial.print(Xmm);
```

```
Serial.print(",");
Serial.print(Ymm);
Serial.println(")");
if (Xmm > lastXmm) {
  xMotorDirction = LOW;
}
else if (Xmm < lastXmm) {
  xMotorDirction = HIGH;
}
if (Ymm > lastYmm) {
  yMotorDirction = HIGH;
}
else if (Ymm < lastYmm) {
  yMotorDirction = LOW;
}
digitalWrite(DIRx, xMotorDirction);
digitalWrite(DIRy, yMotorDirction);


long difXmm = Xmm - lastXmm;
difXmm = abs(difXmm);
lastXmm = Xmm;
long difYmm = Ymm - lastYmm;
difYmm = abs(difYmm);
lastYmm = Ymm;
long pulseCountX = difXmm * stepPerMMx;
long pulseCountY = difYmm * stepPerMMy;

/* Serial.print("pulseCountX to move:");
  Serial.println(pulseCountX);
  Serial.print("pulseCountY to move:");
  Serial.println(pulseCountY);*/

while (pulseCountX != 0 || pulseCountY != 0) {
  if (pulseCountX != 0) {
    if (micros() - Xtimer1 > speedDelayX) {
```

```
      digitalWrite(PULx, HIGH);
      if (micros() - Xtimer1 > speedDelayX * 2) {
        Xtimer1 = micros();
        pulseCountX--;
      }
    }
    else {
      digitalWrite(PULx, LOW);
    }

  }

  if (pulseCountY != 0) {
    if (micros() - Ytimer1 > speedDelayY) {
      digitalWrite(PULy, HIGH);
      if (micros() - Ytimer1 > speedDelayY * 2) {
        Ytimer1 = micros();
        pulseCountY--;
      }
    }
    else {
      digitalWrite(PULy, LOW);
    }
  }

 }
}

void homeingMachine() {
  Serial.println("Homeing ForkLift!");
  reduceForkLift();
  Serial.println("Homeing X Axis And Y Axis..");
  int limitSwitchReadingX = digitalRead(limitSwitchPinX);
  int limitSwitchReadingY = digitalRead(limitSwitchPinY);
  digitalWrite(DIRx, HIGH);
  digitalWrite(DIRy, LOW);
```

```
while (limitSwitchReadingX == 1 || limitSwitchReadingY == 1) {

  if (limitSwitchReadingX == 1) {
    if (micros() - Xtimer1 > homeingSpeedDelay) {
      digitalWrite(PULx, HIGH);
      if (micros() - Xtimer1 > homeingSpeedDelay * 2) {
        Xtimer1 = micros();
        limitSwitchReadingX = digitalRead(limitSwitchPinX);
      }
    }
    else {
      digitalWrite(PULx, LOW);
    }

  }
  else {
    limitSwitchReadingX = digitalRead(limitSwitchPinX);
  }

  if (limitSwitchReadingY == 1) {
    if (micros() - Ytimer1 > homeingSpeedDelay) {
      digitalWrite(PULy, HIGH);
      if (micros() - Ytimer1 > homeingSpeedDelay * 2) {
        Ytimer1 = micros();
        limitSwitchReadingY = digitalRead(limitSwitchPinY);
      }
    }
    else {
      digitalWrite(PULy, LOW);
    }
  }
  else {
    limitSwitchReadingY = digitalRead(limitSwitchPinY);
  }
```

```
  /*

    if (limitSwitchReadingX == 1) {
      digitalWrite(DIRx, HIGH);
      digitalWrite(PULx, HIGH);
      delayMicroseconds(HomeSpeedDelay);
      digitalWrite(PULx, LOW);
      delayMicroseconds(HomeSpeedDelay);
      limitSwitchReadingX = digitalRead(limitSwitchPinX);
    }
    else {
      limitSwitchReadingX = digitalRead(limitSwitchPinX);
    }
    if (limitSwitchReadingY == 1) {
      digitalWrite(DIRy, LOW);
      digitalWrite(PULy, HIGH);
      delayMicroseconds(HomeSpeedDelay);
      digitalWrite(PULy, LOW);
      delayMicroseconds(HomeSpeedDelay);
      limitSwitchReadingY = digitalRead(limitSwitchPinY);
    }
    else {
      limitSwitchReadingY = digitalRead(limitSwitchPinY);
    }
  */
 }
 lastXmm = 0;
 lastYmm = 0;
 delay(1000);
 goToRelativeMM(8, 0);
 lastXmm = 0;
 lastYmm = 0;
 Serial.println("Homeing X and Y Done!");

}
```

```
void sendStrToNX(String atr, String str_) {
  NXSerial.print(atr + ".txt=\"" + str_ + "\"");
  NXSerial.write(0xff);
  NXSerial.write(0xff);
  NXSerial.write(0xff);

}
void goToPageNX(String page) {
  Serial.print("goToPageNX");
  Serial.println(page);
  NXSerial.print("page " + page);
  NXSerial.write(0xff);
  NXSerial.write(0xff);
  NXSerial.write(0xff);
}

void setMem() {
  for (int i = 0; i < maxBarcodeNamesQty; i ++) {
    EEPROM.put(i, 0);
  }
}
void resetMem() {
  for (int i = 0; i < maxBarcodeNamesQty; i ++) {
    EEPROM.put(i, 0);
  }
}
void readMem() {
  bool memVal;
  for (int i = 0; i < maxBarcodeNamesQty; i++) {
    EEPROM.get(i, memVal);
    Serial.print("cell " + String(i) + ":");
    Serial.println(memVal);
  }
```

Arduino UNO Test Code

```
#include <SoftwareSerial.h>
SoftwareSerial BCSerial(7, 8);
void setup()
{
BCSerial.begin(115200); //baud rate
Serial.begin(9600);

}

void loop()
{
if(BCSerial.available())
Serial.write(BCSerial.read());

}
```

ESP32 Test Code

```
//green Barcode TX connect to pin 16 on ESP32

//white Barcode RX connect to pin 17 on ESP32

HardwareSerial BCSerial(1);

void setup(){

Serial.begin(115200);

BCSerial.begin(115200, SERIAL_8N1, 16, 17); // rx,tx

}

void loop(){

if(BCSerial.available())

Serial.write(BCSerial.read());

}
```

Arduino Mega Code

```
#define BCSerial Serial3

void setup(){

Serial.begin(115200);

BCSerial.begin(115200);

}

void loop(){

if(BCSerial.available())

Serial.write(BCSerial.read());
```

Screen’s code for login

```
page 1
if(t0.txt=="4321")
{
  page 1
}else
{
  t0.txt=""
}
```

Screen’s code for setting

```
page 10
if(t0.txt=="6789")
{
  page 10
}else
```

```
{
  t0.txt=""
}
```

Screen’s code for items

```
address.val=0
for(sys0=1;sys0<=9;sys0++)
{
  address.val=sys0*10
  repo buff.txt,address.val
  if(buff.txt=="")
  {
    b[sys0].txt="NotDefined"
  }else
  {
    repo b[sys0].txt,address.val


}
}
address.val=500
for(sys0=1;sys0<=9;sys0++)
{
  repo buff.txt,address.val
  if(buff.txt=="")
  {
    b[sys0].bco=63488
    tsw b[sys0],0
  }else if(buff.txt=="0")
  {
    b[sys0].bco=63488
    tsw b[sys0],0
  }else if(buff.txt=="1")
  {
    b[sys0].bco=50712
    tsw b[sys0],1
```

```
  }
 address.val++
}
```

Screen’s code for adding new item

```
if(page1.lastPage.val==10)
{
 repo t1.txt,page10.buttonID.val*10
}
```