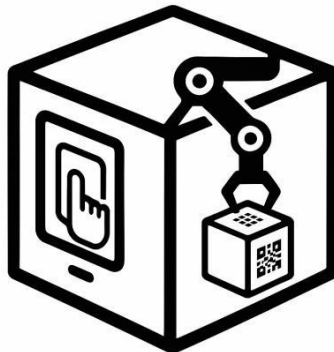




An-Najah National University
Faculty of Engineering & Information Technology
Department of Computer Engineering

Graduation Project II



Ahmad hanani & Abd albaset rajab

Supervisor: Dr. Hekmat darawsheh

Presented in partial fulfillment of the requirements for the
Bachelor degree in Computer Engineering.

june, 2025

Acknowledgement

In the name of Allah, the Most Gracious, the Most Merciful. We are inundated with gratitude to the blessings that have enabled us to accomplish this hardware project successfully. We give our sincere thanks to the Almighty for bestowing us with the strength, capacity, and information needed to accomplish this. Our sincerest appreciation to our supervisor, Dr. Hekmat darawsheh , for the immense value he gave to us through his support and guidance, which added a lot to the direction and success of our project. His experience and expertise have led us to a conclusion that we are glad to submit. We also thank the valuable suggestions from the erudite faculty members of the Department of Computer Engineering, whose inputs have made our learning process richer. We thank our families and friends, whose unflinching support and belief in our abilities have been a source of inspiration and motivation. Your contributions have been a pillar along the way. We hope that our efforts and the learning we have acquired will be to the good of our society and a good contribution to humanity.

Disclaimer

This report was written by students at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Table of Contents

Acknowledgement	II
Disclaimer.....	III
Abstract.....	1
Introduction.....	2
1.1 Statement of the problem	2
1.2 Objectives of the work.....	2
1.3 Scope of the work.....	2
1.4 Significance of our work.....	3
Constraints & Earlier coursework.....	4
2.1 Constraints	4
2.2 Earlier coursework.....	4
Literature Review	5
Methodology	6
4.1 Standards and specifications.....	6
4.2 Hardware components.....	6
Project structure	6
Arduino Mega 2560	7
Raspberry pi 4.....	8
Stepper motor.....	8
IR sensor	10
Keypad	10
I2C 20X4 LCD.....	11
RC522 (RFID).....	11
Solenoid lock.....	12
H-Bridge (L298n).....	12
Power Supply.....	13
LM2596 Step Down Buck Converter Module.....	14
Limit Switch	14
Robotic Arm	15
Idler Pulley	15
GT2 Timig Belt.....	16
Esp8266.....	17

Servo Motor (Mg996r)	17
Sensor Dht11	18
Fan 12v	18
Lead & Lead Screw	19
Project rule	20
Wires	21
4.3 Hardware implementation.....	21
4.4 Project circuit.....	23
4.5 Flowchart of QR Robot: From Product Selection to Delivery	24
24	
Software implementation	25
4.6.1 QR Code Detection.....	25
4.6.2 Determining the Location of Each QR Product.....	27
Results and Discussion	30
Conclusion and Recommendations	31
References	Error! Bookmark not defined.

Table of Figures

Figure 4.1: Creality K1	7
Figure 4.2: Arduino Mega 2560.....	7
Figure 4.3: Raspberry pi 4.....	8
Figure 4.4: Stepper motor, and A4988 driver.....	9
Figure 4.5: IR sensor.....	10
Figure 4.6: Keypad.....	10
Figure 4.7: I2C 20X4 LCD	11
Figure 4.8: RC522 (RFID).....	12
Figure 4.9: Solenoid lock.....	12
Figure 4.10: H-bridge.....	13
Figure 4.11: Power Supply.....	13
Figure 4.12: Step Down Buck Converter Module.....	14
Figure 4.13: Limit Switch.....	14
Figure 4.14: Robotic Arm	15
Figure 4.15: GT2 20T Dual Bearing Aluminum Idler Pulley – 5mm Bore.....	16
Figure 4.16: GT2 Timig Belt	16
Figure 4.17: ESP8266	17
Figure 4.18: Mg996r	18
Figure 4.19: Dht11	18
Figure 4.21: Lead Screw	19
Figure 4.22: stainless steel round bar	20
Figure 4.23: Project rule.....	20
Figure 4.24: Wires	21
Figure 4.25 :Full Design	22
Figure 4.26: Project circuit.....	23
Figure 4.27: Flowchart of QR Robot.....	24
Figure 4.28: QR Code Detection.....	26
Figure 4.29:Code Of QR Code Detection	26
Figure 4.31: Limit Switch In Robot Arm.....	27
Figure 4.32: Code Of Limit Switch In Robot Arm.....	28

Abstract

Robotic QR Arranger is a cutting-edge automated solution that increases the efficiency of the warehouse using QR codes and image processing to accurately sort and retrieve products. With warehouse automation playing a pivotal role in e-commerce, manufacturing, and distribution, this project reduces human error, accelerates sorting, and optimizes inventory management, making operations more independent and cost effective. The project is significant as it reduces operating costs, enhances accuracy in retrieval, and boosts warehouse productivity through the elimination of manual intervention, hence accelerating processing and minimizing errors. Its integration with robotics enhances efficiency for high-scale inventory management industries. Some of the key characteristics of the project include accurate identification of the product location, seamless integration of computer vision and robotics, simple order placement and tracking using a straightforward interface, and utilization of artificial intelligence to enable improved decision-making. The main objectives are to develop an automated retrieval system, improve the tracking of the stock, reduce human interaction, and integrate AI with robotics. Development work entails system analysis, architecture, image processing algorithm for QR recognition, programming of the robotic arm, and overall testing. While other warehouse automation software exists, Robotic QR Arranger is different in that it uses both QR technology and image processing to gain precise localization, offering a novel and cost-efficient solution for smart logistics.

Chapter 1

Introduction

The QR Robot is a smart, interactive product recognition and delivery system that uses QR codes. It operates in an enclosed cube where products are placed, each with its own QR code. The user can request a product of their choice through a mobile app or keypad. The camera reads products; it is processed in the Raspberry Pi to determine the right product, and then a robotic arm picks it up and offers it to the customer. It is an automation, computer vision, and robotics concept that can be applied to warehouses, smart shops, or future service systems.

1.1 Statement of the problem

Traditional product retrieval from stock or retail installations are often reliant on manual search, which can be time-consuming, wasteful, and prone to human inaccuracy. They further lack intelligent interaction and automation. We therefore propose a three-dimensional automated system that uses QR code identification and robot movement in detecting and offering products in an effective and interactive way.

1.2 Objectives of the work

The goal of our project is to create an intelligent and interactive system that can identify and retrieve products using QR codes inside a cube-shaped structure. This project aims to combine automation, image processing, and robotics to deliver a modern solution for product selection. It can be used in public or commercial spaces such as smart stores, exhibitions, or automated storage systems. The main goal is to enhance user experience through smart interaction, efficiency, and innovation.

1.3 Scope of the work

In order to achieve the project goals, we selected required components such as the Raspberry Pi, Arduino, robotic arm, camera, servo motors, and stepper motors. Each part of the system was developed independently: we used Python with the OpenCV and Pyzbar libraries for QR code recognition and product positioning, and the Arduino was programmed to drive the robotic arm. A keypad and LCD screen user interface and a mobile app for product choice were also developed. Finally, all the elements were integrated into a three-dimensional cube shape system capable of identifying and offering the defined product.

1.4 Significance of our work

Our project offers an innovative and efficient product identification and retrieval system through automation and QR code technology. It offers a new application that combines robotics, computer vision, and user-friendly interfaces in order to enhance the user experience. The system is usable in a number of fields that range from smart retail, inventory management, and public service applications. The system's interactive functionality and independent operation make it suitable for modern environments like smart stores, exhibitions, and automated kiosks.

Chapter 2

Constraints & Earlier coursework

2.1 Constraints

During the development of the project, we faced several challenges. One of the main issues was the high cost of CNC systems and robotic arms, which significantly affected the project's budget. Another major challenge was the difficulty in accurately detecting and picking up products that are tilted at certain angles. This is because handling such positions requires a fourth and fifth axis and smooth degrees of movement, which adds complexity and cost to the robotic system.

2.2 Earlier coursework

Our academic coursework played a significant role in equipping us with the knowledge and skills necessary for this project. Courses such as Digital Circuits Design I, II, and III helped us understand the fundamentals of hardware control and system integration. The Algorithms and Mathematics courses were essential in developing the logic for identifying and determining the location of products. Additionally, the Digital Image Processing course provided the foundation for QR code recognition and image analysis. The Python programming course was also crucial, as it enabled us to implement the core functionality of the system using tools like OpenCV and Pyzbar.

Chapter 3

Literature Review

The recent years have seen an upsurge in the development of smart systems that support automation in recognizing, selecting, and retrieving physical goods. Raspberry Pi and Arduino are among the popular technologies utilized when combined with robotic arms and camera modules to mount product recognition systems based on QR codes and computer vision. Such systems are used extensively in use cases like intelligent inventory management, vending machines, and warehouse automation. Libraries like OpenCV for computer vision and QR scanner libraries like Pyzbar can be used to make a system read product labels and identify them, and robotic arms perform actions based on positional information. Real-world applications and research have established that using vision alongside motion control enables independent picking of individual items, simplifying operations and reducing labor. Our solution builds on these ideas with the introduction of an intimate cube-based system that recognizes products through a camera atop, locates them via QR code scanning, and moves the desired product through a robot arm to present it—guided through mobile or touch.

Chapter 4

Methodology

4.1 Standards and specifications

At the beginning of the project, we repurposed an existing frame from a 3D printer machine, specifically the Creality K1, and modified its structure to suit our cube-based product retrieval system. We then conducted initial testing to ensure successful QR code detection and accurate localization of each product on the platform's surface using a top-mounted camera and image processing techniques. After validating the vision system, we began assembling the electronic components, including the Arduino microcontroller, solenoid lock, power supply unit, motors, and the entire motion control system. All components were selected based on availability, compatibility, and performance reliability. The control logic was implemented using Python on the Raspberry Pi and the Arduino IDE for microcontroller programming. Each element of the system will be described in detail in the following sections.

4.2 Hardware components

In this section we are going to talk about all the hardware components we used, and why we used each one of them:

Project structure

We used the frame of a 3D printer, specifically the Creality K1, as the mechanical base for our project. This frame was chosen because of its precise motion system, strong metal structure, and overall mechanical stability. It provided a suitable platform in terms of size and movement capabilities. We modified and adapted the frame to meet the needs of our system by integrating a robotic arm, a camera, and a smart control system to perform the task of identifying and picking products based on QR code recognition.



Figure 4.1: Creality K1

Arduino Mega 2560

The Arduino Mega 2560 is a versatile microcontroller board that serves as an extended version of the Arduino Uno. It boasts an ATmega2560 microcontroller at its core, providing a generous array of digital and analog pins, memory, and computational power with 54 digital I/O pins, 16 analog inputs, and a larger flash memory capacity. We initially used the Arduino Uno, but as the number of components increased at each step, we switched to the Arduino Mega 2560. The Arduino Uno's limited number of pins was insufficient for all the components, making the Arduino Mega 2560 the main controller in our system.



Figure 4.2: Arduino Mega 2560

Raspberry pi 4

The Raspberry Pi 4 is a small, affordable computer used for a variety of projects. It features a 1.5 GHz 64-bit quad-core ARM Cortex-A72 processor, supports up to 8GB of RAM, and includes dual-band Wi-Fi, Bluetooth 5, and gigabit Ethernet for fast networking. It has two USB 3.0 ports, two USB 2.0 ports, and can connect to two 4K monitors using micro HDMI. It's commonly used for tasks like programming, media centers, and IoT projects, making it versatile for hardware and software development.

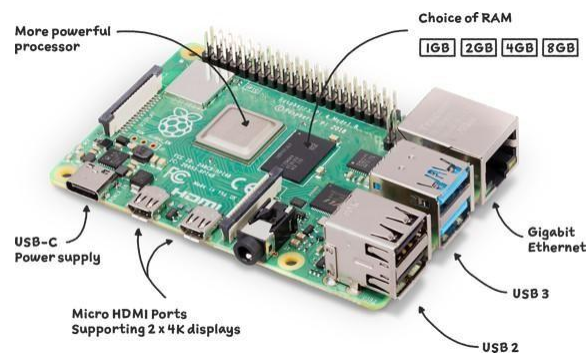


Figure 4.3: Raspberry pi 4

Stepper motor

A stepper motor is a type of electric motor that provides precise control over rotational motion. It works by dividing a complete rotation into individual steps, enabling accurate positioning without requiring feedback systems. The motor moves in fixed increments when it receives electrical pulses from a controller. Stepper motors are available in various types, such as unipolar and bipolar models, giving users options for different torque and speed requirements. Their key advantage is the ability to control position with high precision.

In our project, we adopted the CoreXY motion system due to its high efficiency and speed when moving along the horizontal axes. A Nema 17 stepper motor was used to control the X-axis movement, while another Nema 17 stepper motor was used to drive precise and effective motion along the Y-axis, ensuring full coverage of the workspace. For the Z-axis (the movable platform), we employed an additional Nema 17 stepper motor, allowing us to raise and lower the base accurately and reliably. This configuration provided high stability and smooth control over the robotic arm's movement inside the cube.



((a)) Stepper Motor NEMA17



((b)) A4988 driver

Figure 4.4: Stepper motor, and A4988 driver

IR sensor

The Infrared (IR) sensor is a widely used electronic component designed to detect the presence or absence of objects by emitting infrared light and measuring its reflection. In our project, we integrated the IR sensor into the product delivery slot to determine whether a product has been successfully received or not. This implementation helps ensure accurate delivery and prevents errors or duplication during the dispensing process.



Figure 4.5: IR sensor

Keypad

We used a 4x4 keypad in our project to let the user request a product by entering its number. It works with the LCD screen to display messages and guide the user, making the system easy and interactive.



Figure 4.6: Keypad

I2C 20X4 LCD

The 20×4 I2C LCD is used to display text across 4 lines, with 20 characters per line. It operates using the I2C protocol, which reduces the number of required wires to just two (SDA and SCL). In our project, we used it to show instructions and product selection data from the keypad, making interaction with the system clearer and more user-friendly.



Figure 4.7: I2C 20X4 LCD

RC522 (RFID)

The RC522 is a small and inexpensive RFID card reader used in wireless identification systems. It's an electronic module that can read and write data on RFID or NFC cards operating at 13.56 MHz frequency.

We used it together with a solenoid lock to secure the merchant's (or admin's) door, allowing only authorized users to open it and add products safely.



Figure 4.8: RC522 (RFID)

Solenoid lock

Computer speakers, also known as multimedia speakers, are designed primarily for use with computers. However, they're versatile enough to work with other audio devices, like MP3 players, as well.



Figure 4.9: Solenoid lock

H-Bridge (L298n)

I replaced the regular relay controlling the solenoid lock with an H-Bridge (L298N) to prevent voltage spikes caused by back-EMF.

An H-Bridge is an electronic circuit that allows current to flow in either direction through a load, such as a motor or solenoid, enabling better control. It also helps protect the circuit from voltage spikes generated by inductive loads.

The H-Bridge safely controls current direction and protects the circuit from these inductive spikes.



Figure 4.10: H-bridge

Power Supply



Figure 4.11: Power Supply

LM2596 Step Down Buck Converter Module

A **buck converter**, also known as a step-down converter, is a DC-to-DC converter that reduces voltage while increasing current from its source to the load. We used it in our project to step down the voltage from 12 volts to 5 volts, supplying power to the rest of the system. Specifically, we used it to power the Arduino, which requires 5 volts for proper operation.



Figure 4.12: Step Down Buck Converter Module

Limit Switch

We used limit switches in three locations: one for the X-axis, one for the Y-axis, and one at the base of the Z-axis. These limit switches help us establish a reference point (home position) for each of the X, Y, and Z axis.



Figure 4.13: Limit Switch

Robotic Arm

The robotic arm was custom-designed and 3D-printed to fit the specific needs and dimensions of our project.



Figure 4.14: Robotic Arm

Idler Pulley

In our project, we used a CoreXY motion system with ~~GT2 Timing Belts~~ *GT2 Timing Belts* (5 mm bore). These pulleys help guide the timing belt and keep the movement smooth and accurate. The timing belt connects the motors to the pulleys, making sure the system moves fast and with high precision without slipping. This setup is very useful for CNC machines and robotic arms like the one in our project.



Figure 4.15: GT2 20T Dual Bearing Aluminum Idler Pulley – 5mm Bore

GT2 Timing Belt

In our project, we used a ~~GT2 belt~~ as part of the CoreXY motion system. The timing belt connects to the motors and pulleys and is responsible for transferring the motor power to move the robotic arm. The GT2 belt is strong and flexible, and it keeps the movement smooth, accurate, and without slipping. It helps make sure the arm moves to the exact position we want.



Figure 4.16: GT2 Timing Belt

ESP8266

In our project, we used the ~~ESP826~~ **ESP8266** which is a small Wi-Fi module commonly used in IoT projects. The ESP8266 allows the system to connect to a local network or the internet and makes communication between the hardware and the mobile app possible. We used it to connect the hardware inside the cube to the mobile app, so the user can select the product they want, and the ESP8266 sends the request to the system to start the process.



Figure 4.17: Esp8266

Servo Motor (Mg996r)

In our project, we used the ~~MORSM~~ **MORSM** to grip and carry the product. We installed the servo motor between the gripper jaws, allowing precise control over their movement. This way, the servo can open and close the jaws as needed to firmly hold the product and move it to the desired position inside or outside the cube.



Figure 4.18: Mg996r

Sensor Dht11

The DHT11 sensor is an electronic device used to measure temperature and humidity. It is a small sensor that provides digital readings of the ambient temperature and relative humidity. We used it in our project to give us information about the temperature and humidity inside the system.

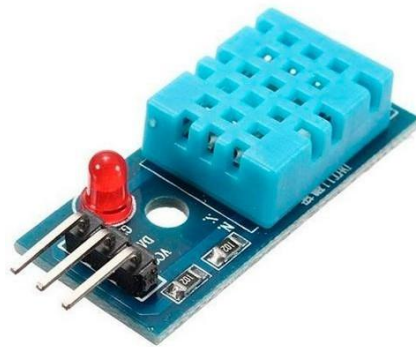


Figure 4.19: Dht11

Fan 12v

In our project, we used a 12V fan to cool the drivers and electronic components inside the system. The fan helps reduce the temperature and prevents overheating, which can damage the components or affect their performance.



Figure 4.20: Fan 12v

Lead & Lead Screw

In our system, we used a smooth ~~Lead~~ rod fixed to the base, and a threaded ~~Lead~~ Screw connected to the stepper motor at the bottom. When the stepper motor turns, it rotates the threaded Lead Screw, which moves the parts holding the base up or down. The smooth Lead rod acts as a fixed guide, allowing the base to move linearly up and down without rotating. This setup lets us control the base height precisely using the stepper motor's rotation.



Figure 4.21: Lead Screw



Figure 4.22: stainless steel round bar

Project rule

We designed the base using a thin wooden piece, which we painted black to give it a clean and modern look. To highlight the edges, we added an orange border around the base. This design not only gave the system a nice appearance, but also helped the camera and QR system easily detect and recognize the base and its boundaries inside the cube.



Figure 4.23: Project rule

Wires

We used different types of wires: male-to-female, male-to-male and female-to-female.



Figure 4.24: Wires

4.3 Hardware implementation

The system consists of three main parts: the mechanical structure (CNC and robot arm), the Raspberry Pi with camera, and the user interface (keypad with LCD and mobile app).

We built the mechanical part using 3D-printed parts and installed motors with a CoreXY system to move the robot arm along the X and Y axes. We also used a lead screw with a stepper motor to lift and lower the base accurately. This allows the arm to reach and grab the selected product inside the cube.

The Raspberry Pi processes images from the camera fixed at the top of the cube. The camera scans the QR codes of all products and identifies their exact positions. Then, the Raspberry Pi sends this data to the Arduino, which controls the arm to pick the correct product.

The user selects the product using the keypad with the LCD screen or the mobile app. The system finds the location of the selected product, moves the arm, and delivers the product out of the cube. The same process repeats for more products if needed.

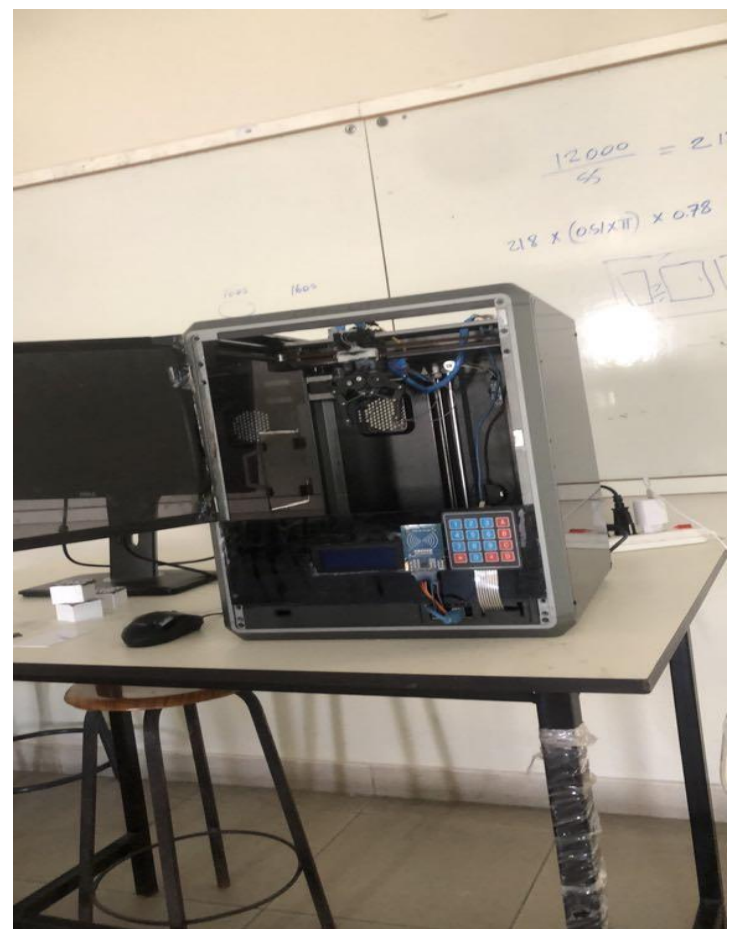
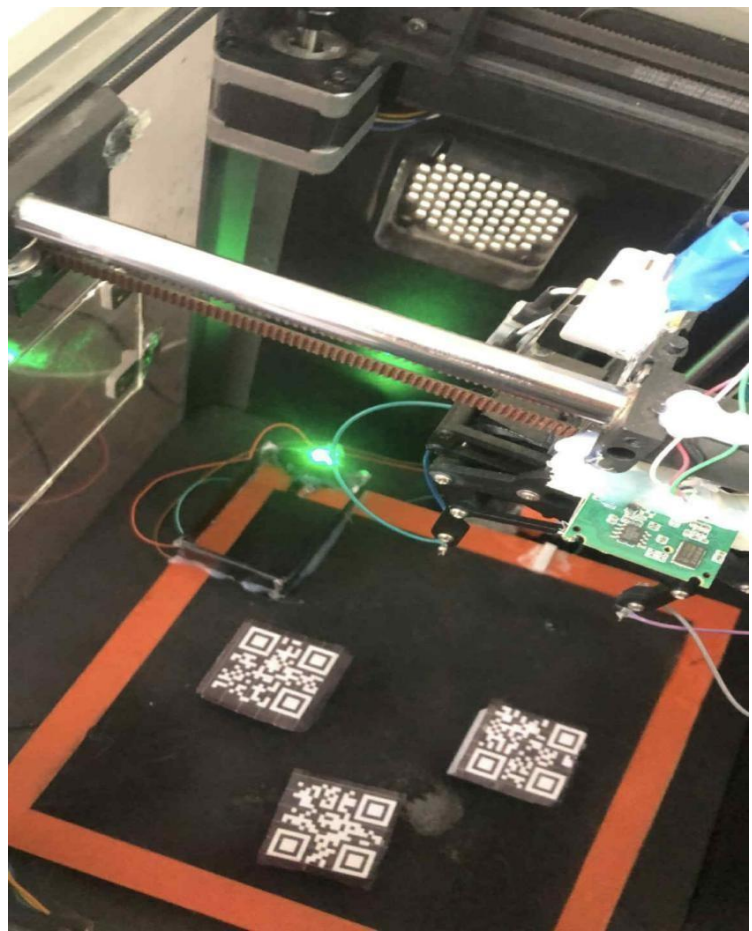
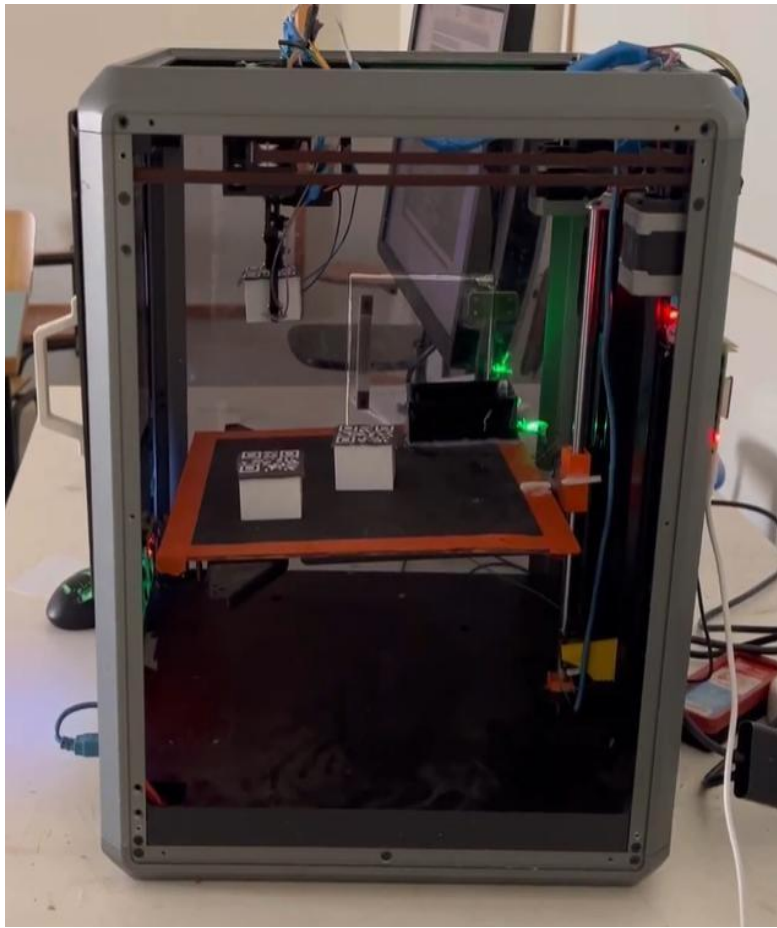


Figure 4.25 :Full Design

4.4 Project circuit

This image shows the hardware and circuit design of our project. It illustrates how we connected the Arduino, stepper motors, drivers, keypad, LCD, and sensors to build a complete and organized system.

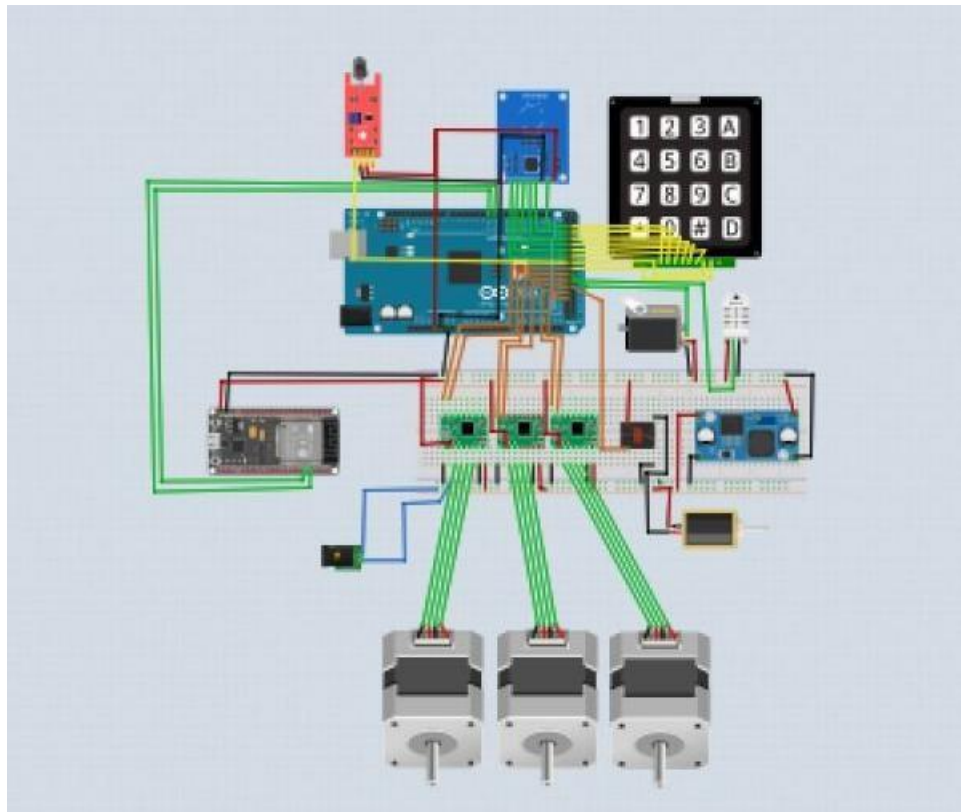


Figure 4.26: Project circuit

4.5 Flowchart of QR Robot: From Product Selection to Delivery

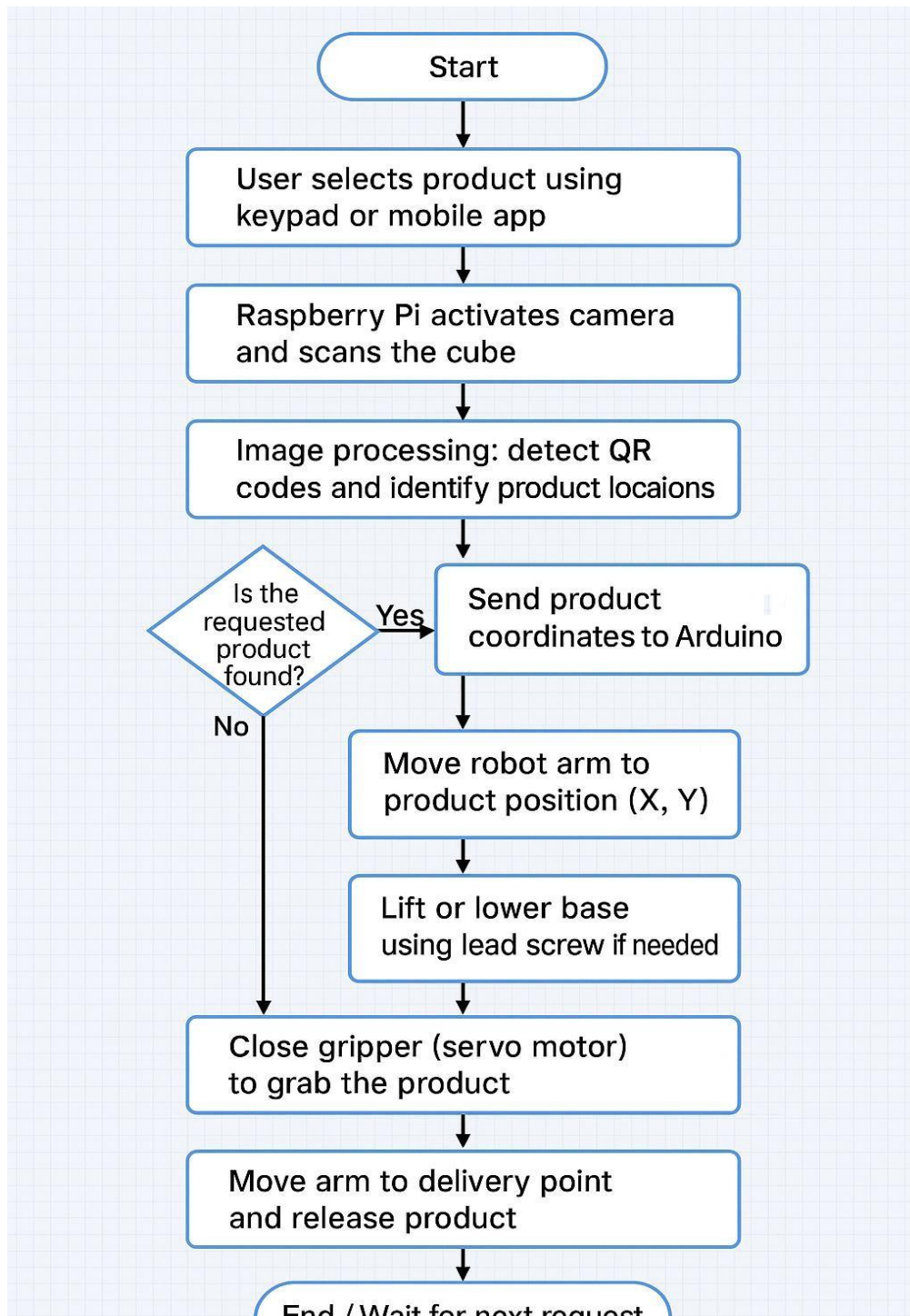


Figure 4.27: Flowchart of QR Robot

Software implementation

The system consists of three main parts: the mechanical structure (CNC and robot arm), the Raspberry Pi with camera, and the user interface (keypad with LCD and mobile app).

We built the mechanical part using 3D-printed parts and installed motors with a CoreXY system to move the robot arm along the X and Y axes. We also used a lead screw with a stepper motor to lift and lower the base accurately. This allows the arm to reach and grab the selected product inside the cube.

The Raspberry Pi processes images from the camera fixed at the top of the cube. The camera scans the QR codes of all products and identifies their exact positions. Then, the Raspberry Pi sends this data to the Arduino, which controls the arm to pick the correct product.

The user selects the product using the keypad with the LCD screen or the mobile app. The system finds the location of the selected product, moves the arm, and delivers the product out of the cube. The same process repeats for more products if needed.

4.6.1 Mobile App:

In our project, we used the **ESP8266** to build a simple and smart web-based application that works without the internet.

The ESP creates its own Wi-Fi network called "**merchant**", and the user can connect to it directly using a phone or browser.

Once connected, the user can add, delete, or order a product through a web interface that communicates with a web server running on the ESP.

Each product is stored in a file named **products.txt** using the **LittleFS** file system.

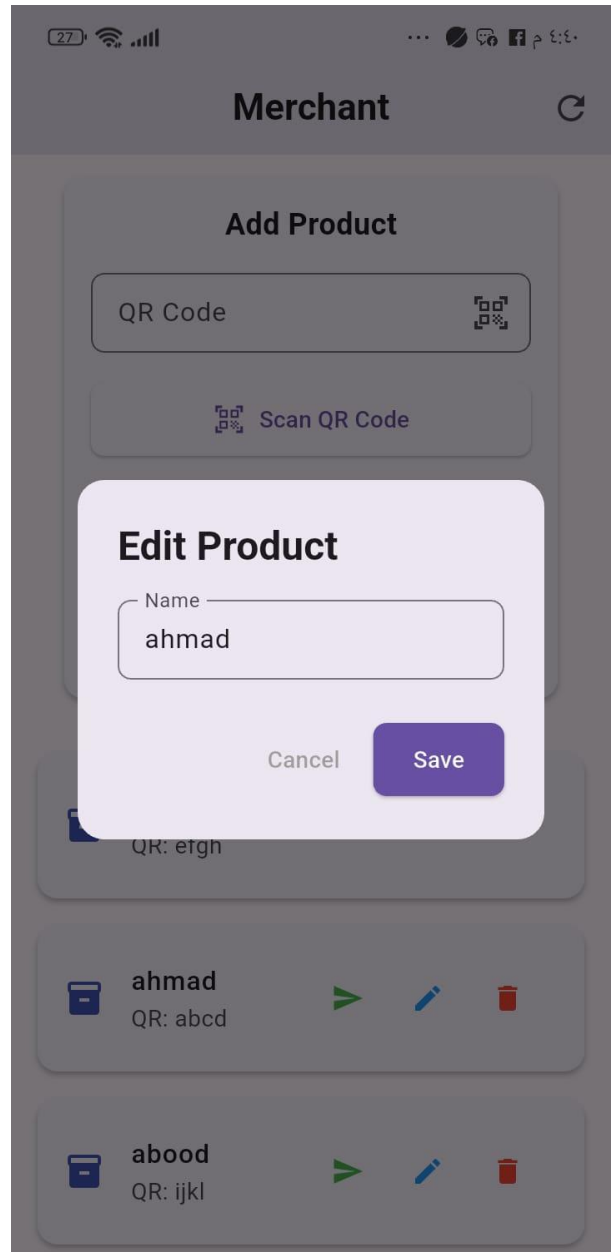
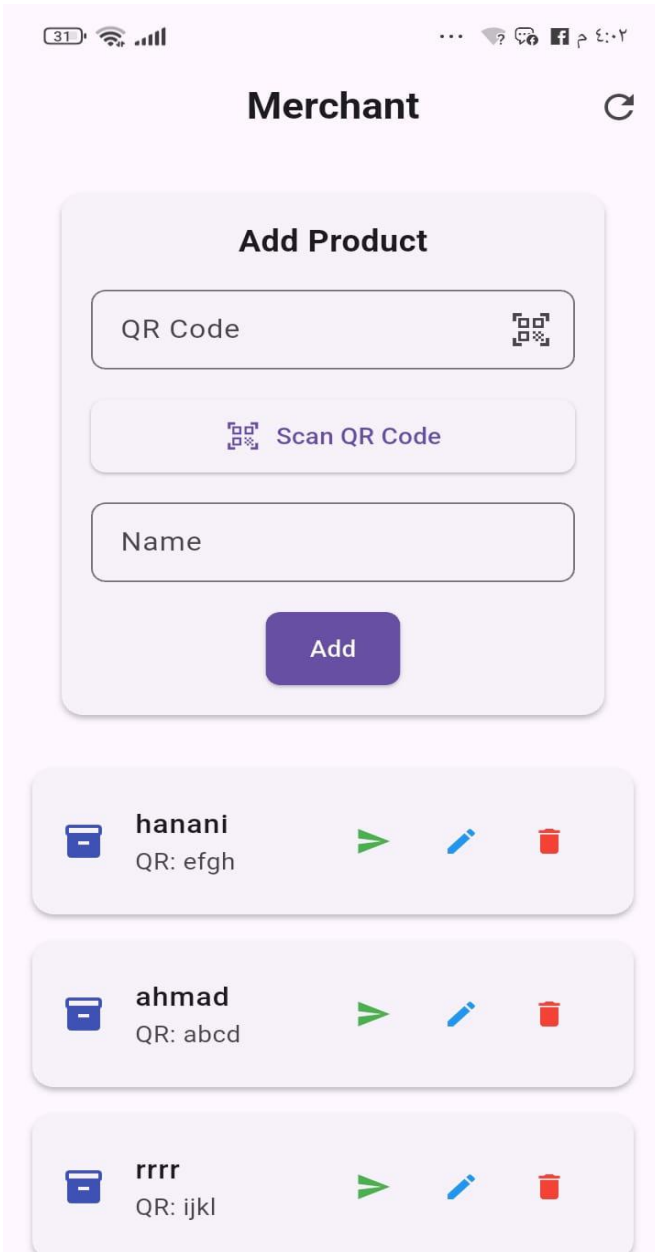
When a product is ordered, the ESP sends the QR code of the selected product to the Arduino via **Serial communication**.

The Arduino then moves the robotic arm to pick up the product and deliver it to the user.

In this way, we successfully connected:

- a smart user interface
- internal data storage
- and physical robotic control

into one organized, efficient, and easy-to-use system.



4.6.2 QR Code Detection

Initially, we used the YOLO model for detection, but it was too slow and resource-heavy for the Raspberry Pi, causing delays in product identification. Therefore, we switched to using the pyzbar library combined with OpenCV for faster and more efficient QR code detection.

The algorithm works as follows:

- Captures frames from the camera and converts them to grayscale.
- Enhances the image contrast using CLAHE (Contrast Limited Adaptive Histogram Equalization).
- Creates several processed variants of the image (such as adaptive thresholding and color inversion) to improve detection chances.
- Attempts to decode QR codes from each variant.
- If a QR code is detected, it identifies its location and center within the frame.
- Compares the detected QR code with the target QR code received from the controller via serial communication.
- Based on the QR code's position relative to the center of the frame, determines the robot's movement direction (left, right, forward, backward, or stop).
- Sends the movement command back to the controller through the serial interface.

This approach enables fast and accurate QR code detection with low resource consumption compared to heavier methods like YOLO.

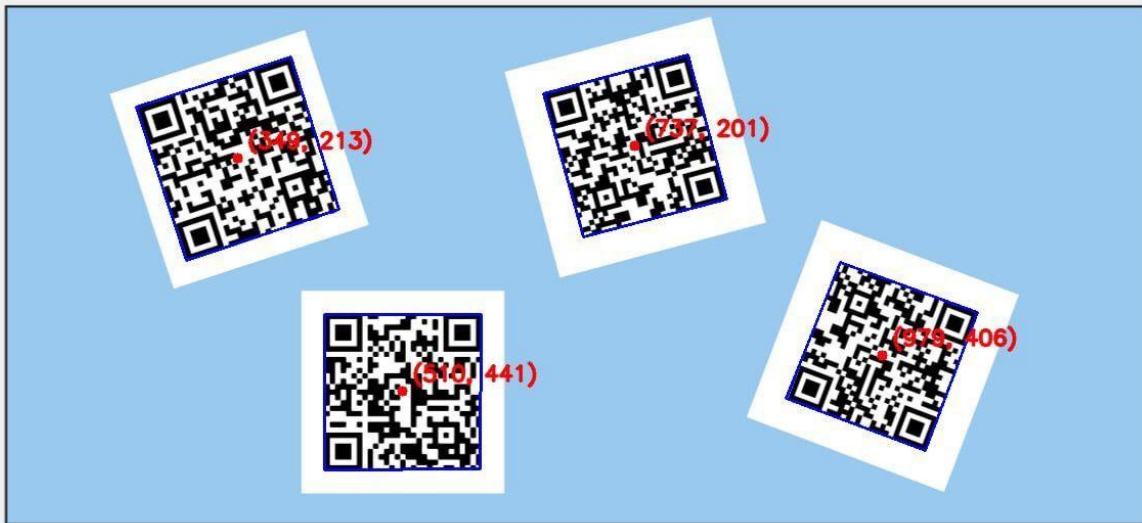


Figure 4.28: QR Code Detection

QR Code Detection:

```

Tabnine | Edit | Test | Explain | Document
def decode_qr(frame):
    variants = [frame]

    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
    clahe_img = clahe.apply(frame)
    variants.append(clahe_img)

    thresh = cv2.adaptiveThreshold(clahe_img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
    cv2.THRESH_BINARY, 11, 2)
    variants.append(thresh)

    variants.append(cv2.bitwise_not(frame))
    variants.append(cv2.bitwise_not(clahe_img))
    variants.append(cv2.bitwise_not(thresh))

    for var in variants:
        decoded = pyzbar.decode(var)
        if decoded:
            return decoded
    return []

```

Figure 4.29:Code Of QR Code Detection

4.6.3 Determining the Location of Each QR Product

In our project, the camera captures a frame that is converted to grayscale to improve QR detection. We use the pyzbar library to scan the frame for QR codes. When a QR code is detected, we get its corner points (polygon). From these points, we calculate the center position (x, y) of the QR code by averaging the minimum and maximum x and y coordinates. This (x, y) location represents the center of the QR code in the frame and helps guide the robotic arm toward the correct product.

Code that determines the (x, y) location:

```
for obj in decoded_objects:
    text = obj.data.decode('utf-8')
    points = obj.polygon

    x_coords = [point.x for point in points]
    y_coords = [point.y for point in points]
    x1, y1, x2, y2 = min(x_coords), min(y_coords), max(x_coords), max(y_coords)

    cx = (x1 + x2) // 2
    cy = (y1 + y2) // 2
```

Figure 4.30: Code Of QR Code Detection Determining the Location of Each QR Product

However, during the robotic arm movement, occasional inaccuracies in positioning occur, causing imperfect alignment. To solve this, we implemented limit switches on the robotic gripper claws. When a limit switch is triggered (pressed), the arm recalibrates its position based on the QR code location. This feedback mechanism improves the precision of the product pickup.

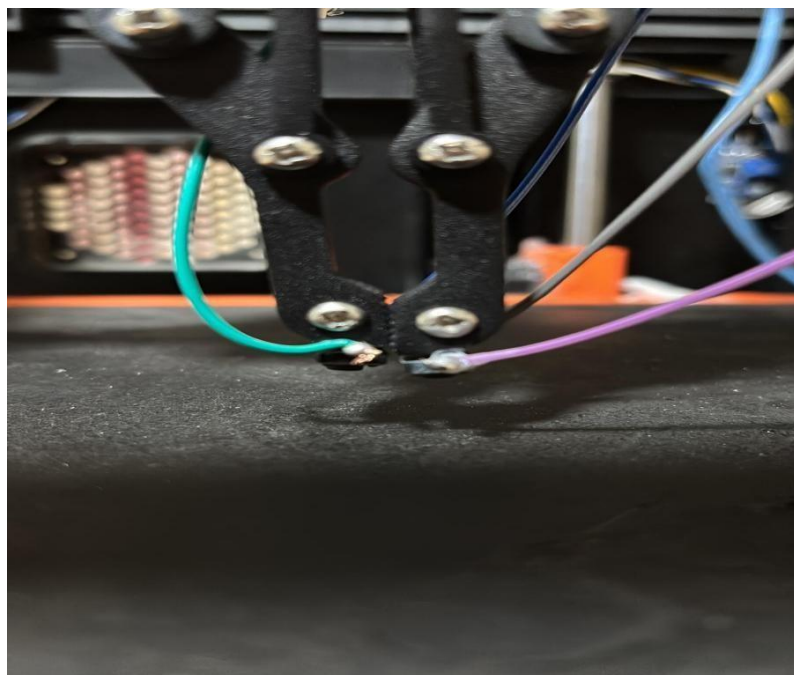


Figure 4.31: Limit Switch In Robot Arm

Below is the code to solve this problem

```
int delayMicrosecondsNonBlockingForUp(unsigned int us)
{
    unsigned long start = micros();
    while (micros() - start < us)
    {
        if(!digitalRead(RIGHTS) && !digitalRead(LEFTS))
        {
            return 1;
        }
        else if (!digitalRead(RIGHTS))
        {
            return 2;
        }
        else if (!digitalRead(LEFTS))
        {
            return 3;
        }
    }
    return -1;
}
```

```
while(upRes == 2)
{
    down(300);
    right(100);
    upRes = up(450, true);
}

while(upRes == 3)
{
    down(300);
    left(100);
    upRes = up(450, true);
}

up(300, true);
controlServo(false);
down(2300);
homePosition(true);
}
```

Figure 4.32: Code Of Limit Switch In Robot Arm

Another challenge we faced in the project was that the robotic arm could not pick up products when they were tilted at certain angles. This is because the arm currently operates with only three axes of motion, which limits its ability to adapt to different product orientations. The system can be improved in the future by upgrading the arm to work with six axes of motion, allowing it to adjust its gripping position and pick up products regardless of their tilt, thereby greatly enhancing the system's efficiency and accuracy. We did not implement this upgrade at this stage because it is financially costly for us as students, but it remains an important option for future development.

Chapter 5

Results and Discussion

Our project successfully developed a smart system capable of automatically recognizing products inside a cube using a Raspberry Pi camera and QR code detection, and controlling a robotic arm to accurately pick the desired item. The hardware components—including the camera, Raspberry Pi, Arduino, and motors—worked seamlessly together to perform tasks efficiently and autonomously.

We faced some challenges, such as improving QR code detection accuracy under varying lighting conditions and fine-tuning the robotic arm's movement to reach precise product locations. These issues were addressed through software enhancements in image processing and by adding limit switches for precise calibration of the arm's position.

Additionally, the system includes an easy-to-use interface via a keypad and mobile app, which simplifies product selection and minimizes errors. While the core objectives were achieved, some limitations remain, including slight delays in detecting QR codes under complex lighting and the need for periodic arm calibration to maintain accuracy.

Chapter 6

Conclusion and Recommendations

The project successfully developed an automated smart cube system that can detect products using QR codes and control a robotic arm to pick the desired items accurately. The integration of image processing, hardware control, and user interfaces made the system efficient and user-friendly. This solution can be applied in various fields like inventory management, automated vending, and smart storage.

For future improvements, we recommend:

- Developing the gripping system to be able to pick up products that are tilted at different angles, which are otherwise difficult to grasp, by adding extra movement axes and smoother, more precise motions.
- Replacing the current grippers with a vacuum gripper system that can pick up any product regardless of its external shape, thereby increasing the system's flexibility and efficiency.
- Enhancing the system's ability to detect QR codes under varying lighting conditions and environments to ensure recognition accuracy at all times.
- Improving the robotic arm's precision and speed through accurate calibration and using advanced motors and electronic components.
- Applying intelligent techniques for obstacle detection and avoidance within the work environment to increase system reliability and safety.
- Adding diverse and user-friendly input interfaces such as voice control or touchscreens to provide a smoother and more interactive user experience.
- Integrating a real-time monitoring and feedback system that allows remote control and supervision, facilitating easier system management and maintenance.
- Expanding the system to include multiple robotic arms working in parallel to increase production speed and efficiency.
- Developing a flexible and scalable database to dynamically manage inventory with smart data analysis and decision-making capabilities.
- Additionally, integrating machine learning techniques to improve product recognition accuracy and demand forecasting, as well as using additional sensors like depth cameras to better understand the working environment.

In conclusion, this project demonstrated how combining computer vision, robotics, and embedded systems can automate complex tasks effectively, opening the door to smarter, more interactive automated solutions. The continuous development and integration of advanced technologies will further enhance system performance, making it adaptable