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

PREPARED BY:

Najat S. Mansour
Mohammad A. Alawneh

SUPERVISED BY:

Dr. Hanaal Abu-Zant
Dr. Aladdin Masri

January 02, 2025

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

Dedication

First and foremost, to God be all the praise. We have reached this point in our educational path thanks to his heavenly guidance.

To the person whose wisdom and advice have consistently inspired us. Our Master Muhammad, may God bless him and grant him peace.

To the souls of our brave martyrs. Whose sacrifices have taught us the true meaning patience and motivated us to continue pursuing dreams on behalf of many of them.

To those whose prayers and supplications were the secret of our success.

To all those who believed in us, enveloped us with love and encouragement, lifted our spirits, and extended a hand when we felt on the brink of giving up our dear family, friends, and teacher thank you from the depths of our hearts for being an inseparable part of this journey.

Here stands this modest effort, a proof of the love and support that have driven us along the way.

Acknowledgment

We would like to express our sincere gratitude to all these individuals for mentoring and supporting us in completing this project. Our first expression of gratitude and appreciation goes to our supervisors **Dr. Hanaal Abu-Zanat** and **Dr. Aladdin Masri**, for their great feedback and assistance during this project. We also extend our heartfelt gratitude to all the teachers in the Computer Engineering department who gave their time for teaching us, working hard to improve our academic production, and recognizing our efforts throughout our university journey until we reached at this point.

Disclaimer Statement

This report was written by ***Najat Mansour*** and ***Mohammad Alawneh*** 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 ***Najat Mansour*** and ***Mohammad Alawneh***. 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.

Contents

List of Figures

List of Tables

Nomenclature

API	Application Programming Interface
ATM	Automated Teller Machine
DC	Direct Current
$EEPROM$	Electrically Erasable Programmable Read-only Memory
$ESP32$	Espressif32
GND	Ground
GUI	Graphical User Interface
HSI	Hyper-Spectral Imaging
$HTTPS$	Hypertext Transfer Protocol Secure
I^2C	Inter-Integrated Circuit
$ICSP$	In-Circuit Serial Programming
IDE	Integrated Development Environment
$IEEE$	Institute of Electrical and Electronics Engineers
ILS	Israeli New Shekel
IR	Infrared
$JSON$	JavaScript Object Notation
LCD	Liquid-Crystal Display
LED	Light-Emitting Diode
PWM	Pulse Width Modulation
$REST$	REpresentational State Transfer
RFC	Request For Comments

RGB	Red, Green, Blue
SCL	Serial Clock
SDA	Serial Data
$UART$	Universal Asynchronous Receiver / Transmitter
URI	Uniform Resource Identifier
USB	Universal Serial Bus
USD	United States Dollar
UV	Ultraviolet
V_{cc}	Voltage Common Collector
V_{DD}	Voltage at the Drain
VCP	Virtual COM Port
$Wi-Fi$	Wireless Fidelity

Abstract

ExchanGo is a currency exchange machine that is important in the need to provide service 24/7, reducing waiting times. In addition, the service was provided with the best possible accuracy. So, this will minimize human errors during currency conversion and counting. The importance of this machine increases in places that witness congestion and where people need immediate, fast, and accurate access to currencies such as various tourist destinations and airports.

This machine allows users to to break larger currency into smaller ones **(breaking exchange)** or combine smaller currencies into larger one **(consolidating exchange)** within the same currency for both **banknotes (paper money or bills)** and **coins** within ILS currency. It can be controlled through a mobile application for administration to update the currencies' states that are stored in a file in the flash memory of ESP32 microcontroller and through a manual control installed on the machine itself through basic I / O devices for normal users.

The mobile application will be implemented using *Flutter* framework in Dart programming language, two microcontrollers (*Arduino Mega 2560* and *ESP32*) will be used to control the hardware of the project. The detection of the banknotes will be according to their color, while the detection of the coins will be according to their sizes.

There may not be a very similar application, but this machine has parts inside it that are already working as separate applications such as counting money, checking if it is counterfeit or not, and sorting it into different categories.

The main goal of this project is to design a currency exchange machine that performs the functions of several different devices in addition to humans; to increase the accuracy, reduce the errors and improve the scalability. This project can be built in the future and integrated with known ATMs.

Chapter 1

Introduction

The number of tourists in the first nine months of 2024 reached about 1.1 billion [1], who quickly think about the currency of the country they are traveling to, and this is an indicator of the size of the currency exchange market in the world, which is growing little by little. According to recent studies, the size of the foreign exchange market is expected to reach at USD 0.89 billion by 2025 and will reach USD 1.18 billion bill by 2030 [2].

With the technological development, the attempt began to automate financial transactions, as the ATM was invented in 1960s and began to spread until the number of ATMs around the world currently reached about 3.5 million ones [3].

Despite the technological progress in the field of financial transactions and the large currency exchange market, exchange shops are still widespread and the machine that will replace humans in this field has not yet appeared.

The goal of this machine is supporting two process: First, **Breaking Exchange** which allows users to break larger currency into smaller ones. Secondly, **Consolidating Exchange** which allows users to combine smaller currencies into larger ones. The machine supports these two process on both banknotes (paper money or bills) and coins within ILS currency.

To ensure the machine is user-friendly, it can be controlled through a mobile application in a wireless (remote) manner for administration to update the currencies' states that are stored in a file in the flash memory of ESP32 microcontroller and using a manual control installed on the machine itself for normal users.

The importance of this machine lies in the fact that it is the first machine specialized in breaking and consolidating currencies instead of resorting to shops to carry out these operations. It ensures providing 24/7 service, reducing wait-

ing times, reducing costs and staff while serving many customers, and ensuring non-discrimination in exchange operations.

It also aims to improve accuracy and avoid human error in financial transactions, especially with the possibility of benefiting from recording financial transactions in various things such as monitoring and conducting research.

The struggle between machine and human will continue, and those who embrace the idea of machine will continue to insist on it as long as human error remains high. According to studies, human error is responsible for 80% of business failures, and a typo in accounting can lead to a loss of USD 225 million. If we talk about data from 10,000 inputs, human error will be between 100 and 400 times, while machine error will be between 1 and 4 times [4].

In **Chapter 2**, we will discuss the related and previous work. **Chapter 3** will address the methodology including the design, the materials and tools, the software development, the mobile application, and the standards and constraints. In **Chapter 4**, we will present the results and analysis, and **Chapter 5** will provide a discussion of the findings. Finally, **Chapter 6** will conclude with our conclusions, recommendations and future work.

Chapter 2

Literature Review

2.1 Similar Machines (Systems)

There are some similar machines in the field of currency in general or the field of currency exchange in particular, such as:

2.1.1 Currency/Money Exchange Machine by Hongzhou Smart

It is an automated self-service kiosk that allows customers of banks and money exchange institutions to exchange currencies independently. It accepts more than 20 currencies and supports up to 4 types of banknotes (paper money or bills) and 2-4 types of coins in the breaking process [5].

2.1.2 Cash Exchange Machine - 4300

It supports exchanging high denomination banknotes (paper money or bills) into coin and low denomination banknotes, It also takes up little space, and offers a simple user interface. [6].

Although similar systems exist, they only support the breaking exchange process and do not support consolidating exchange process. In addition to the fact that our system began with ILS currency, which is the local currency of the State of Palestine.

2.2 Currency Detection Techniques

There are multiple techniques for detecting currencies [7] [8] [9]:

- Ultraviolet (UV): genuine banknotes have UV-sensitive inks that can be detected by UV sensors.
- Magnetic Ink: genuine banknotes printed with magnetic ink that can be detected by magnetic sensors.
- Infrared (IR): genuine banknotes absorb or reflect IR and IR sensors can be used to detect these features.
- Microprint: genuine banknotes contain very small text that can be detected by optical sensors or high-resolution cameras.
- Watermark: genuine banknotes contain watermark that can be detected by optical sensors or high-resolution cameras.
- Serial Number: genuine banknotes contain unique serial number that can be scanned and compared to a database of valid numbers.
- Hyper-Spectral Imaging (HSI): converts the image from RGB to HSI in order to detect unique spectral signature [10].
- Machine Learning Algorithms: by using a model trained on large dataset [11] [12].
- Physical and Chemical Properties: such as the conductivity or chemical composition of the polymers used in the currencies.
- Thickness and Size Measurement: measures the dimensions (Width, length and thickness) of the currencies, it is suitable for coins.
- Color: detects the currencies based on their colors, it is suitable for banknotes (paper money or bills).

NOTE : The last two techniques cannot detect the counterfeit currencies.

The next chapter includes the methodology, which explains the design, the materials and tools used in it including the technique used to detect the currencies, the software programs and technologies used in the mobile application, and the standards and constraints of this project.

Chapter 3

Methodology

3.1 Design

3.1.1 System's User Interface

The user can deal with:

- **Top Thin Horizontal Slot**: for entering banknotes (paper money or bills).
- **Down Four Thin Horizontal Slots**: for outputting the banknotes (paper money or bills) -one for each type (200, 100, 50 or 20 ILS)-.
- **Coins Acceptor**: for entering coins.
- **Flat (Large) Horizontal Slot**: for outputting coins.
- **Keypad**: for entering and processing the transaction.
- **LCD**: for monitoring the transaction.

Figure 3.1: System's User Interface

3.1.2 Getting Inside

The following image shows the inner design of the machine:

Figure 3.2: Inner design of the machine

3.1.3 Currency Storage

The currencies are stored as the following in 3D-printed slots:

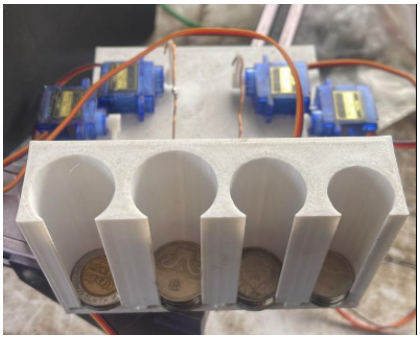

Figure 3.3: Coins Storage Slots

Figure 3.4: Banknotes Storage Slots

Refer to 5.1 to see the working principle.

3.2 Materials and Tools

3.2.1 Microcontrollers

- **Arduino-Mega 2560** [13]: A well-known microcontroller based on ATmega2560, It has the following features:
  - 54 digital input/output pins (of which 15 can be used as PWM outputs).
  - 16 analog inputs
  - 4 UARTs (hardware serial ports)
  - 16 MHz crystal oscillator
  - USB connection
  - Power jack
  - ICSP header
  - Reset button

  It is used to control the hardware.

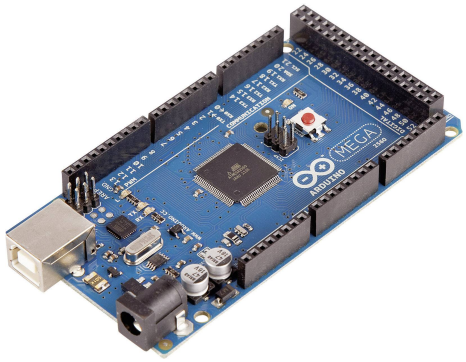

Figure 3.5: Arduino-Mega 2560

- **ESP32** [14]: A well-known microcontroller that supports 2.4 GHz Wi-Fi and Bluetooth, it is used to enable wireless control of the system via the admin mobile application to update the currencies' states that are stored in ESP32 flash memory in a text file.

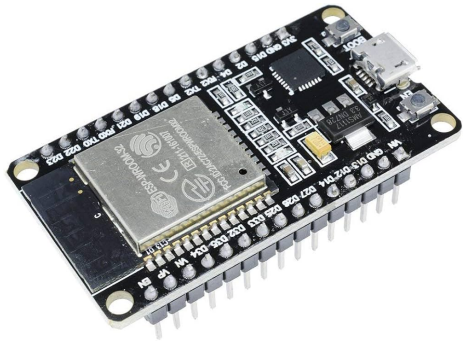

Figure 3.6: ESP32

3.2.2 Motors

- **DC Motor** [15]: Two-wire (power and ground) continuous rotation motor which converts DC electrical energy to a mechanical energy. Two DC motors are used as the following:
  - 1 DC motor for operating the vacuums.
  - 1 DC for banknotes outputting mechanism.

Figure 3.7: DC Motor

- **NEMA 17 / 103H548 Stepper Motor** [16]: A DC motor moves in discrete steps and coverts the electrical power into rotation. It has 1.8°degree/step and it can be controlled in terms of degree of rotation, direction and speed.
  It consists of:
  - Stator: the stationary (fixed) part.
  - Rotator: the moving part.

  It has two modes:
  - Full-Step Mode: the motor is operated with only one phase energized at a time. The advantage of this mode is using least amount of the power from the driver.

    $$\text{Number of steps for one cycle} = \frac{\theta}{1.8^\circ} = \frac{360^\circ}{1.8^\circ} = 200\,\text{steps}$$

  - Half-Step Mode: it is a combination of one phase and two adjacent phases at a time. The advantage of this mode is having smoother operation and better resolution.

    $$\text{Number of steps for one cycle} = \frac{\theta}{\frac{1.8^\circ}{2}} = \frac{360^\circ}{\frac{1.8^\circ}{2}} = 400\,\text{steps}$$

    It is connected to a 3-stage gear-box that moves the arm that holds the vacuums, the desired steps are **1100 steps (5.5 full-cycles)**.

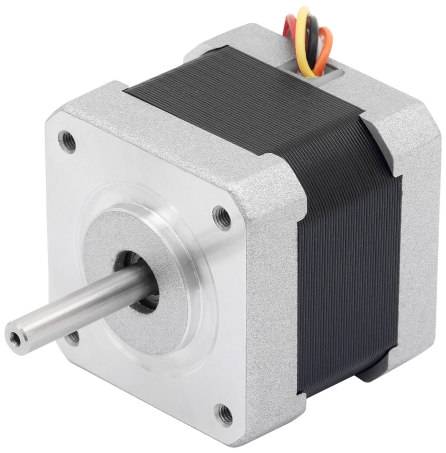

Figure 3.8: Stepper Motor

- **Servo Motor** [17]: A DC / AC motor with some extra features:
  - **Control circuit**: set the angle [0°– 180°].
  - **Gears**: transform speed into torque.
  - **Potentiometer**: keeps track of the motor angle.

  It contains three connection pins:
  - V_{cc}: The red one.
  - GND: The brown one.
  - Control (Signal): The orange one and must be attached to PWM pin in the microcontroller.

  Five servo motors are used as the following:
  - 4 servo motors rotate **180**°for coins outputting mechanism.
  - 1 servo motor rotates **110**°to enter the banknote (paper money or bill) in the input mechanism.

Figure 3.9: Servo Motor

3.2.3 Sensors

- **TCS230/TCS3200 Color Sensor** [18]: It has four LEDs that lights up the object in front of it, and uses RGB sensor chip to detect colors. It contains an array of photodiodes (Device that converts lights into current) as the following:
  - 16 photodiodes with red filter.
  - 16 photodiodes with green filter.
  - 16 photodiodes with blue filter.
  - 16 photodiodes without filter.

Then, it contains current to frequency converter, it has the following connection pins:

- GND
- V_{DD}
- OE (Output Enable)
- S0, S1: Output frequency scaling (100%[11], 20%[10], 2%[01] or None[00]).
- S2, S3: Photodiode type (Red[00], Blue[01], None[10] or Green[11]).
- OUT: Output frequency.

It is used to detect the type of banknote (paper money or bill) as it explained in 5.1.3.

Figure 3.10: TCS230/TCS3200 Color Sensor

3.2.4 I/O Devices

- **4 * 4 Matrix Keypad** [19]: A matrix of 16 keys (Numbers between 0 - 9, letters between A - D and special characters * and #) organized in rows and columns. To be interfaced with a microcontroller, it has 8 digital pins (4 for rows and 4 for columns). It is used in this machine to allow the user to enter and process the transaction.

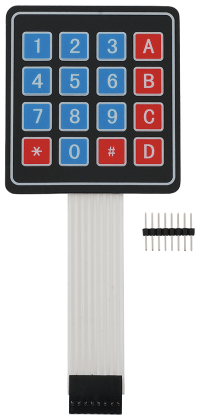

Figure 3.11: 4 * 4 Matrix Keypad

- **16 * 4 LCD** [20]: An informative output that can be used to display characters and sometimes graphics, it supports two types of information: data or control commands (clear the screen, set the cursor, etc.) and it can be used in both byte-mode (8-bits) or nibble mode (4-bits). It has the following connection pins:
  - V_{cc}
  - V_{ss} or GND
  - V_o: for controlling the contrast via a potentiometer.
  - D0 - D7: data pins.
  - R/W (Read / Write): for selecting reading mode or writing one.
  - RS (Register Select): for setting the information type (data or control command).
  - EN (Enable): for enabling the LCD.

  LCD requires two many pins. So, we used I^2C LCD that uses an adapter PCF8574 that converts the LCD multiple connection pins into only four ones (V_{cc}, GND, SCL and SDA).

  It used in this machine to allow the user to monitor and track the transaction (displaying menus, showing results, etc.).

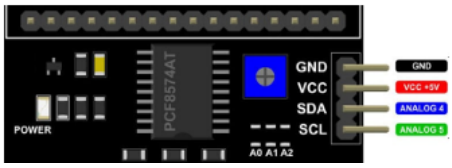

Figure 3.12: PCF8574 Adapter

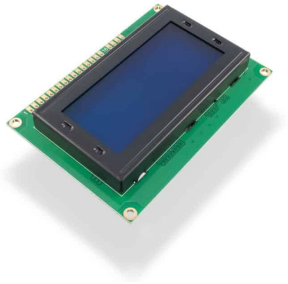

Figure 3.13: 16 * 4 LCD

3.2.5 Drivers

- **Stepper Motor Driver TB6600** [21]: It is a middle layer between the stepper motor NEMA 17 / 103H548 and Arduino-Mega 2560. It has the following pins:
  - V_{cc}
  - GND
  - A+, A-, B+ and B-: Steppe motor windings.
  - PUL+: Pulse control signal.
  - PUL-: Typically is GND.
  - DIR+: Direction control signal.
  - DIR-: Typically is GND.
  - ENA+: Enable control signal.
  - ENA-: Typically is GND.

Figure 3.14: Stepper Motor Driver TB6600

3.2.6 Others

- **Limit Switch** [22]: An electromagnetic device that respond to a physical force, it is typically used to detect objects or physical actions. It has the following connection pins:
  - NO (Normally Open -Output-)
  - NC (Normally Closed -Output-)
  - COM (Common Contact -Output-): The central connection pin.

  It is used in this machine to ensure the stepper motor rotated to a specific reference.

Figure 3.15: Limit Switch

- **Mechanical Relay** [23]: An electrically powered switch that use electrical signals to open and close circuits. It has the following connection pins:
  - V_{cc}
  - GND
  - IN (Input)
  - NO (Normally Open -Output-)
  - NC (Normally Closed -Output-)
  - COM (Common Contact -Output-): The central connection pin.

  Six relays are used as the following:
  - 4 relays for controlling the 4 vacuums.
  - 2 relays for controlling the 2 DC motors.

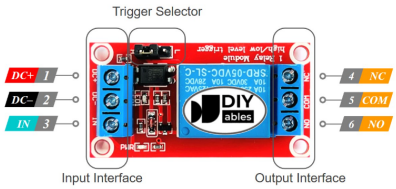

Figure 3.16: Mechanical Relay

- **Coins Acceptor CH-926** [24]: A device for entering and detecting coins based on their magnetic properties. It requires the following configurations:

Coin Type	Number of Samples	Impulses Counter	Precision
10 ILS	15	5	6
5 ILS	15	4	6
2 ILS	15	3	6
1 ILS	15	2	6

NOTE : The impulse length can be fast(30 ms), medium(50 ms) or slow(100 ms), It is **fast (30 ms)** in this machine.

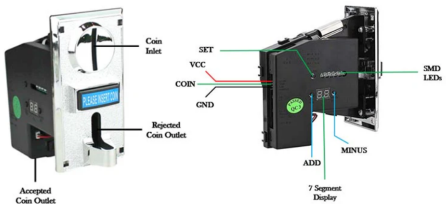

Figure 3.17: CH-926 Coins Acceptor

- **Vacuum**: 4 Vacuums are used for picking up the banknotes in the output mechanism.

Figure 3.18: 4 Vacuums

- **Power Supply**: It is used provide power for the system, the following table shows the voltage required for each device:

Tool	Number of Item(s)
DC Motor NEMA 17 / 103H548 Stepper Motor Stepper Motor Driver TB6600 Vacuum Mechanical Relay Coins Acceptor CH-926	12V
Arduino-Mega 2560 Color Sensor 16 * 4 LCD Limit Switch	5V
ESP32 Servo Motor	3.3V

Figure 3.19: Power Supply

The following table shows the number of each tool used in this machine:

Tool	**Number of Items**
Arduino-Mega 2560	1
ESP32	1
DC Motor	2
NEMA 17 / 103H548 Stepper Motor	1
Servo Motor	5
Color Sensor	1
4 * 4 Matrix Keypad	1
16 * 4 LCD	1
Stepper Motor Driver TB6600	1
Limit Switch	1
Mechanical Relay	6
Coins Acceptor CH-926	1
Vacuum	4
Power Supply	1

3.3 Software Development

3.3.1 Tools

- **Arduino IDE** [25]: An open-source IDE to write Arduino / ESP codes and upload them into the microcontrollers.
- **CP210x USB to UART Bridge VCP Drivers** [26]: Drivers to install ESP32's code on it (to allow the communication between the computer and ESP32).

3.3.2 Arduino and ESP32 Libraries

Arduino and ESP32 provide multiple libraries to deal with various hardware tools, the following table shows the libraries used in this project:

Tool	**Library -If any-**
Arduino-Mega 2560	Arduino.h avr/wdt.h (Reset Watch-Dog Timer)
ESP32	ESPAsyncWebServer.h ArduinoJson.h WiFi.h FS.h & SPIFFS.h (Accessing the ESP32 flash memory)
DC Motor	-
NEMA 17 / 103H548 Stepper Motor	-
Servo Motor	Servo.h
Color Sensor	-
4 * 4 Matrix Keypad	Keypad.h
16 * 4 LCD	LiquidCrystal_I2C.h
Stepper Motor Driver TB6600	-
Limit Switch	-
Mechanical Relay	-
Coins Acceptor CH-926	-
Vacuum	-

3.3.3 Source Codes

- **Arduino Mega Code**: It is attached in Appendix A.
- **ESP32 Code**: It is attached in Appendix B.
- **Mobile Application Code**: It is attached in Appendix C.

3.4 Mobile Application

3.4.1 Main Framework

- **Flutter** [27]: A Dart framework developed by Google for front-end mobile applications.

3.4.2 Tools

Workspace Tools

- **Visual Studio Code** [28]: Very popular code editor that can be customized with multiple extensions to support any programming language and saves time and effort.
- **Android Studio** [29]: Very popular IDE in mobile development, we use it to create virtual devices to build and test the mobile application on them.
- **Postman** [30]: Desktop application (Provides also a website and VS Code extension) to test the back-end APIs and generate documentations for their usage.
- **Git / GitHub**:
  - **Git** [31]: Open-source version control system to deal with cloud-based platforms such as GitHub and GitLab.
  - **GitHub** [32]: Cloud-based platform for saving and sharing codes.

Development Tools

- **Dart** [33]: A programming language developed by Google for web, desktop and mobile applications.

3.5 Standards and Constraints

3.5.1 Standards

- **RESTful API standards**: The RESTful API on ESP32 microcontroller requires the following standards -REST itself is not a standard- in order to communicate with the mobile application and exchange information:
  - Wi-Fi [IEEE 802.11]
  - HTTPS [RFC 2818]
  - URI [RFC 3986]
  - JSON [RFC 8259]
- **UART**: for serial communication between Arduino-Mega 2560 and ESP32 microcontrollers.
- **USB 3.0**: for data transfer and power.

3.5.2 Constraints

- The inability to detect counterfeit banknotes (paper money or bills) due to the technique of detecting them based on their colors.
- The inability of the vacuum to pick up damaged (deteriorated) banknotes, and this can be overcome by changing the method of outputting banknotes (paper money or bills).

Chapter 4

Results

The machine supports both **Breaking Exchange** which allows users to break larger currency into smaller ones and **Consolidating Exchange** which allows users to combine smaller currencies into larger ones for both banknotes (paper money or bills) and coins within ILS currency.

The machine can be controlled in two ways:

- **Mobile Application (Wireless Control)**: for administration to allow the admin to update the currencies' states that are stored in a file in the flash memory of ESP32 microcontroller.
- **Manual Control**: for normal users to process and monitor the transactions.

There are some images for LCD in the following scenarios:

4.1 General Menus

Figure 4.1: Main Menu

Figure 4.2: Breaking and Consolidating Menu

Figure 4.3: Coins Menu

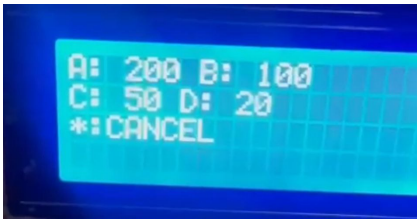

Figure 4.4: Cash Menu

4.2 Breaking Banknotes (Paper Money or Bills)

If the user wants to break 100 ILS to 2X100:

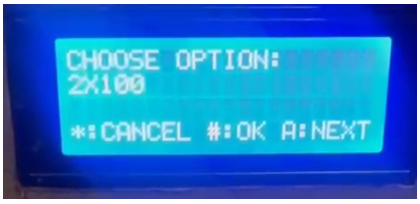

Figure 4.5: Selecting the option

Figure 4.6: Confirming the selection

If the user wants to break 50 ILS to 2X20 + 1X10:

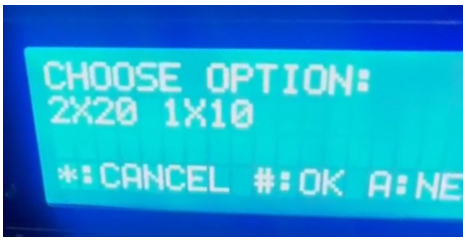

Figure 4.7: Breaking 50 ILS

If the user wants to break 20 ILS to 2X10 or (1X10 + 2X5):

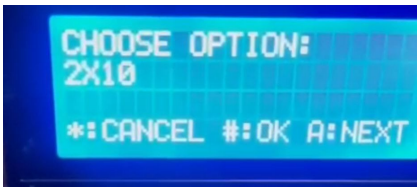

Figure 4.8: Breaking 20 ILS to 2X10

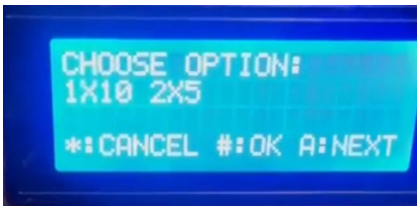

Figure 4.9: Breaking 20 ILS to (1X10 + 2X5)

4.3 Consolidating Banknotes (Paper Money or Bills)

If the user wants to consolidate 20 ILS:

Figure 4.10: Consolidating 20 ILS - Step 1

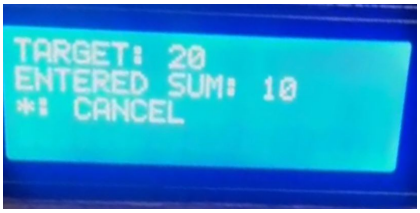

Figure 4.11: Consolidating 20 ILS - Step 2

4.4 Breaking Coins

If we try to break 10 ILS, we could have one or more options according to the available currencies in the system:

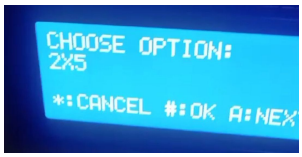

Figure 4.12: Breaking 10 ILS - Only one option

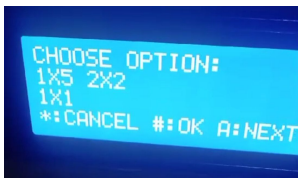

Figure 4.13: Breaking 10 ILS - Multiple options

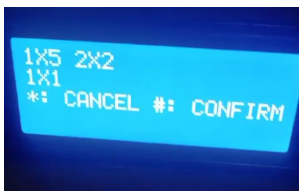

Figure 4.14: Confirm the selection

4.5 Consolidating Coins

If we try to consolidate 10 ILS:

Figure 4.15: Consolidating 10 ILS - Step 1

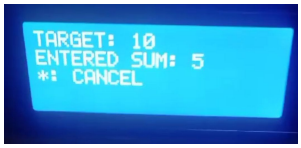

Figure 4.16: Consolidating 10 ILS - Step 2

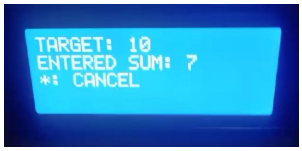

Figure 4.17: Consolidating 10 ILS - Step 3

4.6 Feedback Messages

Figure 4.18: Coins Menu

Figure 4.19: Cash Menu

Chapter 5

Discussion

5.1 Working Principle

5.1.1 Input Mechanism

- **Banknotes (paper money or bills)**: After entering the banknote from the top thin horizontal slot, the color sensor will detect it and a servo motor will rotate 110°to enter the banknote to the system.
- **Coins**: Using the coins acceptor CH-926 itself as explained in 3.2.6.

5.1.2 Output Mechanism

- **Banknotes (paper money or bills)**:
  1. A DC motor will operate the 4 vacuums.
  2. A stepper motor connected to a gear-box will move 1100 steps (5.5 full-cycles) to position the arm that holds the 4 vacuums on the banknotes. A limit switch will ensure the stepper motor moves the required steps and reaches the reference endpoint.
  3. One of the 4 mechanical relays will enable the corresponding vacuum to pick up a banknote.
  4. The stepper motor will rotate again 1100 steps (5.5 full-cycles) to allow the arm that holds the 4 vacuums to move the picked banknote.
  5. The picked banknote will be passed to the user through one of the output horizontal slots using a paper feed roller connected to another DC motor.
- **Coins**: The 3D-printed slots for coins are connected to 4 barrier which are connected to 4 servo motors. The servo motor rotates 180°to push the barrier. So, the coin can go outside to the user.

5.1.3 Detecting Banknotes (Paper Money or Bills)

As explained in 3.2.3, the color sensor contains current to frequency converter. The following table shows the RGB bands of each banknote type:

Banknote Type	Red Band	Green Band	Blue Band
200 ILS	70 - 80	55 - 70	35 - 45
100 ILS	40 - 50	60 - 70	55 - 65
50 ILS	55 - 65	50 - 60	55 - 65
20 ILS	40 - 50	50 - 60	40 - 50

5.1.4 Detecting Coins

The coin acceptor CH-926 itself detects the entered coins as explained in 3.2.6.

CHALLENGE : How to remove the false-positive detection? for e.g. Detecting 10 ILS (getting 5 impulses from the coin acceptor CH-926) as 1 ILS by taking only the first two impulses.

SOLUTION : Checking the impulses' counter after 30 - 40 iterations (Given the pulse length is fast (30 ms)) to ensure all the impulses are arrived. The best way to implement this is to define a counter to avoid using the blocking delay.

5.1.5 Transaction Processing

There are the following options in the system:

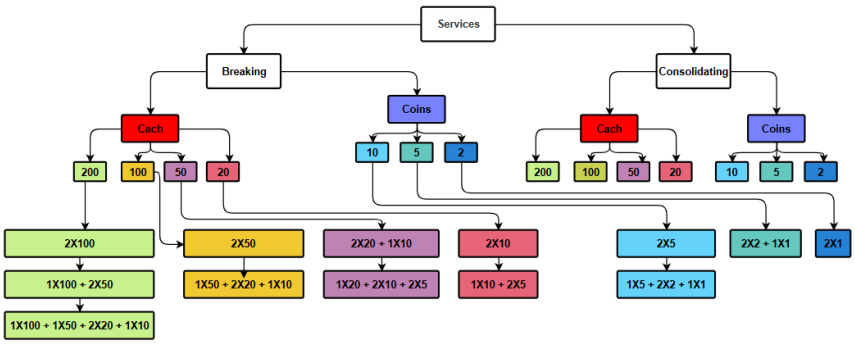

Figure 5.1: Transactions' Options

The system shows only the available options by comparing all the options with the available currencies data.

After each transaction, the system will update the number of available currencies for each type by removing the amount of output currencies in the transaction

itself for both breaking and consolidating exchanges.

For the amount of input currencies, the design requires the admin to classify them and update the amount of the available currencies for each type at the end of day (for e.g.) using the mobile application.

5.2 Mobile Application GUI

The mobile application is used to allow the admin to update the currencies' states (edit the quantities of each currency type) that are stored in a text file in the flash memory of ESP32 microcontroller as the following:

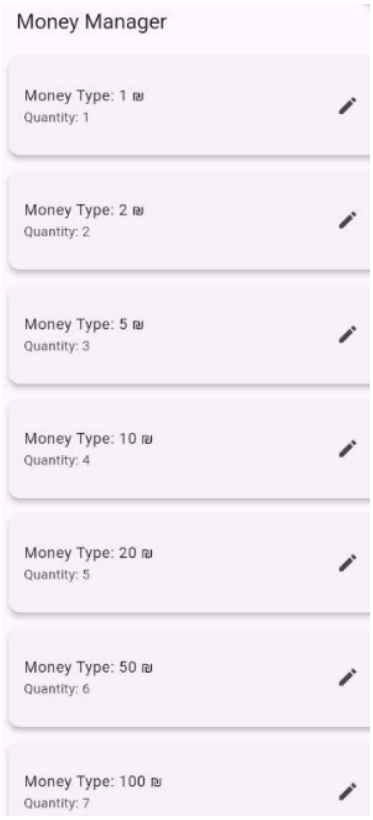

Figure 5.2: Main Screen

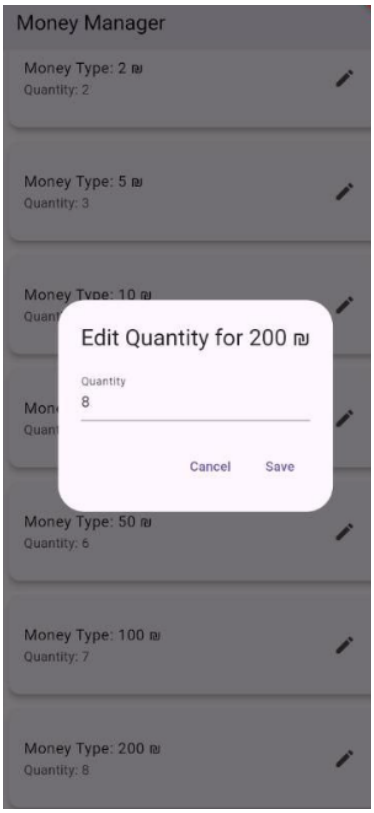

Figure 5.3: Editing Modal

It uses the following end-points from RESTful API on ESP32:

- `GET: http:\\get-data`
- `POST: http:\\edit-data`

Chapter 6

Conclusion, Recommendations and Future Work

6.1 Conclusion

We developed a machine that supports both **Breaking Exchange** which allows users to break larger currency into smaller ones and **Consolidating Exchange** which allows users to combine smaller currencies into larger ones for both banknotes (paper money or bills) and coins within ILS currency.

We offered two methods for controlling the system: a mobile application in a wireless (remote) manner for administration to allow the admin to update the currencies' states that are stored in a file in the flash memory of ESP32 microcontroller and a manual hardware installed on the machine itself for normal users.

6.2 Recommendations

We suggest that the one who intends to engage in similar projects first analyze the system in all its details, determine all the methods that he will use in the processes of inserting currencies, identifying them and removing them, and taking all of these considerations into the design before starting it. To avoid choices that the design itself can impose on the project.

We also suggest specifying the features that the project should include to avoid trading off software (Flexibility, Complexity Management and Cost-Effectiveness) versus hardware (Performance, Security and Reliability).

6.3 Future Work

The current system can be improved with several ideas:

- Detecting counterfeit currencies to make the system more realistic by changing the technique of detecting the currencies since both size and color cannot detect counterfeit currencies.
- Improving the design by adding a smart control panel (iPad for e.g.) instead of the LSD and keypad.
- Making the system support more currencies (USD and euro) in both breaking exchange and consolidating exchange.
- Adding a completely new feature to the system, which is the ability to convert between different currencies using the latest exchange rates from external APIs. This feature will utilize existing hardware for coins in cases where the process conversion needs them.

References

[1] "Global tourism set for full recovery by end of the year with spending growing faster than arrivals - un tourism." https://www.unwto.org/news/global-tourism-set-for-full-recovery-by-end-of-the-year-with-spending-growing-faster-than-arrivals, 2024.

[2] "Foreign exchange market size & share analysis - growth trends & forecasts (2025 - 2030) - modor intelligence." https://www.mordorintelligence.com/industry-reports/foreign-exchange-market, 2024.

[3] "The world of atms - tapix." https://www.tapix.io/resources/post/the-world-of-atms-why-data-and-cash-still-go-hand-in-hand, 2024.

[4] "7 human error statistics for 2025 - docuclipper." https://www.docuclipper.com/blog/human-error-statistics/, 2025.

[5] "Currency/money exchange machine by hongzhou smart." https://www.hongzhousmart.com/currency-exchange-kiosk-manufacturer.html.

[6] "Ccash exchange machine - 4300." https://www.glory-global.com/en-us/products-and-services/hardware/deposit-solutions/ba4300/.

[7] "Currency detection techniques." https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0506/s0128541.pdf.

[8] "How do counterfeit detectors work — counterfeit detection - safe scan." https://www.safescan.com/en-us/counterfeit-detection.

[9] J. W. Lee, H. G. Hong, K. W. Kim, and K. R. Park, "A survey on banknote recognition methods by various sensors," *Sensors*, vol. 17, no. 2, p. 313, 2017.

[10] A. Mukundan, Y.-M. Tsao, W.-M. Cheng, F.-C. Lin, and H.-C. Wang, "Automatic counterfeit currency detection using a novel snapshot hyperspectral imaging algorithm," *Sensors*, vol. 23, no. 4, p. 2026, 2023.

[11] M. Laavanya and V. Vijayaraghavan, "Real time fake currency note detection using deep learning," *Int. J. Eng. Adv. Technol.(IJEAT)*, vol. 9, 2019.

[12] Q. Zhang, W. Q. Yan, and M. Kankanhalli, "Overview of currency recognition using deep learning," *Journal of Banking and Financial Technology*, vol. 3, pp. 59–69, 2019.

[13] "Arduino mega 2560 documentation." https://docs.arduino.cc/hardware/mega-2560/.

[14] "Esp32 datasheet." https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[15] "Dc motor - byjus." https://byjus.com/physics/dc-motor/#dc-motor-diagram.

[16] "Stepper motor basics - oriental motor." https://www.orientalmotor.com/stepper-motors/technology/stepper-motor-basics.html.

[17] "Servo motor - circuit digest." https://circuitdigest.com/article/servo-motor-working-and-basics.

[18] "Tcs230/tcs3200 color sensor - random nerd tutorials." https://randomnerdtutorials.com/arduino-color-sensor-tcs230-tcs3200/.

[19] "4 * 4 matrix keypad - electronic wings." https://www.electronicwings.com/sensors-modules/4x4-keypad-module.

[20] "16 * 4 lcd - winstar." https://www.winstar.com.tw/products/character-lcd-display-module/lcd-16x4.html.

[21] "Tb6600 stepper motor driver with arduino tutorial - makerguides." https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/.

[22] "Arduino limit switch tutorial - the geek pub." https://www.thegeekpub.com/278078/arduino-limit-switch-tutorial/.

[23] "Understanding how a single channel relay module works and how to use it with arduino to control ac loads - circuit digest." https://circuitdigest.com/microcontroller-projects/interface-single-channel-relay-module-with-arduino.

[24] "How to interface ch926 multi coin acceptor with arduino uno - quartz components." https://quartzcomponents.com/blogs/electronics-projects/how-to-interface-ch926-multi-coin-acceptor-with-arduino-uno.

[25] "Arduino ide website." https://www.arduino.cc/en/software.

[26] "Cp210x usb to uart bridge vcp drivers website." https://www.silabs.com/developer-tools/usb-to-uart-bridge-vcp-drivers.

[27] "Flutter website." https://flutter.dev/.

[28] "Visual studio code website." https://code.visualstudio.com/.

[29] "Android studio website." https://developer.android.com/studio.

[30] "Postman website." https://www.postman.com/.

[31] "Git website." https://git-scm.com/.

[32] "Github website." https://github.com/.

[33] "Dart website." https://dart.dev/.

Appendix A

Arduino Mega 2560 Source Code

Available upon request.

Appendix B

ESP32 Source Code

Available upon request.

Appendix C

Mobile Application Code

Available upon request.