بسم الله الرحمن الرحيم

AN-NAJAH NATIONAL UNIVERSITY

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

Computer Engineering Department

Hardware Graduation Project

Smart Shopping Cart

students
Aya Qartallo
Rand Ismail

Advisors
Dr. Luai Malhis

Presented in partial fulfillment of the requirements for Bachelor degree in computer engineering.

Aug,2022

Acknowledgement

We express our sincere gratitude to Allah and to our supervisor, Dr. Luai Malhis, for his guidance, and we thank our family and friends for their psychological support.

Disclaimer

We are a Computer Engineering Students from An-Najah National University Wrote this report. It may contain grammatical and content mistakes because it has not been changed or corrected—only editorial changes have been made—as a result of the evaluation. The opinions expressed in it, along with any conclusions and advice, are all purely the students'. An-Najah National University disclaims all liability and responsibility for the results of using this study for purposes other than those for which it was commissioned.

Contents

List of Figures

Abstract

Nowadays, the world is facing an evolution in technology to make their lives easier and save time. Since shopping is an essential and almost daily thing in people's lives, the project aims to facilitate the shopping and reduce congestion and the time that the user spends at the cashier. The idea of the project is a smart shopping cart that follows the customer and if you move away from him, it alert him by launching a sound using the buzzer and reads the ID of the cart, the price of the product through the barcode reader, and the weight of the product through the balance (HX711), then sends it via Wi-Fi to the cashier (server) after confirming the purchase process, which is done through the keyboard in the cart. On the cashier side, he will compare the weight coming from the cart via Wi-Fi and the actual weight of the products to avoid the occurrence of theft cases. In addition, he can delete certain products that appear on the screen LCD through the ID of the cart. The employee at the cashier takes the purchase invoice from the customer. Upon completion of the purchase and payment process, the employee deletes the list of purchases from the cart. The idea has been done before but with different features, and this year, Amazon has made a smart shopping cart similar to the idea of our project, in addition to other features.

1 Introduction

1.1 Statement of the problem

In the modern era, large buildings and large and spacious facilities from the economic point of view have become important elements and features of the era, as it has become more common for the spread of malls, supermarkets and major shopping stores that occupy large areas. With the spread of the use of technology and computerized techniques and the increase in their importance and reliance on them, we came up with an idea to create a smart shopping cart, among its advantages: reducing the crisis, the number of employees and saving time. It can be benefited from and applied in Palestine in the huge shopping places, as it has not been applied in Palestine until now.

1.2 Objectives

The smart shopping cart aims to save time and alleviate the crisis at the cashier, and this is done when he presses the button "confirm", and the bill is transmitted through WiFi at the cashier. The customer only has to go to the cashier and pay the bill without having to wait for the worker at the cashier to calculate the bill for all products.It is also intended for the convenience of the buyer and does not push the cart while he walks, but the cart follows him.

1.3 Significance of the work:

The smart shopping cart is intended for large shopping centers and supermarkets, which are characterized by a large number of people. This will help speed up the shopping process by reducing congestion and also benefits people who find it difficult to push the cart, as in people with special needs.

1.4 Organization

This report has organized as follows: We explained about the idea, its importance and scope of work. Then the constraints we faced during our work including equipment, tools we used and earlier course work. In addition to finding similar systems to our project with different features, you can read about them to get a background on the project.We also explained the methodology of our work in addition to discussing the results we obtained.Ending it up with the conclusion of the whole work and what is our vision for the future to improve our work.

2 Constraints and Earlier Coursework

2.1 Constraints Limitations

In general, the equipment was not quite as good as we need it to be. Initially, when we ran Load Cell HX711 we did not get similar readings for the same body weight (different decimals) we solved this problem by taking the approximate value of the weight. Some routers isolate devices on the same network from each other. The IR sensor is affected by sunlight and does not measure large distances.

2.2 Earlier coursework

Working on our project requires knowledge from some of the courses we took in the computer engineering program, including:

Microcontroller: In the Microcontroller course, we learned basic information about a PIC microcontroller and how to program hardware components. Also, in the PIC lab we learned how to download the code to the PIC microcontroller and understand each pin and every feature in it. As a result, this course helped us understand how to deal with the ESP32 in the project, so this contributed mainly to help us start working on the project.

Electronic circuits: This course helped us to understand electrical circuits and how to deal with them and to identify the connections of these circuits

3 Literature Review

A visit to the supermarket can be nerve-wracking and time-consuming. The Smart Shopping Cart is a Cart used by people in large supermarkets.

The Smart Shopping Cart is a cart with an LCD screen that displays the name of the product and its price when reading the barcode of the product through the barcode scanner and the keyboard that enables you to move between pages, delete a product and confirm the order. When the order is confirmed, the information on the LCD screen is sent via Wi-Fi to the server at the cashier. It will solve the problem of waiting in a long queue, as the trolley is equipped with a scale to avoid theft cases, and the customer does not need to push the trolley, as it tracks its owner.

In other systems, the smart shopping cart is equipped with an RFID reader which is much easier, as all items can be automatically read by an RFID reader instead

of manually scanned by a laborer.(1)

These project and others in this field are interested in reading product information and sending it via Wi-Fi to the server or paying by credit card without having to go to the cashier to pay.

4 Methodology

4.1 Hardware parts

4.1.1 Overview

Tools and hardware components we used:

- ESP32 Microcontroller
- Barcode Scanner
- LCD 20x4 I2C
- Keypad 4x4
- 5kg Load Cell HX711
- IR Sensor Transmitter and Receiver
- 5V Continuous Piezo Buzzer
- DC Motor
- L298N Dual H-Bridge Motor Drive Module Board
- LM2596 Adjustable Step Down Buck Converter Module 3
- 18650 Original Li-lon Battery 1600mAh 3.7V and 18650 3.7V Battery Parallel Case

4.1.2 Description

Brief explanation and definition of each piece that was used:

- **ESP32 Microcontroller:** It is a microcontroller board such as Arduino with integrated Wi-Fi and dual-mode Bluetooth. It has a 38 pin. It is low cost and low power system. (2)

Figure 1: ESP32

- **Barcode Scanner** optical scanner that can read printed barcodes, decode the data contained in the barcode.

Figure 2: Barcode Scanner

- **LCD 20x4 I2C:** There are 4 rows in display and in 1 row 20 character can be displayed and in 1 display 80 characters can be shown. This liquid crystal module uses HDD44780 (It is a controller used to display monochrome text displays) parallel interfacing. The liquid crystal display interfacing code is easily accessible. We just required 11 input and output pinouts for the interfacing of the LCD screen.The input supply for this module is 3.3 volts or 5 volts. Using I2C with only 3 pins from microcontroller, you can display message on this LCD. Compared to parallel LCD which required at least 6 pins of I/O, this LCD offer more cost effective solution.

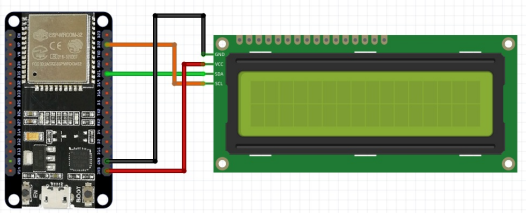


Figure 3: LCD 20x4 I2C

- **Keypad 4x4:** 4X4 Keypad modules are available in different sizes and shapes. It has 8 terminals. In them 4 are rows of matrix and 4 are columns of matrix. These 8 pins are driven out from 16 buttons present in the module. Those 16 alphanumeric digits on the module surface are the 16 buttons arranged in MATRIX formation. (3)

Figure 4: Keypad 4x4

- **5kg Load Cell HX711:** It is measure the weight produced by the load, here most load cells are following the method of a strain gauge, Which converts

the pressure (force) into an electrical signal, these load cells have four strain gauges that are hooked up in a Wheatstone bridge formation. (4)

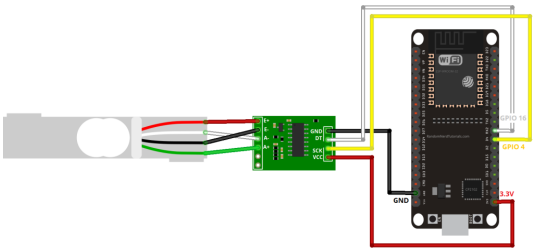


Figure 5: Load Cell HX711

- **5 Channel IR flame Detector Sensor:** This module outputs analog signal, which would be more precisely, and also digital signal which would be more easy to use, you can adjust the digital output sensitivety by the on-board potentiometer. The 5 LED indicators are helpful in your debugging, the high-precision resistors (1

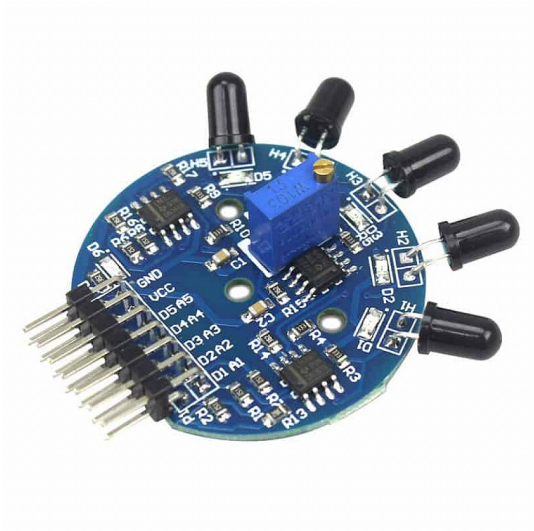


Figure 6: 5 Channel IR flame Detector Sensor

- **5V Continuous Piezo Buzzer:** It has two wires for connection and can work on 3 to 7 V DC. Just connect with power supply and it will give loud sound. Light weight, lightweight construction and low cost. (5)

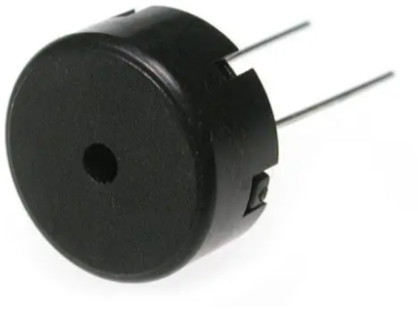


Figure 7: 5V Continuous Piezo Buzzer

- **DC Gear Motor:** The 12V DC gear-motor is a powerful motor to drive the position control of the system. It provides 60 rpm with 12VDC rated voltage and gear ratio is 100:1. In this system, counts per revolution of the gearbox output can be multiplied the gear ratio by 22. (6)

Figure 8: DC Gear Motor

- **L298N Dual H-Bridge Motor Drive Module Board:** It is a high-power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator.L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control. (7)

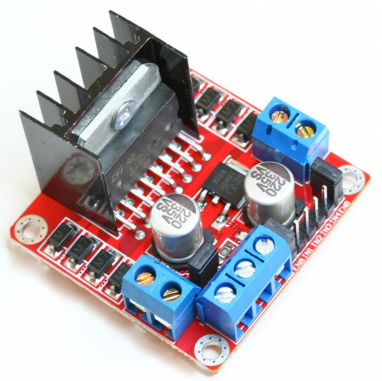


Figure 9: L298N Dual H-Bridge Motor Drive Module Board

- **LM2596 Adjustable Step Down Buck Converter Module 3:** DC-DC Buck Converter Step Down Module LM2596 Power Supply is a step-down (buck) switching regulator, capable of driving a 3-A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, and an adjustable output version. (8)

Figure 10: LM2596 Adjustable Step Down Buck Converter Module 3

- **18650 Original Li-Ion Battery 1600mAh 3.7V and 18650 3.7V Battery Parallel Case:** It is high-energy type lithium-ion battery with high capacity of 2.6 Ah, it is high-power type lithium-ion battery with low capacity of 1.5 Ah6. (9)

Figure 11: 18650 Original Li-lon Battery 1600mAh 3.7V and 18650 3.7V Battery Parallel Case

4.1.3 Hardware Development

To build this project, we went through the following steps:

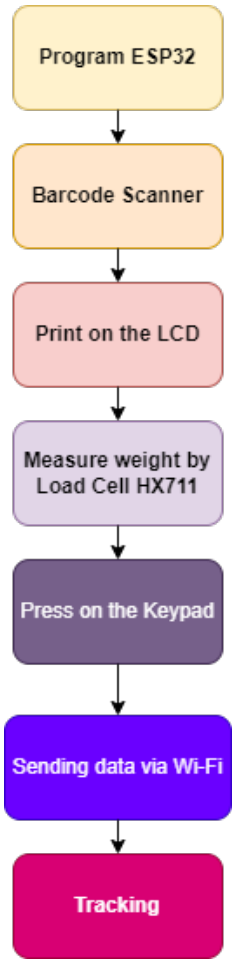


Figure 12: Cart Design

4.1.4 Overall Design

Figure 13: Cart Design

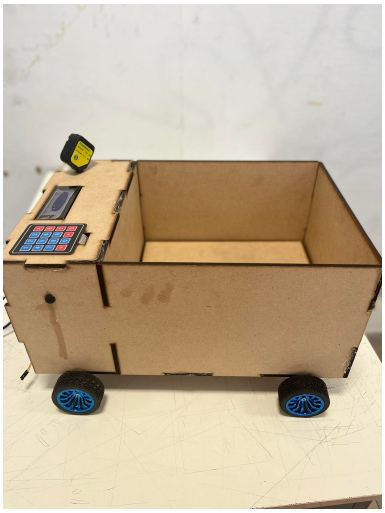


Figure 14: Cart Design

Figure 15: Cart Design

Figure 16: Cart Design

4.2 Software Implementation

```
void setup() {
  // put your setup code here, to run once:
  lcd.init(); // initialize the lcd
  lcd.backlight();
  BCSerial.begin(115200, SERIAL_8N1, 16, 17); // the GPRS/GSM baud rate
  Serial.begin(115200);
  delay(4000);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
  scale.set_scale(calibration_factor); //This value is obtained by using the SparkFun_HX711_Calibra
  scale.tare(); //Assuming there is no weight on the scale at start up, reset the scale to 0
  initMotors();
  pinMode(buzzerPin, OUTPUT);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());

  lcd.setCursor(0,0);
  lcd.print("Product:  Price:");
  lcd.setCursor(17,3);
  lcd.print("1/4");

}
```

Figure 17: Setup Function

After we read barcode of the products and the weight, we press on key 8 on the keypad to confirm the process that send this data to the server via Wi-Fi. If the URL correct then the website open if httpCode that received from server is less than 0 then the process failed.

```
void sendToServer(){
  for(int q=0;q<ARRAYSIZE;q++){
      results[q].trim();
  }
  String server = HOST_NAME + PHP_FILE_NAME +"?idcart=1"+"&barcode1="+results[1]+"&barcode2="+results[2]+"&barcode3="+results[3]+"&barcode4="+res
  http.begin(server);
  Serial.println(server);
  int httpCode = http.GET();

  if(httpCode > 0) {
    if(httpCode == HTTP_CODE_OK) {
      String payload = http.getString();
      Serial.println(payload);

    } else {
      Serial.printf("HTTP GET... code: %d\n", httpCode);
    }
  } else {
    Serial.printf("HTTP GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
  }
}
```

Figure 18: Send to server code

In the first part, We read the barcodes and store them in array of string then print the name of the product and the price on the LCD.
In the second part, read the weight of the products and take the round for it to take the exact value.

```
void loop() {
  // put your main code here, to run repeatedly:
  if(BCSerial.available()>0){//read the barcode
    BCdataIn=BCSerial.readString();
    //Serial.println(BCdataIn);
    results[i]=BCdataIn;
    results[i].trim();

      Serial.println(results[i]);
      printDescription(BCdataIn); //from index=0 start print barcodes
    //Serial.print(results[i]);
    i++;
    }

    if (scale.is_ready()) {//reading the weight
    long reading = scale.read();
    Serial.print("HX711 reading: ");
    weight=round(scale.get_units(5) * 10);
    Serial.println(weight);
  } else {
    Serial.println("HX711 not found.");
  }
```

Figure 19: read barcode and weight

In this function we compare the barcode with the barcodes of the product that already stored in array of barcodes then when the barcodes matches the product name and price that correspond in array of product names and array of prices are print on the LCD.

```
void printDescription(String dta){
  for(int g=0;g<arrsize;g++){
       bcodes[g].trim();
       dta.trim();
      if(bcodes[g].equals(dta)){
        if(c1==0||c1==1){ // c1 counter for page1 which contain 2 lines for print
          tok1[f]=pname[g];//stored name to print(retrieve) it when i press next or b
          tok1[f].trim();
          f++;
          tok1[f]=price[g];//stored price to print(retrieve) it when i press next or
          tok1[f].trim();
          f++;
          page0(pname[g],price[g]); //print in page 1
          }
          else if(c1>1&&c2<3){// c2 counter for page2 which contain 3 lines for print
            tok2[f]=pname[g];
            tok2[f].trim();
            f++;
            tok2[f]=price[g];
            tok2[f].trim();
            f++;
            page1(pname[g],price[g]);//print in page 2
            }
            else if(c1>1&&c2>2&&c3<3){// c3 counter for page3 which contain 3 lines f
               tok3[f]=pname[g];
```

Figure 20: Print on LCD

In the first part, when press key 4 on the keypad we will read the barcode of the product that we need to delete then call function deleteItms>
In the second part, clear the LCD and the array after we press key 5 on the keypad.

```
if(key=='9'){//delete
  arrsize2=(int)sizeof(results)/sizeof(results[0]);
    BCdataDel=BCdataIn;
    Serial.println(BCdataIn);
    del_index=i-1;
   deleteItems();
}
if(key=='5'){//disconnect
  delay(10);
  lcd.clear();
  for(int w=0;w<ARRAYSIZE;w++){
    results[w]=" ";
  }
  for(int w=0;w<ARRAYSIZE;w++){
    Serial.println(results[w]);
  }
  http.end();
  Serial.println("WiFi Disconnected..");
}
```

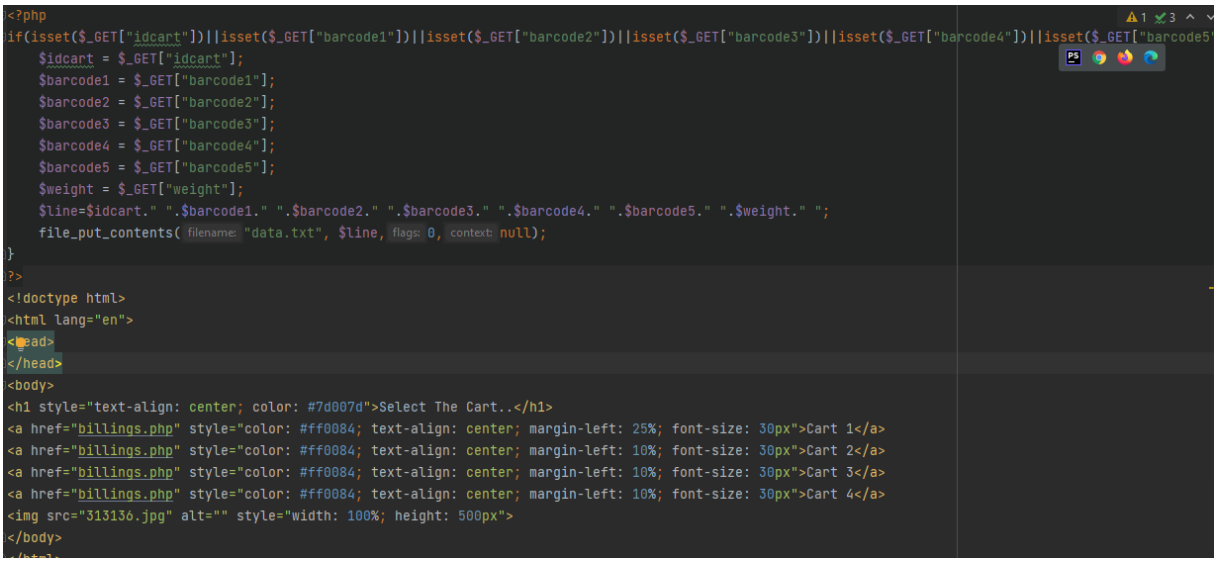
Figure 21: Delete and Disconnect

In the first part, after reading the barcode of the product that we need to delete then comparing it with barcodes in the array that we read when find it, replace it with space and shift the barcodes. Finally print the name of the product and price of the new array on the LCD.
In the second part, clear the array after press key 5 on the keypad.

```
void deleteItems(){
   results[del_index]=" ";
   Serial.println("Item deleted");
  for(int t=0;t<arrsize2;t++){
    Serial.print(results[t]);
    printDescription(results[t]);
  }
}
void clearArray(){
  for(int s=0;s<arrsize;s++){
    results[s]=" ";
  }
}
```

Figure 22: Delete and clear Functions

Here when the values of barcodes and weight revceive the server, the values stored in file. In this page there is 4 buttons each button for cart when click on one of them its going to these page.



```php
<?php
if(isset($_GET["idcart"])||isset($_GET["barcode1"])||isset($_GET["barcode2"])||isset($_GET["barcode3"])||isset($_GET["barcode4"])||isset($_GET["barcode5"
    $idcart = $_GET["idcart"];
    $barcode1 = $_GET["barcode1"];
    $barcode2 = $_GET["barcode2"];
    $barcode3 = $_GET["barcode3"];
    $barcode4 = $_GET["barcode4"];
    $barcode5 = $_GET["barcode5"];
    $weight = $_GET["weight"];
    $line=$idcart." ".$barcode1." ".$barcode2." ".$barcode3." ".$barcode4." ".$barcode5." ".$weight." ";
    file_put_contents( filename: "data.txt", $line, flags: 0, context: null);
}
?>
<!doctype html>
<html lang="en">
<head>
</head>
<body>
<h1 style="text-align: center; color: #7d007d">Select The Cart..</h1>
<a href="billings.php" style="color: #ff0084; text-align: center; margin-left: 25%; font-size: 30px">Cart 1</a>
<a href="billings.php" style="color: #ff0084; text-align: center; margin-left: 10%; font-size: 30px">Cart 2</a>
<a href="billings.php" style="color: #ff0084; text-align: center; margin-left: 10%; font-size: 30px">Cart 3</a>
<a href="billings.php" style="color: #ff0084; text-align: center; margin-left: 10%; font-size: 30px">Cart 4</a>
<img src="313136.jpg" alt="" style="width: 100%; height: 500px">
</body>
```

Figure 23: Page 1 in the Cashier website

In this part we take the value of id cart, barcodes and weight of the products from the file.

```php
<?php
$file=file_get_contents( filename: "data.txt", use_include_path: false,  context: null, offset: 0,  length: null);
$pieces = explode( separator: " ", $file);
?>
<!doctype html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet">
    <title>Cashier</title>
</head>
<body>

<div class="container">
    <div class="row">
        <div class="col-md-12">
            <div class="card mt-4">
                <div class="card-header">
                    <h4>Billings </h4>
                </div>

            </div>
        </div>
```

Figure 24: Page 2 in the Cashier website

Take the barcodes and search about their information in the database.

```
<tr>
    <th style="..."colspan="4">Cart ID: <?php echo $pieces[0];?></th>
</tr>
<tr>
    <th>Barcode</th>
    <th>Product Name</th>
    <th>Price</th>
    <th>Weight</th>
</tr>
</thead>
<tbody>
<?php
$link = mysqli_connect( hostname: "localhost", username: "ayaq", password: "hardwareproj", database: "cash_system") or die("Error in connecting to server
$result = mysqli_query($link, query: "select * from prod_info where barcode='".$pieces[1]."' or barcode='".$pieces[2]."' or barcode='".$pieces[3]."'
$arr=array();
$arrbar=array();
$arrname=array();
$arrprice=array();
$arrweight=array();
while($row = mysqli_fetch_assoc($result)){
    foreach ($row as $value) {
        array_push( &array: $arr,$value);
    }
}
```

Figure 25: Page 2 in the Cashier website

show each barcode and its information on the website as table. Compute total price and total weight that arrived from database.

```php
for ($x = 0; $x < sizeof($arr); $x=$x+4) {
    array_push( &array: $arrbar, $arr[$x]);
    array_push( &array: $arrname, $arr[$x + 1]);
    array_push( &array: $arrprice, $arr[$x + 2]);
    array_push( &array: $arrweight, $arr[$x + 3]);
}
for($y=0; $y<sizeof($arrbar);$y++){
    ?>
    <tr>
        <td><?php echo $arrbar[$y];?></td>
        <td><?php echo $arrname[$y];?></td>
        <td><?php echo $arrprice[$y];?></td>
        <td><?php echo $arrweight[$y];?></td>
    </tr>
    <?php
}
$sum_price = array_sum($arrprice);
$sum_weight = array_sum($arrweight);


?>
```

Figure 26: Page 2 in the Cashier website

Show the Total price of the products, the actual weight and theoretical weight on the website and alert if the two weights are not the same.

```
    </tbody>
</table>
<table class="table table-bordered">
    <thead>
    <tr>
        <th>Total Price:</th>
        <td><?php echo $sum_price;?>NIS</td>
    </tr>
    <tr>
        <th>Actual Weight:</th>
        <td><?php echo $sum_weight;?>g</td>
        <th>Theoretical Weight:</th>
        <td><?php echo $pieces[6];?>g</td>
        <?php
        if($sum_weight!=$pieces[6]){
            echo '<script>alert("The Weight is not the same..")</script>';
        }

        ?>
    </tr>
    </thead>
</table>
v>
```

Figure 27: Page 2 in the Cashier website

here we have two mode, if we press on button 6 then the flag change to 0 which we can go to the purchase process. If we press on button 7, the tarcking on is ON. Configuring motor PWM functionalities to generate the signal and Attaching

```
   char key = keypad.getKey();
if (key){
     if(key=='7'){
       flag=1;
     }
     if(key=='6'){
       flag=0;
     }
   Serial.println(key);
}
```

Figure 28: check the mode

the channel to the GPIO to be controlled.

```
void initMotors()
{
  /* Configuring motor PWM functionalities to generate the signal */
  /* ledcSetup(Channel, Frequency, Resolution) */
  ledcSetup(3, 2000, 8); /* 2000 hz PWM, 8-bit resolution and range from 0 to 255 */
  ledcSetup(4, 2000, 8); /* 2000 hz PWM, 8-bit resolution and range from 0 to 255 */
  ledcSetup(5, 2000, 8); /* 2000 hz PWM, 8-bit resolution and range from 0 to 255 */
  ledcSetup(6, 2000, 8); /* 2000 hz PWM, 8-bit resolution and range from 0 to 255 */

  /* Attaching the channel to the GPIO to be controlled */
  /* ledcAttachPin(GPIO, Channel) */
  ledcAttachPin(RMotor1, 3);
  ledcAttachPin(RMotor2, 4);
  ledcAttachPin(LMotor1, 5);
  ledcAttachPin(LMotor2, 6);

}
```

Figure 29: Init the motors

Here we are reading IR value and depending on it we can go right, left, forward, backward or stop the cart. If each IR reading less than the threshold=1500, then the cart stop and turn ON the buzzer.

```
   if(flag==1){
       reading1 = analogRead(A1);
 reading2 = analogRead(A2);
 reading3 = analogRead(A3);
 reading4 = analogRead(A4);
 reading5 = analogRead(A5);
 Serial.println("reading1: " + String(reading1) + " reading2: " + String(reading2) + " reading3: " + String(reading3) + " reading4: " + String(reading4) + " readi

 if (reading1 < THR && reading2 < THR && reading3 < THR && reading5 < THR && reading5 < THR) {
   stopRobot();
   if(buzzerFlag==1){
     if(millis()-timer>2000){
   digitalWrite(buzzerPin, 1);
   //Serial.println("stopRobot");
   Serial.println("signal lost>>buzzer on");

     }
   }
 }
```

Figure 30: Reading IR Value

If IR readings out of range stop the cart and turn ON the buzzer.

```
  if (reading3 > 4090 ) {
  stopRobot();
  Serial.println("stopRobot");
  buzzerFlag=1;
  timer=millis();
}
  if (reading3 > 3000 && reading4 < 1000 && reading2 < 1000) {
  stopRobot();
  Serial.println("stopRobot");
}
else if (reading3 > 3000 && reading4 > 3000 && reading2 > 3000) {

  stopRobot();
  Serial.println("stopRobot");
}
else if (reading3 > 3000 && reading4 < 2000 && reading2 < 2000) {

  speedCar = 150;
  goForword();
  Serial.println("goForword150");
}
```

Figure 31: Some of IR's reading cases

The cases when the cart move forward.

```
else if (reading3 > 1000 && reading4 < 1000 && reading2 < 1000) {

  speedCar = 150;
  goForword();
  Serial.println("goForword150");
}

else if ((reading3 > 3500) && (reading3 - 1000 > reading4  && reading3 - 1000 > reading2)) {

  speedCar = 150;
  goForword();
  Serial.println("goForword150");
}

else if ((reading3 > 3500) && (reading3 - 1000 > reading4  && reading3 - 1000 > reading2)) {

  speedCar = 150;
  goForword();
  Serial.println("goForword150");
}
```

Figure 32: Some of IR's reading cases

The cases when the cart move left or right.

```
    else if (reading3 > 3500 && reading4 > 3000 && reading2 < 3000) {

      speedCar = 150;
      goLeft();
      Serial.println("goLeft150");
    }
    else if (reading3 > 3500 && reading4 < 3000 && reading2 > 3000) {

      speedCar = 150;
      goRight();
      Serial.println("goRight150");
    }
    else if (reading1 > 3500) {
      speedCar = 250;
      goRight();
      Serial.println("goRight250");
    }

    else if (reading5 > 3500) {
      speedCar = 250;
      goLeft();
      Serial.println("goLeft250");
    }
```

Figure 33: Some of IR's reading cases

```
else if ((reading4 > 1000 || reading5 > 1000 ) && reading4 - 200 > reading3 || reading5 - 200 > reading3) {

  speedCar = 250;
  goLeft();
  Serial.println("goLeft250");
}

else if ((reading1 > 1000 || reading2 > 1000 ) && reading1 - 200 > reading3 || reading2 - 200 > reading3) {

  speedCar = 250;
  goRight();
  Serial.println("goRight250");
}
```

Figure 34: Some of IR's reading cases

When moving forward, LED PWN for motors 3 and 6 is speed of cart=200. When moving Backward, LED PWN for motors 3 and 6 is speed of cart=200 (Rearrangement). When moving right, LED PWN for motors 3 and 5 is speed of cart=200.

```
void goForword() {
      buzzerFlag=1;
    timer=millis();
  digitalWrite(buzzerPin, 0);
  ledcWrite(4, 0);
  ledcWrite(3, speedCar);
  ledcWrite(6, speedCar);
  ledcWrite(5, 0);
}

void goBack() {
  digitalWrite(buzzerPin, 0);
  ledcWrite(3, speedCar);
  ledcWrite(4, 0);
  ledcWrite(5, 0);
  ledcWrite(6, speedCar);

}

void goRight() {
  digitalWrite(buzzerPin, 0);
  ledcWrite(4, 0);
  ledcWrite(3, speedCar);
  ledcWrite(6, 0);
  ledcWrite(5, speedCar);
}
```

Figure 35: movement directions

When moving left, LED PWN for motors 4 and 6 is speed of cart=200.When stop, LED PWN for all motors is zero.

```
void goLeft() {
  digitalWrite(buzzerPin, 0);
  ledcWrite(4, speedCar);
  ledcWrite(3, 0);
  ledcWrite(6, speedCar);
  ledcWrite(5, 0);
}



void stopRobot() {
  ledcWrite(3, 0);
  ledcWrite(4, 0);
  ledcWrite(5, 0);
  ledcWrite(6, 0);
}
```

Figure 36: movement directions

5 Results and Discussion

In the final stage, our project fulfilled all the requirements of reading the barcodes of the products, printing them on the LCD and sending them via Wi-Fi to the server. It also fulfilled the feature of calculating the weight of the products in the cart, in addition to the feature of tracking the customer in the supermarket. end

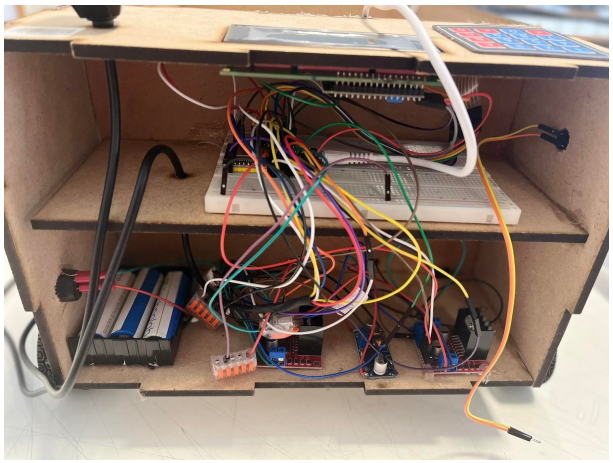


Figure 37: Cart Connections

The figure below shows Wi-Fi connection.

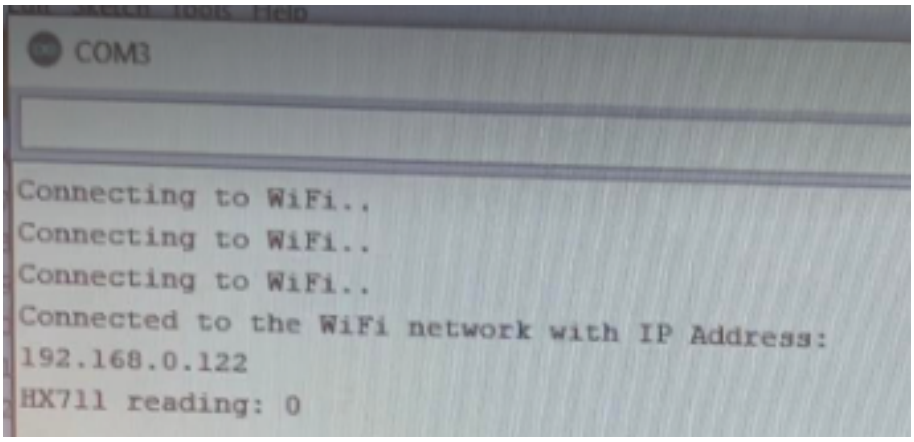


Figure 38: Wi-Fi connection

The figure below shows Load Cell HX711(Weight) and Barcode Scanner readings.

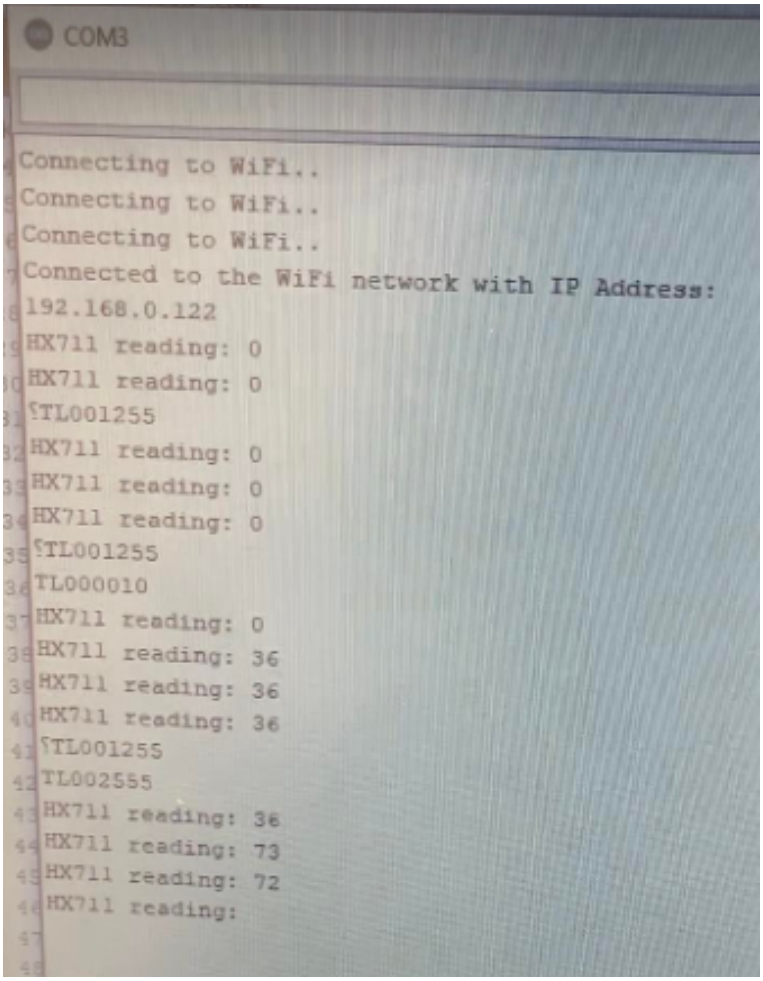


Figure 39: Load Cell HX711(Weight) and Barcode Scanner readings

The figure below shows Display The name of the product and price on the LCD.

Figure 40: Display on LCD

The figure below shows page 1 of the website in the cashier side.

Figure 41: Page 1

The figure below shows page 2 of the website in the cashier side and shows that there is no barcode didn't scan.

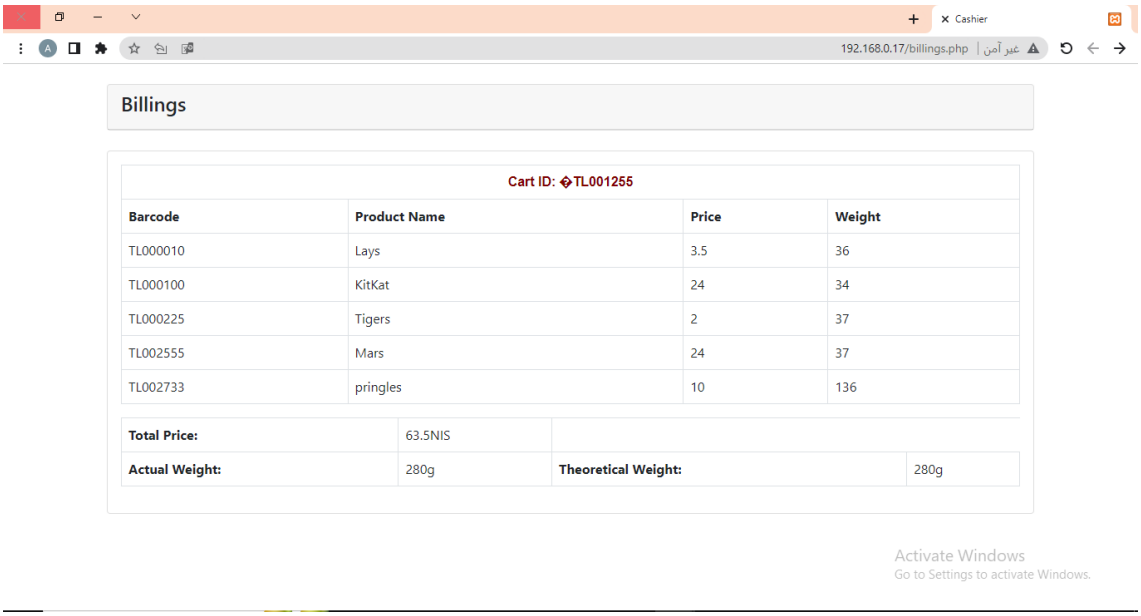


Figure 42: Page 1

6 Conclusion

6.1 Summary

In conclusion, the smart shopping cart is a prototype that helps customers in the supermarket to save their time and budget and reduce the hassle at the cashier. The smart shopping cart reads the products and sends them via Wi-Fi to the server. It also tracks the customer and is equipped with a scale to avoid cases of theft. During our work on the project we learned many things about the devices and how to connect them in the right way to avoid damaging them. Learn about esp32 and how to program and use it. We wish we had more time than we have to make more features.

6.2 Future Work

In the future, we aspire to make payment by credit card, so there is no need for the buyer to go to the cashier to pay. In addition, the cart tracks its owner through image processing, and the cart returns to its designated place after the shopping process is completed. We will also add a place to the cart for the buyer to put his own belongings in it without the need for him to carry it.

References

[1] R. Li, T. Song, N. Capurso, J. Yu, J. Couture, and X. Cheng, "Iot applications on secure smart shopping system," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1945–1954, 2017.

[2] A. Sharp and Y. Vagapov, "Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things," 2017.

[3] E. Orji, U. Nduanya, and C. Oleka, "Microcontroller based digital door lock security system using keypad," *International Journal of Latest Technology in Engineering, Management & Applied Science*, vol. 8, no. 1, pp. 92–97, 2019.

[4] V. Itikala, "Arduino weighing machine using load cell and hx711 module," *Available at SSRN 3918720*, 2021.

[5] N. J. CHRISTOPHER *et al.*, "Transformer monitoring and control using gsm and current sensor," 2020.

[6] T. M. Kyi and K. Z. Latt, "Camera movement control using pid controller in labview [j]," *Journal of Trend in Scientific Research and Development*, vol. 3, no. 5, pp. 519–523, 2019.

[7] P. Peerzada, W. H. Larik, and A. A. Mahar, "Dc motor speed control through arduino and l298n motor driver using pid controller," *International Journal of Electrical Engineering & Emerging Technology*, vol. 4, no. 2, pp. 21–24, 2021.

[8] K. K. Khaing, T. T. Nwet, and T. Naing, "Dancing robot using arduino," Ph.D. dissertation, MERAL Portal.

[9] Q. Huang, *Development of Stable High-capacity Silicon-based Anode and Lithium Metal Anode for Lithium-ion Battery*. The Pennsylvania State University, 2018.