



# **NED UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

**EE-21322**

**MUHAMMAD UZAIR IRFAN**

Smart Trolley  
Billing System

**CEP Report Submitted to**

**Instructor: Ms. Mehwish Jawed**

**Course Code: Tc-306**

**Semester: 6th (Spring-2024), TE**

**Department: Electrical Engineering**

**Credit Hours: 3 (Theory)**

**Batch: 2021**

# **Topic: Smart Trolley Billing System For Malls & Marts**

## **Introduction**

The Smart Trolley Billing System project is an innovative application designed to streamline the shopping experience by integrating Radio Frequency Identification (RFID) technology with an Arduino-based system. This project aims to automate the billing process in retail environments, allowing for quick and efficient transactions without the need for traditional checkout lines. By utilizing RFID tags attached to items, an RFID reader, an LCD display, and various input components such as buttons and a buzzer, the system provides a user-friendly and effective solution for both customers and retailers. This project not only enhances the shopping experience by reducing wait times but also demonstrates the practical implementation of communication systems, digital modulation techniques, and real-time processing. The Smart Trolley Billing System project embodies the principles of modern communication technologies, showcasing how they can be applied to solve real-world problems in a retail setting.

## **Working Principle of the Project:**

The Smart Trolley Billing System is designed to automate the billing process in retail stores using RFID technology. The system comprises an RFID reader, LCD display, buttons for item removal and purchase completion, a buzzer for user feedback, and a servo motor to control the trolley gate for exiting. When a customer places an item in the trolley, the RFID reader scans the item's tag, and the system updates the bill accordingly. The LCD displays the item details and total bill amount. Button 1 & 2 are used to add or remove items, and Button 3 is for Another counter (like J.) and Button 4 is for Reset. and last Button which is 5<sup>th</sup> one used for exiting with record, the servo motor opens the trolley gate for checkout if customer have paid their bill.

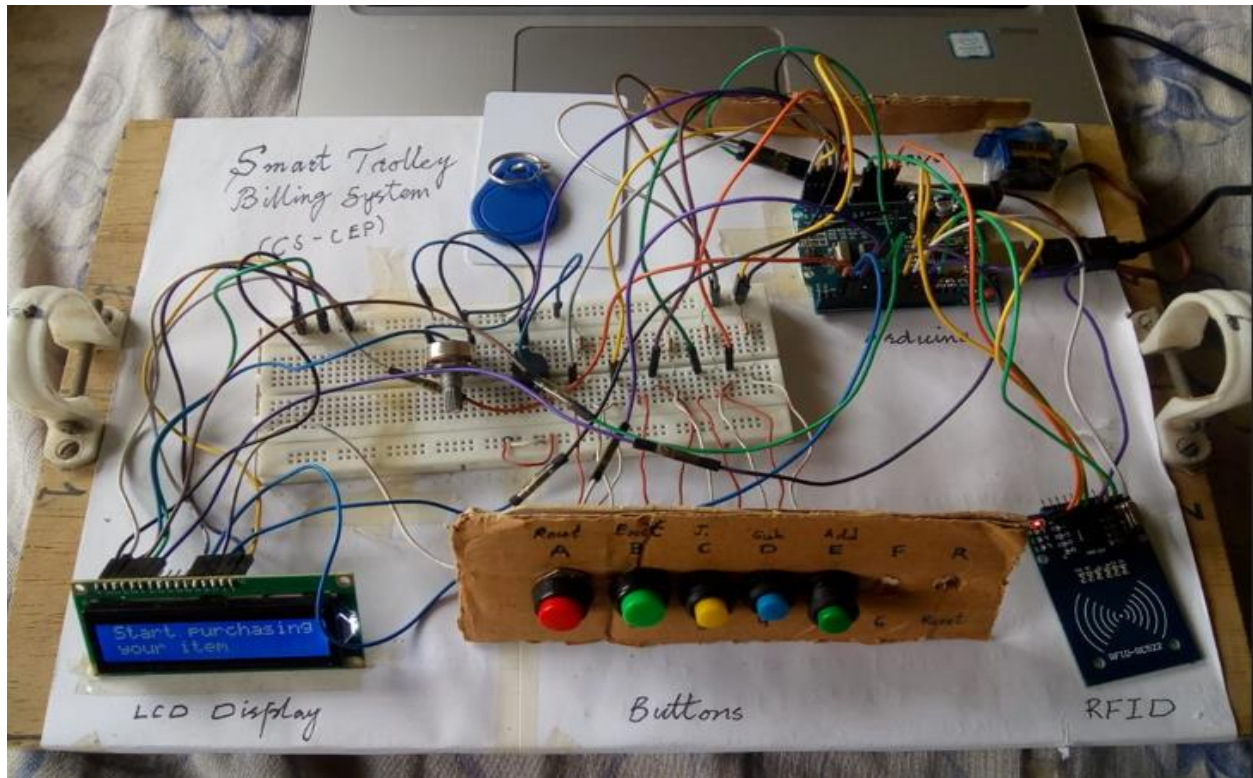
## **Components Used:**

1. **Arduino UNO:** The main controller for the project.
2. **RFID Reader (MFRC522):** To scan item tags.
3. **LCD Display (16x2):** To show item details and total bill.
4. **Buttons:** For adding, removing items, and completing the purchase.
5. **Buzzer:** For auditory feedback.
6. **Servo Motor:** To control the trolley gate.
7. **Power Supply:** For powering the components.



## Circuit Diagram

The circuit involves connecting the RFID reader, LCD display, buttons, buzzer, and servo motor to the Arduino UNO. The connections are detailed below:



### ❖ RFID Reader (MFRC522)

- SDA: Pin 10
- SCK: Pin 13
- MOSI: Pin 11
- MISO: Pin 12
- RST: Pin 9
- 3.3V: 3.3V
- GND: GND

### ❖ Buzzer

- Positive: Pin A0
- GND: GND

### ❖ Buttons

- Remove Button: Pin A1
- Add Button: Pin A2
- Reset Button: Pin A3
- Another Counter A4
- Exit Button: Pin A5
- VCC: 5V
- GND: GND

### LCD Display (16x2)

- RS: Pin 6
- E: Pin 7
- D4: Pin 2
- D5: Pin 3
- D6: Pin 4
- D7: Pin 5
- VSS: GND
- VDD: 5V
- V0: Potentiometer (contrast control)
- RW: GND
- A (LED+): 5V (through a resistor)
- K (LED-): GND

## Code for the Application

```
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#include <Servo.h>
// Define LCD pin connections
const int rs = 6, en = 7, d4 = 2, d5 = 3, d6 = 4, d7 = 5;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
const int remove_button = A1;
const int add_button = A2;
const int done_button = A4;
const int special_button = A3; // New button for special function
const int reset_button = A5; // New button for reset function
const int buzzer_Pin = A0;
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
Servo myServo;
struct item {
    String item_name;
    String item_number;
    int item_price;
};
const int number_of_item = 2;
const item item_list[number_of_item] = {
    //Item Name      Item RFID Number  Item Price
    {"10Kg Flour bag", "2A 57 6A 97", 1500},
    {"5Kg Sugar pack", "73 93 AC 1B", 750},
};
const item special_item_list[number_of_item] = {
    // Special Item Name  Item RFID Number  Item Price
    {"J. Kurta Shalwar", "2A 57 6A 97", 4999},
    {"J. Perfume", "73 93 AC 1B", 1999},
};
int bill_amount = 0;
int remove_buttonState = 0;
int add_buttonState = 0;
int done_buttonState = 0;
int special_buttonState = 0; // State for special button
int reset_buttonState = 0; // State for reset button
int add_item_flag = 1;
int remove_item_flag = 0;
bool purchasing_done = false;
bool special_mode = false; // Flag for special mode
int successful_purchases = 0; // Counter for successful purchas
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;

void setup() {
    pinMode(remove_button, INPUT_PULLUP);
    pinMode(add_button, INPUT_PULLUP);
    pinMode(done_button, INPUT_PULLUP);
    pinMode(special_button, INPUT_PULLUP); // Set special button as input
    pinMode(reset_button, INPUT_PULLUP); // Set reset button as input
    pinMode(buzzer_Pin, OUTPUT);
    myServo.attach(8); // Attach the servo to pin 8
    Serial.begin(9600); // Initiate a serial communication
    SPI.begin(); // Initiate SPI bus
    mfrc522.PCD_Init(); // Initiate MFRC522
    Serial.println("Approximate your card to the reader...");
    Serial.println();
    digitalWrite(buzzer_Pin, LOW);
    lcd.begin(16, 2);
    lcd.clear();
    // Set cursor (Column, Row)
    lcd.setCursor(0, 0);
    lcd.print("Smart Trolley");
    lcd.setCursor(0,1);
    lcd.print("Billing System");
    delay(2000);
    lcd.clear();
    // Set cursor (Column, Row)
    lcd.setCursor(0, 0);
    lcd.print("Counter 1 For");
    lcd.setCursor(0,1);
    lcd.print("Grocery Billing");
    delay(2000);
    // Display initial count of successful purchases
    lcd.clear();
    // Set cursor (Column, Row)
    lcd.setCursor(0, 0);
    lcd.print(String(successful_purchases) + " successfully");
    lcd.setCursor(0,1);
    lcd.print("purchased");
    delay(2000);
    // Prompt to start purchasing
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Start purchasing");
    lcd.setCursor(0, 1);
    lcd.print("your item");
    delay(2000);
    // Initialize PLX-DAQ headers
    Serial.println("CLEARDATA");
    Serial.println("LABEL,Time,Item Name,Item Price,Total Bill");
}
```



```

void loop() {
  int remove_buttonReading = digitalRead(remove_button);
  int add_buttonReading = digitalRead(add_button);
  int done_buttonReading = digitalRead(done_button);
  int special_buttonReading = digitalRead(special_button);
  int reset_buttonReading = digitalRead(reset_button);
  // Debounce the buttons
  if ((millis() - lastDebounceTime) > debounceDelay) {
    if (remove_buttonReading != remove_buttonState) {
      remove_buttonState = remove_buttonReading;
      if (remove_buttonState == LOW) {
        add_item_flag = 0;
        remove_item_flag = 1;
        lcd.clear();
        // Set cursor (Column, Row)
        lcd.setCursor(0, 0);
        lcd.print("You Can now");
        lcd.setCursor(0,1);
        lcd.print("Remove your item");
        delay(2000);
      }
    }
    if (add_buttonReading != add_buttonState) {
      add_buttonState = add_buttonReading;
      if (add_buttonState == LOW) {
        add_item_flag = 1;
        remove_item_flag = 0;
        lcd.clear();
        // Set cursor (Column, Row)
        lcd.setCursor(0, 0);
        lcd.print("You Can now");
        lcd.setCursor(0,1);
        lcd.print("add your item");
        delay(2000);
      }
    }
    if (done_buttonReading != done_buttonState) {
      done_buttonState = done_buttonReading;
      if (done_buttonState == LOW) {
        purchasing_done = true;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Pls Clear the Bill");
        lcd.setCursor(0, 1);
        lcd.print("Before Exit");
        delay(2000);
      }
    }
    if (special_buttonReading != special_buttonState) {
      special_buttonState = special_buttonReading;
      if (special_buttonState == LOW) {
        special_mode = true;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Counter for");
        lcd.setCursor(0, 1);
        lcd.print("J. Brand");
        delay(2000);

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("25% sale on");
        lcd.setCursor(0, 1);
        lcd.print("each item");
        delay(2000);
      }
    }
    if (reset_buttonReading != reset_buttonState) {
      reset_buttonState = reset_buttonReading;
      if (reset_buttonState == LOW) {
        bill_amount = 0;
        purchasing_done = false;
        special_mode = false;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Trolley Reset");
        lcd.setCursor(0,1);
        lcd.print("Start again");
        delay(2000);
      }
    }
    lastDebounceTime = millis();
  }
  // Look for new cards
  if (!mfrc522.PICC_IsNewCardPresent()) {
    return;
  }
  // Select one of the cards
  if (!mfrc522.PICC_ReadCardSerial()) {
    return;
  }

  // Show UID on serial monitor
  Serial.print("UID tag :");
  String content = "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  Serial.println();
  Serial.print("Message : ");
  content.toUpperCase();
  if (purchasing_done) {
    if (content.substring(1) == "2A 57 6A 97") {
      myServo.write(180); // Rotate the servo motor
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Bill Paid, Thnx");
      lcd.setCursor(0, 1);
      lcd.print("For Shopping");
      delay(3000);
    }
  }
}

```

```

myServo.write(90); // Reset the servo position
successful_purchases++;
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(String(successful_purchases) + " successfully");
lcd.setCursor(0, 1);
lcd.print("done");
delay(3000);
// Reset everything
bill_amount = 0;
purchasing_done = false;
special_mode = false; // Reset special mode
lcd.clear();
// Set cursor (Column, Row)
lcd.setCursor(0, 0);
lcd.print("Trolley Reset");
lcd.setCursor(0,1);
lcd.print("Start again");
delay(2000);
} else {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Unauthorized");
  lcd.setCursor(0, 1);
  lcd.print("card");
  delay(2000);
}
return;
}
// Special mode check
const item* active_item_list = special_mode ? special_item_list : item_list;
for (int i = 0; i < number_of_item; i++) {
  if (content.substring(1) == active_item_list[i].item_number) {
    if (add_item_flag == 1) {
      int price_to_add = active_item_list[i].item_price; // Store price in a
      if (special mode) {
        price_to_add *= 0.75; // Apply 25% discount
      }
      bill_amount += price_to_add;
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(active_item_list[i].item_name);
      lcd.setCursor(0, 1);
      lcd.print("Added: Rs." + String(price_to_add));
      digitalWrite(buzzer_Pin, HIGH);
      delay(1000);
      digitalWrite(buzzer_Pin, LOW);
    } else if (remove_item_flag == 1) {
      bill_amount -= active_item_list[i].item_price;
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(active_item_list[i].item_name);
      lcd.setCursor(0, 1);
      lcd.print("Removed: Rs." + String(active_item_list[i].item_price));
      digitalWrite(buzzer_Pin, HIGH);
      delay(1000);
      digitalWrite(buzzer_Pin, LOW);
    }
    // Send data to PLX-DAQ
    Serial.print("DATA,TIME,");
    Serial.print(active_item_list[i].item_name);
    Serial.print(",");
    Serial.print(active_item_list[i].item_price);
    Serial.print(",");
    Serial.println(bill_amount);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Total Bill:");
    lcd.setCursor(0, 1);
    lcd.print("Rs." + String(bill_amount));
    delay(2000);
  }
}
}
}

```

### **How my project links with Communication systems (TC-306)**

This project can be closely linked with the concepts covered in the Communication Systems (TC-306) course, highlighting the practical applications of the theoretical knowledge gained. Below are the ways in which this project aligns with various topics of the course:

#### ❖ **Introduction to Communication:**

- **Elements of Communication System:** The project integrates various components to form a communication system, including the RFID reader (transmitter), Arduino (processor), and LCD display (receiver).
- **Electromagnetic Spectrum:** RFID technology operates at specific frequencies within the electromagnetic spectrum, typically in the UHF band (860-960 MHz). Understanding the allocation and usage of these frequencies is fundamental to communication systems.

#### ❖ **Analog Communication:**

- Although this project predominantly uses digital communication principles, the concepts of modulation and demodulation are still applicable. The RFID reader and tags communicate using modulation techniques to transfer data.
- **Amplitude Modulation (AM), Frequency Modulation (FM):** Understanding how data can be modulated and transmitted over different mediums helps in appreciating the RFID technology, which relies on the modulation of radio waves.

#### ❖ **Digital Communication Systems:**

- **Digital Transmitters and Receivers:** The project uses an RFID reader (digital transmitter) and tags (digital receivers) to communicate data.
- **Pulse Modulation Techniques:** RFID technology can involve pulse modulation for encoding and transmitting information between the reader and the tags.
- **Bit Error Rate (BER):** Ensuring accurate data transmission and reception between the RFID reader and tags involves minimizing bit errors, an essential concept in digital communication.

#### ❖ **Digital Modulation Schemes:**

- **Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK):** RFID systems often use ASK or FSK for communication. Understanding these modulation techniques is directly relevant to how RFID technology works.

## **Future Work**

1. **Inventory Management:** Integrate with a store's inventory management system to update stock in real-time.
2. **Mobile App Integration:** Develop a mobile app for customers to view their shopping history and bills.
3. **Enhanced Security:** Implement more robust security measures for RFID tag identification.
4. **Scalability:** Extend the system to support larger retail environments with multiple trolleys and check-out points.

**Conclusion:** This project demonstrates the integration of RFID technology with an Arduino-based system to create a Smart Trolley Billing System. It automates the billing process in retail stores, making shopping more convenient for customers. The system's functionality, including item addition, removal, and purchase completion, is effectively managed by the Arduino UNO, RFID reader, LCD display, buttons, buzzer, and servo motor. This project can be further enhanced with additional features such as real-time inventory updates and integration with store databases for a more comprehensive retail solution.