ISSN : 0973 - 8355

INTERNATIONAL JOURNAL OF MATHEMATICAL, MODELLING, SIMULATIONS AND APPLICATIONS

IJMMSA

www.ijmmsa.com

E-MAIL
editor.ijmmsa@gmail.com
editor@ijmmsa.com

INTERNATIONAL JOURNAL OF MATHEMATICAL, MODELLING, SIMULATIONS AND APPLICATIONS IJMMSA

ISSN; 0973-8355 www.ijmmsa.com

Vol 17 Issue 1 Jan 2024

Smart Shopping Cart with Automatic Billing System Using RFID and Arduino

Prof. Amol B. Auti[1], Ms. Parimala N. Mittha[2], Mr. Akash J. Nagansur[3], Ms. Durgalaxmi R. Padal[4], Ms. Pragati V. Rathod[5], Mr. Aasam N. Shaikh[6], Ms. Sakshi M. Doijode[7]

Assistant Professor, Department of Electrical Engineering, Nagesh Karajagi Orchid College of Engineering & Technology, Solapur[1]

B.Tech Electrical, Department of Electrical Engineering, Nagesh Karajagi Orchid College of Engineering & Technology, Solapur[2,3,4,5,6,7]

***Abstract*—**Rising footfall at shopping malls has made conventional checkout counters a major bottleneck, with customers spending a disproportionate amount of time waiting in billing queues. This paper presents a Smart Shopping Cart with an Automatic Billing System built around Radio Frequency Identification (RFID) and an Arduino Nano microcontroller. Every product on the shop floor carries a 125 kHz RFID tag, and each shopping trolley is fitted with an EM-18 RFID reader. As a shopper places an item in the cart, the reader captures the tag's unique code, the corresponding price is added to a running total, and the item name and cumulative bill are shown on a 16×2 I2C LCD in real time. Re-scanning an item removes it from the bill, allowing customers to change their mind without visiting a counter. A push button finalizes the bill at the end of shopping, and payment is intended to be settled directly at the trolley using a pre-recharged customer card, removing the need for Wi-Fi, ZigBee, or GSM/ESP connectivity that many existing systems depend on. The prototype was built and tested on a breadboard with three sample products, and it correctly added, removed, and totaled item prices on the LCD in real time. The proposed design offers a low-cost, connectivity-independent alternative to barcode- and app-based smart-cart systems and is well suited for deployment in supermarkets and shopping malls.

Keywords*—*RFID, Arduino Nano, Automatic Billing, EM-18 Reader, Smart Shopping Cart, LCD Display, IoT Retail

I. INTRODUCTION

A supermarket or shopping mall is a place where customers select and purchase daily-use products and then pay for them at a billing counter. Modern retail has raised the standard of living, and with it, the size of the crowds visiting shopping malls has grown substantially. Handling this crowd requires shortening the time consumed by the billing process, since long checkout queues remain one of the most common sources of customer dissatisfaction even when product selection itself is quick and easy.

This paper proposes a smart shopping cart that keeps track of purchased products for billing using RFID and Arduino, so that the billing process is completed inside the trolley itself rather than at a separate counter, even for a single product. As a customer places or removes an item, the cart scans its RFID tag and updates the bill immediately; once shopping is complete, the finalized amount is displayed in the trolley itself.

In the proposed system, every product in the store carries a passive RFID tag, and every cart carries an RFID reader. When a tagged product is scanned, its price is automatically added to (or, on a second scan, subtracted from) the bill, and the item name together with the running price is shown on an onboard LCD. The system is built entirely around the EM-18 RFID reader module and an Arduino Nano microcontroller, using no Wi-Fi, ZigBee, Bluetooth, or ESP module, which keeps the design simple, low-cost, and independent of network availability.

A. Project Objective

In the present-day shopping system, one of the major difficulties is having to follow a queue through the billing process, which is time consuming. This project aims to reduce the average time spent by a customer at a shopping mall by implementing an automatic billing system at the trolley itself using RFID technology.

II. LITERATURE REVIEW

A number of prior smart-cart implementations have explored different combinations of identification and communication technologies for automated billing. Table I summarizes representative approaches and their reported limitations.

TABLE I. SUMMARY OF EXISTING SMART-CART APPROACHES

Author / Year	Approach	Limitation
Sainath (2014)	Barcode-based billing, line-of-sight scanning	Requires fixed boundary and central billing system

Budic (2014)	RFID with online payment, web application	No handling of accidentally dropped products
Dhavale Shraddha (2016)	RFID/IoT billing using ESP module	Distance and interference issues
Komal Ambekar (2015)	RFID/ZigBee-based trolley	Payment restricted to a specific counter
Hsin-Han Chiang (2016)	Face recognition for authentication	Poor accuracy in crowded malls
Narayana Swamy (2016)	RFID with dedicated website	Needs stable internet, prone to server errors
Vinutha (2014)	RFID-based billing	Requires server maintenance and connectivity
Prateek Aryan (2014)	Bluetooth billing to mobile	Limited by smartphone availability
Suganya (2016)	Arduino/Android based system	Needs continuous network and smartphone
Vanitha Sheeba & Brindha Rajkumari (2015)	RFID/ZigBee billing	Manual collection at counter causes queues

Across these works, four recurring drawbacks stand out: (i) dependence on Wi-Fi, ZigBee, Bluetooth, or an ESP module for connectivity, which introduces range and interference problems; (ii) reliance on a remote web server for billing maintenance and payment processing, which fails when the network is unstable; (iii) reduced accuracy in crowded environments for vision-based methods such as face recognition; and (iv) dependence on the customer's own smartphone, which is not guaranteed to be available or charged.

The system proposed in this paper addresses these drawbacks by performing billing and payment entirely within the trolley, using only an RFID reader, an Arduino Nano, and a local LCD — eliminating the connectivity and server dependencies of the approaches summarized above.

III. RFID TECHNOLOGY

Radio Frequency Identification (RFID) is a technology-based identification system that identifies objects through tags attached to them, without requiring line-of-sight contact between the tag and the reader — only radio communication between the two is needed. RFID improves on the barcode system that preceded it (first introduced commercially in 1967) in several respects: it needs no line of sight for reading, multiple items can be read in a single scan, each tag can carry read/write data rather than a fixed category code, individual items rather than just product categories can be identified, and passive tags have a virtually unlimited operating lifetime.

A basic RFID system has three components: an RFID tag (a silicon microchip attached to a small antenna, encapsulated in plastic or glass with an adhesive backing); a reader, which contains a scanner and antenna that communicates with the tag and extracts its data; and a processor or controller — typically a microcontroller — that receives the reader's output and processes it. Tags themselves may be passive (powered by the reader's radio energy, cheapest and used here), active (battery-powered, periodically broadcasting an ID), or battery-assisted passive. RFID systems typically operate at one of four frequency bands: low frequency (120–140 kHz, used for asset tracking), high frequency (13.56 MHz, used for contactless cards), ultra-high frequency (869–928 MHz, used in supply-chain tracking), or microwave (2.4–2.5 GHz, used in electronic toll collection). This project uses 125 kHz passive tags read by an EM-18 low-frequency reader.

IV. PROPOSED SYSTEM AND METHODOLOGY

An RFID tag operating at 125 kHz is attached to every product in the mall, and an EM-18 reader is mounted on the trolley. At the time of purchase, the tag on a product is scanned by the reader; each tag carries a unique Electronic Product Code (EPC). Based on the EPC received, the Arduino identifies the product, adds its price to the running total, and displays the product name and updated bill on the LCD. If the customer decides to remove a product that has already been scanned, scanning the same tag a second time subtracts its cost from the bill rather than adding it again.

A push button mounted on the trolley signals the end of shopping. When it is pressed, the system totals every scanned item and displays the final bill on the LCD, after which payment can be made through a pre-recharged RFID customer card issued by the shopping mall, without the shopper needing to visit a checkout counter. The complete transaction information can additionally be logged to a central PC over the serial interface for the store's own records.

The core control logic follows a simple parity rule: each time a tag is read, a counter for that tag is incremented. If the tag has been read an odd number of times, the item is treated as newly added and its price is added to the bill; if it has been read an even number of times, the item is treated as removed and its price is subtracted. This avoids the need for a separate 'add' and 'remove' mode and keeps the user interaction to a single, uniform scanning action.

The overall algorithm can be summarized as: (1) the cart is reset at the start of a shopping session; (2) each RFID tag read is compared against the previous read-count for that tag — an odd count adds the item, an even count subtracts it; (3) the corresponding item name and updated bill are shown on the LCD after every scan; and (4) pressing the push button ends the session and displays the final bill.

V. SYSTEM ARCHITECTURE

The system architecture consists of four functional blocks connected to the Arduino Nano at its center: the RFID reader (which receives data from the RFID tags on the products), the

ISSN; 0973-8355 www.ijmmsa.com
Vol 17 Issue 1 Jan 2024

16×2 I2C LCD (which displays item and billing information), the push button (used to signal the end of a shopping session), and the power supply. The Arduino Nano forms the processing core, reading the serial data stream from the EM-18 module, applying the odd/even scan-count logic described in Section IV, and driving the LCD accordingly.

Data flows through the system in one direction at a time: the EM-18 reader continuously listens for a 125 kHz tag within its read range and, upon detecting one, transmits the tag's unique 12-character identifier serially to the Arduino. The Arduino buffers these bytes, matches the completed identifier against the set of known product codes, and updates two internal variables — a per-product scan counter and a running total — before pushing the updated state to the LCD over the I2C bus. Because each block communicates over a simple, well-defined interface (UART for the reader, I2C for the display, a digital input for the button), the design remains easy to extend with additional tags, additional products, or a serial link to a central billing PC without restructuring the rest of the circuit.

This modular separation between sensing (RFID reader), processing (Arduino), and output (LCD, buzzer, indicator LEDs) also simplifies fault isolation during testing: each block can be verified independently — for instance, by reading raw tag IDs from the EM-18 on the serial monitor before integrating the LCD logic — which shortened development and debugging time for the prototype.

VI. HARDWARE DESCRIPTION

A. Arduino Nano

The Arduino Nano is a compact microcontroller board built around the 8-bit ATmega328, with an integrated crystal oscillator, USB interface, and voltage regulator. It provides 14 digital I/O pins (6 of which support PWM output) and 8 analog input pins in a breadboard-friendly form factor, and it is programmed through the Arduino IDE over a Mini-B USB connection without requiring any external programmer hardware. Table II lists its key specifications.

TABLE II. KEY HARDWARE SPECIFICATIONS

Parameter	Specification
Microcontroller	Atmel ATmega328
Operating Voltage	5 V
Input Voltage (recommended)	7–12 V
Digital I/O Pins	14 (6 PWM)
Analog Input Pins	8
Flash Memory	32 KB
Clock Speed	16 MHz
RFID Reader	EM-18, 125 kHz, UART / Wiegand26
Display	16×2 LCD with I2C (PCF8574) interface

B. EM-18 RFID Reader

The EM-18 is a low-cost, low-power RFID reader module that reads 125 kHz passive tags. It supports both UART and Wiegand26 output formats, allowing it to be interfaced directly with a microcontroller through UART, or with a PC through an RS232 converter. In this project, its UART output is fed directly to the Arduino Nano's serial input.

C. 16×2 LCD with I2C Interface

A 16×2 character LCD with an I2C back-pack (built around a PCF8574 chip) is used to display item names and the running bill. The I2C interface reduces the wiring to just two data lines (SDA and SCL) instead of the seven to ten connections a standard parallel LCD would require, and the module is addressed at the standard 0x27 (or 0x3F) I2C address.

D. Push Button

A simple push-button switch is used to signal the end of the shopping session. When pressed, it drives the corresponding Arduino digital pin (configured with an internal pull-up) low, which the firmware interprets as a request to finalize and display the bill.

E. Circuit Connections

The EM-18 reader's VCC and GND pins are connected to the Arduino's 5V and GND rails, and its serial data-out pin is wired to the Arduino's RX line so that scanned tag IDs are received over UART. The I2C LCD module's SDA and SCL lines are connected to the Arduino's A4 and A5 pins respectively, and it likewise draws power from the 5V and GND rails. The push button is wired between a digital input pin (configured as INPUT_PULLUP) and ground, so that a press pulls the pin low. An indicator LED and a small buzzer are connected to two further digital output pins and are pulsed briefly on every successful scan to give the shopper audible and visual confirmation that an item has been registered. All grounds — Arduino, reader, display, and indicators — are tied to a common ground to avoid signal-reference errors.

VII. SOFTWARE IMPLEMENTATION

The firmware is written in the Arduino IDE (C/C++) using the Wire and LiquidCrystal_I2C libraries for LCD control. On startup, the LCD displays a welcome banner before clearing to an idle state. In the main loop, incoming serial bytes from the EM-18 reader are buffered into a fixed-length character array; once a complete 12-character tag ID has been received, it is compared against the known product codes using strncmp(). A match triggers the corresponding price update, an LCD message showing the product name and unit price, and a short buzzer/LED indication, after which the LCD switches to show the updated running total. An unrecognized tag ID causes the LCD to display a 'Not in cart' message rather than affecting the bill.

A. Firmware Pseudocode

ISSN; 0973-8355 www.ijmmsa.com

Vol 17 Issue 1 Jan 2024

The core logic executed inside the Arduino's loop() function can be expressed as follows:

```
1. Wait until a complete tag ID is received
over serial.
2. Compare the ID against the known product-
code list.
3. If matched: increment that product's scan
counter.
4. If counter is odd -> add price to total,
show "<item> Added".
   If counter is even -> subtract price from
total, show "<item> Removed".
5. Update the LCD with the item name and
running total.
6. If unmatched: display "Not in cart".
7. If the push button is pressed: display the
final total and end the session.
```

VIII. RESULTS AND DISCUSSION

The proposed smart shopping trolley was assembled on a breadboard using an Arduino Nano, an EM-18 RFID reader, a 16×2 I2C LCD, and a push button, and was tested with three sample RFID-tagged products (butter, tea, and milk). On scanning the 'Butter' tag, the LCD displayed "Butter Added" with its unit price, and the running total updated correctly; scanning 'Tea' and 'Milk' in sequence likewise updated the item name and increased the total to the correct cumulative value in each case. Re-scanning a tag was observed to correctly subtract its price from the running total, confirming the odd/even scan-count logic. The system consistently displayed the correct item name and updated bill on the LCD in real time for every scan performed during testing, demonstrating that the core add/remove/total billing workflow functions as intended for a small product set.

A. Test Observations

Table III summarizes the sequence of test scans performed on the prototype and the corresponding running total shown on the LCD after each scan, confirming that additions accumulate correctly and that the displayed total matched the expected sum in every trial.

TABLE III. SAMPLE SCAN SEQUENCE AND RESULTING BILL

Action	Item Scanned	LCD Display	Running Total (Rs.)
1	Butter (scan 1)	Butter Added, Price 10.00	10.00
2	Tea (scan 1)	Tea Added, Price 20.00	30.00
3	Milk (scan 1)	Milk Added, Price 30.00	60.00
4	Tea (scan 2 – removed)	Tea removed, price subtracted	40.00
5	Push button pressed	Final bill displayed	40.00

B. Discussion

The results confirm that a low-frequency passive-RFID and Arduino combination is sufficient to reproduce the core functionality of a smart-cart billing system — item detection, real-time price accumulation, item removal, and final-bill display — without any network dependency. Response time between a tag scan and the LCD update was consistently under one second in testing, which is adequate for a walking-pace shopping interaction. The main limitation observed during testing was the short read range of the EM-18 module (a few centimetres), which requires the shopper to bring each product close to the reader; this is a reasonable trade-off given the corresponding reduction in cost and the elimination of cross-talk between adjacent trolleys that a longer-range reader could introduce in a crowded aisle.

Compared with the barcode-based and network-dependent approaches summarized in Table I, the prototype avoids the line-of-sight requirement of barcodes and the connectivity dependence of Wi-Fi/ZigBee/ESP-based designs, at the cost of requiring every product to be fitted with an RFID tag rather than a printed barcode — a one-time infrastructure cost for the retailer that is offset by faster, more reliable scanning over the system's lifetime.

IX. ADVANTAGES

- Efficient shopping experience: helps customers track items and cost without waiting at a separate counter.
- Automatic billing: the running bill is maintained by the Arduino as items are added or removed from the cart.
- Queue reduction: billing performed inside the trolley reduces congestion at traditional checkout counters, particularly during peak hours.
- Connectivity-independent design: no Wi-Fi, ZigBee, Bluetooth, or ESP module is required, avoiding the range and reliability issues these introduce.
- Scope for real-time inventory management and anti-theft mechanisms when integrated with a store's central system.

X. FUTURE SCOPE

RFID technology has been in development for over sixty years and continues to see wider adoption across industries, but two challenges still limit its more widespread use in retail: the relatively high cost of tags and readers, and privacy concerns around item-level tracking. As RFID tag prices continue to fall and tag durability improves, this system can be extended with features such as an automated exit gate that cross-verifies purchased items using a master card, integration with a central inventory-management server for real-time stock updates, support for mobile and contactless payment options at the trolley, and additional sensors (such as the IR-based drop-detection and door-locking mechanisms

ISSN; 0973-8355 www.ijmmsa.com
Vol 17 Issue 1 Jan 2024

discussed in the underlying project) to further improve security against accidental or intentional product removal.

XI. CONCLUSION

This paper has presented the design and implementation of a Smart Shopping Cart with an Automatic Billing System using RFID technology, an EM-18 reader, and an Arduino Nano. The system's primary aim is to reduce the time customers spend in long billing queues while simultaneously easing inventory management for the retailer, making it well suited to shopping malls that experience large crowds and heavy rush. By performing scanning, billing, and (in the full design) payment entirely within the trolley — without depending on Wi-Fi, ZigBee, or a customer's smartphone — the proposed approach offers a simpler and more robust alternative to several existing smart-cart solutions. Overall, this automatic billing system is a meaningful step toward replacing the conventional barcode-based checkout process and making the shopping experience faster and more convenient.

The prototype demonstrated in this work validates the core billing logic on a small scale; scaling it to a full retail deployment would mainly involve tagging the complete product catalogue, adding the IR-based drop-detection and card-authentication features outlined in Section X, and integrating the trolley's serial output with the store's central inventory and point-of-sale systems. Given its low component cost and freedom from network dependency, the design is well suited to gradual, low-risk adoption by small and mid-sized retailers before scaling to larger supermarket chains.

ACKNOWLEDGMENT

The authors would like to thank to the Hon. Principal Dr. B. K. Sonage for his continuous support throughout this work, and to Prof. V. B. Bhosale, Head of Department of Electrical Engineering, NKOCET, Solapur, for providing the facilities used to build and test the prototype.

REFERENCES

[1] Chandrashekhar P. and T. Sangeetha, "Smart shopping cart with automatic central billing system through RFID and Zigbee," IEEE, 2014.

[2] M. Hubert, C. Blut, C. Brock, and T. Eberhardt, "Acceptance of smart phone based mobile shopping: mobile benefits, customer characteristics, perceived risks and the impact of application context," IEEE, 2018.

[3] A. Sutagundar, M. Ettinamani, and A. Attar, "IoT based smart shopping mall," Conference Paper.

[4] A. Kumar, S. Srivastava, and U. Gupta, "Internet of Things (IoT) based smart shopping center using RFID," Conference Paper.

[5] R. Li, T. Song, N. Capurso, J. Yu, J. Couture, and X. Cheng, "IoT applications on secure smart shopping system," Conference Paper.