LNMIIT
The LNM Institute of Information Technology

Intelligent Vending Machine

Submitted by:

Parth Garg(22ucs147)
Sanyam Munot(22ucs185)

Submitted to:

Dr. Rajbir Kaur

Introduction

In the era of smart technology, integrating IoT (Internet of Things) solutions into everyday devices enhances their functionality and efficiency. This project focuses on designing a smart vending machine using Arduino Mega, equipped with Zigbee modules, RFID readers, servo motors, push buttons, weight sensors, LEDs, and a buzzer. The machine offers an automated, user-friendly experience for customers, allowing seamless item dispensing and real-time data updates to the owner and customer carts via Zigbee communication.

Components Required

1. Arduino Mega
2. Arduino Uno
3. Zigbee Modules S2C (2)
4. RFID Reader MFRC522
5. Servo Motors specofoc (2)
6. Push Buttons (3)
7. 5kg Weight cell
8. LEDs (Green, Red, White, Blue)
9. Buzzer
10. Breadboard and Jumper Wires
11. Power Supply
12. 16x2 Alphanumeric Display JHD162A
13. 45-ohm potentiometer
14. AK XBee USB Adapter FT232RL
15. Nano USB cable(for 14th component)

Procedure

Initial Setup:

- Install Arduino IDE on your computer.
- Connect the Arduino Mega to your computer via USB.

Connecting Components:

- **Servo Motors:** Connect three servo motors to the Arduino Mega, each controlled by a separate digital pin(5,6,7)
  Red Wire - 5v output
  Brown wire- Ground
  Yellow Wire- Digital pins
- **Push Buttons:** Connect three push buttons for dispensing items and one for ending the session.
  Item Dispense Button - 22,23,24 (Mega)
  Session End Button - 25(Mega)
  NOTE: Connect one end of the push button to the pin mentioned and the other end to the Ground.
- **RFID Reader:** Connect the RFID reader to the Arduino Mega, ensuring proper communication via serial pins.
  SDA - Pin 13 (Mega)
  SCK - Pin 52 (Mega)
  MOSI - Pin 51 (Mega)
  MISO - Pin 50 (Mega)
  RST - Pin 8 (Mega)
- **LEDs and Buzzer:** Connect LEDs to indicate different states and a buzzer for alerts. (Mega)
  Blue- 30
  Green -12
  Yellow- 4
  Red- 31
  Buzzer -3
  NOTE: Connect the Anode of these LEDs to the Pins mentioned and connect the Cathode to the Ground
- **ZigBee Module (Router):**
  Zigbee Pin1 – 3.3 V (Mega)
  Zigbee Pin2 (TX) - pin 19 (RX) (Mega)
  Zigbee Pin3 (RX) - Pin 18 (TX) (Mega)
  Zigbee Pin10 - Ground (Mega)
- **ZigBee Module (Coordinator):**
  Zigbee Pin1 – 3.3 V (UNO)
  Zigbee Pin2 (TX) - pin 2 (RX) (UNO)
  Zigbee Pin3 (RX) - Pin 3 (TX) (UNO)
  Zigbee Pin10 - Ground (UNO)
- **Display :** (Mega)
  VO - Middle Pin of Potentiometer

RS - Pin 9
E - Pin 10
D4 - Pin 11
D5 - Pin 2
D6 - Pin A0
D7 - Pin A1

- **Potentiometer:** Connect 1st and 3rd pin to 5v and to ground. (Mega)

Coding:

- **Arduino Code:** Write and upload code to the Arduino Mega to control the servos, read RFID data, manage the session timer, handle button presses, and communicate with the weight sensor.
- **Zigbee Configuration:**
  - Configure the Zigbee modules for coordinator and router roles using XBee Explorer and install XCTU Software
  - Using XCTU Software, configure both ZIgbee modules one by one and test if they are communicating properly.
  - Ensure proper serial communication between the Arduino Mega and Zigbee modules.
- **Node Js Script for Web App:**
  - Write a Node Js script to receive data from Zigbee and update a web app, displaying the customer and owner carts.

Testing and Simulation:

- Test each component individually in Tinkercad or a similar simulation tool.
- Verify the integration of all components by simulating the entire vending machine operation.

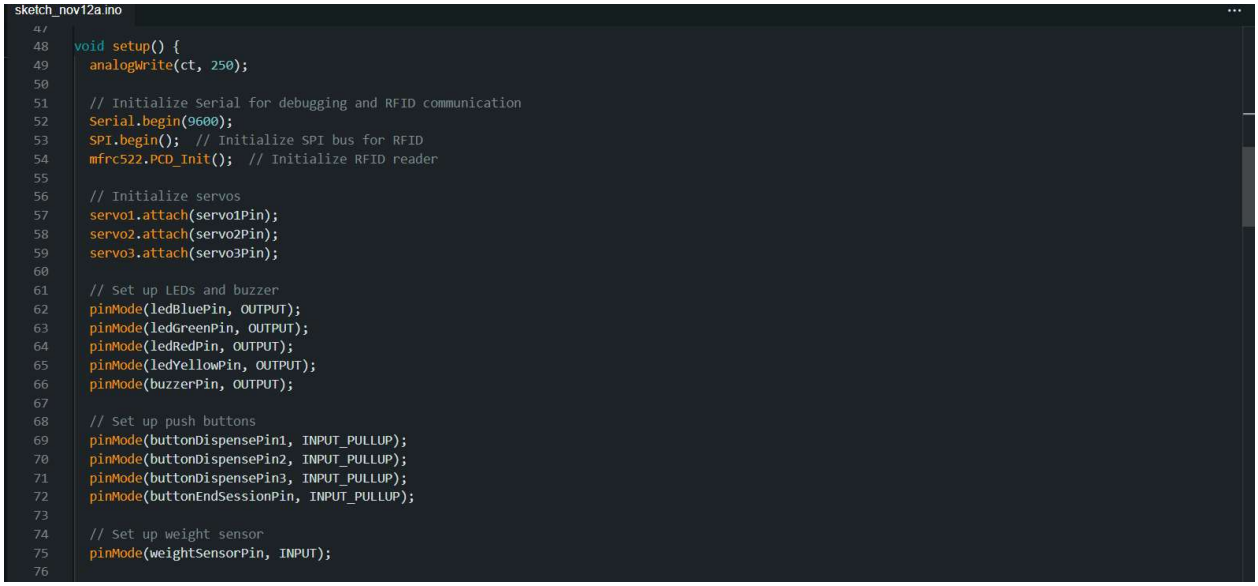
Integration:

- Combine all modules and ensure seamless communication between the Arduino Mega and the Arduino Uno via Zigbee.
- Deploy the web app to visualize real-time data updates.

Problems Faced

- Configuration of the Zigbee module was a very hectic task as these modules are very delicate and if one pin gets badly connected u need to update its firmware again and again . Making Zigbee communication was a difficult task.
- We need to define custom serial pins for data transmission for Zigbee and we used Software Serial library for configuring that.
- Zigbee sends data character by character, So it becomes difficult to club all characters and concatenate them in a string according to time stamps.
- Weight Sensor needs a lot of calibration and hardware for proper functioning and we were not able to implement it in our project.
- The contrast pin VO of the display was initially difficult to understand; afterwards we connected it to the potentiometer so that we could adjust its contrast by rotating the nob.

Circuit & Code Snippets



sketch_nov12a.ino

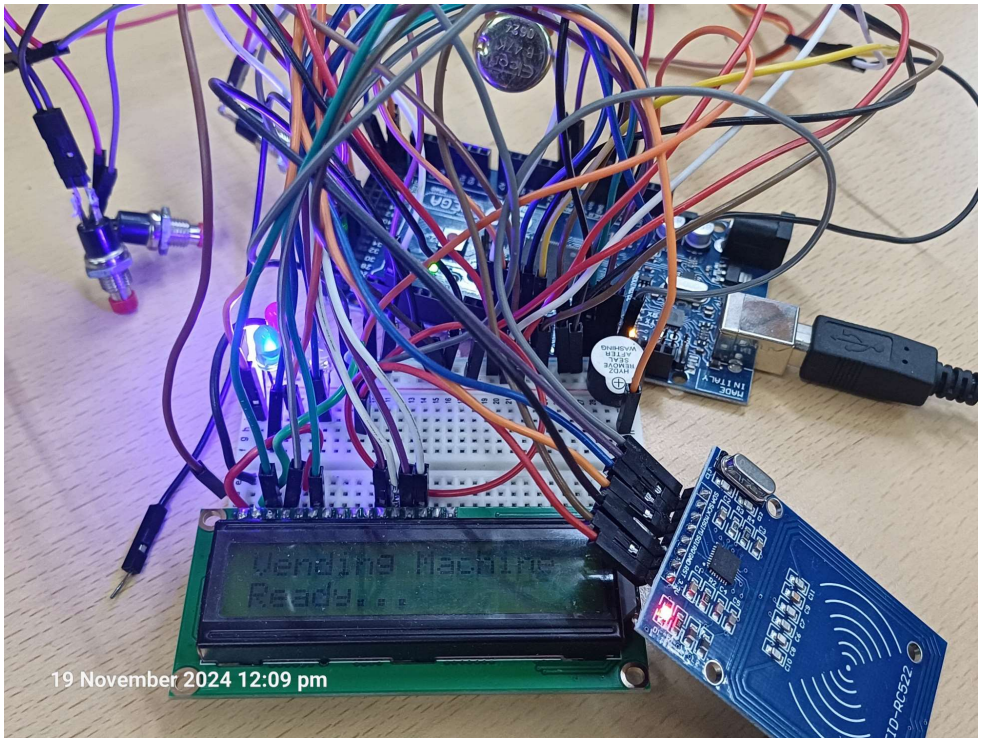
```
  pinMode(weightSensorPin, INPUT);

  // Initialize LCD
  lcd.begin(16, 2);
  displayIdleState();

  // Set initial LED states
  digitalWrite(ledBluePin, HIGH); // Idle indicator
  digitalWrite(ledGreenPin, LOW);
  digitalWrite(ledRedPin, LOW);
  digitalWrite(ledYellowPin, LOW);
}

void loop() {
  // Check RFID tag
  if (!sessionActive) {
    if (checkRFID()) {
      startSession();
    } else {
      denyAccess();
    }
  }

  // Dispense item if corresponding button is pressed
  if (sessionActive && digitalRead(buttonDispensePin1) == LOW) {
    dispenseItem(servo1);
  }
  if (sessionActive && digitalRead(buttonDispensePin2) == LOW) {
    dispenseItem(servo2);
  }
  if (sessionActive && digitalRead(buttonDispensePin3) == LOW) {
```

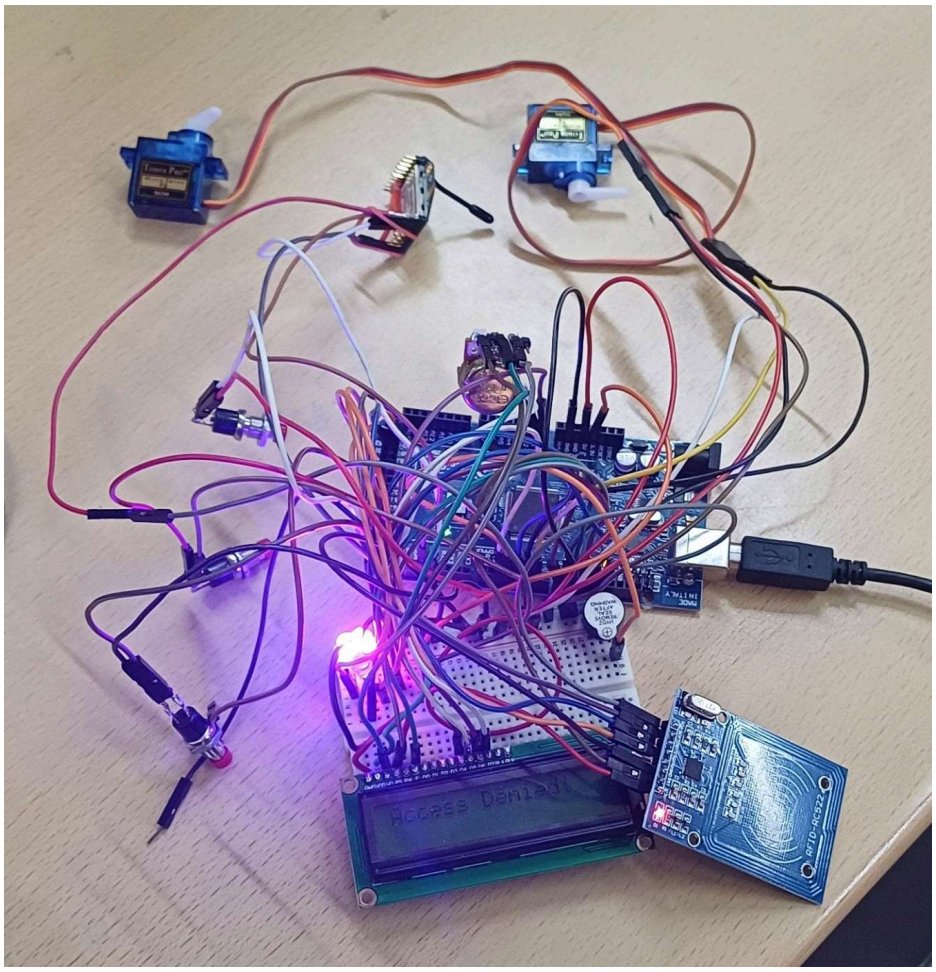
sketch_nov12a.ino

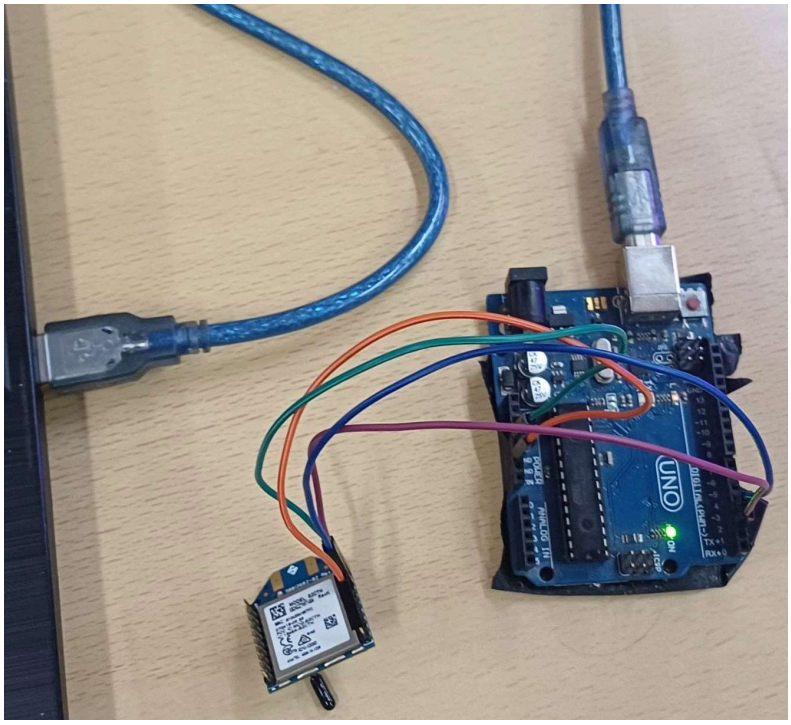
```
  }
  if (sessionActive && digitalRead(buttonDispensePin3) == LOW) {
    dispenseItem(servo3);
  }

  // End session if button pressed
  if (sessionActive && digitalRead(buttonEndSessionPin) == LOW) {
    endSession();
  }

  // Check if item is picked up
  if (sessionActive && checkItemPickup()) {
    itemPickedUp();
  }

  // Check if session has expired
  if (sessionActive && millis() - sessionStartTime > 120000) {
    endSession();
  }
}

// Functions

void displayIdleState() {
  lcd.clear();
  lcd.print("Vending Machine");
  lcd.setCursor(0, 1);
  lcd.print("Ready...");
}
```

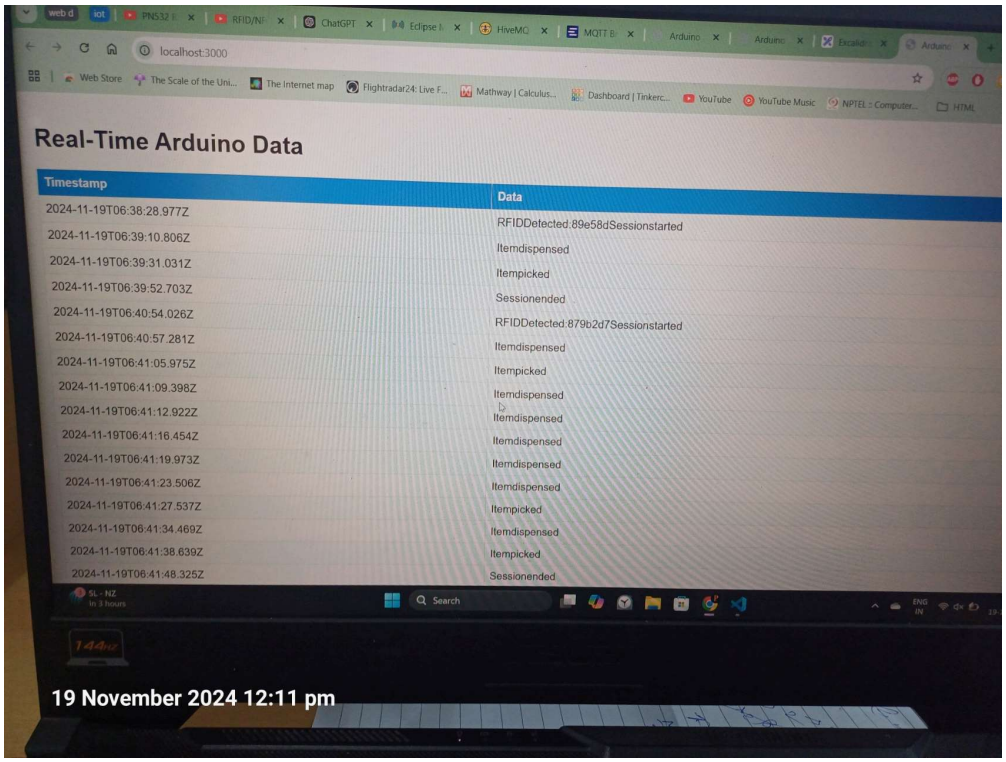
```
163
164   // Activate item waiting indicator
165   digitalWrite(ledYellowPin, HIGH);
166   pickupWaitStartTime = millis();
167
168   lcd.clear();
169   lcd.print("Item Dispensed!");
170   lcd.setCursor(0, 1);
171   lcd.print("Pick it up.");
172
173   Serial.println("Item dispensed");  // Send data to Raspberry Pi
174 }
175
176 void itemPickedUp() {
177   digitalWrite(ledYellowPin, LOW);
178
179   lcd.clear();
180   lcd.print("Item Picked!");
181   delay(1000); // Display briefly
182   lcd.clear();
183   lcd.print("Session Active");
184
185   Serial.println("Item picked");  // Send data to Raspberry Pi
186 }
187
188 bool checkRFID() {
189   if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
190     String rfid = "";
191     for (byte i = 0; i < mfrc522.uid.size; i++) {
192       rfid += String(mfrc522.uid.uidByte[i], HEX);
193     }
```

```
191     for (byte i = 0; i < mfrc522.uid.size; i++) {
192       rfid += String(mfrc522.uid.uidByte[i], HEX);
193     }
194     Serial.println("RFID Detected: " + rfid);
195     return true;  // Assuming the RFID is valid
196   }
197   return false;
198 }
199
200 bool checkItemPickup() {
201   weightValue = analogRead(weightSensorPin);
202   if (weightValue > 100) {  // Adjust the threshold value based on sensor calibration
203     return true;
204   }
205   return false;
206 }
207
208 void endSession() {
209   sessionActive = false;
210   digitalWrite(ledGreenPin, LOW);
211   digitalWrite(ledBluePin, HIGH);
212
213   lcd.clear();
214   lcd.print("Session Ended");
215   delay(2000);
216   displayIdleState();
217
218   // Signal end of session with buzzer
219   tone(buzzerPin, 1000, 500); // Beep
220   delay(500);
221   tone(buzzerPin, 1000, 500); // Beep again
```


Vending Machine
Ready...
19 November 2024 12:09 pm


Access Denied




localhost:3000
Real-Time Arduino Data
Timestamp
Data
2024-11-19T06:38:28.977Z
RFIDDetected:89e58dSessionstarted
2024-11-19T06:39:10.806Z
Itemdispensed
2024-11-19T06:39:31.031Z
Itempicked
2024-11-19T06:39:52.703Z
Sessionended
2024-11-19T06:40:54.026Z
RFIDDetected:879b2d7Sessionstarted
2024-11-19T06:40:57.281Z
Itemdispensed
2024-11-19T06:41:05.975Z
Itempicked
2024-11-19T06:41:09.398Z
Itemdispensed
2024-11-19T06:41:12.922Z
Itemdispensed
2024-11-19T06:41:16.454Z
Itemdispensed
2024-11-19T06:41:19.973Z
Itemdispensed
2024-11-19T06:41:23.506Z
Itemdispensed
2024-11-19T06:41:27.537Z
Itempicked
2024-11-19T06:41:34.469Z
Itemdispensed
2024-11-19T06:41:38.639Z
Itempicked
2024-11-19T06:41:48.325Z
Sessionended
19 November 2024 12:11 pm

Project Summary

This smart vending machine project demonstrates the effective integration of IoT technology to enhance traditional vending machines' functionality. By using Arduino Mega, Zigbee modules, and various sensors and actuators, the machine can autonomously dispense items, validate RFID cards, manage session timings, and update data in real-time. The project showcases the potential of IoT in improving user experience and operational efficiency in everyday applications.

Links:

Github: https://github.com/parthgarg351/Smart-Vending-Machine

Special Thanks

We would like to express our deepest gratitude to our course coordinator, Dr. Rajbir Kaur, for her invaluable guidance, motivation, and support throughout the development of this project. Her expertise and encouragement have been instrumental in helping us overcome challenges and achieve our goals. We would also like to extend our heartfelt thanks to Mr. Shivam Maheshwari for providing us with the necessary components required to bring this project to life.