Vending Machine Firmware – Project Overview

Introduction

This report details the development, implementation, and testing of a software-based vending machine for sodas, using the ESP32-WROOM development board and programmed in C with the Arduino IDE. The machine operates under FreeRTOS to manage real-time tasks and handle concurrent operations. The software also produces 1 of the 4 random faults as the user uses this machine.

Project Requirements and Implementation

Overview of Requirements

- The vending machine accepts only nickels, dimes, and quarters, with a specific starting change count.
- Dispenses four types of sodas with different costs and initial quantities.
- Implements tasks for handling money input, soda dispensing, error handling, and display updates.
- Generates and handles random errors affecting operations.
- Uses task prioritization and to manage concurrency and prevent race conditions.

Assumptions and Decisions

1. **Transaction Handling:** A Task was created to handle coin insertion, soda selection, and soda dispensation. This Task was the last one to be created, and it has the lowest priority. It was assumed that this task could be interrupted by any of the other 2 created tasks.
    - This task allows and verifies user input for coin insertion. This Task only allows characters 'Q', 'D', and 'N', for Quarters, Dimes, and Nickels respectively (including the 'Newline' character that indicates the ends of coin insertion.
    - This task allows and verifies user input for soda selection. This Task only allows characters '1', '2', '3', '4', and 'C', for Cola, Grape, Orange, Lime, and 'Cancel' respectively (including the 'Newline' character that indicates the ends of coin insertion.
    - This Task checks if Soda should be dispensed and if change is required after dispensation.

- A cancel function (C) allows users to abort the transaction at any point before the soda is dispensed.

2. **Soda Inventory, Pricing, and Available Coins:** Another Task was created to handle the displaying of 2 menus. One menu that shows the quantity, the price, and the ID of each Soda, and another menu that shows how many of each acceptable coin is available in the machine. (That menu is very helpful for testing and debugging the machine.)
    - The menu can be easily removed by commenting out one line of code.
    - The quantity, the price, and the ID of each soda can be easily changed by changing their initial variable in the source code (Global Variable).
    - The quantity of each acceptable coin can also be changed easily from the global variables

3. **Task Management:**
    - Input and error handling tasks run concurrently, using both cores of the ESP32.
    - Display updates and error handling are triggered by relevant events (soda dispensed, error occurred).

4. **Error Generation and Handling:** One other Task was created to handle the generation of the 4 random errors. This Task was the first one to be created, and it has the highest priority. It was assumed that this task could interrupt any of the other 2 created tasks to display the errors. The other 2 tasks would not resume until the error is resolved or the reset button is pressed on the ESP32.
    - Random errors are generated with a 4% probability for each second of the program running.
    - vTaskSuspend and vTaskDelay are used to control the other tasks and to control the access of the Global Variables

5. **LED Light:** The blue LED light is used to display the state of the program during the initial set up.
    - Steady and constant light means that everything in the setup function was able to run successfully.
    - 3 blinks of the blue light mean that the Error generator task was not created properly.
    - 4 blinks of the blue light mean that the Display Menu task was not created properly.
    - 3 blinks of the blue light mean that the User Input task was not created properly.

Software Verification and Validation

Verification:

- The source code was analyzed thoroughly to ensure that it was syntax-free and free of logical mistakes.
- The coding style was checked to ensure that the source code adheres to best practices and is well-commented.
- The source code was compiled and ensured that it ran on the ESP32-WROOM development board

Validation:

- Unit testing was conducted for individual tasks and functions. Each Task and function were separately tested to make sure that they did what was expected of them. (
- Integration testing ensured that all tasks worked together and as expected.

Test Plan: Each task can be tested based on their requirements and the Coin available menu can be used to improve testing. For example, the following can be tested:

- Check to see if any other inputs besides 'Q', 'D', and 'N' are accepted (Case-insensitive).
- Check to see if a purchase can be canceled after inserting the coins.
- Check to see if the machine returns the expected change for each purchase.
- Check to see if the machine cancels the purchase if it cannot return the expected coins.
- Check to see if the machine dispenses any soda if the money inserted is not enough.
- Check to see if the machine allows you to purchase a soda that's not available.
- Check to see if the calculation that the machine makes during a transaction is accurate.
- Check to see if the random generated errors stop the machine from making transaction and check is some of them can be resolved by inputting a code

User Manual

**** Note: Upon Initial boot-up, the **reset** button might need to be pressed in order for the Welcome Message to display properly. ****

1. **Starting Up:**
   - The machine displays a welcome message with instructions upon boot-up.

2. **Making a Purchase:**
    - To enter coins, type the respective letter according to the following list:
        1. ‘Q’ for $0.25 (Quarter)
        2. ‘D’ for $0.10 (Dimes)
        3. ‘N’ for $0.05 (Nickels)

    - Then press 'Enter' to finalize the input.
    - Note:
        1. Stack inputs are accepted
        2. Inputs are Case Incentives

        (Ex: Type 'QdN' then press 'Enter' to insert $0.40)

    - After inserting the coin(s), type the ID of the Soda to dispense then press 'Enter'. Or type 'C' to cancel the purchase then press Enter. Valid IDs are 1, 2, 3, 4 for Cola, Grape, Orange, and Lime, respectively.

    - If sufficient funds have been inserted, the soda will be dispensed.

    - Change is dispensed after each purchase, if necessary.

3. **Handling Errors:**
    - There are 4 random errors hardcoded into the program.
    - Each one of them has about a 1% chance of happening every second. Meaning that there is a 4% chance of getting an error every second.
    - If an error occurs, follow on-screen instructions to resolve the issue. or reset the machine.
    - Error with ID ‘1’ is the only error that can be resolved without resetting the machine.
        1. To resolve it, the Activation Code of the machine needs to be entered.
        2. The activation code is case-sensitive and it is ‘IziUnlock’.
        3. After entering that code and pressing ‘Enter’, the machine should be able to resume working properly.
    - The other 3 errors with ID ‘2’, ‘3’, and ‘4’ can only be resolved by resetting the machine. The error descriptions are as follows:
        1. Error with ID ‘2’ is about the Coin Counting part of the machine being jammed.
        2. Error with ID ‘3’ is about the Machine experiencing some power Fluctuation that stops it from receiving enough power to work properly.
        3. Error with ID ‘4’ is about the Cable of the Machine’s Soda Dispenser section, being broken.