COMP2121 Project

Description:

In this project you will develop a vending machine simulator to be run on the AVR development board. The system will need to maintain system and inventory state, take input from the user, and provide audio-visual feedback.

Your system should satisfy the following requirements. Marks will be allocated amongst the points listed in the following section. More marks will be given to more difficult features. Some bonus marks will be awarded for extensions beyond the spec.

Core requirements: (80%)

Start Screen

On device startup, the LCD screen should display the following text:

2121 17s1 X99
Vending Machine

Where 'X99' is replaced with your group's name. Correctly displaying your group name is required before any other parts can be marked.

The device should stay on this screen for 3 seconds, or until any keypad button is pressed. It should then go to the 'select' screen.

Select Screen

This is the main menu screen of the vending machine.

The LCD should display the following:

Select item

Pressing any of the keypad number buttons 1-9 should attempt to retrieve the corresponding item. If any items are in the inventory, the program should go to the 'coin' screen. Otherwise it should go to the 'empty' screen.

Using any other input devices should have no effect on this screen.

Empty Screen

This screen should display the following message:

Out of stock
?

Where '?' should be replaced by the number of the item that was selected on the previous screen.

The program should stay in this state for 3 seconds before returning to the 'select' screen.

All 10 LEDs should be on for the first half of each of the three seconds, and off for the second half.

Pressing either of the push buttons should cause the program to behave as if the 3 seconds had already elapsed.

Coin Screen

This screen prompts the user to enter coins, and displays the number of coins still required.

Insert coins
?

Where '?' is replaced with the number of coins remaining.

The LEDs should count the number of coins that have been entered so far, using one light for each coin starting with the bottom LED.

Coins will be entered using the potentiometer as described in the 'Potentiometer Input' section.

If the '#' key is pressed then the current selection should be aborted and the program should return to the 'select' screen.

Potentiometer Input

While on the 'coin' screen, the potentiometer should be monitored for input. To complete the insertion of a coin, the potentiometer must start at 0 (maximum anticlockwise) then be moved to 0x3FF (maximum clockwise) then be returned back to 0.

If the number of coins remaining after this is 0 then the program should go to the 'deliver' screen and the current inventory count should be decremented. Otherwise the display should be updated to reflect the number of coins remaining and the program should wait for more potentiometer input.

Deliver Screen

Having completed payment, the machine should deliver the item to the user by spinning the motor at full speed for 3 seconds and flash the LEDs in the same way as the 'empty' screen. Afterwards the program should go back to the 'select' screen.

This state cannot be aborted – all input should be ignored. The following should be displayed on the screen:

Delivering item

Coin Return

If the coin screen is aborted after coins have been entered, the program should return each coin by spinning the motor at full speed for 0.25 seconds followed by a 0.25 second pause before returning any other coins. For full marks this must happen concurrently with displaying the select screen again and selecting a new item.

Admin Mode

The user may enter admin mode by holding down the '*' key for 5 seconds while on the 'select' screen. The admin mode screen will allow the user to select and modify the inventory counts and costs of each item.

By default item 1 will be selected, the selection can be changed by pressing the numbers 1-9 on the keypad.

This display should show:

Admin mode ?
$

Where '?' is the number of the currently selected item, '#' is the inventory count of the current item, and '$' is the cost of the current item preceded by a '$' character (e.g "$3").

The LEDs should show the inventory count of the current selected item, using one light for each item starting with the bottom LED.

If the right button is pressed the count for the selected item should be increased, up to a maximum of 10. If the left button is pressed the count should be decreased, down to a minimum of zero.

If the 'A' button is pressed then the cost for the current item should be increased, up to a maximum of 3 coins. Pressing 'B' will decrease the cost down to a minimum of 1 coin.

If the 'C' button is pressed then the count for the selected item should be set to zero immediately.

The user can exit this mode by pushing the '#' key, which will go back to the 'select' screen.

Starting Inventory

When the program is first started, each item should have an inventory count equal to its identifying number. The cost for each item will be 1 coin for odd numbered items and 2 coins for even numbered items.

Advanced Features (20%)

Sounds

The system should use the provided mini speaker to add sounds to the game.

The speaker should be connected between the pin labelled PB3 and the pin labelled GND, in the top-right corner of the board. Despite the labelling, this pin is actually controlled by changing the value of PORTB pin 0, or PB0. The sound may be generated by toggling the pin to produce a square wave. The frequency of the square wave should be as soothing as possible.

The speaker should be used to beep for 250ms every time a key is pressed. It should also beep for 1 second when an 'empty' item is selected, an item is delivered, or admin mode is entered.

LCD Backlight

When on any screen except the admin screen, the LCD backlight should turn off if no keys have been pressed for 5 seconds. It should turn on if any keys are pressed.

When turning on or off, the LCD backlight should fade smoothly over 500ms.

When on the admin screen the LCD backlight should stay on.

Increased Capacity

For this part, implement support for item inventory counts in the range 0 to 255 inclusive. The LCD display should be modified to be able to represent this information, but all display features above must still work as specified when item counts are less than or equal to 10. The LEDs may use any form of display that is backwards compatible, such as saturating at 10 items.

Submission Information

You will need to submit the following items:

1. A soft copy of your complete source (all .asm files). Your program should be well commented.
2. A hard copy of your user manual. The user manual describes how a user interacts you're your device, including how to wire up the AVR lab board. Make sure you indicate which buttons perform each action and how the LED and LCD displays should be interpreted.
3. A hard copy of the design manual. The design manual describes how you designed the vending machine system. It must contain the following components:
    a. System flow control. This component describes the flow control of the system at the module level using a diagram.
    b. Data structures. This component describes the main data structures used in the system.
    c. Algorithm descriptions. This component describes the main algorithms used in the system.
    d. Module specification. This component describes the functions, the inputs and the outputs of each module.

Overall, a person with knowledge about the subject and board should understand how your system is designed after reading this design manual.

Be sure to clearly specify your name, student ID, and lab group on all submitted documentation.

Demonstration

You will need to demonstrate your working project to an assessor on Thursday or Friday of week 13. Demonstration time slots will be determined closer to the due date. As with the labs all group members must be present to be marked, and marks will be deducted for any group members that cannot explain the design and function of the code.

You will need to submit the hard copies of the above documents during the demonstration. You will also need to bring a copy of your source code on a flash drive or similar.

Marking Scheme

This project is worth 100 marks. The marking scheme will be as follows:

- System Implementation (80 marks)
- Design Manual (10 marks)
- User Manual (5 marks)
- Coding style and commenting (5 marks)

Reading the POT hardware

For reference, the following C code will trigger an ADC read from the potentiometer connected to PK8. This can trivially be translated into AVR assembly.

```
ADMUX = (3 << REFS0) | (0 << ADLAR) | (0 << MUX0);
ADCSRB = (1 << MUX5);
ADCSRA = (1 << ADEN) | (1 << ADSC) | (1 << ADIE) | (5 << ADPS0);
```

When the ADC read is complete the ADC interrupt will be triggered, and the measured value can be read from the ADCH:ADCL I/O registers.