CSCE231: Systems Engineering
Final Project: A Classic Vending Machine

Due date: December 15th, 2016

1 Introduction

In this project, your team will build a prototype software system to operate a vending machine. Currently, the actual machine is being built by your company's machine shop. Your job is to use the DE1 to mimic its I/O components in the vending machine so you can develop software in parallel with the development of the vending machine.

Your company will be marketing two versions of the vending machine: the base system and the deluxe system. The basic specifications of these two versions are provided in the next section. At a minimum, you need to complete the base system to earn a decent grade for the project. You can earn more points if you can extend your base system to the deluxe system.

2 Specification of the Two Systems

2.1 Base system (65 pts)

The base system only supports one product that can be purchased through the machine. Your machine must support a function so that the unit price (`unit_price`) can be programmed. You will implement this function using a combination of the slider switches and a push button. As you set the slider switches, the actual value is displayed in decimal through the seven-segment displays. Pressing a push button sets `unit_price` to the value set by the slider switches. This function is performed by the merchant when the unit is put on site (initial state).

When products are delivered to the machine, the delivery person can set the number of units (`units_left`) that will be in the machine after he/she is done (preparation state). Once again, this is done using the slider switches, a push button, and the seven-segment displays to show the set value. Once this value is set, the machine is ready for vending. In the vending state, the machine displays the number of items in the machine as it waits for customers. Each time the machine vends a product, this value is decremented by one. Once this value reach zero, the vending machine stops taking money.

During deployment, you can mimic the three types of coins (nickel, dime, and quarter) using three push buttons. When a customer deposits a coin, the value of the coin is displayed on the seven-segment displays for a second or two. It then displays the total that has been deposited. Once the customer deposits enough coins so that the deposit amount is greater than or equal to the unit price, the customer can use a slider switch to vend the product and retrieve change. Again, vending a product means that `units_left` is decrement by one.

In the case that a customer begins depositing coins but does not have enough enough money to complete the purchase, the customer can use the same switch for vending as a "return coins" button. When the customer presses the "return coins" button, a red light comes on, the `deposit_amount` is reset to zero, and the machine go back to the waiting state. The machine also uses a timer as the back up mechanism to return coins. When the last coin is inserted but the total amount is not enough to vend the product, the machine sets a timer for 30 seconds. If no additional coin is deposited during this time, the machine automatically returns the coins already deposited.

Next, we describe resources needed to complete your tasks and their behaviors.

- *Initial State.* In this state, the machine is first delivered to the deployment site. A technician sets up the machine by plugging it to a backed up power source. **The initial display of $HEX2 - HEX0$ should show blanks.** The technician then sets up `unit_price`. To do so, the technician sets the slider switch $SW9$ to '1' and uses $SW7$ to $SW0$ to set the unit price. For example, if the unit price is 75 cents, the technician would set the 8 switches ($SW7$ - $SW0$) to 01001011, respectively. The technician then pushes $KEY3$ to set the value. The set value should display through $HEX2$ to $HEX0$ **for about a second**.

Note that this machine only supports the unit price from $0 to $2.55. Also note that the decimal place for each seven segment display is not enabled. As such, $2.55 would only show up as 255 on the seven-segment displays. If the technician makes a mistake, he/she simply needs to set the slider switches to the new price and then hits the push button again to set the unit price. Once `unit_price` is set, the technician resets $SW9$ to 0 to indicates that the initial setup is done. **You can indicate that the machine has been initialized but not loaded by displaying "- - -" through $HEX2 - HEX0$.** Throughout this state, the machine cannot not take any coin deposit.

- *Preparation State.* In this state, the machine has been set up but there are no products inside the machine yet. As such, before the completion of this state, the machine still cannot take any money. The delivery person is responsible for completing this state. First, the delivery person sets the slider switch $SW8$ to '1' and then sets the slider switches $SW7$ - $SW0$ to set `units_left` that will be placed in the machine and presses the push button $KEY3$ to store it. $HEX2$ to $HEX0$ **briefly display the value.** Note that this machine only supports up to 255 units. If a mistake is made, he/she can simply enter the new `units_left` through the slider switches and then once again press $KEY3$. Note that this particular state can only be completed after the *Initial State*. Once this state is completed, the system should display `units_left` through $HEX2$ to $HEX0$. Unset $SW8$ to get out of this mode. Throughout this state, the machine cannot not take any coin deposit.

- *Vending State.* Now the machine is ready to vend products. In this state, a customer deposits coins until `deposit_amount` ≥ `unit_price`. Each time a coin is deposited (pressing $KEY1$ - $KEY3$ mimics depositing a nickle, a dime, and a quarter, respectively), $HEX2$ - $HEX0$ display the value of the coin (i.e., 5, 10, or 25, respectively) for approximately one second and then display `deposit_amount`.

  If the customer sets $SW0$ by sliding it up, when `deposit_amount` < `unit_price`, the already deposited coins are returned (i.e., sets `deposit_amount` to 0 and illuminates $LEDR0$). $HEX2 - HEX0$ go back to display `units_left`; that is, the machine is back to the beginning of *Vending State*.

  If the customer sets $SW0$ after `deposit_amount` ≥ `unit_price`, the machine displays the change (`deposit_amount - unit_price`) if there is any and vends the product by setting `deposit_amount` to 0, decrementing `units_left` by one, and illuminating $LEDG7$. $HEX2$ - $HEX0$ go back to display `units_left`; that is, the machine is back to the beginning of *Vending State*.

  If the customer cannot complete the transaction within 30 seconds after the last coin has been deposited, the machine also returns all the coins that have been deposited (sets `deposit_amount` to 0 and illuminates $LEDR0$). This will require that you set a timer for 30 seconds each time a coin is deposit. For example, when a coin is deposited, the timer is set for 30 seconds. If another coin is deposited within 30 seconds after the first, the timer is reset to 30 seconds.

  If the machine just vends the last unit, the machine is now disabled. This means the machine goes back to the beginning of the *Preparation State*.

2.2 Deluxe system (20 pts)

The deluxe system extends the functionality of the base system to support vending of up to three products (three vending channels). This would require the following changes.

1. *Initial State* needs to be extended so that the technician can push $KEY1$ to set the price of `product1`, $KEY2$ to set the price of `product2`, and $KEY3$ to set the price of `product3`. Again, setting the price is done through $SW0$ - $SW7$ while $SW9$ is set. The machine only vends products that their unit prices have been set. This means if a product's price is set to zero, that vending channel is disabled. Unset $SW9$ to get out of this mode.

2. *Preparation State* also needs to be extended so that the delivery person can set the the number of units for each supported product. Again, use the three push buttons, set $SW8$, and input the number of units via $SW0$ - $SW7$ to set the corresponding products. If a product price is not set then the delivery person cannot place that product in the machine. For example, if the unit prices of `product1` and `product2` are set using $KEY1$ an $KEY2$, respectively but not `product3`, then the delivery person cannot set the number of units for `product3`. Note the delivery person can check if he/she sets the number of units correctly by leaving $SW8$ in the set position. $HEX2 - HEX0$ should now show "321" to indicate to the deliver person to push $KEY3$ for the number of `prod3`, $KEY2$ for the number of `prod2`, and $KEY1$ for the number of `prod1`. **If only two products are set as in the above example, then $HEX2 - HEX0$ should display "021" or "21".** Unset $SW8$ to get out of this mode.

3. For *Vending State*, the customer first presses the corresponding push button to first select the product. Next, the number of available units (`units_left[product]`) is displayed. It then follows the normal Vending State to show the inserted coin value and the deposit amount, compute changes, decrement `units_left[product]`, and then vend the product. The same steps as those used in the base machine are used to return the coins.

3 Submission (15 pts)

You need to submit the following via handin.

1. Create a directory using the team leader CSE login as the directory name. Inside this directory, create a "source" subdirectory.

2. Submit the C source files and header files. Put all the files in a directory. All files must provide comments so that students who are not in our section (e.g., folks in the normal 230 section) can read the comments and understand the functionality of your application. Put these files in the "source" subdirectory.

3. Create a project report that contains (i) a state transitional diagram of your application, (ii) five of the most difficult technical challenges you face during the project, (iii) a rating of the difficulty level of the project (1 = way too easy to 5 = way too hard), (iv) the number of hours your team spend on the project (e.g., three members working together for 3 hours = 3 hours and not 9 hours), and (v) five suggestions on how to improve the project.

4. Put the report in the root directory and then zip the entire directory and submit via handin.

We will do live grading **on December 15th, 2016 from 9am to 1pm. It will be in 20/21 Avery hall. As part of grading, prepare a 10 minutes presentation that answers the same 5 questions in your report. The grading will be first come, first serve. The grading process should take about 10 - 15 minutes per group.**

The grading script is provided in the next page. Please use it to check if your system behaves as expected. You can do this before demo. We will use the same script during grading.

Grading Script

Base System

- Initial State.

( ) 1. Observe $HEX2 - HEX0$ displayed value before the machine is initialized (should be blank).

( ) 2. Set $SW9$ to '1' to get the system into this state.

( ) 3. Set $SW0 - SW7$ to 200 (1100 1000).

( ) 4. Push $KEY3$ to set this value.

( ) 5. Observe if $HEX2 - HEX0$ display 200 briefly.

( ) 6. Set $SW0 - SW7$ to 85 (0101 0101).

( ) 7. Push $KEY3$ to set this value.

( ) 8. Observe if $HEX2 - HEX0$ display 85 briefly.

( ) 9. Set $SW9$ to '0' to get out of this state. You can indicate that the machine has been initialized but not loaded by displaying "- - -" through $HEX2 - HEX0$.

- Preparation State.

( ) 1. Try to deposit a quarter by pressing $KEY3$. The system should not take any coin.

( ) 2. Set $SW8$ to '1' to get the system into the preparation state.

( ) 3. Set $SW0 - SW7$ to 100 (0110 0100).

( ) 4. Push $KEY3$ to set this value.

( ) 5. Observe if $HEX2 - HEX0$ display 100.

( ) 6. Set $SW0 - SW7$ to 3 (0000 0011).

( ) 7. Push $KEY3$ to set this value.

( ) 8. Observe if $HEX2 - HEX0$ display 3.

( ) 9. Set $SW8$ to '0' and check if the machine is now in the vending state and displays 3 through $HEX2 - HEX0$.

- Vending State.

( ) 1. **Vend Product.** Press $KEY3$ three times and $KEY1$ twice and observe $HEX2 - HEX0$ for the corresponding displayed values (i.e., 25,25,25,50,25,75,5,80,5,85) as you deposit coins.

( ) 2. Vend the product by setting $SW0$ to '1'.

( ) 3. Observe that the change amount should be zero and $LEDG7$ is illuminated.

( ) 4. Observe that the number of products is now 2.

( ) 5. Reset $SW0$ back to '0' and observe that $LEDG7$ is no longer illuminated.

( ) 6. **Vend Product.** Press $KEY3$ four times and observe $HEX2 - HEX0$ for the corresponding displayed values (i.e., 25,25,25,50,25,75,25,100) as you deposit coins.

( ) 7. Vend the product by setting $SW0$ to '1'.

( ) 8. Observe that the change amount should be 15 and $LEDG7$ is illuminated.

( ) 9. Observe that the number of products is now 1.

( ) 10. Reset $SW0$ back to '0' and observe that $LEDG7$ is no longer illuminated.

( ) 11. **Return coins.** Press $KEY3$ twice and $KEY2$ twice and observe $HEX2 - HEX0$ for the corresponding displayed values (i.e., 25,25,25,50,10,60,10,70) as you deposit coins.

( ) 12. Return coins by setting $SW0$ to '1'.

( ) 13. Observe that the deposit amount should now be zero and $LEDR0$ is illuminated.

( ) 14. Reset $SW0$ back to '0' and observe that $LEDR0$ is no longer illuminated.

( ) 15. Observe that the number of products is still 1.

( ) 16. **Return coins.** Press $KEY3$ twice and wait about 10 seconds.

( ) 17. Press $KEY3$ one more time and wait 30 seconds for all the coins to be returned.

( ) 18. Observe that the deposit amount should now be zero and $LEDR0$ is illuminated briefly.

( ) 19. Observe that the number of products is still 1.

( ) 20. **Vend product.** Vend the last product.

( ) 21. Ensure that the machine is in the preparation state and can no longer take any coin.

- Reinitialization

( ) 1. Redo the initialization state and reset the price of each item to 50 cents.

( ) 2. Load the machine with 5 items.

( ) 3. Vend an item .

Deluxe System

- Initial State.

( ) 1. Set the system to vend two different products. The prices for the two products are 65 cents and 80 cents, respectively. Use $KEY1$ and $KEY2$ to set the prices.

- Preparation State.

( ) 1. Try to load the vending machine with the three products. One product should fail to load and the two that succeed should only have 2 items each.

( ) 2. Observe $HEX2 - HEX0$ should display "021", "-21", or "21".

- Vending State.

( ) 1. Vend an item of a product. Make sure you that the system needs to return some change. Observe that the system behaves correctly (e.g., the deposit amount, the change amount, the number of item left, etc.).

( ) 2. Vend the second item of the same product. Put in the exact amount (i.e., no change) this time. Observe that the system behaves correctly.

( ) 3. Try to vend the same product one more time. This time, you should not be able to. $HEX2 - HEX0$ should display 0 when you try to vend the product to indicate that there are zeor items of this product left in the machine. The machine should not take any coin.

( ) 4. Try to vend the third product. You should not be able to. Again, $HEX2 - HEX0$ should display zero when you push $KEY3$.

( ) 5. Try to vend an item from the remaining product but do not deposit the sufficient amount and wait 30 seconds for the coins to return.

( ) 6. Vend the remaining two items.

- Reinitializing.

( ) 1. Initialize the machine to vend 3 products. Set their prices to 25 cents, 40 cents, and 50 cents.

( ) 2. Load the machine with 1 item of each product.

( ) 3. Vend all three items. The machine should go back to the preparation state.