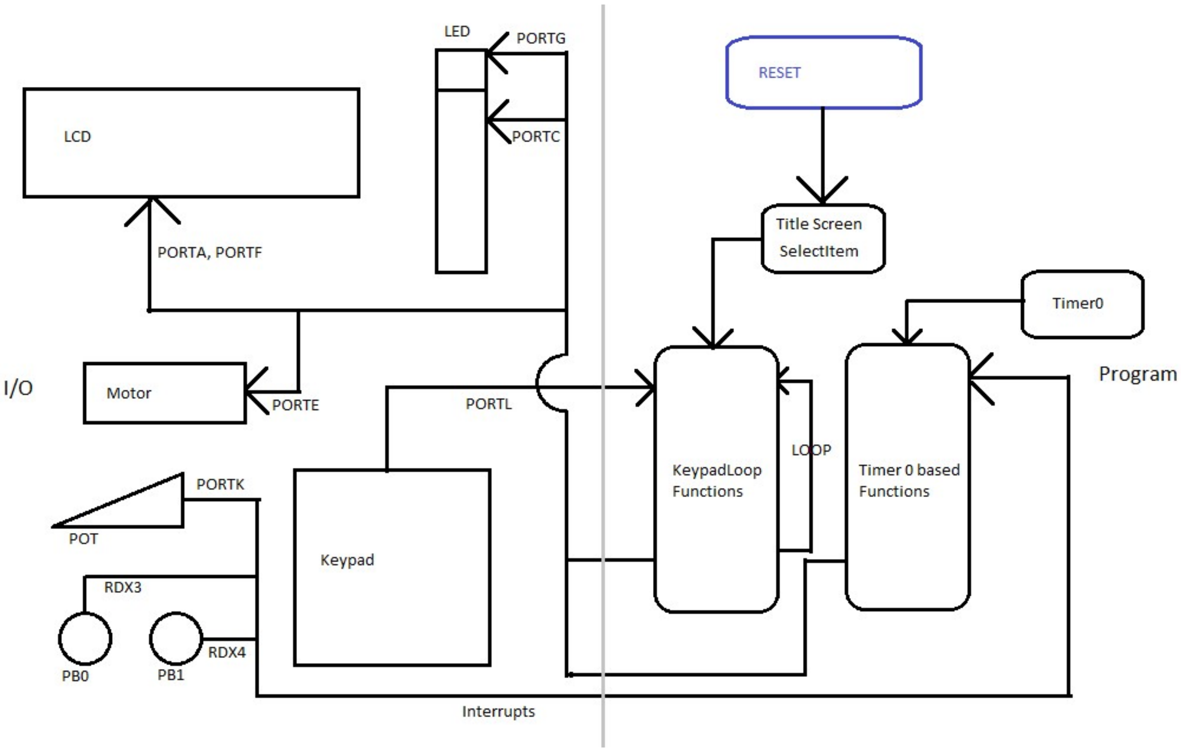
Vending Machine Simulator
Design Manuel

Group O3
Xiaowei Zhou z5108173
Chenkun Hu   z5128773

Table of Contents

1. System Flow Control



2. Data Structure

a) This program used 12 name-defined registers to do most functions and also 5 counters in dseg for temporary usage.

b) Besides, a 2-dimensional array stored the inventory quantity and price.

3. Algorithm

a) The main program is looping in keypad reading, with different pre-defined constants in register waitStatus and debounceStatus to jump to specific functions in main or Timer 0 OVF.

b) In every loop of keypad input and Timer 0 OVF, check the debounceStatus and waitStatus first by using switch() function to disable keypad input or jump to other functions. These status can also be used to identify two different functions of same input. (e.g. PB to break Out Of Stack but change the inventory in Admin Mode)

c) These status can also disable some functions. For example, when in Delivery Item screen, status register can ignore all input to avoid early quit of Delivery.

d) The most part of the code is checking status by using If function or Switch function.

4. Modules

e) Supporting macros are included in Macros.asm. LCD commands are put in the beginning of main.asm. Registers and constants (including status) are defined in defines.asm.

f) The RESET part of program will initialise stack pointers, ports and LCD. It will print the title screen.

g) After initialise timers, PB, quantity and prices, main function will move to loop of function KeypadInitialise, which take in inputs and called for other functions.

h) KeypadInitialise will take input when debounceStatus and waitStatus do NOT disable keypad input. Once get a input, KeypadInitialise function will try to check the current screen and call correct functions.

i) When on SelectItem screen or AdminMode, keypad input will be used to input the number of inventory. After number input, findItem function will take this number to find the price and current quantity of this item. If found in stock, findItem will jump to itemInStock, else, it will jump to itemOutOfStock.

j) When in admin mode, after item founded, it will call to show the quantity of current inventory on LCD and LED. This is seperated into several functions.

   i. If Quantity is 10, print '1' '0' on LCD, else, print 'x' ' '.

   ii. If Quantity is 9 or 10, enable the top 1st/2nd LED on PORTG and light all LED on PORTC. Else, show pattern on PORTC.

k) When on AdminMode screen, press of A,B,C is enabled to change the price and quantity of inventory. Press of A and B will change the price, and press of C will set quantity to zero.

l) When changing the price, we will first check if price reached maximum(3) or minimum (1). If reached, function will return immediately.

m) When press C, the quantity of current item will directly set to 0 and store in dseg.

n) When PB pressed, program will jump to PB_INT. As PB0 for increase and PB1 for decrease quantity, by InventoryNumberPlus/Minus function. It will first check the limit (0-10). If reached either limit, function will directly return.

o) Press of 0 and D will do nothing and return.

p) Press of '#' will start a check. If currently on insert coins screen, # will call coin return. If currently in admin mode, # will abort it and back to select item.

q) Press of '*' will start a check. If currently on title screen, * function normally as press other keys to skip title page and go to select item. If currently on select item screen, * will set a debounce status and start a time count in timer 0. If this count reached 5 sec, program will enter admin mode.

r) When findItem got the inventory and currently in normal mode, program will jump to coin insertion, and initialise POT input. Also, it will disable keypad input but '#' for coin return.

s) POT_INT are set with minimum 0 and maximum 3FF. If and only if POT start from minimum, reached maximum (will set a Status) and turn back to minimum (will detect a Status), a coin will be inserted.

t) After a coin inserted, program will check if coin remaining is 0. If not, coin insertion will be call again. If yes, program will jump to delivery, and all inputs will be disabled.

u) Delivery function will print the screen, start motor and enable LED flash. Since it flashes as same as OutOfStock, we simply use same function. After delivery, program will go back to select item and turn off leds and motor.

v) If '#' pressed when inserting coins, coin return will be called. This will disable all inputs and call motor for return. The return pattern is loaded and simply shift right to show current coins on led.

w) If the selected item is out of stock, program will jump to OutOfStock function and flash LED. At this time, keypad is disabled but PB_INT still work and can break the OutOfStock screen.