In the name of god

Mohammad.amin.allahrabi

97213002

Project: vending machine

Class: assembly

* **Over view:**

In this project we implement Vending machine with AVR and Proteus .

LCD show some Items with and it’s price to user and user see the price and select the Items by push KEY (on/c) and finally , accept and finish his buying and LCD show user the Total cost.

User can changes the list by two ways :

1-

Use bottoms ( + or -), if push these bottoms , LCD will show the next or previous Items.

2-

Use bottoms (1 , 2 , …. , 9) , if push each number , LCD will show the Items with number that ordered in list.

Two Seven segments:

When user select an Items , seven segment will increase number that shows the number of selected items.

Statements :

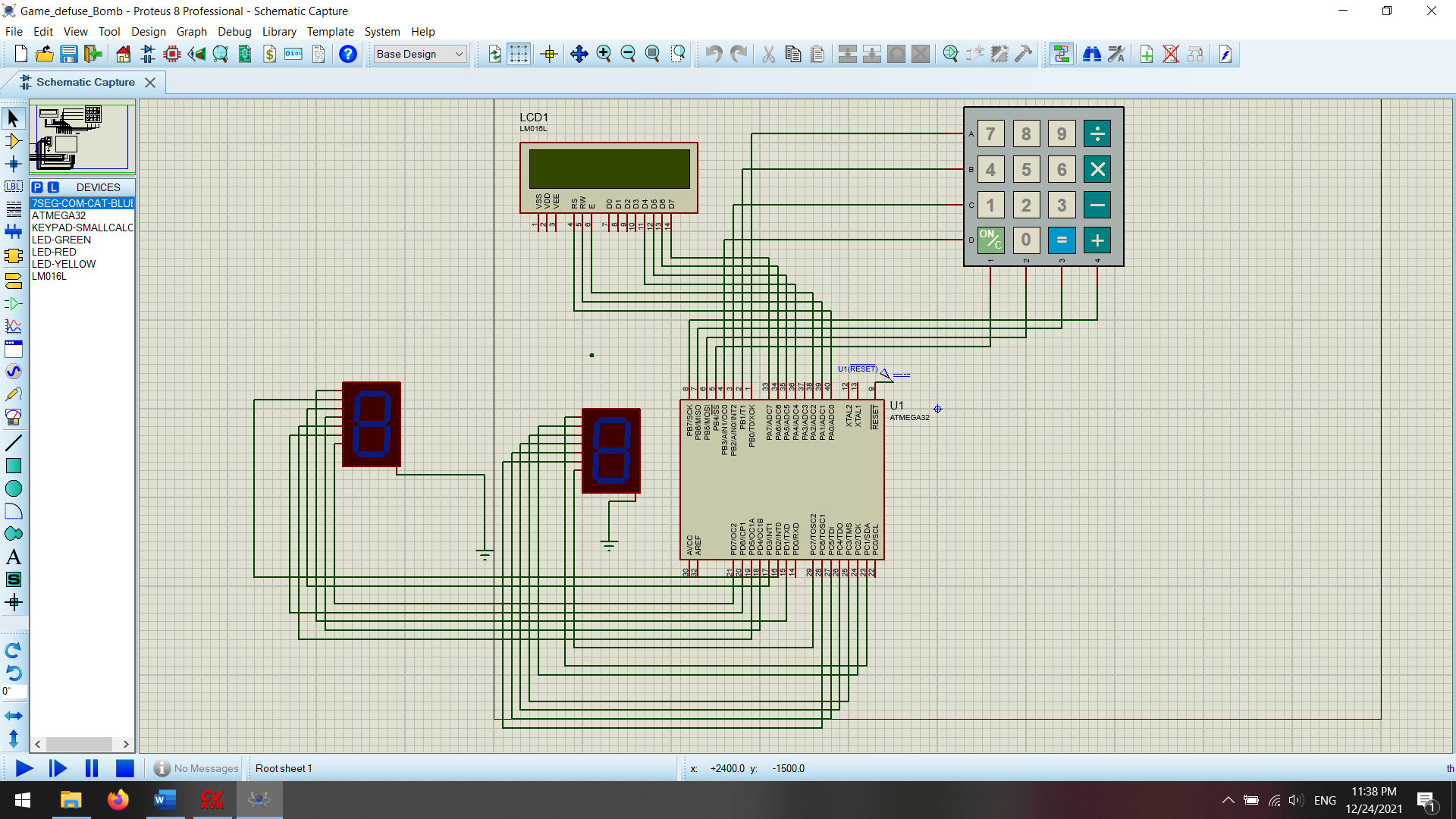
if user push key ( + ) for 10 times , LCD will show the list from the start ( 0 ),

Because it implements like state machine .

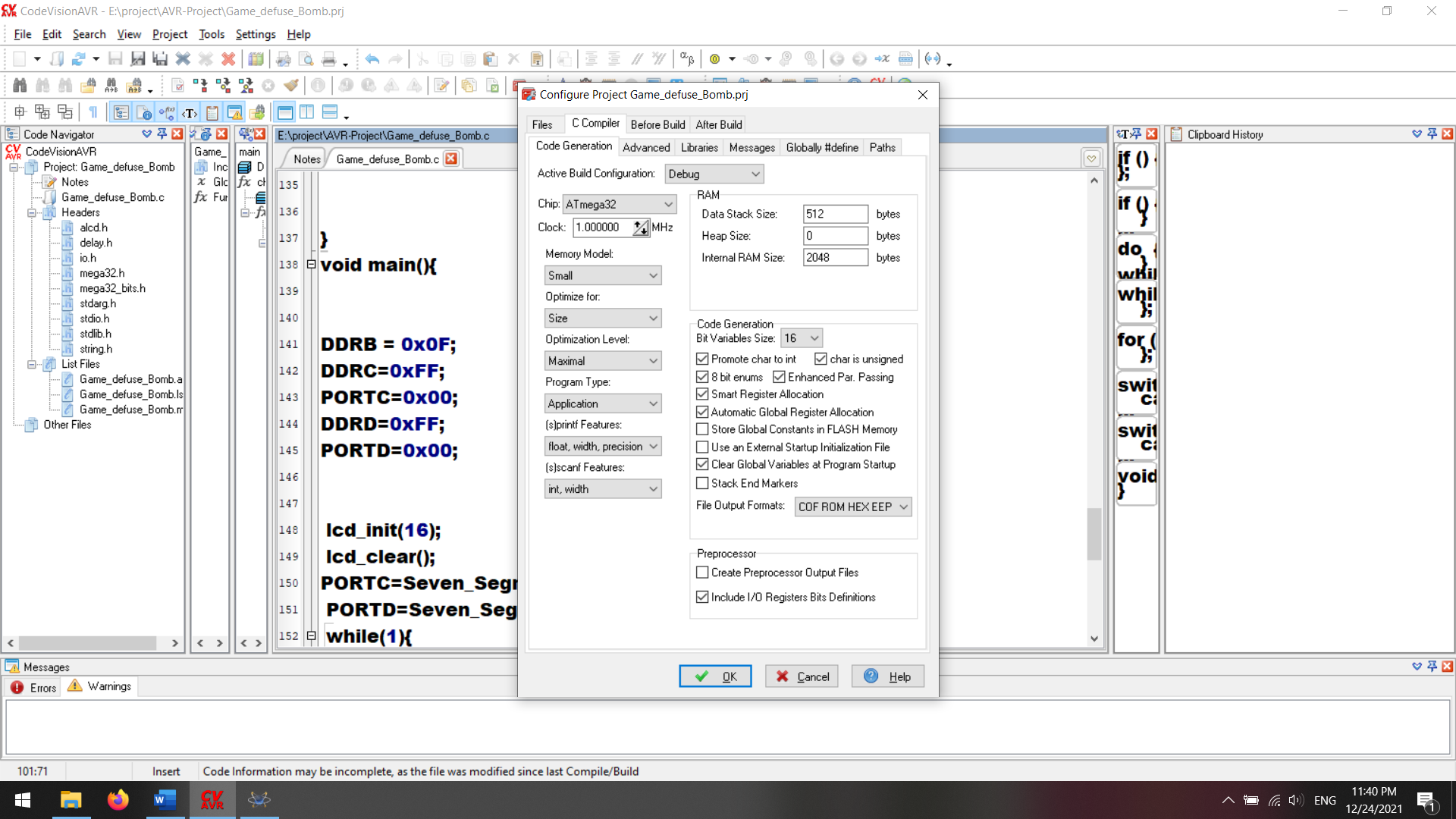
**ADVICE / IDEA:**

It can be promoted by adding this option : UNSELECT ITEMS BY USER

* **View of proteus :**



* **Requirement:**

****

* **Source code AVR:**

**#include <io.h>**

**#include <mega32.h>**

**#include <delay.h>**

**#include <alcd.h>**

**#include <string.h>d**

**#include <stdio.h>**

**#include <stdlib.h>**

**char key= '0';**

**float total = 0;**

**int state = 0;**

**char Buf[16];**

**int row ;**

**int col;**

**char scan [4] = {0XFE,0XFD,0XFB,0XF7};**

**char list [10][16] = { "0- crisp" , "1- water" , "2- pepsi" , "3- coca" , "4- chips" , "5- crisp" , "6- soda" , "7- cold tea" , "8- Ice cream" , "9- pop corn"};**

**float price[10] = {3 , 1.5 , 2 , 2.5 , 2 , 4 , 1 , 5 , 4.5 , 2};**

**char keypad [4][4] = { {'7','8','9',' '},{'4','5','6',' '},{'1','2','3','-'},{'c','0','=','+'}};**

**char Data=0;**

**char Data2=0;**

**char Seven\_Segment(char Input\_Data){**

**char K;**

**switch (Input\_Data){**

**case 0: K=0x7E; return K; break;**

**case 1: K=0x0C; return K; break;**

**case 2: K=0xB6; return K; break;**

**case 3: K=0x9E; return K; break;**

**case 4: K=0xCC; return K; break;**

**case 5: K=0xDA; return K; break;**

**case 6: K=0xFA; return K; break;**

**case 7: K=0x0E; return K; break;**

**case 8: K=0xFE; return K; break;**

**case 9: K=0xDE; return K; break;**

**default: K=0x00; return K; break;} }**

**char key\_pad(){**

**while(1){**

**for(row=0; row <=3; row++){**

**PORTB = scan[row];**

**col = 5;**

**if(PINB.4 ==0){**

**col = 0;**

**}**

**if(PINB.5 ==0){**

**col = 1;**

**}**

**if(PINB.6 ==0){**

**col = 2;**

**}**

**if(PINB.7 ==0){**

**col = 3;**

**}**

**if(col != 5){**

**delay\_ms(250);**

**return keypad[row][col];**

**}**

**}**

**}**

**}**

**int result(){**

**lcd\_clear();**

**lcd\_gotoxy(0,0);**

**lcd\_puts("Total cost :");**

**lcd\_gotoxy(0,1);**

**sprintf(Buf,"%.1f $", total);**

**lcd\_puts(Buf);**

**return 1;**

**}**

**void newState(){**

**lcd\_clear();**

**lcd\_gotoxy(0,0);**

**lcd\_puts(list[state]);**

**lcd\_gotoxy(0,1);**

**sprintf(Buf,"%.1f $", price[state]);**

**lcd\_puts(Buf);**

**}**

**int checking(){**

**if(key == '='){**

**return result();**

**}**

**else if(key == 'c'){**

**total += price[state];**

**Data++;**

**if(Data == 10){**

**Data = 0;**

**Data2++;**

**}**

**PORTC=Seven\_Segment(Data);**

**PORTD=Seven\_Segment(Data2);**

**}**

**else if(key == '+'){**

**state++;**

**if (state == 10){**

**state = 0;**

**}**

**newState();**

**}**

**else if(key == '-'){**

**state--;**

**if (state == -1){**

**state = 9;**

**}**

**newState();**

**}**

**else if(key >= '0' && key <= '9'){**

**state = key - '0' ;**

**newState();**

**}**

**else{**

**lcd\_clear();**

**lcd\_gotoxy(0,0);**

**lcd\_puts(list[state]);**

**lcd\_gotoxy(0,1);**

**sprintf(Buf,"%.1f $", price[state]);**

**lcd\_puts(Buf);**

**}**

**}**

**void main(){**

**DDRB = 0x0F;**

**DDRC=0xFF;**

**PORTC=0x00;**

**DDRD=0xFF;**

**PORTD=0x00;**

**lcd\_init(16);**

**lcd\_clear();**

**PORTC=Seven\_Segment(Data);**

**PORTD=Seven\_Segment(Data2);**

**while(1){**

**checking();**

**key = key\_pad();**

**}**

**}**

* **Source files :**

[**https://github.com/aminallahrabi/Microprocessor-Laboratory**](https://github.com/aminallahrabi/Microprocessor-Laboratory)