BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE , PILANI

DESIGN ASSIGNMENT (SODA DISPENSING MACHINE)

MICROPROCESSORS AND INTERFACING (CS F241)

DONE BY :-

1. **SURAVARAPU V SUBRAHMANYA KRISHNA SASTRY (2017A7PS0237P)**
2. **AREPALLE HIMA SIVA KALYAN REDDY (2017A7PS0235P)**
3. **ABHISHEK GUPTA (2017A7PS0236P)**
4. **DARSHAN AGRAWAL (2017A7PS0233P)**

1. PROBLEM STATEMENT

Three different types of cold drinks can be dispensed by the machine. The cool drink is available in three quantities: Small, Medium and Large.

There are three buttons available to select the cold drink type and another three buttons to select quantity. The user selects the drink, the quantity and then presses a button labelled dispense.

LEDs are available with each button. When a choice is made the corresponding LED glows and turns off when the dispensing is completed.

There are three more LEDs available that are used to indicate when a particular type of cool drink is not available.

The cost is Rs: 5.00, Rs.10.0 and Rs 15.0 respectively. There is a coin slot that accepts five rupee coins only. User can select type of cool drink, desired quantity and then drop the required number of coins. Each type of cool drink has its own dispenser. Based on the user's choice of drink the corresponding outlet will be open. The quantity of drink dispensed has to be accurately monitored. The quantity of drink is based on user's choice and the number of coins dropped in by the user.

2. ASSUMPTIONS

1. The 3 cold drinks come in 3 quantities each: small (50 mL), medium (100 mL), large (150 mL), and the user can order only 1 drink of selected type and size.
2. All 3 tanks are 10" (length) x 6.1" (breadth) x 12.2" (height), and the initial height of liquid in all tanks is 10.2".
3. User is first supposed to select the type of soda he's willing to buy by pressing the corresponding button.

4. There are three LEDs to represent the sodas unavailable (if any). If any type of soda is unavailable, corresponding “UNAVAILABLE” LED glows. So, if he selects the soda which is unavailable and drops the coin, it is going to be wasted.
5. The user is not supposed to insert anything other than FIVE Rupee coins.
6. Once the user selects the soda, he can’t undo the selection. He must finish the transaction by selecting some quantity (least is preferred as he doesn’t want to waste his money).
7. If the number of coins inserted is more than what is required based on the user’s selection, drink is dispensed based on user’s selection, but not the number of coins inserted, and the excess coins are the user’s loss.
8. If the user inserts less coins than what is required based on selection, the liquid will not dispens unless user inserts the correct number of coins.

3. PROCESS FOR DISPENSING :

User selects the Cool-Drink he wants by pressing the button of corresponding drink, then the respective LED of selected cool drink glows. He then chooses the quantity of drink by pressing the button of required quantity, then the respective LED of quantity selected glows. If the cool drink he selected is not available, then he will not be allowed to select the quantity of drink he wanted. After selecting all these things, he needs to insert required number of coins. He cannot insert lesser number of coins. Then he needs to press “Dispense” button. Dispense LED glows and motor of respective drink rotates thereby dispensing required drink.

4. SYSTEM SPECIFICATIONS:

3 LEDs are used to indicate the brand of cold drink selected.

3 LEDs are used to show whether a brand of cold drink is available or unavailable.

3 LEDs are used to show the quantity to be dispensed.

1 LED to show that the drink is being dispensed.

5. HARDWARE DESCRIPTION

NAME	QUANTITY
Intel 8086 Microprocessor	1
Intel 8255A Programmable Peripheral Interface	2
2N2222 Central Semiconductor	12
74LS28	1
74LS32 - Quad 2-Input OR Gate	10
4075	6
74LS138 3-to-8 Line Decoder	1
74LS245 Octal Bus Transceivers	2
74LS373 Octal Latch	3
2732 - EPROM	4
4075 - Triple 3-Input OR Gate	6
6116 - SRAM	2
BUTTON	8
LED-BLUE	10
LM139 - Quad Differential Comparator	3
MOTOR-STEPPER	3
NOT	2
RESISTOR 10 KILO-OHMS	19
RESISTOR 20 OHMS	3
SW-SPDT	1
TSL251RD	3

6. SYSTEM DESIGN

6.1 MEMORY INTERFACING

This system is 8086 microprocessor based, and it uses **4KB of RAM** – 2 (6116) chips of 2KB size each and **16KB ROM** – 4 (2732) chips of 4KB size each. Both RAM and ROM are organized into Even and Odd Banks to facilitate both Byte-Sized and Word-Sized data transfers.

Read Only Memory (ROM - 1):-

Starting address – **00000H**

Ending address – **01FFFH**

Even bank begins at **00000H** and ends at **01FFEH**

Odd bank begins at **00001H** and ends at **01FFFH**

Random Access Memory (RAM):-

Starting address – **02000H**

Ending address – **02FFFH**

Even bank begins at **02000H** and ends at **02FFEH**

Odd bank begins at **02001H** and ends at **02FFFH**

Read Only Memory (ROM - 2) :-

Starting address – **04000H**

Ending address – **05FFFH**

Even bank begins at **04000H** and ends at **05FFEH**

Odd bank begins at **04001H** and ends at **05FFFH**

The code resides in the ROM. The address that is loaded as soon as the system is switched on is FFFF0H. There, there is a JMP instruction at FFFF0H to jump to the beginning of the code.

Address Map

A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08	A07
	A06	A05	A04	A03	A02	A01	A00	HEX ADD		Device		
0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	**00000 H**		**ROM - 1**		
0	0	0	0	0	0	0	1	1	1	1	1	1
	1	1	1	1	1	1	1	**01FFF H**		**ROM - 1**		
0	0	0	0	0	0	1	0	0	0	0	0	0
	0	0	0	0	0	0	0	**02000 H**		**RAM**		
0	0	0	0	0	0	1	0	1	1	1	1	1
	1	1	1	1	1	1	1	**02FFF H**		**RAM**		
0	0	0	0	0	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	**04000 H**		**ROM - 2**		
0	0	0	0	0	1	0	1	1	1	1	1	1
	1	1	1	1	1	1	1	**05FFF H**		**ROM - 2**		

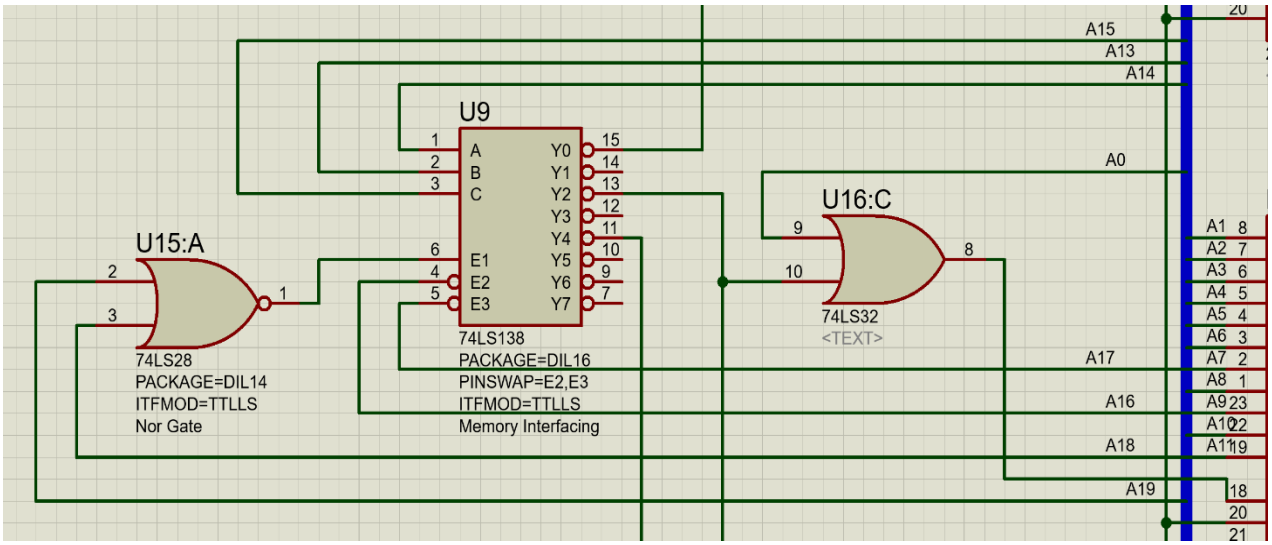


Fig.i Depicting LS-138 through which memory has been interfaced

6.2 I/O INTERFACING

8255 (1)

Port	Port Address	Input/Ouput	Connected to
A	00H	Input	PA-0 to PA-6 connected to Buttons (for taking soda type and quantity as input and dispense button)
B	02H	Output	PB-0 to PB-5 are connected to LEDs (Soda Type (3) followed by Quantity (3))
C Lower	04H	Output	PC-0 to PC-2 are connected to unavailable LEDs and PC-3 to Dispense LED
C Upper	04H	Not Used	Not connected to anything

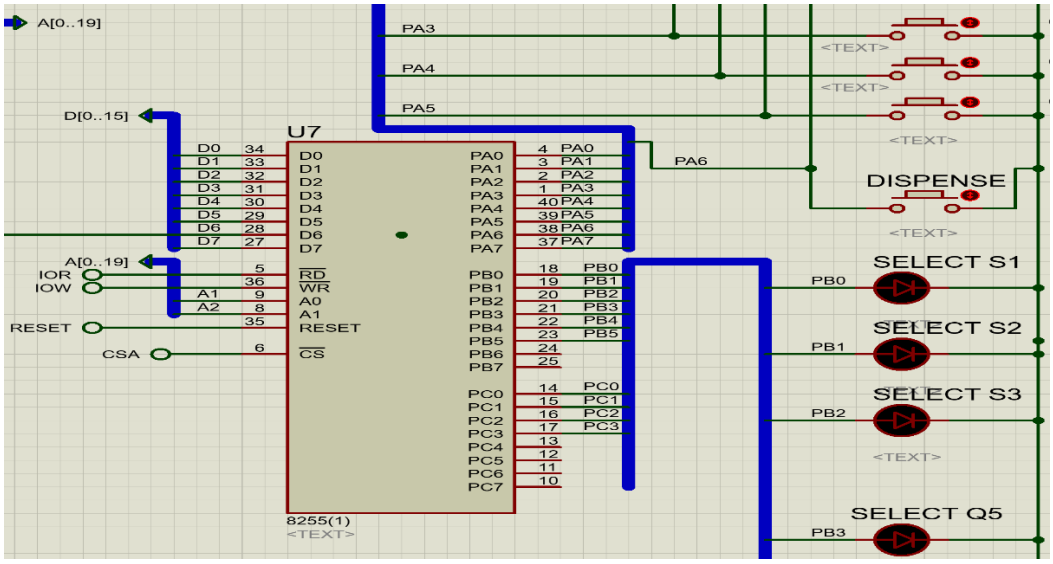


Fig. ii Depicting 8255(1) and all its connections.

8255 (2)

Port	Port Address	Input/Ouput	Connected to
A	10H	Output	PA-0 to PA-3 is connected to Stepper motor to dispense Soda – 1 and PA-4 to PA-7 connected to dispense Soda – 2
B	12H	Input	PB-0 to PB-2 are connected to Quad Differential Comparators to get input about availability
C Lower	14H	Output	PC-0 to PC-3 used to provide output to Stepper Motor for Soda - 3
C Upper	14H	Input	PC-7 is to Coin acceptor switch and measures number of coins inputted.

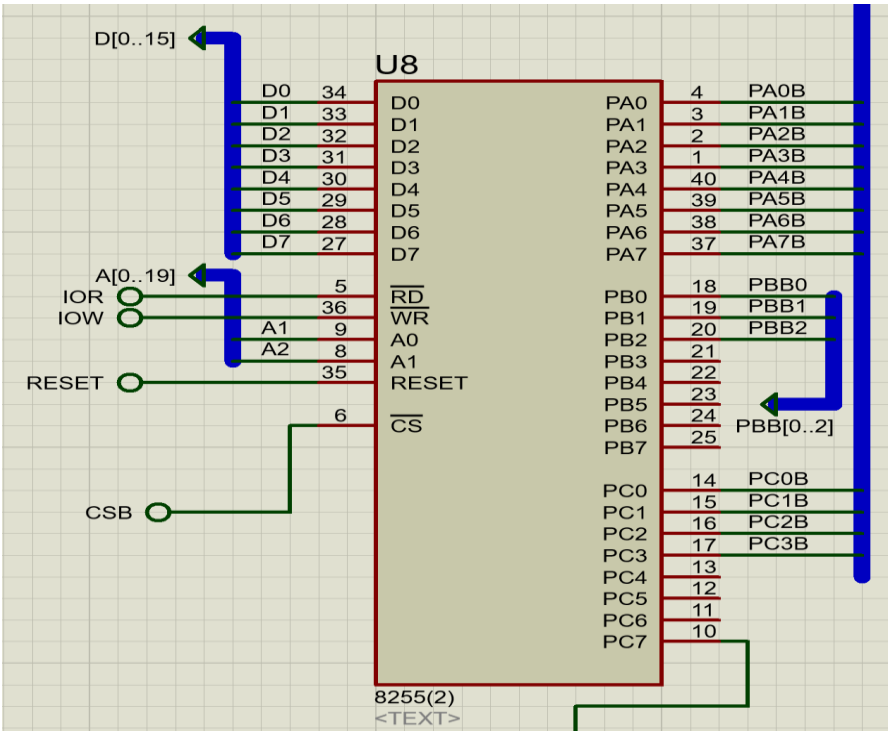


Fig. iii Depicting 8255(2) and all its connections.

7.KEY COMPONENTS

STEPPER MOTOR

A Stepper Motor is a synchronous motor which divides a full rotation into a number of steps. Unlike a DC motor which rotates continuously when a fixed DC voltage is applied to it, a step motor rotates in discrete step angles. The Stepper Motors therefore are manufactured with steps per revolution of 12, 24, 72, 144, 180, and 200, resulting in stepping angles of 30, 15, 5, 2.5, 2, and 1.8 degrees per step. The stepper motor can be controlled with or without feedback. Each stepper motor used here, is connected to 4 voltage regulators, and a Voltage source and ground.

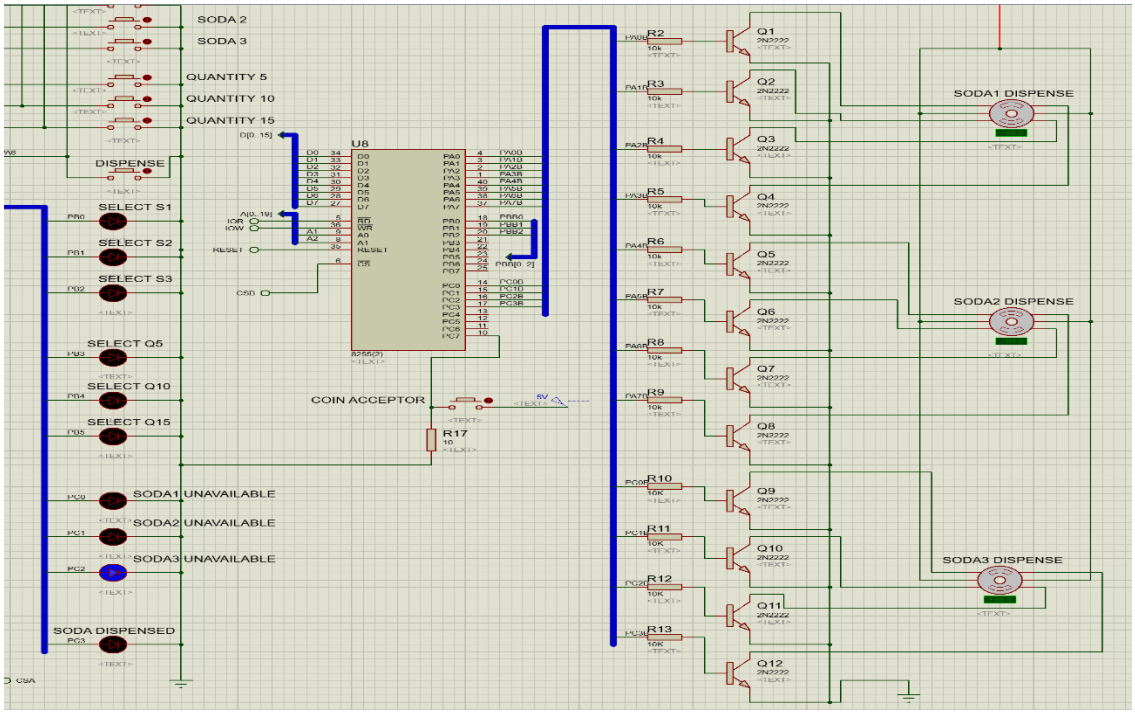


Fig. iv Depicting Stepping Motors connected to 8255(2).

LM-139

It is a "Quad Differential Comparator" which consists of four different independent voltage comparators that operate on a single power supply over a wide range of voltages. The outputs can be connected to other open-collector outputs to achieve 'wired-AND' relationships.

TSL251RD

It is a light-to-voltage optical sensor, combining a photodiode and a trans-impedance amplifier on a single monolithic IC. Output voltage is directly proportional to light intensity on the photodiode. Here the sensors are actually used to notice if any soda is not available using light sensors within them.

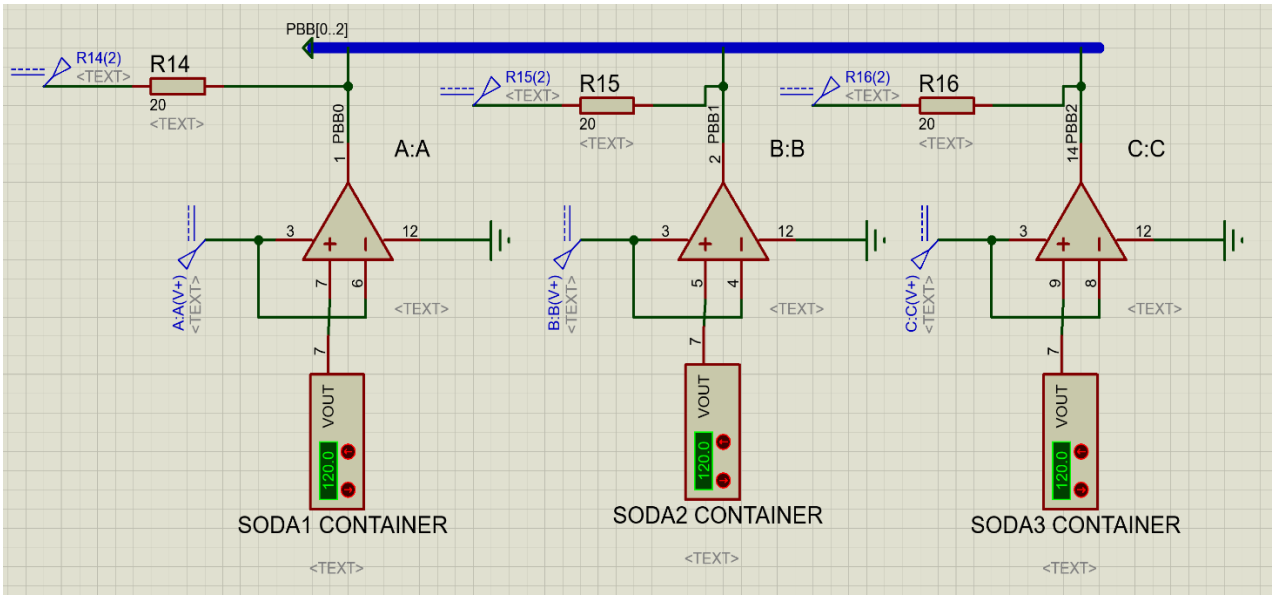


Fig. v Depicting LM-139 and TSL251RD

8. ALGORITHM

1. The 3 switches of the 3 Sodas are checked in an infinite loop. Pressing a particular soda button switches on the respective LED and the Quantity buttons are then checked endlessly.

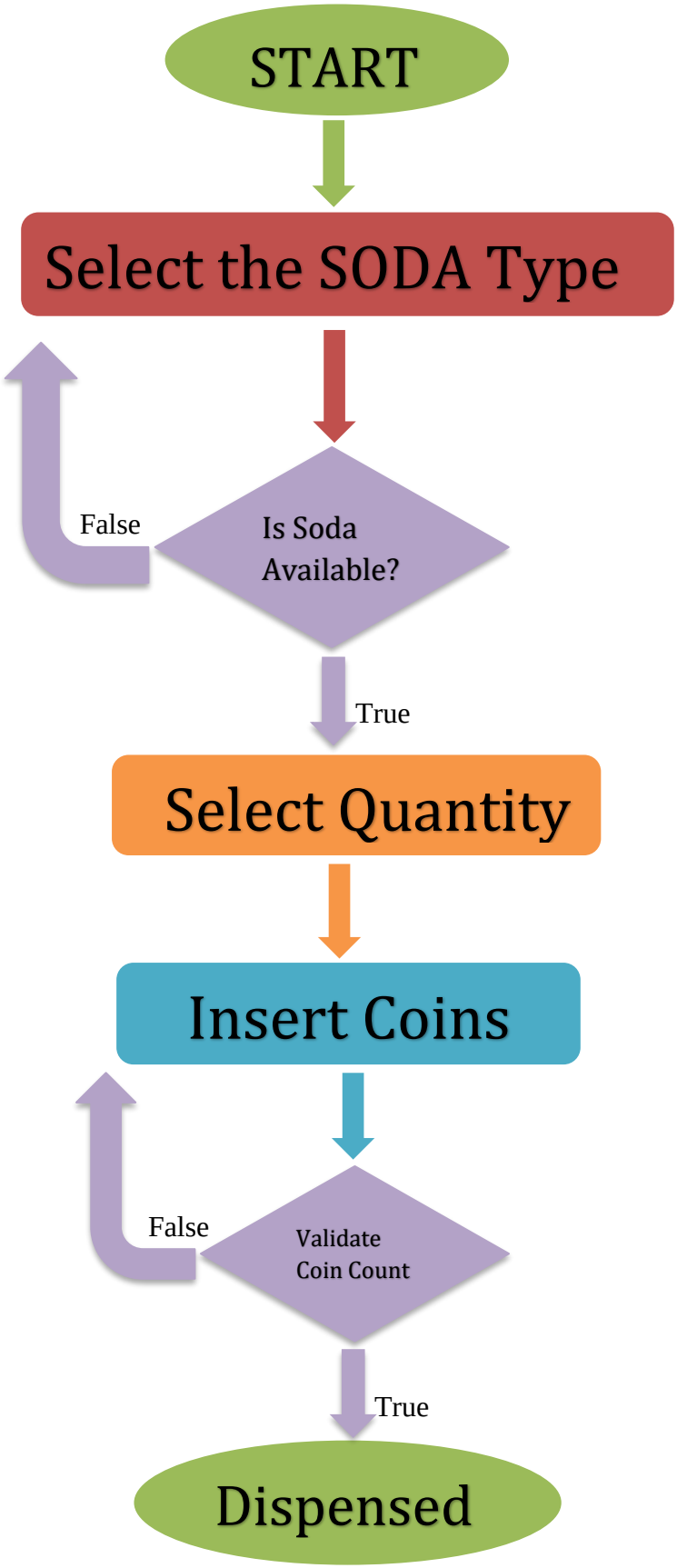
2. Once the user has pressed the Quantity Button the corresponding LED is lit up .

3. It then starts to count the number of coins entered.

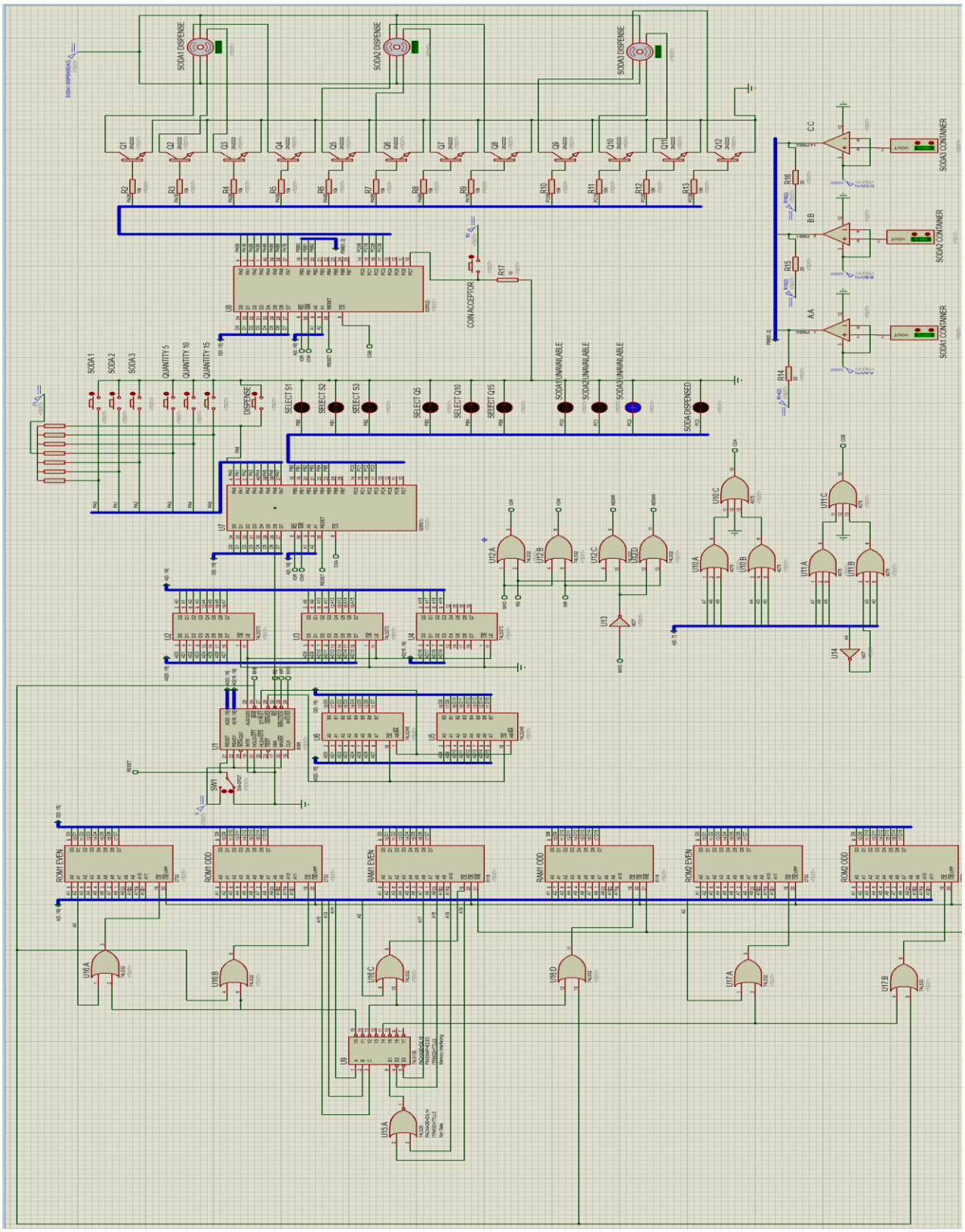
3.The number of coins entered is checked, once the dispense button is pressed.

4.If the correct number of coins have been entered then the drink is dispensed else it waits for user to enter at least required number of coins.

9. FLOW CHART


START
Select the SODA Type
False
Is Soda Available?
True
Select Quantity
Insert Coins
False
Validate Coin Count
True
Dispensed

10. TOTAL SCHEMATIC CAPTURE OF MACHINE



11. CODE

```
.model tiny

.data

PORTA1      EQU   00H          ;INITIALISING ADDRESSES FOR 8255(1)

PORTB1      EQU   02H

PORTC1      EQU   04H

CTR1  EQU   06H                ;CONTROL WORD ADDRESS


PORTA2      EQU   10H          ;INITIALISING ADDRESSES FOR 8255(2)

PORTB2      EQU   12H

PORTC2      EQU   14H

CTR2  EQU   16H                ;CONTROL WORD ADDRESS


UNAVAILABLE      DB     ?      ;UNAVAILABILITY  BYTE OBTAINED FROM
                               PB0-   PB7 OF 8255(2)
QUANTSELECTED  DB     ?       ;QUANTITY SELECTED BYTE OBTAINED FROM
                               PA0-PA7 OF 8255(1)
SODASELECTED    DB     ?      ;SODA SELECTED BYTE OBTAINED FROM
                               PA0-PA7 OF 8255(1)


ANTI_ROTATE1      DB     01H,02H,04H,08H,09H
;ANTICLOCKWISE ROTATION SEQUENCE FOR MOTOR SODA1,3
CLOCK_ROTATE1  DB     08H,04H,02H,01H,09H
;CLOCKWISE ROTATION SEQUENCE FOR MOTOR SODA1,3
ANTI_ROTATE2      DB     10H,20H,40H,80H,90H
```

```
;ANTICLOCKWISE ROTATION SEQUENCE FOR MOTOR SODA2
CLOCK_ROTATE2   DB      80H,40H,20H,10H,90H
;CLOCKWISE ROTATION SEQUENCE FOR MOTOR SODA2
;SODA 1 and 3 can have same rotate and anti rotate signals as they are connected to different ports
whereas 2 is connected to same port as 1 so it should have rotate signal in its upper nibble thereby
rendering lower nibble signal low
.stack
.code
.startup

BEGIN:      MOV   AL, 10010000B
        ;CONTROL WORD INITIALISATION FOR 8255(1) AND 8255(2)
        OUT   CTR1, AL              ; A1 port is used for Input
        MOV   AL, 10001010B
        OUT   CTR2, AL              ; C upper 2 and B2 Ports are used for Input
        MOV   AL, 0
        OUT   PORTB1, AL
        OUT   PORTC1, AL
        IN    AL, PORTB2            ;CHECKING SODA UNAVAILABILITY
        MOV   BL, AL
        AND   BL, 00000111B
        ROR   BL, 1
        JC    X1
        MOV   AL, 00H               ;X1,X2,X3 ARE LOCATED AT END OF CODE
X1A:    ROR   BL, 1
```

```
        JC      X2
X2A:    ROR     BL, 1
        JC      X3
X3A:    OUT     PORTC1, AL              ;OUTPUT for LEDS of Unavailability
        MOV     UNAVAILABLE, AL
XS:     IN      AL,PORTA1               ;SODA SELECTION
        NOT     AL
        AND     AL,00000111B
        CMP     AL,0
        JA      SELECT
        JMP     XS
SELECT:     OUT     PORTB1,AL           ;Glow corresponding LED of SODA TYPE
        MOV     SODASELECTED,AL
        AND     AL,UNAVAILABLE
;CHECKING CONFLICT WITH UNAVAILABILITY
        CMP     AL,0
        JNZ     XT

XQ:     IN      AL,PORTA1               ;QUANTITY SELECTION
        NOT     AL
        AND     AL,00111000B
        CMP     AL,0
        JA      X4
        JMP     XQ
X4:     MOV     QUANTSELECTED,AL
```

```
        OR      AL,SODASELECTED
        OUT     PORTB1,AL
        MOV     AL,QUANTSELECTED        ;COIN ACCEPTANCE
        CMP     AL,00001000B
        JZ      X5
        CMP     AL,00010000B            ;X5,X6,X7 LOCATED AT END OF CODE
        JZ      X6
        CMP     AL,00100000B
        JZ      X7
COIN:   CALL    DELAY
        ;TIME DELAY SO THAT SINGLE PUSH OF BUTTON IS NOT COUNTED AS MANY
        IN      AL,PORTC2
        AND     AL,80H
        CMP     AL,80H
        JNZ     COIN
        DEC     CL
        CMP     CL,0
        JNZ     COIN
DISPENCE:   IN      AL,PORTA1
        ;GLOWING DISPENSED LED AFTER PRESSING DISPENSE BUTTON
        NOT     AL
        AND     AL,01000000B
        CMP     AL,40H
        JNZ     DISPENCE
        MOV     AL,08H
```

```
        OR      AL,UNAVAILABLE
        OUT     PORTC1,AL
        MOV     AL,SODASELECTED
        ;ROTATING THE REQUIRED MOTOR CLOCKWISE AFTER
        CLD
        CMP     AL,00000001B
        ;CHECKING WHICH SODA WAS SELECTED USING SODASEL
        JNZ     Y1
        LEA     SI,CLOCK_ROTATE1
F1:     LODSB
        OUT     PORTA2,AL
        CALL    DELAYS
        CMP     AL,09H
        JNZ     F1
        JMP     Q1
Y1:     CMP     AL,00000010B
        JNZ     Y2
        LEA     SI,CLOCK_ROTATE2
F2:     LODSB
        OUT     PORTA2,AL
        CALL    DELAYS
        CMP     AL,90H
        JNZ     F2
        JMP     Q1
```

```
Y2:   LEA   SI,CLOCK_ROTATE1
F3:   LODSB
      OUT   PORTC2,AL
      CALL  DELAYS
      CMP   AL,09H
      JNZ   F3
Q1:   MOV   AL,QUANTSELECTED     ;SETTING TIME DELAY TO DISPENSE QUANTITY
      CMP   AL,00001000B         ;AFTER CHECKING QUANTITY SELECTED USING
QUASEL
      JNZ   Q2
      MOV   CX,6
      CALL  DELAY1
      JMP   Z1
Q2:   CMP   AL,00010000B
      JNZ   Q3
      MOV   CX,12
      CALL  DELAY1
      JMP   Z1
Q3:   MOV   CX,18
      CALL  DELAY1
Z1:   MOV   AL,SODASELECTED
      ;CLOSING VALVE BY ROTATING MOTOR ANTI CLOCKWISE.
      CLD
      ;SAME LOGIC AS CLOCKWISE ROTATION ABOVE USING SODASEL
      CMP   AL,00000001B
```

```
        JNZ   Y1B
        LEA   SI,ANTI_ROTATE1
F1B:    LODSB
        OUT   PORTA2,AL
        CALL  DELAYS
        CMP   AL,09H
        JNZ   F1B
        JMP   XT
Y1B:    CMP   AL,00000010B
        JNZ   Y2B
        LEA   SI,ANTI_ROTATE2
F2B:    LODSB
        OUT   PORTA2,AL
        CALL  DELAYS
        CMP   AL,90H
        JNZ   F2B
        JMP   XT
Y2B:    LEA   SI,ANTI_ROTATE1
F3B:    LODSB
        OUT   PORTC2,AL
        CALL  DELAYS
        CMP   AL,09H
        JNZ   F3B

XT:     MOV   CX,2              ;END OF PROGRAM
```

```
        CALL  DELAY1
        JMP   BEGIN
X1:     MOV   AL,01H
        JMP   X1A
X2:     OR    AL,00000010B
        JMP   X2A
X3:     OR    AL,00000100B
        JMP   X3A
X5:     MOV   CL,1
        JMP   COIN
X6:     MOV   CL,2
        JMP   COIN
X7:     MOV   CL,3
        JMP   COIN
.EXIT
DELAY   PROC  NEAR
        ;DELAY FOR COIN ACCEPTOR SO THAT SINGLE PUSH IS NOT COUNTED AS MANY
        MOV   BX,3000H
XD:     NOP
        DEC   BX
        JNZ   XD
        RET
DELAY   ENDP

DELAY1  PROC  NEAR          ;STEP DELAY OF 0.5 SECONDS WITH 1.5MHz
```

```
AB:     MOV  BX,8B82H
        ;USED TO SET DELAY FOR QUANTITY DISPENSING 3,6,9 SECONDS
AD:     NOP
        DEC  BX
        JNZ  AD
        DEC  CX
        JNZ  AB
        RET
DELAY1  ENDP
DELAYS  PROC NEAR          ;DELAY FOR MOTOR SIGNALS
        MOV  BX,1800H      ;SO MOTOR ROTATES PROPERLY
AF:     NOP
        DEC  BX
        JNZ  AF
        RET
DELAYS  ENDP
END
```

References

1. LM139 -

https://www.digikey.in/en/datasheets/stmicroelectronics/stmicroelectronics-encd00000460

2. TSL251RD –

https://www.mouser.com/ds/2/588/TSL250R-TSL251R-TSL252R_DS000429_1-00-1149939.pdf

3. Stepper Motor –

https://www.engineersgarage.com/articles/stepper-motors

THE END