Microcontroller and Microprocessors
Experiment - 5
Seven Segment Display and Interrupt Based Operation with Timers

Varun Singh Inda
16BEE0023

Aim :-To use the microcontroller for performing a seven segment display and using interrupts with timers to make a square wave interrupting by a LED.

Seven Segment Display

Procedure :-

1. Click on START.
2. Open Keil µvision5.
3. Open PROJECT. Create New µvision project.
4. Open Legacy Device Database.
5. Click on + next to NXP in the application box that opens.
6. Select P89V51RD2 from the list under NXP.
7. Click OK.
8. Proceed further clicking Yes.
9. In the main screen, select blank page icon present under File.
10. A new Text Window opens, where we are to write the program to be executed.
11. Go to File in the menu bar and save the program with the extension .asm.
12. In project window, select target on clicking + and chose Add existing file to source group1 and chose the program to be executed.
13. On clicking + next to SOURCE GROUP, right click, build target.
14. Click on Debug icon.
15. Start debug session.
16. Click on OK when the window pops in.
17. Press RUN or F5.

ALGORITHM:-

1. START
2. Initialize the R0 value
3. MOV the R0 value to P1 port
4. Assign different values to the P1 port and call Delay
5. Repeat the process till all the required inputs are programmed
6. Write the delay function for the program
7. End

Code1 :-

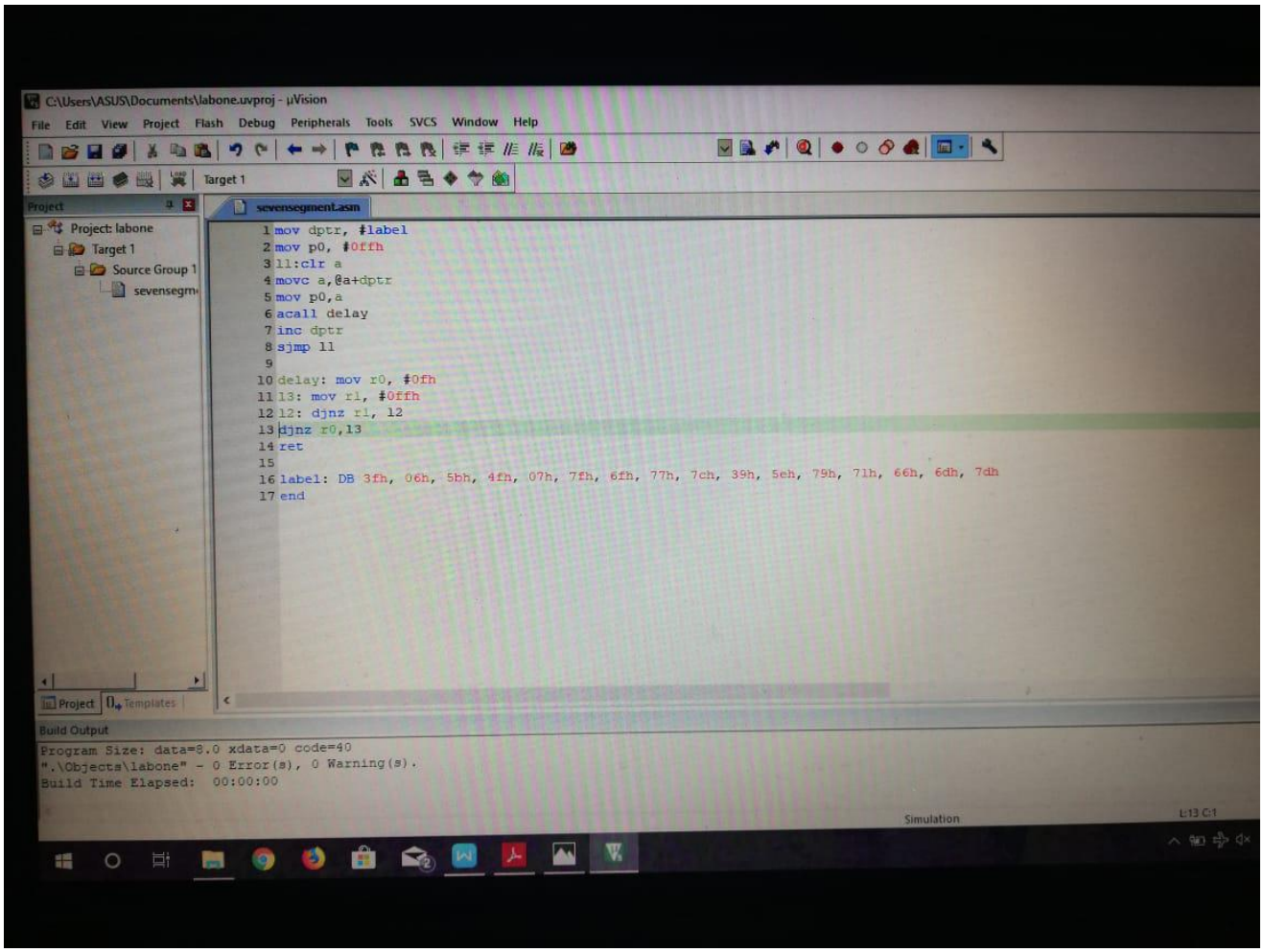
```
mov dptr, #label
mov p0, #0ffh
l1:clr a
movc a,@a+dptr
mov p0,a
acall delay
inc dptr
sjmp l1

delay: mov r0, #0fh
l3: mov r1, #0ffh
l2: djnz r1, l2
djnz r0,l3
ret

label: DB 3fh, 06h, 5bh, 4fh, 07h, 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h,
66h, 6dh, 7dh
End
```

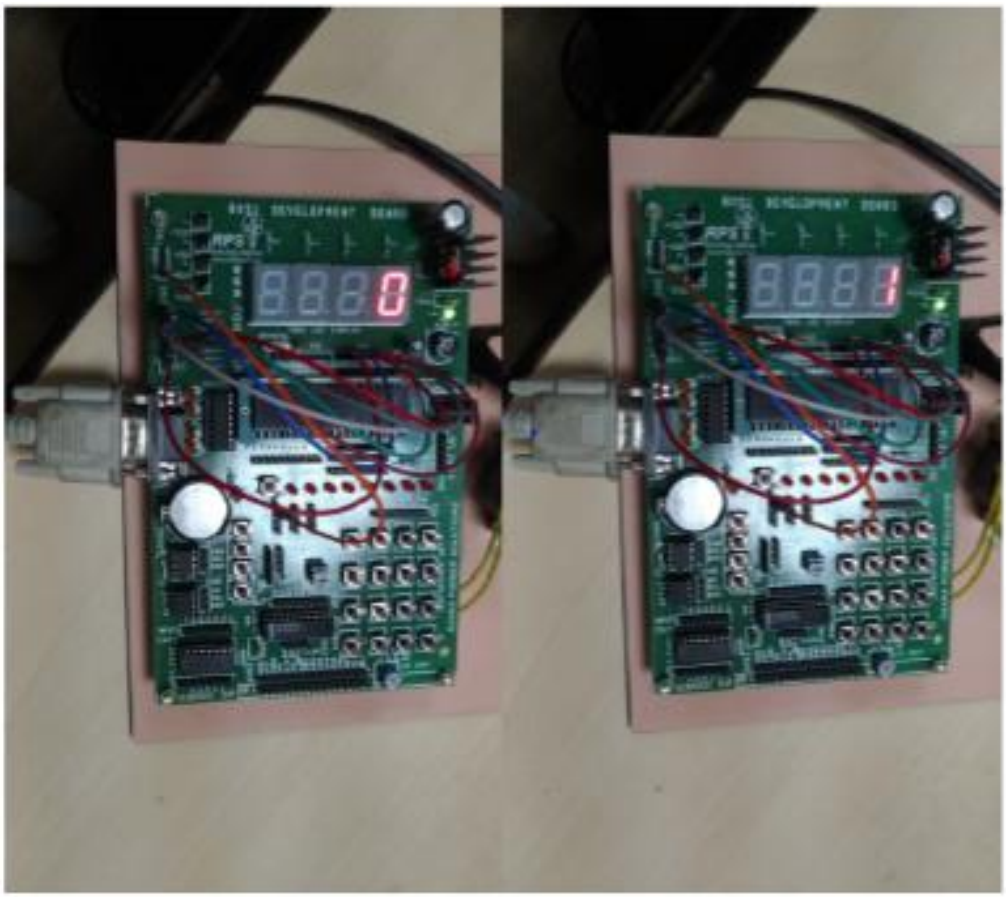
Code2 :-

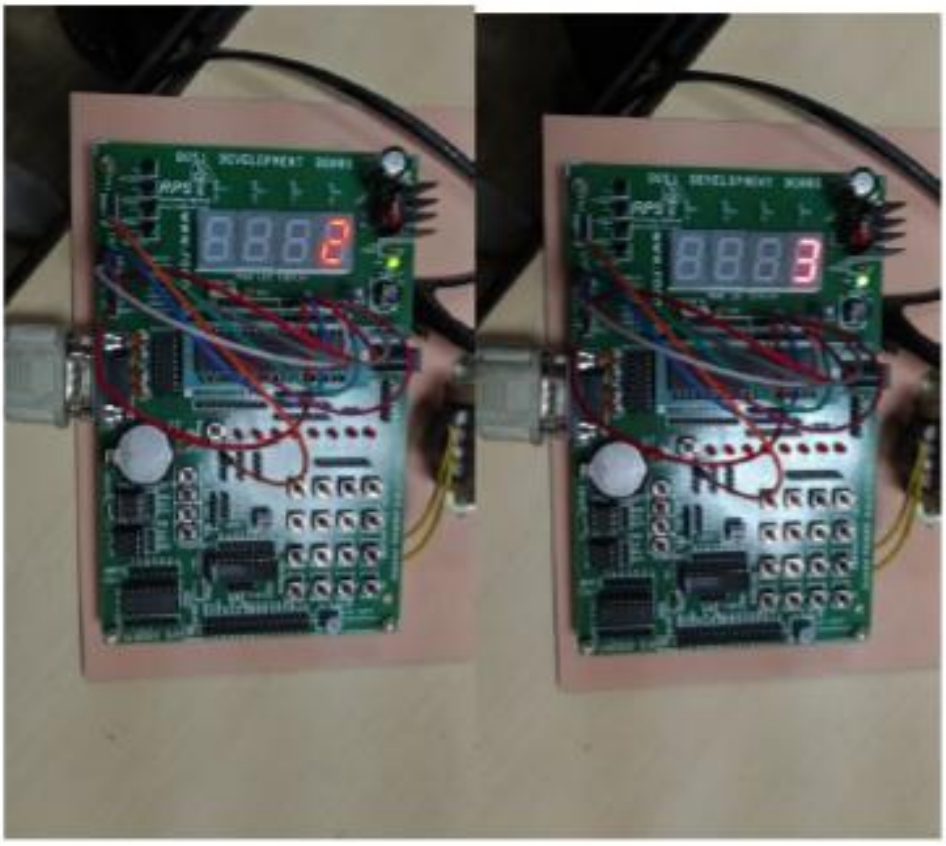
```
MOV R0,#0FFh
START:
MOV P1, R0
MOV P1, #35h
ACALL DELAY
MOV P1, #06h
ACALL DELAY
MOV P1, #5Bh
ACALL DELAY
MOV P1, #4Fh
ACALL DELAY
MOV P1, #66h
ACALL DELAY
MOV P1, #6Dh
ACALL DELAY
MOV P1, #7Dh
ACALL DELAY
MOV P1, #07h
ACALL DELAY
MOV P1, #7Fh
ACALL DELAY
MOV P1, #6Fh
```

```
ACALL DELAY
SJMP START
DELAY:
MOV R1, #0H
L2:
MOV R2, #0FFh
L1:
MOV R3, #0FFh
L:
DJNZ R3, L
DJNZ R2, L1
DJNZ R1, L2
RET
END
```

Pictures :-




0
1


2
3




Conclusion :-The experiment was successfully performed with the verification of the output obtained. The observed output is the same as the expected output.

Interrupt Based Operation with Timers

Procedure :-

1. Click on START.
2. Open Keil µvision5.
3. Open PROJECT. Create New µvision project.
4. Open Legacy Device Database.
5. Click on + next to NXP in the application box that opens.
6. Select P89V51RD2 from the list under NXP.
7. Click OK.
8. Proceed further clicking Yes.
9. In the main screen, select blank page icon present under File.
10. A new Text Window opens, where we are to write the program to be executed.
11. Go to File in the menu bar and save the program with the extension .asm.
12. In project window, select target on clicking + and chose Add existing file to source group1 and chose the program to be executed.
13. On clicking + next to SOURCE GROUP, right click, build target.
14. Click on Debug icon.
15. Start debug session.
16. Click on OK when the window pops in.
17. Press RUN or F5.

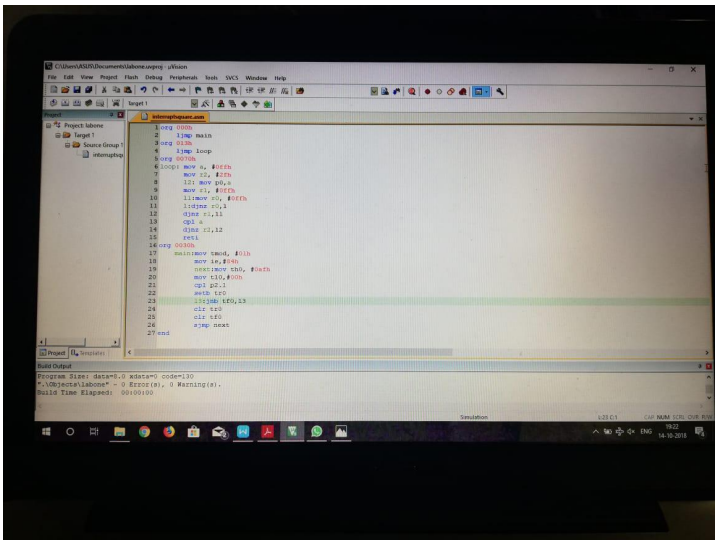
ALGORITHM:-

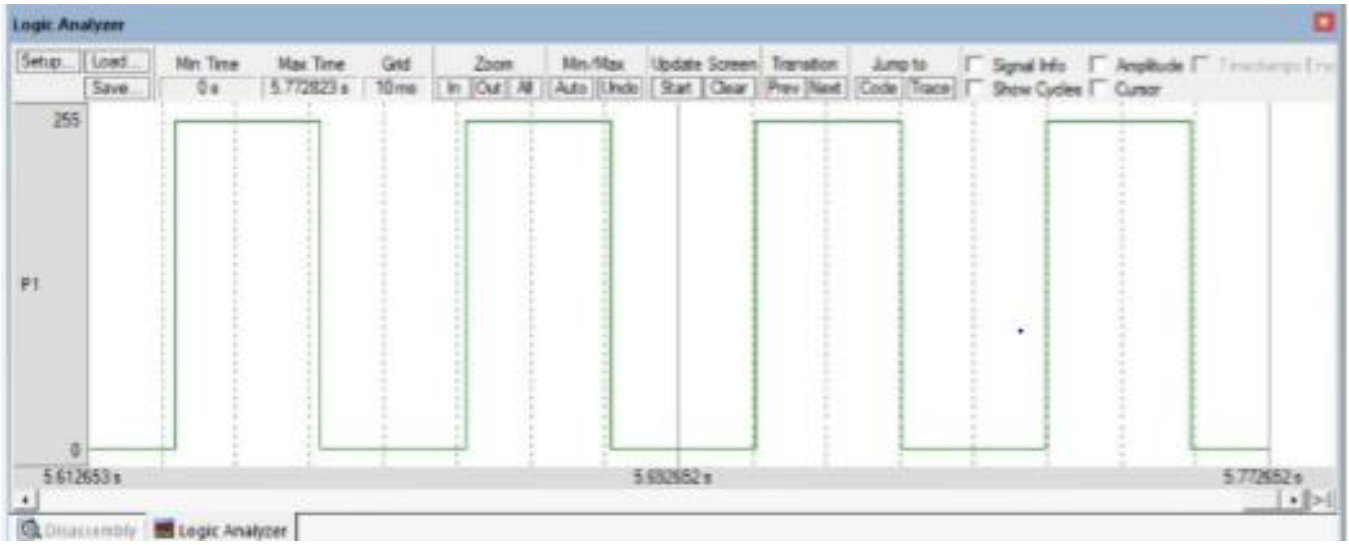
1. Start.
2. Set timer 1 in mode 1 operation
3. Define the label L and assign the values to accumulator
4. MOV the accumulator value to the port P1
5. Call Delay function
6. Change the value assigned to the port P1
7. Repeat the process
8. Write the delay function for the program
9. Set the values for TH0 and TL0
10. End

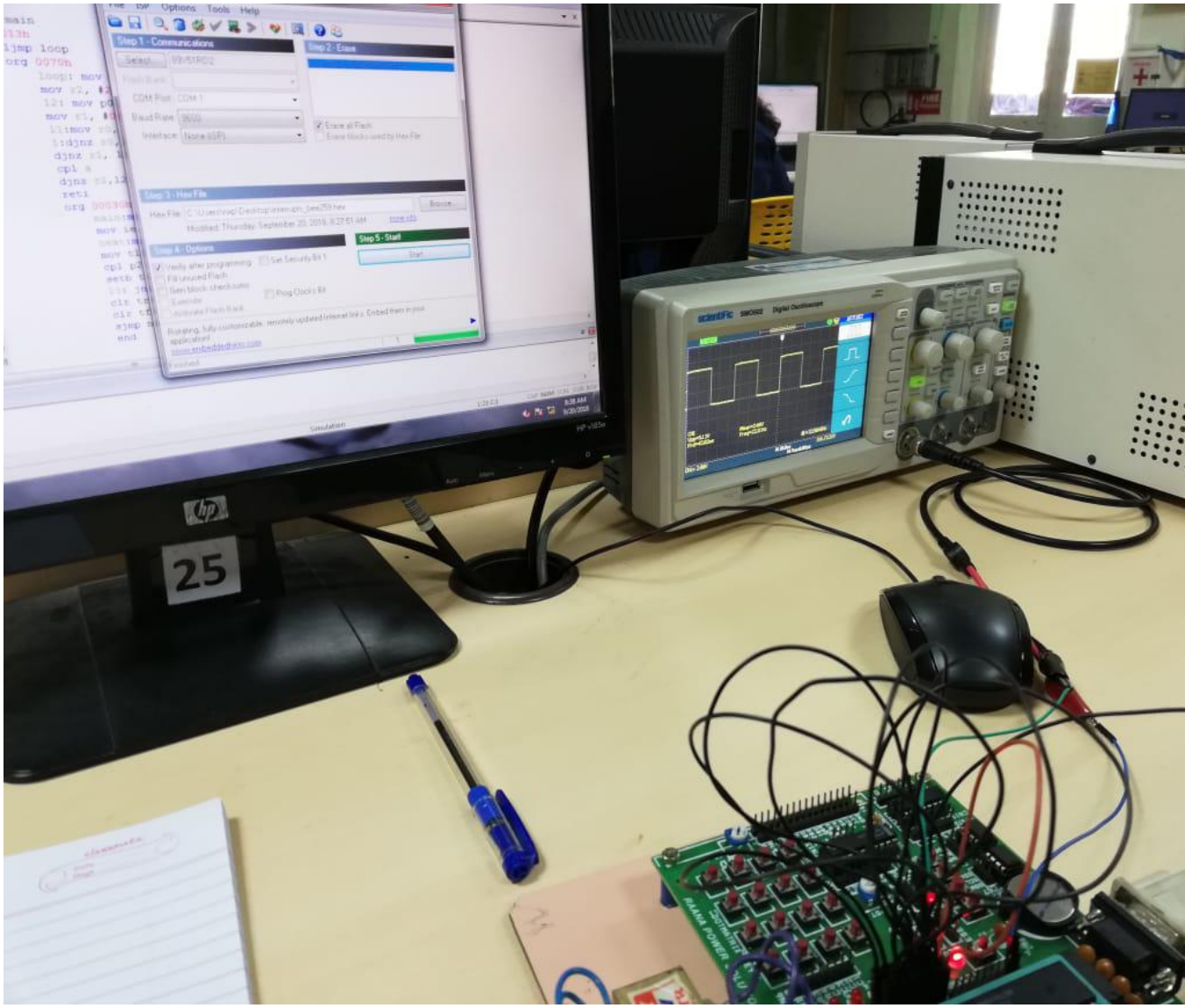
Code:-

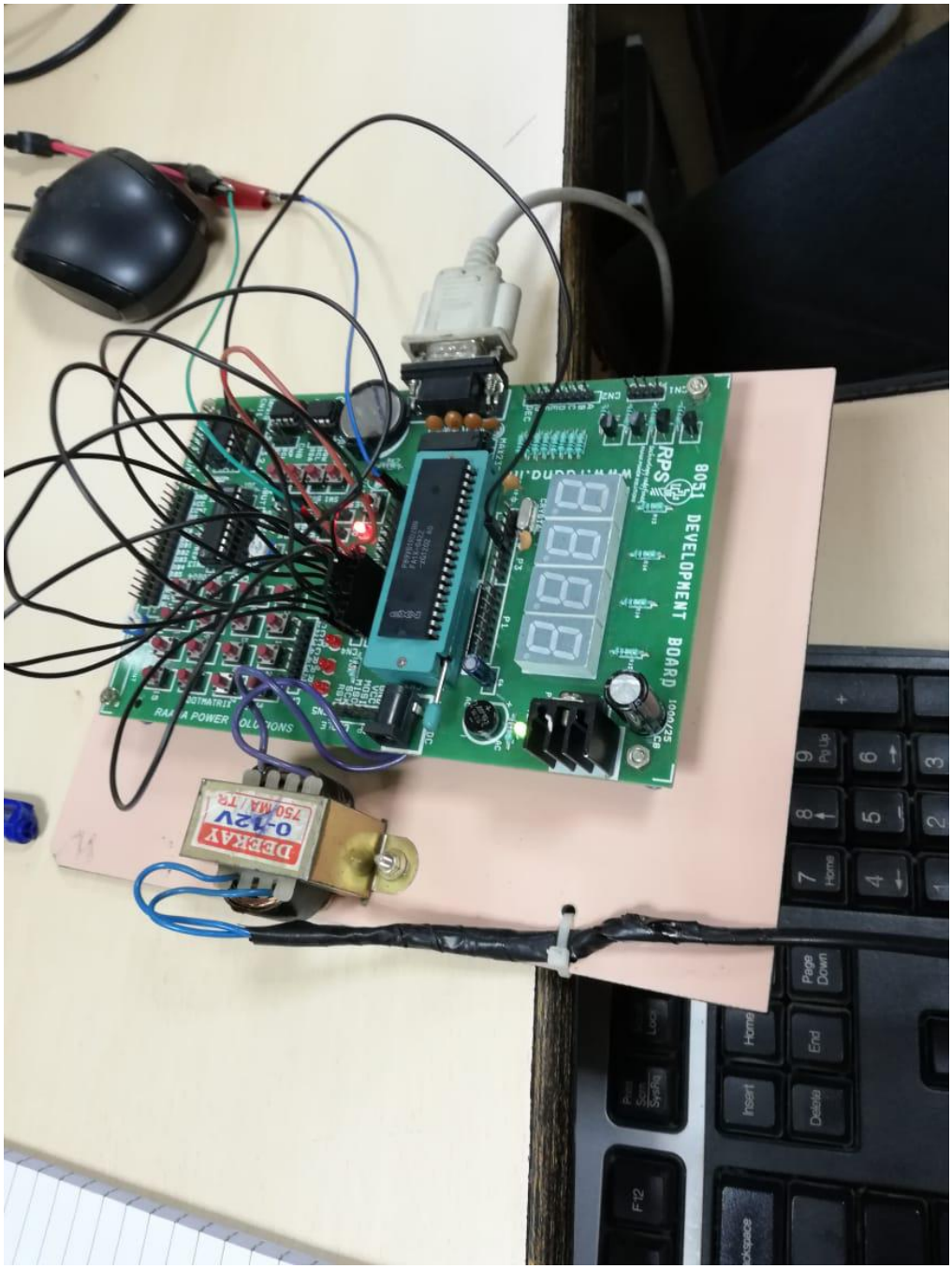
```
org 000h
     ljmp main
org 013h
     ljmp loop
org 0070h
loop: mov a, #0ffh
        mov r2, #2fh
        l2: mov p0,a
        mov r1, #0ffh
        l1:mov r0, #0ffh
        l:djnz r0,l
        djnz r1,l1
        cpl a
        djnz r2,l2
        reti
org 0030h
    main:mov tmod, #01h
           mov ie,#84h
           next:mov th0, #0afh
           mov tl0,#00h
           cpl p2.1
           setb tr0
           l3:jnb tf0,l3
           clr tr0
           clr tf0
           sjmp next
End
```

Pictures




Logic Analyzer
Setup...
Load...
Save...
Min Time
0 s
Max Time
5.772823 s
Grid
10 ms
Zoom
In
Out
All
Min/Max
Auto
Undo
Update Screen
Start
Clear
Transition
Prev
Next
Jump to
Code
Trace
Signal Info
Show Cycles
Amplitude
Cursor
255
P1
0
5.612653 s
5.692652 s
5.772652 s
Disassembly
Logic Analyzer


Step 1 - Communications
Step 2 - Erase
Step 3 - Hex File
Step 4 - Options
Step 5 - Start
25



Conclusion :-The square wave was generated using interrupt programming. Hence the program is verified with the output matching with the desired outcome.