SVKM'S
NMIMS
Deemed-to-be UNIVERSITY

SVKM's NMIMS

Mukesh Patel School of Technology Management and Engineering, Vile Parle, Mumbai- 400056

Programmable Logic Controllers and Data Acquisition Journal

Under the guidance of

Prof. Nirmal Thakur

Department of Mechatronics Engineering

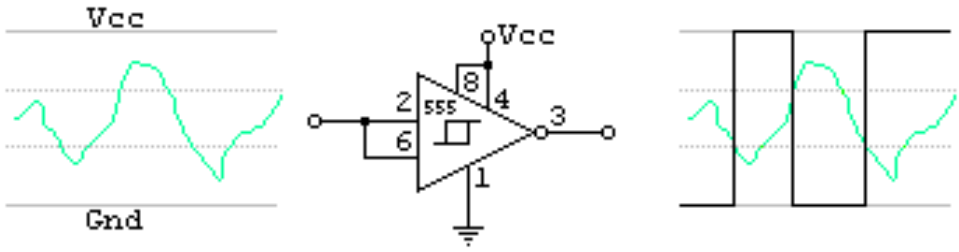
Academic Year:2019-20

Experiment 1

Aim – To implement 1 bit analog to digital converter using Schmitt trigger and to plot the hysteresis characteristics

Apparatus – 555 timer ic, connecting wires, power supply

Circuit diagram



Observation Table

	VH	**VL**
Low Hysteresis	1.6	0.64
High Hysteresis	1.912	0.684

Conclusion

We implemented a 1 bit ADC using Schmitt trigger and saw the hysteresis level of it.

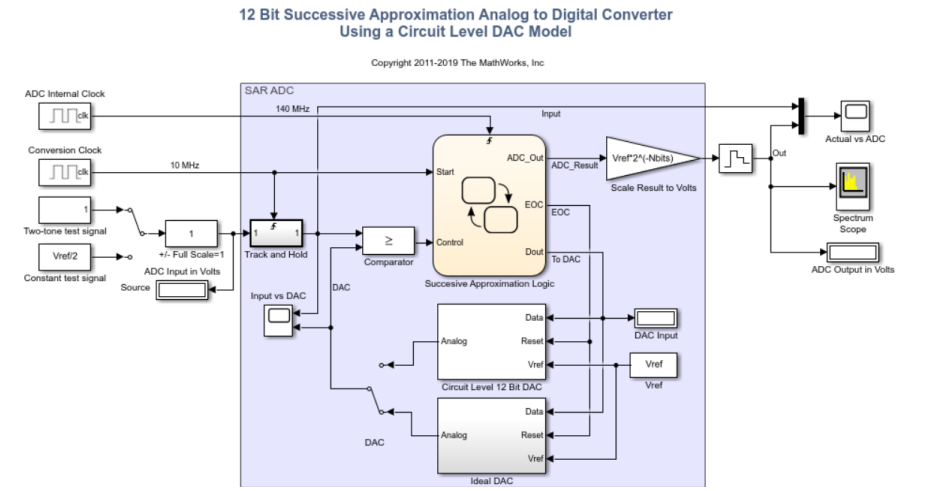
Experiment 2

Aim – To study the implementation of successive approximation ADC in MATLAB Simulink

Apparatus – MATLAB

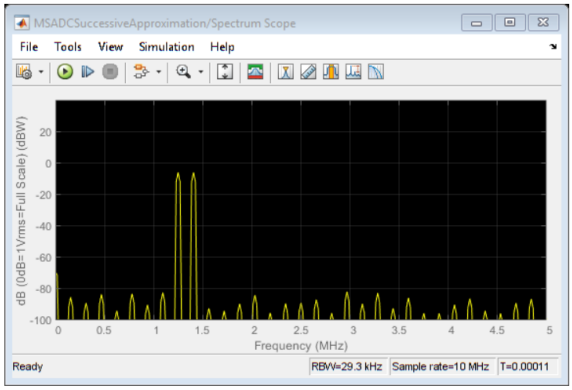
Simulink Diagram

```
model = 'MSADCSuccessiveApproximation';
open_system(model)
```



Observation Table

```
sim(model);
```



Define the number of bits (*NBits*) and ADC conversion rate (*Fs*) in the MATLAB® workspace. The ADC operating clock rate is determined from *Nbits* and *Fs* .

```
Nbits = 12;
Fs = 1e7;
ADC_clock = Fs*(Nbits+2);
```

Conclusion

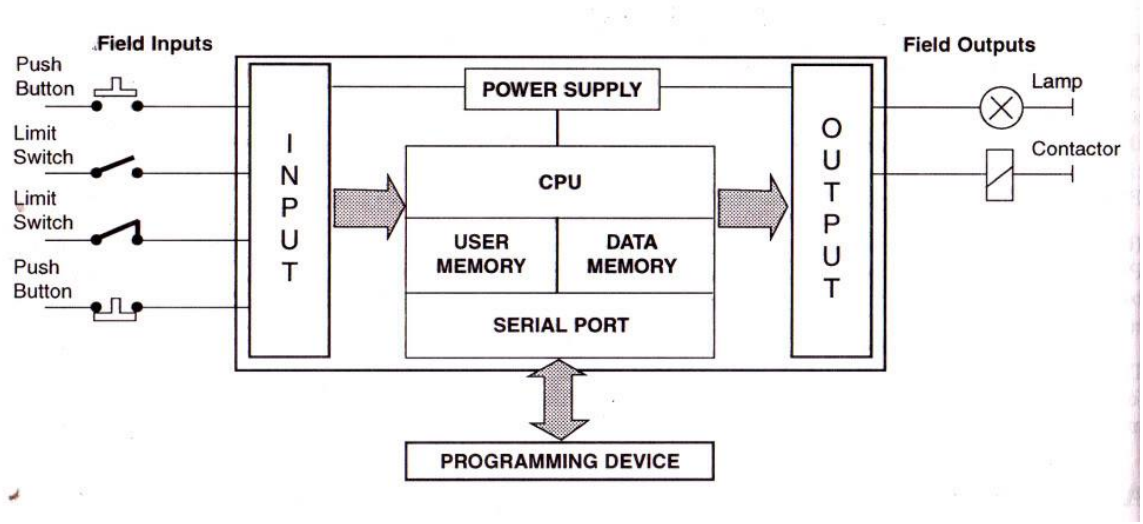
We implemented successive approximation ADC in MATLAB Simulink

Experiment 3

Aim – To demonstrate the hardware configuration of PLC with reference to L20DP PLC

Apparatus – Bosch L20DP

Block Diagram



Configuration

Number of Input – 8(Digital)

Number of Output – 8(Digital)

Network Port – Ethernet, Profibus

Communication Port – RS232

Programming Interface – Computer

Maximum Operating temperature – 55deg

Programming Languages – CFC, FBD, Instruction List, Ladder Logic,SFC,ST

Number of communication port – 1

Memory – 16MB RAM , 64kB Retentive memory

24V DC Output 4-20mA

Conclusion

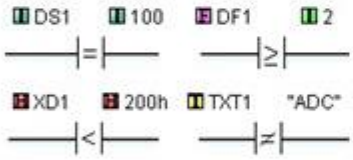
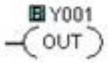
We saw the Hardware configuration with respect to Bosch L20DP

Experiment 4

Aim – To demonstrate features of Bosch Indralogic Programming software and to write PLC ladder logic program for different gates

Apparatus – Bosch L20DP, Bosch Indralogic Software

Relay Logic Instruction set

Ladder Symbol	Title	Type	Description
	Normally Open Contact	Bit Instruction	The Normally Open Contact mimics the behavior of a physical contact and changes in response to the status of a Bit Memory Address. The Normally Open Contact is ON when the related bit is ON.
	Normally Closed Contact	Bit Instruction	The Normally Closed Contact mimics the behavior of a physical contact and changes in response to the status of a Bit Memory Address. The Normally Closed Contact is ON when the related bit is OFF.
	Edge Contact	Bit Instruction	The Edge Contact turns ON when the related bit transitions from OFF to ON (Rising Edge) or ON to OFF (Falling Edge).
	Compare Contact	Word Instruction	The Compare instruction uses a Mathematical Operator as a basis for comparison of two data values. When the data values satisfy the selected mathematical relationship (>, <, =, etc.) the Compare Contact turns ON.
	Out Coil	Bit Instruction	An Out instruction turns ON its associated Bit Memory when the status of the rung is true. The Out instruction turns OFF its associated Bit Memory when the status of the rung is false.
	Set Coil	Bit Instruction	The Set instruction turns ON the associated Bit Memory when the status of the rung is true. The Bit Memory stays on after the rung becomes false.

Ladder Logic Diagram

Observation

Rung 1 is AND Gate, Rung 2 is NOT Gate, Rung 3 is OR Gate, Rung 4 is NAND Gate, Rung 5 is NOR Gate and Rung 6 is XOR Gate

Conclusion

We wrote and implemented PLC ladder logic program for different gates using various contacts and coils

Experiment 5

Aim –

A. Develop PLC Ladder Logic Program for motor ON and OFF control using 2 push buttons, START and STOP , with thermal overload protection with indication LEDs

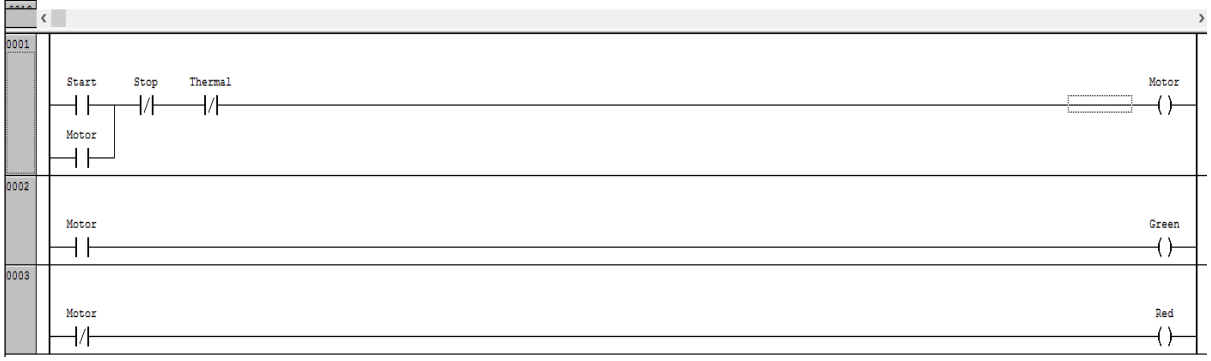
B. Develop PLC Ladder Logic Program to implement 4:1 multiplexer

Apparatus – Bosch L20DP, Bosch Indralogic Software

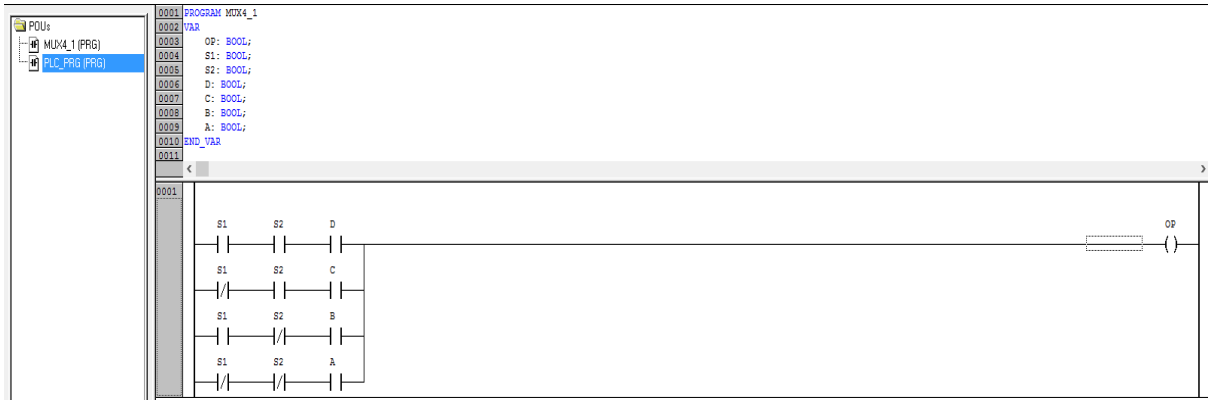
Ladder Logic Diagram

A.

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     Start: BOOL;
0004     Stop: BOOL;
0005     Thermal: BOOL;
0006     Motor: BOOL;
0007     Green: BOOL;
0008     Red: BOOL;
0009 END_VAR
0010
0011
0012
```



B.



Observation

A. When the Start Button is pushed the motor is ON and the green light turns on. When the Stop button is pushed the motor turns OFF and the red light turns on. While the motor is ON and thermal overload sensor is triggered motor turns OFF and Red light turns on.
B. S1 and S2 are the select lines and A,B,C and D are the input to the multiplexer. Based on the logic of S1 and S2 one of the input is given as an output

Conclusion

We learned and implemented the Ladder Logic Diagram Programs using various Contacts and Coils

Experiment 6

Aim –

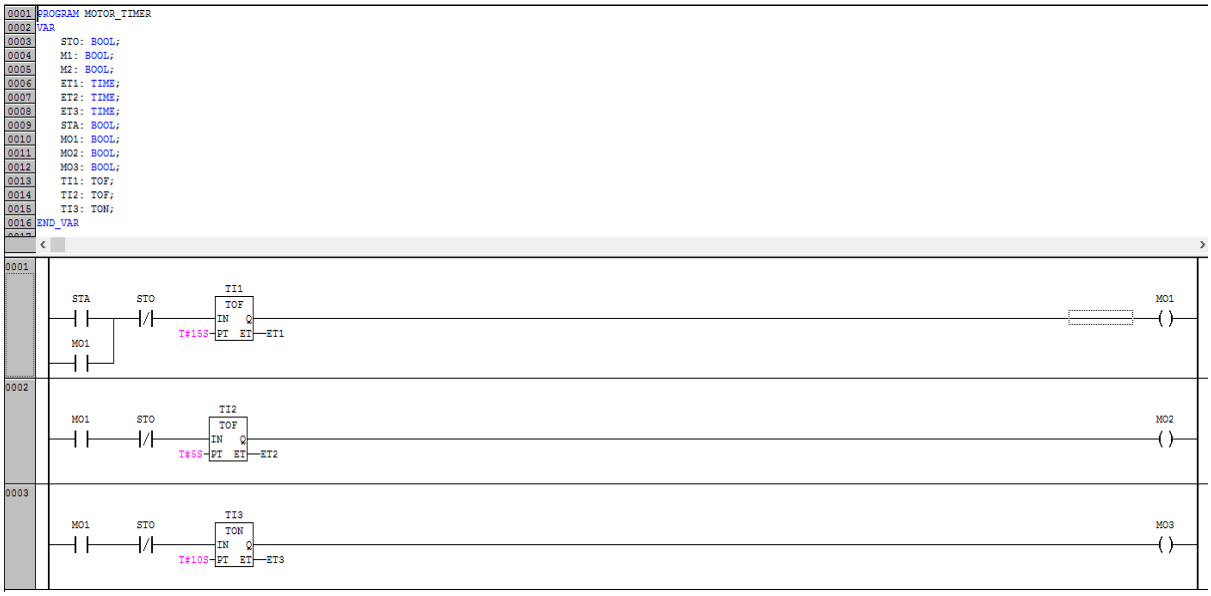
A. Develop a PLC Ladder Logic Program for 3 Motor system. When Start is pressed M1 and M2 starts immediately while M3 Starts after 10s. When stop button is pressed M3 stops immediately after 5s M2 stops and 10s after M2, M1 stops.
B. Develop a PLC Ladder Logic Program for generating a square wave with time duration of 40s

Apparatus – Bosch L20DP, Bosch Indralogic Software

Ladder Logic Diagram

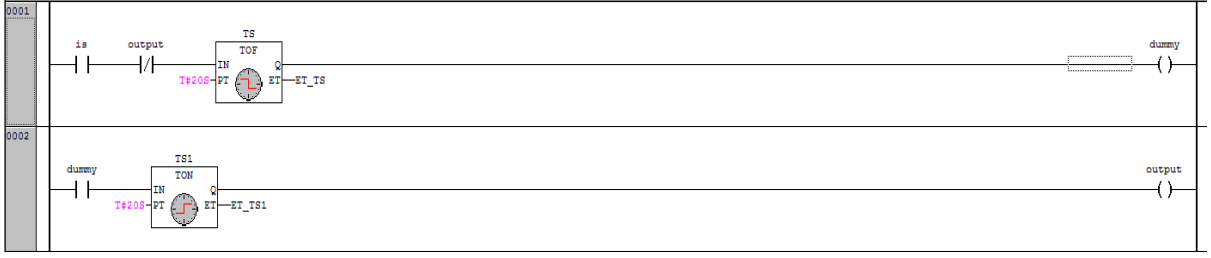
A.

```
0001 PROGRAM MOTOR_TIMER
0002 VAR
0003     STO: BOOL;
0004     M1: BOOL;
0005     M2: BOOL;
0006     ET1: TIME;
0007     ET2: TIME;
0008     ET3: TIME;
0009     STA: BOOL;
0010     MO1: BOOL;
0011     MO2: BOOL;
0012     MO3: BOOL;
0013     TI1: TOF;
0014     TI2: TOF;
0015     TI3: TON;
0016 END_VAR
```



B.

```
0001 PROGRAM MOTOR_TIMER
0002 VAR
0003     start: BOOL;
0004     stop: BOOL;
0005     t1: TP;
0006     et_1: TIME;
0007     op: BOOL;
0008     is: BOOL;
0009     TS: TOF;
0010     ET_TS: TIME;
0011     dummy: BOOL;
0012     TS1: TON;
0013     ET_TS1: TIME;
0014     output: BOOL;
0015 END_VAR
0016
```



Observation

A. When Start is pressed M1 and M2 starts immediately while M3 Starts after 10s. When stop button is pressed M3 stops immediately after 5s M2 stops and 10s after M2, M1 stops.
B. We get a square wave of time duration of 40s at the output coil

Conclusion

We learned and implemented the Ladder Logic Diagram Programs using Timer Blocks .

Experiment 7

Aim –

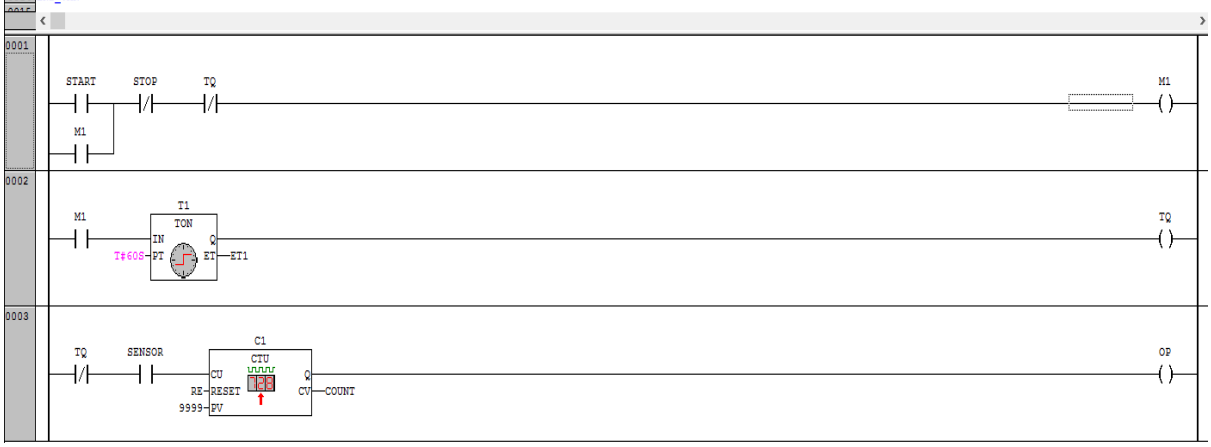
A. To develop PLC Ladder Logic Program for counting number of objects passing on a conveyer belt in time duration of 1min
B. To develop PLC Ladder Logic Program for a box Packaging system having following conditions
   i. 5 boxes to be stacked and bound by a wrapping bot
   ii. Wrapping bot takes 30s to complete operation
   iii. Cycle should repeat continuously until Stop switch is pressed

Apparatus – Bosch L20DP, Bosch Indralogic Software

Ladder Logic Diagram

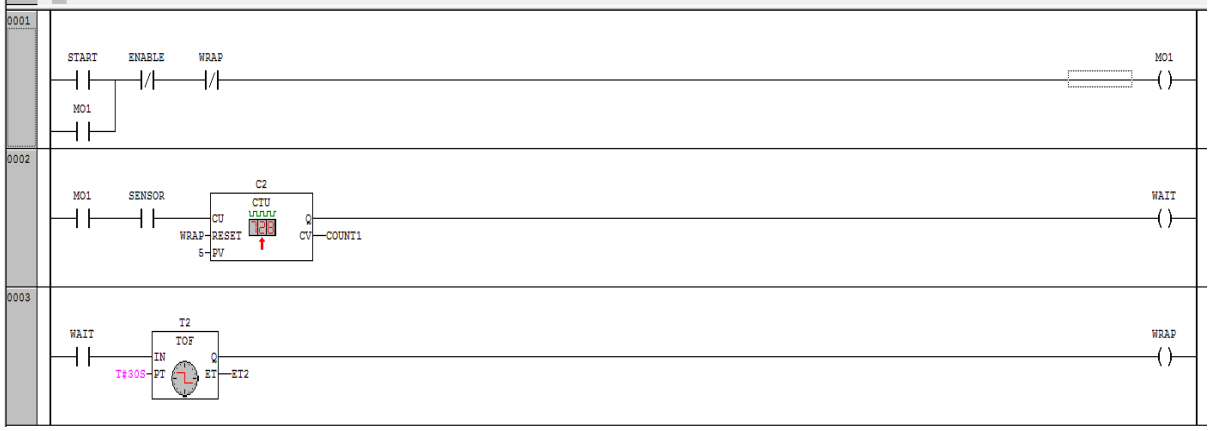
A.

```
0001 PROGRAM exp7
0002 VAR
0003     START: BOOL;
0004     STOP: BOOL;
0005     SENSOR: BOOL;
0006     C1: CTU;
0007     COUNT: INT;
0008     RE: BOOL;
0009     M1: BOOL;
0010     T1: TON;
0011     ET1: TIME;
0012     TQ: BOOL;
0013     OP: BOOL;
0014 END_VAR
```



B.

```
0001 PROGRAM EXP7B
0002 VAR
0003     START: BOOL;
0004     ENABLE: BOOL;
0005     SENSOR: BOOL;
0006     C2: CTU;
0007     COUNT1: INT;
0008     WAIT: BOOL;
0009     T2: TOF;
0010     ET2: TIME;
0011     WRAP: BOOL;
0012     OP: BOOL;
0013     M01: BOOL;
0014 END_VAR
```



Observation

A. The program counted number of objects passing on a conveyer belt in time duration of 1min. The number of objects counted are in C1 Counter and T1 runs the system for 60s
B. Start button turns on the system and the coveyer betlt M1 starts and Sensor senses objects and gives output to counter c1. As soon as the counter reaches 5. The conveyer belt stops and wrapping bot is on for 60s.

Conclusion

We learned and implemented the Ladder Logic Diagram Programs using Timer and Counter Blocks .

Experiment 8

Aim – To write assembly Language Program for generating a square Wave of time duration 2s on pin number 0 of port 1 of 8051 microcontroller

Apparatus – 8051 Microcontroller and microvision software

Program

```
START:
setb p1.0
        MOV R2, # 0AH
LOOP1:  MOV R1, # 0C8H
LOOP:   MOV R0, # 0FFH
WAIT:   DJNZ R0,WAIT
        DJNZ R1, LOOP
        DJNZ R2, LOOP1

clr p1.0
        MOV R2, # 0AH
LOOP3:  MOV R1, # 0C8H
LOOP2:  MOV R0, # 0FFH
WAIT1:  DJNZ R0,WAIT1
        DJNZ R1, LOOP2
        DJNZ R2, LOOP3
SJMP    START
END
```

Observation

We saw the LED light connected to the pin 0 blink on and off for the interval of 2s

Conclusion

We learned and wrote the assembly language code for generating a square wave.

Experiment 9

Aim – To interface DAC 0808 with 8051 microcontroller and to write assembly language program to generate triangular wave

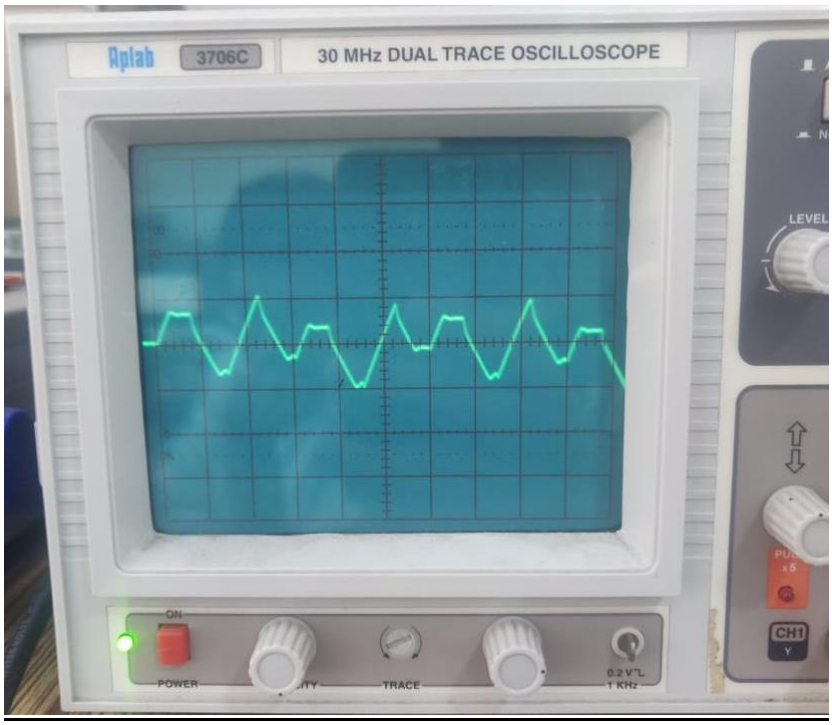
Apparatus – 8051 Microcontroller and microvision software

Program

dac.asm | STARTUP.A51

```
1  ORG 0000H
2  LJMP START
3  ORG 0100H
4  START:
5  MOV A,#00H
6  HERE:
7  MOV P1,A
8  INC A
9  CJNE A,#0FFH,HERE
10 NXT:
11 DEC A
12 MOV P1,A
13 CJNE A,#00H,NXT
14 SJMP START
15 END
```

Observation



Conclusion

We learned and wrote the assembly language code for generating a triangular wave.