**Name: Nimra Nasir**

**Reg No: 19-CP-35**

**Submitted to: Sir Sharoon Saleem**

**Microprocessor & Interfacing Assignment**

**Question:**

**Implementation of ADC or DAC interfaces with processor on proteus and its program.**

1. **Implementation of ADC interfaces with processor on proteus and its program.**

**Analog to Digital Conversion (ADC):**

ADC is a device that converts Analog signal to digital signal. It is used in different application.

**Conversion Formula:**

**Step Size = Vref/2^n**

Where **n** is the number of bits.

**Dout = Vin/Step Size**

Where **Vin** is input applied signal.

**Source Code:**

DATA SEGMENT

PORTA EQU 00H

PORTB EQU 02H

PORTC EQU 04H

PORT\_CON EQU 06H

DATA ENDS

CODE SEGMENT

MOV AX,DATA

MOV DS,AX

ORG 0000H

START:

MOV DX, PORT\_CON

MOV AL, 10010000B

OUT DX, AL

MOV AL,00H

XX:

MOV DX, PORTA

IN AL, DX

MOV DX, PORTC

OUT DX, AL

MOV DX, PORTB

MOV AL, 00000000B

OUT DX, AL

MOV CX, 0ffh

D1: LOOP D1

MOV DX, PORTB

MOV AL, 00000001B

OUT DX, AL

MOV CX, 0ffh

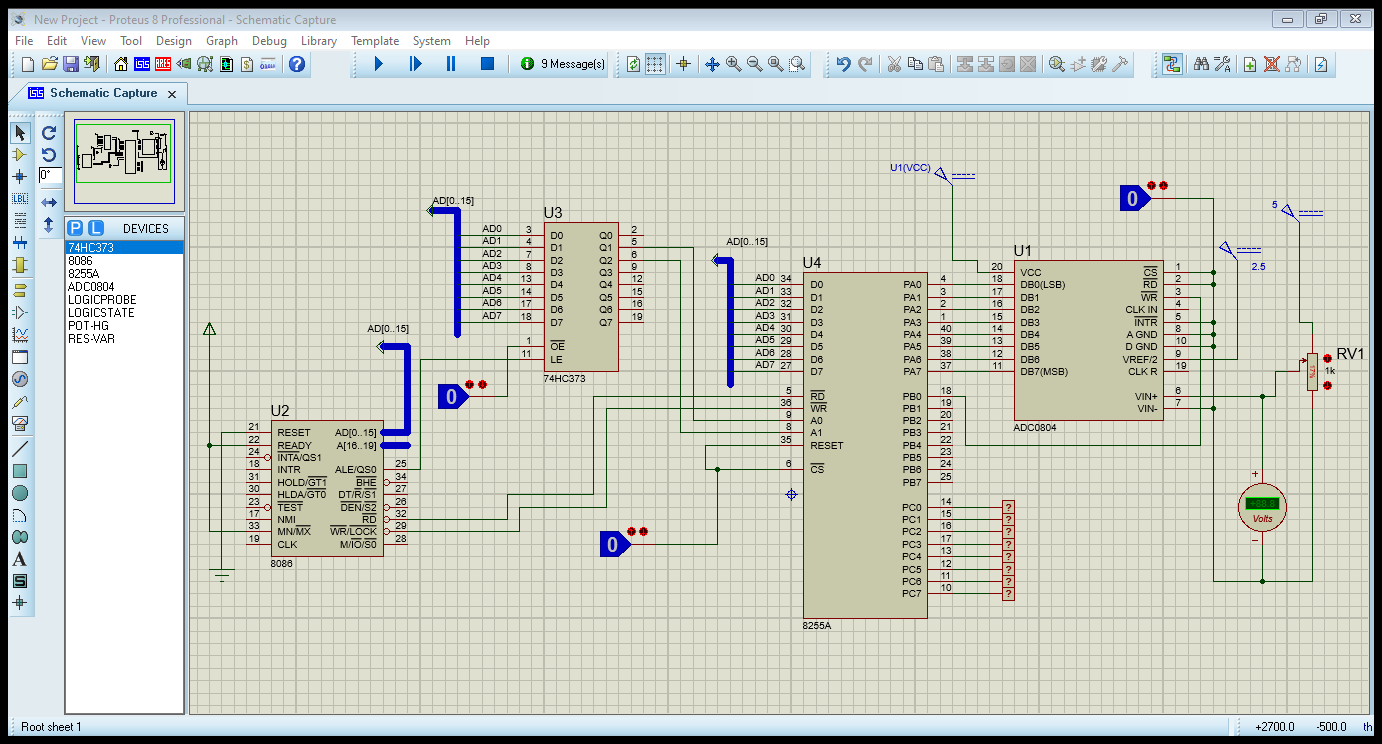
D2: LOOP D2

JMP XX

CODE ENDS

END

ret

****

**Output:**

When applied voltage **=** Vin= +3.00V;

Vref = +5.00V

By using conversion formula;

Step Size = Vref/2^n

Step Size = 5/2^8 (Where 8 is the number of bits)

Step Size = 5/256 = 0.0195V

Dout = Vin/Step Size

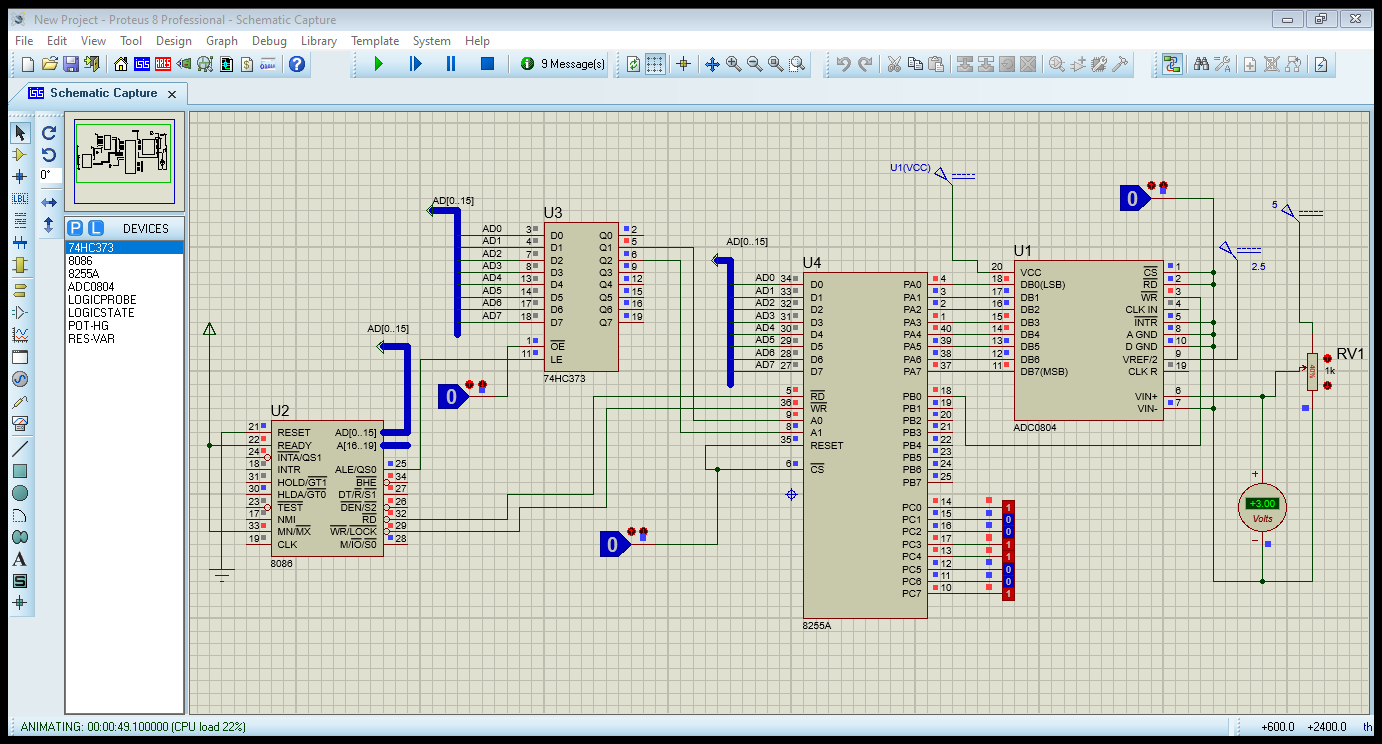
Dout = +3.00/0.0195V

Dout = 153

Now

When we convert 153 into binary

Decimal (153) = Binary (10011001)

****

1. **Implementation of DAC interfaces with processor on proteus and its program.**

A digital-to-analog converter (DAC), as the name implies, is a data converter which generates an analog output from a digital input. A DAC converts a limited number of discrete digital codes to a corresponding number of discrete analog output values.

**Source Code:**

CODE SEGMENT

PORTA EQU 00H

Config EQU 06H

ORG 100H

MOV DX, Config

MOV AL, 10000000B

OUT DX,AL

START:

MOV AL, 00000000B

MOV DX, PORTA

OUT DX, AL

MOV CX,0ffh

loopy1:

loop loopy1

MOV AL, 11111111B

MOV DX, PORTA

OUT DX, AL

MOV CX,0ffh

loopy2:

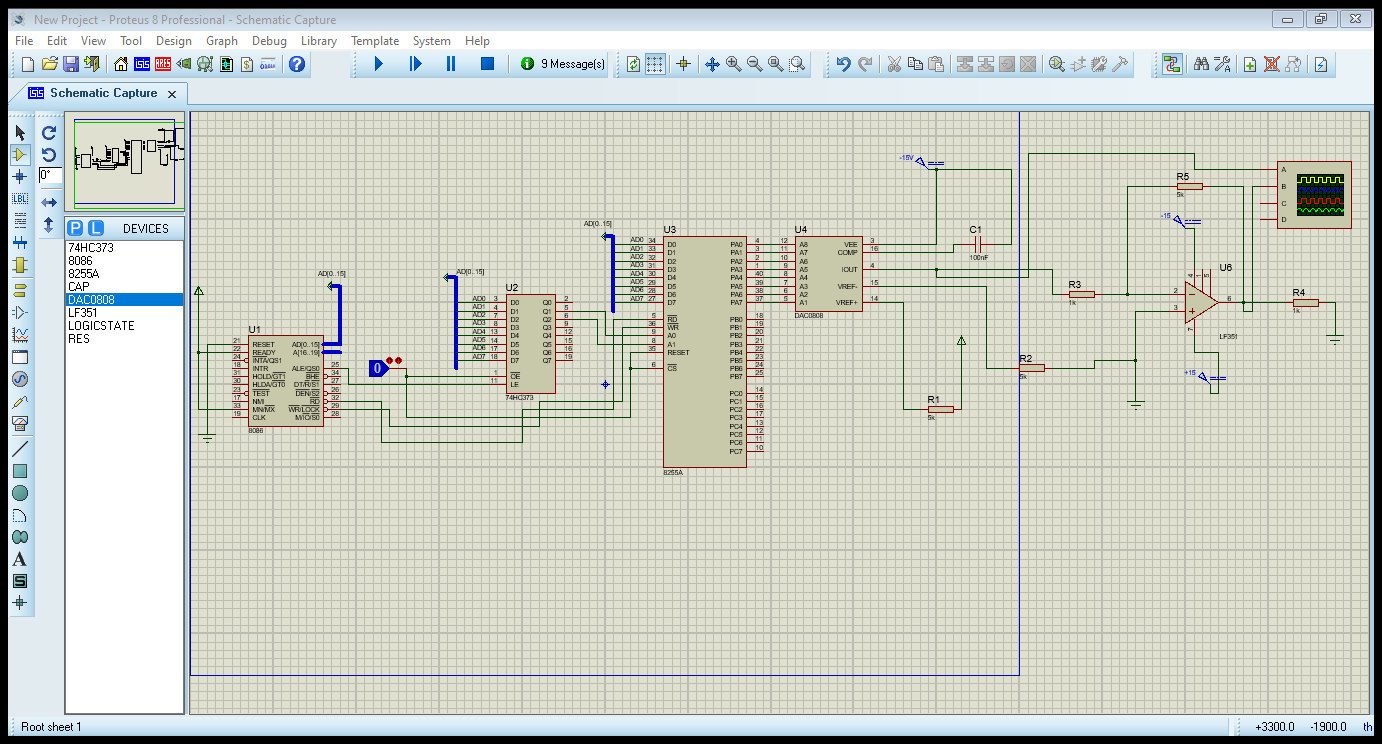
loop loopy2

JMP START

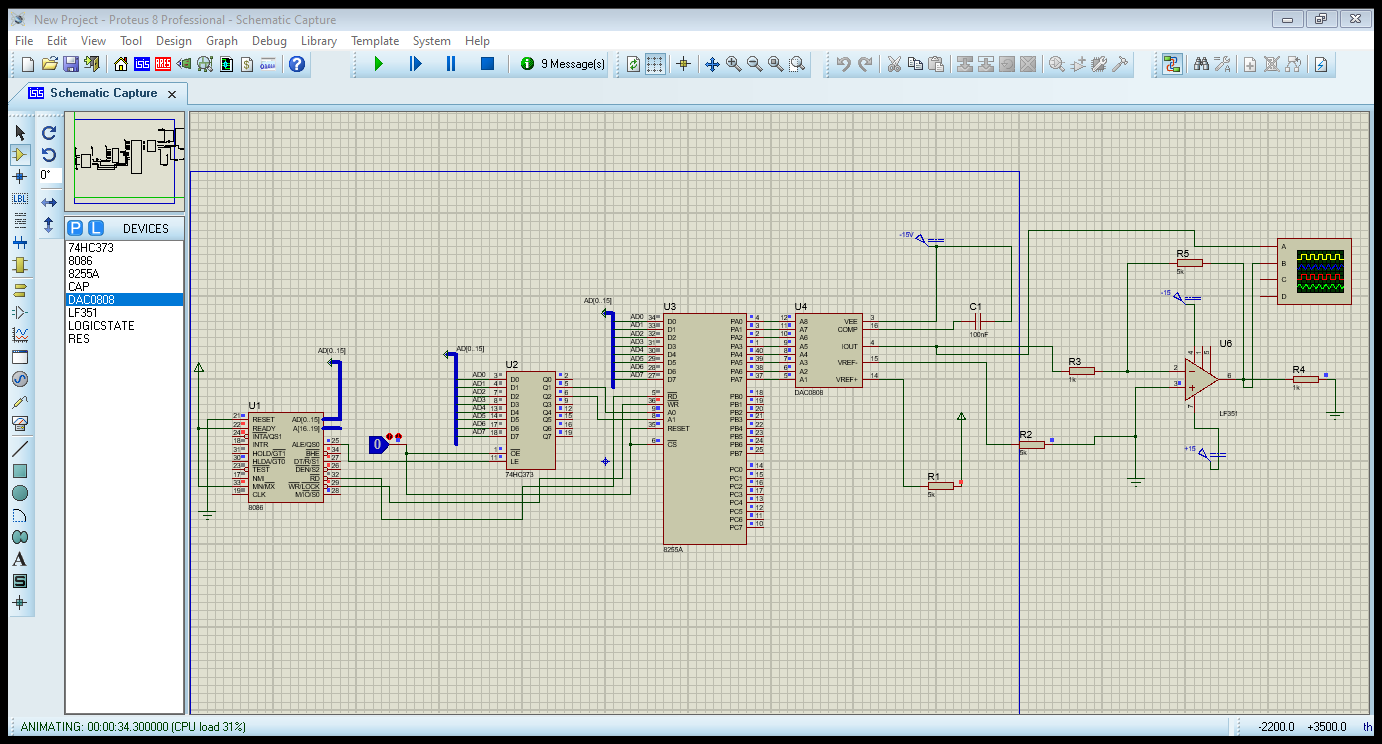
CODE ENDS

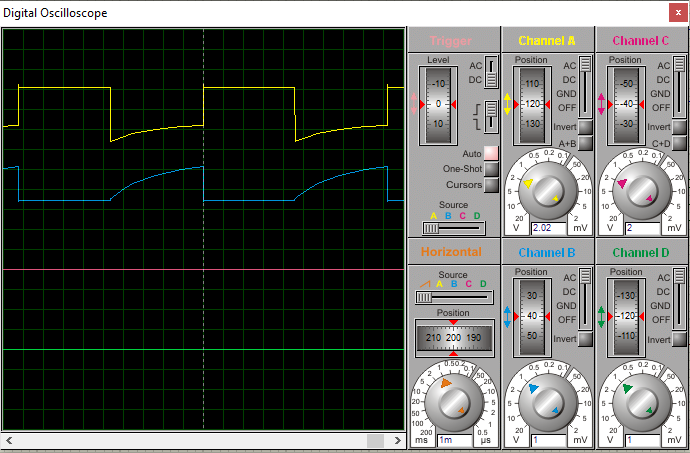
END

ret



**Output:**





**------------------------------------------------THE END-------------------------------------------------**