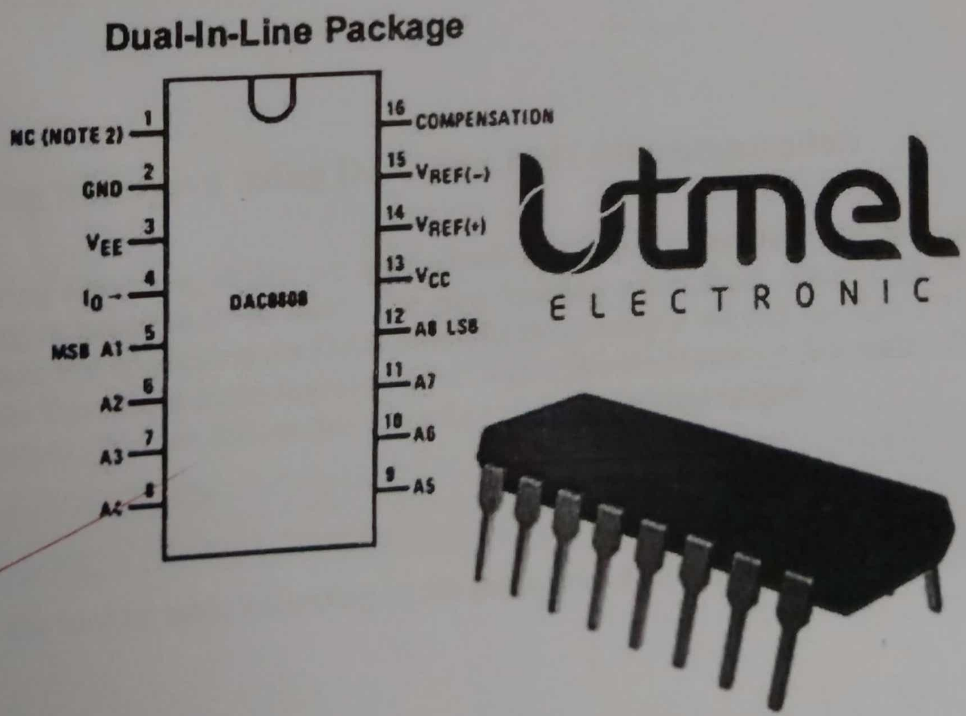
Practical No : 7

Aim: Interfacing DAC0808 with 8051 microcontroller.

Theory:

The Digital to Analog converter (DAC) is a device, that is widely used for converting digital pulses to analog signals. There are two methods of converting digital signals to analog signals. These two methods are binary weighted method and R/2R ladder method. In this article we will use the MC1408 (DAC0808) Digital to Analog Converter. This chip uses R/2R ladder method. This method can achieve a much higher degree of precision. DACs are judged by its resolution. The resolution is a function of the number of binary inputs. The most common input counts are 8, 10, 12 etc. Number of data inputs decides the resolution of DAC. So if there are n digital input pin, there are 2^n analog levels.



v

DAC0808 Pinout

DAC0808 Pin Description:

pin Number	Pin Name	Description
1	NC	No connection
2	GND	Ground
3	VEE	Negative power supply
4	IO	Output signal pin
5	A1	Digital input bit 1 (Most Significant Bit)
6	A2	Digital input bit 2
7	A3	Digital input bit 3
8	A4	Digital input bit 4
9	A5	Digital input bit 5
10	A6	Digital input bit 6
11	A7	Digital input bit 7
12	A8	Digital input bit 8 (Least Significant Bit)
13	VCC	Positive power supply
14	VREF+	Positive reference voltage
15	VREF-	Negative reference voltage
16	COMPENSATION	Compensation **capacitor** pin

Generating Sinewave using DAC and 8051 Microcontroller:

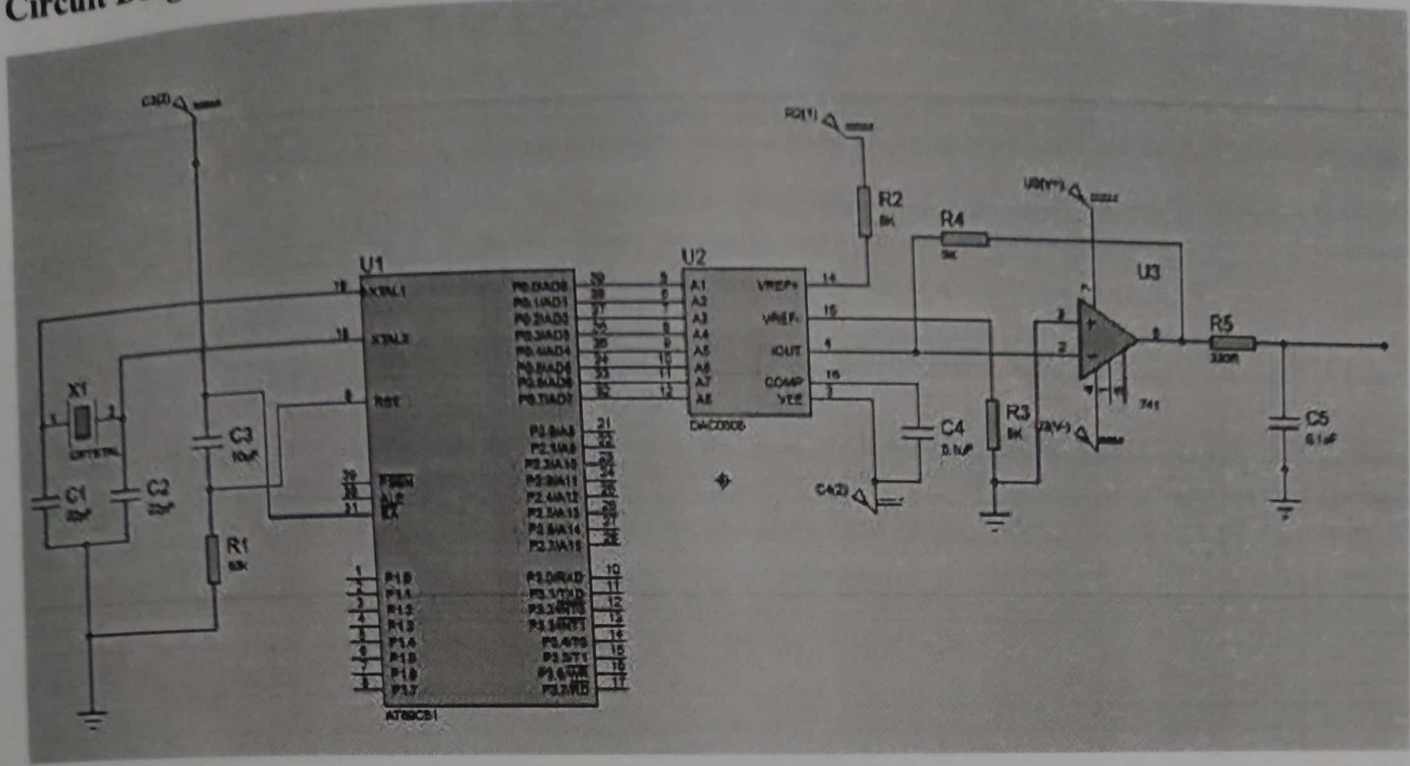
For generating sinewave, at first we need a look-up table to represent the magnitude of the sine value of angles between 0° to 360°. The sine function varies from -1 to +1. In the table only integer values are applicable for DAC input. In this example we will consider 30° increments and calculate the values from degree to DAC input. We are assuming full-scale voltage of 10V for DAC output. We can follow this formula to get the voltage ranges.

$V_{out} = 5V + (5 \times \sin\theta)$

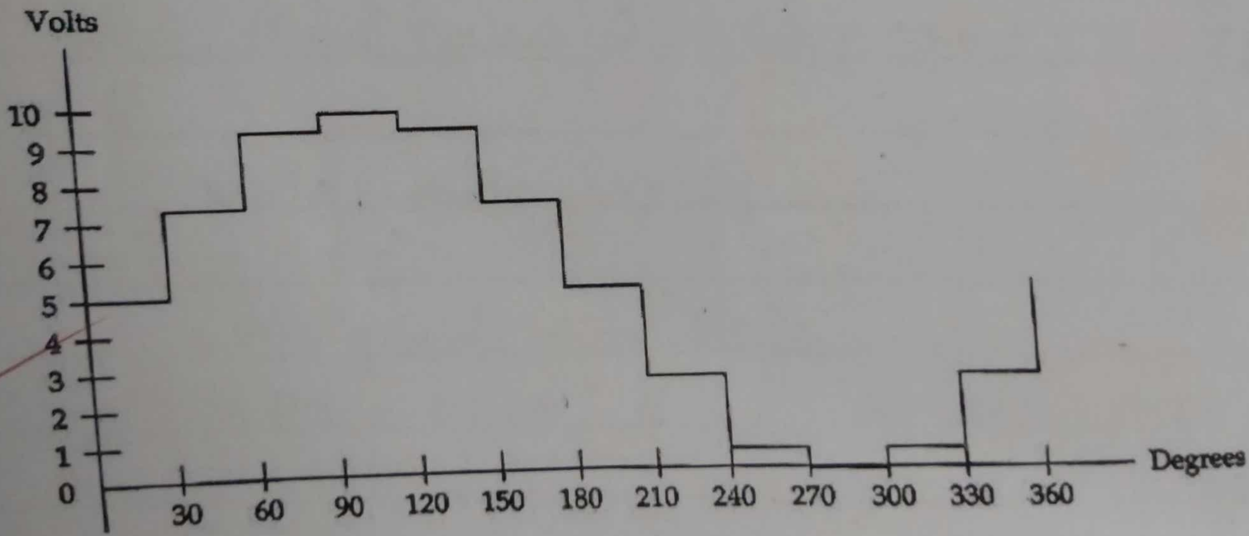
Let us see the lookup table according to the angle and other parameters for DAC.

Angle(in θ)	sinθ	V_{out} (Voltage Magnitude)	Values sent to DAC
0	0	5	128
30	0.5	7.5	192
60	0.866	9.33	238
90	1.0	10	255
120	0.866	9.33	238
150	0.5	7.5	192
180	0	5	128
210	-0.5	2.5	64
240	-0.866	0.669	17
270	-1.0	0	0
300	-0.866	0.669	17
330	-0.5	2.5	64
360	0	5	128

Circuit Diagram –



The output will look like this –



Rajdhani
DATE / /

Experiment - 7

Name - Amey Sanjay Vanave
Roll no. - 20151206
Branch - ETX
Batch - EX-I

Aim : Interfacing DAC0808 with 8051 microcontroller.

Program :

1)

```
#include <reg51.h>
#include <math.h>
unsigned char val[] = {0X80, 0X86, 0X8C,
                       0X93 . . . 0X7A};

// void delay (unsigned int itime)
// {
// int i,j;
// for (i=0 ; i <= itime ; i++)
//    for (j=0 ; j <= 50 ; j++);
// }
void main (void)
{
  unsigned char i;
    P1 = 0X00; // P1 As o/p port.
    while (1)
  {
    P1 = val[i];
    i++;
    if (i >= 112)
    i = 0;
  // delay (1);
  }
}
```

DATE

2)

```
# include <reg51.h>
unsigned int i=0;
void delay (unsigned int itime);
void main (void)
{
    P1 = 0x00;          // P1 As o/p port
    {
      while (1)
        {
          P1 = 0xFF;
          delay (5);
          P1 =0x00
          delay (5);
        }
    }
}
void delay (unsigned int itime)
{
  int i,j;
  for  (i=0; i <= itime; i++)
  for  (j=0; j <= 100; j++);
}
```

Rajdhani
DATE / /

3)

```
# include <reg51.h>
unsigned int i=0;
void delay (unsigned int itime);
void main (void)
{
    unsigned char i, j;
        P1 = 0x00          // P1 as output port
    while (1)
    {
      for (i = 0x00); i <= 0x00 ; i++)
     {
  //    P1 = i ;
           delay (5);
     }
      for (j = 0xF0 ; j >= 0x01 ; j--)
      {
          P1 = j ;
  //   delay (1);
      }
    }
}
   void delay (unsigned int itime)
  {
     int i, j ;
     for (i = 0 ; i <= itime ; j++)
     for (j = 0 ; j <= 5 ; j++);
  }
```

Rajdha

DATE /

- Conclusion :

~~Do~~ Interfacing of DAC 0808 with Microcontroller, using C programming.

C

m 13/5/22