Group - 11

Soumava Paul 16EE10056
Yerramsetty Rohit 16EE10055

Experiment 7

Tuneable Signal Generator

Aim:

a. Interface a DAC 0808 with Arduino and a resistive load
b. Generate an analog value transmitted from PC via serial monitor
c. Generate a sinusoidal wave at a given frequency transmitted from PC via serial monitor
d. Control the amplitude of the sinusoidal wave using a variable resistor
e. Control the frequency of the sinusoidal source using a variable resistor

Apparatus Required:

1. Arduino Uno Board
2. Resistors
3. DAC 0808
4. Op-amp lmp358
5. Capacitors
6. Breadboard
7. Jumper Wires

Theory:

The digital-to-analog converter (DAC) is a device widely used to convert digital pulses to analog signals. The vast majority of integrated circuit DACs, including the MC1408 (DAC0808) used in this section, use the R/2R method since it can achieve a much higher degree of precision. The first criterion for judging a DAC is its resolution, which is a function of the number of binary inputs. The common ones are 8, 10, and 12 bits. The number of data bit inputs decides the resolution of the DAC since the number of analog output levels is equal to 2″, where *n* is the number of data bit inputs. Therefore, an 8-input DAC such as the DAC0808 provides 256 discrete voltage (or current) levels of output. Similarly, the 12-bit DAC provides 4096 discrete voltage levels.

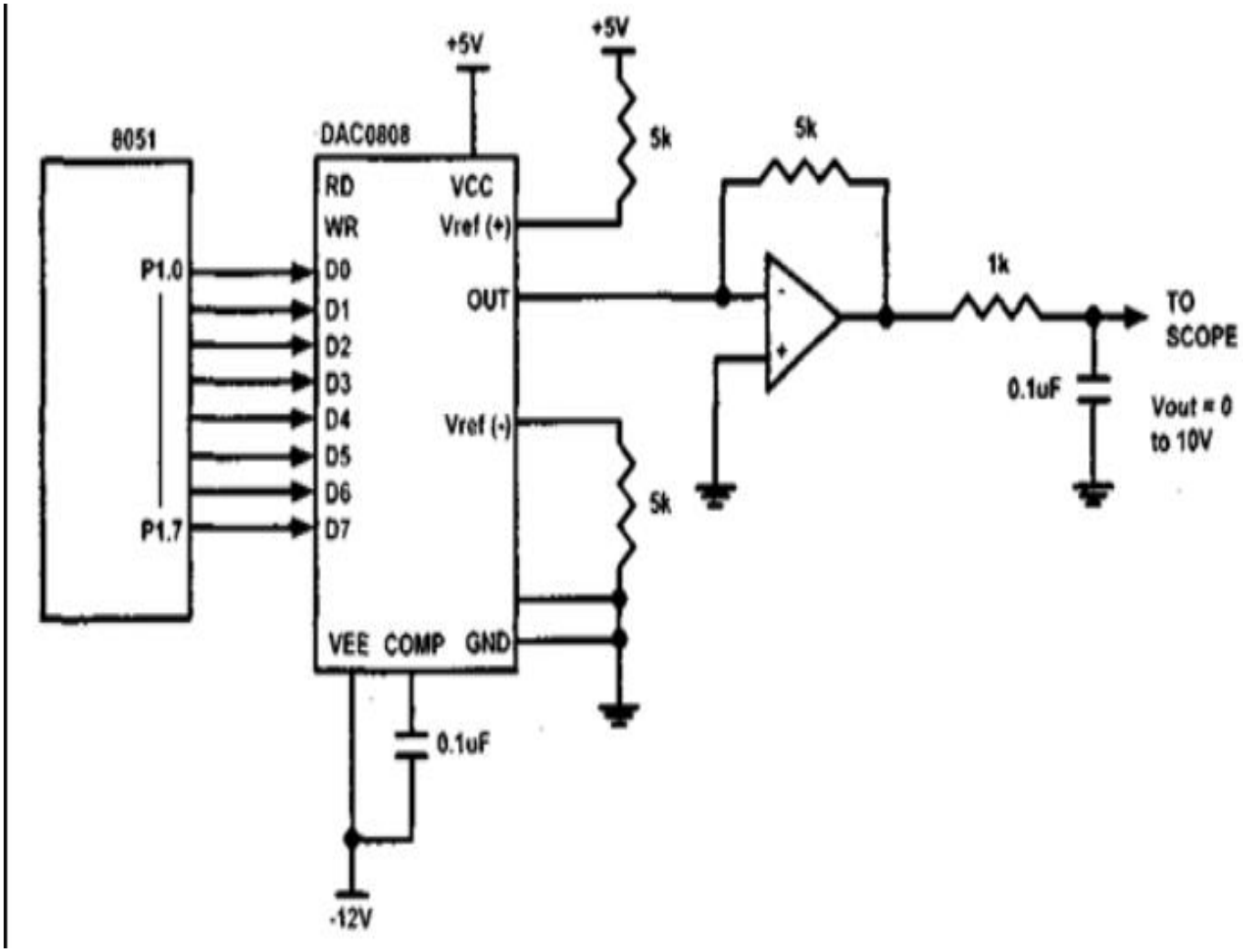
MC1408 DAC (or DAC0808)

In the MC1408 (DAC0808), the digital inputs are converted to current (I_{out}), and by connecting a resistor to the I_{out} pin, we convert the result to voltage. The total current provided by the I_{out} pin is a function of the binary numbers at the DO – D7 inputs of the DAC0808 and the reference current (I_{ref}), and is as follows:

$$I_{out} = I_{ref} \left(\frac{D7}{2} + \frac{D6}{4} + \frac{D5}{8} + \frac{D4}{16} + \frac{D3}{32} + \frac{D2}{64} + \frac{D1}{128} + \frac{D0}{256} \right)$$

where DO is the LSB, D7 is the MSB for the inputs, and I_{ref} is the input current that must be applied to pin 14. The I_{ref} current is generally set to 2.0 mA. Some DACs also use the zener diode (LM336), which overcomes any fluctuation associated

Circuit Diagram:



Codes:

7a

7a

```
int bit0 = 2;
int bit1 = 3;
int bit2 = 4;
int bit3 = 5;
int bit4 = 6;
int bit5 = 7;
int bit6 = 8;
int bit7 = 9;
int output = A0;
void setup() {
pinMode(bit0,OUTPUT);
pinMode(bit1,OUTPUT);
pinMode(bit2,OUTPUT);
pinMode(bit3,OUTPUT);
pinMode(bit4,OUTPUT);
pinMode(bit5,OUTPUT);
pinMode(bit6,OUTPUT);
pinMode(bit7,OUTPUT);
pinMode(A0,INPUT);
Serial.begin(9600);
}
void loop() {
digitalWrite(bit0,HIGH);
digitalWrite(bit1,HIGH);
digitalWrite(bit2,HIGH);
digitalWrite(bit3,HIGH);
digitalWrite(bit4,HIGH);
digitalWrite(bit5,HIGH);
digitalWrite(bit6,HIGH);
digitalWrite(bit7,HIGH);
analogRead(A0);
Serial.println(A0);
delay(1000);
```

7c

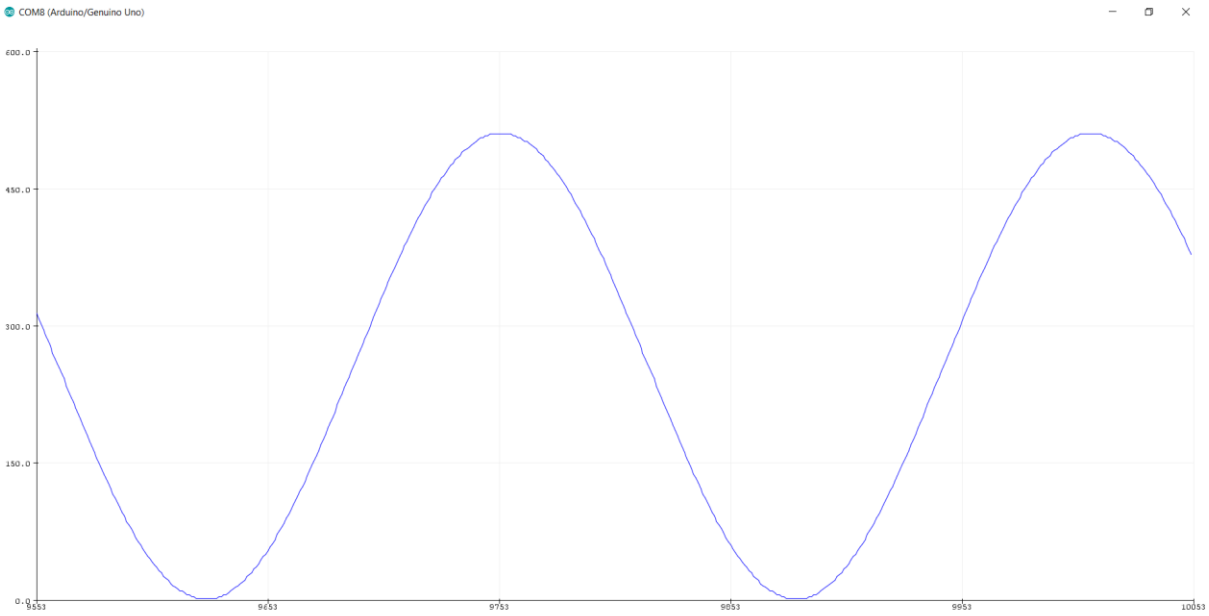
```
int sine [255];
const int POT = A0;
int value = 0;
int freq = 0;
int amp = 0;
int data1 = 0;
int data2 = 0;
```

```
int data3 = 0;
int frequency;
int freqCurrent;
unsigned int freqscaled;
void setup () {
pinMode ( 2 , OUTPUT );
pinMode ( 3 , OUTPUT );
pinMode ( 4 , OUTPUT );
pinMode ( 5 , OUTPUT );
pinMode ( 6 , OUTPUT );
pinMode ( 7 , OUTPUT );
pinMode ( 8 , OUTPUT );
pinMode ( 9 , OUTPUT );
Serial . begin ( 9600 ); // Initialize variables
frequency = analogRead ( A4 ); // initialize frequency
// A4 gets the value from DAC output.
freqscaled = 48 * frequency + 1 ; // from 1 to ~50,000\
period = samplerate / freqscaled;
delay ( 3000 ); // So we can see the nice splash screen
// Generate the values of a sine function float
float x , y;
for ( int i = 0 ; i < 255 ; i ++){
x =( float ) i;
y = sin (( x / 255 )* 2 * PI );
sine [ i ]=( int ( y * 128 )+ 128 );
}
}
/*
* @Description : This function generate a sine signal
* @input : freq
```

```
*/
void Sine_Function ( int freq ){
for ( int i = 0 ; i < 255 ; i ++){
PORTD = sine [ i ];
amp = analogRead ( A0 ); // A pot is used to give a value between 0 and 5 V to pin
amp = amp / 255.0;
Serial . println ( amp * sine [ i ]); // amplitude control using variable register.
delay(10);
delayMicroseconds ( freq ); }
}
/*
* @Description : This function check the value of the input Analog 4 ( A4 ),
* which configure the frequency of the signal.
* This value will be displayed by the display
*/
void checkFreq () {
freqCurrent = analogRead ( A4 );
}
void loop () {
digitalWrite ( 2 , LOW );
digitalWrite ( 3 , LOW );
digitalWrite ( 4 , LOW );
digitalWrite ( 5 , LOW );
digitalWrite ( 6 , LOW );
digitalWrite ( 7 , LOW );
digitalWrite ( 8 , LOW );
digitalWrite ( 9 , HIGH ); // setting up the DAC output to set up the frequency..
value = analogRead ( A4 );
freq = value * 10;
checkFreq ();
```

```
Sine_Function ( freq );
}
```

Magnitude Controlled Sine Wave Generation



Discussions

The DAC 0808 was checked and tested for various inputs and corresponding output was checked by giving various combinations of the 8-bit input. A sine function wave was generated using the arduino.