Experiment 09: Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.

```
#include <LPC21xx.H>
 void clock_wise(void);
 void anti_clock_wise(void);
  unsigned long int var1,var2;
 unsigned int i=0,j=0,k=0;

 int main(void)
 {
        PINSEL0 = 0x00FFFFFF;              //P0.12 to P0.15 GPIo
        IO0DIR |= 0x0000F000;              //P0.12 to P0.15 output
        while(1)
        {
              for(j=0;j<50;j++)            // 20 times in Clock wise Rotation
              clock_wise();

              for(k=0;k<65000;k++);   // Delay to show  anti_clock Rotation

              for(j=0;j<50;j++)       // 20 times in  Anti Clock wise Rotation
              anti_clock_wise();

              for(k=0;k<65000;k++);   // Delay to show clock Rotation

        }                                    // End of while(1)

 }                                          // End of main

 void clock_wise(void)
 {
        var1 = 0x00000800;                   //For Clockwise
        for(i=0;i<=3;i++)                    // for A B C D Stepping
        {
           var1 = var1<<1;                   //For Clockwise
           var2 = ~var1;
            var2 = var2 & 0x0000F000;
            IO0PIN = ~var2;
          for(k=0;k<3000;k++);               //for step speed variation
    }

 }

void anti_clock_wise(void)
 {
        var1 = 0x00010000;                   //For Anticlockwise
       for(i=0;i<=3;i++)                     // for A B C D Stepping
        {
          var1 = var1>>1;                    //For Anticlockwise
          var2 = ~var1;
```

```
            var2 = var2 & 0x0000F000;
            IO0PIN = ~var2;
           for(k=0;k<3000;k++);                    //for step speed variation
        }
}
```

—----------

OUTPUT

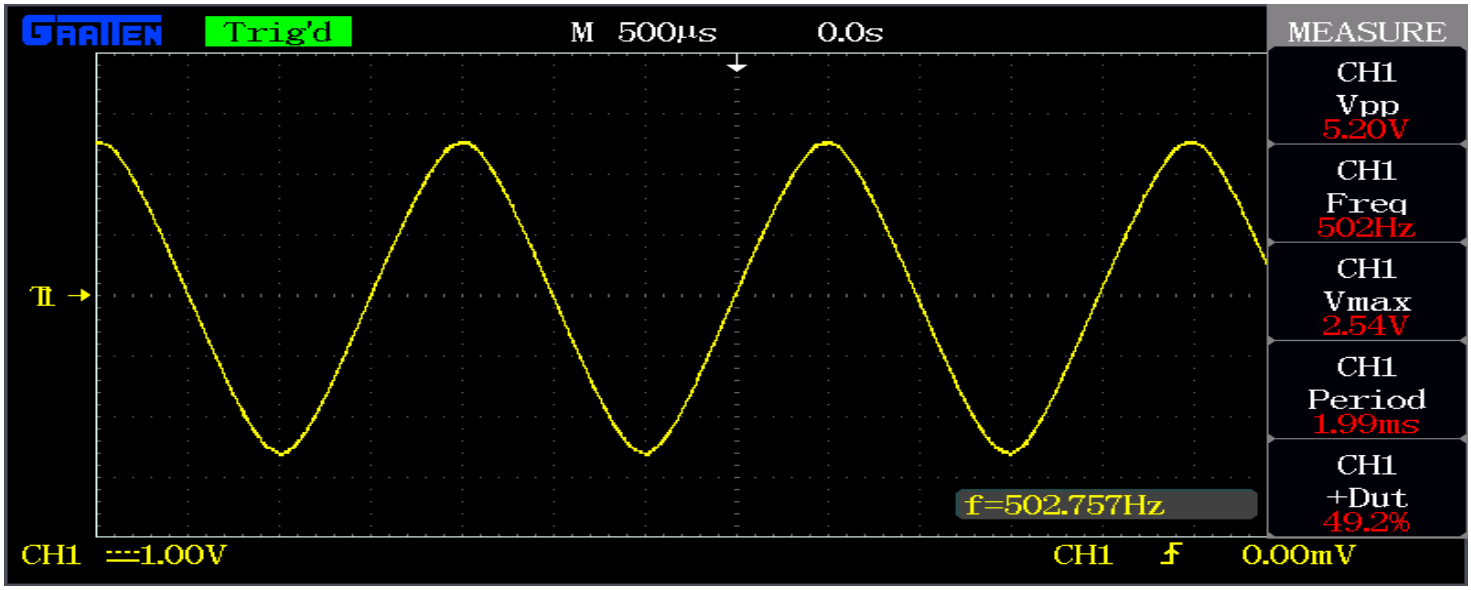
Rotation of Stepper Motor in clockwise & anticlockwise direction has been observed.

Experiment 10a: Interface a DAC and generate Triangular and Square waveforms.

// Sine wave

```
#include <LPC21xx.h>

int count=0,sinevalue,value;
unsigned char sine_tab[49]=
     { 0x80,0x90,0xA1,0xB1,0xC0,0xCD,0xDA,0xE5,0xEE,0xF6,0xFB,0xFE,
      0xFF,0xFE,0xFB,0xF6,0xEE,0xE5,0xDA,0xCD,0xC0,0xB1,0xA1,0x90,
       0x80,0x70,0x5F,0x4F,0x40,0x33,0x26,0x1B,0x12,0x0A,0x05,0x02,
       0x00,0x02,0x05,0x0A,0x12,0x1B,0x26,0x33,0x40,0x4F,0x5F,0x70,0x80};

int main(void)
{
   PINSEL0 = 0x00000000 ;                                  // Configure P0.0 to P0.15 as GPIO
   IO0DIR  = 0x00FF0000 ;
   count = 0;
         while(1)
         {
                  for(count=0;count<48;count++)
                  {
                           sinevalue = sine_tab[count];                        //+0X10 ;
                           value= 0x00FF0000 & (sinevalue << 16);
                           IO0PIN = value;
                  }
         }
}
```

OUTPUT

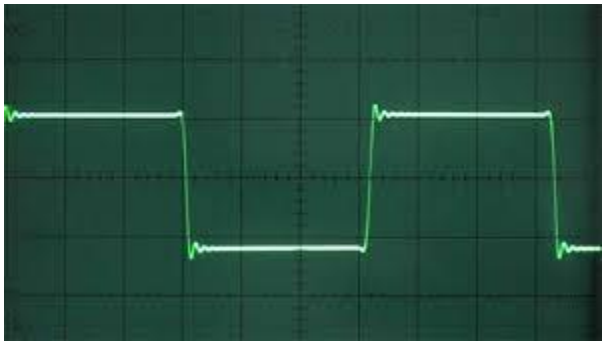


Experiment 10b: Interface a DAC and generate Triangular and Square waveforms.

// Square wave

```
#include <lpc21xx.h>
void delay(void);
int main ()
{
        PINSEL0 = 0x00000000 ;                  // Configure P0.0 to P0.15 as GPIO
        PINSEL1 = 0x00000000 ;                  // Configure P0.16 to P0.31 as GPIO
        IO0DIR  = 0x00FF0000 ;
        while(1)
   {
        IO0PIN = 0x00000000;
      delay();
      IO0PIN = 0x00FF0000;
      delay();
   }
}

void delay(void)
{
        unsigned int i=0;
        for(i=0;i<=95000;i++);
}
```

OUTPUT

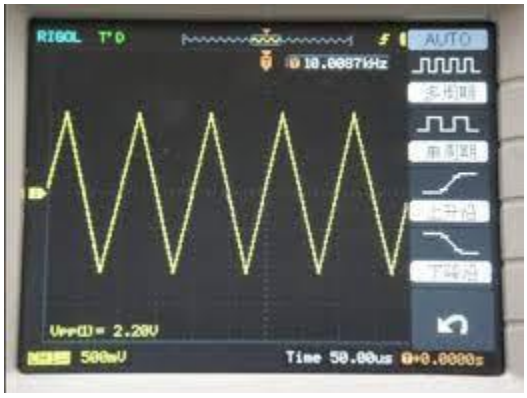


Experiment 10c: Interface a DAC and generate Triangular and Square waveforms.

// Triangular wave

```
#include <LPC21xx.h>

int main ()
{
        unsigned long int temp=0x00000000;
        unsigned int i=0;

        IO0DIR=0x00FF0000;

   while(1)
   {
                                                    // output 0 to FE
      for(i=0;i!=0xFF;i++)
      {
         temp=i;
         temp = temp << 16;
         IO0PIN=temp;
      }

                                                  // output FF to 1
      for(i=0xFF; i!=0;i--)
      {
         temp=i;
         temp = temp << 16;
         IO0PIN=temp;
      }
  }     //End of while(1)
}       //End of main()
```

Output



Experiment 11: Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in between

```
#include <LPC21XX.h>
unsigned int delay;
unsigned int Switchcount=0;
unsigned int Disp[16]={0x003F0000, 0x00060000, 0x005B0000, 0x004F0000, 0x00660000,0x006D0000,
                       0x007D0000, 0x00070000, 0x007F0000, 0x006F0000, 0x00770000,0x007C0000,
                        0x00390000, 0x005E0000, 0x00790000, 0x00710000 };

#define SELDISP1 0x10000000              //P0.28
#define SELDISP2 0x20000000              //P0.29
#define SELDISP3 0x40000000              //P0.30
#define SELDISP4 0x80000000              //P0.31
#define ALLDISP  0xF0000000              //Select all display
#define DATAPORT 0x00FF0000              //P0.16 to P0.23 Data lines connected to drive Seven Segments

int main (void)
{
        PINSEL0 = 0x00000000;
        PINSEL1 = 0x00000000;
        IO0DIR  = 0xF0FF0000;
        IO1DIR  = 0x00000000;

        while(1)
        {
                IO0SET |= ALLDISP;                      // select all digits
                IO0CLR = 0x00FF0000;                    // clear the data lines to 7-segment displays
                IO0SET = Disp[Switchcount];         // get the 7-segment display value from the array

                if(!(IO1PIN & 0x00800000))          // if the key is pressed
                {
                        for(delay=0;delay<100000;delay++)   // delay
                          {}

                   if((IO1PIN & 0x00800000))   // check to see if key has been released
                        {
                                Switchcount++;
                                if(Switchcount == 0x10)        // 0 to F has been displayed ? go back to 0
                          {
                                  Switchcount = 0;
                                  IO0CLR  =   0xF0FF0000;
                          }
                        }}}}
```

OUTPUT-
On 7-segment display numbers display from 0000–1111–2222— ,,, FFFF OBSERVED.

Experiment 12: Interface a simple Switch and display its status through Relay, Buzzer and LED.

OUTPUT–

simple Switch and display its status through Relay, Buzzer and LED

```
#include <lpc214x.h>

void delay_ms(unsigned int count) {
    unsigned int i, j;
    for (i = 0; i < count; i++)
        for (j = 0; j < 6000; j++);
}

int main(void) {
    IO0DIR |= (1 << 10) | (1 << 11) | (1 << 12) | (1 << 13); // Set P0.10, P0.11, P0.12, P0.13 as output
    IO0DIR &= ~(1 << 10); // Set P0.10 as input

    while (1) {
        if (IO0PIN & (1 << 10)) {
            // Switch is not pressed
            IO0SET = (1 << 11);  // Activate relay
            IO0CLR = (1 << 12);  // Turn off buzzer
            IO0SET = (1 << 13);  // Turn on LED
        } else {
            // Switch is pressed
            IO0CLR = (1 << 11);  // Deactivate relay
            IO0SET = (1 << 12);  // Turn on buzzer
            IO0CLR = (1 << 13);  // Turn off LED
        }

        delay_ms(100); // Add a delay for stability
    }

    return 0;
}
```