Q11. Construct appropriate functions to send and write 8-bit data to the data register of LCD operating in 4-bit mode. Data lines are connected to P0.4 to P0.7, RS line is connected to P0.8 and Enable line is connected to P0.9. (2)

```
  void lcd_data(void)
    { temp2 = temp1 & 0xf0;
     write_data();
     delay_lcd(30000);
     temp2= temp1 & 0x0f;
     temp2= temp2 << 4;
     write_data();
     delay_lcd(30000);                          1M
     return; }
void delay(int j)
{int i;
for (i=0; i<j; i++);}
  void write_data(void)
  {      LPC_GPIO0->FIOPIN = temp2;
         LPC_GPIO0->FIOSET = 1<<8;
         LPC_GPIO0->FIOSET = 1<<9;
         delay_lcd(25);
         LPC_GPIO0->FIOCLR = 1<<9;
         Return; }                                  1M
```

Q12. Discuss the relationship between i) Timer Counter(TC) ii) Prescale Register(PR) iii) Prescale Counter (PC). (2)

When a Timer is Reset and Enabled, the TC is set to 0 and incremented by 1 every 'PR+1' clock cycles – where PR is the value stored in Prescale Register. When it reaches its maximum value it gets reset to 0 and hence restarts counting.

Prescale Register is used to define the resolution of the timer. If PR is 0 then TC is incremented every 1 clock cycle of the peripheral clock. 1M

Prescale Counter (PC) register increments on every PCLK(Peripheral clock). This register controls the resolution of the timer. When PC reaches the value in PR , PC is reset back to 0 and Timer Counter is incremented by 1 1M

Q13. Develop an embedded C program to rotate the stepper motor in clockwise and anticlockwise direction continuously. The stepper motor is connected through the port pins P0.4 to P0.7 (3)

```
#include <LPC17xx.H>
```

```
void clock_wise(void);
void anti_clock_wise(void);
unsigned long int var1,var2;
unsigned int i=0,j=0,k=0;                                   0.5M
int main(void)
{   SystemInit();
    SystemCoreClockUpdate();
    LPC_GPIO0->FIODIR = 0x000000F0;          0.5M
    while(1)
    {    for(j=0;j<50;j++)
            clock_wise();
            for(k=0;k<65000;k++);
            for(j=0;j<50;j++)
            anti_clock_wise();
  for(k=0;k<65000;k++); }  }                  1M
 void clock_wise(void)
 {   var1 = 0x00000008;
     for(i=0;i<=3;i++)
     {  var1 = var1<<1;
        LPC_GPIO0->FIOPIN = var1;
        for(k=0;k<3000;k++); } }                    0.5M
 void anti_clock_wise(void)
 {   var1 = 0x00000100;
    for(i=0;i<=3;i++)
     {  var1 = var1>>1;
        LPC_GPIO0->FIOPIN = var1;
        for(k=0;k<3000;k++);   } }                      0.5M
```

Q14. Develop an embedded C program to display 1-digit BCD up counter in any one of the four Seven Segment Display units. Configure port pins P1.23 to P1.30 as data lines and P0.15 to P0.18 as enable lines. The hexadecimal values corresponding to digits 0 to 9 are {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F}. (3)

```
#include<LPC17xx.h>

int i,j;

unsigned int seg[10] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};

int d[4];                                   0.5M

int main(void)

{          LPC_GPIO1->FIODIR = 0X7F800000 | 0xFF<<23; //SETTING OUTPUT PINS          0.5M

          LPC_GPIO0->FIODIR |= 0X00078000 | 0xF<<15; // SETTING OUTPUT PINS          0.5M

          while(1)
```

```
{       LPC_GPIO0->FIOPIN = 0 << 15;                    0.5M
        for(d[0] = 0; d[0] <10; d[0]++) {
                LPC_GPIO1->FIOPIN = seg[d[0]]<<23;      0.5M
                for(j = 0; j<10000 ; j++);
                LPC_GPIO1->FIOCLR = 0X00000FF0;  } } }      0.5M
```