EXP NO:1	LED FLASHING WITH ARM LPC2148 MICROCONTROLLER
DATE	

AIM:

To write and execute the program for LED Flashing with ARM7 (LPC2148) Processor.

HARDWARE & SOFTWARE TOOLS REQUIRED:

S.No	Hardware & Software Requirements	Quantity
1	ARM Processor LPC2148	1
3	LEDs and Resistors	1
5	Proteus ISIS Software and Keil Software	

PROCEDURE

1. Create a New project, Go to "Project" and close the current project "Close Project".

2. Next Go to the Project New µvision Project Create New Project

Select Device forTarget.

3. Select the data base NXP LPC2148.

4. Add Startup file and Next go to "File" and click "New".

5. Write a program on the editor window and save as "xxxxx.c".

6. Add this source file to Group and click on "Build Target" or F7.

7. Create a Hex file from "Project" menu and click on "Rebuild all target Files".

8. Create the circuit diagram in Proteus Software and upload the hex file of your program by double clicking on the microcontroller in Proteus workspace.

9. Debug and run your Proteus file to see the working of the circuit.

LED INTERFACING CIRCUIT DIAGRAM:
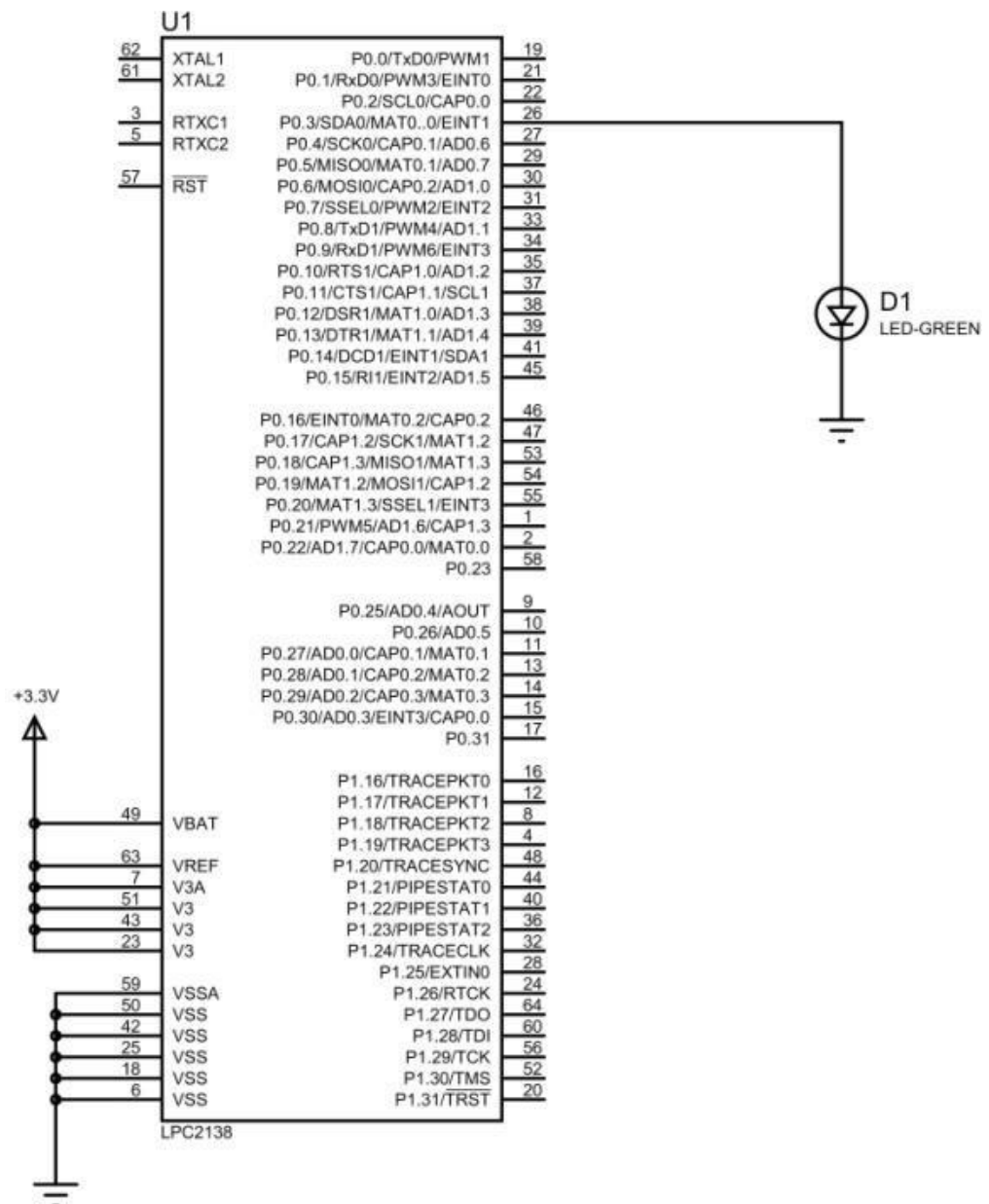
PROGRAM:
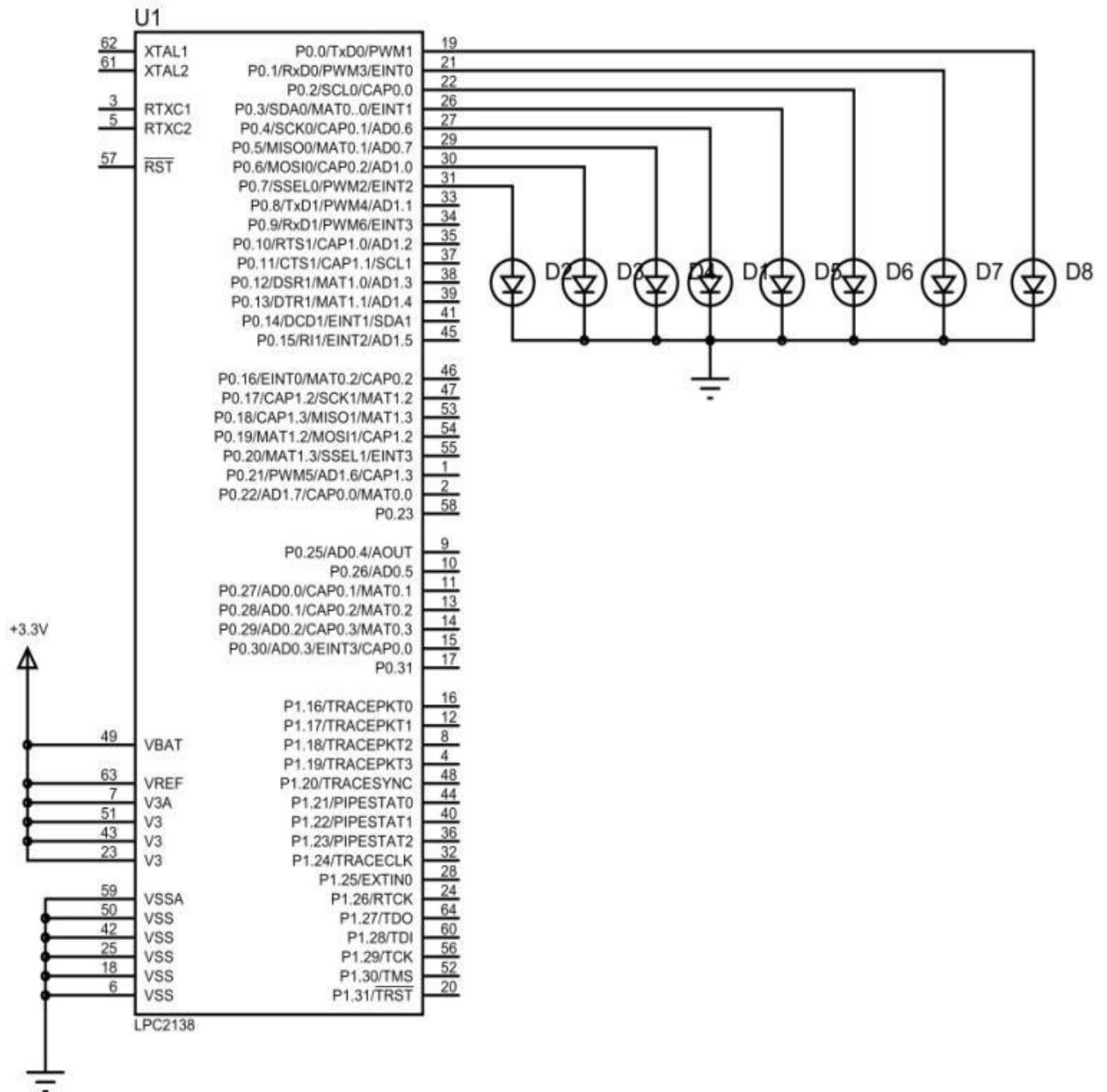
```c
#include
<lpc214x.h>int i;
int main()
{ IODIR0=(1<<
3);
        while(1)
        {
         IOSET0=(1<<3);
         for(i=0;i<120000;i++);
         IOCLR0=(1<<3);
         for(i=0;i<120000;i++);
        }
}
```

PROGRAM:

FLASHING OF MORE LEDS

PROGRAM TYPE-I:

```c
#include <lpc214x.h>
int i;
int main()
{ IODIR0=0x000000FF;
        while(1)
        {
         IOSET0=0x000000AA;
         for(i=0;i<120000;i++);
         IOCLR0=0x000000AA;
         for(i=0;i<120000;i++);
        }
}
```

TYPE-II:

```c
#include <lpc214x.h>
int i,b;
int main()
{ IODIR0=0x000000FF;
while(1)
 {
 for(b=0;b<8;b++)
   {
        IOSET0=(1<<b);
        for(i=0;i<120000;i++);
        IOCLR0=(1<<b);
        for(i=0;i<120000;i++);
   }
 }
}
```

EXP NO:2	SWITCH CONTROLLED LED
DATE	

AIM:

      To write and execute the program for switch controlled LED with ARM7 (LPC2148) Processor.

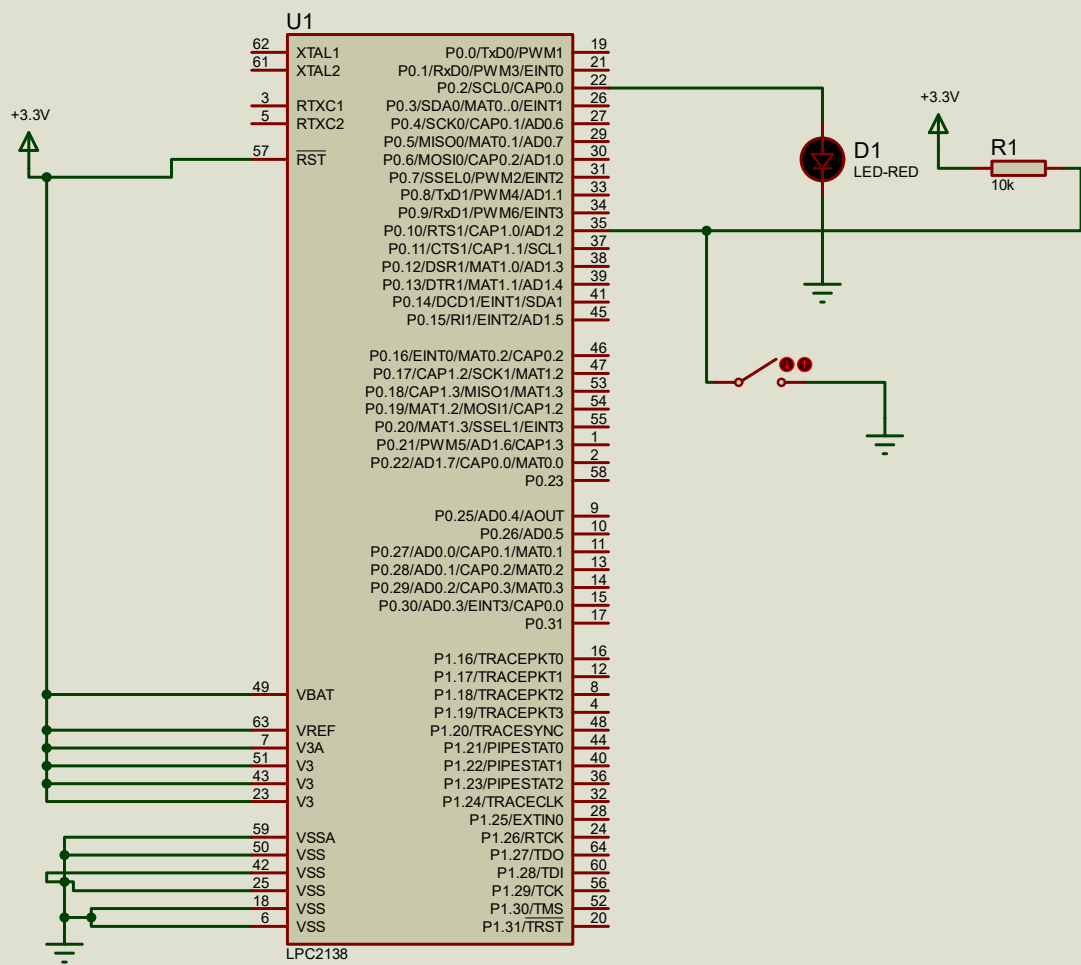**HARDWARE & SOFTWARE TOOLS REQUIRED:**

S.No	Hardware & Software Requirements	Quantity
1	ARM Processor LPC2148	1
3	LEDs and Switch	
5	Proteus ISIS Software and Keil Software	

PROCEDURE

1. Create a New project, Go to "Project" and close the current project "Close Project".
2. Next Go to the Project New µvision Project Create New Project Select Device forTarget.
3. Select the data base NXP LPC2148.
4. Add Startup file and Next go to "File" and click "New".
5. Write a program on the editor window and save as "xxxxx.c".
6. Add this source file to Group and click on "Build Target" or F7.
7. Create a Hex file from "Project" menu and click on "Rebuild all target Files".
8. Create the circuit diagram in Proteus Software and upload the hex file of your program by double clicking on the microcontroller in Proteus workspace.
9. Debug and run your Proteus file to see the working of the circuit.

      SWITCH CONTROLLED LED CIRCUIT DIAGRAM:

U1

Pin	Label
62	XTAL1
61	XTAL2
3	RTXC1
5	RTXC2
57	RST
49	VBAT
63	VREF
7	V3A
51	V3
43	V3
23	V3
59	VSSA
50	VSS
42	VSS
25	VSS
18	VSS
6	VSS

P0.0/TxD0/PWM1 19
P0.1/RxD0/PWM3/EINT0 21
P0.2/SCL0/CAP0.0 22
P0.3/SDA0/MAT0..0/EINT1 26
P0.4/SCK0/CAP0.1/AD0.6 27
P0.5/MISO0/MAT0.1/AD0.7 29
P0.6/MOSI0/CAP0.2/AD1.0 30
P0.7/SSEL0/PWM2/EINT2 31
P0.8/TxD1/PWM4/AD1.1 33
P0.9/RxD1/PWM6/EINT3 34
P0.10/RTS1/CAP1.0/AD1.2 35
P0.11/CTS1/CAP1.1/SCL1 37
P0.12/DSR1/MAT1.0/AD1.3 38
P0.13/DTR1/MAT1.1/AD1.4 39
P0.14/DCD1/EINT1/SDA1 41
P0.15/RI1/EINT2/AD1.5 45

P0.16/EINT0/MAT0.2/CAP0.2 46
P0.17/CAP1.2/SCK1/MAT1.2 47
P0.18/CAP1.3/MISO1/MAT1.3 53
P0.19/MAT1.2/MOSI1/CAP1.2 54
P0.20/MAT1.3/SSEL1/EINT3 55
P0.21/PWM5/AD1.6/CAP1.3 1
P0.22/AD1.7/CAP0.0/MAT0.0 2
P0.23 58

P0.25/AD0.4/AOUT 9
P0.26/AD0.5 10
P0.27/AD0.0/CAP0.1/MAT0.1 11
P0.28/AD0.1/CAP0.2/MAT0.2 13
P0.29/AD0.2/CAP0.3/MAT0.3 14
P0.30/AD0.3/EINT3/CAP0.0 15
P0.31 17

P1.16/TRACEPKT0 16
P1.17/TRACEPKT1 12
P1.18/TRACEPKT2 8
P1.19/TRACEPKT3 4
P1.20/TRACESYNC 48
P1.21/PIPESTAT0 44
P1.22/PIPESTAT1 40
P1.23/PIPESTAT2 36
P1.24/TRACECLK 32
P1.25/EXTIN0 28
P1.26/RTCK 24
P1.27/TDO 64
P1.28/TDI 60
P1.29/TCK 56
P1.30/TMS 52
P1.31/TRST 20

LPC2138

+3.3V

D1
LED-RED

+3.3V

R1
10k

PROGRAM:

```c
#include <lpc214x.h>
int i,b;

int main()
{ IODIR0=0x000000FF;
IODIR0=~(1<<16);

while(1)
{
if((IOPIN0&(1<<16))==0)
{
for(b=0;b<8;b++)
{
IOSET0=(1<<b);
for(i=0;i<120000;i++);
IOCLR0=(1<<b);
for(i=0;i<120000;i++);
}
}
else
{
IOCLR0=0x000000FF;
}
}
}
```

EXP NO:3	7-segment LED Display with LPC2148 Microcontroller
DATE	

AIM:

To write and execute the program for seven segment led with ARM7 (LPC2148) Processor.

HARDWARE & SOFTWARE TOOLS REQUIRED:

S.No	Hardware & Software Requirements	Quantity
1	ARM Processor LPC2148	1
3	7seg LED	1
5	Proteus ISIS Software and Keil Software	

PROCEDURE

1. Create a New project, Go to "Project" and close the current project "Close Project".
2. Next Go to the Project New µvision Project Create New Project

   Select Device forTarget.
3. Select the data base NXP LPC2148.
4. Add Startup file and Next go to "File" and click "New".
5. Write a program on the editor window and save as "xxxxx.c".
6. Add this source file to Group and click on "Build Target" or F7.
7. Create a Hex file from "Project" menu and click on "Rebuild all target Files".
8. Create the circuit diagram in Proteus Software and upload the hex file of your program by double clicking on the microcontroller in Proteus workspace.
9. Debug and run your Proteus file to see the working of the circuit.

```c
#include <lpc214x.h>
void delay();
unsigned int i, j, k;
unsigned int
ar[10]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
int main(void){
    IODIR0= 0x000000FF;
    while(1){
        for(i=0; i<10; i++) {
            IOSET0=ar[i];
            delay();
            IOCLR0=ar[i];
        }
    }
    return 0;
}
void delay(){
                                for(j=0;j<1000;j++);
    for(k=0;k<400;k++);
}
```
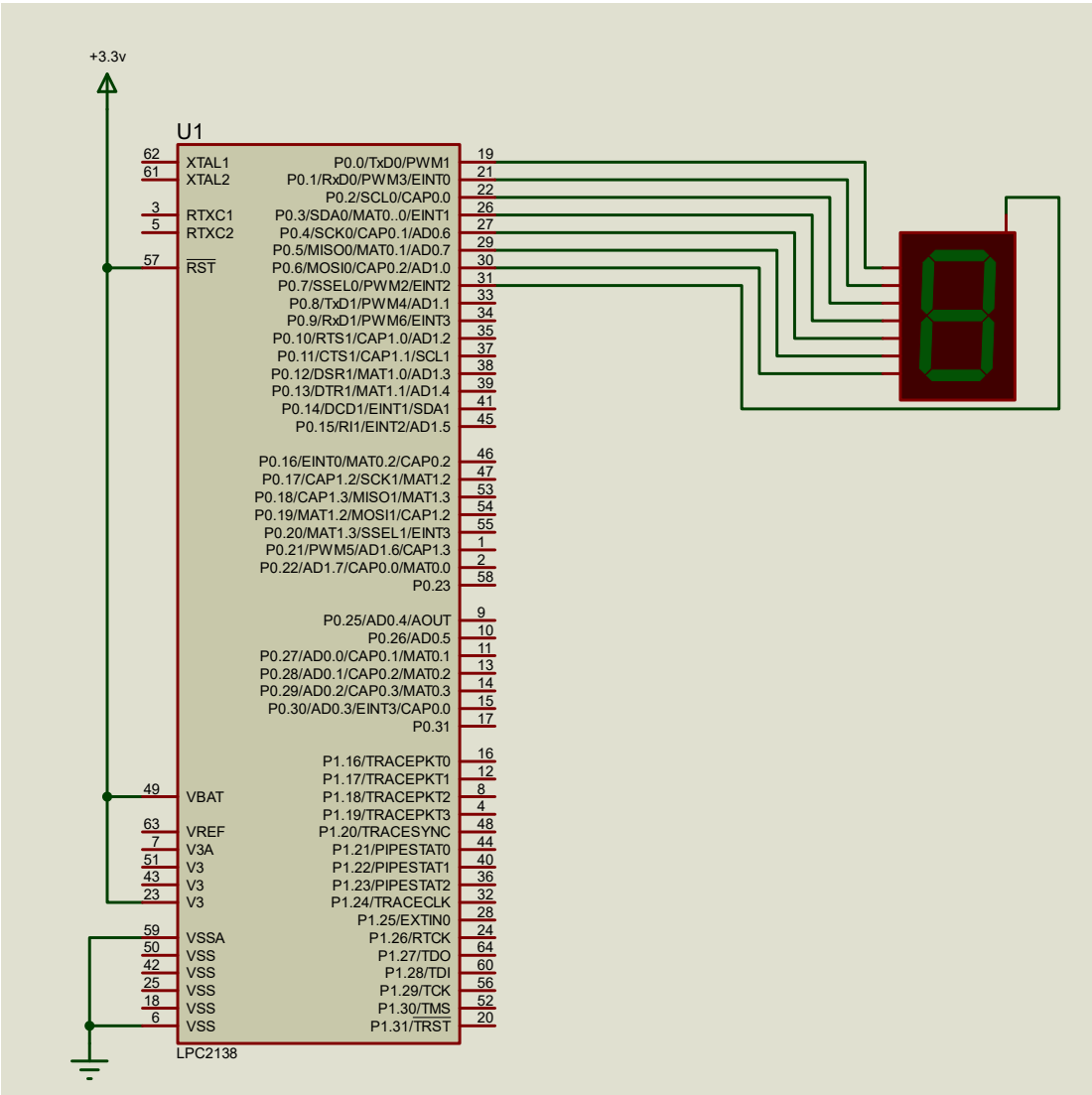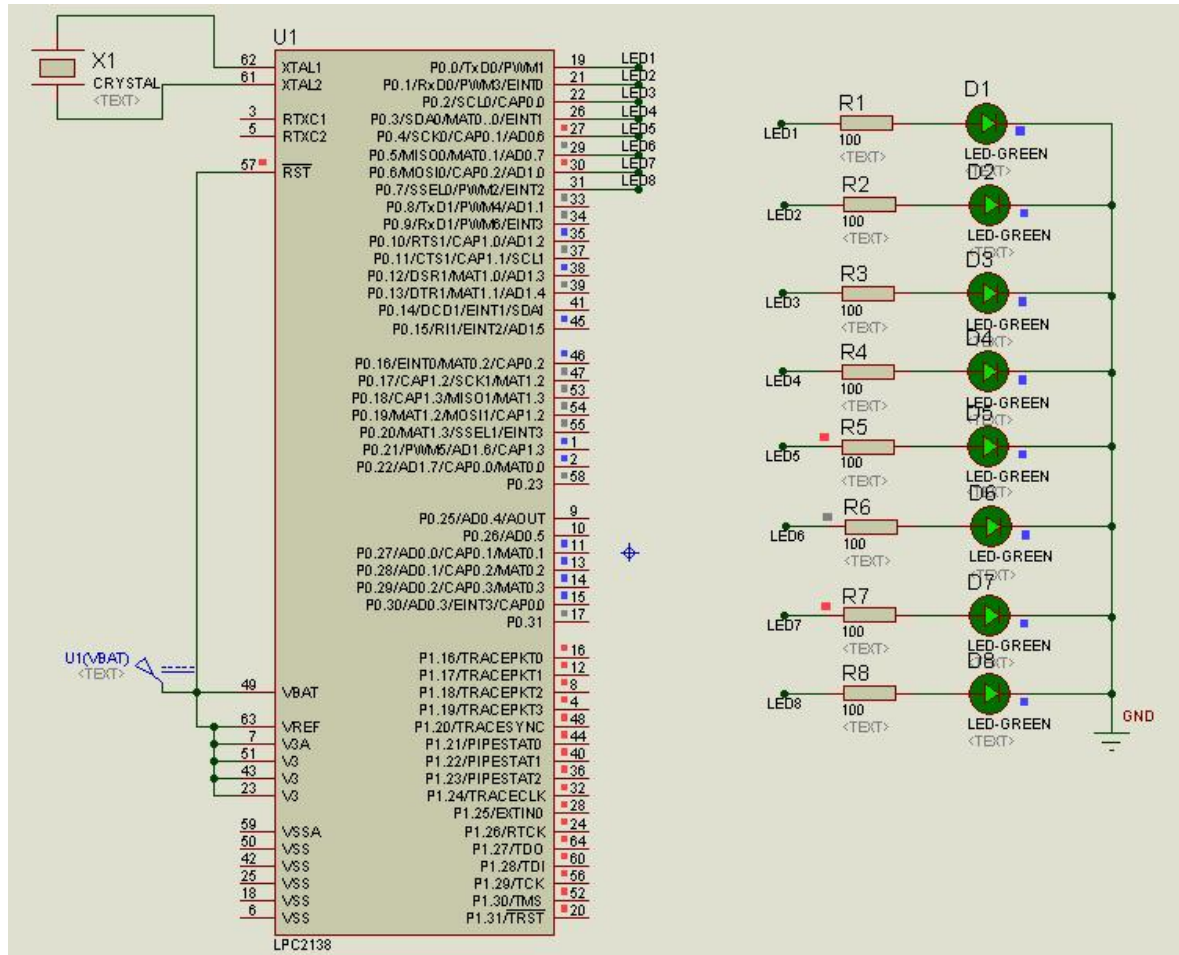
EXP NO:4	TIMERS
DATE	

```
#include <LPC214x.H>
void timer_sec(unsigned int x)
{
 T0MR0 = x;
 T0MCR = 1;
 T0PR = 0x00E4E1C0;
 T0TCR = 0x1;
 while(T0TC < T0MR0);
 T0TCR = 0x0;
 T0TC = 0x0;
}

int main (void)
{
  unsigned int n;
  PINSEL2=0x00000000;
  IODIR0 = 0x00FF;
  while (1)
  {


      IOSET0 = 0x00FF;
      timer_sec(1);
      IOCLR0 = 0x00FF;
      timer_sec(1);

  }
}
```
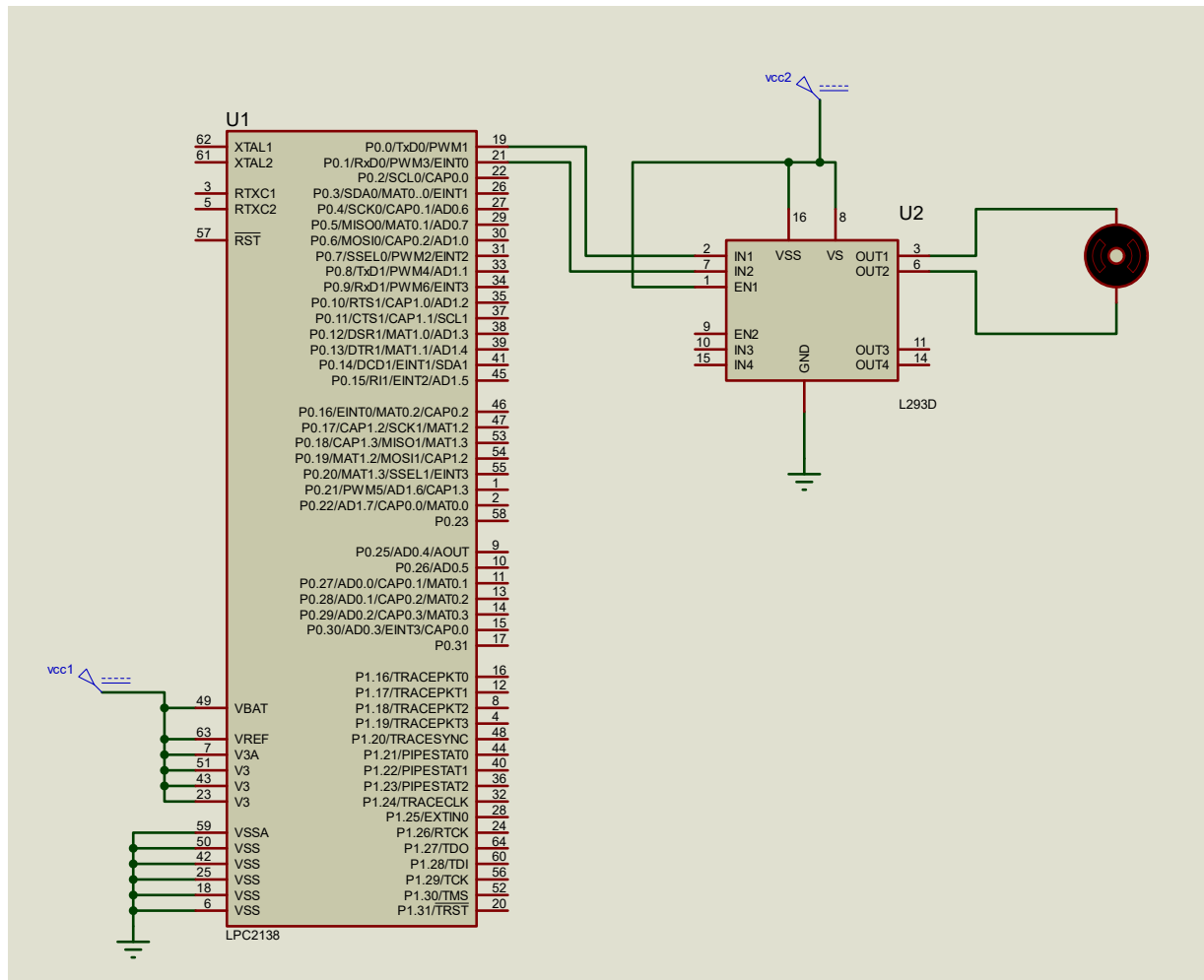
EXP NO:5	**DC Motor Interfacing**
DATE:	

AIM:

To design a circuit and develop a C program for controlling a DC motor.

HARDWARE AND SOFTWARE REQUIRED:

S.No	Hardware & Software Requirements	Quantity
1	ARM Processor LPC2148	1
2	L293D IC	1
3	Proteus Software and Keil Software	-
4	Simple DC Motor active	1

CIRCUIT DIAGRAM:

PROGRAM:

```c
#include <lpc214x.h>
void delay();
unsigned int j,k;
void main(void)
{
IO0DIR=0x03;
while(1)
{
IOSET0 = (1<<0);
IOCLR0=(1<<1) ;
delay();
delay();
delay();
delay();
IOCLR0=0x03;
delay();
delay();
delay();
delay();
IOSET0 = (1<<1);
IOCLR0=(1<<0) ;
delay();
delay();
delay();
delay();
IOCLR0=0x03;
delay();
delay();
delay();
delay();
}
}
```

```
void delay()
{ for (j=0;j<1000;j++)
for (k=0;k<400;k++);
}
```

EXP NO:6	Stepper Motor Interfacing with LPC2138
DATE:	

AIM:

        To design and develop a C program for interfacing Stepper Motor with LPC2138.

HARDWARE AND SOFTWARE REQUIRED:

S.No	Hardware & Software Requirements	Quantity
1	ARM Processor LPC2138	1
2	ULN2003A	1
3	Proteus Software and Keil Software	-
4	Stepper Motor Animated Unipolar.	1

PROCEDURE:

1. Create a New project, Go to "Project" and close the current project "Close Project".
2. Next Go to the Project New µvision Project Create New Project Select Device forTarget.
3. Select the data base NXP LPC2148.
4. Add Startup file and Next go to "File" and click "New".
5. Write a program on the editor window and save as "Main.c".
6. Add this source file to Group and click on "Build Target" or F7.
7. Create a Hex file from "Project" menu and click on "Rebuild all target Files".
8. Open Flash magic and select the device LPC2148 in ARM 7
9. Next browse the path of hex file and click ok.
10. Start debugging process and run the simulation for getting output.

CIRCUIT DIAGRAM:
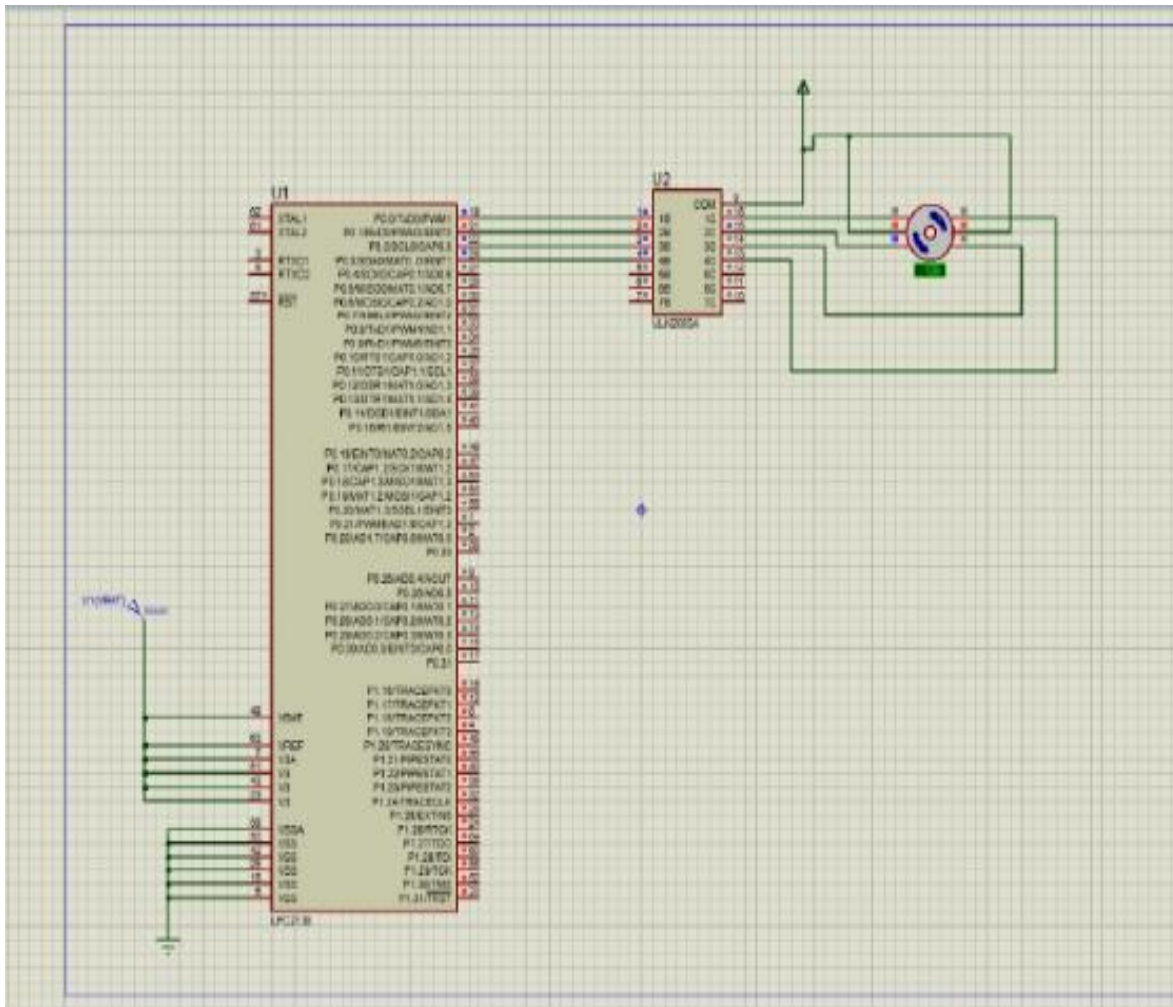
PROGRAM:

```c
#include<lpc213x.h>
#include<stdint.h>
uint8_t i,j;
void delay_ms(uint16_tj)
{
uint16_t x,i;
for(i=0;i<j;i++)
{
for(x=0;x<6000;x++);
}
}
int main(void)
{
IO0DIR=(IO0DIR | 0x0000000F);
while(1) {
for(j=0;j<12;j++)
{
```

```
IO0PIN=1<<0;
delay_ms(500);
IO0PIN=1<<3;
delay_ms(500);
}
}
}
```