EXPERIMENT 1 LED INTENSITY CONTROL

SOURCE CODE:

```
#include <LPC214X.H>

int main(void)
{
        PINSEL0=2;                      // P0.0 is configured as PWM1 output
        PWMPC=0;                        //PWM Pre scale counter value is 0; it is not used
        PWMPR=0;                        //PWM Pres scale register value is 0; it is not used
        PWMMR0 = 12000000;// Match register 0 has count for period; in this case approx. 1 sec.
                                        // Peripheral clock is 12 MHz
        PWMMR1 = 1200000;               // On time approx. 0.45 sec
        PWMMCR=0X2;          // when match occurs with the value in Match Register 0, reset

        //PWMLER = 0x3;                 // Enable PWM Match 0 Latch and PWM Match1 Latch
                                        // to transfer the values to shadow registers
        PWMPCR = 1<<9;                  // to enable PWM1 output (9th bit)
        PWMTCR = 0x9;                   // to enable Timer counter (0th bit) and PWM(3rd bit)
        while(1);                       // idling
}
// to vary the on time, the content of PWMMR1 is to be changed.  Maximum time is period
// which is stored in PWMMR0
```

EXPERIMENT 2 DISPLAY CHAR IN LCD

SOURCE CODE:

```
#include <lpc214x.h>
#include <string.h>
void delay(void)                    // delay routine
{
        unsigned int i;
        i = 0xfff;
        while (i--);
}
void write(unsigned char data)
{
        IOSET0 = data <<16;
                        // data is shifted left by 16 bits for sending to data bits (P0.23 - P0.16)
        delay();                        // delay is called
        IOSET1 = 1 << 31;     // enable bit is set to 1
        delay();                        // delay is called
        IOCLR1 = 1 << 31;     // enable bit is made to 0
        delay();                        // delay is called
        IOCLR0 = data <<16; // data lines are cleared
}
```

```
void cmd_write(unsigned char data)
{
        IOCLR1 = 1 << 30;     // make RS as 0
        IOCLR1 = 1 <<31;      // make enable bit as 0
        write(data);          // write routine is called
}

void data_write(unsigned char data)
{
        IOSET1 = 1 << 30;     // make RS as 1
        IOCLR1 = 1 <<31;      // make enable bit as 0
        write(data);          // write routine is called
}
void lcd_init()
{
        cmd_write(0x38);       // 5/8 matrix and 8 bit data
        cmd_write(0xe);        // first byte of the initialization sequence
        cmd_write(0x1);        // clear the display
        cmd_write(0x6);        // the cursor moves right when a data is entered
}
void disp(unsigned char *msg)         // display message routine, * denotes the address
pointer of the msg array
{
        unsigned int i,j;
        j = strlen(msg);              // function to find the length of the message
```

```
        for (i = 0; i<j; i ++)       // for the string length, for loop is repeated
                {
                data_write (msg[i]); // each character is written
                }
}
int main(void)                       // main program
{
        PINSEL1 = 0 ;                       // P0.16 to P0.31 are I/O
        IODIR0 = 0xff << 16;  // P0.16 to P0.23 are output lines (data lines)
        PINSEL2 = 0;              // P1 is I/O
        IODIR1 = 0x3 <<30;   // make P1.30 and P1.31 as output lines
        lcd_init();                  // LCD initialization routine is called
        delay();                            // delay routine is called
        cmd_write(0x80);        // cursor is in I row and I column
        disp ("    Welcome to      ");     // message is displayed in first row
        cmd_write (0xc0);       // cursor is in II row and I column
        disp ("Electronics Dept"); // message is displayed in second row
        while (1) ;                  // idling
}
```

EXPERIMENT 3 STEPPER MOTOR

```
// P1.16 (TIP1) and P1.18 (TIP2) are used to switch on one of the windings at a time in one pair
// P1.17 (TIP3) and P1.19 (TIP4) are used to switch on one of the windings at a time in another pair
// 4 steps are used to run the motor – full step mode
// program makes it to revolve in one direction, the speed can be increased by decreasing the delay
// For reverse direction, the sequence is to be reversed
// bit 0 switches on the winding
#include <lpc214x.h>
void delay(void) {
   unsigned int i;
   for(i=0; i<0x7ffff; i++);
}
int main(void) {
   PINSEL2 = 0;            // configure P1.16 to P1.31 as I/O
   IODIR1 = 0x0F<<16;      // configure stepper motor port as output
   IOSET1 = 0x0F<<16;      // no winding is switched on
   while(1) {              // continuous loop
      IOSET1 = 0x03<<16;   // 0011 - in each pair, first winding is switched in both pairs
      delay();
      IOCLR1 = 0x03<<16;   // all the outputs are momentarily switched off
      IOSET1 = 0x09<<16;   // 1001 - in first pair, the current is switched on to second winding
      delay();
      IOCLR1 = 0x09<<16;   // all the outputs are momentarily switched off
      IOSET1 = 0x0C<<16;   // 1100 - in second pair, the current is switched on to second winding
      delay();
      IOCLR1 = 0x0C<<16;   // all the outputs are momentarily switched off
      IOSET1 = 0x06<<16;   // 0110 - in first pair, the current is switched on to first winding
      delay();
      IOCLR1 = 0x06<<16;   // all the outputs are momentarily switched off

      // 1010 - again the cycle repeats
      // for reverse direction, the pattern should be 0x03, 0x06, 0x0C, 0x09 and repeats
   }
}
```

EXPERIMENT 4 REALTIME CLOCK

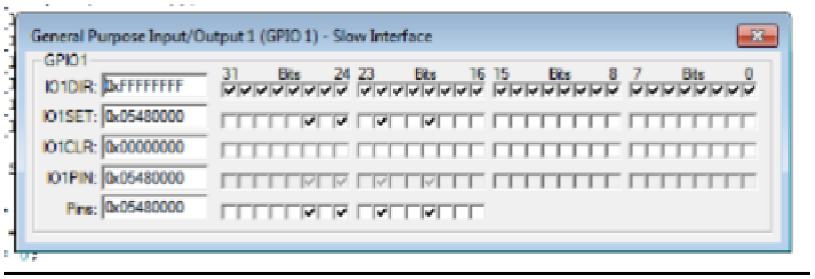
SOURCE CODE:

```
#include<lpc21xx.h>  // header file for LPC21XX series
#define  rs (1<<24)    // register select pin
#define  rw (1<<25)    // read write pin
#define  en (1<<26)    // enable pin
void  delay(int  j )       // Time delay function in milli seconds
{
int i; for(;j;j--)
for(i=6000;i;i--);
}
void data_lcd(char  ch)          // Function to send data to LCD
{
int i =0; i = ch;
i = i<<16;
IOPIN1 &=(0XFF00FFFF); IOPIN1 |= i;
IOSET1 = rs; IOCLR1 = rw; IOSET1 = en;
delay(2); IOCLR1  = en;
}
void cmd_lcd(char  ch)          // Function to send command to LCD
{
int i =0; i = ch;
i = i<<16;
IOPIN1 &=(0XFF00FFFF); IOPIN1 |= i;
IOCLR1 = rs; IOCLR1 = rw;

IOSET1 = en;
delay(2); IOCLR1  = en;
}
void init_lcd()  / Funtion to Initialize LCD
{
cmd_lcd(0x38);          // for using 8-bit 2 row mode and 5x7 Dots of LCD
cmd_lcd(0x01);          // clear screen
cmd_lcd(0x06);          // display ON
cmd_lcd(0x0c);          // force cursor to beginning of second row
cmd_lcd(0x80);          // clear screen
}
void str_lcd(char  *str)          // Function to display it in LCD
{
while(*str) data_lcd(*str++);
}
void time(void)          // function to perfom the operation of clock
{
cmd_lcd(0x80); str_lcd("HH:MM:SS"); cmd_lcd(0xc0); data_lcd(48+(HOUR/10)); data_lcd(48+(HOUR%10));
data_lcd(':'); data_lcd(48+(MIN/10)); data_lcd(48+(MIN%10)); data_lcd(':'); data_lcd(48+(SEC/10));
```

```
data_lcd(48+(SEC%10));
}
void SetTime(void)     // function to initialize RTC
{
CCR = 0x02; HOUR = 0;
MIN = 0;
SEC = 0; CCR = 0x11;
}
int  main(void)
{
SetTime();
PINSEL2 = 0X00000000;                  // select PORT1 as GPIO mode
IODIR1 = 0XFFFFFFFF;       // make PORT1 pin as Output mode init_lcd();
while  (1)        // Repeat(loop) forever
{
time();
}
}
```

OUTPUT:



EXPERIMENT 5 PIR SENSOR

```
#include<lpc214x.h>
#define bit(x) (1<<x)
#define delay for(i=0;i<7000;i++);

#define PIR (IO1PIN & (1<<24))

unsigned int i;
void lcd_int();
void dat(unsigned char);
void cmd(unsigned char);
void string(unsigned char *);
```

```
void main()
{
   IO0DIR =0XFFF;
   IO1DIR = 0x0;
   lcd_int();
   cmd(0x80);
   string("EMBETRONICX.COM ");
   while(1) {
         if(PIR == 0) {
               string("Intruder Detcted");
         }
         delay;delay;
         cmd(0x01);
      }
}

void lcd_int()
{
   cmd(0x38);
   cmd(0x0c);
   cmd(0x06);
   cmd(0x01);
   cmd(0x80);
}

void cmd(unsigned char a)
{
   IO0PIN&=0x00;
   IO0PIN|=(a<<0);
   IO0CLR|=bit(8);              //rs=0
   IO0CLR|=bit(9);              //rw=0
   IO0SET|=bit(10);              //en=1
   delay;
   IO0CLR|=bit(10);              //en=0
}

void dat(unsigned char b)
{
   IO0PIN&=0x00;
   IO0PIN|=(b<<0);
   IO0SET|=bit(8);              //rs=1
   IO0CLR|=bit(9);              //rw=0
   IO0SET|=bit(10);              //en=1
   delay;
   IO0CLR|=bit(10);              //en=0
}

void string(unsigned char *p)
{
   while(*p!='\0') {
      dat(*p++);
   }
}
```

EXPERIMENT 6

INTERFACING LED TO TOGGLE AT EQUAL TIME DELAY USING ARDUINO

PROGRAM

```
int LEDpin = 13; int
delayT = 1000; void
setup() {
  // put your setup code here, to run once:
pinMode(LEDpin, OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
digitalWrite(LEDpin, HIGH);
delay(delayT); digitalWrite(LEDpin,
LOW); delay(delayT);
}
```

EXPERIMENT 7 GAS LEAKAGE USING ARDUINO UNO

```
#define GAS_SENSOR A0
int threshold = 400;
int gasValue = 0;

void setup() {
  Serial.begin(9600);
  pinMode(GAS_SENSOR, INPUT);
}

void loop() {
  gasValue = analogRead(GAS_SENSOR);
  Serial.print("Gas Sensor Value: ");
  Serial.println(gasValue);
  if (gasValue > threshold) {
    Serial.println("Gas leak detected!");
  } else {
    Serial.println("Gas level normal.");
  }
  delay(1000);
}
```