```
#include "LPC23xx.h"

 /*******************************************************************************************
**************
   Routine to set processor and pheripheral clock
 ********************************************************************************************
**************/

 void  TargetResetInit(void)
 {
  // 72 Mhz Frequency
  if ((PLLSTAT & 0x02000000) > 0)
  {
     /* If the PLL is already running   */
     PLLCON  &= ~0x02;                /* Disconnect the PLL                        */
     PLLFEED  =  0xAA;                /* PLL register update sequence, 0xAA, 0x55          */
     PLLFEED  =  0x55;
  }
  PLLCON   &= ~0x01;                  /* Disable the PLL                           */
  PLLFEED   =  0xAA;                 /* PLL register update sequence, 0xAA, 0x55          */
  PLLFEED   =  0x55;
  SCS      &= ~0x10;               /* OSCRANGE = 0, Main OSC is between 1 and 20 Mhz       */
  SCS      |=  0x20;             /* OSCEN = 1, Enable the main oscillator              */
  while ((SCS &  0x40) == 0);
  CLKSRCSEL = 0x01;                /* Select main OSC, 12MHz, as the PLL clock source      */
  PLLCFG    = (24 << 0) | (1 << 16);         /* Configure the PLL multiplier and divider          */
  PLLFEED   = 0xAA;               /* PLL register update sequence, 0xAA, 0x55          */
  PLLFEED   = 0x55;
  PLLCON   |= 0x01;              /* Enable the PLL                            */
  PLLFEED   = 0xAA;               /* PLL register update sequence, 0xAA, 0x55          */
  PLLFEED   = 0x55;
  CCLKCFG   = 3;              /* Configure the ARM Core Processor clock divider       */
  USBCLKCFG = 5;              /* Configure the USB clock divider                    */
  while ((PLLSTAT & 0x04000000) == 0);
  PCLKSEL0  = 0xAAAAAAAA;                  /* Set peripheral clocks to be half of main clock        */
  PCLKSEL1  = 0x22AAA8AA;
  PLLCON   |= 0x02;              /* Connect the PLL. The PLL is now the active clock source  */
  PLLFEED   = 0xAA;               /* PLL register update sequence, 0xAA, 0x55          */
  PLLFEED   = 0x55;
  while ((PLLSTAT & 0x02000000) == 0);
  PCLKSEL0 = 0x55555555;       /* PCLK is the same as CCLK */
  PCLKSEL1 = 0x55555555;
 }


 // serial Reception routine
 int serial_rx(void)
 {
  while (!(U0LSR & 0x01));
  return (U0RBR);
 }
 //serial transmission routine
```

```
void serial_tx(int ch)
{
//  while ((U0LSR & 0x20)!=0x20);
  while ((U0LSR & 0x20)==0);
  U0THR = ch;
}
// serial transmission routine for string of characters
void string_tx(char *a)
{
      while(*a!='\0')
      {
        while((U0LSR&0X20)!=0X20);
        U0THR=*a;
        a++;
      }
}
/************************ main routine
**********************************************************/
int  main ()
{
 unsigned int Fdiv;
 char value;
 TargetResetInit();

 /*************************** uart1 initialization ***********************************/
 PINSEL0 = 0x00000050;

 U0LCR = 0x83;               // 8 bits, no Parity, 1 Stop bit
 Fdiv = ( 72000000 / 16 ) / 19200 ;  //baud rate
 U0DLM = Fdiv / 256;
 U0DLL = Fdiv % 256;
  U0LCR = 0x03;               // DLAB = 0


  while(1)
 {
  value=serial_rx();
  serial_tx(value+2);
 }
        return 0;
}
```