Problem :

Configure the ADC, give input to the ADC using Pot Resistors, and display the digital value on Tera Term

```
/********************* serial Transmission
routine***********************************/

void serial_tx(int ch)

{

  while ((U0LSR & 0x20)!=0x20);

  U0THR = ch;

}

/**************************** Routine for converting hex
value to ascii value *****************/

int atoh(int ch)

{

        if(ch<=0x09)

                ch = ch + 0x30;

        else

                ch = ch + 0x37;

        return(ch);

}

/*********************************** main routine
***************************************************
*/

int  main ()

{
        unsigned int Fdiv,value,i,j;

//      char value;

        TargetResetInit();

//      init_timer( ((72000000/100) - 1) );


        PCONP |=0X00001000;                     //switch adc from
disable state to enable state

        PINSEL0 = 0x00000050;                   //Pinselection for
uart tx and rx lines

        PINSEL1         = 0X01554000;
        //Pinselection for adc0.0

        /************** Uart initialization
*********************************/


        U0LCR = 0x83;           // 8 bits, no Parity, 1 Stop bit

        Fdiv = ( 72000000 / 16 ) / 19200 ;  //baud rate

        //Fdiv = ( 72000000 / 16 ) / 2400 ;  //baud rate

        U0DLM = Fdiv / 256;

        U0DLL = Fdiv % 256;

        U0LCR = 0x03;           // DLAB = 0


        AD0CR = 0X01210F01; // Adc initialization

        while(1)

        {

                while((AD0DR0 & 0X80000000)!
=0X80000000){};        // Wait here until adc make conversion
complete
```

```
                /************* To get converted value and
display it on the serial port****************/

                value = (AD0DR0>>6)& 0x3ff ;

                        //serial_tx(value);

                serial_tx('\t');

                serial_tx(atoh((value&0x300)>>8));

                serial_tx(atoh((value&0xf0)>>4));

                serial_tx(atoh(value&0x0f));

                serial_tx(0x0d);

                serial_tx(0x0a);


                        for(i=0;i<=0xFF;i++)

                        {

           for(j=0;j<=0xFFF;j++);

                        }

        }

 return 0;

}
```

Explanation:

There are two important steps in this program. The first one is to set UART0 for serial communication. For this please refer the serial communication program. The second part is to configure the ADC whose explanation is as given below:

```
PCONP |=0X00001000;
```

PCONP is Power Control for Peripherals register (please refer page number 68 in the LPC2378 manual).

PCONP sets the bit 12 which sets the ADC0.

```
PINSEL1        = 0X01554000;
```

Refer page number 158 in LPC2378 manual. PINSEL1 bit 15 and 14 has to be set to 01 to chose ADC0.

```
AD0CR = 0X01210F01;
```

Please refer page number 598 and 599 in the LPC2378 manual and decipher the significance of the value.

```
while((AD0DR0 & 0X80000000)!=0X80000000){};
```

We are waiting for the ADC conversion to be over.This is indicated by setting of bit 31 to 1. So we check bit 31 wait until it becomes 1.

```
value = (AD0DR0>>6)& 0x3ff ;
```

The AD0DR0 contains the converted 10-bit value from bit 15 to bit 6. Hence we have to right shift the value 6 times to the right and mask the other bits.

```
serial_tx(atoh((value&0x300)>>8));

serial_tx(atoh((value&0xf0)>>4));

serial_tx(atoh(value&0x0f));
```

We have to take each nibble of the 10 bit value, convert it ascii value and send it to the PC via serial port.