/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Expt. 3b.: On-chip ADC Programming using Interrupt

Platform: Smart Logic LPC2148 Development Board.

Clock Settings:

FOSC >> 12MHz (onboard)

PLL >> M=5, P=2

CCLK >> 60MHz

PCLK >> 15MHz

Hardware Setup:-

LCD data pin :-P0.16-P0.23

RS-P1.16

RW-P1.17

EN-P1.18

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Include Controller specific header file

#include <lpc214x.h>

//Function to generate software delay

//Calibrated to 1ms

void delay\_ms(unsigned char time)

{

unsigned int i, j;

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

for(i=0; i<8002; i++);

}

void LCD\_command(unsigned char command)

{

IOCLR0 = 0xFF<<16; // Clear LCD Data lines

IOCLR1=1<<16; // RS=0 for command

IOCLR1=1<<17; // RW=0 for write

IOSET0=command<<16; // put command on data line

IOSET1=(1<<18); // en=1

delay\_ms(10) ; // delay

IOCLR1=(1<<18); // en=0

}

LCD\_data(unsigned char data)

{

IOCLR0 = 0xFF<<16; // Clear LCD Data lines

IOSET1=1<<16; // RS=1 for data

IOCLR1=1<<17; // RW=0 for write

IOSET0= data<<16; // put command on data line

IOSET1=(1<<18); //en=1

delay\_ms(10) ; //delay

IOCLR1=(1<<18); //en=0

}

LCD\_init()

{

LCD\_command(0x38); //8bit mode and 5x8 dotes (function set)

delay\_ms(10) ; // delay

LCD\_command(0x80); //set cursor to 0th location 1st lne

delay\_ms(10) ; // delay

LCD\_command(0x01); //clear lcd(clear command)

delay\_ms(10) ; // delay

LCD\_command(0x06); //cursor increament and display shift(entry mode set)

delay\_ms(10) ; // delay

LCD\_command(0x0c); //display on, cursor off, cursor char blinking off(display on/off)

}

LCD\_write\_string(unsigned char \*string)

{

while(\*string) //Check for End of String

LCD\_data(\*string++); //sending data on LCD byte by byte

}

void \_\_irq ADC0\_ISR(void)

{

unsigned int ADC\_Result=0;

unsigned char i;

ADC\_Result = AD0DR2; //Store converted data

ADC\_Result = (ADC\_Result>>6) & 0x3FF;

LCD\_command (0xCA); //Goto 10th place on second line of LCD

i = (ADC\_Result/1000)+0x30 ; //Get the thousands place

LCD\_data (i); // Display thousands place

i = ((ADC\_Result%1000)/100)+0x30; //Get the Hundreds place

LCD\_data (i); //Display Hundreds place

i = (((ADC\_Result%1000)%100)/10)+0x30; //Get the Tens place

LCD\_data (i); //Display Tens place

i = (ADC\_Result%10)+0x30 ; //Get the Ones place

LCD\_data (i); //Display Ones place

VICVectAddr = 0x00; //Acknowledge Interrupt

}

int main(void)

{

PINSEL1 = 0x04000000;

PINSEL2 = 0X00; //Configure PORT1 as GPIO

IODIR1= 0x07<<16; //Configure P1.18, P1.17, P1.16 as output

IODIR0= 0xFF<<16; //Configure P0.23 - P0.16 as output

LCD\_init(); //Initialize LCD 16x2

LCD\_write\_string("Smart Logic Tech");

LCD\_command(0xc0);//second line

LCD\_write\_string("AD0.2 O/P=");

VICIntSelect = 0x00000000; // Setting all interrupts as IRQ(Vectored)

VICVectCntl0 = 0x20 | 18; // Assigning Highest Priority Slot to ADC0 and enabling this slot

VICVectAddr0 = (unsigned long)ADC0\_ISR; // Storing vector address of ADC0

VICIntEnable = (1<<18); //Enable AD0 Interrupt

/\* Configure ADC0 for following

ADC Channel = AD0.2

ADC Clock = 3 MHz

Clock Selection = 11 Clock Cycles/10bit

Start Condition = No start

Power Down = 1, EDGE = 0, BURST = 1 \*/

AD0CR = 0x00210404;

while (1);

}