T. Y. B. Tech (ECE)

Trimester: VI **Subject:** ESD & RTOS

Name: Satej Zunjarrao **Division: B**

Roll No: PB-30 **Batch: B3**

Experiment No: 04

Name of the Experiment: Programming ADC of LPC 2148 using interrupt controller

Performed on:

Submitted on:

Marks	Teacher's Signature with date

Aim: Write Embedded C program for programming ADC of LPC 2148 using interrupt controller.

Part List:

- Educational practice board for ARM7 LPC2148
- All in one general purpose board (ASK25)
- +9V power supply
- USB A to B type cable
- 20 pin flat cable
- Single lead wire
- PC
- Eclipse IDE
- Flash Magic Utility

Theory:

Analog to digital converter is among the most widely used device for data acquisition. It is used to convert the Analog signals to digital numbers so that microcontroller can read and process them. LPC2148 has two 10-bit successive approximation Analog to Digital converters named ADC0 and ADC1 with 06 and 08 channels respectively. Basic clocking for the A/D converters is provided by

the APB clock. A programmable divider is included in each converter, to scale this clock to the 4.5 MHz (max) clock needed by the successive approximation process. A fully accurate conversion requires 11 of these clocks. The result of A/D conversion is stored in A/D Global Data Register as well as A/D Data Register corresponding to the input channel. The result of conversion is displayed on 16x2 LCD display.

Hardware Connection:

For ADC Connection:

Pin No.	PL10 Connector of Ask25	J4 Connector of EPBARM7
1	POT-RV2	AD0.1
2	POT-RV3	AD0.2
3	GND	GND

The ASK-25A has a 3 pin Relimate connector PL10, which brings out 2 analog signals from potentiometer RV2 and RV3.

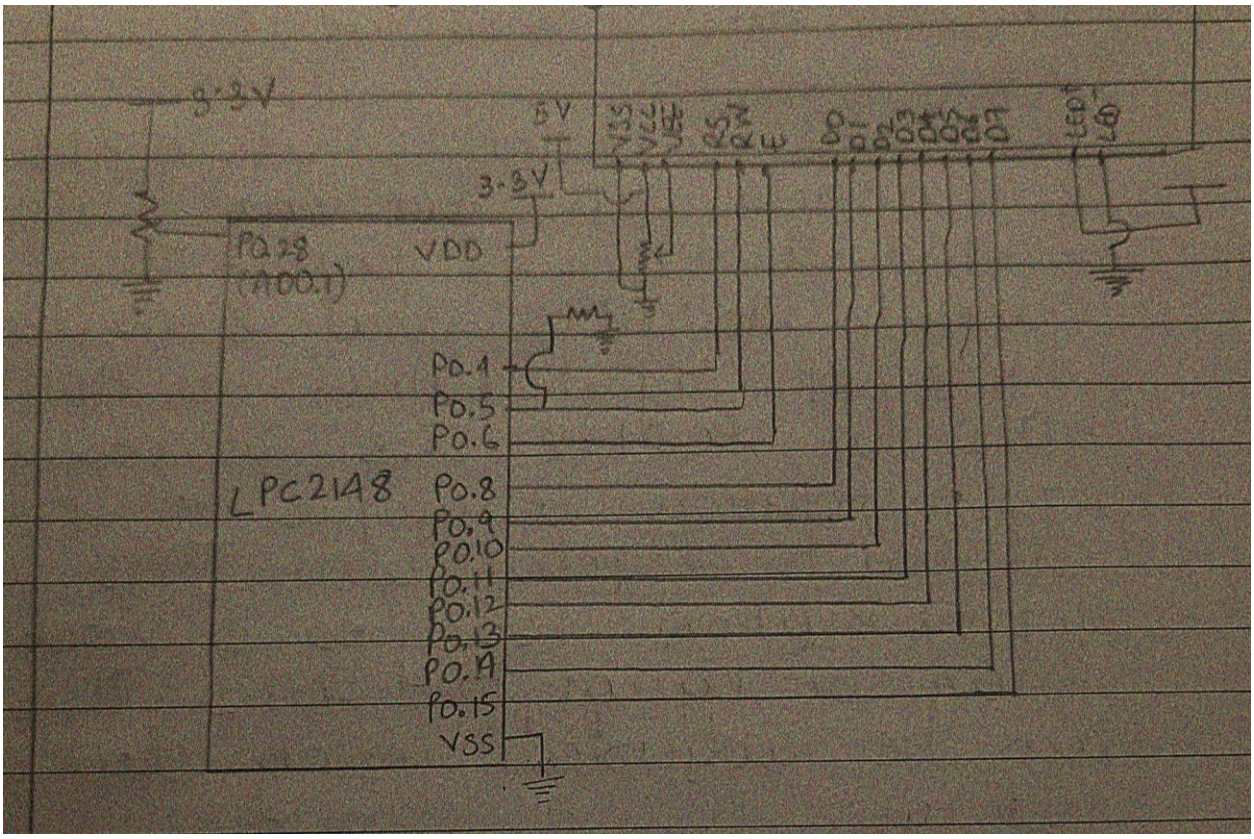
The EPBARM7 has 3 pin Relimate connector J4, which brings out ADC-0 module signals. Two channels of the ADC are available on this connector.

For ADC Connection:

Pin No.	PL3 Connector of Ask25	J1 Connector of EPBARM7
1		P0.0
2		P0.1
3		P0.2
4		P0.3
5		P0.4
6	RS	P0.5
7	R/W*	P0.6
8	E	P0.7
9		NC
10	D0	P0.8
11	D1	P0.9
12	D2	P0.10
13	D3	P0.11
14	D4	P0.12
15	D5	P0.13
16	D6	P0.14
17	D7	P0.15
18		3.3V
19	5V	5.0V

20	GROUND	GND

Interfacing Diagram:



Observation Table:

A: ADC with no interrupt

Vin	Dout HEX	Dout DEC	Dout calculated	%Error
0.71V	0DF	223	220.315	1.2
1.10V	15A	346	341.33	1.3
2.442V	2F8	760	757.76	0.29

3.006V	**3A6**	**934**	**932.77**	**0.13**
3.29V	**3FF**	**1023**	**1020.896**	**0.205**

B: ADC with interrupt

Vin	Dout HEX	Dout DEC	Dout calculated	%Error
0.475V	**095**	**149**	**147.39**	**1.08**
1.422V	**1BA**	**442**	**441.25**	**0.16**
2.323V	**2D3**	**723**	**720.83**	**0.3**
3.228V	**3EC**	**1004**	**1001.66**	**0.23**
3.298V	**3FF**	**1023**	**1023.38**	**0.37**

Program:

A: ADC without interrupt

```
#include "LPC214x.h"
#define RS (1<<5)
#define RW (1<<6)
#define EN (1<<7)
void delay_fv(unsigned int x,int y);
void lcd_display(unsigned int x);
void cmd (unsigned int x);
void lcd_ini(void);
void lcd_str(unsigned char x[]);
void DisplayHexLcd(int LcdData);
unsigned int AdcData =0;
```

```
int main()
{
 int Result;

 PINSEL0 = 0x00000000;
 PINSEL1 = 0x01000000;
 VPBDIV = 0x01;

 IODIR0 = 0X0000FFE0;
 AD0CR = 0X01200402;
 AD0GDR = 0X01000000;
 lcd_ini();
 cmd(0x80);
 lcd_str ("ADC o/p:");
 cmd (0x8B);
 lcd_display ('H');

 AD0CR = 0x01200402;
 AD0CR = AD0CR|(1<<24);
 while((AD0GDR & 0x80000000)!=0x80000000);
 Result = ((AD0GDR & 0x0000FFC0)>>6);
 DisplayHexLcd(Result);
 while(1);
```

```
}
void lcd_display(unsigned int x)
{
 IOCLR0 = 0x0000FF00;
 x =(x<<8);
 IOSET0 = x;
 IOSET0 =RS;
 IOCLR0 =RW;
 IOSET0 =EN;
 delay_fv(100,10);
 IOCLR0 =EN;
}
void delay_fv(unsigned int x,int y)
{
 unsigned int i,j;
 for(i=0;i<x;i++)
 for(j=0;j<y;j++);
}
void cmd (unsigned int x)
{
 IOCLR0 = 0x0000FF00;
 x =(x<<8);
 IOSET0 = x;
 IOCLR0 =RS;
 IOCLR0 =RW;
 IOSET0 =EN;
```

```
 delay_fv(100,10);
 IOCLR0 =EN;
}

void lcd_ini()
{
 cmd(0X38);
 cmd(0X0e);
 cmd(0X01);
 cmd(0X06);
}

void lcd_str(unsigned char x[])
{
 int i;
 for(i=0;x[i]!='\0';i++)
 {
  lcd_display(x[i]);
 }
}

void DisplayHexLcd(int LcdData)
{
    unsigned char Character[17]={"0123456789ABCDEF"};
    unsigned int DivValue=0x100,BaseValue=0x10;
    char j = 0;
```

```
  while(DivValue)
  {
 cmd(0x88+j++);
   lcd_display(Character[LcdData/DivValue]);
      LcdData%=DivValue;
      DivValue/=BaseValue;

   }
}
```

B: ADC with interrupt

```
#include "LPC214x.h"
#include "VIClowlevel.h"

#define RS (1<<5)
#define RW (1<<6)
#define EN (1<<7)
void delay_fv(unsigned int x,int y);
void lcd_display(unsigned int x);
void cmd (unsigned int x);
void lcd_ini(void);
void lcd_str(unsigned char x[]);
//void lcd_str(unsigned char *x);
void DisplayHexLcd(int LcdData);
unsigned int AdcData =0;
```

```
void __attribute__((interrupt("IRQ"))) ADC0_routine(void)
{
        ISR_ENTRY();
        AdcData = (AD0GDR & 0x0000FFC0)>>6;
        DisplayHexLcd(AdcData);
        AD0INTEN = 0x00;
        VICVectAddr = 0x00;
        ISR_EXIT();
}

int main()
{
        int Result;

        PINSEL0 = 0x00000000;
//      PINSEL1 = 0x01000000;
        VPBDIV = 0x01;

        IODIR0 = 0XFFE0;
//      AD0CR = 0X01200402;
//      AD0GDR = 0X01000000;
        lcd_ini();
        cmd(0x80);
        lcd_str ("ADC o/p:");
        cmd (0x8B);
```

```
        lcd_display ('H');

//      AD0CR = 0x01200402;
//      AD0CR = AD0CR|(1<<24);
        while(1)
        {
                VICIntEnable |= (0x01 << 18);
                VICVectAddr0 = (unsigned) ADC0_routine;
                VICVectCntl0 = (0x32);
                enableIRQ();
                PINSEL1 = 0x1000000;
                VPBDIV = 0x01;
        AD0CR = 0X01200402;
           ADGSR = 0X01000000;
                AD0INTEN = 0x02;

        }
        Result = ((AD0GDR & 0x0000FFC0)>>6);
        DisplayHexLcd(Result);
        while(1);

}
void lcd_display(unsigned int x)
{
        IOCLR0 = 0x0000FF00;
        x =(x<<8);
```

```
        IOSET0 = x;
        IOSET0 =RS;
        IOCLR0 =RW;
        IOSET0 =EN;
        delay_fv(100,10);
        IOCLR0 =EN;
}
void delay_fv(unsigned int x,int y)
{
        unsigned int i,j;
        for(i=0;i<x;i++)
        for(j=0;j<y;j++);
}
void cmd (unsigned int x)
{
        IOCLR0 = 0x0000FF00;
        x =(x<<8);
        IOSET0 = x;
        IOCLR0 =RS;
        IOCLR0 =RW;
        IOSET0 =EN;
        delay_fv(100,10);
        IOCLR0 =EN;
}

void lcd_ini()
```

```
{
        cmd(0X38);
        cmd(0X0e);
        cmd(0X01);
        cmd(0X06);
}

void lcd_str(unsigned char x[])
{
        int i;
        for(i=0;x[i]!='\0';i++)
        {
                lcd_display(x[i]);
        }
}

void DisplayHexLcd(int LcdData)
{
   unsigned char Character[17]={"0123456789ABCDEF"};
   unsigned int DivValue=0x100,BaseValue=0x10;
   char j = 0;
   while(DivValue)
   {
                cmd(0x88+j++);
        lcd_display(Character[LcdData/DivValue]);
      LcdData%=DivValue;
```

```
        DivValue/=BaseValue;


    }
}
```

Result:

The digital value corresponding to analog input should be displayed on LCD display.

Conclusion:

In this experiment, two embedded C programmes for ADC with and Without were performed and outputs were displayed on LCD. Different registers used for ADC were studied along with the understanding of the difference between polling and interrupt were carried out.

Name :- Sateej Zunjarrao Div :- B
Subject :- ESP Batch :- B3
Roll no :- PB-30

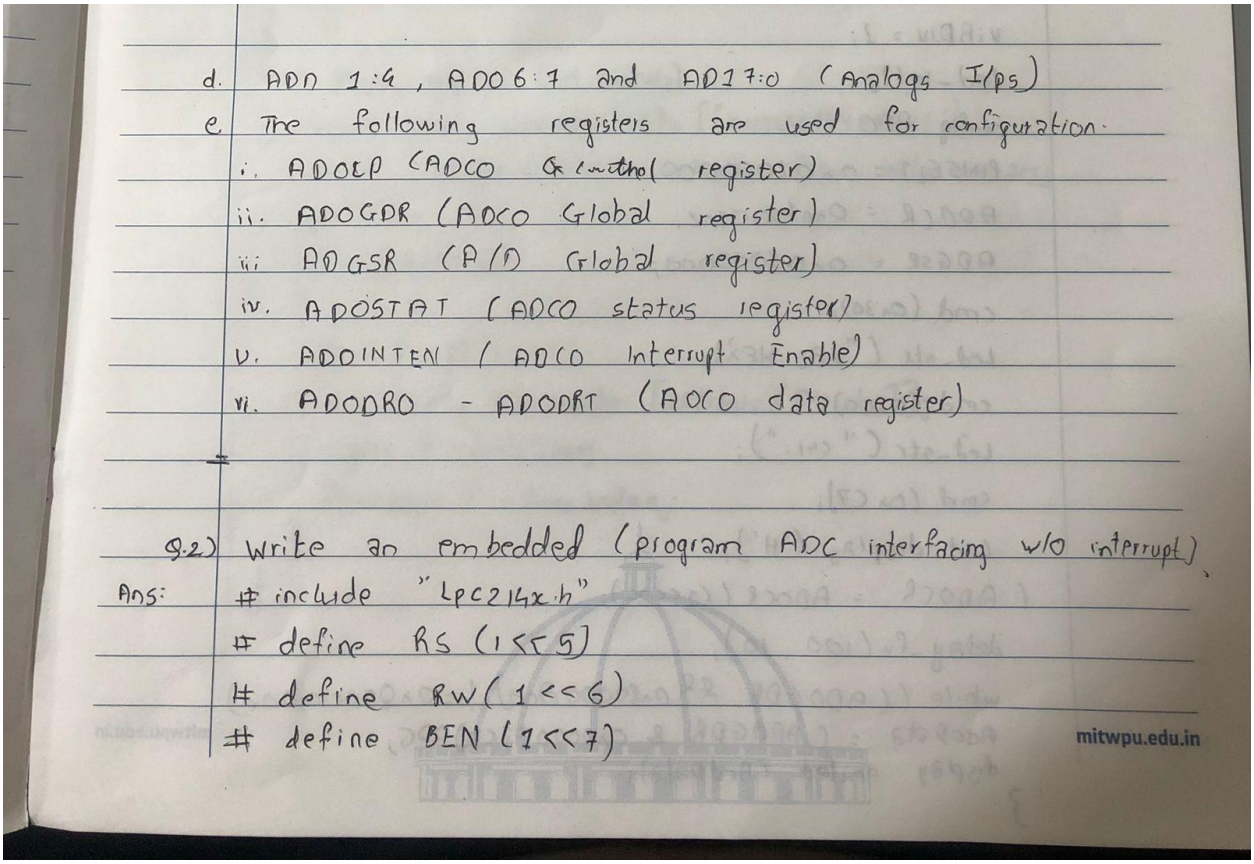
Lab-4 - Assignment

Q.1 Describe the features of ADC in LPC2148.

a. LPC2148 has 2 built in 10-bit ADC i.e ADC0 and ADC1

b. ADC0 has 6 channel while ADC1 has 8 channel. Hence we can connect 6 distinct types of input analog signal to ADC0 and 8 distinct types of input signal to ADC1.

c. ADC in LPC2149 use successive approximation requires to connect analog to digital form. This successive approximation requires a clock less than or equal to 4.5 MHz. This clock can be adjusted using clock divides settings.



d. AD0 1:4, AD0 6:7 and AD1 7:0 (Analogs I/ps)

e. The following registers are used for configuration.

i. AD0CR (ADC0 Control register)

ii. AD0GDR (ADC0 Global register)

iii. AD0GSR (A/D Global register)

iv. AD0STAT (ADC0 status register)

v. AD0INTEN (ADC0 Interrupt Enable)

vi. AD0DR0 - AD0DR7 (ADC0 data register)

Q.2) Write an embedded (program ADC interfacing w/o interrupt).

Ans:

```
#include "Lpc214x.h"
#define RS (1<<5)
#define RW (1<<6)
#define EN (1<<7)
```

mitwpu.edu.in

ASSIGNMENT / TEST Page No.

```
void delay_fu (unsigned int x, int y);
void delay_display (unsigned int x);
void cmd (unsigned int x);
void Lcd_in (void);
void Lcd_str (unsigned char x);
void Led_str (unsigned char x);
void Display Hexled ;
unsigned int AcdData = 0;

int main()
{
  PINSEL0 = 0x00;
  FIODIR0 = 0xFFFFFFFF;
  VIBDIV = 1;
  Led_in();
  config ADC
  PINSEL1 = 0x01000000;
  AD0CR = 0x01200702;
  AD0GSR = 0x01000000;
  cmd (0x30);
  Led_str ("VAL HEX");
  cmd (0xC0);
  Led_str ("C41:");
  cmd (0xC7);
  Lcd_display ('H');
{ AD0CR = AD0CR | (1<<24);
  delay_fu (100, 10);
  while ((AD0GDR && 0x80000000) != 0x80000000)
  AcdData = (AD0GDR & 0x0000FFC0) >>6;
  display Hexled (AcdData);
}
```

Since 1983
Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

ASSIGNMENT / TEST Page No.

```
while(1);
}
void delay_fv(unsigned int x, int y);
{
  unsigned int i, j;
  for(i=0; i<x; i++)
    for(j=0; j<y; j++);
}
void lcd_display (unsigned int x)
{ x = 0x41 = 01000001;
  IOCLR0 = 0x0000FF00;
  IOSET0 = RS;
  IOSET0 = RW;
}
void Display HexLcd (int lcdData)
{ unsigned char character[17] = {"0123456789ABCDEF"};
  unsigned int Div value = 0x100, Base value = 0x10;
  char j = 0;
  while (DivValue)
  {
    cmd (0x88 + j++);
    lcd_display (character [LCDdata (Divvalue]];
    LCDAd % = DivValue;
    Divvalue / = Base Value;
  }
}
```