**Experiment-1**

**Write and execute C program to blink LEDs using software delay routine in LPC2148 kit**

**Aim:** To write and execute a C program to blink LEDs using software delay routine in LPC 2148 kit

**Apparatus Required:**

Keil uVision5 Software

Philips Flah Programmer

LPC 2148 kit

**Program:**

#include "lpc214x.h"

void delay (unsigned int k);

void main(void)

{

IODIR0 = 0xFFFFFFFF; //Configure Port0 as output Port

PINSEL0 = 0; //Configure Port0 as General Purpose IO

while(1)

{

IOSET0 = 0x0000ff00; //Set P0.15-P0.8 to '1'

delay(1000); //1 sec Delay

IOCLR0 = 0x0000ff00; //Set P0.15-P0.8 to '0'

delay(1000); //1 Sec Delay

}

}

//Delay Program

//Input - delay value in milli seconds

void delay(unsigned int k)

{

unsigned int i,j;

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

for(i = 0; i<=800; i++);

}

**Output:** LEDs P0.15-P0.8 are blinking

**Result:**

Thus the C program to blink LEDs using software delay routine was written and executed in LPC 2148 kit  
  
  
**Experiment-2  
  
Write and execute C program to display a number in seven segment LED in LPC2148 kit**

**Aim:** To write and execute C program to display a number in seven segment LED in LPC2148 kit

**Apparatus Required:**

Keil uVision5 Software

Philips Flah Programmer

LPC 2148 kit

**Program:**

//SEVEN SEGMENT LED DISPLAY INTERFACE IN C

/\* Program to Count 0-9 and Display it in 7 segment Display (MUX) DS4

\* Display Select DS3 ==> "P0.13" Enable --> '0', Disable --> '1'

\* Display Select DS4 ==> "P0.12" Enable --> '0', Disable --> '1'

\*/

/\* Segment Connection Display 1 & 2 Enable --> '1', Disable --> '0'

\*--------------------------------------------------------------------

\* MSB LSB

\* Dp G F E D C B A

\* P0.23 P0.22 P0.21 P0.20 P0.19 P0.18 P0.17 P0.16

\* 0 0 0 0 0 1 1 0 --> 6 => '1'

\*---------------------------------------------------------------------\*/

#include <LPC214X.H>

#define DS3 1<<13 // P0.13

#define DS4 1<<12 // P0.12

#define SEG\_CODE 0xFF<<16 // Segment Data from P0.16 to P0.23

unsigned char const seg\_dat[]={0x3F, 0x6, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x7, 0x7F, 0x67};

void delayms(int n)

{

int i,j;

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

{for(j=0;j<5035;j++) //5035 for 60Mhz \*\* 1007 for 12Mhz

{;}

}

}

int main (void)

{

unsigned char count;

PINSEL0 = 0; // Configure Port0 as General Purpose IO => P0.0 to P0.15

PINSEL1 = 0; // Configure Port0 as General Purpose IO => P0.16 to P0.31

IODIR0 = SEG\_CODE | DS3 | DS4; //Configure Segment data & Select signal as output

IOSET0 = SEG\_CODE | DS3 ; //Disable DS3 display

IOCLR0 = DS4; //Enable DS4 Display

count = 0; //Initialize Count

//Display Count value

IOCLR0 = SEG\_CODE;

IOSET0 = seg\_dat[count]<<16;

while(1)

{

delayms(1000); //1 sec delay

count++; //Increment count

if(count>9) count=0; //Limit 0-9

//Display Count value

IOCLR0 = SEG\_CODE;

IOSET0 = seg\_dat[count]<<16;

}

}

**Output:** 7-Segment display counting from 0 to 9

**Result:**

Thus C program, was written and executed to display a number in seven segment LED in LPC2148 kit

1. **Displaying Decimal Numbers (0-9)**

#include <LPC214X.H>

#define DS3 1<<13 // P0.13

#define DS4 1<<12 // P0.12

#define SEG\_CODE 0xFF<<16 // Segment Data from P0.16 to P0.23

unsigned char const seg\_decimal[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F};

void delayms(int n) {

int i, j;

for(i = 0; i < n; i++) {

for(j = 0; j < 5035; j++) {;} // Delay for 60 MHz clock

}

}

int main(void) {

int count;

PINSEL0 = 0; // Configure Port0 as General Purpose IO => P0.0 to P0.15

PINSEL1 = 0; // Configure Port0 as General Purpose IO => P0.16 to P0.31

IODIR0 = SEG\_CODE | DS3 | DS4; // Configure Segment data & Select signal as output

IOSET0 = SEG\_CODE | DS3; // Disable DS3 display

IOCLR0 = DS4; // Enable DS4 Display

while (1) {

for (count = 0; count < 10; count++) {

IOCLR0 = SEG\_CODE;

IOSET0 = seg\_decimal[count] << 16;

delayms(1000); // 1 sec delay

}

}

}

**b.display a hexadecimal numbers in seven segment LED in LPC2148 kit**

#include <LPC214X.H>

#define DS3 1<<13 // P0.13

#define DS4 1<<12 // P0.12

#define SEG\_CODE 0xFF<<16 // Segment Data from P0.16 to P0.23

unsigned char const seg\_hexadecimal[] = {

0x3F, // '0'

0x06, // '1'

0x5B, // '2'

0x4F, // '3'

0x66, // '4'

0x6D, // '5'

0x7D, // '6'

0x07, // '7'

0x7F, // '8'

0x6F, // '9'

0x77, // 'A'

0x7C, // 'b'

0x39, // 'C'

0x5E, // 'd'

0x79, // 'E'

0x71 // 'F'

};

void delayms(int n) {

int i, j;

for(i = 0; i < n; i++) {

for(j = 0; j < 5035; j++) {;} // Delay for 60 MHz clock

}

}

int main(void) {

int count;

PINSEL0 = 0; // Configure Port0 as General Purpose IO => P0.0 to P0.15

PINSEL1 = 0; // Configure Port0 as General Purpose IO => P0.16 to P0.31

IODIR0 = SEG\_CODE | DS3 | DS4; // Configure Segment data & Select signal as output

IOSET0 = SEG\_CODE | DS3; // Disable DS3 display

IOCLR0 = DS4; // Enable DS4 Display

while (1) {

for (count = 0; count < 16; count++) {

IOCLR0 = SEG\_CODE;

IOSET0 = seg\_hexadecimal[count] << 16;

delayms(1000); // 1 sec delay

}

}

}  
  
**c.Alphabets**  
  
#include <LPC214X.H>

#define DS3 (1 << 13) // P0.13 - Display Select 3

#define DS4 (1 << 12) // P0.12 - Display Select 4

#define SEG\_CODE (0xFF << 16) // P0.16 to P0.23 - 7-segment control

// Segment data for displaying the characters 'S', 'A', 'V', 'E', 'E', 'T', 'H', 'A'

unsigned char const seg\_dat[] = {

0x6D, // S

0x76, // A

0x37, // V

0x79, // E

0x79, // E

0x74, // T

0x76, // H

0x76 // A

};

void delayms(int n) {

int i, j;

for (i = 0; i < n; i++) {

for (j = 0; j < 5035; j++) { // Adjust based on the microcontroller clock (60 MHz)

// Empty loop to create a delay

}

}

}

int main(void) {

unsigned char count; // Variable to hold the current character index

// Configure the GPIO pins for 7-segment display

PINSEL0 = 0; // Set Port0 pins as General Purpose I/O (GPIO) for 0-15

PINSEL1 = 0; // Set Port0 pins as General Purpose I/O (GPIO) for 16-31

// Configure the Direction for Port0 (P0.16 to P0.23 for segments, P0.12, P0.13 for display selects)

IODIR0 = SEG\_CODE | DS3 | DS4;

// Initially disable both displays

IOSET0 = SEG\_CODE | DS3; // Disable DS3

IOCLR0 = DS4; // Enable DS4

count = 0; // Start with character 'S'

// Main loop to display "SAVEETHA" on the 7-segment displays

while (1) {

// Display the current character on the 7-segment displays

IOCLR0 = SEG\_CODE; // Clear previous segment data

IOSET0 = seg\_dat[count] << 16; // Set the current character for DS4

delayms(1000); // 1-second delay to keep the display steady

// Increment the count to show the next character in "SAVEETHA"

count++;

if (count > 7) {

count = 0; // Reset to 0 after displaying "A"

}

// Switch display (multiplexing DS3 and DS4)

IOCLR0 = DS4; // Disable DS4

IOSET0 = DS3; // Enable DS3

delayms(500); // Shorter delay for DS3 to give it time to display

IOCLR0 = DS3; // Disable DS3

IOSET0 = DS4; // Enable DS4

}

}  
  
**Exp.**

**Write and execute C program for accessing an internal ADC and display the binary output in LEDS in LPC2148 kit**

**Date:**

**Aim:** To write and execute C program for accessing an internal ADC and display the binary output in LEDS in LPC2148 kit.

**Apparatus Required:**

Keil uVision5 Software

Philips Flah Programmer

LPC 2148 kit

**Program:**

#include <LPC214X.H>

#define LEDS 0xFF<<8 //LED => P0.8 to P0.15

/////////////////////////////////////////

/\*--- ADC Signal Declaration \*/

/////////////////////////////////////////

#define AD0\_1 1<< 24

#define CLK\_DIV 1<<8

#define PDN 1<<21

#define SOC 1<<24

#define BURST 1<<16

#define DONE 1<<31

/\*-----------------------------------------------------------\*/

//Delay Program

//Input - delay value in milli seconds

void delay(unsigned int k)

{

unsigned int i,j;

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

for(i = 0; i<=800; i++);

}

/\*-----------------------------------------------------------\*/

void adc\_init()

{

unsigned long int ADC\_CH;

ADC\_CH = 0 | 1 << 1; //Channel AD0.1

AD0CR = SOC | PDN | CLK\_DIV | ADC\_CH | BURST ;

}

/\*-----------------------------------------------------------\*/

unsigned int adc\_read( unsigned char channel)

{

unsigned int aval;

unsigned long int val;

if (channel == 1) val = AD0DR1;

else if (channel == 2) val = AD0DR2;

else if (channel == 3) val = AD0DR3;

val = val >> 6;

val = val & 0x3FF;

aval = val;

return (aval);

}

/\*-----------------------------------------------------------\*/

//////////////////////////

/\*----Main Program------\*/

//////////////////////////

int main(void)

{

unsigned int tp1;

IODIR0 = LEDS; //Configure Port0 as output Port

PINSEL0 = 0; //Configure Port0 as General Purpose IO

PINSEL1 = 0 | AD0\_1; // Enable AD0.1

adc\_init(); //Initialise on-chip ADC

do

{ tp1 = adc\_read(1); // Channel AD0 0.1

tp1 = tp1 >> 2; // ADC 10 bit But LED 8bit, Truncate lsb 2 bits

IOSET0 = LEDS; //Switch OFF all LEDS

IOCLR0 = tp1 << 8; //Set VAlue

delay(1000);

}while(1);

}

**Output:** The Potentiometer knob was adjusted to generate Analog input and Digital display is observed

**Result:**

Thus C program was Written and executed for accessing an internal ADC and display the binary output in LEDS in LPC2148 kit.

**TRIANGLE WAVEFORM GENERATION WITH 50% DUTY CYCLE USING 10 BIT DAC**

**AIM:**

To write the C program to generate a triangular wave form with 50% duty cycle using

internal 10-bit DAC using LPC2148 ARM Micro controller.

**APPARATUS REQUIRED:**

1. LPC 2148 ARM Microcontroller Development board.
2. Keil µVision version 5
3. Flash Magic

**PROGRAM:**

#include "lpc214x.h"

// PLL Register Values

#define MSEL 4 // M-1 since M=5

#define PSEL (1 << 5) // P-1 since P=2

#define AOUT (1 << 19) // PINSEL1 value for AOUT config.

// Delay function (creates delay in milliseconds)

void delayms(int n) {

int i, j;

for (i = 0; i < n; i++) {

for (j = 0; j < 5035; j++) { // 5035 for 60MHz, 1007 for 12MHz

;

}

}

}

/\* Function to Set PLL0 so CPU Clock = 60MHz, PCLK = 15MHz \*/

void clock\_select(void) {

// Fosc = 12MHz

// Select CCLK = 60MHz & Fcco = 240MHz

PLL0CFG = PSEL | MSEL;

// PLL Feed Sequence

PLL0FEED = 0xAA;

PLL0FEED = 0x55;

// Enable PLL0

PLL0CON = 3;

// PLL Feed Sequence

PLL0FEED = 0xAA;

PLL0FEED = 0x55;

// Set VPBDIV for PCLK = CCLK / 4, so PCLK = 15MHz

VPBDIV = 0;

}

/\* Function to Generate a Triangle Wave using the DAC \*/

void generate\_triangle\_wave(void) {

unsigned int i;

// Incrementing part of the triangle

for (i = 0; i < 1023; i++) {

DACR = (i << 6); // Write to DAC (10-bit left justified)

delayms(1); // Delay to control waveform frequency

}

// Decrementing part of the triangle

for (i = 1023; i > 0; i--) {

DACR = (i << 6); // Write to DAC

delayms(1); // Delay to control waveform frequency

}

}

int main(void) {

// CPU Clock Configuration

clock\_select();

// Configure Port0.0 to P0.15 as General Purpose IO

PINSEL0 = 0;

// Configure Port0.25 as Analog Output PIN

PINSEL1 = AOUT;

while (1) {

generate\_triangle\_wave(); // Generate the triangle wave continuously

}

return 0;

}

**SQUARE WAVEFORM GENERATION WITH 50% DUTY CYCLE USING 10 BIT DAC**

**AIM:**

To write the C program to generate a square wave form with 50% duty cycle using internal 10-bit DAC using LPC2148 ARM Micro controller.

**APPARATUS REQUIRED:**

1. LPC 2148 ARM Microcontroller Development board.
2. Keil µVision version 5
3. Flash Magic

**PROGRAM:**

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

\* Program to Generate Square Wave at DAC Output

\* Square wave Frequency fixed to 500Hz

\* DAC Ouput = (VALUE / 1024) \* Vref => DAC Resolution = 10 Bits

\* Vref = 2.2V fixed in trainer kit

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

#include "lpc214x.h"

//PLL Register Values

#define MSEL 4 //M-1 since M=5

#define PSEL 1<<5 //P-1 since P=2

#define AOUT 1<<19 // PINSEL1 value for AOUT config.

void delayms(int n)

{

int i,j;

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

{for(j=0;j<5035;j++) //5035 for 60Mhz \*\* 1007 for 12Mhz

{;}

}

}

/-------------------------------------------------------/

/-------------------------------------------------------/

/\* Function Sets PLL0 So CPU Clock=60Mhz PCLK=15Mhz \*/

/-------------------------------------------------------/

void clock\_select(void)

{

//Fosc = 12Mhz

//Select CCLK = 60Mhz & Fcco = 240Mhz

PLL0CFG = PSEL | MSEL;

//PLL FEED

PLL0FEED=0xAA;

PLL0FEED=0x55;

PLL0CON = 3; //Enable PLL0

//PLL FEED

PLL0FEED=0xAA;

PLL0FEED=0x55;

VPBDIV = 0; //PCLK = CCLK/4 So PCLK = 15Mhz

}

/-------------------------------------------------------/

int main(void)

{

clock\_select(); //CPU Clock Configuration

PINSEL0 = 0; //Configure Port0.0 to P0.15 as General Purose IO

PINSEL1 = AOUT; //Configure Port0.25 as Analog Output PIN

while(1)

{

DACR = 0; //DAC ouput = 0V

delayms(1); //1 msec Delay

DACR = 0x3FF << 6; //DAC ouput = 2.2V

delayms(1); //1 mSec Delay

}

}

/-------------------------------------------------------/