**Ayush Goyal**

**190905522 CSE D 62**

**DAA Lab-6 (Week 6) – Interfacing LED to ARM Microcontroller**

**Solved Exercise: Write a program to turn on/off the LEDs serially.**

**CODE:**

#include<LPC17xx.h>

unsigned int i,j;

unsigned long LED = 0x00000010;

int main(void)

{

    SystemInit();

    SystemCoreClockUpdate();

    LPC\_PINCON->PINSEL0 &= 0xFF0000FF;

    LPC\_GPIO0->FIODIR |= 0x00000FF0;

    while(1){

        LED = 0x00000010;

        for(i=1;i<9;i++)

        {

            LPC\_GPIO0->FIOSET = LED;

            for(j=0;j<10000;j++);

            LED <<= 1; *//Shift LED to the left by one unit*

        }

        LED = 0x00000010;

        for(i=1;i<9;i++)

        {

            LPC\_GPIO0->FIOCLR = LED;

            for(j=0;j<10000;j++);

            LED<<=1;

        }

    }

}

**OUTPUT:**

After the first LED is turned on:

Graphical user interface, text, application

Description automatically generated

After the 8th LED is turned on:

Graphical user interface, text, application

Description automatically generated

After 1st and 2nd LED is turned off:

Graphical user interface, text, application, email

Description automatically generated

After 8 LEDs are turned off:

Graphical user interface, text, application

Description automatically generated

Thus, we can see that this while loop iterates continuously serially turning on and off the LEDs.

1. **Write a C program to display an 8-bit binary up counter on the LEDs.**

**CODE:**

#include<LPC17xx.h>

unsigned int i,j;

int main(void)

{

    SystemInit();

    SystemCoreClockUpdate();

    LPC\_PINCON->PINSEL0 &= 0xFF0000FF;

    LPC\_GPIO0->FIODIR |= 0x00000FF0;*//configuring as output pins*

*//using same pins P0.4 to P0.11 as GPIO functions*

    while(1){

        for(i=0;i<256;i++)

        {

            LPC\_GPIO0->FIOPIN = i<<4;

            for(j=0;j<10;j++);

        }

    }

}

**OUTPUT:**

After 1st count it should be just 4th bit on:

Graphical user interface, text, application

Description automatically generated

After 17 counts or iterations, the binary counter should be 10001:

Graphical user interface, text, application

Description automatically generated

After 255 counts, the binary up counter should be at value 11111111:

Graphical user interface, text, application

Description automatically generated

Thus, we can see the functioning of an 8-bit binary up counter on the LEDs.

1. **Write a C program to read a key and display an 8-bit up/down counter on the LEDs.**

**Hint: Use key SW2(if SW2=1, up counter else down counter), which is available at CNB1 pin 7. Connect CNB1 to any controller connector like CNB, CNC, etc. Configure the corresponding port pin as GPIO using corresponding PINSEL register and as input pin using corresponding FIODIR register.**

**CODE:**

**OUTPUT:**

1. **Write a program to simulate an 8-bit ring counter with key press (SW2).**

**CODE:**

**OUTPUT:**

**THE END**