Lambton College
In Toronto

Course Number	ESE3025
Section Number	**1**
Course Title	Embedded Real Time Operating Systems
Semester/Year	3/2019

Instructor	**Takis Zourntos**

Assignment No.	**1**

Submission Date	**March 15, 2019**
Due Date	**March 15, 2019**

Student Name	**Akshit Patel**	**Jaivik Patel**	**Tejas Shingala**
Student ID	C0724035	C0732075	C0732079
Signature*	**AP**	**JP**	**TS**

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties.*

Akshit Patel(C0724035)

Jaivik Patel(C0732075)

Tejas Shingala(C0732079)

Assignment -1

Write a basic embedded-C program to get the LEDs to blink in the pattern shown in class (sequentially, Red, then Green, then Blue, with no color mixing).

MCUXpresso IDE helps users to create easy to use Eclipse based environment. It includes Arm Cortex-M cores, LPC and kinrtis microcontroller. It offers advanced compiling, debugging and editing features. It also provides code trace, debugging views, multicore debugging and integrated configuration tools.

Features of MCUXpresso

- It provides free toolchain to developers without any restriction on code or the size of debugging.
- It provides powerful interface with power measurement, profiling on supported boards, multicore capable debugger and many more.
- It provides complete C/C++ integrated development environment

Advance project wizard:

- Trace functionality
- Link Server Power Measurement
- MCUXpresso Configuration Tools

Supported Debug Probes:

There is built in support for three debug solutstions.

- Native LinkServer
- P&E Micro
- Segger J-Link

LPC1769

LPCXpresso is a new, cost effective development program provide by NXP which support NXP's Arm-based microcontrollers. It is a good platform for embedded engineers where they can develop any applications from starting generation to ending product.

LPC1769 is a populated board having lots of in-built features that enables us to work it with ease.

- It has CORTEX-M3 Arm based micro-controller.
- Tricolor LED – Red, Green, and Blue.
- Reset and ISP boot mode buttons.
- For external debug probe, it has 10 pin SWD connector.
- Integrated CMSIS-DAP debug probe.
- Ethernet 10/100 PHY
- Memory- 64kB RAM, 512kB ROM
- 4 x 32-bit Timer

Program – LED Blinking

Mostly LPC1769 pins are multiplexed to support more than one function. Every pin has a minimum of one and maximum of four functions. Here, we perform a LED blinking program in which we have to blink LED's sequentially Red, then Green, then Blue, with no color mixing.

In this program, we used LPC1769 user defined library. With the help of two integers k,j, we create a loop in which we used bunch of delays to turn on and off the led and to generate a gap between the LED's.

```c
/*
==============================================================================
=
 Name        : led.c
 Author      : $(TEJAS SHINGALA)/ AKSHIT PATEL/ JAIVIK PATEL
 Version     : 1.0
 Copyright   : $(copyright)
 Description : main definition
==============================================================================
=
*/

#if defined (__USE_LPCOPEN)
#if defined(NO_BOARD_LIB)
#include "chip.h"
#else
#include "board.h"
#endif
#endif

#include <cr_section_macros.h>

// TODO: insert other include files here

// TODO: insert other definitions and declarations here

int main(void) {

#if defined (__USE_LPCOPEN)
    // Read clock settings and update SystemCoreClock variable
    SystemCoreClockUpdate();
#if !defined(NO_BOARD_LIB)
    // Set up and initialize all required blocks and
    // functions related to the board hardware
    Board_Init();
    // Set the LED to the state of "On"
    Board_LED_Set(0, true);
    //Board_LED_Set(1, true);
   // Board_LED_Set(2, true);
#endif
#endif

    // TODO: insert code here

    // Force the counter to be placed into memory
    volatile static int i = 0 ;
    // Enter an infinite loop, just incrementing a counter
    while(1) {
        int k,j;
                    Board_LED_Set(0,false);
                    Board_LED_Set(1,true);
                    Board_LED_Set(2,true);
                    for(k=0;k<5000;k++)
                    {
                        for(j=0;j<2000;j++)
                    {}
                    }

                            Board_LED_Set(0,true);
                            Board_LED_Set(1,false);
                            Board_LED_Set(2,true);
                            for(k=0;k<5000;k++)
                            {
                                for(j=0;j<2000;j++)
                            {}
                                }
                                        Board_LED_Set(0,true);
                                        Board_LED_Set(1,true);
                                        Board_LED_Set(2,false);
                                        for(k=0;k<5000;k++)
                                        {
                                            for(j=0;j<2000;j++)
                                        {}
                                        }




            i++ ;
    }
    return 0 ;
}

```