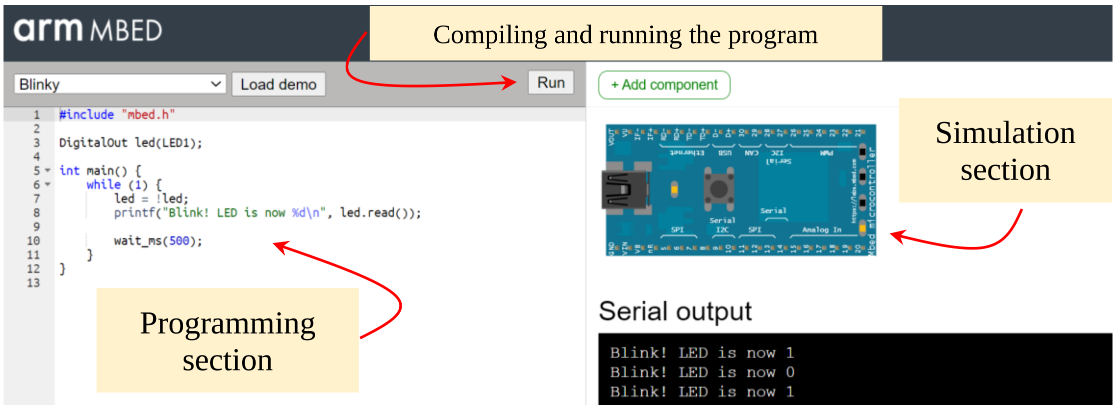
CE323/CE860 Lab Notes

Two assignments are based on the ARM mbed NXP LPC1768 microcontroller. Some of the details about the mbed will be provided in the **lecture slides**, but full details can be found from the website (Website link). The extension. The offline lab has a development board that we will try to develop using an online simulator.
In the following, a brief description on online simulator for the mbed NXP LPC1768, is provided. Some exercises related to the assignment 1 and 2 are then provided to practice. These exercises are helpful for you to better solve the two assignments.

1. The ARM mbed and Online simulator

Each student will develop and online simulation using mbed online simulator. The mbed simulator uses an online compiler to edit and compile the source code. You can add push bottoms, LEDS, LCD, etc. to develop a prototype embedded device. You do not need a user account to access to the online simulator. However, For the resource you will need a user account which is free of charge please check this link 1.
The online simulator is in this link 2. You can select sample programs on the left side menu. After going to the workspace, you can create the new projects, and edit and compile the source codes. After compiling the code, you can check the execution on the righthand simulator. The mbed starts to execute the code sample code shown in the below figure. The details of using the online compiler can be found here.



```
#include "mbed.h"

DigitalOut led(LED1);

int main() {
    while (1) {
        led = !led;
        printf("Blink! LED is now %d\n", led.read());

        wait_ms(500);
    }
}
```

```
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
```

Fig. 1 The mbed online simulator

2. Serial Communication with a PC

Online simulator

There is a direct link between your program and the simulator. You can compile and run your program directly using the run bottom (Fig. 1) on the simulator.

For your information

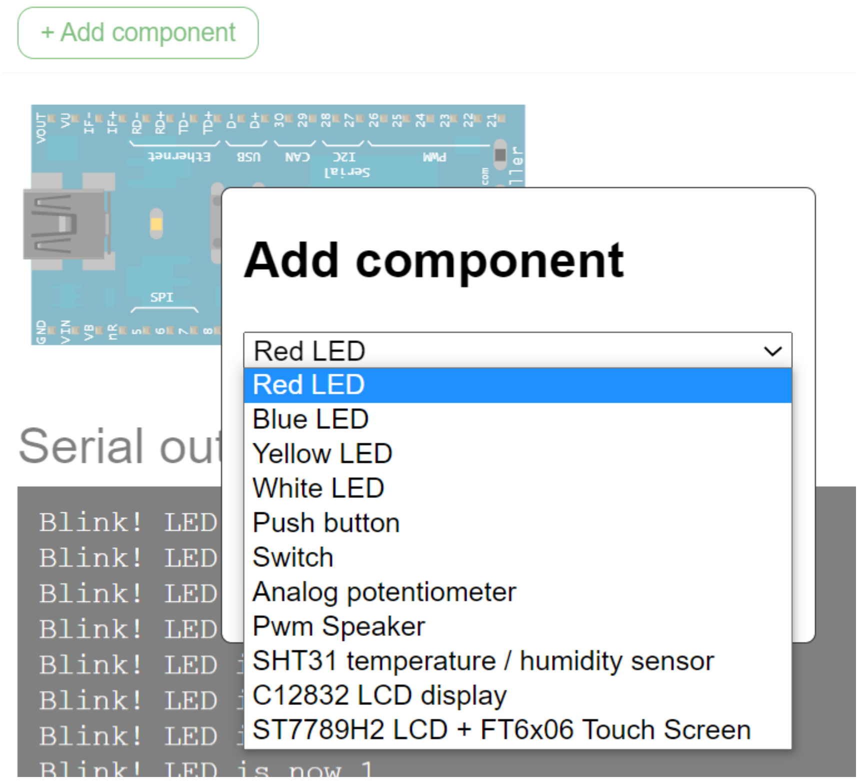
The physical mbed microcontroller can communicate with a host PC through a "USB Virtual Serial Port" over the same USB cable that is used for programming. First install the serial port driver at https://os.mbed.com/handbook/Windows-serial-configuration to your host PC. You can then use Putty or HyperTerminal in host PC to send and receive information between the mbed and the host PC. You need to open Putty or HyperTerminal in host PC and configure it with baud rate: 9600, data: 8 bits, parity: none, stop: 1 bit, and flow control: none.

University of Essex

You need to program the mbed to communicate with Putty or HyperTerminal. It can send the PC keyboard input to the mbed and receive information sent from the mbed. The example is provided in the lab session below.

3. The Extension Board

The physical extension board is developed with additional devices to enhance the system functionality. The main devices included in the extension board are a 16×2 text LCD, a 4×4 keypad, 8 LEDs (2 bits per LED), 8 switches, a temperature sensor, and an IMU sensor.
The online simulator has some of these functionalities and the labs/assignments are designed to cover most of the functionalities provided by the extension board. The list of comments is shown in Fig. 2.



```
Blink! LED
Blink! LED
Blink! LED
Blink! LED
Blink! LED
Blink! LED
Blink! LED
Blink! LED is now 1
```

Fig. 2 The mbed simulator components

4. Lab Exercises

The following information provides a guide to the work to be conducted in lab sessions. This is just a minimum requirement to be completed in each lab session. It is not necessary to strictly follow the guide. However, if your progress is far behind the work required in each session, extra time is needed to complete the assignments on time.

Very Important Note:

Please **make backup of for every step** of your program as the online simulator is not very stable and you might lose all your developments.

Lab Session 3: LEDs and Switches

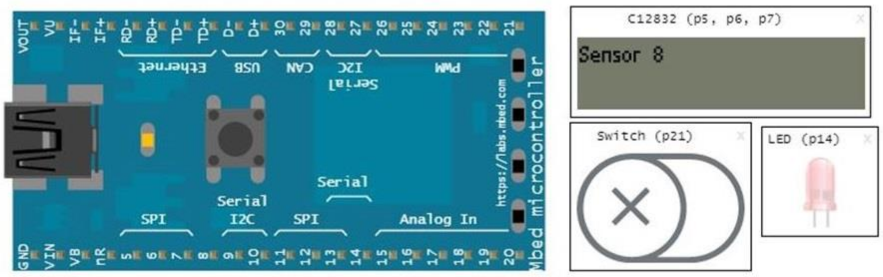
On the physical extension board, there are 8 LEDs and 8 switches. The LEDs are driven by a driver TLC59281, which is connected to a SPI port (P5, P6, P7) on the mbed. Each LED requires 2 bits to control (0 off, 1 red, 2 green, and 3 yellow). On the online simulator, however, the LED driver does not exist, and you can choose LEDs with different collars from the add component pop up window. You can enter 8 different LEDs and connect all of them to the same pin and can make all the LEDs blink (example code below).

```
#include "mbed.h"

DigitalOut led(p5);

int main() {
    while (1) {
        led = !led;
        printf("Blink! LED is now %d\n", led.read());

        wait_ms(500);
    }
}
```

1. You can try to make all the LEDs blink while the first three LEDs are on (you need to simply use different pin).

The 8 switches on the extension board are separated into two groups. Use the switch component on the online simulator and connect 1 LED to the switch (example code below).

```
#include "mbed.h"
#include "C12832.h"
C12832 lcd(SPI_MOSI, SPI_SCK, SPI_MISO, p8, p11); //initiate the LCD
DigitalOut myled1(p13); //initiate the LCD
DigitalIn toggle1(p21); //initiate the switches (for sensor)

int main() {
    while (1) {
      if(toggle1==1)  //activating the sensor
       {
       lcd.locate(0,3);
       lcd.printf("Sensor 1"); //Writing the message on the LCD
       myled1 = 1; //Turning on the LED to show that the sensor is on
       wait_ms(500);
       myled1 = 0;
       }
      // Serial Output
      wait(0.2);
      printf("Switch! is: %d\n", toggle1.read());
      wait(0.2);
    }
}
```



2. Ideally, we want to activate 8 different state of a program using these 8 different switches. Each swich should be connected to a LED and show the activation of one sensor on the LCD.