
Southern Polytechnic College at Kennesaw State University
Electrical Engineering Department
EE 3501 Laboratory Exercise 3: GPIO

Name: _____

Date: _____

Learning Objectives:

- Demonstrate register-based peripheral control.
- Create a C or C++ program to output and input from the ARM processor.

Laboratory procedure

1. Start a new project.
2. Copy and paste the code below into the main.cpp (Note that we will be using the DigitalIn and DigitalOut classes included in the Mbed API to interface with LED1 and the push button of the microcontroller board)

```
#include "mbed.h"

DigitalOut myLed(LED1);
DigitalIn myButton(BUTTON1);

int main() {
    while (1){
        if (myButton ==1){
            wait (0.1);
            if (myButton ==1)
                myLed=1;
            else
                myLed=0;
        }
    }
}
```

3. Compile the program, download it to the board, and run it.
4. Press the blue button on the board to turn the LED on, and release it to turn the LED off.
5. Modify the code in main.cpp so that LED1 alternates between on and off states with each press of the push button. In other words, the push button will toggle the state of LED1.
6. Compile the program, download it to the board, and run it.
7. Press the blue button on the board to turn the LED on, and press it again to turn the LED off.

Questions and Discussion:

- 1) Refer to Chapter 14 of your Embedded Systems textbook and define the term *memory-mapped I/O*. You may also use online resources.
- 2) Refer to the STM32F401xB/C and STM32F401xD/E Reference Manual located in the Resources subfolder of the Laboratories folder in D2L. Within the manual, locate the memory map for the MCU and state the base address of GPIO port A and Port C.
- 3) Refer to the Board pinout on the NUCLEO-F401RE page <https://os.mbed.com/platforms/ST-Nucleo-F401RE/> and state the bit location and GPIO port that corresponds to the LED1 output and the bit location and GPIO port that corresponds to the BUTTON1 input.
- 4) State what you have learned from the experiment.