

# GPIO Lab Assignment

## Objective

Gain experience doing the following:

1. Manipulating a registers in order to access and control physical pins
2. Use implemented driver to sense input signals and control LEDs.

## Assignment

Test your knowledge by doing the following:

### Part 0. Basic GPIO Driver to blink an onboard LED

```
int main()
{
    // 1) Find and choose an onboard LED to manipulate.
    // 2) Use the schematic to figure out which pin it is connected to
    // 3) Use FIODIR to set that pin as an output
    while (true)
    {
        // 4) use FIOCLR to set the pin LOW, turning ON the LED
        LOG_INFO("Turning LED ON!");
        Delay(500); // Delay in milliseconds
        // 5) use FIOSET to set the pin HIGH, turning OFF the LED
        LOG_INFO("Turning LED OFF!");
        Delay(500);
    }
    return 0;}

```

# Part 1. Implement the **LabGPIO** Driver

Using the following class template

1. Implement ALL class methods.
2. All methods must function work as expected of their method name.
3. Must be able to handle pins in port 0, 1, and 2.

```
#pragma once
#include <stdint>

class LabGPIO
{
public:
    enum class Direction : uint8_t
    {
        kInput = 0,
        kOutput = 1
    };
    enum class State : uint8_t
    {
        kLow = 0,
        kHigh = 1
    };
    /// You should not modify any hardware registers at this point
    /// You should store the port and pin using the constructor.
    ///
    /// @param port - port number between 0 and 5
    /// @param pin - pin number between 0 and 32
    constexpr LabGPIO(uint8_t port, uint8_t pin);
    /// Sets this GPIO as an input
    void SetAsInput();
    /// Sets this GPIO as an output
    void SetAsOutput();
    /// Sets this GPIO as an input
    /// @param output - true => output, false => set pin to input
    void SetDirection(Direction direction);
```

```

/// Set voltage of pin to HIGH
void SetHigh();
/// Set voltage of pin to LOW
void SetLow();
/// Set pin state to high or low depending on the input state parameter.
/// Has no effect if the pin is set as "input".
///
/// @param state - State::kHigh => set pin high, State::kLow => set pin low
void set(State state);
/// Should return the state of the pin (input or output, doesn't matter)
///
/// @return level of pin high => true, low => false
State Read();
/// Should return the state of the pin (input or output, doesn't matter)
///
/// @return level of pin high => true, low => false
bool ReadBool();
private:
/// port, pin and any other variables should be placed here.
/// NOTE: Pin state should NEVER be cached! Always check the hardware
/// registers for the actual value of the pin.
};

```

## Part 2. Use Driver for an application

The application is to use all 4 internal buttons to control the on board LEDs above them.

```

int main(void)
{
    LabGpio button0(?, ?);
    LabGpio led0(?, ?);

    // Initialize button and led here
    while(true)
    {
        // Logic to read if button has been RELEASED and if so, TOGGLE LED state;
    }
}

```

```
}  
return 0;}
```

## Requirements:

You **MUST NOT** use any pre-existing library such as a GPIO class for this assignment.

You **MAY USE** LPC40xx.h as it is not a library but a list of registers mapped to the appropriate locations.

The code must read from the internal button. If a button is **RELEASED**, toggle the state of the LED.

Upload only relevant source files into canvas. A good example is: *main.cpp*, *LabGPIO.hpp*, *LabGPIO.cpp*. See Canvas for rubric and grade breakdown.

### Extra Credit

Add a flashy easter egg feature to your assignment, with your new found LED and switch powers! The extra credit is subject to the instructor's, ISA's and TA's discretion about what is worth the extra credit.

Consider using additional switches and/or LEDs.

?

?

Revision #3

Created 5 years ago by [Khalil Estell](#)

Updated 5 years ago by [Khalil Estell](#)

?