

ECE 153B - Lab 1: Handout and Turn-in Questions

Part 1: GPIO Initialization (Masking and Operations)

3.2 Enable the Clock of GPIOA and GPIOC

AHB2ENR Mask = $(1UL \ll 0) | (1UL \ll 2)$

Masking Operation:

`RCC->AHB2ENR |= (1UL << 0) | (1UL << 2);`

3.3 Pin Initialization for Green LED (PA5)

(a) MODER Mask = $\sim(3UL \ll (5 * 2))$

MODER Operation:

`GPIOA->MODER &= $\sim(3UL \ll (5 * 2))$;`

`GPIOA->MODER |= (1UL << $(5 * 2)$);`

(b) OTYPER Mask = $\sim(1UL \ll 5)$

OTYPER Operation:

`GPIOA->OTYPER &= $\sim(1UL \ll 5)$;`

(c) PUPDR Mask = $\sim(3UL \ll (5 * 2))$

PUPDR Operation:

`GPIOA->PUPDR &= $\sim(3UL \ll (5 * 2))$;`

3.4 Pin Initialization for User Button (PC13)

(a) MODER Mask = $\sim(3UL \ll (13 * 2))$

MODER Operation:

`GPIOC->MODER &= $\sim(3UL \ll (13 * 2))$;`

(b) PUPDR Mask = $\sim(3UL \ll (13 * 2))$

PUPDR Operation:

```
GPIOC->PUPDR &= ~(3UL << (13 * 2));
```

Part 2: Turn-in Questions

1. Mask to clear the mode bits of PB11:

```
GPIOB->MODER &= ~(3UL << (11 * 2));
```

2. Set PC7 to output mode (assuming cleared):

```
GPIOC->MODER |= (1UL << (7 * 2));
```

3. Read input data from PC3:

```
uint32_t value = (GPIOC->IDR & (1UL << 3)) != 0;
```

Part 3: main.n code

```
#include "stm321476xx.h"
// Initialize GPIOA (PA5: LED) and GPIOC (PC13: Button)
void GPIO_Init(void);

int main(void) {
    GPIO_Init(); // Initialize GPIOA and GPIOC

    int pressed = 0; // Flag to track if button was previously pressed
    int ledState = 0; // 0 = OFF, 1 = ON

    while (1) {
        // Polling the button: check if PC13 is LOW (button is pressed)
        if ((GPIOC->IDR & (1UL << 13)) == 0) {
            if (!pressed) {
                pressed = 1; // Mark that the button is now pressed
                ledState ^= 1; // Toggle the LED state
            }
            if (ledState)
                // Turn on LED
            else
                // Turn off LED
        }
    }
}
```

```

        GPIOA->ODR |= (1UL << 5); // Turn ON LED (PA5 high)
    else
        GPIOA->ODR &= ~(1UL << 5); // Turn OFF LED (PA5 low)
    }
} else {
    pressed = 0; // Button released, allow for next press
}
}
}

void GPIO_Init(void) {
    // Enable clock for GPIOA and GPIOC
    RCC->AHB2ENR |= (1UL << 0) | (1UL << 2); // Bit 0: GPIOAEN, Bit 2: GPIOCEN

    // Configure PA5 as output
    GPIOA->MODER &= ~(3UL << (5 * 2)); // Clear mode bits for PA5
    GPIOA->MODER |= (1UL << (5 * 2)); // Set PA5 to output mode (01)

    // Configure PA5 as push-pull (default, no action needed)
    GPIOA->OTYPER &= ~(1UL << 5); // Ensure PA5 is push-pull

    // Disable pull-up/pull-down for PA5
    GPIOA->PUPDR &= ~(3UL << (5 * 2));

    // Configure PC13 as input
    GPIOC->MODER &= ~(3UL << (13 * 2)); // Set PC13 to input mode (00)

    // Disable pull-up/pull-down for PC13
    GPIOC->PUPDR &= ~(3UL << (13 * 2));
}

```