https://github.com/uofu-emb-25/5780_alex_harris

1. What are the GPIO control registers that the lab mentions? Briefly describe each of their

functions.

MODER: GPIO port mode register. Configures the GPIO pin between input mode, general purpose output mode, alternate function mode, and analog mode.

OSPEEDR: Configures the speed of the GPIO pin between one of three speed modes.

PUPDR: Connects the internal pull up/pull down resistors in the GPIO pin. (not actual resistors)

OTYPER: Sets the pin to output push-pull or output open-drain

ODR: Sets the state of the output pin, 0 sets it low and 1 pulls it high.

IDR: Reports the state of the pins in the GPIO port.

BSRR: Write only register that allows you to set and clear bits in the output register.

LCKR: Locks the configuration registers for the pin.

AFRL/AFRH: Two 32 bit registers that configure the alternate functions of the pins.

BRR: A copy of the BSRR but with the clearing bits in the lower half

2. What values would you want to write to the bits controlling a pin in the GPIOx_MODER

register in order to set it to analog mode?

11

3. Examine the bit descriptions in GPIOx_BSRR register: which bit would you want to set to

clear the fourth bit in the ODR?

20

4. Perform the following bitwise operations:

- 0xAD | 0xC7 = 11101111

- 0xAD & 0xC7 = 10000101

- 0xAD & ~(0xC7) = 101000

- 0xAD ^0xC7 = 1101010

5. How would you clear the 5th and 6th bits in a register while leaving the other's alone?

GPIOx->BSRR = GPIO_Pin << 16;

6. What is the maximum speed the STM32F072R8 GPIO pins can handle in the lowest speed setting?

• Use the chip datasheet: lab section 1.4.1 gives a hint to the location. You'll want to search the I/O AC characteristics table. You will also need to view the OSPEEDR settings to find the bit pattern indicating the slowest speed.

2 MHz when bits are x0

7. What RCC register would you manipulate to enable the following peripherals: (use the comments next to the bit defines for better peripheral descriptions)

• TIM1 (TIMER1): **RCC_APB2ENR register**

• DMA1: **RCC_AHBENR register**

• I2C1: **RCC_APB1ENR register**

8. How many commits did you generate over the course of the lab?

26 commits