

Embedded Systems Prelab 1

1. How much memory and FLASH storage does the STM32F072R8 have? (section 1.2)
2. What does the acronym "HAL" stand for? (section 1.3)
3. What is the STM32CubeMX program used for? (section 1.4)
4. Why can't a "bare-metal" embedded application return from the main function? (section 2.2)
5. In the system's memory table, are the peripheral registers higher or lower in address than the SRAM? (section 2.3)
6. What information does each of the four main datasheets/manuals used in the labs provide? (section 2.4)
7. Why do STM32F0 devices not recognize inputs/outputs on a chip by physical pin numbering? (section 2.4.1)
8. What is the name of ST's header file that defines names for the peripheral registers? (section 2.4.3)
9. What bitwise operator would you use to set a bit in a register? (section 2.5.1)
10. What peripheral enables the system clock to other peripherals? (section 2.5.2)
11. What peripheral do the HAL library delay functions use? (section 2.5.3)
12. Why should you avoid floating-point values on an STM32F0? (section 2.5.4)

Answers:

1. 16 Kbytes of static RAM and 128 Kbytes of Flash memory.
2. Hardware Abstraction Library
3. "To graphically configure the project parameters and generate a ready-to-use microVision project."
4. It can't return from the main function because it is executing directly on the processor core and because it has no operating system, there will be nothing to launch or clean up afterwards. Because of this, there is nothing there to catch the processors execution after the main program exits. This creates undefined behavior which could result in anything from resetting the device to executing random data.
5. Peripheral registers are higher in memory addresses than Sram.
6. Manuals
 1. Datasheet – Provides deice specific details for the processor including pin connections for available chip packages and available peripherals.
 2. Core Manual – Information about ARM core peripherals and assembly instruction set.
 3. Peripheral Manual – contains detailed info on all peripheral available within the device but you must use the datasheet to know which peripherals are on the specific board.
 4. Discovery board manual – schematics and tables to show onboard devices and the connectors.
7. This is because there are different chip packages and have differing numbers of pins as well as the ordering in those packages being inconsistent.
8. stm32f072xb.h is the specific header file with names and locations for the peripheral registers.
9. To set bits you must use the bitwise-OR which is "|=".

10. Reset and Clock Control (RCC)
11. SysTick timer peripheral
12. This is because many devices do not have the correct hardware support and must make up for this with other slow methods like large libraries.

Embedded Systems Post Lab 1

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

- The control registers are the MODER, OTYPER, OSPEEDR and PUPDR. The MODER is used to select the I/O mode (whether it is input, output, AF, or analog). The OTYPER is to select the output type as either push-pull or open drain and OSPEEDR is also for the output, but it is for the speed level. The last one, PUPDR, is used to select whether it will pull up or down depending on the I/O direction.

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?

- You would want to send it the bits '11' to set it to analog mode.

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

- Bit number 3 is the one you would want to set to clear the 4th bit in the ODR.

4. Perform the following bitwise operations:

- $0xAD \mid 0xC7 = 0xEF$
- $0xAD \& 0xC7 = 0x85$
- $0xAD \& \sim(0xC7) = 0x7A$
- $0xAD \wedge 0xC7 = 0x6A$

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

- (Register & 0xCF) Assuming the LSB is the first bit. 0xCF is 0b11001111 where the 5th and 6th bit are zero which when we and it with the register it will clear the 5th and 6th bit.

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.

-It is between 1 and 2 MHz depending on the conditions of voltage and capacitance.

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_APB2RSTR register
- DMA1 – RCC_AHBENR register
- I2C1 – RCC_APB1ENR register