Name: Khoi Nguyen

Class: ECE 5780

Prelab 01 (Intro/GPIO)

1. How much memory and FLASH storage does the STM32F072R8 have? (section 1.2)
   1. It has 16 Kbytes of static RAM and 128 Kbytes of Flash memory for program and data storage.
2. What does the acronym "HAL" stand for? (section 1.3)
   1. HAL stand for “hardware abstraction library”
3. What is the STM32CubeMX program used for? (section 1.4)
   1. STM32CubeMX program used to graphically configure the project parameters and generate a ready-to-use μVision project.
4. Why can't a "bare-metal" embedded application return from the main function? (section 2.2)
   1. In a "bare-metal" embedded application, the main function cannot return because there is no operating system to catch the processor's execution after the main program exits.
5. In the system's memory table, are the peripheral registers higher or lower in address than the SRAM? (section 2.3)
   1. The peripheral registers are higher in address than the SRAM.
6. What information does each of the four main datasheets/manuals used in the labs provide? (section 2.4)
   1. **(STM32F072RBT6 Datasheet) DM00090510.pdf**: The chip datasheet provides device-specific details for the processor; this includes pin connections for available chip packages and a list of available peripherals.
   2. **(Programming & Core Manual) DM00051352.pdf**: The core programming manual provides information on the ARM-core peripherals as well as the assembly instruction set; it is generic to all of the processors within the STM32F0 family.
   3. **(Peripheral Manual) DM00031936.pdf**: The peripheral reference manual contains detailed information on all peripherals available within an STM32F0 device; however, not all STM32F0 devices contain every peripheral! The chip datasheet is necessary to determine which peripherals are available for use.
   4. **(Discovery Board Manual) DM00099401.pdf**: The Discovery board manual contains schematics and tables that show the onboard devices and connectors attached to the STM32F0; the Discovery board silkscreen also documents many device connections.
7. Why do STM32F0 devices not recognize inputs/outputs on a chip by physical pin numbering? (section 2.4.1)
   1. Because different chip packages with differing numbers of pins, and the pin ordering between these is inconsistent; GPIO pins are instead labeled with a port name (PA0 for example) which describes where to go to configure it. Within the chip datasheet, we see a table mapping GPIO pin names to physical pin numbers on the specific chip package.
8. What is the name of ST's header file that defines names for the peripheral registers? (section 2.4.3)
   1. stm32f0xx.h
9. What bitwise operator would you use to set a bit in a register? (section 2.5.1)
   1. To set bits in a register, bitwise-OR its value with a bitmask.
10. What peripheral enables the system clock to other peripherals? (section 2.5.2)
    1. The STM32F0 family has a dedicated peripheral called the Reset and Clock Control (RCC) which enables or disables clock signals around the chip.
11. What peripheral do the HAL library delay functions use? (section 2.5.3)
    1. The SysTick timer peripheral is a device which raises a system signal at a configurable periodic rate; since the duration between these signals is a known quantity, the SysTick is useful as an application heartbeat. The HAL library uses the SysTick to trigger periodic tasks such as updating a global system time variable.
12. Why should you avoid floating-point values on an STM32F0? (section 2.5.4)
    1. Because many embedded devices, including the STM32F0, do not have hardware support for floating-point mathematics and must emulate it with large and slow code libraries. This can lead to inefficient code and slower execution times.