Brian Burton

ECE 5780

2/ 2/2022

**Prelab 01 (Intro/GPIO)**

1. How much memory and FLASH storage does the STM32F072R8 have? (section 1.2)

It has 16 Kbytes of static RAM and 128 Kbytes of Flash memory for program and data storage.

1. What does the acronym "HAL" stand for? (section 1.3)

The acronym “HAL” stands for the hardware abstraction library.

1. What is the STM32CubeMX program used for? (section 1.4)

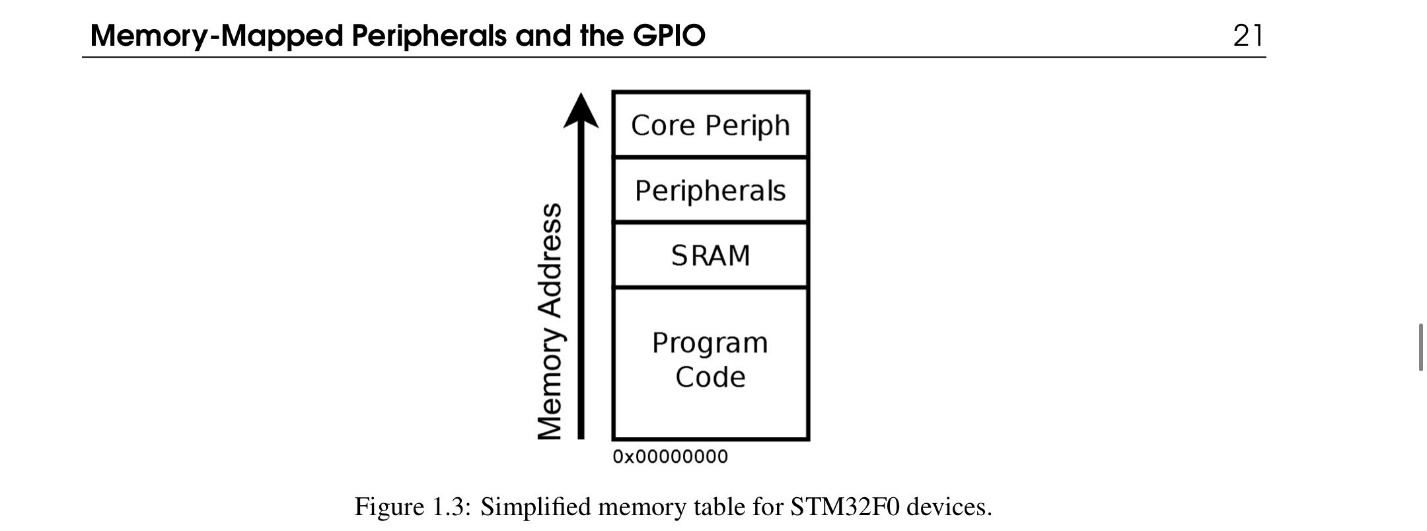
We will use the STM32CubeMX utility to graphically configure the project parameters and generate a ready-to-use µVision project.

1. Why can't a "bare-metal" embedded application return from the main function? (section 2.2)

After device initialization, the startup code calls the main() function within the user’s application; embedded programs generally begin by configuring hardware peripherals that they require and subsequently enter into an infinite loop where the majority of the applications resides. This endless loop is necessary since the main function of an embedded program should never return.

1. In the system's memory table, are the peripheral registers higher or lower in address than the SRAM? (section 2.3)

The peripheral registers are higher in address than the SRAM. (See image below).



1. What information does each of the four main datasheets/manuals used in the labs provide? (section 2.4)

Each of the four main datasheets/manuals provides insights and information about the board that we use in lab.

1. Why do STM32F0 devices not recognize inputs/outputs on a chip by physical pin numbering? (section 2.4.1)

This is due to 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.

1. What is the name of ST's header file that defines names for the peripheral registers? (section 2.4.3)

The answer lies within the CMSIS Cortex-M0 Device Peripheral Access Layer Header Files, also known as stm32f0xx.h and its derivatives. Originally, stm32f0xx.h defined all peripheral information for the entire STM 32F0 line. However, in recent versions or the HAL, ST Microelectronics has split the information for each device sub-family into separate files.

1. What bitwise operator would you use to set a bit in a register? (section 2.5.1)

To set bits in a register, bitwise-OR its value with a bitmask. Any bits set in the bitmask will set the corresponding bit in the register. The bitwise-OR operator is a single vertical-pipe character ‘I’.

1. What peripheral enables the system clock to other peripherals? (section 2.5.2)

The STM32F0 family has a dedicated peripheral called the Reset and Clock Control (RCC) which enables or disables clock signals around the chip.

1. What peripheral do the HAL library delay functions use? (section 2.5.3)

The SysTick timer peripheral is a device which raises a system signal at a configurable periodic rate; since the duration between these signals is 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.

1. Why should you avoid floating-point values on an STM32F0? (section 2.5.4)

The STM32F0 does not have the hardware support for floating-point mathematics and must emulate it with large and slow code libraries. Higher-end devices such as the STM32F4 family of chips have a hardware floating-point unit. (FPU).