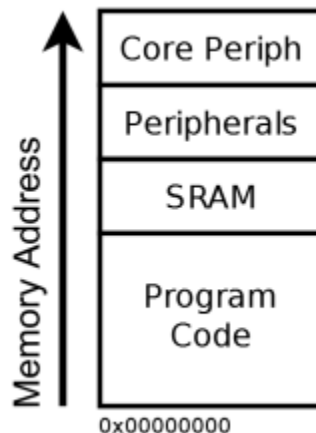


1. How much memory and FLASH storage does the STM32F072R8 have?
 - 16 Kbytes of static RAM
 - 128 Kbytes of Flash memory for program and data storage
2. What does the acronym "HAL" stand for?
 - HAL = Hardware Abstraction Library. The HAL provides a C/C++ API for controlling peripherals
3. What is the STM32CubeMX program used for?
 - It's used to configure STM32 microcontrollers and microprocessors as well as generate initialization code (in C) for ARM Cortex-M Core processors through a step-by-step process with a GUI. With STM32CubeMX we can graphically configure the project parameters and generate a ready-to-use μ Vision project.
4. Why can't a "bare-metal" embedded application return from the main function?
 - With "bare-metal" applications, an OS to launch or clean up after user applications exit does not exist. This means that once the reset vector executes, more complex initialization code executes, and the startup code calls the main() function within the user's application entering into an infinite loop, if main returns nothing is there to catch the processor's execution resulting in undefined behavior. This undefined behavior could range from resetting the device to executing random data.
5. In the system's memory table, are the peripheral registers higher or lower in address than the SRAM?
 - The peripheral registers higher in address:



6. What information does each of the four main datasheets/manuals used in the labs provide?
- Discovery Board user manual: board specific, contains information about the board's physical layout and what connections go where.
 - Chip datasheet: specific to the processor that is used, contains pinout information, package information, and available protocols and pin definitions. Could contain information about the different packages within the group of processors.
 - Programming/core manual: addresses information on the STM32 Cortex-M0 (our processor's family of processors). Provides information on the instruction set, core peripherals and all-around general information about the family of processors.
 - Peripheral reference manual: contains information about all the peripherals that *might* be available to use with our processor. Includes descriptions for the registers for those peripherals, which may include clocks, memory, GPIO, etc.

Or verbatim from the lab manual:

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?
- 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.
8. What is the name of ST's header file that defines names for the peripheral registers?
- stm32f072xb.h

9. What bitwise operator would you use to set a bit in a register?

- You would use a bitwise-OR operator. As per the manual: “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 ‘|’”

10. What peripheral enables the system clock to other peripherals?

- Reset and Clock Control (RCC) peripheral enables the system clock to other peripherals. Specifically within the “RCC_TypeDef” definition in the stm32f072xb.h header file, there are three “peripheral clock registers” labels. These three registers control the clock signals to all other peripherals except for those in the ARM-core itself.

11. What peripheral do the HAL library delay functions use?

- The SysTick peripheral. The SysTick timer peripheral is a device which raises a system signal at a configurable periodic rate.

12. Why should you avoid floating-point values on an STM32F0?

- Many embedded devices do not have hardware support for floating-point mathematics and must emulate it with large and slow code libraries. The STM32F0 is one of those devices.