



## Lab 3: ARM Assembly Sort

### 1 Learning Objectives

By the end of this lab you will have:

- Written a simple assembly program to sort an array of signed bytes
- Used the debugger in PlatformIO to monitor a location in memory.

### 2 Lab Assignment

#### 2.1 Requirements

Write an ARM assembly-language program to sort an array of 12 signed bytes on the STM32F401RE.

#### 2.2 Instructions

- Set up a project in PlatformIO by going to PlatformIO home and clicking “New Project”. Name your project `lab3_xx` where `xx` stands for your initials. Select the STM32F401RE board from the dropdown list and the desired location for your project.
- Once your project is created, download the starter code from [the course Github](#) and put it in your `src` folder or create a new file inside your `src` directory titled `sort.S` and copy in the contents of the starter code. N.B. the uppercase `S` for the file extension, this is important for your code to correctly compile with the current build configuration in PlatformIO!
- Finish writing the assembly language subroutine under label `main` in `sort.S` to sort 12 signed bytes (the ones on the `.byte ...` line under `arr`). Remember that assembly language code is nearly unreadable without line-by-line comments. If needed, use online reference sheets for ARM Thumb2 Assembly language (like the one [here](#)) or refer to Chapter 6 of Digital Design and Computer Architecture.

#### 2.3 Running and Testing Your Code

- Build your code by selecting the build command from the PlatformIO “Project Tasks” menu, by clicking the checkmark icon in the bottom toolbar, or by using the keyboard shortcut `ctrl + alt + b`.
- Double check your `platformio.ini` file and add a line that says `debug_tool = stlink` at the bottom of the file.
- With the Nucleo board connected via USB, start a debugging session by going to the debug section of the sidebar and pressing the green arrow in the top of the menu (or by using the keyboard shortcut `F5`).
- Examine the memory location (`0x20000000`) where the array of signed bytes has been loaded using the memory location viewer in the bottom left of the debugger toolbar. You may also choose to use the “Watch” section of the debugger. For example, if you want to view the contents of the byte array stored in memory, you can enter `*(char *)(&arr)@12` as a Watch expression. This tells the watch window to cast the address of the `arr` label to a pointer to a character and dereference 12 consecutive values from that memory location.
- Step through the operation of your program and see how it changes the values of the array stored at memory location `0x20000000`.
- To test your code, try various cases with the array in the `.byte ...` line under the `arr` label. Rebuild and start a new debug session every time you make changes.

### 3 What to Turn In

- A short writeup of your design approach and the sort algorithm you chose to implement.



- A copy of your assembly code file `sort.S`.
- A listing of the test cases you used and the output of the tests. Be sure your tests would convince a skeptic that your algorithm works.
- For your lab checkoff, please have your lab set up and connected to a running PlatformIO session so that the instructor may provide a test case for your code.

### 3.1 General Info

- How many hours did you spend on the lab?
- Any comments, suggestions, or complaints about the assignment? Please fill out the form here to help improve the lab for the future! This will not count toward your grade.