

Laboratory 1

(Due date: September 26th)

OBJECTIVES

- ✓ Introduce the Software Development Tool: CodeWarrior for HCS12(X).
- ✓ Introduce Assembly Language instructions.
- ✓ Compiling and Debugging on the Dragon12-Light Board.

ASSEMBLY PROGRAMMING WITH THE CODEWARRIOR

- ✓ For a comprehensive description of Assembly Instructions, refer to the [HCS12 CPU Reference Manual Rev. 4.0](#). Go to the Appendix A. Instruction Reference.
- ✓ Refer to the [CodeWarrior Assembler Manual for HCS12](#) for a comprehensive guide on Assembly Programming using CodeWarrior.

CODEWARRIOR FOR HCS12(X) AND DRAGON12-LIGHT BOARD SETUP

- Turn on your Board by connecting your board to your computer via the USB cable (use the USB connector below the LCD). If it does not turn on, connect the power supply of the Board.
- The COM drivers will be installed automatically. You can verify this by opening the Device Manager (it is usually installed as COM4)
- Make sure that left switch of SW7 is set to the LOAD position (down).

FIRST ACTIVITY (50/100)

- Create a New Project for Assembly Code in CodeWarrior:
 - ✓ File → New Project. Select the HCS12 → HCS12D Family → MC9S12DG256B device.
 - ✓ Default Connection Window: Select *Full Chip Simulation*.
 - ✓ Languages to Support: Check ONLY *Absolute assembly*.
 - ✓ Project name: **lab1a** Uncheck the *Create main C/.asm* file (otherwise a main.asm will be created)
- Use the provided file (`main_lab1.asm`) as the main source. You can copy the file into the `/Sources` directory and then add the file to the project.
- The code stores 8 bytes in contiguous locations in memory (starting from `$1000`) and adds them all. The final result is stored in the address provided by the contents of `x`.
- Compile the project (Project → Compile). Then, build the project (Project → Make). If something goes wrong, ask your TA.
- **Debugger:**
 - ✓ Go to Project → Debug. The Debugger Window will open.
 - ✓ Run the program instruction by instruction. Take a look at how the registers (D, A, B, IX, IY, PC, CCR) vary as the program executes.
 - ✓ You can also see how the memory contents (starting at `$1000`) are updated (they are usually shown in blocks of 8 bytes) in the Memory Window.
 - ✓ Familiarize yourself with the use of Breakpoints, Step, etc (consult with your TA). Verify that the addition result is correct.

SECOND ACTIVITY (50/100)

- Create a new project (**lab1b**). In this new project, you will redo the previous example, but you will treat the initially stored 8-bit numbers as signed. Compile the project and run the debugger. Go over your code step by step and take a look at how the CPU registers vary. Verify the execution of the program and that your addition result is correct.
- You need to provide the `.asm` file you create to your TA for its evaluation.

TA signature: _____

Date: _____