

## Running and Debugging a Program in Keil MDK-ARM's Simulator

Keil MDK-ARM is relatively easy to use IDE, which has a powerful simulator so that we do not have to use a HW kit when we do not operate peripherals. . We can run or debug a program in the **Debug** mode with the simulator. When you are “Creating a New Keil MDK-ARM Project by Modifying an Existing One” (see [cec32x\\_devtool\\_16\\_creating\\_new\\_Keil\\_proj\\_via\\_modifying\\_an\\_existing\\_one.pdf](#)), you will have the simulator set up already.

### Running a program in the Keil simulator

Many times, when we use a simulator, we want to print out some results to a terminal or console. For example, when we run the “Hello World” project in the Keil MDK-ARM simulator, we want to have “Hello World” printed out. To this end, we need to do the following:

- Compile the project by clicking the **Build** button (or F7). Make sure there is no error with the compilation. Sometimes, if you see mysterious errors, you can force to rebuild all files by clicking the **Rebuild** button.
- After seeing no build error, go to the **Debug** mode by clicking the **Stop/Start Debug Session** button (Ctrl + F5).
- Make sure the **Debug (printf) Viewer** window is activated; if no such a window is activated, click the triangle beside the **Serial Windows** button and click **Debug (printf) Viewer**.
- Run the project by clicking the **Run** button (F5). The output will be printed in the **Debug (printf) Viewer** window.

### Debugging a program in the Keil simulator

We don't have to print out the values of some variables to see their values. We can use various techniques listed in Lab 1 to see their values, such as:

- Displaying the local variables in the **Watch** window.
- Displaying the global variables in the **Memory** window.

To see the variables using the **Watch** window, we have to be in the right **scope**, which can be guaranteed by setting and running to a breakpoint just after the variable is assigned the needed value. Here, the following buttons are extremely important:

- The **Run** button (F5).
- The **Reset** button (to the left of the **Run** button).

Note that the debugger is a **Source level** debugger which can step through each source code line. To do source level debugging to execute a number of lines, we can use the following:

- The **Step** button (F11). This is used to step to the next statement if the current one is a basic statement line or into a function if the current one has such a function.
- The **Step Over** button (F10). This is used to step over to the next statement even if the current statement has a function. (Step over the function in the current statement.)
- The **Step Out** button (Ctrl + F11). This is used to step out of the function. Sometimes, this button does not work well. If this is the case, use the **Step over** to proceed.
- The **Run to Cursor Line** button (Ctrl + F10). If we place the cursor in a line of the code, we can

Note also that the source level is for assembly language as well. If we want to see the values of the general purpose registers after each assembly instruction in the **Debug** mode, as will be demonstrated in the class soon, we can use these **Step** related debug techniques.