

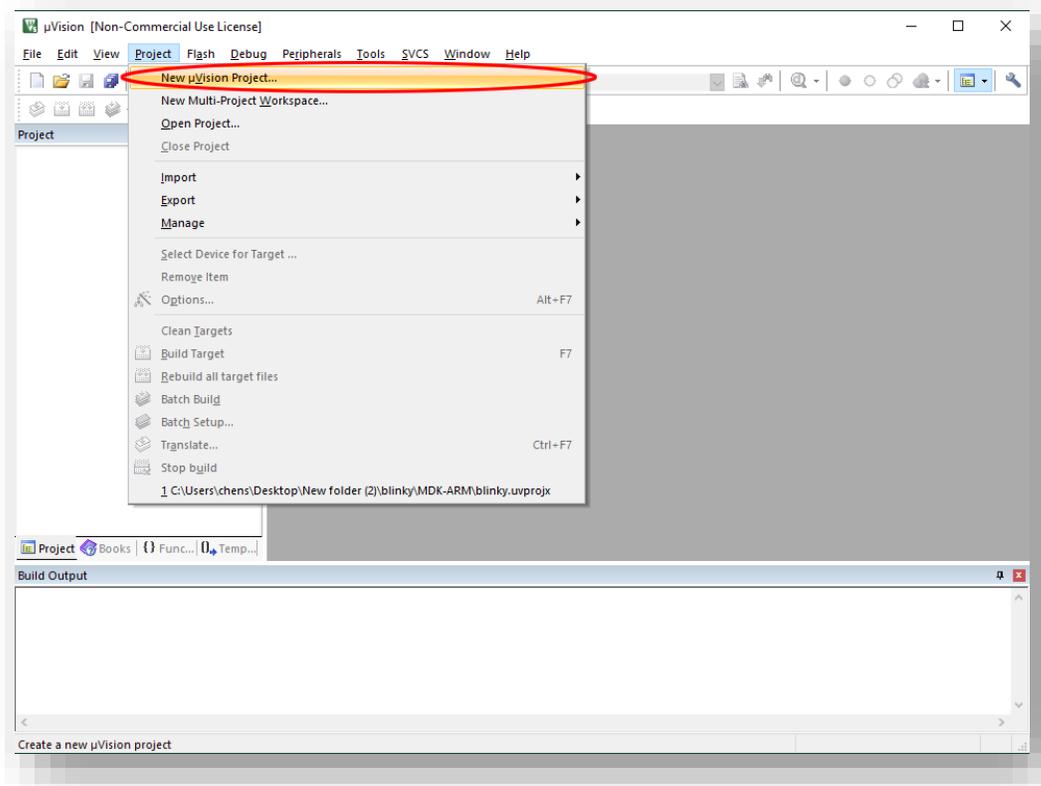
Start A New Project with Keil MDK-ARM Version 5 for STM32F030

This tutorial is intended for starting a new project to develop software with STM32FG030R8 Nucleo-64 board using Keil Microcontroller Development Kit for ARM (MDK-ARM) version 5.34.

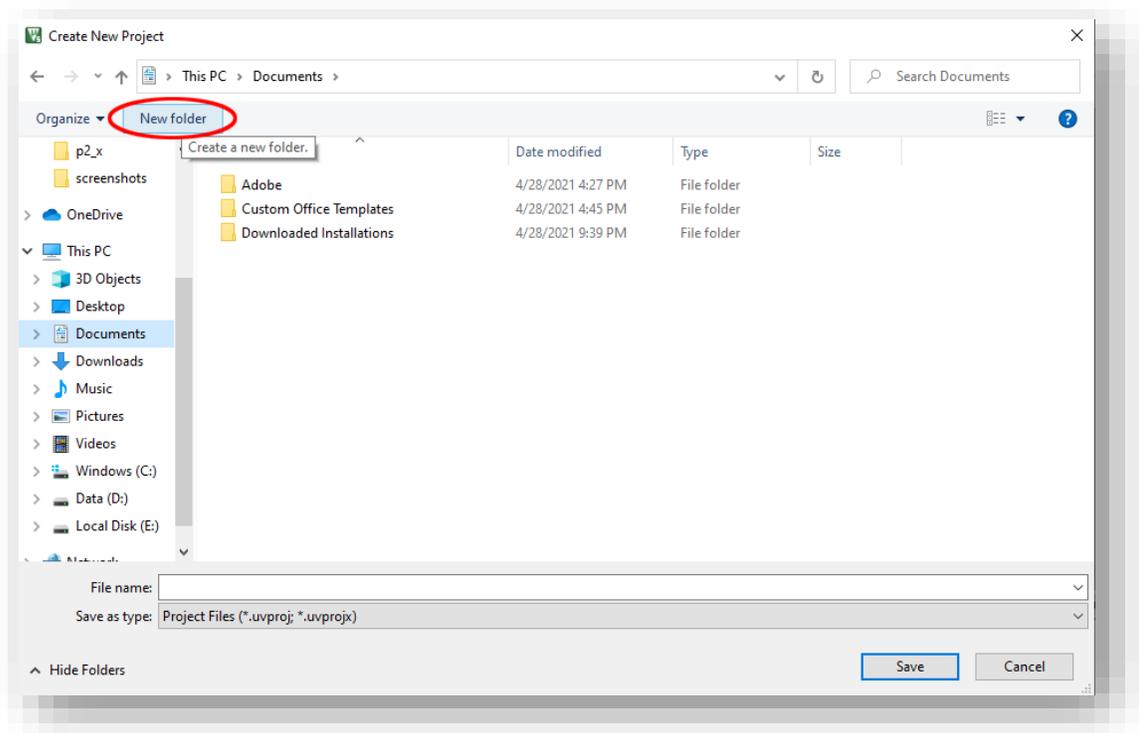


Create a New Project in Keil MDK v5.34

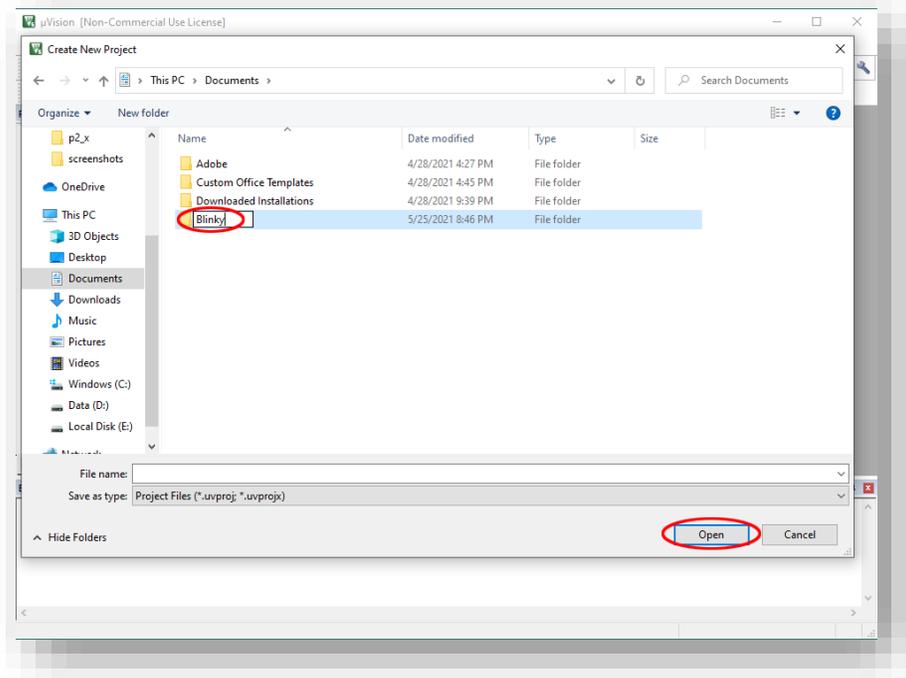
1. Launch Keil uVision IDE by double clicking on the icon .
2. From the menu, select Project>New uVision Project...



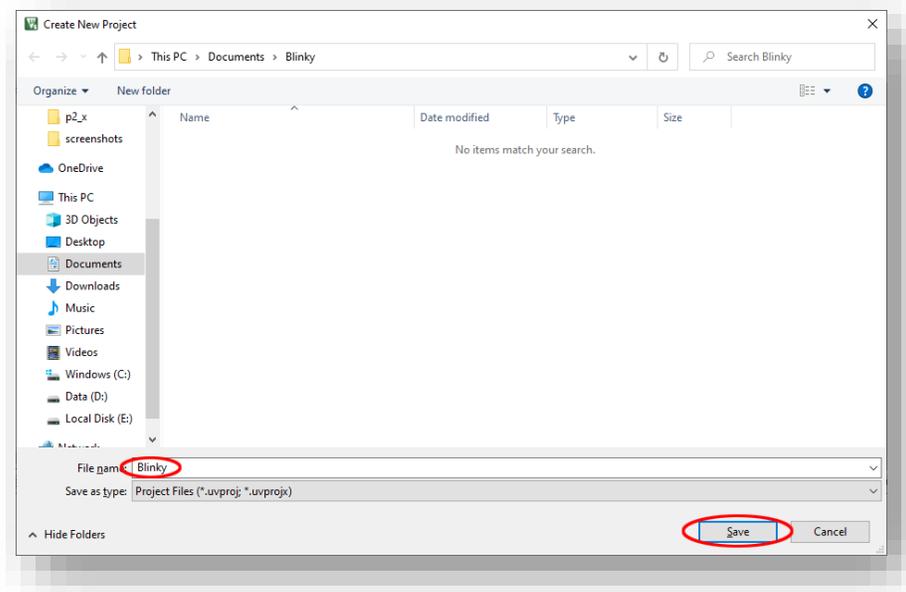
3. In the “Create New Project” dialog box, click “New folder”. Browse to a folder location where you would like to create your project folder.



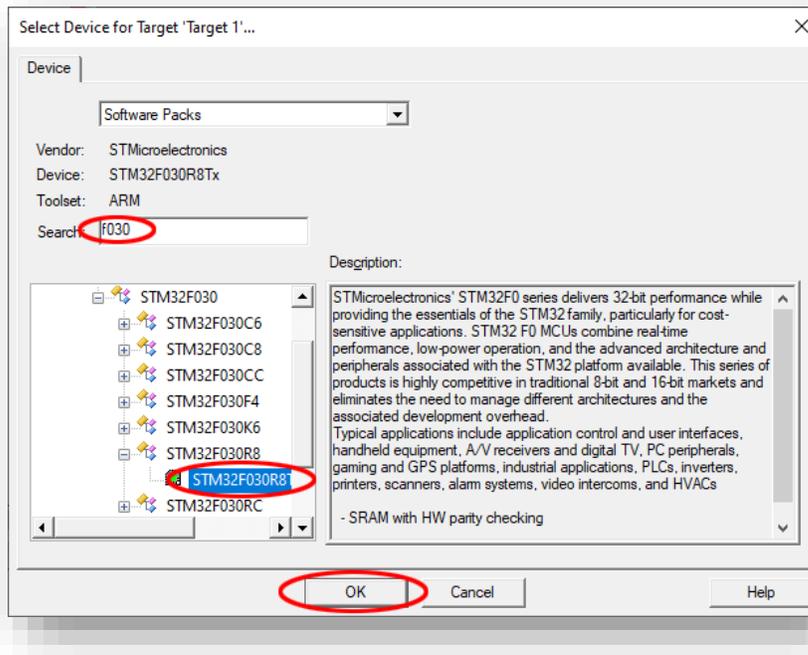
4. Type in the folder name, for example, “Blinky”, then click the “Open” button. This will create a folder named “Blinky” to hold all the files for the new project. It will also take you inside that folder.



5. While in the project folder, type “Blinky” in the File name field and click the “Save” button. This creates a project with the project named “Blinky”. Although we used the same name for the project folder and the project, they do not have to be identical.



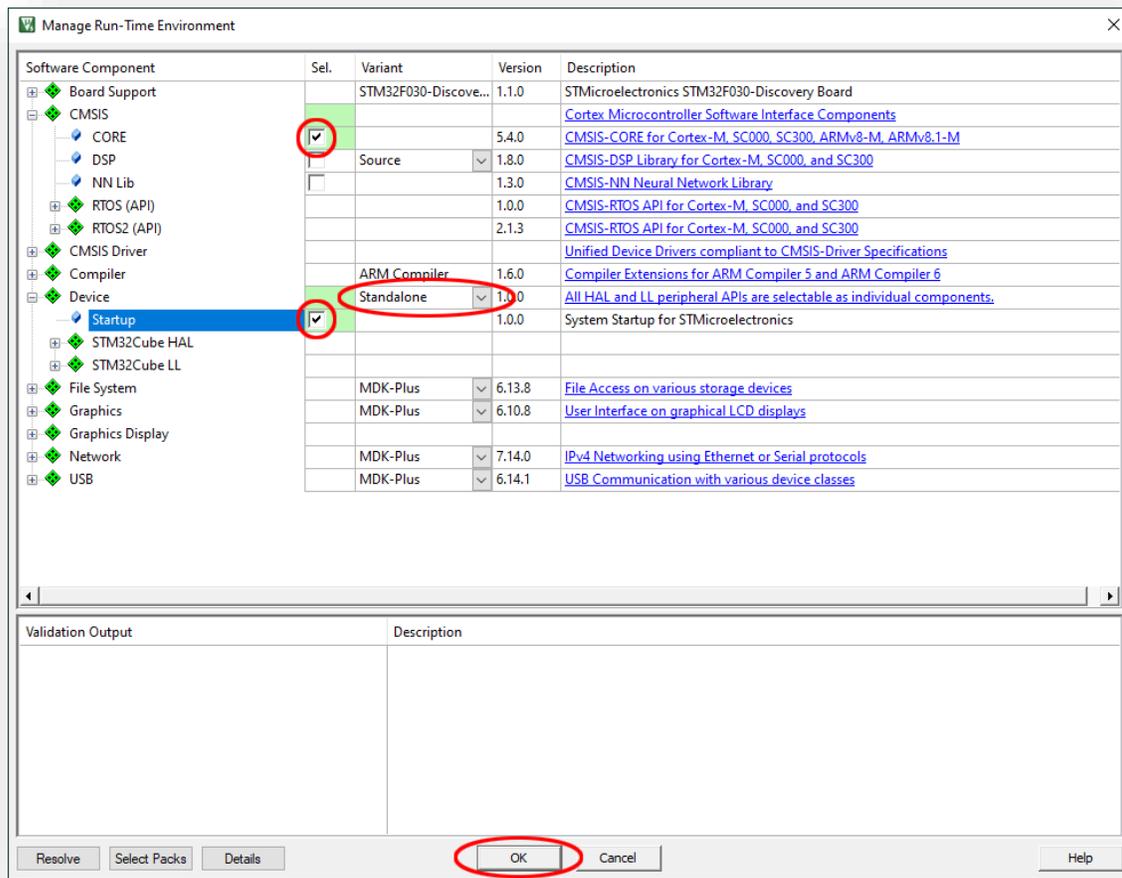
- The Project Wizard will prompt you to select the device type for the project target. You may drill down the device family tree to find “STM32F030R8”. If you installed a large number of supported devices, it may take a while to find the device in the family trees. In that case, type in a substring of the device name in the “Search” field and the matching devices will appear in the window below. Click select the device type then click the “OK” button.



If the device “STM32F030R8” could not be found, you need to install its Software Pack. Please refer to the Keil uVision IDE installation.

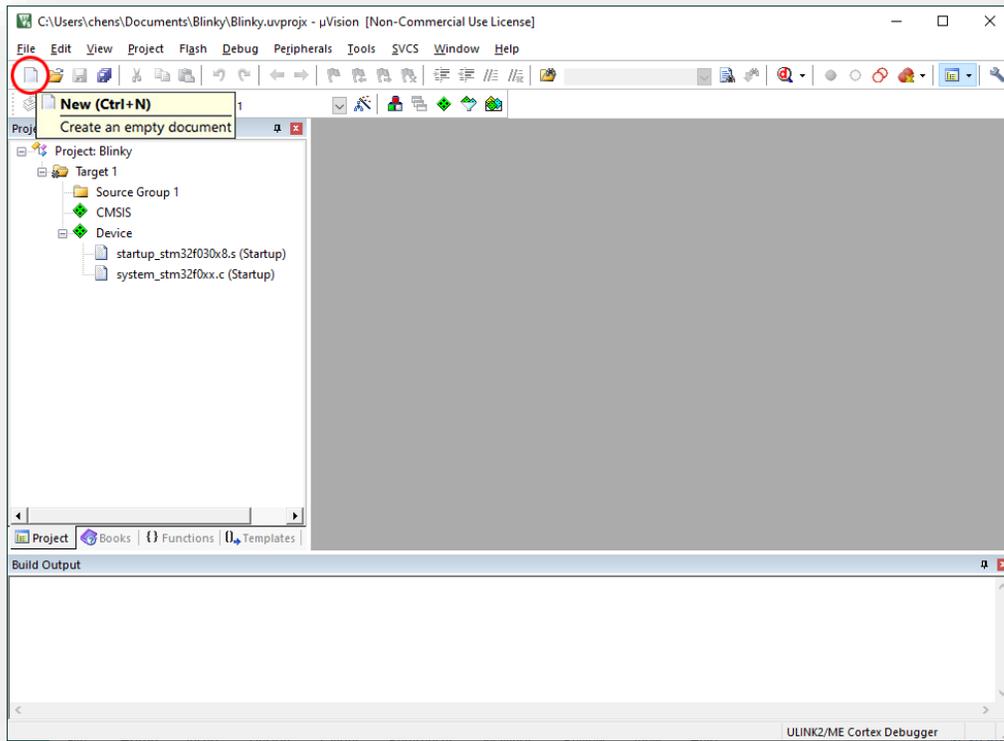
- Next, the Project Wizard will help you manage the run-time environment. Expand the Software Component selections, check “CMSIS>CORE”, select “Standalone” in the pull-down of Device

category then check “Device>Startup”. Click the “OK” button to close the window.



8. You should see a project with a target created in the Project pane. If you click the + signs to open the target, you would see the startup codes for the specific device that were created in the project.

Click the “New” button for a new text file in the editing window.



9. Copy the code below into the new text file window. This is a simple program to blink the LED LD2 on the STM32F030 Nucleo-64 board.

```
/* use delay loop, 1 sec on 1 sec off
 * default 8 MHz clock
 * LD2 connects to PA5
 */
#include "stm32f0xx.h"

void delayMs(int n);

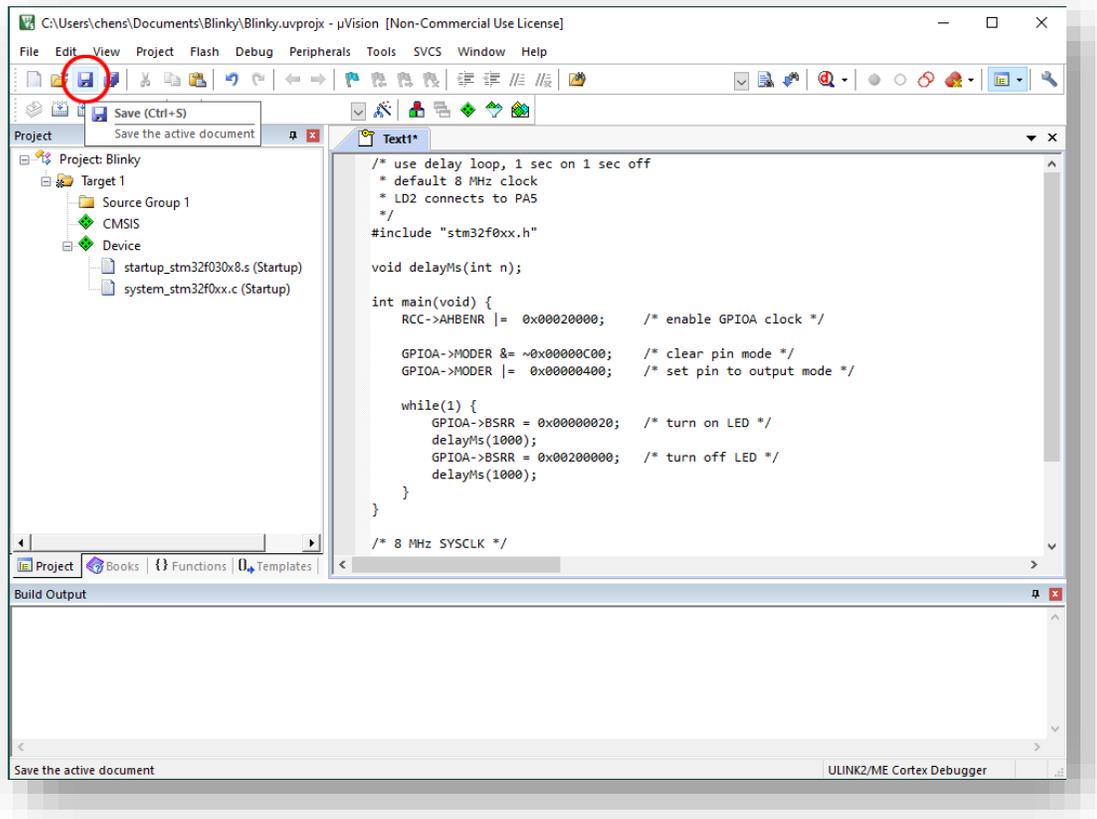
int main(void) {
    RCC->AHBENR |= 0x00020000;    /* enable GPIOA clock */

    GPIOA->MODER &= ~0x00000C00; /* clear pin mode */
    GPIOA->MODER |= 0x00000400;  /* set pin to output mode */

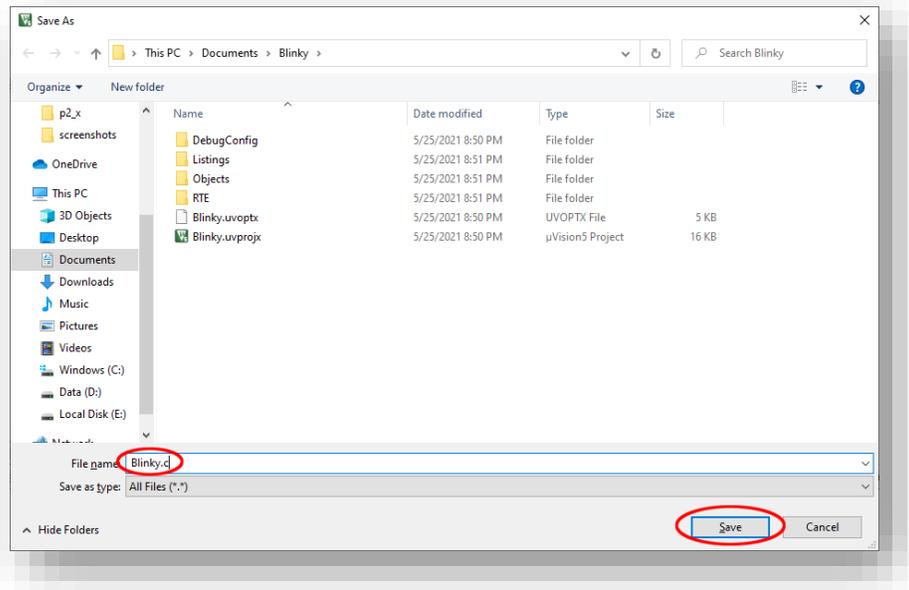
    while(1) {
        GPIOA->BSRR = 0x00000020; /* turn on LED */
        delayMs(1000);
        GPIOA->BSRR = 0x00200000; /* turn off LED */
        delayMs(1000);
    }
}

/* 8 MHz SYSCLK */
void delayMs(int n) {
    int i;
    for (; n > 0; n--)
        for (i = 0; i < 1142; i++) ;
}
```

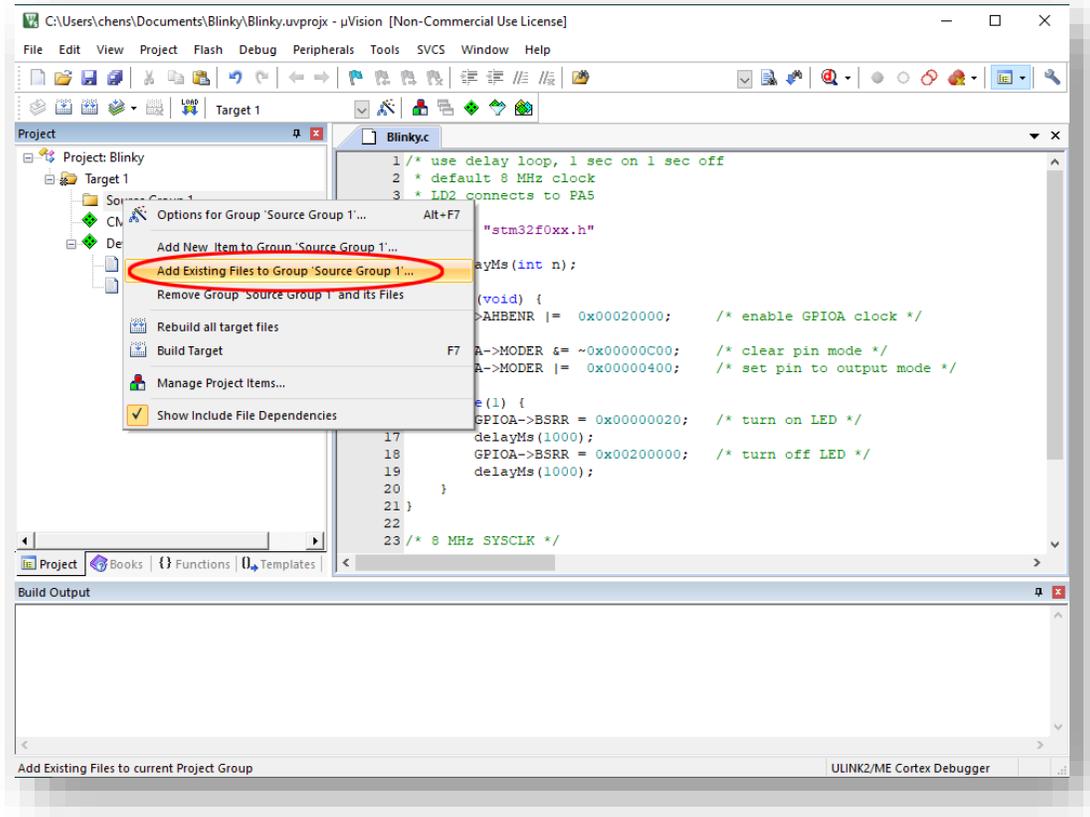
10. Click the "Save" button to save the file with the code. By default, the file will be saved in the project folder.



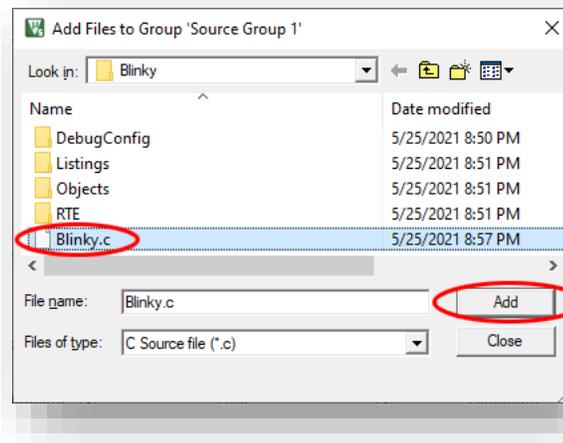
11. Give the file name “Blinky.c” and click the “Save” button. The editor is context-sensitive. Once you make it a C source file, the file content will change the color.



12. Once the source file is saved, it needs to be associated with the project. Right click on “Source Group 1” and select “Add Existing Files to Group ‘Source Group 1’...”.

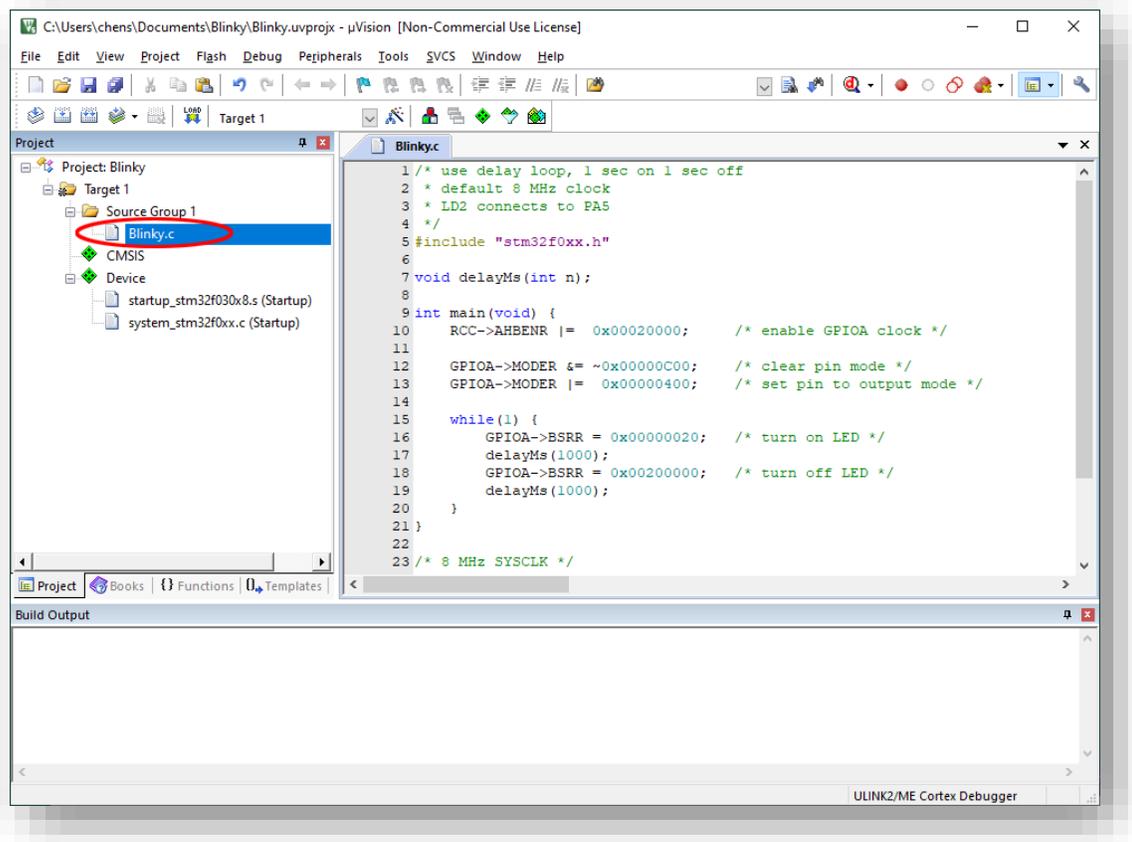


13. Locate the file Blinky.c, click select it then click the “Add” button. Click the “Close” button to

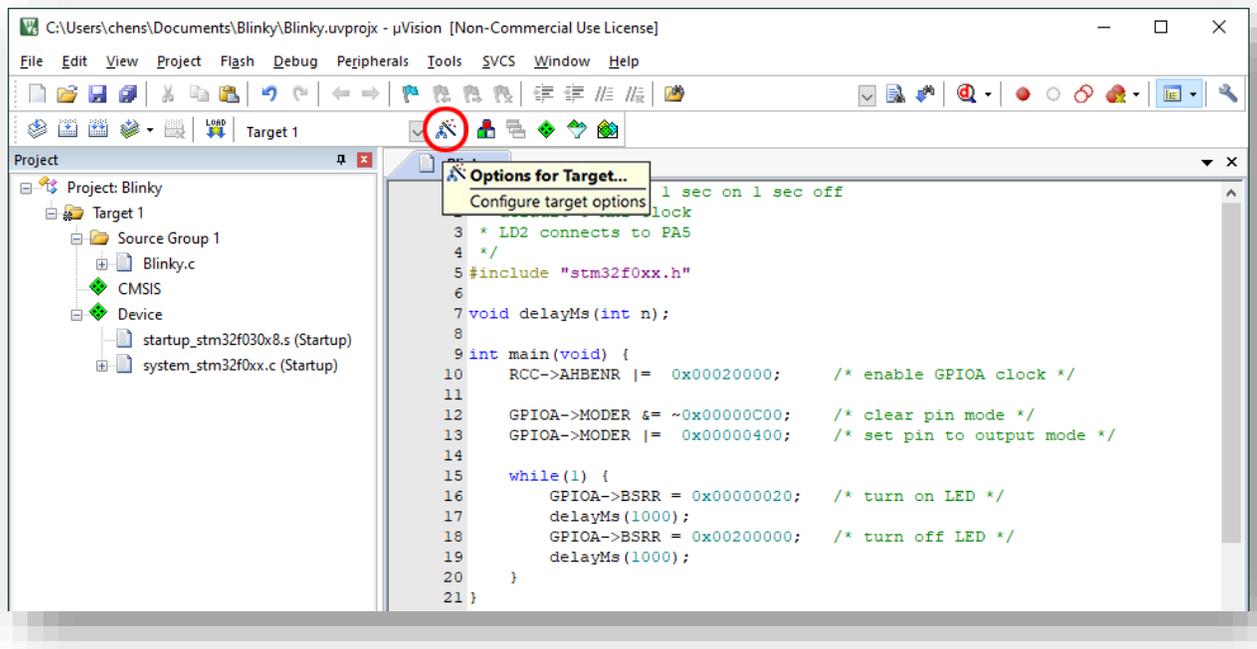


proceed.

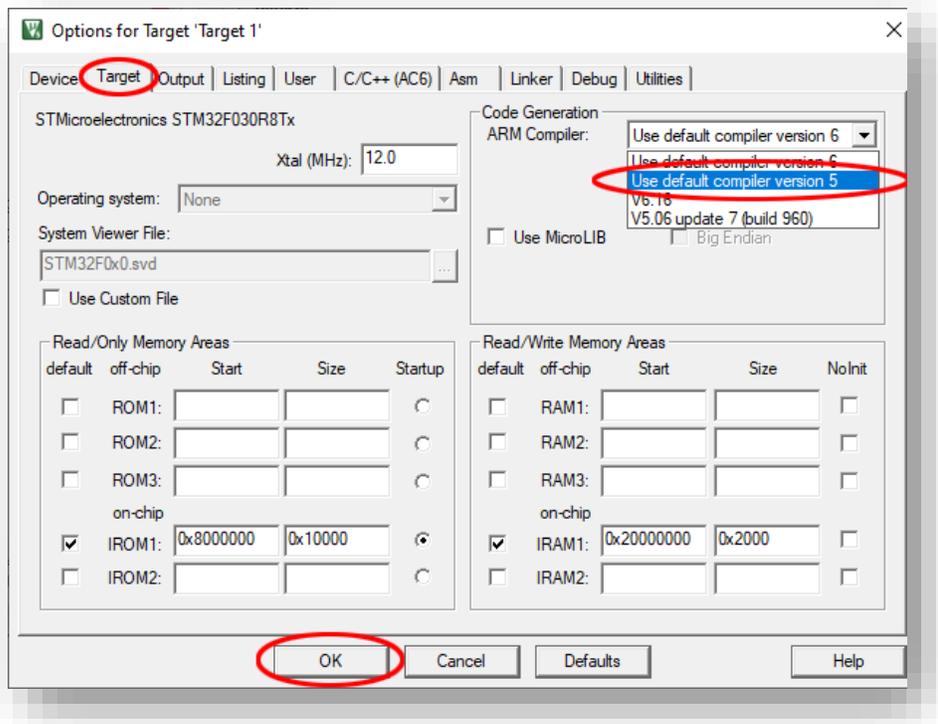
14. After the file is added to the project, it will appear under “Source Group 1”.



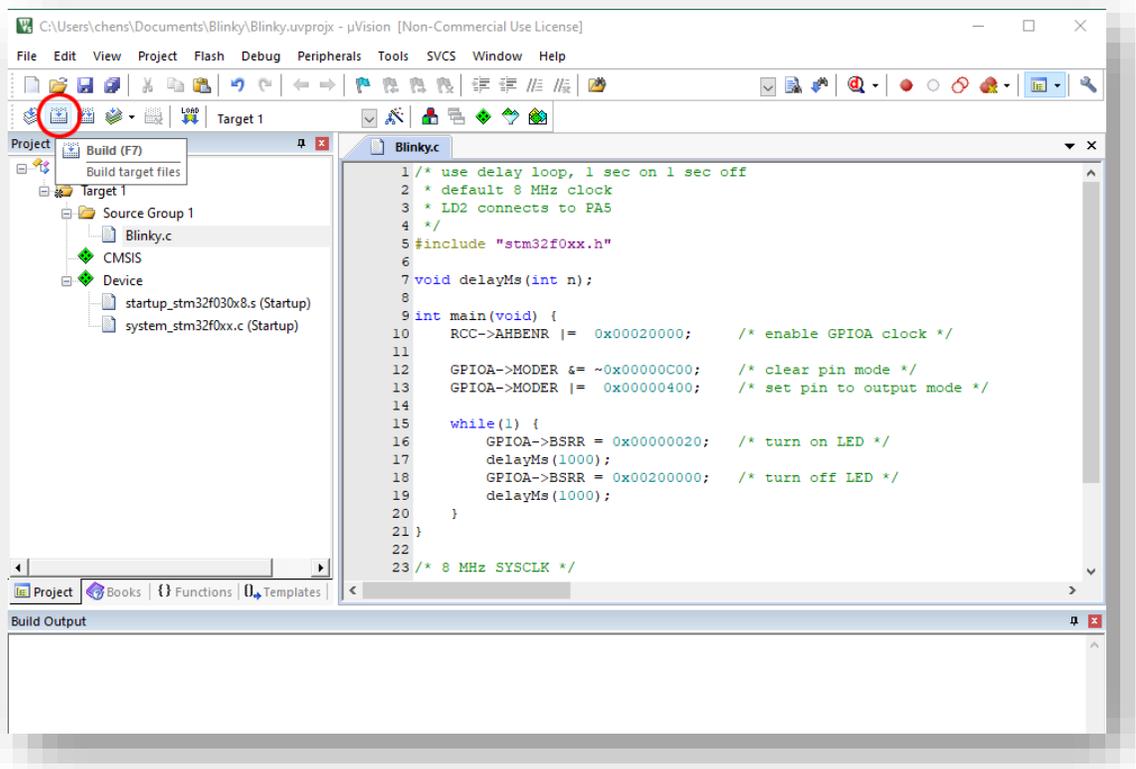
15. Before we build the project, we need to select the compiler used. Click the “Options for Target...” button to open up a dialog box.



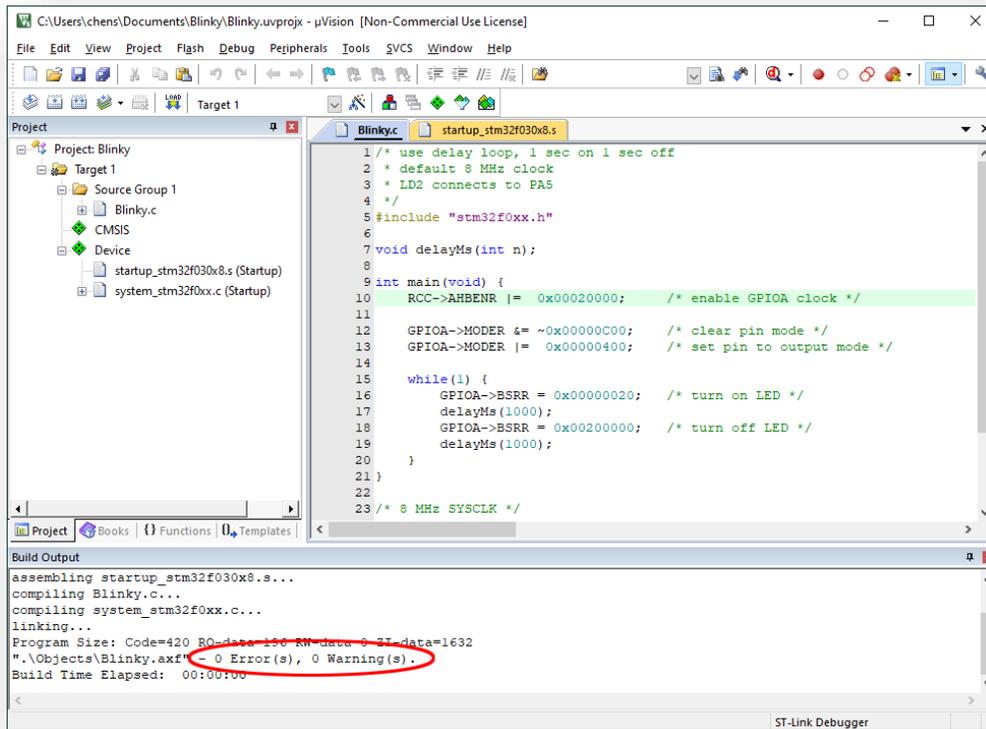
16. In the dialog box, select the “Target” tab, in the Code Generation area, pick “Use default compiler version 5” from the pull-down menu. Click “OK” to close the dialog box.
In the older versions of Keil uVision IDE, the default compiler was version 5 and the programs in this book were tested using the default version 5 compiler. The recent versions of Keil uVision IDE changed the default compiler to version 6 and came with a new set of default compiler settings. Some of the programs in this book may not work properly with the new settings in version 6. Once you select the version 5 compiler, your computer should remember your choice.



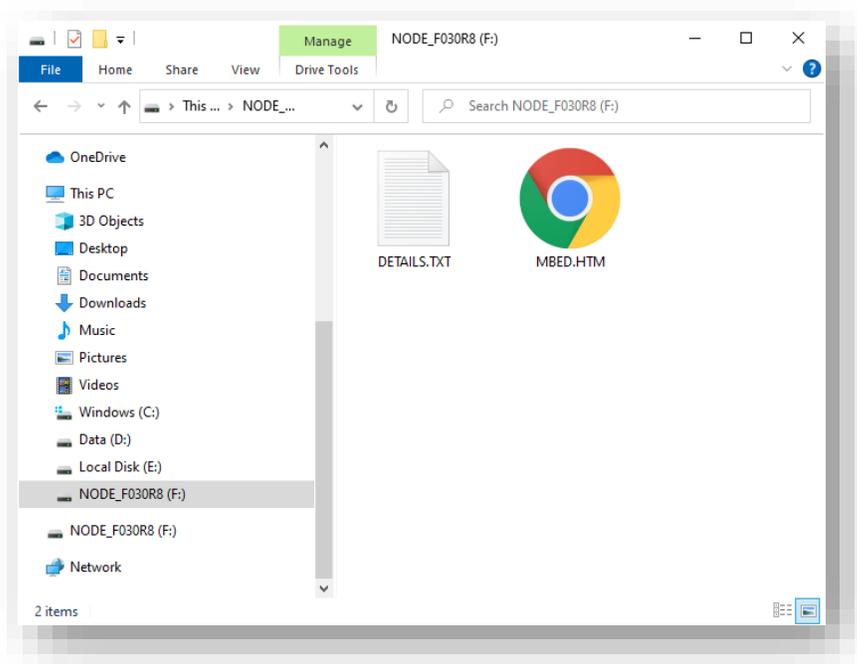
17. Click the "Build" button to build the project.



18. You should get a clean build with this project with 0 Error and 0 Warning.

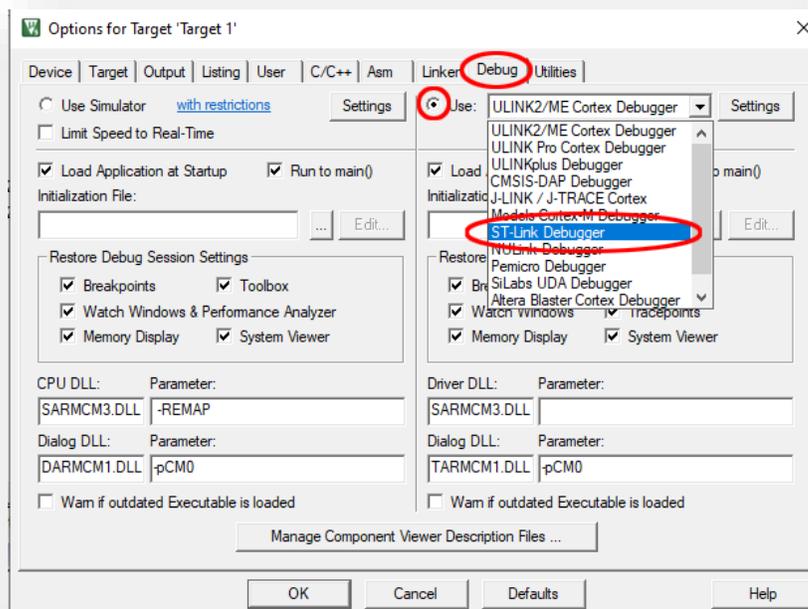


19. Once the project is built, we need to configure the debug interface so that the program may be downloaded to the target. Make sure the STM32F030 Nucleo-64 board is connected to the computer with the USB cable. The ST-LINK debugger on the Nucleo board is MBED compatible. It will appear as a USB thumb drive as below.

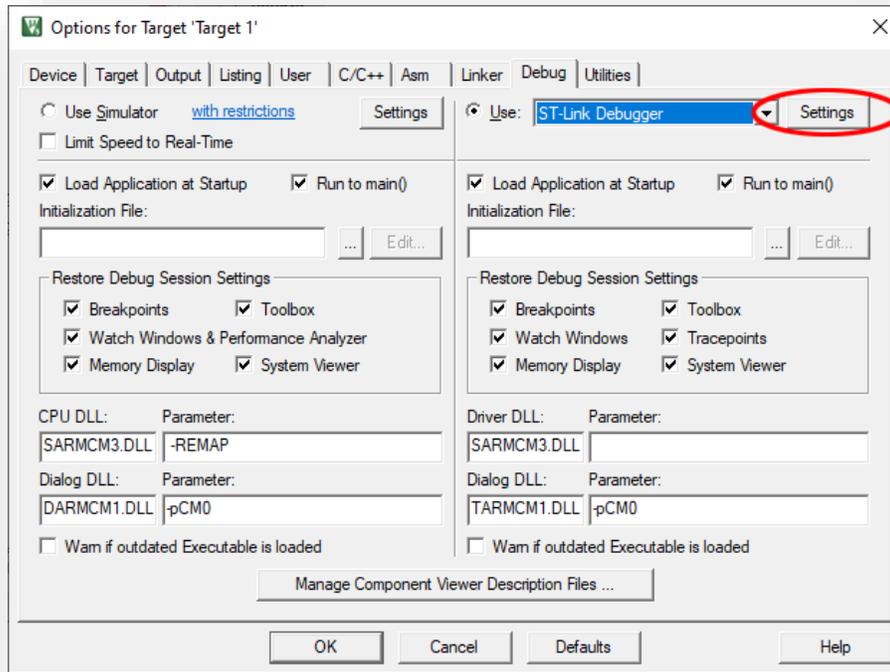


20. Click the “Options for Target...” button again as in Step 15.

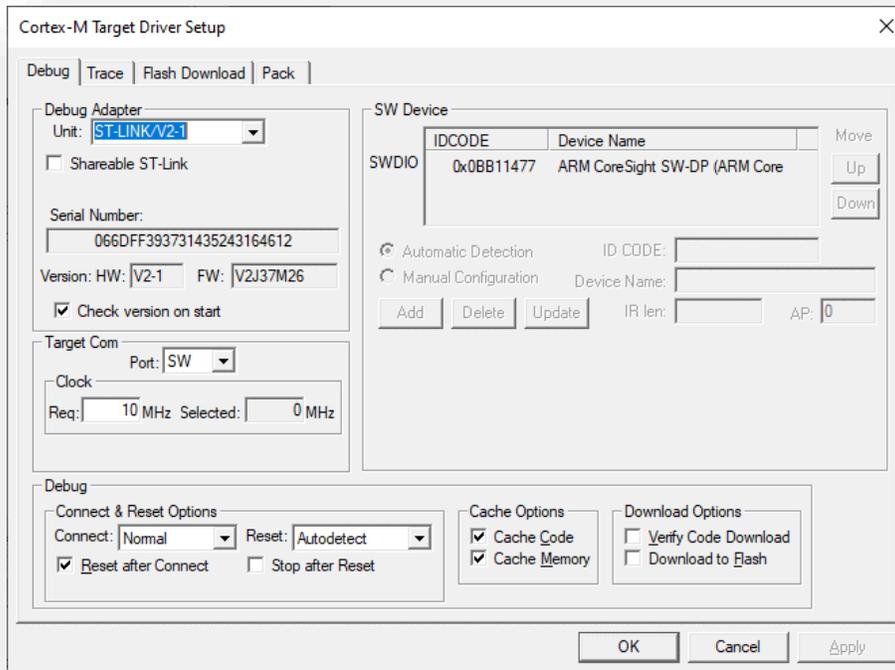
21. Select Debug tab. Select “ST-Link Debugger” from the pull-down menu. Click the radio button before “Use:”.



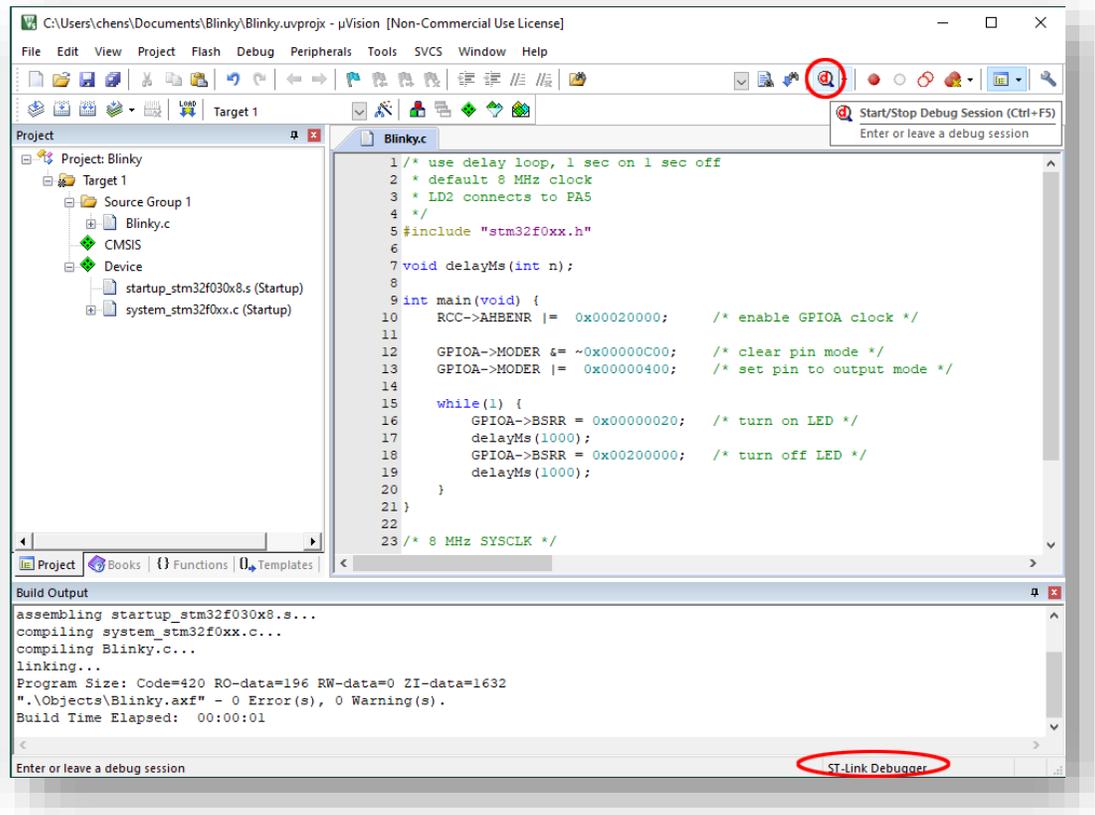
22. Verify that “Load Application at Startup” and “Run to main()” are checked. Click the “Settings” button.



23. In the next dialog box, if the board is properly connected to the computer, the debug interface information should be filled in the dialog box automatically. You should be able to see the Serial Number, the Target Com Port is “SW”, and the SW Device should be found with an IDCODE. Click the “OK” button to close it then click the “OK” button to close the Options dialog box.



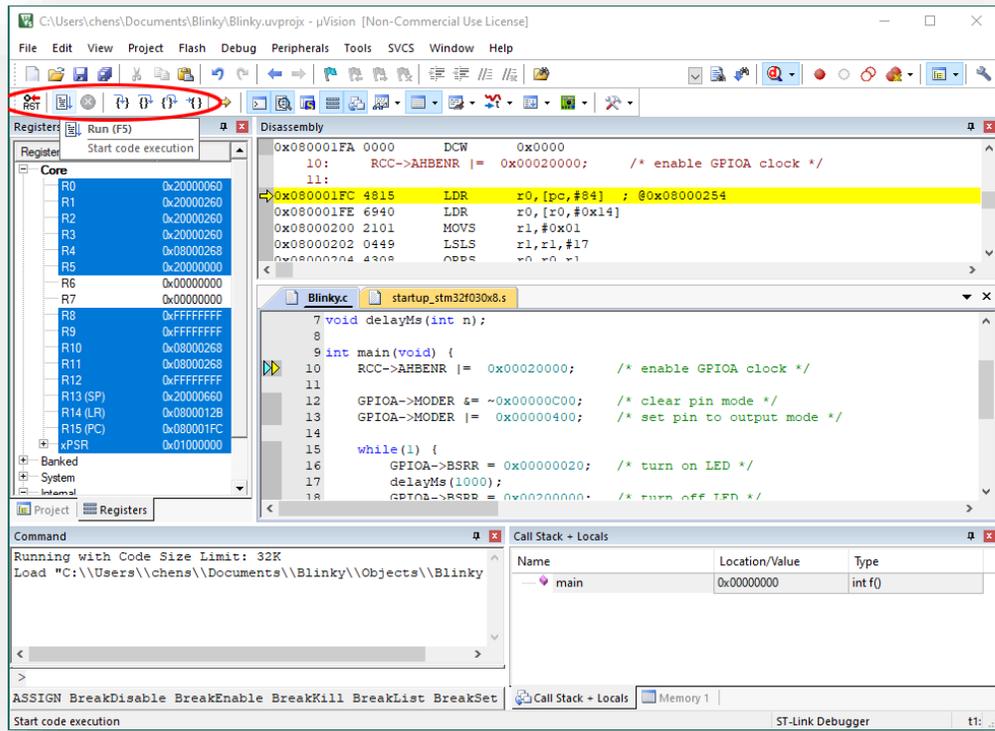
24. The selected debug interface (ST-Link) should appear at the lower right corner. Click on the “Start/Stop Debug Session” button to launch the debugger.



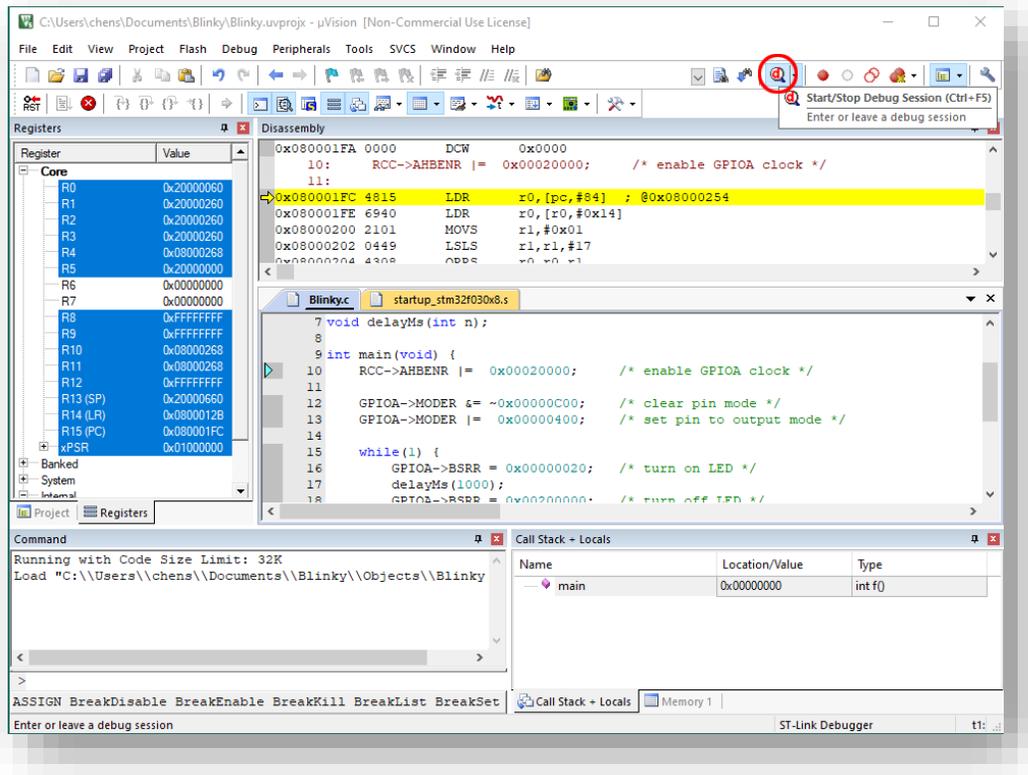
25. The programmer progress bar appears at the bottom of the window. When done, the message “Running with Code Size Limit: 32K...”¹ should appear in the Command pane. The uVision IDE should change into debug perspective. The left pane displays the CPU registers. The right pane is split into Disassembly and source C code panes. The most commonly used debug buttons are at the beginning of the second row: Reset, Run, Stop, Step In (the function), Step over (the function), Step out (of the function), and Run to Cursor Line.

¹ We are using a free non-commercial version, which has a 32K code size limit.

Click the “Run” button and the program will start running and the LED labeled LD2 blinks.



26. To stop the debug session and return to the build project perspective, click on the “Start/Stop Debug Session” button.



27. Congratulations! You have successfully finished the first programming project for the STM32F030 Nucleo-64 board using Keil MDK v5 uVision IDE!