ECE3810 Microprocessor System Design Laboratory

Lab 0: Introduction and Warm-up Lab Practice

School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen

2024-2025 Term 1

1. Introduction

Welcome to ECE3810 Microprocessor System Design Laboratory. The objective of this document is to provide you with

- Basic information of ECE3810
- Grading policies
- Report format
- General lab policies
- Guide for Lab 0

In the first lab practice, you will learn to build a simple project, compile and download to the MDK (Microcontroller Development Kit) board.

Description

Through a series of experiments, this course will strengthen students' capability to use a Cortex-M3 based microprocessor platform to implement an embedded system that utilizes interrupts, timers, keys, beeper, communications port, color display, and other peripherals.

Instruction

- Instructor
  Prof. Tin Lun LAM (tllam@cuhk.edu.cn)
- Teaching Assistant
  Yuxiao TU (116010210@link.cuhk.edu.cn)
  Jindi ZHANG (224010186@link.cuhk.edu.cn)
  Weijia WANG (224010092@link.cuhk.edu.cn)
  Wentao YE (221019027@link.cuhk.edu.cn)
  Chongyang WAN (223010167@link.cuhk.edu.cn)
  Yan YANG (224010146@link.cuhk.edu.cn)
  Junguang YAO (224010136@link.cuhk.edu.cn)
  Yanbing MO (223010149@link.cuhk.edu.cn)
- Venue
  Research Building A 207
  Research Building A 208
- Time
  L01: Monday 15:00-17:50
  L02: Wednesday 15:30-18:20

Prerequisite

- CSC1001 Introduction to Computer Science: Programming Methodology

Co-requisites

- ECE3080 Microprocessors and Computer Systems

Reading

- Required
  - *Lab handouts*
  - *STM32F103xx Reference Manual RM0008*
- Reference
  - N.A.

■ Lab Contents

There are altogether 6 labs, spanning different number of weeks. The contents are shown in Table 1.

Table 1. Seven Labs

Week	Experiment Content
1	Lab 0: Introduction and Warm-up Lab Practice
2-3	Lab 1: GPIO (General Purpose Inputs and Outputs)
4-5	Lab 2: USART (Universal Asynchronous Receiver/Transmitter)
6-7	Lab 3: FSMC (Flexible Static Memory Controller)
8-9	Lab 4: EXTI (External Interrupts)
10-11	Lab 5: Timer

■ Grading Policies

Lab 0 is a warm-up lab and will not be graded. Lab 1-5 are supposed to be completed within two 3-hour sessions in two consequent weeks. Each Lab has an equal weight for grading. The overall grade consists of two parts in the table below.

Table 2. Final Score

Parts	Percentage
Lab performance	60%
Lab report	40%

■ Lab performance

There will be a number of stages in each lab. When you have completed each stage within the lab sessions, demonstrate it to the instructor or TAs. We will sign on the check table to indicate that you have completed the stage. When you complete all stages, you will get the full marks for that lab. **Failing to demonstrate stages within the lab session will not get the corresponding marks.**

■ Lab report and source code

Lab 0 does not need to have a report. For Lab 1 - 5, the report's grading will be based on the rubric below.

- Correct format [20%]
- Content (i.e. items requested in lab, etc.) [50%]
- Presentation (i.e., neatness, language, level of effort) [10%]
- Source code (correctness, adequate comments, etc.) [20%]

Hand in softcopies of lab reports, and zip the required source code files, and upload to Blackboard in due course.

■ Report Format

Lab 1-5 should have one report each. The report should consist of the following parts.

1) **Cover page:** You should include your name, the course number and title, and the date.

2) **Design:** This section contains the designed program flowchart, an explanation of some key source code, etc.

3) **Results:** Use this section to discuss the observations you made during the lab.

4) **Conclusion:** This section should briefly summarize what you have done, and provide some insight you have learned.

Note: if you have any figure or table in the report, provide the caption with figure/table number, similar to an academic paper.

An Important Note on Plagiarism!

Students shall pay attention to University policy and regulations on honesty in academic work, and to the disciplinary guidelines and procedures applicable to breaches of such policy and regulations. Plagiarism is considered as a disciplinary offence which can result in reduced grades, failed subjects, and suspension from the University. ***If plagiarism is detected in any image, data, answer to question, etc., that reports mark will be 0!***

Hence, **DO NOT** let others copy your work.

■ General Lab Policies

1) Food and drink are NOT allowed in the lab.

2) Never, ever modify or attach any high-voltage equipment.

3) Turn off all the devices and clean up your work area before leaving.

2. Warm-up Lab Practice

■ Basic Equipment and Software

In this lab, you will use the following equipment and software.

1) Project board: ALIENTEK Warship
   ALIENTEK Warship project board (Figure 1) adopts STM32F103ZET6, which is an ARM®-based 32-bit Cortex®-M3 MCU (microcontroller unit). This MCU is produced by STMicroelectronics, commonly called ST, which is a French-Italian multinational electronics and semiconductor manufacturer headquartered in Geneva, Switzerland.

Figure 1. ALIENTEK Warship Project Board

The series part number breaking down is shown in Figure 2 below. STM32F103ZET6 has 512K bytes of Flash memory, with 144 pins in LQFP package.

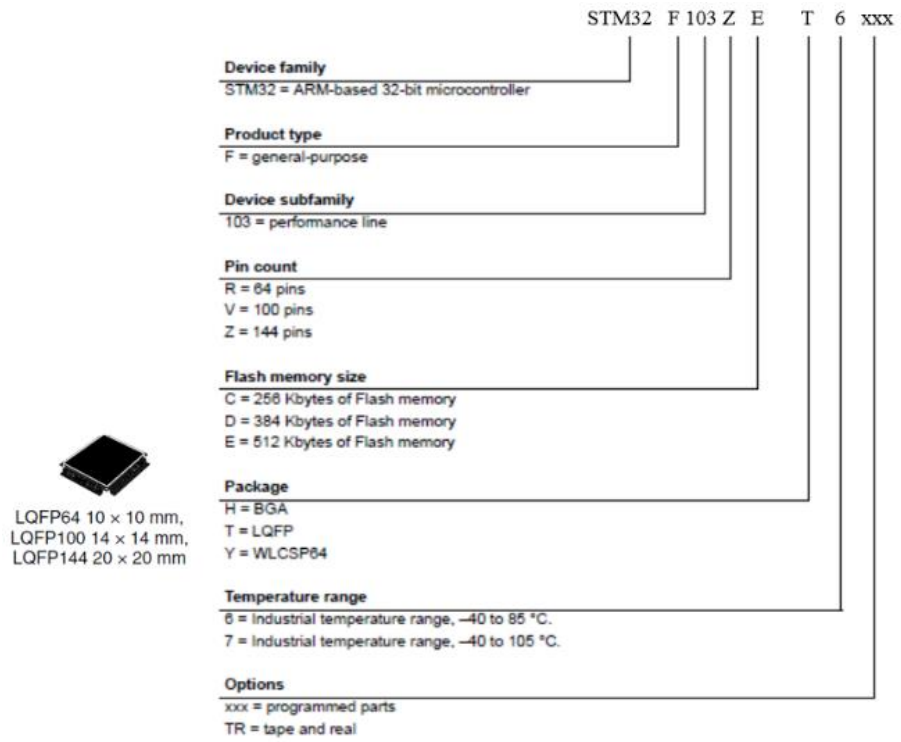


Figure 2. Series part number break down of STM32F103ZET6

2) MDK-ARM

MDK-ARM (Microcontroller Development Kit - Advanced RISC Machine), produced by Keil[1], is one of the development tools for 32-bit ARM microcontrollers. MDK-ARM is based on µVision (Windows only). It is already installed on the desktops of the lab. You can also try to download and install it on your personal computer (https://www2.keil.com/mdk5).

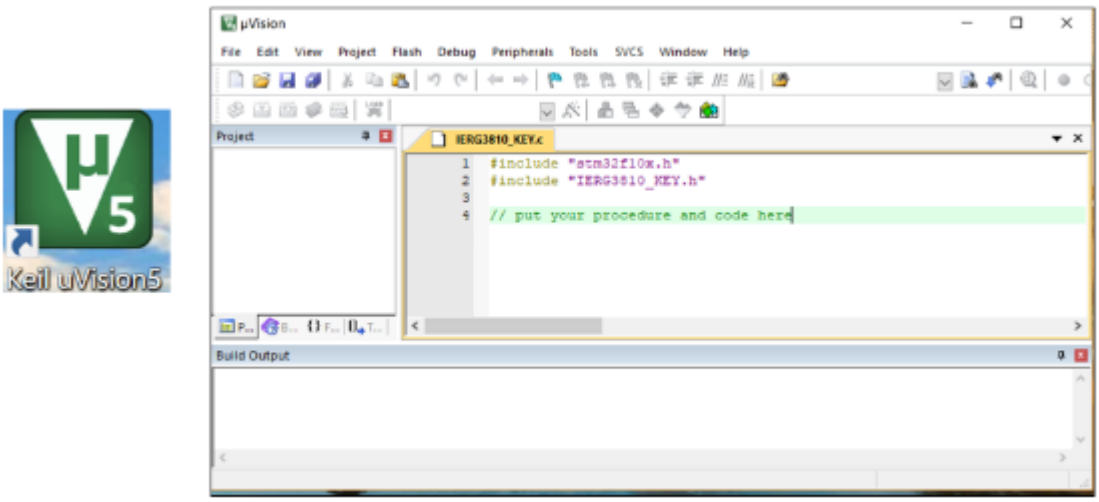


Figure 3. Icon and interface of MKD-ARM

[1] Keil was founded in 1982 by Günter and Reinhard Keil, initially as a German GbR (i.e. simple partnership). In October 2005, Keil (Keil Elektronik GmbH in Munich, Germany, and Keil Software, Inc. in Plano, Texas) were acquired by ARM

- Procedures

  At this moment, you can start to familiarize with how to program STM32 with MDK-ARM

  1) Create an empty project
     - Before starting your project, create a folder to store your project. E.g., Lab0.
     - In your project folder, create a few folders, including Board, Core, Fw_lib, and User.

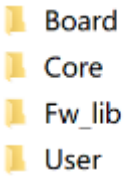


Figure 4. Folders to be created

       - ***Board*** will include .c and .h files that are needed for your project board hardware design.
       - ***Core*** will contain core_cm3.c and core_cm3.h for Cortex-M3. You should not need to modify these files.
       - ***Fw_lib*** will contain standard library files
       - ***User*** will contain some .c and .h files that are created by the user for the project.
     - Download STM32F10x_StdPeriph_Lib_V3.5.0.zip from Blackboard (or download here https://www.st.com/en/embedded-software/stsw-stm32054.html), and copy some essential files to your project.
       - Copy the following files to Lab0\Core\
         - ..\Libraries\CMSIS\CM3\CoreSupport\core_cm3.c
         - ..\Libraries\CMSIS\CM3\CoreSupport\core_cm3.h
         - ..\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\stm32f10x.h
         - ..\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\system_stm32f10x.h
         - ..\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\system_stm32f10x.c
         - ..\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\startup\arm\startup_stm32f10x_hd.s
           (STM32F103ZET6 is a high-density device. So we use startup_stm32f10x.hd.s startup file)
       - Copy the following folders '\inc' and 'src' to Lab0\Fw_lib\
         - ..\Libraries\STM32F10x_StdPeriph_Driver\inc
         - ..\Libraries\STM32F10x_StdPeriph_Driver\src
       - Copy the following files to Lab0\User\
         - ..\Project\STM32F10x_StdPeriph_Template\stm32f10x_conf.h
         - ..\Project\STM32F10x_StdPeriph_Template\stm32f10x_it.c
         - ..\Project\STM32F10x_StdPeriph_Template\stm32f10x_it.h
     - Create an empty project in Keil µVision.
       - Open Keil µVision on your desktop, and follow the steps below.

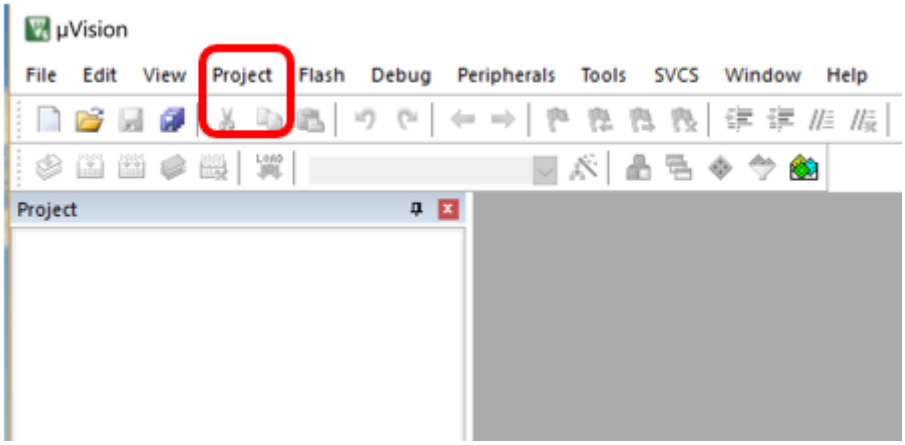


Figure 5. Click Project => new µVision Project

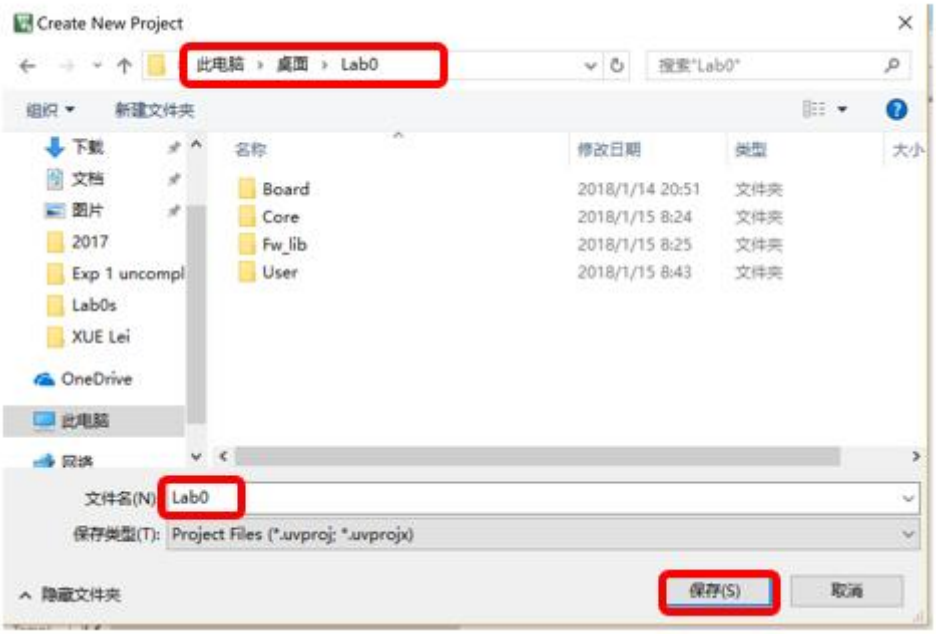


Figure 6. Select the directory and name your project file

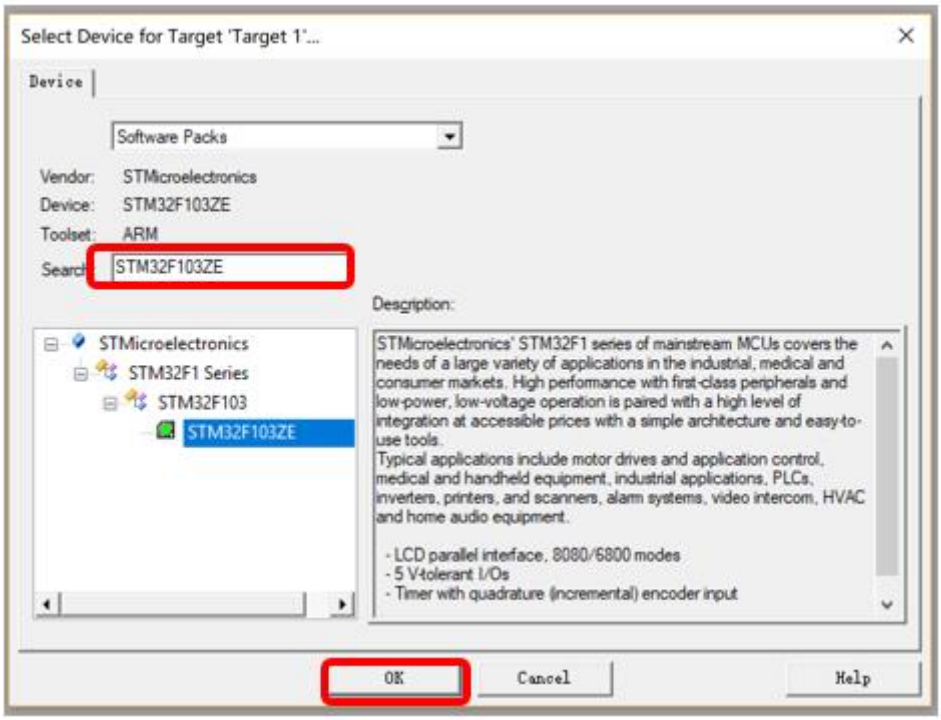
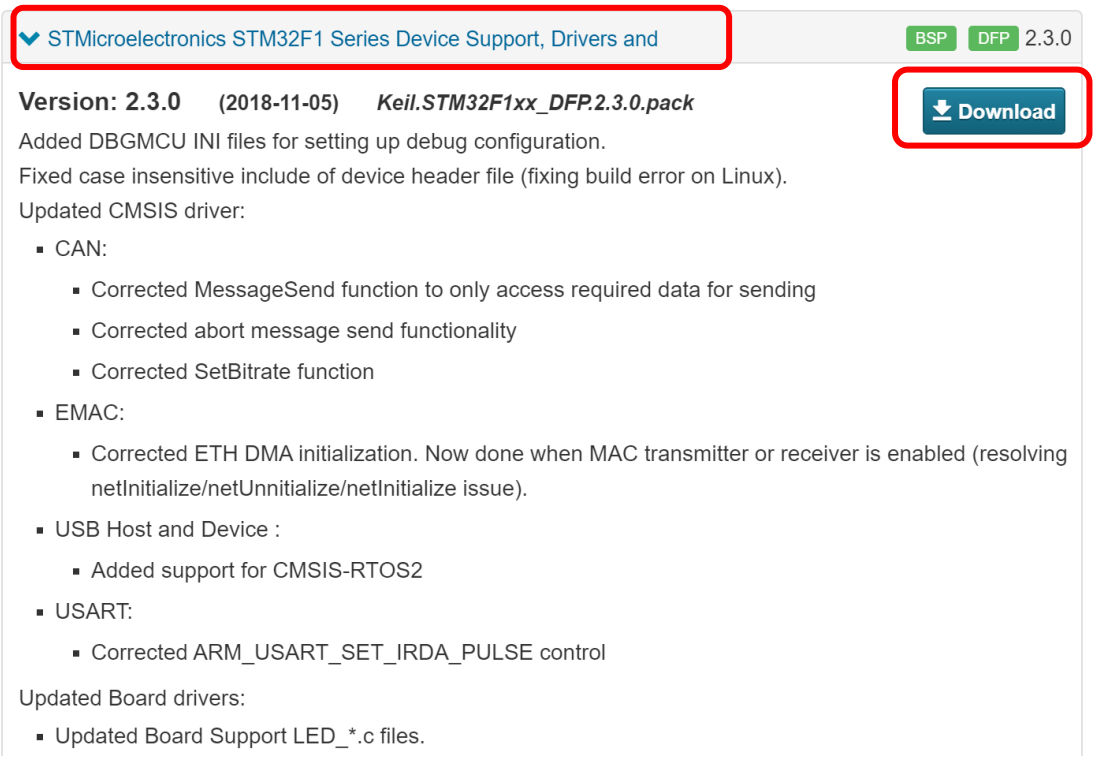


Figure 7. Select the device STM32F103ZE

Remark: If you cannot find STM32F103ZE, download and install the corresponding package (https://www.keil.com/dd2/pack).



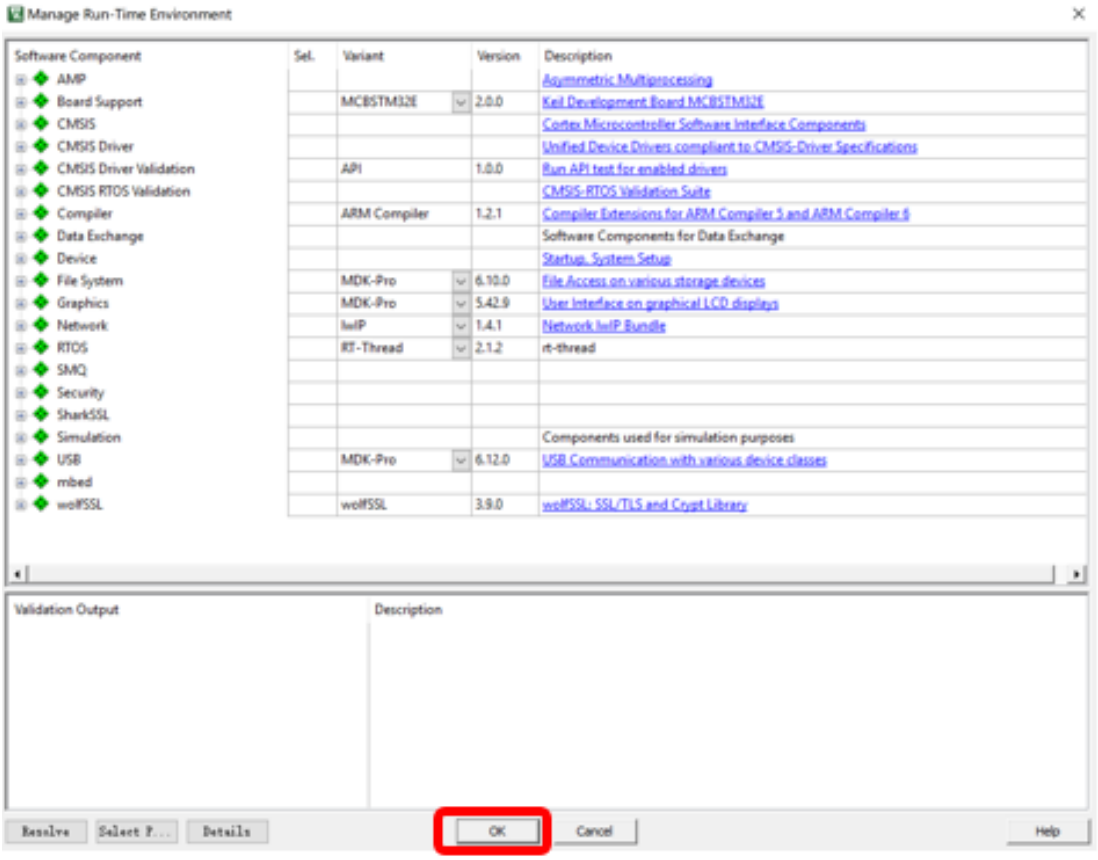
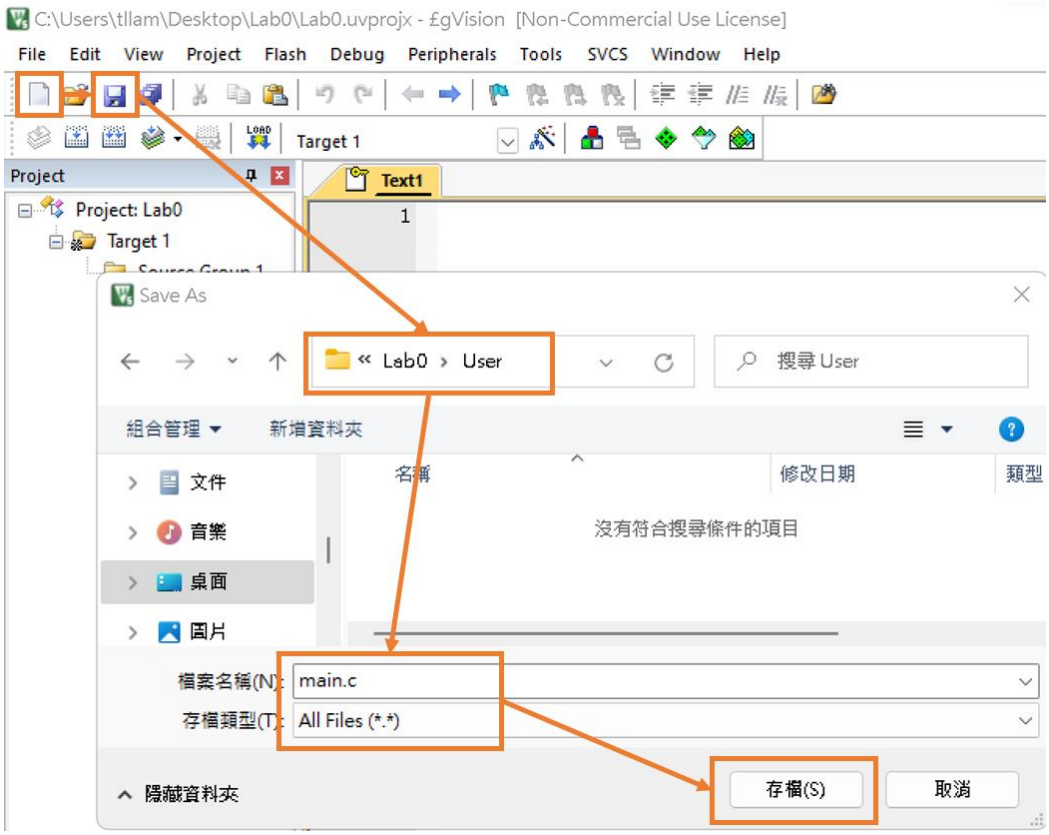


Figure 8. Click OK in the Manage Run-Time Environment panel

- Create a file named main.c under Lab0\User\.



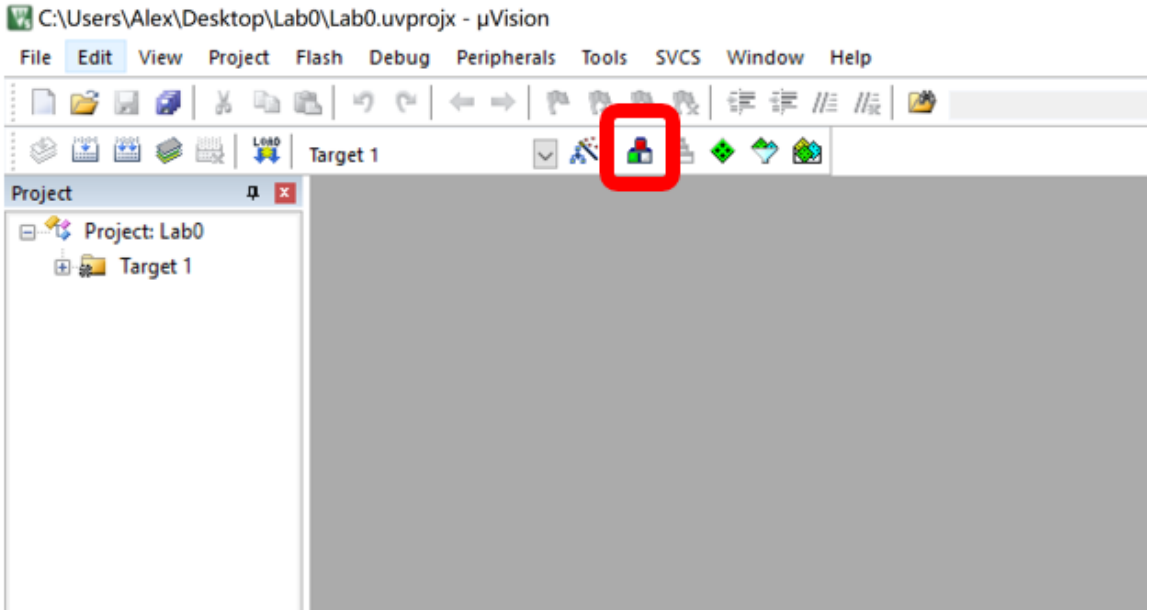


Figure 9. Click the "File Extensions, …" icon

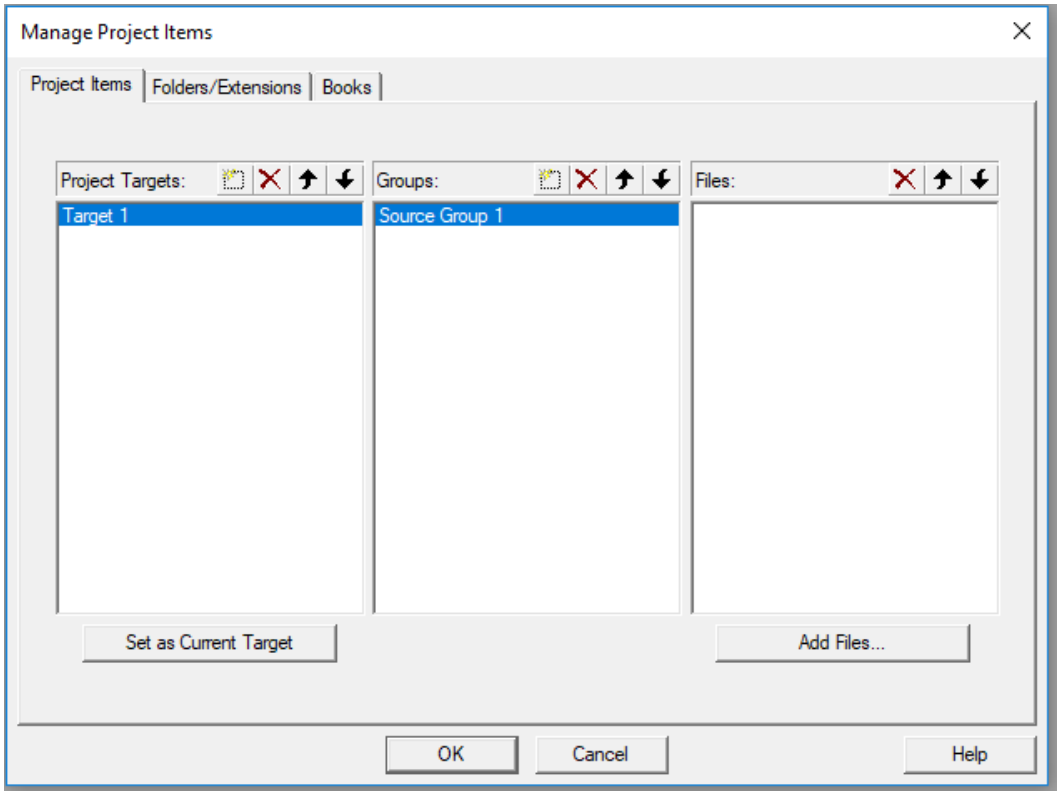


Figure 10. The default "Manage Project Item" dialogue box

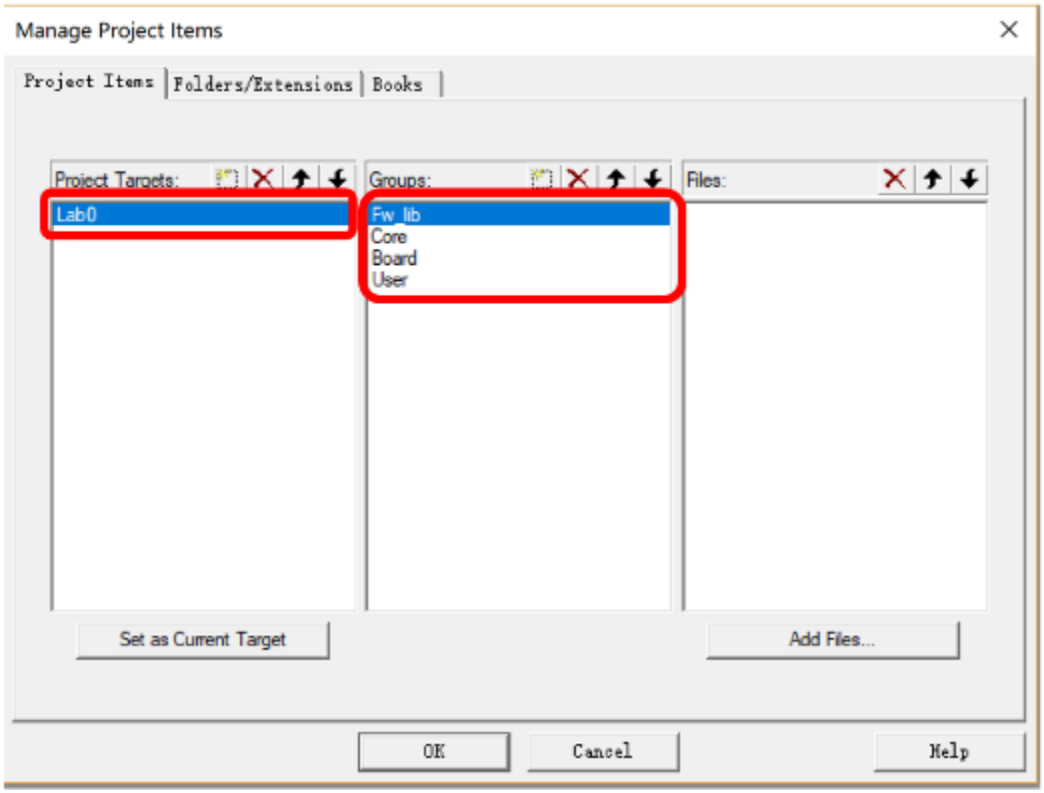


Figure 11. Double click to re-name “Project Targets” and “Groups”

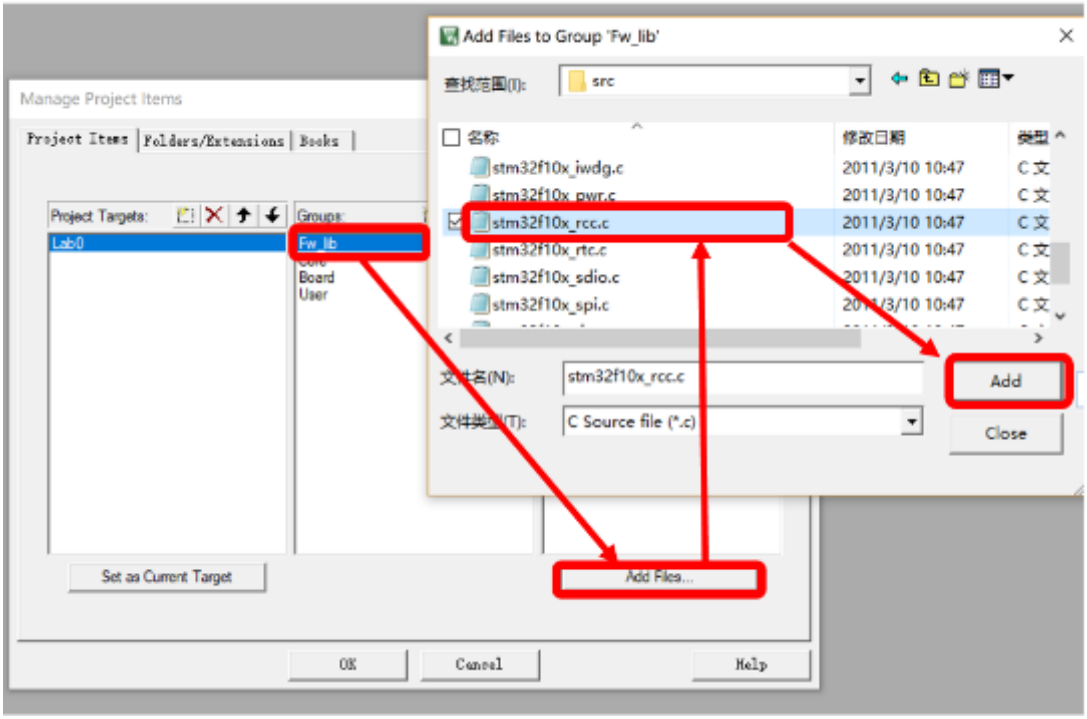


Figure 12. Add files to Fw_lib

You can include all .c files from Fw_lib. Minimally includes stm32f10x_rcc.c, which is the system clock configuration file.

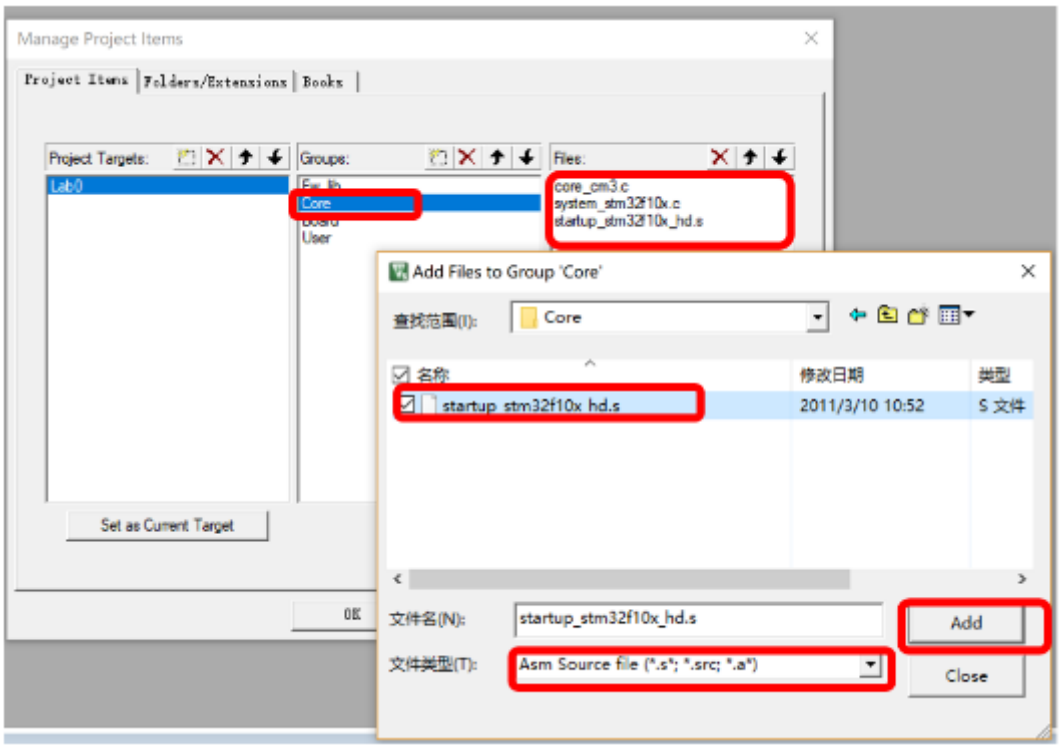


Figure 13. Include .c and .s files into Core separately

(Remark: You can only select .c and .s separately.)

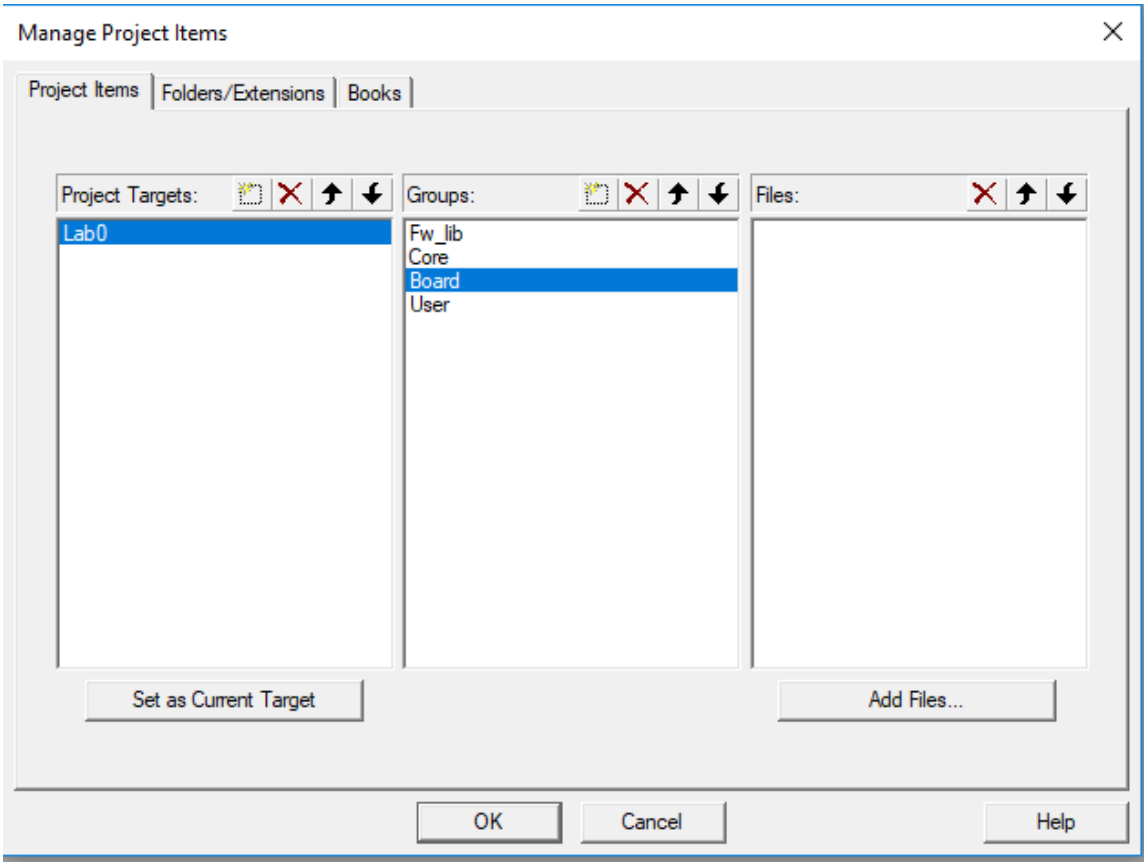


Figure 14. No need to include any file to Board in this experiment

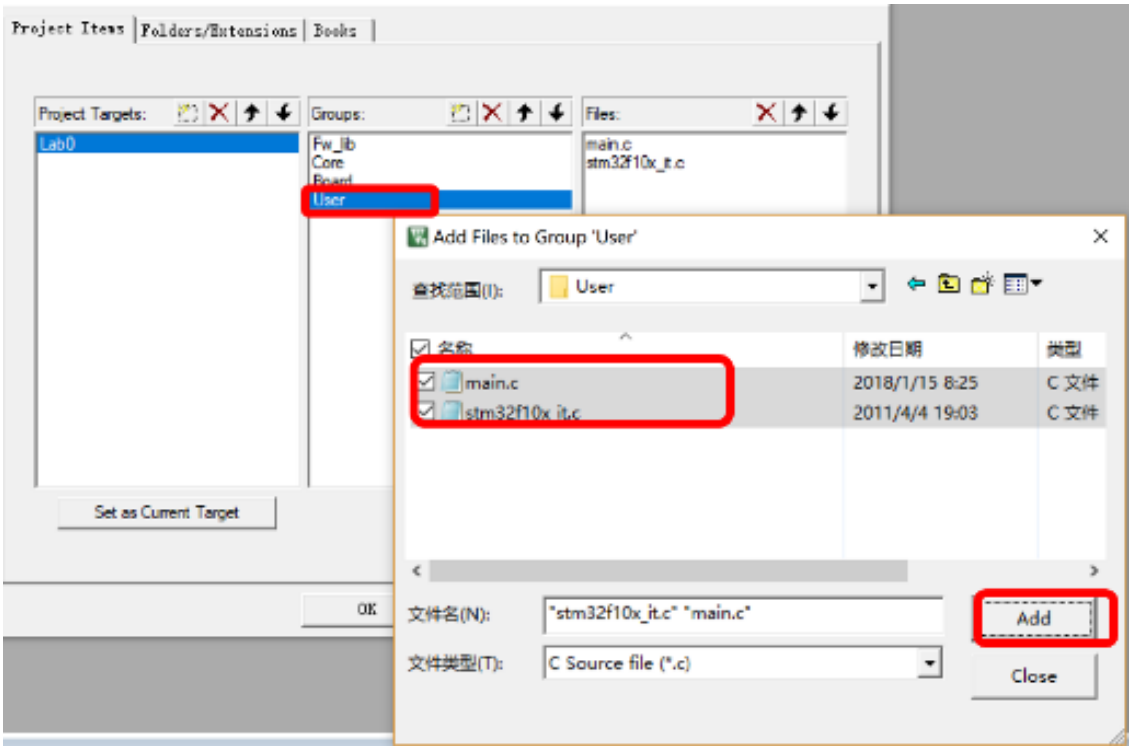


Figure 15. Add .c file from the User folder into User

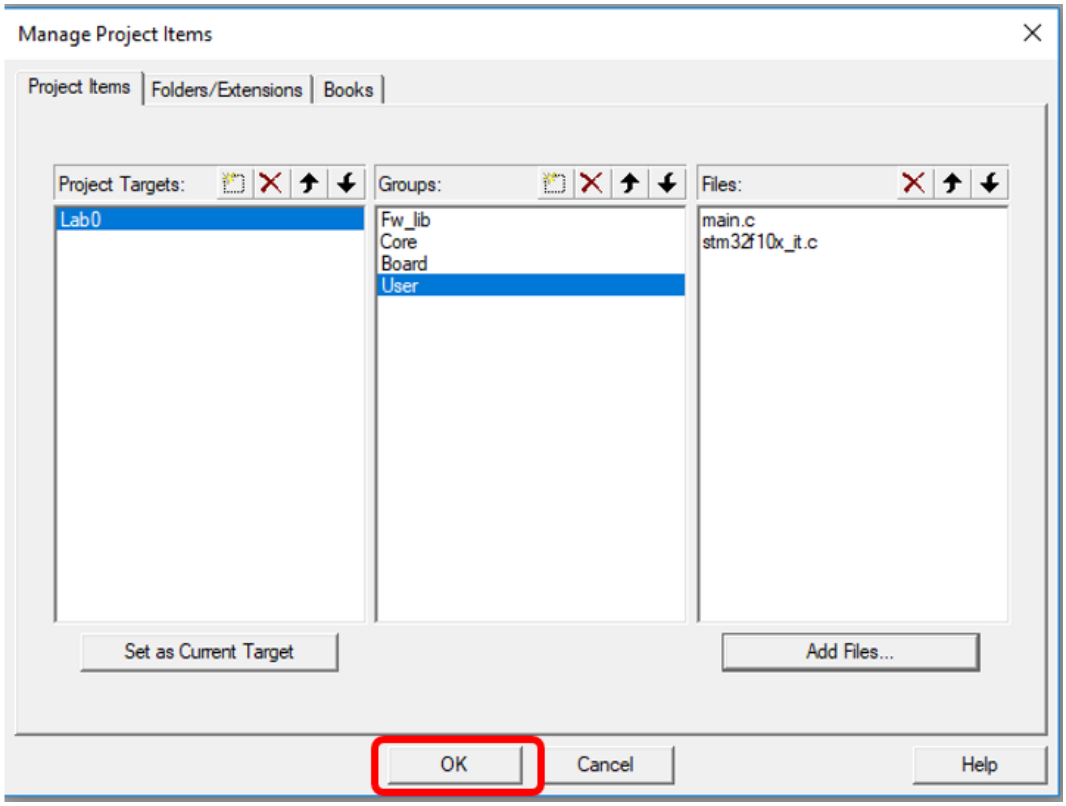


Figure 16. After adding files, click “OK”

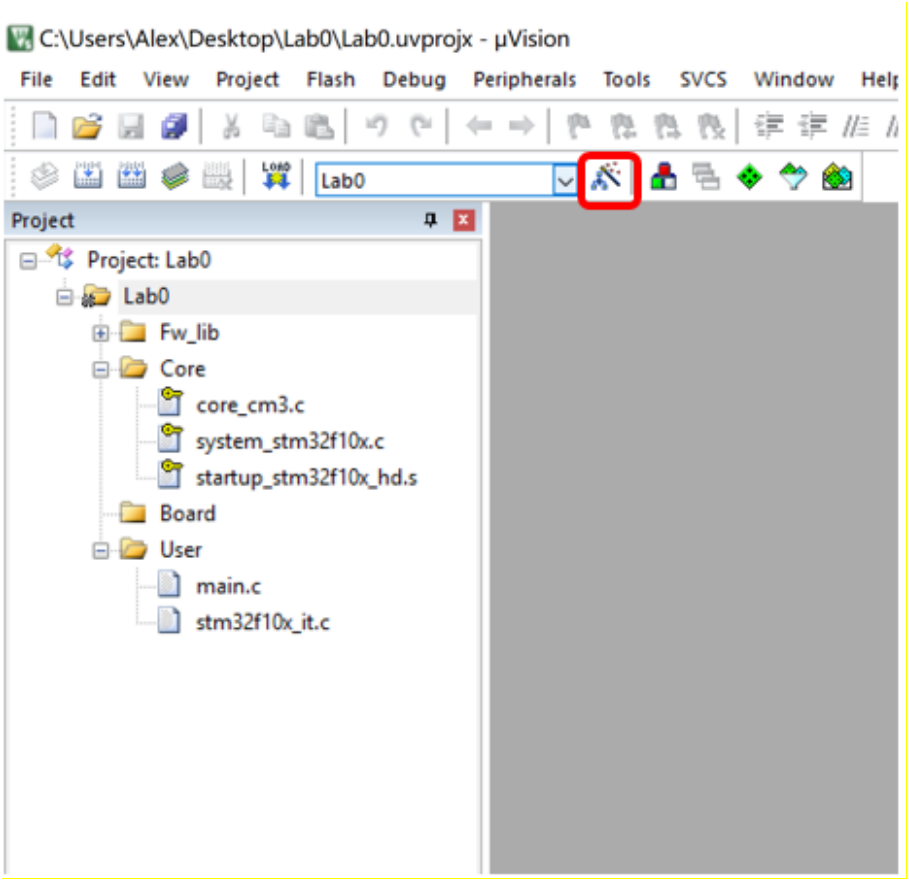


Figure 17. The view after adding files. Then click “Options for Target…”

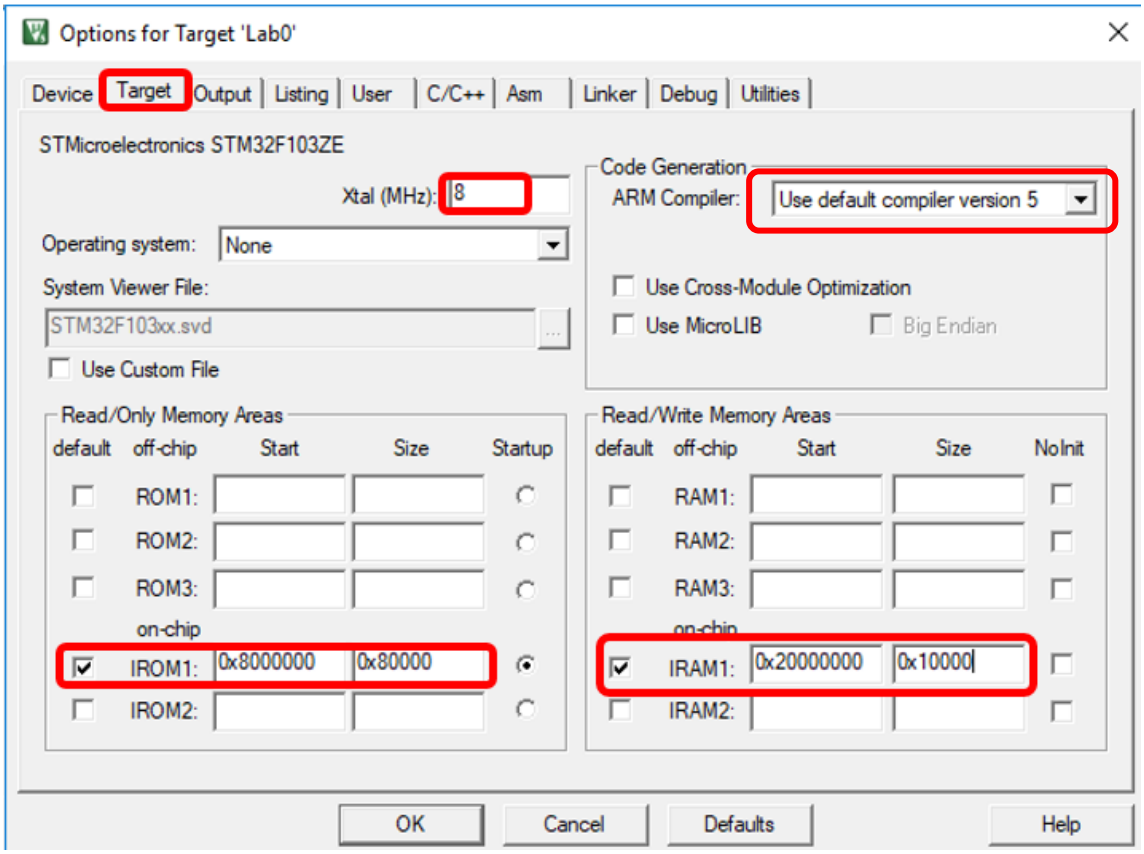


Figure 18. Set Xtal to 8 Hz, and check chip memories selected by default

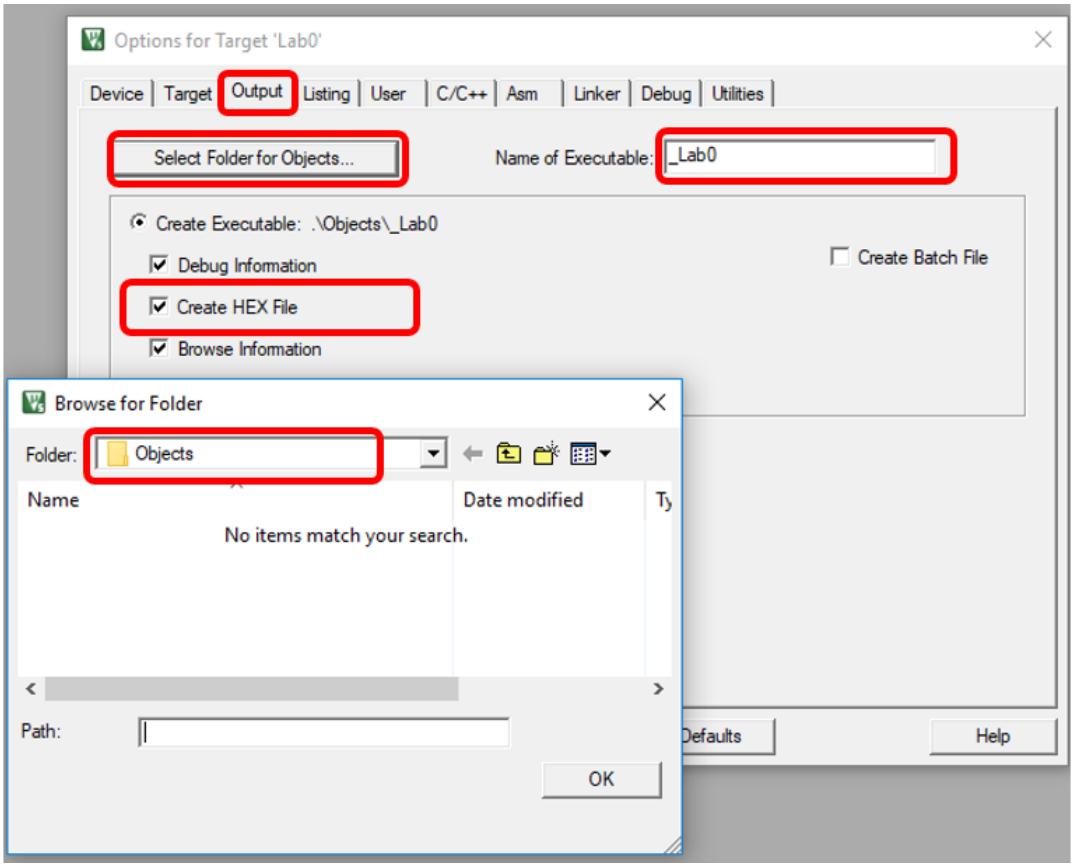


Figure 19. Set the Output tab

Select the "Objects" folder. This is a folder created by the software. Modify the executable file name with a prefix "_". This will help you to find the file easier. Tick "Create HEX File".

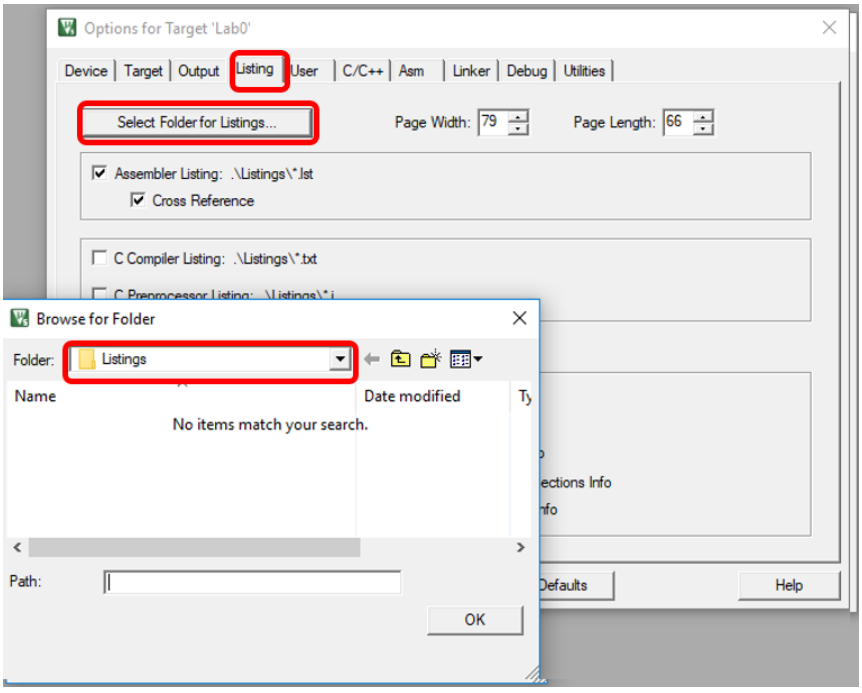


Figure 20. The List tab

In the Listing tab, the folder selected should be “Listings”, which is also created by the software.

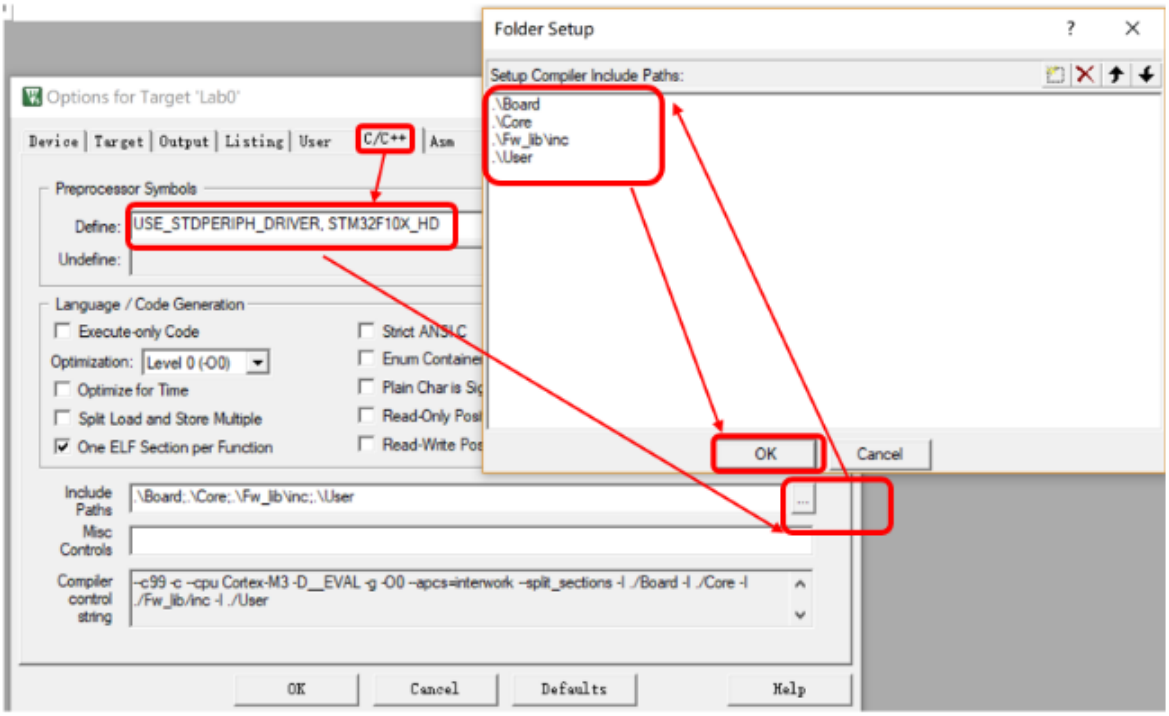


Figure 21. C/C++ tab

Key in “USE_STDPERIPH_DRIVER, STM32F10X_HD” and include the 4 folders in the figure above.
Now, you have set up an environment for Cortex-M3 development, and can modify the file main.c for your coding.

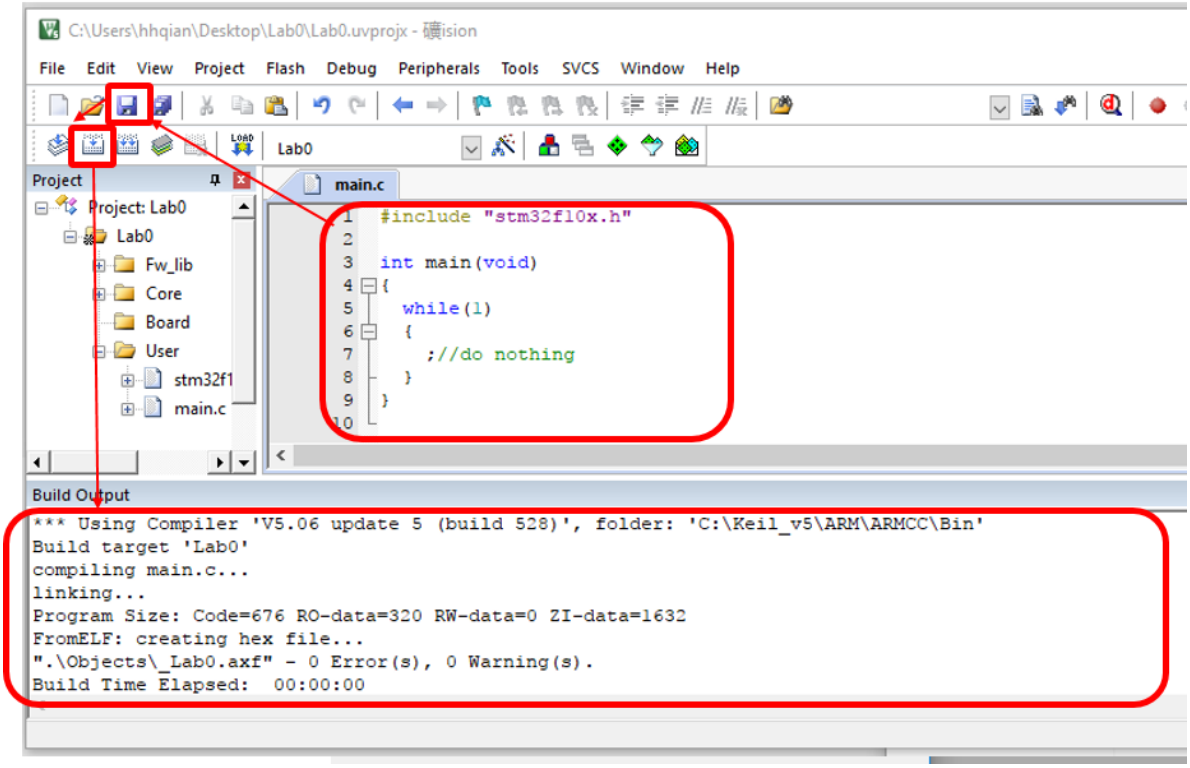


Figure 22. Modify the coding in main.c, save, and click Build button to compile the program

In the Build Output window, you shall have 0 error and 0 warning. Then you can find the .hex file in the Object folder.

Figure 23. The hex file generated

2) Download binary file

The next stage is to download the coding to the development board.

Hardware Preparation:

- On the board, find PA9(RXC) and PA10(TXD), and check that their jumper caps are closed.

Figure 24. RXD and TXD caps

- Connect USB_232 (left-lower part of the board) to computer. Press the power switch. There is no need to connect the external power in this project.

Figure 25. Steps on the board before download

Software Preparation:

- **CH-341SER.zip** for chip CH340, which is a converter chip for USB-to-serial on your project boards. The driver for CH-340 has been installed in lab's PCs. If you need it, you can obtain it at http://www.wch.cn/download/CH341SER_ZIP.html.
- **FlyMcu.exe**. Find this program in the computer. It can also be downloaded from http://www.mcuisp.com/software/FlyMcu.rar

Download:

- Open FlyMcu.exe. Click EnumPort to discover available serial ports, and select the correct port with CH340.

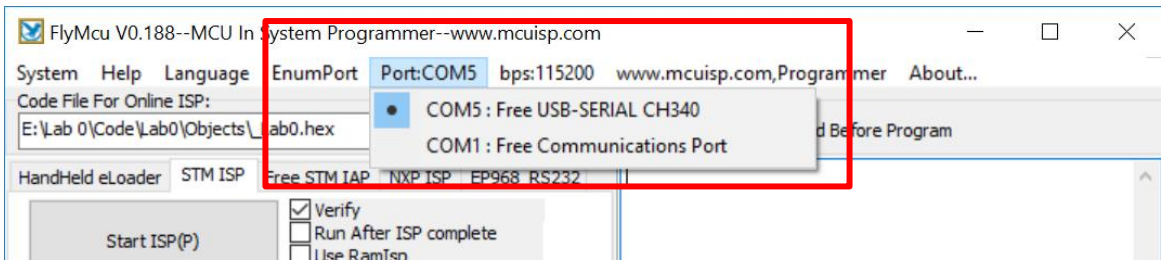


Figure 26. FlyMcu setup steps

- Then select the location which stores the binary file. Choose “Reset@DTR Low(<-3V), ISP@RTS High”.

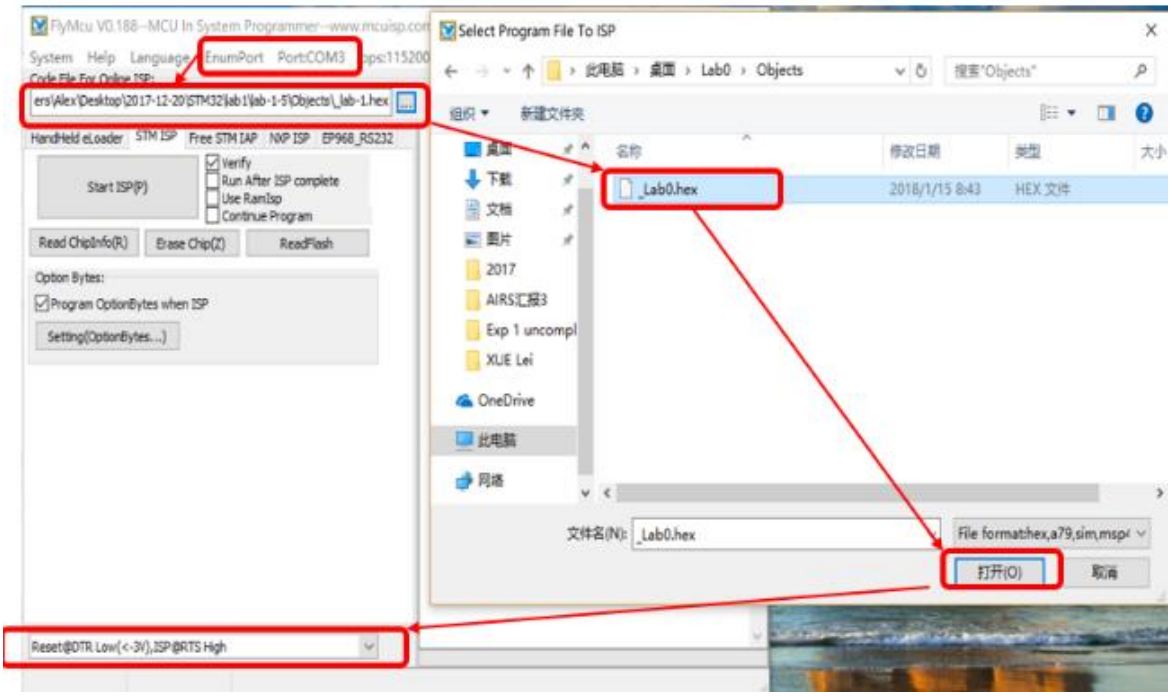


Figure 27. FlyMcu setup steps

- In the STM ISP tab, click “Start ISP(P)”. The program will be downloaded to the board. If FlyMcu gets stuck, press the Reset button (red button in the right lower part of the board).

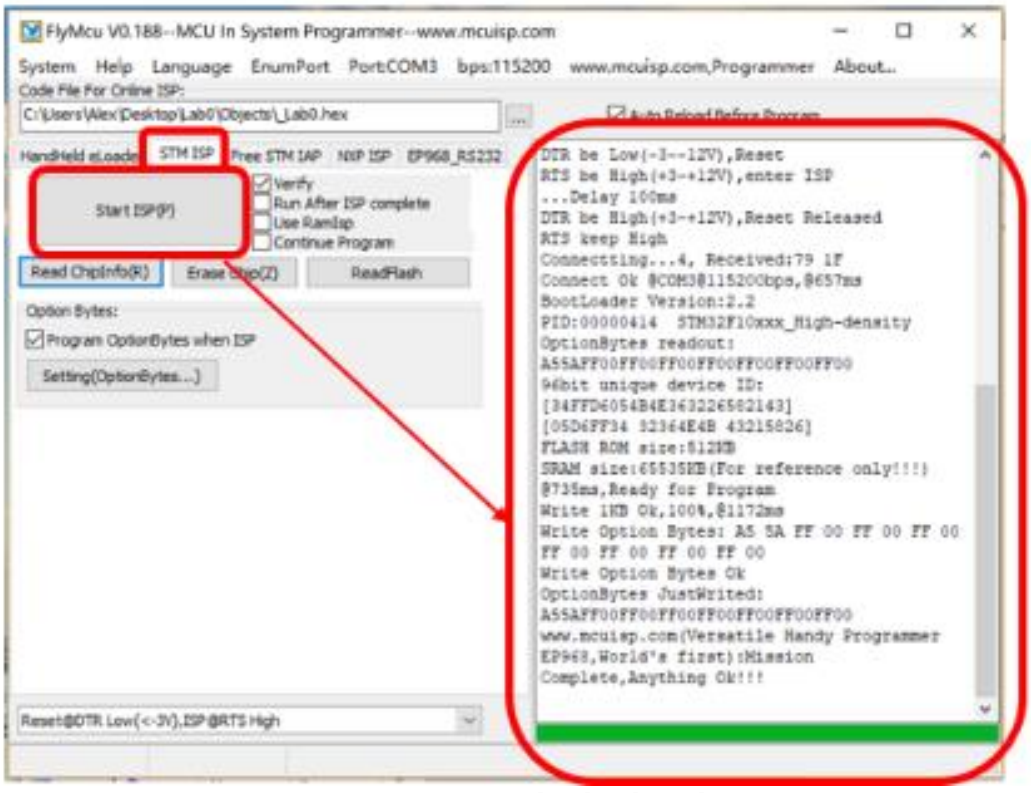


Figure 28. Download to board

3) After downloading, press the Reset button on the board to start the program. Of course, you will see nothing change on the board, as the coding does nothing. In Lab 1, you will program the board with input/output.

Before you leave the lab:

1. **Save the entire 'Lab 0' folders to your usb drive or send them to your email account for future use. Don't keep them in the computer.**
2. **Turn off all the devices and tidy up your desk.**