UNIVERSITY OF MICHIGAN JOINT INSTITUTE 交大密西根学院

Ve373 Microprocessor Based System Design

Lab 1 Development Tools for STM32 MCU

OBJECTIVE

- **Build an environment for STM32 programming & debugging.** This includes the installing of:
  1. **STM32CubeMX**, a tool with graphic interface that helps you configure a project. Library references and starter files are hard to manage for an embedded project, and STM32CubeMX generates a configured project for you with all those stuffs arranged.
  2. **Keil v5,** a tool for compiling, downloading your code into the STM32 board and debugging.
  3. **ST-LINK V2,** a downloader and its driver. You'll get a physical downloader in the on-site lab, and you may need to install its driver ahead of time.

  Note that Keil only supports Windows. If you use Mac or Linux, you can use **STM32CubeIDE** instead. CubeIDE is an all-in-one solution that covers the function of CubeMX + Keil. Also, when installing CubeIDE, it offers a convenient choice to install ST-LINK driver together.
- Generate and run a simple program on an embedded chip.
- Understand the whole loop of developing a STM32 project.

PROJECT DESCRIPTION

In this lab, you will set up the embedded system IDE for the rest of this course. The lab involves:

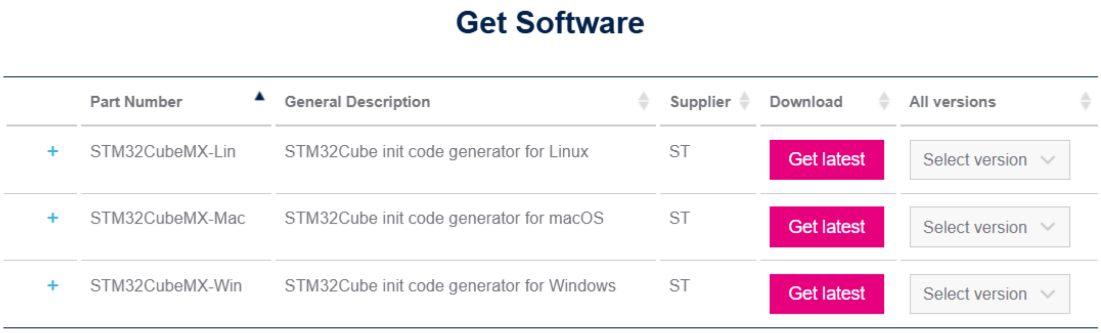
1. Install the software above one by one.
2. Using CubeMX / CubeIDE, create a new project and select the MCU, configure it correctly;
3. Using Keil / CubeIDE, open your project and compile it, then download it to a STM32 board with a ST-LINK downloader.

a. Install Software (Please do this before lab)

If you choose **STM32CubeMX + Keil**, you should:

1. Install STM32CubeMX.

Open the official link STM32CubeMX - STM32Cube initialization code generator - STMicroelectronics. Click "Get Software" (or just scroll down the page) and you'll see this:

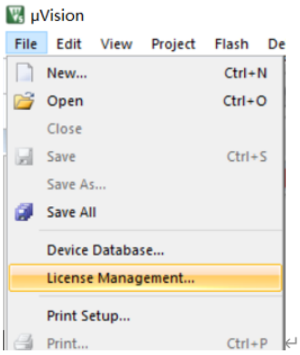


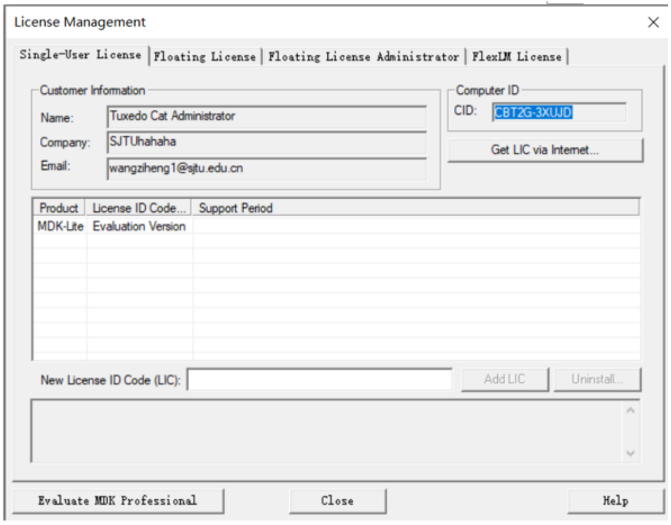
Get Software

	Part Number	General Description	Supplier	Download	All versions
+	STM32CubeMX-Lin	STM32Cube init code generator for Linux	ST	Get latest	Select version
+	STM32CubeMX-Mac	STM32Cube init code generator for macOS	ST	Get latest	Select version
+	STM32CubeMX-Win	STM32Cube init code generator for Windows	ST	Get latest	Select version

Then, choose the latest version for your operating system. (The website will ask you to create an account, it's okay to follow its instructions. After logging in with your account, you can scroll down and see the above page again.) The installation should go smoothly.

2. Install Keil V5 and activate a license.
   In Canvas-Files-Lab-Resources, you'll see 3 files. Please do the following sequentially:
   1) Download & unpack Keil uVision5 MDK V5.20. After installing Keil, you'll be led to a "Release Note Page" and see a "Pack Installer" pop-up window. No need to get scared and just close them anyway.
   2) Download & unpack Keil uVision5 MDK V5.20 Device Support Pack.
   3) Download Keygen.
      Open Keil as administrator. Click "File->License Management" and copy the CID.





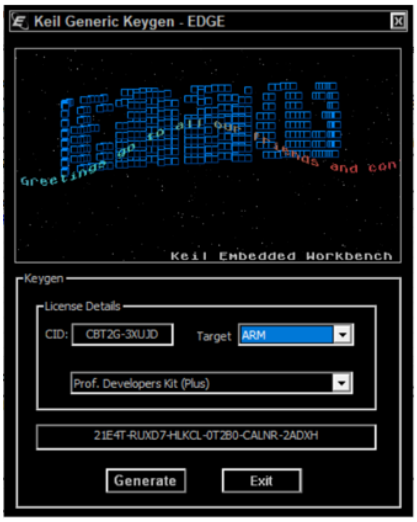
Customer Information
Name: Tuxedo Cat Administrator
Company: SJTUhahaha
Email: wangziheng1@sjtu.edu.cn

Computer ID
CID: CBT2G-3XUJD

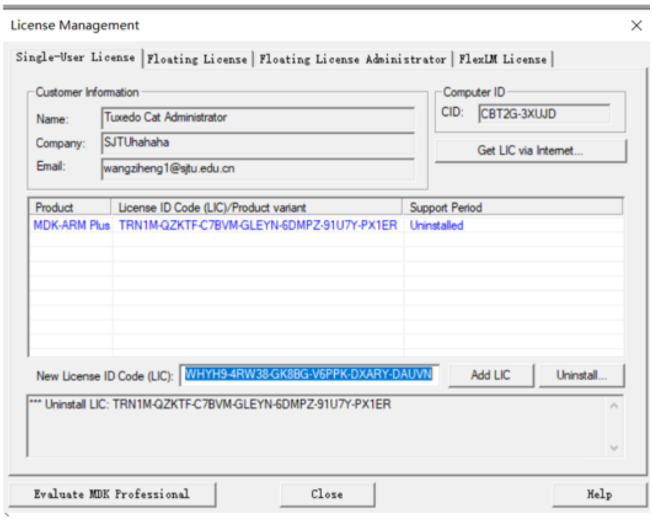
Product	License ID Code...	Support Period
MDK-Lite	Evaluation Version	

New License ID Code (LIC):

Run Keygen. Note that you'll hear **a powerful piece of music** when Keygen is running, and turning down the volume ahead of time is recommended. Paste the CID into Keygen, and **choose ARM as the target**. Then, click "Generate" and copy the license code you get.


Keil Generic Keygen - EDGE
Greetings go to all our friends and con
Keil Embedded Workbench
Generate
Exit

Keygen
License Details
CID: CBT2G-3XUJD
Target: ARM
Prof. Developers Kit (Plus)
21E4T-RUXD7-HLKCL-0T2B0-CALNR-2ADXH



Customer Information
Name: Tuxedo Cat Administrator
Company: SJTUhahaha
Email: wangziheng1@sjtu.edu.cn

Computer ID
CID: CBT2G-3XUJD

Product	License ID Code (LIC)/Product variant	Support Period
MDK-ARM Plus	TRN1M-QZKTF-C7BVM-GLEYN-6DMPZ-91U7Y-PX1ER	Uninstalled

New License ID Code (LIC): WHYH9-4RW38-GK8BG-V6PPK-DXARY-DAUVN

*** Uninstall LIC: TRN1M-QZKTF-C7BVM-GLEYN-6DMPZ-91U7Y-PX1ER

Go back to "License Management" in Keil. Paste the license code into "New License ID Code", click "Add LIC" and you've cracked Keil.

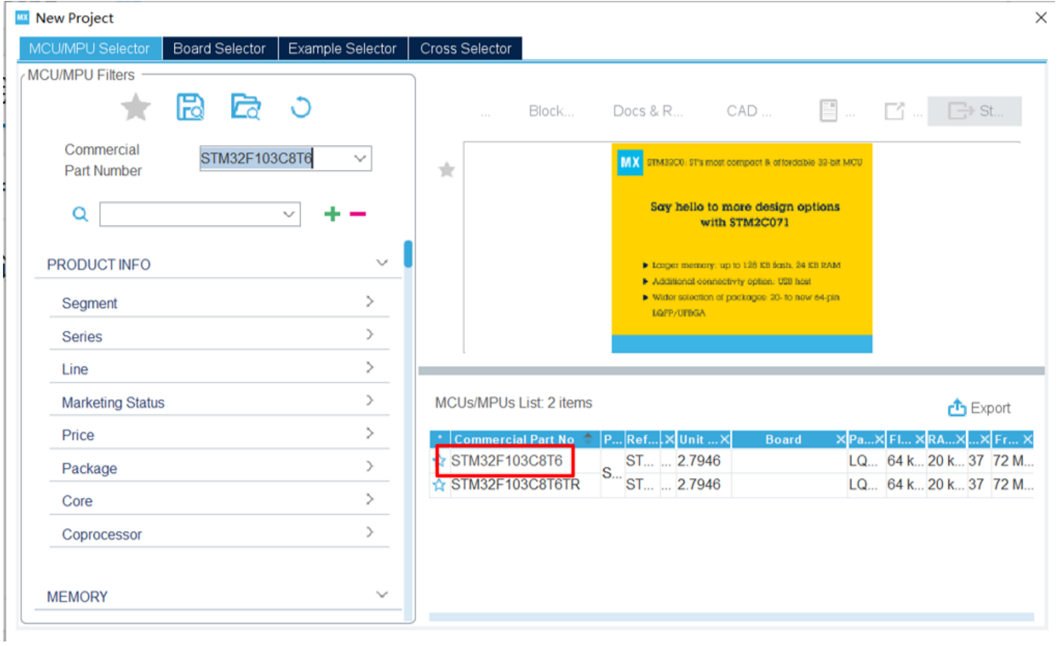
3. Install the driver for ST-LINK at https://jbox.sjtu.edu.cn/l/c16Jfm.

If you choose **STM32CubeIDE** as an all-in-one solution, you should:

1. Just install it.
   The official website is [STM32CubeIDE - Integrated Development Environment for STM32 - STMicroelectronics]() and the installation process is alike to STM32CubeMX, as mentioned above.

b. Write & Run a Simple Program

1. Generate a configured project. The screenshots come from **CubeMX**, and the same procedure applies to **CubeIDE**.
   Click “File -> New Project”. Search for STM32F103C8T6 in “Commercial Part Number” and double-click the item you get.



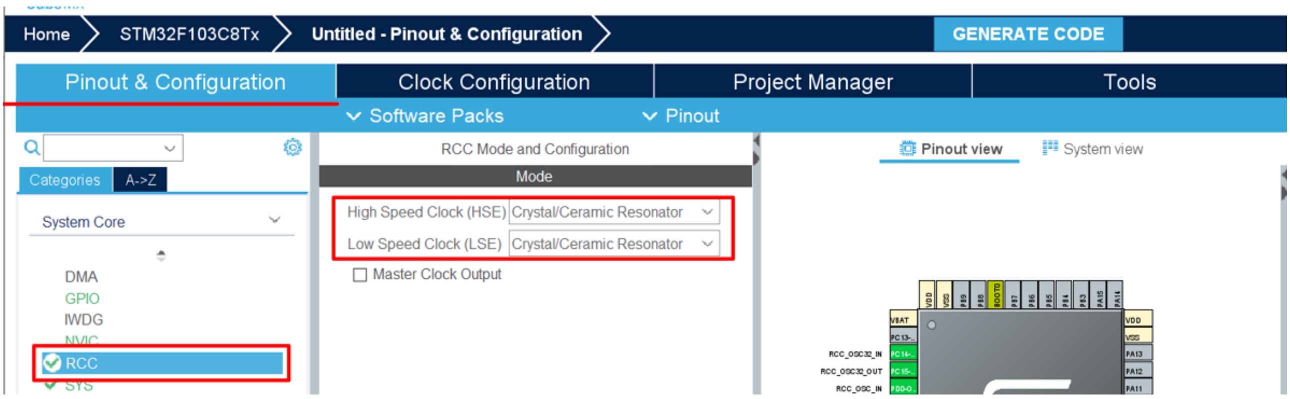
New Project

Commercial Part Number: STM32F103C8T6

MCUs/MPUs List: 2 items

Commercial Part No	P...	Ref...	Unit ...	Board	Pa...	Fl...	RA...	...	Fr...
STM32F103C8T6	S...	ST...	... 2.7946		LQ...	64 k...	20 k...	37	72 M...
STM32F103C8T6TR		ST...	... 2.7946		LQ...	64 k...	20 k...	37	72 M...

   In Pinout & Configuration, go to “RCC” and configure “High Speed Clock” and “Low Speed Clock” as “Crystal/Ceramic Resonator”.

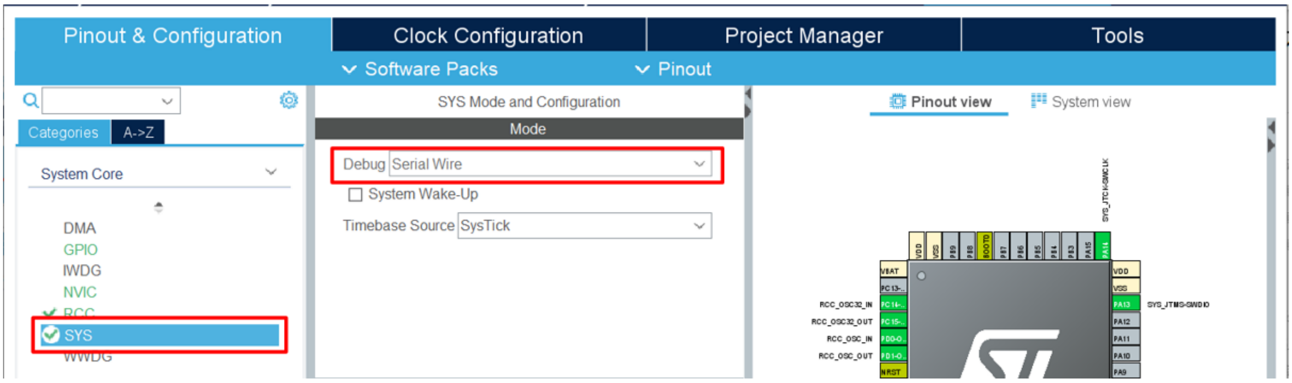


RCC Mode and Configuration

Mode

High Speed Clock (HSE): Crystal/Ceramic Resonator
Low Speed Clock (LSE): Crystal/Ceramic Resonator
□ Master Clock Output

   Then, go to “SYS” and config “Debug” as “Serial Wire”.

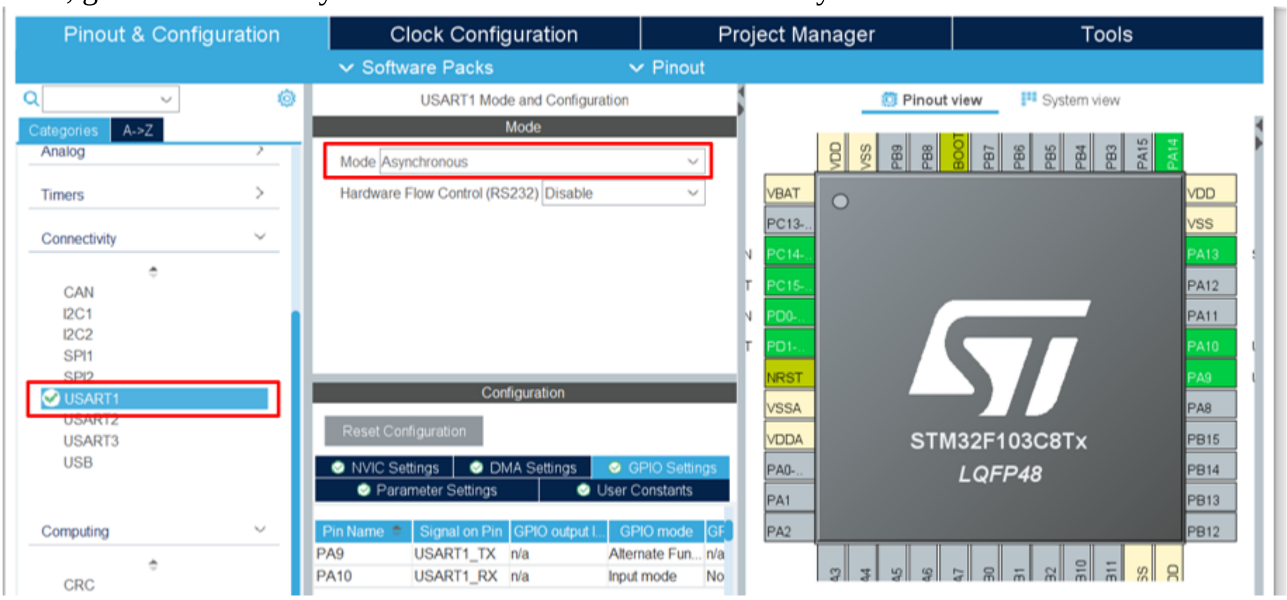


Debug Serial Wire

□ System Wake-Up

Timebase Source SysTick

Then, go to “Connectivity -> USART1” and set “Mode” as “Asynchronous”.

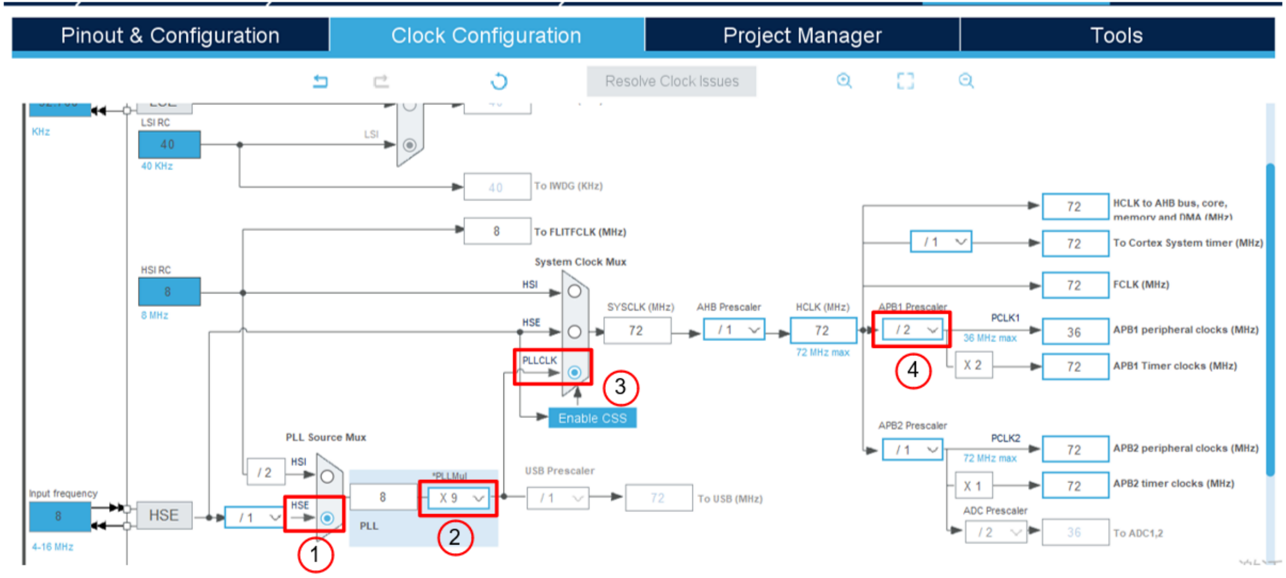


Mode Asynchronous

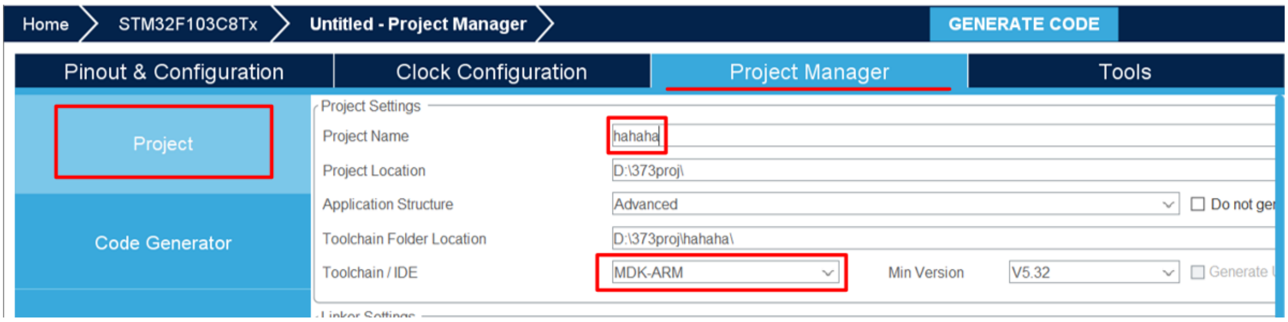
Hardware Flow Control (RS232) Disable

Pin Name	Signal on Pin	GPIO output l...	GPIO mode	GF
PA9	USART1_TX	n/a	Alternate Fun...	n/a
PA10	USART1_RX	n/a	Input mode	No

Now the Pinout are configured, and let’s go to “Clock Configuration”. Configure the clock step-by-step with same settings as below.



Now configurations are down, let’s go to “Project Manager”. In “Project”, name your project and choose “MDK-ARM” as the target toolchain.



Project Settings

Project Name: hahaha

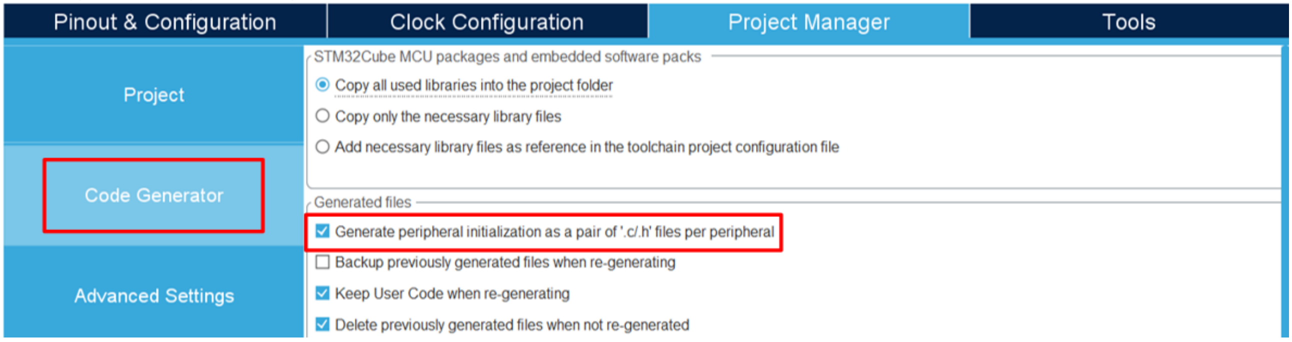
Project Location: D:\373proj\

Application Structure: Advanced ☐ Do not ge

Toolchain Folder Location: D:\373proj\hahaha\

Toolchain / IDE: MDK-ARM Min Version V5.32 ☐ Generate

In “Code Generator”, choose “Generate peripheral initialization as a pair of ‘.c/.h’ files per peripheral”. This is not necessary step but a good habit, since separated files are easier to manage when you’ve got a lot of peripherals.



STM32Cube MCU packages and embedded software packs

● Copy all used libraries into the project folder

○ Copy only the necessary library files

○ Add necessary library files as reference in the toolchain project configuration file

Generated files

☑ Generate peripheral initialization as a pair of '.c/.h' files per peripheral

☐ Backup previously generated files when re-generating

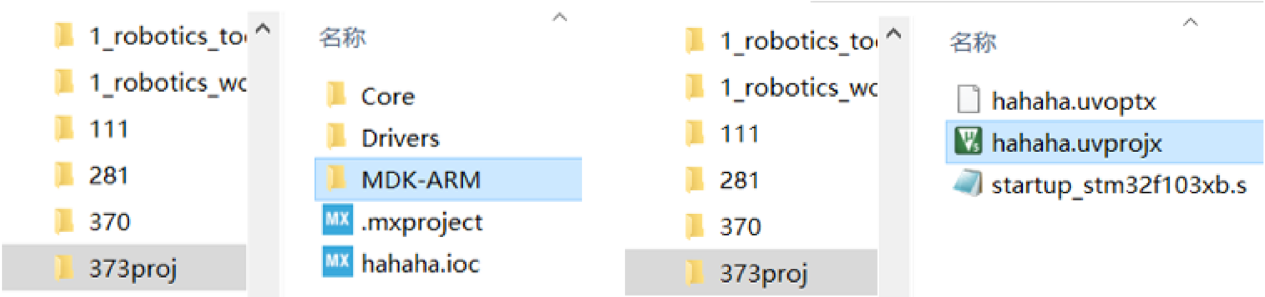
☑ Keep User Code when re-generating

☑ Delete previously generated files when not re-generated

Now all settings are done. To generate your project, click “GENERATE CODE” if you’re using CubeMX or just press Ctrl + S to save the project if you’re using CubeIDE. Note that if you’re doing this for the first time, you’ll be asked to login and download some firmware. If the pop-up window breaks down, just try again.

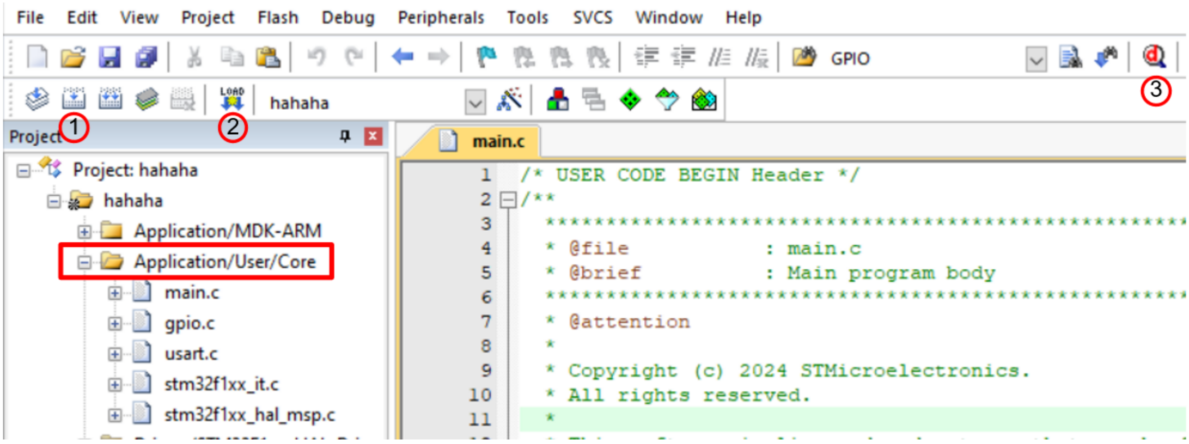
2. Compile and download your project.

Here we show the step-to-step process for Keil, and a similar approach holds for CubeIDE. After generated, your project files would look like something as below. Click into “MDK-ARM” and open the file with a “.uvprojx”.



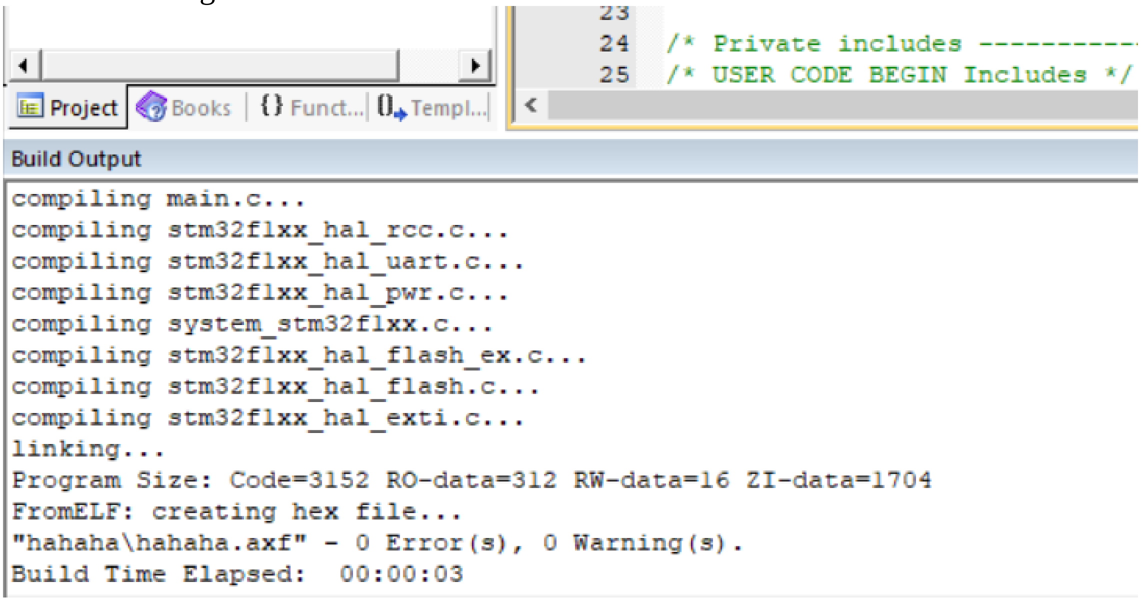
Now we’re at Keil. The folder “Application/User/Core” is your main workplace. Button (1) “Build” compiles your project, button (2) “Load” downloads your compiled project into a STM32 board.

Button (3) “Debug” helps you debug your code by showing values of variables operated in your board. It’s a powerful tool and a major merit of Keil (while CubeIDE can do similar things), and we’ll explore it in future labs.



```
/* USER CODE BEGIN Header */
/**
  ******************************************************
  * @file           : main.c
  * @brief          : Main program body
  ******************************************************
  * @attention
  *
  * Copyright (c) 2024 STMicroelectronics.
  * All rights reserved.
  *
```

Press “Compile” (or shortcut F7) and check the log messages in “Build Output”. The message should be something like below.

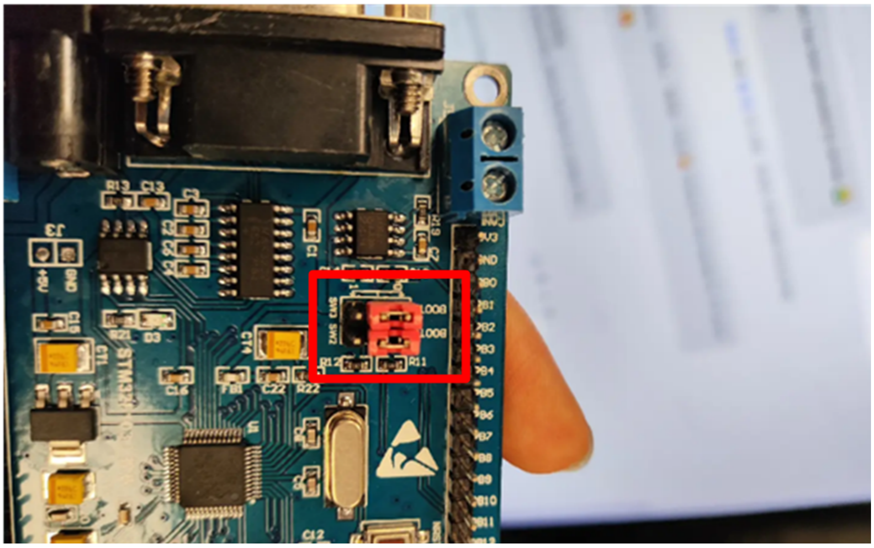


```
/* Private includes ----------
/* USER CODE BEGIN Includes */
```

```
compiling main.c...
compiling stm32f1xx_hal_rcc.c...
compiling stm32f1xx_hal_uart.c...
compiling stm32f1xx_hal_pwr.c...
compiling system_stm32f1xx.c...
compiling stm32f1xx_hal_flash_ex.c...
compiling stm32f1xx_hal_flash.c...
compiling stm32f1xx_hal_exti.c...
linking...
Program Size: Code=3152 RO-data=312 RW-data=16 ZI-data=1704
FromELF: creating hex file...
"hahaha\hahaha.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:03
```

Now the compilation is done. If you’re testing the correctness of installation, congrats for getting everything settled. Later steps of downloading require a physical board and downloader to be distributed in on-site lab.

After you get the board, adjust the jumpers (“跳线帽” in Chinese) as below.



You’ll get a ST-LINK downloader and several wires. Connect SWDIO, SWCLK, 3.3V, GND on the downloader to PA13, PA14, 3.3V, GND on your board. Plug the downloader into your labtop and click “Load” button in Keil. A successful download means you’ve finished everything (except the report) of Lab1.

c. Other Options for Developing

As you would see in future development, Keil / CubeIDE are good tools for compiling, downloading and debugging but not for writing codes. Functions like code navigation and auto-completion are hard to use and the interface is crude. For other choices, you can refer to “*Lab1_Additional_Contents*”, a guidance written by a former TA with ample experience in embedded programming. As long as you could make up a toolchain that covers configuration, compilation, download and debug together with coding, you can work well with that.