ECE 373 Microprocessor Based System Design

Lab 1 Development Tools for STM32 MCU Developing

OBJECTIVE

- Install keil V5, STM32CubeMX for STM32F103C8T6 Coding and debugging ;Install ST-LINK V2 for programming the MCU
- Learn to use STM32CubeMX to configure a C project with specific STM32 MCU
- Learn to use Keil V5 to build ,compile the C project, download the hex file to the MCU with the ST-LINK.
- Learn to use CubeIDE for developing STM32 projects
- Understand the whole loop of developing a STM32 C language project

PROJECT DESCRIPTION

In this lab, you will need to setup the embedded system IDE for the rest of this course. The lab involves:

1. Install Keil V5 (software) and Crack it successfully on your personal computer;
2. Install ST-LINK V2 and update the drive software ;
3. Install STM32CubeMX , create a New project and select the MCU , configure it correctly ;
4. Open the STM32CubeMX create project with Keil V5 to build and download the project to object MCU with ST-LINK.
5. Download and install CubeIDE, use CubeIDE as the develop software instead of STM32CubeMX Keil V5.

Install Softwares

As mentioned in the Project Description , Keil , ST-LINK, STM32CubeMX would be installed in your computer .

- Install Keil V5 as programming and debugging tool for STM32 arm MCU. The Keil uVision files are in the folder below:

https://jbox.sjtu.edu.cn/l/w1jxtK

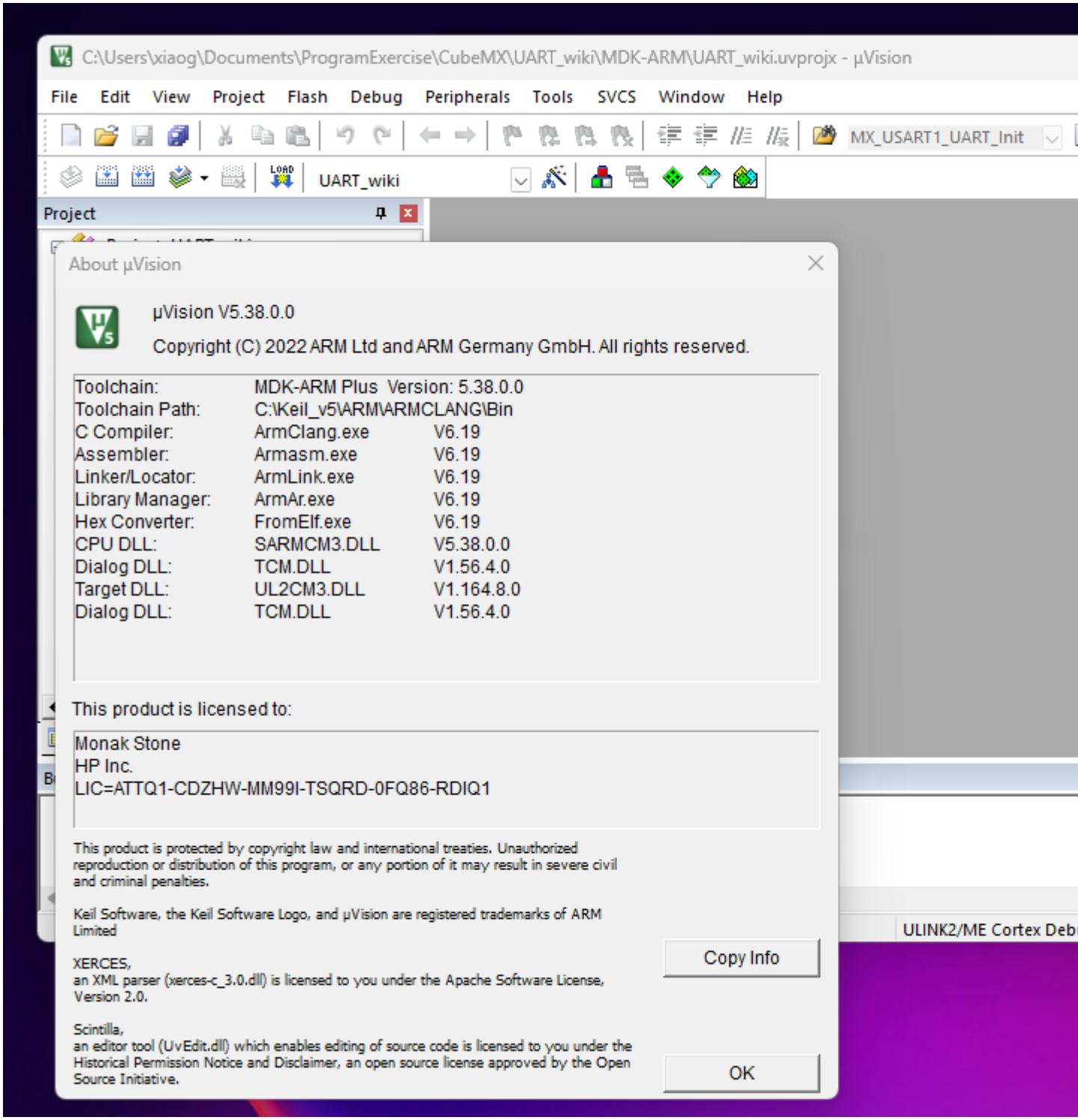
There are four files under this folder.When install the Keil uVision V5 , you would install the software sequentially ::

1. Unpacked and install Keil uVision5 MDK V5.20.rar:
2. Install Keil uVision5 MDK V5.2;
3. Run the crack file of keil;

Download and install drive for ST-LINK V2 . The software is in the folder below:

https://jbox.sjtu.edu.cn/l/c16Jfm

Open Keil V5 and check the “Help” menu, click the “About uVision “ , if they have been installed and cracked correctly , following information would show:



After the Keil Assistant extension is installed, you should set Keil Assistant extension to Keil as shown below :

- Download and install STM32CubeMX and CubeIDE

Open the link : https://www.st.com/content/st_com/zh/stm32cubemx.html,

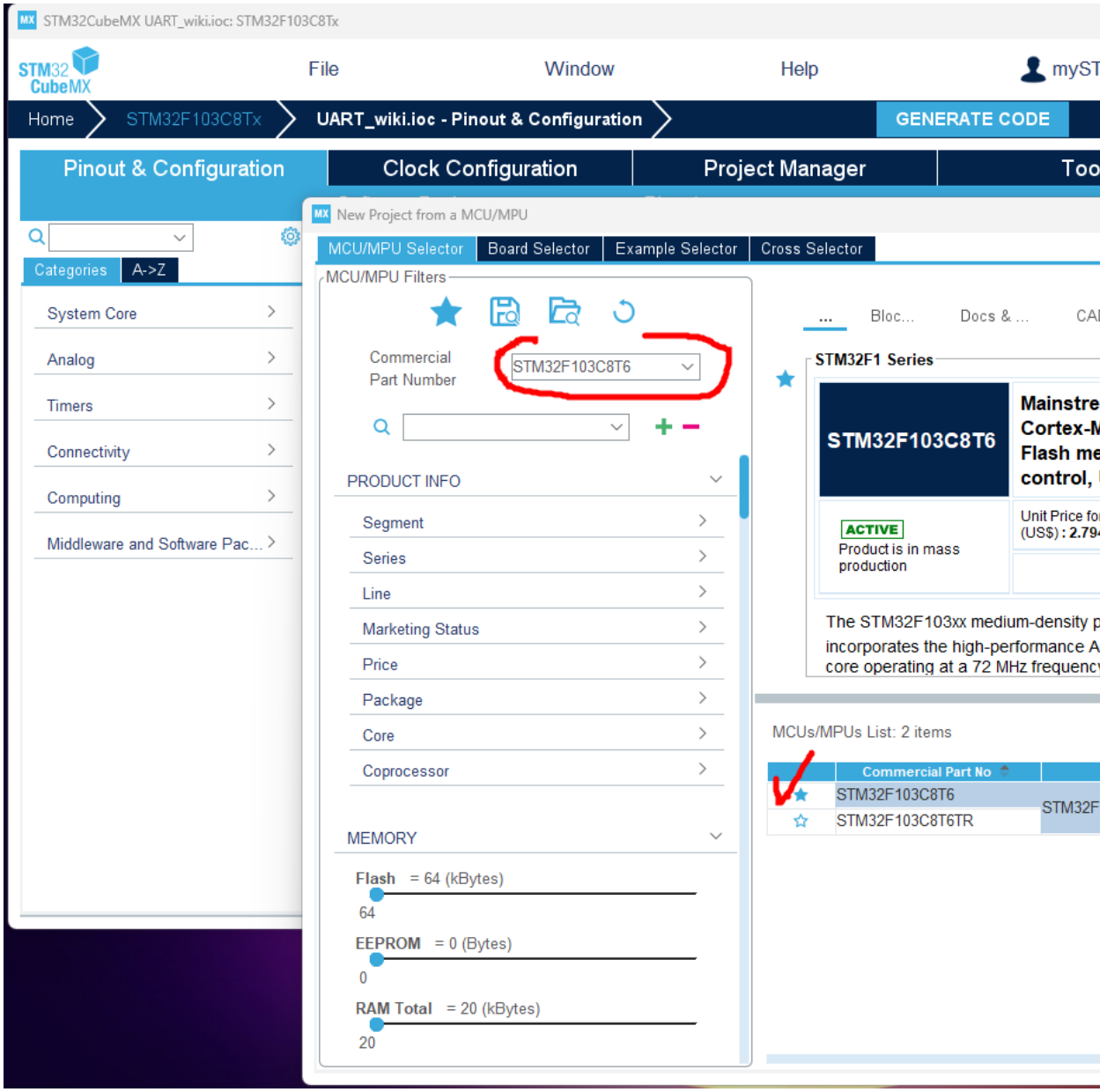
Download the STM32CubeMX and STM32CubeIDE and install them on your computer.

As described previously , normally there are two options to develop a STM32 project :

Option 1, use STM32CubeMX to configure the MCU in your project , and use Keil to build, load and debug the project files to
the object board .

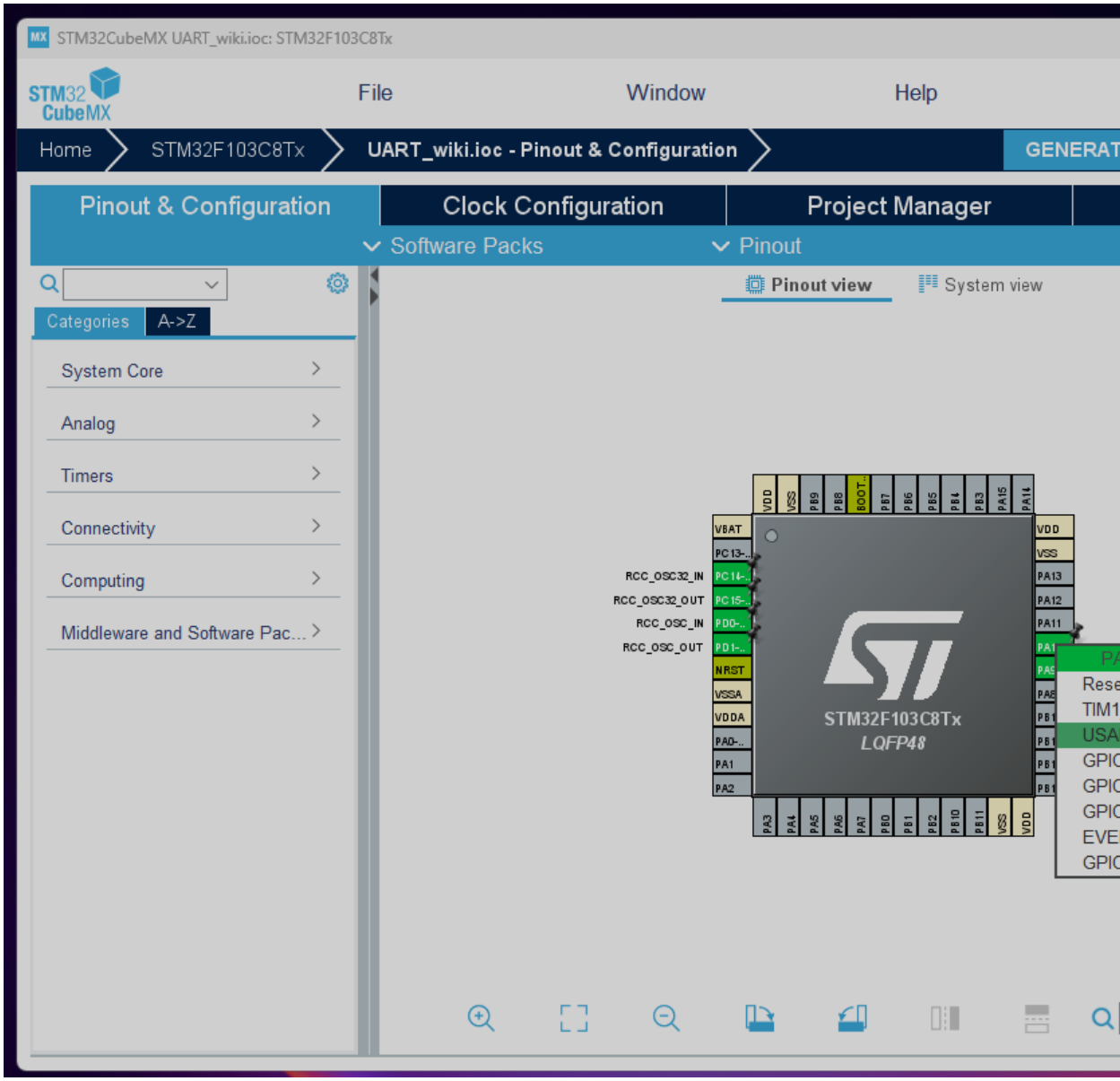
Below pictures is an example of STM32F103C8Tx for UART communication project.:

i. MCU Selector
Select STM32F103C8T6 for example project:

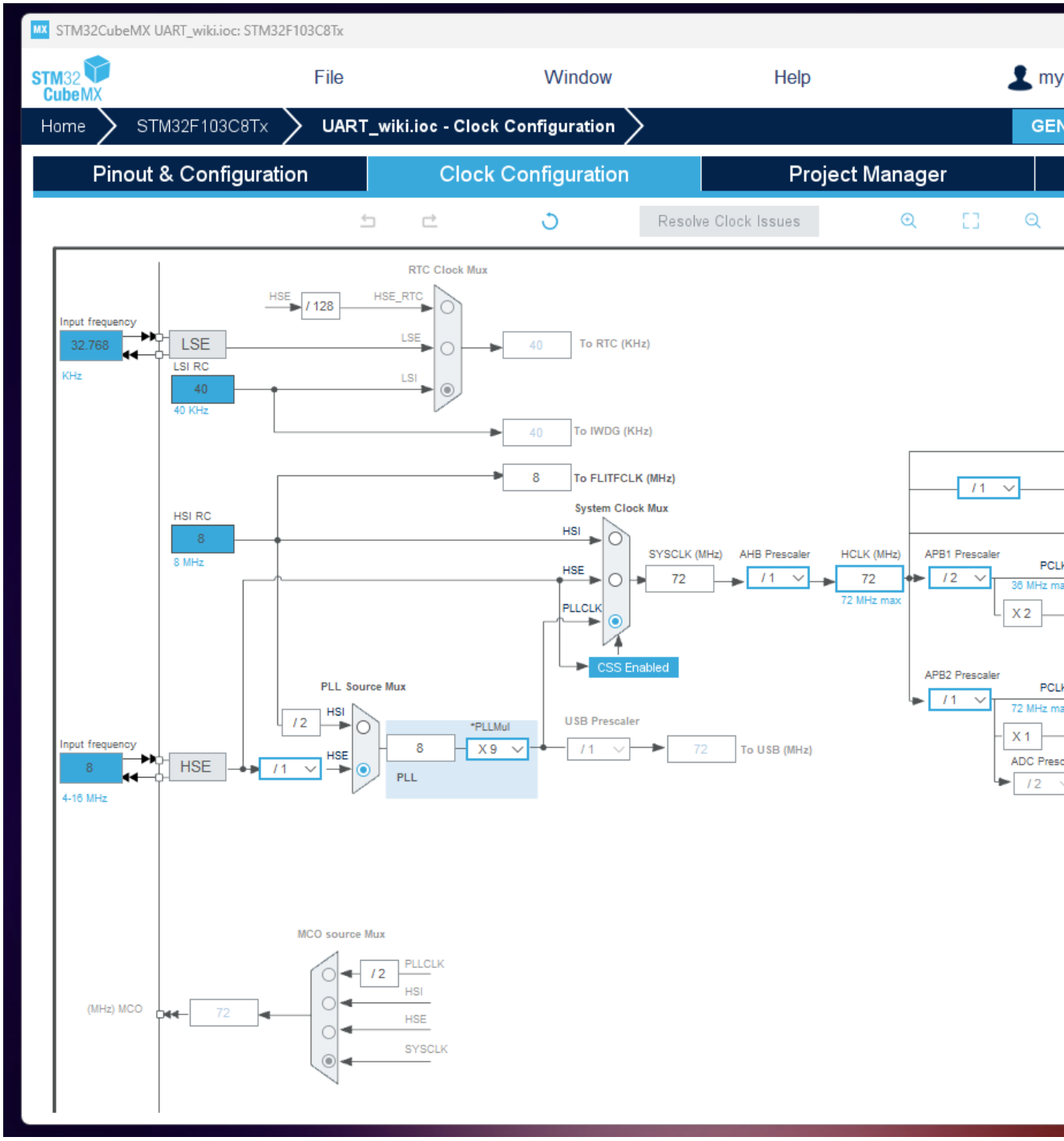


ii. Pinout & Configuration

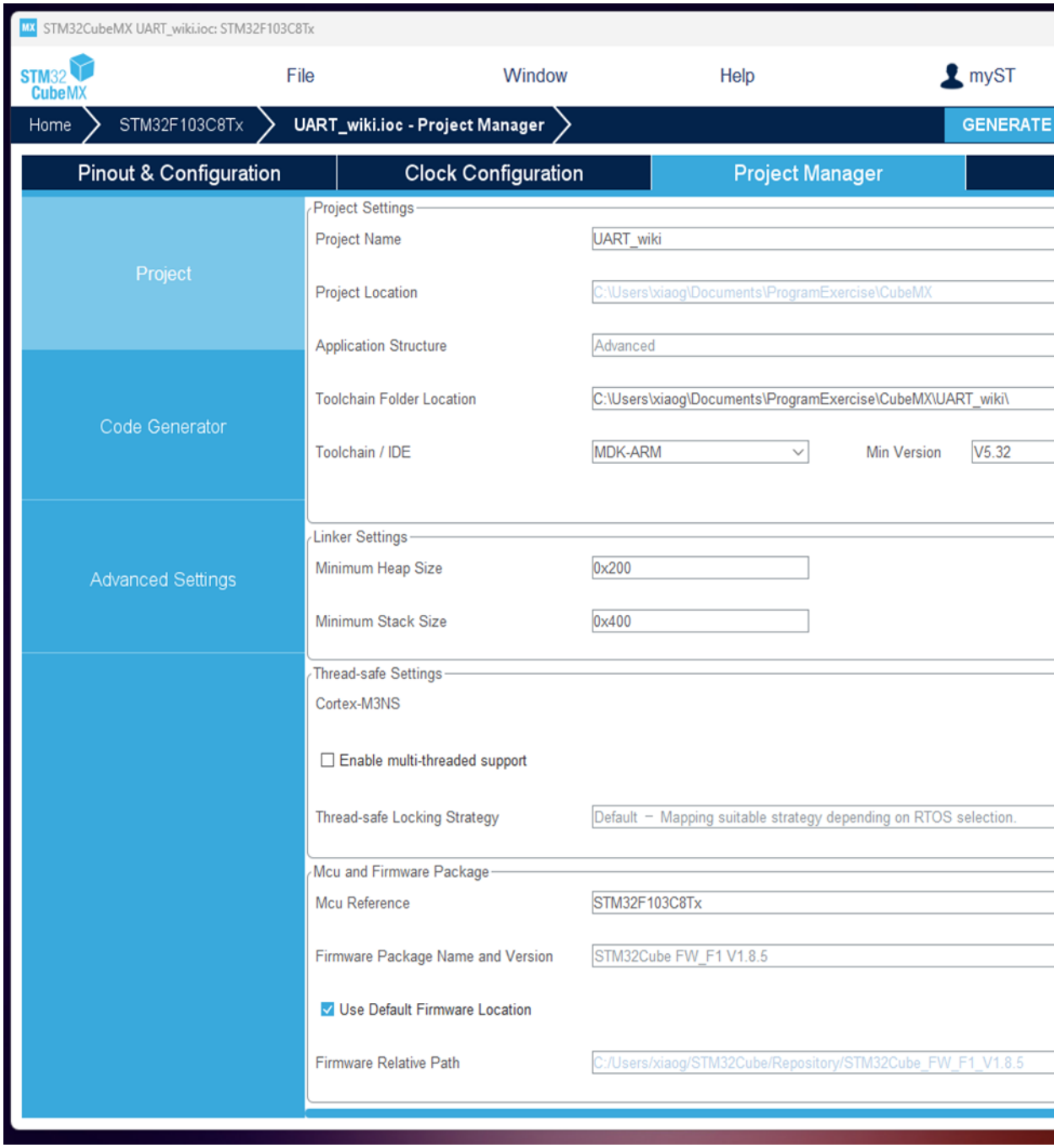
Configure PA10 as UART_RX and PA9 as UART_TX .



iii. Clock Configuration:

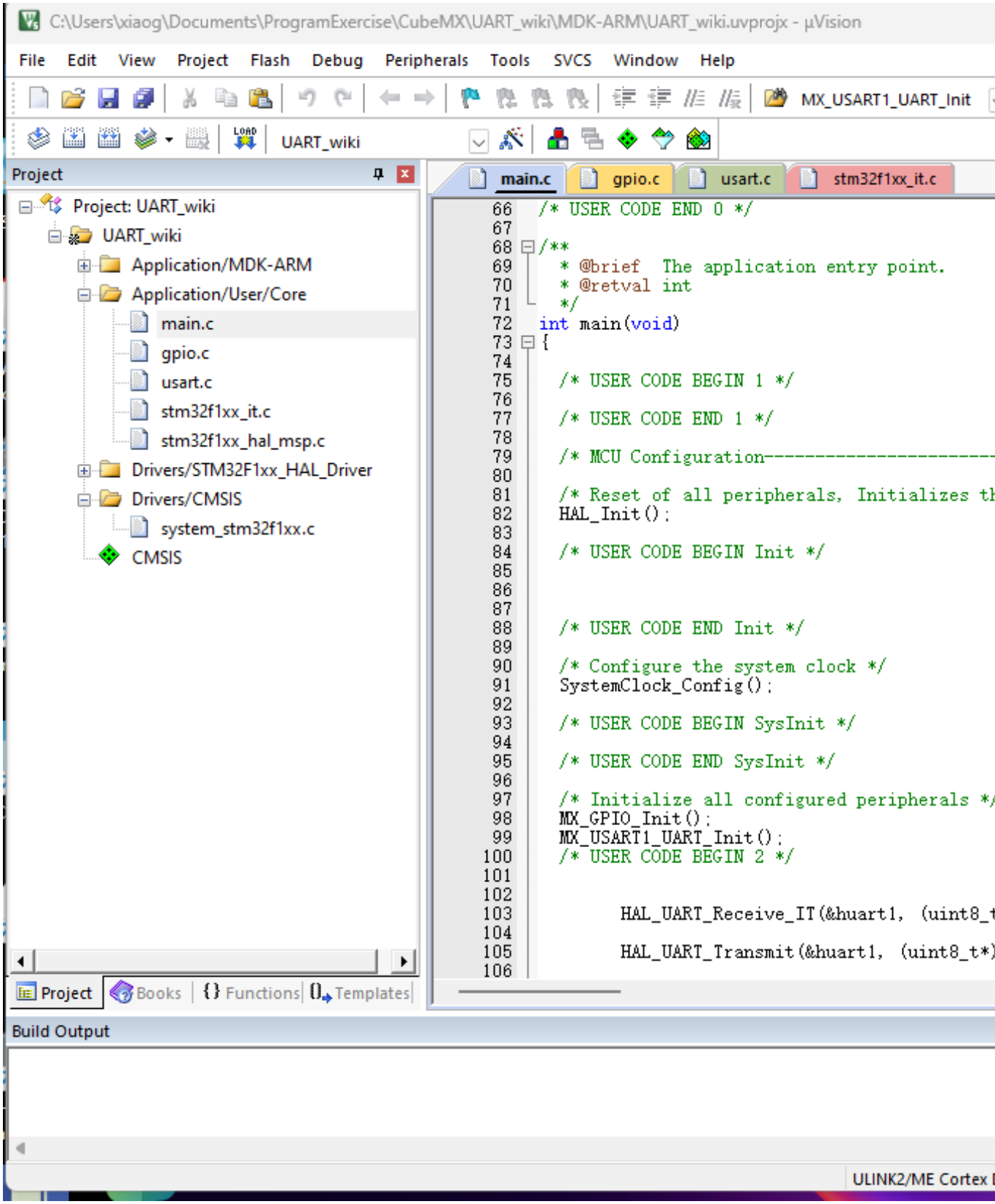


iv. Project Manger set

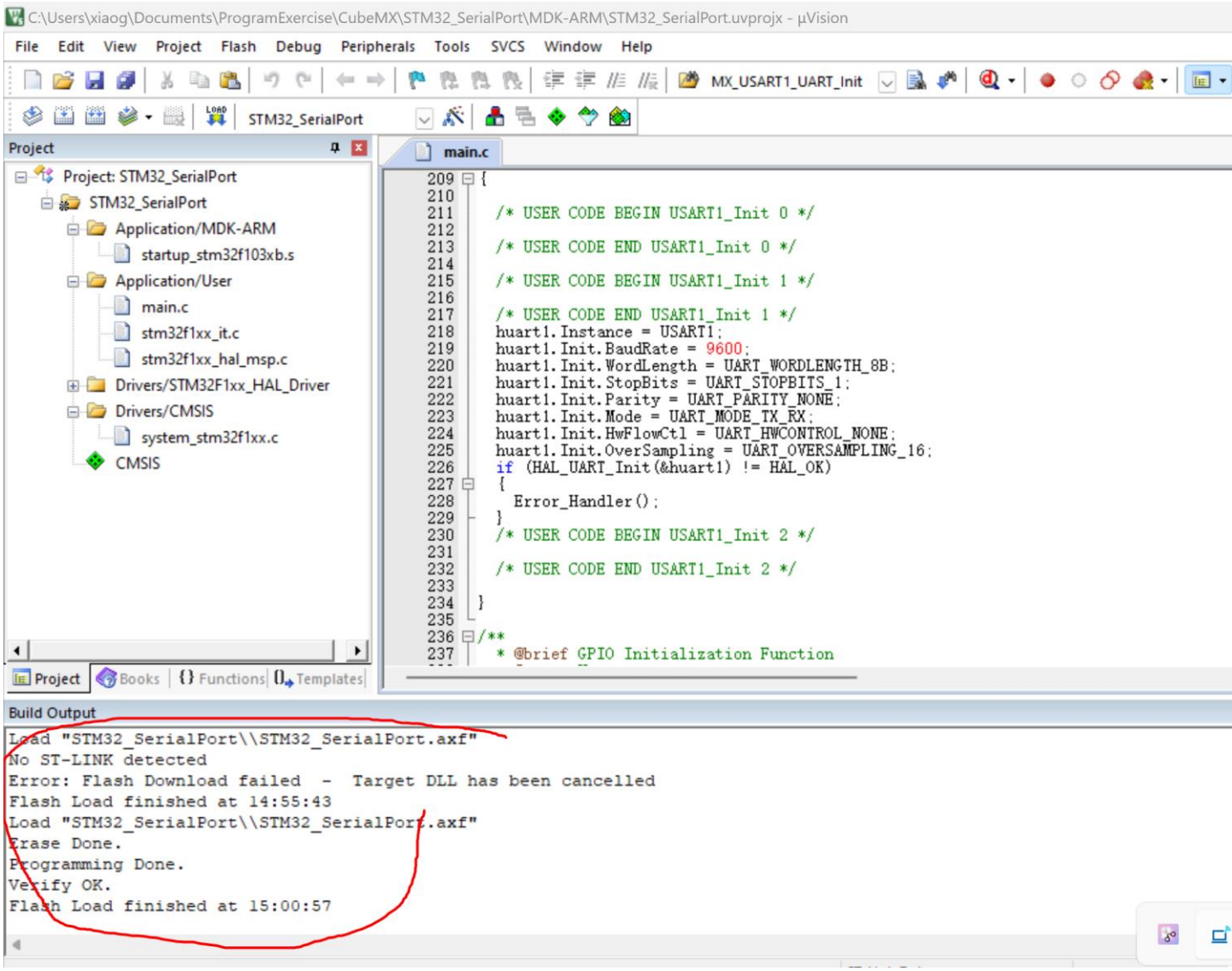


After the MCU configuration, Save the project. For the sample , we created and saved the project named UART_wiki .
Now we can open it with Keil V5 software , now you can edit and build the project files synchronizely .
After the configurations , the project configurations are created and can be saved.

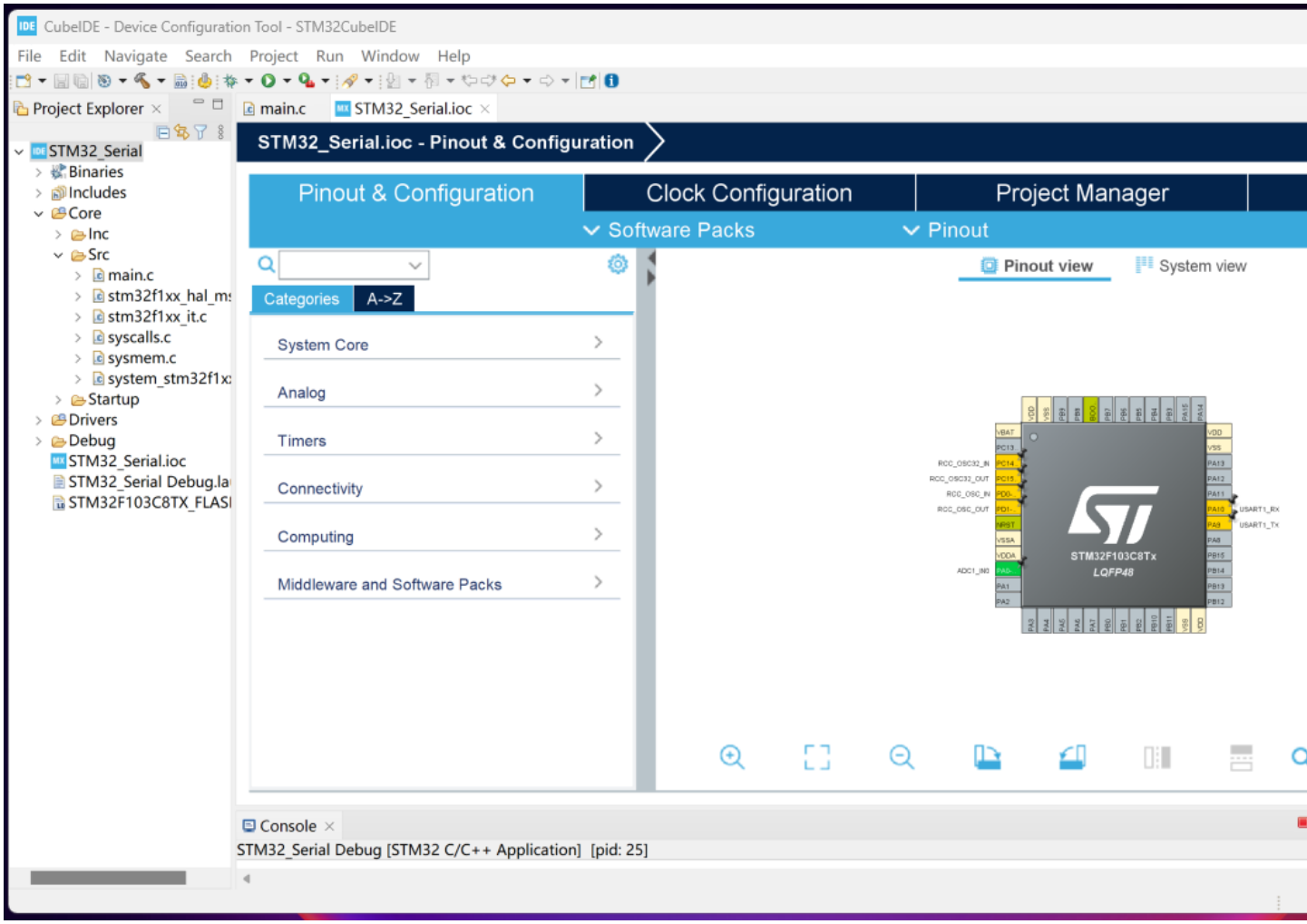
v. Open the sample project with Keil V5 .



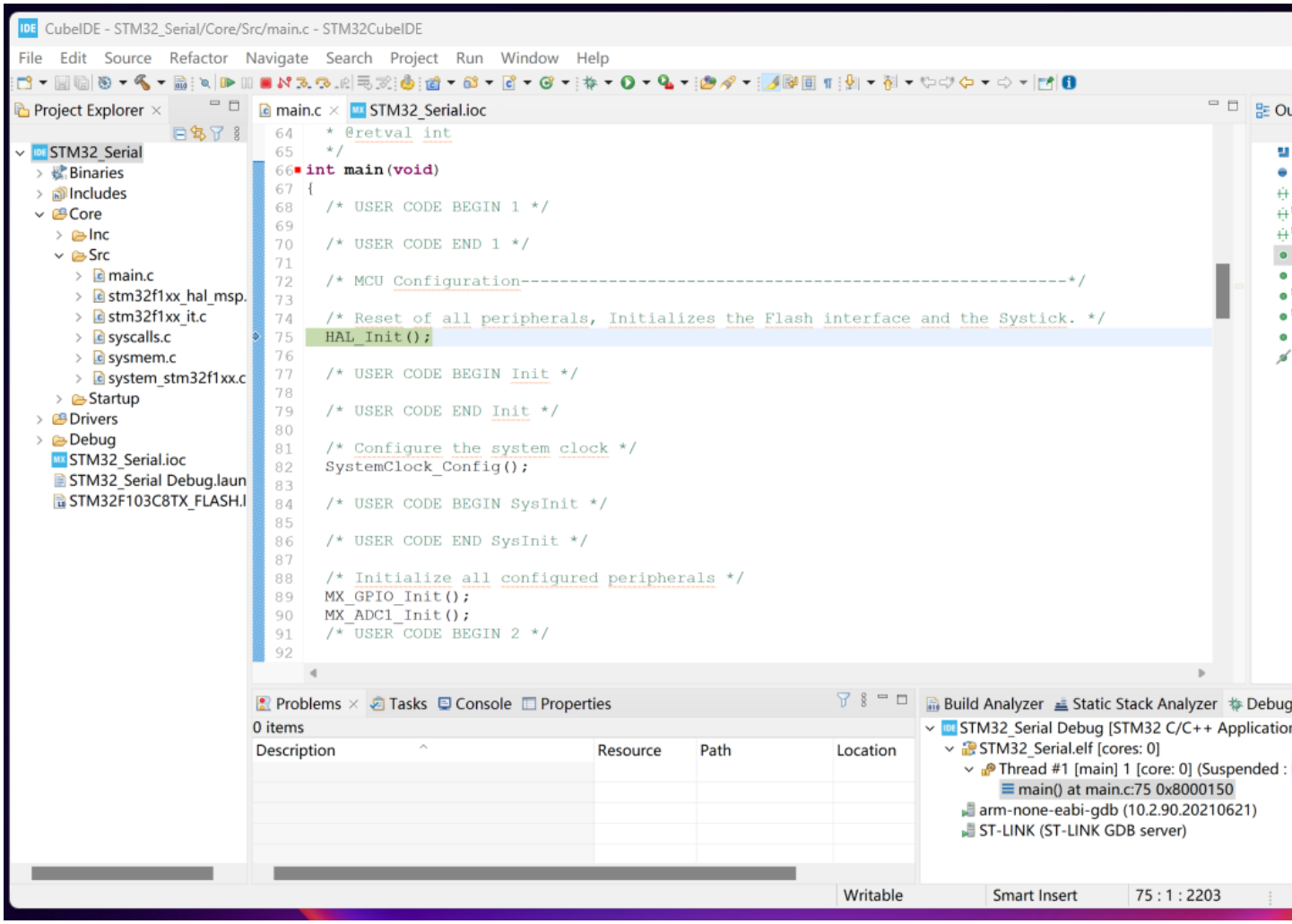
Use Keil to edit, build program files and load the compiled project files to the object MCU board, the result shows as below.



Option 2 : Use CubeIDE as the project developing tool to configure , edit , compile , build and download the project .
CubeIDE combine the functions of configurations editing , building and debugging , the following pictures show CubeIDE sample.



After configured the MCU , one can run the configurations. A project related to the Configuration be created . Save the project in the folder preferred and now there are several files of C in Src folder. The files can be edited ,built, compiled and download to object MCU board.


CubeIDE - STM32_Serial/Core/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
STM32_Serial
Binaries
Includes
Core
Inc
Src
main.c
stm32f1xx_hal_msp.
stm32f1xx_it.c
syscalls.c
sysmem.c
system_stm32f1xx.c
Startup
Drivers
Debug
STM32_Serial.ioc
STM32_Serial Debug.laun
STM32F103C8TX_FLASH.l
main.c
STM32_Serial.ioc
64 * @retval int
65 */
66 int main(void)
67 {
68 /* USER CODE BEGIN 1 */
69
70 /* USER CODE END 1 */
71
72 /* MCU Configuration--*/
73
74 /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
75 HAL_Init();
76
77 /* USER CODE BEGIN Init */
78
79 /* USER CODE END Init */
80
81 /* Configure the system clock */
82 SystemClock_Config();
83
84 /* USER CODE BEGIN SysInit */
85
86 /* USER CODE END SysInit */
87
88 /* Initialize all configured peripherals */
89 MX_GPIO_Init();
90 MX_ADC1_Init();
91 /* USER CODE BEGIN 2 */
92
Problems Tasks Console Properties
0 items
Description Resource Path Location
Build Analyzer Static Stack Analyzer Debug
STM32_Serial Debug [STM32 C/C++ Application
STM32_Serial.elf [cores: 0]
Thread #1 [main] 1 [core: 0] (Suspended :
main() at main.c:75 0x8000150
arm-none-eabi-gdb (10.2.90.20210621)
ST-LINK (ST-LINK GDB server)
Writable
Smart Insert
75 : 1 : 2203