

EE3220 System-on-Chip Design

Tutorial 2: Write a program and debug with Keil Studio

Objective:

- Get familiar with the NUCLEO board and ARM Keil Studio platform.
- Learn how to write a program to control the blinking of LEDs.
- Learn how to use debug tools in ARM Keil Studio.

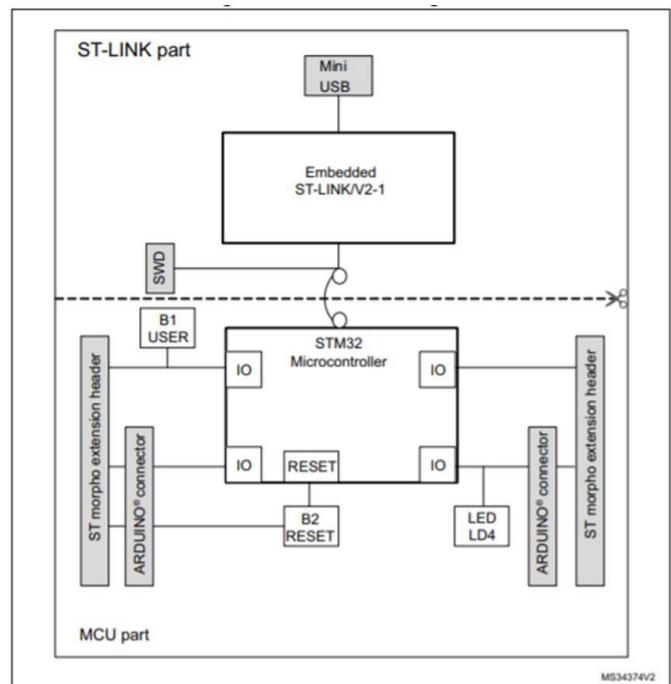
In this tutorial, we aim to get familiar with the targeted device (NUCLEO-F401RE ARM board), write a program, and debug on the ARM Keil Studio platform. During this process, you will create a project and write a program to control the blinking of LED lights. You can change LED blink frequency by changing the C code and running debug on Keil Studio. At the end of this tutorial, you will gain more experience playing the ARM board, writing and debugging your programs, and being ready to design your own system.

In this tutorial, students don't need the board. If you want to use the board, you can borrow it from P1800.

Overview of NUCLEO F401-RE

Common features

- STM32 microcontroller in LQFP64 package
- 1 user LED shared with ARDUINO®
- 1 user and 1 reset push-buttons
- 32.768 kHz crystal oscillator
- Board connectors:
 - ARDUINO® Uno V3 expansion connector
 - ST morpho extension pin headers for full access to all STM32 I/Os
- Flexible power-supply options:
 - ST-LINK, USB VBUS, or external sources
- On-board ST-LINK debugger/programmer with USB re-enumeration capability: mass storage, Virtual COM port and debug port
- Comprehensive free software libraries and examples available with the STM32Cube MCU Package
- Support of a wide choice of Integrated Development Environments (IDEs) including IAR Embedded Workbench®, MDK-ARM, and STM32CubeIDE Board-specific features
- External SMPS to generate Vcore logic supply
- 24 MHz HSE – Board connectors:
 - External SMPS experimentation dedicated connector
 - Micro-AB or Mini-AB USB connector for the ST-LINK



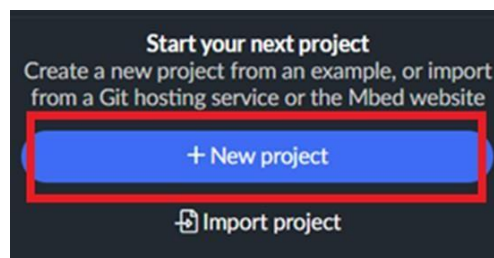
- MIPI® debug connector
- Arm® Mbed Enabled™ compliant

Development tools: the board is supported by many tools and IDEs many vendors

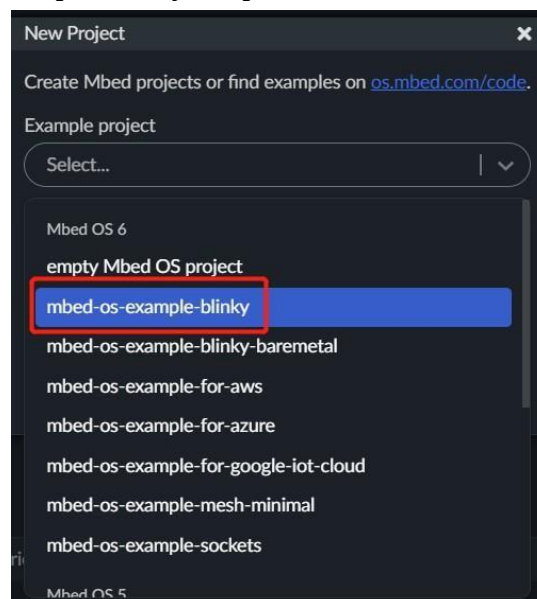
- ARM Development Studio (licensed)
- ARM® Mbed online
- ARM Keil Studio Cloud
- STM32CubeIDE (free) IAR Systems
- IAR Embedded Workbench® (licensed)
- Keil®: MDK-ARM(a) (licensed)
- GCC-based IDEs
- CrossWorks
- SEGGER Embedded Studio (commercial with free-trial)
- SW4STM32 (free)
- TrueSTUDIO (free)
- MATLAB® and Simulink® (licensed)

Create a blinky project and change the parameter

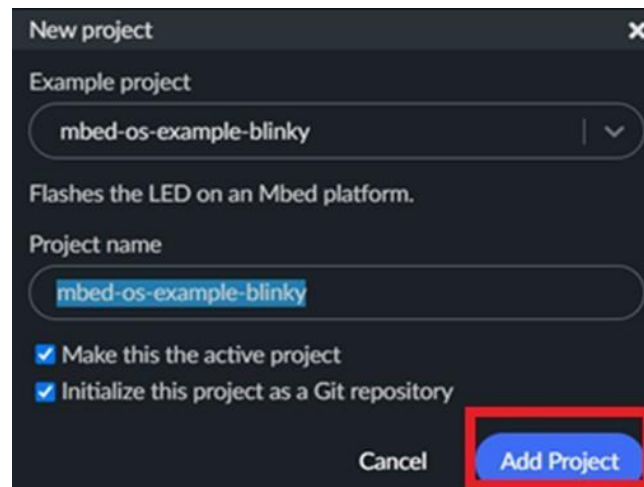
- Go to the Keil Studio Cloud website and log in
<https://studio.keil.arm.com/auth/login/>
- Click New project



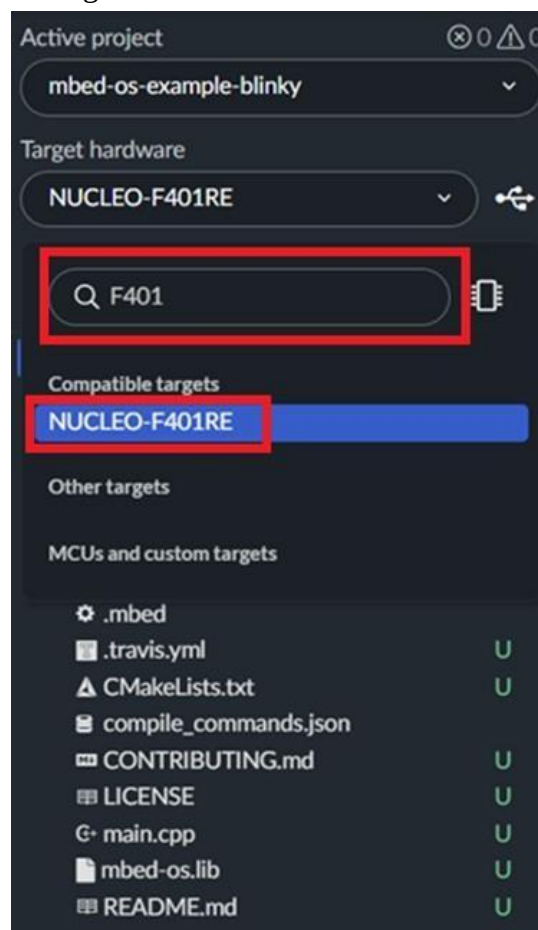
- Select mbed-os-example-blinky template



- Add Project

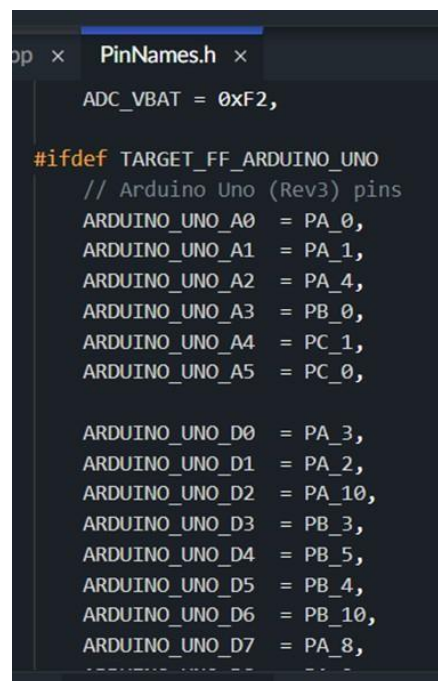
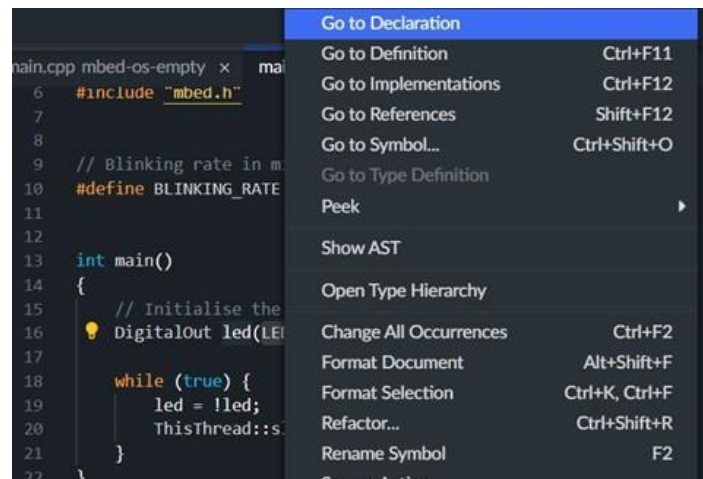


- Search and select the target hardware: **F401**



- Install the ST-link USB driver from the web page, prior to connecting the board.
<https://www.st.com/zh/development-tools/stsw-link009.html#get-software>
 (After downloading, go to device manager -> Ports, select driver)

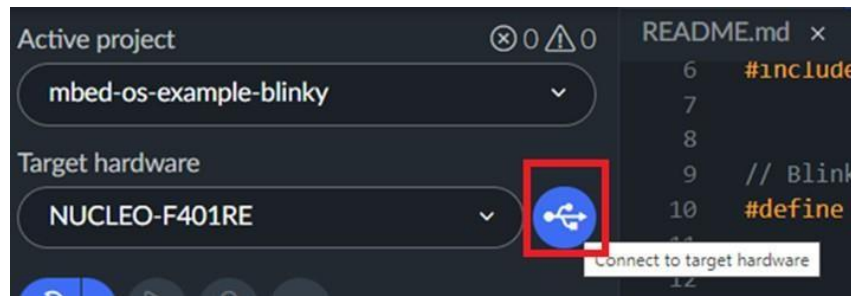
- Open PinNames.h, read more about pin definition on NUCLEO_F401RE board
(Open **main.cpp** file, select and right click **LED1**, Click **Go to declaration**)



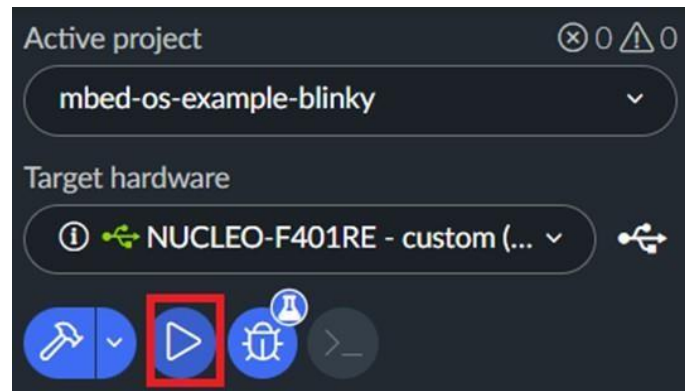
- Click Build Project, it starts to compile and generate mbed-os-example-blinky.NUCLEO_F401RE.bin file



- Connect your board with USB port and then click connect to target hardware on website. Choose STM32Link and connect.



- Click run project and the generated executable file is downloaded to your board.



Debug system overview

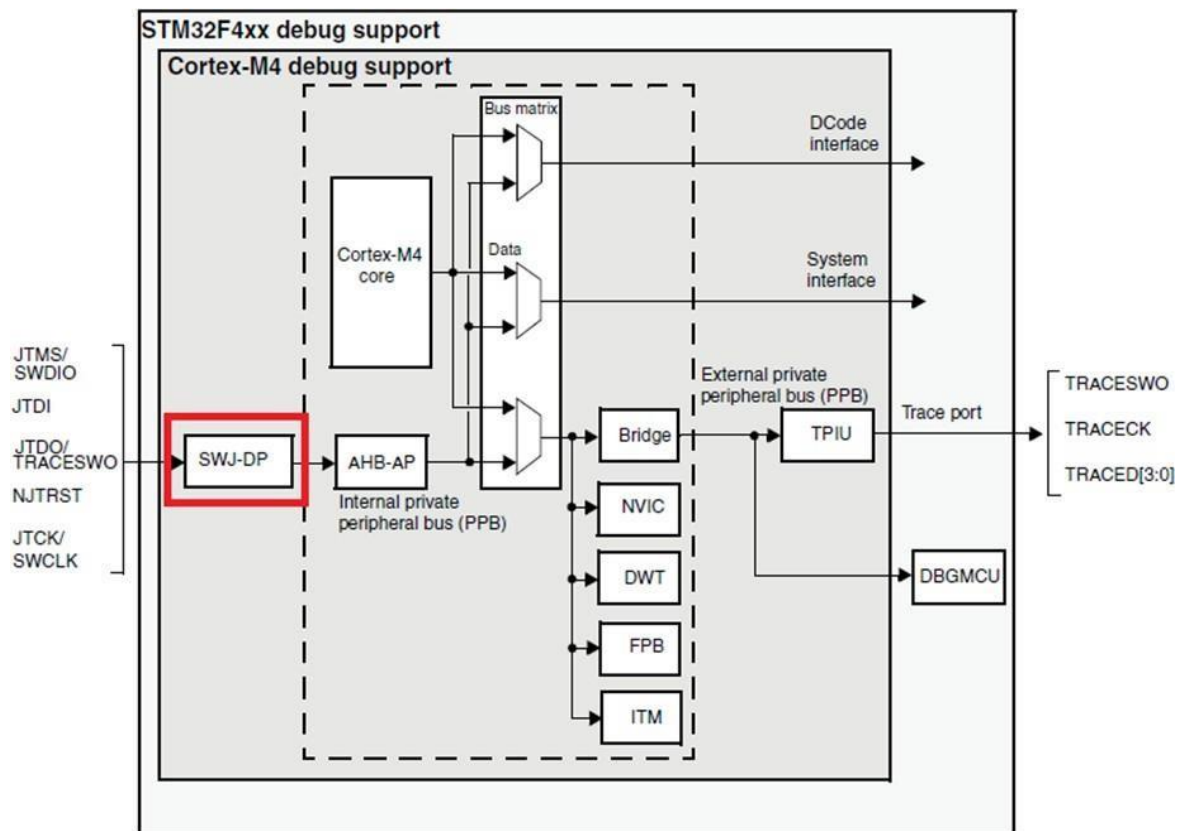
The STM32F401RE is built around a Cortex®-M4 with FPU core which contains hardware extensions for advanced debugging features:

- The debug extensions allow the core to be stopped either on a given instruction fetch (breakpoint) or data access (watchpoint).
- When stopped, the core's internal state and the system's external state may be examined.
- Once examination is complete, the core and the system may be restored and program execution resumed.

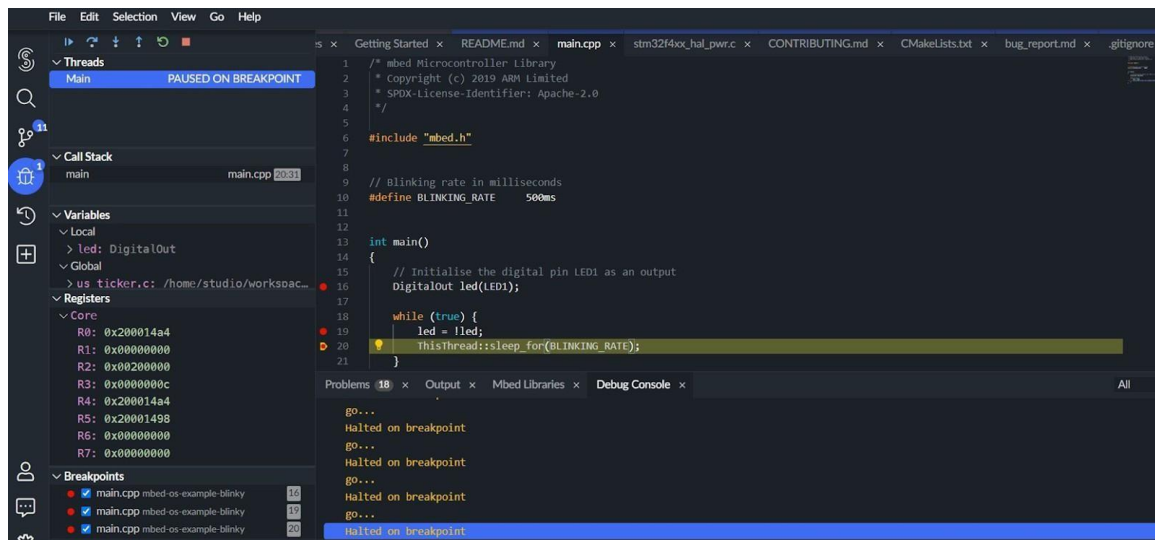
Two interfaces for debug are available:

- Serial wire
- JTAG debug port

In STM32F401RE, debugging is performed using the Serial Wire Debug Port (SW-DP).

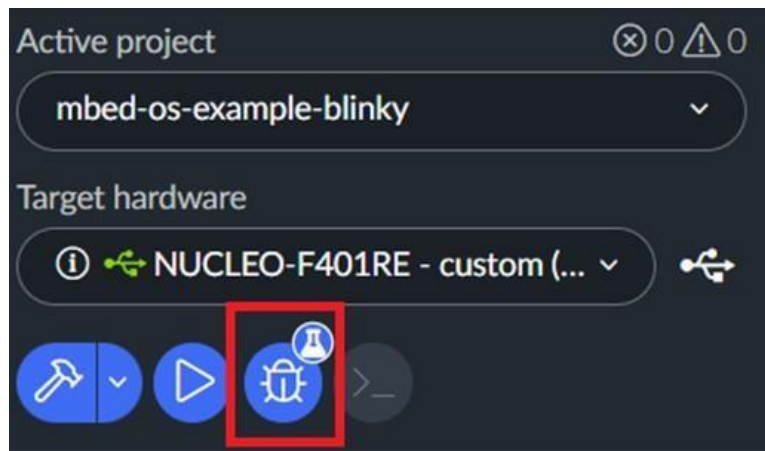


Debug window in ARM Keil Studio

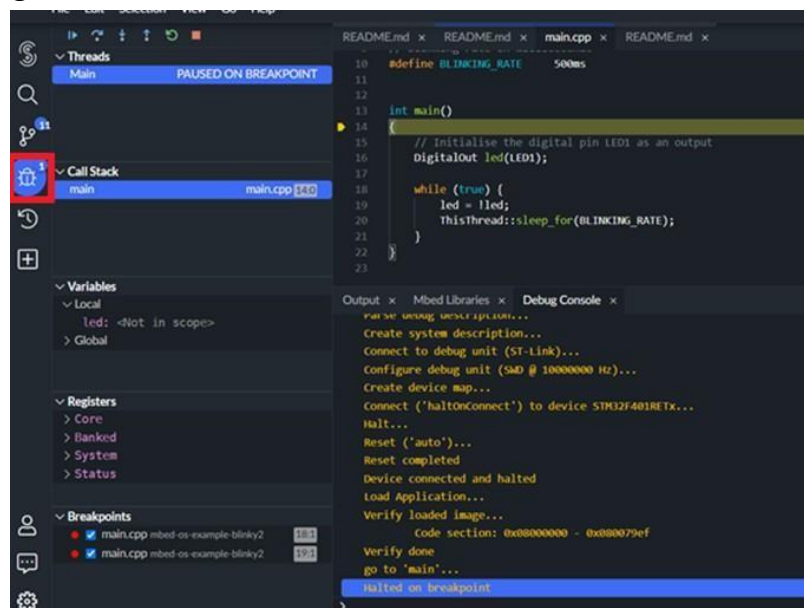


Keil Studio Debug process

- Click Debug project



- Enter the Debug window when the buildup project is over. Confirm the generation of the debug interface.



- Find out the Debug menu.



Continue button: To start debugging, click the Continue button. The debugger runs to the first breakpoint it encounters and stops.



Step Over: Advance the debugger to the next source line that follows in the program execution to go straight to the parts of code you are more interested in.



Step Into: Advance the debugger into each function. The debugger then breaks on the first line that gets executed in the function.



Step Out: Advance the debugger until the current function returns (in other words, advance all the way through the current function).

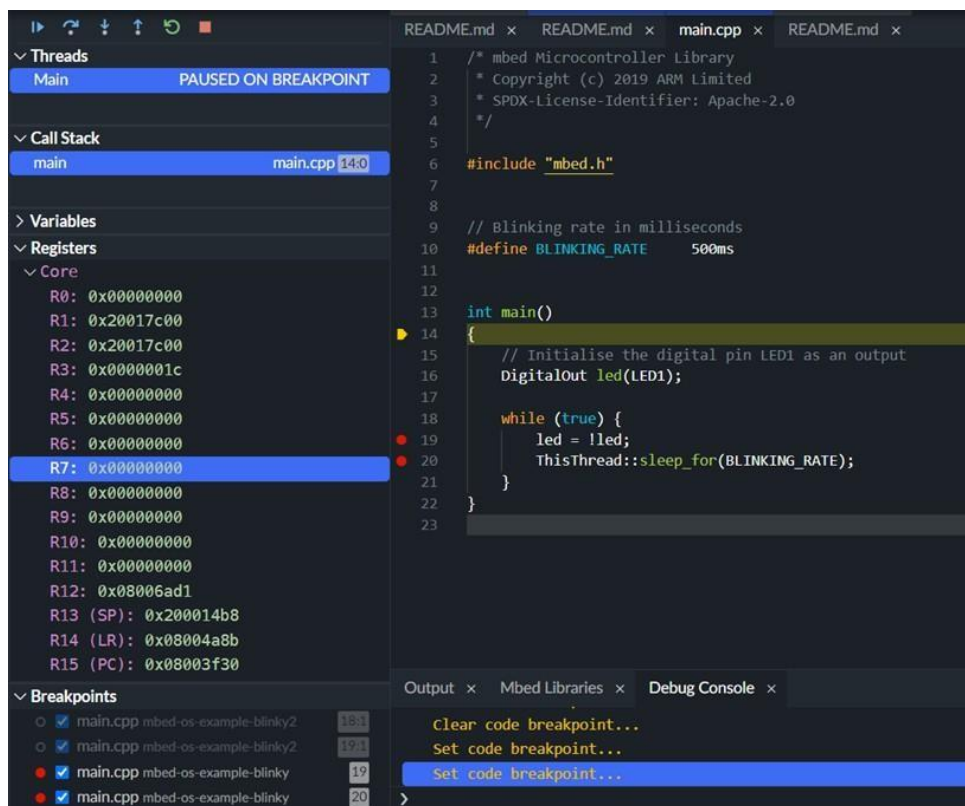


Restart button



Stop button: To stop the debugger and return to the editor.

- Select the breakpoint you want to debug, then click the continue button.
- Observe the changes of the small LED lights during debugging
- Look at the value changes in Registers.



- Observe carefully: How many registers change the value? How it is changed?
- Change the BLINKING_RATE parameter, is there any value changed in the register during debug?

Reference documents and websites

- Go to board website read more reference document

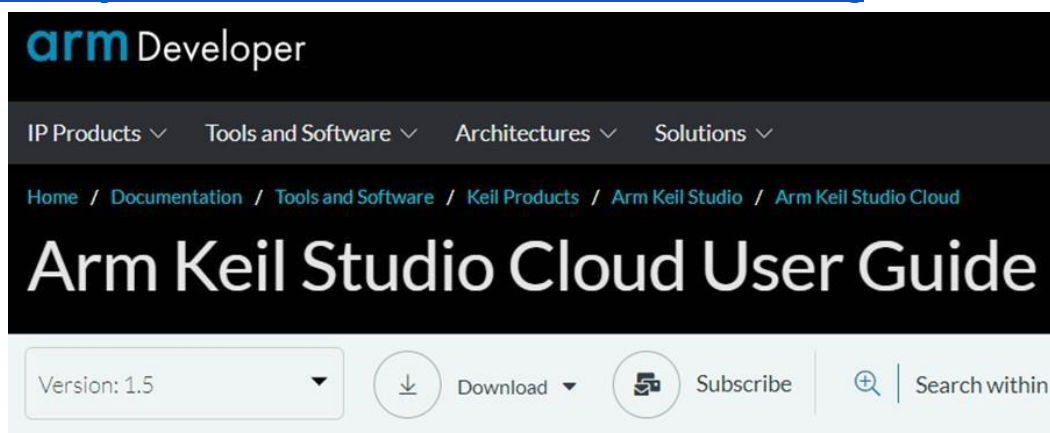
<https://www.st.com/en/microcontrollers-microprocessors/stm32f401re.html#documentation>

- Reference document: RM0368 Reference manual

https://www.st.com/resource/en/reference_manual/dm00096844-stm32f401xb-c-and-stm32f401xd-e-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf

- Arm Keil Studio Cloud User Guide

<https://developer.arm.com/documentation/102497/1-5/Monitor-and-debug>



- Download the full user guide pdf file

<https://documentation-service.arm.com/static/616ea229ac265639eac59559?token=>

END