



MakeNTU STM32 工作坊

ST MCU Taiwan Team

2021.03.27

1 軟體安裝

2 開發板

3 STM32CubeIDE 簡介

4 Hands-on: GPIO

5 Hands-on: UART

6 Hands-on: Timer

7 Hands-on: TouchGFX

軟體安裝
建立myST帳號

建立myST帳號

<https://my.st.com/>

The screenshot shows the STMicroelectronics myST account login page. At the top right, there is a 'Create Account' button highlighted with a pink box. Below the login form, there is a section titled 'New myST user?' which lists various benefits such as managing product updates, ordering samples, and accessing newsletters. At the bottom of the page, there is a footer with links to About STMicroelectronics, Media Center, Investor Relations, Sustainability, and Careers sections.

New myST user?

Brings you a set of personalized features:

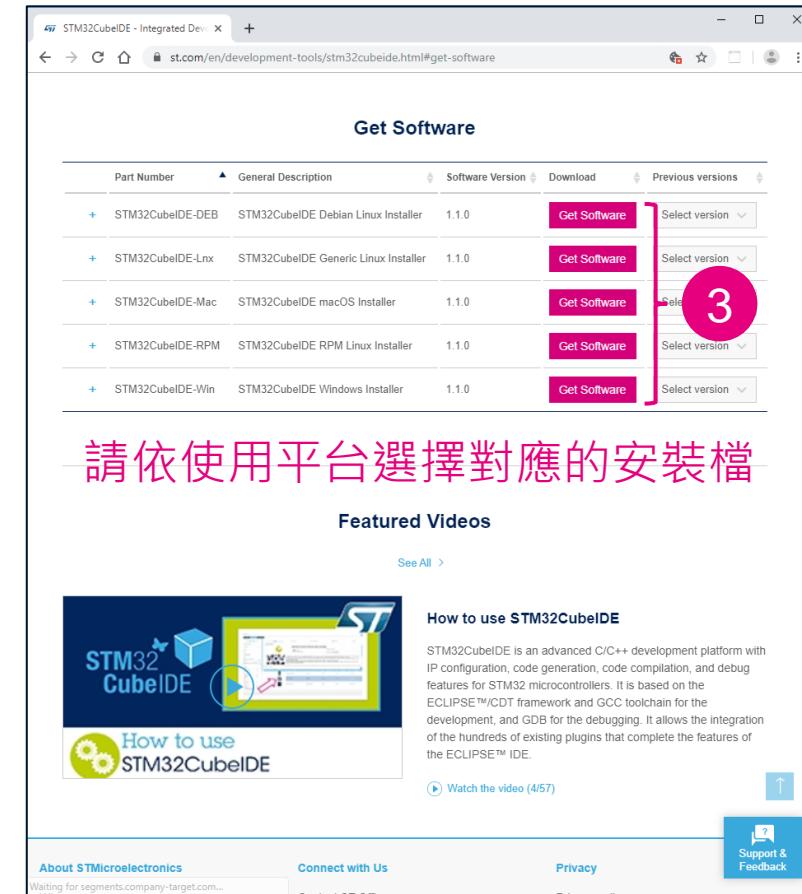
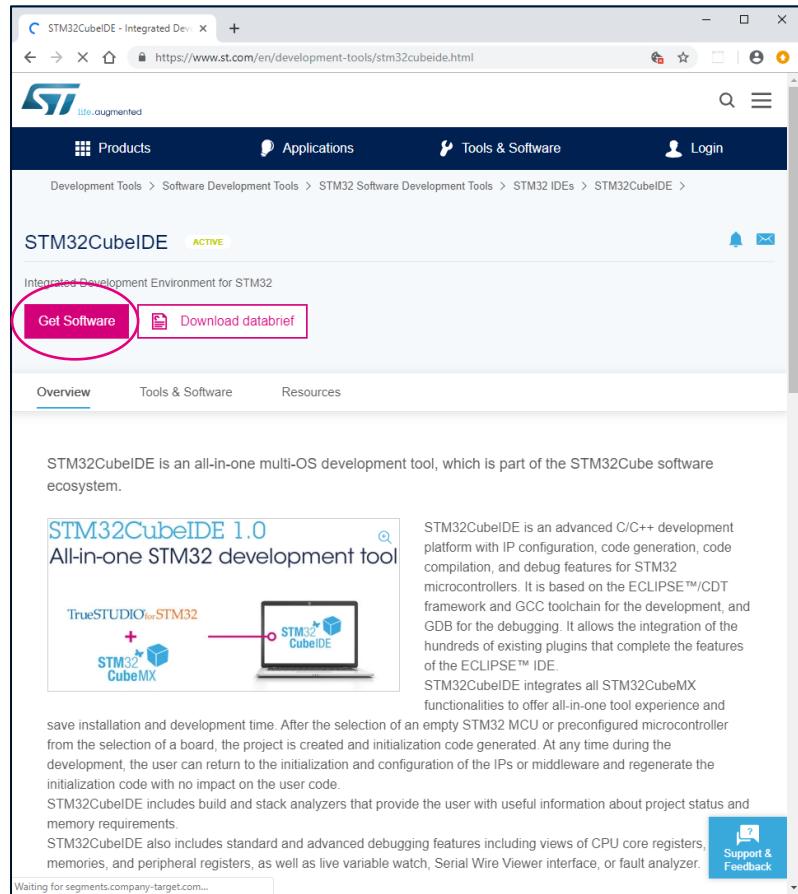
- Manage your weekly product updates
- Order free sample
- Download Software
- Buy products and tools
- Stay informed with ST eNewsletters
- Access ST Online Support
- Join our engineer-to-engineer Communities
- Benefit from our Online Design Tools
- Participate to ST events

軟體安裝

下載並安裝STM32CubeIDE

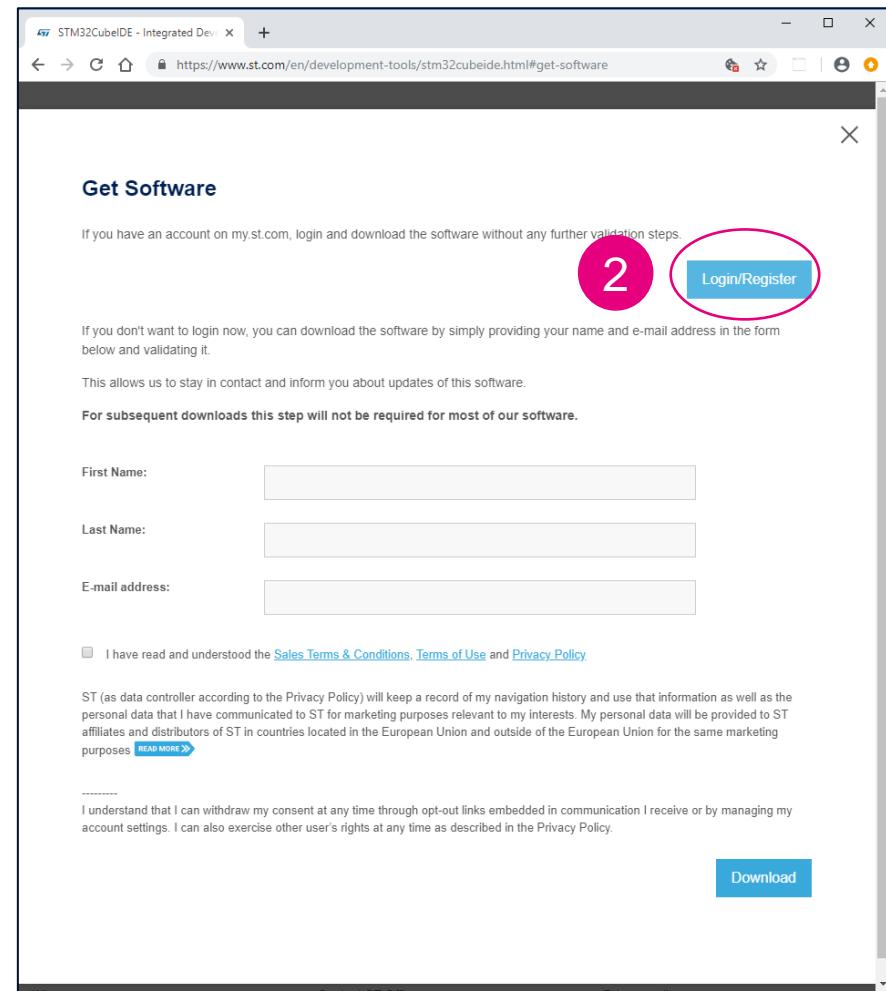
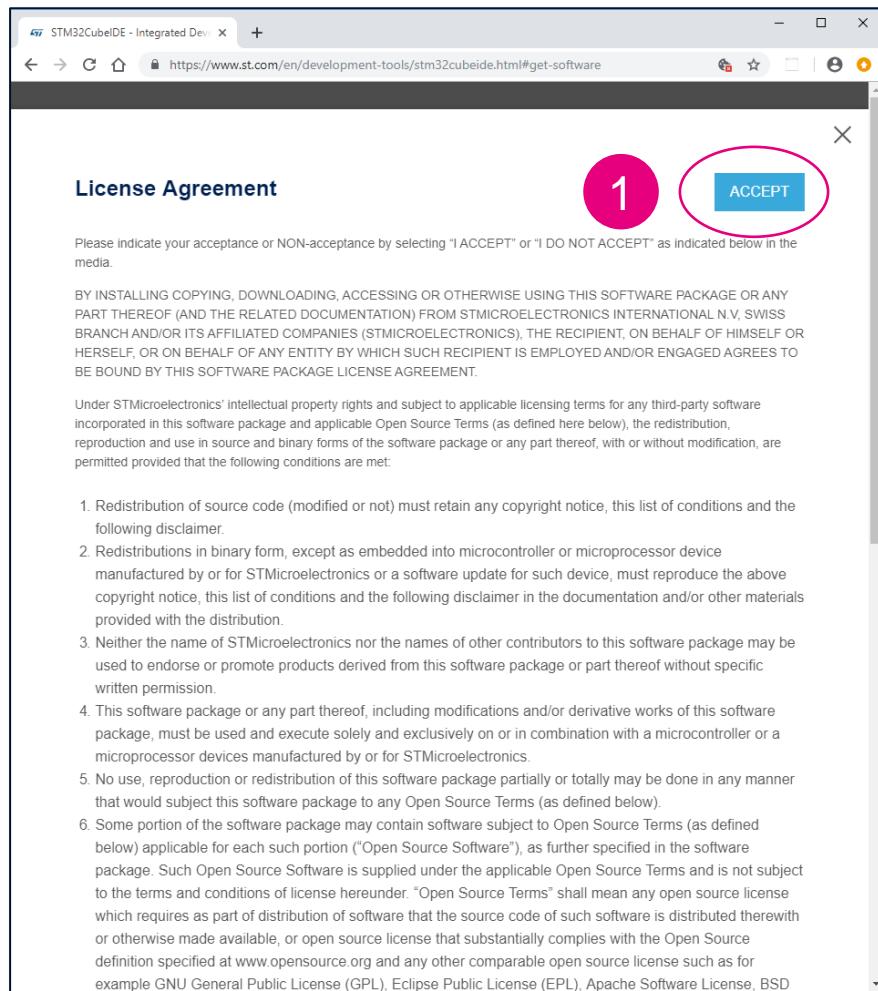
下載STM32CubeIDE 下載頁面

1 <https://www.st.com/en/development-tools/stm32cubeide.html>

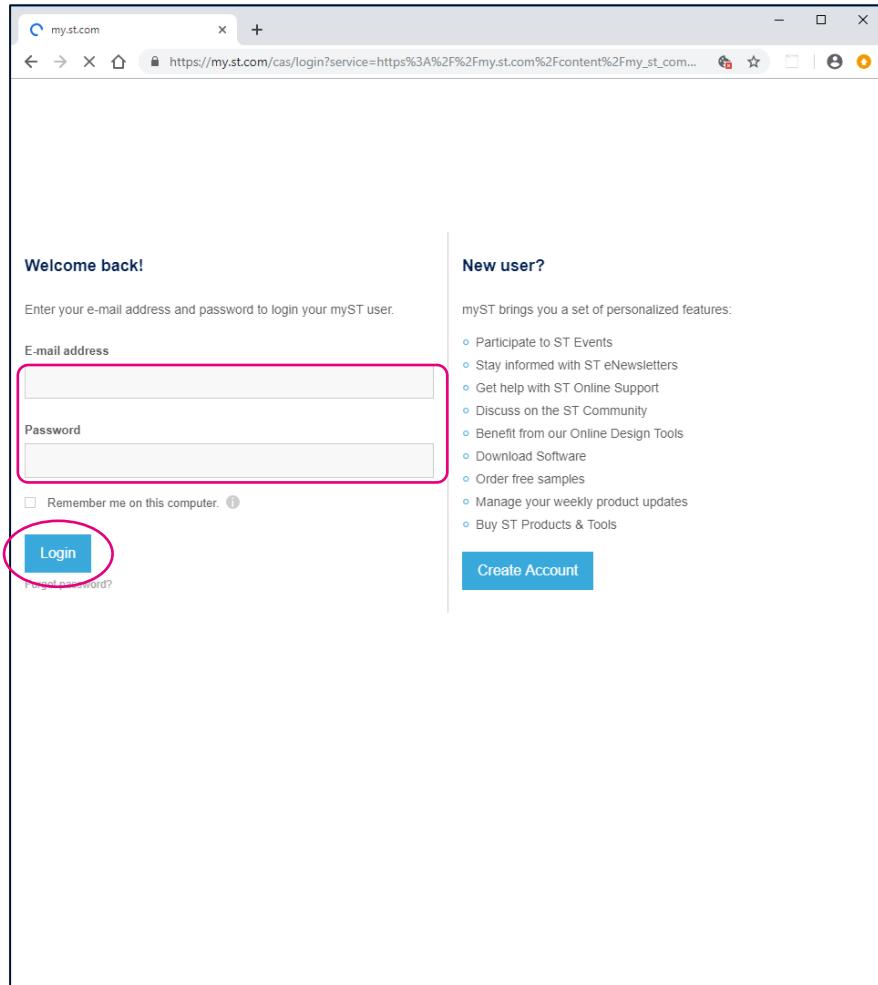


請依使用平台選擇對應的安裝檔

下載STM32CubeIDE License Agreement



下載STM32CubeIDE 登入及下載



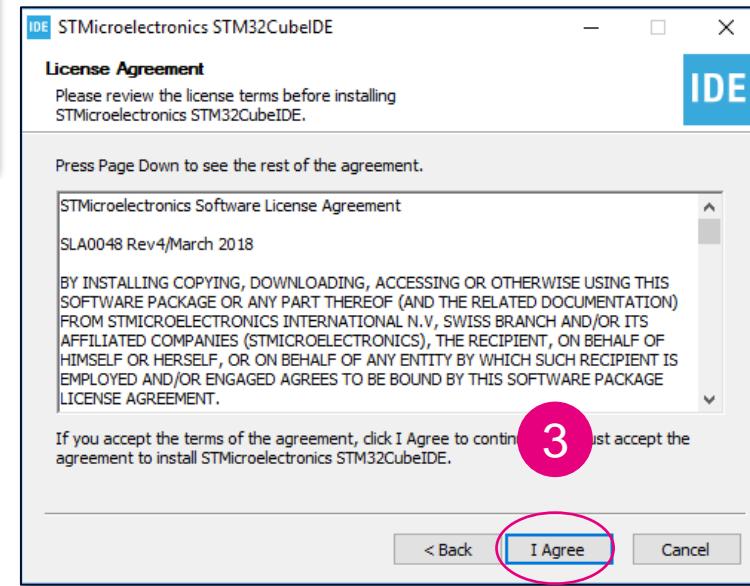
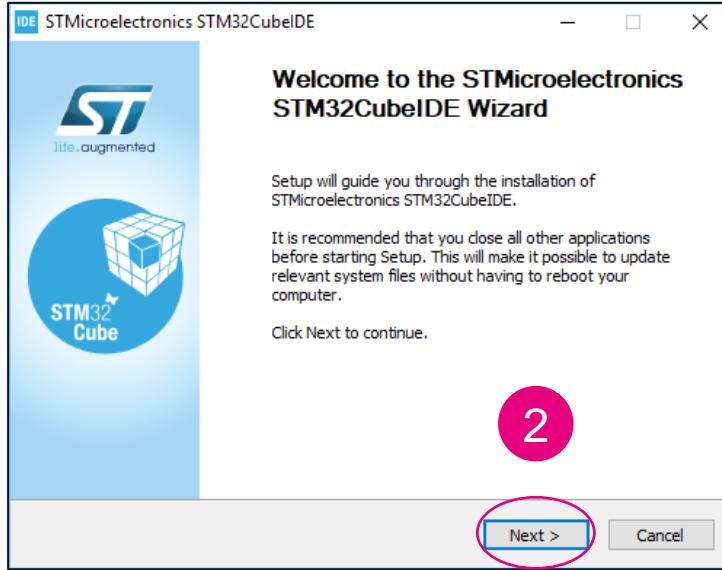
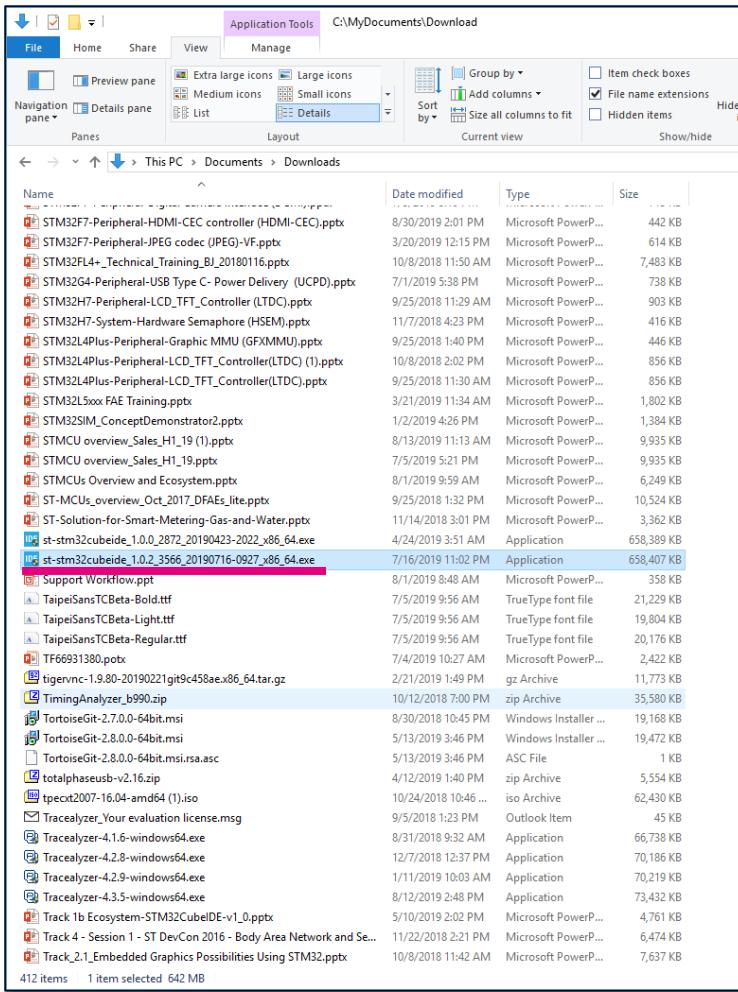
1

2

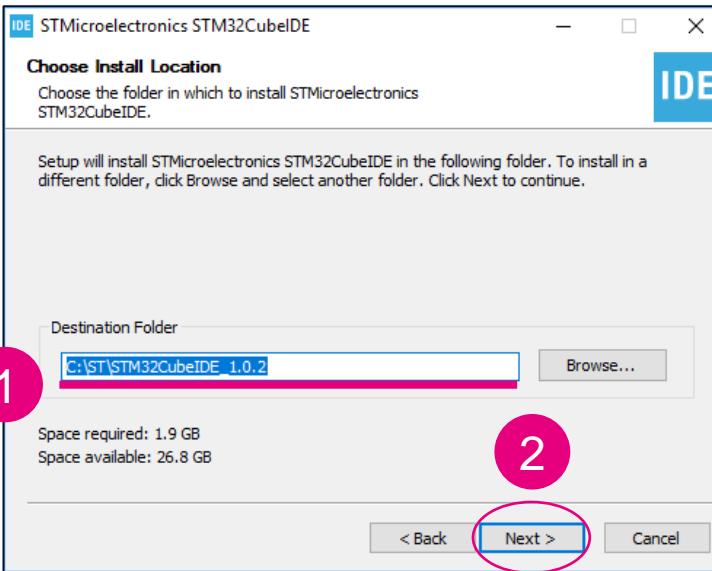
The screenshot shows the STM32CubeIDE product page on the ST website. At the top, there are navigation links for 'Products', 'Applications', 'Tools & Software', and a user profile. Below this, a banner for 'Integrated Development Environment for STM32' features two buttons: 'Get Software' and 'Download databrief'. The 'Get Software' button is highlighted with a pink rectangular box. The main content area is titled 'STM32CubeIDE 1.0' and describes it as an 'All-in-one STM32 development tool'. It highlights integration with TrueSTUDIO for STM32 and STM32CubeMX, and shows a laptop icon with the STM32CubeIDE interface. To the right, a detailed description of the tool's capabilities is provided, mentioning its use of the Eclipse CDT framework and GCC toolchain for development, and GDB for debugging. A pink oval points to the 'Support & Feedback' button at the bottom right of the page. A download progress bar at the bottom indicates a file named 'en.st-stm32cubeede...' is being downloaded at 5.1/638 MB, with 6 mins left.

下載會在完成登入之後開始

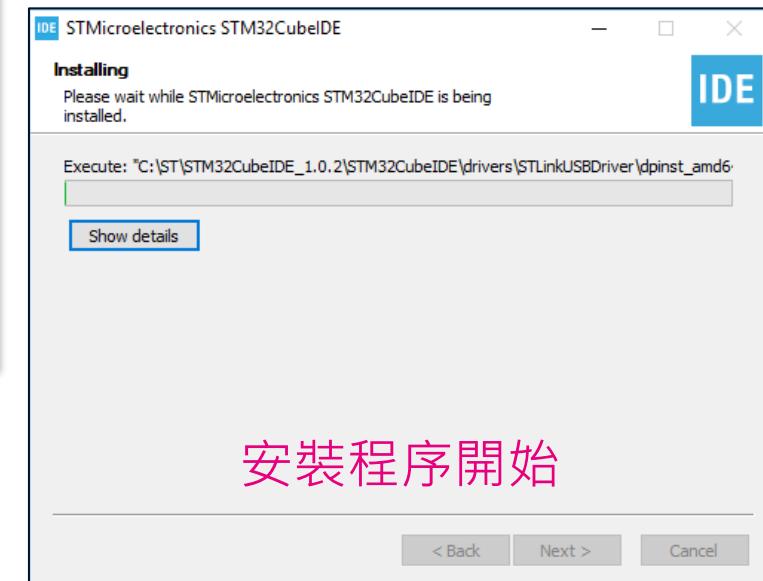
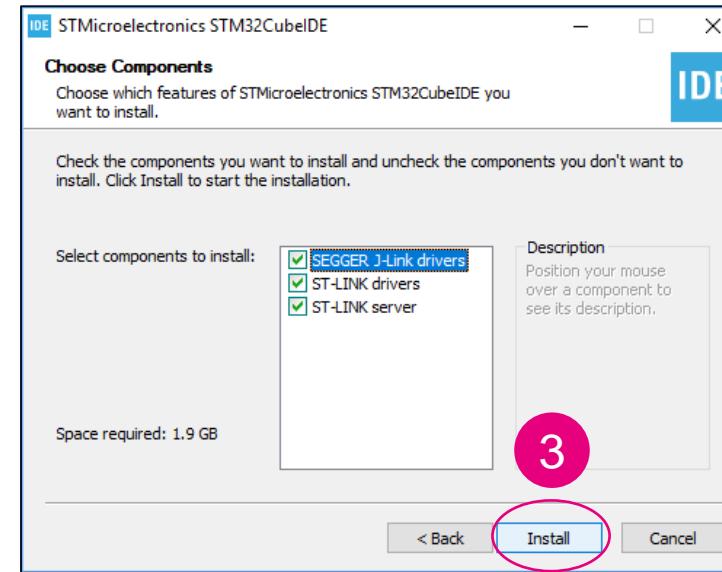
安裝STM32CubeIDE – 1



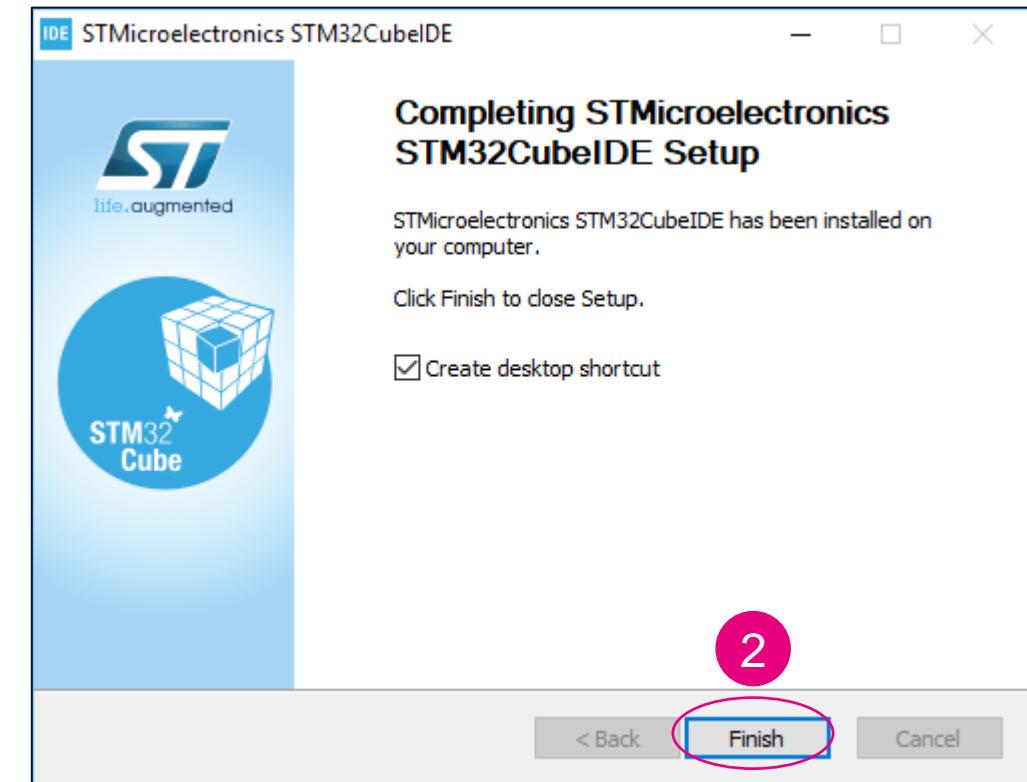
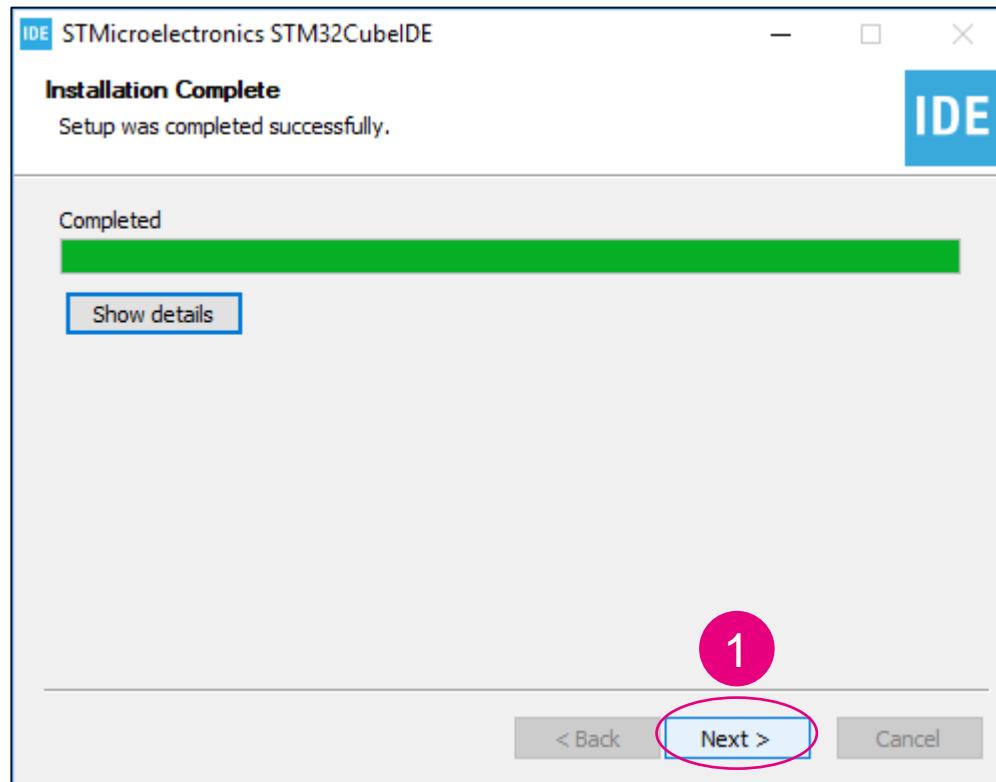
安裝STM32CubeIDE – 2



- 建議使用預設安裝路徑
- 若不使用預設安裝路徑，請確認安裝路徑名稱當中沒有空格



安裝STM32CubeIDE – 3

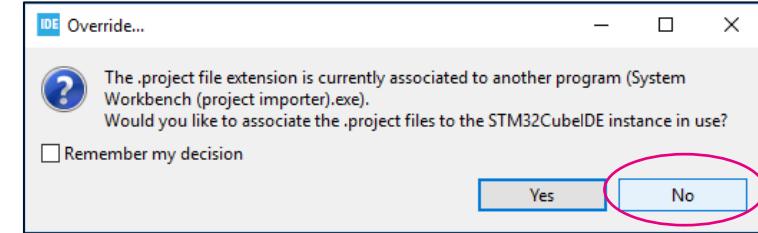
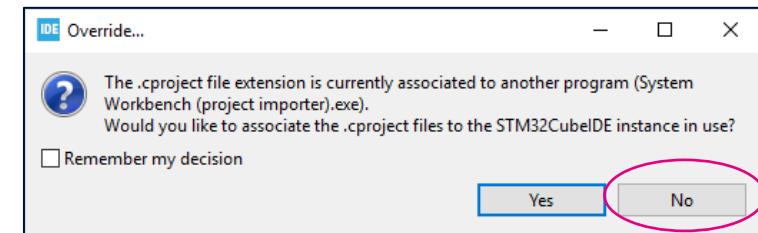
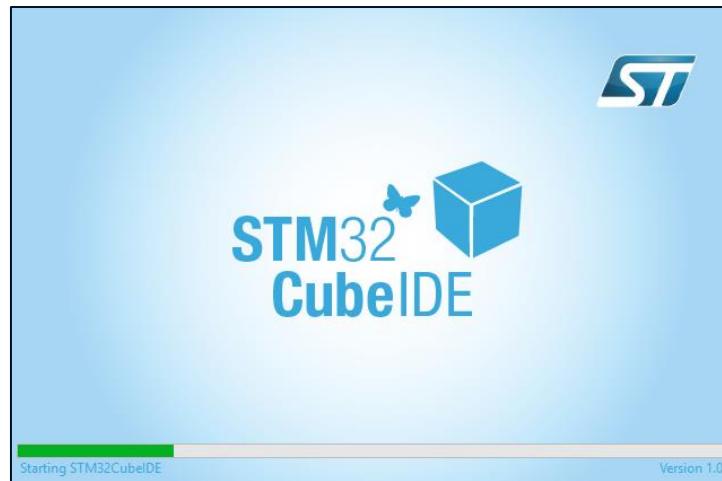
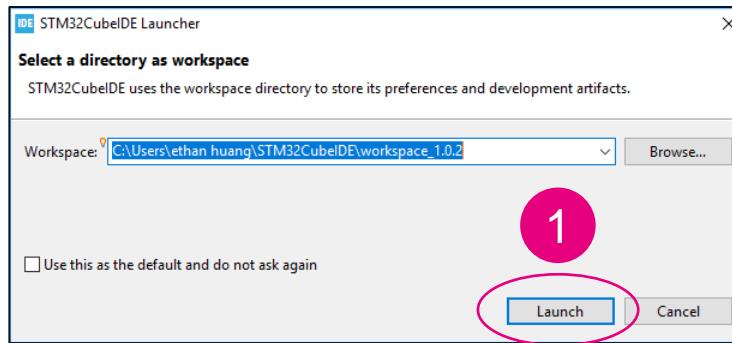


安裝完成

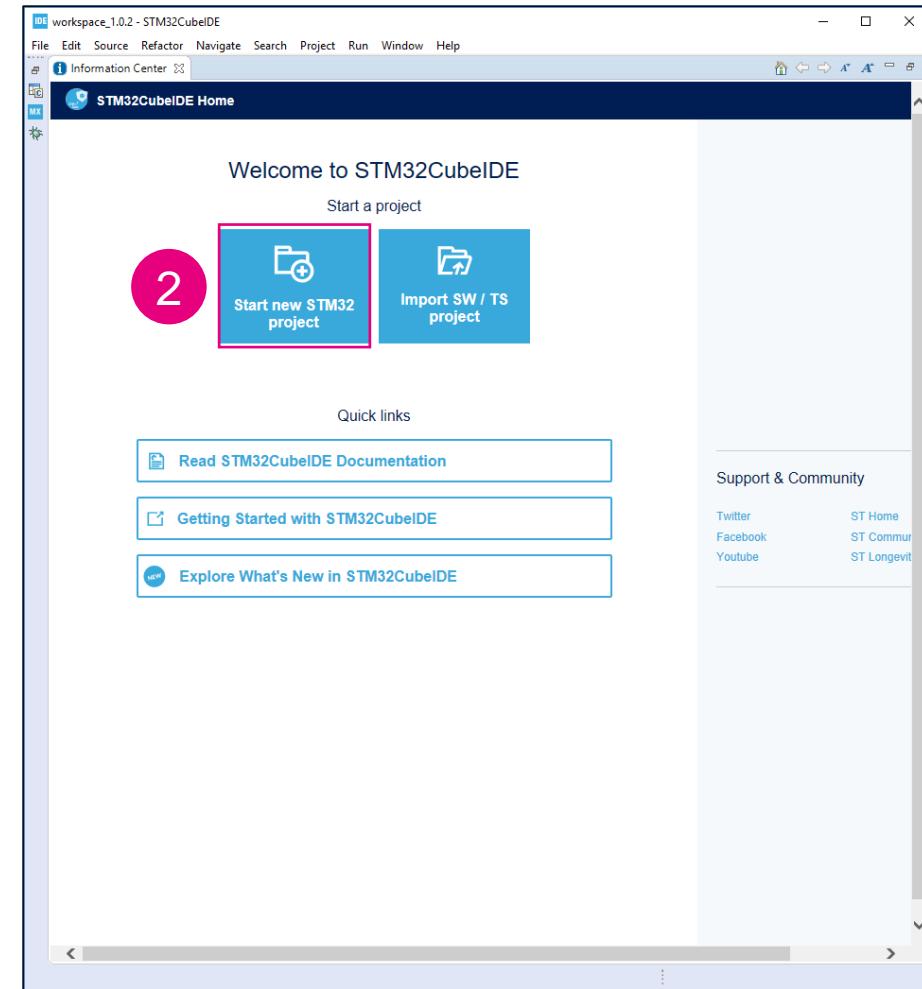
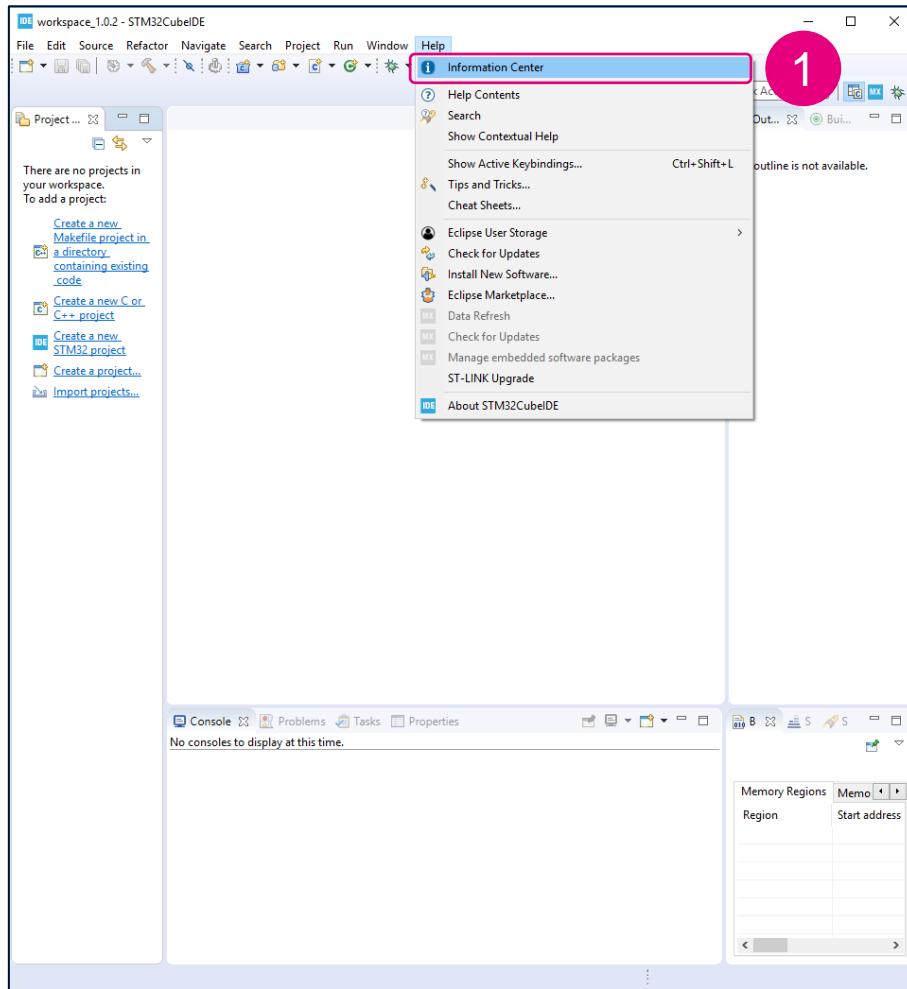
軟體安裝

STM32CubeF7 MCU Package 安裝

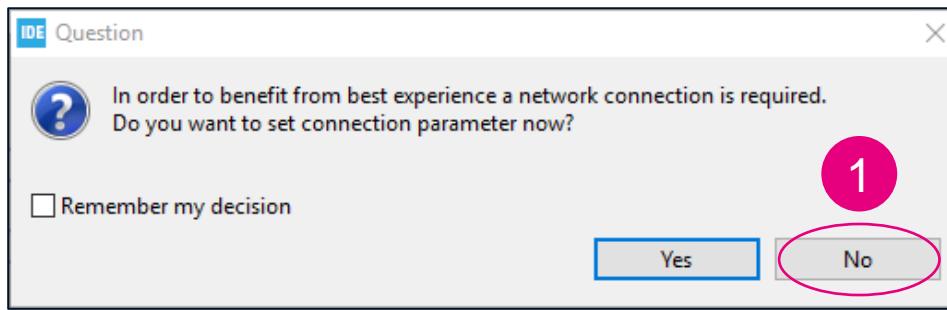
開啟STM32CubeIDE



開始STM32F7新專案 (1)



開始STM32F7新專案 (2)



The screen shows the "Target Selection" step of the STM32 Project setup. A pink circle labeled "2" is around the "Board Selector" tab. A pink circle labeled "3" is around the search bar where "F746" is typed. A pink circle labeled "4" is around the "32F746GDISCOVERY" board entry in the "Boards List". A pink circle labeled "5" is around the "Next >" button at the bottom right.

STM32 Project
Target Selection
Select STM32 target

MCU/MPU Selector Board Selector Cross Selector

Part Number Search: F746

Vendor: >

Type: >

MCU/MPU Series: >

Other: >

Price: From 23.0 to 54.0
23.0 54.0

Oscillator Freq.: From 0 to 25 (MHz)
0 25

Peripheral:

- Accelerometer: 0
- Analog I/O: 0
- Arduino Form Factor: 0
- Audio Line In: 0
- Audio Line Out: 0
- Battery: 0
- Button: 0 2
- CAN: 0 0
- Camera: 0
- Compass: 0 0
- Custom Form Factor: 0 0
- Digital I/O: 0 256
- Ethernet: 0

STM32F746GDISCOVERY

STM32F746GDISCOVERY Board Support and Examples

ACTIVE Active Product is in mass production Unit Price (US\$): 54.0
Mounted device: STM32F746NGHx

The STM32F7 discovery kit allows users to develop and share applications with the STM32F7 Series microcontrollers based on ARM® Cortex®-M7 core. The discovery kit enables a wide diversity of applications taking benefit from audio, multi-sensor support, graphics, security, video and high-speed connectivity features. The Arduino connectivity support provides unlimited expansion capabilities with a large choice of specialized add-on boards.

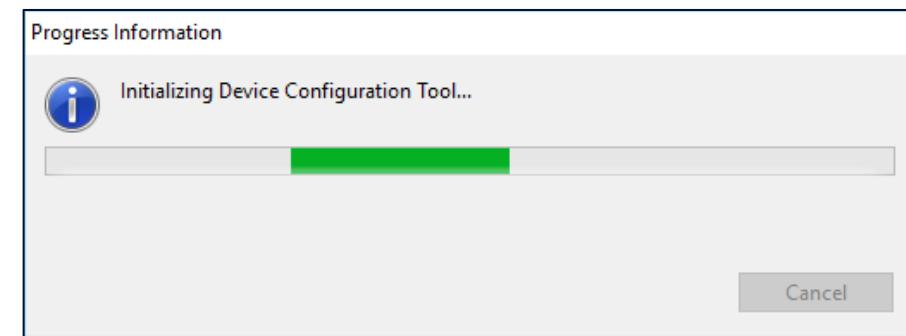
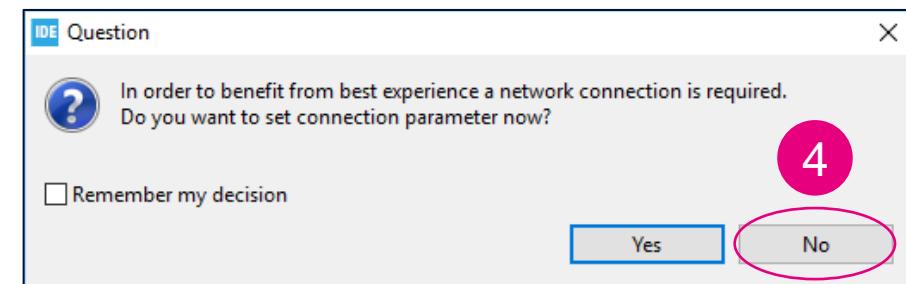
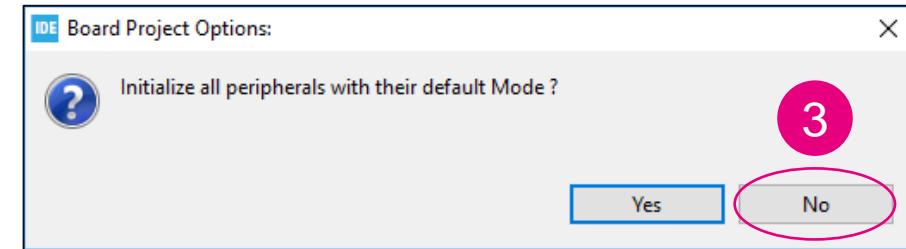
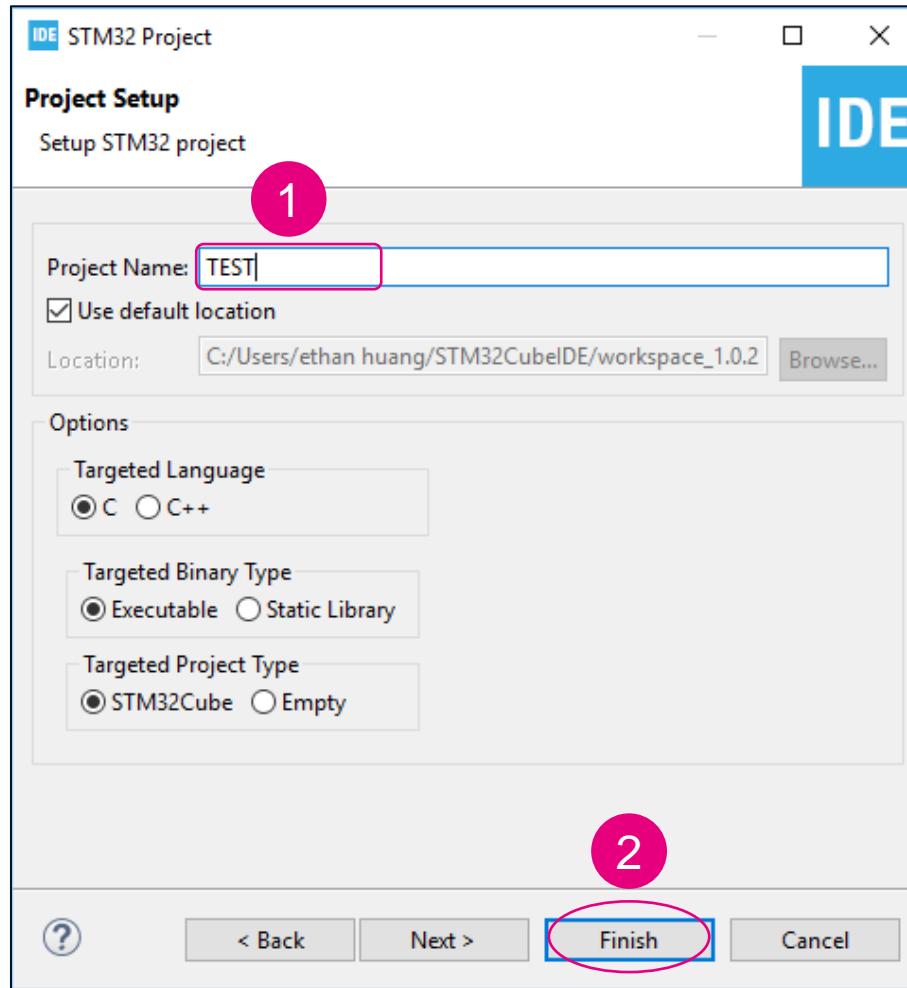
Features

Boards List: 2 items

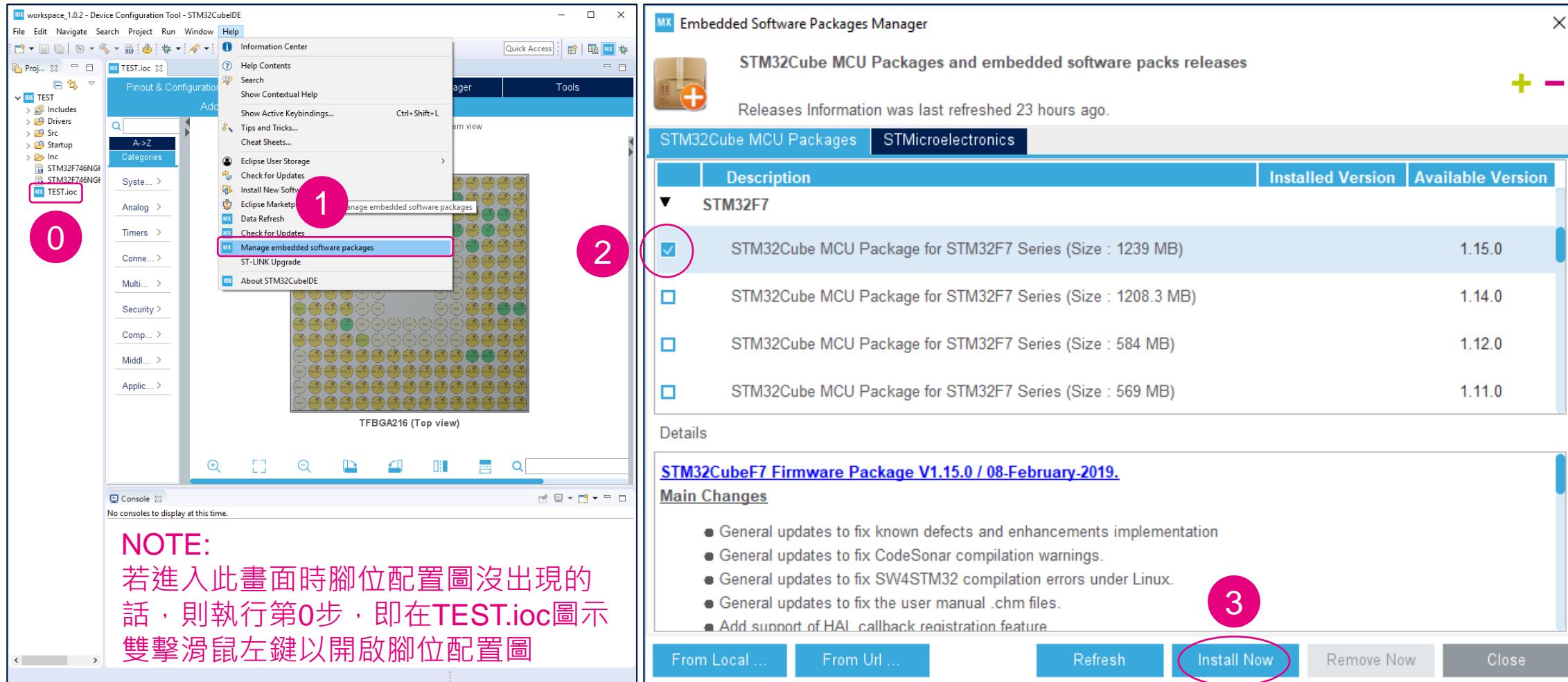
Overview	Part No.	Type	Marketing Status	Unit Price (US\$)	Mounted Device
	32F746GDISCOVERY	Discovery kit	Active	54.0	STM32F746NGHx
	NUCLEO-F746ZG	Nucleo144	Active	23.0	STM32F746ZGTx

< Back Next > Finish Cancel

開始STM32F7新專案 (3)

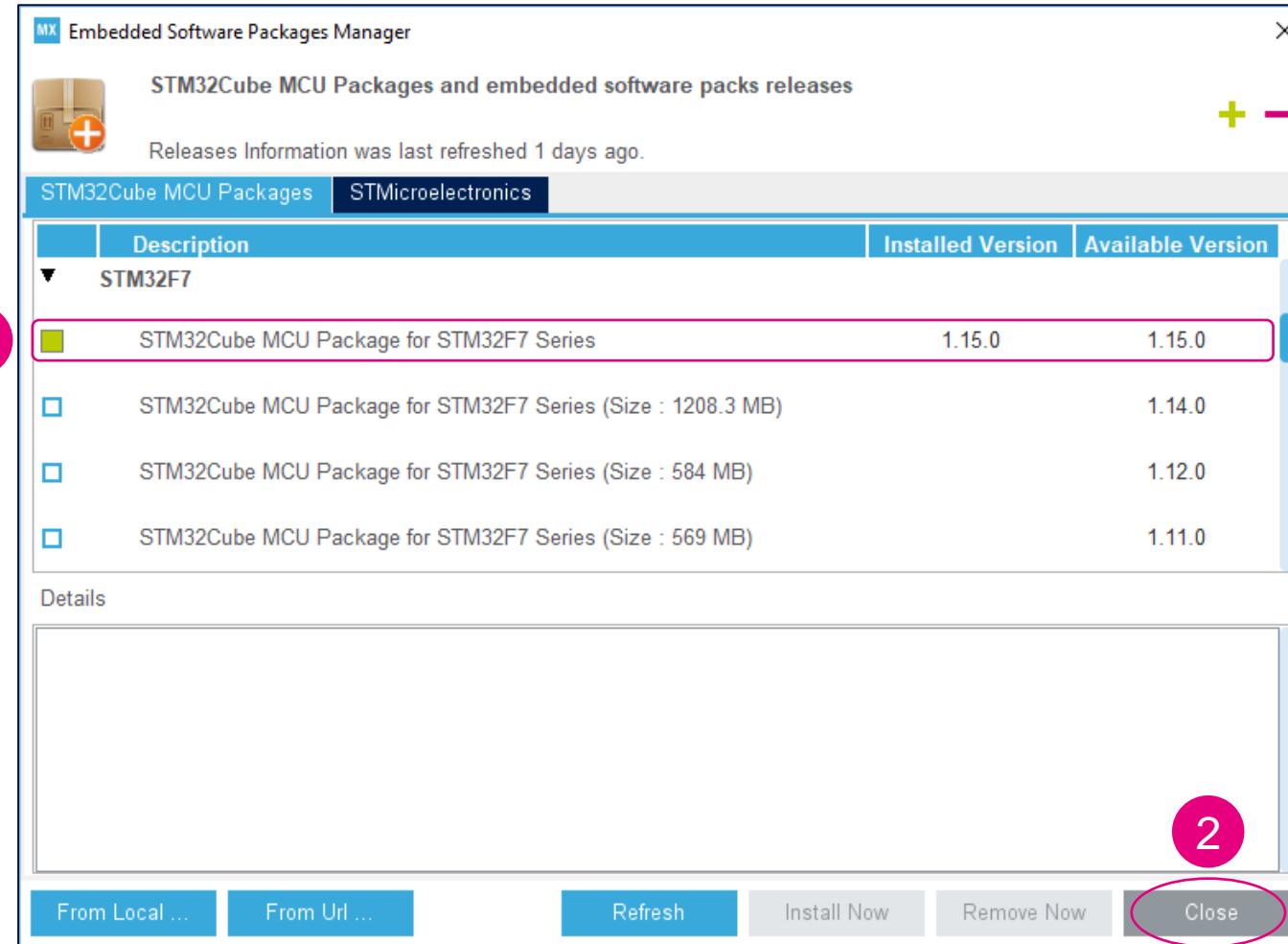


開始STM32F7新專案 (4)



STM32CubeF7 MCU Package 安裝完成

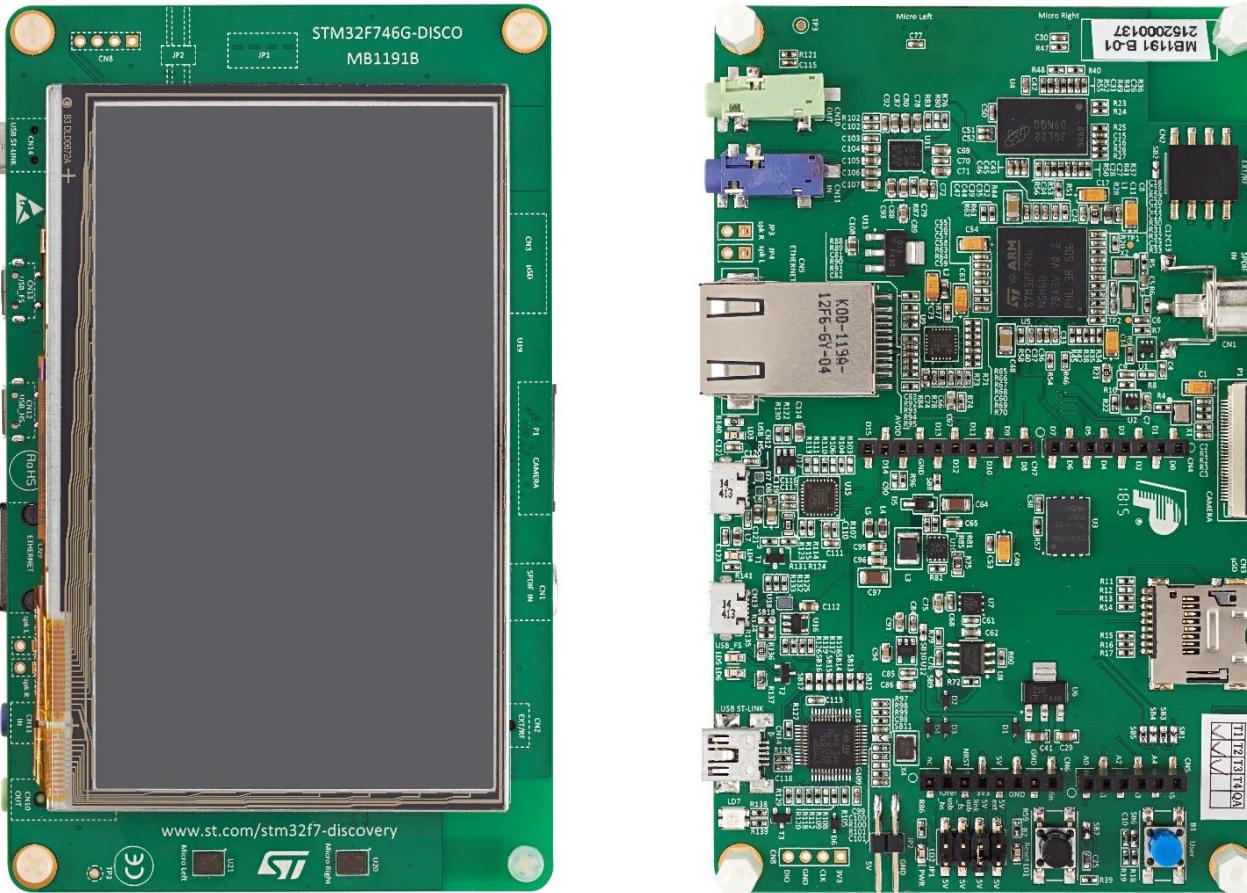
完裝完成



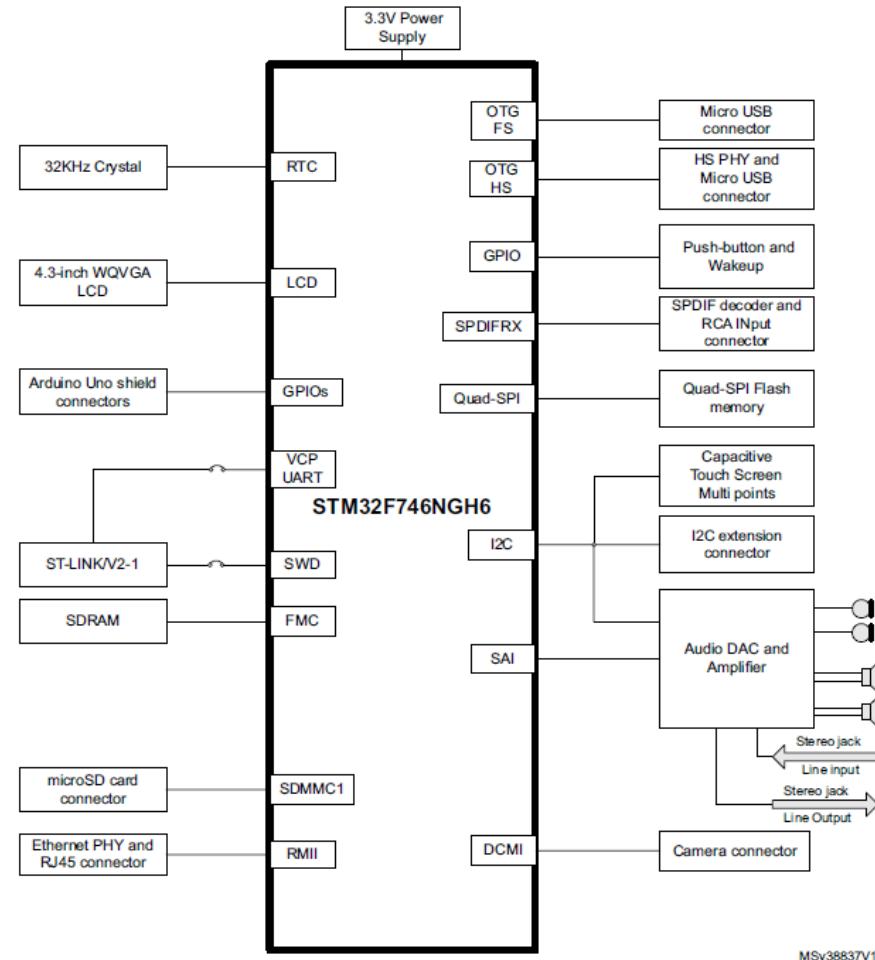
STM32F746 Discovery開發板



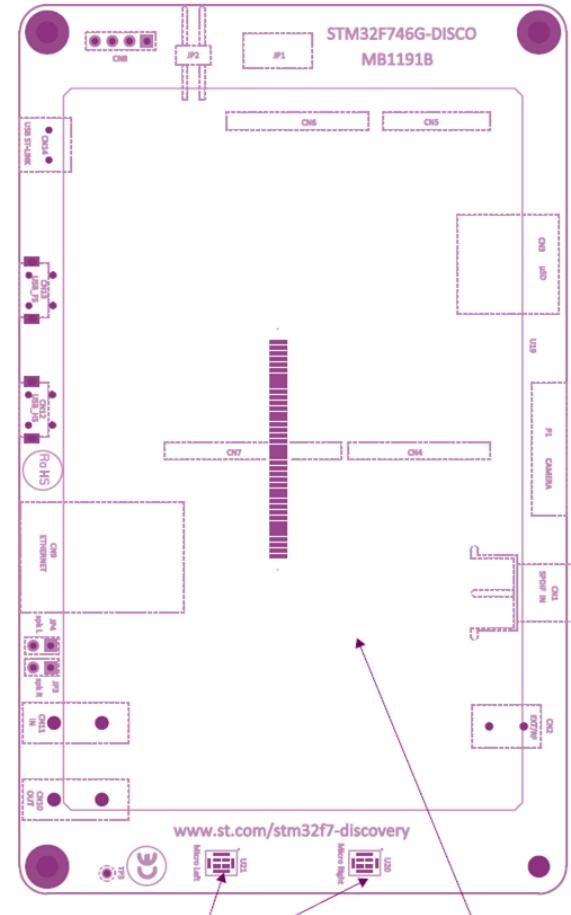
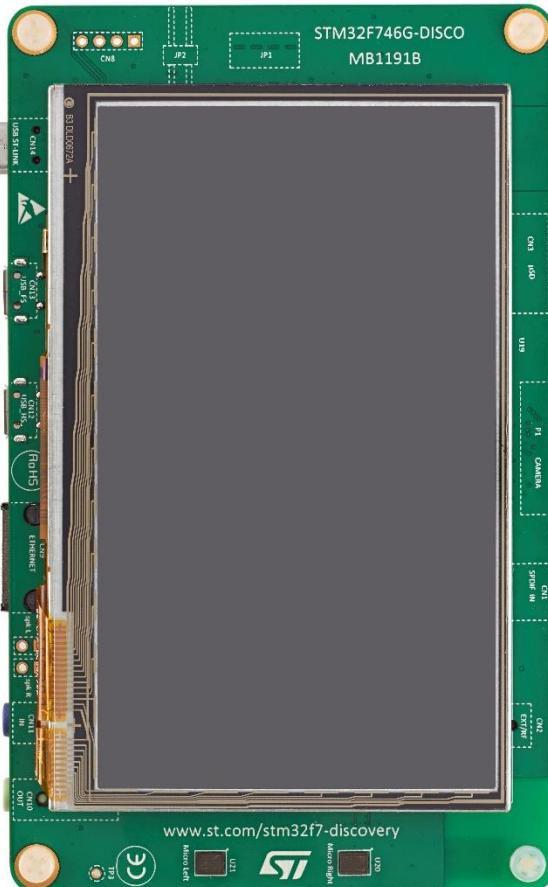
32F746GDISCOVERY



方塊圖



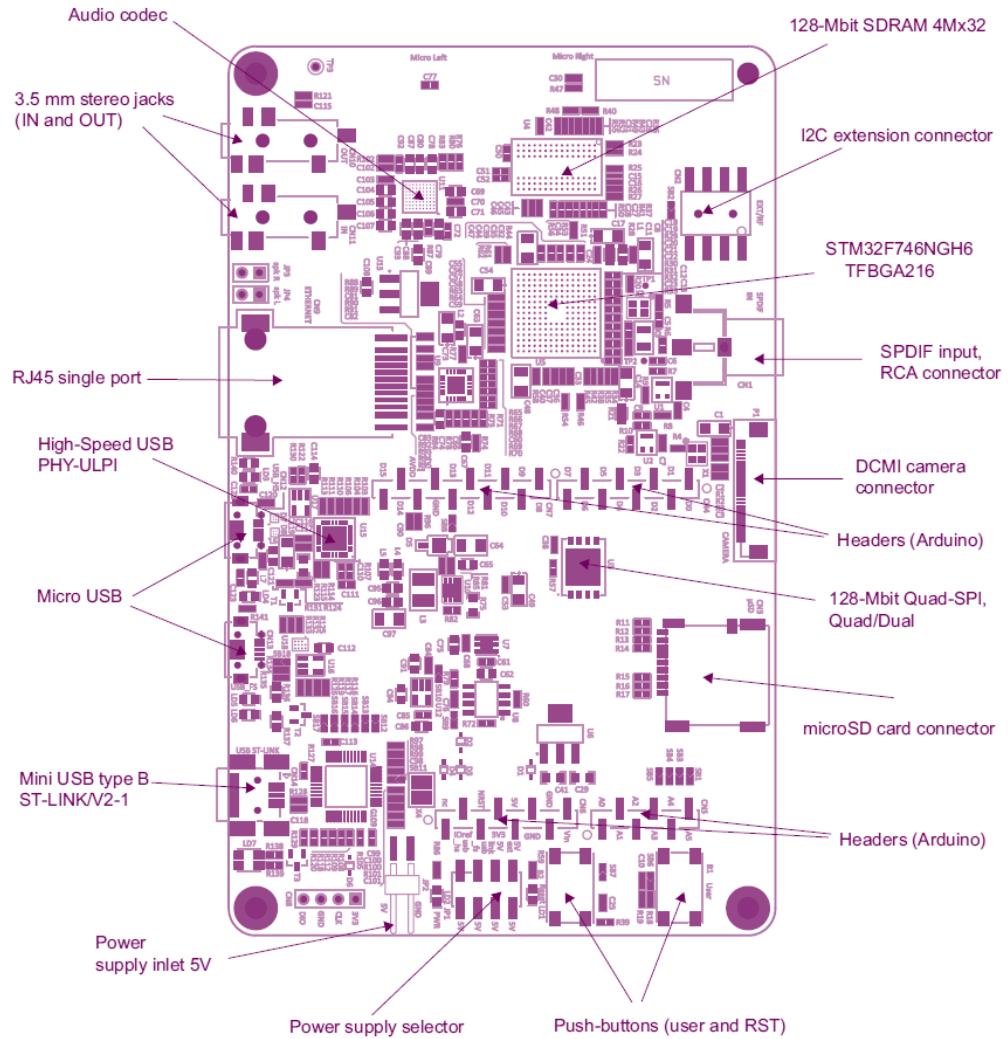
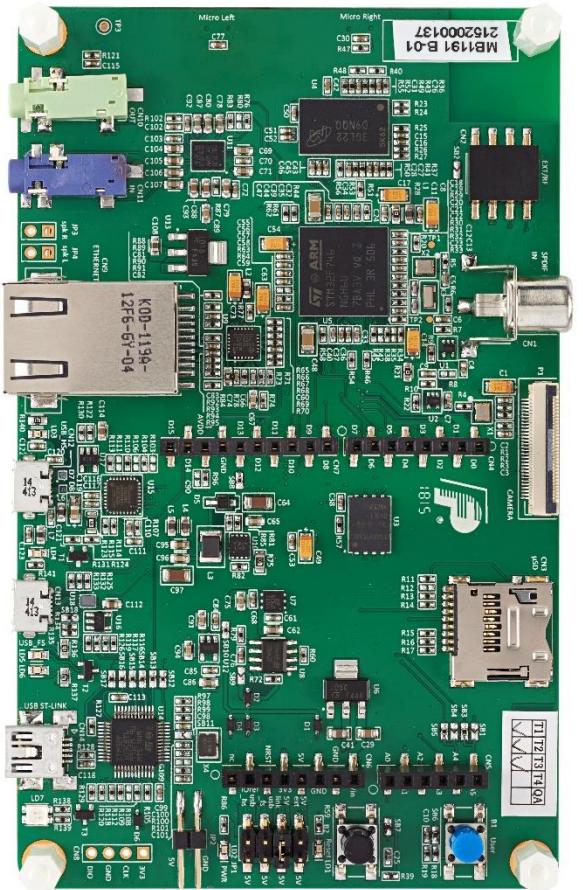
開發板說明



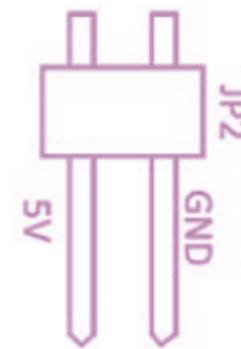
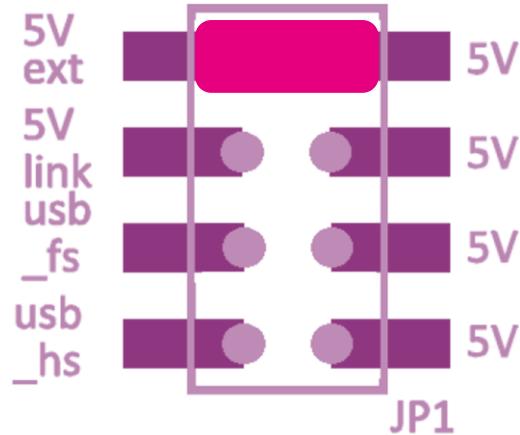
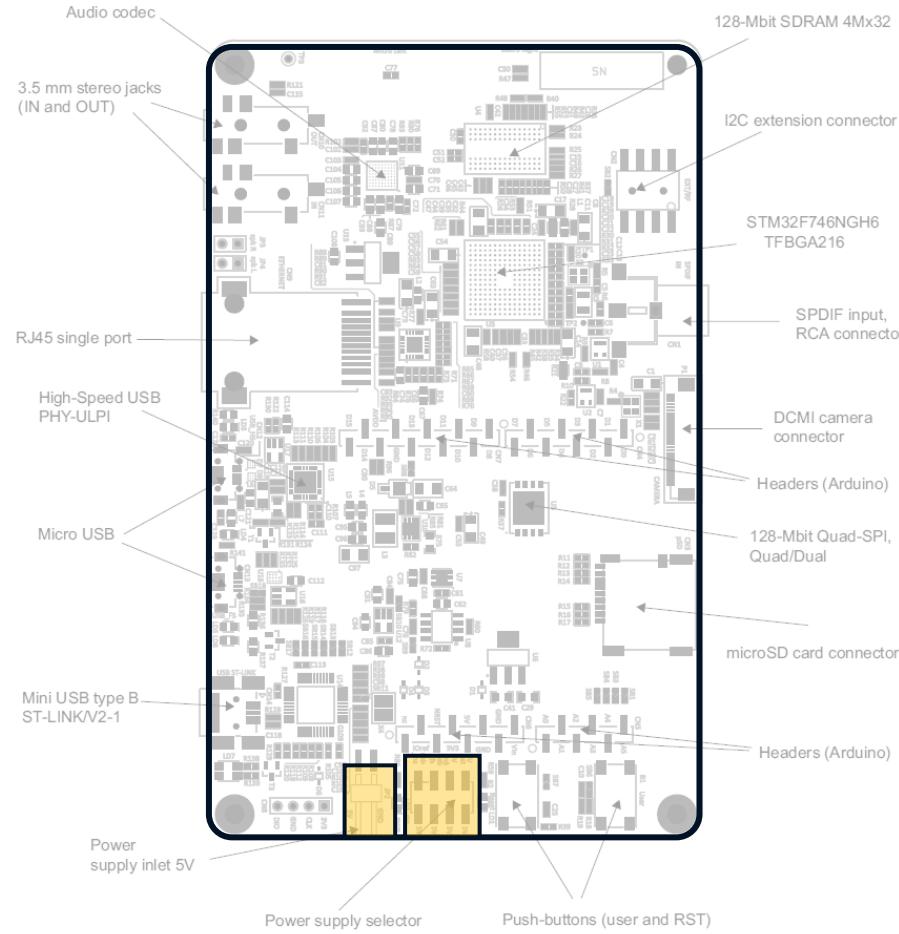
Two ST-MEMS microphones

4.3-inch 480x272 color LCD-TFT
with capacitive touch screen

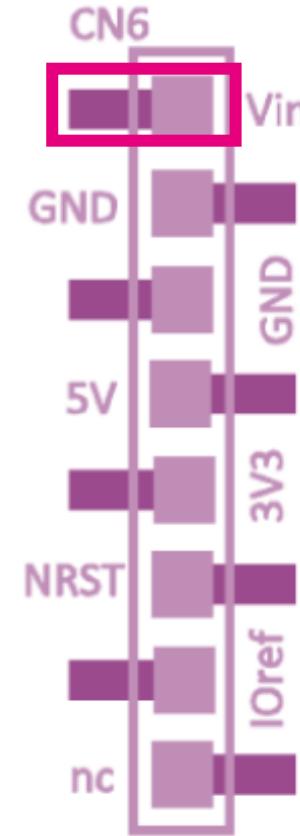
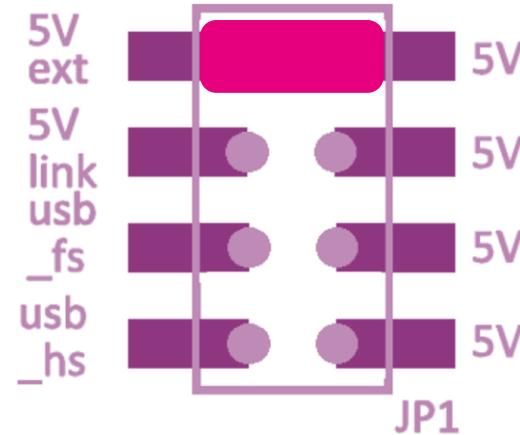
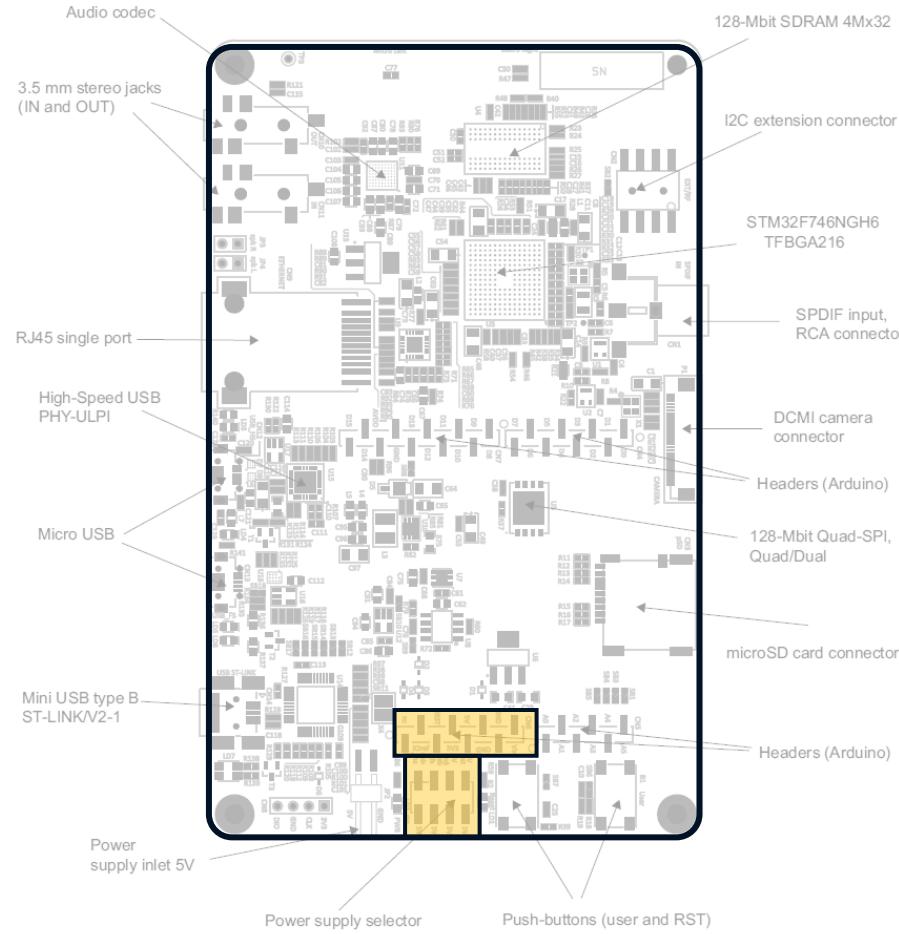
開發板說明



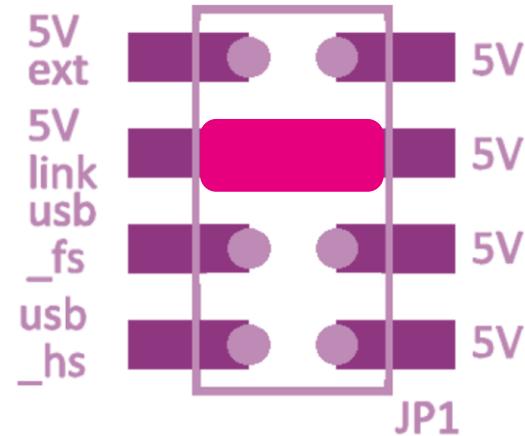
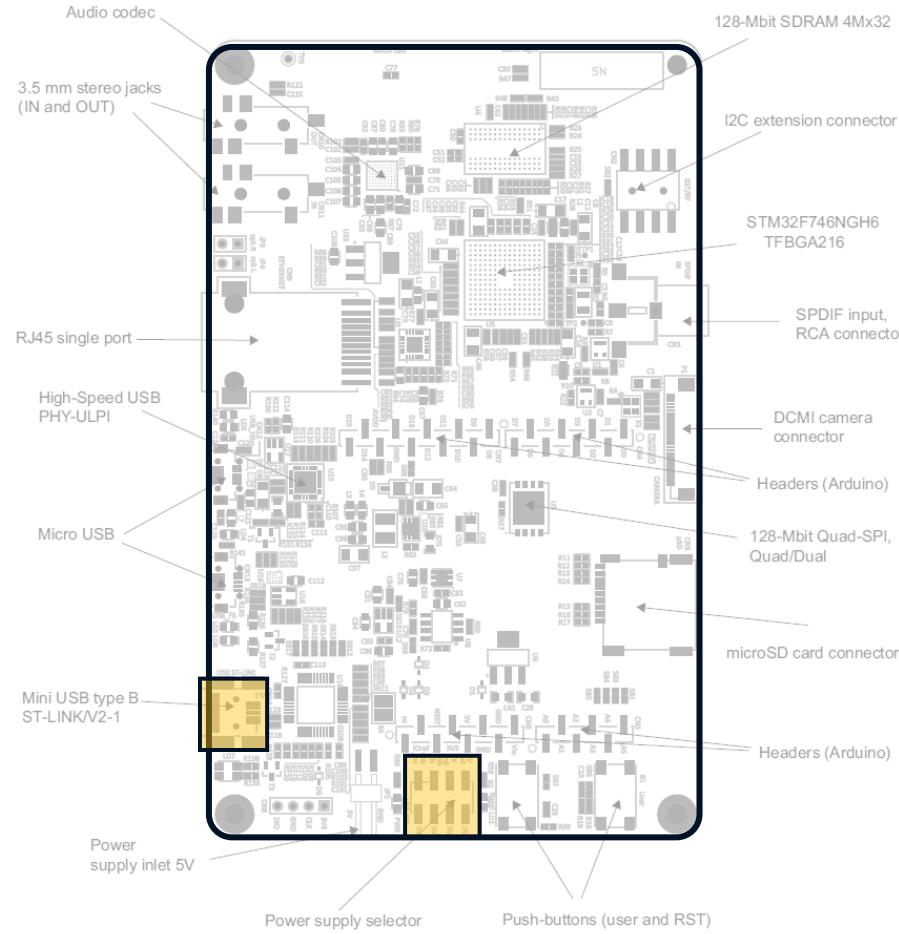
電源供應(1)



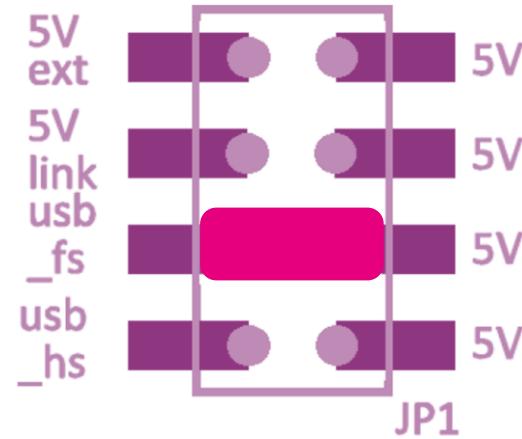
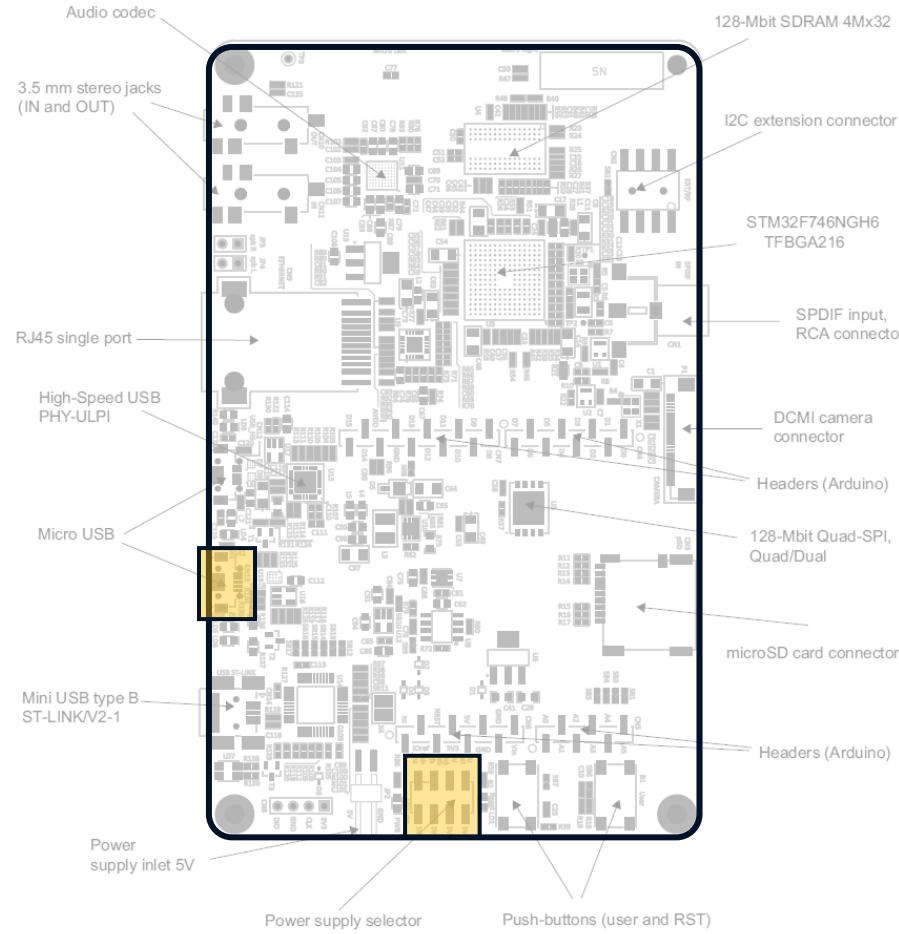
電源供應(2)



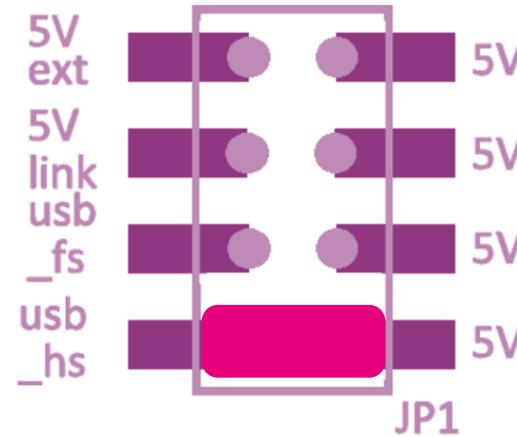
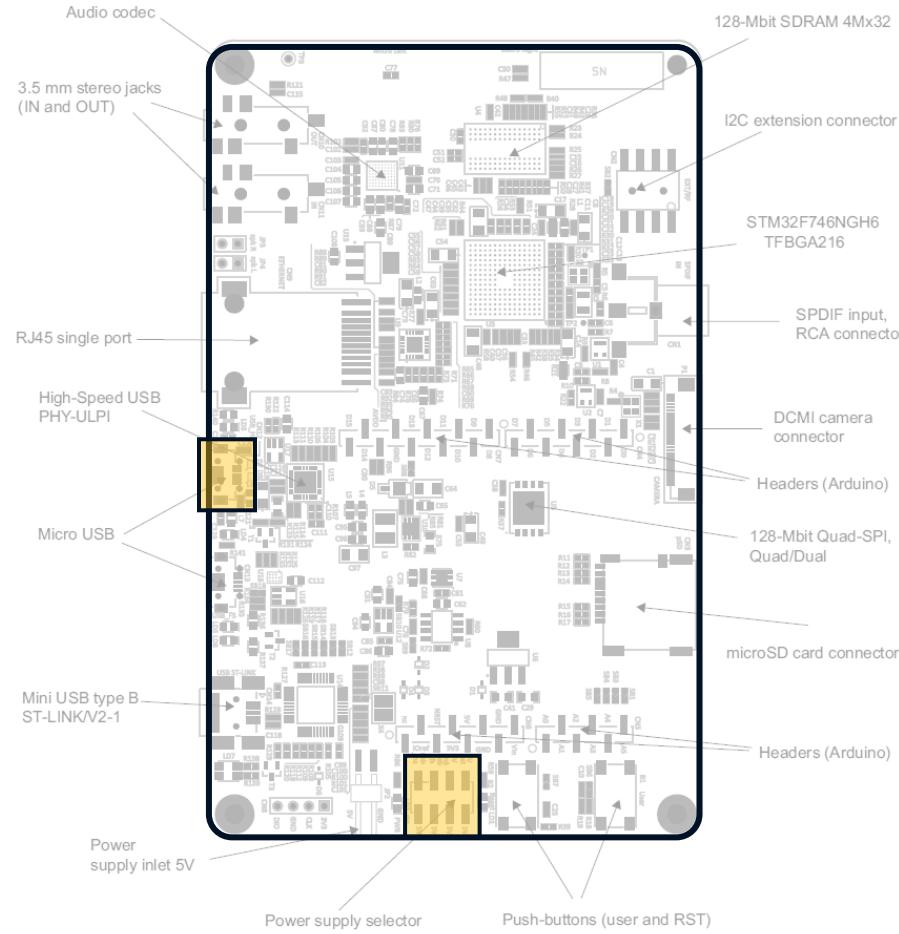
電源供應(3)



電源供應(4)

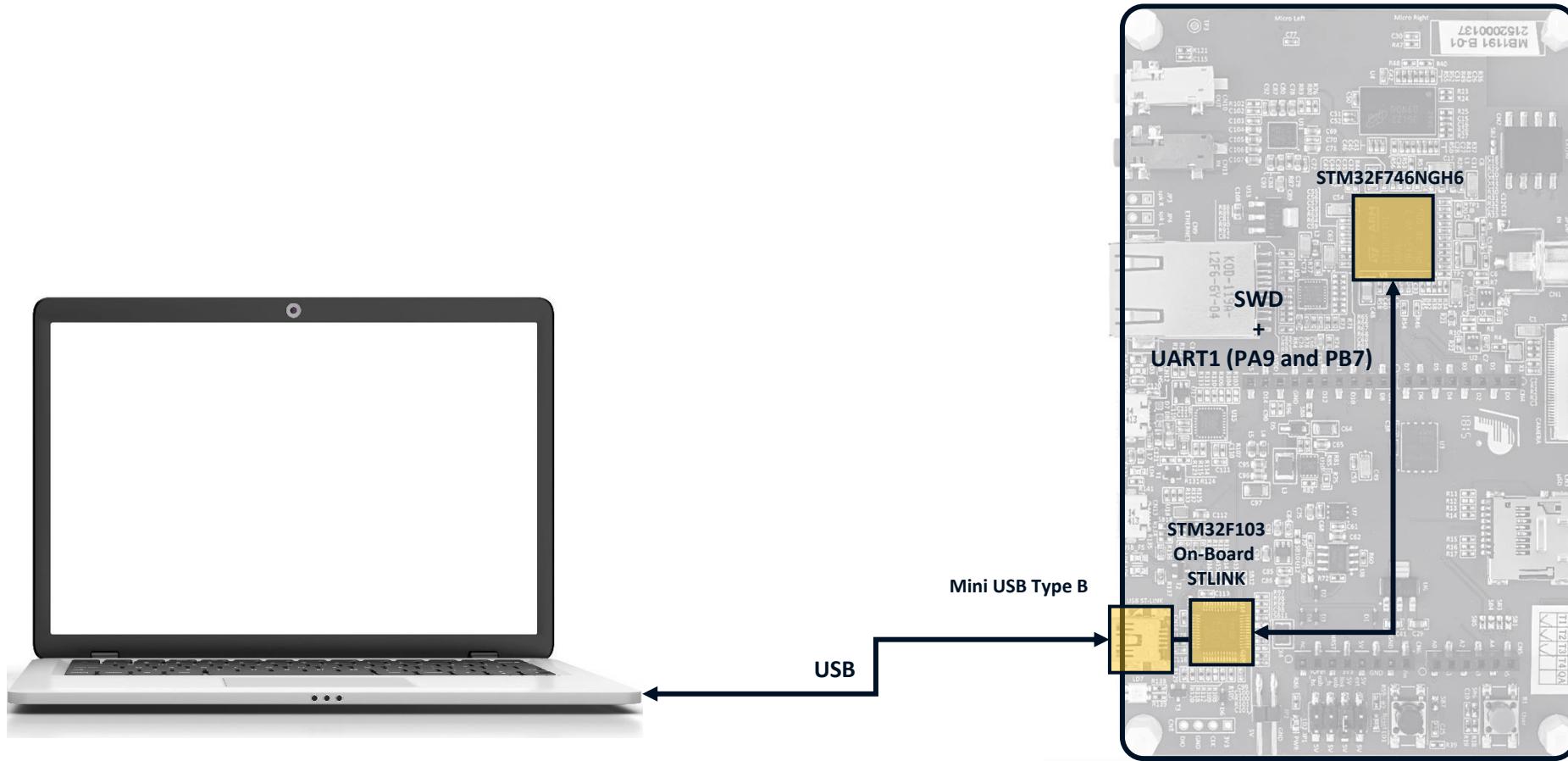


電源供應(5)

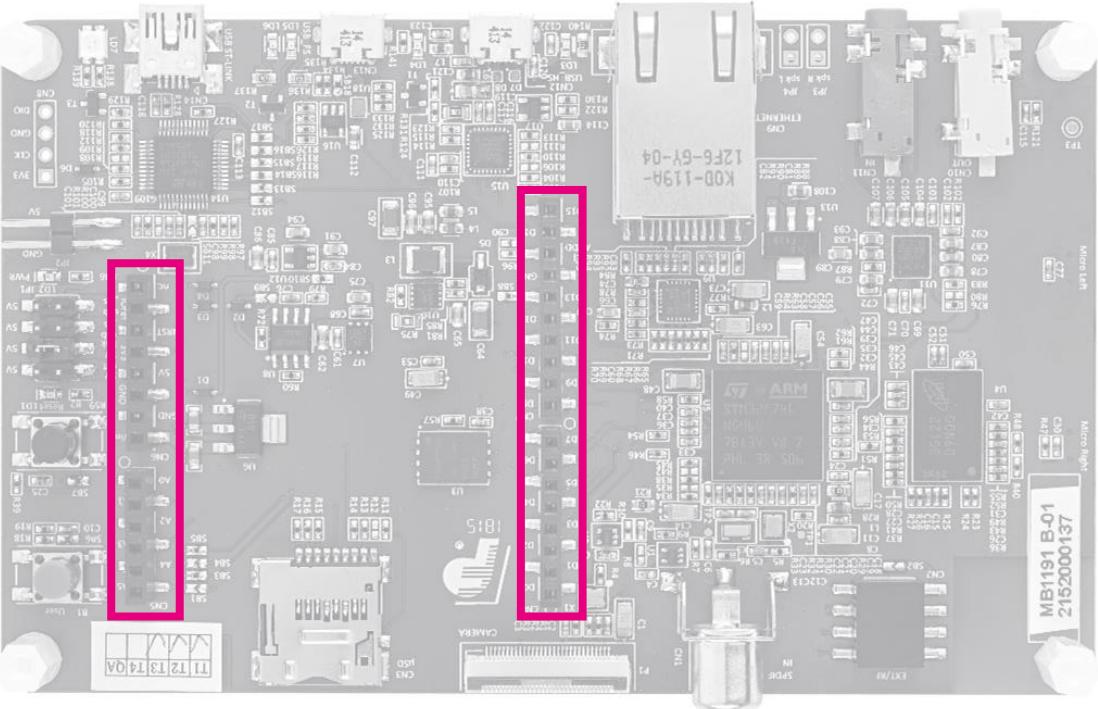




Virtual Comport



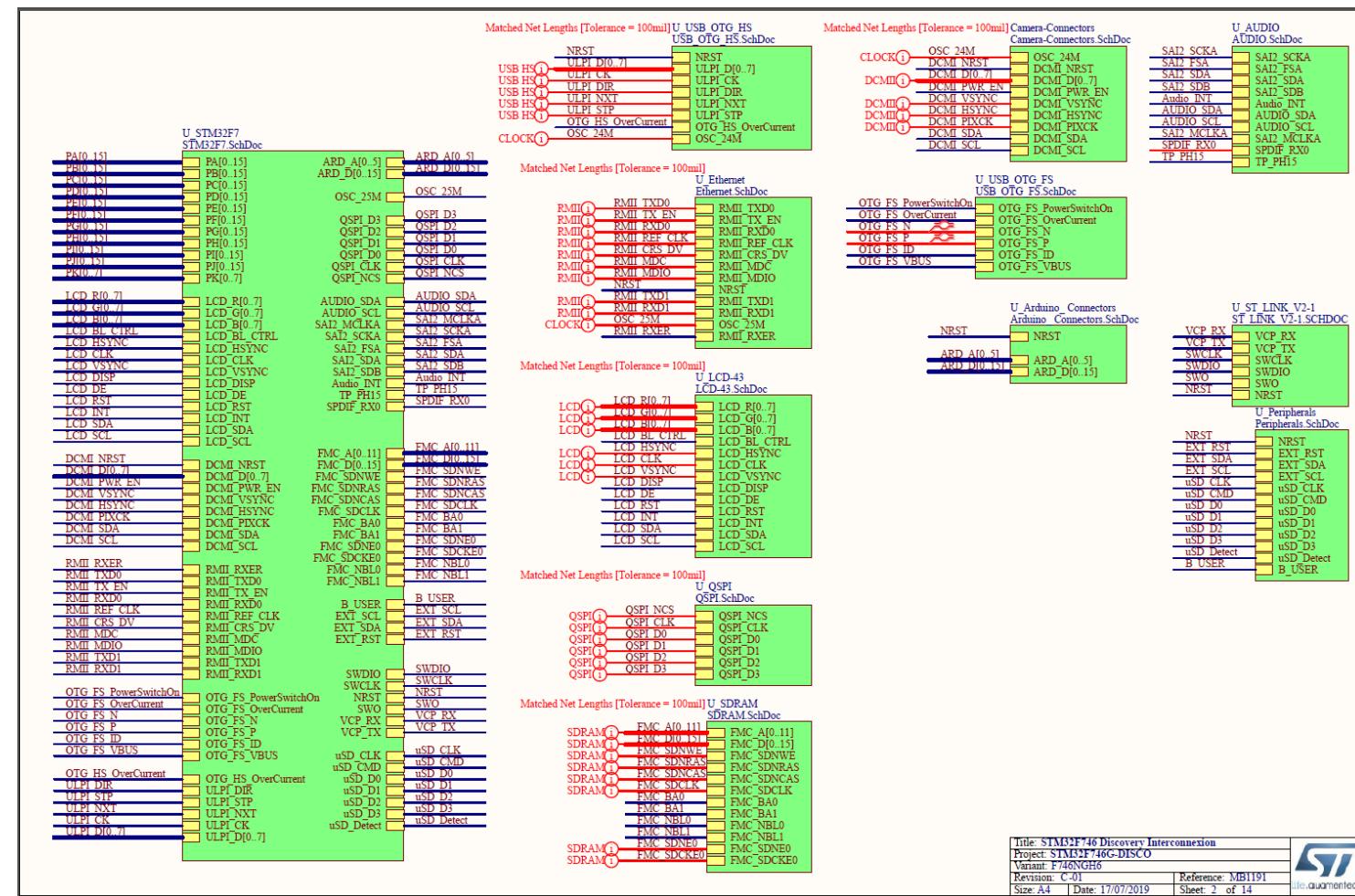
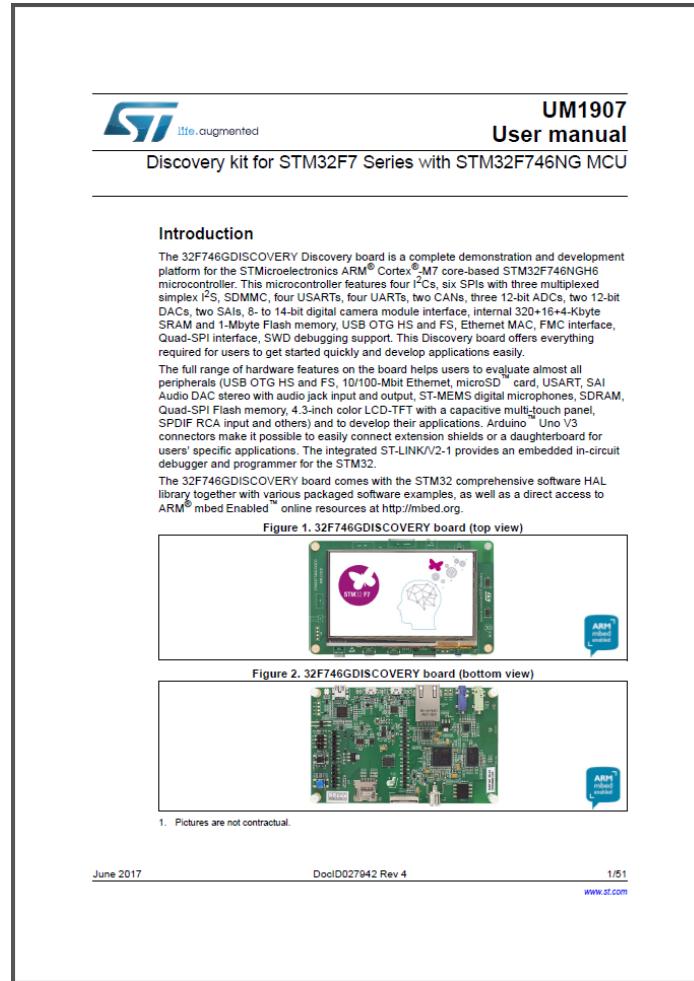
Arduino Connectors



	CN6	CN5
Reserved	1 NC	
3V3	2 IOREF	
NRST	3 RESET	
3V3	4 3V3	
5V	5 5V	
GND	6 GND	
GND	7 GND	
VIN	8 VIN	
PA0	1 A0	
PF10	2 A1	
PF9	3 A2	
PF8	4 A3	
PF7	5 A4	
PF6	6 A5	

	CN7	CN4
SCL/D15	10 PB8	
SDA/D14	9 PB9	
AVDD	8 AVDD	
GND	7 GND	
SCK/D13	6 PI1 / LD1	
MISO/D12	5 PB14	
PWM/MOSI/D11	4 PB15	
PWM/D10	3 PA8	
PWM/D9	2 PA15	
D8	1 PI2	
D7	8 PI3	
PWM/D6	7 PH6	
PWM/CS/D5	6 PIO	
D4	5 PG7	
PWM/D3	4 PB4	
D2	3 PG6	
TX/D1	2 PC6	
RX/D0	1 PC7	

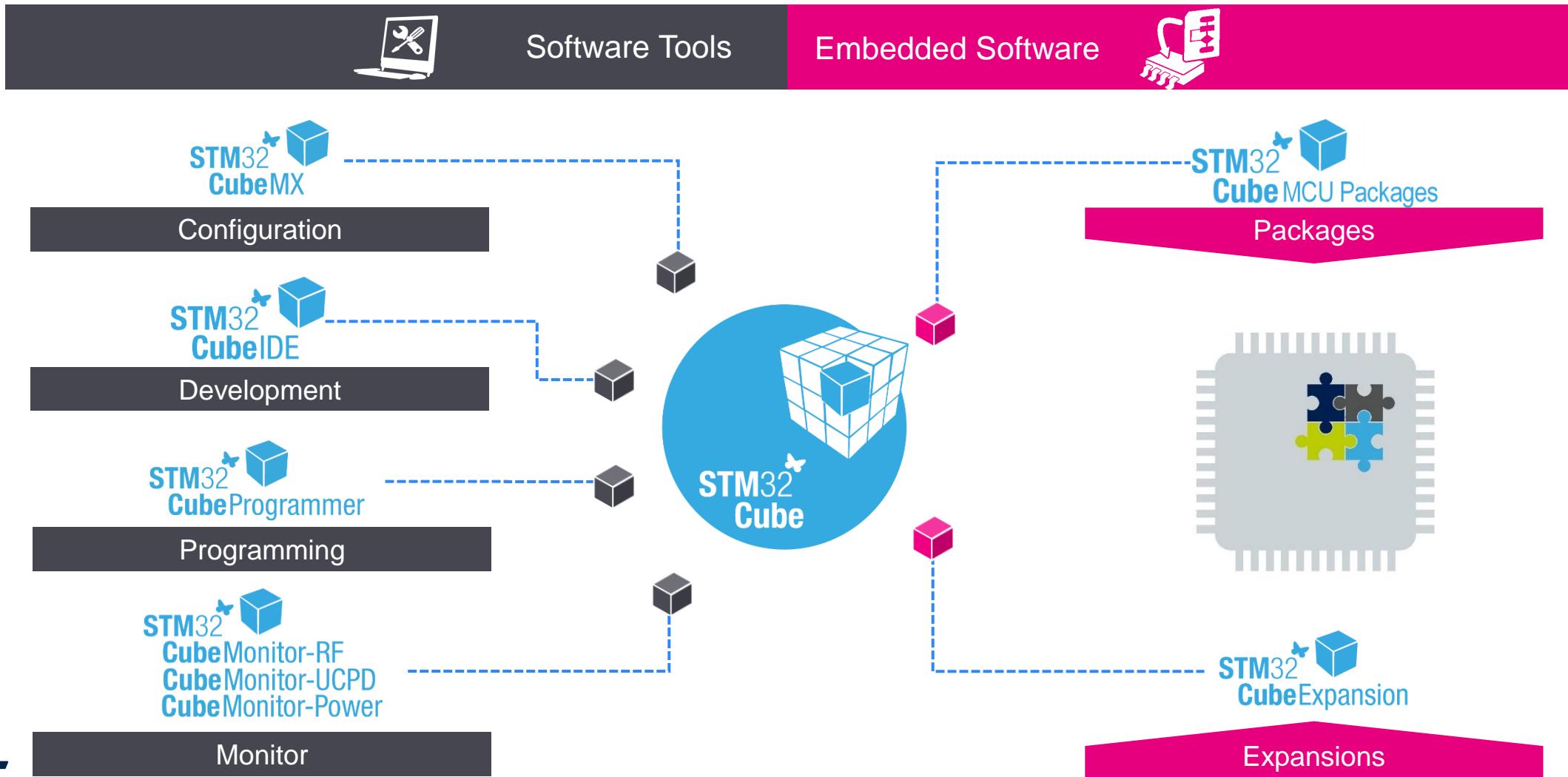
32F746DISCOVERY文件



STM32CubeIDE簡介



STM32Cube Ecosystem



Background of STM32CubeIDE

History



2017 December

2019 May

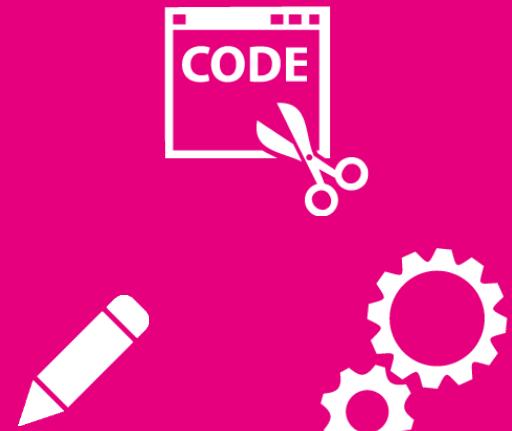


One Tool for All Your STM32 Development

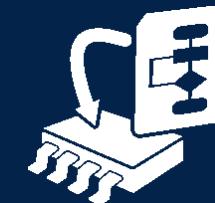
Chipset / Board Configuration



Code Development



Validation Debug

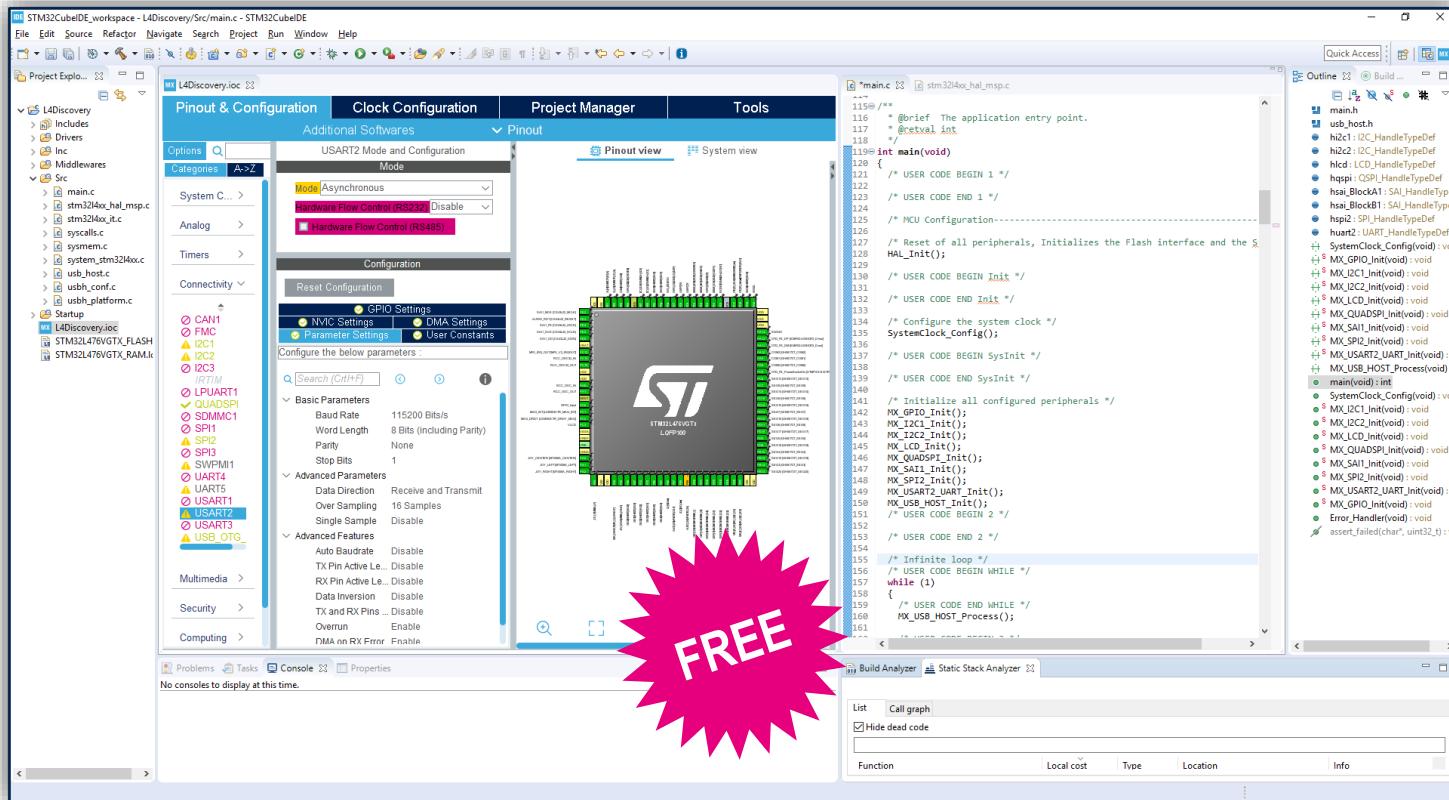


Free Multi-Platform Development Tool

eclipse



Eclipse/GCC Based



Windows



macOS

Project Management



Importer

Welcome to STM32CubeIDE

Start a project

Start new STM32 project

Import SW / TS project

TrueSTUDIO Project

SW4STM32 Project

IOC Editor

st-community - Device Configuration Tool - STM32CubeIDE

F429-TEST.ioc

Pinout & Configuration

USB_HOST Mode and Configuration

Mode

Class for HS IP Communication Host Class (Virtual)

Class for FS IP Disable

Configuration

User Constants Platform Settings

Host Configuration

USBH_MAX_NUM... 2

USBH_MAX_NUM... 2

USBH_MAX_NUM... 1

usb_host.c

```
76 if (USBH_Init(&hUsbHostHS, USBH_UserProcess, HOST_HS) != USBH_OK)
77 {
78     Error_Handler();
79 }
80 if (USBH_RegisterClass(&hUsbHostHS, USBH_CDC_CLASS) != USBH_OK)
81 {
82     Error_Handler();
83 }
```



Code Editor – Navigation

Symbol Hyperlink

```
BSP_LED_Init(LED1);
BSP_LED_Init(LED2);
BSP_LED_Init(LED3);
BSP_LED_Init(LED4);

void BSP_LED_Init(L...
```

Include Browser

Include Browser X

Files included by 'main.c(/F746-DISCO-HID/Src)' - in workspace

- main.c
- main.h
- stm32f7xx_hal.h
- stm32f7xx_hal_conf.h
- stm32f7xx_hal_rcc.h
- stm32f7xx_hal_exti.h
- stm32f7xx_hal_gpio.h
- stm32f7xx_hal_dma.h
- stm32f7xx_hal_cortex.h

Call Hierarchy

```
/* ... */
= HAL_StatusTypeDef USB_WritePacket(USB_OTG_GlobalTypeDef *USBx, uint8_t *src, uint8_t
```

Brace Navigation

```
294 /* Check the parameters */
295 assert_param(IS_OPTIONBYTE(pOBInit->OptionType));
296
297 /* Write protection configuration */
298 if((pOBInit->OptionType & OPTIONBYTE_WRP) == OPTIONBYTE_WRP)
00 {
01     assert_param(IS_WRPSTATE(pOBInit->WRPState));
02     if(pOBInit->WRPState == OB_WRPSTATE_ENABLE)
03     {
04         /*Enable of Write protection on the selected Sector*/
05         status = FLASH_OB_EnableWRP(pOBInit->WRPSector);
06     }
07     else
08     {
09         /*Disable of Write protection on the selected Sector*/
10         status = FLASH_OB_DisableWRP(pOBInit->WRPSector);
11     }
12 }
```

Macro Expansion Browser

Explore Macro Expansion - 2 step(s)

```
#define USB_OTG_FS ((USB_OTG_GlobalTypeDef *) USB_OTG_FS_PERIPH_BASE)
```

Original	Fully Expanded
USB_OTG_FS	((USB_OTG_GlobalTypeDef *) 0x50000000UL)

Explore Macro Expansion - 2 step(s)

```
#define USB_OTG_FS ((USB_OTG_GlobalTypeDef *) USB_OTG_FS_PERIPH_BASE)
```

Original	Expansion #1 of 2
USB_OTG_FS	((USB_OTG_GlobalTypeDef *) USB_OTG_FS_PERIPH_BASE)

Explore Macro Expansion - 2 step(s)

```
#define USB_OTG_FS_PERIPH_BASE 0x50000000UL
```

Expansion #1 of 2	Fully Expanded
USB_OTG_FS	((USB_OTG_GlobalTypeDef *) 0x50000000UL)

Type Hierarchy

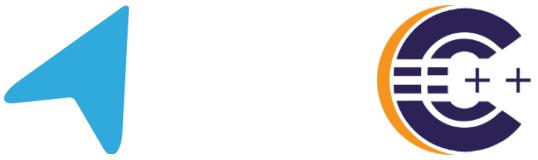
```
46 UART_HandleTypeDef huart2;
47 /* USER CODE BEGIN PV */
48
49
50
51
52
```

Type Hierarchy X

UART_HandleTypeDef

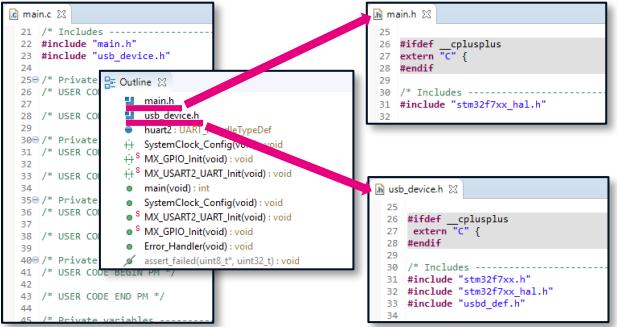
- _UART_HandleTypeDef
- UART_HandleTypeDef

- AdvancedInit : UART_AdvFeatureInitTypeDef - _UART_HandleTypeDef
- ErrorCode : volatile uint32_t - _UART_HandleTypeDef
- gState : volatile HAL_UART_StateTypeDef - _UART_HandleTypeDef
- hdmarx : DMA_HandleTypeDef * - _UART_HandleTypeDef
- hdmatx : DMA_HandleTypeDef * - _UART_HandleTypeDef
- Init : UART_InitTypeDef * - _UART_HandleTypeDef
- Lock : HAL_LockTypeDef * - _UART_HandleTypeDef
- Mask : uint16_t - _UART_HandleTypeDef
- pRxBuffPtr : uint8_t * - _UART_HandleTypeDef
- pTxBuffPtr : uint8_t * - _UART_HandleTypeDef

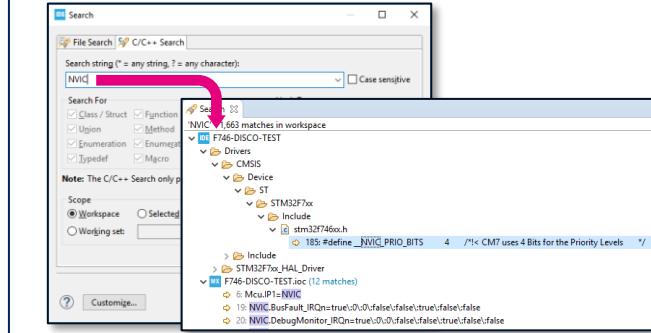


Code Editor – Navigation (2/2)

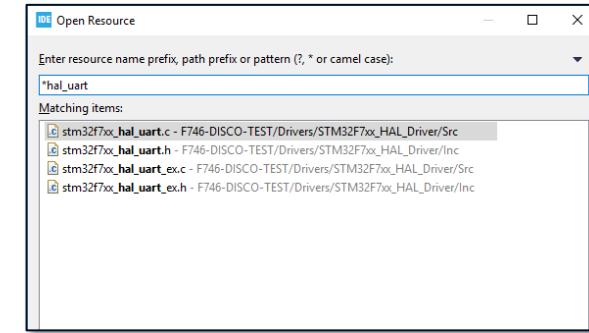
Outline View



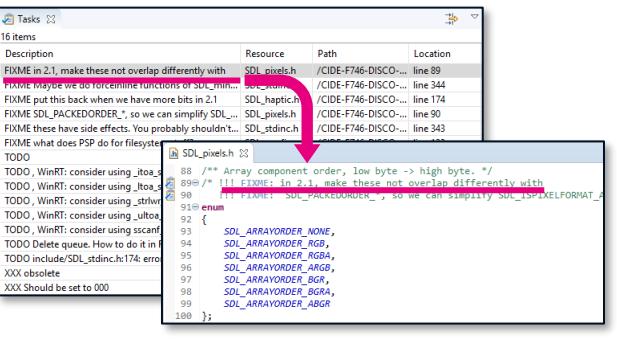
Powerful Search



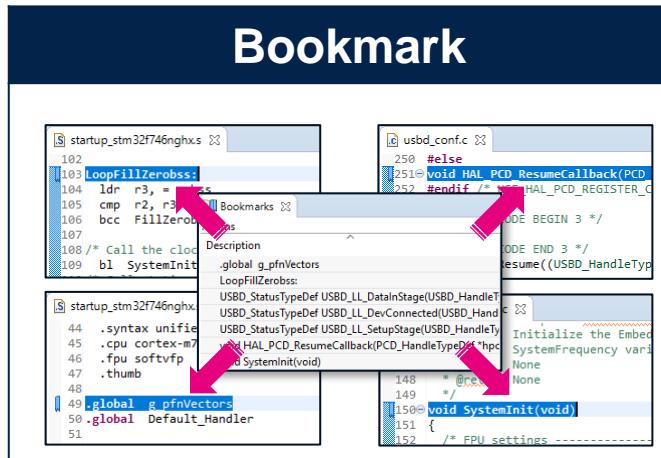
Open Resource



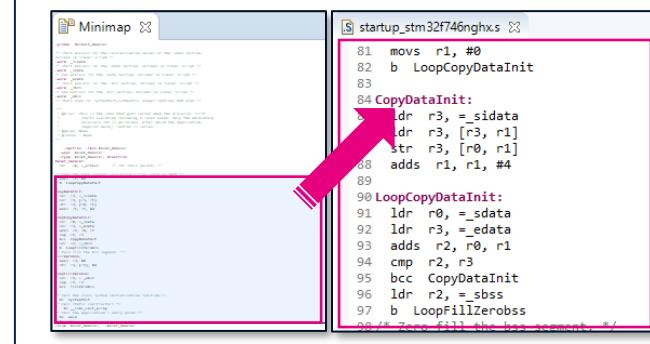
Task Tags



Bookmark



Minimap





Code Editor – Writing

Highlight Inactive Code

A screenshot of the Code Editor showing a file named `usbd_desc.c`. The code contains several sections of code highlighted in different colors: grey for comments, light blue for strings, and various shades of green and yellow for different code blocks. A specific section of code is highlighted in red, indicating it is inactive or commented out.

```

157
158 #if defined ( __ICARM__ ) /* IAR Compiler */
159 #pragma data_alignment=4
160 #endif /* defined ( __ICARM__ ) */
161 /* USB standard device descriptor. */
162 __ALIGN_BEGIN uint8_t USBD_F5_DeviceDesc[USB_LEN_DEV_DESC] __ALIGN_END =
163 {
164     0x12,           /*bLength */
165     USB_DESC_TYPE_DEVICE,
166     #if (USBD_LPM_ENABLED == 1)
167         0x01,           /*bcdUSB */ changed to USB version 2.01
168         /* in order to support LPM L1 suspend
169         resume test of USBCV3.0*/
170     #else
171         0x00,           /*bcdUSB */
172     #endif /* (USBD_LPM_ENABLED == 1) */
173     0x02,
174     0x00,           /*bDeviceClass*/
175     0x00,           /*bDeviceSubClass*/
176     0x00,
177     USB_MAX_EP0_SIZE,
178     LOBYTE(USBD_VID),
179     HIBYTE(USBD_VID),
180
181     /*bDescriptorType*/
182     #if (USBD_LPM_ENABLED == 1)
183         0x01,           /*bcdUSB */
184     #endif /* (USBD_LPM_ENABLED == 1) */
185     0x00,
186     0x00,           /*bDeviceProtocol*/
187     0x00,
188     0x00,           /*bMaxPacketSize*/
189     /*idVendor*/
190     /*idVendor*/

```

Auto-Complete

A screenshot of the Code Editor showing a file named `main.c`. The cursor is at the end of the word `HAL_GPIO_`, and a dropdown menu shows several completion suggestions: `HAL_GPIO_DelInit`, `HAL_GPIO_EXTI_Callback`, `HAL_GPIO_EXTI_IRQHandler`, `HAL_GPIO_Init`, `HAL_GPIO_LockPin`, `HAL_GPIO_ReadPin`, `HAL_GPIO_TogglePin`, `HAL_GPIO_WritePin`, and `#HAL_GPIO_MODULE_ENABLED`. The suggestion `HAL_GPIO_Init` is highlighted.

```

97  /* USER CODE BEGIN 2 */
98  HAL_GPIO_
99  /* Infinite loop */
100 /* USER CODE END 2 */
101 while (1)
102 {
103     /* Infinite loop */
104     HAL_GPIO_Init(GPIOx, GPIO_InitStructDef * GPIO_InitStruct);
105     /* Infinite loop */
106     HAL_GPIO_LockPin(GPIOx, uint16_t GPIO_Pin);
107     /* Infinite loop */
108     HAL_GPIO_ReadPin(GPIOx, uint16_t GPIO_Pin);
109     /* Infinite loop */
110     HAL_GPIO_TogglePin(GPIOx, uint16_t GPIO_Pin);
111     /* Infinite loop */
112     HAL_GPIO_WritePin(GPIOx, uint16_t GPIO_Pin, GPIO_PinState);
113     /* @brief Software tick
114     * @retval None

```

Syntax Highlight

A screenshot of the Code Editor showing a file named `SystemClock_Config(void)`. The code is highlighted with various colors: grey for comments, light blue for strings, and different shades of green and yellow for code blocks. The entire file is highlighted in red, indicating it is inactive or commented out.

```

116 void SystemClock_Config(void)
117 {
118     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
119     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
120     RCC_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};
121
122     /* Configure the main internal regulator output voltage */
123
124     __HAL_RCC_PWR_CLK_ENABLE();
125     __HAL_RCC_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE3);
126
127     /* Initializes the CPU, AHB and APB busses clocks */
128     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI|RCC_OSCILLATOR_TYPE_NONE;
129     RCC_OscInitStruct.HSEState = RCC_HSE_ON;
130     RCC_OscInitStruct.HSISState = RCC_HSI_ON;
131     RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
132     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
133     RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
134     RCC_OscInitStruct.PLL.PLLM = 15;
135     RCC_OscInitStruct.PLL.PLLN = 144;

```

File Diff/Compare

A screenshot of the Code Editor showing a comparison between two files: `F746-DISCO-TEST/Src/main.c` and `F746-DISCO-TEST-2/Src/main.c`. The interface includes a tree view of changes and a detailed viewer showing the differences in the code. The code is highlighted with various colors: grey for comments, light blue for strings, and different shades of green and yellow for code blocks.

Block Select

A screenshot of the Code Editor showing a file with code blocks selected. Two specific blocks of code are highlighted with red boxes and connected by a pink arrow. The first block is located between `/* USER CODE BEGIN 2 */` and `/* USER CODE END 2 */`, and the second block is located between `/* USER CODE BEGIN WHILE */` and `/* USER CODE END WHILE */`.

```

/* USER CODE BEGIN 2 */
HAL_GPIO_Toggle(GPIOA, GPIO_PIN1);
HAL_GPIO_Toggle(GPIOA, GPIO_PIN2);
HAL_GPIO_Toggle(GPIOA, GPIO_PIN3);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
/* USER CODE BEGIN 2 */
HAL_GPIO_Toggle(GPIOA, GPIO_PIN1);
HAL_GPIO_Toggle(GPIOA, GPIO_PIN2);
HAL_GPIO_Toggle(GPIOA, GPIO_PIN3);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)

```

Code Style

A screenshot of the Code Editor showing the preferences dialog for code style. The "Formatter" tab is selected, showing a dropdown menu for active profiles. The "K&R [built-in]" profile is selected. The dialog also includes sections for "Configure Project Specific Settings..." and "Editor".

Active profile: K&R [built-in]

Configure Project Specific Settings...

Editor

Build Tools



Build Analyzer

Memory Regions						
Region	Start address	End address	Size	Free	Used	Usage (%)
FLASH	0x08000000	0x08200000	2048 KB	2043.22 KB	4.78 KB	0.23%
RAM	0x20000000	0x20080000	512 KB	510.45 KB	1.55 KB	0.30%
ITCMRAM	0x00000000	0x00004000	16 KB	15.48 KB	528 B	3.22%

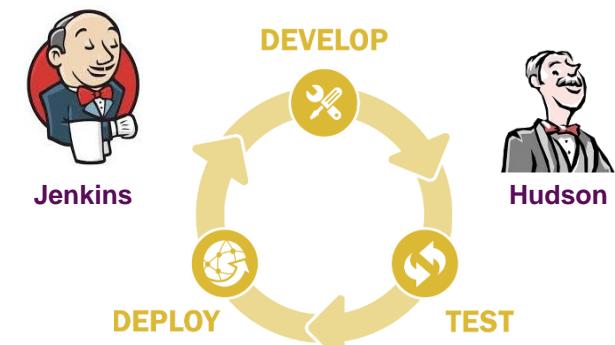
Search			
Name	Run address (VMA)	Load address (LMA)	Size
ITCMRAM	0x00000000		16 KB
> .itcmram	0x00000000	0x0800110c	528 B
ITCMRAM	0x00000000		2048 KB
> .itcmram	0x00000000	0x0800110c	528 B
FLASH	0x08000000		2048 KB
> .itcmram	0x08000000	0x0800110c	528 B
> .isr_vector	0x08000000	0x08000000	60 B
> .text	0x08000000	0x08000000	4.16 KB
> .rodata	0x080010e0	0x080010e0	16 B
> .ARM	0x080010f0	0x080010f0	8 B
> .preinit_array	0x080010f8	0x080010f8	0 B
> .init_array	0x080010f8	0x080010f8	4 B
> .fini_array	0x080010fc	0x080010fc	4 B
> .data	0x20000000	0x08001100	12 B
RAM	0x20000000		512 KB
> .data	0x20000000	0x08001100	12 B
> .bss	0x20000000		32 B
> _user_heap_stack	0x20000000		1.5 KB

Static Stack Analyzer

List				
Function	Depth	Type	Location	Info
SystemClock_Config	88	STATIC	main.c:108	
NVIC_EncodePriority	40	STATIC	core_cm7.h:2071	
HAL_RCC_GetSysClockFreq	40	STATIC	stm32f7xx_hal_rcc.c:982	
HAL_NVIC_SetPriority	32	STATIC	stm32f7xx_hal_cortex.c:165	
HAL_NVIC_OscConfig	32	STATIC	stm32f7xx_hal_rcc.c:344	
_NVIC_SetPriorityGrouping	24	STATIC	core_cm7.h:1865	
HAL_RCC_ClockConfig	24	STATIC	stm32f7xx_hal_rcc.c:703	
HAL_InitTick	16	STATIC	stm32f7xx_hal.c:231	

List					
Function	Depth	Max cost	Local cost	Type	Location
LoopFillZeroBoss	7	208	0		stm32f7xx_it.c:182
SysTick_Handler	0	8	8	STATIC	stm32f7xx_it.c:70
NMI_Handler	0	4	4	STATIC	stm32f7xx_it.c:128
UsageFault_Handler	0	4	4	STATIC	stm32f7xx_it.c:160
PendSV_Handler	0	4	4	STATIC	stm32f7xx_it.c:168
HardFault_Handler	0	4	4	STATIC	stm32f7xx_it.c:83
HAL_IncTick	0	4	4	STATIC	stm32f7xx_hal.c:290
SVC_Handler	0	4	4	STATIC	stm32f7xx_it.c:143
DebugMon_Handler	0	4	4	STATIC	stm32f7xx_it.c:156
MemManage_Handler	0	4	4	STATIC	stm32f7xx_it.c:98
BusFault_Handler	0	4	4	STATIC	stm32f7xx_it.c:113
Reset_Handler	0	0	0		
main	0	0	0		

Headless Build



• Continuous Integration

Build project without opening IDE

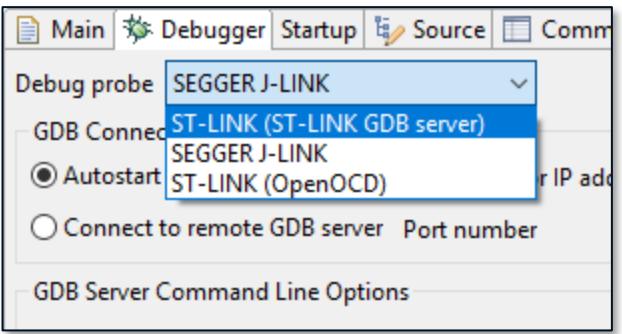
No GUI shown but build system becomes active

• Supported for makefile and managed projects



Debug

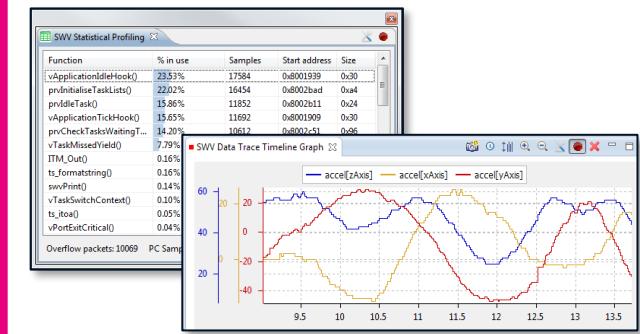
Debugger



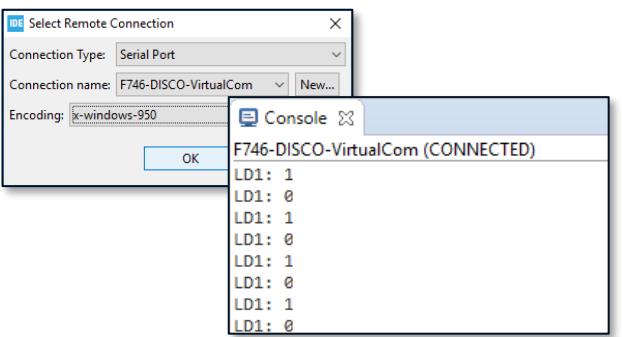
Live expressions

Expression	Type	Value
(x)= uwTick	volatile uint32_t	1603
Add new expression		

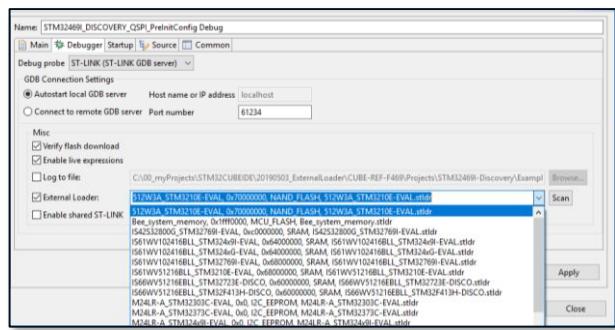
SWV



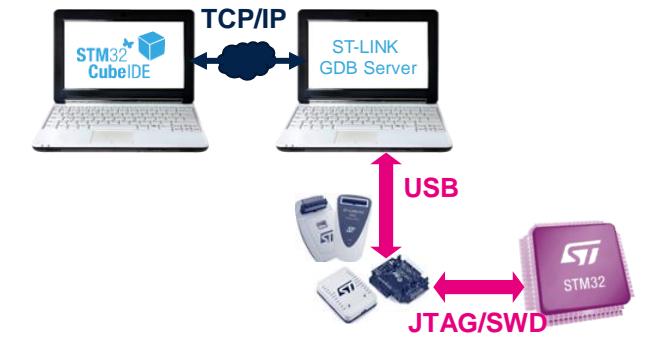
Integrated UART Terminal



External Loader

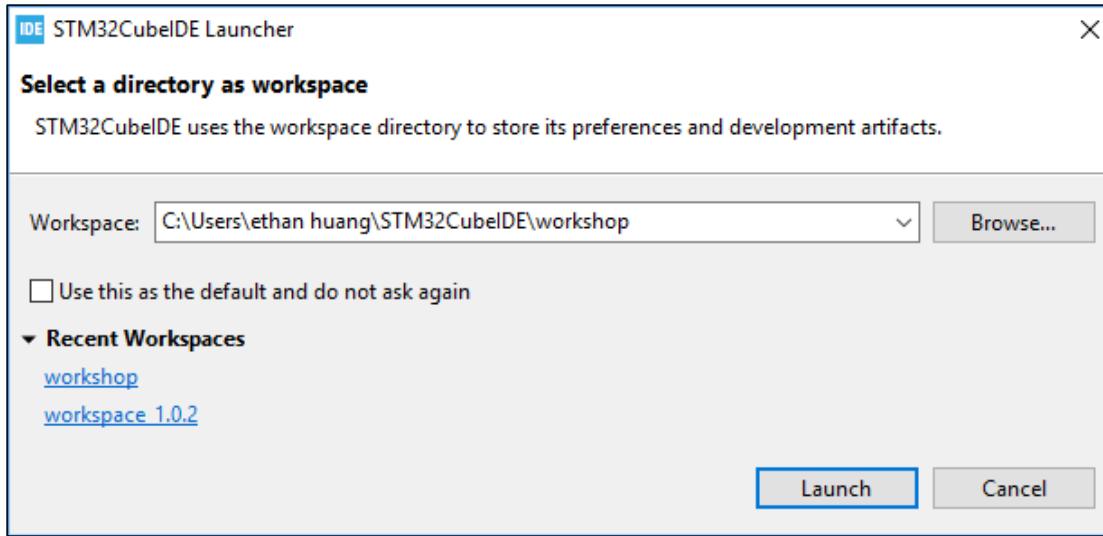


Remote Debug



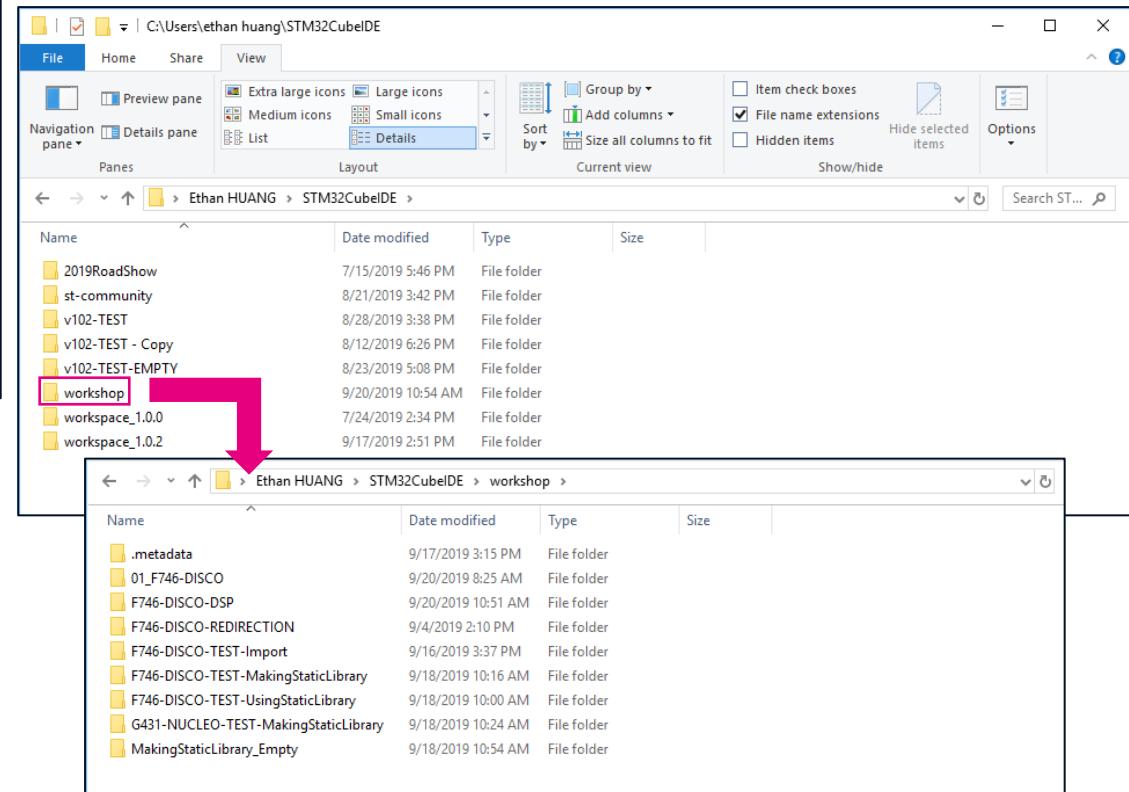
關於STM32CubeIDE

Workspace – Workspace的內容



- Workspace是Eclipse當中用來存放專案目錄或是專案目錄參照的地方
- Workspace可存放多個專案
- Workspace也是用來儲存操作環境設定的地方 (.metadata)

Workspace的預設目錄為：
C:\Users\XXX huang\STM32CubeIDE\



Workspace - Workspace與Project

Workspace Folder 1

Option 1



Workspace Folder 2



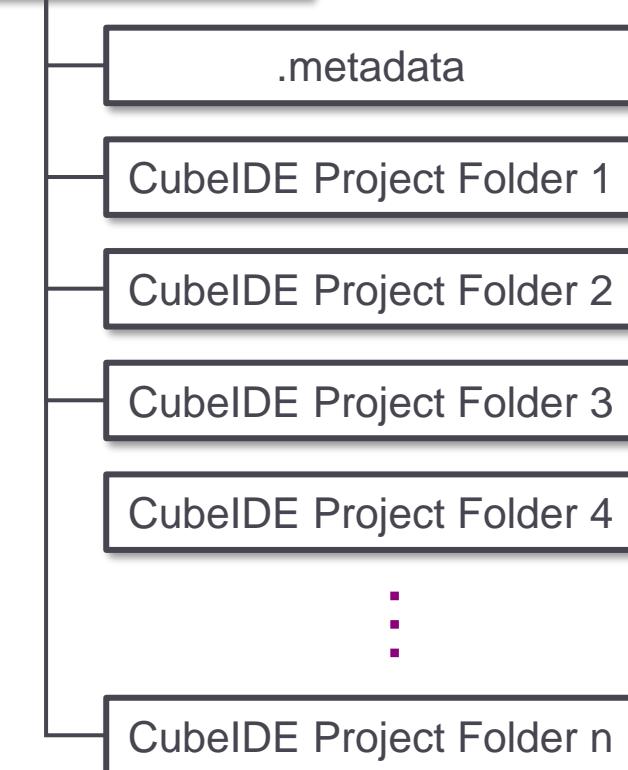
Workspace Folder 3



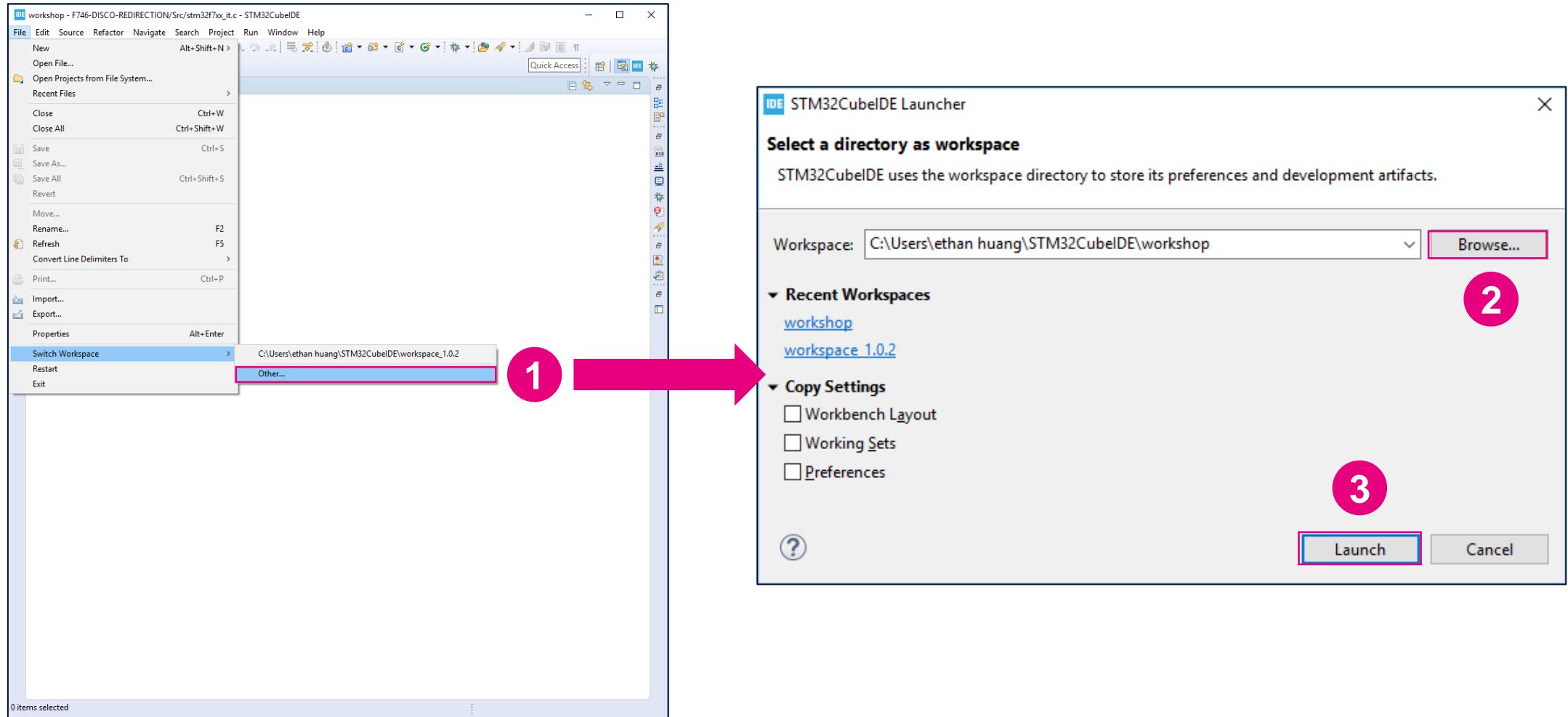
Option 2

建議使用

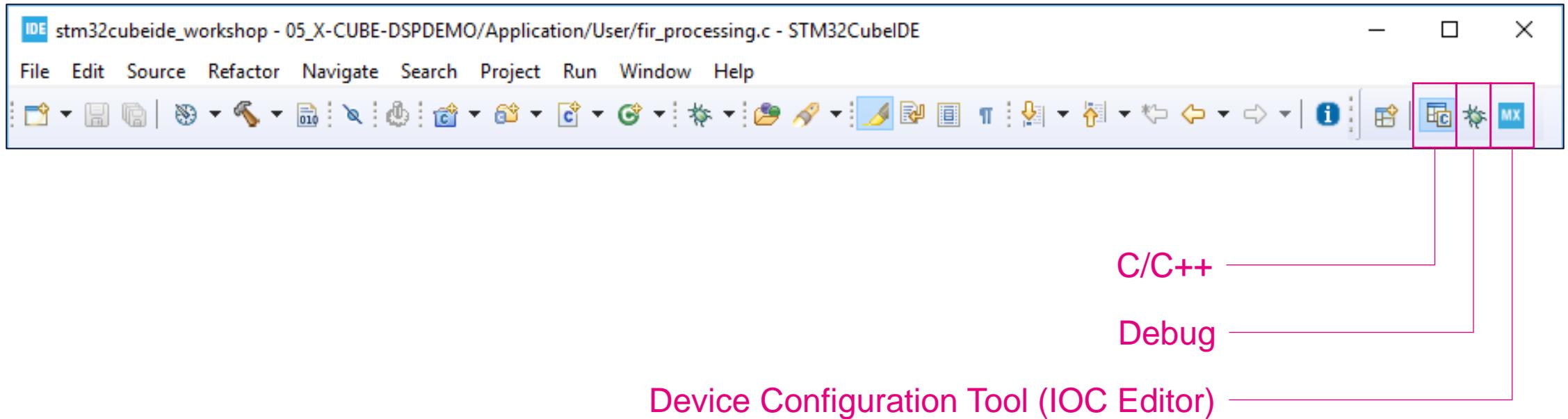
Workspace Folder 1



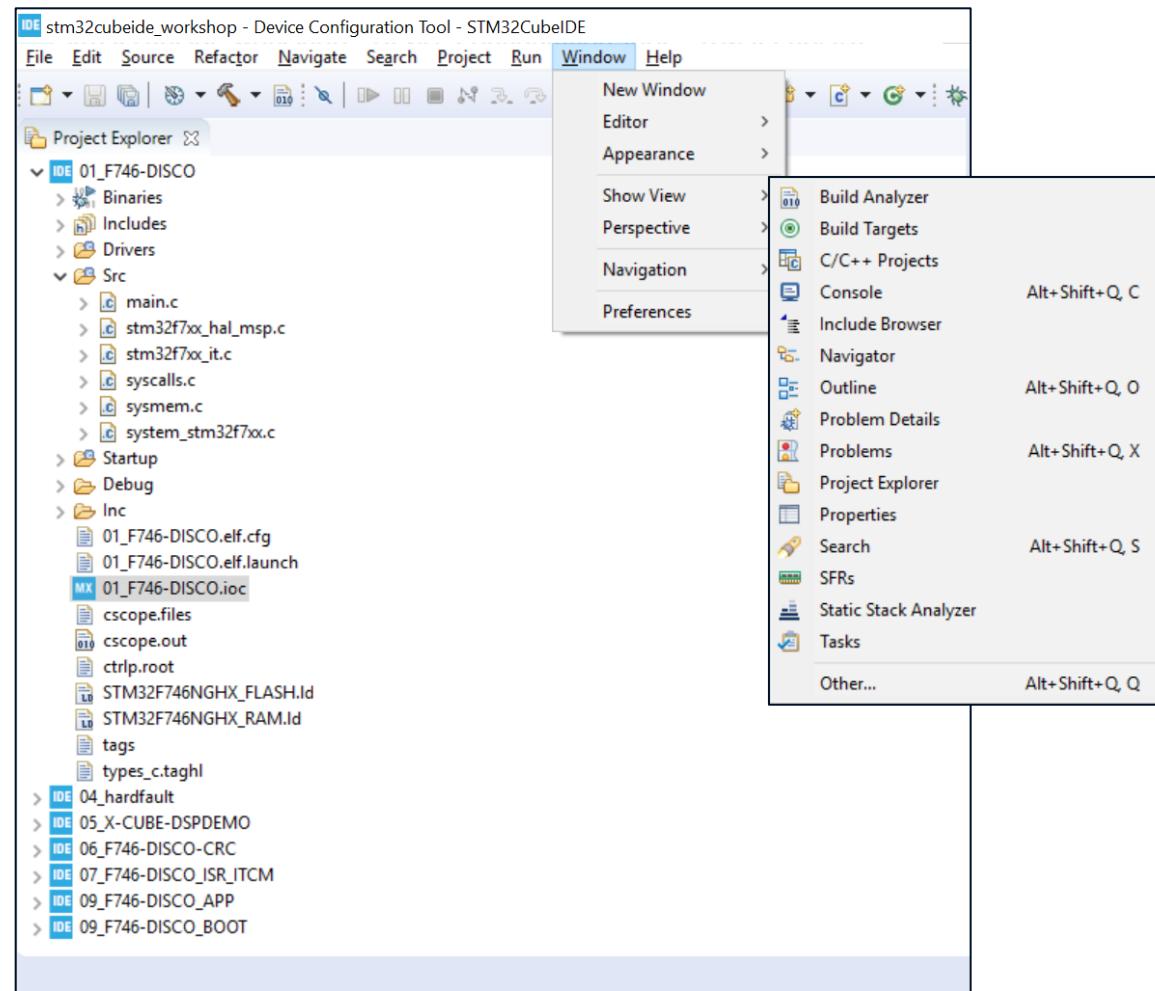
Workspace – Workspace的切換



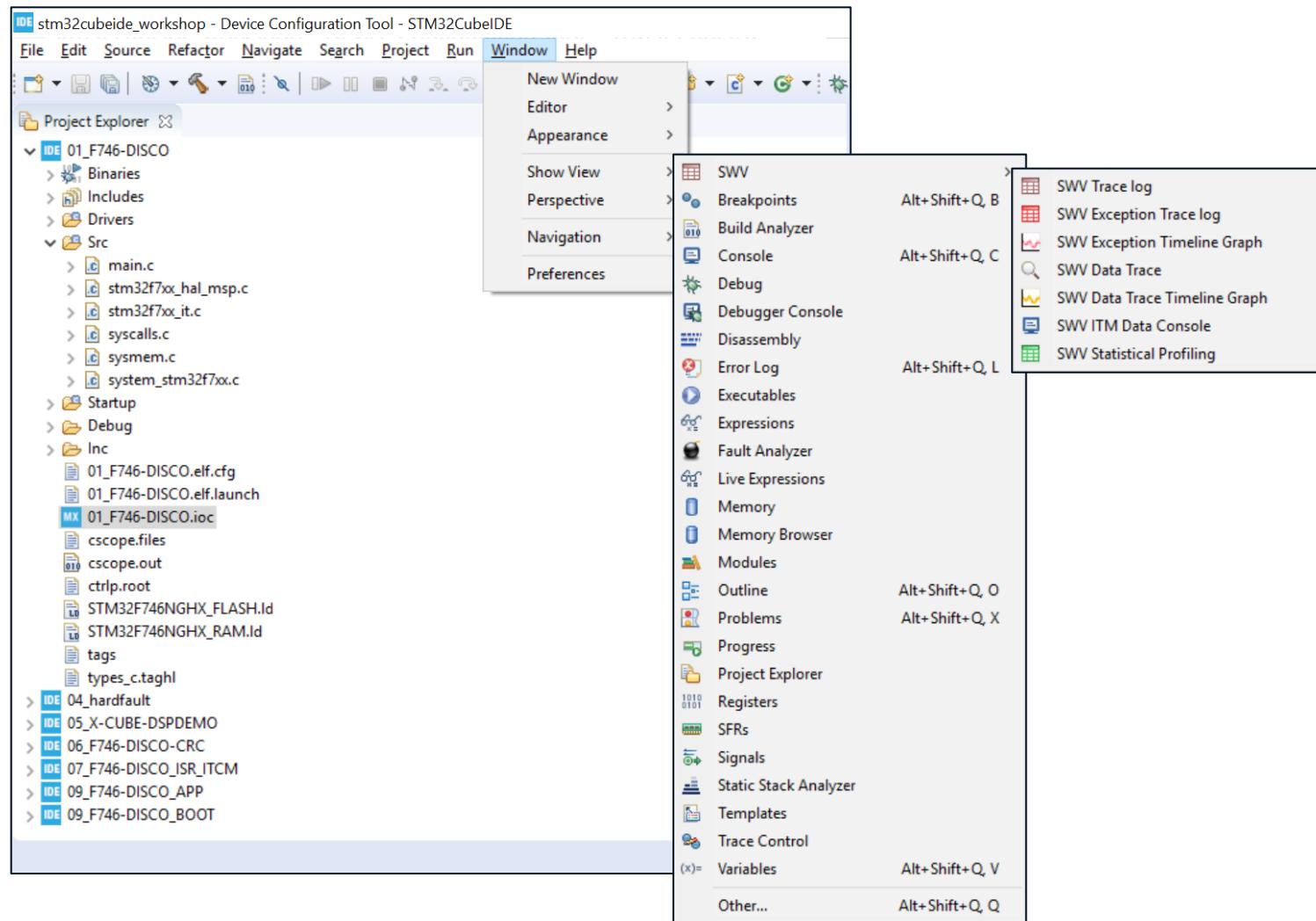
Perspectives (視角)



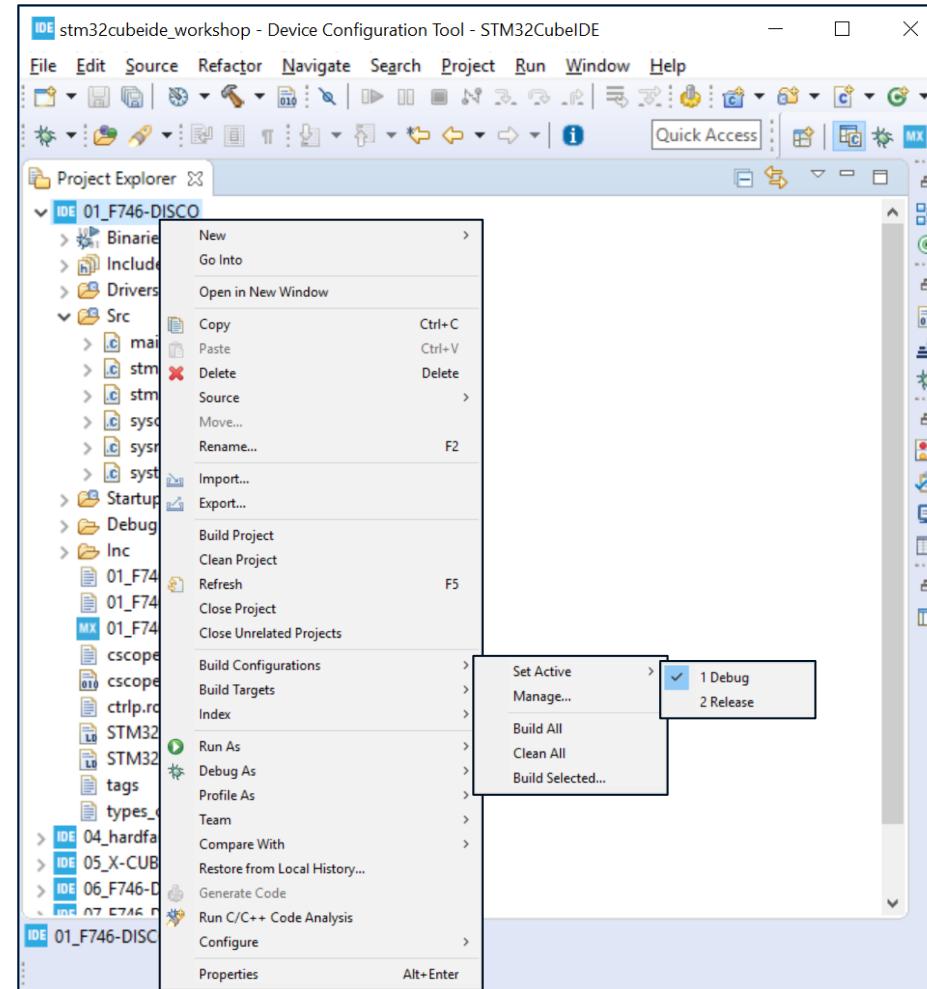
Perspectives (視角) Views in C/C++ Perspective



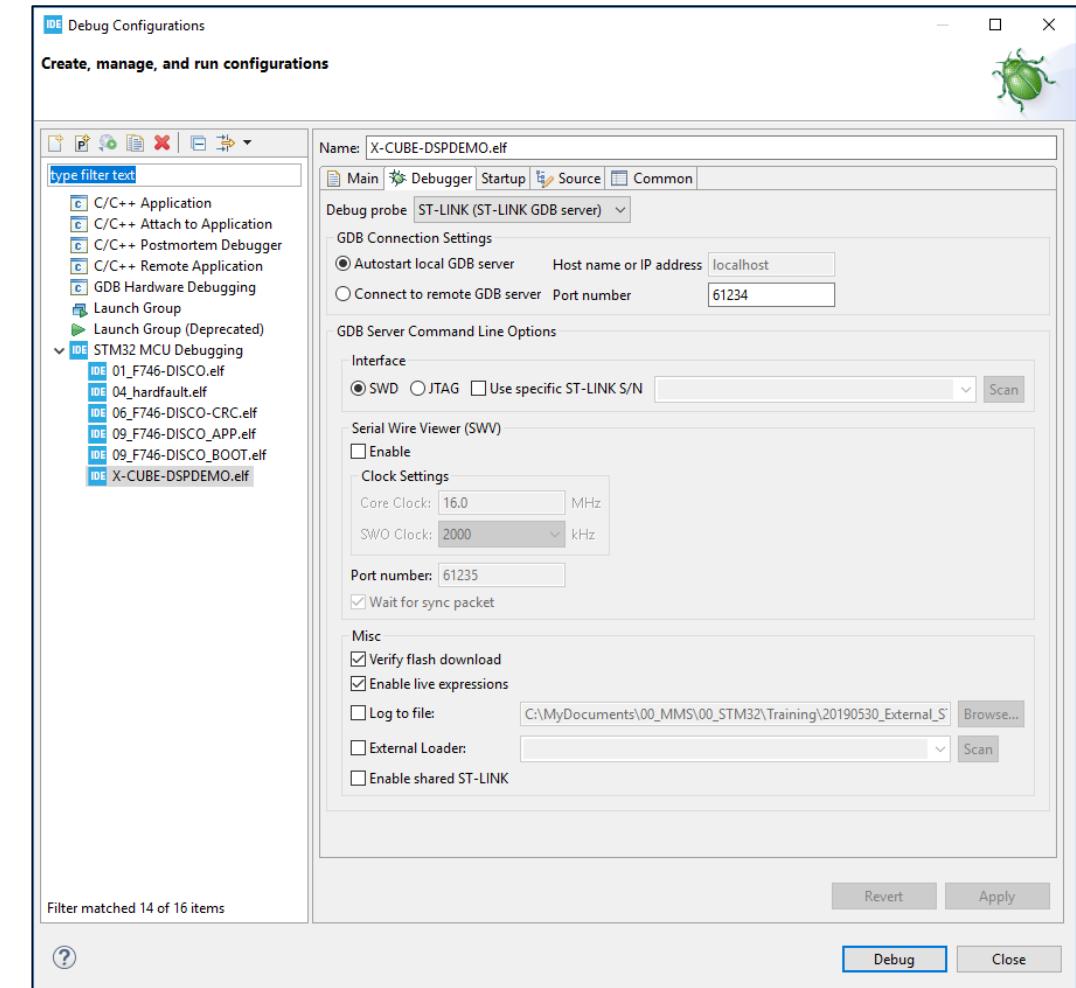
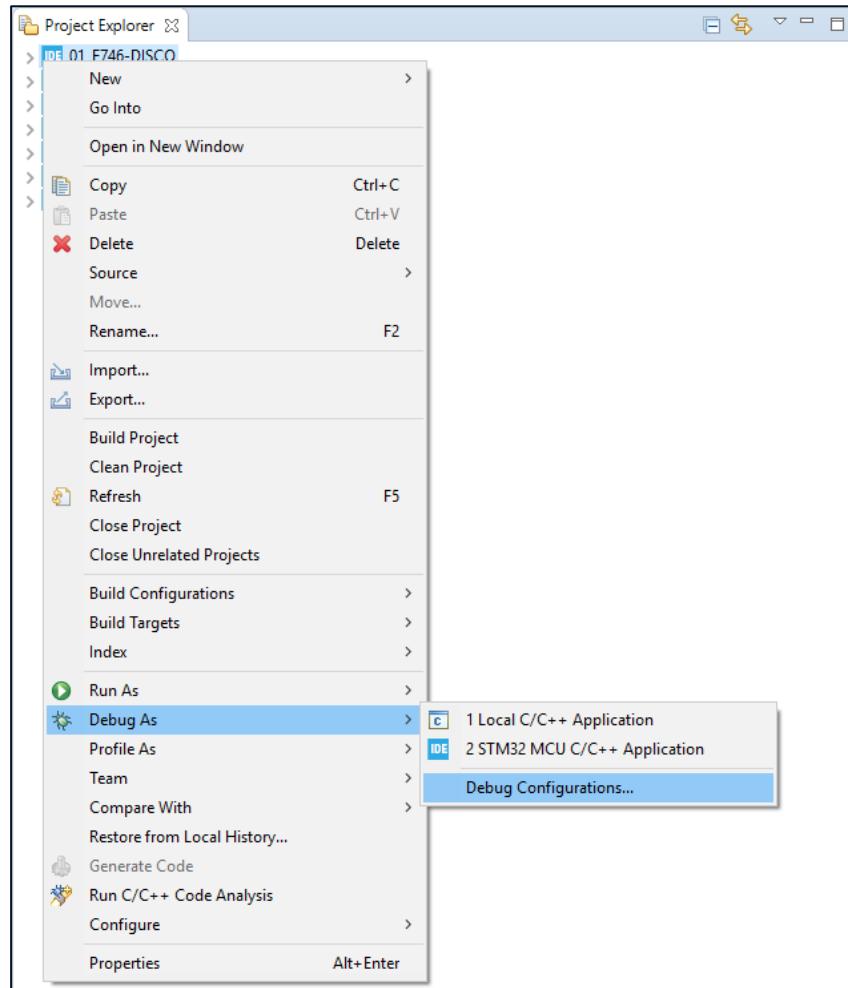
Perspectives (視角) Views in Debug Perspective



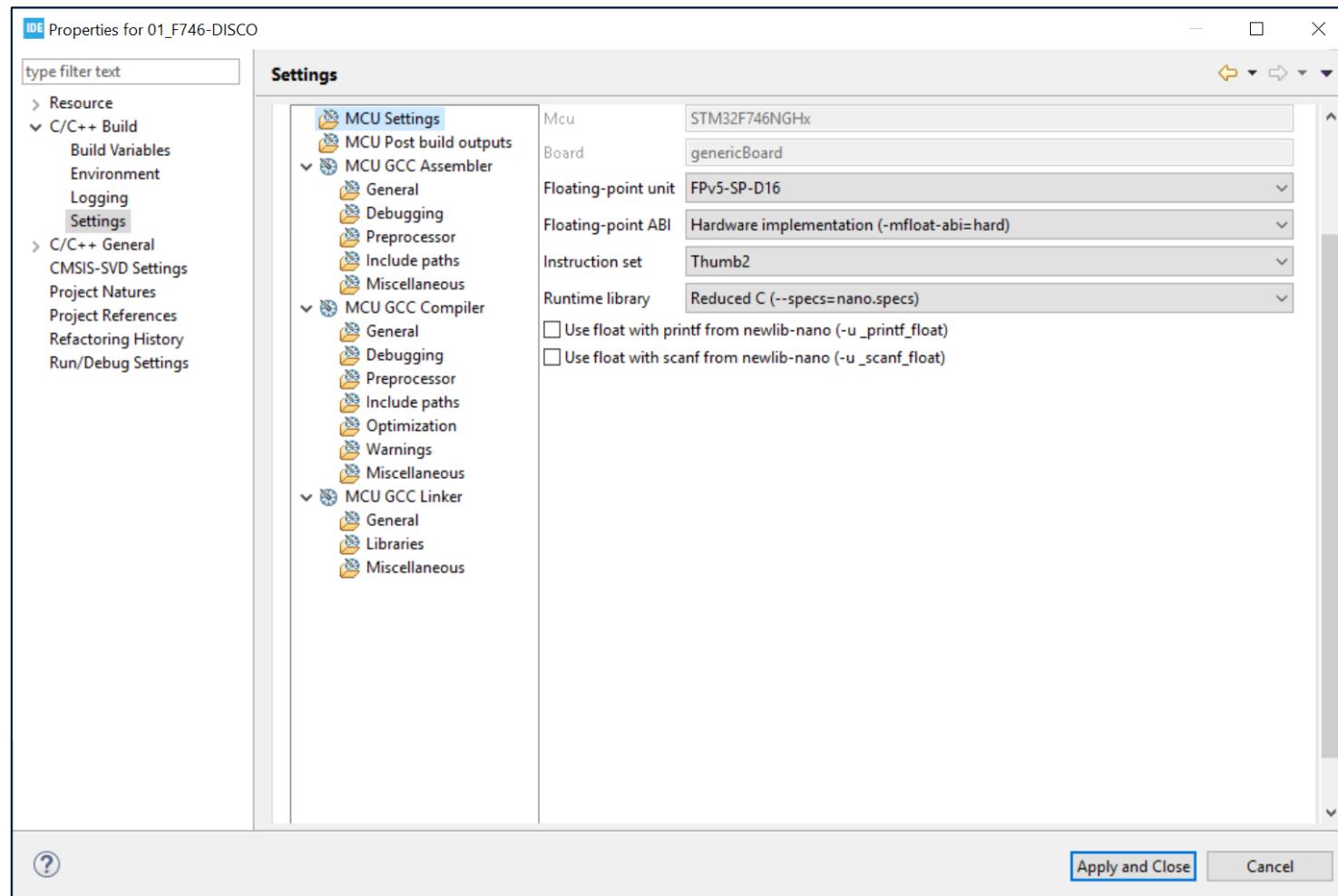
Configuration – Build Configuration



Configuration – Debug Configuration



C/C++ Build Settings



Known Issues and Limitations

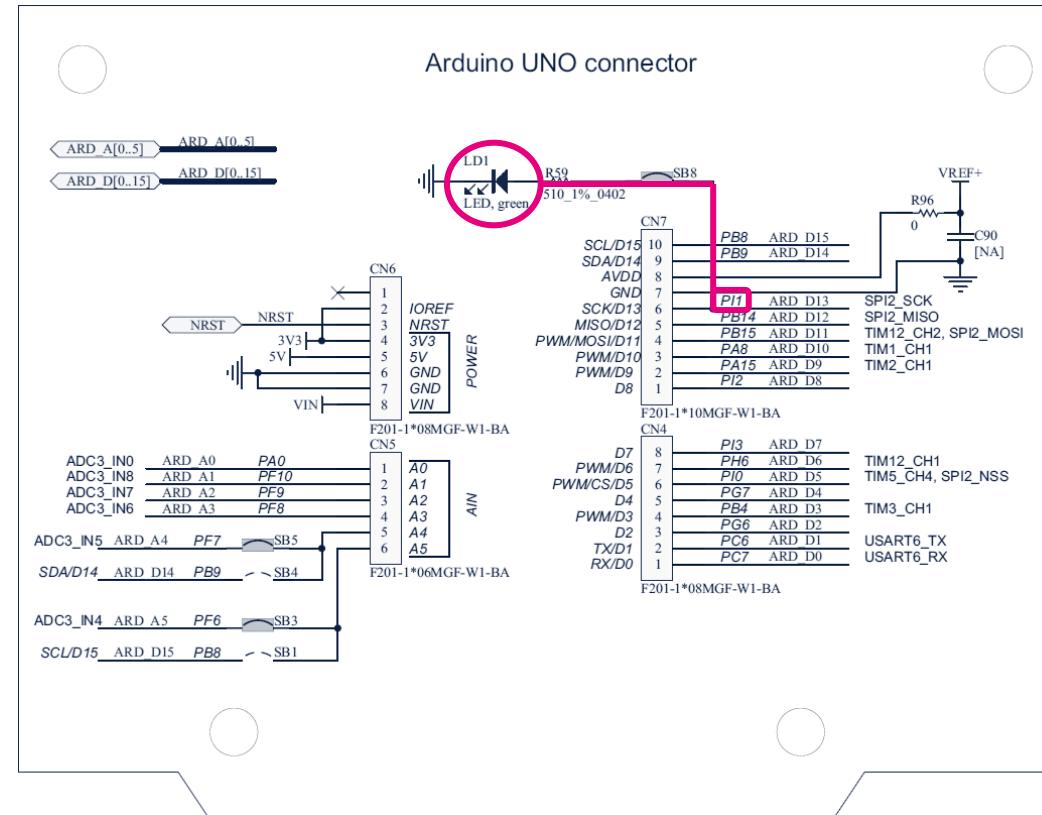
- The [Run] button is not yet implemented and is therefore hidden from the toolbar menu and Run menu.
- The creation of static libraries with the option add libraries as reference leads to the unintended generation of the Drivers folder.
- The USE_HAL_DRIVER symbol is not properly removed when switching from HAL to LL driver, which can cause build failure.
- Creating BOARD projects with Code generator options == add necessary library files as reference.. and Initialise all peripherals with default settings can cause build errors if BOARD depends on the USB library.
- The user cannot change [Application Structure] from Basic to Advanced or vice versa without losing user code.
- Importing the ioc file created by stand-alone STM32CubeMX is not fully supported.
- Editor hyperlinks sometimes jump to declaration instead of definition.
- It is not possible to open an SW4STM32 or TrueSTUDIO® workspace with STM32CubeIDE. Refer to Migration guide from TrueSTUDIO® to STM32CubeIDE (UM2578) and Migration guide from System Workbench to STM32CubeIDE (UM2579).
- The [Help]>[Data refresher] can be invoked several times without pop-up dialog.
- Some Linux® installers install a few packages before the license agreement has been accepted.
- The macOS® installer displays incompatible version dialog when installing the stlink-server package. This can safely be ignored.
- Some STM32CubeMX pop-up dialogs are not opened in front of the STM32CubeIDE workbench on all OS.
- On macOS®, the ioc editor does not show the Peripheral categories frame on the left side by default.
- STM32CubeIDE does not support switching from one MCU to another once the project is created.
- Anti-virus tools may be sensitive to STM32CubeIDE and bundled exe files despite the addition of vendor certificates since it is not yet deployed widely.
- STM32CubeIDE SWV selecting large amounts of data to copy to the clipboard may crash STM32CubeIDE.
- Conditional breakpoints do not work with OpenOCD.
- The project importer for SW4STM32 cannot import all settings in projects from very old versions (older than 2.0).
- The Generate code operation re-includes excluded files in folders that are created by STM32CubeMX
- Having a space or non-ascii character in the project/workspace path or installation path is not fully supported
- Updating field [HCLK] in tab Clock Configuration is difficult
- When creating an empty project, the FPU is enabled in the build settings but the project is generated without a SystemInit function to initialize the FPU at run-time.
- In the STM32CubeMX .ioc editor under [Project Manager]>[Code Generator], there is a [settings] button that is not yet implemented.
- Importing a project from an earlier version into the current one will hang STM32CubeIDE when opening the .ioc editor. This only affects macOS®.
- Some radio or check buttons in the debugger tab have unexpected rendering on any Ubuntu® 14.04.

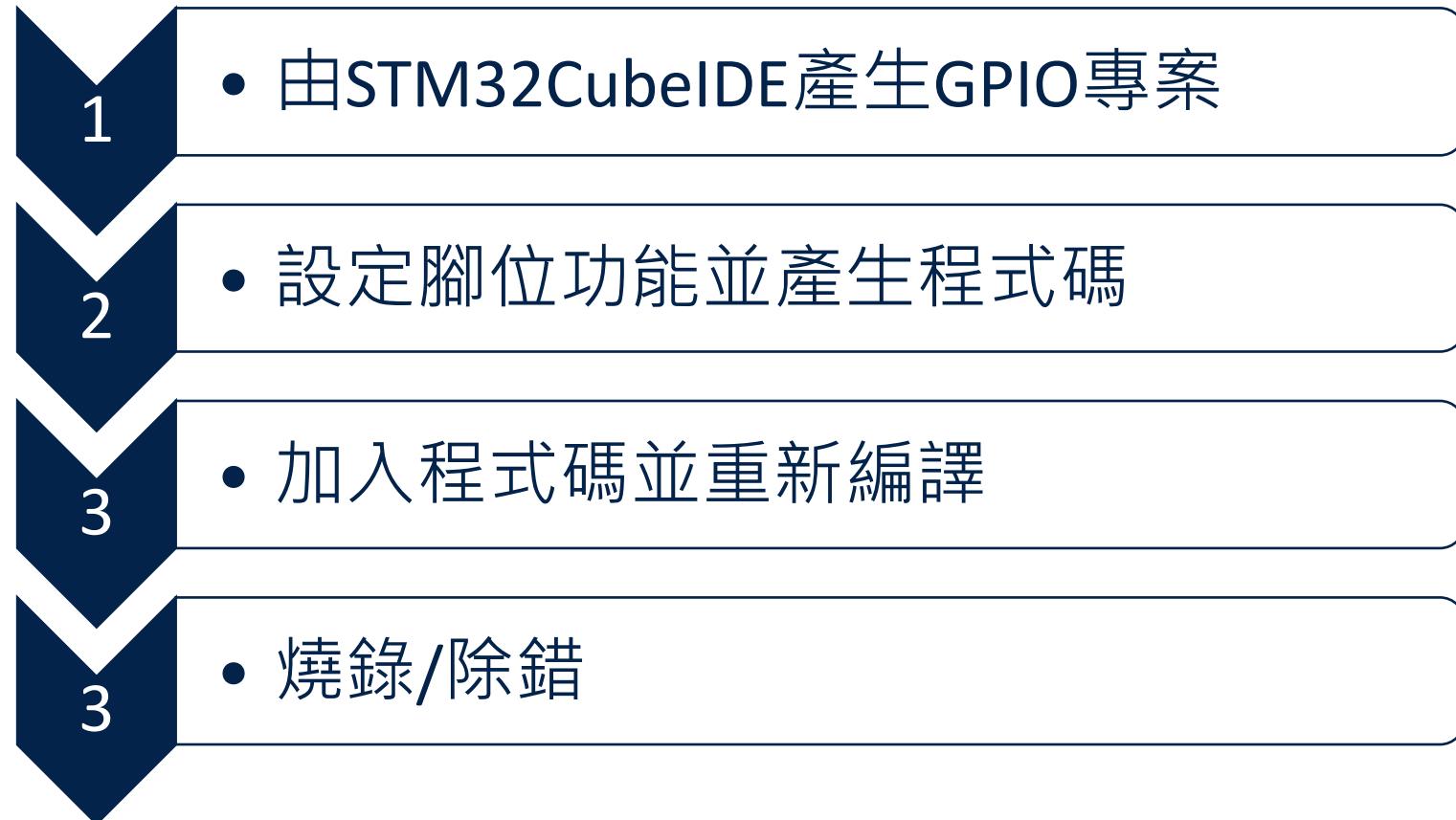
Hands-on

01_GPIO_Output

實作目標

每500ms改變LD1閃滅狀態





Step 1: 產生專案

The image displays two windows from the STM32 Project IDE.

Left Window: Target Selection

This window allows selecting an STM32 target. It includes:

- Target Selection:** Select STM32 target
- MCU MPU Selector:** Board Selector (selected) and Cross Selector
- Board Filters:** Part Number Search (32F746), Vendor, Type, MCU MPU Series, Other (Price = 54.0, Oscillator Freq. = 25 MHz), Peripheral (Accelerometer, Analog I/O, Arduino Form Factor, Audio Line In, Audio Line Out, Battery, Button, CAN).
- 32F746GDISCOVERY:** STMicroelectronics STM32F746G Discovery kit Board Support and Examples. It shows the STM32 F7 logo, a product status of ACTIVE, Unit Price (US\$): 54.0, and Mounted device: STM32F746NGHx. A description states: "The 32F746GDISCOVERY Discovery kit is a complete demonstration and development platform for STMicroelectronics Arm Cortex-M7-core-based STM32F746NG".
- Boards List:** 1 item (32F746GDISCOVERY, Discovery kit, Active, 54.0, STM32F746NG...).

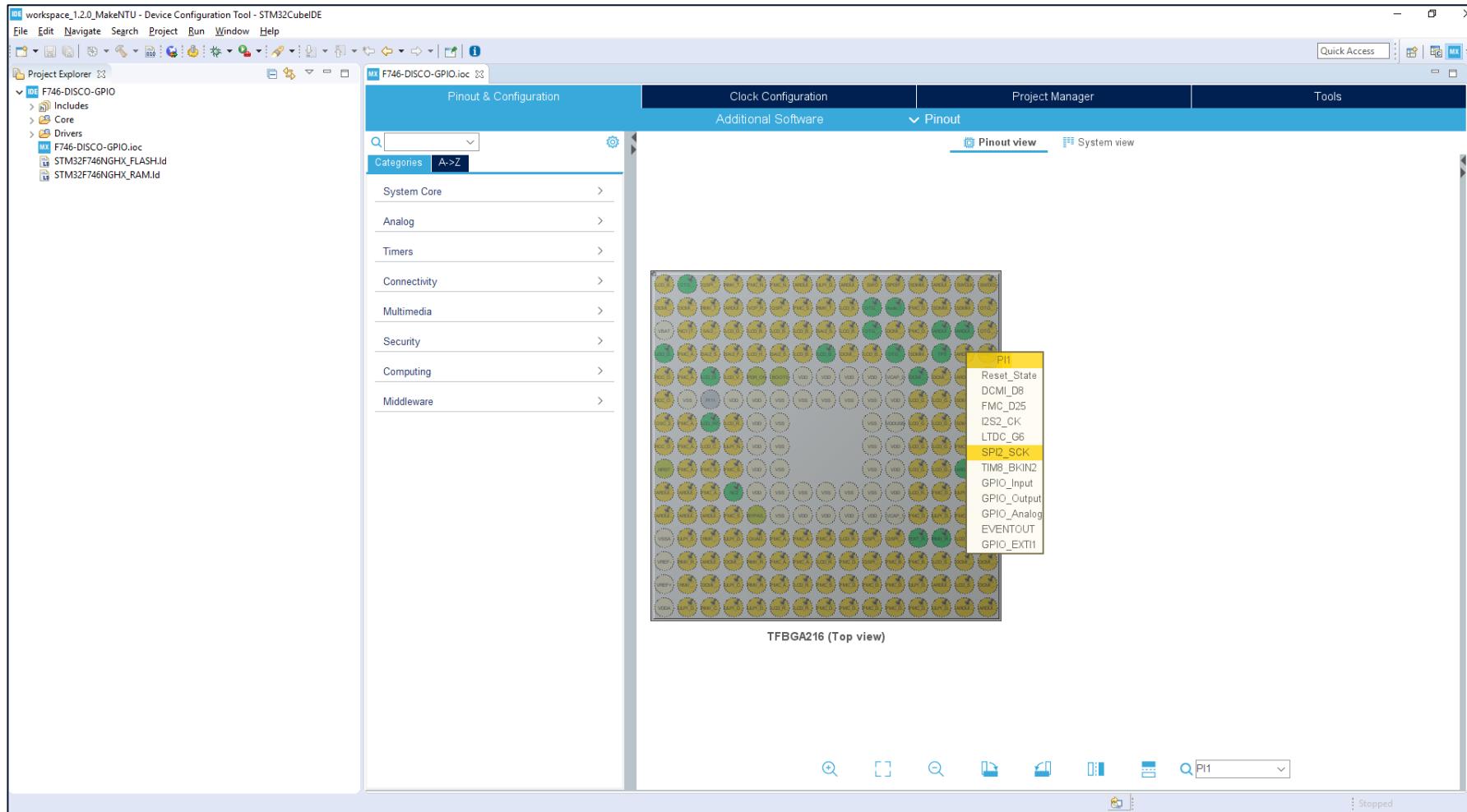
Right Window: Setup STM32 project

This window configures a new project:

- Project:** Project Name: F746-DISCO-GPIO, Use default location (unchecked).
- Location:** D:\xx_MakeNTU\Hands-on-Examples\F746-DISCO-GPIO (Browse... button).
- Options:**
 - Targeted Language:** C (radio button selected).
 - Targeted Binary Type:** Executable (radio button selected).
 - Targeted Project Type:** STM32Cube (radio button selected).

Buttons at the bottom include: ? (Help), < Back, Next >, Finish (highlighted in blue), and Cancel.

Step 2: 腳位設定



Step 3: 編譯

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** workspace_1.2.0_MakeNTU - F746-DISCO-GPIO/Core/src/main.c - STM32CubeIDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Includes, Project, Build, Run, etc.
- Project Explorer:** Shows the project structure:
 - F746-DISCO-GPIO (selected)
 - Includes
 - Core
 - Inc
 - Src
 - main.c
 - stm32f7xx_hal_msp.c
 - stm32f7xx_it.c
 - syscalls.c
 - sysmem.c
 - system_stm32f7xx.c
 - Startup
 - Drivers
 - F746-DISCO-GPIO.ioc
 - STM32F746NGHX_FLASH.Id
 - STM32F746NGHX_RAM.Id
- Code Editor:** The main window displays the content of `main.c`. The code is heavily annotated with multi-line comments explaining the structure of the user code. It includes sections for header files, private includes, typedefs, PTDs, defines, macros, and function prototypes. The annotations are as follows:

```
1 /* USER CODE BEGIN Header */
2 /**
3  * @file    : main.c
4  * @brief   : Main program body
5  *
6  * @attention
7  *
8  * <h2><center>&copy; Copyright (c) 2020 STMicroelectronics.
9  * All rights reserved.</center></h2>
10 *
11 *
12 * This software component is licensed by ST under BSD 3-Clause license,
13 * the "License"; You may not use this file except in compliance with the
14 * License. You may obtain a copy of the License at:
15 *          opensource.org/licenses/BSD-3-Clause
16 *
17 */
18 /* USER CODE END Header */
19
20 /* Includes -----*/
21 #include "main.h"
22
23 /* Private includes -----*/
24 /* USER CODE BEGIN Includes */
25
26 /* USER CODE END Includes */
27
28 /* Private typedef -----*/
29 /* USER CODE BEGIN PTD */
30
31 /* USER CODE END PTD */
32
33 /* Private define -----*/
34 /* USER CODE BEGIN PD */
35
36 /* USER CODE END PD */
37
38 /* Private macro -----*/
39 /* USER CODE BEGIN PM */
40
41 /* USER CODE END PM */
42
43 /* Private variables -----*/
44
45 /* USER CODE BEGIN PV */
46
47 /* USER CODE END PV */
48
49 /* Private function prototypes -----*/
50 void SystemClock_Config(void);
51 static void MX_GPIO_Init(void);
52 /* USER CODE BEGIN PFP */
53
54 /* USER CODE END PFP */
55
56 /* Private user code -----*/
57 /* USER CODE BEGIN 0 */
58
59 /* USER CODE END 0 */
60
```

workspace_1.2.0_MakeNTU - F746-DISCO-GPIO/Core/src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

Console

CDT Build Console [F746-DISCO-GPIO]

```
16:53:08 **** Build of configuration Debug for project F746-DISCO-GPIO ****
make: jst all
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal cortex.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal dma.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal dma ex.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal exti.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal flash.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal gpio.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal flash ex.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal i2c ex.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal rcc c.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal rcc ex.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc ".../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal tim c.c" -mcpu=cortex-m7 -std=gnu11
arm-none-eabi-gcc -c -x assembler-with-cpp --specs=nano.specs -mfpu=fpv5-sp-d
arm-none-eabi-gcc ".../Core/Src/main.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIVER -DSTM32F74
arm-none-eabi-gcc ".../Core/Src/stm32f7xx_hal_msp.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIVER
arm-none-eabi-gcc ".../Core/Src/stm32f7xx_hal_it.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIVER -DSTM32F74
arm-none-eabi-gcc ".../Core/Src/syscalls.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIVER -DSTM32F74
arm-none-eabi-gcc ".../Core/Src/system.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIVER -DSTM32F74
arm-none-eabi-gcc ".../Core/Src/system_stm32f7xx.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIVER
arm-none-eabi-gcc -o "F746-DISCO-GPIO.elf" "@objects.list" -mcpu=cortex-m7 -T"C:\00_myProjects\F746-DISCO-GPIO.elf"
Finished building target: F746-DISCO-GPIO.elf

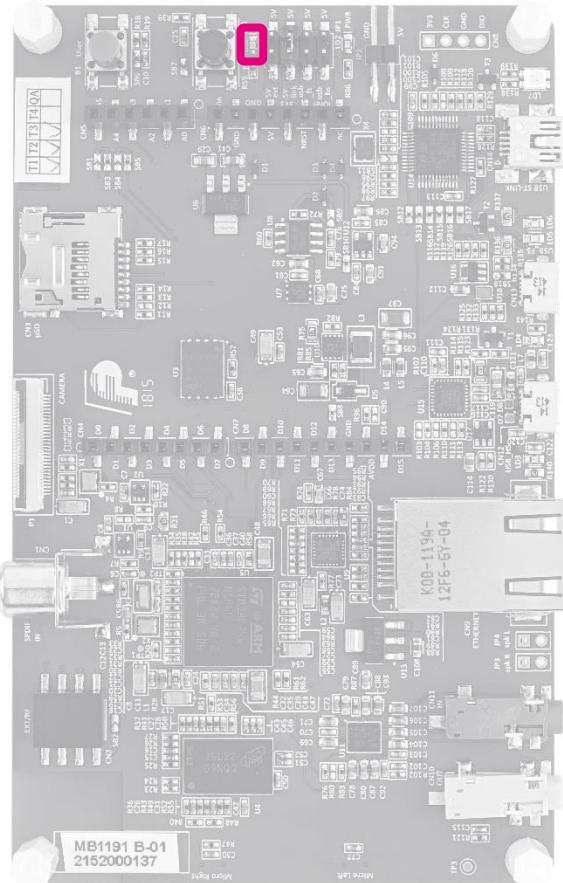
arm-none-eabi-size F746-DISCO-GPIO.elf
arm-none-eabi-objdump -h -S F746-DISCO-GPIO.elf > "F746-DISCO-GPIO.list"
arm-none-eabi-objcopy -O binary F746-DISCO-GPIO.elf "F746-DISCO-GPIO.bin"
    text     data     bss     dec   hex filename
 8100      20    1572    9692    25dc F746-DISCO-GPIO.elf
Finished building: default.size.stdout

Finished building: F746-DISCO-GPIO.bin
Finished building: F746-DISCO-GPIO.list

16:53:13 Build Finished. 0 errors, 0 warnings. (took 9s.551ms)
```

Step 4: 燒錄/除錯

每500ms改變LD1閃滅狀態



CUBE HAL使用建議

- 永遠使用STM32CubeIDE IOC Editor來新增刪減週邊裝置和變更週邊設定

如此可以確保永遠保持程式碼和IOC檔的設定是一致的

- 不要修改變更週邊的驅動程式stm32f7xx_hal_xxx.c

所有在週邊驅動程式所做的修改會在用STM32CUBEMX重新產生程式碼之後被移除

- 將自己的程式碼置於USER CODE區塊當中

若將自己的程式碼置於USER CODE區塊之外，則自己的程式碼會在用STM32CUBEMX重新產生程式碼之後被移除

CUBE HAL使用建議

- 永遠使用STM32CubeIDE IOC Editor來新增刪減週邊裝置和變更週邊設定

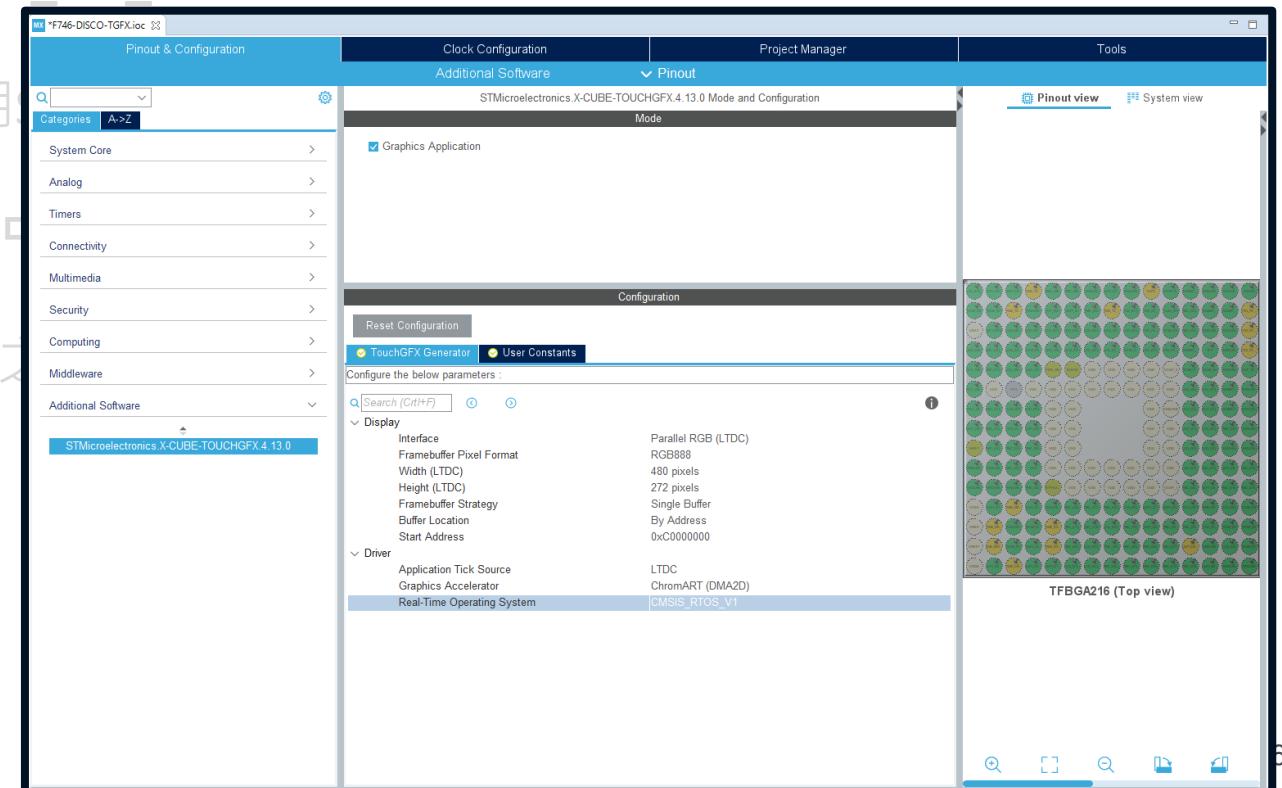
如此可以確保永遠保持程式碼和IOC檔的設定是一致的

- 不要修改變更週邊的驅動程式stm32f7xx_hal_xxx.c

所有在週邊驅動程式所做的修改會在用

- 將自己的程式碼置於USER CODE區塊當中

若將自己的程式碼置於USER CODE區塊之後
新產生程式碼之後被移除



CUBE HAL使用建議

- 永遠使用STM32CubeIDE IOC Editor來新增刪減週邊裝置和變更週邊設定

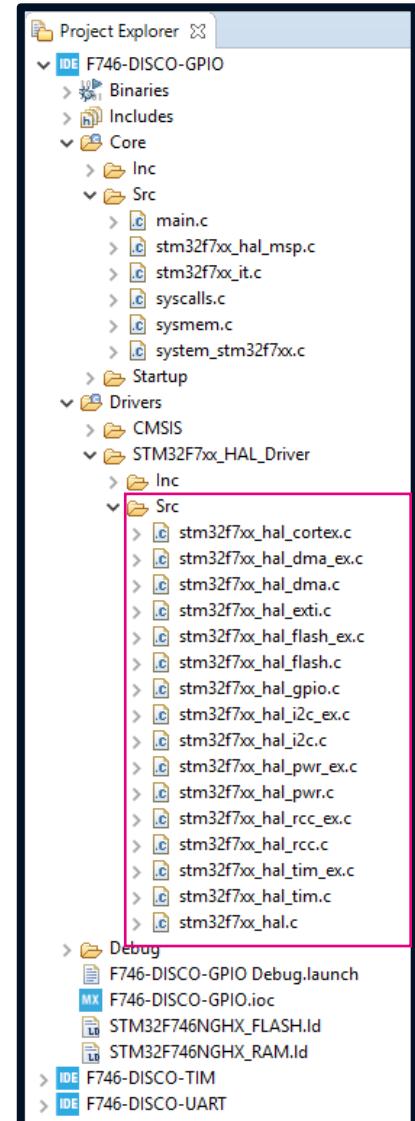
如此可以確保永遠保持程式碼和IOC檔的設定是一致的

- 不要修改變更週邊的驅動程式stm32f7xx_hal_xxx.c

所有在週邊驅動程式所做的修改會在用STM32CubeIDE重新產生程式碼之後被移除

- 將自己的程式碼置於USER CODE區塊當中

若將自己的程式碼置於USER CODE區塊之外，則自己的程式碼會在用STM32CUBEMX重新產生程式碼之後被移除



CUBE HAL使用建議

- 永遠使用STM32CubeIDE IOC Editor來新增刪減週邊裝置和變更週邊設定

如此可以確保永遠保持程式碼和IOC檔的設定是一致的

- 不要修改變更週邊的驅動程式stm32f7xx_hal_xxx.c

所有在週邊驅動程式所做的修改會在用STM32CUBEMX重新產生程式碼之後被移除

- 將自己的程式碼置於USER CODE區塊當中

若將自己的程式碼置於USER CODE區塊之外，則自己的程式碼會在用STM32CUBEMX重新產生程式碼之後被移除

```
int main(void)
{
    /* USER CODE BEGIN 1 */

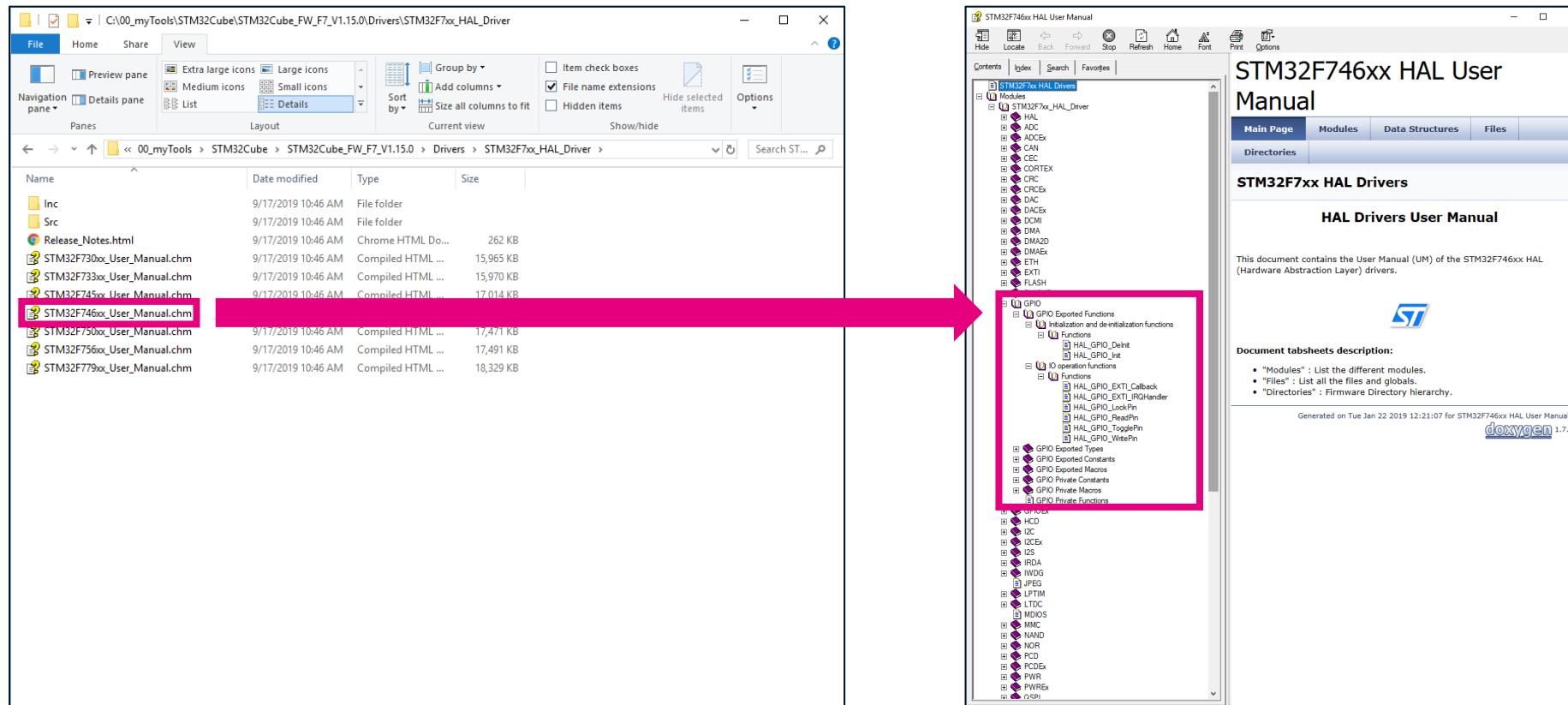
    /* USER CODE END 1 */
    ...

    /* USER CODE BEGIN Init */
    /* USER CODE END Init */
    ...
    /* USER CODE BEGIN SysInit */
    /* USER CODE END SysInit */
    ...
    /* USER CODE BEGIN 2 */
    /* USER CODE END 2 */
    ...
    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        /* USER CODE BEGIN 3 */
        HAL_Delay(500);
        HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
    }
    /* USER CODE END 3 */
}
```

補充：如何查找HAL API

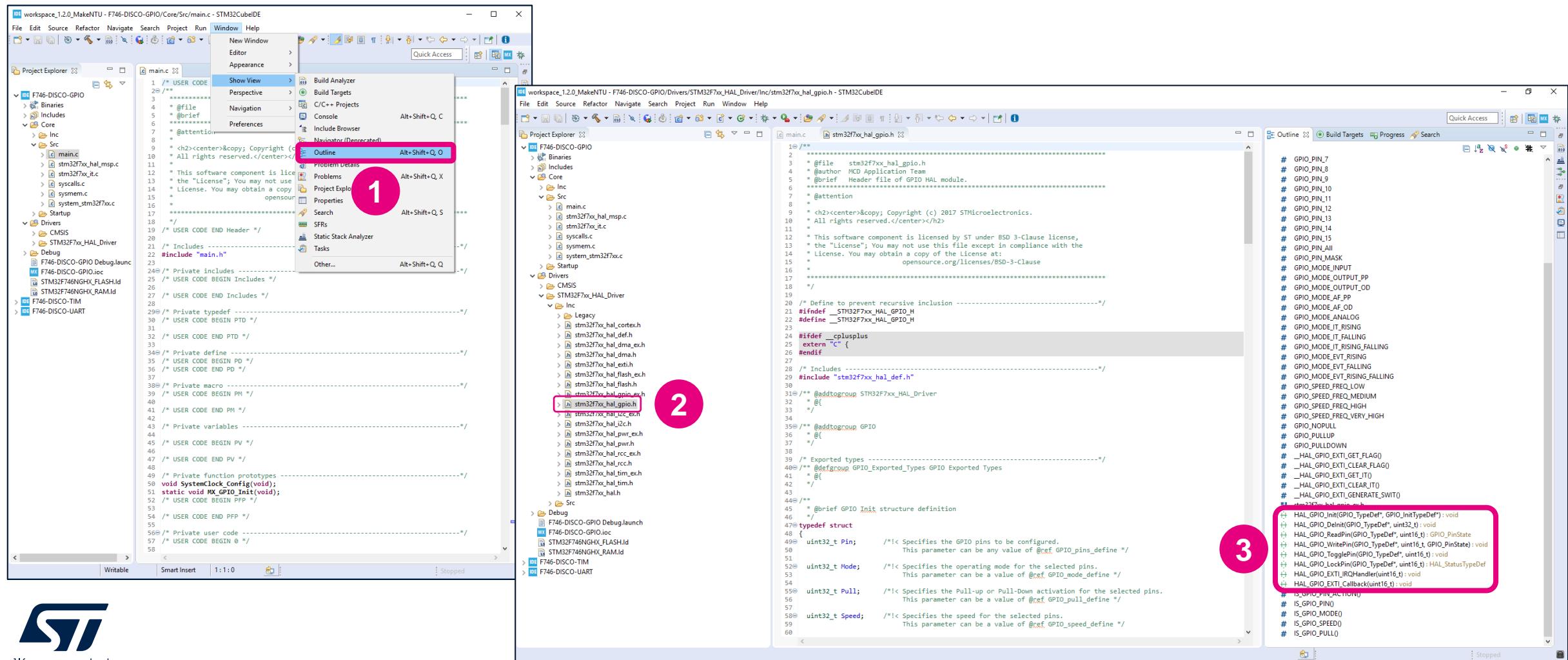
搜尋Doxygen文件 (以搜尋GPIO HAL API為例)

Doxygen文件置於目錄: STM32F7 MCU Package Repository\Drivers\STM32F7xx_HAL_Driver



補充：如何查找HAL API

從STM32CubeIDE的Outline視窗(以搜尋GPIO HAL API為例)



STM32CubeIDE基本功能介紹



功能體驗 – Auto-Complete (1)

欲輸入HAL_Delay()

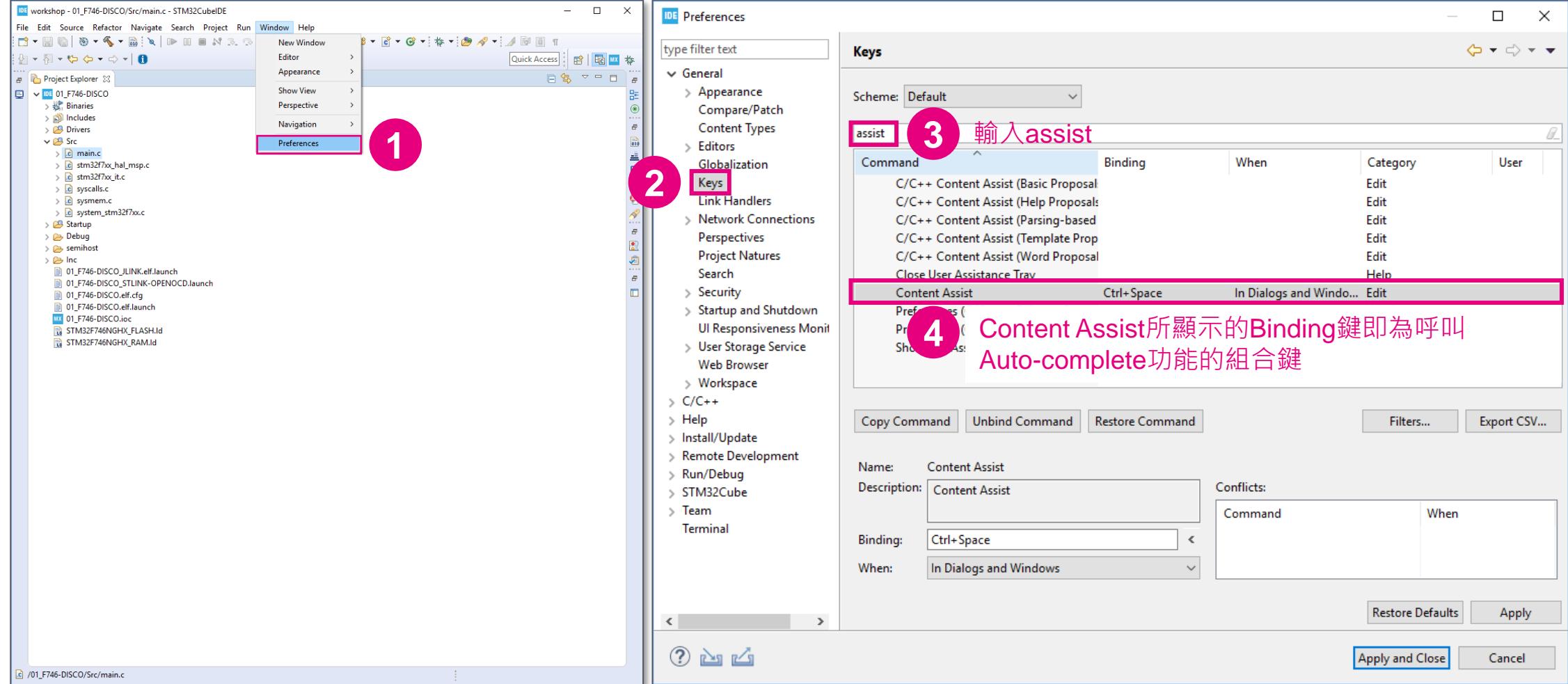
The screenshot shows the STM32CubeIDE interface with the main.c file open in the editor. A tooltip is displayed at the cursor position, showing the text "Press 'Ctrl+Space' to show Template Proposals". The tooltip also lists two template proposals:

- HAL_Delay(uint32_t Delay) : void
- HAL_TIM_OC_DelayElapsedCallback(TIM_HandleTypeDef * htim) : void

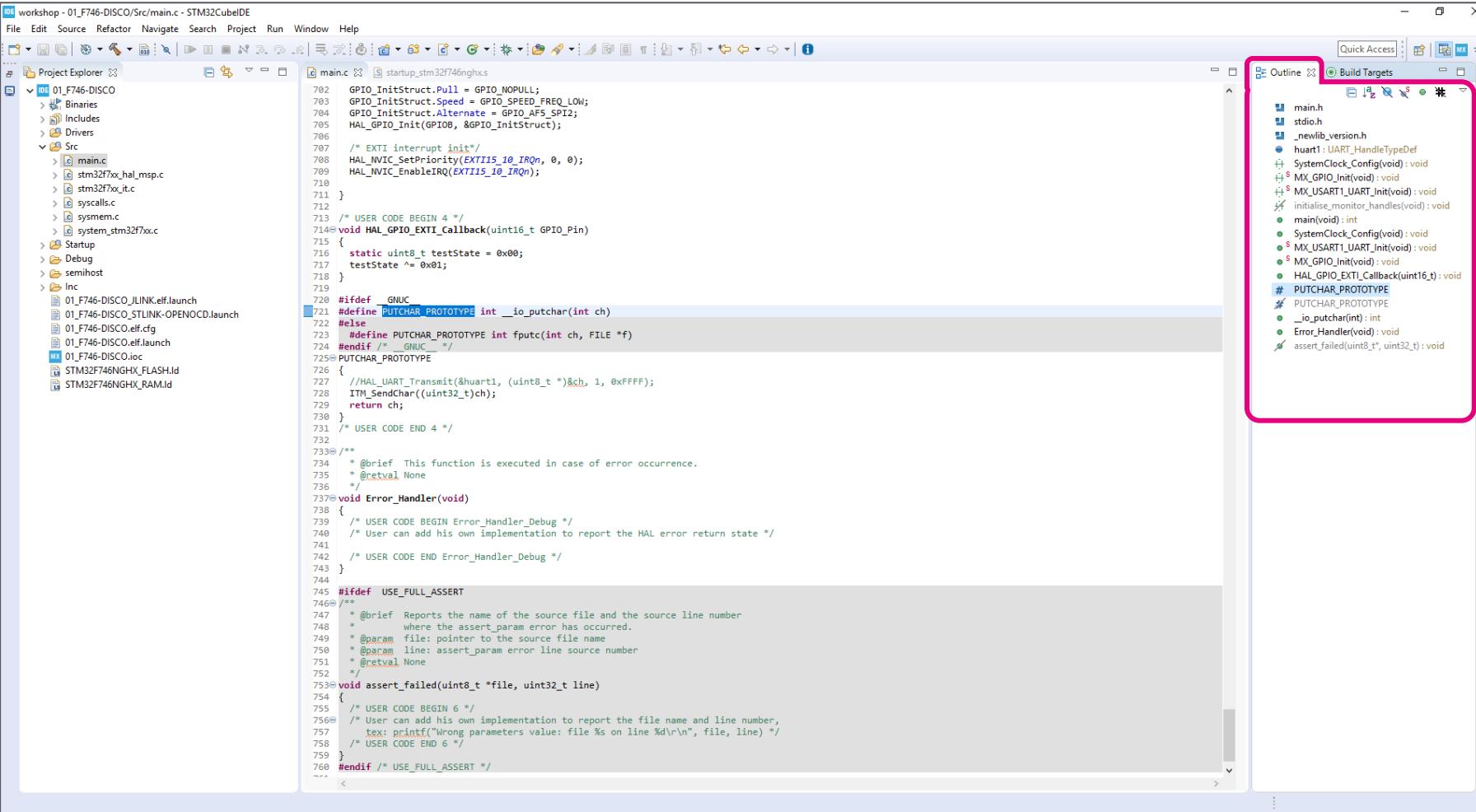
A pink circle labeled "1" points to the cursor position where "HAL Del" is typed. A pink circle labeled "2" points to a button containing "Ctrl + Space" which is highlighted with a pink border.

```
103 printf("TEST: pi = %f\n\r", 3.1415926);
104 /* USER CODE END 2 */
105
106 /* Infinite loop */
107 /* USER CODE BEGIN WHILE */
108 while (1)
109 {
110     /* USER CODE END WHILE */
111     /* USER CODE BEGIN */
112 }
113
114 /**
115  * @brief S
116  */
117
118 /**
119  * @brief S
120  * @retval
121  */
122 void SystemC
123 {
124     RCC_OscInit
125     RCC_ClkInit
126     RCC_Periph
127
128     /**
129      */
130     __HAL_RCC_PWR_CLK_ENABLE();
131     __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
132     /**
133      */
134     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
135     RCC_OscInitStruct.HSISState = RCC_HSI_ON;
136     RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
```

功能體驗 – Auto-Complete (2) 組合鍵查詢



功能體驗 – Outline View



The screenshot shows the STM32CubeIDE interface with the 'Outline' view panel highlighted by a red border. The main window displays the 'main.c' file from the '01_F746-DISCO' project. The 'Outline' view lists all the symbols defined in the current file, including function prototypes and variable declarations.

```
main.h
stdio.h
_newlib_version.h
uart1: UART_HandleTypeDef
SystemClock_Config(void) : void
MX_GPIO_Init(void) : void
MX_USART1_UART_Init(void) : void
initialise_monitor_handles(void) : void
main(void) : int
SystemClock_Config(void) : void
MX_USART1_UART_Init(void) : void
MX_GPIO_Init(void) : void
HAL_GPIO_EXTI_Callback(uint16_t) : void
PUTCHAR_PROTOTYPE
PUTCHAR_PROTOTYPE
_i_o_putchar(int ch)
_i_o_putchar(int ch, FILE *f)
_error_handler(void)
assert_failed(uint8_t *, uint32_t) : void
```

The code editor shows the implementation of the `main.c` file, which includes GPIO initialization, NVIC setup, and USART1 configuration. It also defines prototypes for `PUTCHAR_PROTOTYPE`, `_i_o_putchar`, and `Error_Handler`.

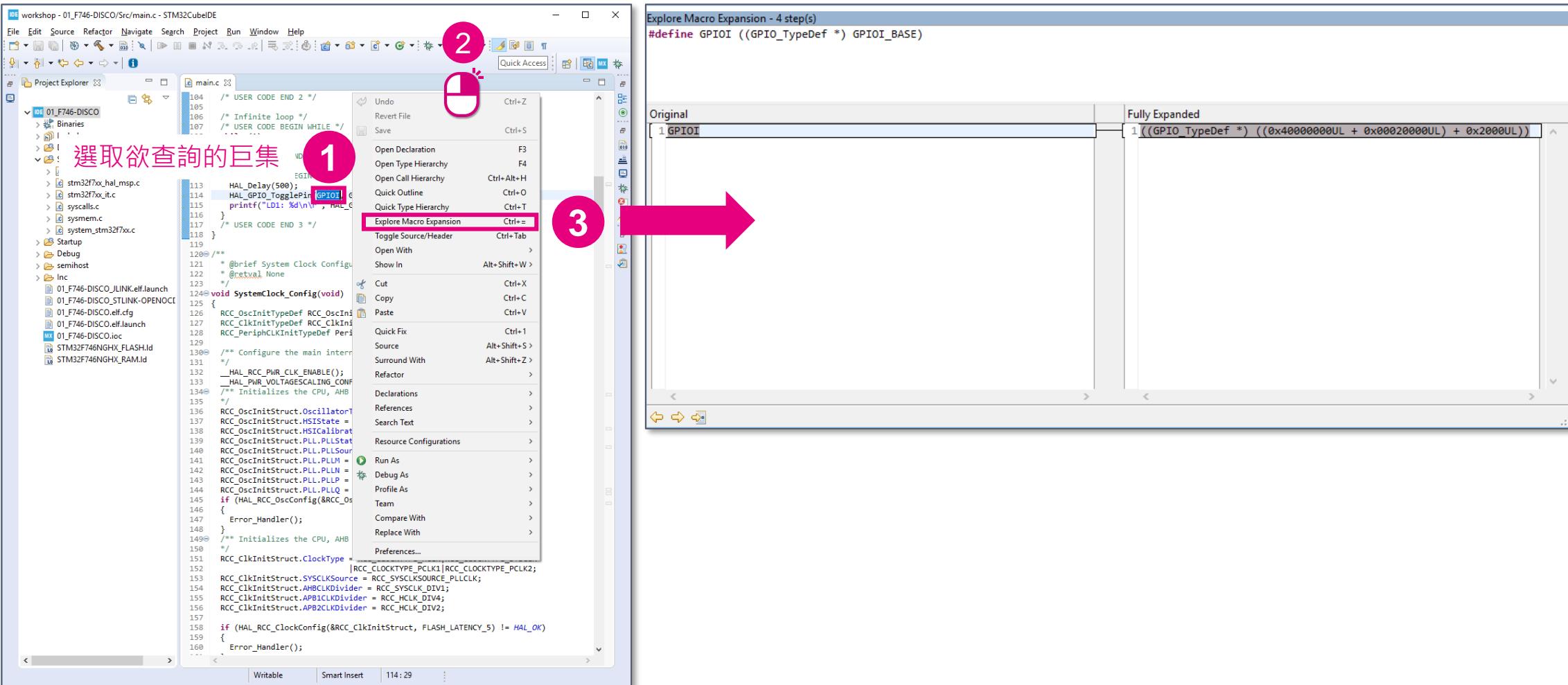
功能體驗 – 巨集的展開 (1)



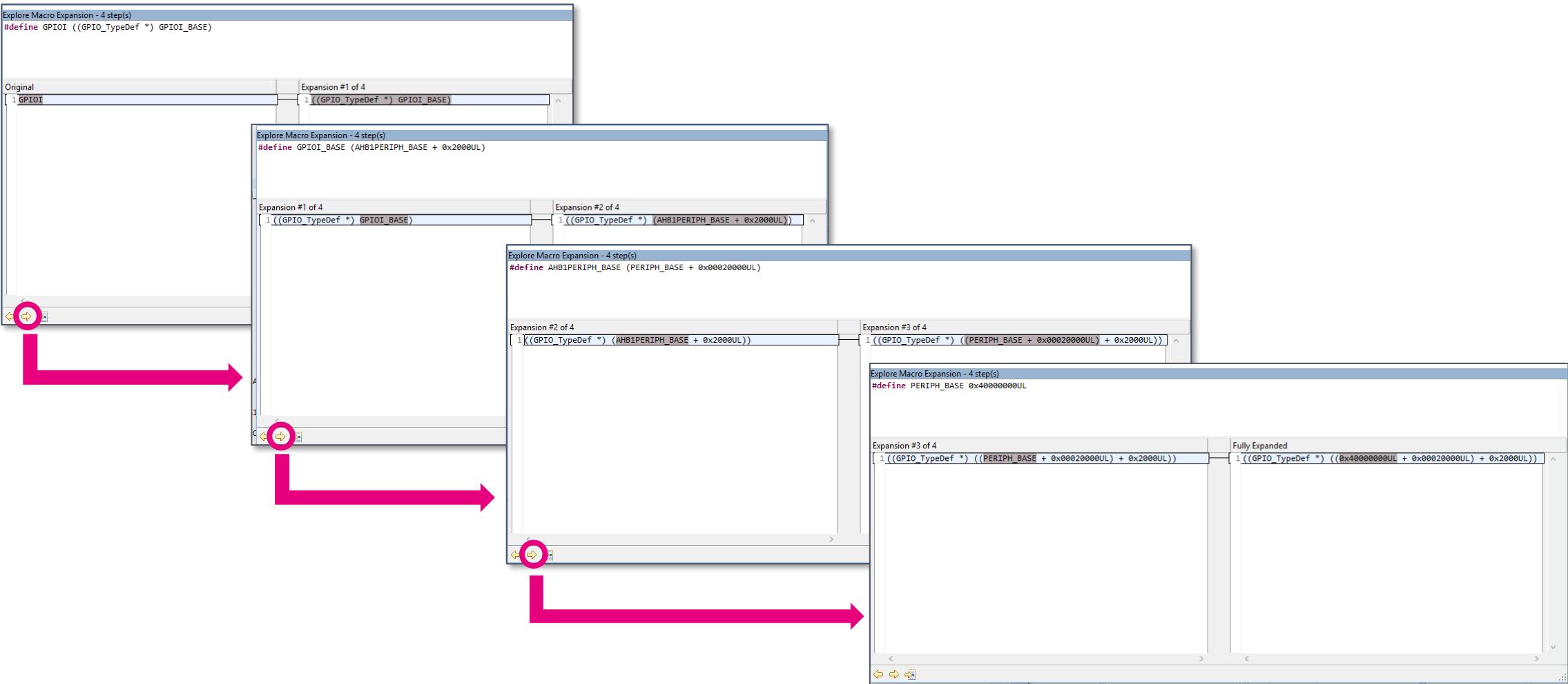
The screenshot shows the STM32CubeIDE interface with the project '01_F746-DISCO' open. The main window displays the 'main.c' file. A red circle labeled '1' highlights the macro expansion step, with the text '將滑鼠游標停在欲查詢處' (Stop the mouse cursor over the query point). Another red circle labeled '2' highlights the expanded code, with the text '一秒後出現巨集展開的結果' (The result of macro expansion appears after one second).

```
104  /* USER CODE END 2 */
105
106  /* Infinite loop */
107  /* USER CODE BEGIN WHILE */
108  while (1)
109  {
110      /* USER CODE END */
111
112      /* USER CODE BEGIN */
113      HAL_Delay(500);
114      HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
115      printf("LD1: %d\r\n", LD1);
116  } /* USER CODE END 3 */
117  ((GPIO_TypeDef *) ((0x40000000UL + 0x00020000UL) + 0x2000UL))
118
119 /**
120  * @brief System Clock Configuration
121  * @retval None
122  */
123
124 void SystemClock_Config(void)
125 {
126     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
127     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
128     RCC_PeriphClkInitTypeDef PeriphClkInitStruct = {0};
129
130     /* Configure the main internal regulator output voltage
131     */
132     __HAL_RCC_PWR_CLK_ENABLE();
133     __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
134     /* Initializes the CPU, AHB and APB busses clocks
135     */
136     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
137     RCC_OscInitStruct.HSISState = RCC_HSI_ON;
138     RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
139     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
140     RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
141     RCC_OscInitStruct.PLL.PLLN = 102;
142     RCC_OscInitStruct.PLL.PLLP = 210;
143     RCC_OscInitStruct.PLL.PLLQ = 2;
144     RCC_OscInitStruct.PLL.PLLR = 2;
145     if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
146     {
147         Error_Handler();
148     }
149     /* Initializes the CPU, AHB and APB busses clocks
150     */
151     RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_SYSCLK
152     | RCC_CLOCKTYPE_PCLK1 | RCC_CLOCKTYPE_PCLK2;
153     RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
154     RCC_ClkInitStruct.AHCLKDivider = RCC_SYSCLK_DIV2;
155     RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
156     RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
157
158     if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
159     {
160         Error_Handler();
161     }
162 }
```

功能體驗 – 巨集的展開 (2)



功能體驗 – 巨集的展開 (3)





功能體驗 – Symbol Hyperlink

1 將滑鼠游標移到函式處

```

159 /* USER CODE END DebugMonitor_IRQn 0 */
160 /* USER CODE BEGIN DebugMonitor_IRQn 1 */
161 // ...
162 /* USER CODE END DebugMonitor_IRQn 1 */
163 // ...
164 }
165
166 /**
167 * @brief This function handles Pendable request for system service.
168 */
169 void PendSV_Handler(void)
170 {
171     /* USER CODE BEGIN PendSV_IRQn 0 */
172     // ...
173     /* USER CODE END PendSV_IRQn 0 */
174     /* USER CODE BEGIN PendSV_IRQn 1 */
175     // ...
176     /* USER CODE END PendSV_IRQn 1 */
177 }
178
179 /**
180 * @brief This function handles System tick timer.
181 */
182 void SysTick_Handler(void)
183 {
184     /* USER CODE BEGIN SysTick_IRQn 0 */
185     // ...
186     /* USER CODE BEGIN SysTick_IRQn 1 */
187     HAL_IncTick();
188     // ...
189 }
190
191 /**
192 * @brief This function handles Interrupt Handlers for the used peripherals.
193 */
194 /**
195 * For available peripheral interrupt handler names,
196 * please refer to the startup file (startup_stm32f7xx.s).
197 */
198
199 /**
200 * Handles EXTI line[15:10] interrupts.
201 */
202 void EXTI15_10_IRQHandler(void)
203 {
204     /* USER CODE BEGIN EXTI15_10_IRQn 0 */
205     // ...
206     /* USER CODE END EXTI15_10_IRQn 0 */
207     HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_11);
208     /* USER CODE BEGIN EXTI15_10_IRQn 1 */
209     // ...
210     /* USER CODE END EXTI15_10_IRQn 1 */
211 }
212
213 /* USER CODE BEGIN 1 */
214
215 /* USER CODE END 1 */
216 /* USER CODE END 1 */
217 //***** (C) COPYRIGHT STMicroelectronics *****END OF FILE*** */
218

```

2 按下Ctrl鍵後函式呈現超連結狀態

```

159 /* USER CODE END DebugMonitor_IRQn 0 */
160 /* USER CODE BEGIN DebugMonitor_IRQn 1 */
161 // ...
162 /* USER CODE END DebugMonitor_IRQn 1 */
163 // ...
164 }
165
166 /**
167 * @brief This function handles Pendable request for system service.
168 */
169 void PendSV_Handler(void)
170 {
171     /* USER CODE BEGIN PendSV_IRQn 0 */
172     // ...
173     /* USER CODE END PendSV_IRQn 0 */
174     /* USER CODE BEGIN PendSV_IRQn 1 */
175     // ...
176     /* USER CODE END PendSV_IRQn 1 */
177 }
178
179 /**
180 * @brief This function handles System tick timer.
181 */
182 void SysTick_Handler(void)
183 {
184     /* USER CODE BEGIN SysTick_IRQn 0 */
185     // ...
186     /* USER CODE BEGIN SysTick_IRQn 1 */
187     HAL_IncTick();
188     // ...
189 }
190
191 /**
192 * @brief This function handles Interrupt Handlers for the used peripherals.
193 */
194 /**
195 * For available peripheral interrupt handler names,
196 * please refer to the startup file (startup_stm32f7xx.s).
197 */
198
199 /**
200 * Handles EXTI line[15:10] interrupts.
201 */
202 void EXTI15_10_IRQHandler(void)
203 {
204     /* USER CODE BEGIN EXTI15_10_IRQn 0 */
205     // ...
206     /* USER CODE END EXTI15_10_IRQn 0 */
207     HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_11);
208     /* USER CODE BEGIN EXTI15_10_IRQn 1 */
209     // ...
210     /* USER CODE END EXTI15_10_IRQn 1 */
211 }
212
213 /* USER CODE BEGIN 1 */
214
215 /* USER CODE END 1 */
216 /* USER CODE END 1 */
217 //***** (C) COPYRIGHT STMicroelectronics *****END OF FILE*** */
218

```

3 按下Ctrl+滑鼠左鍵後即可跳轉到函式的實作

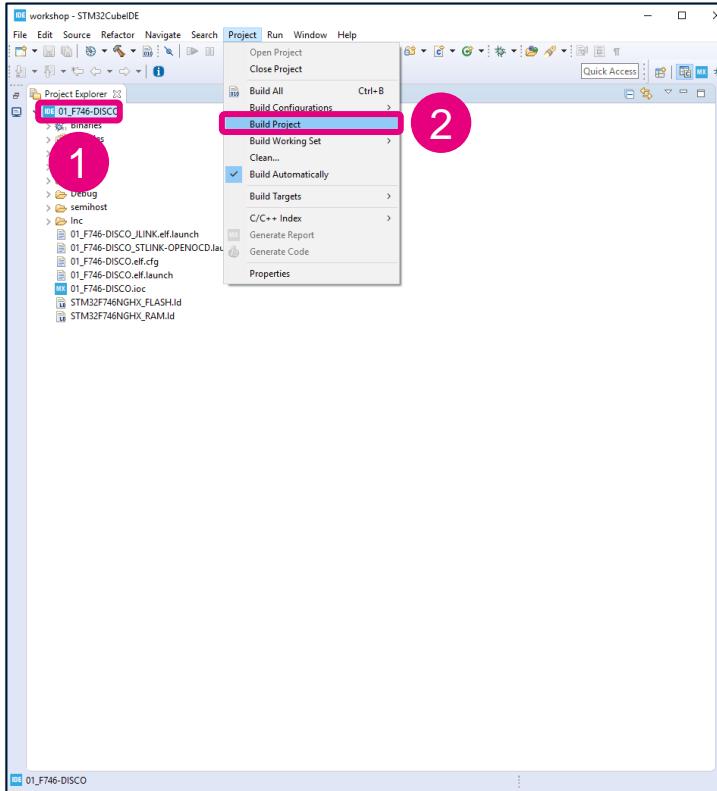
```

271 /**
272 * @brief Get the device identifier
273 */
274 void HAL_GetDeviceID(void)
275 {
276     /* @note In the default implementation, this variable is incremented each ms
277      * in SysTick ISR.
278     */
279     /* @note This function is declared as __weak to be overwritten in case of other
280      * implementations in user file.
281     */
282     /* @note None */
283
284     uwTick += uwTickFreq;
285 }
286
287 /**
288 * @brief Provides a tick value in millisecond.
289 */
290 /* @note This function is declared as __weak to be overwritten in case of other
291      * implementations in user file.
292 */
293 /* @note None */
294
295 /**
296 * @brief Returns the tick priority
297 */
298 /* @note None */
299
300 /**
301 * @brief Set the tick priority
302 */
303 void HAL_SetTickPriority(void)
304 {
305     /* @note None */
306
307     return uwTick;
308 }
309
310 /**
311 * @brief Set the tick frequency
312 */
313 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
314 {
315     /* @note None */
316
317     uwTickFreq = Freq;
318
319     /* Apply the new tick Freq */
320     status = HAL_InitTick(uwTickPriority);
321 }
322
323 /**
324 * @brief Set the tick frequency
325 */
326 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
327 {
328     /* @note None */
329
330 }
331
332 /**
333 * @brief Set the tick frequency
334 */
335 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
336 {
337     /* @note None */
338
339 }
340
341 /**
342 * @brief Set the tick frequency
343 */
344 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
345 {
346     /* @note None */
347
348 }
349
350 /**
351 * @brief Set the tick frequency
352 */
353 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
354 {
355     /* @note None */
356
357 }
358
359 /**
360 * @brief Set the tick frequency
361 */
362 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
363 {
364     /* @note None */
365
366 }
367
368 /**
369 * @brief Set the tick frequency
370 */
371 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
372 {
373     /* @note None */
374
375 }
376
377 /**
378 * @brief Set the tick frequency
379 */
380 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
381 {
382     /* @note None */
383
384 }
385
386 /**
387 * @brief Set the tick frequency
388 */
389 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
390 {
391     /* @note None */
392
393 }
394
395 /**
396 * @brief Set the tick frequency
397 */
398 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
399 {
400     /* @note None */
401
402 }
403
404 /**
405 * @brief Set the tick frequency
406 */
407 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
408 {
409     /* @note None */
410
411 }
412
413 /**
414 * @brief Set the tick frequency
415 */
416 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
417 {
418     /* @note None */
419
420 }
421
422 /**
423 * @brief Set the tick frequency
424 */
425 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
426 {
427     /* @note None */
428
429 }
430
431 /**
432 * @brief Set the tick frequency
433 */
434 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
435 {
436     /* @note None */
437
438 }
439
440 /**
441 * @brief Set the tick frequency
442 */
443 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
444 {
445     /* @note None */
446
447 }
448
449 /**
450 * @brief Set the tick frequency
451 */
452 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
453 {
454     /* @note None */
455
456 }
457
458 /**
459 * @brief Set the tick frequency
460 */
461 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
462 {
463     /* @note None */
464
465 }
466
467 /**
468 * @brief Set the tick frequency
469 */
470 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
471 {
472     /* @note None */
473
474 }
475
476 /**
477 * @brief Set the tick frequency
478 */
479 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
480 {
481     /* @note None */
482
483 }
484
485 /**
486 * @brief Set the tick frequency
487 */
488 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
489 {
490     /* @note None */
491
492 }
493
494 /**
495 * @brief Set the tick frequency
496 */
497 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
498 {
499     /* @note None */
500
501 }
502
503 /**
504 * @brief Set the tick frequency
505 */
506 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
507 {
508     /* @note None */
509
510 }
511
512 /**
513 * @brief Set the tick frequency
514 */
515 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
516 {
517     /* @note None */
518
519 }
520
521 /**
522 * @brief Set the tick frequency
523 */
524 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
525 {
526     /* @note None */
527
528 }
529
530 /**
531 * @brief Set the tick frequency
532 */
533 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
534 {
535     /* @note None */
536
537 }
538
539 /**
540 * @brief Set the tick frequency
541 */
542 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
543 {
544     /* @note None */
545
546 }
547
548 /**
549 * @brief Set the tick frequency
550 */
551 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
552 {
553     /* @note None */
554
555 }
556
557 /**
558 * @brief Set the tick frequency
559 */
560 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
561 {
562     /* @note None */
563
564 }
565
566 /**
567 * @brief Set the tick frequency
568 */
569 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
570 {
571     /* @note None */
572
573 }
574
575 /**
576 * @brief Set the tick frequency
577 */
578 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
579 {
580     /* @note None */
581
582 }
583
584 /**
585 * @brief Set the tick frequency
586 */
587 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
588 {
589     /* @note None */
590
591 }
592
593 /**
594 * @brief Set the tick frequency
595 */
596 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
597 {
598     /* @note None */
599
600 }
601
602 /**
603 * @brief Set the tick frequency
604 */
605 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
606 {
607     /* @note None */
608
609 }
610
611 /**
612 * @brief Set the tick frequency
613 */
614 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
615 {
616     /* @note None */
617
618 }
619
620 /**
621 * @brief Set the tick frequency
622 */
623 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
624 {
625     /* @note None */
626
627 }
628
629 /**
630 * @brief Set the tick frequency
631 */
632 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
633 {
634     /* @note None */
635
636 }
637
638 /**
639 * @brief Set the tick frequency
640 */
641 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
642 {
643     /* @note None */
644
645 }
646
647 /**
648 * @brief Set the tick frequency
649 */
650 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
651 {
652     /* @note None */
653
654 }
655
656 /**
657 * @brief Set the tick frequency
658 */
659 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
660 {
661     /* @note None */
662
663 }
664
665 /**
666 * @brief Set the tick frequency
667 */
668 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
669 {
670     /* @note None */
671
672 }
673
674 /**
675 * @brief Set the tick frequency
676 */
677 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
678 {
679     /* @note None */
680
681 }
682
683 /**
684 * @brief Set the tick frequency
685 */
686 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
687 {
688     /* @note None */
689
690 }
691
692 /**
693 * @brief Set the tick frequency
694 */
695 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
696 {
697     /* @note None */
698
699 }
699
700 /**
701 * @brief Set the tick frequency
702 */
703 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
704 {
705     /* @note None */
706
707 }
708
709 /**
710 * @brief Set the tick frequency
711 */
712 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
713 {
714     /* @note None */
715
716 }
717
718 /**
719 * @brief Set the tick frequency
720 */
721 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
722 {
723     /* @note None */
724
725 }
726
727 /**
728 * @brief Set the tick frequency
729 */
730 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
731 {
732     /* @note None */
733
734 }
735
736 /**
737 * @brief Set the tick frequency
738 */
739 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
740 {
741     /* @note None */
742
743 }
744
745 /**
746 * @brief Set the tick frequency
747 */
748 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
749 {
750     /* @note None */
751
752 }
753
754 /**
755 * @brief Set the tick frequency
756 */
757 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
758 {
759     /* @note None */
760
761 }
762
763 /**
764 * @brief Set the tick frequency
765 */
766 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
767 {
768     /* @note None */
769
770 }
771
772 /**
773 * @brief Set the tick frequency
774 */
775 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
776 {
777     /* @note None */
778
779 }
779
780 /**
781 * @brief Set the tick frequency
782 */
783 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
784 {
785     /* @note None */
786
787 }
788
789 /**
790 * @brief Set the tick frequency
791 */
792 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
793 {
794     /* @note None */
795
796 }
797
798 /**
799 * @brief Set the tick frequency
800 */
801 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
802 {
803     /* @note None */
804
805 }
806
807 /**
808 * @brief Set the tick frequency
809 */
810 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
811 {
812     /* @note None */
813
814 }
815
816 /**
817 * @brief Set the tick frequency
818 */
819 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
820 {
821     /* @note None */
822
823 }
824
825 /**
826 * @brief Set the tick frequency
827 */
828 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
829 {
830     /* @note None */
831
832 }
833
834 /**
835 * @brief Set the tick frequency
836 */
837 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
838 {
839     /* @note None */
840
841 }
842
843 /**
844 * @brief Set the tick frequency
845 */
846 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
847 {
848     /* @note None */
849
850 }
851
852 /**
853 * @brief Set the tick frequency
854 */
855 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
856 {
857     /* @note None */
858
859 }
859
860 /**
861 * @brief Set the tick frequency
862 */
863 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
864 {
865     /* @note None */
866
867 }
868
869 /**
870 * @brief Set the tick frequency
871 */
872 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
873 {
874     /* @note None */
875
876 }
877
878 /**
879 * @brief Set the tick frequency
880 */
881 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
882 {
883     /* @note None */
884
885 }
886
887 /**
888 * @brief Set the tick frequency
889 */
890 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
891 {
892     /* @note None */
893
894 }
895
896 /**
897 * @brief Set the tick frequency
898 */
899 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
900 {
901     /* @note None */
902
903 }
904
905 /**
906 * @brief Set the tick frequency
907 */
908 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
909 {
910     /* @note None */
911
912 }
913
914 /**
915 * @brief Set the tick frequency
916 */
917 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
918 {
919     /* @note None */
920
921 }
922
923 /**
924 * @brief Set the tick frequency
925 */
926 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
927 {
928     /* @note None */
929
930 }
931
932 /**
933 * @brief Set the tick frequency
934 */
935 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
936 {
937     /* @note None */
938
939 }
940
941 /**
942 * @brief Set the tick frequency
943 */
944 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
945 {
946     /* @note None */
947
948 }
949
950 /**
951 * @brief Set the tick frequency
952 */
953 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
954 {
955     /* @note None */
956
957 }
958
959 /**
960 * @brief Set the tick frequency
961 */
962 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
963 {
964     /* @note None */
965
966 }
967
968 /**
969 * @brief Set the tick frequency
970 */
971 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
972 {
973     /* @note None */
974
975 }
976
977 /**
978 * @brief Set the tick frequency
979 */
980 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
981 {
982     /* @note None */
983
984 }
985
986 /**
987 * @brief Set the tick frequency
988 */
989 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
990 {
991     /* @note None */
992
993 }
994
995 /**
996 * @brief Set the tick frequency
997 */
998 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
999 {
999     /* @note None */
1000
1001 }
1002
1003 /**
1004 * @brief Set the tick frequency
1005 */
1006 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1007 {
1008     /* @note None */
1009
1010 }
1011
1012 /**
1013 * @brief Set the tick frequency
1014 */
1015 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1016 {
1017     /* @note None */
1018
1019 }
1020
1021 /**
1022 * @brief Set the tick frequency
1023 */
1024 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1025 {
1026     /* @note None */
1027
1028 }
1029
1030 /**
1031 * @brief Set the tick frequency
1032 */
1033 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1034 {
1035     /* @note None */
1036
1037 }
1038
1039 /**
1040 * @brief Set the tick frequency
1041 */
1042 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1043 {
1044     /* @note None */
1045
1046 }
1047
1048 /**
1049 * @brief Set the tick frequency
1050 */
1051 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1052 {
1053     /* @note None */
1054
1055 }
1056
1057 /**
1058 * @brief Set the tick frequency
1059 */
1060 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1061 {
1062     /* @note None */
1063
1064 }
1065
1066 /**
1067 * @brief Set the tick frequency
1068 */
1069 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1070 {
1071     /* @note None */
1072
1073 }
1074
1075 /**
1076 * @brief Set the tick frequency
1077 */
1078 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1079 {
1080     /* @note None */
1081
1082 }
1083
1084 /**
1085 * @brief Set the tick frequency
1086 */
1087 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1088 {
1089     /* @note None */
1090
1091 }
1092
1093 /**
1094 * @brief Set the tick frequency
1095 */
1096 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1097 {
1098     /* @note None */
1099
1100 }
1101
1102 /**
1103 * @brief Set the tick frequency
1104 */
1105 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1106 {
1107     /* @note None */
1108
1109 }
1110
1111 /**
1112 * @brief Set the tick frequency
1113 */
1114 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1115 {
1116     /* @note None */
1117
1118 }
1119
1120 /**
1121 * @brief Set the tick frequency
1122 */
1123 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1124 {
1125     /* @note None */
1126
1127 }
1128
1129 /**
1130 * @brief Set the tick frequency
1131 */
1132 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1133 {
1134     /* @note None */
1135
1136 }
1137
1138 /**
1139 * @brief Set the tick frequency
1140 */
1141 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1142 {
1143     /* @note None */
1144
1145 }
1146
1147 /**
1148 * @brief Set the tick frequency
1149 */
1150 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1151 {
1152     /* @note None */
1153
1154 }
1155
1156 /**
1157 * @brief Set the tick frequency
1158 */
1159 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1160 {
1161     /* @note None */
1162
1163 }
1164
1165 /**
1166 * @brief Set the tick frequency
1167 */
1168 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1169 {
1170     /* @note None */
1171
1172 }
1173
1174 /**
1175 * @brief Set the tick frequency
1176 */
1177 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1178 {
1179     /* @note None */
1180
1181 }
1182
1183 /**
1184 * @brief Set the tick frequency
1185 */
1186 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1187 {
1188     /* @note None */
1189
1190 }
1191
1192 /**
1193 * @brief Set the tick frequency
1194 */
1195 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1196 {
1197     /* @note None */
1198
1199 }
1200
1201 /**
1202 * @brief Set the tick frequency
1203 */
1204 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1205 {
1206     /* @note None */
1207
1208 }
1209
1210 /**
1211 * @brief Set the tick frequency
1212 */
1213 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1214 {
1215     /* @note None */
1216
1217 }
1218
1219 /**
1220 * @brief Set the tick frequency
1221 */
1222 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1223 {
1224     /* @note None */
1225
1226 }
1227
1228 /**
1229 * @brief Set the tick frequency
1230 */
1231 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1232 {
1233     /* @note None */
1234
1235 }
1236
1237 /**
1238 * @brief Set the tick frequency
1239 */
1240 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1241 {
1242     /* @note None */
1243
1244 }
1245
1246 /**
1247 * @brief Set the tick frequency
1248 */
1249 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1250 {
1251     /* @note None */
1252
1253 }
1254
1255 /**
1256 * @brief Set the tick frequency
1257 */
1258 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1259 {
1260     /* @note None */
1261
1262 }
1263
1264 /**
1265 * @brief Set the tick frequency
1266 */
1267 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1268 {
1269     /* @note None */
1270
1271 }
1272
1273 /**
1274 * @brief Set the tick frequency
1275 */
1276 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1277 {
1278     /* @note None */
1279
1280 }
1281
1282 /**
1283 * @brief Set the tick frequency
1284 */
1285 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1286 {
1287     /* @note None */
1288
1289 }
1290
1291 /**
1292 * @brief Set the tick frequency
1293 */
1294 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1295 {
1296     /* @note None */
1297
1298 }
1299
1300 /**
1301 * @brief Set the tick frequency
1302 */
1303 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1304 {
1305     /* @note None */
1306
1307 }
1308
1309 /**
1310 * @brief Set the tick frequency
1311 */
1312 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1313 {
1314     /* @note None */
1315
1316 }
1317
1318 /**
1319 * @brief Set the tick frequency
1320 */
1321 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1322 {
1323     /* @note None */
1324
1325 }
1326
1327 /**
1328 * @brief Set the tick frequency
1329 */
1330 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1331 {
1332     /* @note None */
1333
1334 }
1335
1336 /**
1337 * @brief Set the tick frequency
1338 */
1339 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1340 {
1341     /* @note None */
1342
1343 }
1344
1345 /**
1346 * @brief Set the tick frequency
1347 */
1348 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1349 {
1350     /* @note None */
1351
1352 }
1353
1354 /**
1355 * @brief Set the tick frequency
1356 */
1357 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1358 {
1359     /* @note None */
1360
1361 }
1362
1363 /**
1364 * @brief Set the tick frequency
1365 */
1366 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1367 {
1368     /* @note None */
1369
1370 }
1371
1372 /**
1373 * @brief Set the tick frequency
1374 */
1375 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1376 {
1377     /* @note None */
1378
1379 }
1380
1381 /**
1382 * @brief Set the tick frequency
1383 */
1384 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1385 {
1386     /* @note None */
1387
1388 }
1389
1390 /**
1391 * @brief Set the tick frequency
1392 */
1393 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1394 {
1395     /* @note None */
1396
1397 }
1398
1399 /**
1400 * @brief Set the tick frequency
1401 */
1402 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1403 {
1404     /* @note None */
1405
1406 }
1407
1408 /**
1409 * @brief Set the tick frequency
1410 */
1411 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1412 {
1413     /* @note None */
1414
1415 }
1416
1417 /**
1418 * @brief Set the tick frequency
1419 */
1420 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1421 {
1422     /* @note None */
1423
1424 }
1425
1426 /**
1427 * @brief Set the tick frequency
1428 */
1429 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1430 {
1431     /* @note None */
1432
1433 }
1434
1435 /**
1436 * @brief Set the tick frequency
1437 */
1438 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1439 {
1440     /* @note None */
1441
1442 }
1443
1444 /**
1445 * @brief Set the tick frequency
1446 */
1447 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1448 {
1449     /* @note None */
1450
1451 }
1452
1453 /**
1454 * @brief Set the tick frequency
1455 */
1456 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1457 {
1458     /* @note None */
1459
1460 }
1461
1462 /**
1463 * @brief Set the tick frequency
1464 */
1465 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1466 {
1467     /* @note None */
1468
1469 }
1470
1471 /**
1472 * @brief Set the tick frequency
1473 */
1474 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1475 {
1476     /* @note None */
1477
1478 }
1479
1480 /**
1481 * @brief Set the tick frequency
1482 */
1483 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1484 {
1485     /* @note None */
1486
1487 }
1488
1489 /**
1490 * @brief Set the tick frequency
1491 */
1492 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1493 {
1494     /* @note None */
1495
1496 }
1497
1498 /**
1499 * @brief Set the tick frequency
1500 */
1501 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1502 {
1503     /* @note None */
1504
1505 }
1506
1507 /**
1508 * @brief Set the tick frequency
1509 */
1510 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1511 {
1512     /* @note None */
1513
1514 }
1515
1516 /**
1517 * @brief Set the tick frequency
1518 */
1519 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1520 {
1521     /* @note None */
1522
1523 }
1524
1525 /**
1526 * @brief Set the tick frequency
1527 */
1528 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1529 {
1530     /* @note None */
1531
1532 }
1533
1534 /**
1535 * @brief Set the tick frequency
1536 */
1537 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1538 {
1539     /* @note None */
1540
1541 }
1542
1543 /**
1544 * @brief Set the tick frequency
1545 */
1546 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1547 {
1548     /* @note None */
1549
1550 }
1551
1552 /**
1553 * @brief Set the tick frequency
1554 */
1555 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1556 {
1557     /* @note None */
1558
1559 }
1560
1561 /**
1562 * @brief Set the tick frequency
1563 */
1564 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1565 {
1566     /* @note None */
1567
1568 }
1569
1570 /**
1571 * @brief Set the tick frequency
1572 */
1573 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1574 {
1575     /* @note None */
1576
1577 }
1578
1579 /**
1580 * @brief Set the tick frequency
1581 */
1582 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1583 {
1584     /* @note None */
1585
1586 }
1587
1588 /**
1589 * @brief Set the tick frequency
1590 */
1591 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1592 {
1593     /* @note None */
1594
1595 }
1596
1597 /**
1598 * @brief Set the tick frequency
1599 */
1600 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1601 {
1602     /* @note None */
1603
1604 }
1605
1606 /**
1607 * @brief Set the tick frequency
1608 */
1609 void HAL_SetTickFreq( HAL_TickFreqTypeDef Freq )
1610 {
1611     /* @note None */
1612
1
```

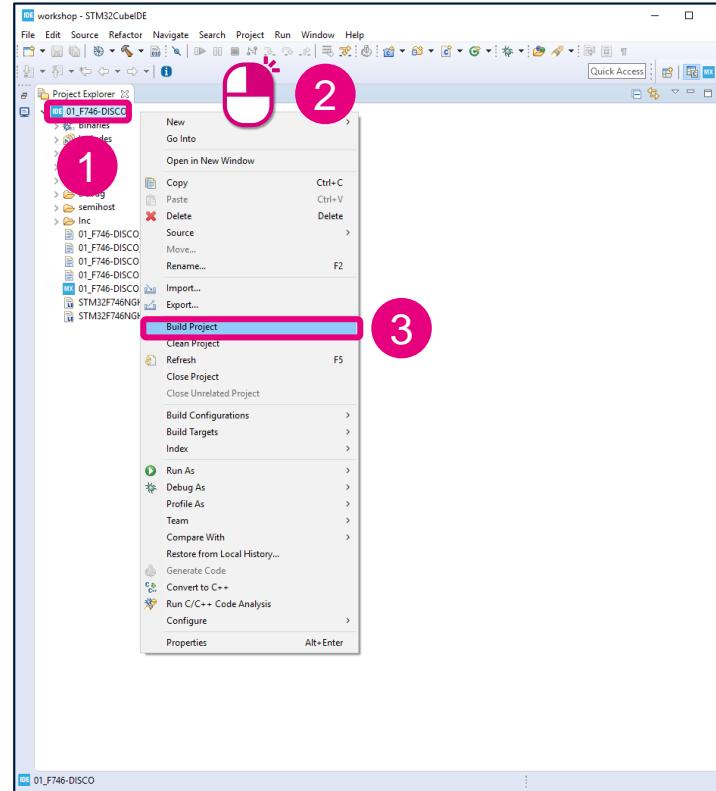


基本功能 – 編譯專案 (Build Project) 的方式

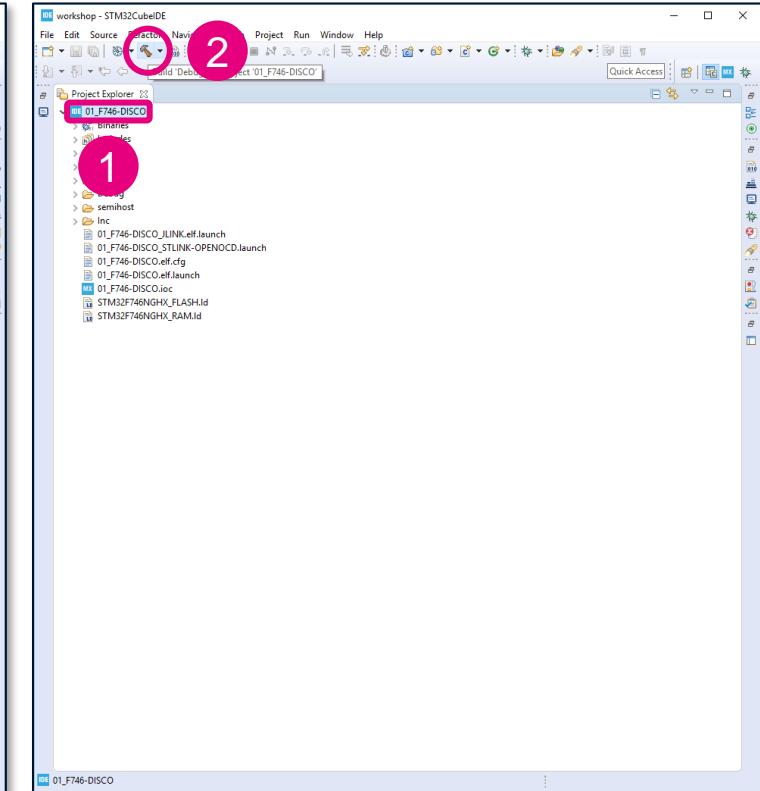
Option 1



Option 2



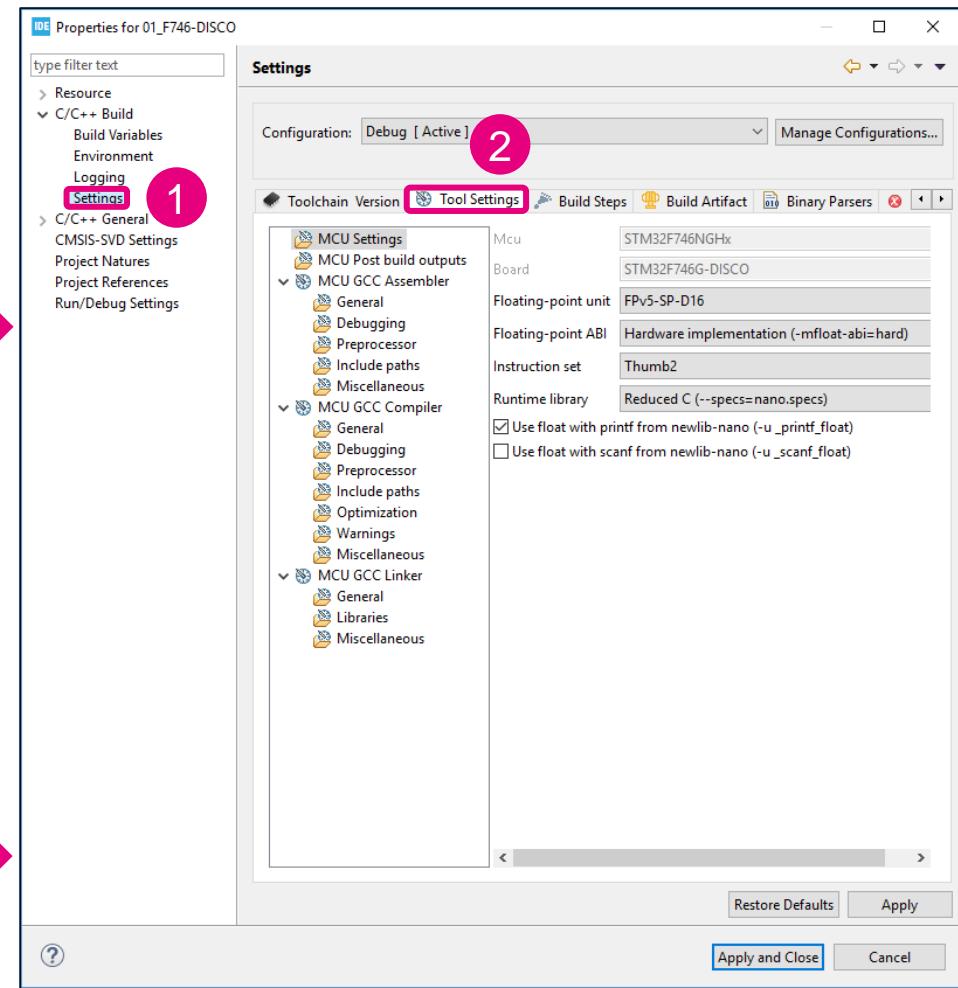
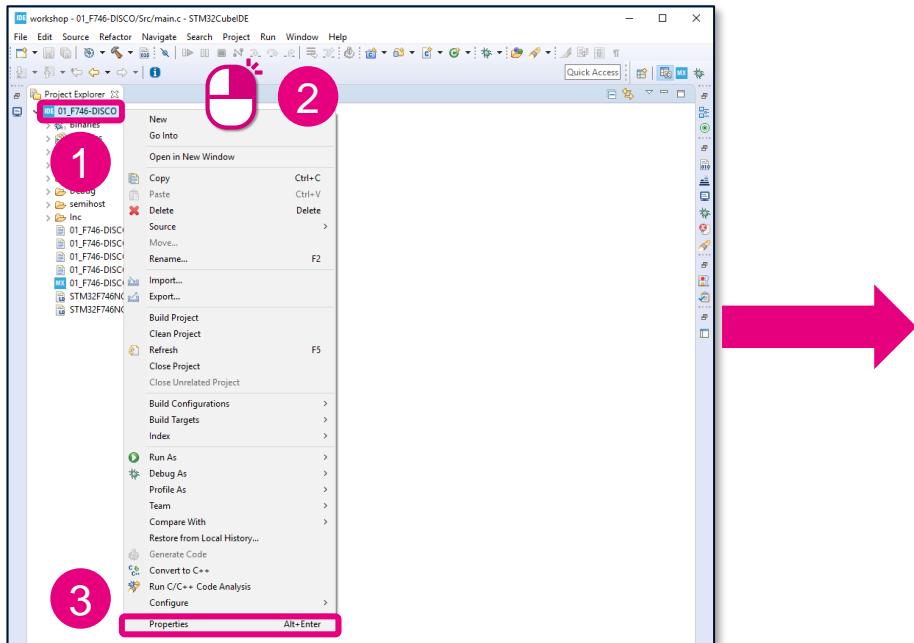
Option 3



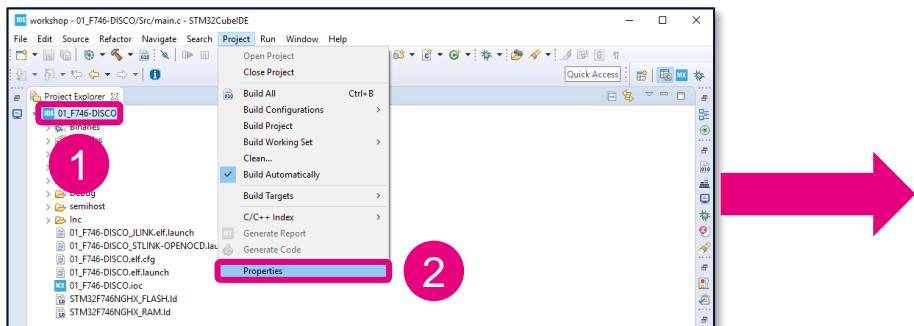


基本功能 – 開啟編譯設定視窗的方式

Option 1



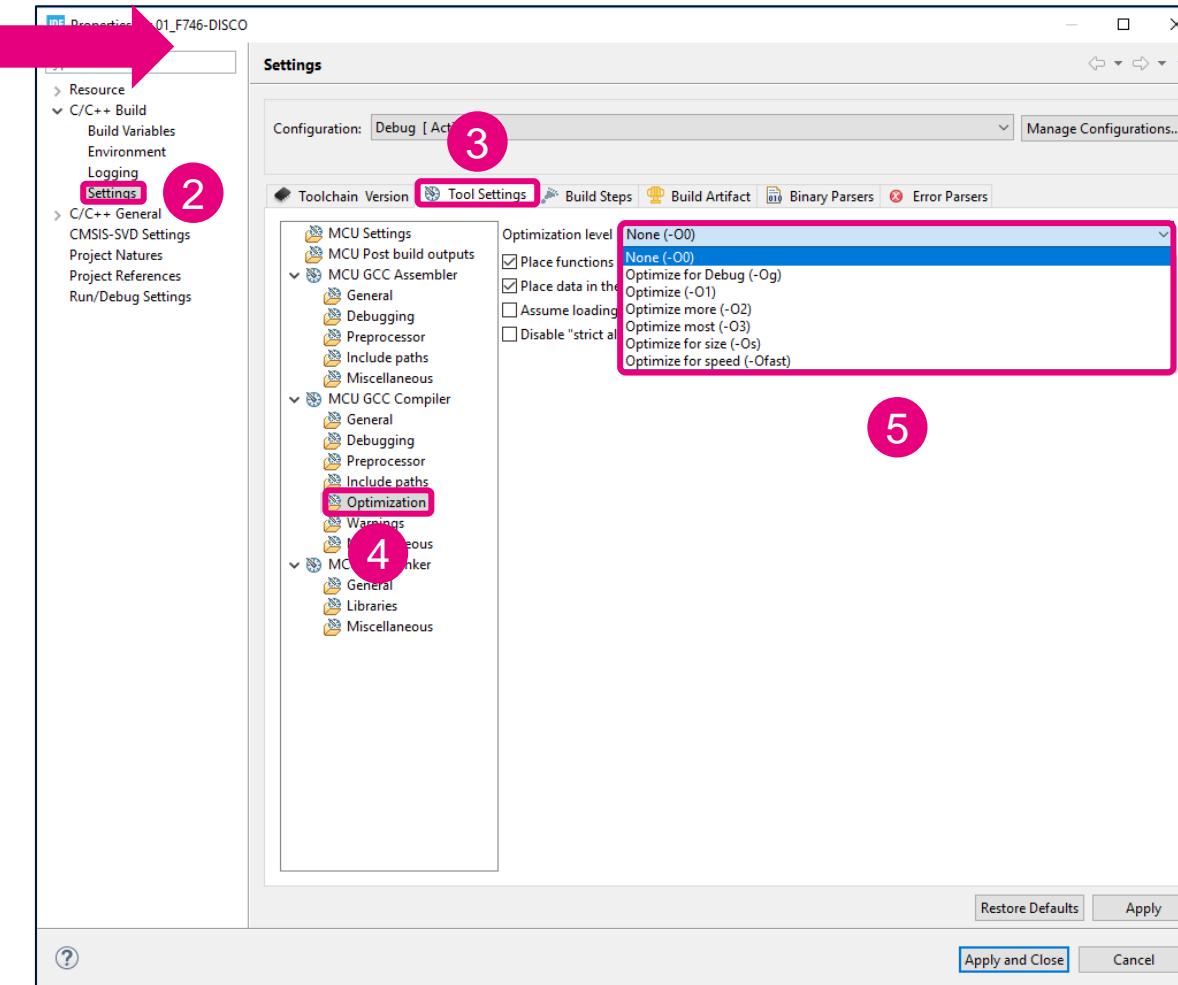
Option 2



基本功能 – 最佳化設定



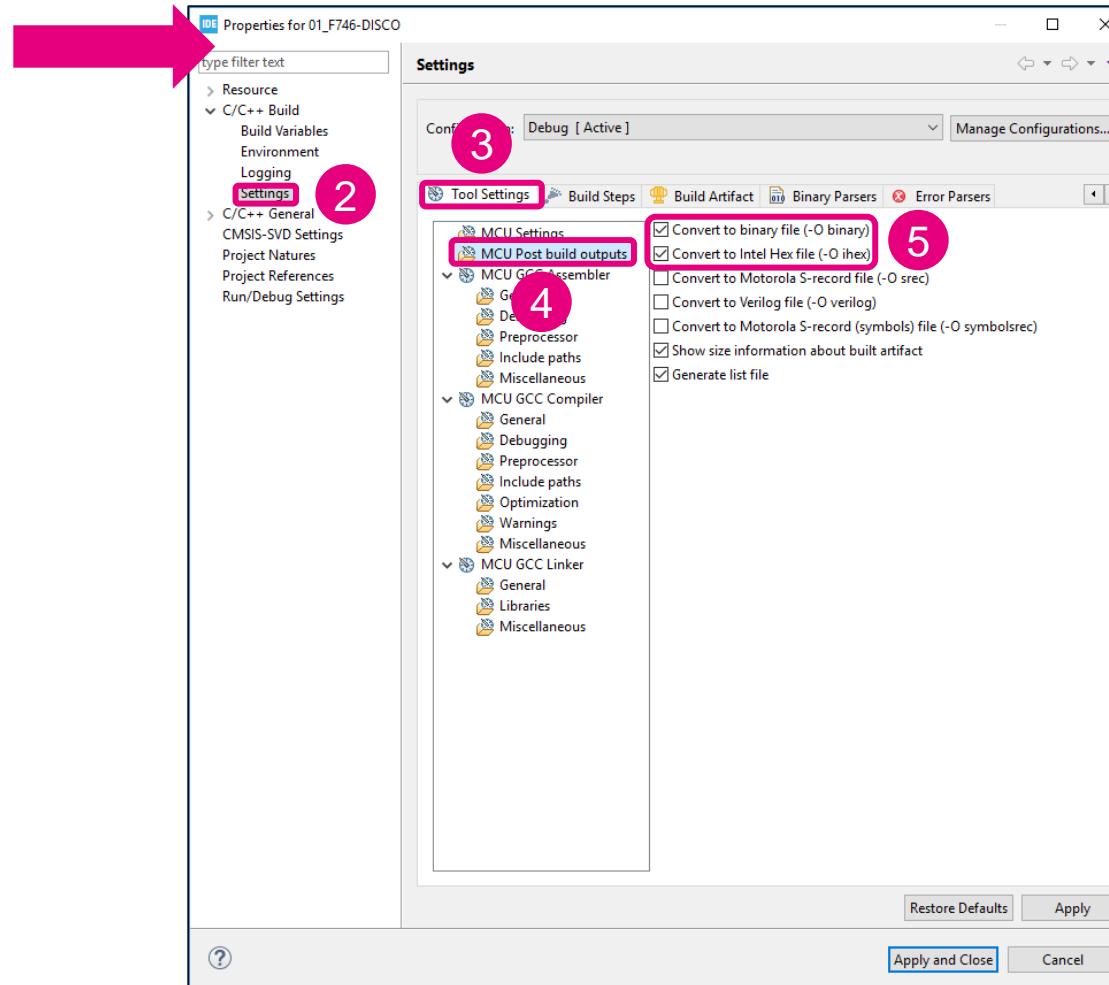
1 開啟編譯設定視窗



基本功能 – 產生Hex或BIN檔



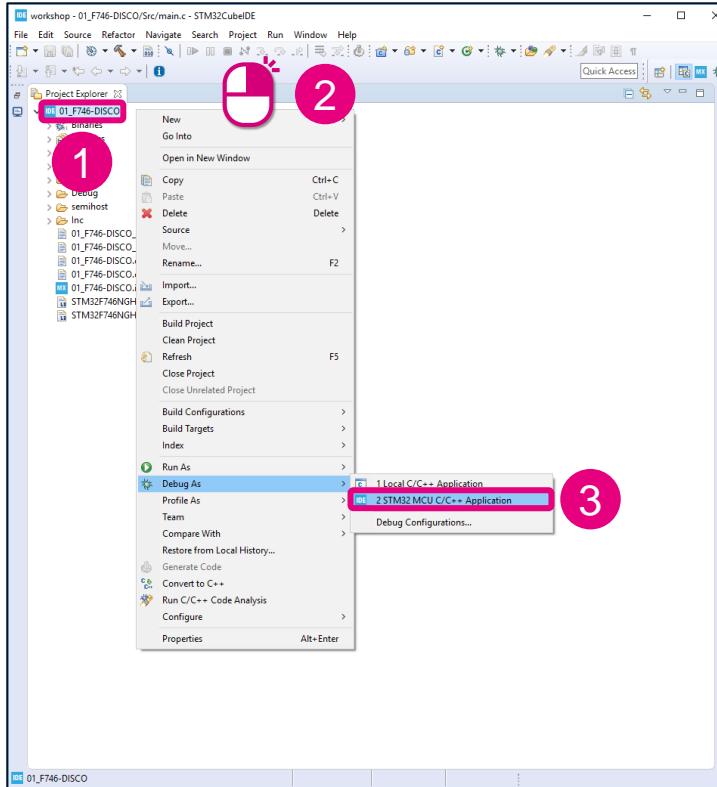
1 開啟編譯設定視窗



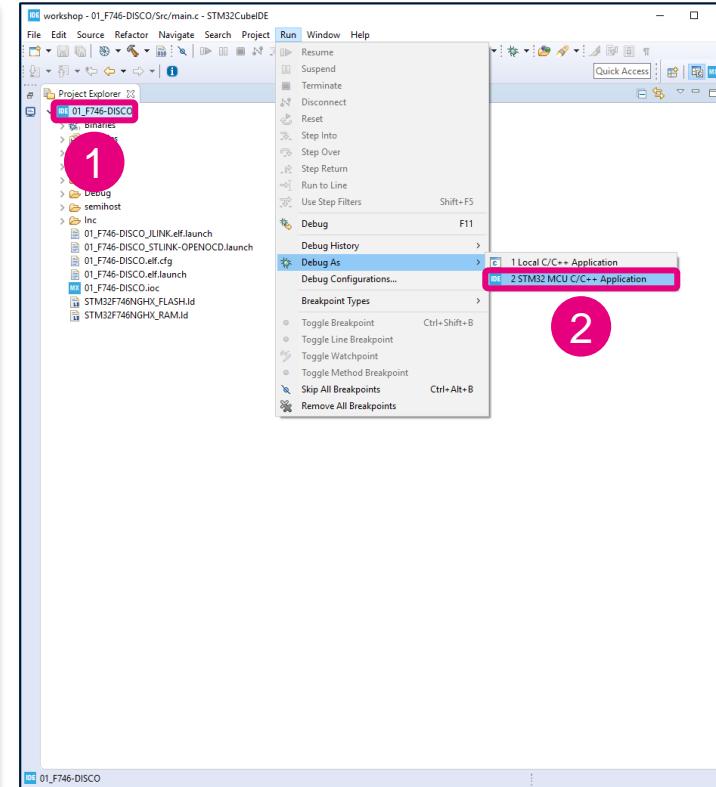
基本功能 – 執行專案的方式



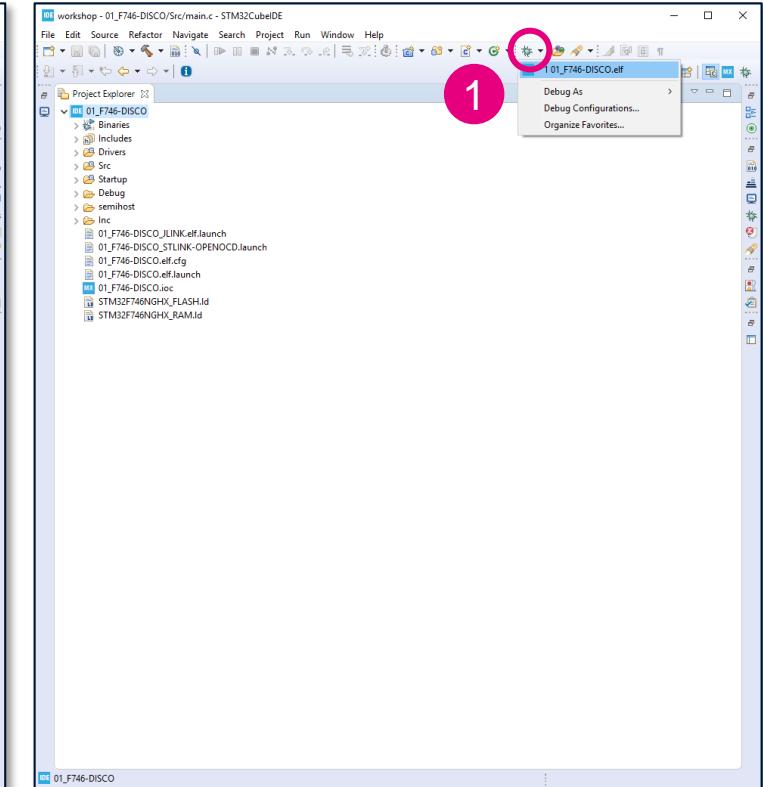
Option 1



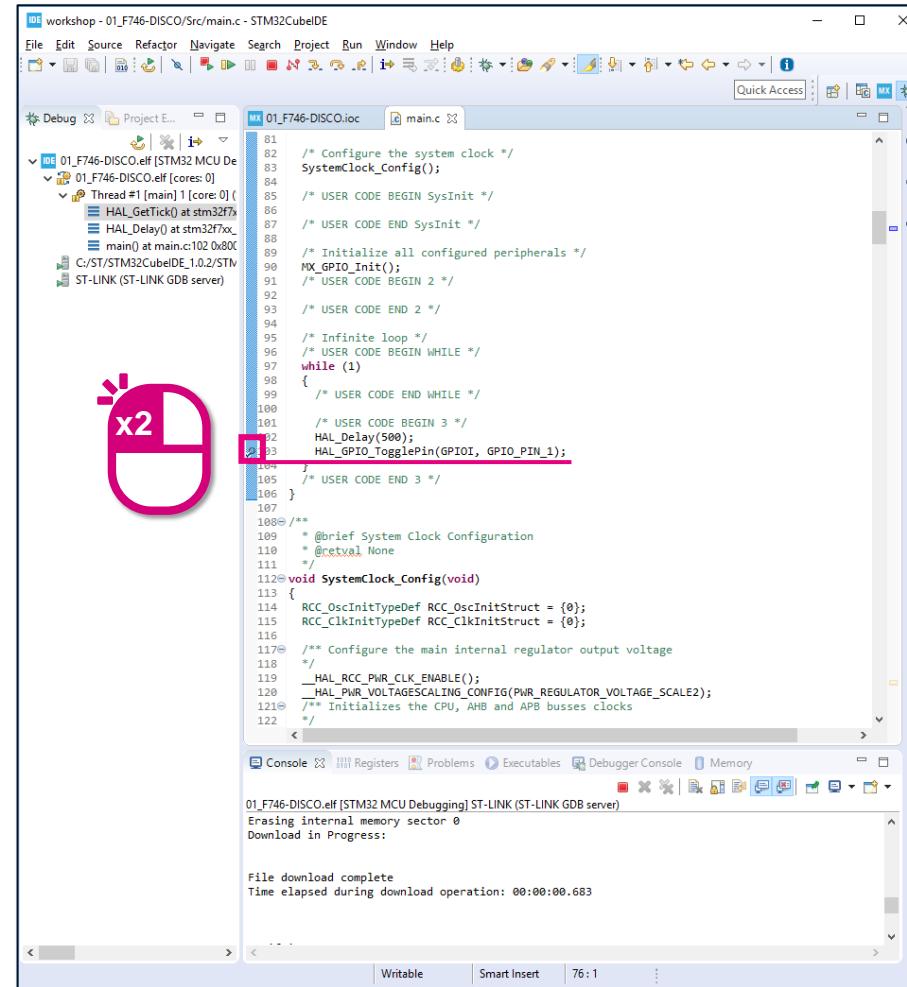
Option 2



Option 3



基本功能 – 設置中斷點



The screenshot shows the STM32CubeIDE interface with the project 'workshop - 01_F746-DISCO/Src/main.c - STM32CubeIDE' open. The code editor displays the main.c file, which contains the following code:

```
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
91  /* USER CODE BEGIN 2 */
92
93  /* USER CODE END 2 */
94
95  /* Infinite loop */
96  /* USER CODE BEGIN WHILE */
97  while (1)
98  {
99      /* USER CODE END WHILE */
100
101     /* USER CODE BEGIN 3 */
102     HAL_Delay(500);
103     HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
104
105     /* USER CODE END 3 */
106 }
107
108 /**
109 * @brief System Clock Configuration
110 * @param None
111 */
112 void SystemClock_Config(void)
113 {
114     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
115     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
116
117     /** Configure the main internal regulator output voltage
118     */
119     HAL_RCC_PWR_CLK_ENABLE();
120     HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
121
122     /* Initializes the CPU, AHB and APB busses clocks
123     */

```

A red box highlights the line of code where a breakpoint is set: `HAL_Delay(500);`. The line number 102 is also highlighted with a pink oval containing the text 'x2'. The STM32CubeIDE status bar at the bottom indicates: 'File download complete' and 'Time elapsed during download operation: 00:00:00.683'.



基本功能 – Live Expressions

The screenshot shows the STM32CubeIDE interface. A pink circle labeled '1' highlights the 'Window' menu, specifically the 'Show View' option. A pink arrow points from this menu to a pink-bordered window labeled '2' on its top-left corner. This window is titled 'Live Expressions' and displays a table with one row:

Expression	Type	Value
uvTick	volatile uint32_t	2295

A pink circle labeled '2' is also placed on the left side of the 'Live Expressions' window. Below the window, a note in pink text reads: '藉由Live Expressions可以觀察全域變數內容的即時變化' (Using Live Expressions, you can observe the real-time changes of global variable contents). At the bottom of the slide, another note in pink text states: 'NOTE: 此功能在OpenOCD 無法執行' (Note: This function cannot be executed in OpenOCD).

基本功能 – Registers



The screenshot shows the STM32CubeIDE interface. A pink circle labeled '1' highlights the 'Registers' option under the 'Window' menu. A pink arrow labeled '2' points from the 'Registers' menu item to the 'Registers' view window, which is also highlighted with a pink border. The Registers view displays a table of general purpose and FPU registers with their current values. The table includes columns for Name, Value, and Description.

Name	Value	Description
r0	536870912	General Purpose and FPU Register...
r1	1073887232	
r2	134217728	
r3	134218229	
r4	0	
r5	0	
r6	0	
r7	537198584	
r8	0	
r9	0	
r10	0	
r11	0	
r12	0	
sp	0x2004fff8	
lr	134233655	
pc	0x8002fc4 <main+4>	
xpsr	1627389952	
d0	0	
d1	0	
d2	0	
d3	0	
d4	0	
d5	0	
d6	0	
d7	-nan(0xffffffff00000000)	
d8	-nan(0xffffffff00000000)	
d9	-nan(0xffffffff00000000)	
d10	-nan(0xffffffff00000000)	
d11	-nan(0xffffffff00000000)	
d12	-nan(0xffffffff00000000)	
fpcr	0	
PRIMASK	0	
BASEPRI	0	
FAULTMASK	0	
CONTROL	0	
MSP	537198584	
PSP	0	
s0	0	
s1	0	
s2	0	
s3	0	
s4	0	
s5	0	
s6	0	

藉由Registers視窗可以觀察
核心暫存器的內容



基本功能 – SFR

The screenshot shows the STM32CubeIDE interface. A pink circle labeled '1' highlights the 'Window' menu, which is open to show options like 'Registers', 'SFRs', and 'Memory'. A pink arrow labeled '2' points from the 'Registers' tab in the menu to the 'SFRs' tab in the main window. The main window displays the 'Registers' view, which lists various SFRs such as RNG, HASH, CRYP, DCMI, FMC, DBG, DMA2, DMA1, RCC, GPIOA, GPIOB, GPIOC, GPIOE, GPIOF, GPIOH, GPIOI, GPIOJ, GPIOK, GPIOO, GPIOQ, SPI1, SPI2, SPI3, SPI4, SPI5, SPI6, WWDG, CADC, TIM1, TIM2, and TIM3. The 'SFRs' tab is highlighted with a pink border.

藉由SFRs視窗可以
觀察週邊暫存器的內容

NOTE:
SFR = Special function Registers

基本功能 – Memory

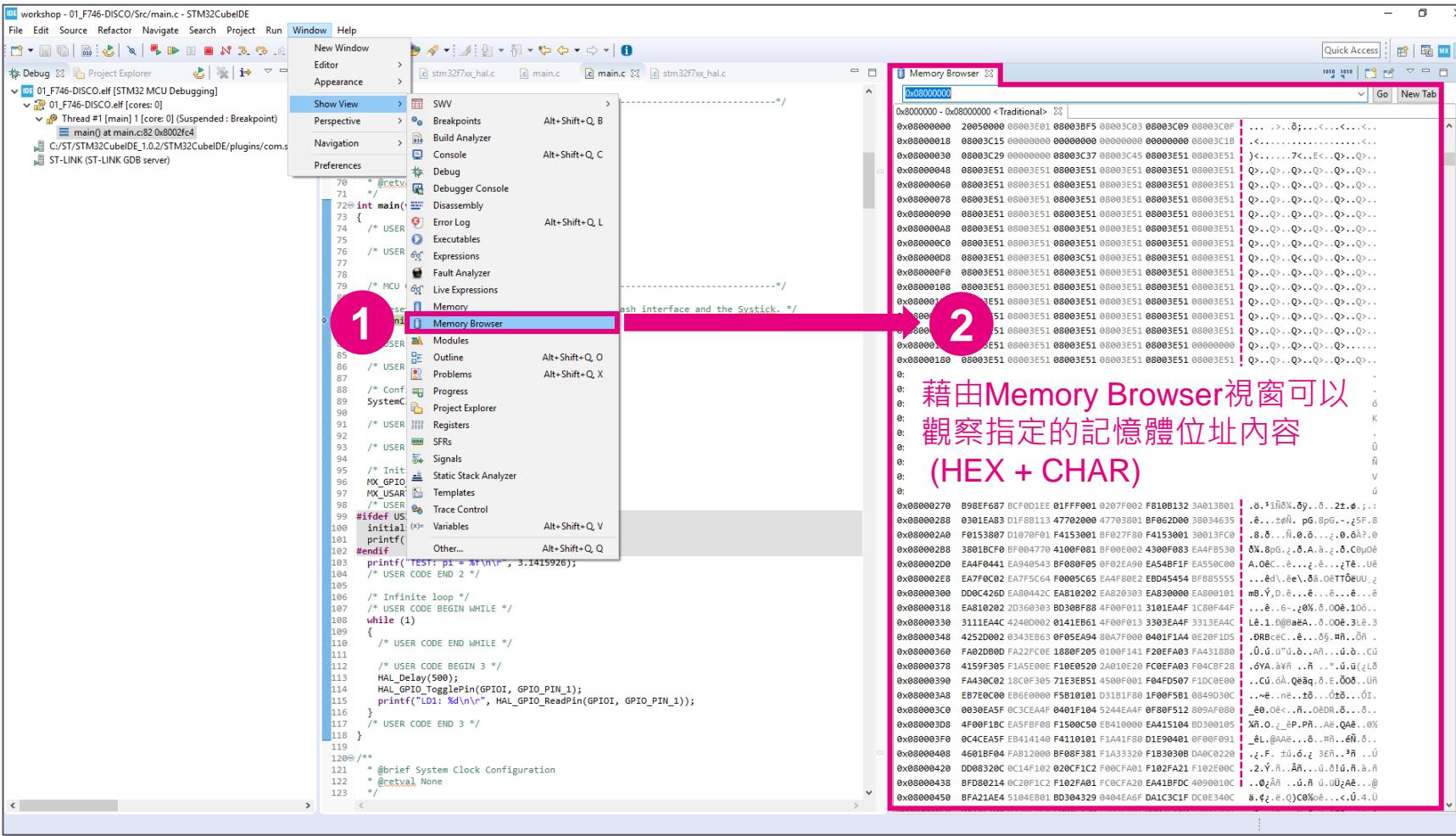


藉由Memory視窗可以觀察
指定的記憶體位址HEX內容

Address	0 - 3	4 - 7	8 - B	C - F
08000000	00000520	013E0008	F53B0008	033C0008
08000010	093C0008	0F3C0008	153C0008	00000000
08000020	00000000	00000000	00000000	183C0008
08000030	293C0008	00000000	00000000	373C0008
08000040	513E0008	513E0008	513E0008	513E0008
08000050	513E0008	513E0008	513E0008	513E0008
08000060	513E0008	513E0008	513E0008	513E0008
08000070	513E0008	513E0008	513E0008	513E0008
08000080	513E0008	513E0008	513E0008	513E0008
08000090	513E0008	513E0008	513E0008	513E0008
080000A0	513E0008	513E0008	513E0008	513E0008
080000B0	513E0008	513E0008	513E0008	513E0008
080000C0	513E0008	513E0008	513E0008	513E0008
080000D0	513E0008	513E0008	513E0008	513E0008
080000E0	513C0008	513E0008	513E0008	513E0008

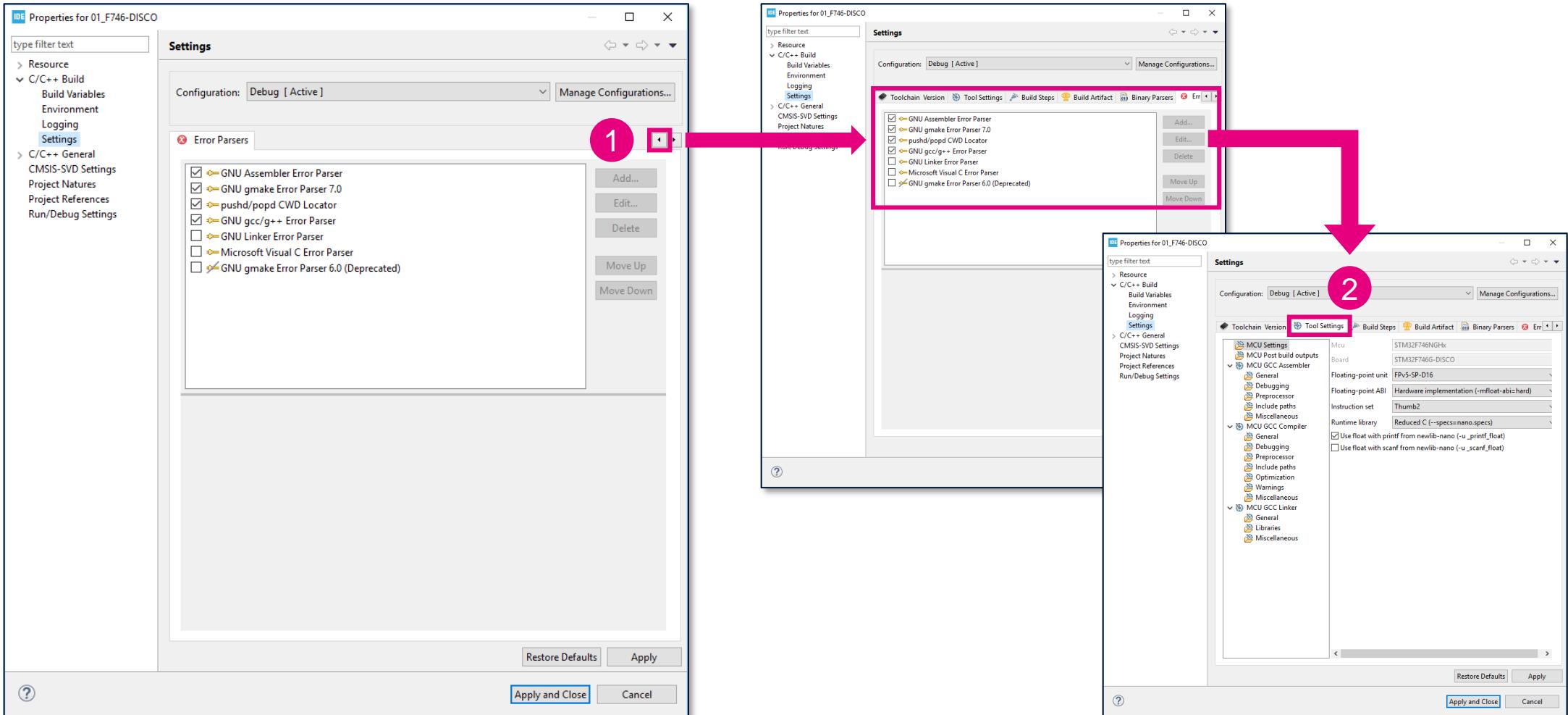


基本功能 – Memory Browser



!?

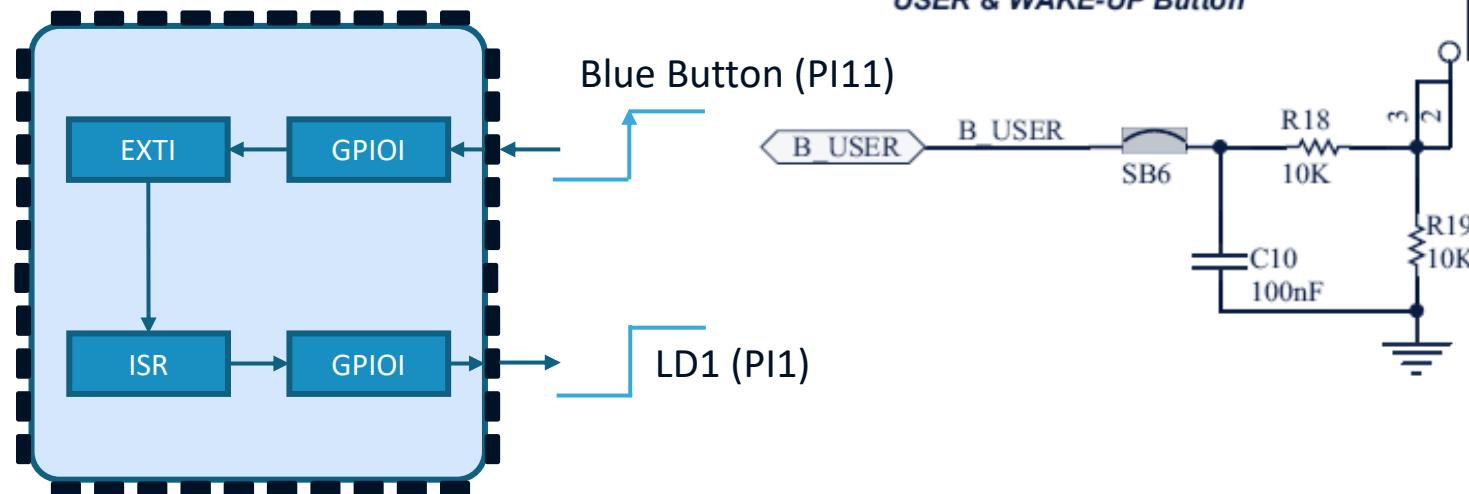
補充 – Tool Settings消失了？！

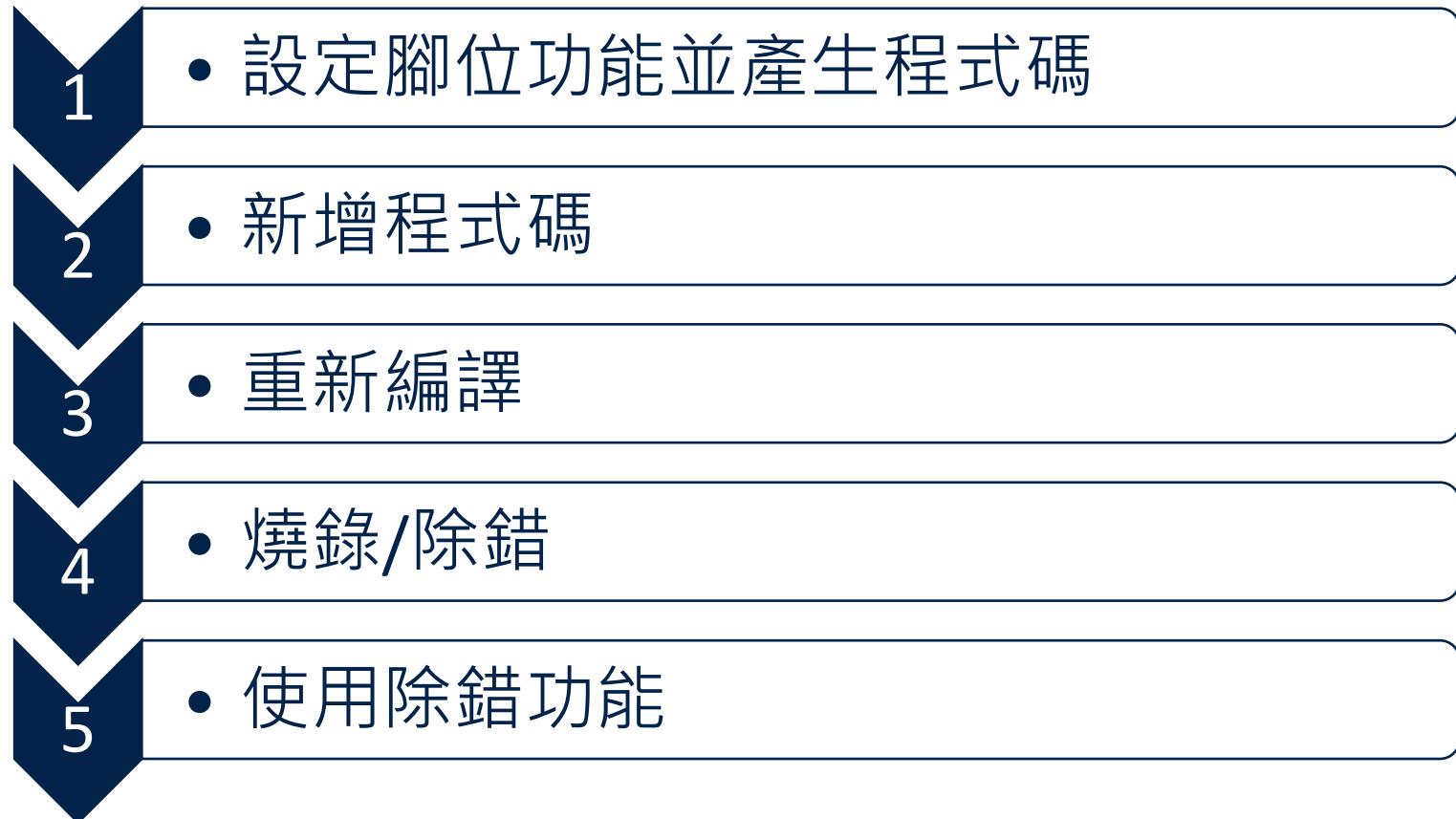


Hands-on

02_GPIO_Interrupt

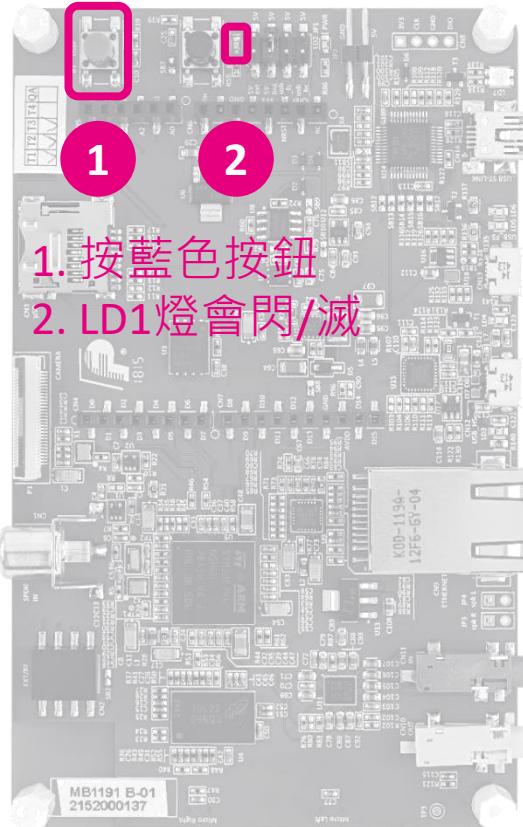
- 按下藍色按鈕後改變LD1閃滅狀態
- 同時在Debug Console看到狀態訊息



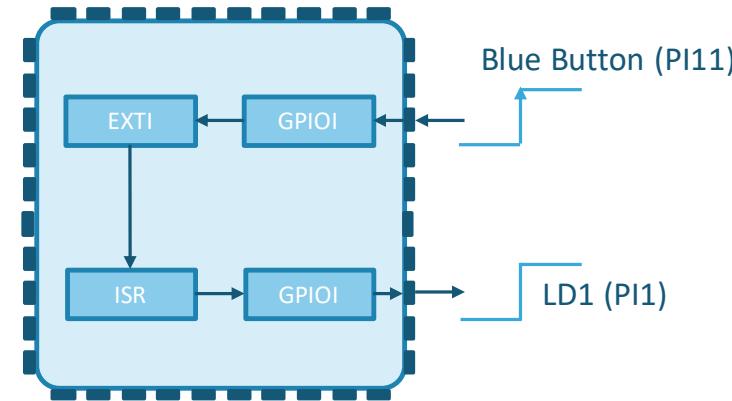


Step 4: 燒錄/除錯

中斷處理說明

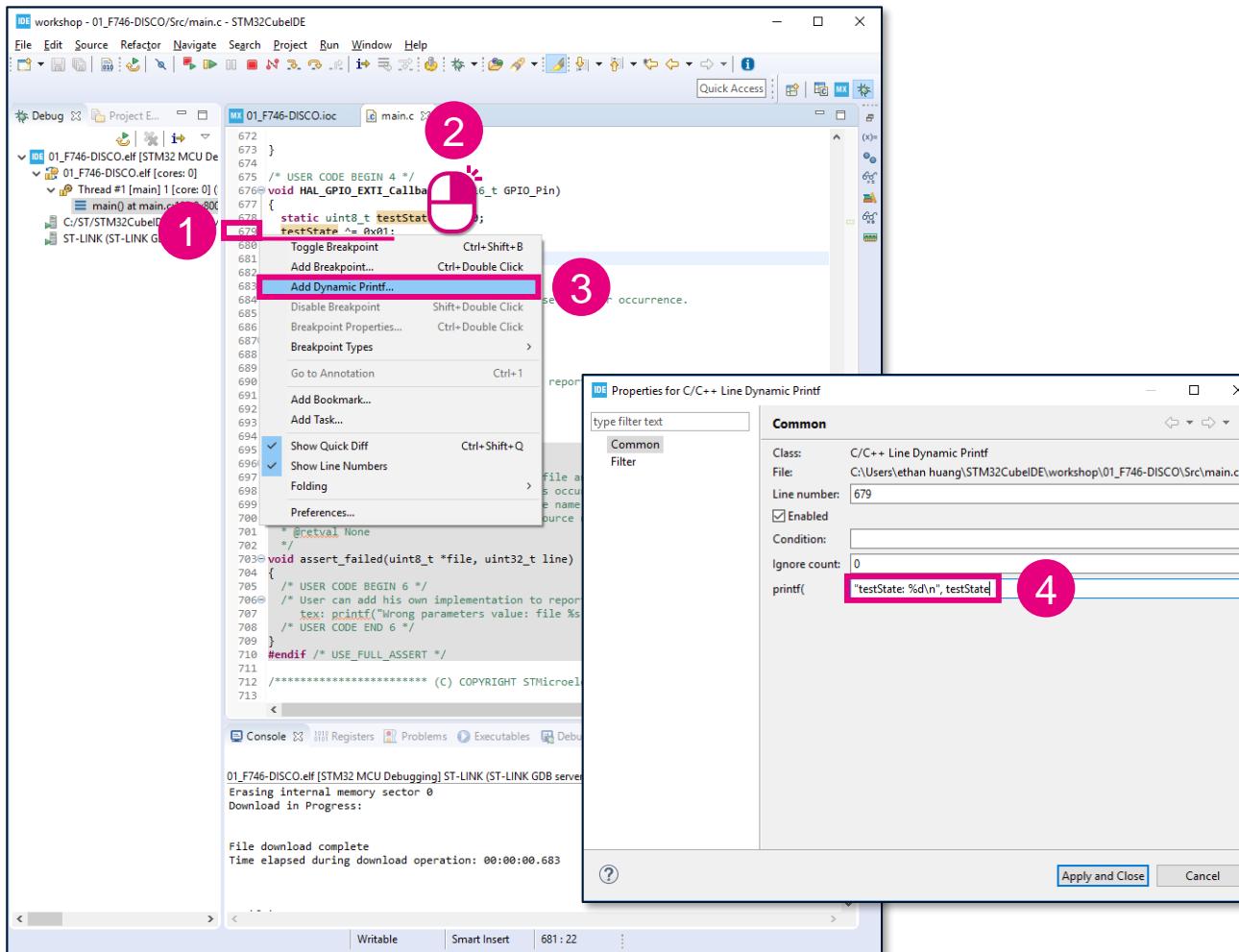


- Blue Button = PI11
- LD1 = PI1
- Blue button pressed
 - PI11 goes from HIGH to LOW,
 - PI11 rising edge detected
 - EXTI11 fires
 - EXTI15_10_IRQHandler(){...} //in stm32f7xx_it.c
 - HAL_GPIO_EXTI_IRQHandler(KEY_BUTTON_PIN) //in stm32f7xx_hal_gpio.c
 - HAL_GPIO_EXTI_Callback(){...} //in main.c
 - HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1); //in main.c





功能體驗 – 使用Dynamic Printf



```
01_F746-DISCO.elf [STM32 MCU Debugging] C:\ST\STM32CubeIDE_1.0.2\STM32CubeIDE\plugins\com.st.stm32cube.ide.mcu.external\01_F746-DISCO.elf in HAL_Delay (Delay=500) at ...\\Drivers\\STM32F7xx_HAL_Driver\\Src\\stm32f7xx_hal.c:366
while ((HAL_GetTick() - tickstart) < wait)
Breakpoint 7, main () at ../../Src/main.c:103
103      HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
testState: 0
testState: 1
testState: 0
testState: 1
```

Hands-on

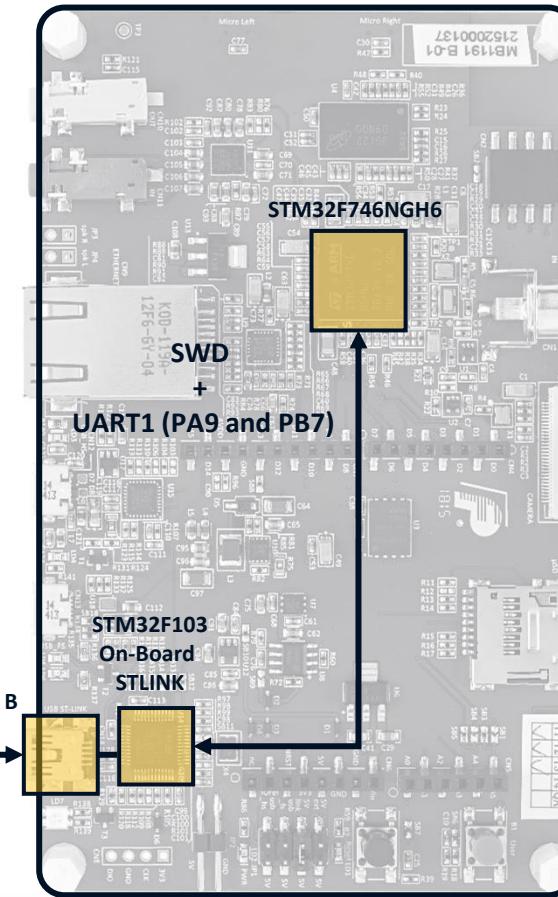
03_UART_printf

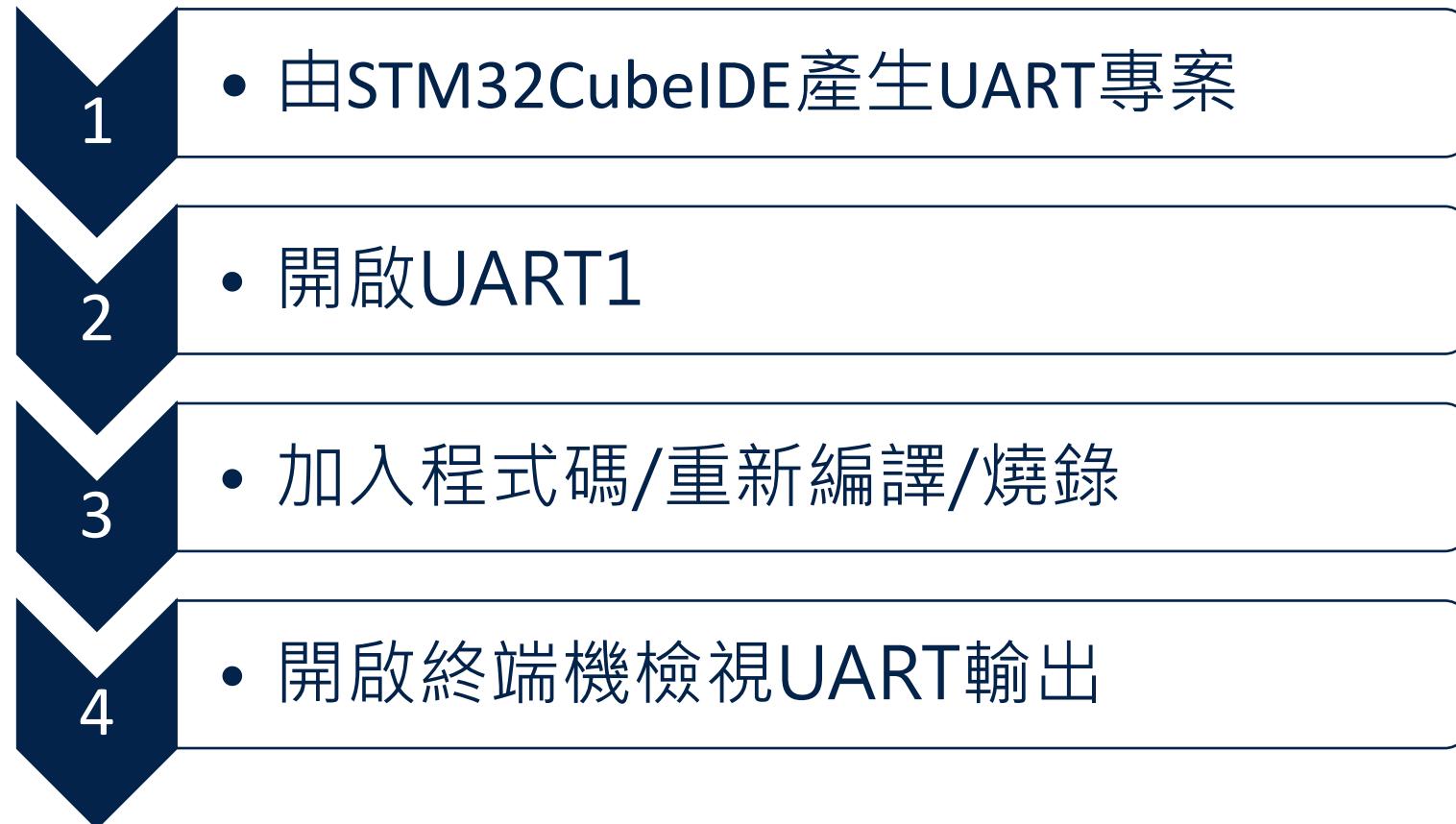
實作目標

每500ms改變LD1閃滅狀態，
並透過終端機顯示LD1狀態



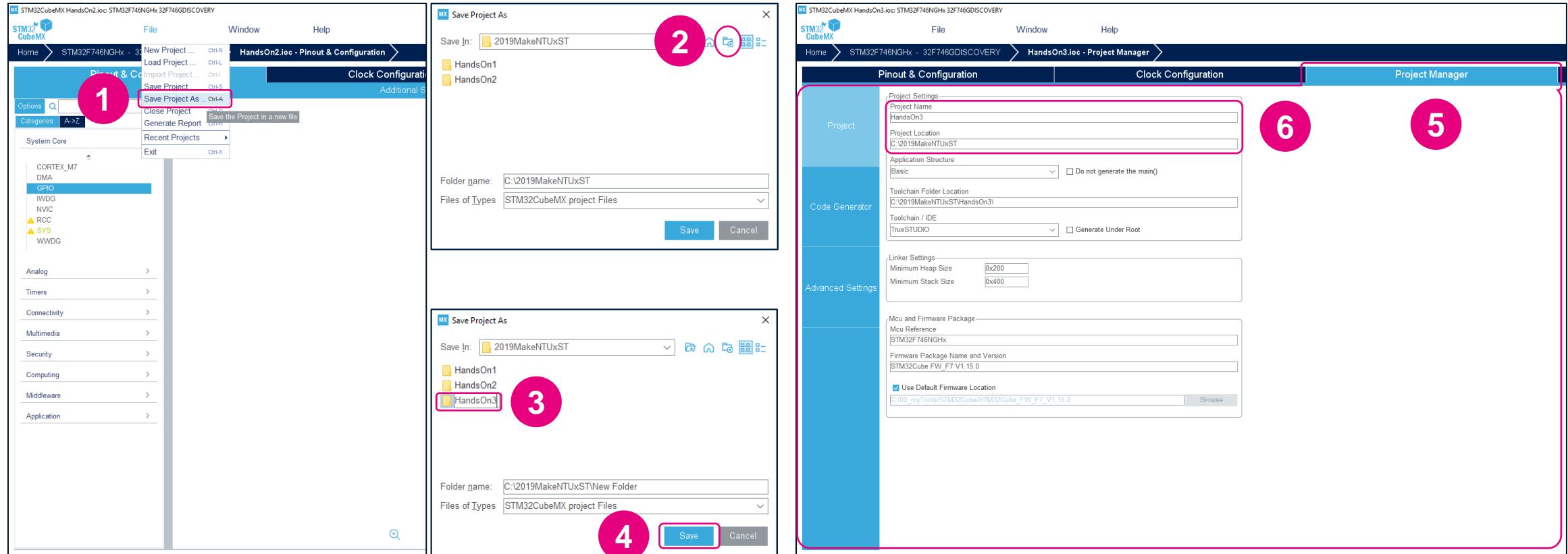
USB





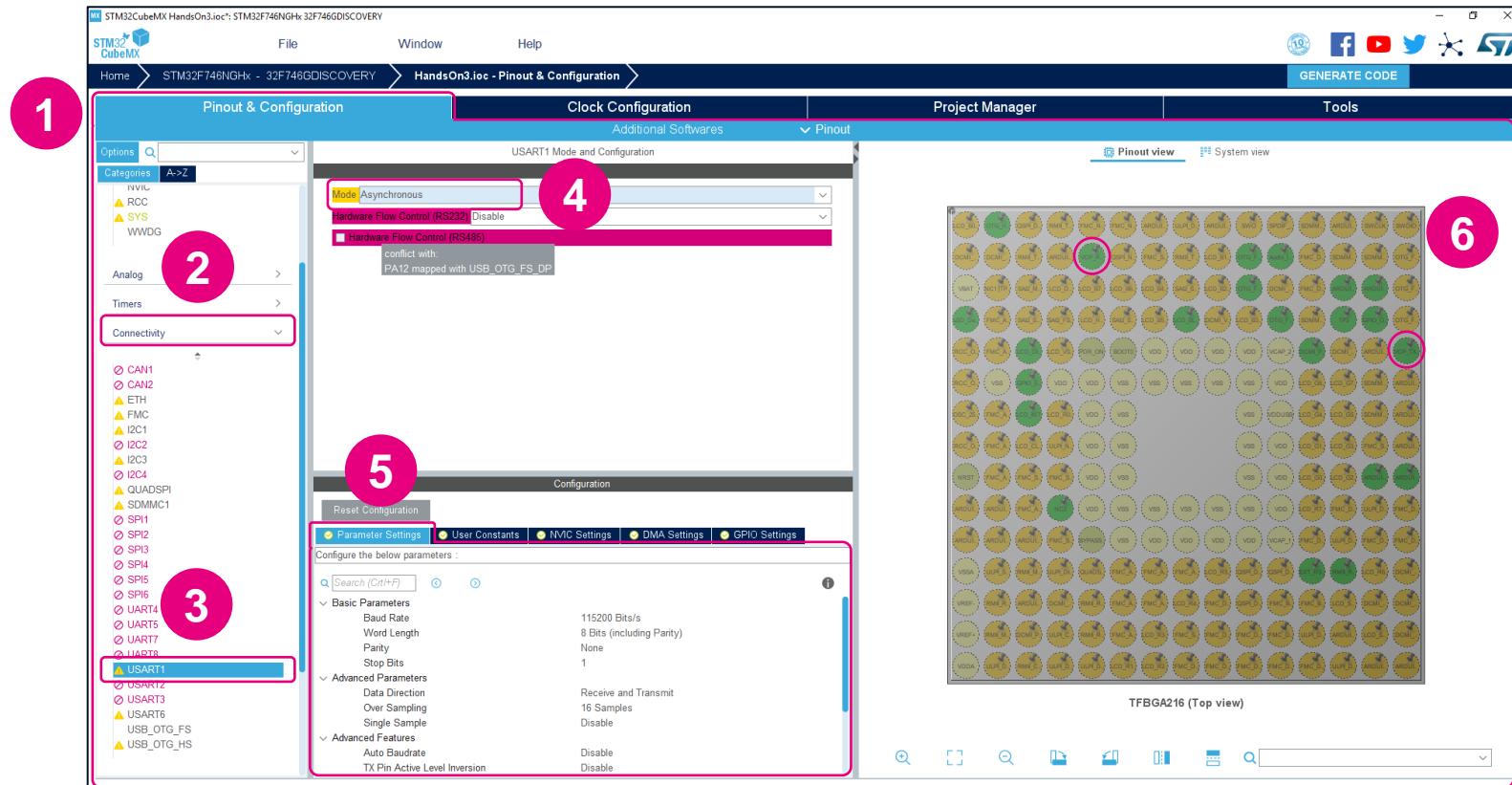
Step 1: 另存CUBEMX專案

由HandsOn 2另存CUBEMX Project



Step 2: 開啟UART1

於Pinout & Configuration分頁中的Connectivity項目開啟USART1



Step 3: 加入程式碼/重新編譯/燒錄

於main.c加入程式碼

STM32_workspace_9.3 - C/C++ - HandsOn3/Application/User/main.c - Atollic TrueSTUDIO for STM32

```
File Edit Source Refactor View Navigate Search Project Run Window Help
```

Project Explorer

```
main.c
```

1

```
699 GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
700 GPIO_InitStruct.Pull = GPIO_NOPULL;
701 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
702 GPIO_InitStruct.Alternate = GPIO_AF5_SPI2;
703 HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
704
705 /* EXTI interrupt init*/
706 HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
707 HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
708
709 }
710
711 /* USER CODE BEGIN 4 */
712 int __io_putchar(int ch)
713 {
714     HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
715     return ch;
716 }
717
718 /* USER CODE END 4 */
```

```
int __io_putchar(int ch)
{
    HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
    return ch;
}
```

STM32_workspace_9.3 - C/C++ - HandsOn3/Application/User/main.c - Atollic TrueSTUDIO for STM32

```
File Edit Source Refactor View Navigate Search Project Run Window Help
```

Project Explorer

```
main.c
```

2

```
10 * All rights reserved.</center></h2>
11
12 * This software component is licensed by ST under BSD 3-Clause license,
13 * the "License"; You may not use this file except in compliance with the
14 * License. You may obtain a copy of the License at:
15 *           opensource.org/licenses/BSD-3-Clause
16
17 ****
18 */
19 /* USER CODE END Header */
20
21 /* Includes -----
22 #include "main.h"
23
24 /* Private includes -
25 /* USER CODE BEGIN Incl */
26 #include <stdio.h>
27 /* USER CODE END Incl
```

```
#include <stdio.h>
```

3

```
92 MX_GPIO_Init();
93 MX_USART1_UART_Init();
94 /* USER CODE BEGIN 2 */
95
96 /* USER CODE END 2 */
97
98 HAL_Delay(500);
99 HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
100 printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));
101
102 /* USER CODE BEGIN 3 */
103 HAL_Delay(500);
104 HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
105 printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));
106
107 /* USER CODE END 3 */
108
109 /* USER CODE END 3 */
110 }
```

```
HAL_Delay(500);
HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));
```

Step 3: 加入程式碼/重新編譯/燒錄

於main.c加入程式碼

The image shows two side-by-side screenshots of the Atollic TrueSTUDIO IDE. Both windows have the title "STM32_workspace_9.3 - C/C++ - HandsOn3/Application/User/main.c - Atollic TrueSTUDIO for STM32".

Left Window (Annotation 1): Shows the beginning of the main.c file. A pink box highlights the line "#include <main.h>".

```
699 GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
700 GPIO_InitStruct.Pull = GPIO_NOPULL;
701 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
702 GPIO_InitStruct.Alternate = GPIO_AF5_SPI2;
703 HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
704
705 /* EXTI interrupt init*/
706 HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
707 HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
708
709 }
710
711 /* USER CODE BEGIN 4 */
712 int __io_putchar(int ch)
713 {
714     HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
715     return ch;
716 }
717
718 /* USER CODE END 4 */
```

Left Window (Annotation 2): Shows the implementation of the __io_putchar function. A pink box highlights the function body.

```
int __io_putchar(int ch)
{
    HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
    return ch;
}
```

Right Window (Annotation 3): Shows the end of the main.c file. A pink box highlights the inclusion of stdio.h.

```
10 * All rights reserved.</center></h2>
11
12 * This software component is licensed by ST under BSD 3-Clause license,
13 * the "License"; You may not use this file except in compliance with the
14 * License. You may obtain a copy of the License at:
15 * opensource.org/licenses/BSD-3-Clause
16
17 ****
18 */
19 /* USER CODE END Header */
20
21 /* Includes -----
22 #include "main.h"
23
24 /* Private includes -
25 /* USER CODE BEGIN Incl */
26 #include <stdio.h>
27 /* USER CODE END Incl */

92 MX_GPIO_Init();
93 MX_USART1_UART_Init();
94 /* USER CODE BEGIN 2 */
95
96 /* USER CODE END 2 */

HAL_Delay(500);
HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));

104 /* USER CODE BEGIN 3 */
105 HAL_Delay(500);
106 HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
107 printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));
108
109 /* USER CODE END 3 */
110 }
```

Right Window (Annotation 4): Shows the addition of a HAL_Delay(500) call and a printf statement. A pink box highlights the new code.

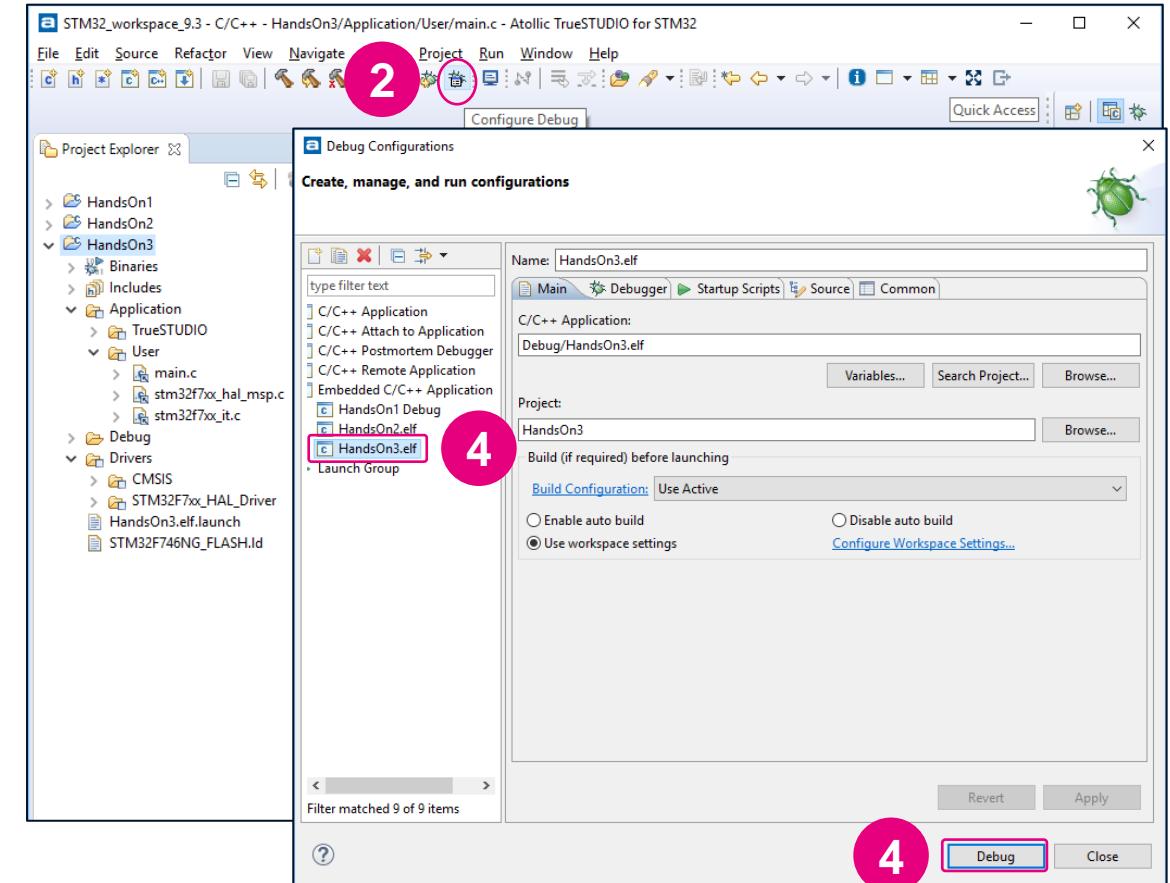
```
HAL_Delay(500);
HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));
```

Step 3: 加入程式碼/重新編譯/燒錄

重新編譯/燒錄

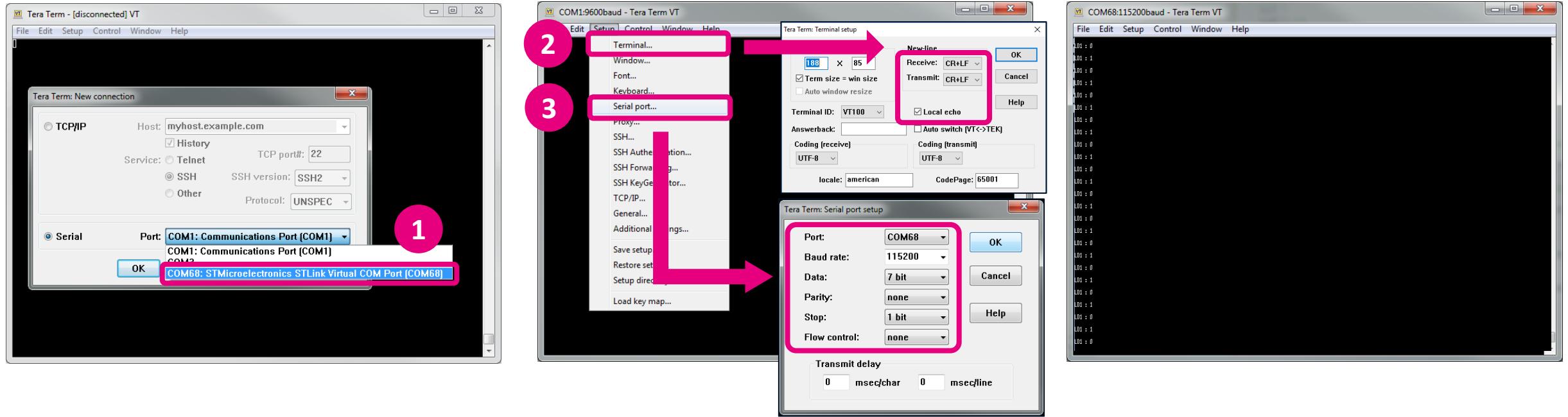
The screenshot shows the Atollic TrueSTUDIO interface. On the left is the Project Explorer with three projects: HandsOn1, HandsOn2, and HandsOn3. The HandsOn3 project is selected and highlighted with a red circle labeled '1'. In the center is the main.c code editor. A right-click context menu is open over the code, with the 'Build Project' option highlighted with a red rectangle and a red circle labeled '2'. The code itself contains several lines of C code related to GPIO initialization.

```
689 // Configure GPIO pin : ARDUINO_PWM_D6_Pin
690 GPIO_InitStruct.Pin = ARDUINO_PWM_D6_Pin;
691 GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
692 GPIO_InitStruct.Pull = GPIO_NOPULL;
693 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
694
695 // Configure GPIO pin : ARDUINO_MISO_D12_Pin ARDUINO_MOSI_PWM_D11_Pin */
696 Pin = ARDUINO_MISO_D12_Pin|ARDUINO_MOSI_PWM_D11_Pin;
697 Mode = GPIO_MODE_AF_PP;
698 Pull = GPIO_NOPULL;
699 Speed = GPIO_SPEED_FREQ_LOW;
700 Alternate = GPIO_AF5_SPI2;
701 IOB, &GPIO_InitStruct);
702
703 /* init*/
704 priority(EXTI15_10_IRQn, 0, 0);
705 RQ(EXTI15_10_IRQn);
```



Step 4: 開啟終端機檢視UART輸出

以TeraTerm為例

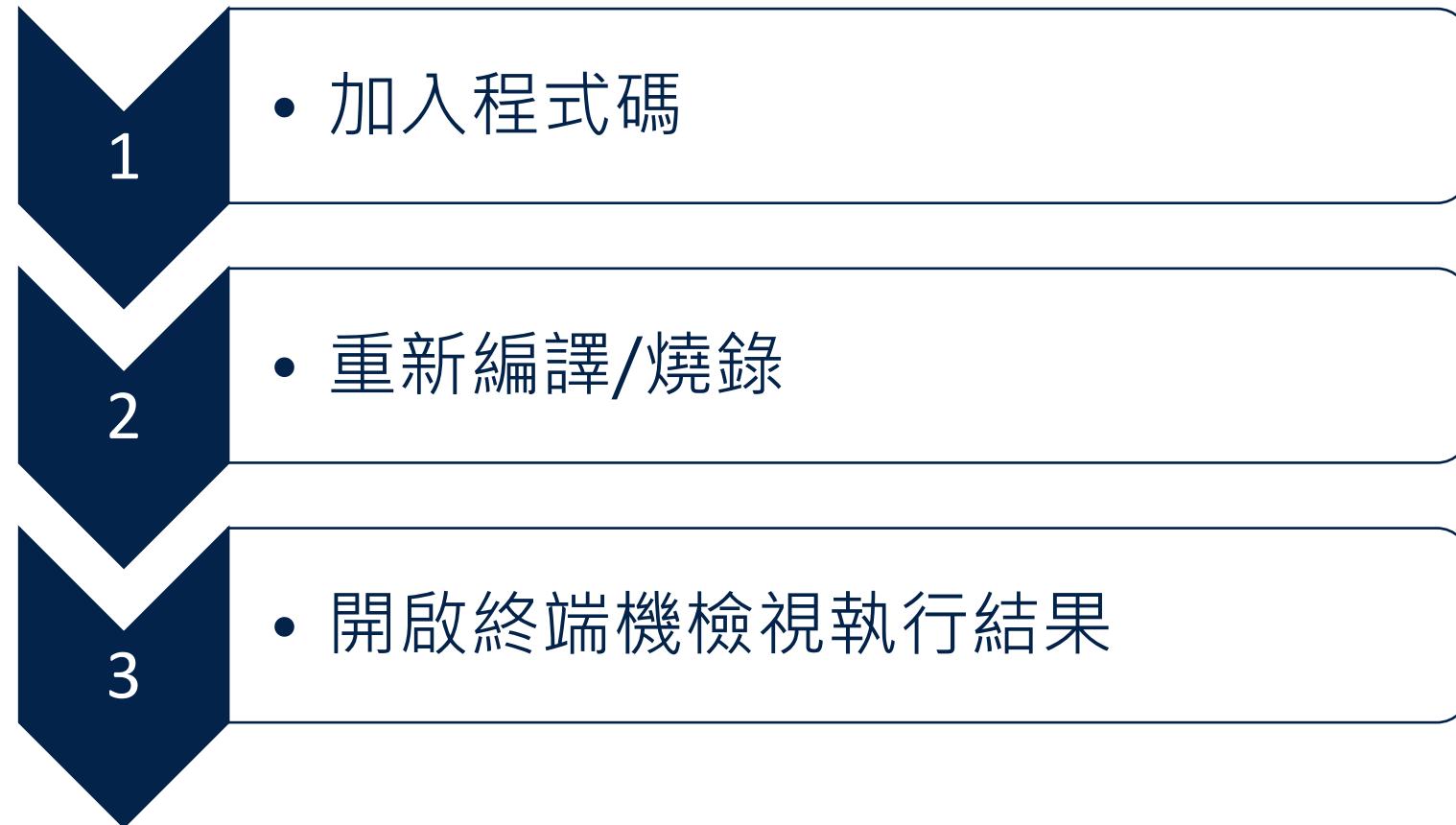


本次工作坊範例使用的終端機為Tera-Term，可由以下網址下載：

<https://ttssh2.osdn.jp/index.html.en>

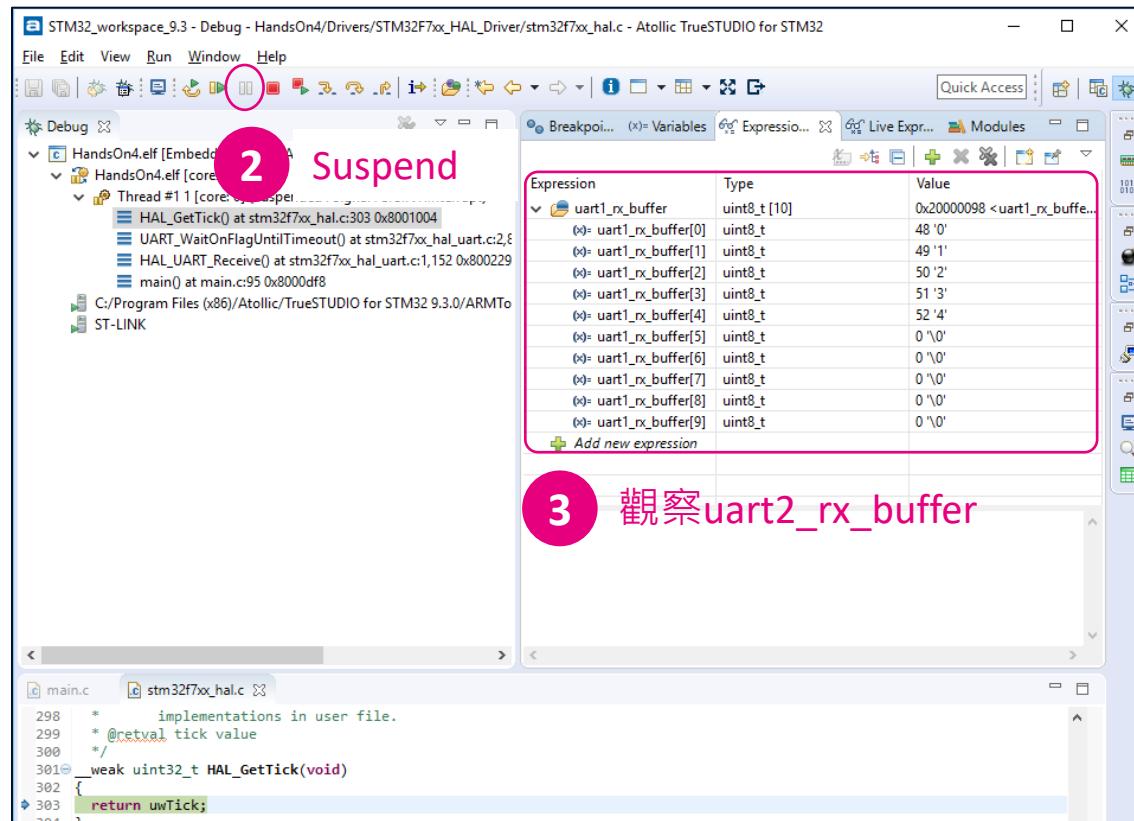
Hands-on

04_UART_Receiving_Polling



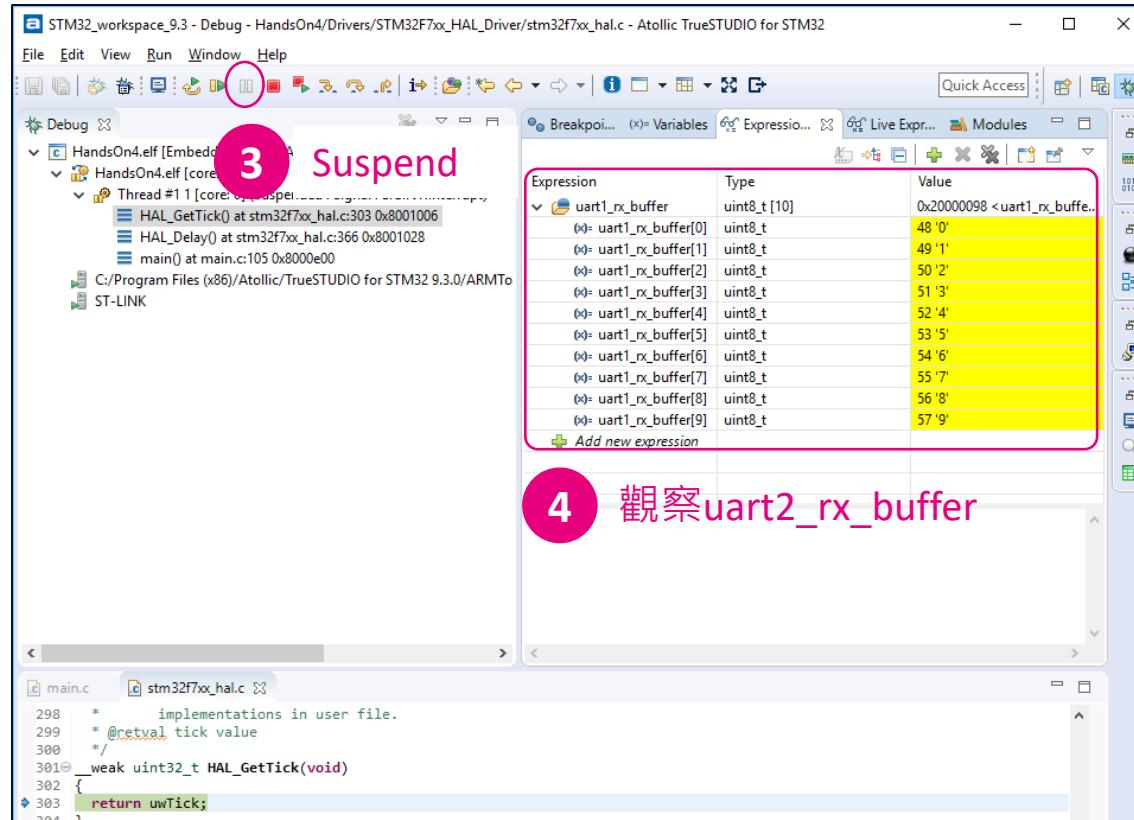
Step 3: 開啟終端機檢視執行結果

在終端機下打字，並觀察uart1_rx_buffer



Step 3: 開啟終端機檢視執行結果

在終端機下打字，並觀察uart1_rx_buffer

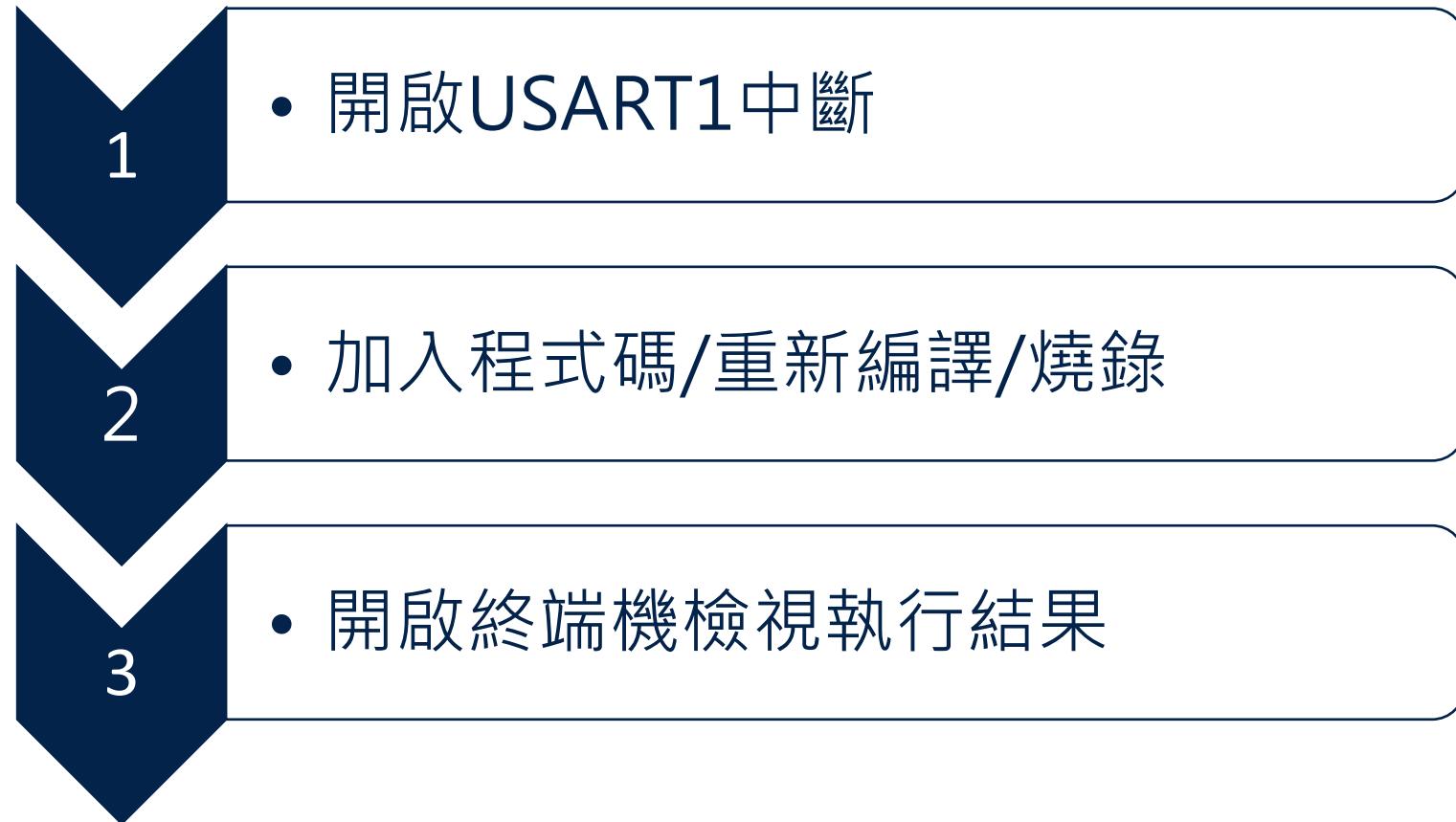


The terminal window shows the following sequence of characters: '0123456789' followed by '01 : 1'. This is followed by a series of 'LD1 : 1' outputs, indicating the state of the LD1 pin. The terminal window has a red circle around the input text and another red circle around the output pattern.

```
0123456789 01 : 1
LD1 : 0
LD1 : 1
```

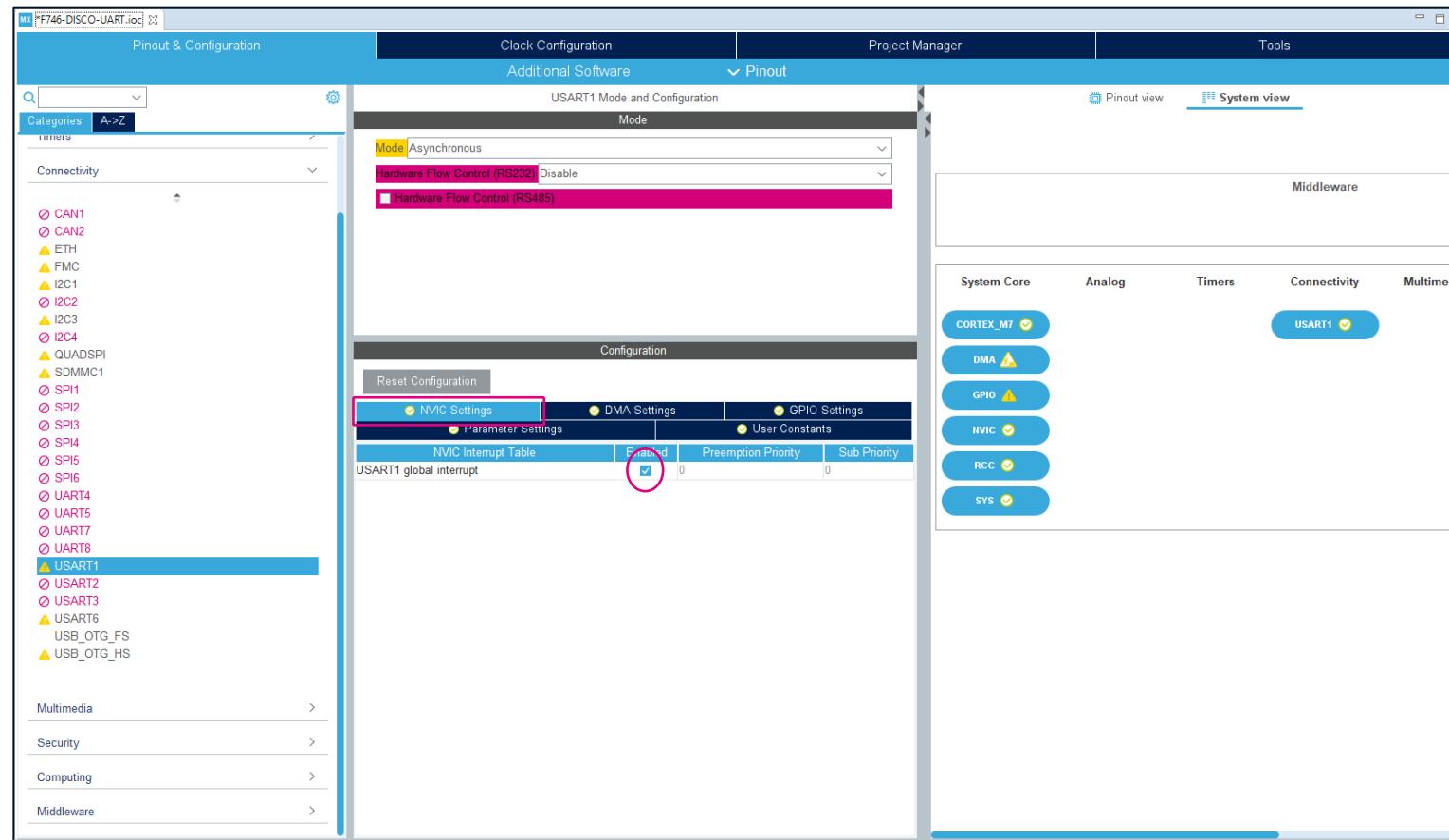
Hands-on

05_UART_Receiving_IT



Step 1: 開啟USART1中斷

開啟USART1中斷



Step 3: 開啟終端機檢視執行結果

在終端機下打字，並觀察uart1_rx_buffer

The screenshot shows the Atollic TrueSTUDIO for STM32 interface with two main windows:

- Left Window (Atollic TrueSTUDIO):** Shows the project tree with "HandsOn5.elf [Embedded]" selected. A pink circle labeled "2 Suspend" highlights the "Suspend" button in the toolbar. Below it, the variable viewer displays the "uart1_rx_buffer" array with values: A, B, C, D, E, 0, 0, 0, 0, 0.
- Right Window (Tera Term VT):** Shows the terminal window titled "COM43 - Tera Term VT". A pink circle labeled "1 在終端機下打字" highlights the terminal window. The terminal output shows the characters A, B, C, D, E being printed sequentially.

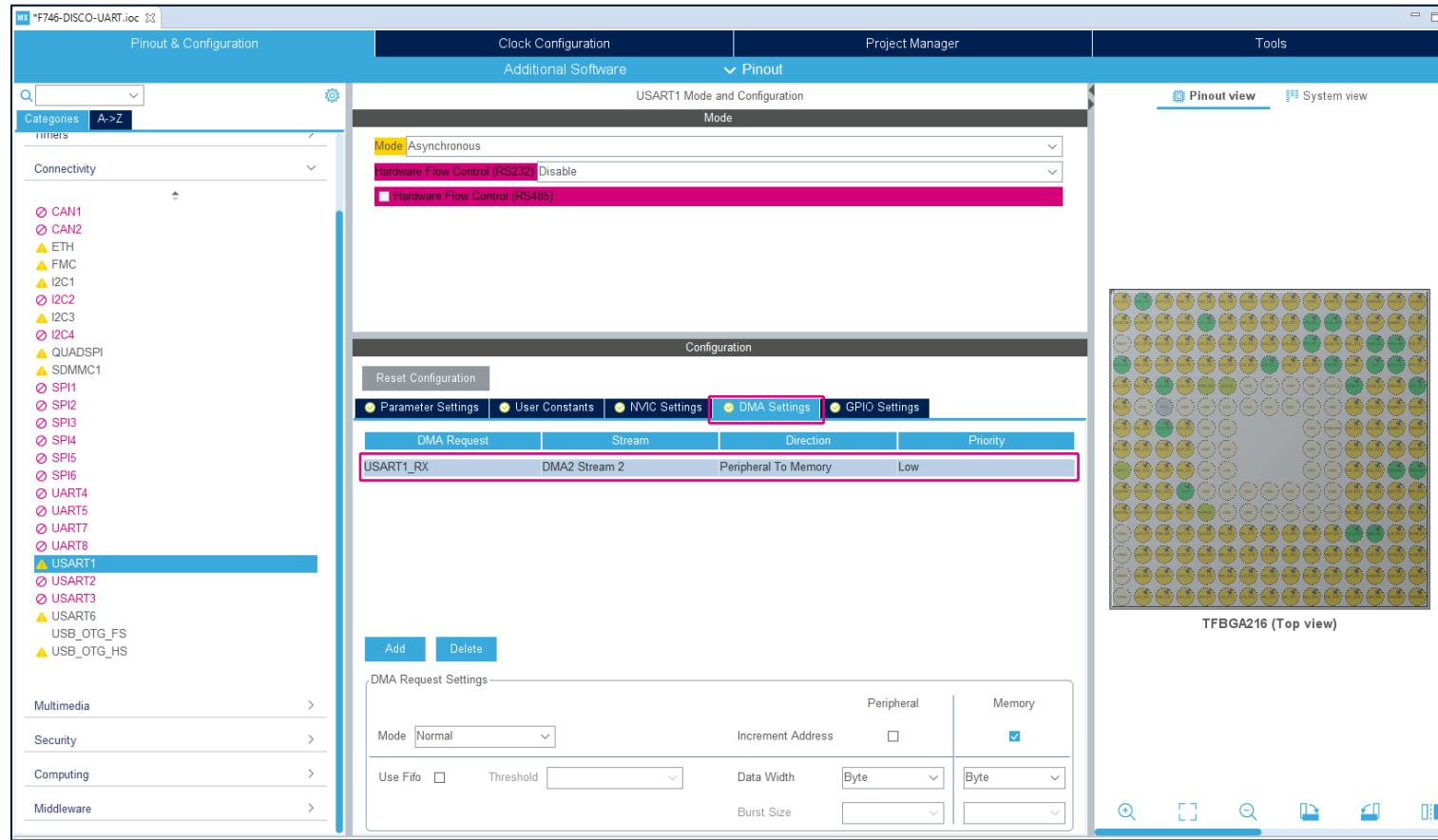
Hands-on

06_UART_Receiving_DMA



Step 1: 開啟USART1 DMA

開啟USART1 DMA



Step 3: 開啟終端機檢視執行結果

在終端機下打字，並觀察uart1_rx_buffer

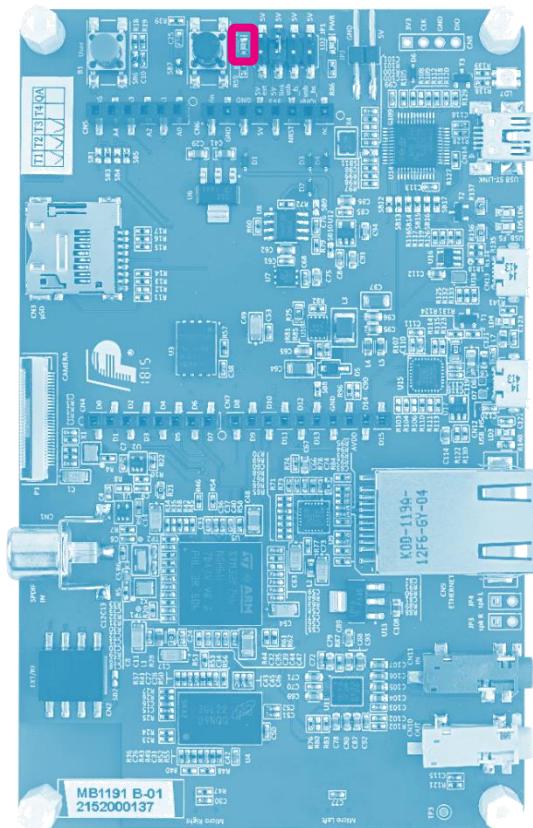
The image shows two windows side-by-side. On the left is the Atollic TrueSTUDIO IDE interface. A pink circle labeled '2 Suspend' highlights the 'Suspend' button in the toolbar. Below it, the 'Variables' tab of the debugger is open, showing a list of variables under the expression 'uart1_rx_buffer'. The first five entries (indices 0 to 4) have their values highlighted in yellow: '65 'A'', '66 'B'', '67 'C'', '68 'D'', and '69 'E''. A pink circle labeled '3 觀察uart2_rx_buffer' points to this list. At the bottom of the IDE, code snippets from main.c and stm32f7xx_hal.c are visible. On the right is a terminal window titled 'COM43 - Tera Term VT'. A pink circle labeled '1 在終端機下打字' points to the window. The terminal displays a series of alternating '0's and '1's and letters 'A', 'B', 'C', 'D', and 'E' on separate lines, representing the received data.

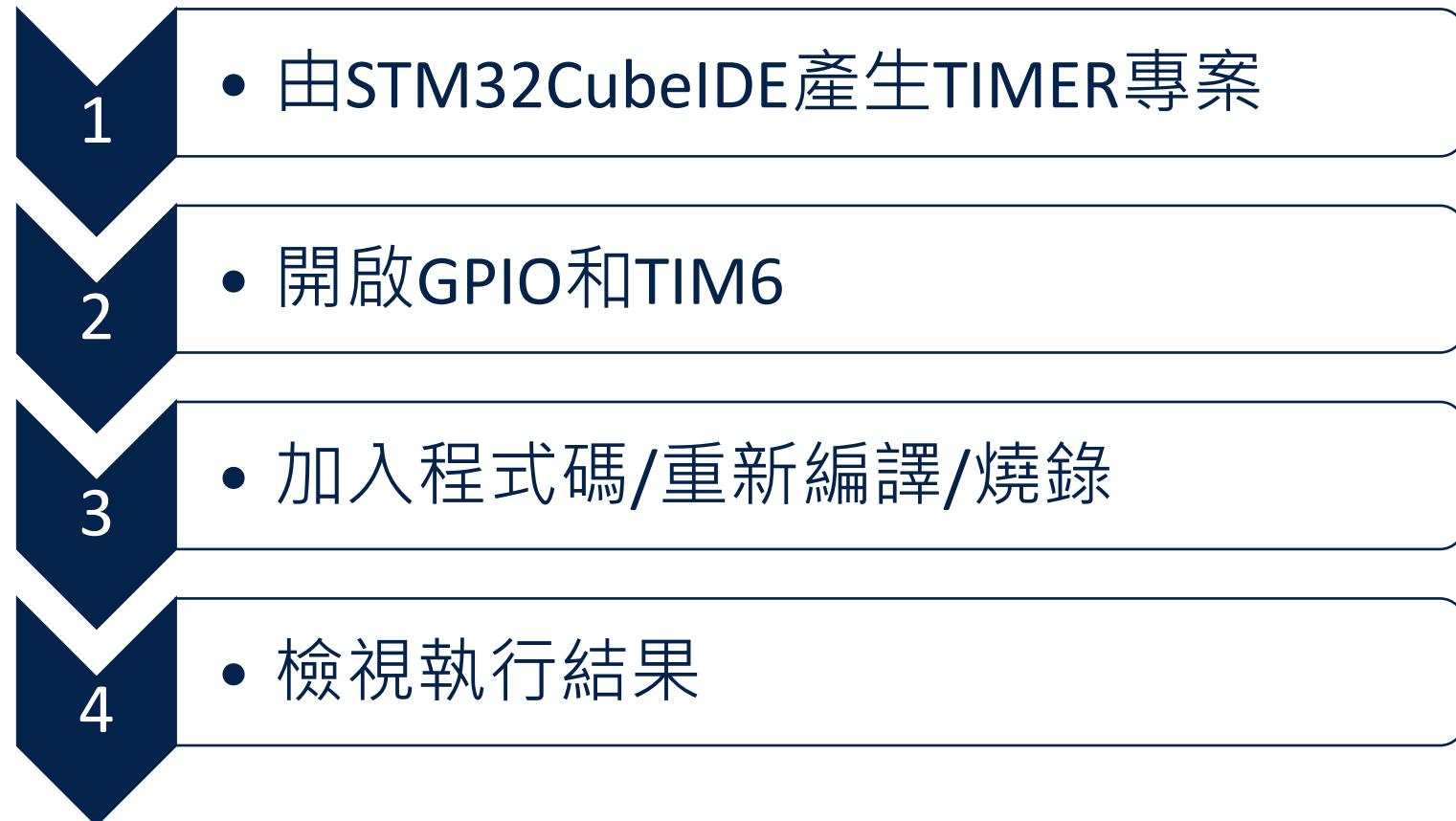
Hands-on

07_TIMER

程式碼實作目標

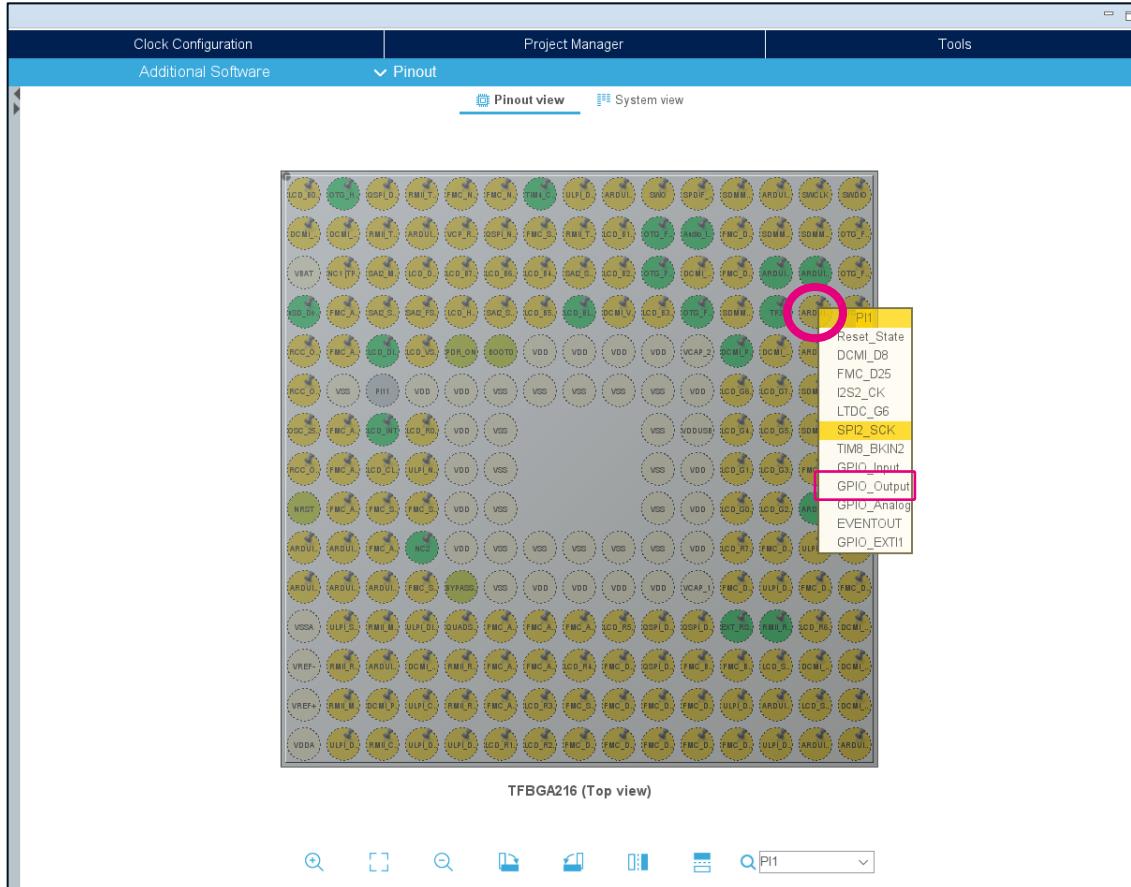
每500ms變LD1閃滅狀態



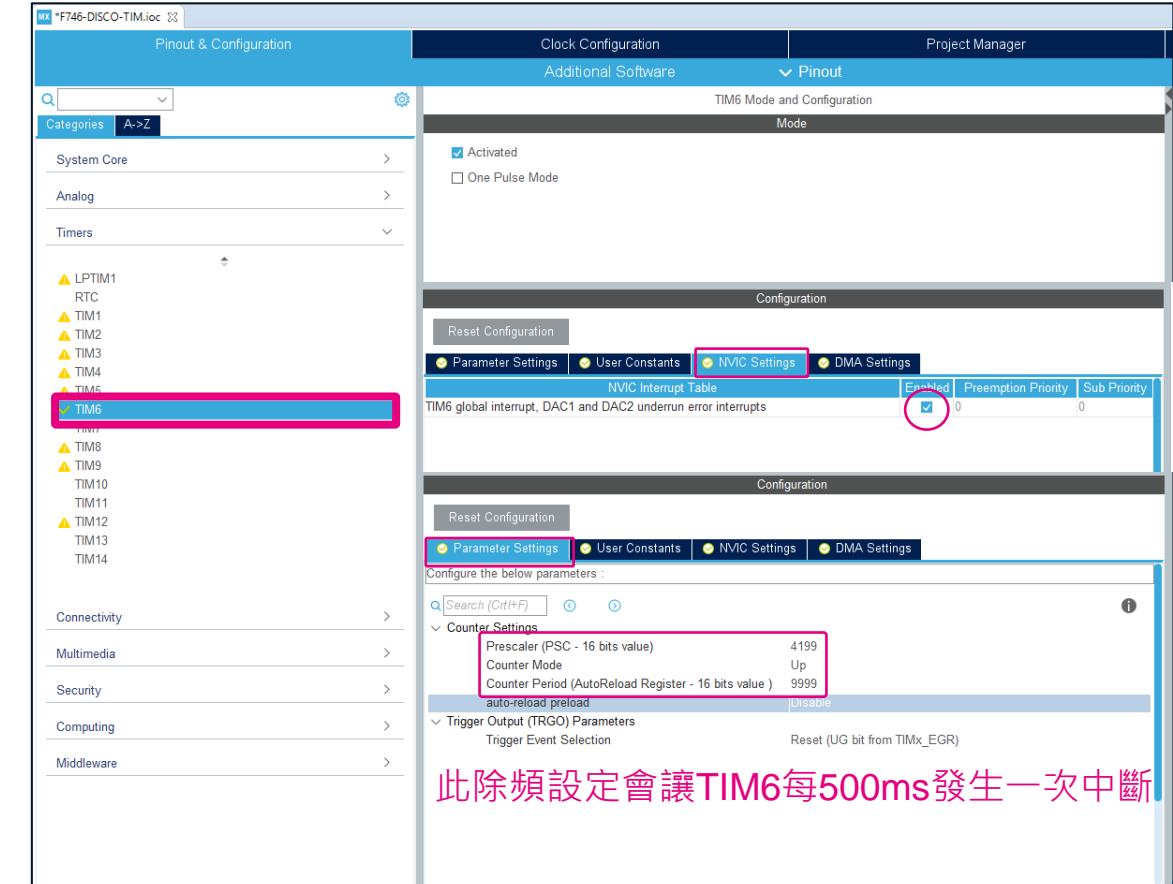


Step 2: 開啟GPIO和TIM6

PI1 = GPIO-Output

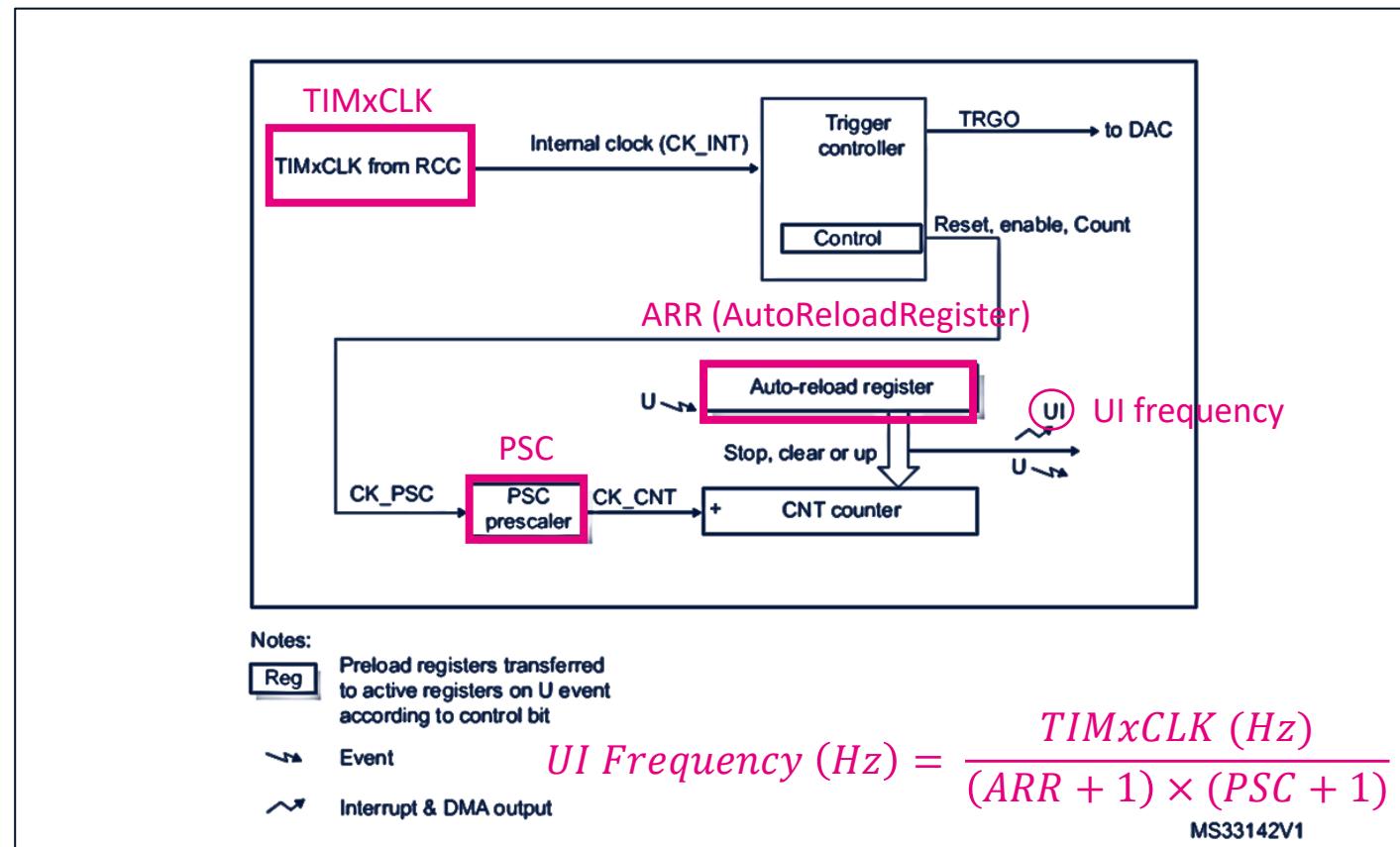


TIM6



Timer功能方塊示意圖

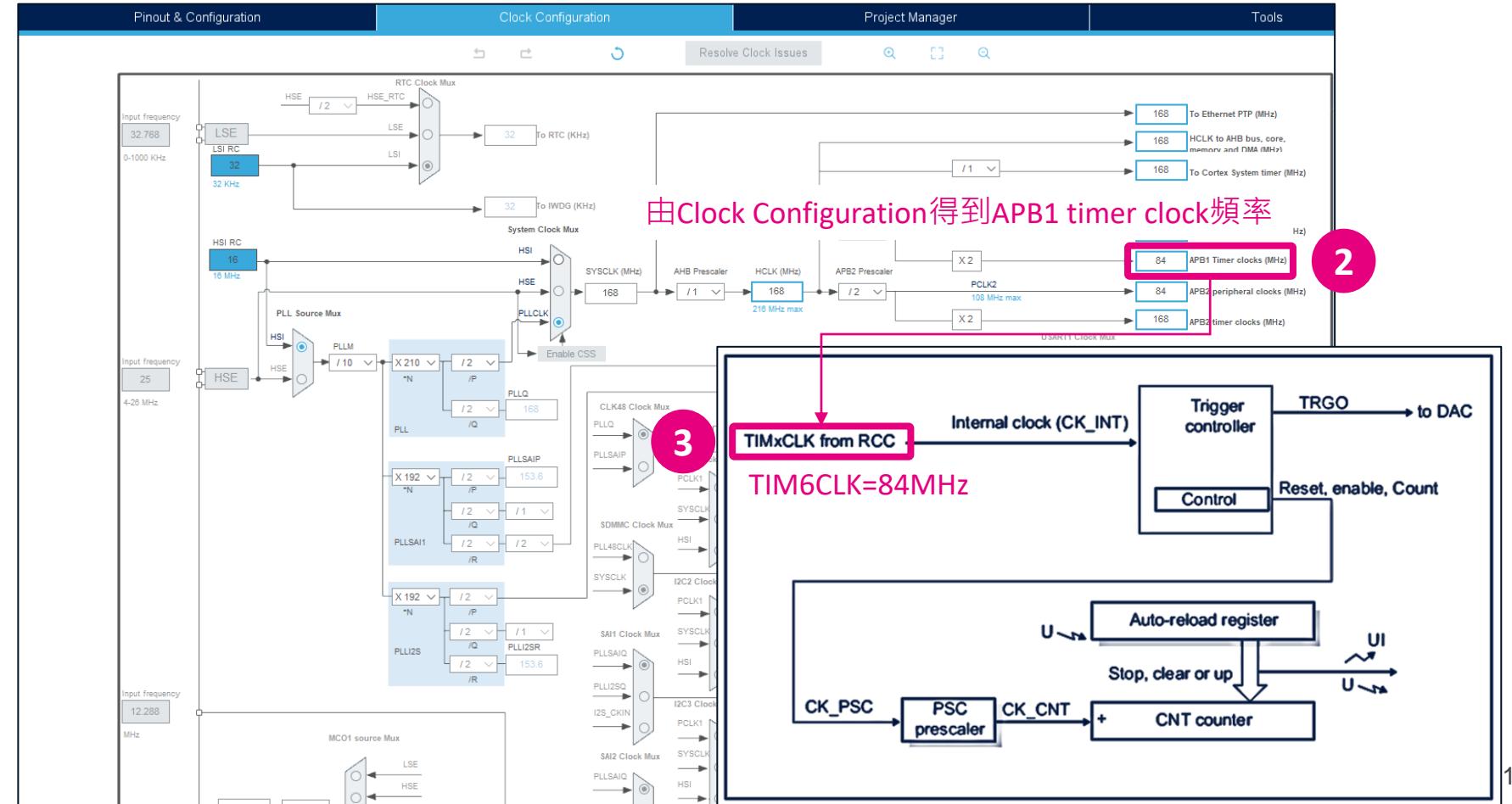
以Basic Timer (TIM6)為例



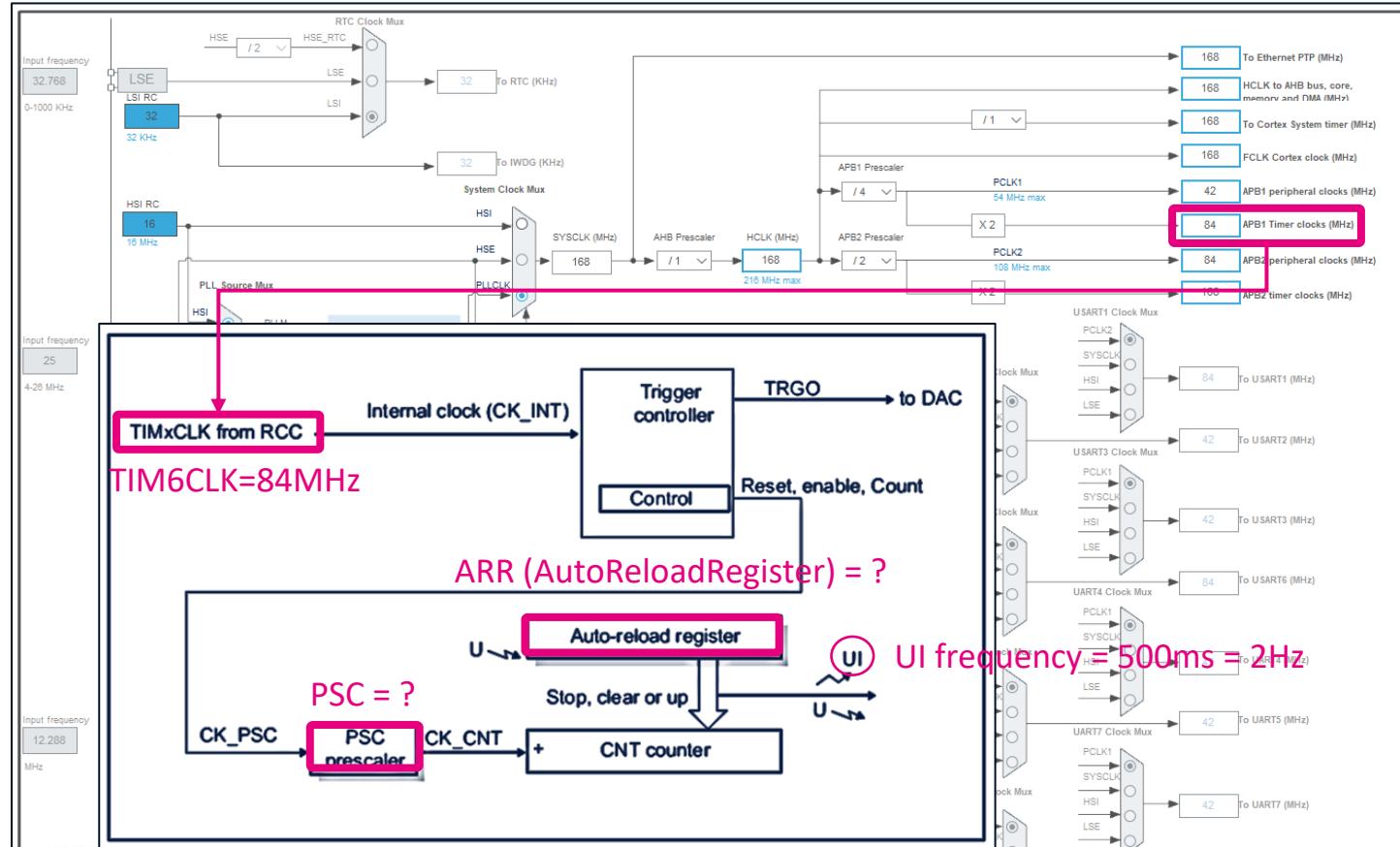
TIMxCLK = ?

查閱RM0385

RM0385			
Boundary address	Peripheral	Bus	Register map
0x4000 7C00 - 0x4000 7FFF	UART8		Section 31.8.12: USART register map on page 1048
0x4000 7800 - 0x4000 7BFF	UART7		Section 16.5.15: DAC register map on page 479
0x4000 7400 - 0x4000 77FF	DAC		Section 4.4.4: PWR power control register 2 (PWR_CSR2) on page 124
0x4000 7000 - 0x4000 73FF	PWR		Section 39.7.7: HDMI-CEC register map on page 1620
0x4000 6C00 - 0x4000 6FFF	HDMI-CEC		Section 36.9.5: bxCAN register map on page 1292
0x4000 6800 - 0x4000 6BFF	CAN2		Section 30.7.12: I2C register map on page 984
0x4000 6400 - 0x4000 67FF	CAN1		Section 30.7.12: I2C register map on page 984
0x4000 6000 - 0x4000 63FF	I2C4		Section 30.7.12: I2C register map on page 984
0x4000 5C00 - 0x4000 5FFF	I2C3		Section 30.7.12: I2C register map on page 984
0x4000 5800 - 0x4000 5BFF	I2C2		Section 30.7.12: I2C register map on page 984
0x4000 5400 - 0x4000 57FF	I2C1		Section 30.7.12: I2C register map on page 984
0x4000 5000 - 0x4000 53FF	UART5		Section 31.8.12: USART register map on page 1048
0x4000 4C00 - 0x4000 4FFF	UART4		Section 32.9.10: SPI/I2S register map on page 1109
0x4000 4800 - 0x4000 4BFF	USART3		Section 24.5.12: TIM10/TIM11/TIM13/TIM14 register map on page 820
0x4000 4400 - 0x4000 47FF	USART2		Section 24.4.13: TIM9/TIM12 register map on page 810
0x4000 4000 - 0x4000 43FF	SPDIFRX		Section 25.4.9: TIM6/TIM7 register map on page 835
0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3		Section 23.4.21: TIMx register map on page 771
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2		
0x4000 3000 - 0x4000 33FF	IWDG		
0x4000 2C00 - 0x4000 2FFF	WWDG		
0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers		
0x4000 2400 - 0x4000 27FF	LPTIM1		
0x4000 2000 - 0x4000 23FF	TIM14		
0x4000 1C00 - 0x4000 1FFF	TIM13		
0x4000 1800 - 0x4000 1BFF	TIM12		
0x4000 1400 - 0x4000 17FF	TIM7		
0x4000 1000 - 0x4000 13FF	TIM6	APB1	由Clock Configuration得到APB1 timer clock頻率
0x4000 0C00 - 0x4000 0FFF	TIM5		
0x4000 0800 - 0x4000 0BFF	TIM4		
0x4000 0400 - 0x4000 07FF	TIM3		
0x4000 0000 - 0x4000 03FF	TIM2		



Prescaler = ? Counter Period = ?



$$UI \text{ Frequency (Hz)} = \frac{TIM6CLK \text{ (Hz)}}{(ARR + 1) \times (PSC + 1)}$$

$$2 \text{ Hz} = \frac{84 \text{ MHz}}{(ARR + 1) \times (PSC + 1)}$$

$$\therefore (ARR + 1) \times (PSC + 1) = \frac{84 \text{ MHz}}{2 \text{ Hz}} = 42000000$$

取 (ARR+1) = 10000 · 則 (PSC+1) = 4200

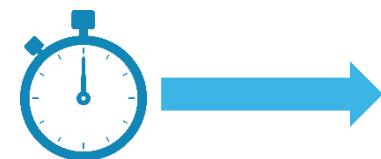
$$\therefore ARR = 9999, \quad PSC = 4199$$

加入程式碼

```
int main(void)
{
    /* USER CODE BEGIN 2 */
    HAL_TIM_Base_Start_IT(&htim6);
    /* USER CODE END 2 */

    ...
    while(1) {}

    /* USER CODE BEGIN 4 */
    void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef*htim)
    {
        HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
    }
    /* USER CODE END 4 */
```



```
void TIM6_DAC_IRQHandler(void)
{
    HAL_TIM_IRQHandler(&htim6);
}
```

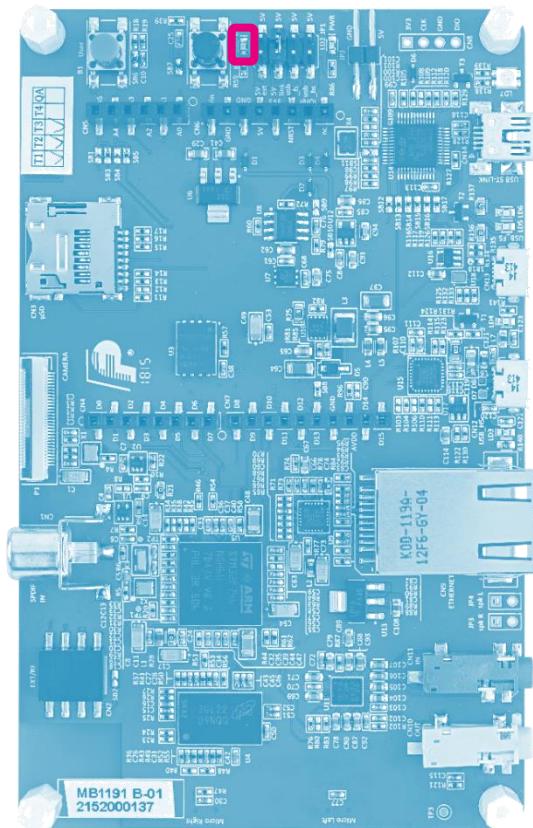
```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef*htim)
{
    HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
}
```

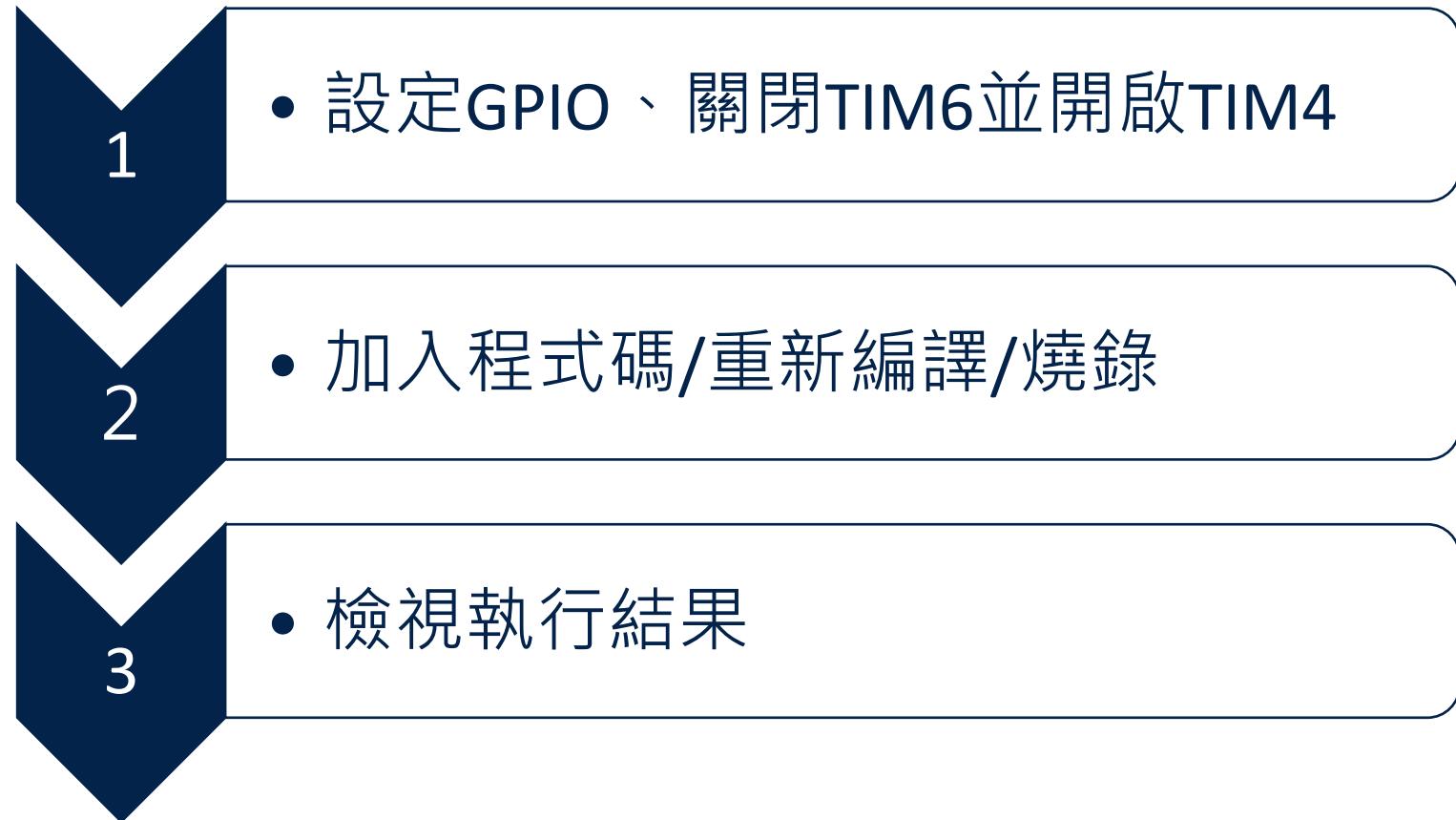
Hands-on

08_TIMER-PWM

程式碼實作目標

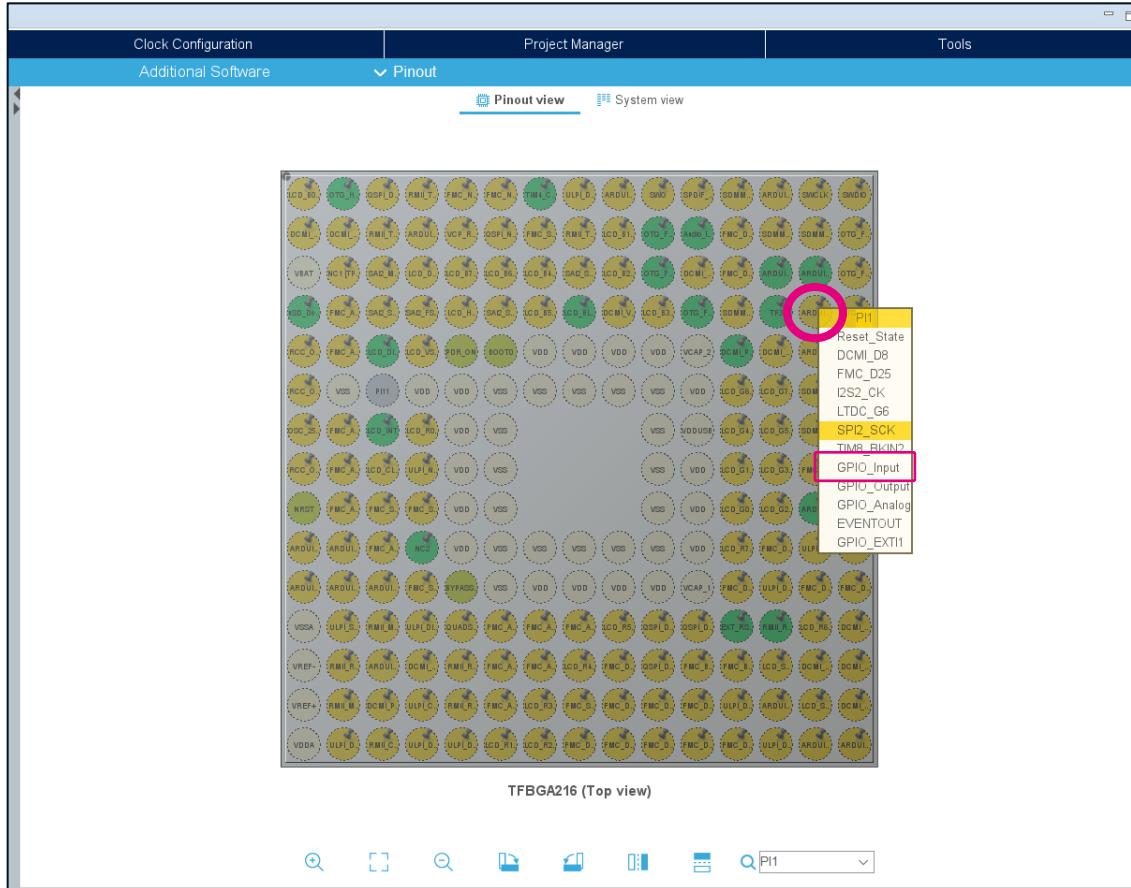
每500ms變LD1閃滅狀態



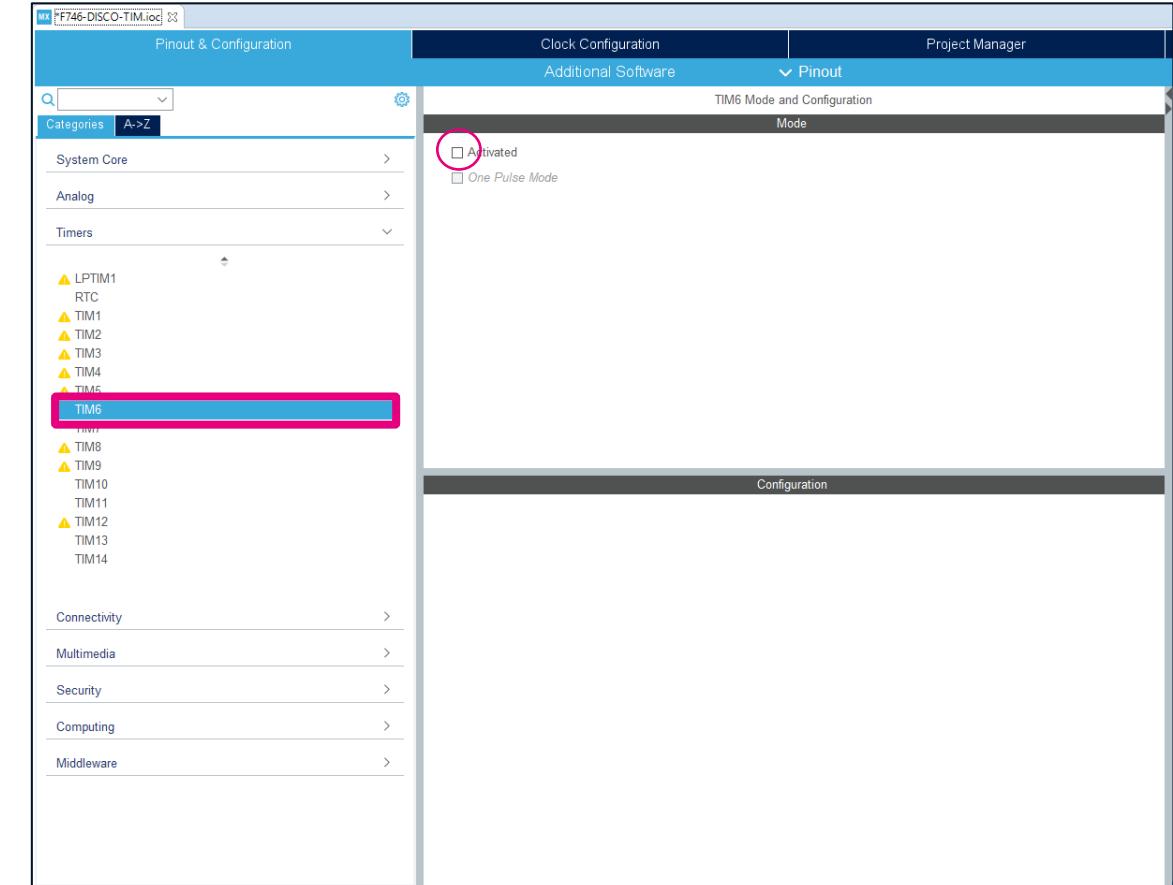


Step 2: 設定PI1和關閉TIM6

PI1 = GPIO-Input

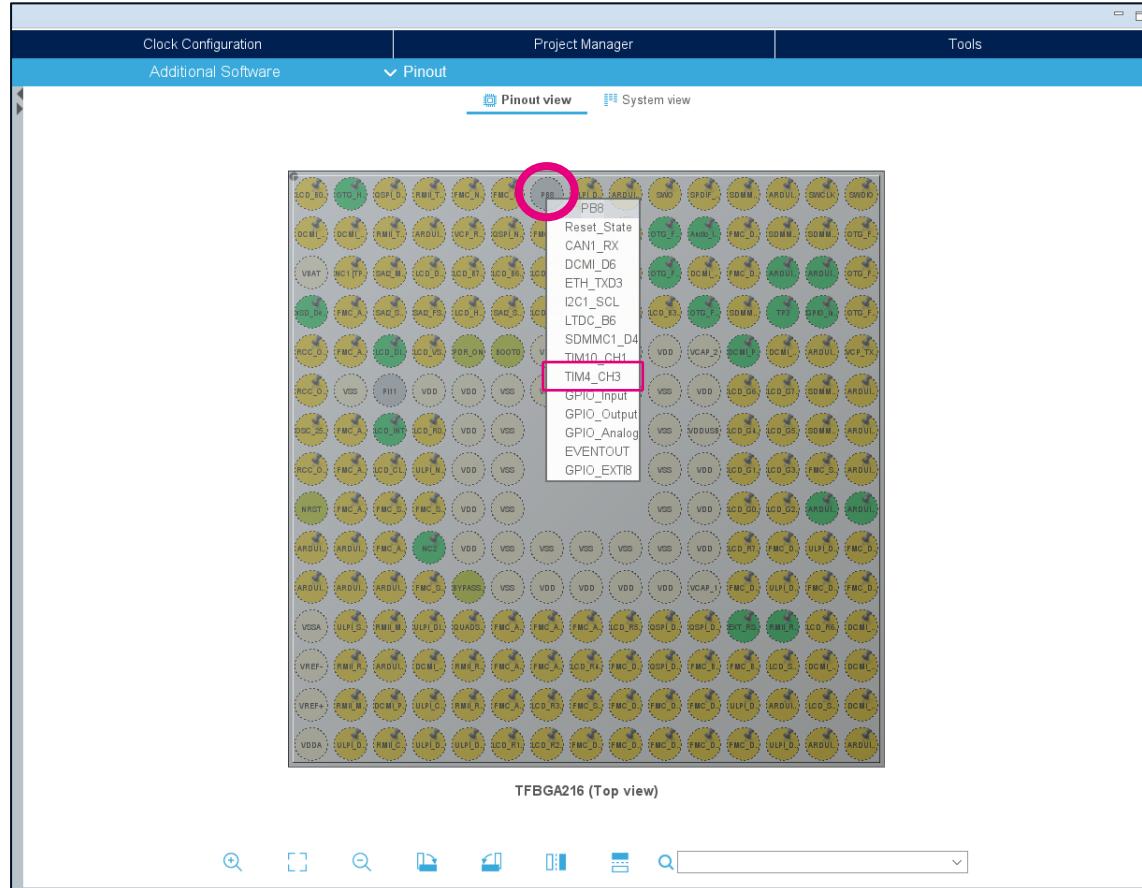


TIM6



Step 2: 設定TIM4

PB8 = TIM4_CH3 (PWM Output)



TIM4-CH3

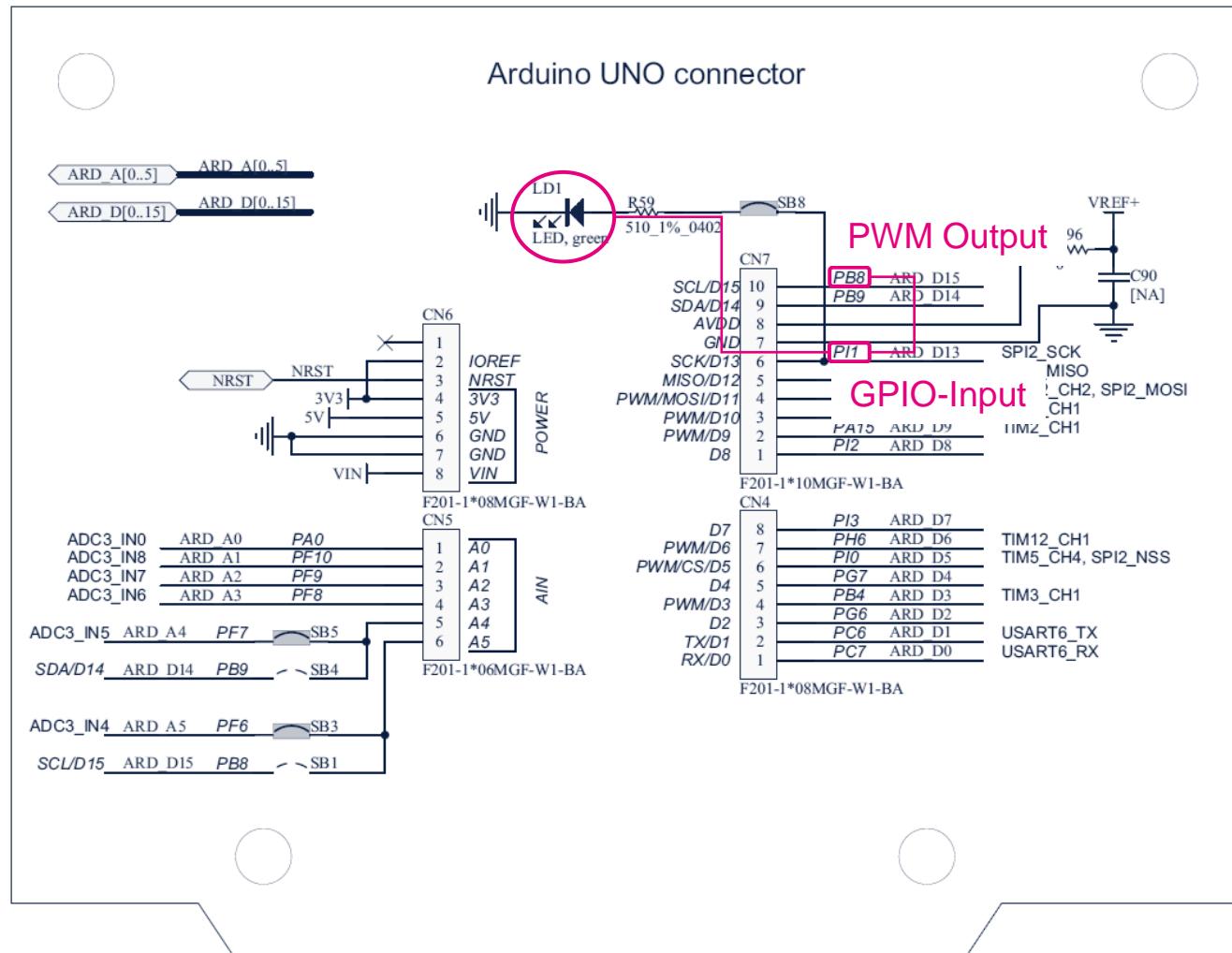
The screenshot shows the Pinout & Configuration interface for TIM4-CH3. The left sidebar lists categories like System Core, Analog, Timers, and various timers (LPTIM1, RTC, TIM1, TIM2, TIM3, TIM4, etc.). The right panel shows the TIM4 Mode and Configuration settings. A red box highlights the 'Mode' dropdown set to 'PWM mode 1'. Another red box highlights the 'Pulse (16 bits value)' field set to '4999'. The 'Prescaler (PSC - 16 bits value)' is set to '8399' and the 'Counter Mode' is set to 'Up'. The 'Counter Period (AutoReload Register - 16 bits value)' is set to '9999'.

Setting	Value
Mode	PWM mode 1
Pulse (16 bits value)	4999
Prescaler (PSC - 16 bits value)	8399
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	9999

此除頻設定會讓TIM4產生週期為1Hz且duty cycle為50%的PWM訊號由PB8輸出

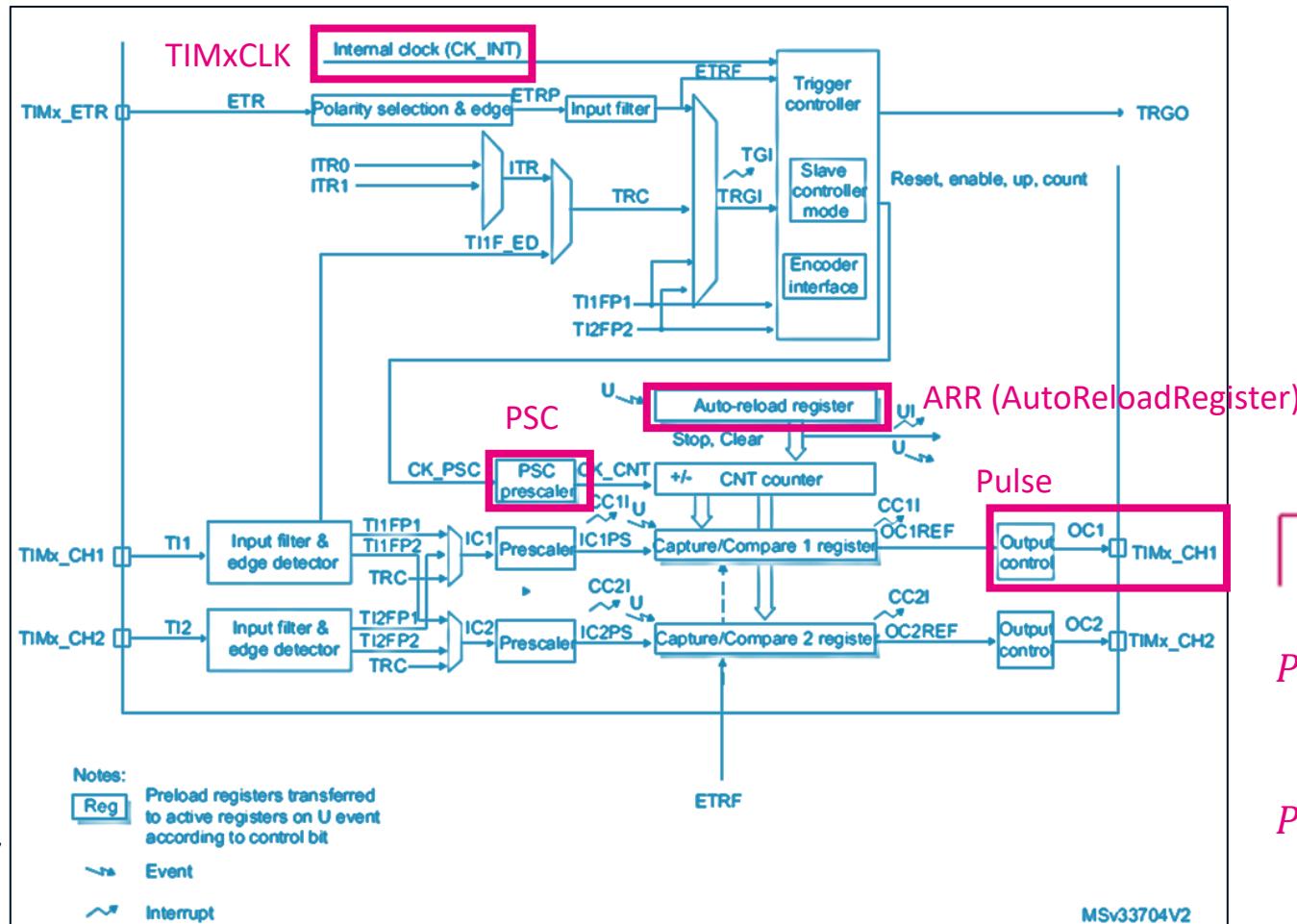
PI1 = GPIO-Input

PB8 = TIM4_CH3 (PWM Output)



Timer-PWM功能方塊示意圖

以General Purpose Timer (TIM4)為例



PWM Frequency (Hz) = $\frac{\text{TIMxCLK (Hz)}}{(ARR + 1) \times (PSC + 1)}$

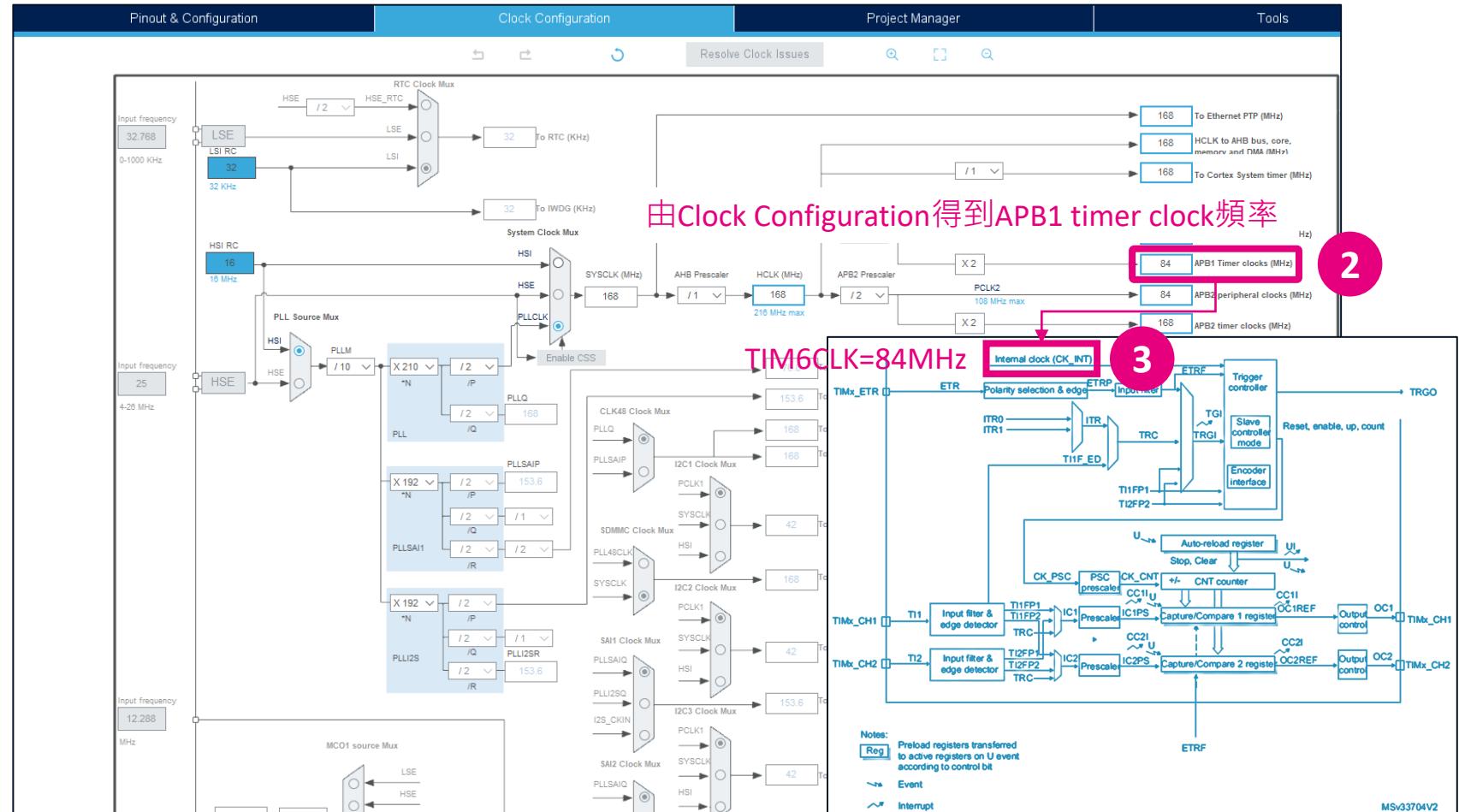
PWM Duty Cycle (%) = $\frac{(\text{Pulse} + 1)}{(ARR + 1)}$

TIMxCLK = ?

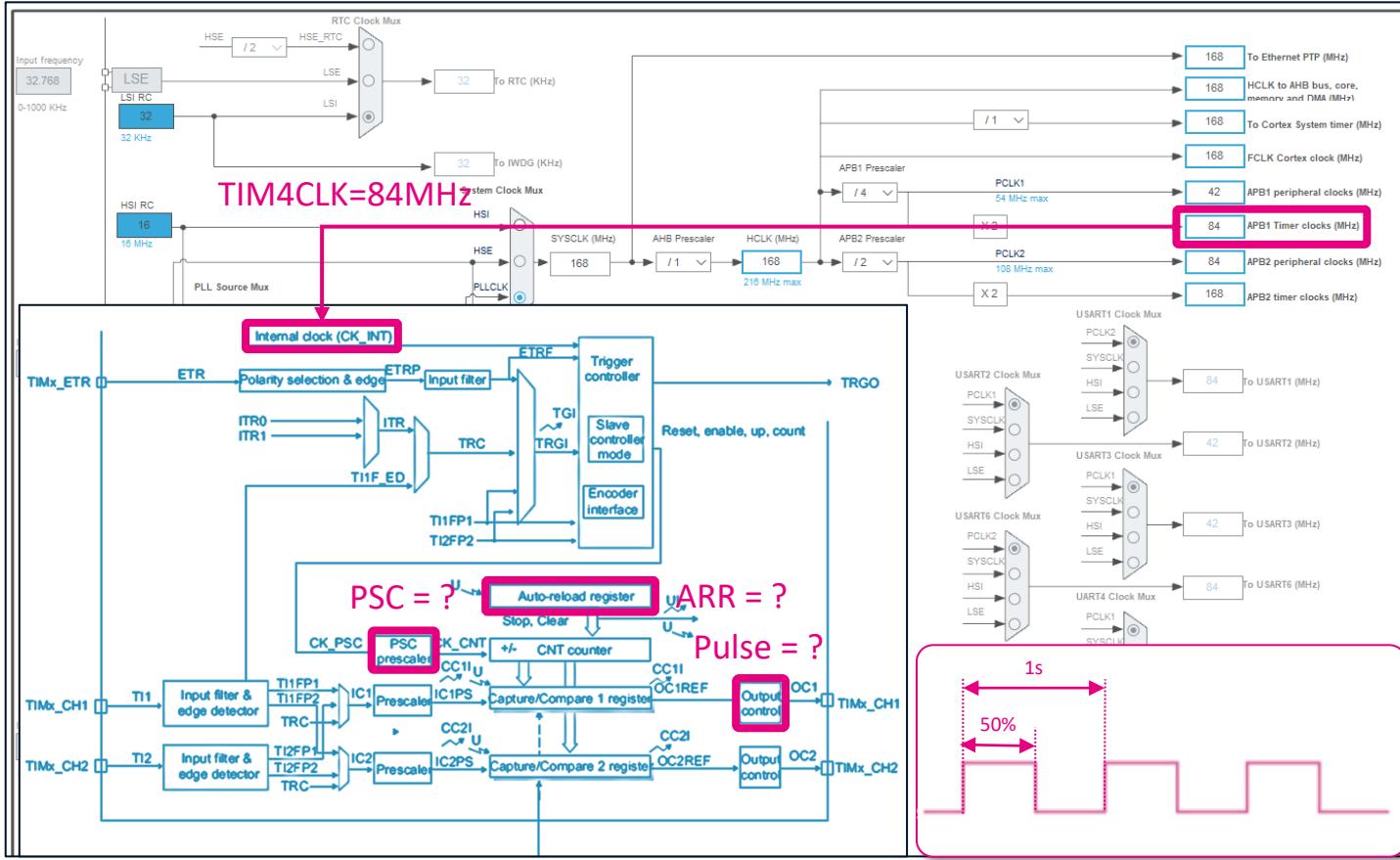
查閱RM0385

RM0385			
Boundary address	Peripheral	Bus	Register map
0x4000 7C00 - 0x4000 7FFF	UART8		
0x4000 7800 - 0x4000 7BFF	UART7		
0x4000 7400 - 0x4000 77FF	DAC		
0x4000 7000 - 0x4000 73FF	PWR		
0x4000 6C00 - 0x4000 6FFF	HDMI-CEC		
0x4000 6800 - 0x4000 6BFF	CAN2		
0x4000 6400 - 0x4000 67FF	CAN1		
0x4000 6000 - 0x4000 63FF	I2C4		
0x4000 5C00 - 0x4000 5FFF	I2C3		
0x4000 5800 - 0x4000 5BFF	I2C2		
0x4000 5400 - 0x4000 57FF	I2C1		
0x4000 5000 - 0x4000 53FF	UART5		
0x4000 4C00 - 0x4000 4FFF	UART4		
0x4000 4800 - 0x4000 4BFF	USART3		
0x4000 4400 - 0x4000 47FF	USART2		
0x4000 4000 - 0x4000 43FF	SPDIFRX		
0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S		
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2		
0x4000 3000 - 0x4000 33FF	IWDG		
0x4000 2C00 - 0x4000 2FFF	WWDG		
0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers		
0x4000 2400 - 0x4000 27FF	LPTIM1		
0x4000 2000 - 0x4000 23FF	TIM14		
0x4000 1C00 - 0x4000 1FFF	TIM13		
0x4000 1800 - 0x4000 1BFF	TIM12		
0x4000 1400 - 0x4000 17FF	TIM7		
0x4000 1000 - 0x4000 13FF	TIM6		
0x4000 0C00 - 0x4000 0FFF	TIM5		
0x4000 0800 - 0x4000 0BFF	TIM4		
0x4000 0400 - 0x4000 07FF	TIM3		
0x4000 0000 - 0x4000 03FF	TIM2		

1
查閱RM得到
TIM6為APB1-Bus



PSC / ARR / Pulse = ?



- $\text{PWM Frequency (Hz)} = \frac{\text{TIM}x\text{CLK (Hz)}}{(\text{ARR}+1) \times (\text{PSC}+1)}$

$$1 \text{ Hz} = \frac{84000000}{(\text{ARR} + 1) \times (\text{PSC} + 1)}$$

$$\therefore (\text{ARR} + 1) \times (\text{PSC} + 1) = 84000000$$

$$\text{取 } (\text{ARR}+1) = 10000 \quad \text{則 } (\text{PSC}+1) = 8400$$

$$\therefore \text{ARR} = 9999, \quad \text{PSC} = 8399$$

- $\text{PWM Duty Cycle (\%)} = \frac{(\text{Pulse}+1)}{(\text{ARR}+1)}$

$$50\% = \frac{(\text{Pulse} + 1)}{10000} \quad \therefore \text{Pulse} = 4999$$

加入程式碼

```
/* USER CODE BEGIN 2 */  
HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_3);  
/* USER CODE END 2 */
```

軟體安裝

X-CUBE-TOUCHGFX的安裝

X-CUBE-TouchGFX的安裝 – 1



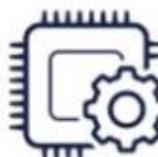
STM32CubeMx plugin

Graphic HW and SW configuration and project generation



TouchGFX PC tool

Graphic application development and simulation



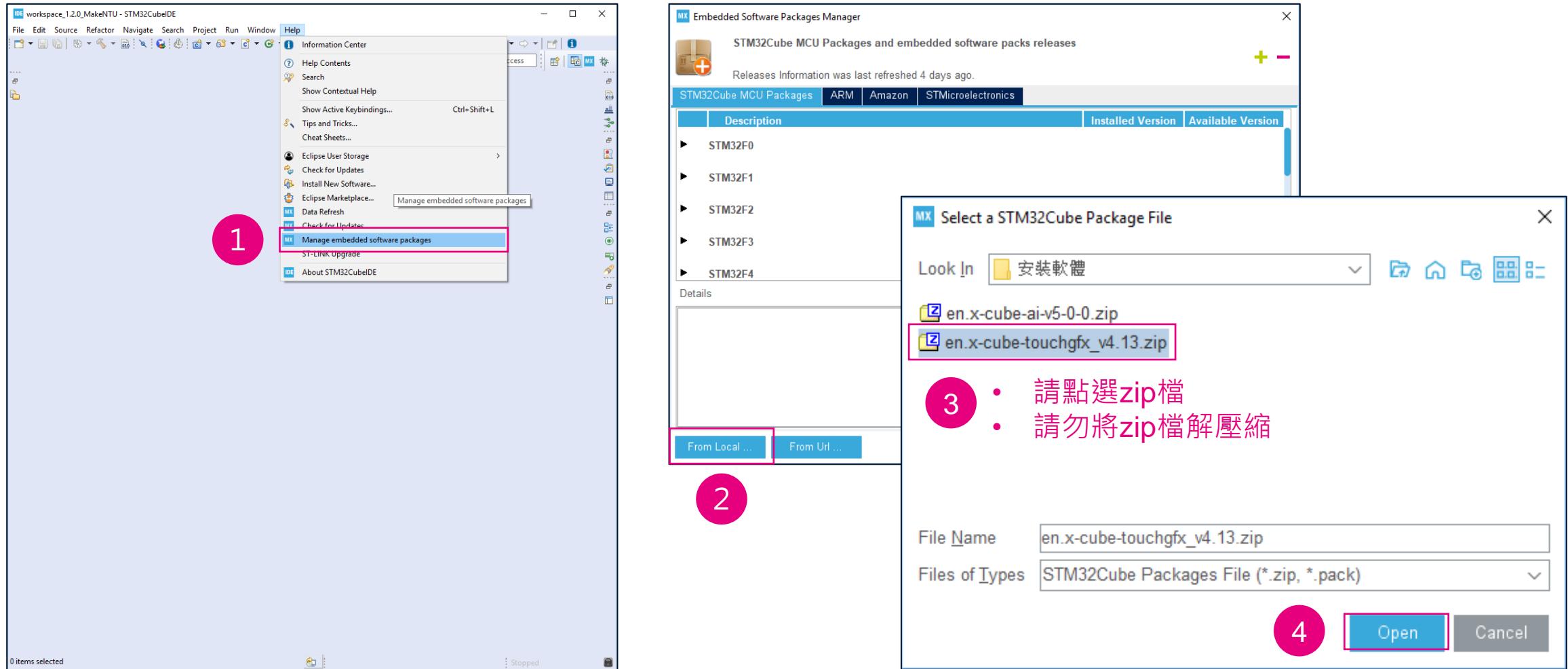
TouchGFX embedded library

Optimized and Hardware accelerated graphic library

 **Windows**

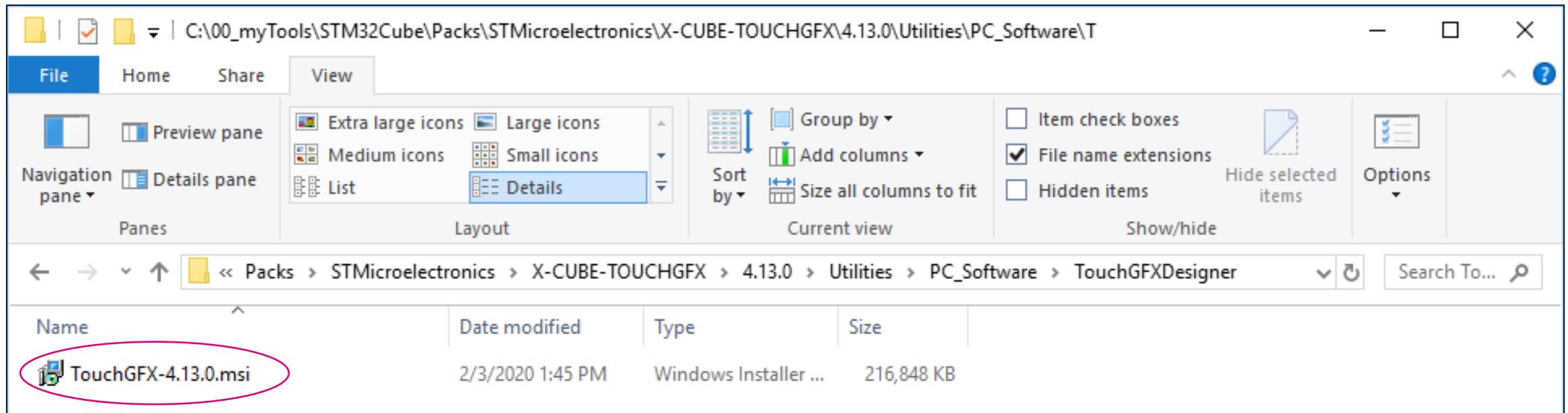
 **macOS**

X-CUBE-TouchGFX的安裝 – 2

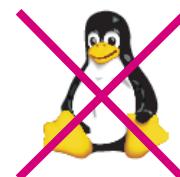


X-CUBE-TouchGFX的安裝 – 3

STM32CubeIDE Repository的位置



STM32CubeIDE-Reposiboty\Packs\STMicroelectronics\X-CUBE-TOUCHGFX\4.13.0\Utilities\PC_Software\TouchGFXDesigner

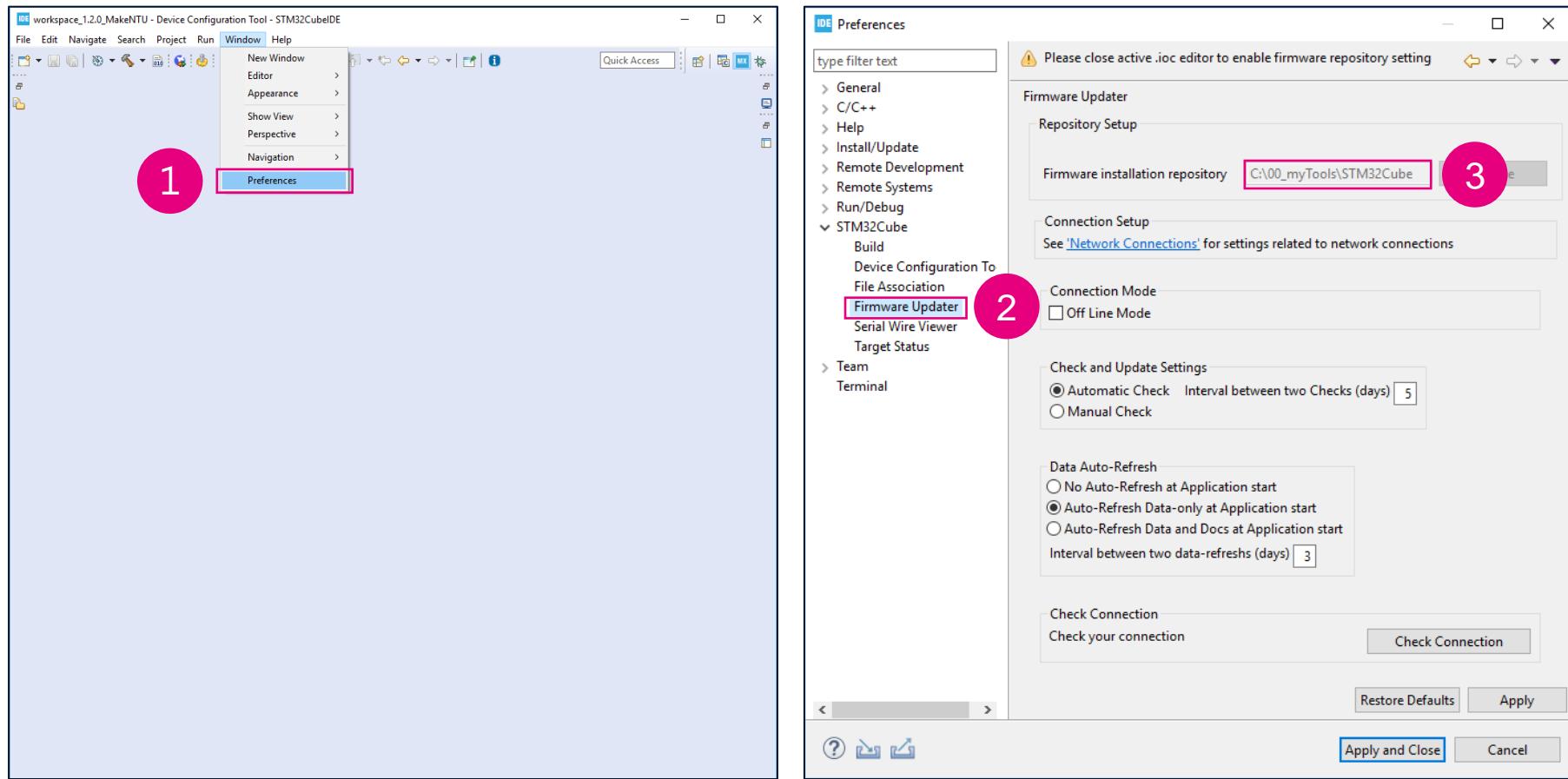


Windows

macOS

X-CUBE-TouchGFX的安裝 – 4

STM32CubeIDE Repository的位置



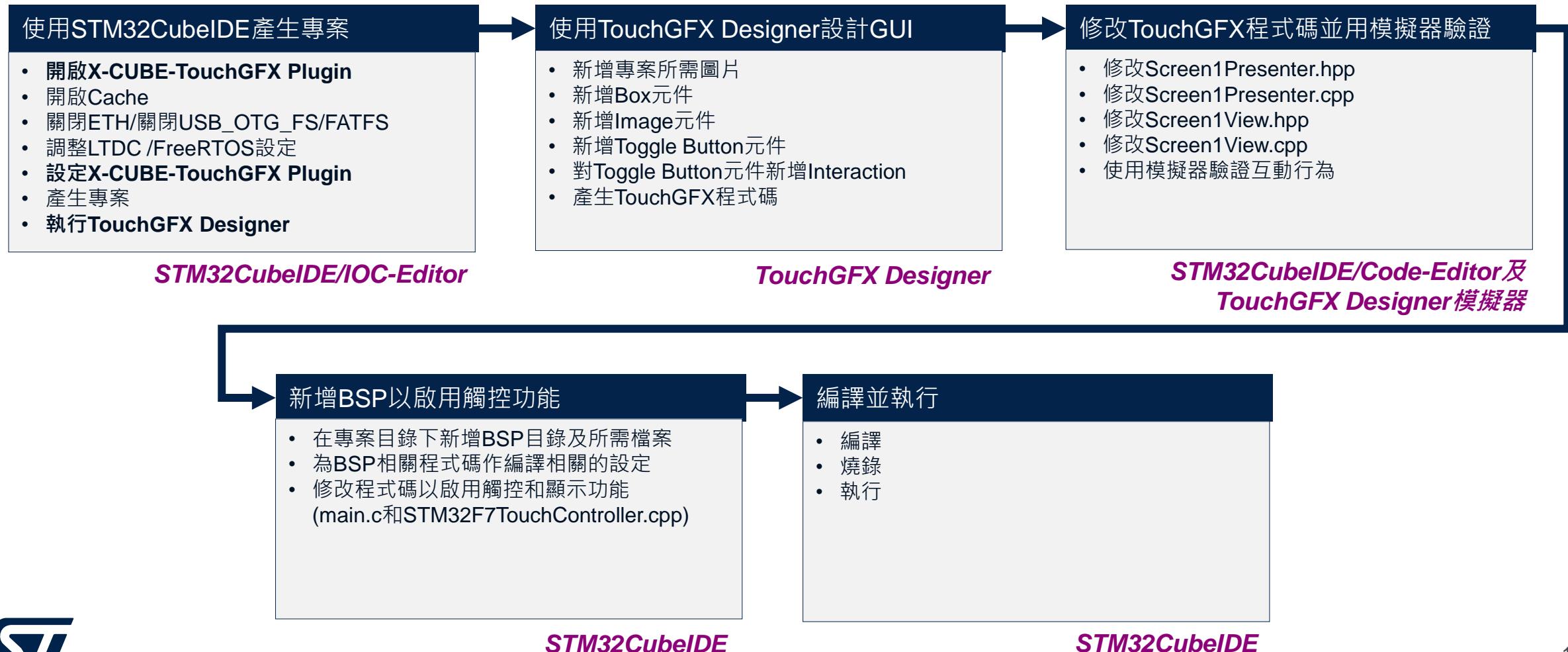
Hands-on

09_TouchGFX

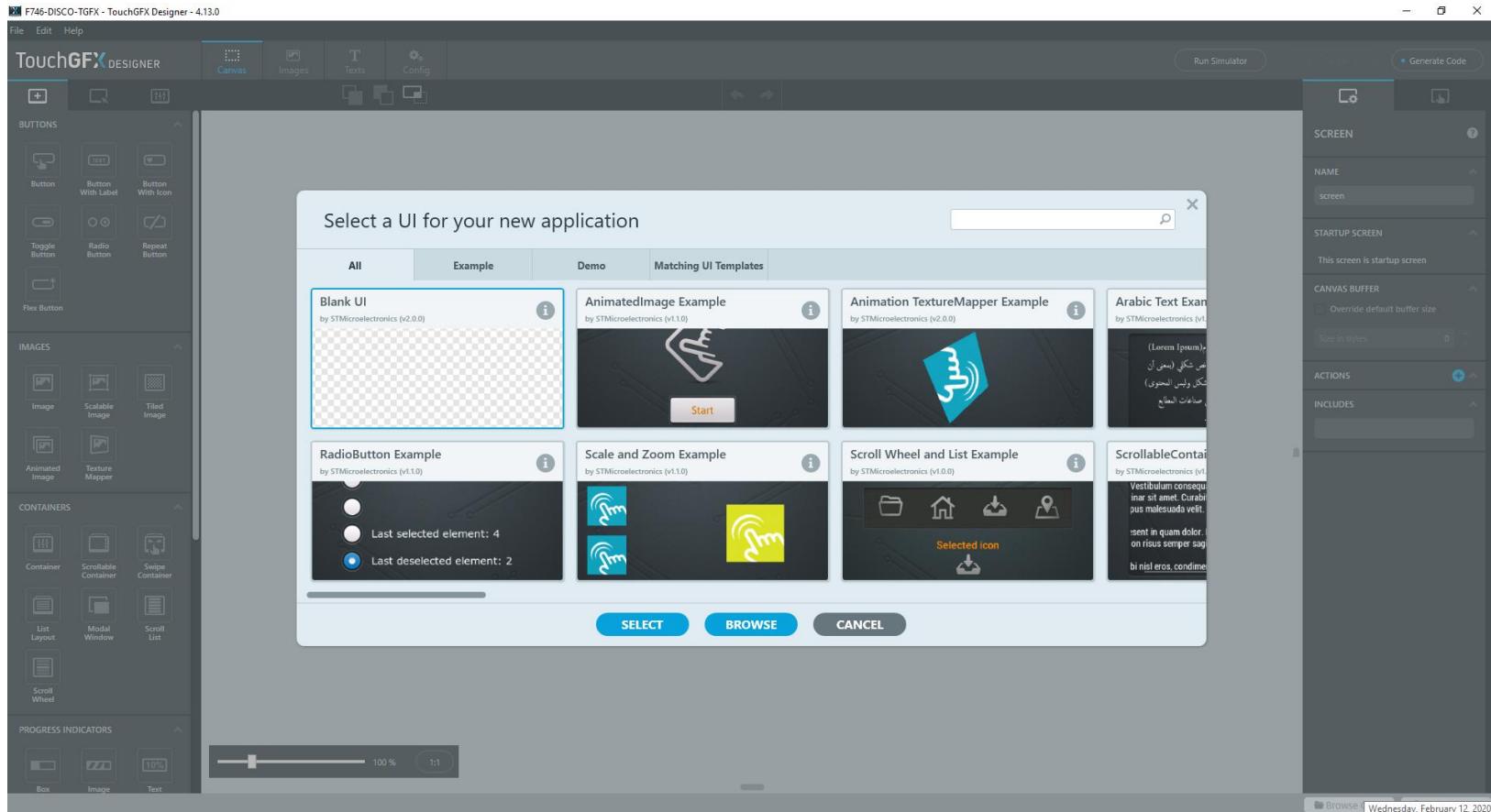
實作目標



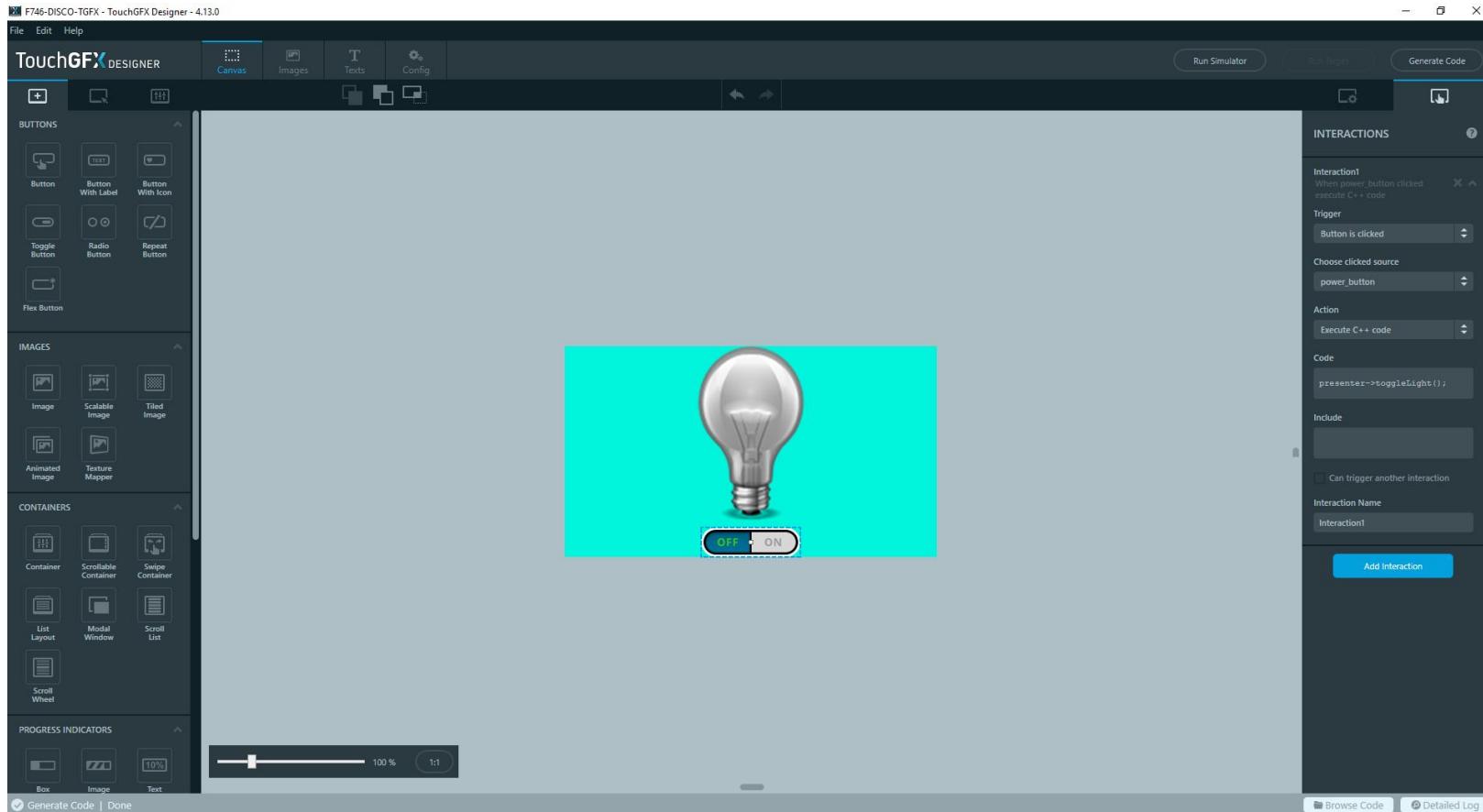
流程說明



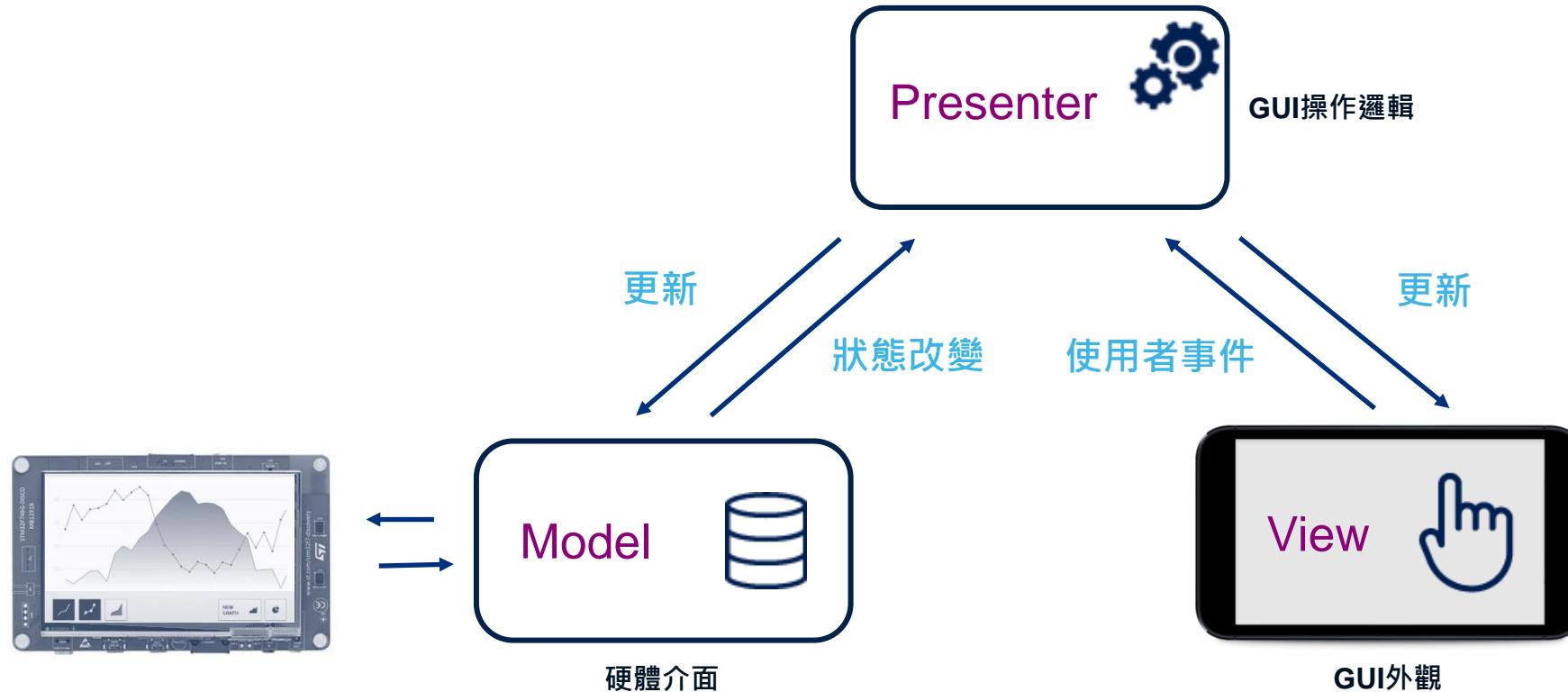
TouchGFX Designer – 1



TouchGFX Designer – 2



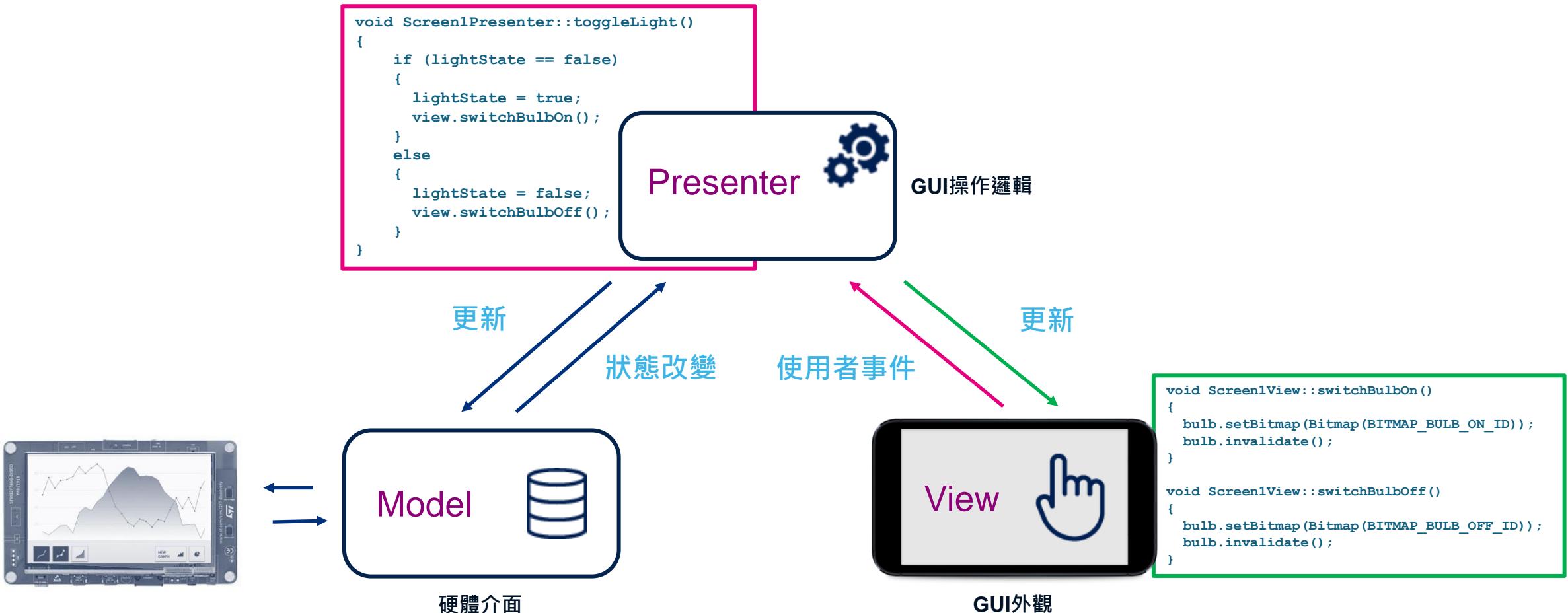
MVP (Model – View - Presenter) 模型



程式碼說明 (1)

```
24 Screen1ViewBase::Screen1ViewBase() :  
25     buttonCallback(this, &Screen1ViewBase::buttonCallbackHandler)  
26 {  
27     background.setPosition(0, 0, 480, 272);  
28     background.setColor(touchgfx::Color::getColorFrom24BitRGB(5, 250, 163));  
29 }  
30  
31 bulb.setXY(134, 0);  
32 bulb.setImageBitmap(Bitmap(BITMAP_BULB_OFF_ID));  
33  
34 power_button.setXY(176, 234);  
35 power_button.setBitmaps(Bitmap(BITMAP_BLUE_TOGGLEBARS_TOGGLE_ROUND_LAR  
36     _ON_ID), Bitmap(BITMAP_BLUE_TOGGLEBARS_TOGGLE_ROUND_LARGE_BUTTON  
37     _ON_ID));  
38 power_button.setAction(buttonCallback);  
39  
40 add(background);  
41 add(bulb);  
42 add(power_button);  
43 }  
44  
45 void Screen1ViewBase::setupScreen()  
46 {  
47 }  
48  
49 void Screen1ViewBase::buttonCallbackHandler(const touchgfx::AbstractButton  
50     & src)  
51 {  
52     if (&src == &power_button)  
53     {  
54         //Interaction1  
55         //When power_button clicked execute C++ code  
56         //Execute C++ code  
57         presenter->toggleLight();  
58     }  
59 }  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1869  
1870  
1871  
1872  
1
```

程式碼說明 (2)



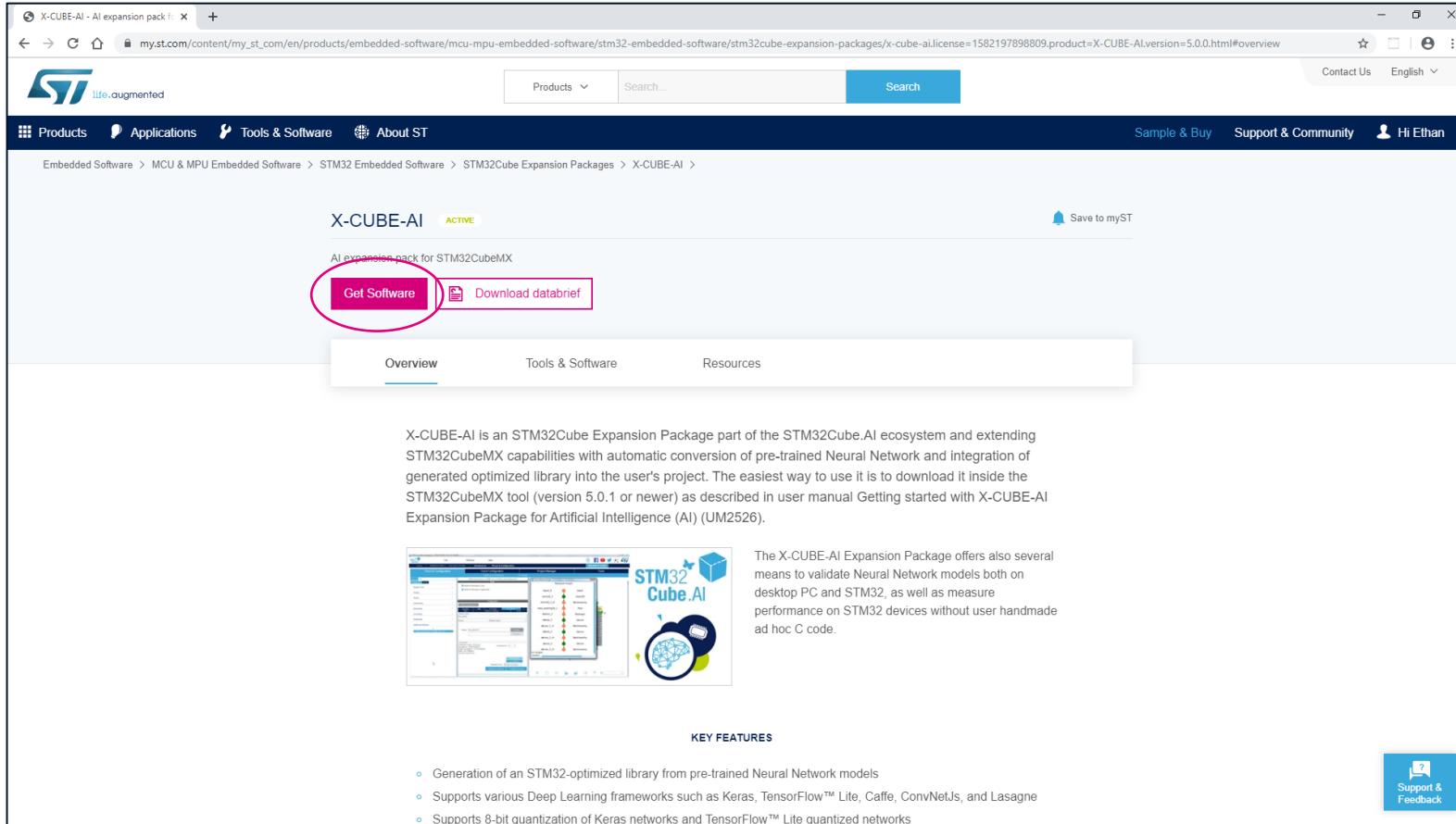
TouchGFX Documentation

<https://support.touchgfx.com/docs/introduction/welcome>

軟體安裝

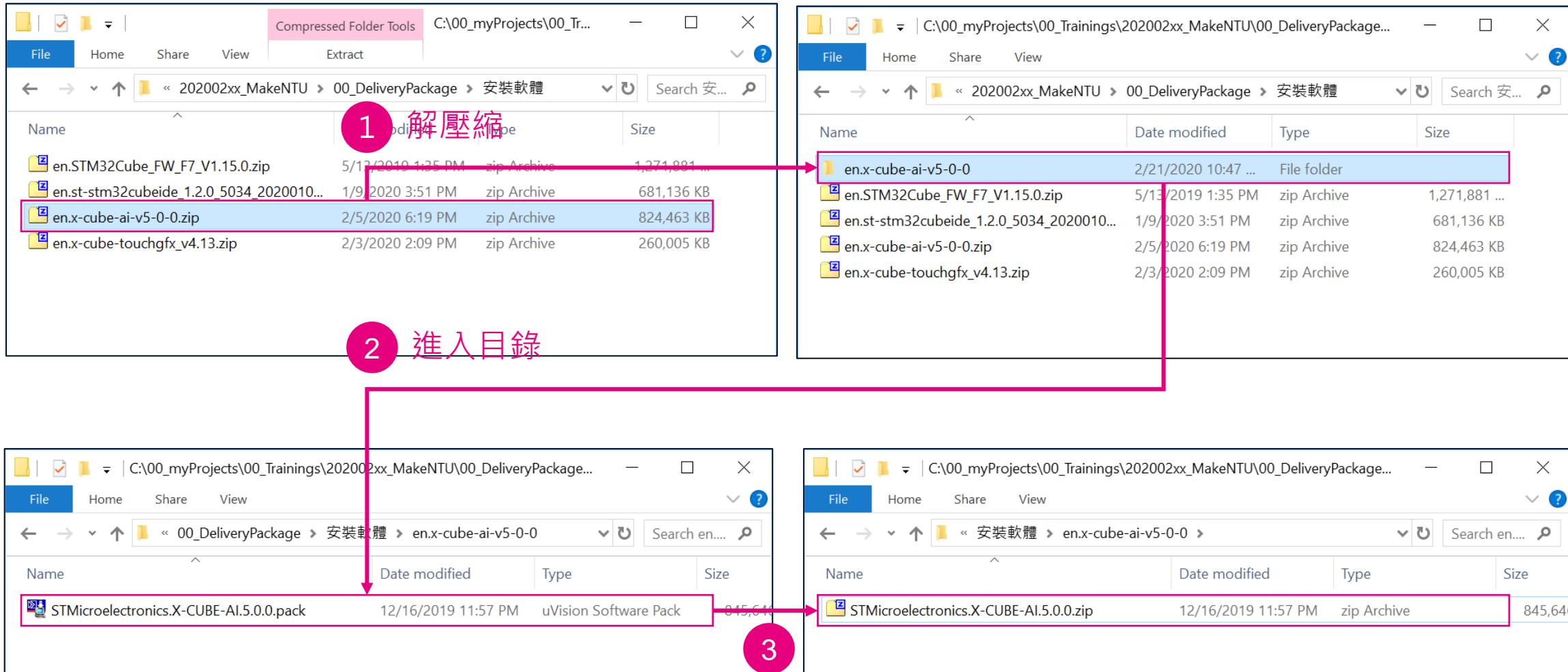
X-CUBE-AI的安裝

X-CUBE-AI的安裝 – 1



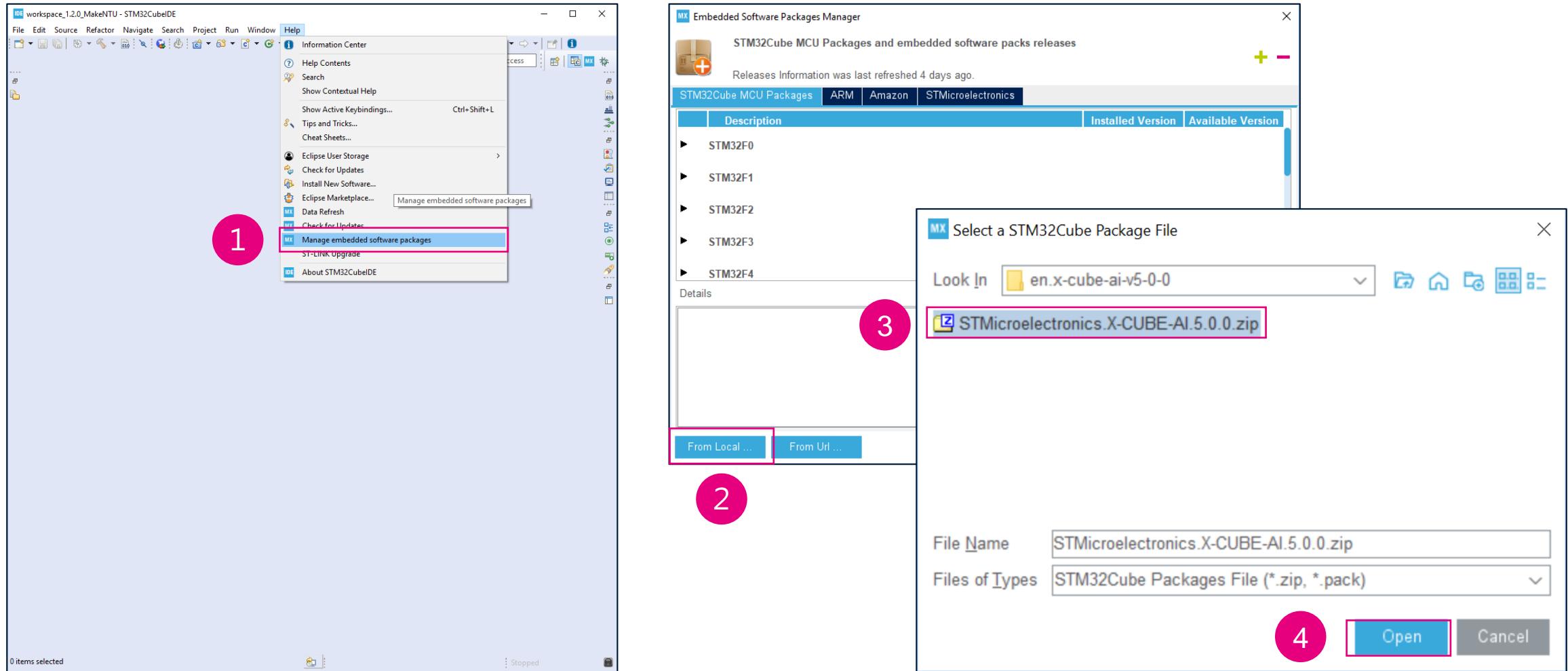
The screenshot shows the X-CUBE-AI product page on the STMicroelectronics website. The URL is https://my.st.com/content/my_st_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/x-cube-ai.license=1582197898809.product=X-CUBE-AI.version=5.0.0.html#get-software. The page features a navigation bar with links for Products, Applications, Tools & Software, About ST, Sample & Buy, Support & Community, and a user profile. The main content area displays the X-CUBE-AI expansion package for STM32CubeMX. It includes a brief description, two prominent buttons ('Get Software' and 'Download databrief'), and tabs for Overview, Tools & Software, and Resources. Below the description, there's a screenshot of the STM32Cube.AI software interface and a section titled 'KEY FEATURES' listing three bullet points. A 'Support & Feedback' button is located in the bottom right corner.

X-CUBE-AI的安裝 – 2



將副檔名pdsc由改為zip

X-CUBE-AI的安裝 – 3



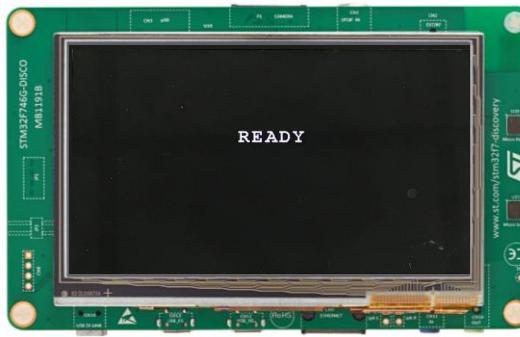
Hands-on

10_AI_HCR

流程說明

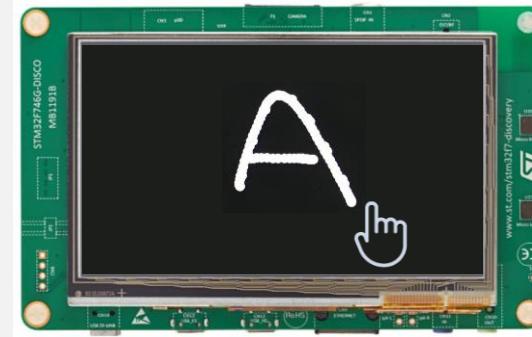
STAGE 1: 在LCD上顯示「READY」

1. 新增專案
2. 加入BSP
3. 新增程式碼
4. 編譯並執行



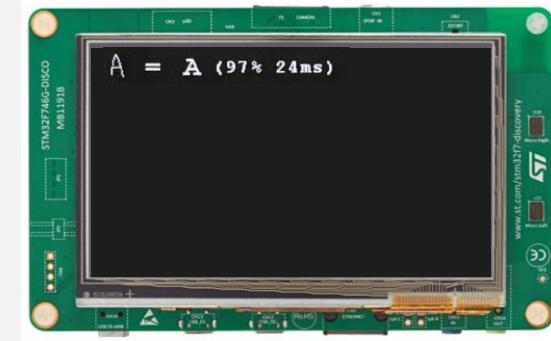
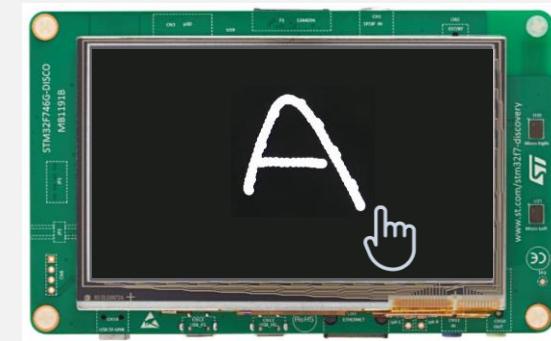
STAGE 2: 顯示手寫軌跡並作縮圖顯示

1. 新增「Utilities/ST_Image」目錄及程式碼
2. 編譯設定的調整
3. 新增程式碼
4. 編譯並執行



STAGE 3: 使用AI模型作手寫辨識

1. 使用X-CUBE-AI
2. 新增「Utilities/ST_DWT」目錄及程式碼
3. 編譯設定的調整
4. 新增程式碼
5. 編譯並執行



The Key Steps Behind Neural Networks



Neural Network (NN) Model Creation



Operating Mode

Capture data



1

2

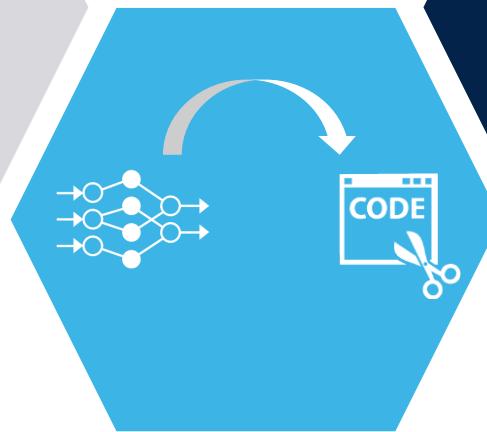
Train NN Model



3

Clean, label data
Build NN topology

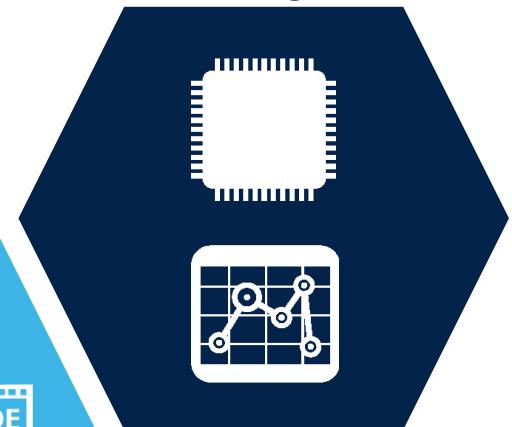
4



Convert NN into
optimized code for MCU

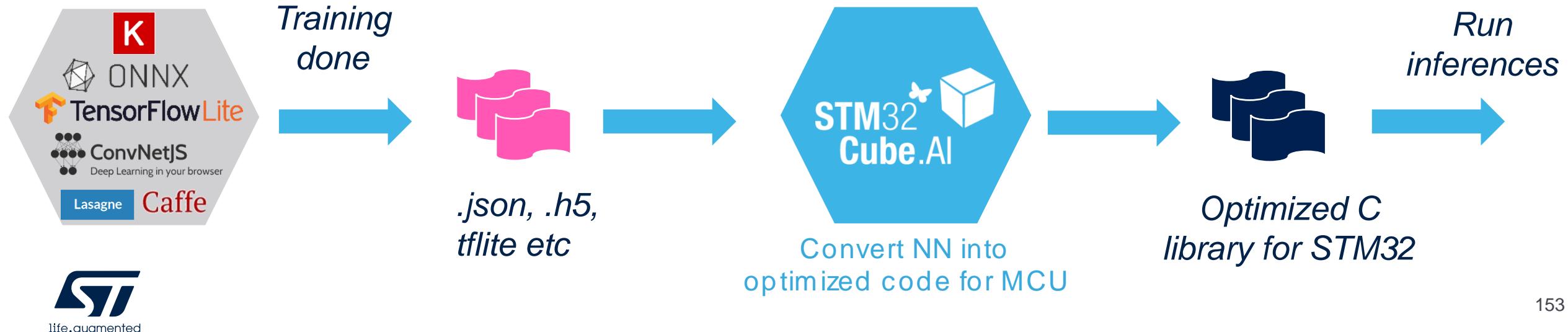
5

Process & analyze
new data using trained NN

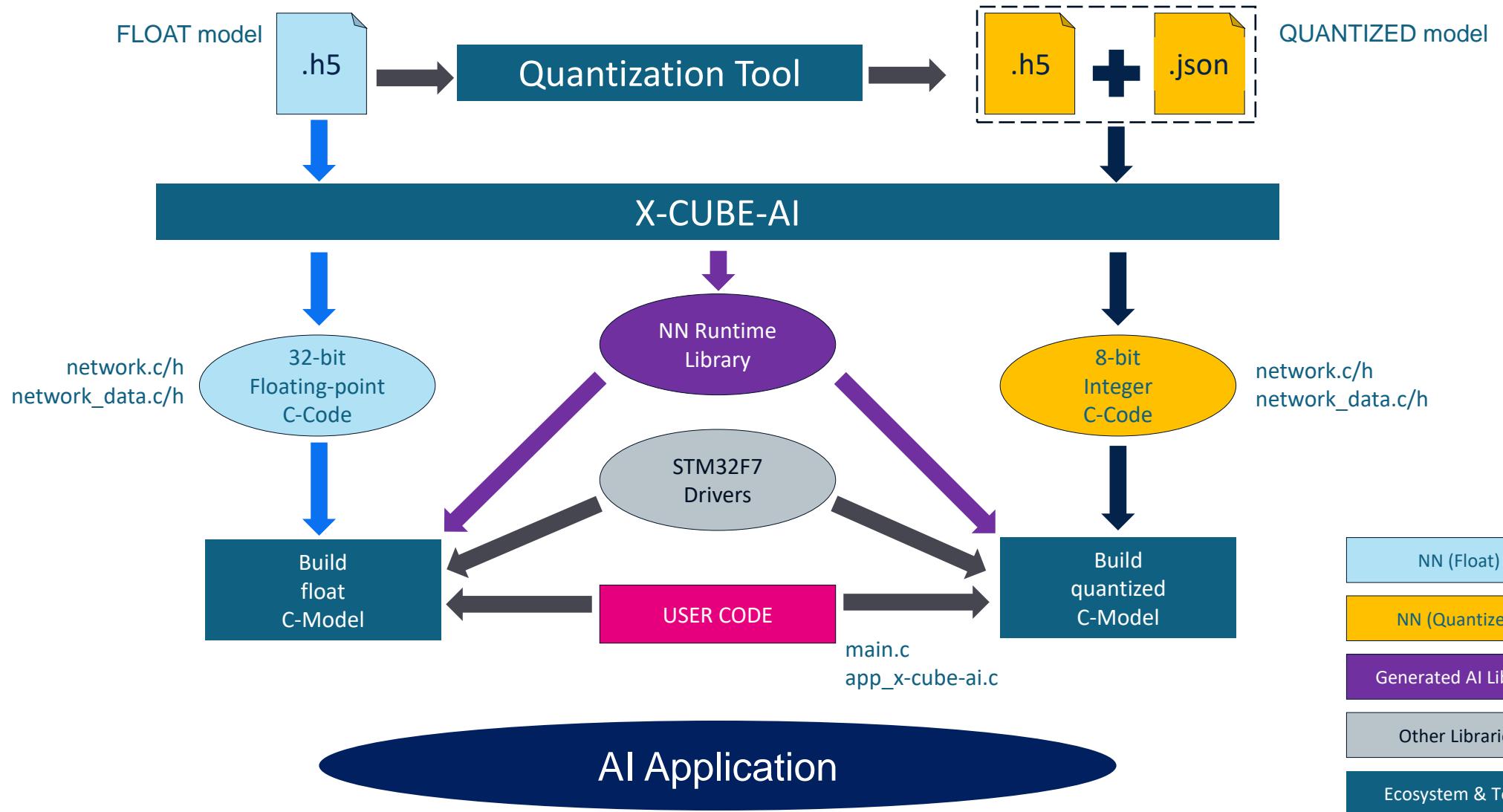


Link between AI frameworks and X-Cube-AI

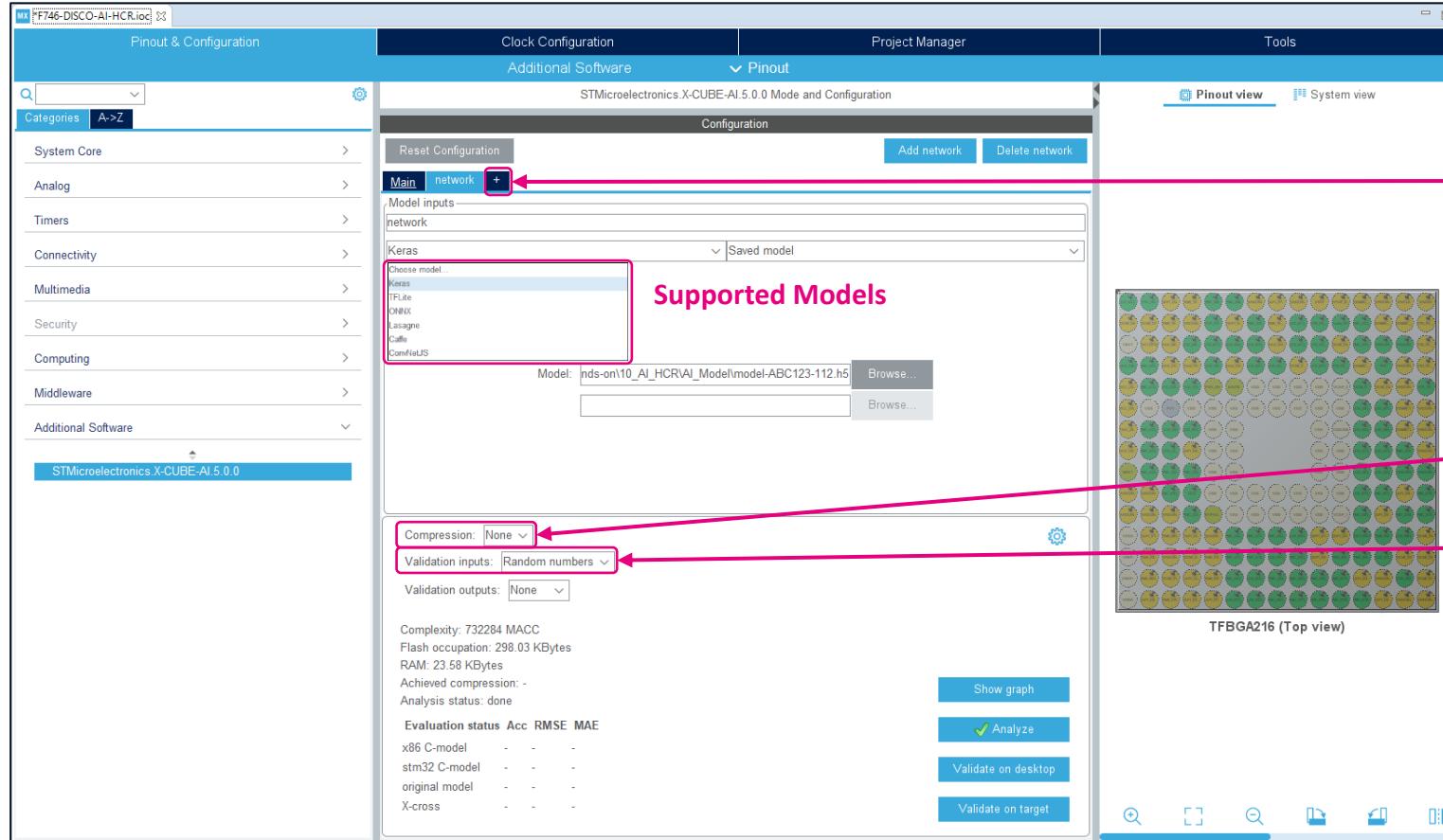
- When training done → Use AI frameworks to generate some output files
 - JSON file, binary file, python script file etc
 - Those files represent the neural network architecture, configuration and training parameters
- With the output files → Feed the X-Cube-AI
 - The tool uses those files as inputs to generate the optimized library for STM32



使用X-CUBE-AI產生AI專案



X-CUBE-AI Plugin – 1



Multi-network

You can upload several NN models (limited by memory)

Compression tool

Reduce the ROM size by compressing the training parameters

Custom data

run validation process with your own data (csv format)

X-CUBE-AI Plugin – 2

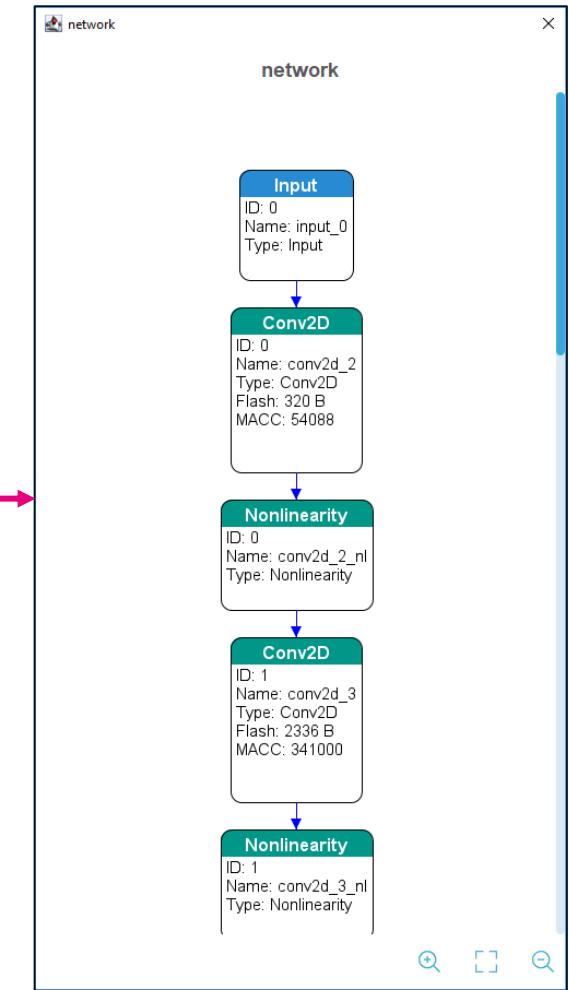
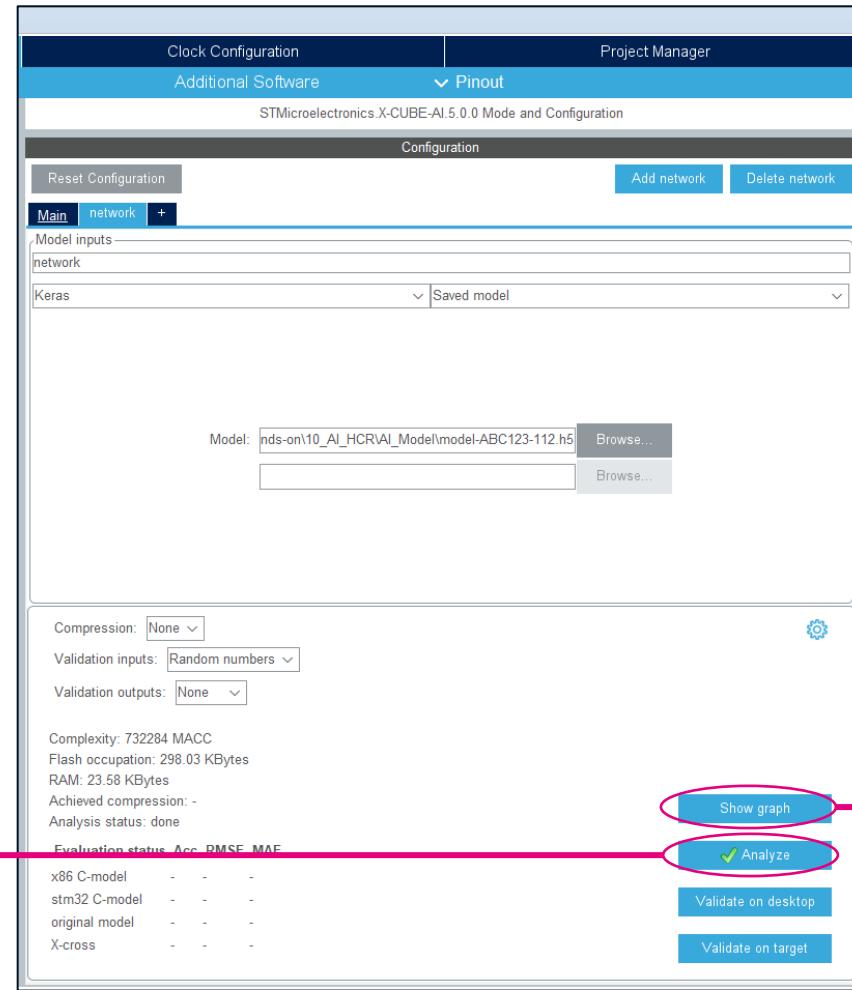
Be wait...

Analyzing Network

Complexity per-layer - macc=732,284 rom=298.032

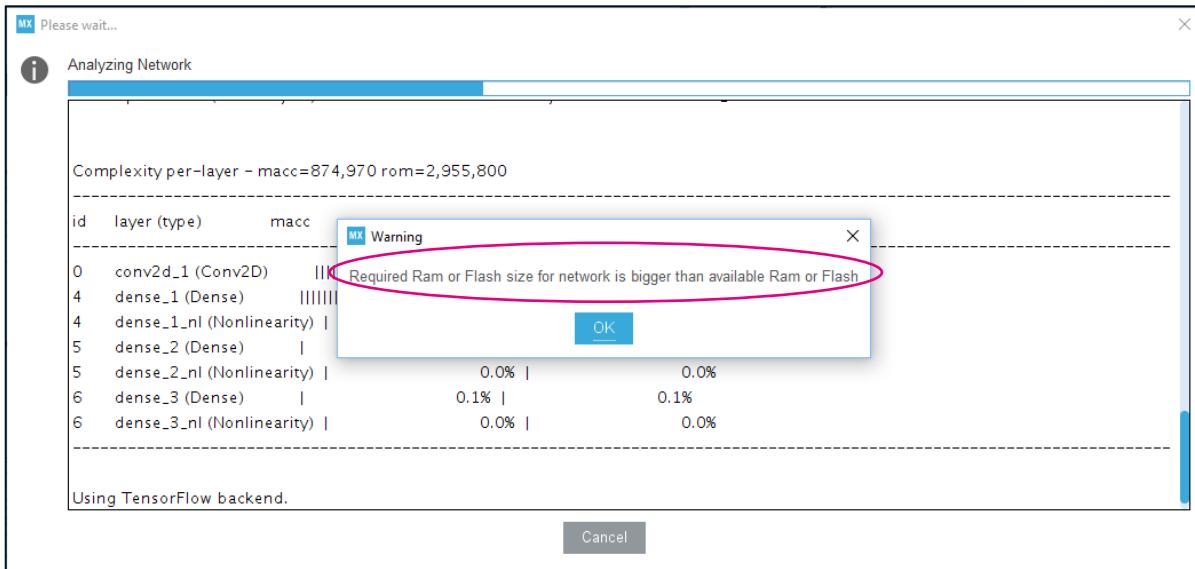
id	layer (type)	macc	rom
0	conv2d_2 (Conv2D)	7.4%	0.1%
1	conv2d_3 (Conv2D)	46.6%	0.8%
3	conv2d_4 (Conv2D)	16.0%	1.6%
4	conv2d_5 (Conv2D)	20.4%	3.1%
8	dense_3 (Dense)	8.4% 82.8%	
8	dense_3_nl (Nonlinearity)	0.0%	0.0%
10	dense_4 (Dense)	1.2%	11.6%
10	dense_4_nl (Nonlinearity)	0.1%	0.0%

Using TensorFlow backend.



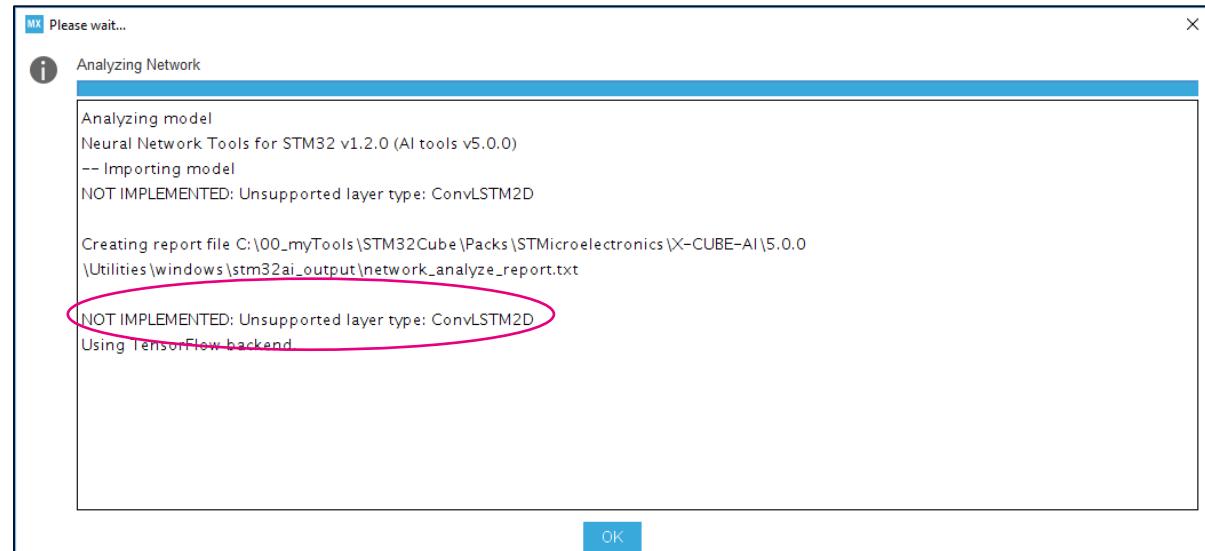
警示/錯誤訊息

Over Size

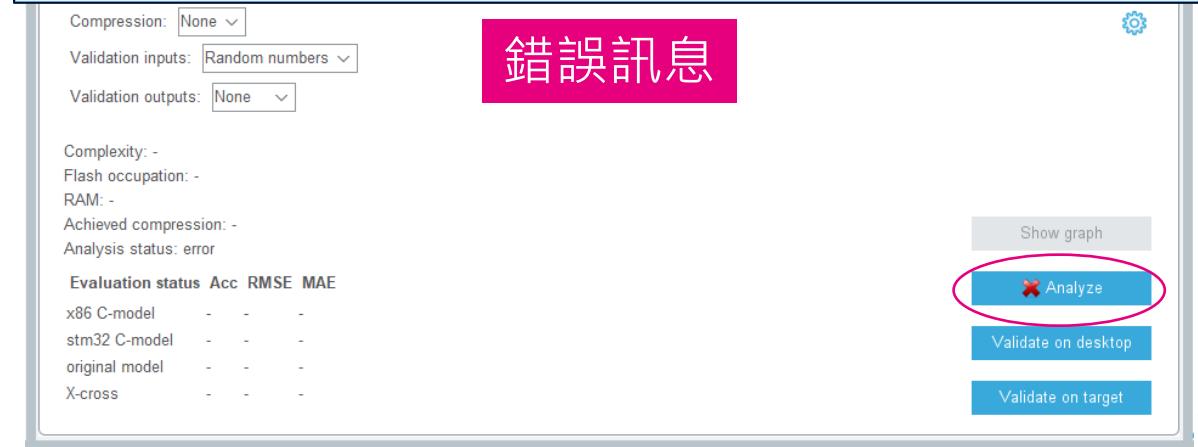


警示訊息

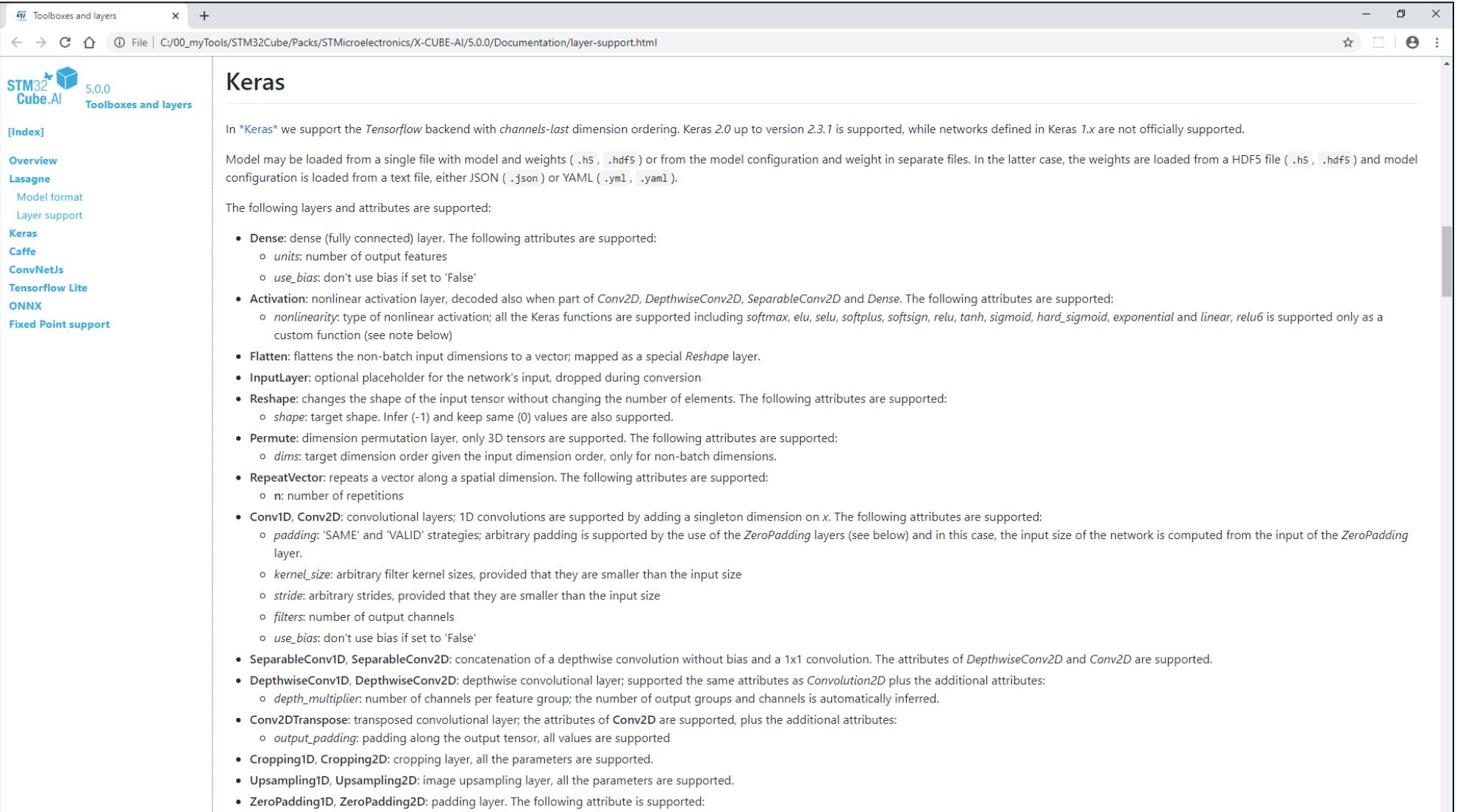
Unsupported Layer



錯誤訊息



Supported Layers

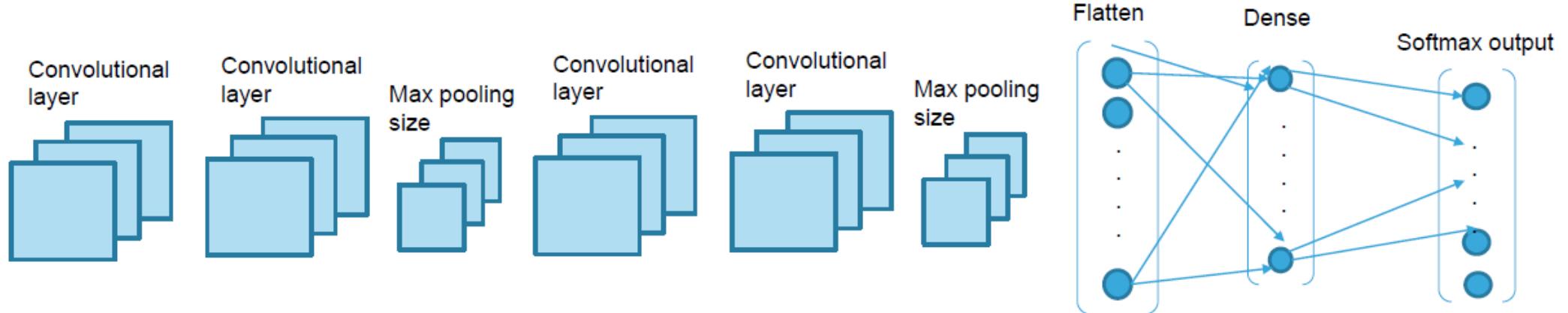


The screenshot shows a web browser window displaying the STM32Cube.AI 5.0.0 documentation for 'Toolboxes and layers'. The left sidebar contains links for Overview, Lasagne, Model format, Layer support, Keras, Caffe, ConvNetJs, Tensorflow Lite, ONNX, and Fixed Point support. The main content area is titled 'Keras' and discusses the support for the Keras library. It states that Tensorflow backend with 'channels-last' dimension ordering is supported up to version 2.3.1, while networks defined in Keras 1.x are not officially supported. It also mentions that models can be loaded from a single file or from separate files for weights and configuration. A detailed list of supported Keras layers and their attributes follows:

- **Dense:** dense (fully connected) layer. The following attributes are supported:
 - *units*: number of output features
 - *use_bias*: don't use bias if set to 'False'
- **Activation:** nonlinear activation layer, decoded also when part of *Conv2D*, *DepthwiseConv2D*, *SeparableConv2D* and *Dense*. The following attributes are supported:
 - *nonlinearity*: type of nonlinear activation; all the Keras functions are supported including *softmax*, *elu*, *selu*, *softplus*, *softsign*, *relu*, *tanh*, *sigmoid*, *hard_sigmoid*, *exponential* and *linear*, *relu6* is supported only as a custom function (see note below)
- **Flatten:** flattens the non-batch input dimensions to a vector; mapped as a special *Reshape* layer.
- **InputLayer:** optional placeholder for the network's input, dropped during conversion
- **Reshape:** changes the shape of the input tensor without changing the number of elements. The following attributes are supported:
 - *shape*: target shape. Infer (-1) and keep same (0) values are also supported.
- **Permute:** dimension permutation layer, only 3D tensors are supported. The following attributes are supported:
 - *dims*: target dimension order given the input dimension order, only for non-batch dimensions.
- **RepeatVector:** repeats a vector along a spatial dimension. The following attributes are supported:
 - *n*: number of repetitions
- **Conv1D, Conv2D:** convolutional layers; 1D convolutions are supported by adding a singleton dimension on x. The following attributes are supported:
 - *padding*: 'SAME' and 'VALID' strategies; arbitrary padding is supported by the use of the *ZeroPadding* layers (see below) and in this case, the input size of the network is computed from the input of the *ZeroPadding* layer.
 - *kernel_size*: arbitrary filter kernel sizes, provided that they are smaller than the input size
 - *stride*: arbitrary strides, provided that they are smaller than the input size
 - *filters*: number of output channels
 - *use_bias*: don't use bias if set to 'False'
- **SeparableConv1D, SeparableConv2D:** concatenation of a depthwise convolution without bias and a 1x1 convolution. The attributes of *DepthwiseConv2D* and *Conv2D* are supported.
- **DepthwiseConv1D, DepthwiseConv2D:** depthwise convolutional layer; supported the same attributes as *Convolution2D* plus the additional attributes:
 - *depth_multiplier*: number of channels per feature group; the number of output groups and channels is automatically inferred.
- **Conv2DTranspose:** transposed convolutional layer; the attributes of *Conv2D* are supported, plus the additional attributes:
 - *output_padding*: padding along the output tensor, all values are supported
- **Cropping1D, Cropping2D:** cropping layer, all the parameters are supported.
- **Upsampling1D, Upsampling2D:** image upsampling layer, all the parameters are supported.
- **ZeroPadding1D, ZeroPadding2D:** padding layer. The following attribute is supported:
 - *padding*: list of padding values, symmetric and asymmetric padding supported

關於model-ABC123-112.h5

- Covering handwriting characters (10 digits + 26 letters)
- Dataset (Grayscale images of size 28 x 28)
<https://www.nist.gov/itl/products-and-services/emnist-dataset>
- Referent paper (should be cited)
 - Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters.
 - <https://arxiv.org/abs/1702.05373>



STAGE 3 程式碼解說 – 1



STAGE 3 程式碼解說 – 2



STAGE 3 程式碼解說 – 3

```
void MX_X_CUBE_AI_Process(void)
{
    /* Get status and positions of the touch screen */
    if( BSP_TS_GetState(&ts_state) == TS_OK )
    {
        /* Touch event*/
        if( ts_state.touchDetected > 0 && data_ready == false )
        {
            BSP_LCD_Clear(LCD_COLOR_BLACK);
            data_ready = true;
        }

        /*Draw display*/
        for(int i=0;i<ts_state.touchDetected;i++)
        {
            touch_time = HAL_GetTick();
            BSP_LCD_FillCircle(ts_state.touchX[i],ts_state.touchY[i],PEN_POINT_SIZE);
        }

        /* Major block of drawing mini patch and running AI related code */
        if( data_ready && (HAL_GetTick() - touch_time) > TOUCH_TIMEOUT)
        {
            ...
        } // end of if( data_ready && (HAL_GetTick() - touch_time) > TOUCH_TIMEOUT)
    } // end of if( BSP_TS_GetState(&ts_state) == TS_OK )
}
```

```
void MX_X_CUBE_AI_Process(void)
{
    /* Get status and positions of the touch screen */
    if( BSP_TS_GetState(&ts_state) == TS_OK )
    {
        /* Touch event*/
        /*Draw display*/

        /* Major block of drawing mini patch and running AI related code */
        if( data_ready && (HAL_GetTick() - touch_time) > TOUCH_TIMEOUT)
        {
            /* retrieve downsampled image 58*32 from display memory (480*272) in SDRAM */
            int ii = 0;
            for(uint16_t y=16;y<BSP_LCD_GetYSize();y+=8)
            {
                for(uint16_t x=16;x<BSP_LCD_GetXSize();x+=8)
                {
                    g_patch8888_5832[ii++] = BSP_LCD_ReadPixel(x,y);
                }
            }

            /* Resize from 58*32 to 28*28 */
            ImageResize((uint8_t*)g_patch8888_5832,58,32,4,0,0,58,32,(uint8_t*)g_patch8888_2828, 28, 28);
            BSP_LCD_Clear(LCD_COLOR_BLACK);

            /* Draw 28*28 and prepare input array of AI model: in_data */
            ii=0;
            for(uint16_t y=0;y<PATCH_SIZE;y+=1)
            {
                for(uint16_t x=0;x<PATCH_SIZE;x+=1)
                {
                    in_data[ii] = (g_patch8888_2828[ii] > 0xFF000000)?1.0F:0.0F;
                    BSP_LCD_DrawPixel(x,y,g_patch8888_2828[ii++]);
                }
            }
            ...
        }
    }
}
```

STAGE 3 程式碼解說 – 4

```
void MX_X_CUBE_AI_Process(void)
{
    /* Get status and positions of the touch screen */
    if( BSP_TS_GetState(&ts_state) == TS_OK )
    {
        /* Touch event*/
        /*Draw display*/

        /* Major block of drawing mini patch and running AI related code */
        if( data_ready && (HAL_GetTick() - touch_time) > TOUCH_TIMEOUT)
        {
            /* retrieve downsampled image 58*32 from display memory (480*272) in SDRAM */
            ...
            /* Resize from 58*32 to 28*28 */
            ...

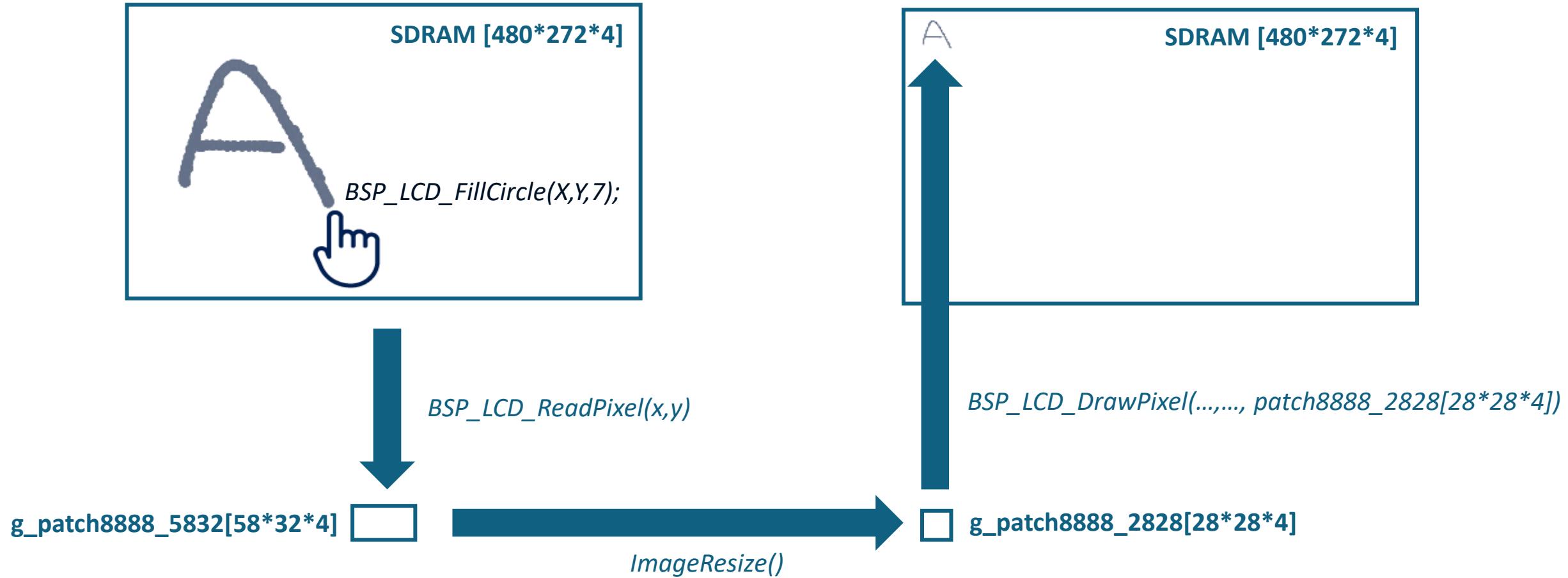
            /* Draw 28*28 and prepare input array of AI model: in_data */
            ii=0;
            for(uint16_t y=0;y<PATCH_SIZE;y+=1)
            {
                for(uint16_t x=0;x<PATCH_SIZE;x+=1)
                {
                    in_data[ii] = (g_patch8888_2828[ii] > 0xFF000000)?1.0F:0.0F;
                    BSP_LCD_DrawPixel(x,y,g_patch8888_2828[ii++]);
                }
            }
            ...
        }
    }
}
```

```
void MX_X_CUBE_AI_Process(void)
{
    /* Get status and positions of the touch screen */
    if( BSP_TS_GetState(&ts_state) == TS_OK )
    {
        /* Touch event*/ --> /* Draw display*/
        /* Major block of drawing mini patch and running AI related code */
        if( data_ready && (HAL_GetTick() - touch_time) > TOUCH_TIMEOUT)
        {
            /* retrieve downsampled image 58*32 from display memory (480*272) in SDRAM */ -->
            /* Resize from 58*32 to 28*28 */ -->
            /* Draw 28*28 and prepare input array of AI model: in_data */ --> /* DWT reset*/ -->
            /* inference duration measurement */

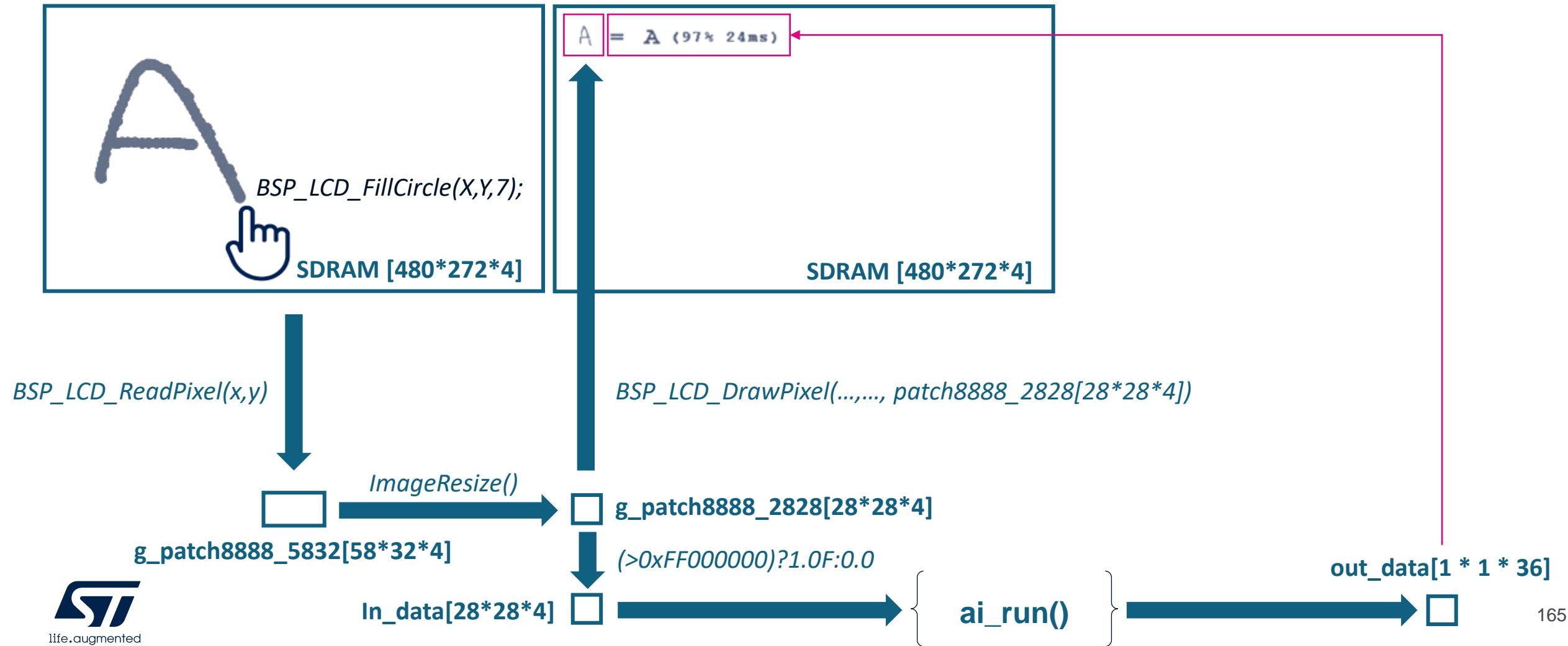
            /* run NN*/
            aiRun(in_data, out_data);

            /* show results*/
            max_probability = 0;
            max_index = -1;
            for(ii=0;ii<AI_NETWORK_OUT_1_SIZE;ii++)
            {
                if( out_data[ii]>max_probability ){ max_probability = out_data[ii]; max_index = ii; }
            }
            BSP_LCD_SetFont(&Font24);
            if(max_index >= 0 && max_probability > 0.5F)
            {
                prediction = (max_index <10) ? max_index+48 : max_index+55;
                sprintf(msg,"= %c",prediction);
                BSP_LCD_DisplayStringAt(38,4,(uint8_t*)msg,LEFT_MODE);
                BSP_LCD_SetFont(&Font16);
                sprintf(msg,"(%d%% %dms)",(int)(max_probability*100),t_ms_inference_duration.ms);
                BSP_LCD_DisplayStringAt(98,8,(uint8_t*)msg,LEFT_MODE);
            }
            else
            {
                BSP_LCD_DisplayStringAt(38,4,(uint8_t*)"= ?",LEFT_MODE);
            }
            data_ready = false;
        } // end of if( data_ready && (HAL_GetTick() - touch_time) > TOUCH_TIMEOUT)
    } // end of if( BSP_TS_GetState(&ts_state) == TS_OK )
}
```

STAGE 2 示意圖



STAGE 3示意圖



Thank you

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to www.st.com/trademarks.

All other product or service names are the property of their respective owners.

