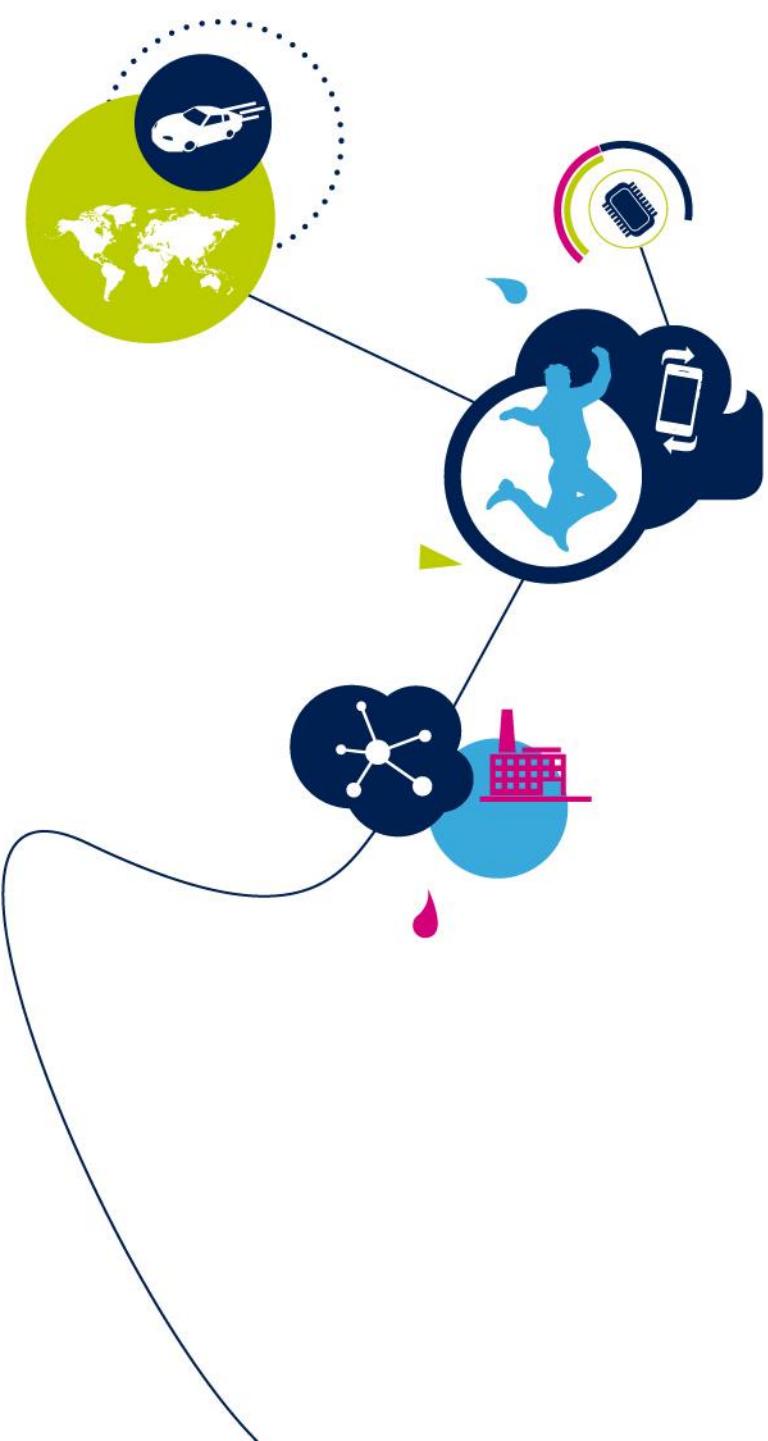




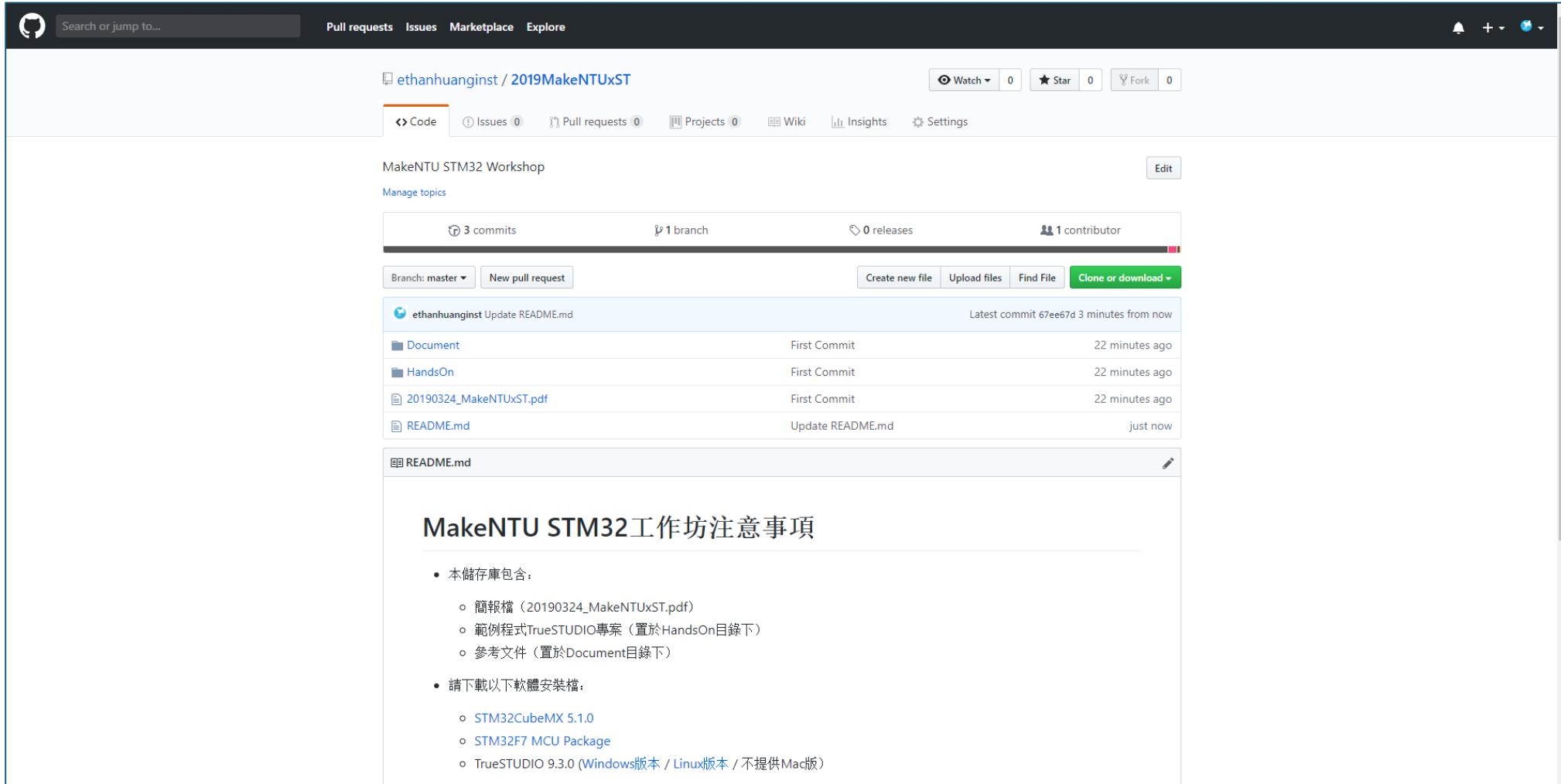
MakeNTU STM32 工作坊

2019.03.19

ST MCU Team



ethanhuanginst / 2019MakeNTUxST



Search or jump to...

Pull requests Issues Marketplace Explore

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

ethanhuanginst / 2019MakeNTUxST

MakeNTU STM32 Workshop

Manage topics

3 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

ethanhuanginst Update README.md Latest commit 67ee67d 3 minutes from now

Document First Commit 22 minutes ago

HandsOn First Commit 22 minutes ago

20190324_MakeNTUxST.pdf First Commit 22 minutes ago

README.md Update README.md just now

README.md

MakeNTU STM32工作坊注意事項

- 本儲存庫包含：
 - 簡報檔 (20190324_MakeNTUxST.pdf)
 - 範例程式TrueSTUDIO專案（置於HandsOn目錄下）
 - 參考文件（置於Document目錄下）
- 請下載以下軟體安裝檔：
 - STM32CubeMX 5.1.0
 - STM32F MCU Package
 - TrueSTUDIO 9.3.0 (Windows版本 / Linux版本 / 不提供Mac版)

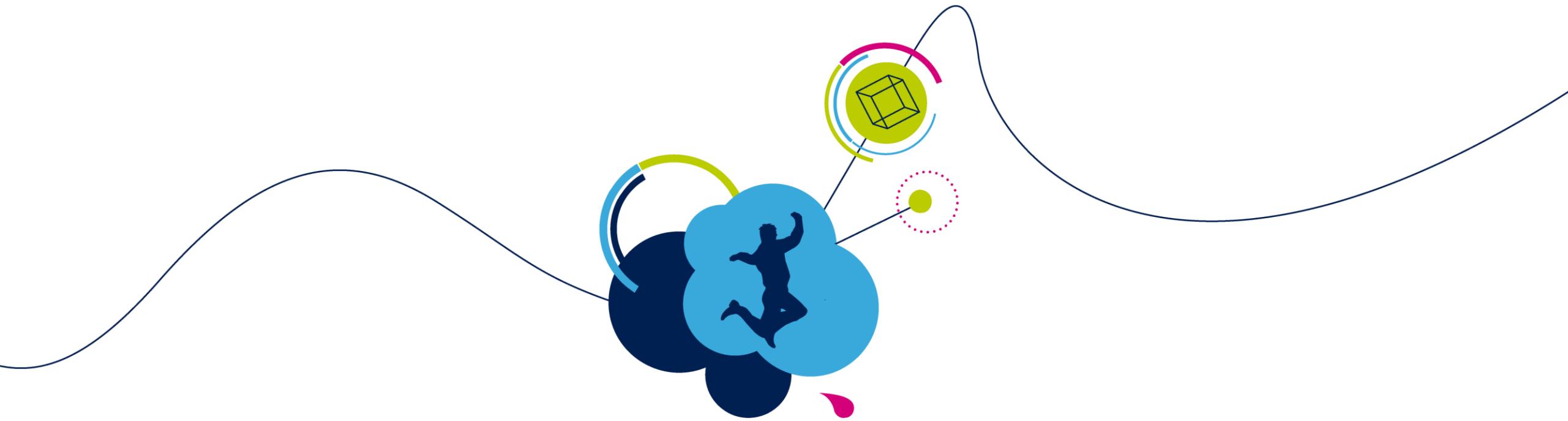
1. Atollic_TrueSTUDIO_for_STM32_linux_x86_64_v9.3.0_20190212-0734.tar.gz
2. Atollic_TrueSTUDIO_for_STM32_windows_x86_v9.3.0_20190212-0734.exe
3. en.SetupSTM32CubeMX-5.1.0-RC6.zip
4. en.STM32Cube_FW_F7_V1.15.0.zip

- 熟悉STM32開發環境
 - STM32 CUBEMX
 - TrueSTUDIO
- 熟悉以下功能
 - GPIO輸出作LED控制
 - GPIO中斷
 - printf的使用
 - UART的傳輸和接收

Agenda

5

Time	Length	Topic
12:30 –13:00	30mins	<p>STM32開發生態系統介紹</p> <ul style="list-style-type: none">• STM32F746 Discovery 開發板• STM32CUBE介紹和安裝• TrueSTUDIO安裝
13:00 – 14:30	90mins	<p>Hands-on動手實作</p> <ol style="list-style-type: none">1. GPIO 輸出控制2. GPIO 中斷3. printf over UART4. UART Polling5. UART-DMA <p>額外練習</p> <ul style="list-style-type: none">• Timer• Timer-PWM



STM32 開發生態系統

STM32 開發生態系統

7

STM32 32-bit ARM Cortex MCUs



STM32 Ecosystem



Software tools

STM32CubeMX
Configuration and
initialization tool

Integrated Development
Environments (IDE)

STM Studio
Monitoring tool

[More software tools](#)



arm KEIL



Embedded software

STM32Cube MCU Packages

STM32Cube Expansion
Packages

[More embedded software](#)

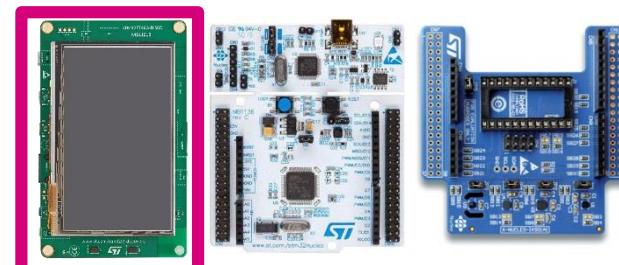


Hardware tools

STM32 Nucleo
development boards,
Discovery kits,
Evaluation boards

STM32 Nucleo
expansion boards

ST-LINK in-circuit
debugger/programmer



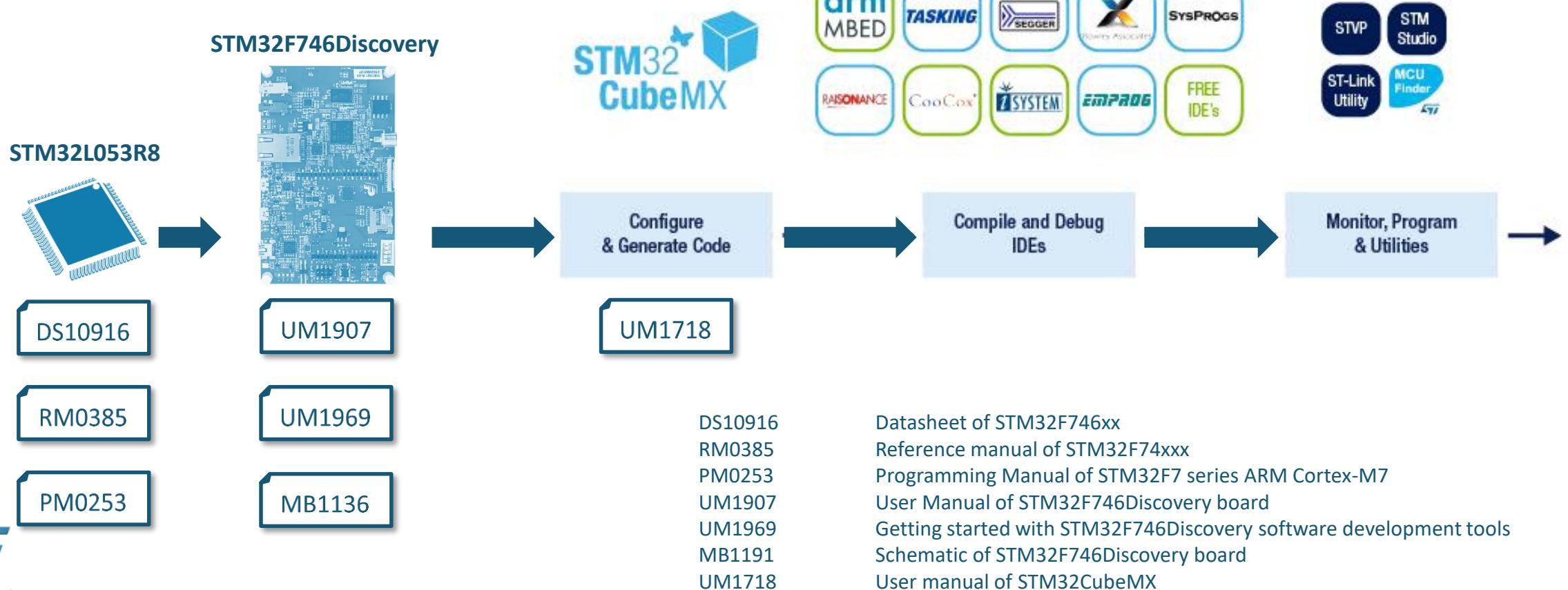
STM32 開發工具

8



STM32 Software Development Tools





參考文件(2)

<https://www.st.com/en/microcontrollers-microprocessors/stm32f746ng.html>

QUICK VIEW	RESOURCES	TOOLS AND SOFTWARE	SAMPLE & BUY	QUALITY & RELIABILITY
Application Notes				
		Description	Version	Size
	AN3126	Audio and waveform generation using the DAC in STM32 microcontrollers	3.0	816.19 KB
	AN4775	Basics and low-cost solution proposals to move from legacy USB2.0 connector to USB Type-C™ connector with STM32 devices	1.0	443.47 KB
	AN3154	CAN protocol used in the STM32 bootloader	6.0	630.15 KB
	AN5020	Digital camera interface (DCMI) on STM32 MCUs	1.0	2.03 MB
	AN1181	Electrostatic discharge sensitivity measurement	1.6	49.01 KB
	AN4566	Extending the DAC performance of STM32 microcontrollers	2.0	365.16 KB
	AN4776	General-purpose timer cookbook	2.0	1.95 MB
	AN4661	Getting started with STM32F7 Series MCU hardware development	5.0	1.09 MB
	AN4750	Handling of soft errors in STM32 applications	2.1	410.84 KB
	AN4803	High-speed SI simulations using IBIS and board-level simulations using HyperLynx SI on STM32 32-bit ARM® Cortex® MCUs	1.1	2.03 MB
	AN2834	How to get the best ADC accuracy in STM32 microcontrollers	3.1	1.08 MB
	AN4229	How to implement a vocoder solution using STM32 microcontrollers	1.1	466 KB
	AN4221	I2C protocol used in the STM32 bootloader	6.0	686.2 KB
	AN5156	Introduction to STM32 microcontrollers security	2.0	3.07 MB
	AN4861	LCD-TFT display controller (LTDC) on STM32 MCUs	2.0	2.21 MB
	AN4839	Level 1 cache on STM32F7 Series and STM32H7 Series	2.0	151.16 KB

QUICK VIEW	RESOURCES	TOOLS AND SOFTWARE	SAMPLE & BUY	QUALITY & RELIABILITY
X-CUBE-32F7PERF				
	ST	STM32F7 performance software expansion for STM32Cube (AN4667)		
	X-CUBE-AI	AI expansion pack for STM32CubeMX		
	X-CUBE-AUDIO	Audio effects software expansion for STM32Cube		
	X-CUBE-CRYPTOLIB	STM32 cryptographic firmware library software expansion for STM32Cube (UM1924)		
	X-CUBE-DSPDEMO	Digital Signal Processing with STM32, software expansion for STM32Cube (AN4841)		
	X-CUBE-FPUDEMO	Floating point unit demonstration on STM32 microcontrollers, software expansion for STM32Cube (AN4044)		
	X-CUBE-IAP-USART	STM32Cube in-application programming using the USART embedded software (AN4657)		
	X-CUBE-LPDEMO-F7	Managing low-power consumption on STM32F7 series microcontrollers, software expansion for STM32Cube (AN4749)		
	X-CUBE-LPTIMER	Low-power timer (LPTIM) applicative use-cases on STM32 MCUs software expansion for STM32Cube (AN4865)		
	X-CUBE-RTC	Real Time Clock (RTC) software expansion for STM32Cube (AN4759)		
	X-CUBE-SMBUS	STM32Cube SMBus/PMBus embedded software (AN4502)		
Evaluation Tools				
PRODUCT EVALUATION TOOLS				
Part Number	▲ Manufacturer	◆ Description		
32F723EDISCOVERY	ST	Discovery kit with STM32F723IE MCU		
32F746GDISCOVERY	ST	Discovery kit with STM32F746NG MCU		
32F769IDISCOVERY	ST	Discovery kit with STM32F769NI MCU		
EasyMx PRO v7 for STM32	MikroElektronika	Development board for STM32 ARM Cortex M0, M3, M4, and M7 mcus.		
I-NUCLEO-NETX	Hilscher	netSHIELD industrial ethernet protocols expansion board for STM32 Nucleo		
mikromedia 7 for STM32F7	MikroElektronika	Vivid 7" touch display and a development platform with STM32F746ZG mcu, including Ethernet and WiFi.		
mikromedia Plus for STM32F7	MikroElektronika	Smart 4.3" programmable touch display with STM32F746ZG mcu.		

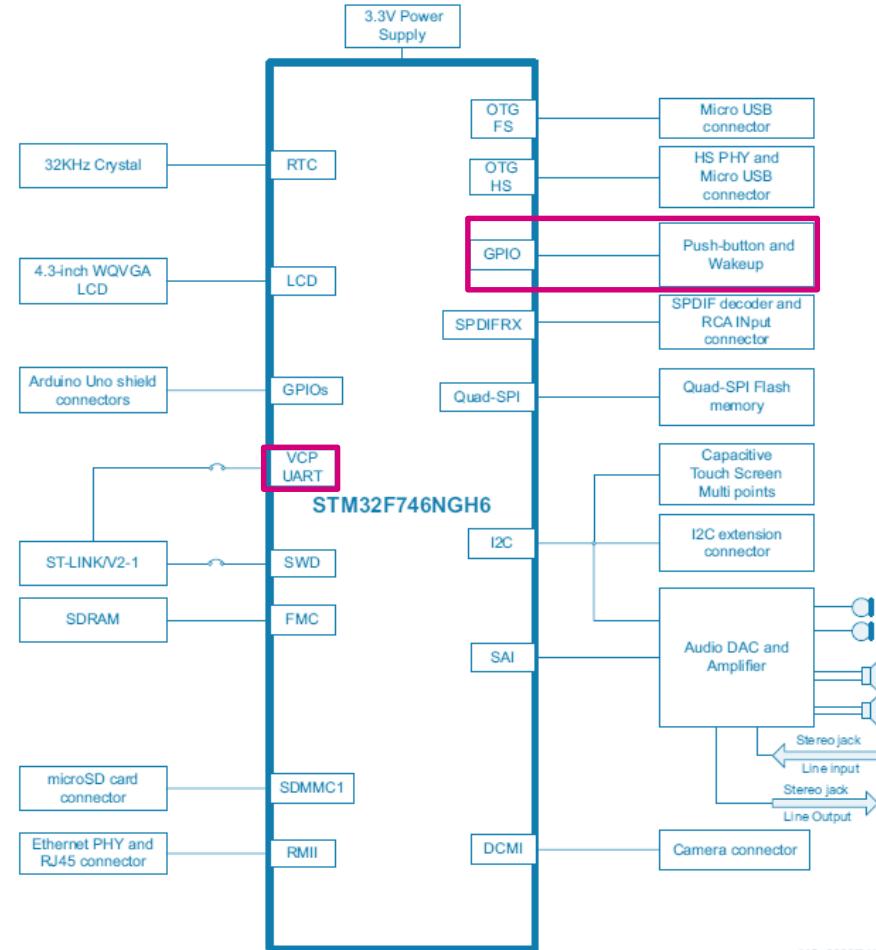


STM32F746 Discovery 開發板



開發板功能方塊圖

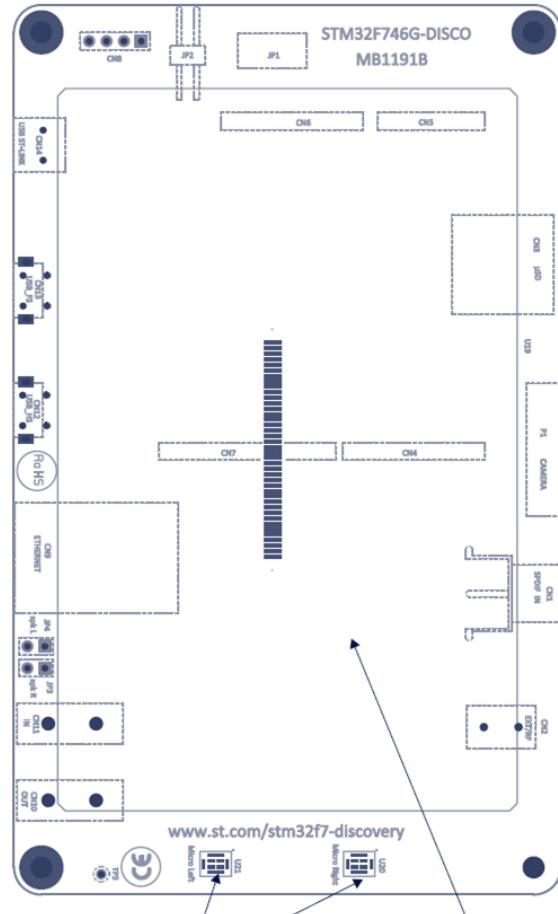
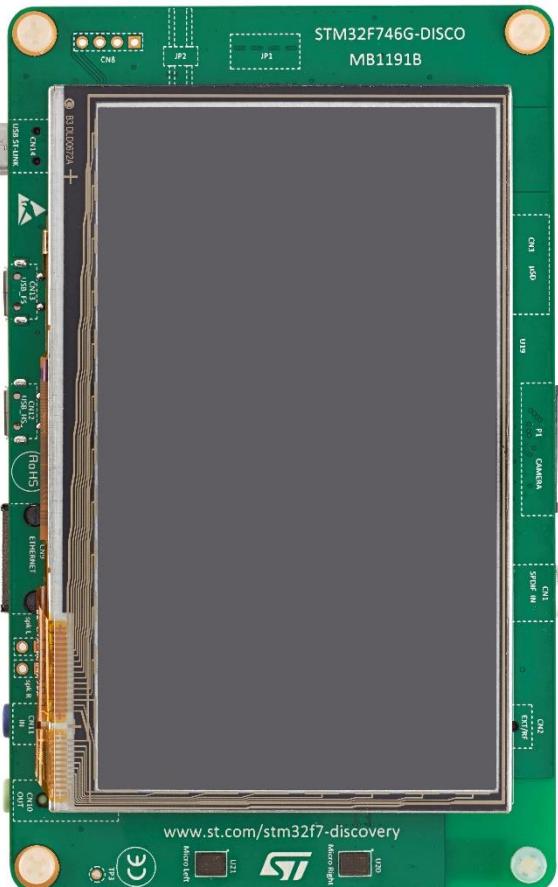
12





開發板說明

13



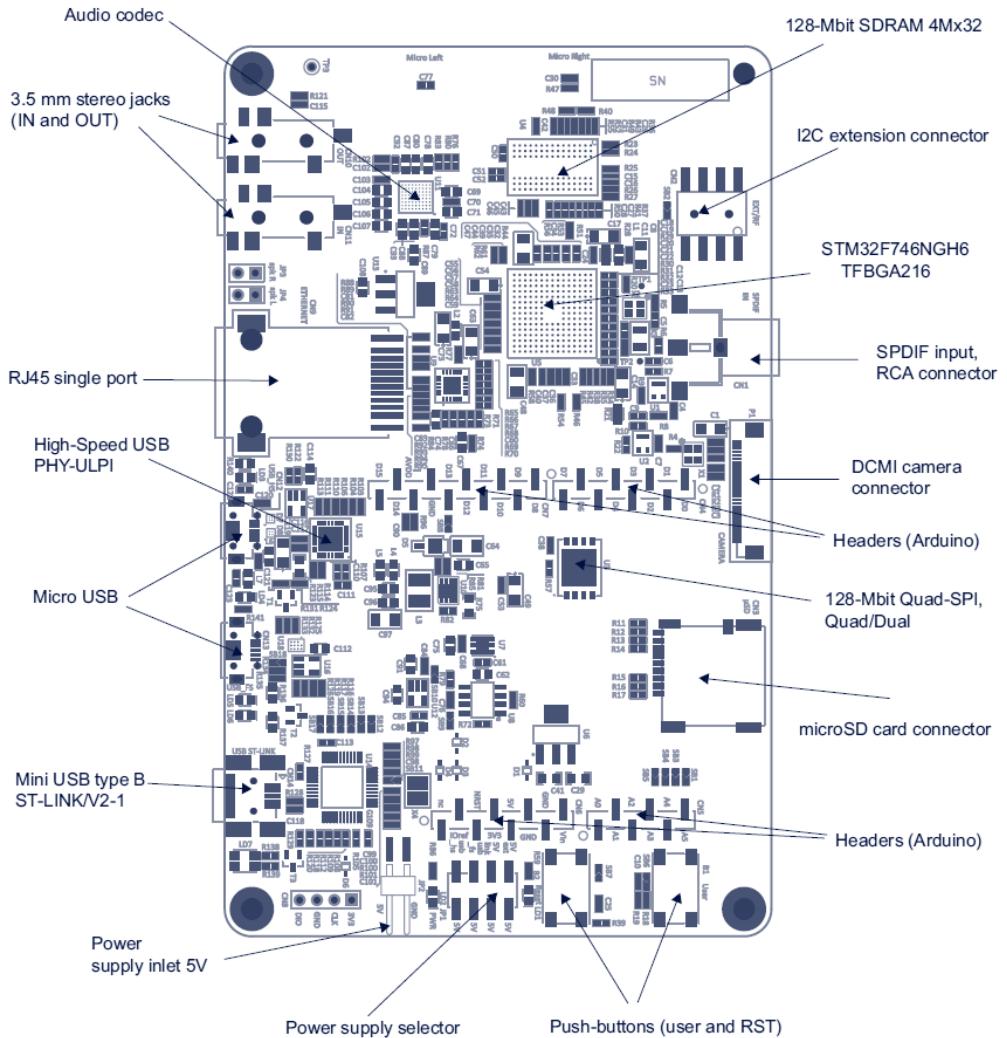
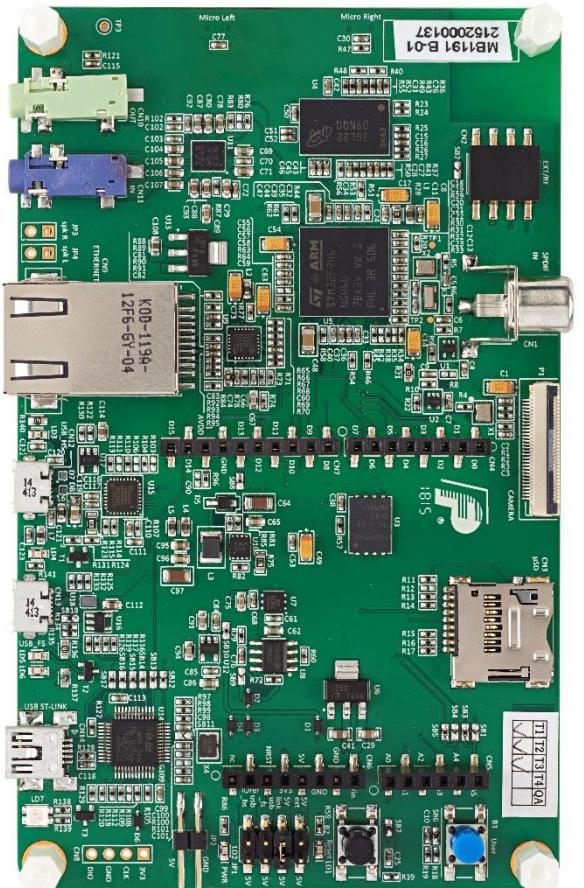
Two ST-MEMS microphones

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



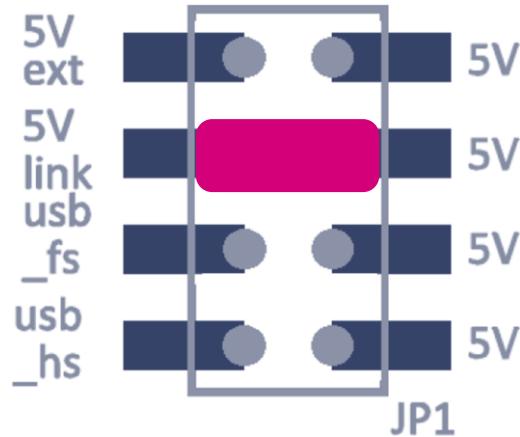
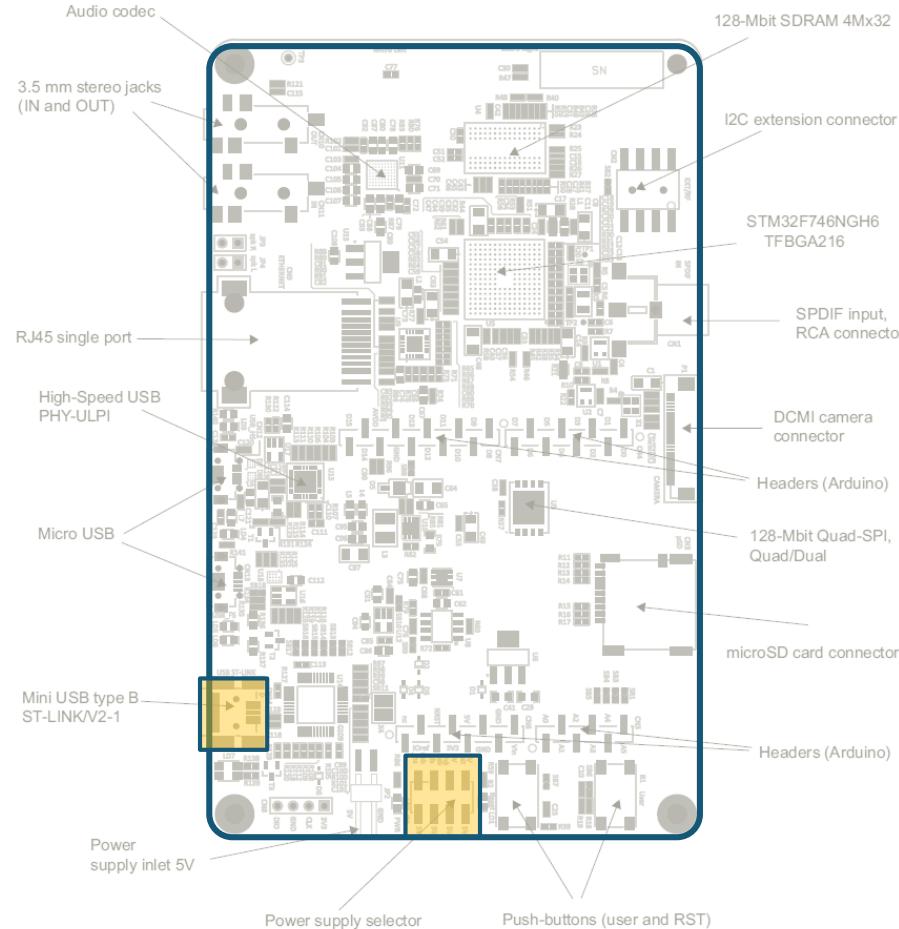
開發板說明

14



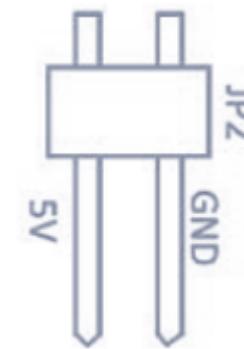
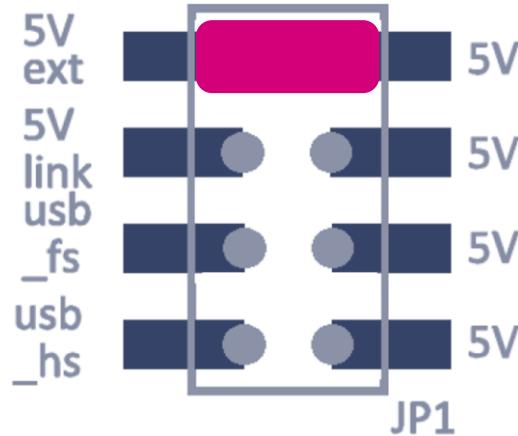
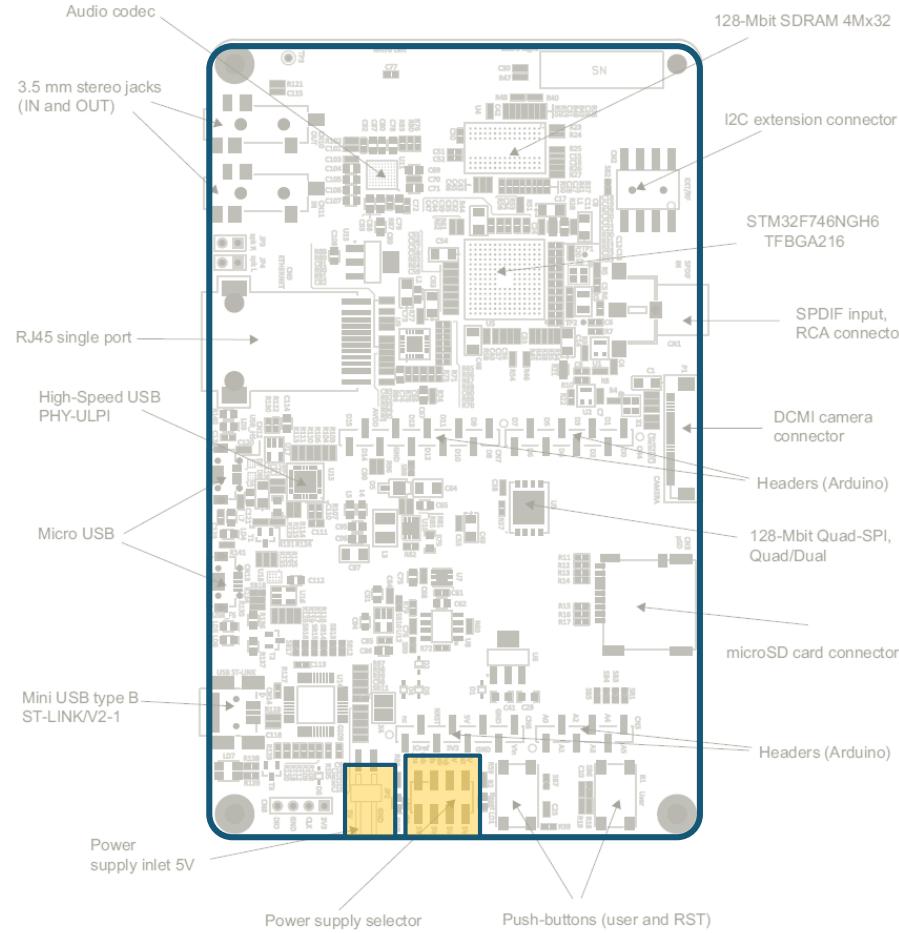
電源供應 (1)

15



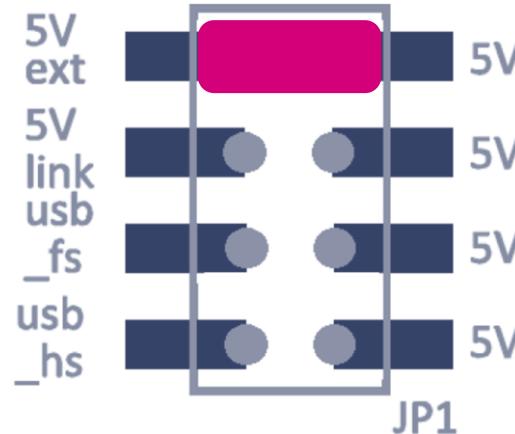
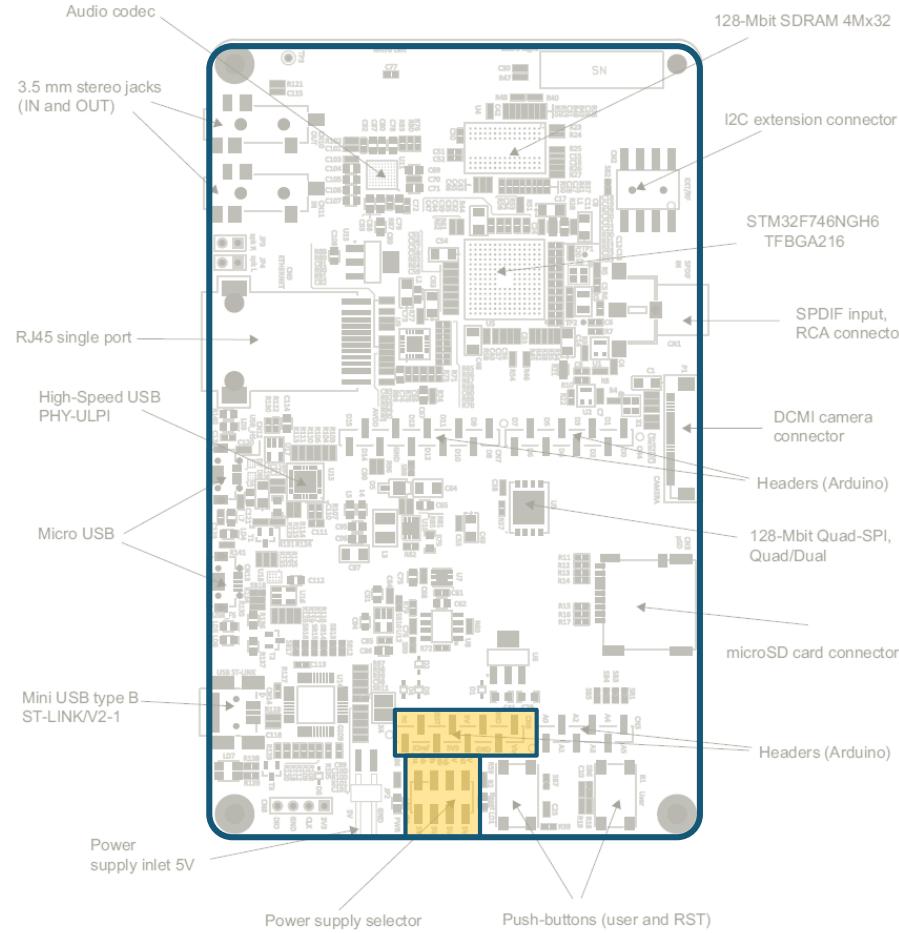
電源供應 (2)

16

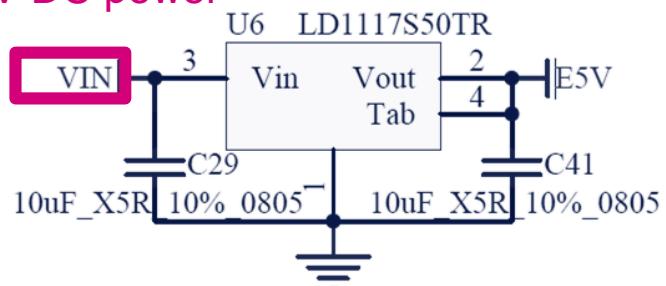


電源供應 (3)

17

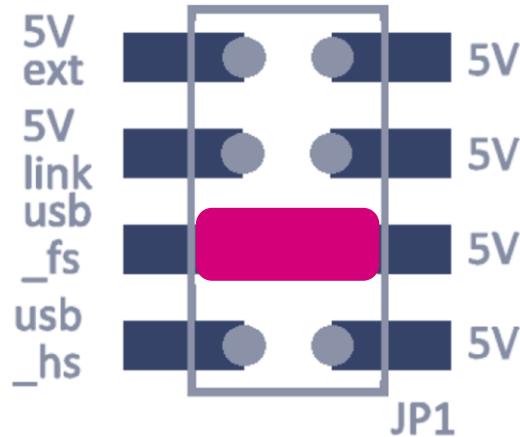
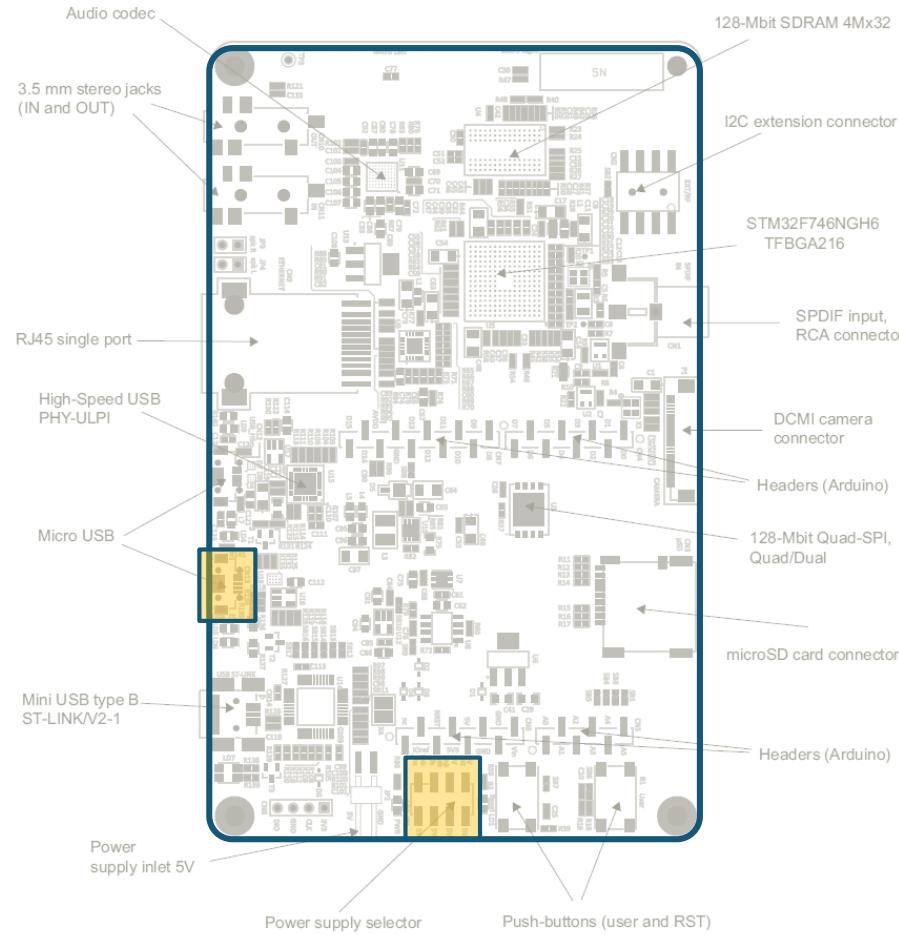


7-12V DC power



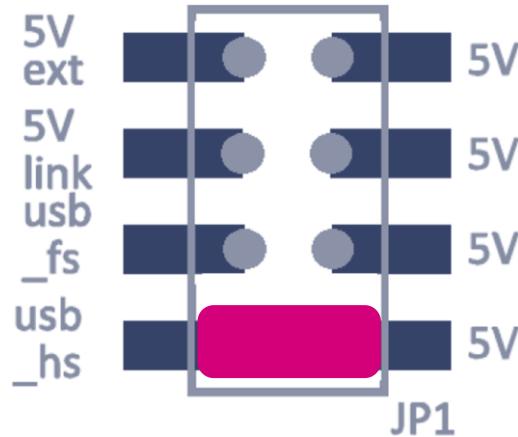
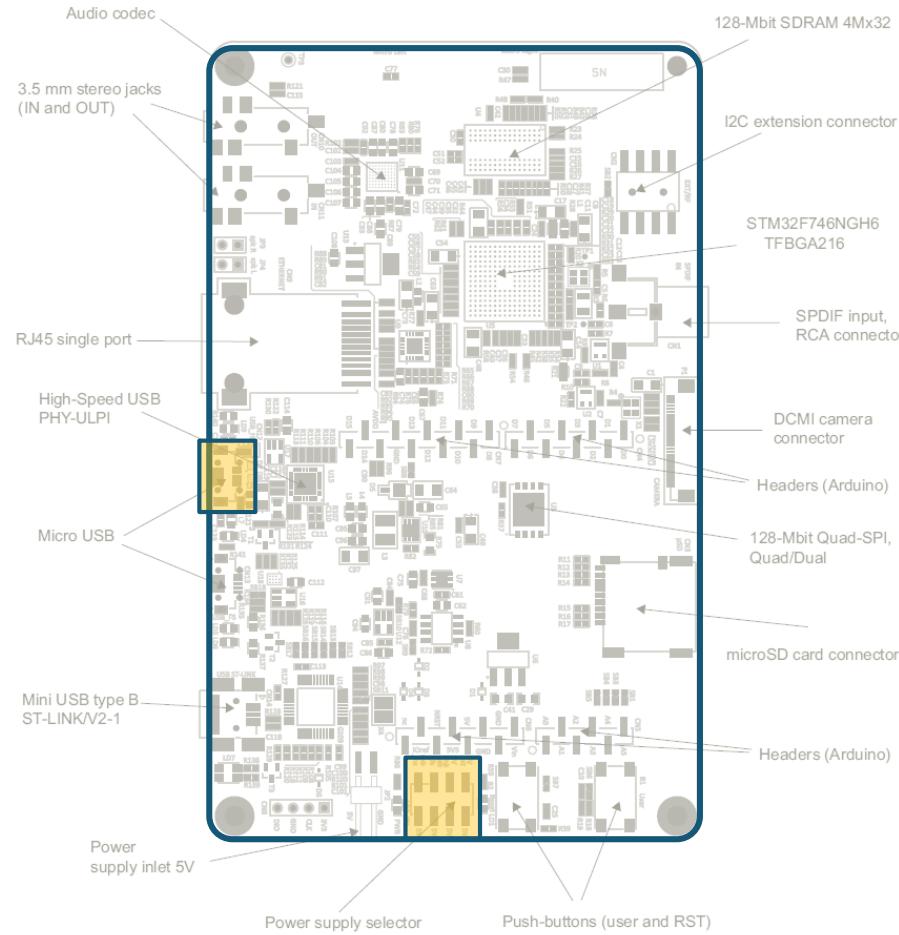
電源供應 (4)

18



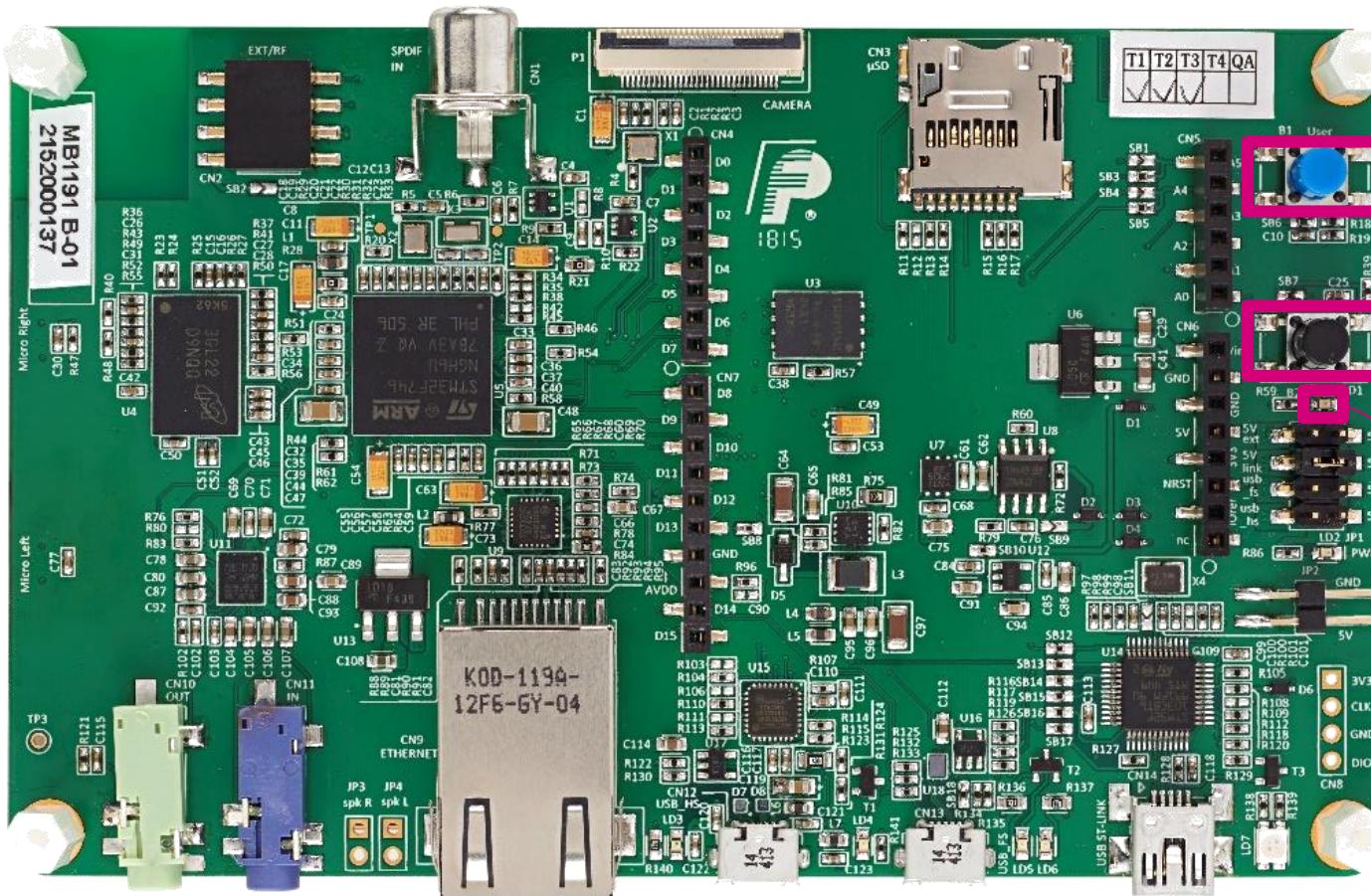
電源供應 (5)

19



按鈕及可控制LED

20



藍色按鈕

USER

黑色按鈕

RESET

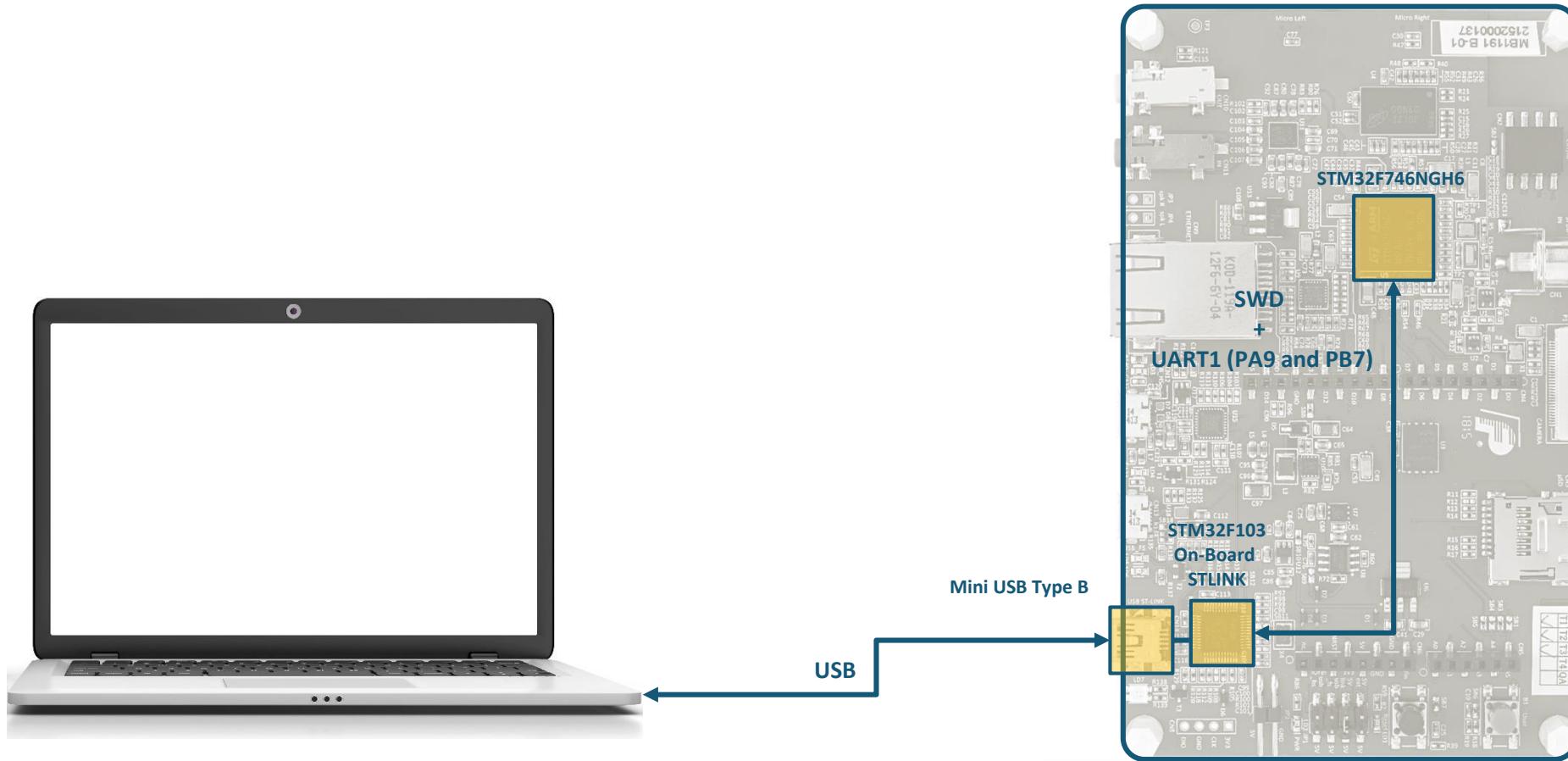
LD1

可控制LED



虛擬串列埠 (Virtual Comport)

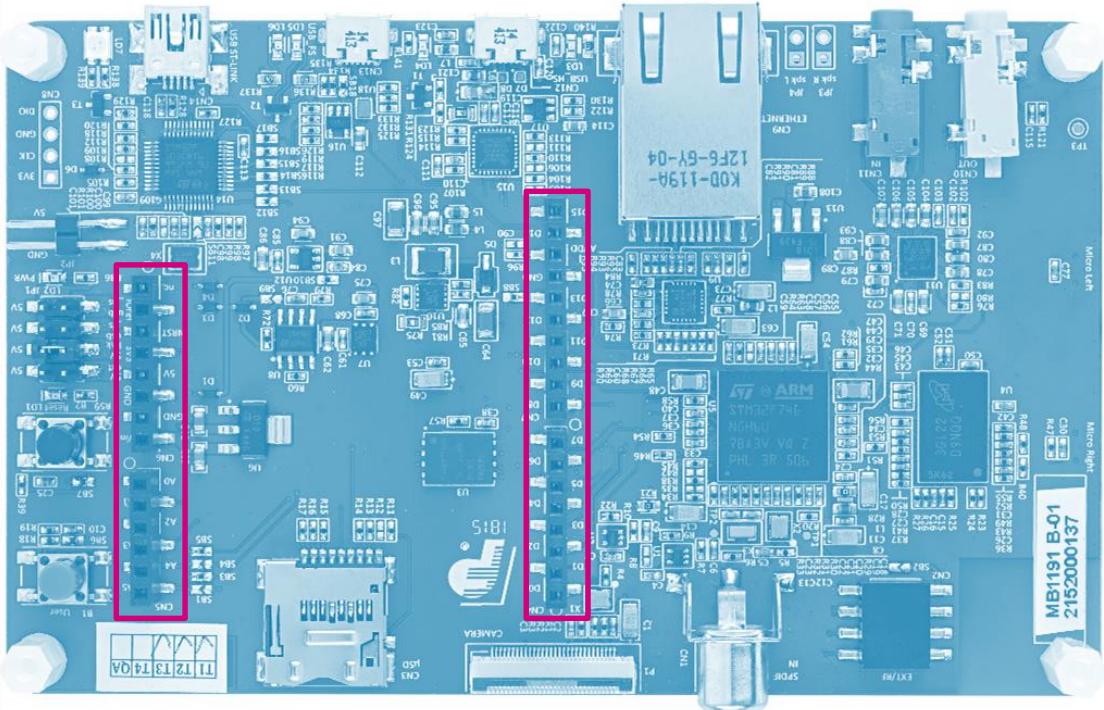
21





Arduino Connectors

22



CN6	
Reserved	1 NC
3V3	2 IOREF
NRST	3 RESET
3V3	4 3V3
5V	5 5V
GND	6 GND
GND	7 GND
VIN	8 VIN

CN5

PA0	1 A0
PF10	2 A1
PF9	3 A2
PF8	4 A3
PF7	5 A4
PF6	6 A5

CN4

CN7	
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



STM32CUBE

STM32開發生態系統



STM32 Ecosystem



Software tools

STM32CubeMX
Configuration and initialization tool

Integrated Development Environments (IDE)

STM Studio
Monitoring tool

[More software tools](#)



arm KEIL



STM32Cube MCU Packages

STM32Cube Expansion Packages

[More embedded software](#)

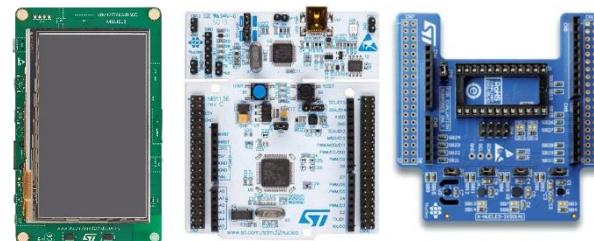


Hardware tools

STM32 Nucleo development boards,
Discovery kits,
Evaluation boards

STM32 Nucleo expansion boards

ST-LINK in-circuit debugger/programmer



Join the STM32 Community!
community.st.com/stm32

STM32開發工具

25



STM32 Software Development Tools



Configure
& Generate Code

Compile and Debug
IDEs

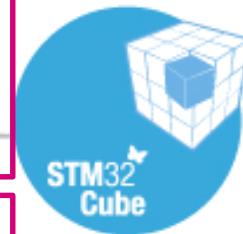
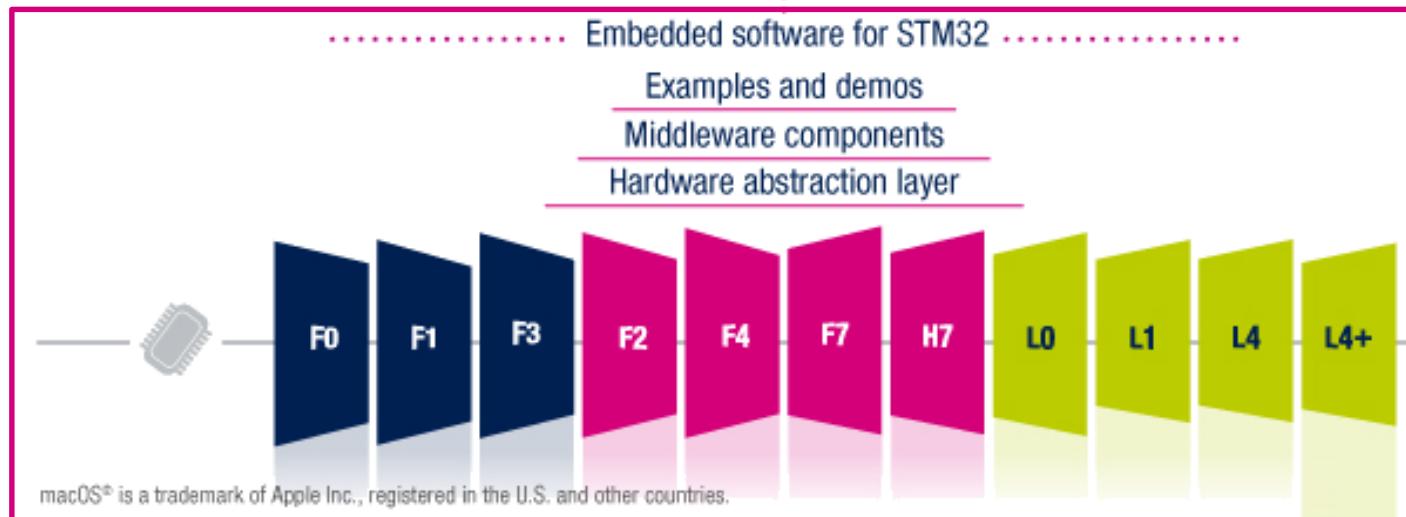
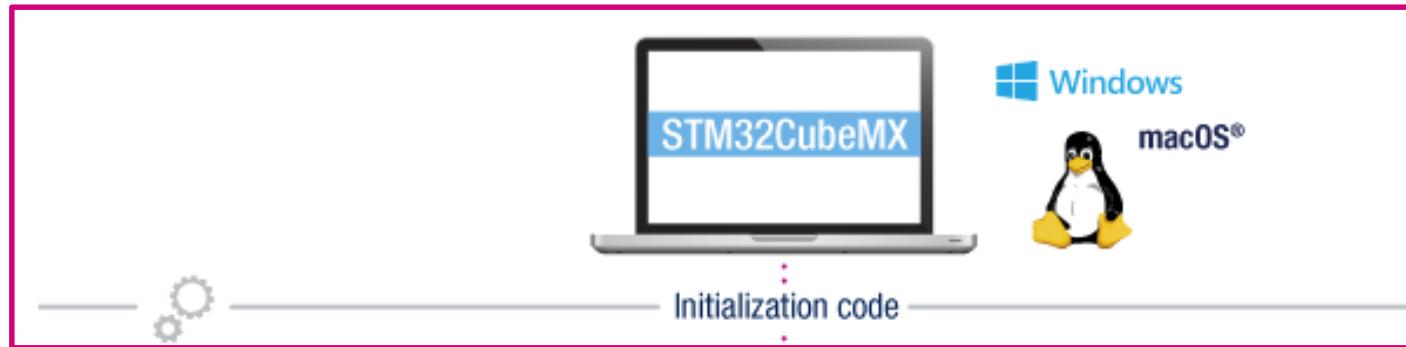
Monitor, Program
& Utilities



STM32 Ecosystem

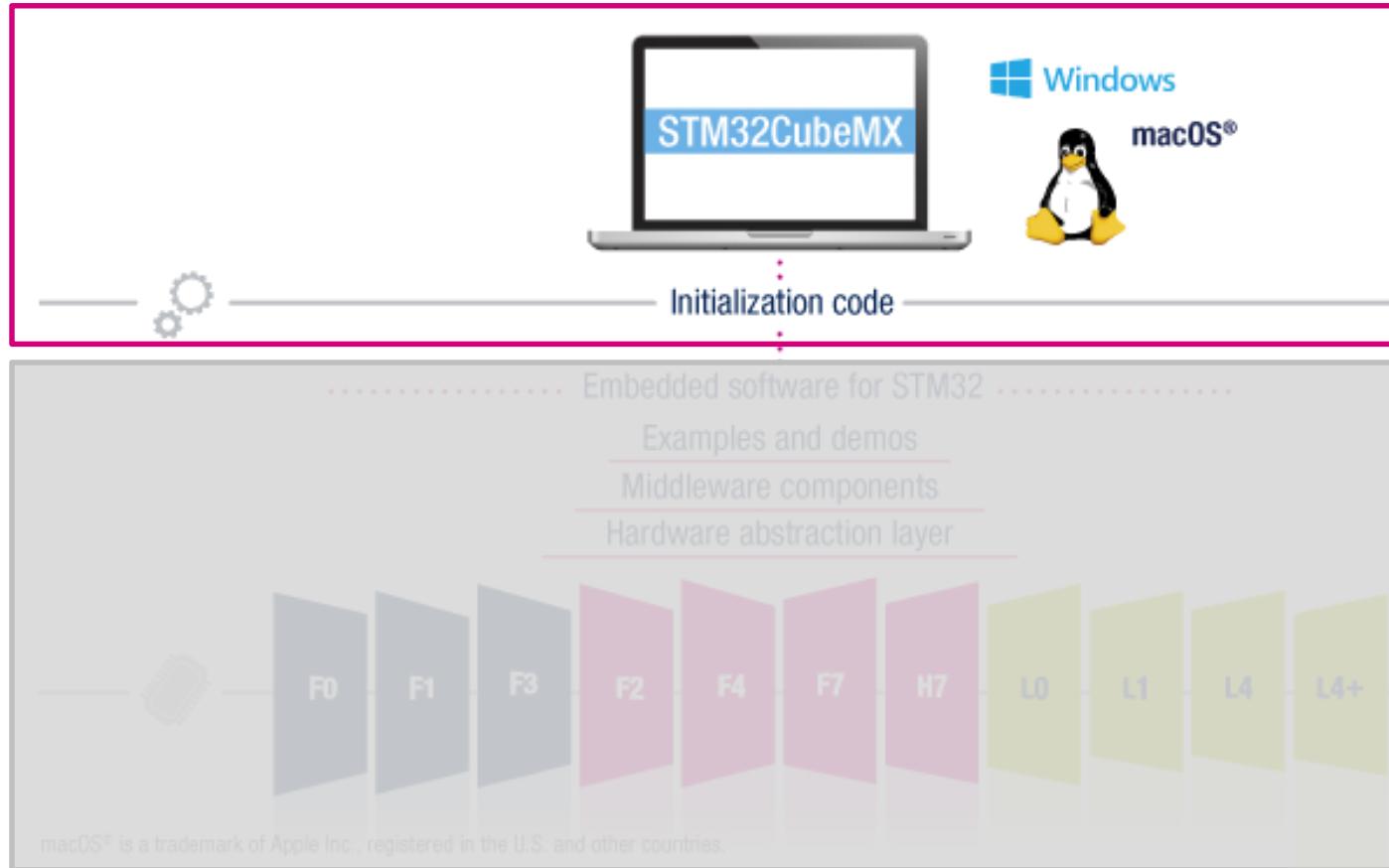


STM32CubeMX



STM32CubeMX (1)

27



- Intuitive STM32 microcontroller selection
- Microcontroller graphical configuration:
 - Pinout with automatic conflict resolution
 - Clock tree with dynamic validation of configuration
- Peripherals and middleware functional modes and initialization with dynamic validation of parameter constraints
- Power sequence with estimate of consumption results
- C code project generation with IAR™, Keil™ and GCC compilers.
- Available as a standalone software running on Windows®, Linux® and macOS® operating systems, or through Eclipse plug-in

STM32CubeMX (2)

28

MCU 選擇

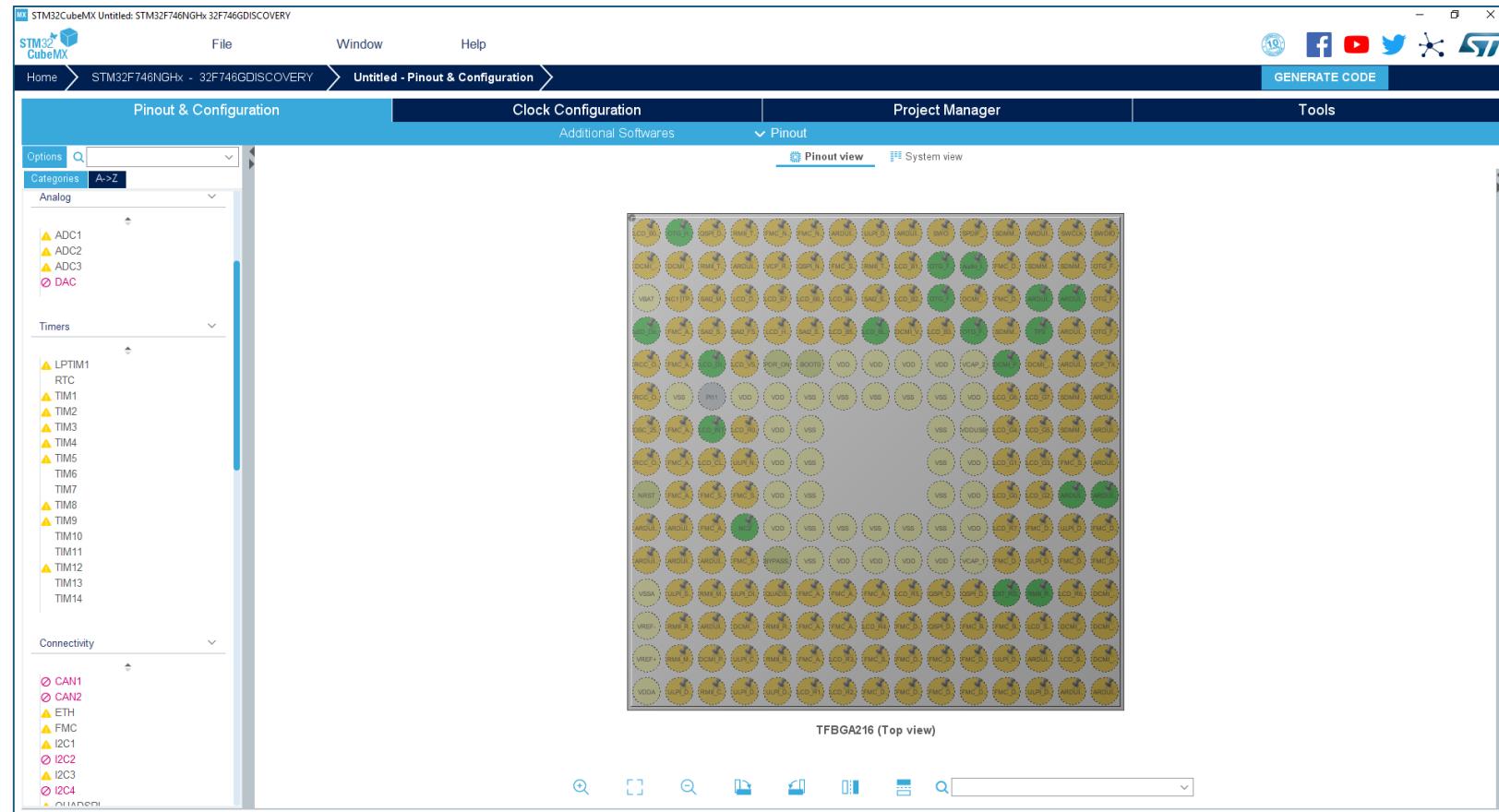
The screenshot shows the STM32CubeMX software interface. On the left, there is a sidebar with 'Existing Projects' and 'Recent Opened Projects'. The main area is divided into 'New Project' and 'Manage software installations' sections. The 'New Project' section contains buttons for 'Start My project from MCU' (with 'ACCESS TO MCU SELECTOR') and 'Start My project from STBoard' (with 'ACCESS TO BOARD SELECTOR'). A large central image features the text 'New multicore STM for Industrial and IoT' and an STM32MP1 chip. On the right, a 'New Project from a Board' dialog box is open, showing a list of boards. The dialog includes sections for 'Board Filters' (Part Number Search, Vendor, Type, Check/Uncheck All), 'MCU Series' (Check/Uncheck All, STM32F0, STM32F1, STM32F2), and a 'Boards List' table with 107 items. The table columns are: Overview, Part No, Type, Marketing Status, Unit Price (US\$), Mounted Device, Kit Cont., and Included in. The first two rows are visible:

	Overview	Part No	Type	Marketing Status	Unit Price (US\$)	Mounted Device	Kit Cont.	Included in
★		32F0308DISCOVERY	Discovery	Active	8.9	STM32F0308Tx		
★		32F072BDISCOVERY	Discovery	Active	10.4	STM32F072Rx		

STM32CubeMX (3)

29

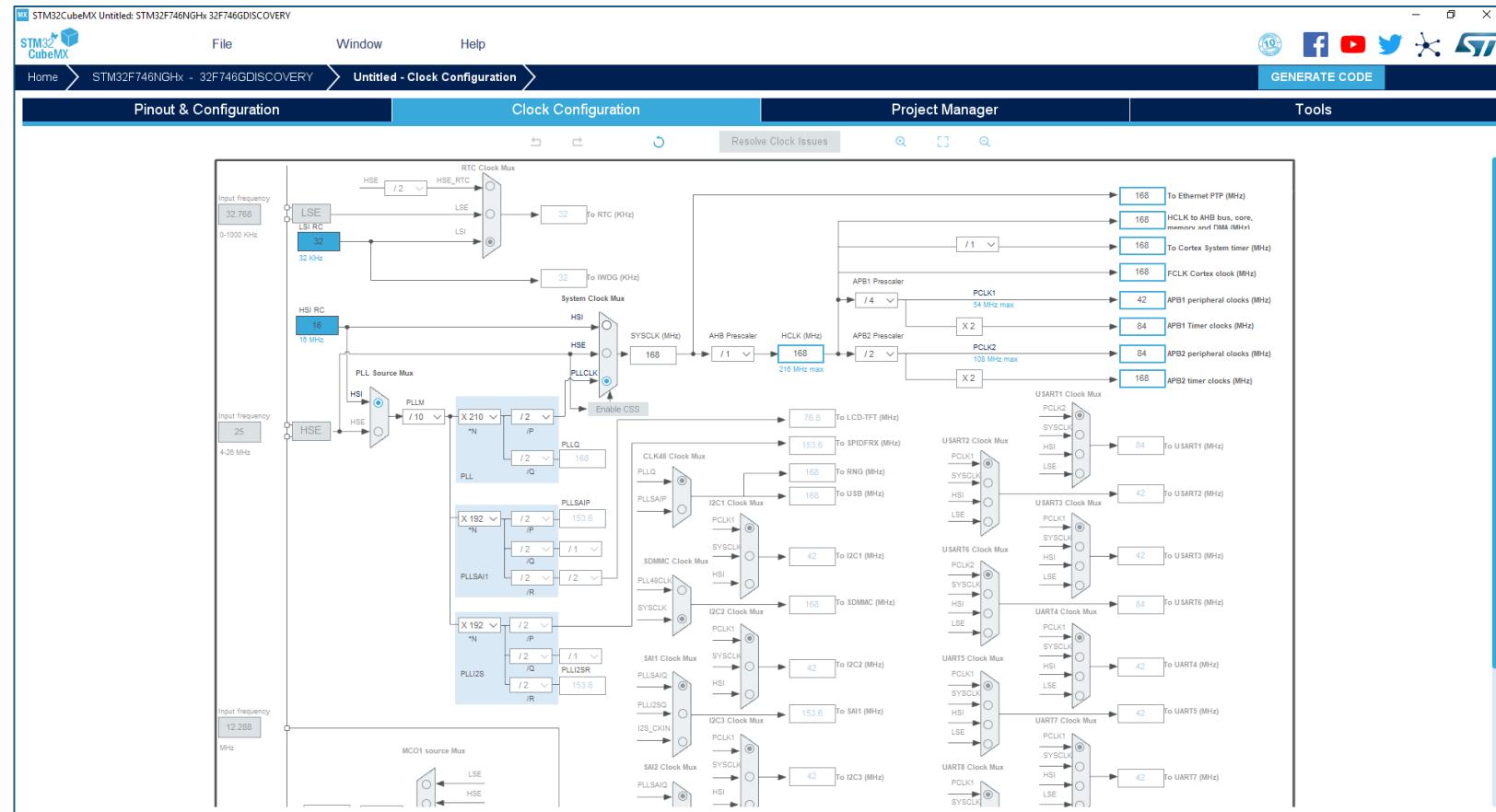
週邊及功能的開啟



STM32CubeMX (4)

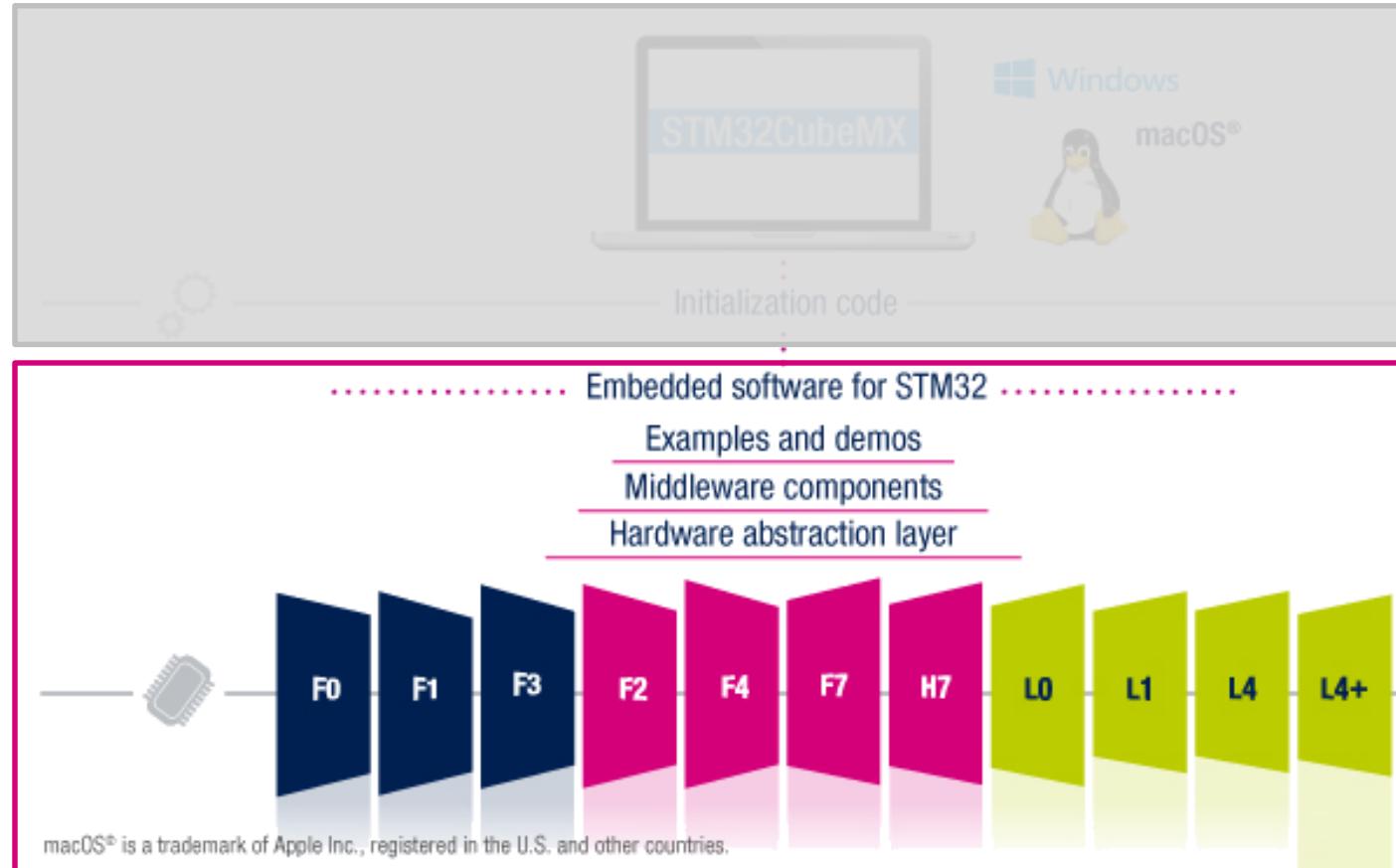
30

Clock Tree



STM32Cube MCU Package (1)

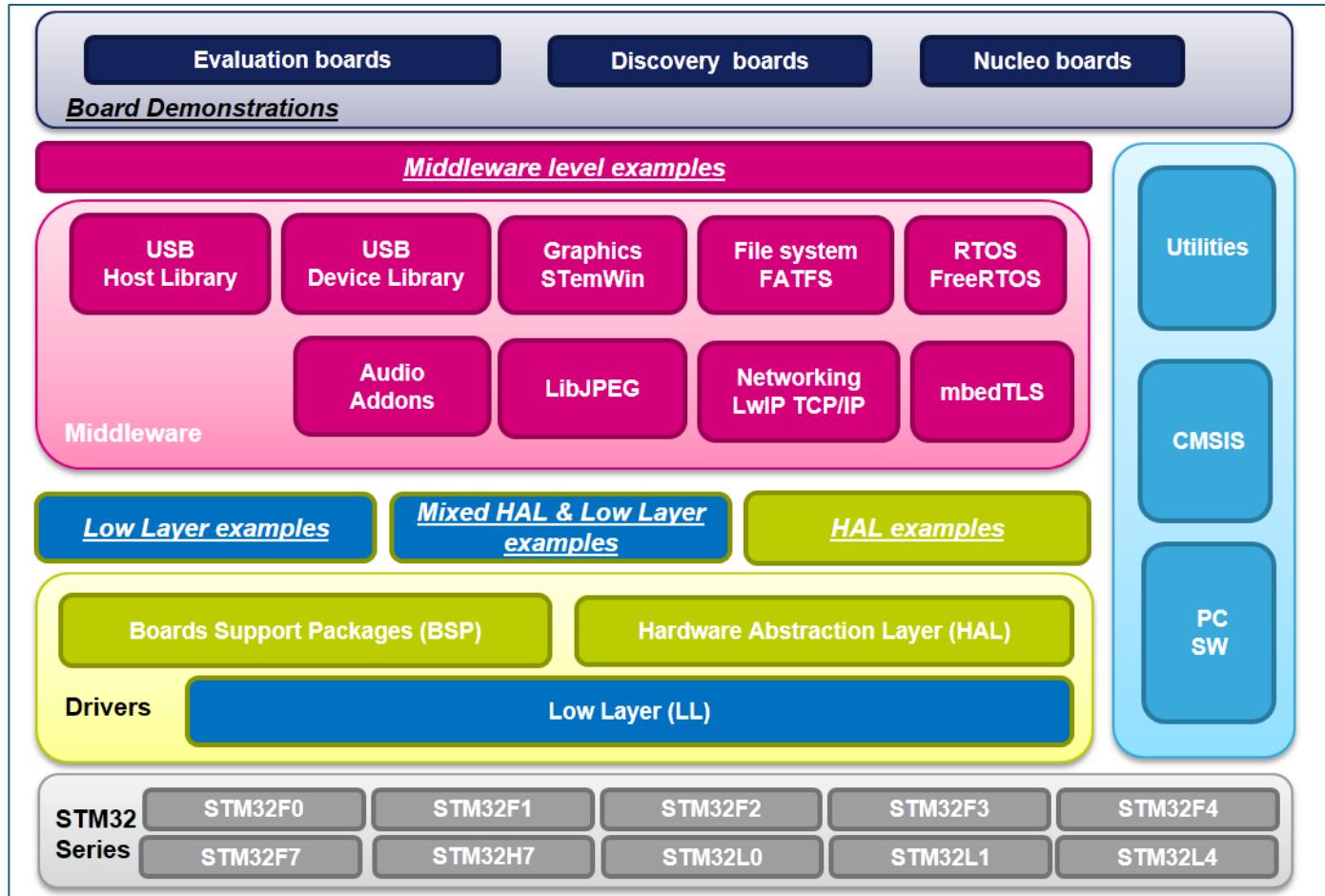
31



- Consistent and complete embedded software offer that frees the user from dependency issues
- Maximized portability between all STM32 Series supported by STM32Cube
- Hundreds of examples for easy understanding
- High quality HAL and low-layer API drivers using CodeSonar® static analysis tool
- STM32F4-dedicated middleware including USB Host and Device, and TCP/IP
- Free user-friendly license terms
- Update mechanism that can be enabled by the user to be notified of new releases

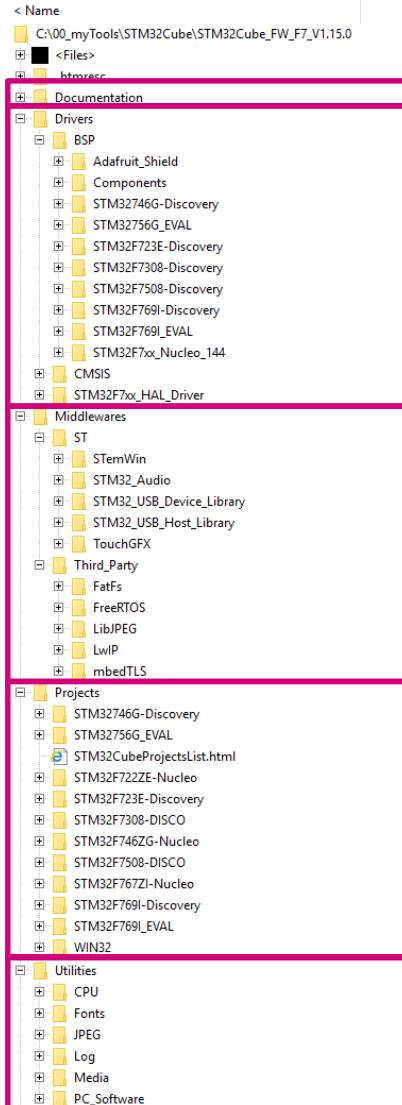
STM32Cube MCU Package (2)

32



STM32CubeF7 MCU Package (1)

33

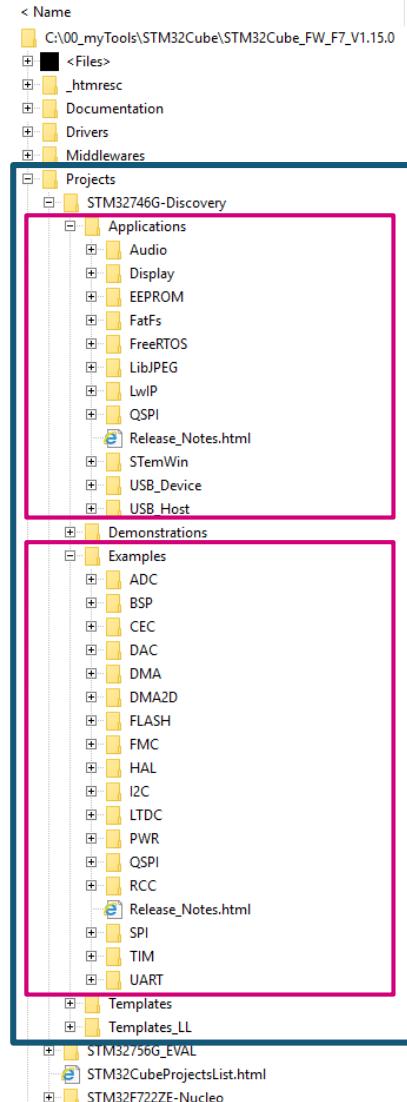


STM32CubeF7 FW package 包含下幾個部份:

- Drivers
 - Drivers\BSP
開發板的驅動程式
 - Drivers\CMSIS
STM32f7xx CMSIS 檔案(週邊暫存器的定義和宣告和位址的對映)
 - Drivers\STM32L0xx_HAL_Driver:
STM32f7xx各類週邊的 HAL驅動程式
- Middlewares
 - 由ST提供的middlewares (如USB host、USB device和audio)
 - 由第三方提供的middlewares (如FasFS和FreeRTOS)
- Projects
開發板的範例程式
- Utilities
其餘工具

STM32CubeF7 MCU Package (2)

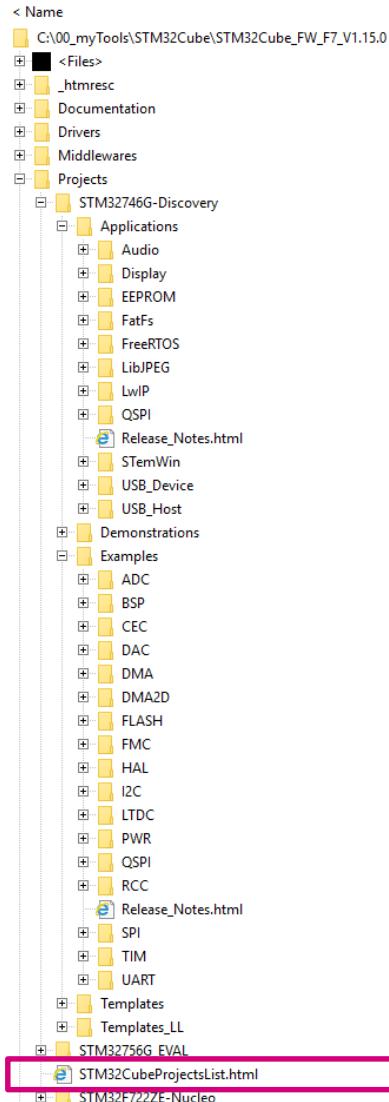
34



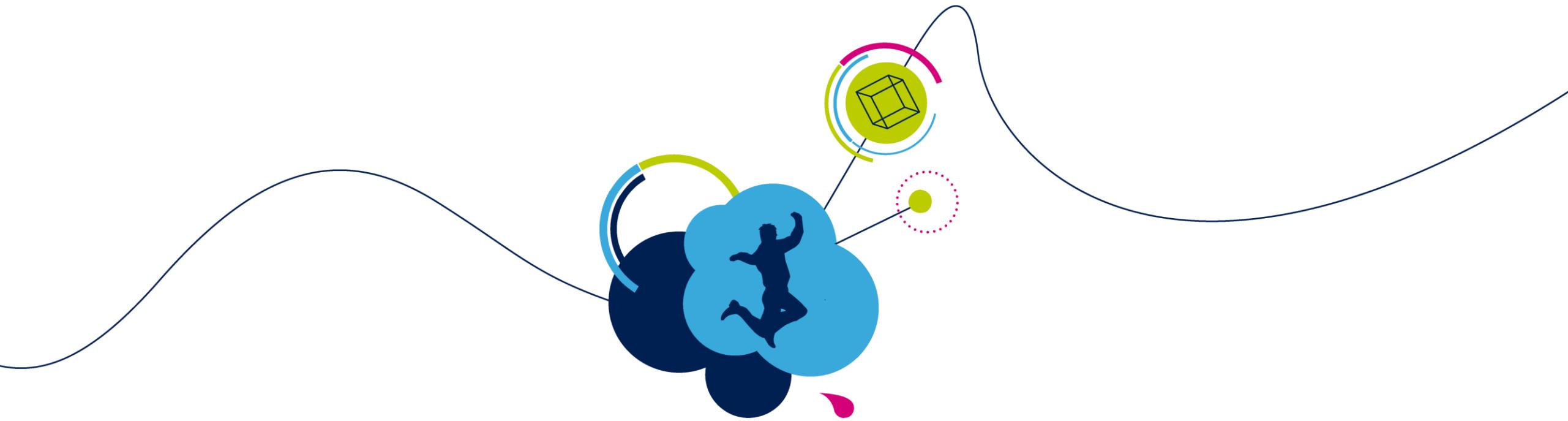
- STM32Cube 範例程式在STM32Cube FW package當中的“Project” 目錄
- 開發板範例程式可用EWARM、MDK-ARM和TrueSTUDIO 開啟
- 範例程式依STM32Cube 的層級分為以下幾類：
 - Example 只用到HAL 驅動的範例
 - Application 使用到HAL 驅動和middleware的範例
 - Demonstration 使用到HAL驅動、middleware和BSP的範例
- 所有範例的檔案目錄結構都是使用如下的結構：
 - \Inc 使用者程式碼檔頭檔 (header file)
 - \Src 使用者程式碼
 - \EWARM, \MDK-ARM \TrueSTUDIO IDE專案檔
 - readme.txt 說明範例執行的行為和開發環境的設定

STM32CubeF7 MCU Package (3)

35



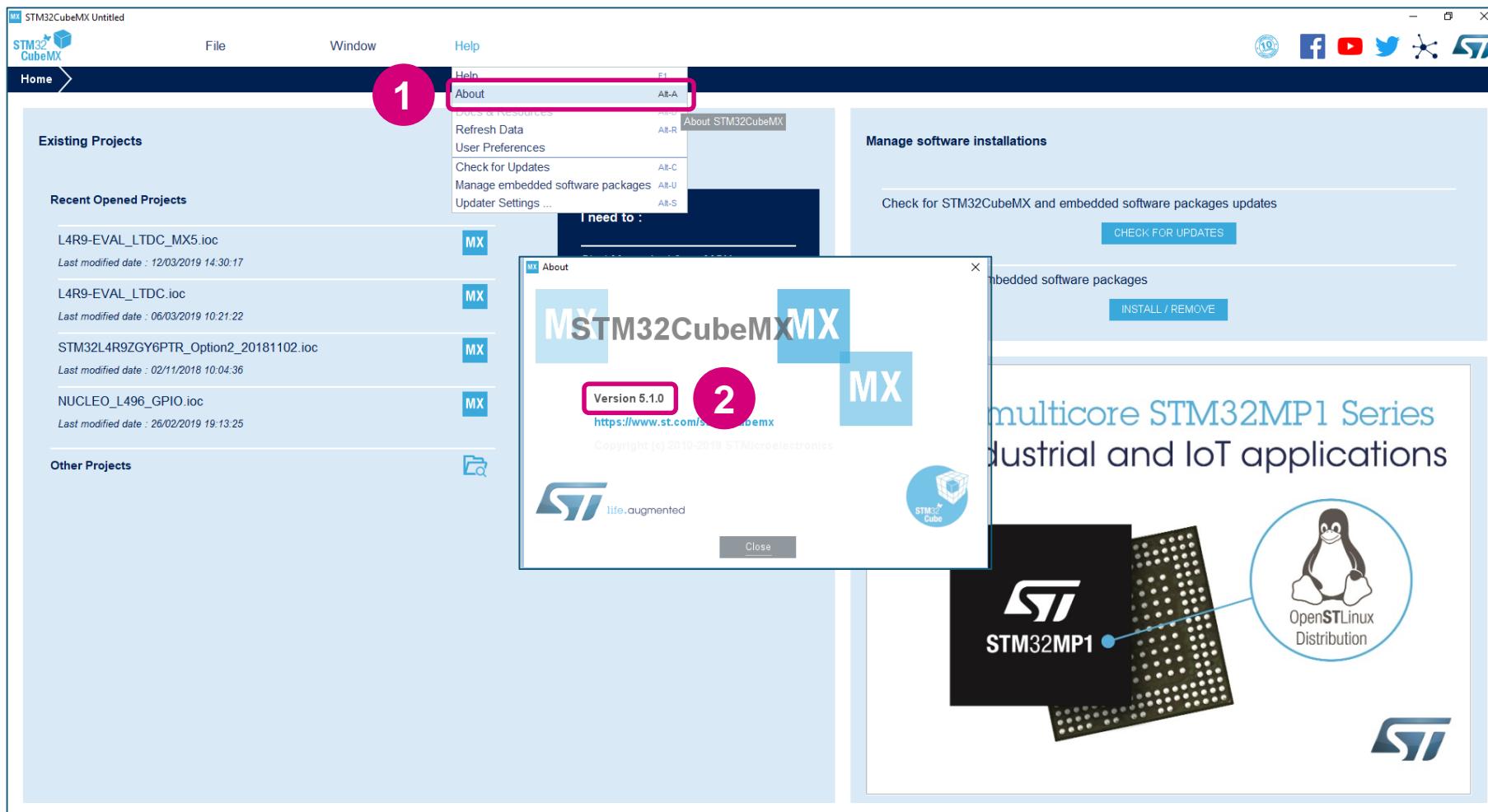
範例列表可在STM32CubeF7 FW package當中的STM32CubeProjectsList.htm檔案找到



STM32CUBEMX安裝

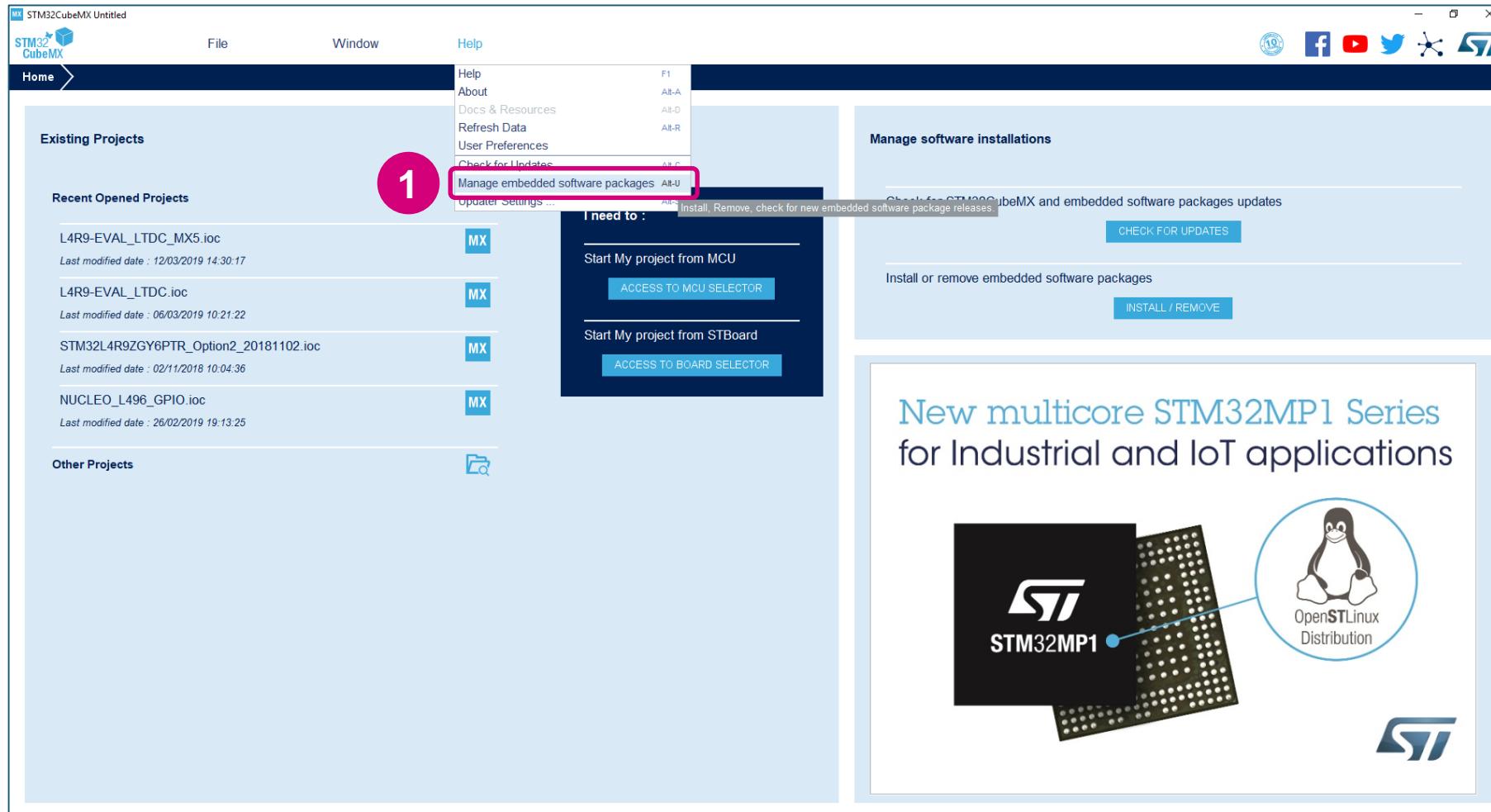
版本確認

37



安裝STM32F7 MCU Package (1)

38



安裝STM32F7 MCU Package (2)

39

MX Embedded Software Packages Manager

STM32Cube MCU Packages and embedded software packs releases

Releases Information was last refreshed 17974 days ago.

STM32Cube MCU Packages STMMicroelectronics

Description	Installed Version	Available Version
STM32F0		
STM32F1		
STM32F2		
STM32F3		
STM32F4		

MX Select a STM32Cube Package File

Look In Download

en.SetupSTM32CubeMX-5.1.0-RC6.zip
en.STM32Cube_FW_F7_V1.15.0.zip

File Name
Files of Types STM32Cube Packages File (*.zip, *.pack)

From Local ... From Url ... Open Cancel

MX Embedded Software Packages Manager

STM32Cube MCU Packages and embedded software packs releases

Releases Information was last refreshed 17974 days ago.

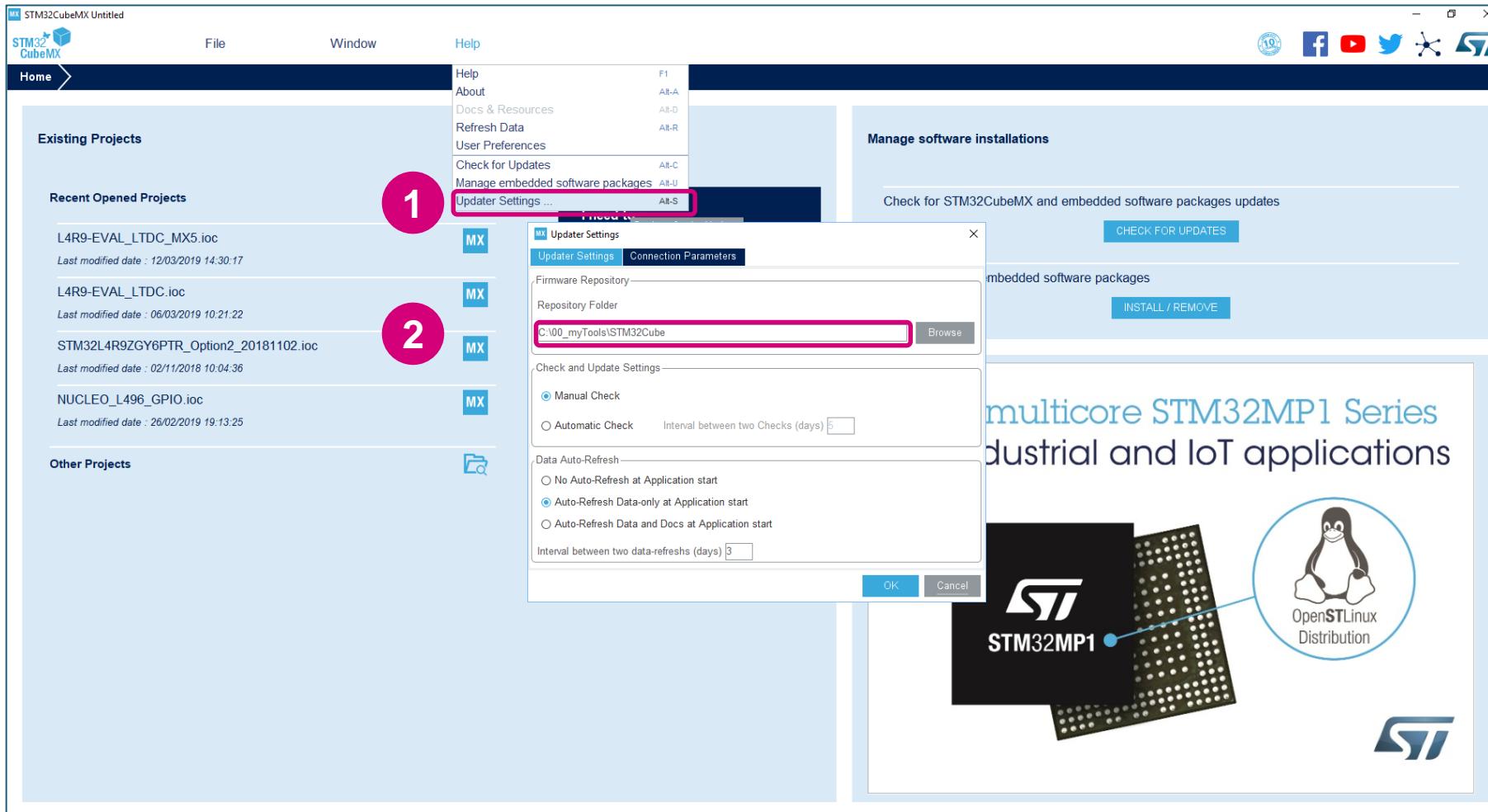
STM32Cube MCU Packages STMMicroelectronics

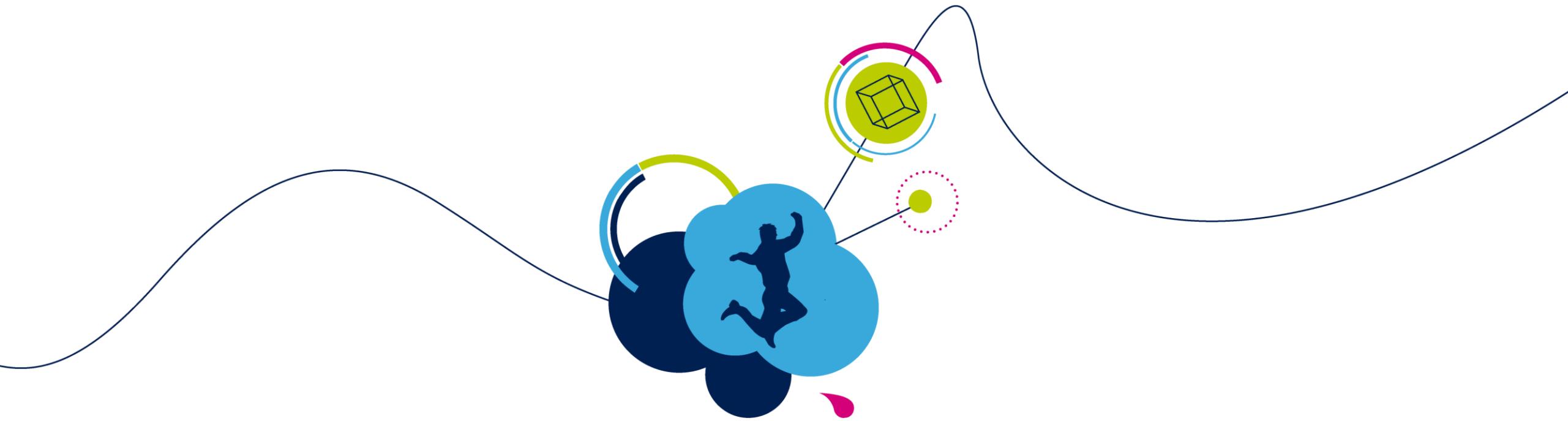
Description	Installed Version	Available Version
STM32F3		
STM32F4		
STM32F7		
STM32Cube MCU Package for STM32F7 Series	1.15.0	1.15.0
STM32Cube MCU Package for STM32F7 Series	1.14.0	1.14.0

From Local ... From Url ... Refresh Install Now Remove Now Close

MCU Package Repository 路徑

40





TrueSTUDIO 安裝

STM32 開發生態系統

42

STM32 32-bit ARM Cortex MCUs

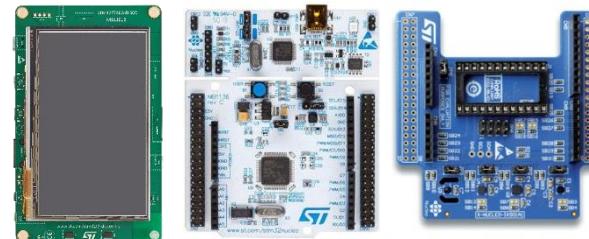


STM32 Ecosystem

- Software tools
 - STM32CubeMX Configuration and initialization tool
 - Integrated Development Environments (IDE)
 - STM STUDIO Monitoring tool
 - More software tools
- Embedded software
 - STM32Cube MCU Packages
 - STM32Cube Expansion Packages
 - More embedded software
- Hardware tools
 - STM32 Nucleo development boards, Discovery kits, Evaluation boards
 - STM32 Nucleo expansion boards
 - ST-LINK in-circuit debugger/programmer



arm KEIL



STM32 開發工具

43

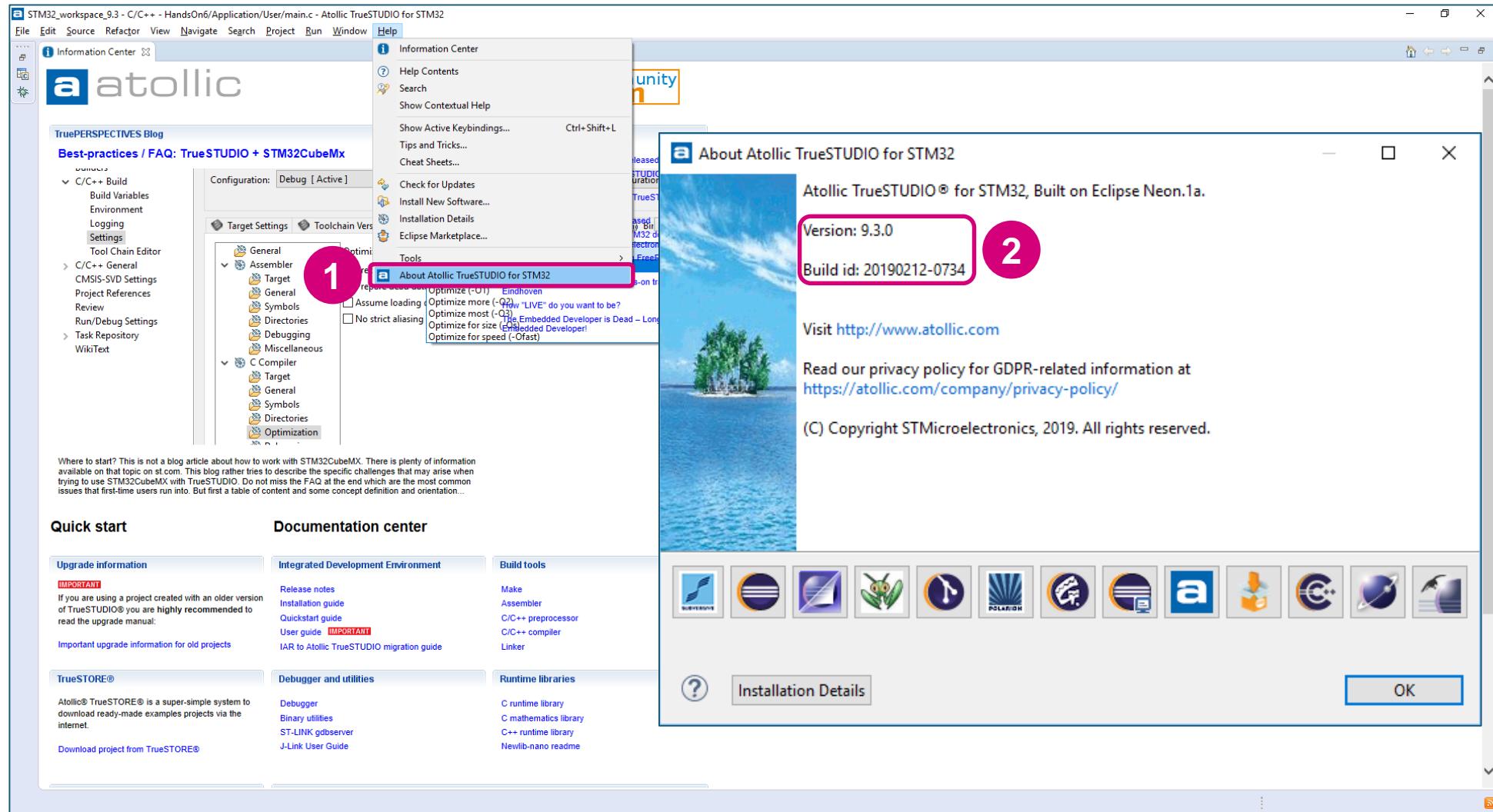


STM32 Software Development Tools



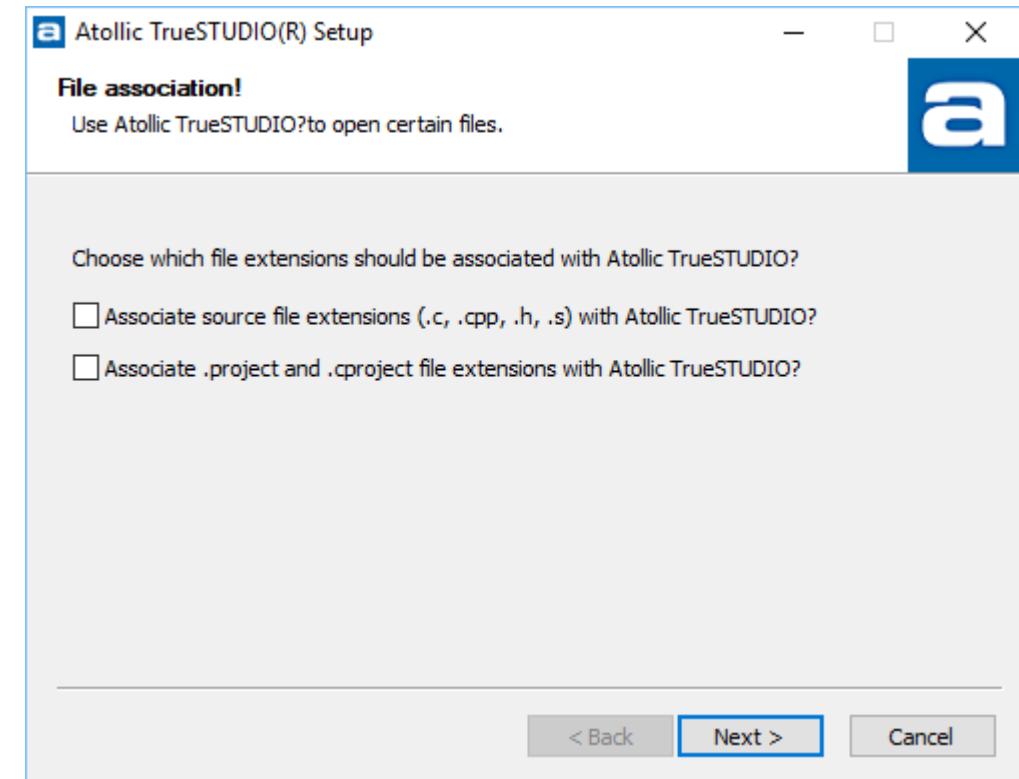
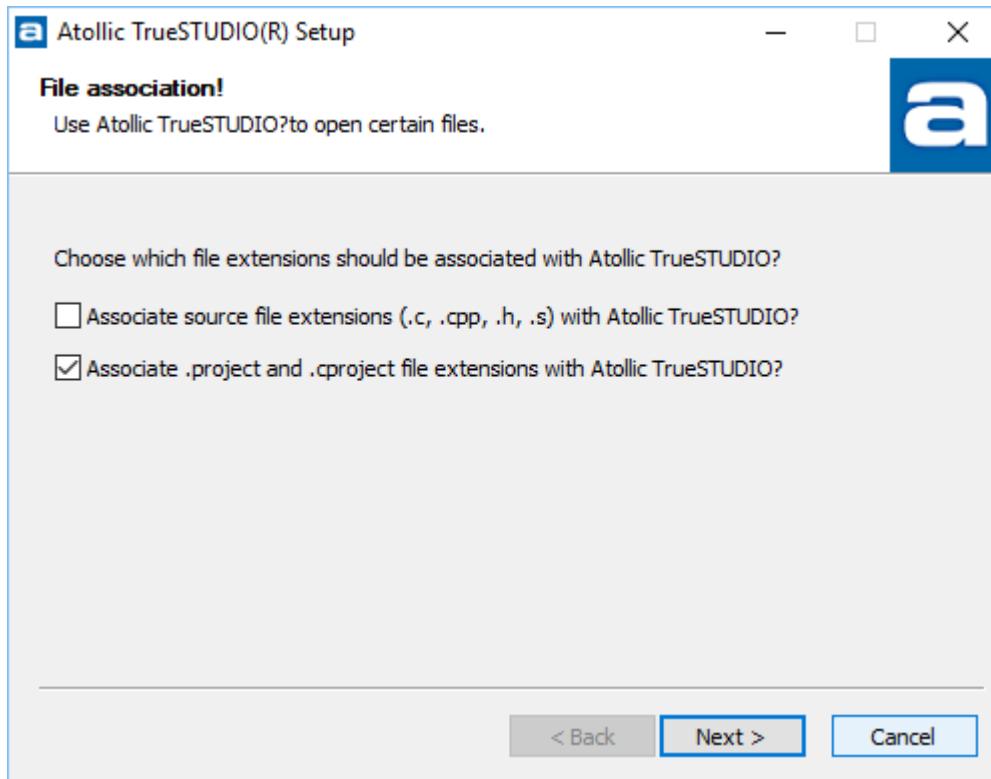
版本確認

44



設定TrueSTUDIO專案檔關聯

45

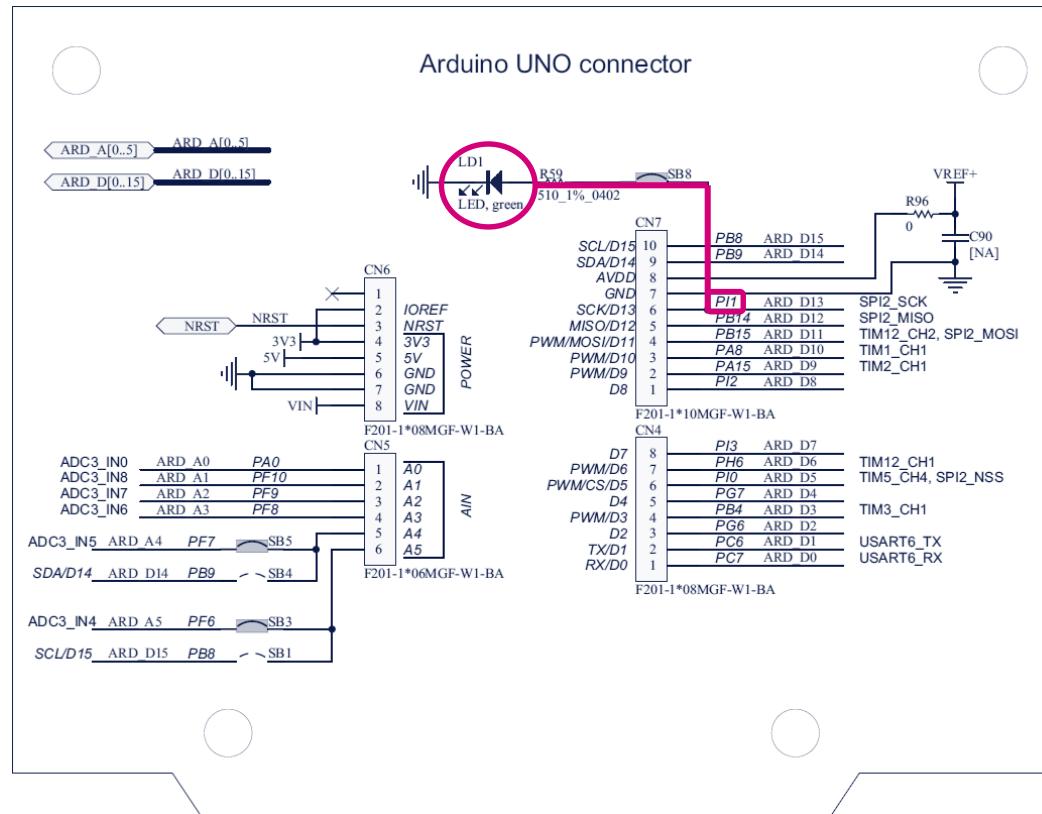


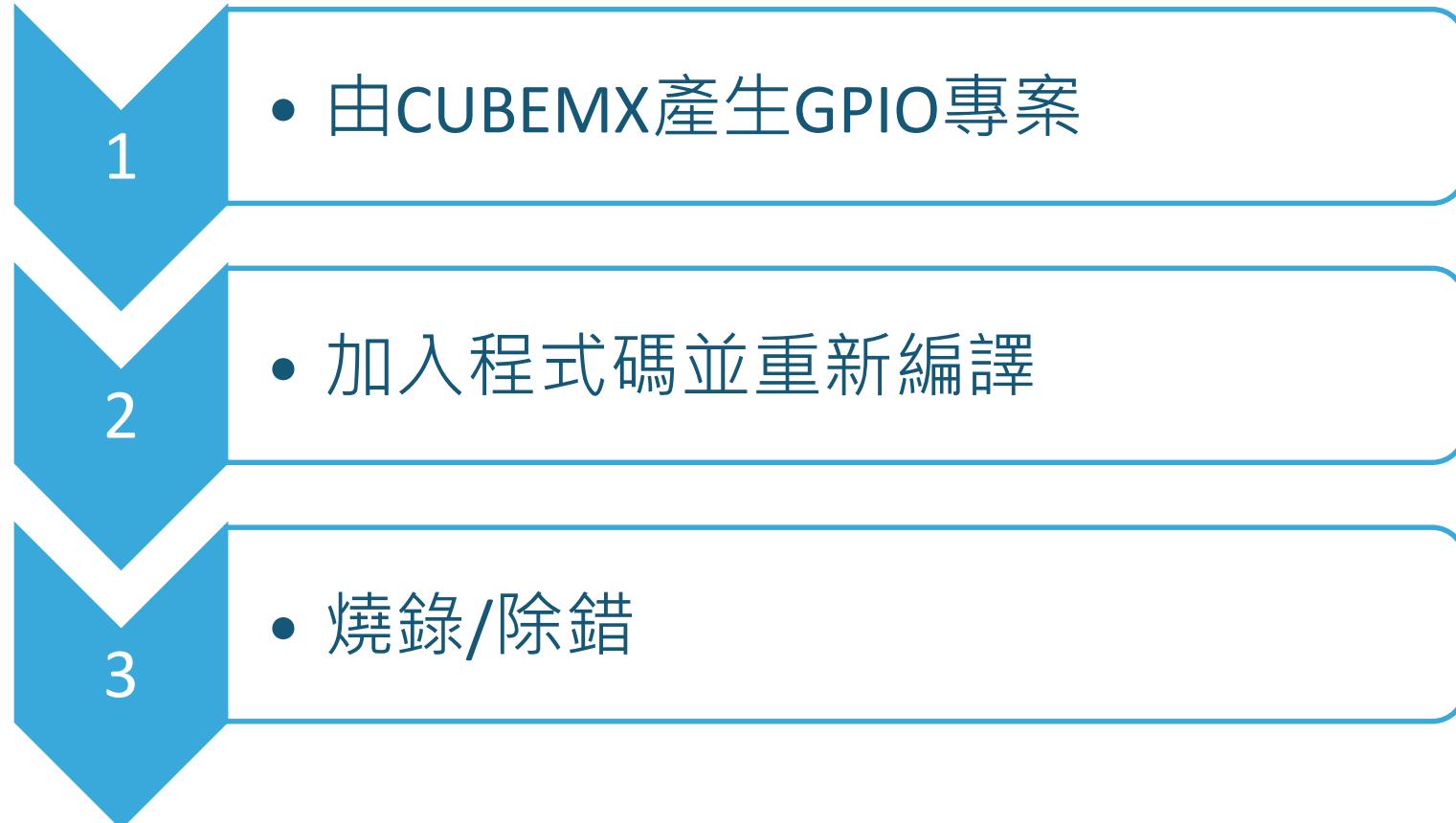


Hands-on 1

GPIO Control: Build from STM32CubeMX

每500ms改變LD1閃滅狀態

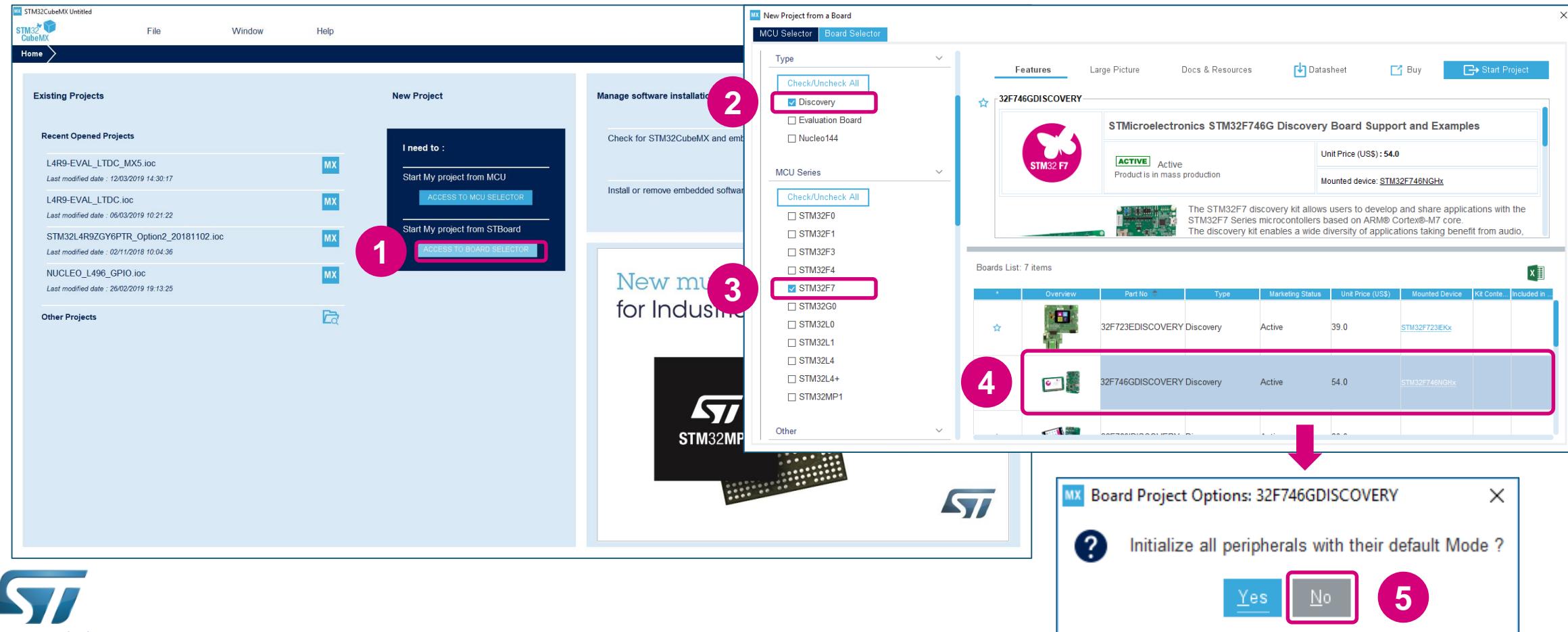




Step 1: 由CUBEMX產生GPIO專案

49

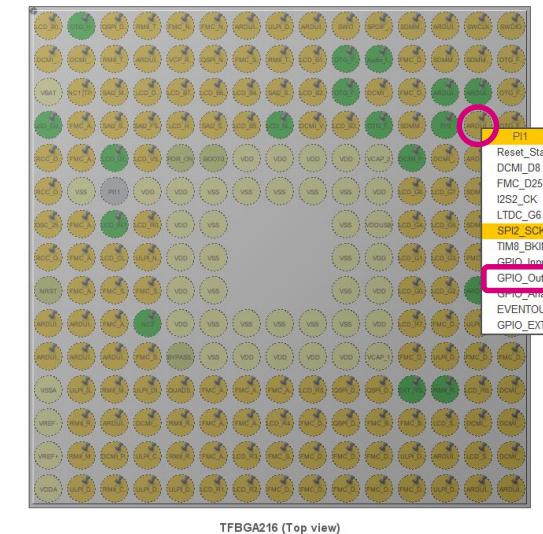
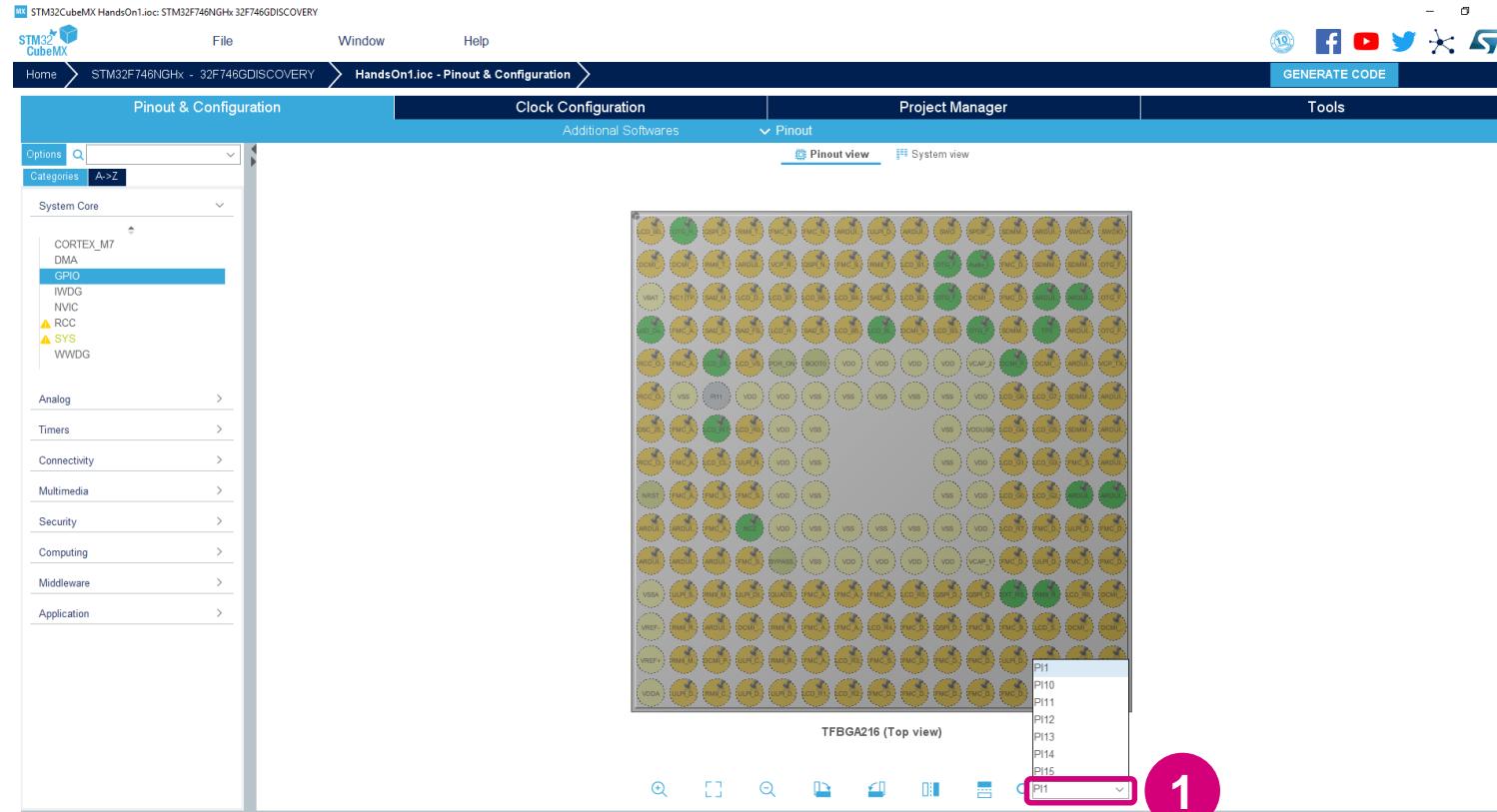
由Board Selector開始選取



Step 1: 由CUBEMX產生GPIO專案

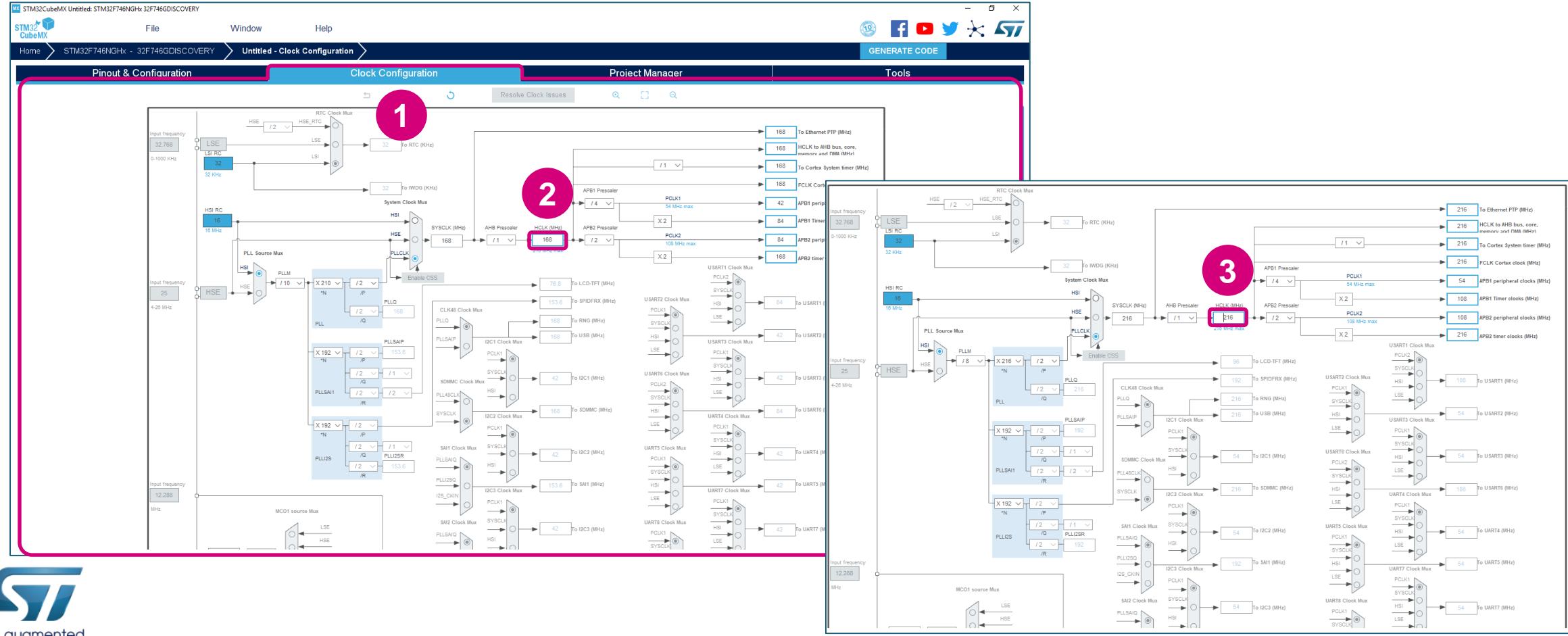
50

檢視Pinout圖並新增LD1控制腳位PI1



Step 1: 由CUBEMX產生GPIO專案

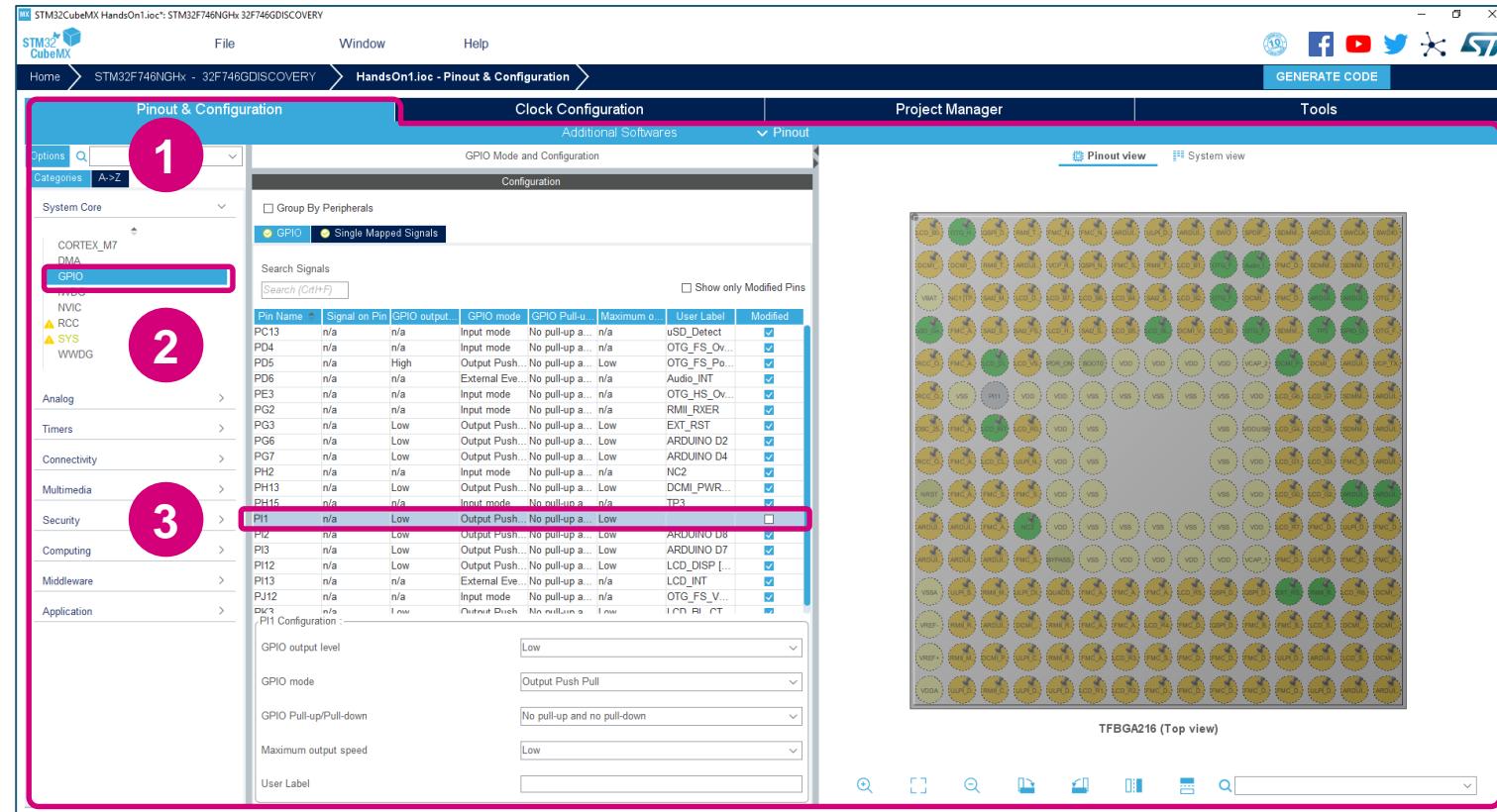
檢視/設定Clock Tree



Step 1: 由CUBEMX產生GPIO專案

52

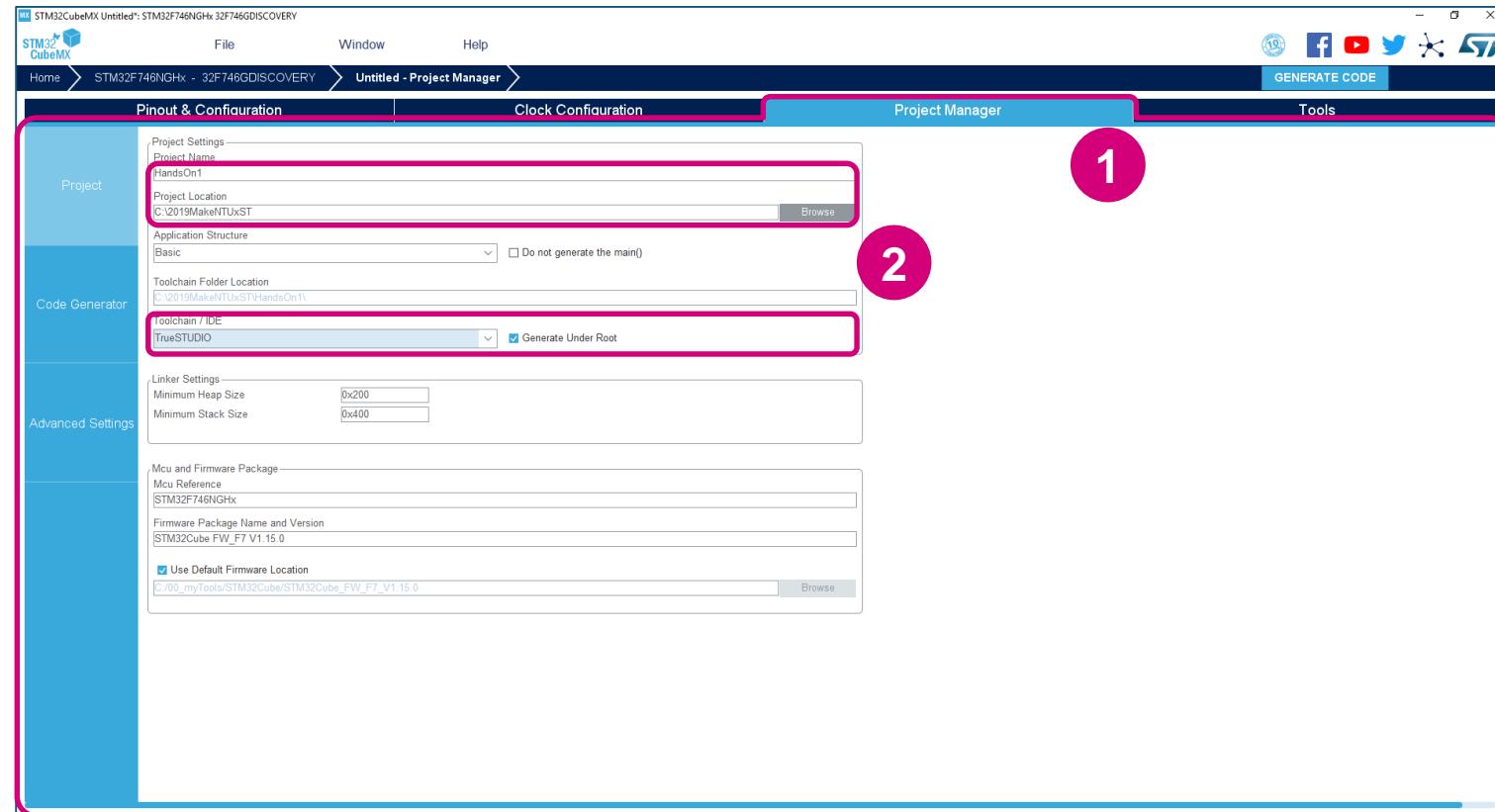
檢視/設定Peripheral Configuration



Step 1: 由CUBEMX產生GPIO專案

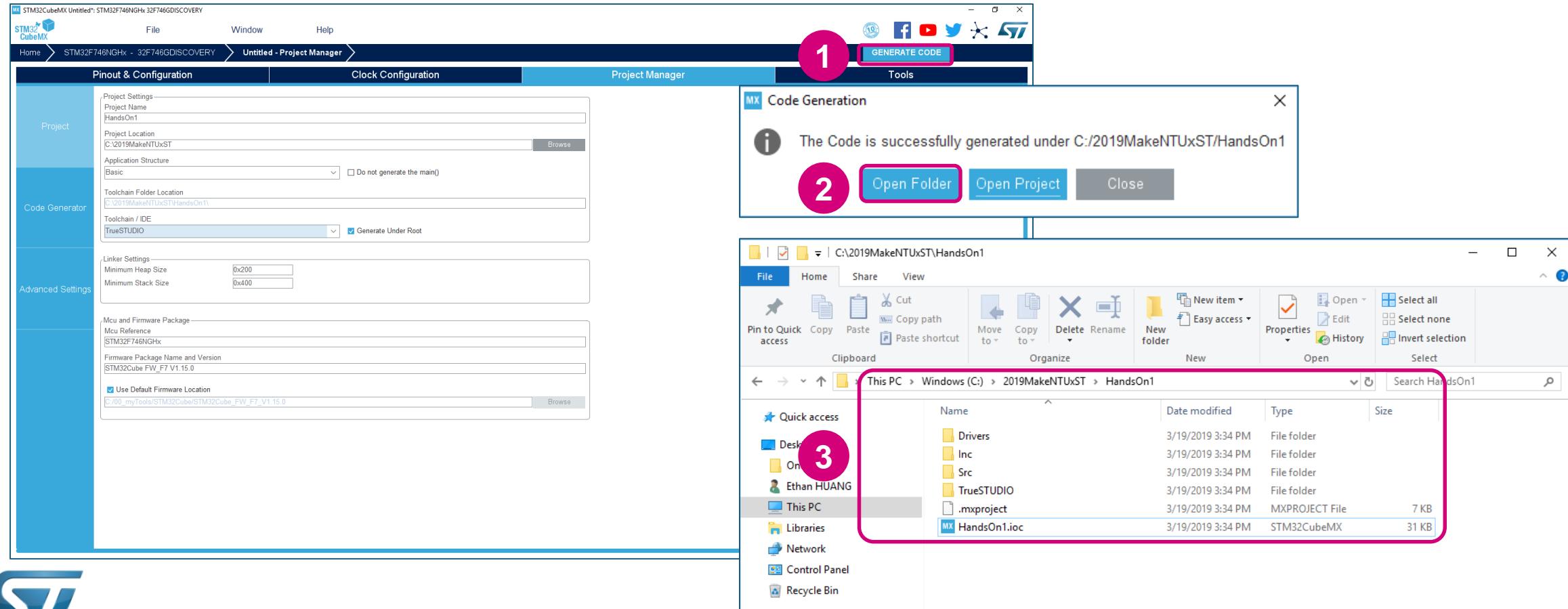
53

檢視/設定Project Settings



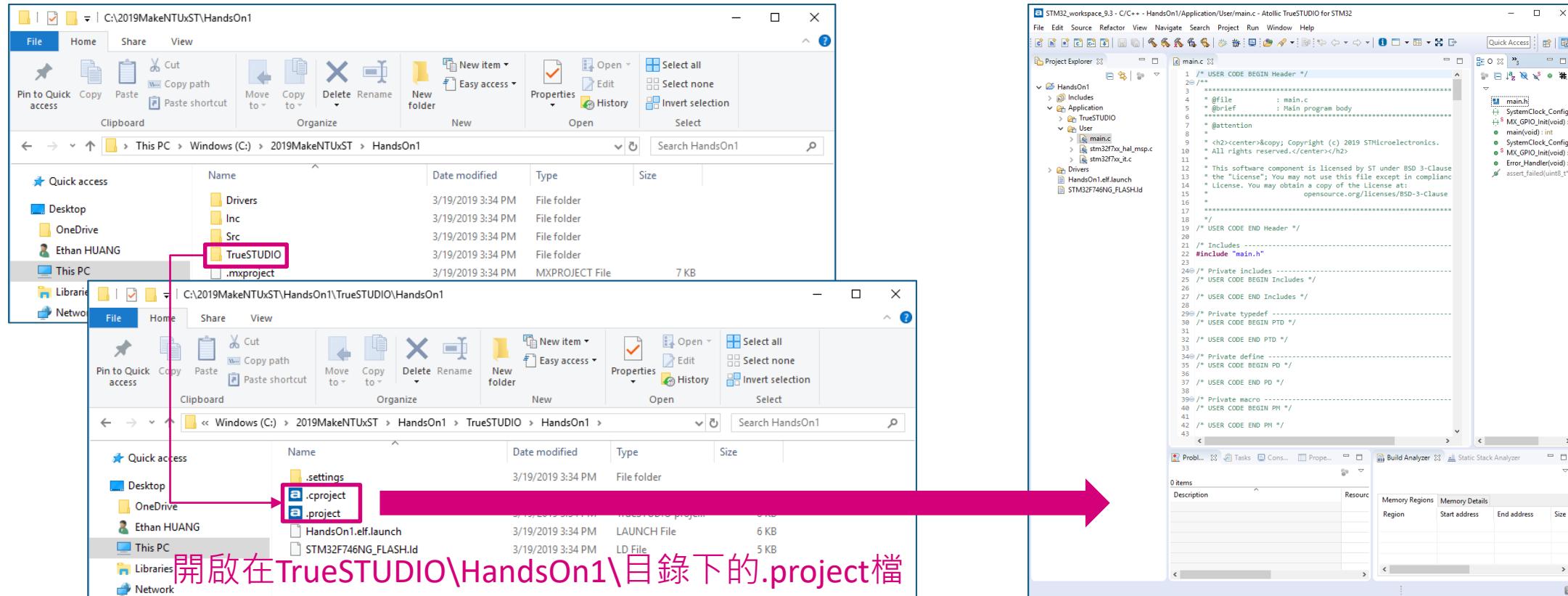
Step 1: 由CUBEMX產生GPIO專案

產生TrueSTUDIO專案檔和程式碼



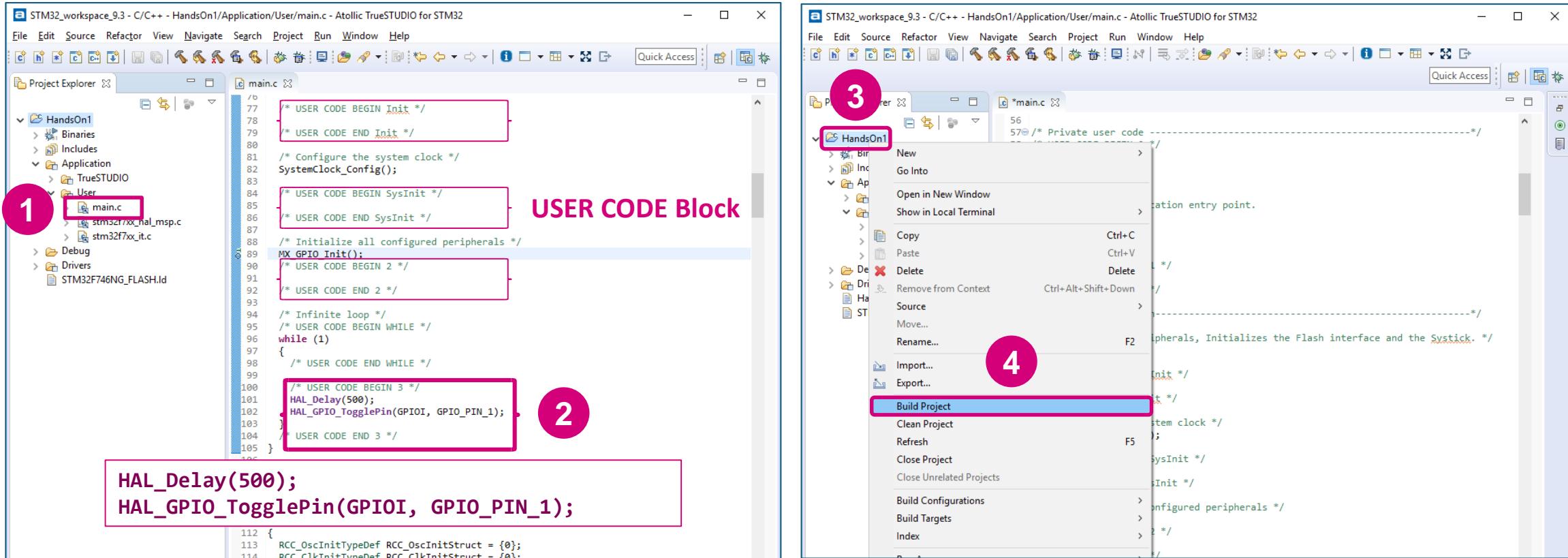
Step 2:加入程式碼並重新編譯

開啟TrueSTUDIO專案檔



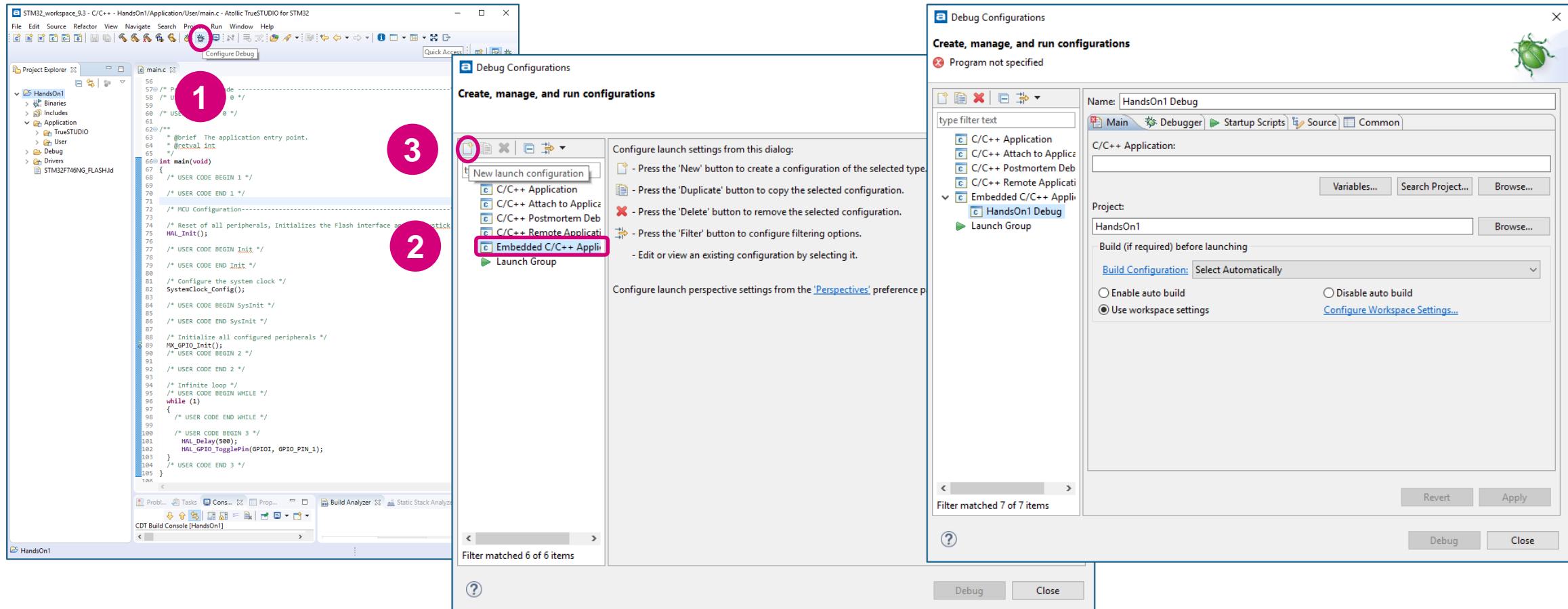
Step 2:加入程式碼並重新編譯

於main.c加入程式碼並重新編譯



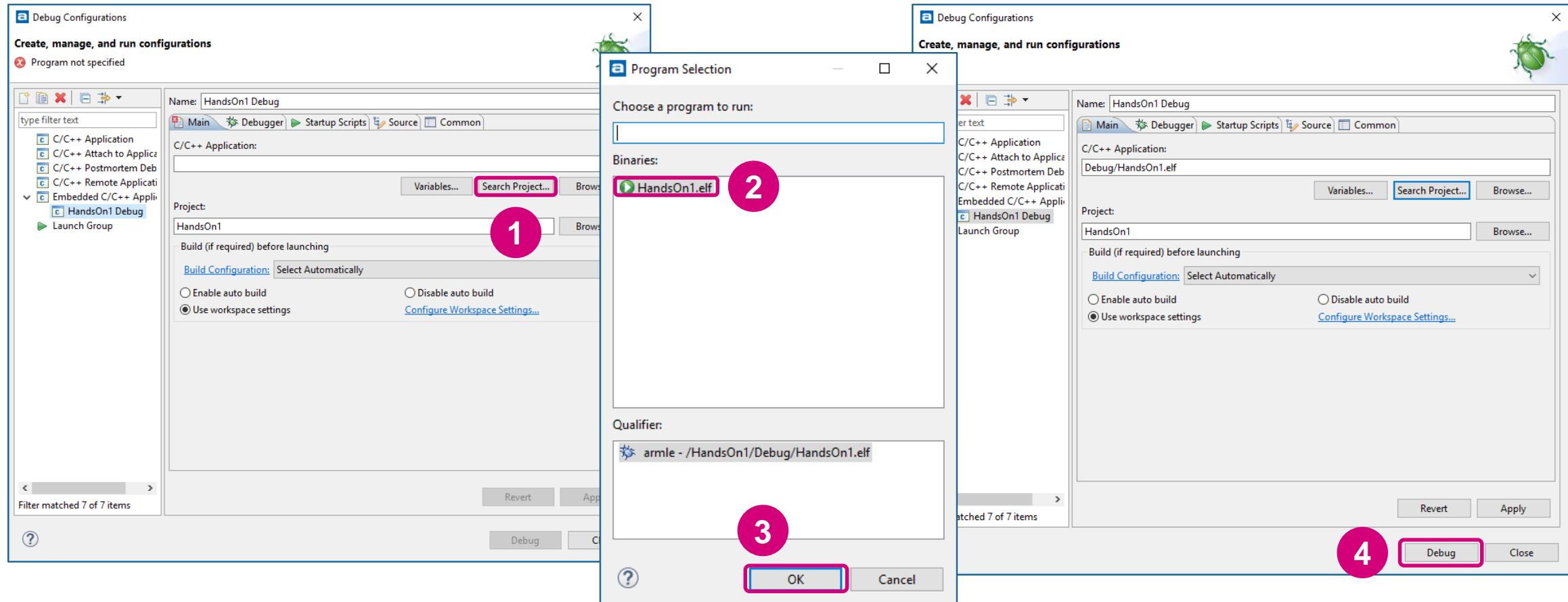
Step 3: 燒錄/除錯

57



Step 3: 燒錄/除錯

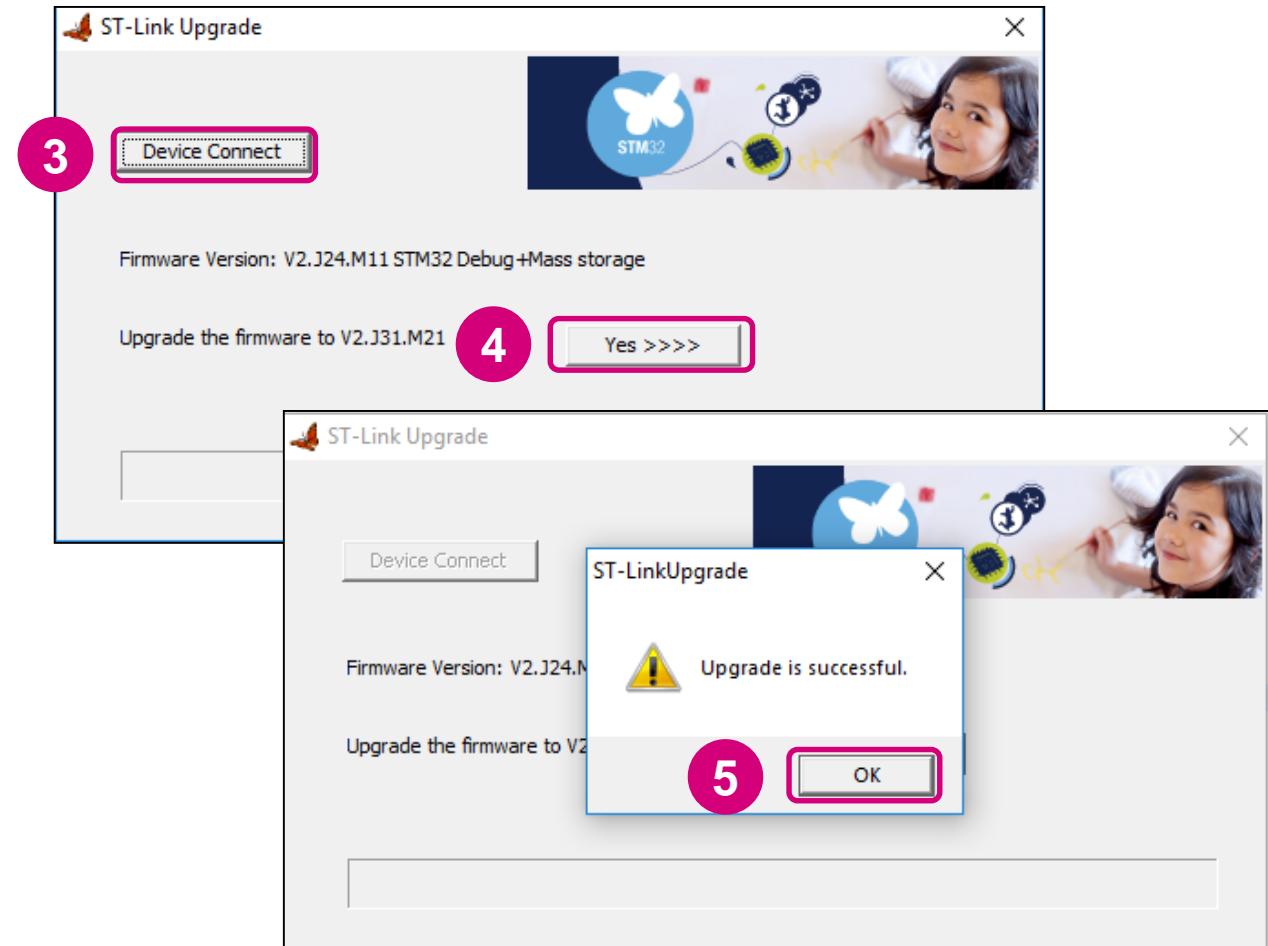
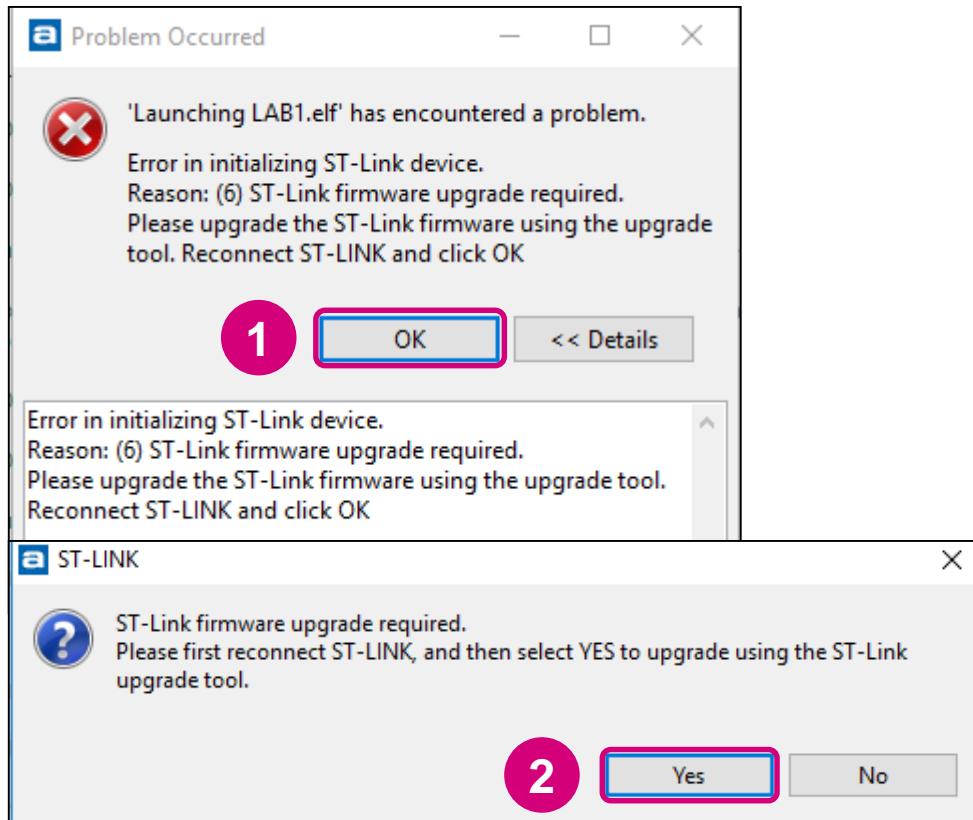
58



Step 3: 燒錄/除錯

59

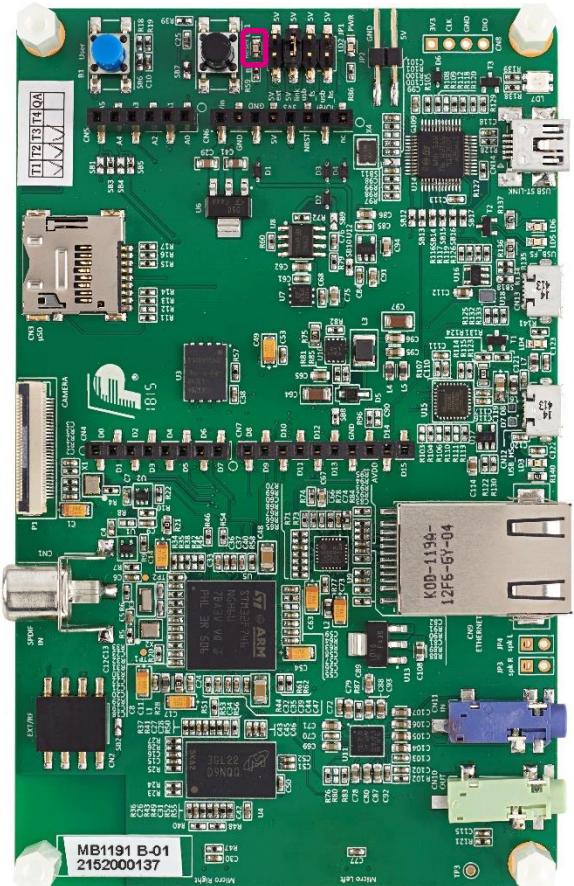
可能會發生的錯誤訊息



Step 3: 燒錄/除錯

60

每500ms改變LD1閃滅狀態



- **永遠使用STM32CUBEMX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STM32CUBEMX 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**

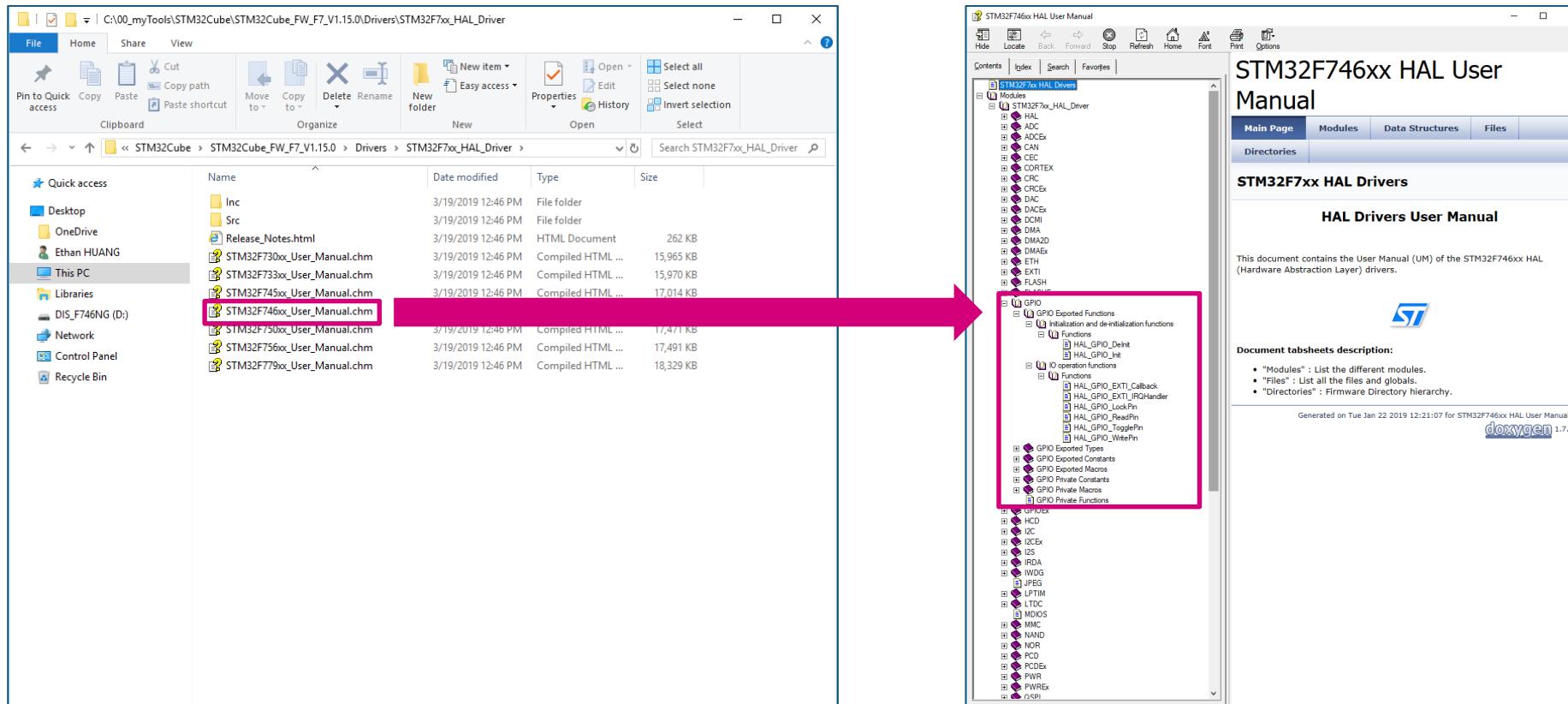
若將自己的程式碼置於USER CODE區塊之外，則自己的程式碼會在用STM32CUBEMX重新產生程式碼之後被移除

補充：如何查找HAL API

62

搜尋Doxygen文件 (以搜尋GPIO HAL API為例)

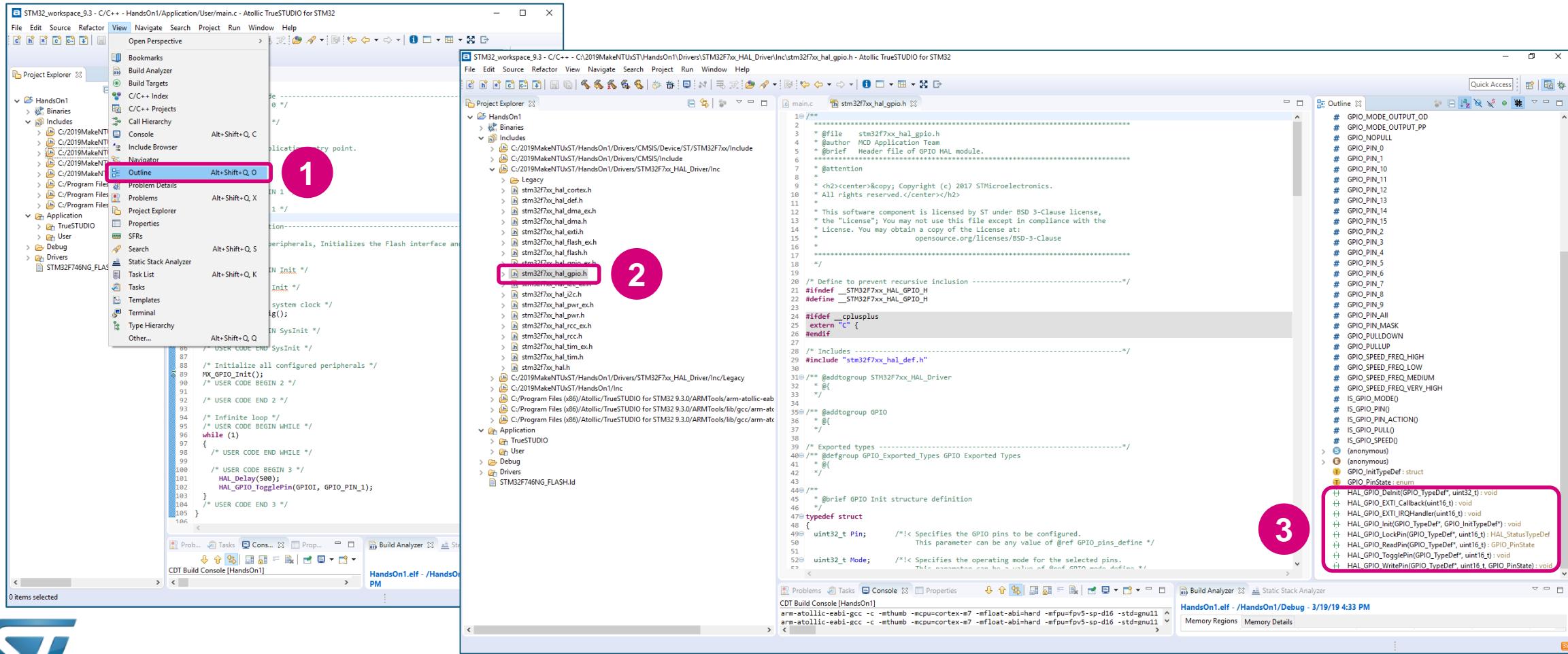
Doxygen文件置於目錄: STM32F7 MCU Package Repository\Drivers\STM32F7xx_HAL_Driver



補充：如何查找HAL API

63

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

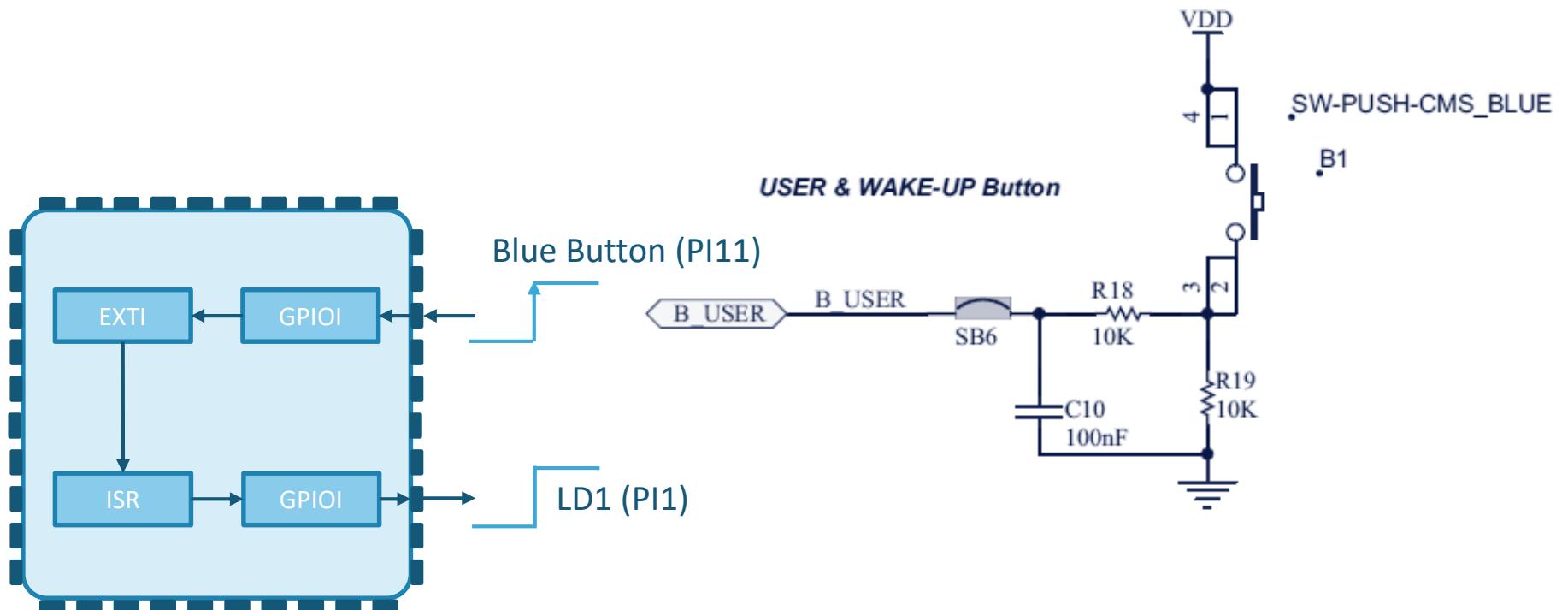




Hands-on 2

Button Interrupt

每500ms改變LD1閃滅狀態

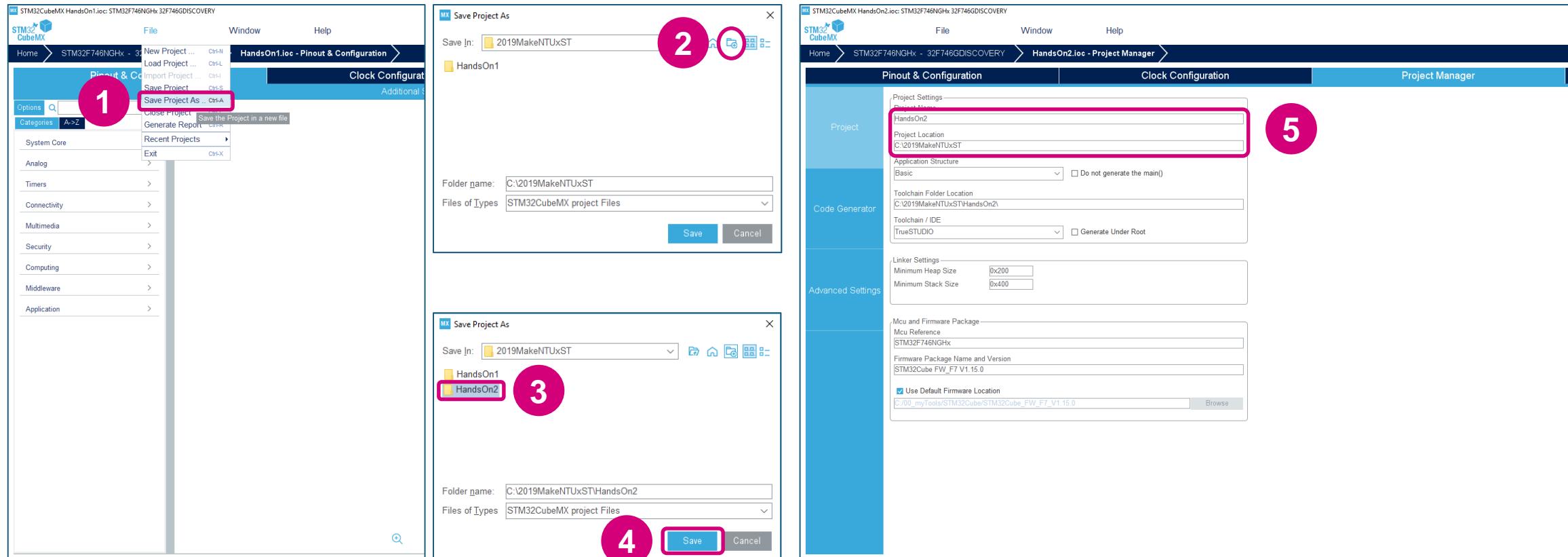




Step 1: 另存CUBEMX專案

67

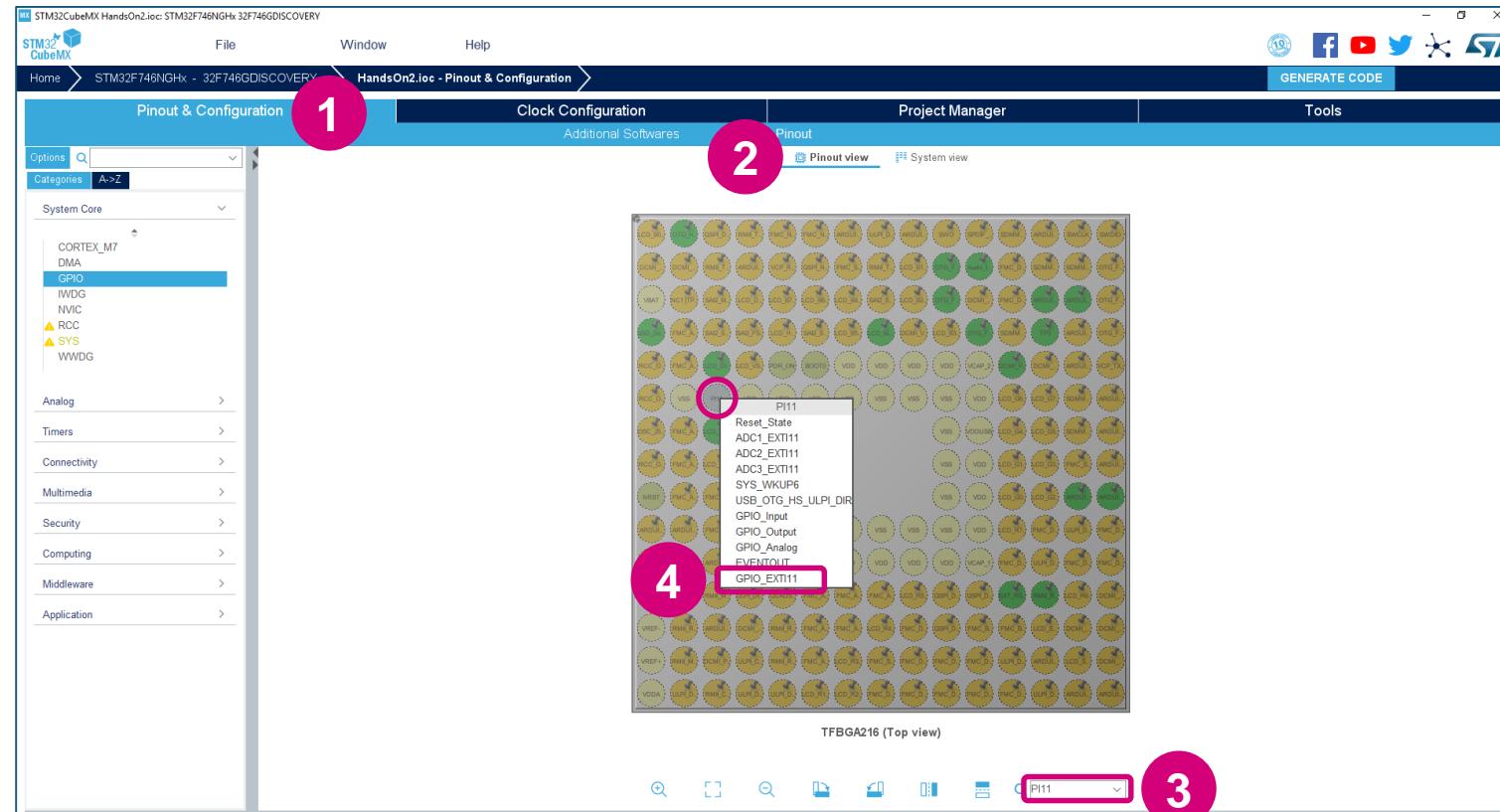
由HandsOn 1另存CUBEMX Project



Step 2: 新增Button設定

68

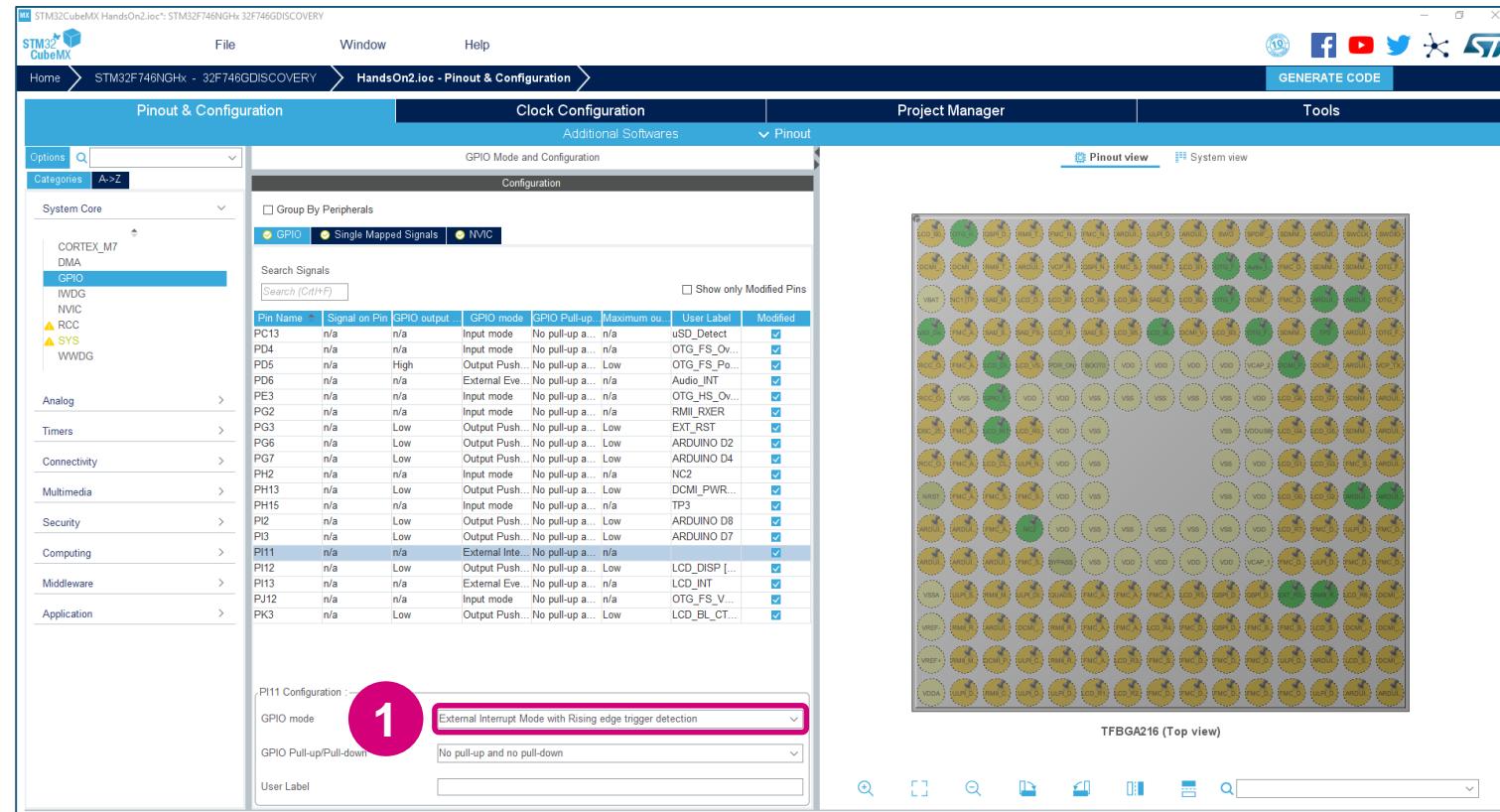
搜尋PI11並設為GPIO_EXTI11



Step 2: 新增Button設定

69

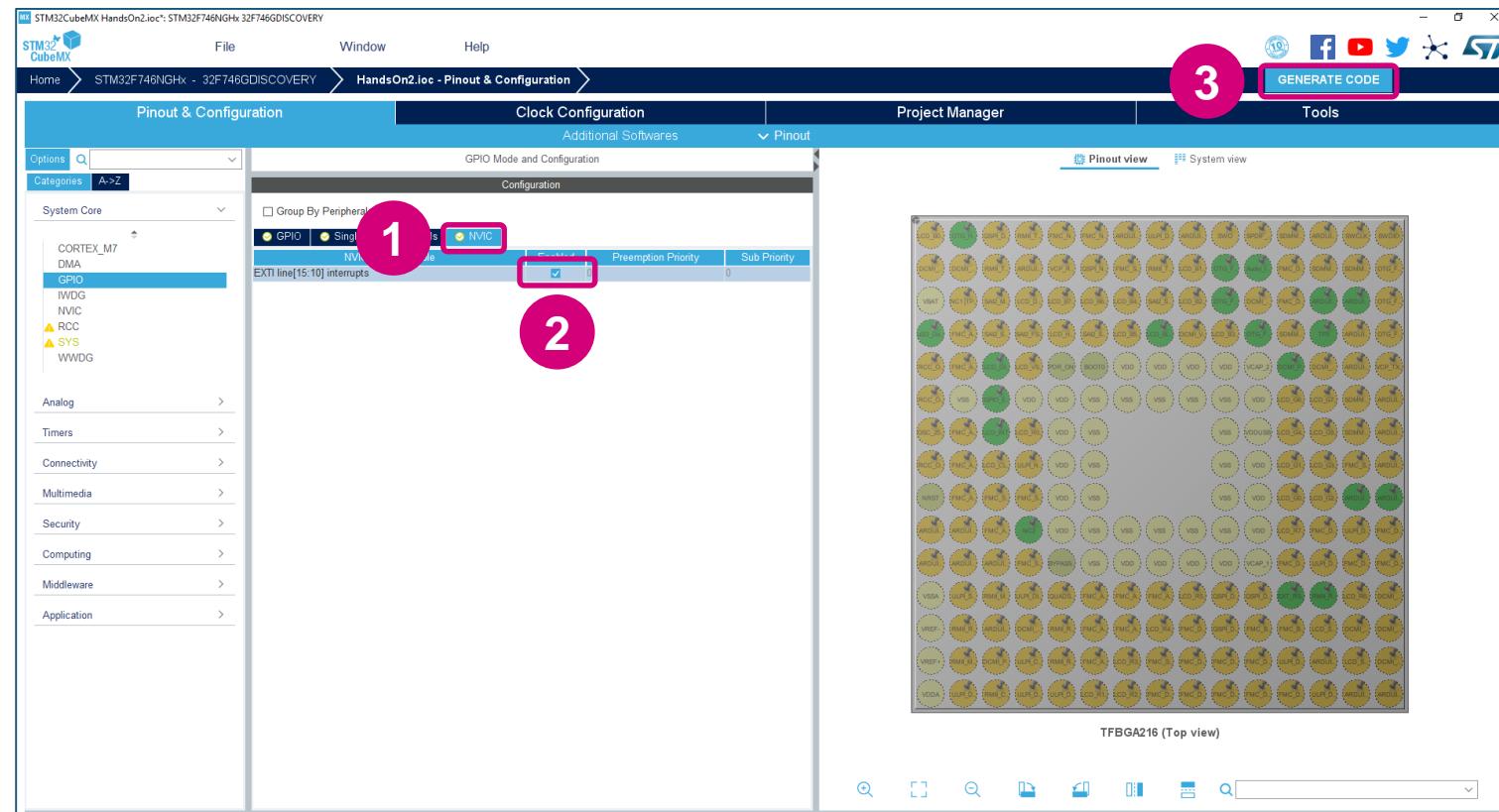
檢視PI11的中斷設定



Step 2: 新增Button設定

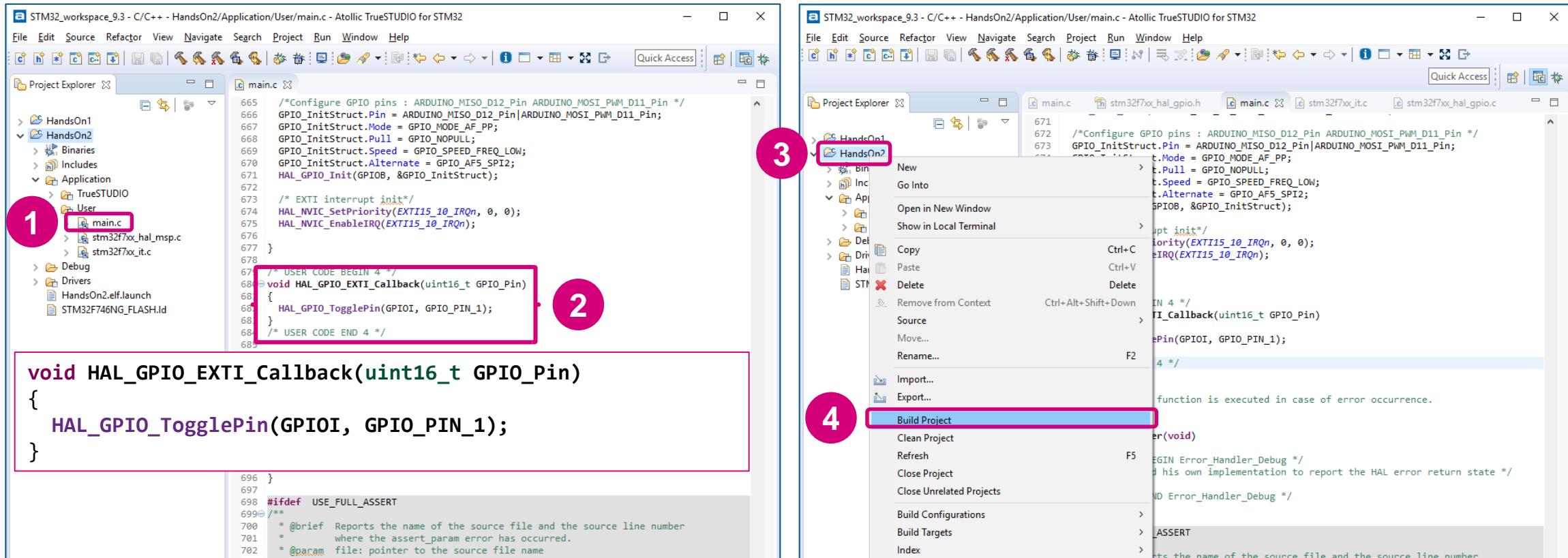
70

檢視PI11的NVIC設定



Step 3:加入程式碼並重新編譯

於main.c加入程式碼並重新編譯



The image shows two screenshots of the Atollic TrueSTUDIO IDE interface. The left screenshot shows the main.c file with a block of code highlighted and numbered 2. The right screenshot shows a context menu for a project named 'HandsOn2' with 'Build Project' highlighted and numbered 4. Numbered 1 points to the main.c file in the project explorer. Numbered 3 points to the 'HandsOn2' project in the project explorer.

```

1  main.c
2
3  HandsOn2
4  Build Project

```

main.c (Left Screenshot):

```

665  /*Configure GPIO pins : ARDUINO_MISO_D12_Pin ARDUINO_MOSI_PWM_D11_Pin */
666  GPIO_InitStruct.Pin = ARDUINO_MISO_D12_Pin|ARDUINO_MOSI_PWM_D11_Pin;
667  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
668  GPIO_InitStruct.Pull = GPIO_NOPULL;
669  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
670  GPIO_InitStruct.Alternate = GPIO_AF5_SPI2;
671  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
672
673  /* EXTI interrupt init*/
674  HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
675  HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
676
677 }
678
679 /* USER CODE BEGIN 4 */
680 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
681 {
682     HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
683 }
684 /* USER CODE END 4 */
685

```

main.c (Right Screenshot):

```

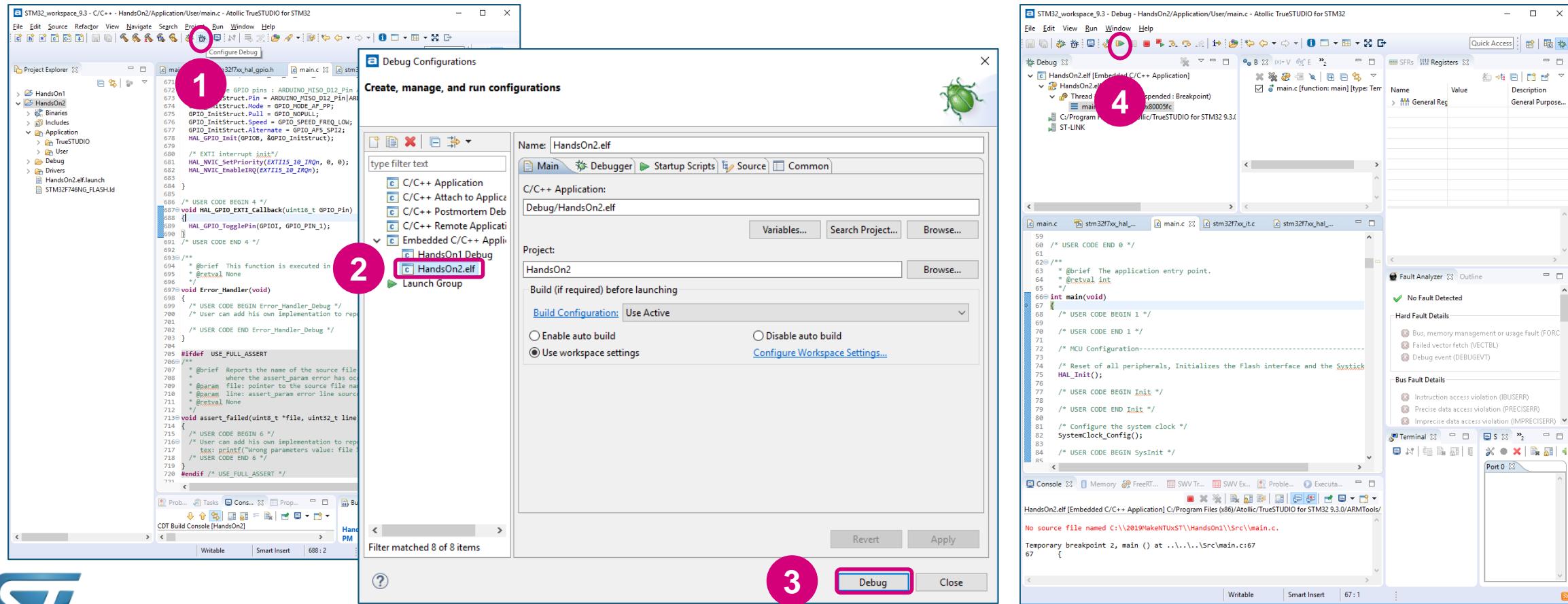
671  /*Configure GPIO pins : ARDUINO_MISO_D12_Pin ARDUINO_MOSI_PWM_D11_Pin */
672  GPIO_InitStruct.Pin = ARDUINO_MISO_D12_Pin|ARDUINO_MOSI_PWM_D11_Pin;
673  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
674  GPIO_InitStruct.Pull = GPIO_NOPULL;
675  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
676  GPIO_InitStruct.Alternate = GPIO_AF5_SPI2;
677  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
678
679  /* EXTI interrupt init*/
680  HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
681  HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
682
683 /* USER CODE BEGIN 4 */
684 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
685 {
686     HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
687 }
688
689 #ifdef USE_FULL_ASSERT
690 /**
691  * @brief  Reports the name of the source file and the source line number
692  * where the assert_param error has occurred.
693  * @param  file: pointer to the source file name
694  */
695

```

Step 4: 燒錄/除錯

72

Debug Configurations → Debug → Resume



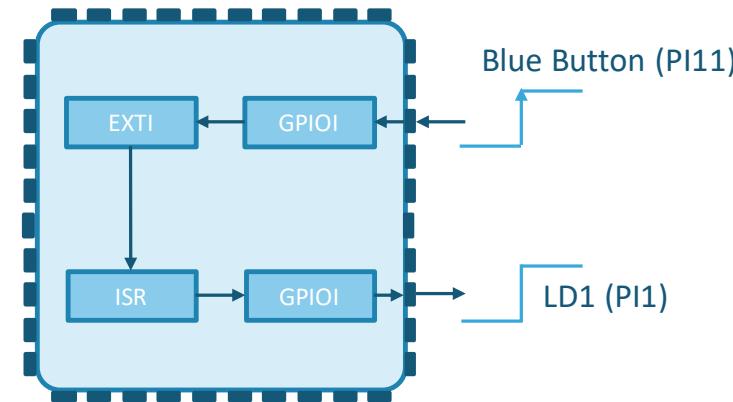
Step 4: 燒錄/除錯

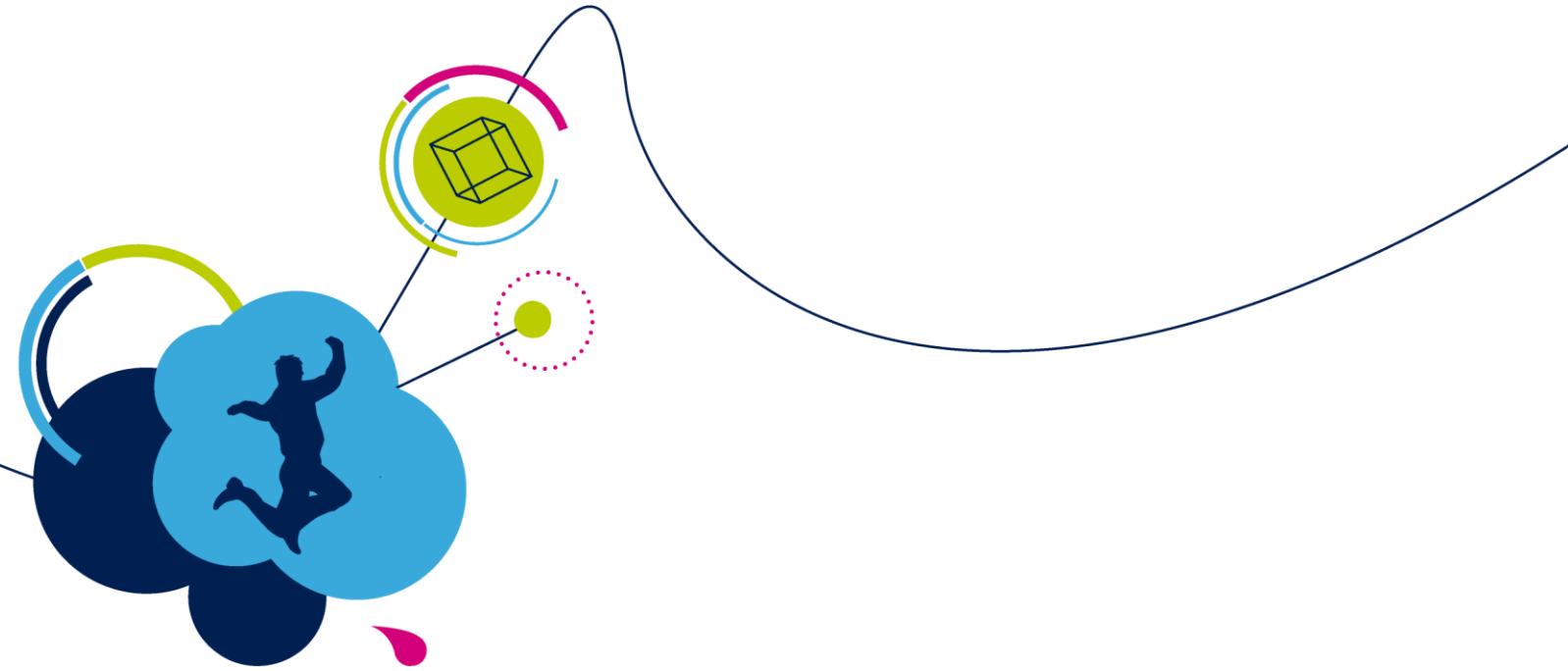
73

中斷處理說明



- Blue Button = PI11
- LD1 = PI1
- Blue button pressed
 - PI11 goes from LOW to HIGH
 - 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





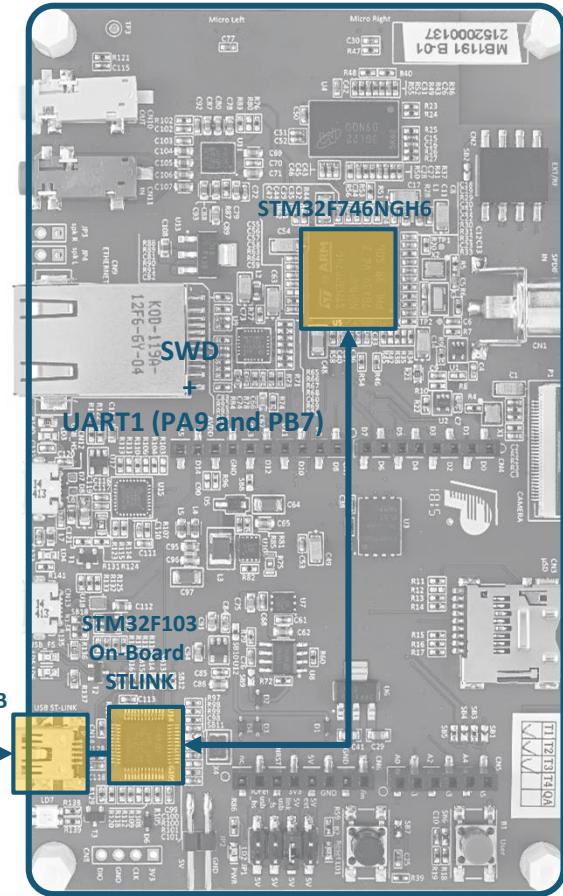
Hands-on 3

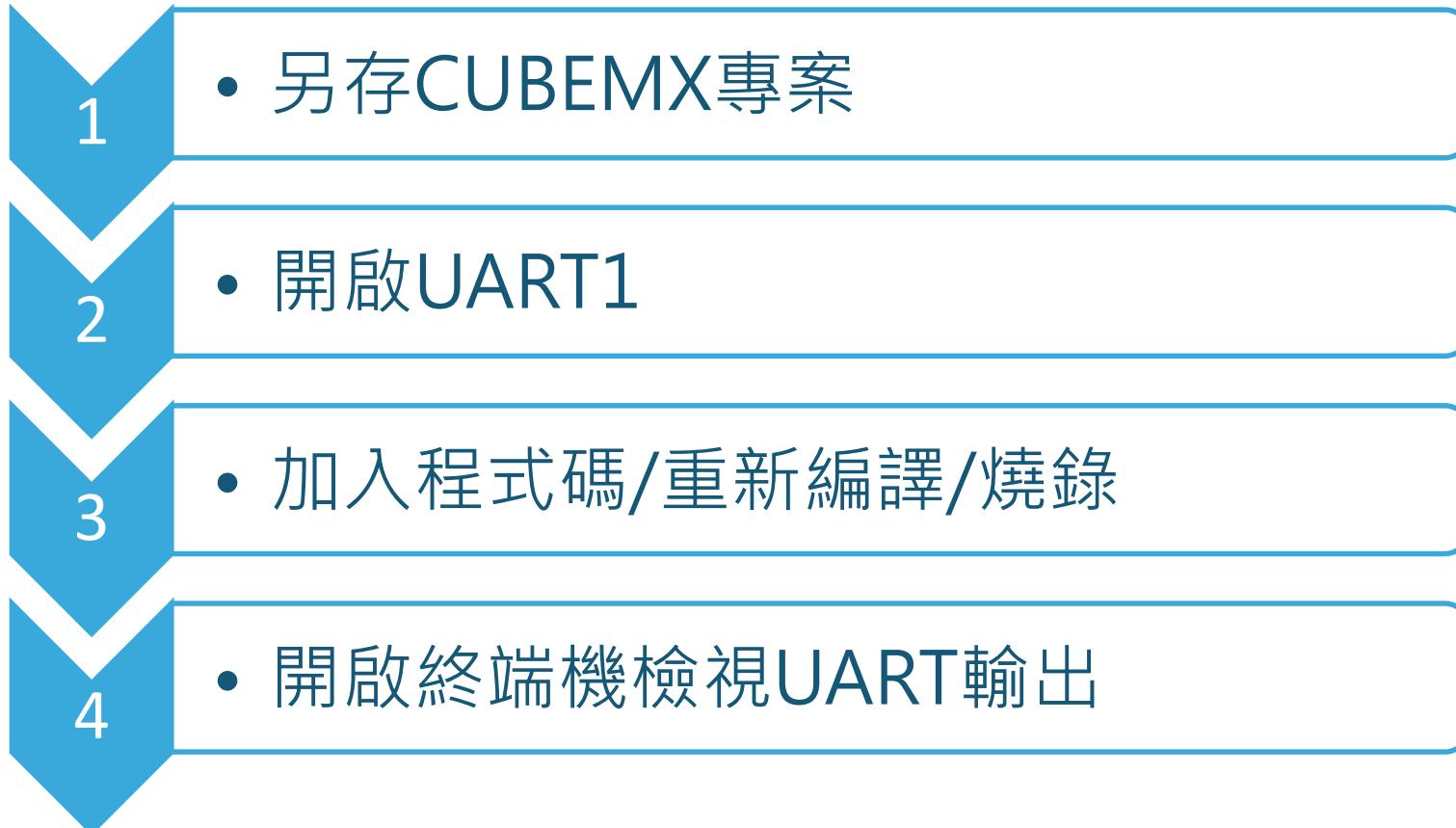
Printf over UART1-VCP

每500ms改變LD1閃滅狀態，
並透過終端機顯示LD1狀態



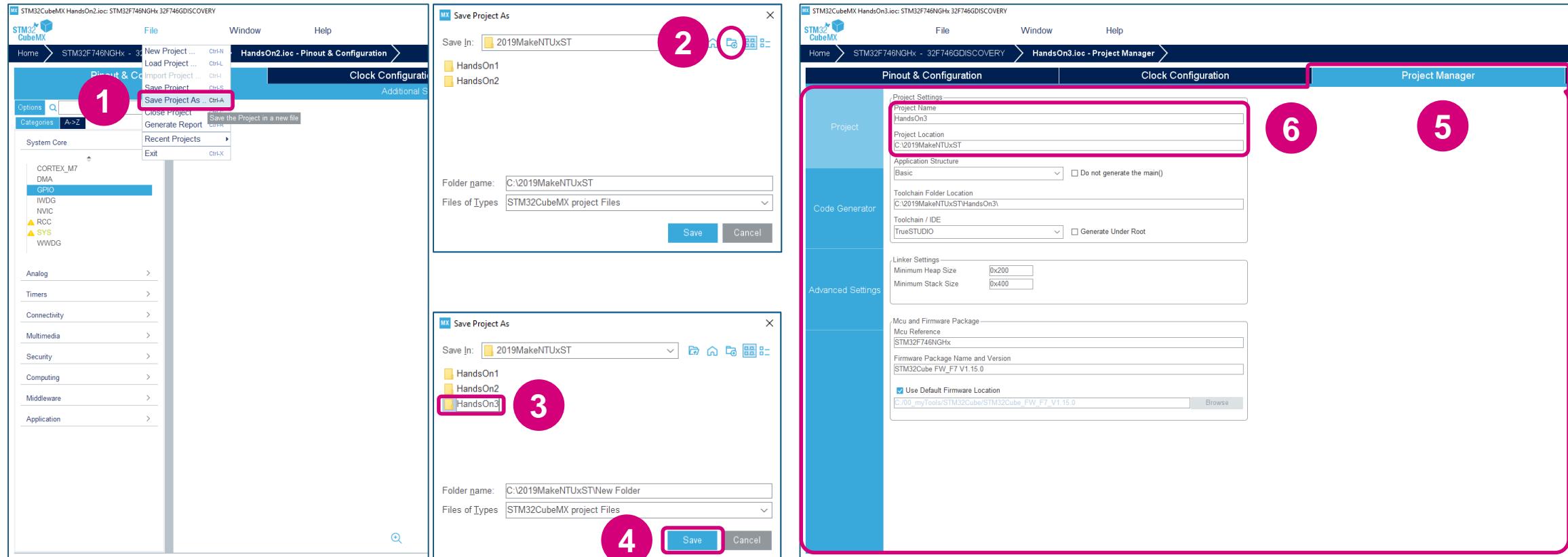
USB





Step 1: 另存CUBEMX專案

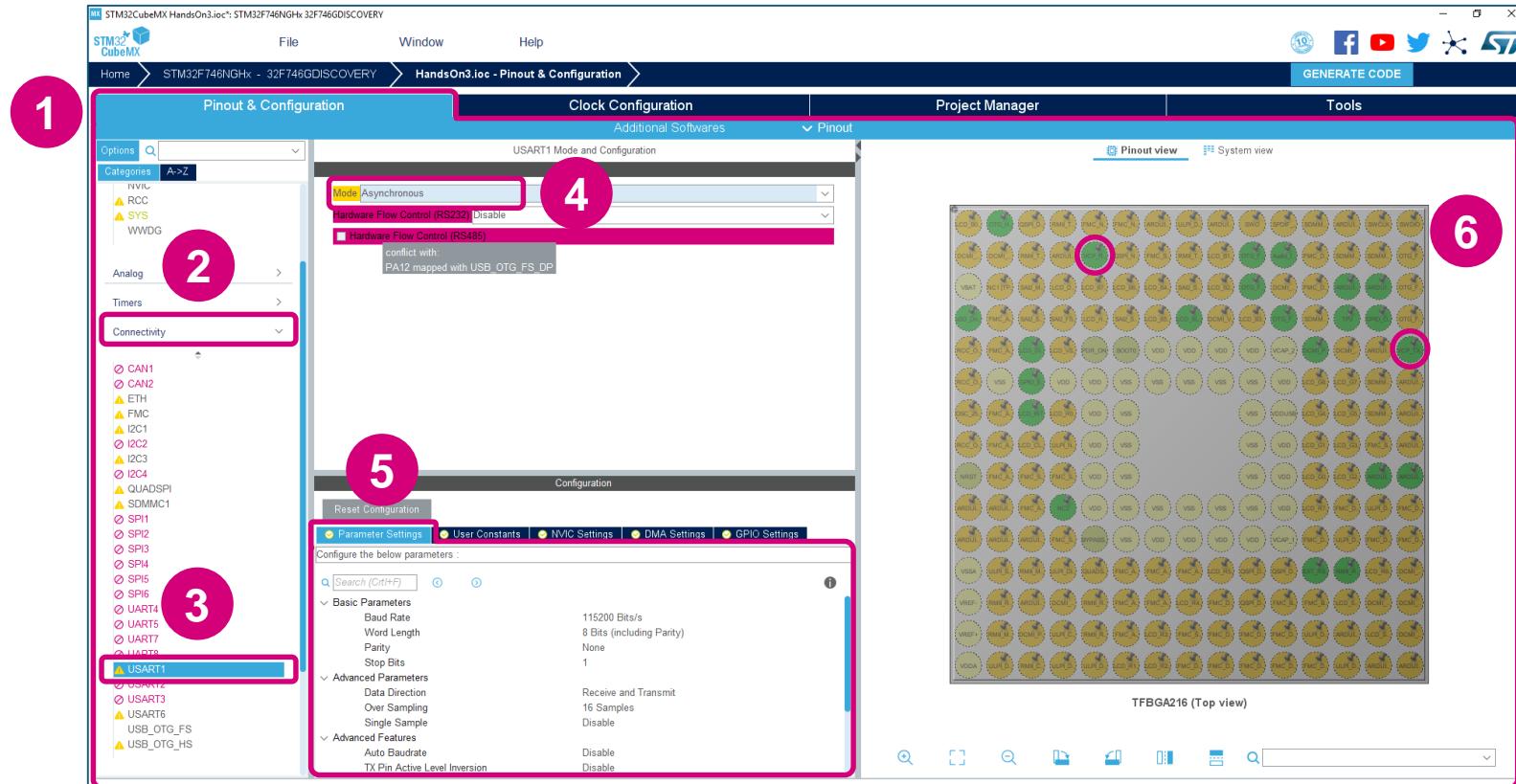
由HandsOn 2另存CUBEMX Project



Step 2: 開啟UART1

78

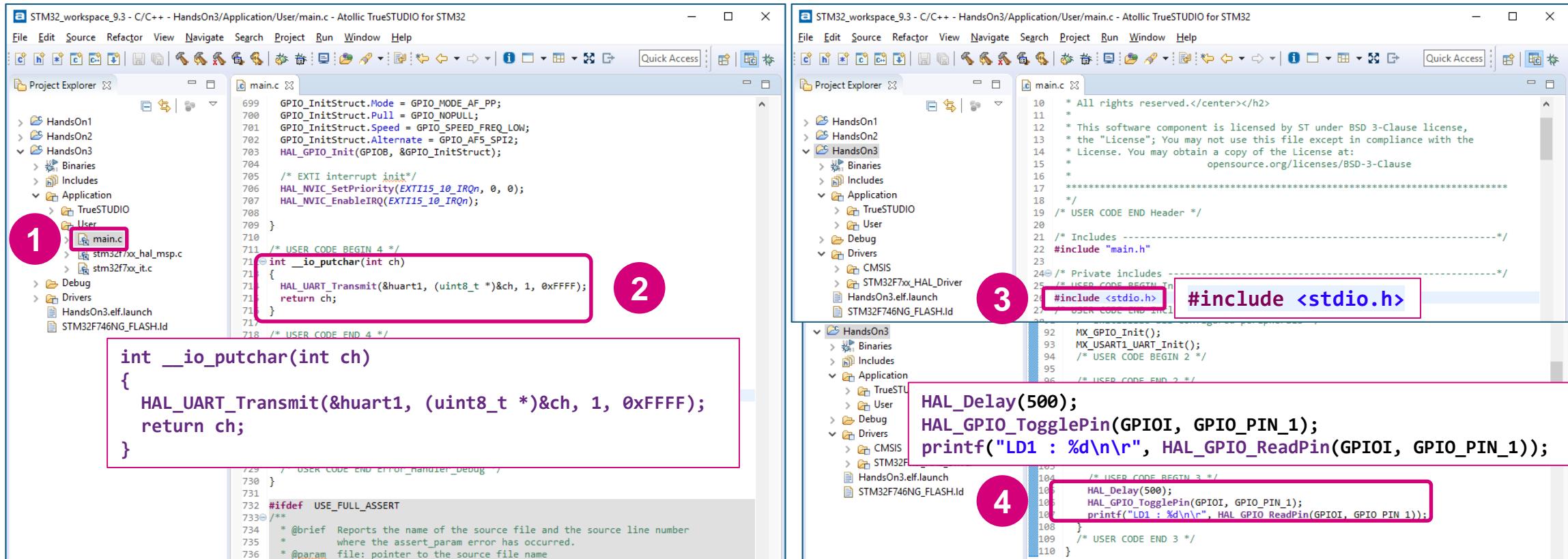
於Pinout & Configuration分頁中的Connectivity項目開啟USART1



Step 3: 加入程式碼/重新編譯/燒錄

79

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

```
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 */
```

1

2

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

```
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 1 */
26 #include <stdio.h>
27 /* USER CODE END 1 */
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
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 MX_GPIO_Init();
93 MX_USART1_UART_Init();
94 /* USER CODE BEGIN 2 */
95
96 /* USER CODE END 2 */
97
98
99
100
101
102
103
104
105
106
107
108
109
110
```

3

4

#include <stdio.h>

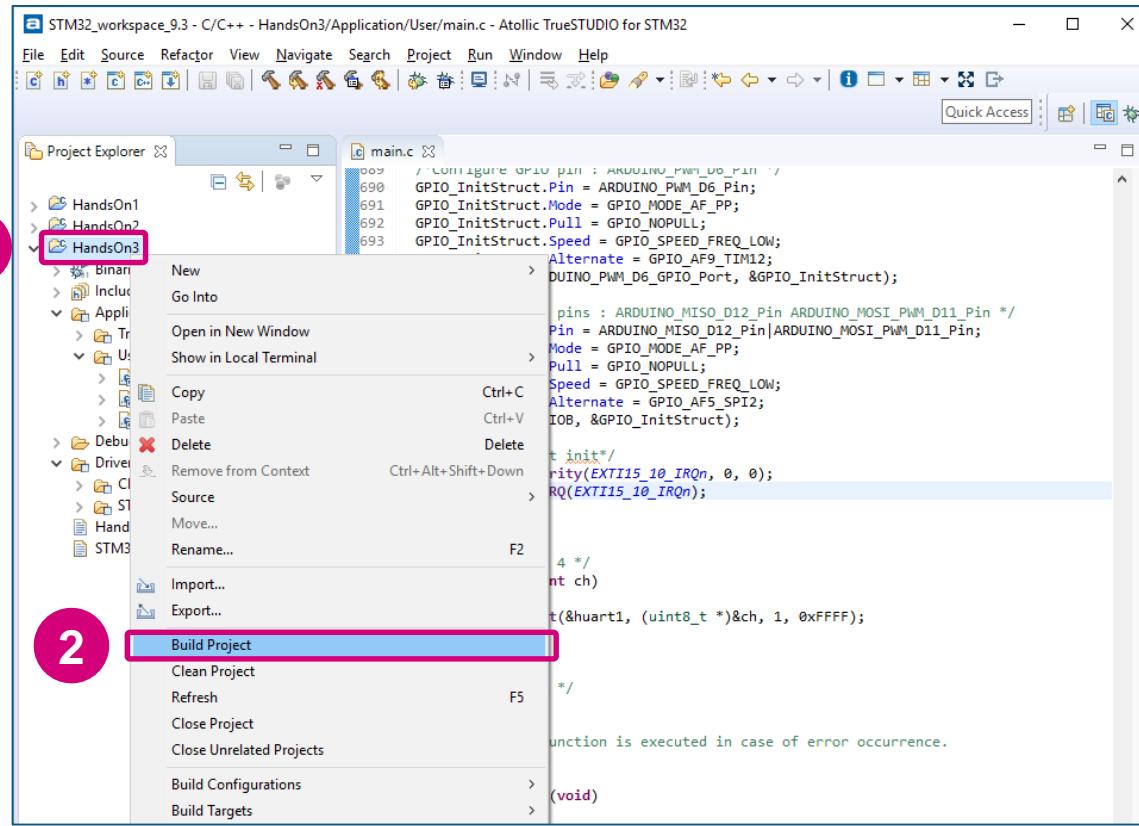
HAL_Delay(500);
HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));

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

80

重新編譯/燒錄



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

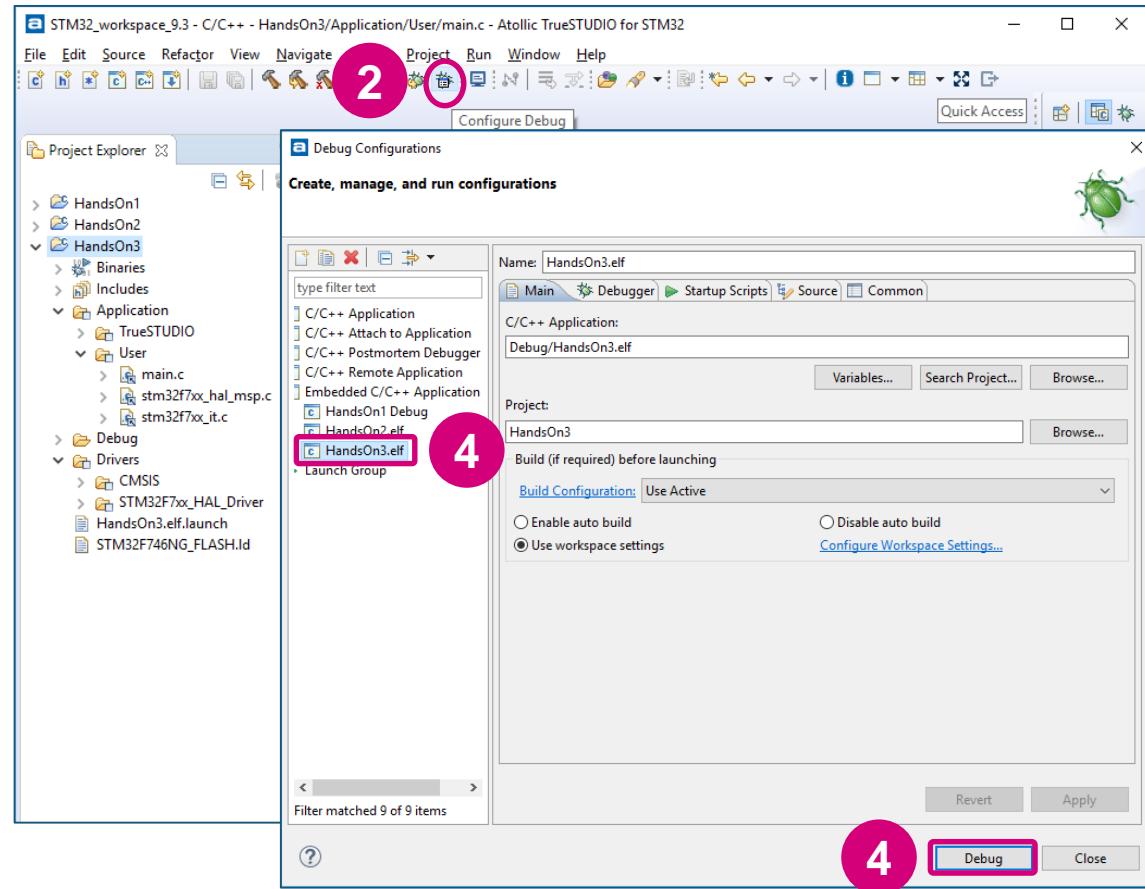
File Edit Source Refactor View Navigate Search Project Run Window Help

Project Explorer

- HandsOn1
- HandsOn2
- HandsOn3**

main.c

```
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 GPIO_InitStruct.Alternate = GPIO_AF9_TIM12;
695 DUEINO_PWM_D6_GPIO_Port, &GPIO_InitStruct);
696
697 pins : ARDUINO_MISO_D12_Pin ARDUINO_MOSI_PWM_D11_Pin */
698 Pin = ARDUINO_MISO_D12_Pin|ARDUINO_MOSI_PWM_D11_Pin;
699 Mode = GPIO_MODE_AF_PP;
700 Pull = GPIO_NOPULL;
701 Speed = GPIO_SPEED_FREQ_LOW;
702 Alternate = GPIO_AF5_SPI2;
703 IOB, &GPIO_InitStruct);
704
705 /* init*/
706 priority(EXTI15_10_IRQn, 0, 0);
707 IRQ(EXTI15_10_IRQn);
```



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

File Edit Source Refactor View Navigate Project Run Window Help

Project Explorer

- HandsOn1
- HandsOn2
- HandsOn3**

Debug Configurations

Create, manage, and run configurations

Name: HandsOn3.elf

C/C++ Application: Debug/HandsOn3.elf

Project: HandsOn3

Build (if required) before launching

Build Configuration: Use Active

Enable auto build Disable auto build

Use workspace settings Configure Workspace Settings...

Launch Group: HandsOn3.elf

Buttons: Revert Apply

Filter matched 9 of 9 items

4

4

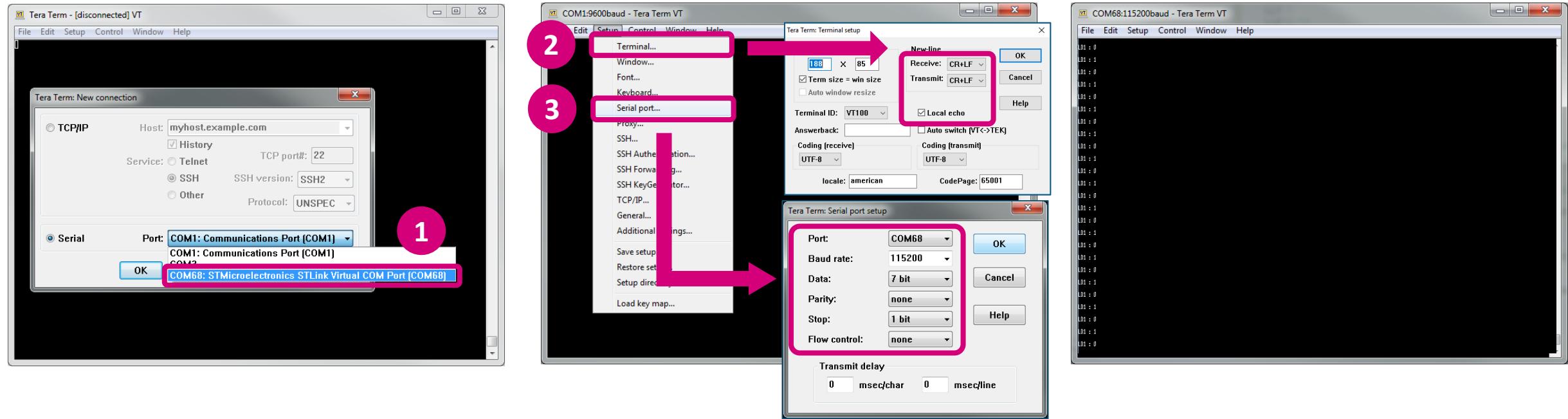
Debug

Close

Step 4: 開啟終端機檢視UART輸出

81

以TeraTerm為例



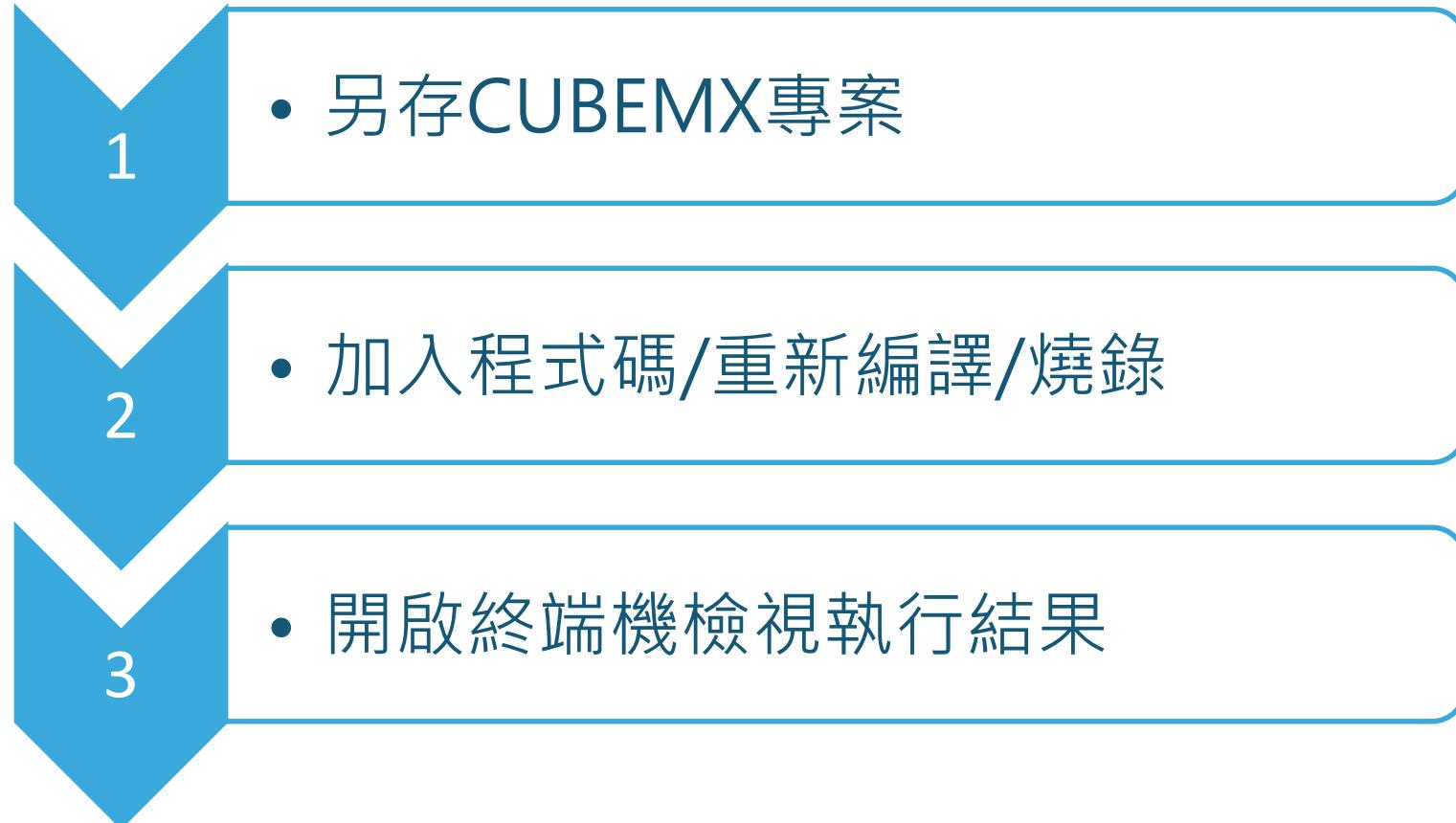
本次工作坊範例使用的終端機為Tera-Term，可由以下網址下載：

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



Hands-on 4

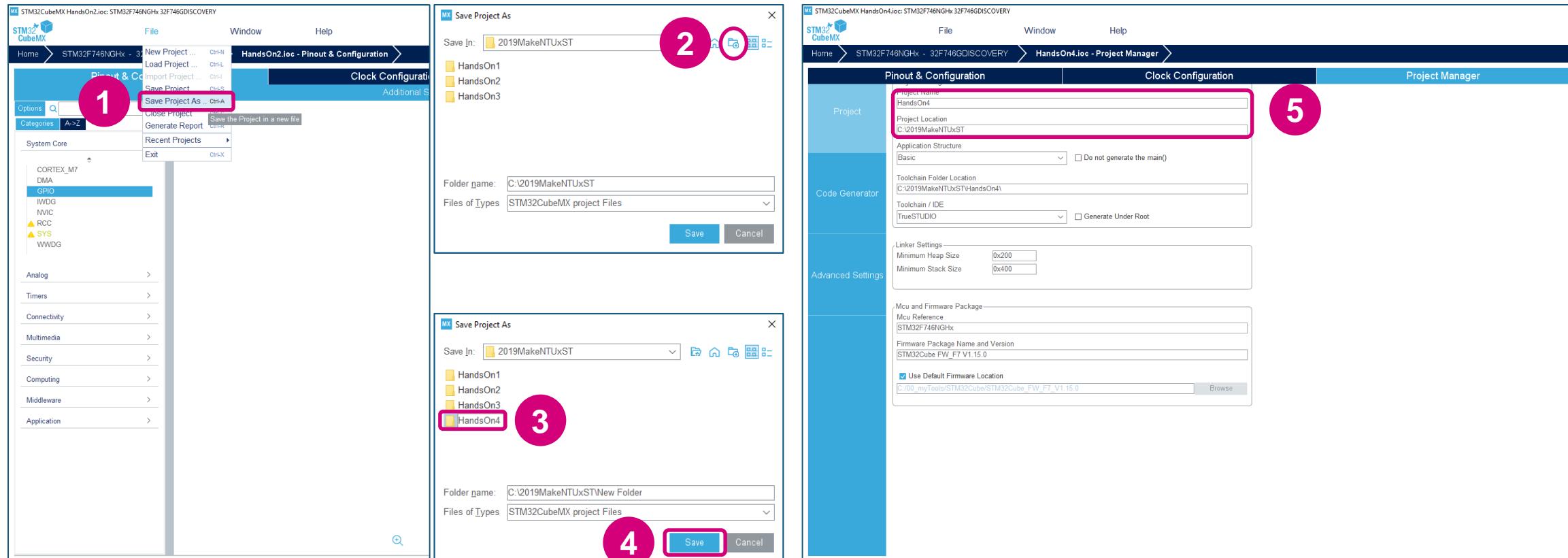
UART Polling Receiving over UART1-VCP



Step 1: 另存CUBEMX專案

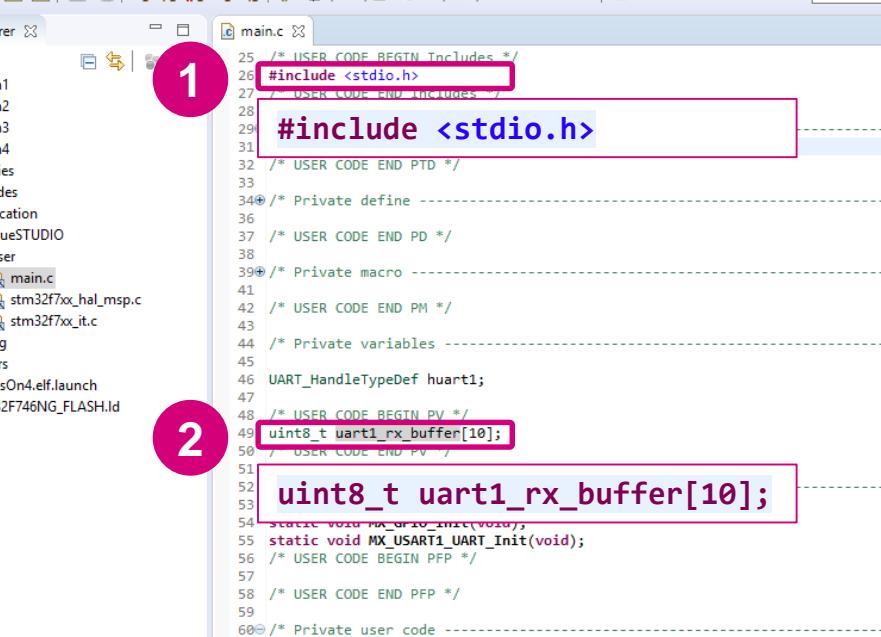
84

由HandsOn 3另存CUBEMX Project



Step 2: 加入程式碼/重新編譯/燒錄

於main.c加入程式碼



The screenshot shows the Atollic TrueSTUDIO IDE interface with the following details:

- Project Explorer:** Shows a tree structure with projects: HandsOn1, HandsOn2, HandsOn3, and HandsOn4. HandsOn4 is expanded to show Binaries, Includes, Application (containing main.c, stm32f7xx_hal_msp.c, and stm32f7xx_it.c), and User (containing Debug, Drivers, HandsOn4.elf.launch, and STM32F746NG_FLASH.ld).
- Code Editor:** The main.c file is open. The code includes a header file and a variable declaration. The variable declaration is highlighted with a pink box and circled with a pink number 2.
- Annotations:** Two pink circles with numbers 1 and 2 highlight specific code elements: the `#include <stdio.h>` line and the `uint8_t uart1_rx_buffer[10];` line.

```
25  /* USER CODE BEGIN Includes */
26  #include <stdio.h>
27  /* USER CODE END Includes */
28
29 #include <stdio.h>
30
31
32 /* USER CODE END PTD */
33
34 /* Private define */
35
36 /* USER CODE END PD */
37
38
39 /* Private macro */
40
41 /* USER CODE END PM */
42
43
44 /* Private variables */
45
46 UART_HandleTypeDef huart1;
47
48 /* USER CODE BEGIN PV */
49 uint8_t uart1_rx_buffer[10];
50 /* USER CODE END PV */
51
52
53
54 static void MX_GPIO_Init(void);
55 static void MX_USART1_UART_Init(void);
56 /* USER CODE BEGIN PFP */
57
58 /* USER CODE END PFP */
59
60 /* Private user code */
61 /* USER CODE BEGIN 0 */
62
63 /* USER CODE END 0 */
64
65 */
```

STM32_workspace_9.3 - C/C++ - HandsOn4/Application/User/main.c - Atollic TrueSTUDIO for STM32

File Edit Source Refactor View Navigate Search Project Run Window Help

Quick Access

Project Explorer

main.c

```
91  /* Initialize all configured peripherals */
92  MX_GPIO_Init();
93  MX_USART1_UART_Init();
94  /* USER CODE BEGIN 2 */
95  HAL_UART_Receive(&huart1, uart1_rx_buffer, 10, 0xFFFF);
96  /* USER CODE END 2 */
97
98
99
100
101
102
103
104
105
106
107
108
109
```

3

```
HAL_UART_Receive(&huart1, uart1_rx_buffer, 10, 0xFFFF);
```

4

```
HAL_UART_Receive(&huart1, uart1_rx_buffer, 10, 0xFFFF);

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

5

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

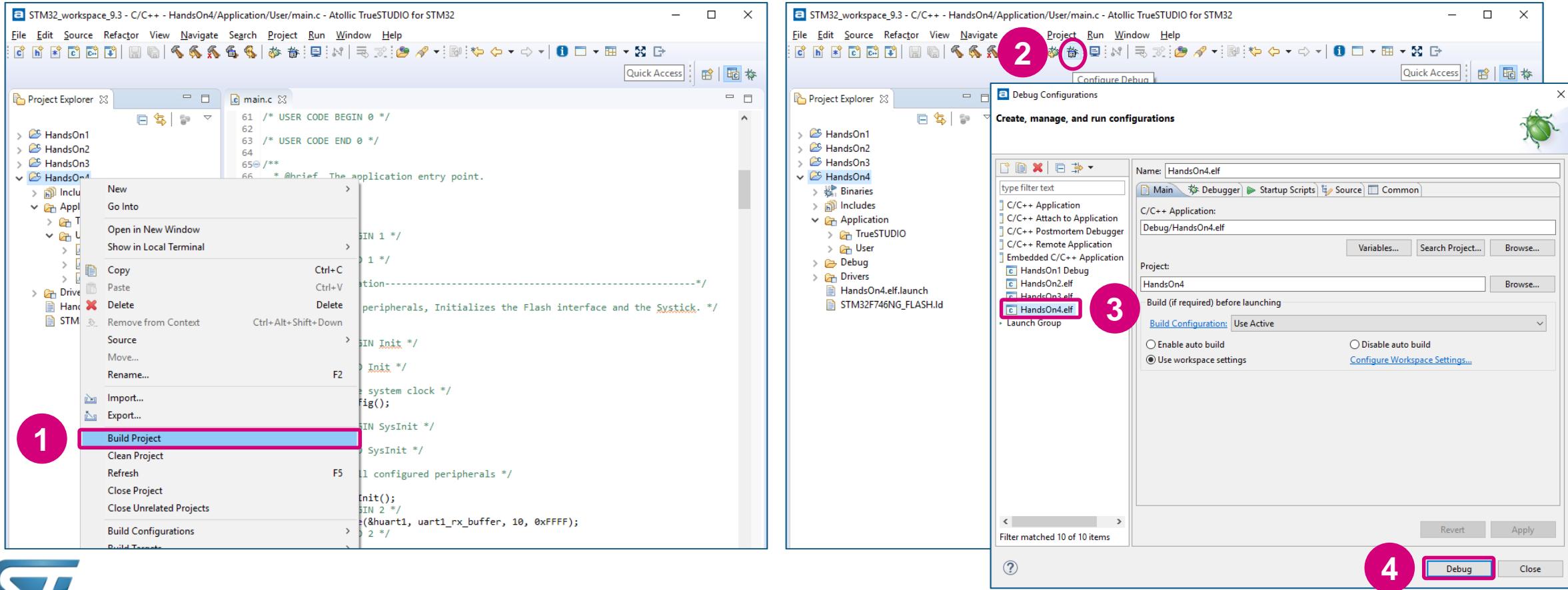
703  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
704
705  /* EXTI interrupt init*/
706  HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
707  HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
708
709 }
710
711 /* USER CODE BEGIN 4 */
712 int __io_putchar(int ch)
713 {
714  HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
715  return ch;
716 }
717 /* USER CODE END 4 */
```

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

Step 2: 加入程式碼/重新編譯/燒錄

86

重新編譯/燒錄



Step 3: 開啟終端機檢視執行結果

87

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 for STM32 debugger interface. A pink circle labeled '2 Suspend' highlights the 'Suspend' button in the toolbar. The 'Variables' tab is selected, showing a table for the 'uart1_rx_buffer' variable, which is defined as a uint8_t[10] array. The table lists the first 10 elements of the buffer, each with a value of '0x20000098 <uart1_rx_buffer[0]' and a hex value from '48 0' to '52 4'. A pink circle labeled '3 觀察uart2_rx_buffer' highlights the 'uart2_rx_buffer' entry in the table. Below the table, the code editor shows the 'main.c' and 'stm32f7xx_hal.c' files. The 'main.c' file contains a weak implementation for the HAL_GetTick function. On the right is the Tera Term VT terminal window, titled 'COM43 - Tera Term VT'. A pink circle labeled '1 在終端機下打字' highlights the text '01234' entered into the terminal. The terminal window is black with white text.

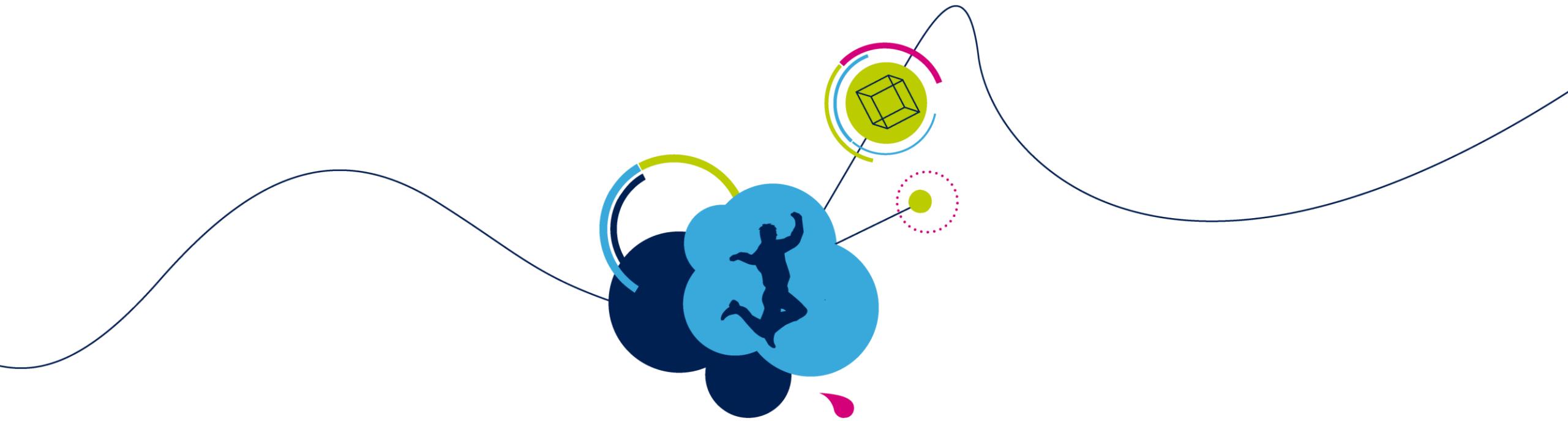
Expression	Type	Value
uart1_rx_buffer	uint8_t[10]	0x20000098 <uart1_rx_buffer[0]
(x): uart1_rx_buffer[0]	uint8_t	48 0'
(x): uart1_rx_buffer[1]	uint8_t	49 1'
(x): uart1_rx_buffer[2]	uint8_t	50 2'
(x): uart1_rx_buffer[3]	uint8_t	51 3'
(x): uart1_rx_buffer[4]	uint8_t	52 4'
(x): uart1_rx_buffer[5]	uint8_t	0 \0'
(x): uart1_rx_buffer[6]	uint8_t	0 \0'
(x): uart1_rx_buffer[7]	uint8_t	0 \0'
(x): uart1_rx_buffer[8]	uint8_t	0 \0'
(x): uart1_rx_buffer[9]	uint8_t	0 \0'

Step 3: 開啟終端機檢視執行結果

88

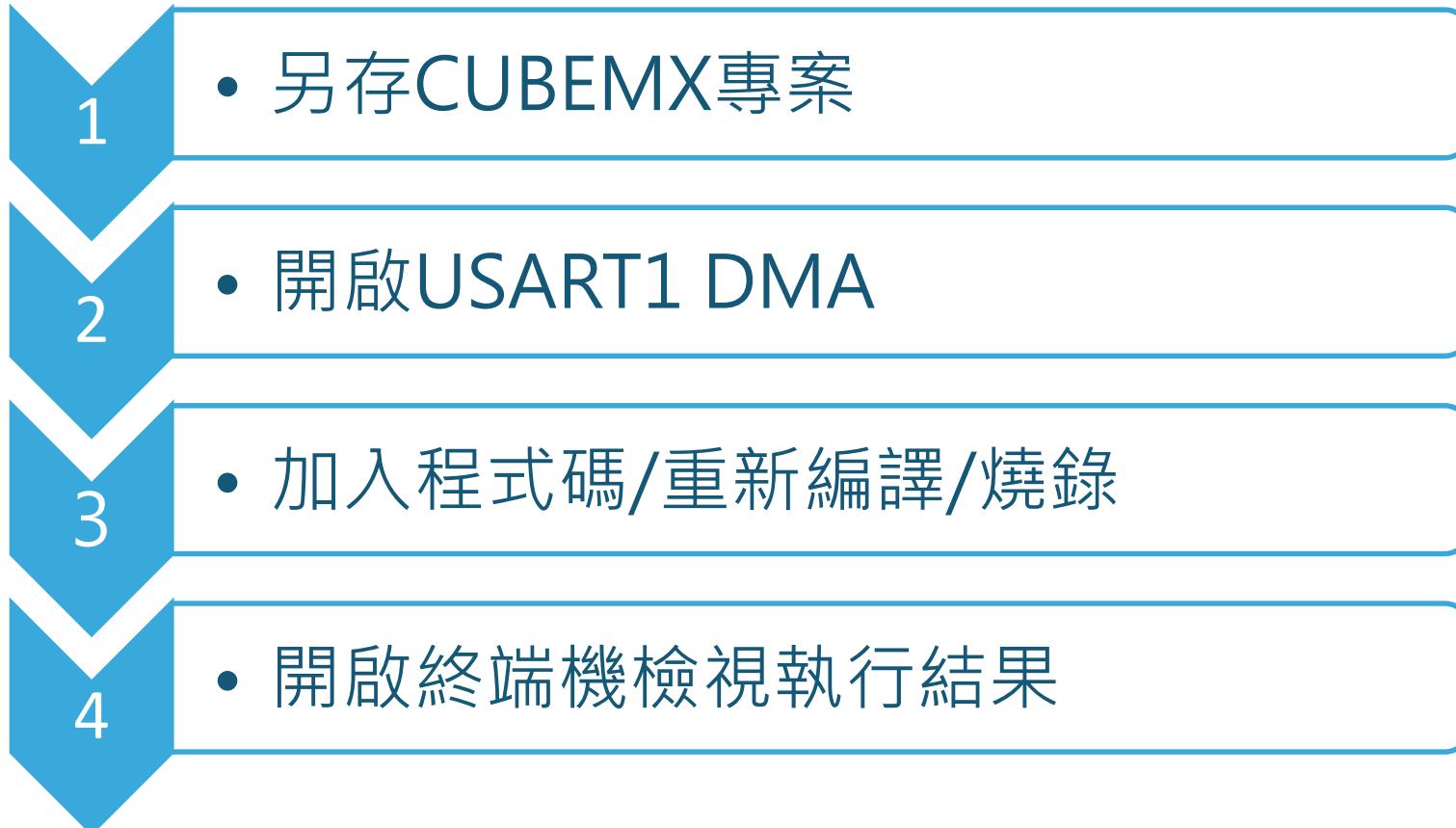
在終端機下打字，並觀察uart1_rx_buffer

The image shows the Atollic TrueSTUDIO for STM32 interface with two main windows. On the left is the 'Variables' window, which is highlighted with a yellow box. It displays the 'uart1_rx_buffer' variable as a uint8_t [10] array. The array contains the following values: 48 '0', 49 '1', 50 '2', 51 '3', 52 '4', 53 '5', 54 '6', 55 '7', 56 '8', and 57 '9'. A pink circle with the number '3' is placed over the 'Suspend' button in the toolbar. A pink circle with the number '4' is placed over the 'uart1_rx_buffer' table in the Variables window. On the right is a terminal window showing the text 'D123456789 01 : 1'. Below this, the terminal is outputting the character 'L' followed by a series of '0' and '1' bits. A pink circle with the number '1' is placed over the terminal input line. A pink circle with the number '2' is placed over the terminal output line. The terminal window has a black background with white text.



Hands-on 5

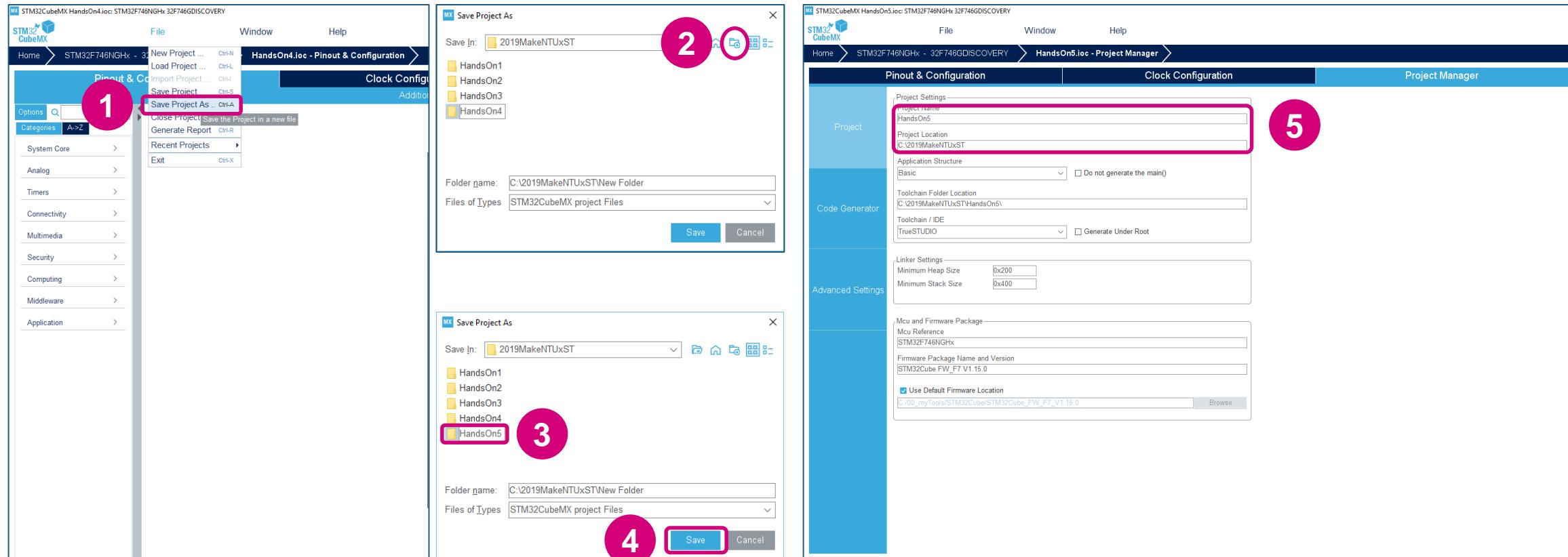
UART-DMA Receiving over UART1-VCP



Step 1: 另存CUBEMX專案

91

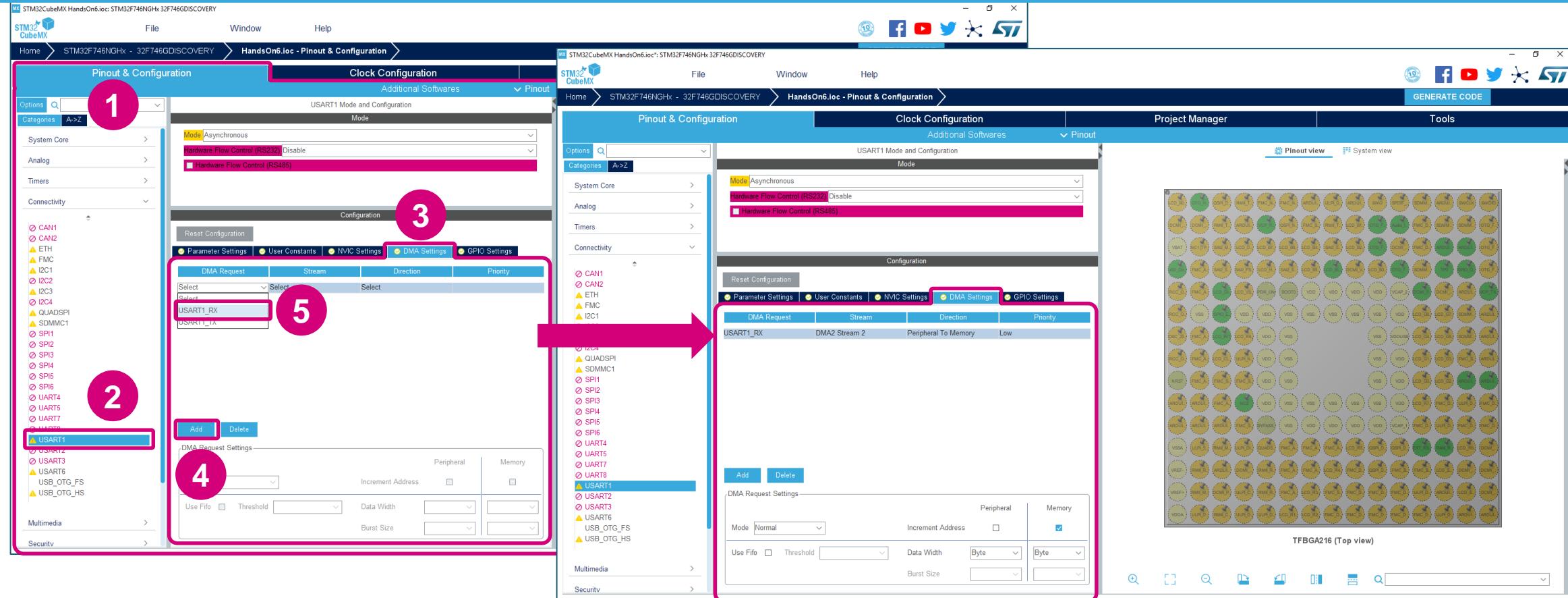
由HandsOn4另存CUBEMX Project



Step 2: 開啟USART1 DMA

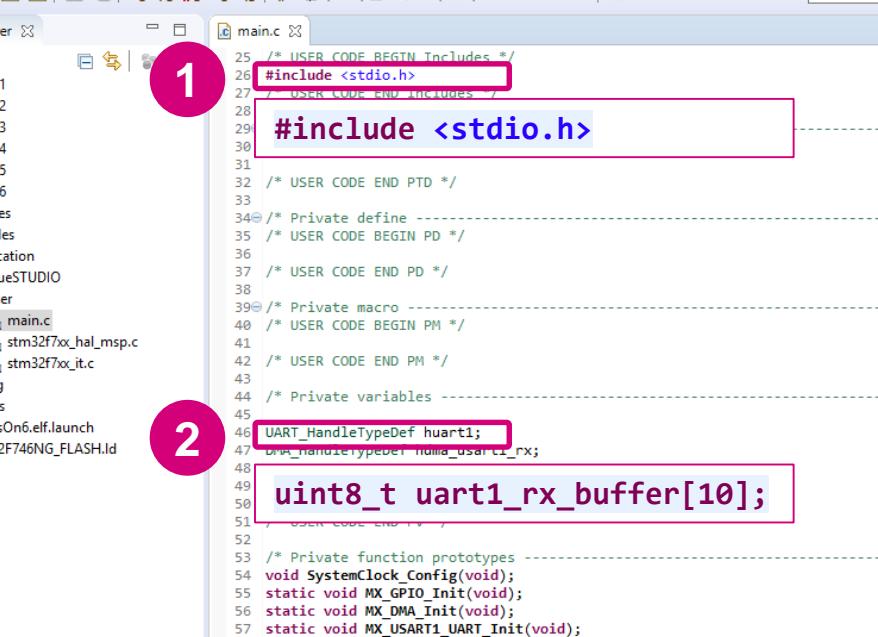
92

開啟USART1 DMA



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++ - HandsOn6/Application/User/main.c - Atollic TrueSTUDIO for STM32

File Edit Source Refactor View Navigate Search Project Run Window Help

Project Explorer

main.c

```
25 /* USER CODE BEGIN Includes */
26 #include <stdio.h>
27 /* USER CODE END Includes */
28
29 #include <stdio.h>
30
31
32 /* USER CODE END PTD */
33
34 /* Private define -----
35 /* USER CODE BEGIN PD */
36
37 /* USER CODE END PD */
38
39 /* Private macro -----
40 /* USER CODE BEGIN PM */
41
42 /* USER CODE END PM */
43
44 /* Private variables -----
45
46 UART_HandleTypeDef huart1;
47 DMA_HandleTypeDef handleDMA1;
48
49 uint8_t uart1_rx_buffer[10];
50
51 /* USER CODE END IV */
52
53 /* Private function prototypes -----
54 void SystemClock_Config(void);
55 static void MX_GPIO_Init(void);
56 static void MX_DMA_Init(void);
57 static void MX_USART1_UART_Init(void);
58 /* USER CODE BEGIN PFP */
59
60 /* USER CODE END PFP */
61
62 /* Private user code -----*/
```

STM32_workspace_9.3 - C/C++ - HandsOn6/Application/User/main.c - Atollic TrueSTUDIO for STM32

File Edit Source Refactor View Navigate Search Project Run Window Help

Quick Access

Project Explorer

main.c

```
91  /* USER CODE END SysInit */  
92  
93  /* Initialize all configured peripherals */  
94  MX_GPIO_Init();  
95  MX_DMA_Init();  
96  MX_USART1_UART_Init();  
97  /* USER CODE BEGIN 2 */  
98  HAL_UART_Receive_DMA(&huart1, uart1_rx_buffer, 10);  
99  /* USER CODE END 2 */  
100  
101  
102  while (1)  
103  {  
104      /* USER CODE END WHILE */  
105  
106      /* USER CODE BEGIN 3 */  
107      HAL_Delay(500);  
108      HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);  
109      printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);  
125  HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);  
126  
127 }  
128  
129  /* USER CODE BEGIN 4 */  
130  int __io_putchar(int ch)  
131  {  
132      HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);  
133      return ch;  
134 }  
135 /* USER CODE END 4 */  
136  
137 int __io_putchar(int ch)  
138 {  
139     HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);  
140     return ch;  
141 }
```

3

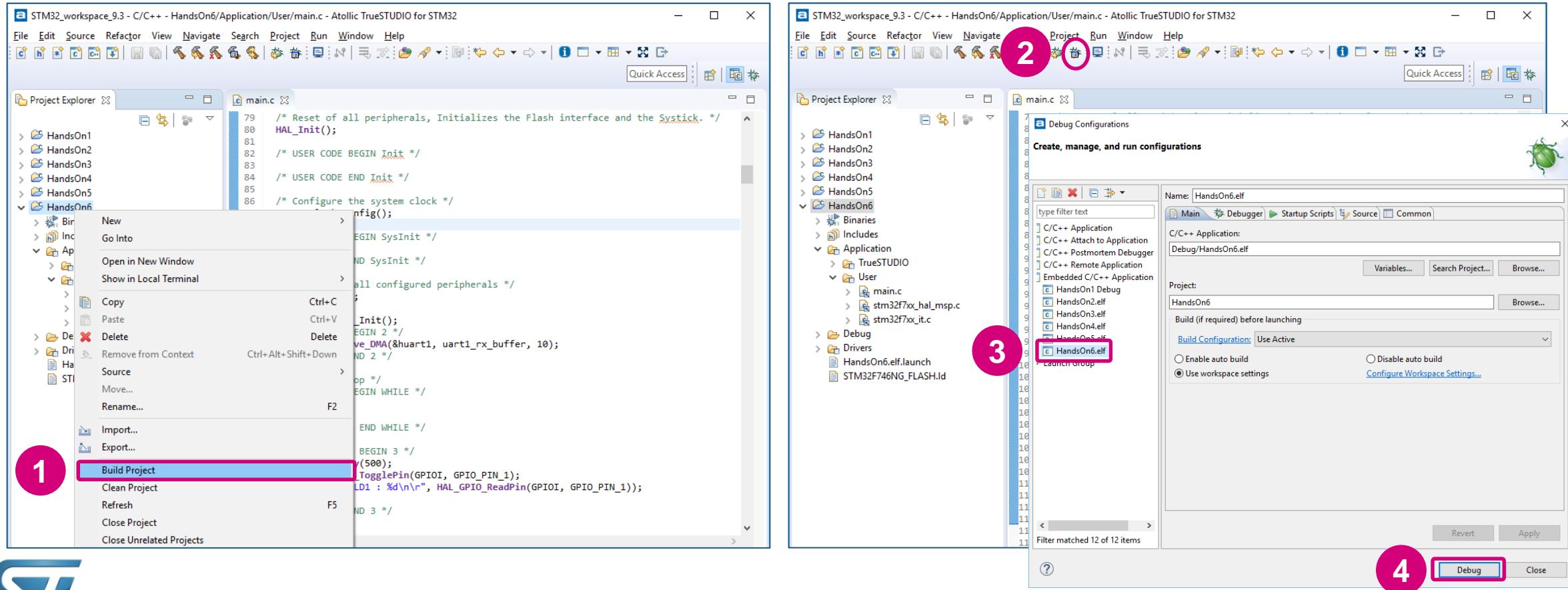
4

5

Step 3: 加入程式碼/重新編譯/燒錄

94

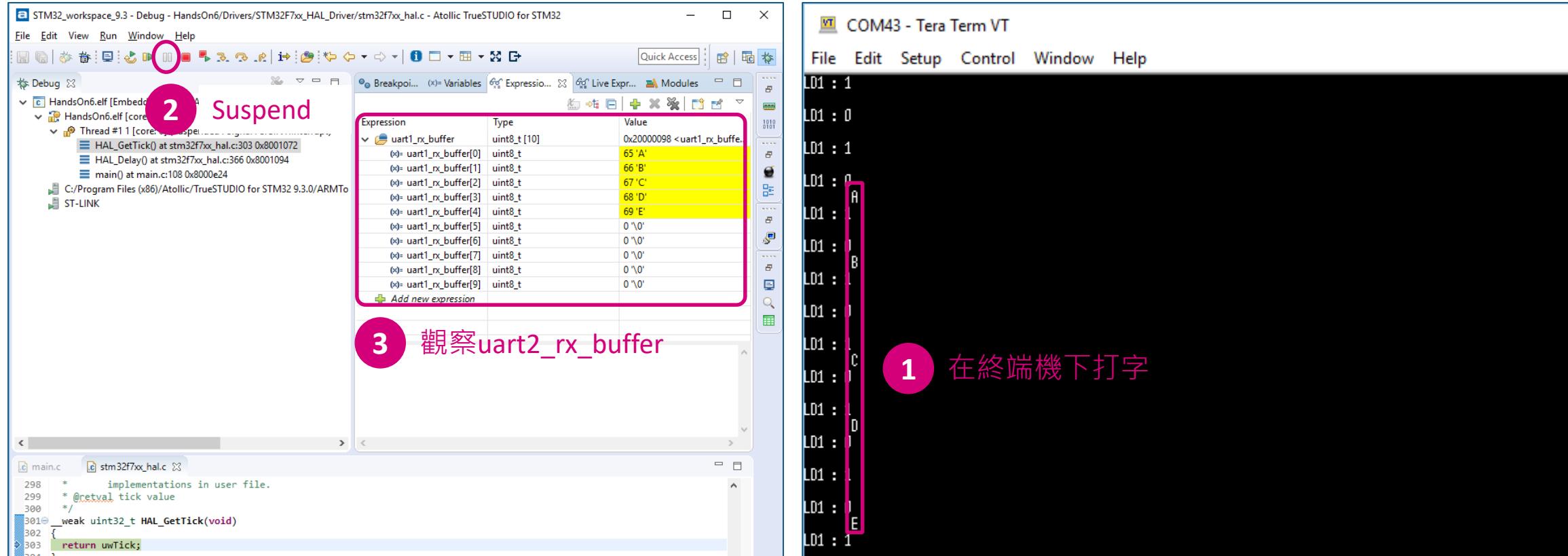
重新編譯/燒錄



Step 4: 開啟終端機檢視執行結果

95

在終端機下打字，並觀察uart1_rx_buffer





Hands-on 6

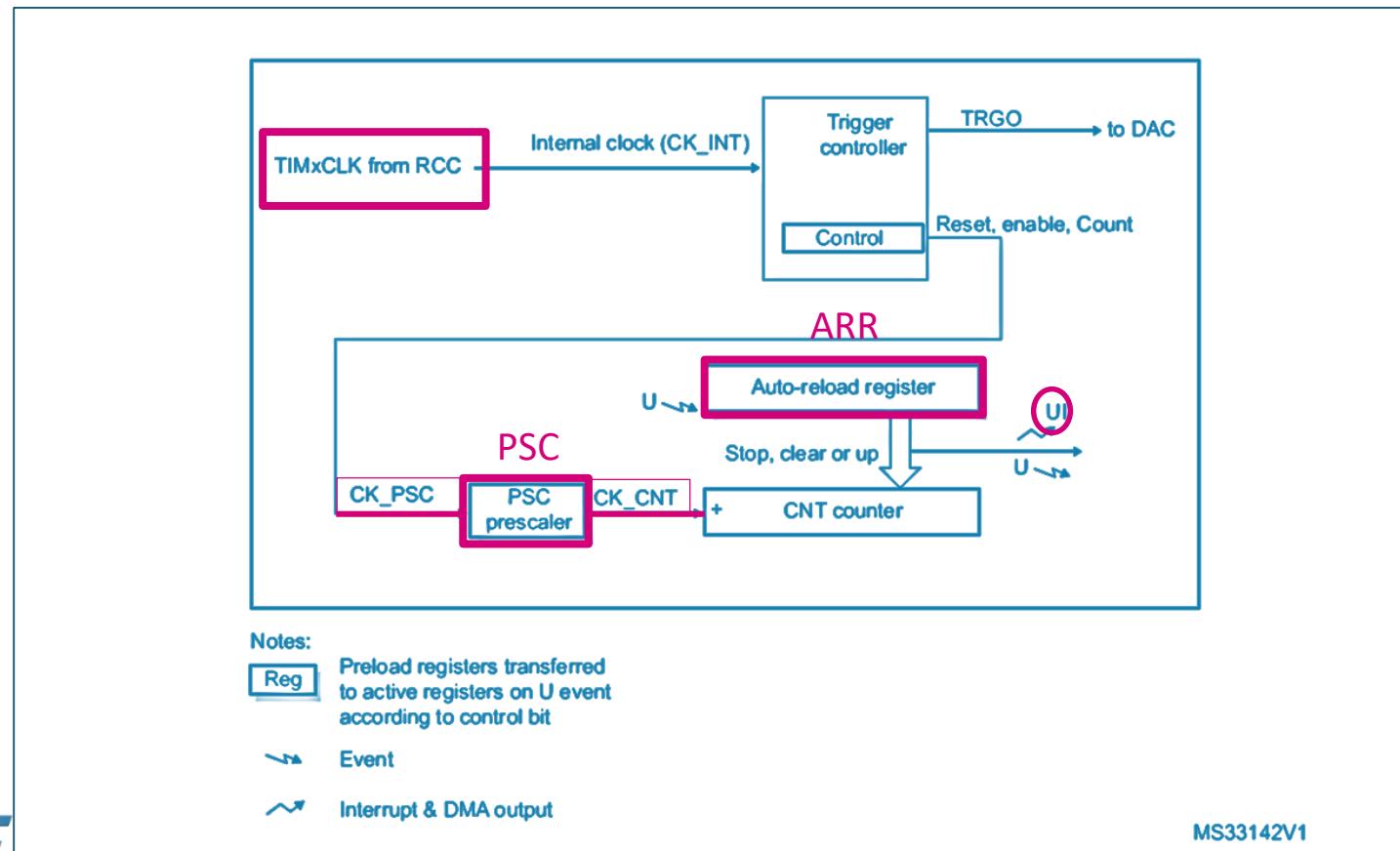
Timer

不靠HAL_Delay() 達成每1秒改變LD1閃滅狀態

Timer功能方塊示意圖

98

以Basic Timer (TIM6)為例

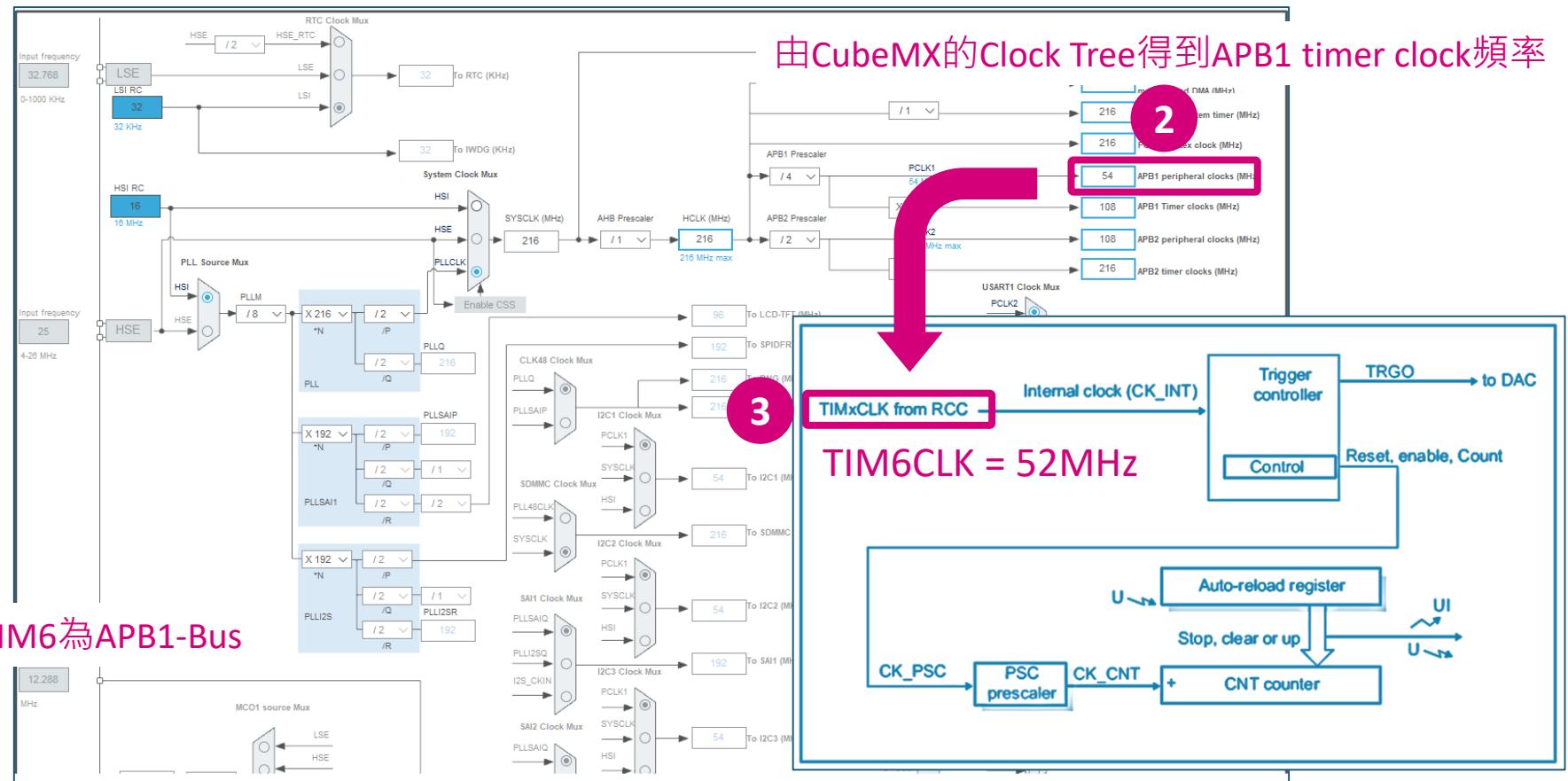


- $CK_{PSC} \text{ (Hz)} = TIMxCLK \text{ (Hz)}$
- $CK_{CNT} \text{ (Hz)} = CK_{PSC} \text{ (Hz)} / (PSC + 1)$
- $UI \text{ (Hz)} = CK_{CNT} \text{ (Hz)} / (ARR + 1)$

查閱RM0385

Boundary address	Peripheral	Bus	Register map
0x4000 7C00 - 0x4000 7FFF	UART8		Section 31.8.12: USART register map on page 1086
0x4000 7800 - 0x4000 7BFF	UART7		Section 16.5.15: DAC register map on page 482
0x4000 7400 - 0x4000 77FF	DAC		Section 4.4.4: PWR power control register 2 (PWR_CSR2) on page 129
0x4000 7000 - 0x4000 73FF	PWR		Section 39.7.7: HDMI-CEC register map on page 1668
0x4000 6C00 - 0x4000 6FFF	HDMI-CEC		
0x4000 6800 - 0x4000 6BFF	CAN2		Section 36.9.5: bxCAN register map on page 1336
0x4000 6400 - 0x4000 67FF	CAN1		
0x4000 6000 - 0x4000 63FF	I2C4		Section 30.7.12: I2C register map on page 1022
0x4000 5C00 - 0x4000 5FFF	I2C3		
0x4000 5800 - 0x4000 5BFF	I2C2		Section 30.7.12: I2C register map on page 1022
0x4000 5400 - 0x4000 57FF	I2C1		
0x4000 5000 - 0x4000 53FF	UART5		
0x4000 4C00 - 0x4000 4FFF	UART4		Section 31.8.12: USART register map on page 1086
0x4000 4800 - 0x4000 4BFF	USART3		
0x4000 4400 - 0x4000 47FF	USART2		
0x4000 4000 - 0x4000 43FF	SPDIFRX		Section 34.5.10: SPDIFRX interface register map on page 1235
0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3		
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2		Section 32.9.10: SPI/I2S register map on page 1146
0x4000 3000 - 0x4000 33FF	IWDG		Section 27.4.6: IWDG register map on page 905
0x4000 2C00 - 0x4000 2FFF	WWDG		Section 28.4.4: WWDG register map on page 912
0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers		Section 29.6.21: RTC register map on page 956
0x4000 2400 - 0x4000 27FF	LPTIM1		Section 26.7.9: LPTIM register map on page 896
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	Section 23.4.21: TIMx register map on page 810
0x4000 0C00 - 0x4000 0FFF	TIM5		
0x4000 0800 - 0x4000 0BFF	TIM4		
0x4000 0400 - 0x4000 07FF	TIM3		
0x4000 0000 - 0x4000 03FF	TIM2		

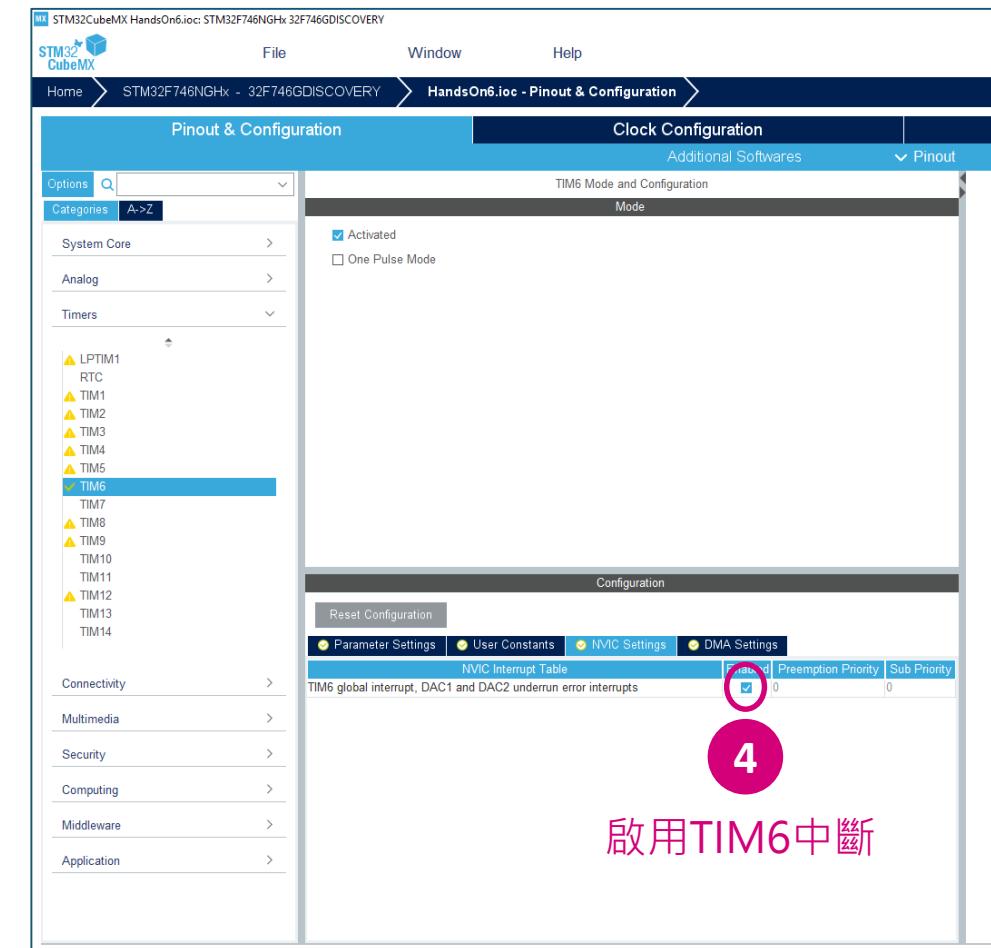
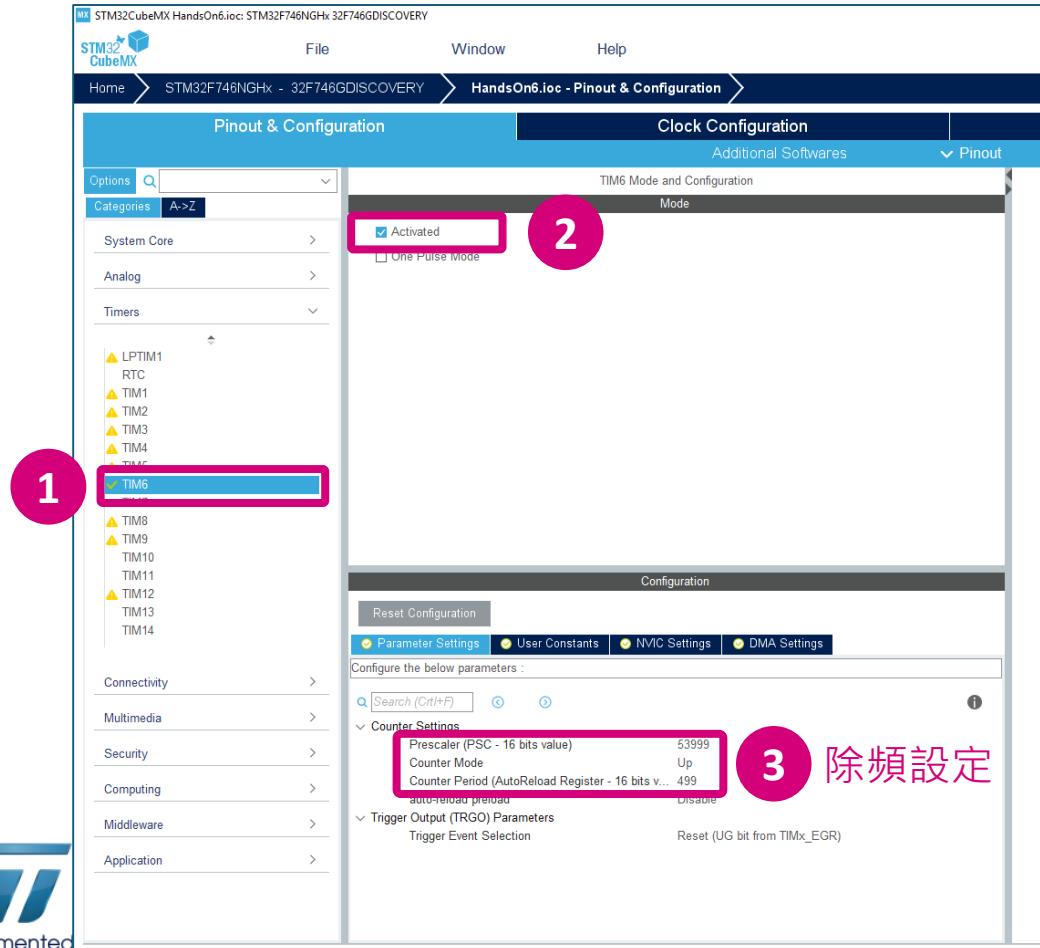
查閱RM得到TIM6為APB1-Bus



由CUBEMX產生TIM6專案

100

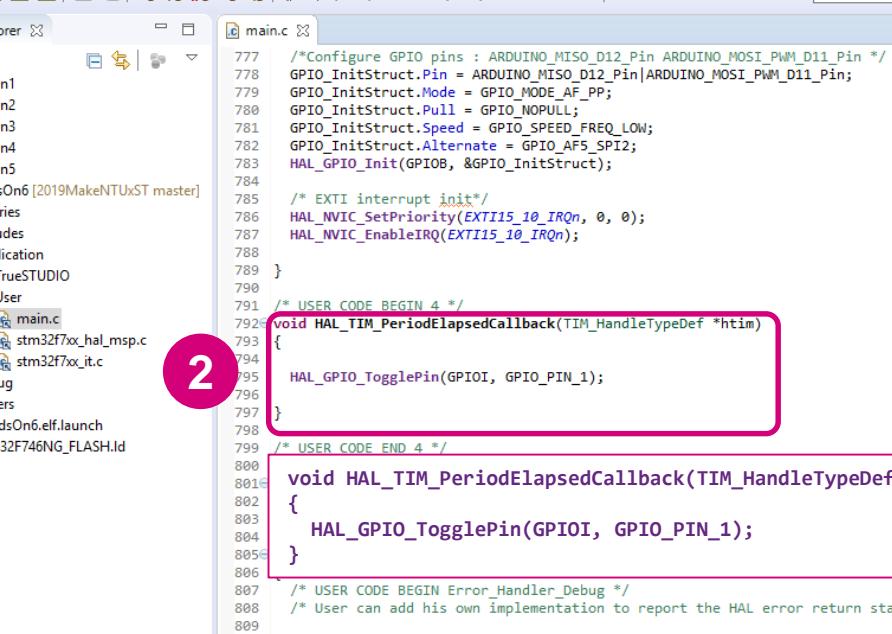
啟動TIM6並設定除頻參數和開啟TIM6中斷



加入程式碼

101

於main.c加入程式碼



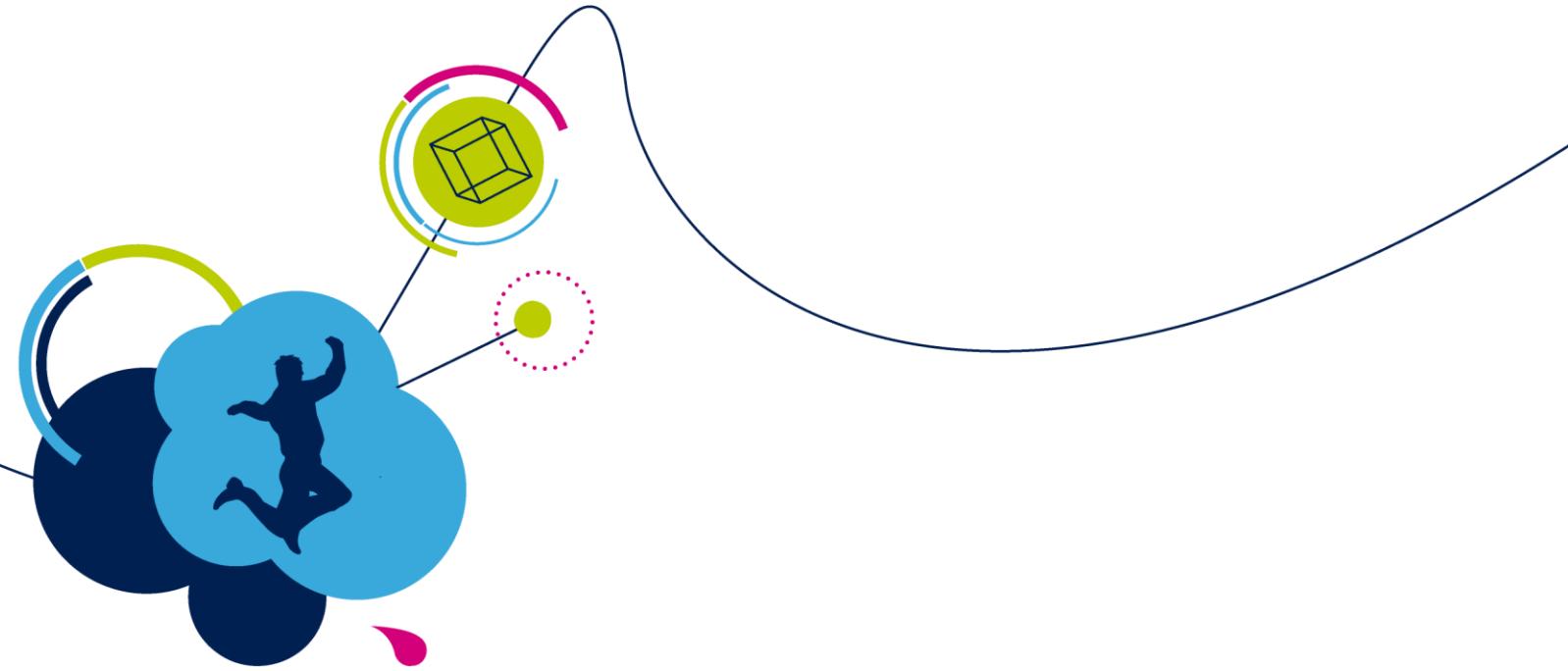
STM32_workspace_9.3 - C/C++ - HandsOn6/Application/User/main.c - Atollic TrueSTUDIO for STM32

File Edit Source Refactor View Navigate Search Project Run Window Help

Project Explorer

main.c

```
777  /*Configure GPIO pins : ARDUINO_MISO_D12_Pin ARDUINO_MOSI_PWM_D11_Pin */  
778  GPIO_InitStruct.Pin = ARDUINO_MISO_D12_Pin|ARDUINO_MOSI_PWM_D11_Pin;  
779  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;  
780  GPIO_InitStruct.Pull = GPIO_NOPULL;  
781  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;  
782  GPIO_InitStruct.Alternate = GPIO_AF5_SPI2;  
783  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);  
784  
785  /* EXTI interrupt init*/  
786  HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);  
787  HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);  
788 }  
789  
790 /* USER CODE BEGIN 4 */  
791 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)  
792 {  
793  
794  HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);  
795  
796 }  
797  
798 /* USER CODE END 4 */  
799  
800 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)  
801 {  
802  
803  HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);  
804  
805 }  
806  
807 /* USER CODE BEGIN Error_Handler_Debug */  
808 /* User can add his own implementation to report the HAL error return state */  
809  
810 /* USER CODE END Error_Handler_Debug */  
811 }  
812  
813 #ifdef USE_FULL_ASSERT  
814 /*
```



Hands-on 7

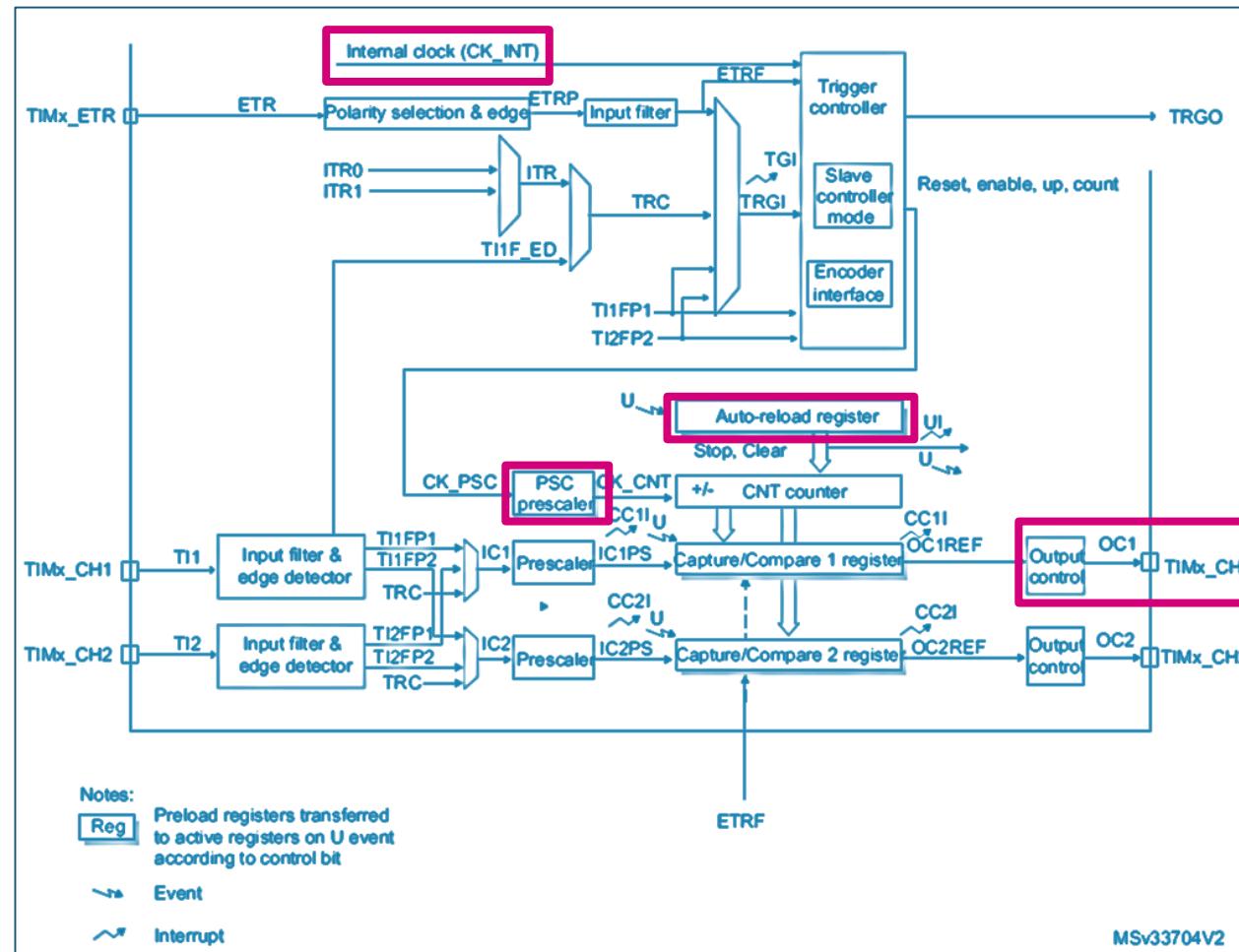
Timer-PWM

不靠HAL_Delay() 且不靠任何ISR 達成每500ms改變LD1閃滅狀態

Timer功能方塊示意圖

104

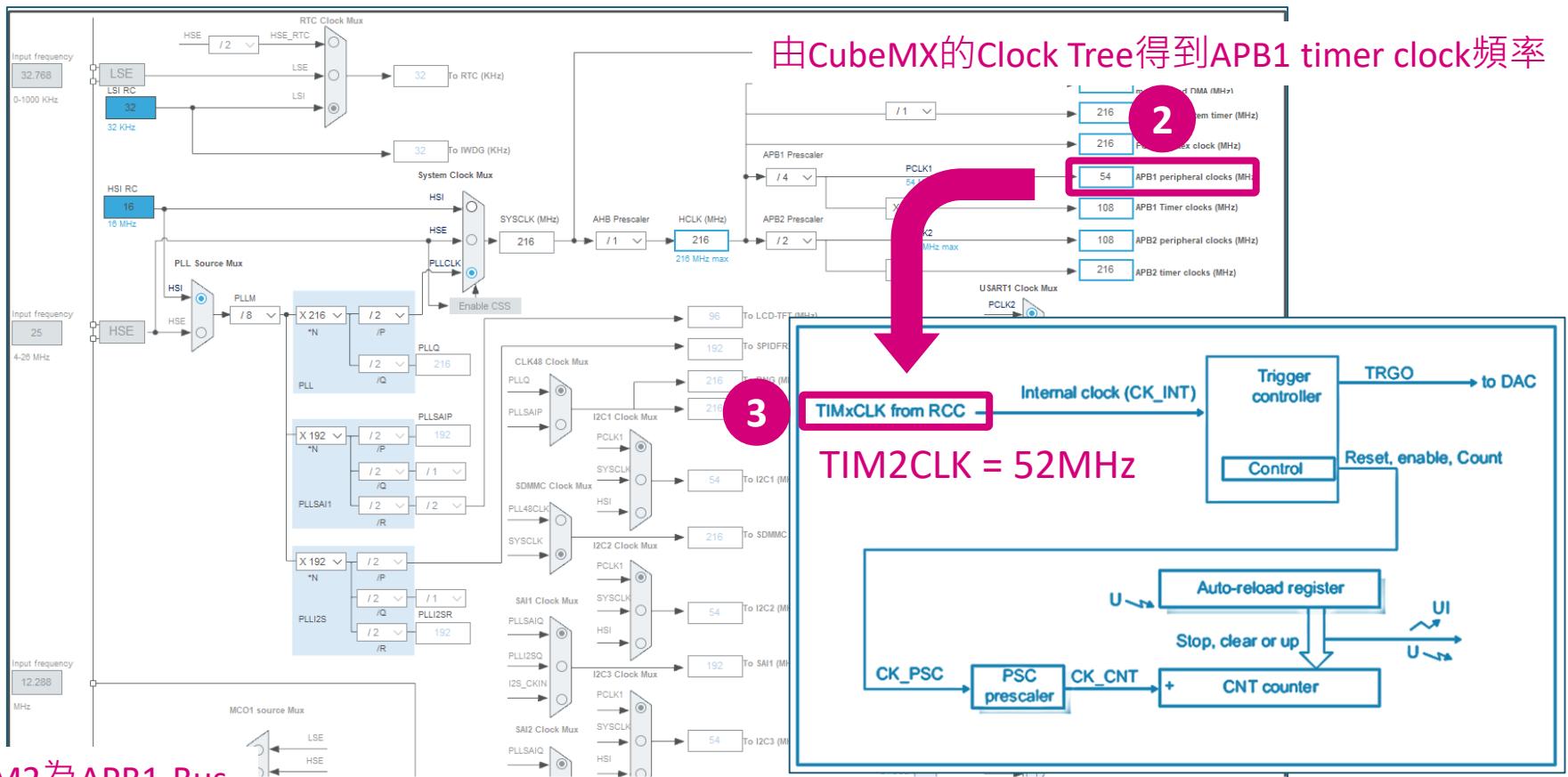
以General Purpose Timer (TIM2)為例



查閱RM0385

Boundary address	Peripheral	Bus	Register map
0x4000 7C00 - 0x4000 7FFF	UART8	APB1	Section 31.8.12: USART register map on page 1086
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 / I2S3		
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2		
0x4000 3000 - 0x4000 33FF	IWDG		
0x4000 2C00 - 0x4000 2FFF	WWDG		
0x4000 2800 - 0x4000 2BFF	RTC & BKP Register		
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		

查閱RM得到TIM2為APB1-Bus



由CubeMX的Clock Tree得到APB1 timer clock頻率

1

2

3

由CUBEMX產生TIM2專案

106

啟動TIM2並設定除頻參數

1

2

3 除頻設定

4 設定Duty Cycle

1

2

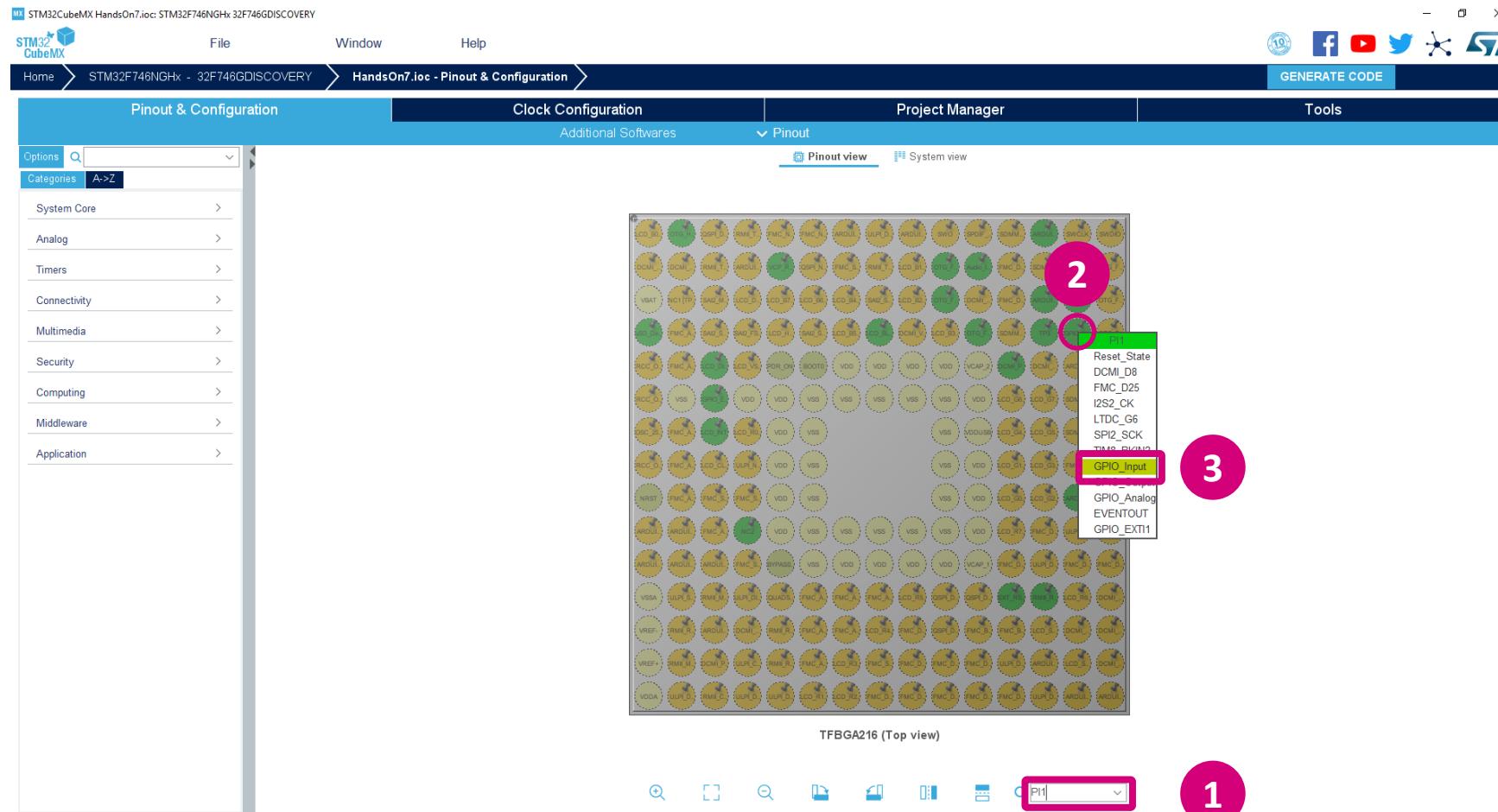
3 除頻設定

4 設定Duty Cycle

由CUBEMX產生TIM2專案

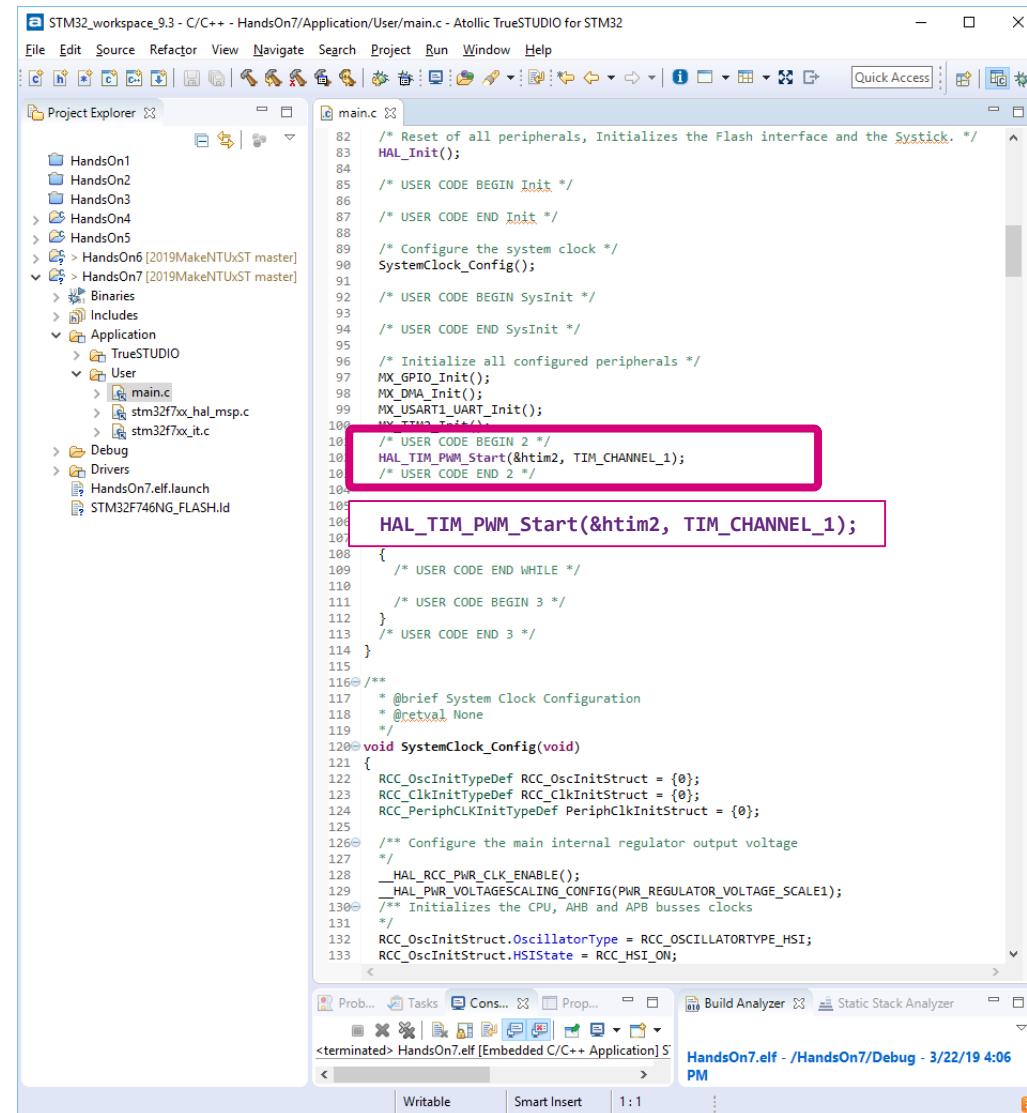
107

變更PI1的設定(由GPIO Output變為GPIO Input)



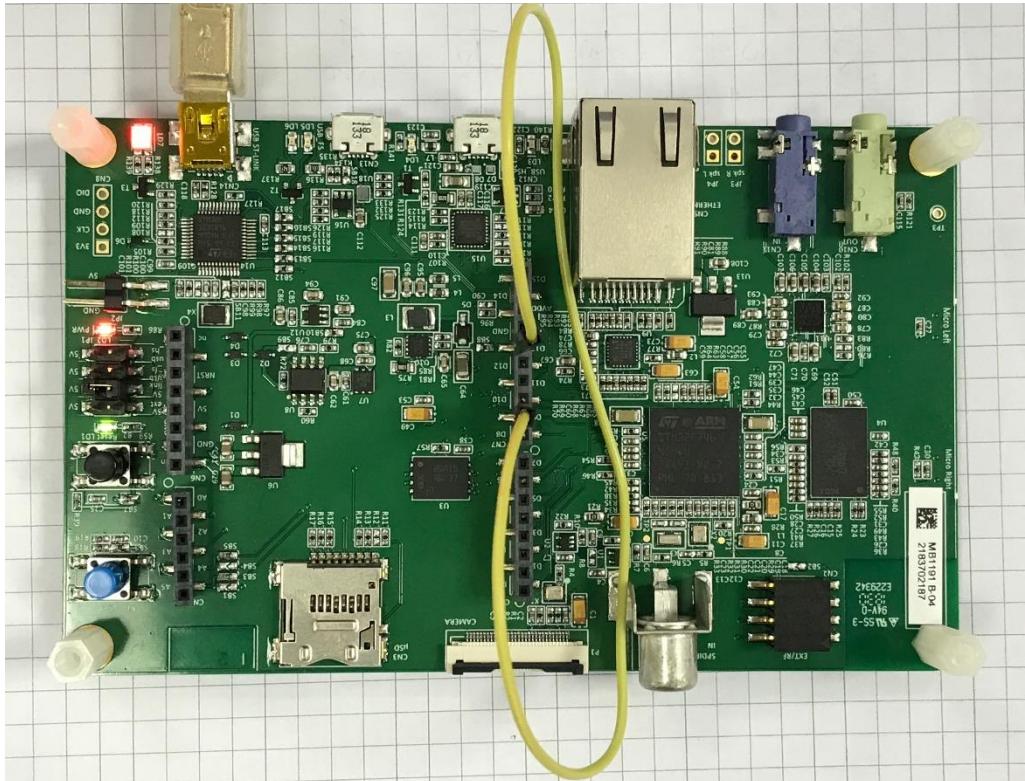
加入程式碼

108



```
82  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
83  HAL_Init();
84
85  /* USER CODE BEGIN Init */
86
87  /* USER CODE END Init */
88
89  /* Configure the system clock */
90  SystemClock_Config();
91
92  /* USER CODE BEGIN SysInit */
93
94  /* USER CODE END SysInit */
95
96  /* Initialize all configured peripherals */
97  MX_GPIO_Init();
98  MX_DMA_Init();
99  MX_USART1_UART_Init();
100 /* TIM2 init */
101 /* USER CODE BEGIN 2 */
102 HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
103 /* USER CODE END 2 */
104
105
106 HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
107
108 {
109  /* USER CODE END WHILE */
110
111  /* USER CODE BEGIN 3 */
112 }
113 /* USER CODE END 3 */
114 }
115
116 /**
117  * @brief System Clock Configuration
118  * @retval None
119  */
120 void SystemClock_Config(void)
121 {
122  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
123  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
124  RCC_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};
125
126  /* Configure the main internal regulator output voltage
127  */
128  __HAL_RCC_PWR_CLK_ENABLE();
129  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
130  /* Initializes the CPU, AHB and APB busses clocks
131  */
132  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
133  RCC_OscInitStruct.HSIStrate = RCC_HSI_ON;
```

CN6		CN7	
Reserved	1 NC	SCL/D15	10 PB8
3V3	2 IOREF	SDA/D14	9 PB9
NRST	3 RESET	AVDD	8 AVDD
3V3	4 3V3	GND	7 GND
5V	5 5V	SCK/D13	6 PI1 / LD1
GND	6 GND	MISO/D12	5 PB14
GND	7 GND	PWM/MOSI/D11	4 PB15
VIN	8 VIN	PWM/D10	3 PA8
PA0		PWM/D9	2 PA15
PF10	1 A0	D8	1 PI2
PF9	2 A1	CN4	
PF8	3 A2	D7	8 PI3
PF7	4 A3	PWM/D6	7 PH6
PF6	5 A4	PWM/CS/D5	6 PIO
	6 A5	D4	5 PG7
		PWM/D3	4 PB4
		D2	3 PG6
		TX/D1	2 PC6
		RX/D0	1 PC7
CN5		CN7	



Releasing Your Creativity

110



 /STM32

 @ST_World

 ST community