



# **SVB2640 BLE Wireless Module**

## **User Guide**

Rev. 1.1

Aug. 2018

## 目錄

一、介紹 .....	2
1.1 工具及開發環境 .....	2
1.2 硬體規格 .....	2
二、硬體配置 .....	3
三、接腳圖 .....	3
四、接腳描述 .....	4
五、開發環境下載與安裝 .....	5
5.1 安裝 Code Composer Studio .....	5
5.2 搜索 CCS 產品 .....	6
5.3 匯入 CCS 專案 .....	7
六、硬體設置 .....	9
七、程式碼編譯與燒錄 .....	10
八、腳位操作說明 .....	12
九、版本歷史 .....	13

## 一、介紹

Sivann SVB2640 BLE Wireless Module (低功耗藍牙無線模組) 採用佐臻 (Jorjin) 所生產之 ZB7412 模組。它是一款基於德州儀器 (Texas Instruments) CC2640R2F 藍牙單晶片之系統級封裝模組，內有 32 位元的 ARM Cortex-M3 微控制器，並提供豐富的周邊 I/O。適用於 Bluetooth 4.2 和 Bluetooth 5 之低功耗應用，且通過 FCC、ETSI、NCC 等多國射頻法規認證。其韌體能透過 CCS 或 IAR 等集成開發環境進行開發，可依需求實現藍牙的物聯網應用。



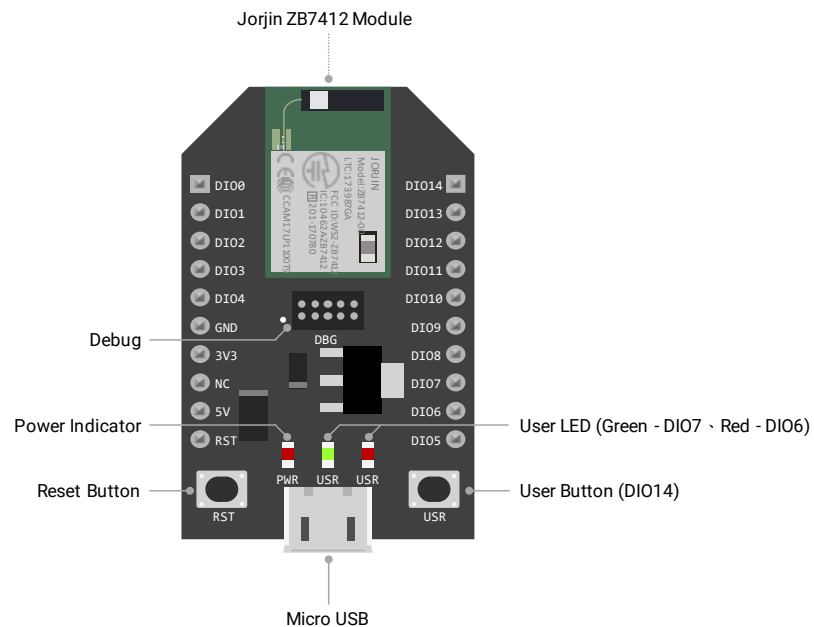
### 1.1 工具及開發環境

- Full-Feature and Low-Cost Development Kits
- Packet Sniffer PC Software
- Sensor Controller Studio
- SmartRF Studio
- SmartRF Flash Programmer 2
- IAR Embedded Workbench for ARM
- Code Composer Studio

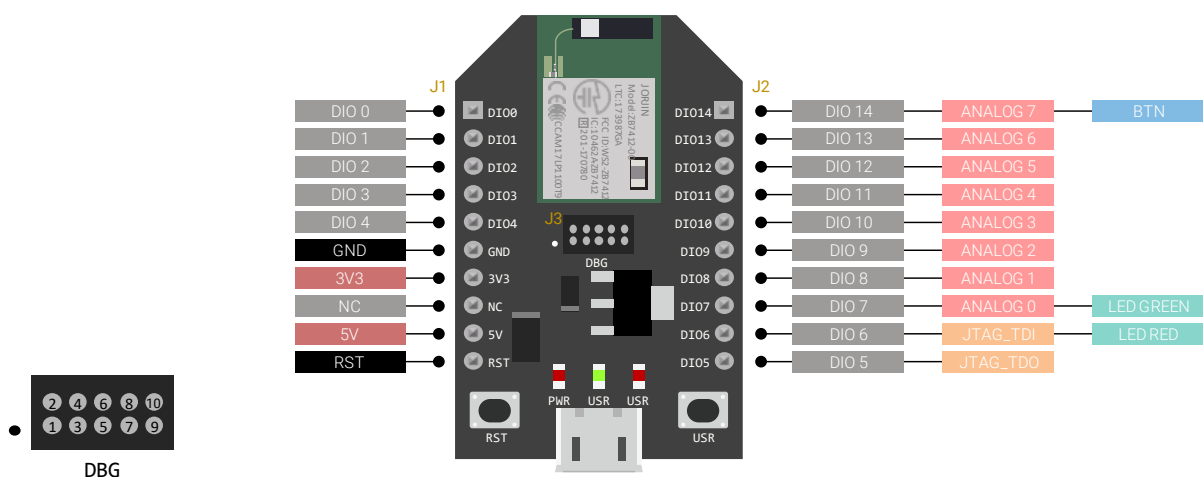
### 1.2 硬體規格

SVB2640 BLE Wireless Module	
無線模組	Jorjin ZB7412-00 (TI CC2640R2F 5x5mm, 15 GPIOs)
藍牙標準	Bluetooth 5
電 源	5 V (Micro USB)
周 邊	GPIO、UART、I2C、I2S、SPI、PWM、Timer、ADC、RTC
燒錄介面	2-Pin cJTAG
尺 寸	42 x 26 mm
重 量	5.3 g
產 地	台灣

## 二、硬體配置



## 三、接腳圖



## 四、接腳描述

Pin No.	Pin Name	Type	Description
J1			
1	DIO 0	Digital I/O	GPIO, Sensor Controller
2	DIO 1	Digital I/O	GPIO, Sensor Controller
3	DIO 2	Digital I/O	GPIO, Sensor Controller, High drive capability
4	DIO 3	Digital I/O	GPIO, Sensor Controller, High drive capability
5	DIO 4	Digital I/O	GPIO, Sensor Controller, High drive capability
6	GND	GND	Ground
7	3V3	Power	3V3
8	NC	NC	No Connection
9	5V	Power	5V
10	RST	Digital Input	Reset, Active-low
J2			
1	DIO 14	Digital/Analog I/O	GPIO, Sensor Controller, Analog
2	DIO 13	Digital/Analog I/O	GPIO, Sensor Controller, Analog
3	DIO 12	Digital/Analog I/O	GPIO, Sensor Controller, Analog
4	DIO 11	Digital/Analog I/O	GPIO, Sensor Controller, Analog
5	DIO 10	Digital/Analog I/O	GPIO, Sensor Controller, Analog
6	DIO 9	Digital/Analog I/O	GPIO, Sensor Controller, Analog
7	DIO 8	Digital/Analog I/O	GPIO, Sensor Controller, Analog
8	DIO 7	Digital/Analog I/O	GPIO, Sensor Controller, Analog
9	DIO 6	Digital I/O	GPIO, High drive capability, JTAG_TDI
10	DIO 5	Digital I/O	GPIO, High drive capability, JTAG_TDO
J3 Debug			
1	3V3	Power	3V3
2	JTAG_TMSC	Digital I/O	JTAG_TMSC, High drive capability
3	GND	GND	Ground
4	JTAG_TCKC	Digital I/O	JTAG_TCKC
5	GND	GND	Ground
6	JTAG_TDO	Digital I/O	JTAG_TDO, High drive capability, DIO 5
7	NC	NC	No Connection
8	JTAG_TDI	Digital I/O	JTAG_TDI, High drive capability, DIO 6
9	GND	GND	Ground
10	RST	Digital Input	Reset, Active-low

## 五、開發環境下載與安裝

CC2640R2F 支援 IAR Embedded Workbench for ARM、Code Composer Studio™ (CCS) 和 CCS Cloud™ 三種 IDE，此處將以 Code Composer Studio (CCS) 做介紹。

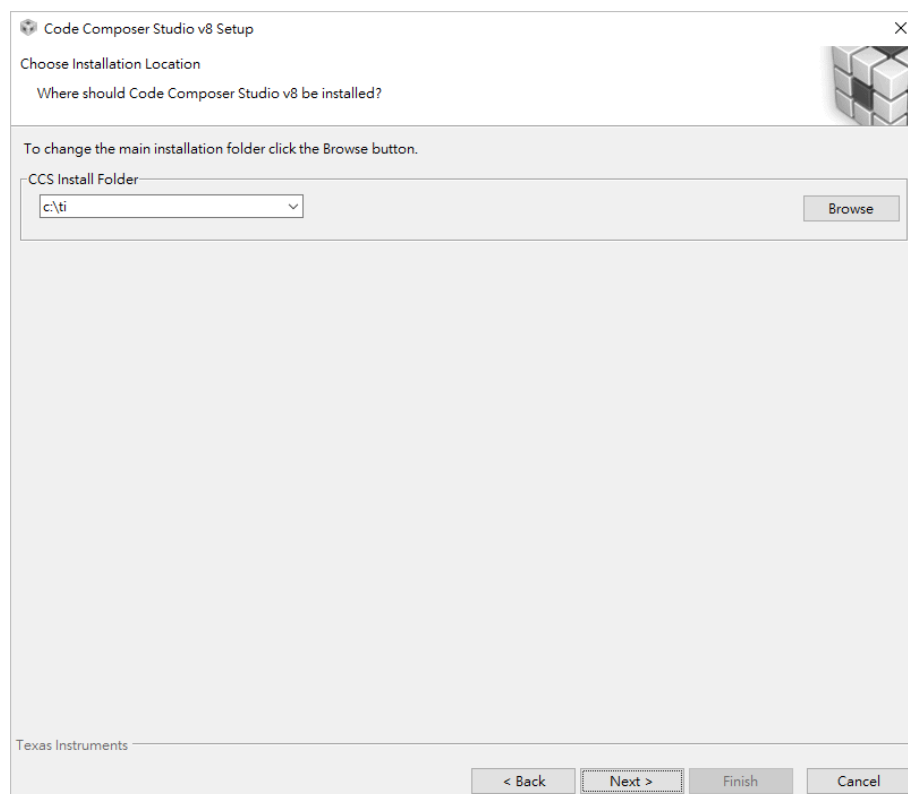
Code Composer Studio 是一個整合式的開發環境，支援 TI 的微控制器和嵌入式處理器產品。它提供了一系列工具來為嵌入式應用開發與除錯，包含經過優化的 C/C++ 編譯器、原始碼編輯器、原始碼環境整合、調試器、分析器和許多其他功能。CCS 提供直觀的 IDE 使用者介面，能夠引導使用者完成應用程式開發流程的每一步。

以下過程介紹了如何安裝和配置 CCS 的正確版本和必要的工具：

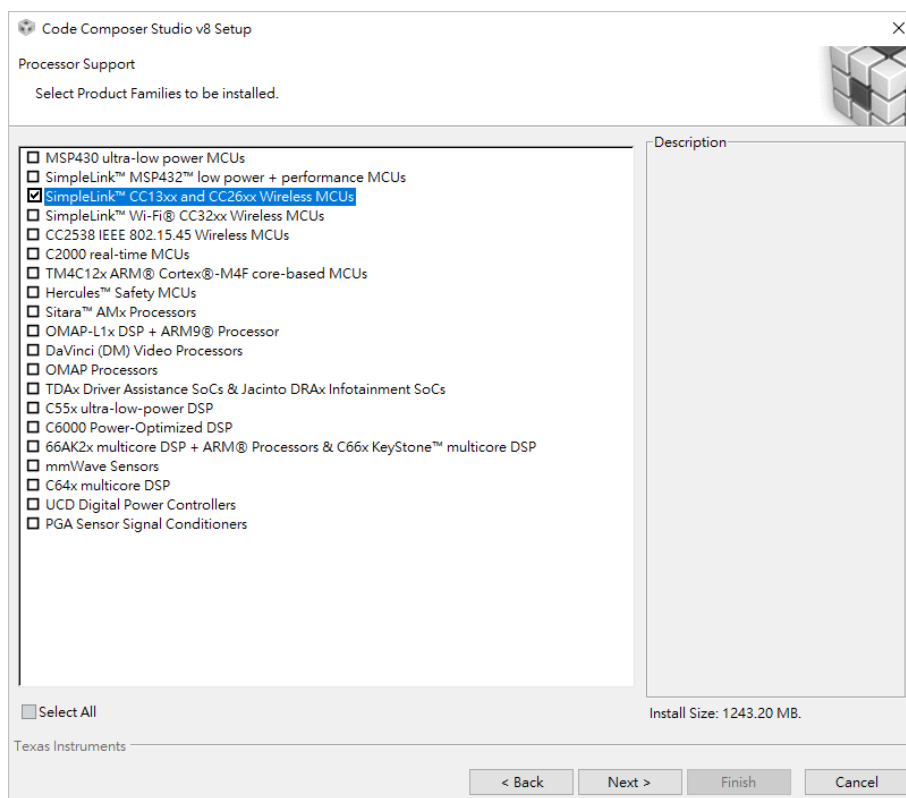
### 5.1 安裝 Code Composer Studio

#### 1. 安裝 Code Composer Studio

- 下載 [Code Composer Studio](#)
- 開始安裝程序並接受授權協議，建議將 CCS 安裝在預設路徑



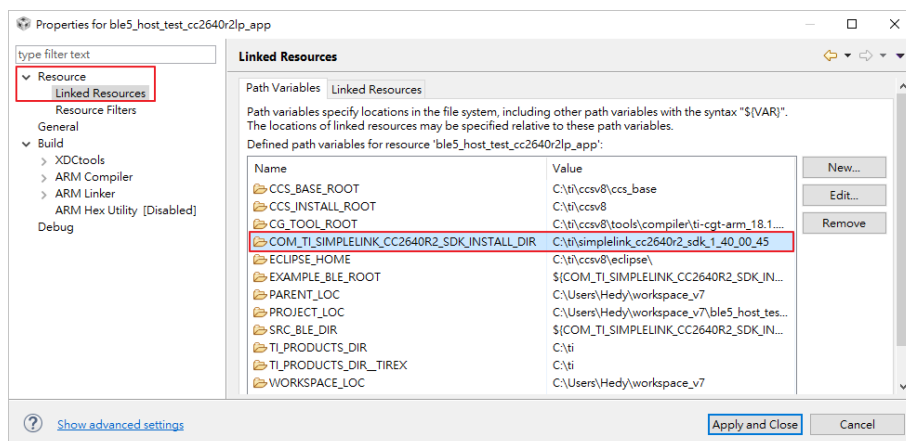
- 在 Processor Support 清單，選擇 SimpleLink CC13xx and CC26xx Wireless MCUs



- 在 Debug Probes 部分，CCS 會安裝對 TI XDS Debug Probe Support 的支援。該選項支援 XDS110 除錯器
- 選擇 Finish 開始安裝程序

## 5.2 搜索 CCS 產品

如果將 Code Composer Studio 安裝在預設目錄之下 (c:/ti)，它會自動搜索 SimpleLink CC2640R2 SDK。一旦 SDK 被 CCS 發現，它會定義一個名為 COM\_TI\_SIMPLELINK\_CC2640R2\_SDK\_INSTALL\_DIR 的建構環境變數給 BLE-Stack 專案使用。



如果 BLE-Stack 專案是由其他地方匯入，而不是 SimpleLink CC2640R2 SDK 在安裝期間所指定的路徑，就必須在匯入程序完成後重新定義 COM\_TI\_SIMPLELINK\_CC2640R2\_SDK\_INSTALL\_DIR 變數。重新定義該變數步驟如下所示：

1. 開啟 CCS 專案屬性 (Project → Properties)
2. 點選 Resource → Linked Resources 並編輯 COM\_TI\_SIMPLELINK\_CC2640R2\_SDK\_INSTALL\_DIR，讓該變數指向匯入的根目錄位置

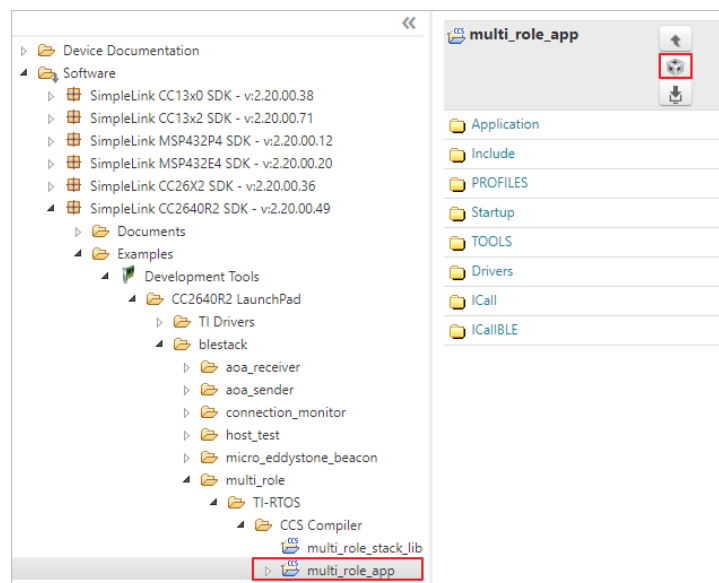
## 5.3 匯入 CCS 專案

本節將引用 multi-role 專案說明如何匯入並編譯一個已存在的專案，開發套件中所有的 BLE-Stack 專案都具有相似的結構。

1. 從開始選單中開啟 CCS IDE
2. 建立一個 workspace (請確認 CCS workspace 的路徑名稱不包含空格)
3. 匯入 CCS 專案

### (1) 開啟 Resource Explorer

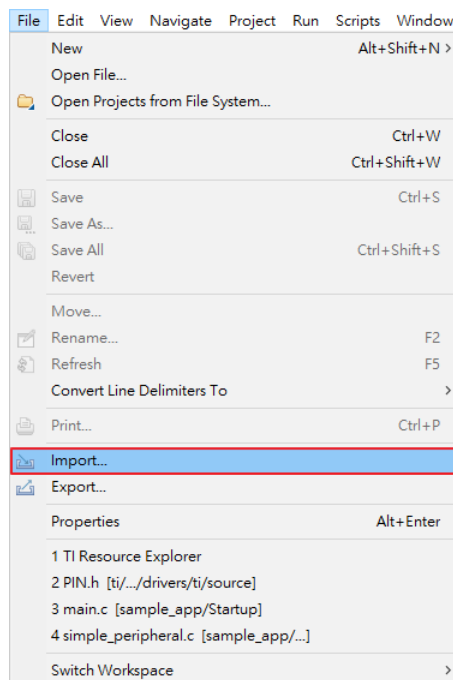
- 點選 View → Resource Explorer，在選擇的範例專案中選取 CCS 資料夾  
Software→SimpleLink CC2640R2 SDK → Examples → CC2640R2 Launchpad → blestack → multi\_role → TI-RTOS → CCS Compiler → multi\_role\_app
- 選取 CCS Logo 按鈕匯入專案到 Project Explorer  
若是 SimpleLink CC2640R2 SDK 本地版本不存在，匯入過程中將會一併安裝，必須接受使用者授權協議才能繼續



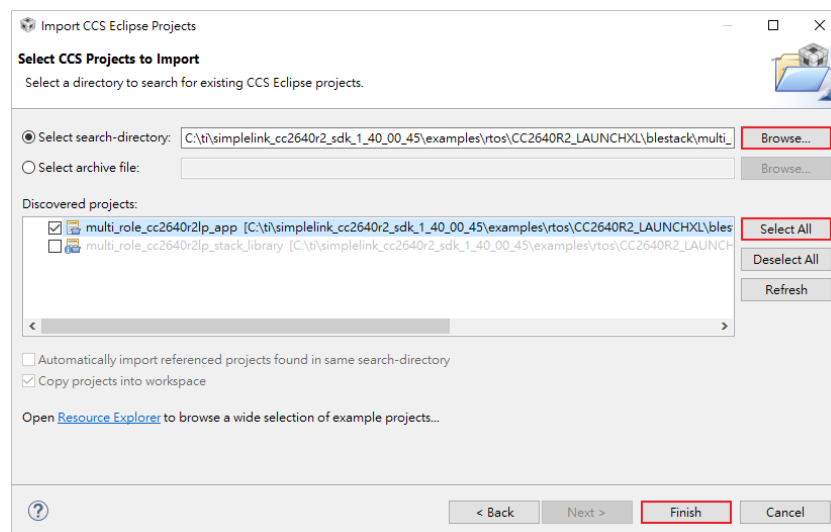


## (2) 選擇範例專案

- 點選 Import: File → Import



- 選擇 CCS Projects : Code Composer Studio → CCS Project
- 瀏覽 multi\_role 檔案位置  
C:\ti\simplelink\_cc2640r2\_sdk\_1\_40\_00\_45\examples\rtos\CC2640R2\_LAUNCHXL\blestack\multi\_role
- Select All 選擇搜索到的專案
- 點選 Finish 開始匯入程序

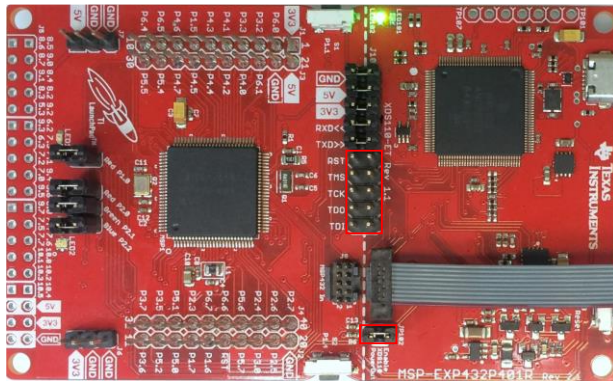


4. 若匯入成功，Project Explorer 將會出現兩個專案，分別為 app、stack 之程式碼

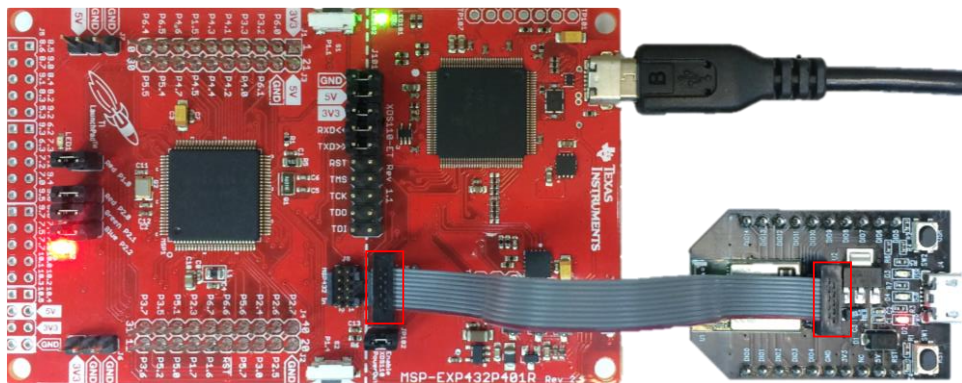
## 六、硬體設置

燒錄流程中需要使用 TI 開發的 MSP-EXP432P401R 開發套件作為 SVB2640 之燒錄板，請遵循以下流程連接無線模組與燒錄板：

1. 請先確認移除下圖中標示區域的 jumpers，目的是為了將開發板上燒錄器 XDS110 與微控制器 MSP-EXP432P401R 斷開。另外，需確認 XDS110 電源 jumper (JP102) 被連接，以提供電源至 SVB2640。



2. 準備 10-pin 之 JTAG 連接線，將其連接至 SVB2640 的 Debug Port (J3)，並將另一端連接至 MSP-EXP432P401R 上的 XDS110 Out Port (J102)，接著插上 Micro USB 連接至電腦。

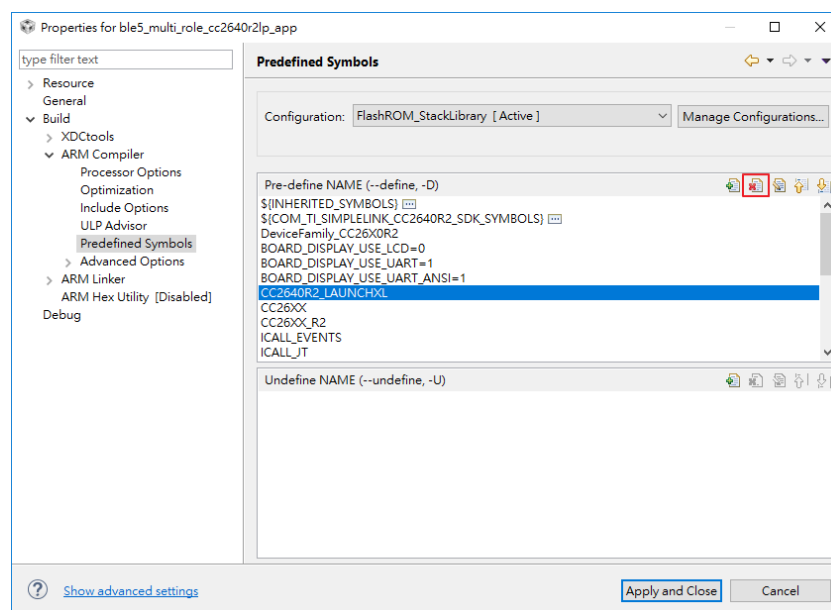


## 七、程式碼編譯與燒錄

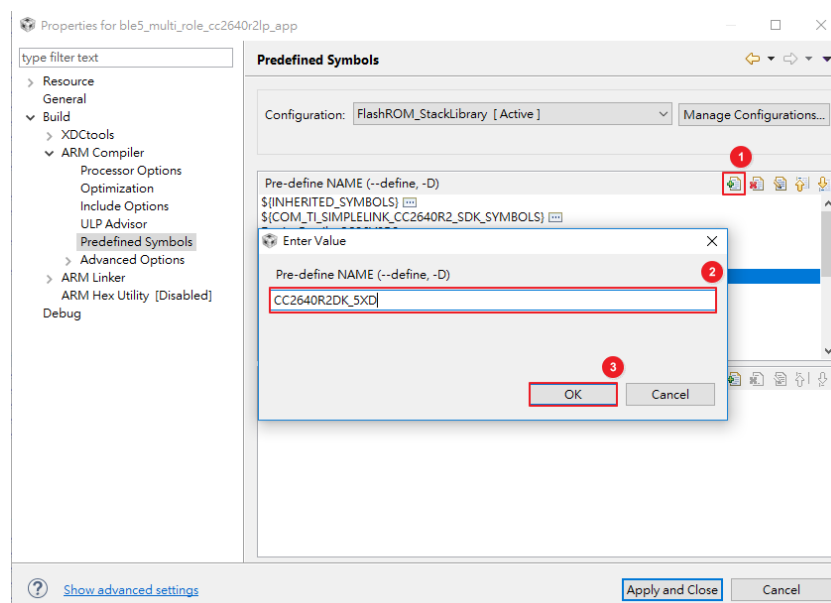
注意：由於 SVB2640 使用之晶片型號為 CC2640R2DK\_5XD，因此需移除專案預編譯旗標 CC2640R2\_LAUNCHXL，並新增預編譯旗標 CC2640R2DK\_5XD。

下列過程將描述如何存取並修改預編譯旗標：

1. 開啟 Project Properties
2. 選取到 Build → ARM Compiler → Predefined Symbols
3. 點選下圖標註的按鈕，刪除預編譯旗標 CC2640R2\_LAUNCHXL



4. 點選下圖標註的按鈕，新增預編譯旗標 CC2640R2DK\_5XD



程式碼開發完畢後，可依照下列流程編譯程式碼，並燒錄至開發板中：

1. 編譯 stack library 專案
  - 將 stack 專案設置為 active
  - 點選 Project → Build All 編譯 stack 專案
2. 編譯 app 專案
  - 將 app 專案設置為 active
  - 點選 Project → Build All 編譯 app 專案
3. 載入整個應用程式
  - Run → Debug

## 八、腳位操作說明

SVB2640 開發板分別提供了紅色、綠色兩顆 LED 燈及一顆按鈕供使用者測試或開發應用時使用，紅色、綠色 LED 分別連接至晶片的 DIO6、DIO7，按鈕則是連接至 DIO14，詳細腳位請參考第三章接腳圖說明。為了讓開發者在韌體中存取 LED 與按鈕做應用開發，另外提供修改後的開發板層級的接腳定義檔 board.h 與 CC2640R2DK\_5XD.h，並以使用範例簡單的說明如何操作 LED 接腳。

下列為接腳的使用範例，此應用範例將每隔五秒輪流切換紅色、綠色 LED 的亮滅：

### 1. 修改 SDK 接腳定義檔

- 下載修改後的 board.h 與 CC2640R2DK\_5XD.h ( [SVB2640 接腳定義](#) )
- 取代下列 SDK 目錄兩個資料夾中兩支同名檔案  
C:\ti\simplelink\_cc2640r2\_sdk\_1\_40\_00\_45\source\ti\blestack\boards\CC2640R2DK\_5XD  
C:\ti\simplelink\_cc2640r2\_sdk\_1\_40\_00\_45\source\ti\ble5stack\boards\CC2640R2DK\_5XD
- 修改接腳定義檔後，在應用程式中可透過 Board\_GLED、Board\_RLED 和 Board\_BUTTON 操作 LED 跟按鈕

### 2. LED 範例應用程式，此處主要說明如何在韌體中操作 LED 接腳，詳細應用程式實作請參考 [simple\\_peripheral.c](#)

- 宣告接腳相關變數，並配置 LED 接腳

```
static PIN_Config ledPinTable[] =
{
    Board_RLED    | PIN_GPIO_OUTPUT_EN | PIN_GPIO_LOW | PIN_PUSHPULL | PIN_DRVSTR_MAX,
    Board_GLED    | PIN_GPIO_OUTPUT_EN | PIN_GPIO_LOW | PIN_PUSHPULL | PIN_DRVSTR_MAX,
    PIN_TERMINATE
};
```

- 設置一週期性 clock，並於 clock 到期時切換 LED 亮滅

```
static void SimpleBLEPeripheral_performPeriodicTask(void)
{
    if (PIN_getOutputValue(Board_GLED) == 0) {
        PIN_setOutputValue(hPins, Board_GLED, Board_LED_ON);
        PIN_setOutputValue(hPins, Board_RLED, Board_LED_OFF);
    } else {
        PIN_setOutputValue(hPins, Board_GLED, Board_LED_OFF);
        PIN_setOutputValue(hPins, Board_RLED, Board_LED_ON);
    }
}
```

## 九、版本歷史

Revision	Date	Description
<b>1.0</b>	2018/08/01	Initial release
<b>1.1</b>	2018/08/08	新增腳位操作說明章節