

## 摘要

此應用說明可協助於從 STMicroelectronics STM32® 平台移轉至德州儀器 MSPM0 MCU 生態系統。本指南主要介紹 MSPM0 開發與工具生態系統、核心架構、周邊設備考量及軟體開發套件。用意是要強調兩個系列之間的差異，並利用 STM32 生態系統的現有知識快速升級至 MSPM0 系列 MCU。

## 目錄

<b>1 MSPM0 產品組合概覽</b>	2
1.1 簡介	2
1.2 STM32 MCU 與 MSPM0 MCU 的產品組合比較	2
<b>2 生態系統與移轉</b>	3
2.1 軟體生態系統比較	3
2.2 硬體生態系統	4
2.3 偵錯工具	5
2.4 移轉程序	6
2.5 移轉和移植範例	6
<b>3 核心架構比較</b>	15
3.1 CPU	15
3.2 嵌入式記憶體比較	15
3.3 開機和重設摘要與比較	17
3.4 時鐘摘要和比較	19
3.5 MSPM0 運作模式摘要和比較	20
3.6 中斷與事件比較	21
3.7 偵錯和編程比較	23
<b>4 數位周邊設備比較</b>	24
4.1 通用 I/O (GPIO、IOMUX)	24
4.2 通用非同步接收器-發射器 (UART)	25
4.3 序列周邊介面 (SPI)	25
4.4 I <sup>2</sup> C	26
4.5 計時器 (TIMGx、TIMAx)	27
4.6 窗型監視計時器 (WWDT)	28
4.7 即時時鐘 (RTC)	28
<b>5 類比周邊設備比較</b>	29
5.1 類比轉數位轉換器 (ADC)	29
5.2 比較器 (COMP)	29
5.3 數位轉類比轉換器 (DAC)	31
5.4 運算放大器 (OPA)	31
5.5 電壓參考 (VREF)	32
<b>6 修訂記錄</b>	32

## 商標

MSP430™, TI E2E™, Code Composer Studio™, LaunchPad™, EnergyTrace™, and BoosterPack™ are trademarks of Texas Instruments.

STM32® is a registered trademark of STMicroelectronics International N.V.

Arm® and Cortex® are registered trademarks of Arm Limited.

所有商標均為其各自所有者的財產。

## 1 MSPM0 產品組合概覽

### 1.1 簡介

MSP430™ MCU 近 30 年來皆是 TI 的傳統微控制器。最新世代 MSPM0 系列。MSPM0 微控制器 (MCU) 是 MSP 以強化型 Arm® Cortex®-M0+ 32 位元核心平台為架構的高度整合超低功耗 32 位元 MCU 系列之一。這些成本最佳化的 MCU 提供高效能類比周邊設備整合，支援廣泛的溫度範圍，並提供小體積套件。TI MSPM0 系列的低功耗 MCU 包含不同程度的類比及數位整合裝置，讓工程師能找到符合專案需求的 MCU。MSPM0 MCU 系列結合 Arm Cortex-M0+ 平台與超低功耗系統架構，讓系統設計人員得以提升效能，同時降低功耗。

MSPM0 MCU 提供了 STM32 MCU 的競爭替代方案。此應用說明可透過比較裝置功能與生態系統，協助從 STM32 MCU 移轉至 MSPM0 MCU。

### 1.2 STM32 MCU 與 MSPM0 MCU 的產品組合比較

表 1-1. TI MSPM0Gx/LX 和 STM32G0/F0 系列的比較

	ST 微型 STM32G0 系列	ST 微型 STM32F0 系列	TI MSPM0 MSPM0Gx 系列	TI MSPM0 MSPM0Lx 系列
核心/頻率	CM0+ / 64 MHz	CM0 / 48 MHz	CM0+ / 80 MHz	CM0+ / 32 MHz
電源電壓	1.7 V 至 3.6 V	2 V 至 3.6 V	1.62 V 至 3.6 V	1.62 V 至 3.6 V
溫度	-40°C 至 125°C	-40°C 至 105°C	-40°C 至 125°C	-40°C 至 125°C
記憶體	512KB 至 16KB	256KB 至 16KB	128KB 至 32KB	64KB 至 8KB
RAM	最高 144KB	最高 32KB	最高 32KB	最高 4KB
GPIO (最大值)	90	88	60	28
類比	1x 2.5-Msps 12 位元 ADC 1x 12 位元 DAC 3x 比較器	1x 1-Msps 12 位元 ADC 1x 12 位元 DAC 2x 比較器	2x 4-Msps 12 位元 ADC 1x 12 位元 DAC 3x 高速比較器 2x 運算放大器	1x 1-Msps 12 位元 ADC 1x 高速比較器 2x 運算放大器
通訊 (最大值)	3x SPI 3x I <sup>2</sup> C Fast+ 6x UART (LIN) 2x CAN-FD 1x USB	2x SPI 2x I <sup>2</sup> C Fast+ 8x UART (LIN) 1x CAN	2x SPI 2x I <sup>2</sup> C Fast+ 4x UART (LIN) 1x CAN-FD	1x SPI 2x I <sup>2</sup> C Fast+ 2x UART (LIN)
計時器	8	4	7	4
進階計時器	是 (1)	是 (1)	是 (3x)	否
硬體加速器	不適用	不適用	選擇性	不適用
安全性	CRC、TRNG、AES256	CRC	CRC、TRNG、AES256	CRC
低功耗	主動式：100 µA/MHz 待機 (RTC)：1.5 µA	主動式：281 µA/MHz 待機 (RTC)：2.5 µA	主動式：85 µA/MHz 待機 (RTC)：1.5 µA	主動式：85 µA/MHz 待機：1.5 µA

## 2 生態系統與移轉

廣泛硬體與軟體生態系統為 MSPM0 MCU 提供參考設計和程式碼範例的支援，有助於快速開始進行設計。MSPM0 MCU 還擁有線上資源、MSP Academy 培訓和 [TI E2E™ 支援論壇的線上支援](#)。

### 2.1 軟體生態系統比較

表 2-1. 適用於 MSPM0 的 STM32 軟體工具等效產品

	STM32	MSPM0
IDE	CubeIDE	<a href="#">Code Composer Studio™ IDE (CCS)</a>
軟體配置	CubeMX	<a href="#">SysConfig</a>
獨立編程	CubeProgrammer	<a href="#">UniFlash</a>
顯示/示範 GUI 編輯器	CubeMonitor	<a href="#">GuiComposer</a>

#### 2.1.1 MSPM0 軟體開發套件 (MSPM0 SDK)

MSPM0 SDK 交付軟體 API、範例、文件和程式庫，協助工程師快速開發德州儀器 MSPM0+ 微控制器裝置上的應用程式。提供的範例示範了如何在每個受支援的裝置上使用每個功能區域，並可做為您自己專案的起始點。此外，MSPM0 SDK 也包含互動式 MSP Academy 訓練課程，提供引導式學習路徑。

範例資料夾分為 RTOS 和非 RTOS 子資料夾 (目前僅支援非 RTOS)。這些資料夾內含各 LaunchPad™ 開發套件的範例，並依照如低階 DriverLib 範例、高階 TI 驅動程式範例，以及如 GUI Composer、LIN、IQMath 等中介軟體範例進行分類。如需詳細資訊，請參閱 [MSPM0 SDK 使用指南](#)。

#### 2.1.2 CubeIDE 與 Code Composer Studio IDE (CCS)

[Code Composer Studio IDE \(CCS\)](#) 是 TI 與 STM32 CubeIDE 等效的產品。CCS 為免費的 Eclipse 架構 IDE，其支援 TI 的微控制器 (MCU) 及嵌入式處理器產品組合。CCS 包含一套用於開發和偵錯嵌入式應用程式的工具，其中包括最佳化 C/C++ 編譯器、原始碼編輯器、專案建置環境、偵錯器、性能評測工具以及其他多種功能。CCS 可用作桌上型或雲端型 IDE。

CCS 在整合式 TI Resource Explorer 中整合了 SysConfig 的 MSPM0 裝置配置和自動產生程式碼功能，以及 MSPM0 程式碼範例和學院訓練。CCS 提供多合一開發工具體驗。

除了 CCS 外，下表所列的業界標準 IDE 也支援 MSPM0 裝置。

表 2-2. MSPM0 支援的 IDE

IDE	MSPM0
CCS	✓
IAR	✓
Keil	✓

#### 2.1.3 CubeMX 與 SysConfig

SysConfig 是一款直覺式且全方位的圖形式公用程式集合，可用於配置針腳、周邊設備、無線電、子系統和其他零組件。是 TI 與 STM32 CubeMX 等效的產品。SysConfig 可協助以視覺化的方式管理、找出與解決衝突，讓您在撥出更多時間來建立與眾不同的應用。此工具的輸出包含 C 標頭和程式碼檔案，可與 MSPM0 SDK 範例一起使用，也可用於配置自訂軟體。SysConfig 已整合至 CCS，但也可當成獨立程式使用。

如需詳細資訊，請參閱 [MSPM0 SysConfig 指南](#)。

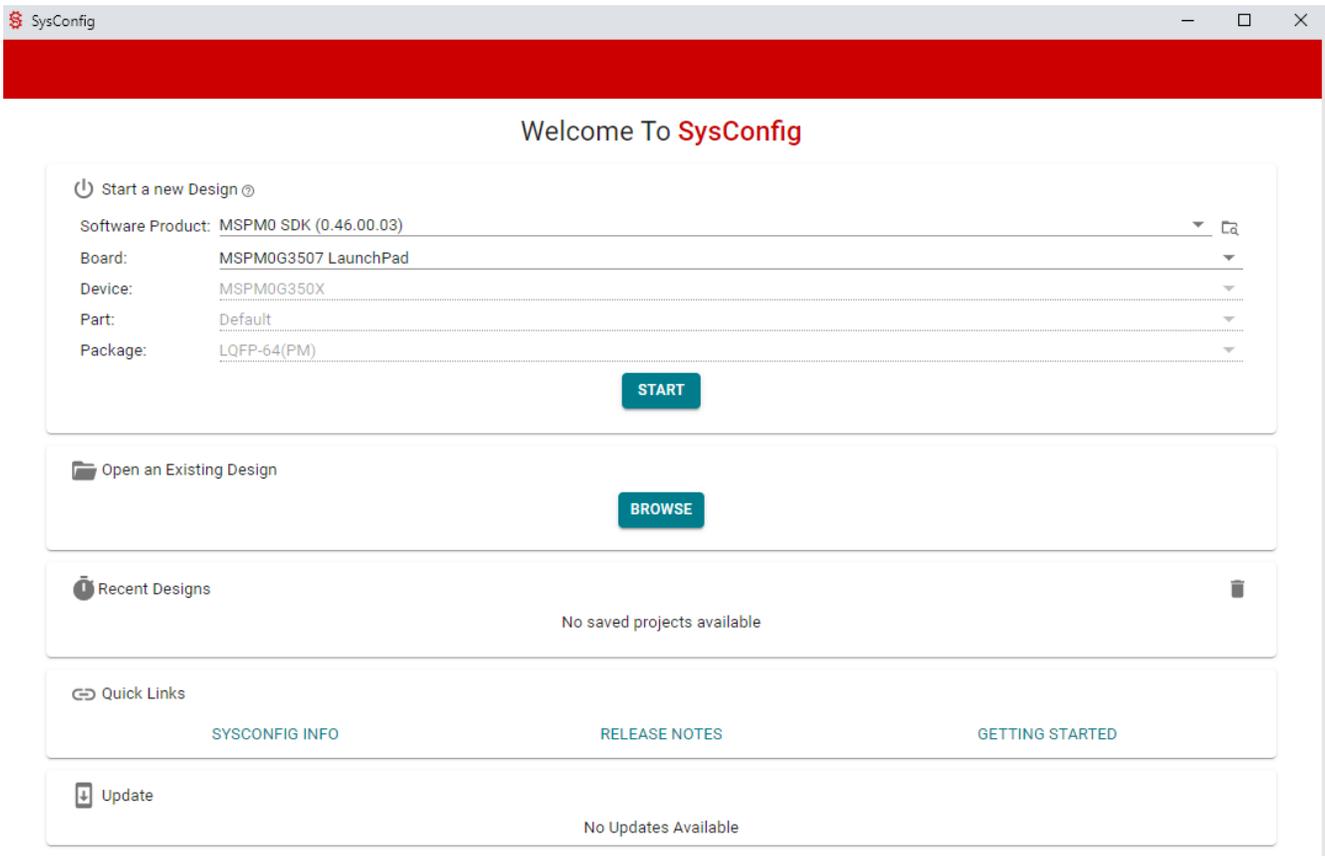


图 2-1. MSPM0 SysConfig

## 2.2 硬體生態系統

LaunchPad 開發套件是 MSPM0 的唯一評估模組。LaunchPad 套件是易於使用的 EVM，內含開始在 MSPM0 上進行開發所需的一切。其中包括使用 EnergyTrace™ 技術進行程式設計、偵錯和測量功耗的板載偵錯探針。MSPM0 LaunchPad 套件也具備板載按鈕、LED 和溫度感測器以及其它電路。40 針腳 BoosterPack™ 外掛模組接頭可簡化快速原型設計，同時支援多種可用的 BoosterPack 外掛模組。您可快速新增無線連線、圖形顯示器、環境感測等功能。

- [LP-MSPM0G3507 LaunchPad 開發套件](#)
- [LP-MSPM0L1306 LaunchPad 開發套件](#)

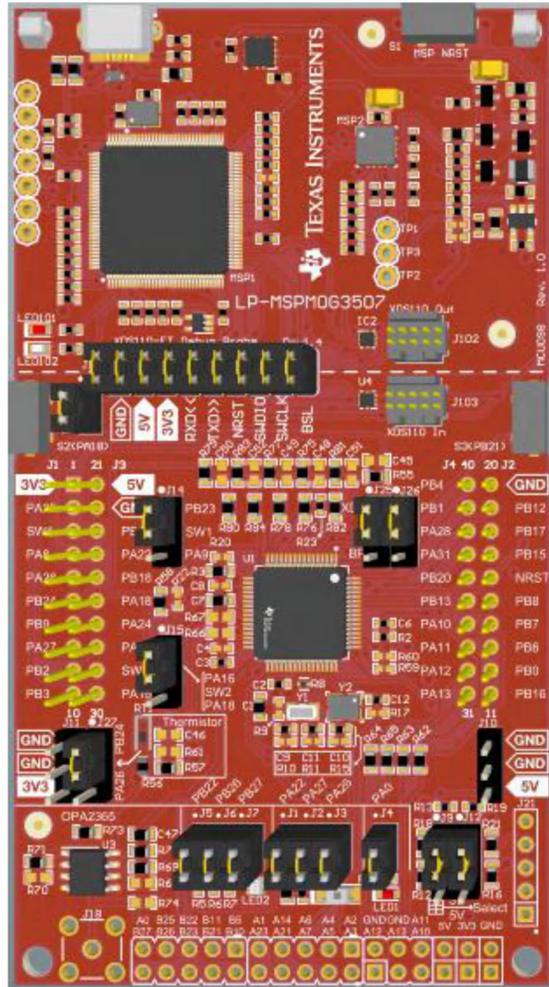


图 2-2. LP-MSPM0G3507 LaunchPad 開發套件

### 2.3 偵錯工具

偵錯子系統 (DEBUGSS) 可將序列線偵錯 (SWD) 雙線實體介面連接至裝置內的多個偵錯功能。MSPM0 裝置支援對處理器執行、裝置狀態和電源狀態 (使用 EnergyTrace 技術) 進行偵錯。图 2-3 顯示偵錯器的連線。

MSPM0 支援適用於標準序列線偵錯的 XDS110 和 J-Link 偵錯器。

德州儀器 XDS110 是針對 TI 嵌入式處理器而設計的。XDS110 透過 TI 20 針腳連接器 (具有適用於 TI 14 針腳和 Arm 10 針腳和 Arm 20 針腳的多重轉接器) 連接到目標電路板, 並透過 USB2.0 2.0 高速 (480 Mbps) 連接到主機 PC。它支援單一 Pod 中更廣泛的標準 (IEEE1149.1、IEEE1149.7、SWD)。所有 XDS 偵錯探針都支援所有配備嵌入式追蹤緩衝器 (ETB) 的 Arm 與 DSP 處理器的核心與系統追蹤。如需詳細資訊, 請參閱 [XDS110 偵錯探針](#)。

J-Link 偵錯探針是偵錯與快閃記憶體編程體驗最佳化的最受歡迎選擇。受益於破紀錄的快閃載入程式, 具備高達 3 MiB/s RAM 下載速度, 以及在 MCU 快閃記憶體中設定數量無限的斷點的能力。J-Link 也支援廣泛的 CPU 及架構, 包含 CortexM0+。如需詳細資訊, 請造訪 [Segger J-Link 偵錯探針頁面](#)。

图 2-3 顯示 XDS110 探針至 MSPM0 目標的主要功能區和介面的高階圖。

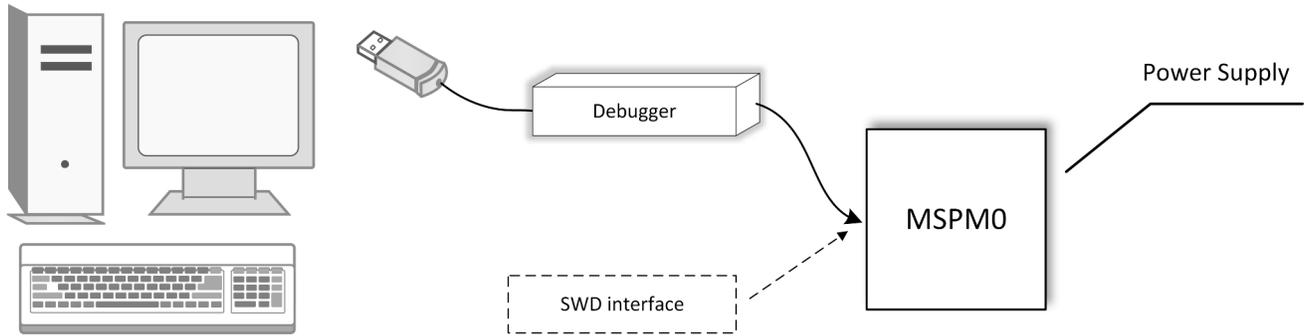


图 2-3. MSPM0 偵錯

## 2.4 移轉程序

移轉的第一步是檢閱產品組合，並選擇最佳的 MSPM0 MCU。選擇 MSPM0 MCU 後，選擇開發套件。開發套件包含可供購買的 LaunchPad 套件，以及目標插座板的设计檔案。TI 也提供免費的 MSPM0 軟體開發套件 (SDK)，此套件可在 TI 資源總管中做為 [Code Composer Studio IDE](#) 桌面和雲端版本的零組件使用。請使用此應用說明的周邊設備章節，以協助將軟體從 STM32 移植到 MSPM0。最後，在軟體移植後，請下載並利用我們的偵錯工具偵錯應用程式。

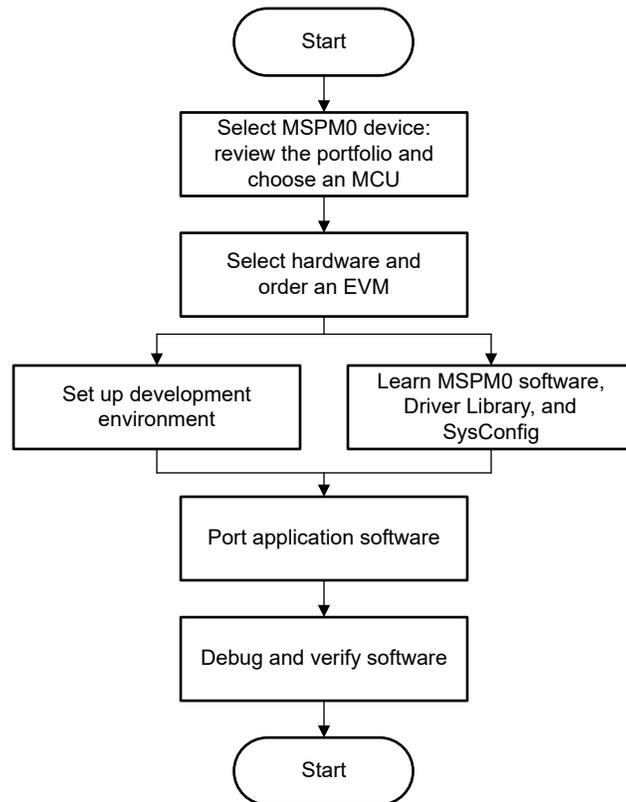


图 2-4. MSPM0 移轉流程圖

## 2.5 移轉和移植範例

為了更熟悉 TI 生態系統並說明如何以最佳方式開始使用 MSPM0，本節內容會說明基本應用程式的逐步移轉程序。

為了示範從 STM32 移植到 MSPM0 的程序，本說明內容包含了使用現有 ST UART 範例作為起點，將基本低功耗 UART 監控應用程式從 STM32G0x 移植到 MSPM0 裝置的步驟。

### 步驟 1. 選擇正確的 MSPM0 MCU

移轉的第一步是為應用程式選擇正確的 MSPM0 裝置。為達此目的，本指南的產品組合章節可用於選擇 MSPM0 系列。使用 [產品選擇工具](#) 縮小到特定裝置。STM32G0 和 MSPM0 共享 M0+ 內核，但也必須考慮記憶體大小、電源和關鍵周邊設備等功能。MSPM0 也提供許多針腳對針腳可調整選項，讓您能夠輕鬆調整為更大或更小的記憶體裝置、而不需變更系統中的其他任何項目。

為達到本範例之目的，我們選擇 MSPM0G3507 做為最適合其應用的產品。

## 步驟 2. 選擇硬體並訂購 EVM

使用評估模組 (EVM) 可加快移轉程序。對於 MSPM0 MCU，LaunchPad 套件是可供開始使用的最簡單硬體。LaunchPad 套件隨附內建程式設計工具，其設計功用可加快開發速度，因此使用起來十分方便。

MSPM0G3507 具有可用於移植軟體的 LaunchPad 開發套件 ([LP-MSPM0G3507](#))。

## 步驟 3. 設定軟體 IDE 和 SDK

在移植軟體之前，必須選擇並設定軟體開發環境。[節 2.1](#) 顯示 MSPM0 支援的所有 IDE。移轉和移植過程與選擇的任何 IDE 相似。應該安裝最新版的 [MSPM0 SDK](#)。

在此範例中，TI 的 CCS 是選擇的 IDE。

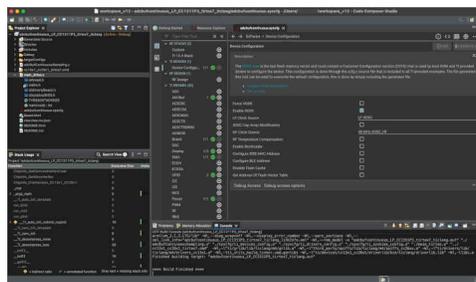


圖 2-5. Code Composer Studio IDE

## 步驟 4. 軟體移植

環境準備好後，請開始使用 MSPM0 SDK。如前所述，MSPM0 SDK 與 STM32Cube 軟體套件相似。MSPM0 SDK 為軟體開發提供不同階層。MSPM0 TI 驅動程式的執行階層與 STM32Cube HAL 相似，而 MSPM0 DriverLib 與 STM32Cube 低階驅動程式不相上下。大多數 MSPM0 使用者都會發現，DriverLib 階層的軟體最適合他們的應用程式，因此大多數 MSPM0 軟體範例也都是以 DriverLib 為基礎。此範例使用 DriverLib。

移植專案時，其中一個選項是嘗試以同等的 MSPM0 DriverLib API 取代程式碼的每個區段，但這通常不是最簡單的路徑。一般而言，最好先了解移植的應用程式碼。然後從最接近的 MSPM0 範例專案開始修改專案，以符合原始程式碼功能。此程序使用 STM32CubeG0 的低功耗 UART 範例，如下所示。對於使用許多周邊設備的複雜專案，通常會對每個周邊設備重複一遍此程序。

## 步驟 4a：認識應用程式

以下描述摘自名為 'LPUART\_WakeUpFromStop\_Init' 的 STM32CubeG0 的範例專案。

### @par 範例說明

GPIO 和 LPUART 周邊設備的配置允許在 LPUART RX 針腳上收到的字元從低功耗模式喚醒 MCU。此範例是以 LPUART LL API 為基礎。周邊設備初始化使用 LL 初始化功能來示範 LL init 用法。

LPUART 周邊設備配置為異步模式 (9600 傳輸速率、8 個數據位元、1 個起始位元、1 個停止位元、無同位檢查)。  
未使用硬體流量控制。

LPUART 時鐘以 HSI 為基礎。

### 執行範例：

從重設和系統配置啟動後，LED3 會在 3 秒內快速閃爍，然後 MCU 會進入「停止 0」模式 (LED3 關閉)。當 LPUART 在「停止 0」模式期後從 PC Com 連接埠 (例如：使用 HyperTerminal) 接收第一個字元時，MCU 會從「停止 0」模式喚醒。

已核取接收的字元值：

- 在特定的值 ('S' 或 's') 上，LED3 開啟，程序結束。
- 如果與 'S' 或 's' 不同，程式會在 3 秒內快速閃爍 LED3，並再次進入「停止 0」模式，等待下一個字元喚醒。

第一步是了解 MCU 的主要設定。這通常是時鐘速度和電源政策。此範例未指定一般時鐘頻率，因為唯一重要的設定是 UART 在低功率 Stop0 模式下運作。此設定表示低功耗 UART 時鐘以「HSI」或高速內部振盪器為基礎，意思是未使用外部晶體。UART 以 9600 傳輸速率、8 個數據位元、1 個啟動和停止位元、無同位檢查來執行。未使用硬體流程控制。應用端檢查要接收的「S」或「s」，並閃爍 LED。

## 步驟 4b：尋找最接近的 MSPM0 範例

下一步是了解 STM32G0 和 MSPM0 的 UART 模組之間的任何差異，然後在 MSPM0 SDK 中找出最接近的範例。這可以透過參閱節 4 中的 UART 區段輕鬆完成。本節強調 UART 模組之間的差異，以及連結至 UART 相關的 MSPM0 SDK 程式碼範例。此範例的 SDK 中最接近的範例可能是 [uart\\_echo\\_interrupts\\_standby](#)，其中「當裝置處於待機模式時，UART RX/TX 回波使用中斷」。

此 MSPM0 範例類似，但與正在移植的範例不同。此範例將進入待機模式，這是比停止模式更低的功率模式。必須檢查 UART 通訊設定以及使用的 GPIO。最後，必須增加特定字元的監視應用層。

## 步驟 4c：匯入並修改範例

找到類似的範例後，開啟 CCS 並前往專案 > 匯入 CCS 專案... 匯入程式碼範例，並導覽至 MSPM0 SDK 範例資料夾。匯入範例。這是匯入的 [uart\\_echo\\_interrupts\\_standby](#) 範例。這是 SysConfig 專案，所以主要 C 檔案十分簡單。它會先呼叫 SysConfig driverlib 初始化，是由 SysConfig 自動產生的函數，用於配置裝置。接著會啟用 UART 中斷。最後會進入睡眠狀態等待任何 UART 交易。如果收到 UART 交易，就會立即回顯並喚醒資料。

```

25 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
26 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
27 * OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
28 * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
29 * OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
30 * EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
31 */
32
33 #include "ti_msp_dl_config.h"
34
35 uint8_t data = 0;
36
37 int main(void)
38 {
39     SYS_CFG_DL_init();
40
41     NVIC_ClearPendingIRQ(UART_0_INST_INT_IRQN);
42     NVIC_EnableIRQ(UART_0_INST_INT_IRQN);
43     DL_SYSCFG_enableSleepOnExit();
44
45     while (1) {
46         __WFI();
47     }
48 }
49
50 void UART_0_INST_IRQHandler(void)
51 {
52     switch (DL_UART_Main_getPendingInterrupt(UART_0_INST)) {
53         case DL_UART_MAIN_IIDX_RX:
54             data = DL_UART_Main_receiveData(UART_0_INST);
55             DL_UART_Main_transmitData(UART_0_INST, data);
56             break;
57         default:
58             break;
59     }
60 }
61

```

圖 2-6. uart\_echo\_interrupts\_standby 範例

若要查看 SysConfig 配置，請開啟 .syscfg 檔案，預設情況下在 SYSCFG 標籤處開啟該檔案。如需有關使用 SysConfig 的詳細指引，請參閱 MSPM0 SDK 中的 [SysConfig 指南](#)。

首先要注意的是電源政策。此 MSPM0 範例使用 Standby0 模式，但目標是使用 Stop0 模式。按下拉清單，選擇正確的低功耗模式。所有時鐘及振盪器也可在此標籤上配置，但目前仍可正常使用。

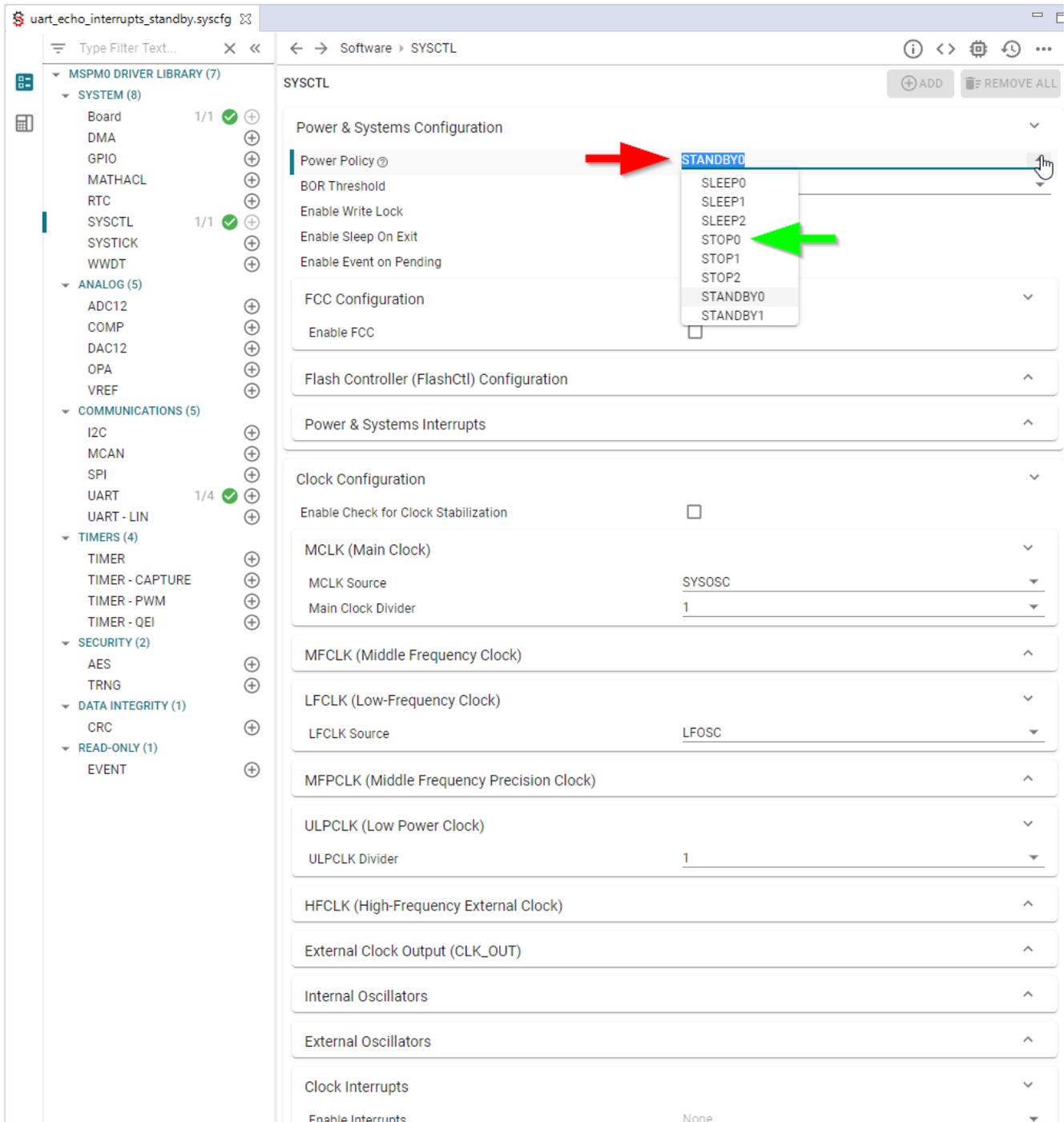


图 2-7. 功率模式配置

接下來，檢查 UART 標籤上的 UART 通訊設定 (請參閱 图 2-8)。在這種情況下，傳輸速率已設定為 9600，其餘的通訊設定是正確的。接收中斷已啟用，並在主程式中使用。另外請按一下右上方的晶片圖示，並檢查 UART 的重點標示針腳，以檢查 UART 模組和使用的針腳。這裡沒有任何東西需要改變，因為這些都已經連接到 MSPM0G3507 LaunchPad 套件的後通道 UART。

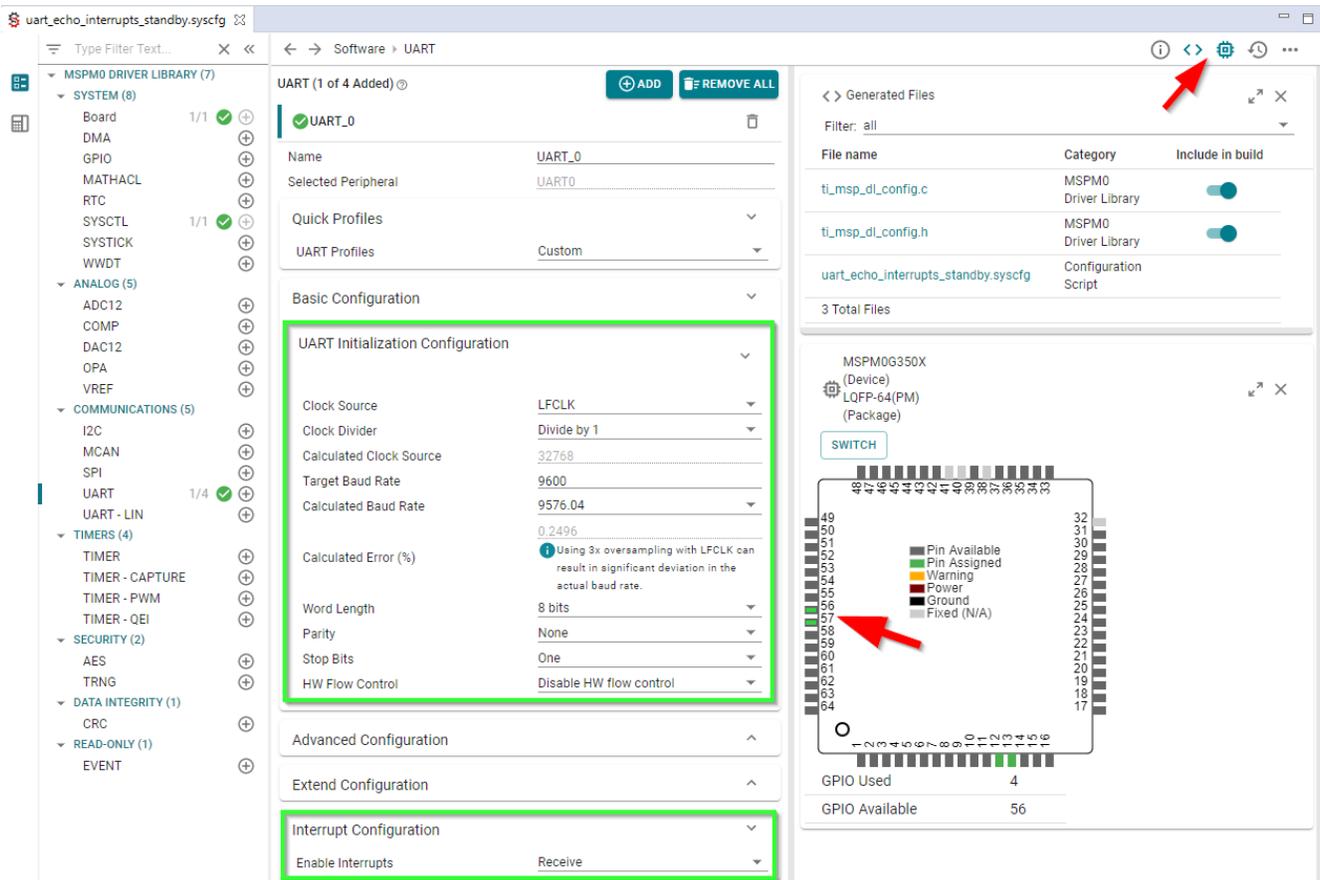
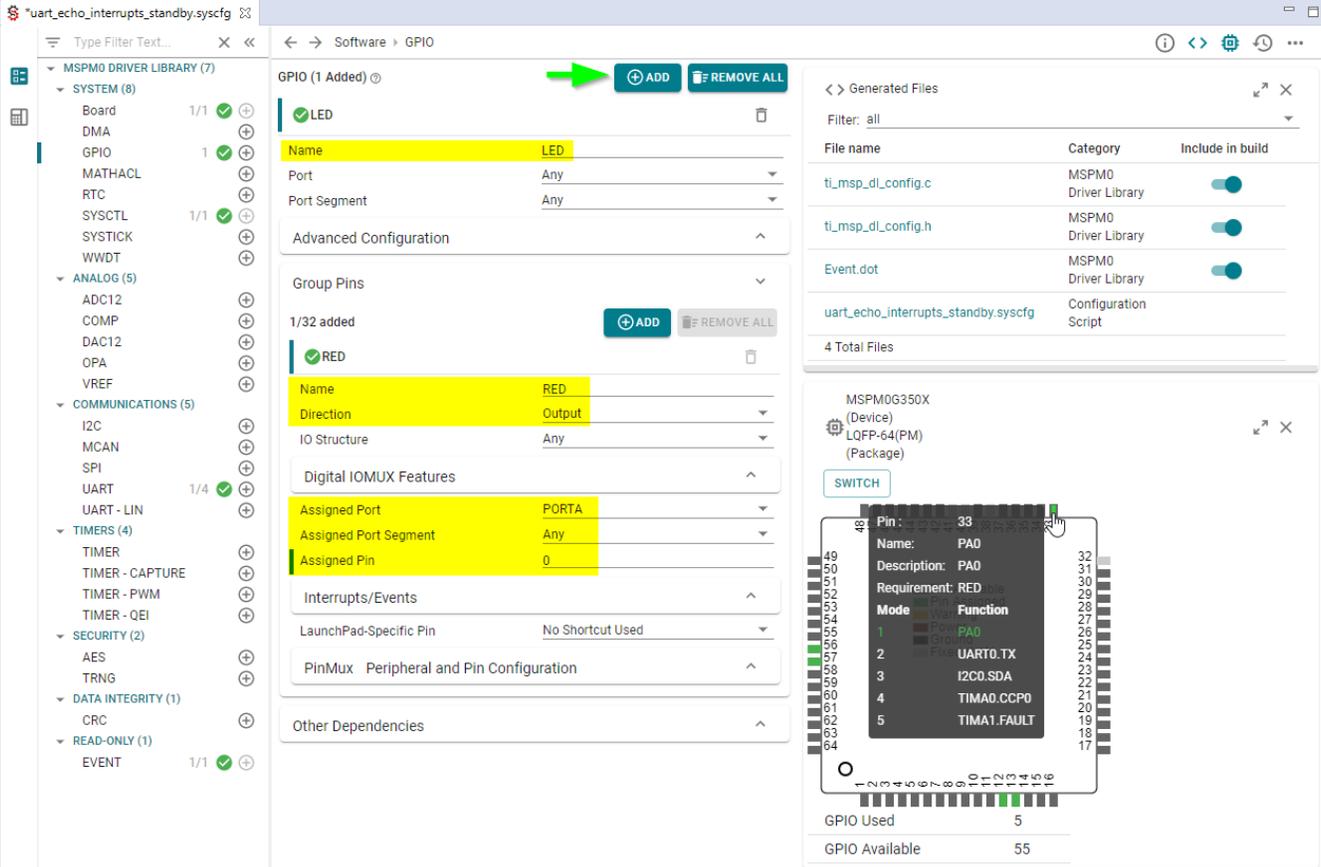


图 2-8. UART 配置

此範例目前沒有為驅動 LED 配置 GPIO，但可以輕鬆增加配置 (請參閱 图 2-9)。可以透過使用頁面頂部的 **+ADD** 按鈕新增 GPIO。GPIO 連接埠和針腳可各自命名，在本例中各命名為「LED」和「D」。此 GPIO 設定為輸出，然後放置於連接埠 A 針腳 0 (PA0) 上。在 LaunchPad 套件上，此 GPIO 連結至簡單的紅色 LED。



The screenshot displays the SysConfig tool interface for configuring GPIO on an MSPM0G350X device. The left sidebar shows a tree view of components under 'MSPM0 DRIVER LIBRARY (7)', including SYSTEM (8), ANALOG (5), COMMUNICATIONS (5), TIMERS (4), SECURITY (2), DATA INTEGRITY (1), and READ-ONLY (1). The central pane shows the 'GPIO (1 Added)' configuration for a component named 'LED'. The configuration includes: Name: LED, Port: Any, Port Segment: Any, Direction: Output, Assigned Port: PORTA, and Assigned Pin: 0. A green arrow points to the 'ADD' button. The right pane shows 'Generated Files' with a table of files and their categories, and a 'MSPM0G350X (Device)' section with a pin configuration diagram. The diagram shows pin 33 (PA0) is used for the LED. Below the diagram, it indicates 'GPIO Used: 5' and 'GPIO Available: 55'.

File name	Category	Include in build
ti_msp_dl_config.c	MSPM0 Driver Library	<input checked="" type="checkbox"/>
ti_msp_dl_config.h	MSPM0 Driver Library	<input checked="" type="checkbox"/>
Event.dot	MSPM0 Driver Library	<input checked="" type="checkbox"/>
uart_echo_interrupts_standby.syscfg	Configuration Script	<input type="checkbox"/>

圖 2-9. GPIO 配置

儲存並重建專案時，SysConfig 會更新範例中的 `ti_msp_dl_config.c` 和 `ti_msp_dl_config.h` 檔案。此時，範例硬體配置已被修改以符合要移植的原始軟體的全部功能。唯一剩餘工作是應用層軟體檢查傳入的 UART 位元組並切換 LED。這是透過將少量程式碼移入主 C 檔案來完成的。

```

uart_echo_interrupts_standby.c  uart_echo_interrupts_standby.syscfg
31 ^/
32
33 #include "ti_msp_dl_config.h"
34
35 uint8_t data = 0;
36
37 int main(void)
38 {
39     SYSCFG_DL_init();
40
41     DL_GPIO_clearPins(LED_PORT, LED_RED_PIN);
42
43     NVIC_ClearPendingIRQ(UART_0_INST_INT_IRQN);
44     NVIC_EnableIRQ(UART_0_INST_INT_IRQN);
45 //     DL_SYSCCTL_enableSleepOnExit();
46     DL_SYSCCTL_disableSleepOnExit();
47
48     while (1) {
49         __WFI();
50
51         if((data == 'S') || (data == 's')){
52             DL_GPIO_setPins(LED_PORT, LED_RED_PIN);
53
54         }else{
55             DL_GPIO_clearPins(LED_PORT, LED_RED_PIN);
56         }
57     }
58 }
59
60 void UART_0_INST_IRQHandler(void)
61 {
62     switch (DL_UART_Main_getPendingInterrupt(UART_0_INST)) {
63         case DL_UART_MAIN_IIDX_RX:
64             data = DL_UART_Main_receiveData(UART_0_INST);
65             DL_UART_Main_transmitData(UART_0_INST, data);
66             break;
67         default:
68             break;
69     }
70 }
71
72

```

图 2-10. 應用程式碼的變更

對應用程式碼進行兩次變更。首先，使用 `DL_SYSCCTL_disableSleepOnExit()` 以便在每個 UART RX 上短暫喚醒 MSPM0。接著，新增 UART RX 資料的簡易檢查，若收到「S」或「s」，代表紅色 LED 已開啟。其他任何功能都會關閉。

### 步驟 5：偵錯並驗證

以下圖形是從邏輯分析器所擷取，該分析器是以 9600 傳輸速率顯示 UART 通訊，且紅色 LED 正確地開啟和關閉。程式碼回應每個 UART 字元，但只有在收到正確的字元時才會開啟 LED。

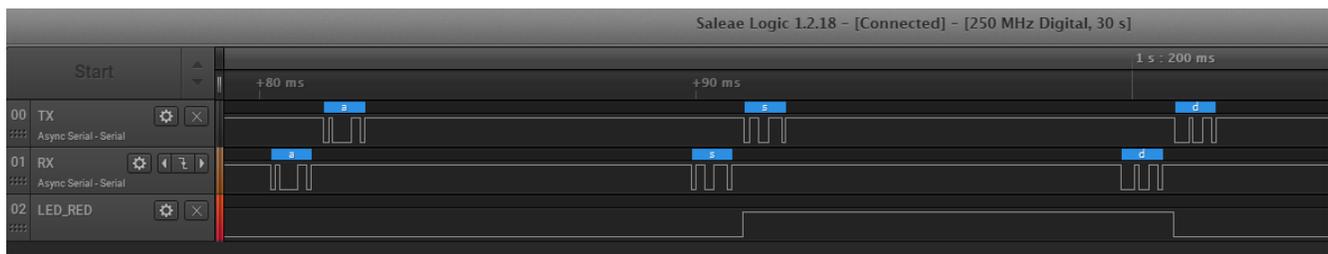


图 2-11.

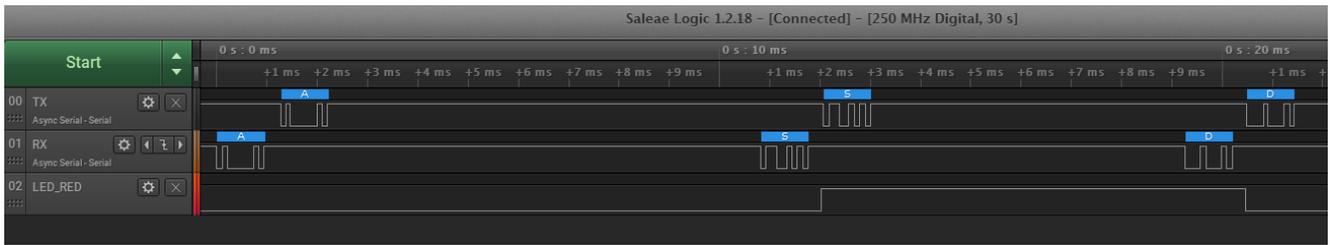


圖 2-12.

軟體已成功移植！如果這只是許多周邊設備的第一個，請繼續重複此程序，並使用 SysConfig 組合每個區塊。

## 3 核心架構比較

### 3.1 CPU

STM32G0 和 MSPM0 系列的零件都是以 Arm Cortex® M0+ CPU 核心架構和指令集為基礎。下表概括介紹了 MSPM0G 和 MSPM0L 系列 CPU 與 STM32G0 相比的一般特性。中斷和例外提供了中斷和例外的比較，以及它們如何映射到每個裝置 M0 結構中包含的內嵌向量中斷控制器 (NVIC) 周邊設備中。

表 3-1. CPU 功能集的比較

特點	STM32G0	MSPM0G	MSPM0L
架構	Arm Cortex-M0+	Arm Cortex-M0+	Arm Cortex-M0+
最大 MCLK	64 MHz	80 MHz	32 MHz
CPU 指令快取	2x64 位元行 (16 位元組)	4x64 位元行 (32 位元組)	2x64 位元行 (16 位元組)
處理器追蹤能力	否	是，整合式微追蹤緩衝器	否
記憶體保護單元 (MPU)	是	是	否
系統計時器 (SYSTICK)	是	是 - 24 位元	是 - 24 位元
NVM 預取	是	是	是
硬體乘法	是	是	是
硬體斷點/監視點	4 / 2	4 / 2	4 / 2
開機例行儲存	快閃 (系統記憶體)	ROM	ROM
Bootstrap 載入程式儲存	快閃 (系統記憶體)	ROM	ROM
開機載入器介面支援 <sup>(1) (2)</sup>	UART、I2C、SPI、USB、FDCAN	UART、I2C、使用者可延長	UART、I2C、使用者可延長
DMA	是	是	是

(1) 有關可用性，請參閱裝置專屬產品規格表。

(2) 其他介面則可在後續裝置版本中使用。

### 3.2 嵌入式記憶體比較

#### 3.2.1 快閃記憶體功能

MSPM0 和 STM32G0 系列的 MCU 具有非揮發性記憶體，用於儲存可執行程式碼和應用程式資料。

表 3-2. 快閃記憶體功能的比較

特色	STM32G0	MSPM0
快閃記憶體	STM32G0B1xx、G0C1xx (高達 512KB) STM32G071xx、G081xx (高達 128KB) STM32G031xx、G041xx、G051xx、G061xx (高達 64KB)	MSPM0Gx 範圍 128KB 至 32KB MSPM0Lxx 範圍 64KB 至 8KB
記憶體組織	1 組裝置高達 128KB 2 組裝置具備 >128KB	1 組裝置高達 256KB 2 組裝置具備 >256KB
快閃記憶體等待狀態	0 (HCLK ≤ 24 MHz) 1 (HCLK ≤ 48 MHz) 2 (HCLK ≤ 64 MHz)	0 (MCLK、CPUCLK ≤ 24 MHz) 1 (MCLK、CPUCLK ≤ 48 MHz) 2 (MCLK、CPUCLK ≤ 80 MHz)
快閃字大小	64 位元加 8 ECC 位元	相同
編程解析度	單字大小	單字、32 位元、16 位元或 8 位元 (位元組)
多字編程	32 個字 (256 位元組)	2、4 或 8 個字 (最多 64 個位元組)
清除	頁面大小= 2KB 銀行清除 (單一銀行) 大量清除 (所有銀行)	部門大小= 1KB 銀行清除 (高達 256KB)
防寫保護	是 (每個銀行有 2 個防寫保護區域)	是，靜態和動態
讀取保護	是	是
快閃記憶體讀取作業	64 位元快閃字大小加 8 個 ECC 位元	同樣 - 如果存在可選 ECC
快閃記憶體寫入作業	64 位元快閃字大小加 8 個 ECC 位元	同樣 - 如果存在可選 ECC

**表 3-2. 快閃記憶體功能的比較 (continued)**

特色	STM32G0	MSPM0
錯誤代碼修正 (ECC)	8 位元 (用於 64 位元)	相同
安全的記憶體區域	是, 主記憶體	否
資訊記憶體	是	是 (NONMAIN)
OTP 資料區域	1KB	否
預取	是	是
CPU 指令快取	兩條 64 位元快取行 (16 位元組) 4x 32 位元指令或 8x 16 位元指令	四條 64 位元快取行 (32 位元組) 8x 32 位元指令或 16x 16 位元指令

除了上表所列的快閃記憶體功能外, MSPM0 快閃記憶體也具備下列功能:

- 在電路中編程並清除整個供應電壓範圍內支援的項目
- 產生內部編程電壓
- 在快閃記憶體的較低 32KB 上支援高達 100 000 個程式/清除週期的 EEPROM 模擬, 在剩餘的快閃記憶體上最多可有 10 000 個程式/清除週期 (具 32KB 的裝置在整個快閃記憶體上支援 100 000 個週期)

### 3.2.2 快閃記憶體組織

快閃記憶體用於儲存應用程式碼和資料、裝置開機配置, 以及 TI 出廠時預編程的參數。快閃記憶體會排列成一個或多個記憶體組, 每個記憶體組中的記憶體會進一步映射到一個或多個邏輯記憶體區域, 並指派系統位址空間給應用程式使用。

#### 記憶體組

大多數 MSPM0 裝置都會執行單一快閃記憶體組 (BANK0)。在具有單一快閃記憶體組的裝置上, 正在進行的程式/清除作業會停止所有對快閃記憶體的讀取請求, 直到作業完成, 且快閃記憶體控制器已釋放對快閃記憶體組的控制。在具有多個快閃記憶體組的裝置上, 快閃記憶體組上的程式/清除作業也會停止核發給正在執行程式/清除作業的快閃記憶體組的讀取請求, 但不會停止對其他快閃記憶體組核發的讀取請求。因此, 多個快閃記憶體組的存在會啓用以下應用案例:

- 雙映像韌體更新 (當第二個映像編程到第二個對稱快閃記憶體組時, 應用程式可從一個快閃記憶體組執行程式碼, 而不會阻礙應用程式的執行)
- EEPROM 模擬 (當第二個快閃記憶體組用於寫入資料, 而不停止執行應用程式時, 應用程式可從一個快閃記憶體組執行應用程式)

#### 快閃記憶體區域

每個記憶體組中的記憶體會根據每個記憶體組中的記憶體支援的功能, 映射到一個或多個邏輯區域。共有四個區域:

- 工廠 - 裝置 ID 和其他參數
- 非主要 - 裝置開機設定 (BCR 和 BSL)
- 主要 - 應用程式碼和資料
- 資料 - 資料或 EEPROM 模擬

具有一個記憶體組的裝置在 BANK0 上實作工廠、非主要和主要區域 (唯一存在的記憶體組), 而且資料區域不可用。具有多個記憶體組的裝置也會在 BANK0 上實作工廠、非主要和主要區域, 但包括可實作主要或資料區域的其他記憶體組 (BANK1 至 BANK4)。

#### 非主要記憶體

非主要記憶體是專用快閃記憶體區域, 儲存 BCR 和 BSL 用於啟動裝置的配置資料。該區域不用於其他任何目的。BCR 和 BSL 都有配置政策, 可以保留預設值 (在開發和評估過程中是典型值), 也可以透過更改編程到非主要快閃區域中的值來修改特定用途 (在生產編程過程中為典型值)。

### 3.2.3 嵌入式 SRAM

MSPM0 和 STM32G0 系列的 MCU 具有用於儲存應用資料的 SRAM。

表 3-3. SRAM 功能比較

特點	STM32G0	MSPM0
SRAM 記憶體	STM32G0B1xx、G0C1xx：144KB (啟用 SRAM 同位檢查時為 128KB) STM32G071xx、G081xx：36KB (啟用 SRAM 同位檢查時為 32KB) STM32G051xx、G061xx：18KB (啟用 SRAM 同位檢查時為 16KB) STM32G031xx、G041xx：8KB (啟用 SRAM 同位檢查時為 8KB) 零等待狀態	MSPM0Gxx：32KB 至 16KB MSPM0Lxx：4KB 至 2KB 零等待狀態 特定裝置包括 SRAM 同位檢查和 ECC。如需詳細資訊，請參閱裝置產品規格表
最大 CPU 時鐘頻率下的零等待狀態	是	是
存取解析度	位元組、半字 (16 位元) 或全字 (32 位元)	位元組、半字 (16 位元) 或全字 (32 位元)
同位檢查	是	是

MSPM0 MCU 包含低功耗高效能 SRAM，可在裝置支援的 CPU 頻率範圍內進行零等待狀態存取。除了程式碼外，SRAM 還可用於儲存不穩定的資訊，例如呼叫堆疊、堆積和全域資料。SRAM 內容會在執行、睡眠、停止和待機運作模式中完全保留，但會在關機模式中遺失。提供防寫保護機制，允許應用程式以 1KB 解析度動態防寫保護較低的 32KB SRAM。在低於 32KB SRAM 的裝置上，整個 SRAM 皆提供防寫保護。將可執行程式碼放入 SRAM 時，防寫保護非常受用，因為它提供一種防護等級，可防止 CPU 或 DMA 意外覆寫程式碼。將程式碼置於 SRAM 中可啟用零等待狀態作業並降低功耗，藉此改善關鍵迴路的效能。

### 3.3 開機和重設摘要與比較

MSPM0 裝置與 STM32G0 裝置相似，具有最低運作電壓，並將模組安裝到位，可將裝置或部分裝置保持在重設狀態，確保裝置能正常啟動。表 3-4 顯示兩個系列執行方式的比較，以及控制開機程序和跨系列重設的模組。

表 3-4. 開機比較

STM32G0 裝置		MSPM0 裝置	
管理開機和重設的模組	PWR (電源) 和 RCC (重設和時鐘控制) 模組	模組管理開機與重設	PMCU (電源管理和時鐘裝置)
<b>基於電壓電平的重設</b>			
POR (開機重設)	完成裝置重設。用於開機的第一級釋放電壓。用於關機的最低電壓電平。	POR (開機重設)	完成裝置重設。用於開機的第一級釋放電壓。用於關機的最低電壓電平。
具有可配置電平的 BOR (電壓不足重設)	有時可編程。設定開機時釋放重設狀態，或在關機時重設裝置的電壓電平。	可配置的 BOR (電壓不足重設)	可配置為具有不同電壓閾值，且結合 STM32G0 BOR 與 PVD 功能的重設或中斷。
PVD (可編程電壓偵測器)	可配置且可提供中斷的電壓監控器。		

STM32G0 定義不同的重設領域，而 MSPM0 裝置則有不同程度的重設狀態。對於 MSPM0 裝置，重設等級具有設定順序，當觸發某電平時，所有後續電平都會重設，直到裝置進入執行模式為止。表 3-5 提供 STM32G0 重設領域和 MSPM0 重設狀態之間的簡短說明和比較。圖 3-1 顯示所有 MSPM0 重設狀態之間的關係。

表 3-5. 重設領域的比較

STM32G0 重設領域		MSPM0 重設狀態 <sup>(1)</sup>	
電源重設領域	典型的觸發方式是 POR、BOR 和退出待機或關機模式。除了 VCore 領域外的所有暫存器都會重設。	POR	典型觸發器：POR 電壓電平、SW 觸發器、NRST 保持低電平 >1 秒。重設關機記憶體，重新啟用 NRST 和 SWD，觸發 BOR
		BOR	典型觸發器：POR 或 BOR 電壓電平，退出關機模式。重設 PMU、VCore 和關聯的邏輯。觸發 BOOTRST。
沒有完全同等。重設後，在 SYSCLK 的第四個時鐘週期上讀取開機配置。		開機重設 (BOOTRST)	典型觸發器：BOR 或軟體觸發、時鐘嚴重錯誤、NRST 維持低於電平 <1 秒。執行開機配置例行作業。重設多數核心邏輯和暫存器，包括 RTC、時鐘與 IO 配置。 <sup>(2)</sup> SRAM 電源循環並遺失。觸發 SYSRST。

表 3-5. 重設領域的比較 (continued)

STM32G0 重設領域		MSPM0 重設狀態 <sup>(1)</sup>	
系統重設領域	系統重設會將所有暫存器設定為其重設值，但時鐘控制和狀態暫存器 (RCC_CSR) 中的重設旗標以及 RTC 領域中的暫存器除外。	系統重設 (SYSRST)	典型觸發器：BOOTRST、BSL 進入或退出、監視計時器、軟體觸發、偵錯子系統。重設 CPU 狀態和除 RTC、LFCLK、LFXT 和 SYSOSC 頻率校正迴路之外的所有周邊設備。裝置在退出時進入執行模式。
沒有同等		僅 CPU 重設 (CPURST)	僅軟體和偵錯子系統觸發。僅重設 CPU 邏輯。周邊設備狀態不受影響。
RTC 領域	由軟體或 VDD 或 VBAT 電源開啟進行觸發，如果二者電源先前都已關閉的話。僅重設 LSE 振盪器、RTC、備份暫存器和 RCC RTC 領域控制暫存器。	RTC 和關聯的時鐘透過 BOOTRST、BOR 或 POR 重設。 <sup>(2)</sup>	

- (1) 並非所有重設觸發程序均已說明。如需有關所有可用的重設觸發器的詳細資訊，請參閱裝置 TRM 的 PMCU 一章。  
 (2) 如果 BOOTRST 原因來自 NRST 或軟體觸發器，則 RTC、LFCLK 和 LFXT/LFCLK\_IN 配置和 IOMUX 設定都不會重設，進而允許 RTC 透過外部重設來維護運作。

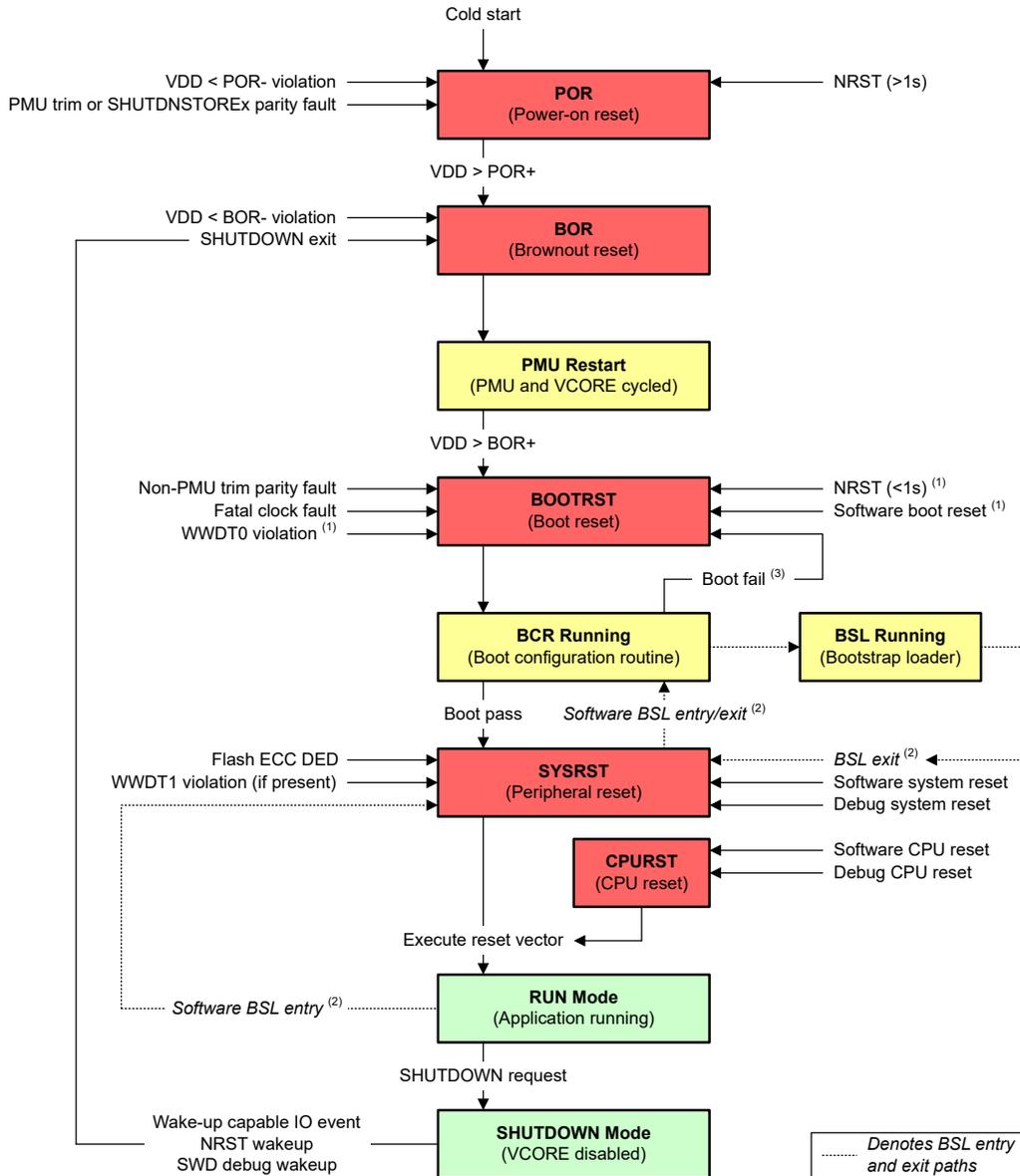


图 3-1. MSPM0 重設等級

### 3.4 時鐘摘要和比較

STM32G 和 MSPM0 包含主時鐘來源的內部振盪器。時鐘可劃分為其他時鐘來源、並可分佈於多種周邊設備。

表 3-6. 震盪器比較

STM32G0 振盪器	MSPM0 振盪器
HSI16RC 16 MHz	SYSOSC <sup>(1)</sup>
HSI48RC 48 MHz	SYSOSC
LSI RC 32 kHz	LFOSC
HSE OSC 4-48 MHz	HFXT
LSE OSC 32 kHz	LFXT
I2S_CLKIN	HFCLK_IN (數位時鐘)

(1) SYSOSC 可編程為 32 MHz、24 MHz、16 MHz 或 4 MHz。

表 3-7. 時鐘比較

STM32G 時鐘	MSPM0 時鐘
HSISYS	不適用
PLLCLK	SYSPLLCLK1
PLLQCLK	SYSPLLCLK1
PLLCLK	SYSPLLCLK0
不適用	SYSPLLCLK2x <sup>(1)</sup>
SYSCLK	BUSCLK <sup>(2)</sup>
HCLK	MCLK
HCLK8	CPCLK
PCLK	BUSCLK
TIMCLK	BUSCLK
LPTIMx_IN	LFCLK_IN

(1) SYSPLLCLK2x 是 PLL 模組輸出速度的兩倍，可以向下分隔。

(2) BUSCLK 取決於電源域。若為電源域 0，則 BUSCLK 為 ULCLK。若為電源域 1，則 BUSCLK 為 MCLK。

表 3-8. 周邊時鐘來源

周邊設備	STM32G 時鐘來源	MSPM0 時鐘來源
RTC	LSI、LSE、HSE/32	LFCLK (LFOSC、LFXT)
UART	PCLK、LSE、HSI16、SYSCLK	BUSCLK、MFCLK、LFCLK
SPI	需要尋找	BUSCLK、MFCLK、LFCLK
I2C	PCLK、HSI16、SYSCLK	BUSCLK、MFCLK
ADC	HSI16、SYSCLK、PLLCLK	ULCLK、HFCLK、SYSOSC
控制器區域網路	PCLK、HSE、PLLQCLK	PLLCLK1、HFCLK
計時器	PCLK、TIMCLK、PLLQCLK	BUSCLK、MFCLK、LFCLK
LPTIM 1/2 (TIM0/1)	PCLK、LSI、LSE、HSI16、LPTIMX_IN	LFCLK、ULCLK、LFCLK_IN
RNG	HSI48、PLLQCLK、HSI16/8、SYSCLK	MCLK

每個裝置系列的 TRM 都有一個時鐘樹，可協助視覺化時鐘系統。Sysconfig 可協助提供時鐘分配與尋找周邊設備供應來源選項。

### 3.5 MSPM0 運作模式摘要和比較

MSPM0 MCU 提供五種主要運作模式 (功率模式)，可根據應用需求最佳化裝置功耗。要降低功率，這些模式為：RUN (執行)、SLEEP (睡眠)、STOP (停止)、STANDBY (待機) 和 SHUTDOWN (關機)。CPU 在執行模式下執行程式碼。周邊設備中斷事件可將裝置從睡眠、停止或待機模式喚醒至執行模式。關機模式會完全停用內部核心穩壓器，以將功耗降至最低，而且只能透過 NRST、SWD 或特定 IO 上的邏輯位準相符項目進行喚醒。執行、睡眠、停止和待機模式也包括了幾個可配置的政策選項 (例如，RUN.x)，可用於平衡效能和功耗。

為進一步平衡效能與功耗，MSPM0 裝置採用兩種功率域：PD1 (適用於 CPU、記憶體和高效能週邊設備) 和 PD0 (適用於低速、低功耗周邊設備)。PD1 始終在執行和睡眠模式下通電，但在所有其它模式下均被停用。PD0 始終在執行、睡眠、停止和待機模式下通電。在關機模式下，PD1 和 PD0 都被停用。

#### 運作模式比較

STM32G0 裝置具有類似的運作模式。下表提供了 STM32G0 與 MSPM0 裝置之間的簡短比較。

**表 3-9. STM32G0 與 MSPM0 裝置之間的運作模式比較**

STM32G0		MSPM0	
模式	說明	模式	說明
執行	提供全時脈與周邊設備	執行	0 提供全時脈與周邊設備
LP 執行	CPU 限制為 2 MHz		1 SYSOSC 在設定頻率時，CPUCLK 和 MCLK 限制為 32 kHz
			2 SYSOSC 停用；CPUCLK 和 MCLK 限制為 32 kHz
睡眠	CPU 未計時	睡眠	0 CPU 未計時
LP 睡眠	與 LP 執行相同，但 CPU 不計時		1 與 Run1 相同，但 CPU 不計時
			2 與 Run2 相同，但 CPU 不計時
停止	0 VCORE 領域時鐘已停用	停止	0 睡眠 0 + PD1 已停用
	1 停止 0 + 主電源穩壓器關閉		1 睡眠 1 + SYSOSC 檔位偏移至 4 MHz
			2 睡眠 2 + ULPCLK 限制為 32 kHz
待機	具有 BOR 功能的最低功耗；RTC 可用；暫存器設定遺失。	待機	0 具有 BOR 功能的最低功耗；所有 PD0 周邊設備均可在 32 kHz 時接收 ULPCLK 和 LFCLK；RTC 可隨 RTCCLK 使用
			1 只有 TIMG0 和 TIMG1 可以在 32 kHz 時接收 ULPCLK 或 LFCLK；RTC 可隨 RTCCLK 使用
停機	無時鐘或 BOR。核心調節關閉。RTC 域仍可在作用狀態。退出觸發器重設。	停機	無時鐘、BOR 或 RTC。核心調節關閉。PD1 和 PD0 已停用。退出觸發器重設位準 BOR。

#### 低功耗模式下的 MSPM0 功能

如表 3-9 中所示，MSPM0 周邊設備或周邊設備模式在低功耗運作模式下的可用性或運作速度可能受到限制。如需詳細資訊，請參閱 MSPM0 裝置特定產品規格表中的「運作模式支援的功能」表格，例如：

[MSPM0G350x 混合訊號微控制器產品規格表](#)

[MSPM0L134x、MSPM0L130x 混合訊號微控制器產品規格表](#)

MSPM0 裝置的另一項功能是讓一些周邊設備能夠執行非同步快速時鐘要求。這可讓 MSPM0 裝置以較低的功率模式執行，其中周邊設備並未作用，但仍可觸發或啟用周邊設備。當發生非同步快速時鐘要求時，MSPM0 裝置能夠快速將內部振盪器提升至更高速及/或暫時進入更高運作模式，以處理即將發生的動作。這可讓 CPU 從計時器、比較器、GPIO 和 RTC 快速喚醒；接收 SPI、UART 和 I2C；或觸發 DMA 傳輸和 ADC 轉換，同時在最低功率模式下進入睡眠。如需實作非同步時鐘要求以及周邊設備支援與用途的詳細資訊，請參閱 MSPM0 TRM 中的適當章節。

[MSPM0 G 系列 80-MHz 微控制器技術參考手冊](#)

[MSPM0 L 系列 32-MHz 微控制器技術參考手冊](#)

## 進入低功耗模式

與 STM32G0 裝置類似，MSPM0 裝置在執行等待事件 `_WFE()`；或等待中斷 `_WFI()`；指令時進入低功耗模式。低功耗模式由目前的電源政策設定決定。裝置電源政策由驅動程式庫函數設定。以下函數呼叫會將該電源政策設定為待機 0。

```
DL_SYSCTL_setPowerPolicySTANDBY0();
```

STANDBY0 可替換為所選的運作模式。如需管理電源政策的 driverlib API 的完整單，請參閱 [MSPM0 SDK DriverLib API 指南](#) 的這個章節。另請參閱以下示範進入不同運作模式的程式碼範例。每個 MSPM0 裝置都有類似的範例。

## 低功耗模式程式碼範例

瀏覽至 SDK 安裝，並在範例 > nortos > LP 名稱 > driverlib 中尋找低功耗模式程式碼範例

## 3.6 中斷與事件比較

### 中斷和例外

MSPM0 和 STM32G0 會根據裝置的可用周邊設備來登錄和對應中斷和例外狀況向量。表 3-10 中包含每個設備系列的中斷向量的摘要和比較。中斷或異常的優先順序較低的值，優先於具有較高優先順序值的中斷。對於其中的一些向量，優先順序是可供使用者選擇的，對於其他向量則是固定的。

在 MSPM0 和 STM32G0 中，NMI、RESET 和硬錯誤處理常式等例外會被指定為負優先順序值，以表示它們始終優先於周邊設備中斷。對於具有可選中斷優先等級的周邊設備，兩個系列裝置上最多可以使用 4 個可編程優先等級。

表 3-10. 中斷比較

NVIC 編號	STM32G0		MSPM0x	
	中斷/例外	優先順序	中斷/例外	優先順序
-	重設	固定：-3	重設	固定：-3
-	NMI 處理常式	固定：-2	NMI 處理常式	固定：-2
-	硬錯誤處理常式	固定：-1	硬錯誤處理常式	固定：-1
-	SVCALL 處理常式	可選	SVCALL 處理常式	可選
-	PendSV	可選	PendSV	可選
-	SysTick	可選	SysTick	可選
0	窗型監控系統中斷	可選	INT_GROUP0：WWDT0、DEBUGSS、FLASHCTL、WUC FSUBx 和 SYSCTL	可選
1	功耗電壓偵測器中斷	可選	INT_GROUP1：GPIO0 和 COMP0	可選
2	RTC 和時間戳記	可選	計時器 G1 (TIMG1)	可選
3	快閃記憶體全域中斷	可選	UART3 <sup>(1)</sup>	可選
4	RCC 全域中斷	可選	ADC0	可選
5	EXTI0 和 EXTI1 中斷	可選	ADC1 <sup>(1)</sup>	可選
6	EXTI2 和 EXTI3 中斷	可選	CANFD0 <sup>(1)</sup>	可選
7	EXTI4-EXTI15 中斷	可選	DAC0 <sup>(1)</sup>	可選
8	UCPD1/UCPD2/USB	可選	已保留	可選
9	DMA1 通道 1	可選	SPI0	可選
10	DMA1 通道 2 和 3	可選	SPI1 <sup>(1)</sup>	可選
11	DMA1 通道 4-6 和 DMA2 通道 1-5	可選	已保留	可選
12	ADC 和比較器	可選	已保留	可選
13	計時器 1 (TIM1)、中斷、更新、觸發和變換	可選	UART1	可選
14	TIM1 擷取比較	可選	UART2 <sup>(1)</sup>	可選
15	TIM2 全域中斷	可選	UART0	可選

表 3-10. 中斷比較 (continued)

NVIC 編號	STM32G0		MSPM0x	
	中斷/例外	優先順序	中斷/例外	優先順序
16	TIM3 和 TIM4 全域中斷	可選	TIMG0	可選
17	TIM6、LPTIM1 和 DAC 中斷	可選	TIMG10 <sup>(1)</sup>	可選
18	TIM6 和 LPTIM2 全域中斷	可選	TIMA0 <sup>(1)</sup>	可選
19	TIM14 全域中斷	可選	TIMA1	可選
20	TIM15 全域中斷	可選	TIMA2 <sup>(2)</sup>	可選
21	TIM16 和 FDCAN0 全域中斷	可選	TIMH0 <sup>(1)</sup>	可選
22	TIM17 和 FDCAN1 全域中斷	可選	已保留	可選
23	I2C1 全域中斷	可選	已保留	可選
24	I2C2 和 I2C3 全域中斷	可選	I2C0	可選
25	SPI1 全域中斷	可選	I2C1	可選
26	SPI2 和 SPI3 全域中斷	可選	已保留	可選
27	USART1 全域中斷	可選	已保留	可選
28	USART2 和 LPUART2 全域中斷	可選	AES <sup>(1)</sup>	可選
29	USART 3-6 和 LPUART1 全域中斷	可選	已保留	可選
30	CEC 全域中斷	可選	RTC <sup>(1)</sup>	可選
31	AES 和 RNG 全域中斷	可選	DMA	可選

(1) 僅能在 MSPM0G 系列裝置中使用。

(2) MSPM0L 系列裝置上的 TIMG4

### 事件處理常式和 EXTI (延伸中斷和事件控制器)

MSPM0 裝置包含專屬事件管理器周邊設備，此裝置延伸 NVIC 的概念，允許將周邊設備的數位事件以中斷形式傳輸至 CPU、以觸發形式傳輸至 DMA，或傳輸至另一個周邊設備來觸發硬體動作。事件管理器也可與電源管理和時鐘單元 (PMCU) 執行交握，確保存在必要的時鐘和電源域，以執行觸發的事件動作。

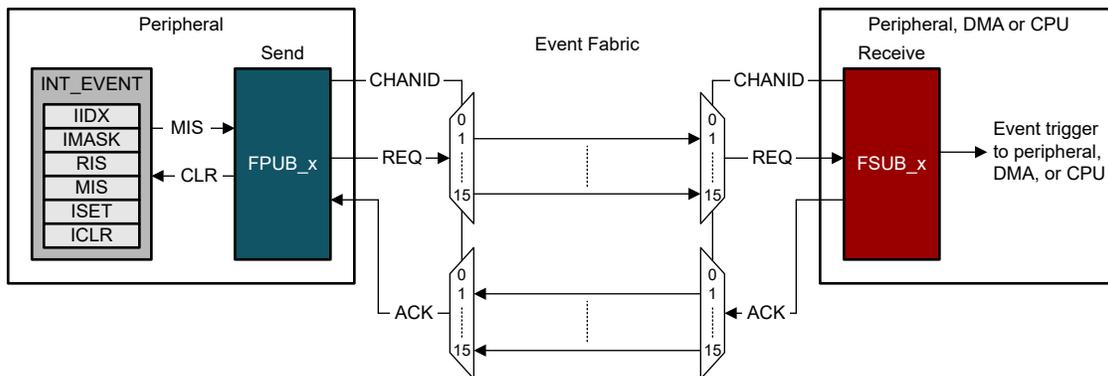


图 3-2. 一般事件路由

在 MSPM0 事件管理器中，產生事件的周邊設備稱為發佈者，而以發佈器為基礎的周邊設備、DMA 或 CPU 稱為訂閱者。潛在的可用發佈者和訂閱者組合極度靈活，可以在移轉軟體時使用，以取代以前由中斷向量和 CPU 處理的功能，進而完全繞過 CPU。例如，I<sup>2</sup>C-to-UART 橋接器可能在先前收到 I<sup>2</sup>C STOP 時即已觸發 UART 傳輸 (使用 ISR 設定旗標)，或直接載入 UART TX 緩衝器。使用 MSPM0 事件處理常式時，I<sup>2</sup>C 交易完成事件可能會觸發 DMA 以直接載入 UART TX 緩衝區，因此不需要 CPU 採取任何動作。

請參閱 [MSPM0G 技術參考手冊](#) 或 [MSPM0L 技術參考手冊](#) 的「活動」一節，以取得有關使用 MSPM0 事件處理常式的詳細資訊。

為不與 MSPM0 事件處理常式混淆，STM32G0 系列裝置實作延伸中斷和事件控制器 (EXTI)，如此可透過 IOS 或周邊設備的可配置事件從停止模式喚醒系統。STM32G0 EXTI 的喚醒功能在 MSPM0 中可使用 IO 喚醒功能 (請參閱 [MSPM0 技術參考手冊](#) 的 IOMUX 一節) 和 GPIO FastWake (請參閱 [MSPM0 技術參考手冊](#) 的 GPIO 一節) 進行

最佳複製。如果喚醒是針對單一動作，則事件處理常式周邊設備可以請求必要的 PMCU 資源以使周邊設備運作，並在之後返回適用的低功耗模式。

### 3.7 偵錯和編程比較

Arm SWD 2 線 JTAG 埠是 MSPM0 和 STM32G0 裝置的主要偵錯和編程介面。此介面通常會在應用程式開發期間及生產編程期間使用。表 3-11 會比較兩個裝置系列之間的功能。有關 MSPM0 偵錯介面安全功能的其他資訊，請參閱 *MSPM0 MCU 中的 Cybersecurity Enablers 應用說明*。

表 3-11. Arm SWD JTAG 功能比較

	STM32G0	MSPM0
偵錯埠	Arm SWD 連接埠 (2 線)	Arm SWD 連接埠 (2 線)
斷點單元 (BPU)	4 硬體斷點	4 硬體斷點
觀看數據裝置 (DWT)	2 個觀看點	2 個觀看點
微追蹤緩衝器 (MTB)	否	MTB 支援 4 個追蹤封包 <sup>(1)</sup>
低功耗偵錯支援	是	是
EnergyTrace s 支援	否	EnergyTrace+ 支援 (具有功率分佈的 CPU 狀態)
偵錯期間的周邊設備執行支援	是	是
偵錯介面鎖定	可以暫時阻止偵錯讀取權限	可以永久停用偵錯功能，也可用密碼鎖定

(1) 僅限 MSPM0Gxxxx 裝置

### Bootstrap 載入程式 (BSL) 編程選項

Bootstrap 載入程式 (BSL) 編程介面是 Arm SWD 的替代編程介面。此介面僅提供編程功能，通常會透過標準嵌入式通訊介面來使用。這允許透過與系統或外部連接埠中的其他嵌入式裝置的現有連接進行韌體更新。雖然編程更新是此介面的主要用途，但也可用於初始生產編程。表 3-12 顯示 MSPM0 和 STM32G0 裝置系列之間不同選項和功能的比較。

表 3-12. BSL 功能比較

BSL 功能	STM32G0	MSPM0
BSL 已在空白裝置上啟動	是	是
自動偵測編程介面	是	是
安全性	記憶體安全和存取限制選項	安全開機選項；CRC 防護
可自訂	否	是，可配置的叫用針腳和外掛程式功能
叫用方法	模式 <sup>(1)</sup> 涵蓋多達 2 個針腳和 RESET、SW 進入點的裝置暫存器設定	位於 BOOTRST、SW 進入點的 1 針腳高電平
支援的介面		
UART	是	是
I2C	是	是
SPI	是 <sup>(2)</sup>	需要自訂外掛程式
控制器區域網路	是 <sup>(2)</sup>	已規劃外掛程式 <sup>(2)</sup>
USB	是 <sup>(2)</sup>	目前沒有具備 USB 功能的 MSPM0 裝置。

(1) 模式選項可用性取決於裝置。

(2) 僅於特定裝置

## 4 數位周邊設備比較

### 4.1 通用 I/O (GPIO、IOMUX)

MSPM0 GPIO 功能幾乎涵蓋 STM32G0 GPIO 提供的所有功能。STM32G0 使用術語 GPIO 來提及負責管理裝置針腳的所有功能。然而，MSPM0 使用的命名方式稍有不同，也就是：

- MSPM0 GPIO 是指能夠讀寫 IO、產生中斷等的硬體。
- MSPM0 IOMUX 是指負責將不同的內部數位周邊設備連接至針腳的硬體。IOMUX 服務許多不同的數位周邊設備，包括但不限於 GPIO。

MSPM0 GPIO 和 IOMUX 的功能與 STM32G0 GPIO 相同。此外，MSPM0 提供的功能在 STM32G0 裝置中不可用，例如 DMA 連線能力、可控制的輸入過濾和事件功能。

**表 4-1. GPIO 功能比較**

特點	STM32G0	MSPM0G 和 MSPM0L
輸出模式	推挽 具上拉或下拉功能的開漏	同等
GPIO 速度選擇	每個 I/O 的速度選擇	相似 MSPM0 在所有 IO 針腳上提供標準 IO (SDIO)。SDIO 可與 STM GPIO 速度=01 相當或更好。 MSPM0 高速 IO (HSIO) 可在特定針腳上使用。HSIO 相當於 STM GPIO 速度=10。
高驅動 GPIO	大約 20 mA	等效，稱為高驅動器 IO (HDIO)
輸入模式	浮動 上拉或下拉 類比	同等
原子位元設定和重設	是	同等
GPIO 鎖定	暫存器上鎖機制	沒有同等的 MSPM0
替代功能	選擇暫存器	同等 MSPM0 使用 IOMUX
快速切換	每兩個時鐘變更	等效的 MSPM0 可以在每個時鐘週期切換針腳
喚醒	GPIO 針腳狀態變更	同等
由 DMA 控制的 GPIO	否	僅適用於 MSPM0
使用者控制輸入篩選以排除少於 1、3 或 8 ULPCLK 期間的干擾	否	僅適用於 MSPM0
使用者可控制的輸入磁滯	否	僅適用於 MSPM0

### GPIO 程式碼範例

有關 GPIO 程式碼範例的資訊可在 [MSPM0 SDK 範例指南](#) 中找到。

## 4.2 通用非同步接收器-發射器 (UART)

STM32G0 和 MSPM0 都提供了執行非同步 (無時鐘) 通訊的周邊設備。這些 UART 周邊設備有兩種變體，一個具有標準功能，一個具有進階功能。命名差異如 [表 4-2](#) 所示。

**表 4-2. STM32G0 和 MSPM0 之間的 UART 命名差異**

	STM32G0 命名	MSPM0 命名
標準功能	基本	主要
進階功能	完整	延伸

**表 4-3. UART 進階功能集比較**

特點	STM32G0 USART 完整功能集	MSPM0L 和 MSPM0G UART 延伸功能集
硬體流程控制	是	是
使用 DMA 連續通訊	是	是
多處理器	是	是
同步模式	是	否
智慧卡模式 (ISO7816)	是	是
單線半雙工通訊	是	是 <sup>(1)</sup>
IrDA 硬體支援	是	是
LIN 硬體支援	是	是
DALI 硬體支援	否	是
曼徹斯特編碼硬體支援	否	是
從低功耗模式喚醒	是	是
自動傳輸速率偵測	是	否
驅動程式啟用	是	是
資料長度	7、8、9	5、6、7、8
Tx/Rx FIFO 深度	8	4

(1) 需要在傳輸和接收之間重新配置周邊設備

**表 4-4. UART 標準功能集比較**

特點	STM32G0 USART 基本功能集	MSPM0 UART 主要功能集
硬體流程控制	是	是
使用 DMA 連續通訊	是	是
多處理器	是	是
同步模式	是	否
單線半雙工通訊	是	是 <sup>(1)</sup>
從低功耗模式喚醒	否	是
驅動程式啟用	是	是
資料長度	7、8、9	5、6、7、8
Tx/Rx FIFO 深度	無	4

(1) 需要在傳輸和接收之間重新配置周邊設備

### UART 程式碼範例

有關 UART 程式碼範例的資訊可在 [MSPM0 SDK 範例指南](#) 中找到。

## 4.3 序列周邊介面 (SPI)

MSPM0 和 STM32G0 都支援序列周邊介面 (SPI)。總體來說，MSPM0 和 STM32G0 SPI 支援與 [表 4-5](#) 中列出的差異相當。

**表 4-5. SPI 功能比較**

特點	STM32G0x	MSPM0L 和 MSPM0G
控制器或周邊設備運作	是	是
資料位元寬度 (控制器模式)	4 至 16 位元	4 至 16 位元
資料位元寬度 (周邊設備模式)	4 至 16 位元	7 至 16 位元
速度上限	32 MHz	MSPM0L : 16 MHz
		MSPM0G : 32 MHz
全雙工傳輸	是	是
半雙工傳輸 (雙向資料線路)	是	否
單工傳輸 (單向資料線路)	是	是
多控制器能力	是	否
硬體晶片選擇管理	是 (1 個周邊設備)	是 (4 個周邊設備)
可編程時鐘極性和相位	是	是
MSB 優先或 LSB 優先變化的可編程資料順序	是	是
SPI 格式支援	Motorola、TI	Motorola、TI、MICROWIRE
硬體 CRC	是	否, MSPM0 提供 SPI 同位檢查模式
TX FIFO 深度	視資料大小而定	4
RX FIFO 深度	視資料大小而定	4

### SPI 程式碼範例

有關 SPI 程式碼範例的資訊可在 [MSPM0 SDK 範例指南](#) 中找到。

### 4.4 I<sup>2</sup>C

MSPM0 和 STM32G0 都支援 I<sup>2</sup>C。總體 MSPM0 和 STM32G0 I<sup>2</sup>C 支援可與下表中列出的顯著差異進行比較。

**表 4-6. I<sup>2</sup>C 功能比較**

特點	STM32G0	MSPM0L 和 MSPM0G
控制器和目標模式	是	是
多控制器能力	是	是
標準模式 (高達 100 kHz)	是	是
快速模式 (高達 400 kHz)	是	是
加快模式 (高達 1 MHz)	是	是
定址模式	7、10 位元	7 位元
周邊設備位址	2 個位址和 1 個可配置遮罩	2 個位址
全體呼叫	是	是
可編程設定和保留時間	是	否
事件管理	是	是
時脈擴展	是	是
軟體重設	是	是
FIFO/緩衝器	1 個位元組	TX : 8 個位元組
		RX : 8 個位元組
DMA	是	是
可編程類比及數位雜訊濾波器	是	是

### I<sup>2</sup>C 程式碼範例

有關 I<sup>2</sup>C 程式碼範例的資訊可在 [MSPM0 SDK 範例指南](#) 中找到。

## 4.5 計時器 (TIMGx、TIMAx)

STM32G0 和 MSPM0 都提供各種計時器。MSPM0 提供各種功能的計時器，可支援從低功耗監控到進階馬達控制的應用實例。

表 4-7. 計時器命名

STM32G0		MSPM0	
計時器名稱	縮寫名稱	計時器名稱	縮寫名稱
進階控制	TIM1	進階控制	TIMA0
一般用途	TIM2-4、TIM14/-17	一般用途	TIMG0-11
		高解析度	TIMG12
基本	TIM6/7		
低功耗	LPTIM		

表 4-8. 計時器功能比較

特點	STM32G0 計時器	MSPM0G 計時器	MSPM0L 計時器
解析度	16 位元、32 位元	16 位元、32 位元	16 位元
PWM	是	是	是
擷取	是	是	是
比較	是	是	是
單次	是	是	是
上下計數功能	是	是	是
功率模式	是	是	是
QEI 支援	是	是	否
可編程的預分頻器	是	是	是
遮蔽暫存器模式	是	是	是
事件/中斷	是	是	是
故障事件機制	是	是	否
自動重新載入功能	是	是	是

表 4-9. 計時器模組更換

STM32G0 計時器	MSPM0 同等	推論
TIM1	TIMA、TIMG8-12	進階控制、兩個 16 位元解析度、QEI 支援
TIM2	TIMG12	32 位元解析度
TIM3/4	TIMG0-7	通用 16 位元解析度
TIM6/7	任意	基本計時器
TIM14	任意	與 TIM3/4 相同的功能
TIM15/16/17	任意	一般用途
LPTIM	PD0 中的任何計時器	LPTIM 來源 LFCLK、PD0 - MSPM0 中的低功耗模式

表 4-10. 計時器應用實例比較

特點	STM32G0 計時器	MSPM0 計時器
PWM	TIM1-4 具有邊緣和中心對齊選項，TIM6-7 沒有 PWM 功能。TIM15-17 僅對齊邊選項。	所有計時器都有邊對齊或置中對齊選項
擷取	沒有重大差異	沒有重大差異
比較	沒有重大差異	沒有重大差異
單次	沒有重大差異	沒有重大差異
預分頻器	除 LPTIM (3 位元預分頻器) 之外的 16 位元預分頻器	8 位元預分頻器
同步	TIM1-4, TIM15	所有計時器都具有此功能

## 計時器程式碼範例

有關計時器程式碼範例的資訊可在 [MSPM0 SDK 範例指南](#) 中找到。

### 4.6 窗型監視計時器 (WWDT)

STM32G0 和 MSPM0 都有提供窗型監視計時器。當應用程式在指定的時段無法簽入時，窗型監視計時器 (WWDT) 會啟動系統重設。

表 4-11. WWDT 命名

金鑰	STM32G0	MSPM0
名稱	獨立監視計時器，窗型監視計時器	窗型監視計時器
縮寫名稱 (相同順序)	IWDG、WWDG	WWDT

表 4-12. WDT 功能比較

特點	STM32G0	MSPM0G	MSPM0L
視窗模式	是	是	是
間隔計時器模式	是	是	是
LFCLK 來源	是	是	是
中斷	是	是	是
計數器解析度	7 位元	25 位元	25 位元
時脈除頻器	WWDG 否、IWDG 是	是	是

## WWDT 程式碼範例

有關 WWDT 程式碼範例的資訊可在 [MSPM0 SDK 範例指南](#) 中找到。

### 4.7 即時時鐘 (RTC)

STM32G0 和 MSPM0<sup>1</sup> 兩者皆提供即時時鐘 (RTC)。即時時鐘 (RTC) 模組以可選擇二進位或二進位編碼十進位格式提供應用程式的時間追蹤，其計數器可顯示秒鐘、分鐘、小時、週間日、月份日期和年份。

表 4-13. RTC 功能比較

特點	STM32G0	MSPM0G
功率模式	是	是
二進位編碼格式	是	是
閏年修正	是	是
可自訂的警報數	2	2
內部和外部晶體	是	是
晶體偏移校準	是	是
預分頻器區塊	是	是
中斷	是	是

## RTC 程式碼範例

有關 RTC 程式碼範例的資訊可在 [MSPM0 SDK 範例指南](#) 中找到。

<sup>1</sup> 僅 MSPM0G 裝置支援 RTC。

## 5 類比周邊設備比較

### 5.1 類比轉數位轉換器 (ADC)

STM32G0 和 MSPM0 都提供 ADC 周邊設備，可將類比訊號轉換為數位等效訊號。兩種裝置系列皆配備 12 位元 ADC。下表比較 ADC 的不同功能和模式。

表 5-1. 功能集比較

特點	STM32G0	MSPM0G	MSPM0L
解析度 (位元)	12	12	12
轉換率 (Msps)	2.5	4	1.4
超取樣 (位元)	16	14	不適用
硬體超取樣	256x	128x	不適用
FIFO	否	是	是
ADC 參考 (V)	內部：2.048、2.5	內部：1.4、2.5、VDD	內部：1.4、2.5、VDD
	當 $V_{DD} < 2$ 時 外部： $V_{REF} = V_{DD}$	外部： $1.4 \leq V_{REF} \leq V_{DD}$	外部： $1.4 \leq V_{REF} \leq V_{DD}$
	當 $V_{DD} \geq 2$ 時 外部： $2 \leq V_{REF} \leq V_{DD}$		
操作功率模式	執行、睡眠	執行、睡眠、停止、待機 <sup>(1)</sup>	執行、睡眠、停止、待機 <sup>(1)</sup>
自動關機	是	是	是
外部輸入通道 <sup>(2)</sup>	最高 16	最高 16	最高 16
內部輸入通道	溫度感測器、VREF、VBAT	溫度感測器、供應監控、類比訊號鍵	溫度感測器、供應監控、類比訊號鍵
DMA 支援	是	是	是
ADC 窗口比較器裝置	否	是	是
同時取樣	否	是	否
ADC 數量 <sup>(3)</sup>	最高 1	最高 2	最高 1

(1) ADC 可在待機模式中觸發，進而變更運作模式。

(2) 外部輸入通道的數量因裝置而異。

(3) ADC 的數量因裝置而異。

表 5-2. 轉換模式

STM32G0	MSPM0	說明
單次轉換模式	單通道單次轉換	ADC 取樣並轉換一次單通道
掃描一系列通道	通道轉換順序	ADC 會取樣一連串的通道，並轉換一次。
連續轉換模式	重複單通道轉換	重複單通道連續取樣並轉換單一通道
	重複通道轉換順序	取樣並轉換一系列通道，然後重複相同序列
不連續模式	重複通道轉換順序	取樣並轉換不連續的通道集。這可以透過將 MEMCTRLx 對應至不同通道在 MSPM0 上完成。

### ADC 程式碼範例

有關 ADC 程式碼範例的資訊，可在 [MSPM0 SDK 範例指南](#) 中找到。

### 5.2 比較器 (COMP)

STM32G0 和 MSPM0 系列的零件都有提供整合式比較器，做為部分裝置上的選用周邊設備。在這兩個裝置系列中、這些都會標示為 COMPx，其中「x」最終字元是指所考慮的特定比較器模組。在 STM32G0 系列中編號為 1-3，在 MSPM0 系列中編號為 0-2。比較器模組可在配備超過 1 個比較器的裝置中提供窗型比較器功能，並可接

收各種內部及外部來源的輸入，且可用於觸發功率模式變更或截斷/控制 PWM 訊號。表 5-3 中包含了有關 MSPM0 和 STM32G0 比較器模組如何比較功能的摘要。

表 5-3. COMP 功能集比較

特點	SMT32G0	MSPM0G	MSPM0L
可用比較器	最高 3	最高 3	最高 1
輸出路由	多工 I/O 針腳	多工 I/O 針腳	多工 I/O 針腳
	EXTI 中斷	中斷/事件介面	中斷/事件介面
非反相輸入來源	多工 I/O 針腳	多工 I/O 針腳	多工 I/O 針腳
		DAC12 輸出 <sup>(1)</sup>	DAC8 輸出
		DAC8 輸出	OPA1 輸出 <sup>(2)</sup>
		內部 $V_{REF}$ : 1.4 V 和 2.5 V	
OPA1 輸出 <sup>(2)</sup>			
反相輸入來源	多工 I/O 針腳	多工 I/O 針腳	多工 I/O 針腳
	DAC 通道 1 和 2	內部溫度感測器	內部溫度感測器
	內部 $V_{REF}$ : 2.048 V 和 2.5 V	內部 $V_{REF}$ : 1.4 V 和 2.5 V	DAC8 輸出
	緩衝的 $V_{REF}$ 分頻器包括： $\frac{1}{4}V_{REF}$ 、 $\frac{1}{2}V_{REF}$ 和 $\frac{3}{4}V_{REF}$	DAC8 輸出 OPA0 輸出 <sup>(3)</sup>	OPA0 <sup>(3)</sup> 輸出
可編程的磁滯	無、10 mV、20 mV、30 mV	無、10 mV、20 mV、30 mV	無、10 mV、20 mV、30 mV
		其他使用 DAC8 之 0 V 至 $V_{REF}/V_{DD}$ 的值	其他使用 DAC8 之 0 V 至 $V_{DD}$ 的值
暫存器鎖定	是，所有 COMP 暫存器 (在裝置重設時停用)	是，有些 COMP 暫存器 (寫入需要金鑰)	是，有些 COMP 暫存器 (寫入需要金鑰)
窗型比較器配置	是	是	無 (單一 COMP)
輸入短模式	否	是	是
運作模式	高速、中速	高速、低功耗	高速、低功耗
快速 PWM 關機	是	是 (藉由 TIMA 故障處理常式)	否
輸出濾波	消隱濾波器	消隱濾波器	消隱濾波器
		可調式類比濾波器	可調式類比濾波器
輸出極性控制	是	是	是
中斷	正緣	正緣	正緣
	負緣	負緣	負緣
	雙邊緣	輸出就緒	輸出就緒
交換輸入模式	否	是	是

(1) 僅在具有 DAC12 周邊設備的裝置上

(2) 僅在具有 OPA1 周邊設備的裝置上

(3) 僅在具有 OPA0 周邊設備的裝置上

## COMP 程式碼範例

有關 COMP 程式碼範例的資訊可在 [MSPM0 SDK 範例指南](#) 中找到。

### 5.3 數位轉類比轉換器 (DAC)

STM32G0 和 MSPM0 系列的零件都有提供 12 位元 DAC 周邊設備，可針對各種應用執行數位轉類比的轉換作業。在 STM32G0 文件中，此周邊設備稱為 DAC。在 MSPM0 技術參考手冊、MSPM0 系列產品規格表和 MSPM0 SDK 中，12 位元 DAC 周邊設備稱為 DAC12。這使得 DAC12 與 8 位元 DAC 不同，後者可與指定的 MSPM0 裝置中的每個比較器周邊設備搭配使用。這些額外的 8 位元 DAC 都涵蓋在本文件的比較器一節中。此 DAC12 周邊設備僅適用於 MSPM0G 系列裝置。

STM32G0 和 MSPM0G 的 12 位元 DAC 周邊設備的功能摘要可在 [表 5-4](#) 中找到。

表 5-4. DAC 功能集比較

特點	STM32G0	MSPM0
解析度	12 位元 (11.4 至 11.5 ENOB)	12 位元 (11 ENOB)
輸出率	1 MSPS	1 MSPS
輸出通道	2. <sup>(1)</sup>	1. <sup>(2)</sup>
資料格式	8 位元向右對齊，12 位元向右對齊，12 位元向左對齊	8 位元向右對齊，12 位元向右對齊，二補數或直接二進位
DMA 整合	是	是
輸出路由	外部針腳	外部針腳
	內部周邊設備連線：COMP IN-、ADC	內部周邊設備連線：OPA IN+、COMP IN+、ADC0
內部參考電壓	是，2.5 V 或 2.048 V	是，2.5 V 或 1.4 V
外部參考電壓	是	是
FIFO	否	是
輸出緩沖器	是	是
可配置的輸出偏移	是	是
自我校正模式	是	是
自動產生波形	雜訊波、三角波	否
取樣保持模式	是	否
觸發來源	外部針腳、內部計時器訊號、DAC 保持時鐘、DMA 欠負載運轉	內部專用取樣時間產生器、DMA 中斷/事件、FIFO 閾值中斷/事件、2 個硬體觸發器 (可從事件結構獲得)

(1) 僅適用於某些裝置。

(2) 未來的 MSPM0G 裝置已規劃雙 DAC 通道。

### DAC12 程式碼範例

有關 DAC12 程式碼範例的資訊可在 [MSPM0 SDK 範例指南](#) 中找到。

### 5.4 運算放大器 (OPA)

STM32G0 系列裝置不提供整合式運算放大器 (OPA) 周邊設備，但從 STM32G0 移轉至 MSPM0 系列時，您可以使用 MSPM0 內部 OPA 來取代外部分離式裝置，或視需要緩衝內部訊號。MSPM0 OPA 模組極具彈性，可以個別或合併取代感測或控制應用的許多分離式放大器。MSPM0 OPA 模組的主要功能包括在 [表 5-5](#) 中，您可以重新建立的常用 OPA 配置範例均包括在 [OPA 程式碼範例](#) 中

表 5-5. MSPM0 OPA 功能集

特點	MSPM0 實作
輸入類型	軌對軌 (可啟用或停用)
增益頻寬	1 MHz (低功耗模式)
	6 MHz (標準模式)
放大器配置	通用模式
	緩衝模式
	PGA 模式 (反相或非反相)
	差分放大器模式
	級聯放大器模式

表 5-5. MSPM0 OPA 功能集 (continued)

特點	MSPM0 實作
輸入/輸出路由	外部針腳路由
	與 ADC 和 COMP 模組的內部連線
故障偵測	燒斷電流來源 (BCS)
截波穩定	標準 (可選截波頻率)
	ADC 輔助截波
	停用
參考電壓	內部 VREF (僅限 MSPM0G 裝置)
	DAC12 (僅限 MSPM0G 裝置)
	DAC8 (僅限配備 COMP 模組的裝置)

### OPA 程式碼範例

有關 OPA 程式碼範例的資訊可在 [MSPM0 SDK 範例指南](#) 中找到。

### 5.5 電壓參考 (VREF)

STM32G0x 和 MSPM0 都具有內部參考，可用於為內部周邊設備提供參考電壓，並輸出到外部周邊設備。

表 5-6. 功能集比較

特點	STM32G0	MSPM0G	MSPM0L
內部參考 (V)	2.048、2.5	1.4、2.5	1.4、2.5
外部參考 (V)	當 $V_{DD} < 2$ 時, $V_{REF} = V_{DD}$	外部: $1.4 \leq V_{REF} \leq V_{DD}$	外部: $1.4 \leq V_{REF} \leq V_{DD}$
	當 $V_{DD} \geq 2$ 時, $2 \leq V_{REF} \leq V_{DD}$		
輸出內部參考	是	是	是
內部連線至 ADC	是	是	是
內部連線至 DAC	是	是	否
內部連線至 COMP	否	是	否
內部連線至 OPA	不適用	是	否

表 5-7. 控制位元比較

STM32G0x VREFBUF 位元	MSPM0 同等
VREFBUF Bit3 (VRR)	CTL1 Bit0 (就緒)
VREFBUF Bit2 (VRS)	CTL0 Bit7 (BUFCONFIG)
VREFBUF Bit1 (HIZ)	不適用
VREFBUF Bit0 (ENVR)	CTL0 Bit0 (啟用)
	用於取樣和保持模式: CTL0 Bit8 (SHMODE)

對於 MSPM0 VREF，您必須啟用功率位元 PWREN Bit0 (啟用)。

### VREF 程式碼範例

使用 VREF 的程式碼範例可在 [MSPM0 SDK 範例指南](#) 中找到。

## 6 修訂記錄

注：以前版本的页码可能与当前版本的页码不同

日期	修訂	備註
2023 年 3 月	A	第一個公開版本

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2023, Texas Instruments Incorporated