

部署處理特定任務的微控制器來簡化複雜設計

處理特定任務的微控制器可減輕主微控制器或微處理器的任務和工作負荷，進而有助於簡化各種應用的設計流程。

■作者：Robert Perkel

Microchip 8 位元微控制器產品部

如今，運行即時操作系統 (RTOS) 的大型 32 位元微控制器 (MCU) 和微處理器 (MPU) 日益普及。不過，如果使用一個大型微控制器處理複雜的應用，可能會在執行小型幕後處理任務時遇到 CPU 資源方面的問題，這些任務雖然並不複雜，但十分耗時。8 位元和 16 位元 MCU 等小型元件可用於減輕 32 位元元件的工作負荷。

試想一下這樣一個案例：將一個 32 位元 MCU 用於控制汽車的非安全功能，如娛樂系統、環境照明和空調。此 32 位元元件必須對其資源進行分配，以便處理與這些功能相關的所有任務。這樣的任務還包括測量駕駛艙內多個點的溫度、打開 / 關閉空調系統、更新圖形顯示、處理使用者輸入、調整照明條件和播放音樂。即使對於大型 32 位元處理器，這些工作量也過於繁重。

但是，如果 32 位元處理器將部分任務負荷轉移給幾乎不需要監控的子處理器，每個子處理器僅負責處理其中的 1 或 2 個任務，那麼這些任務會更易於管理。這可以釋放主處理器上的 CPU 資源，進而降低軟體的複雜性，同時提高效能並縮短執行時間。

這種解決方案與微控制器中的周邊有異曲同工之妙。周邊是專用硬體的小型模組，可以添加新功能（例如運算放大器或類比數位轉換器），也可以減少執行給定功能時 CPU 必須承擔的工作量。在某些情況下，初始化後，周邊可獨立於 CPU 運行。

為了說明周邊的優勢，我們以產生脈寬調製

(PWM) 信號為例。要在沒有專用周邊的情況下產生 PWM，只需將 I/O 線設為高電位，等待一定數量的週期後，將其設為低電位，再等待一段時間，然後重複操作。這會佔用大量 CPU 週期，並且對於某些功能（如 RTOS）來說，難以可靠地執行。相比之下，PWM 周邊允許 CPU 在執行其他任務的同時設置所需的波形參數。

本文中介紹的第一個案例說明了減輕 CPU 密集型任務負荷的優勢。在該案例中，使用了一個 8 位元 MCU 來創建 I/O 擴展器。I/O 擴展器並不複雜；然而，由於需要頻繁處理中斷，因此它們會佔用大量的 CPU 時間。藉由使用專用 MCU 來完成這項任務，大型 32 位元元件可以減少 I/O 使用和需要處理的中斷次數。此外，I/O 擴展器的功能集可在軟體中設置，因此支援針對應用進行客製和調整。

本文中的第二個範例以創建獨立於 CPU 運行的電壓頻率 (V/F) 轉換器為例，展示了獨立於核心的周邊的效能。在這個範例中，CPU 的唯一功能是初始化周邊並將調試列印消息發送到 UART。在大型系統中，當 V/F 在後臺運行時，CPU 可以執行另一個簡單的任務。

I/O 擴展器

使用 8 位元 MCU 創建 I/O 擴展器的最大好處是提高靈活性。I/O 擴展器 ASIC 的功能集已嵌入到元件中，而 MCU 可基於其執行的軟體定義其行為。

這種靈活性使基於 MCU 的版本能夠滿足最終應用的需求。

實現先進 I/O 擴展器

在元件內部，先進 I/O 擴展器在基於查閱資料表的結構上運行。在讀取或寫入之前，會發送一個虛擬位址。該位址與微控制器上的暫存器無關——僅特定於查閱資料表。這意味著，可以透明地添加不在微控制器硬體暫存器中的功能。此外，還可以針對特定用途，輕鬆地重新排列表格中的條目。這種結構的另一個優勢是，能夠向查閱資料表添加許可權。例如，要創建一個唯讀暫存器，只需省略查閱資料表的寫條目即可。

這種較為複雜的結構也適用於非標準功能。“MEM OP”功能允許 MCU 將其當前的通用輸入和輸出 (GPIO) 配置保存或載入到記憶體中。

MEM OP 也可以將 GPIO 配置重置為編譯時設置的參數。

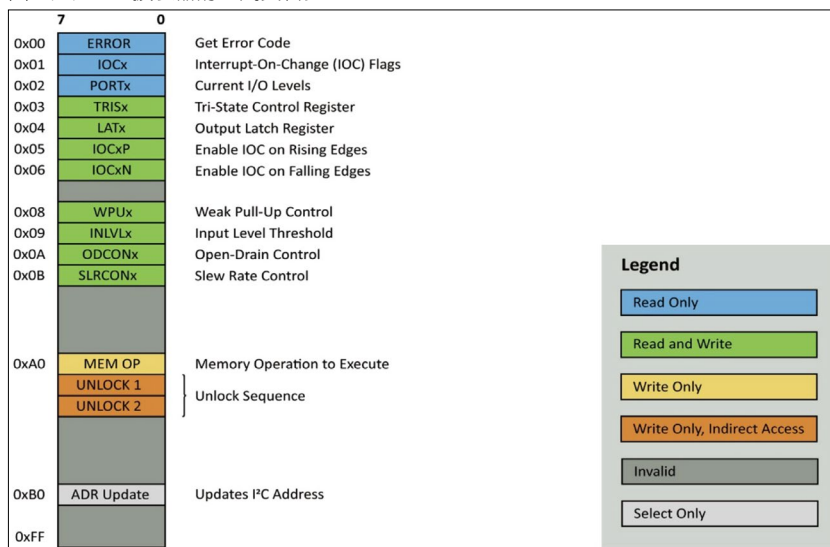
此外，也可以選擇將微控制器設置為在上電時載入保存的設置。如果已啟動，微控制器會嘗試載入配置 0 中的設置。如果配置執行校驗和驗證失敗，則 MCU 將恢復為編譯時常數。如果不需要，可以在軟體中禁用此功能。

該解決方案的要點

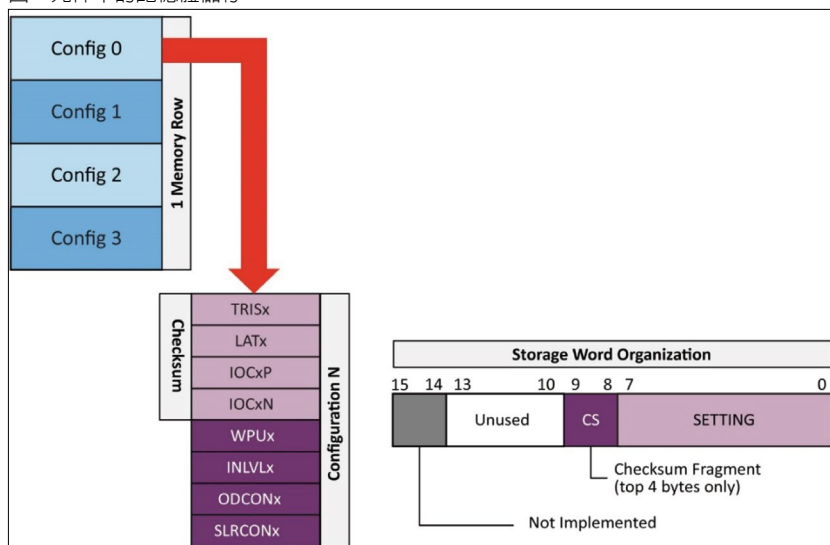
基於 MCU 的解決方案的優勢在於出色的靈活性。與市場上的 ASIC 不同，我們可以為 MCU 配置特定於應用場景的非標準功能。此應用程式針對通用 PIC16F15244 系列 MCU 開發。

如需深入瞭解該實現或想要試用該範例，請參見資源庫中的 README

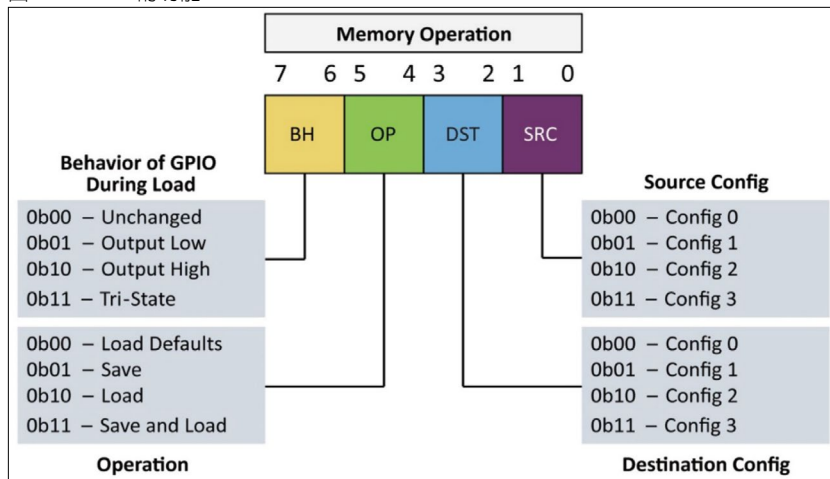
圖：先進 I/O 擴展器的查閱資料表



圖：元件中的記憶體儲存



圖：MEM OP 的功能



注：並非所有欄位均可用於所有操作

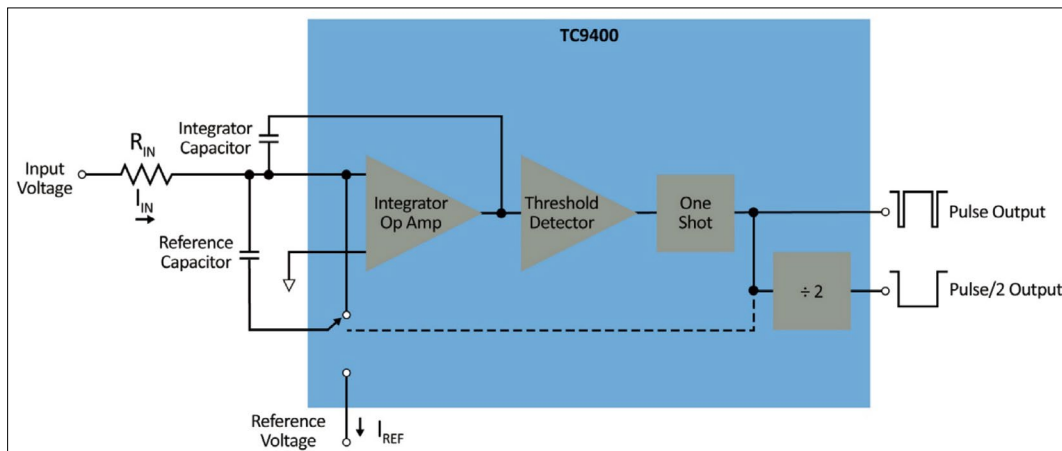
文件。此外，還提供帶有 Arduino 的先進 I/O 擴展器的示範。

原始程式碼、文件和示範：<https://github.com/microchip-pic-avr-examples/pic16f15244-family-advanced-i2c-io-expander>

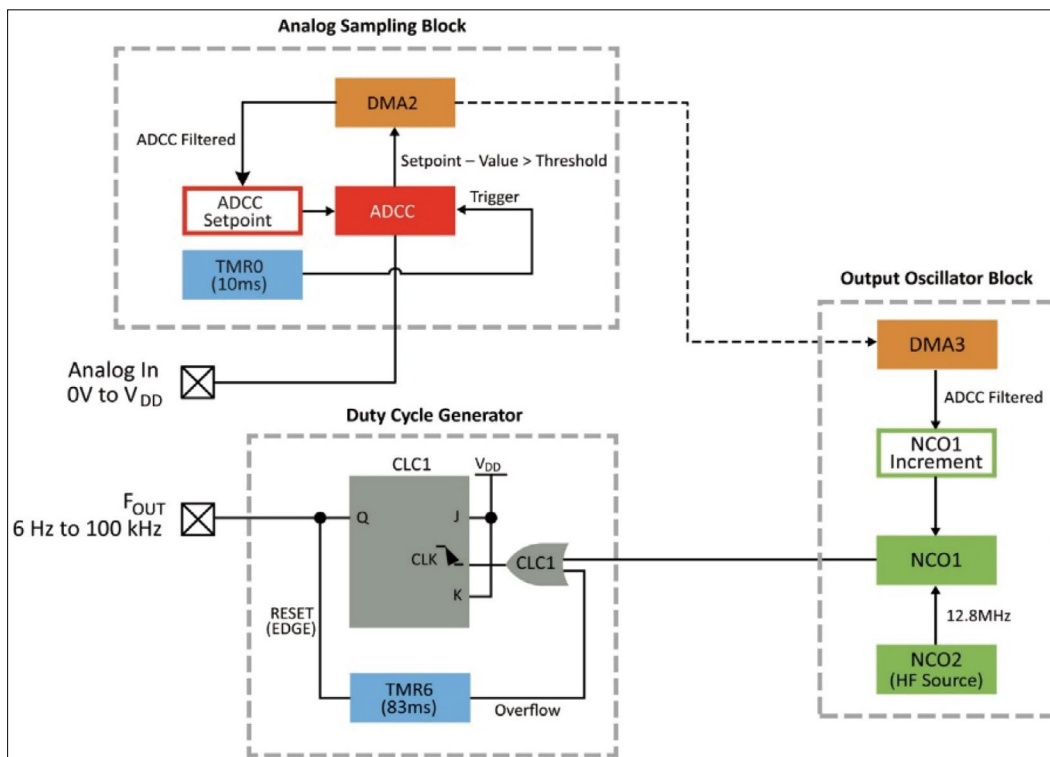
電壓頻率 (V/F) 轉換器

透過降低物料清單 (BOM) 成本，進而減小設計面積，電壓頻率轉換器可改進傳統的類比解決方案。

圖：TC9400/TC9401/TC9402 10 Hz 至 100 kHz V/F 轉換器的應用原理圖



圖：解決方案框圖



市場上的許多 V/F 轉換器需要配備外部電阻和電容才能運行，而微控制器只需使用通用去耦和上拉元件 (所有 MCU 的必備元件) 即可運行。

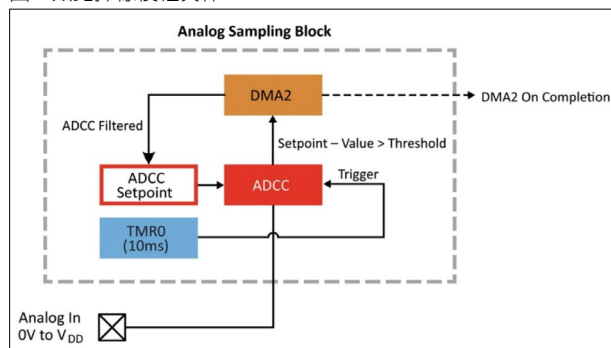
MCU 不使用類比技術進行數位化，而是使用獨立於核心的周邊和功能的組合。MCU 使用內部帶計算功能的類比數位轉換器 (ADCC) 測量輸入信號，然後對時鐘信號進行分頻，以創建可變頻率輸出。在該範例中，周邊已設置為在初始化後獨立於 CPU 運行。這意味著，CPU 可以用於最終應用中的其他任務。

對於基於 MCU 的方案，其挑戰在於效能不如類比解決方案。輸出的解析度本身受到 ADCC 的限制。表面上看，ADCC 為 12 位元，但它會以配置為過取樣的 14 位元解析度運行，具體取決於程式的配置方式。同樣，用於合成輸出頻率的內建數位控振盪器 (NCO) 具有有限的解析度，並且其輸出中可能存在抖動，具體取決於 ADC 測得的值。

基於 MCU 的解決方案可以分為三個不同的周邊模組——類比採樣模組、輸出振盪器模組和工作週期產生器。

類比採樣模組

圖：類比採樣模組實作



類比採樣模組負責執行類比數位轉換。為了在元件的頻率限制下實現 100 kHz 的輸出，已將 ADCC 配置為過取樣，然後透過平均值處理獲得 14 位元結果。

這種過取樣配置有一個缺點，即向結果中增加額外的統計雜訊，可採取計算過取樣的平均值並增加滯後的方法來補償雜訊。要實現滯後，可使用 ADCC 的閾值中斷功能。(為簡單起見，將僅介紹有關此範例如何使用閾值中斷功能的細節。)

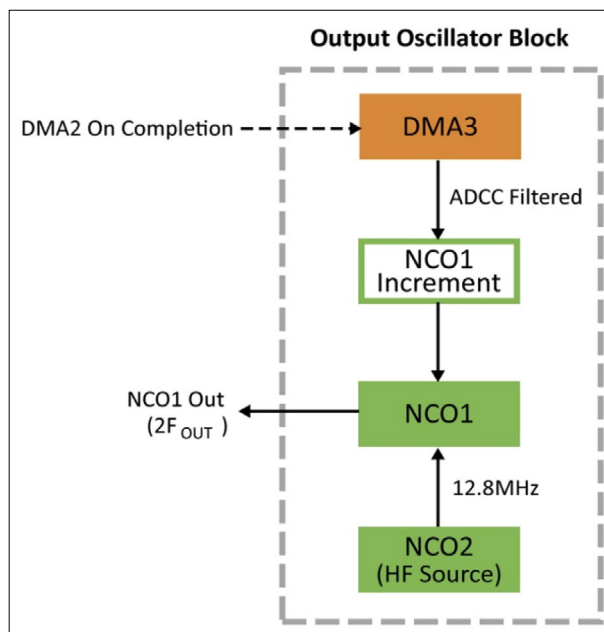
在 ADCC 完成過取樣的平均值計算後，將得出的值與周邊中的設定值暫存器進行比較。如果兩者之差大於或小於設定閾值，則觸發中斷。CPU 可遮罩此中斷且不受影響，然而，此中斷會觸發直接記憶體訪問 (DMA)，將經過平均值處理的過取樣結果複製到 ADCC 的設定值暫存器，進而產生滯後。如果未超過閾值，則不會發生 DMA 複製，進而不會觸發輸出振盪器模組的 DMA 更新。

輸出振盪器模組

該解決方案的輸出振盪器模組負責以所需輸出頻率產生時鐘信號。該輸出信號在內部連接到工作週期產生器，該元件將輸出頻率減半，但會產生 50% 的工作週期輸出。因此，輸出振盪器模組以輸出頻率的兩倍運行。

輸出振盪器模組的核心是數位振盪器 (NCO)。NCO 周邊的工作原理是在輸入時鐘的上升沿向累加器添加增量值，然後根據累加器溢出匯出周邊的輸

圖：輸出振盪器模組的結構



出。(有關 NCO 的完整說明，請參見資料手冊。)

在該範例中，已將 NCO2 設置為在內部創建所需的輸入時鐘頻率，以透過 14 位元輸入獲得 100 kHz 輸出。之所以使用 14 位元結果，是因為 ADCC 本身的 12 位元結果不足以在沒有外部時鐘源的情況下產生 100 kHz 輸出。

表：100 kHz V/F 轉換器的理想輸出 (看門狗已關閉)。

ADC 結果	NCO1 輸出 (翻倍)	輸出頻率
0x0000	0 Hz	0 Hz
0x0001	12.2 Hz	6.1 Hz
0x0100	3.1 kHz	1.6 kHz
0x1000	50 kHz	25 kHz
0x3FFF	200 kHz	100 kHz

如果改變 NCO2 的輸出頻率或使用備用源，則輸出頻率將調整為不同的輸出範圍。例如，如果

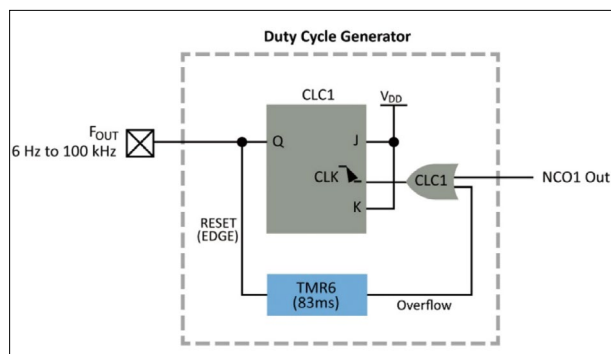
表：10 kHz V/F 轉換器的理想輸出 (看門狗已關閉)。

ADC 結果	NCO1 輸出頻率 (翻倍)	輸出頻率
0x0000	0 Hz	0 Hz
0x0001	12.2 Hz	0.6 Hz
0x0100	312.5 kHz	156.3 kHz
0x1000	5 kHz	2.5 kHz
0x3FFF	20 kHz	10 kHz

NCO2 的頻率降低到 1.28 MHz，則輸出最大為 10 kHz。

工作週期產生器

圖：工作週期產生器框圖



該解決方案的工作週期產生器模組負責創建 50% 的工作週期輸出。這是一個可選功能——可以直接使用 NCO 的輸出，但這樣做會增加工作週期的變化幅度。

該生成器使用一個可配置邏輯單元 (CLC) 實現。CLC 是可配置邏輯的小型模組，類似於現場可程式設計閘陣列 (FPGA) 的一個單元。CLC 可用作離散邏輯門 (例如 AND-OR 或 OR-XOR)，也可以配置為鎖存器或觸發器。在該解決方案中，CLC 實現為帶復位功能的 J-K 觸發器。J 和 K 保持在邏輯高電位。輸出振盪器模組用作觸發器的時鐘。每個輸入時鐘脈衝均會導致輸出翻轉，進而產生 50% 的工作週期。注意：輸出振盪器模組的頻率抖動將對工作週期產生影響。

Timer 6 用作不穩定的“看門狗”計時器。如果

輸出沒有產生邊沿 (上升沿或下降沿)，則計時器將溢出，並將產生的時鐘脈衝發送到 CLC，這可以控制輸出頻率範圍的下限。輸出翻轉到計時器頻率的一半 (輸出為 6 Hz)，而不是達到直流。

該解決方案的要點

該範例表明，要使用硬體周邊創建獨立於核心的功能，通常必須使用外部積體電路。這種配置的一個最大優勢在於，周邊操作可在軟體中設置，這樣便可輕鬆地根據最終應用調整範例。由於使用了大量周邊，因此選擇 PIC18-Q43 系列 MCU 來實現該範例。

有關該範例的更多資訊，請參見範例資源庫中的 README 文件。此外，範例資源庫還包含頻率電壓轉換器的實例，可與電壓頻率轉換器在同一個元件上實現。

按一下以下連結獲取原始程式碼和文件：
<https://github.com/microchip-pic-avr-examples/pic18f57q43-v-to-f-mplab-mcc>

總結

儘管高效能微控制器和微處理器都有一席之地，但在執行小型專門任務時，8 位元和 16 位元 MCU 的作用不容小覷。這類任務並不一定十分複雜，但可能十分耗時，或者是時間關鍵型任務。任務負荷減輕後，32 位元元件可擁有更簡單的實現，進而提高可靠性、減少記憶體佔用率並降低功耗。

CTA

COMPOTECHAsia 臉書

每週一、三、五與您分享精彩内容

<https://www.facebook.com/lookcompotech>