部署处理特定任务的单片机来简化复杂设计

Microchip Technology Inc.

8 位单片机产品部

Robert Perkel

摘要：处理特定任务的单片机可减轻主单片机或微处理器的任务和工作负荷，从而有助于简化各种应用的设计流程。

如今，运行实时操作系统（RTOS）的大型 32 位单片机（MCU）和微处理器（MPU）日益普及。不过，如果使用一个大型单片机处理复杂的应用，可能会在执行小型后台处理任务时遇到 CPU 资源方面的问题，这些任务虽然并不复杂，但十分耗时。8 位和 16 位 MCU 等小型器件可用于减轻 32 位器件的工作负荷。

试想一下这样一个示例：将一个 32 位 MCU 用于控制汽车的非安全功能，如娱乐系统、环境照明和空调。此 32 位器件必须对其资源进行分配，以便处理与这些功能相关的所有任务。这样的任务还包括测量驾驶室内多个点的温度、打开/关闭空调系统、更新图形显示、处理用户输入、调整照明条件和播放音乐。即使对于大型 32 位器件，这些工作量也过于繁重。

但是，如果 32 位器件将部分任务负荷转移给几乎不需要监控的子处理器，每个子处理器仅负责处理其中的 1 或 2 个任务，那么这些任务会更易于管理。这可以释放主处理器上的 CPU 资源，从而降低软件的复杂性，同时提高性能并缩短执行时间。

这种解决方案与单片机中的外设有异曲同工之妙。外设是专用硬件的小型模块，可以添加新功能（例如运算放大器或模数转换器），也可以减少执行给定功能时 CPU 必须承担的工作量。在某些情况下，初始化后，外设可独立于 CPU 运行。

为了说明外设的优势，我们以产生脉宽调制（PWM）信号为例。要在没有专用外设的情况下产生 PWM，只需将 I/O 线设为高电平，等待一定数量的周期后，将其设为低电平，再等待一段时间，然后重复操作。这会占用大量 CPU 周期，并且对于某些功能（如 RTOS）来说，难以可靠地执行。相比之下，PWM 外设允许 CPU 在执行其他任务的同时设置所需的波形参数。

本文中介绍的第一个示例说明了减轻 CPU 密集型任务负荷的优势。在该案例中，使用了一个 8 位 MCU 来创建 I/O 扩展器。I/O 扩展器并不复杂；然而，由于需要频繁处理中断，因此它们会占用大量的 CPU 时间。通过使用专用 MCU 来完成这项任务，大型 32 位器件可以减少 I/O 使用和需要处理的中断次数。此外，I/O 扩展器的功能集可在软件中设置，因此支持针对应用进行定制和调整。

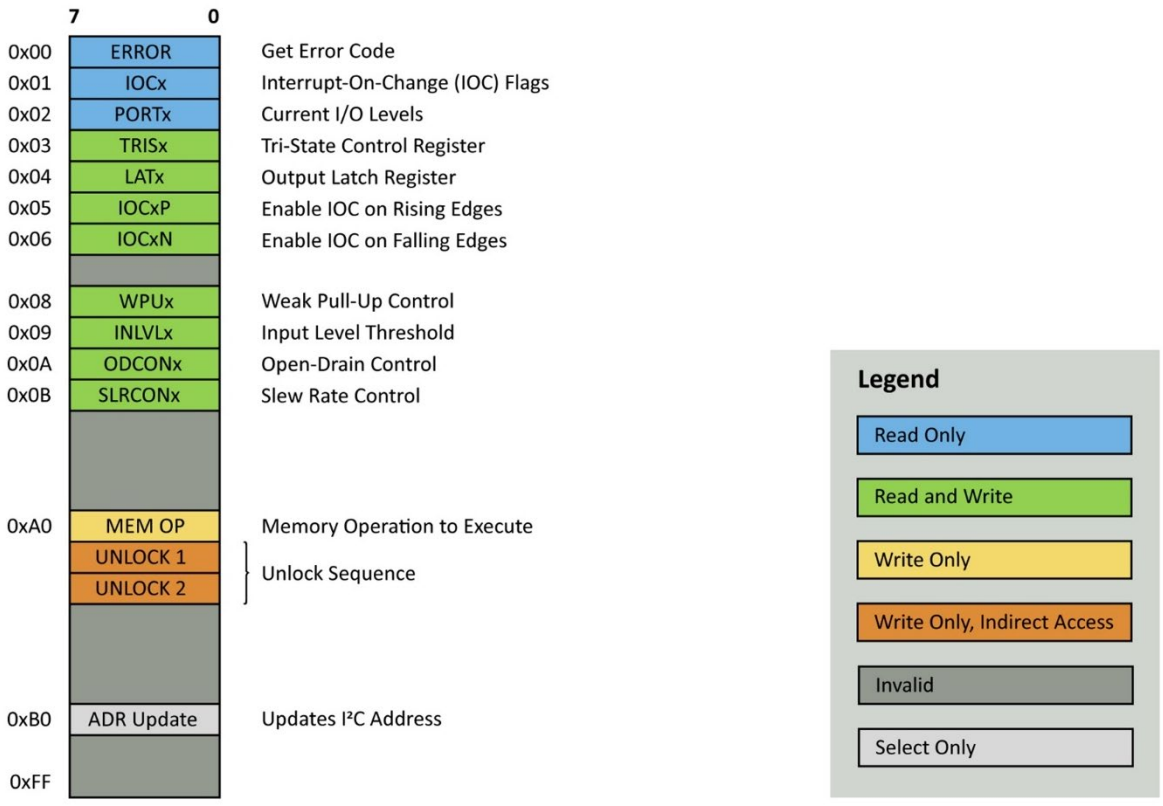
本文中的第二个示例以创建独立于 CPU 运行的电压频率（V/F）转换器为例，展示了独立于内核的外设的性能。在这个示例中，CPU 的唯一功能是初始化外设并将调试打印消息发送到 UART。在大型系统中，当 V/F 在后台运行时，CPU 可以执行另一个简单的任务。

I/O 扩展器

使用 8 位 MCU 创建 I/O 扩展器的最大好处是提高灵活性。I/O 扩展器 ASIC 的功能集已嵌入到器件中，而 MCU 可基于其执行的软件定义其行为。这种灵活性使基于 MCU 的版本能够满足最终应用的需求。

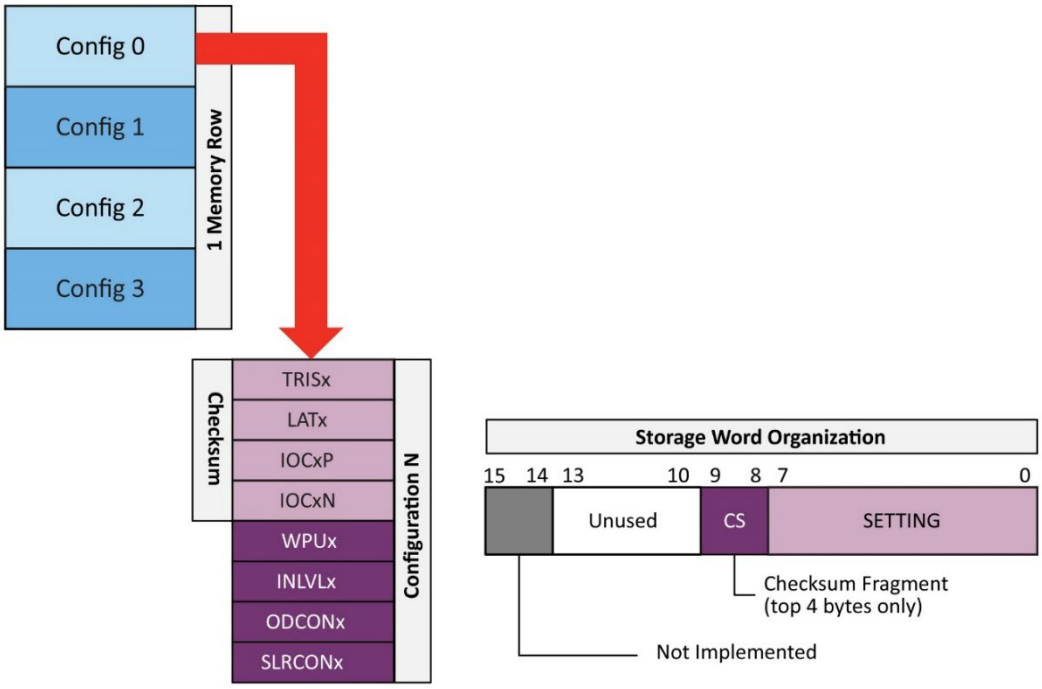
实现高级 I/O 扩展器

在器件内部，高级 I/O 扩展器在基于查找表的结构上运行。在读取或写入之前，会发送一个虚拟地址。该地址与单片机上的寄存器无关——仅特定于查找表。这意味着，可以透明地添加不在单片机硬件寄存器中的功能。此外，还可以针对特定用途，轻松地重新排列表格中的条目。这种结构的另一个优势是，能够向查找表添加权限。例如，要创建一个只读寄存器，只需省略查找表的写条目即可。



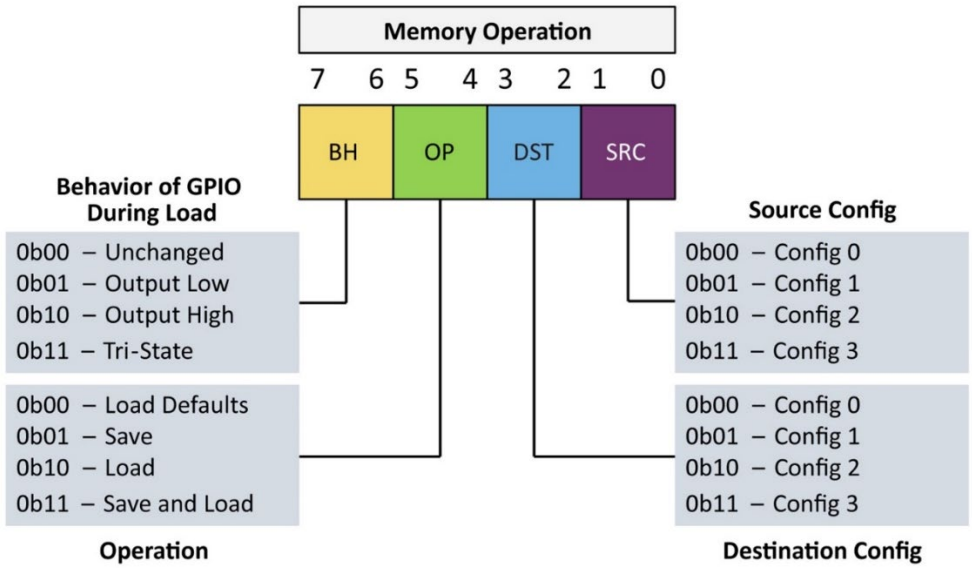
高级 I/O 扩展器的查找表

这种较为复杂的结构也适用于非标准功能。“MEM OP”功能允许 MCU 将其当前的通用输入和输出（GPIO）配置保存或加载到存储器中。



器件中的存储器存储

MEM OP 也可以将 GPIO 配置重置为编译时设置的参数。



注：并非所有字段均可用于所有操作

MEM OP 的功能

此外，也可以选择将单片机设置为在上电时加载保存的设置。如果已使能，单片机会尝试加载配置 0 中的设置。如果配置执行校验和验证失败，则 MCU 将恢复为编译时常数。如果不需要，可以在软件中禁用此功能。

该解决方案的要点

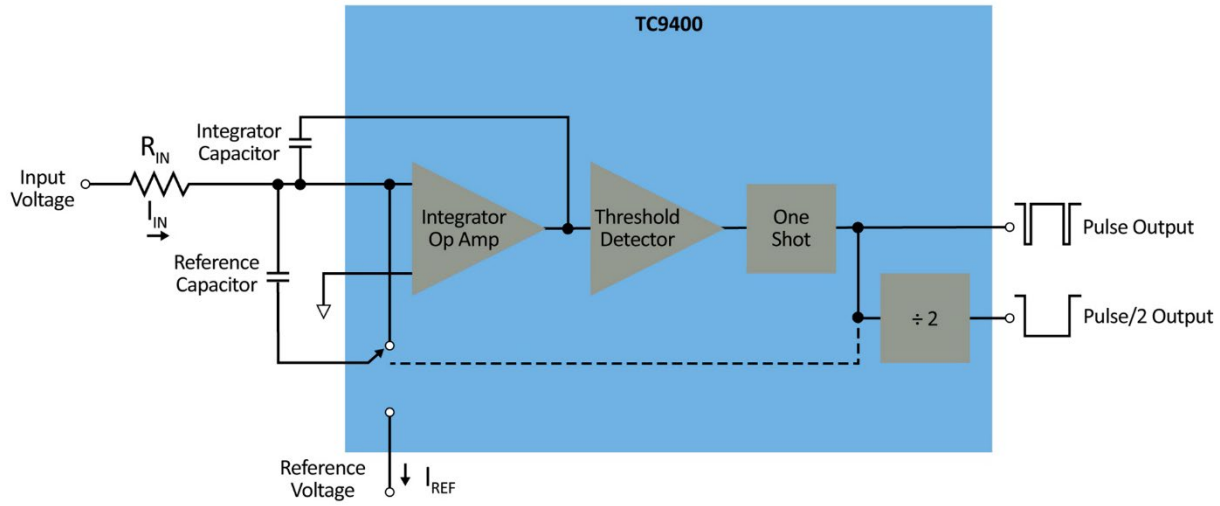
基于 MCU 的解决方案的优势在于出色的灵活性。与市场上的 ASIC 不同，我们可以为 MCU 配置特定于应用场景的非标准功能。此应用程序针对通用 PIC16F15244 系列 MCU 开发。

如需深入了解该实现或想要试用该示例，请参见源资源库中的 README 文件。此外，还提供带有 Arduino 的高级 I/O 扩展器的演示。

源代码、文档和演示：https://github.com/microchip-pic-avr-examples/pic16f15244-family-advanced-i2c-io-expander

电压频率（V/F）转换器

通过降低物料清单（BOM）成本，进而减小设计面积，电压频率转换器可改进传统的模拟解决方案。市场上的许多 V/F 转换器需要配备外部电阻和电容才能运行，而单片机只需使用通用去耦和上拉组件（所有 MCU 的必备组件）即可运行。

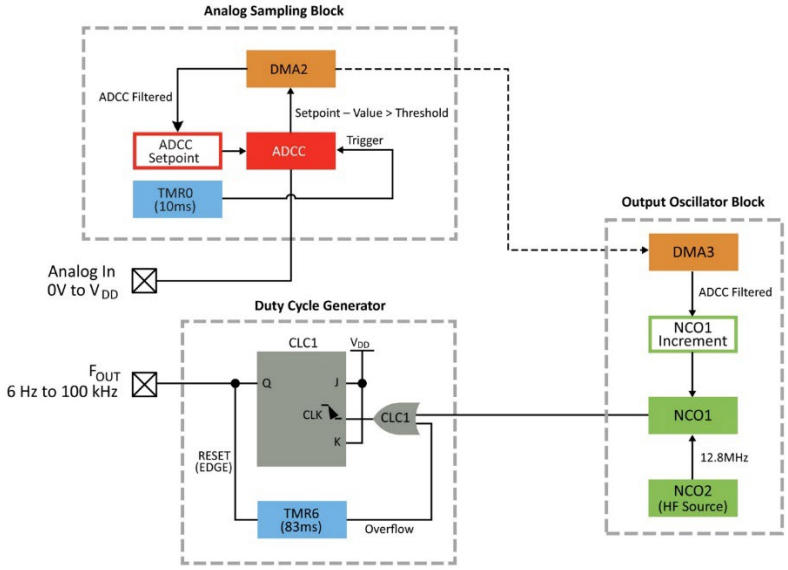


TC9400/TC9401/TC9402 10 Hz 至 100 kHz V/F 转换器的应用原理图

MCU 不使用模拟技术进行数字化，而是使用独立于内核的外设和功能的组合。MCU 使用内部带计算功能的模数转换器（ADCC）测量输入信号，然后对时钟信号进行分频，以创建可变频率输出。在该示例中，外设已设置为在初始化后独立于 CPU 运行。这意味着，CPU 可以用于最终应用中的其他任务。

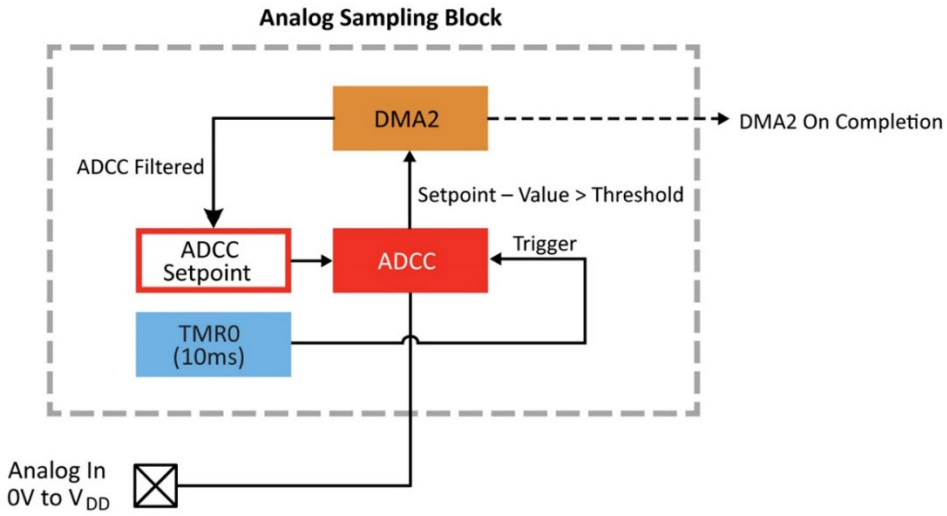
对于基于 MCU 的方案，其挑战在于性能不如模拟解决方案。输出的分辨率本身受到 ADCC 的限制。表面上看，ADCC 为 12 位，但它会以配置为过采样的 14 位分辨率运行，具体取决于程序的配置方式。同样，用于合成输出频率的片内数控振荡器（NCO）具有有限的分辨率，并且其输出中可能存在抖动，具体取决于 ADC 测得的值。

基于 MCU 的解决方案可以分为三个不同的外设模块——模拟采样模块、输出振荡器模块和占空比发生器。



解决方案框图

模拟采样模块



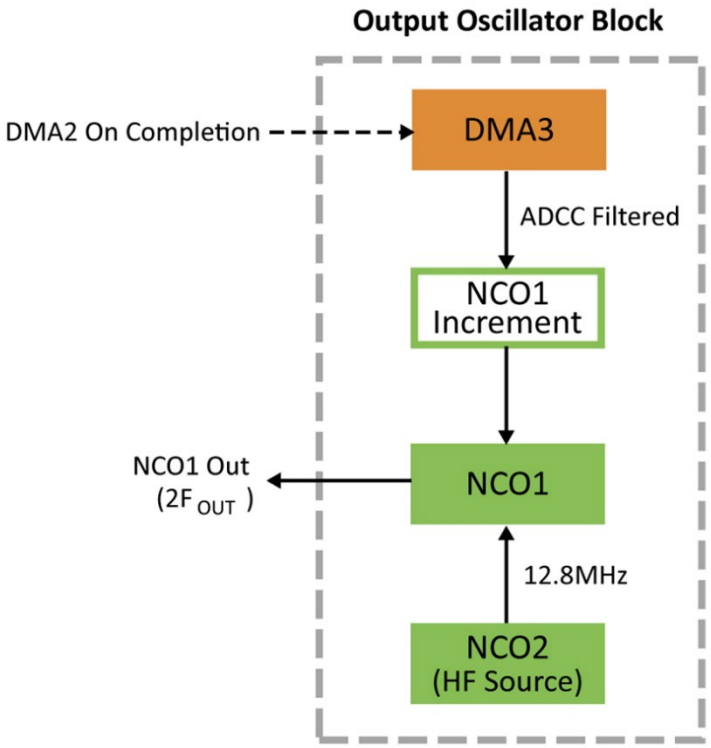
模拟采样模块实现

模拟采样模块负责执行模数转换。为了在器件的频率限制下实现 100 kHz 的输出，已将 ADCC 配置为过采样，然后通过平均值处理获得 14 位结果。

这种过采样配置有一个缺点，即向结果中增加额外的统计噪声，可采取计算过采样的平均值并增加滞后的方法来补偿噪声。要实现滞后，可使用 ADCC 的阈值中断功能。（为简单起见，将仅介绍有关此示例如何使用阈值中断功能的细节。）

在 ADCC 完成过采样的平均值计算后，将得出的值与外设中的设定值寄存器进行比较。如果两者之差大于或小于设定阈值，则触发中断。CPU 可屏蔽此中断且不受影响，然而，此中断会触发直接存储器访问（DMA），将经过平均值处理的过采样结果复制到 ADCC 的设定值寄存器，从而产生滞后。如果未超过阈值，则不会发生 DMA 复制，从而不会触发输出振荡器模块的 DMA 更新。

输出振荡器模块



输出振荡器模块的结构

该解决方案的输出振荡器模块负责以所需输出频率产生时钟信号。该输出信号在内部连接到占空比发生器，该元件将输出频率减半，但会产生 50%的占空比输出。因此，输出振荡器模块以输出频率的两倍运行。

输出振荡器模块的核心是数控振荡器（NCO）。NCO 外设的工作原理是在输入时钟的上升沿向累加器添加增量值，然后根据累加器溢出导出外设的输出。（有关 NCO 的完整说明，请参见数据手册。）

在该示例中，已将 NCO2 设置为在内部创建所需的输入时钟频率，以通过 14 位输入获得 100 kHz 输出。之所以使用 14 位结果，是因为 ADCC 本身的 12 位结果不足以在没有外部时钟源的情况下产生 100 kHz 输出。

ADC 结果	NCO1 输出（翻倍）	输出频率
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

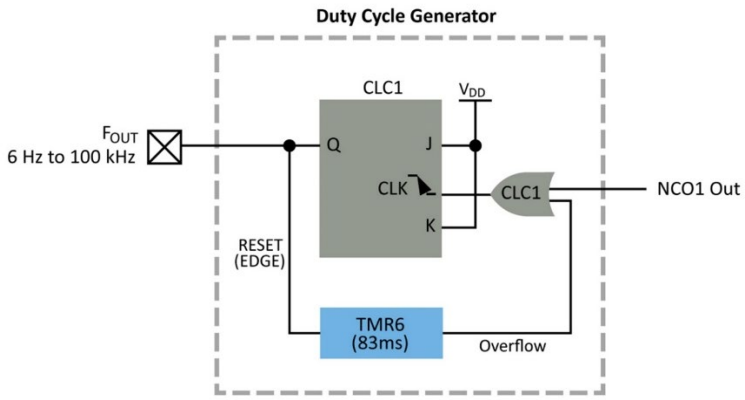
*100 kHz V/F 转换器的理想输出（**看门狗已关闭**）。*

如果改变 NCO2 的输出频率或使用备用源，则输出频率将调整为不同的输出范围。例如，如果 NCO2 的频率降低到 1.28 MHz，则输出最大为 10 kHz。

ADC 结果	NCO1 输出频率（翻倍）	输出频率
0x0000	0 Hz	0 Hz
0x0001	1.2 Hz	0.6 Hz
0x0100	312.5 Hz	156.3 Hz
0x1000	5 kHz	2.5 kHz
0x3FFF	20 kHz	10 kHz

*10 kHz V/F 转换器的理想输出（**看门狗已关闭**）。*

占空比发生器



占空比发生器框图

该解决方案的占空比发生器模块负责创建 50%的占空比输出。这是一个可选功能——可以直接使用 NCO 的输出，但这样做会增加占空比的变化幅度。

该生成器使用一个可配置逻辑单元（CLC）实现。CLC 是可配置逻辑的小型模块，类似于现场可编程门阵列（FPGA）的一个单元。CLC 可用作离散逻辑门（例如 AND-OR 或 OR-XOR），也可以配置为锁存器或触发器。在该解决方案中，CLC 实现为带复位功能的 J-K 触发器。J 和 K 保持在逻辑高电平。输出振荡器模块用作触发器的时钟。每个输入时钟脉冲均会导致输出翻转，从而产生 50%的占空比。注意：输出振荡器模块的频率抖动将对占空比产生影响。

Timer 6 用作不稳定的“看门狗”定时器。如果输出没有产生边沿（上升沿或下降沿），则定时器将溢出，并将产生的时钟脉冲发送到 CLC，这可以控制输出频率范围的下限。输出翻转到定时器频率的一半（输出为 6 Hz），而不是达到直流。

该解决方案的要点

该示例表明，要使用硬件外设创建独立于内核的功能，通常必须使用外部集成电路。这种配置的一个最大优势在于，外设操作可在软件中设置，这样便可轻松地根据最终应用调整示例。由于使用了大量外设，因此选择 PIC18-Q43 系列 MCU 来实现该示例。

有关该示例的更多信息，请参见示例资源库中的 README 文档。此外，示例资源库还包含频率电压转换器的实现，可与电压频率转换器在同一个器件上实现。

单击以下链接获取源代码和文档：https://github.com/microchip-pic-avr-examples/pic18f57q43-v-to-f-mplab-mcc

总结

尽管高性能单片机和微处理器都有一席之地，但在执行小型专门任务时，8 位和 16 位 MCU 的作用不容小觑。这类任务并不一定十分复杂，但可能十分耗时，或者是时间关键型任务。任务负荷减轻后，32 位器件可拥有更简单的实现，从而提高可靠性、减少存储器占用率并降低功耗。