

## mTouch™ 触摸传感解决方案采集方法 电容分压器

作者: *Burke Davison*  
*Microchip Technology Inc.*

### 简介

将 PIC® MCU 的引脚通过串联电阻（可选）连接到一块由导电材料制成的区域即可构成电容传感器。随着传感器周围环境的变化，导电材料相对于地的电容将发生改变。尽管有许多种方法可测量引脚电容，但大多数都需要特殊硬件或高级数字滤波系统才能获得干净的信号。

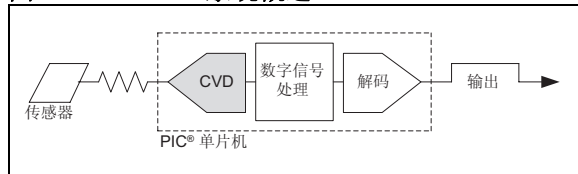
Microchip 开发的差分电容分压器（Capacitive Voltage Divider, CVD）采集技术仅需要一个模数转换器（Analog-to-Digital Converter, ADC）以及最少量的数字处理开销。这使得 CVD 可在最广泛的器件上实现。

本应用笔记将介绍如何实现 mTouch™ 触摸传感解决方案 CVD 电容检测方法，分析其信号相对于环境变化的行为方式，并定义几种可选的性能增强功能来提高灵敏度以及减少输出中的噪声。

本应用笔记提供的代码仅供教学使用。对于所有需要可靠噪声抑制的实际应用，强烈建议使用 Microchip 应用程序库（MLA，<http://www.microchip.com/mla>）中提供的 mTouch 触摸传感框架和库。

CVD 并不是惟一可用于测量 PIC 器件电容的技术。我们的网站（[www.microchip.com/mTouch](http://www.microchip.com/mTouch)）上提供了有关充电时间测量单元备选检测方法的应用笔记（AN1250，《Microchip 用于电容触摸应用的 CTMU》）以及硬件和软件设计准则（AN1334，“Techniques for Robust Touch Sensing Design”）。

图 1: CVD 系统概述



### 基本电容触摸概述

电容传感器最常见的设计方式是在印刷电路板上放置一块金属填充的区域，但也可以简单到用一片铝箔实现。然后通过一条细走线和一个可选串联电阻将该导电焊盘与 PIC 器件连接。如图 1 所示，PIC 器件将不断轮询焊盘的电容并等待出现重大的转变。传感器在扫描的测量阶段处于高阻态，而在所有其他阶段处于低阻态。

“重大”的定义取决于噪声等级。电容的变化量必须明显高于最坏情况下的噪声等级。如果应用环境可能会迅速变化，则电容变化量必须高于由环境引起的最大可能变化。因此，整体噪声是来自各种可能源的高频和低频干扰的组合。

为此，电容传感器信号的质量应该始终采用公式 1 中定义的信噪比（signal-to-noise ratio, SNR）来定义，而不是简单地用激活时的信号变化来定义。

### 公式 1: 信噪比

$$SNR = \frac{\mu}{\sigma}$$

$\mu$  是激活时的信号变化量  
 $\sigma$  是噪声的标准偏差

本应用笔记介绍了 CVD 测量技术如何将传感器的电容转换为用于数字处理的整型值。但是，Microchip 不建议手动实现 CVD！Microchip 应用程序库中提供的 mTouch 触摸传感框架和 mTouch 触摸传感库会自动实现扫描，并且经测试可提供高级抗噪声性能。强烈建议使用这些资源而不要从头实现扫描。

## 电容分压器概述

CVD 是一项基于电荷 / 电压的技术，仅使用模数转换器 (ADC) 模块即可测量引脚上的相对电容。由于只需要一个常见的 PIC 器件外设，因此此技术可在大多数器件上实现。

此技术根据内部 ADC 采样和保持电容的大小来执行相对电容测量。PIC 器件的电气规范将定义此电容的典型值；但是，由于制造公差缘故，该典型值可能有最多 20% 的变化。因此，不建议使用 CVD 来生成绝对测量值，除非已进行校准并可确保环境条件不会发生变化。触摸和接近传感应用只需要一个相对测量值。这样便可对环境变化进行跟踪和过滤，无需进行校准。

## 检测方法的优势

CVD 技术在实际应用中性能出众有以下几个原因。这些特性可提高最终触摸决策的可靠性并最大程度地降低电容触摸集成的成本。

- 温度依赖性低

信号在  $-20^{\circ}\text{C}$  至  $+60^{\circ}\text{C}$  范围内偏移量的典型值为 1-3%。通常根据传感器值的缓慢变化在软件中将此偏移量移除。

- VDD 依赖性低

CVD 波形对 VDD 的依赖性不大，因为传感器的电荷和 ADC 的正参考电压使用的值相同。因此，VDD 中的低频变化会大幅衰减。但是，VDD 中的高频干扰可能产生无用的信号噪声。

- 硬件要求极低

建议选用串联电阻来降低信号上的高频噪声。如果噪声无关紧要，则无需任何外部元件。

- 低频噪声抑制

低频噪声导致的偏移量将在同一方向上影响 CVD 波形的两个 ADC 采样。然而，增大电容将在相反的方向上对它们产生影响。减去这两个采样将使信号强度加倍，同时消除噪声偏移量。

**注：** 波形从根本上抑制的频率范围将取决于两个采样之间的延时。两个采样越接近，噪声抑制的带宽越大。

并非每种电容检测技术都具有这些优势，这使得 CVD 独树一帜，成为触摸应用的理想选择。

## 工作原理

由于时序对于传感器的最终 SNR 很重要，因此仅建议使用汇编编程方法来手动实现 CVD。Microchip 在其应用程序库中提供了实现扫描所需的库，强烈建议使用此库而不要从头实现扫描。可以从我们的网站 [www.microchip.com/mla](http://www.microchip.com/mla) 下载该库。本应用笔记中还针对几种可能的波形配置提供了汇编示例。

电容传感器与 PIC 器件的一个模拟引脚相连接。可在电路中放置一个可选的串联电阻以创建低通滤波器，使信号上的高频噪声衰减。随后只需控制输入/输出端口和 ADC 便可执行采样。

### 步骤 1：预充电电容

将两个电容充电至相反的电压。第一次执行为“采样 A”。第二次（在步骤 4-6 中介绍）为“采样 B”。如图 4 所示。

采样 A:

- 外部传感器放电至 VSS。
- 内部传感器充电至 VDD。

采样 B:

- 外部传感器充电至 VDD。
- 内部传感器放电至 VSS。

步骤 2：连接电容并使其稳定

将两个电容并联并使电荷稳定。随着外部电容的增大，初始电荷也会增加（公式 2）。内部电容不发生变化，因此其电荷保持恒定。此步骤如图 5 所示。

步骤 3：ADC 转换

$C_{hold}$  上的最终电压由外部电容大小与内部电容大小的关系决定（公式 3）。

步骤 4-6：反转预充电电压并重复

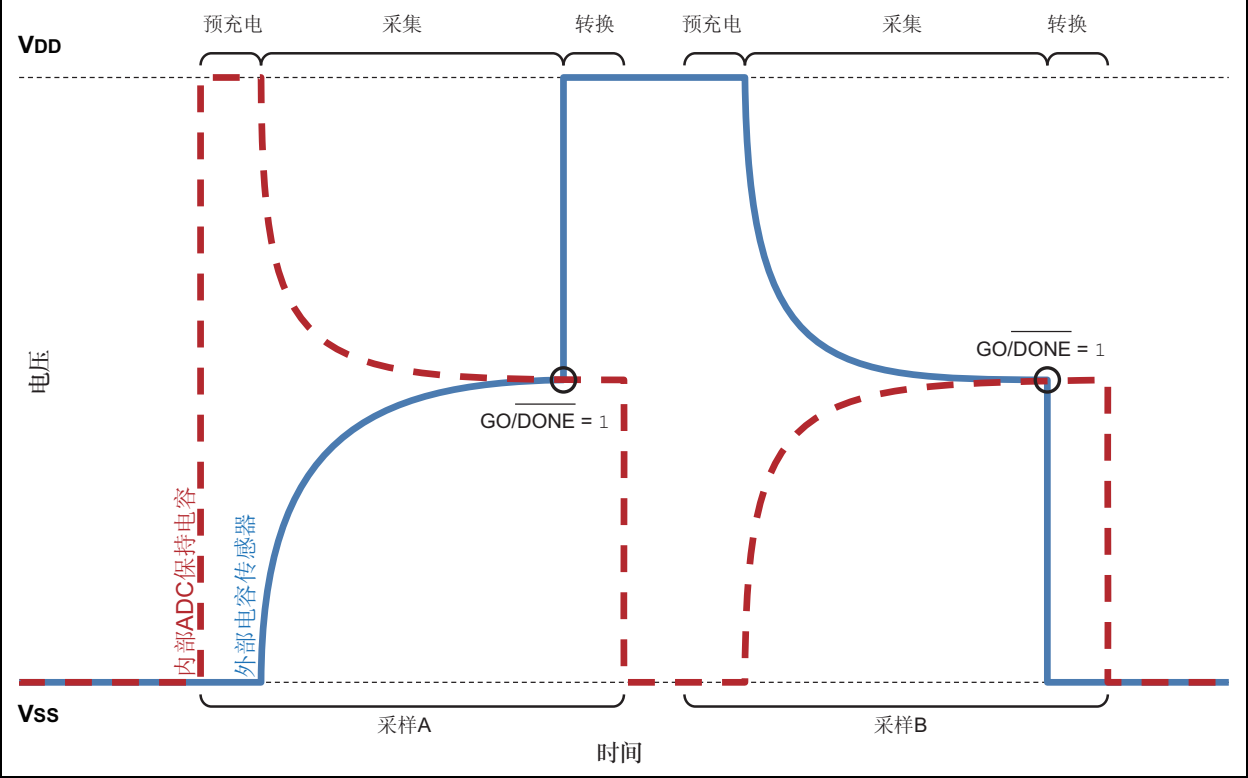
然后，再次执行上述操作，但这次将预充电电压反转。  
两次结果的差值用作当前传感器读数。这就是该扫描技术通常被称为“差分 CVD”的原因。差分 CVD 检测方法的完整波形如图 2 所示。

当用户接近电容传感器时，外部电容相对于内部电容的大小将增大。这将导致两次采样的稳定电压发生变化。

对于采样 A，当外部传感器的电容增大并放电至  $V_{ss}$  时，最终稳定电压将减小。对于采样 B，当外部传感器的电容增大并充电至  $V_{DD}$  时，最终稳定电压将增大。因此，随着外部电容的增大，CVD 波形的两个稳定点将发生偏移，进而导致传感器读数发生变化。

噪声将导致稳定电压发生变化，具体取决于噪声信号的相位。两次采样的低频噪声相位近似相同。由于噪声对两次采样的影响大致相同，我们可通过两个稳定点之间的差值使信号中的噪声大幅衰减，从而使噪声偏移量的影响自行抵消。随着噪声频率的增大，噪声抑制的质量会降低，这使得进一步滤波技术变得必要。随着两次采样之间的时间间隔的缩短，抑制高频噪声的能力会增强。这是采用汇编语言（而不是 C 语言）可以提高检测方法抗噪声性能的原因之一。

图 2： 差分 CVD 波形



[illegible]

将其从公书中移除。这一假定 (M2) 简写

\_\_\_\_\_

--

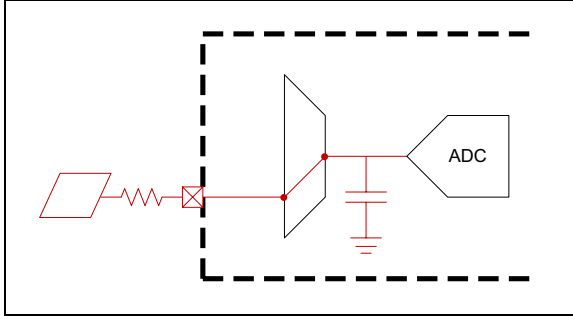
\_\_\_\_\_

[illegible]

## 采集阶段

两个电容此时已充电至相反电压状态，并且并联在一起，如图5所示。这会将两个电容的电荷合计在一起，如公式3中定义。根据  $C_{hold}$  与外部电容的关系使电容两端的电压相等，为一中间值。

图 5: 采集阶段



由于两个电容此时处于并联状态，我们可以将它们的值合并来获取电路上的总电容。

$$C_{total} = C_{hold} // C_{base} = C_{hold} + C_{base}$$

两个电容之间的总电荷是各电容电荷的总和。

$$Q_{total} = C_{hold}V_{hold} + C_{base}V_{base}$$

## 公式 3: 采集阶段

$$V_{settle} = \frac{C_{hold}V_{hold} + C_{base}V_{base}}{C_{hold} + C_{base}}$$

$Q_{total}$  为采集阶段两个电容连接后的总电荷。

$C_{total}$  为采集阶段中内部电容和外部电容连接并共享电荷时电路的总电容。

$V_{settle}$  为标准 CVD 波形采集阶段中的最终稳定电压。

## 差分结果

上面的稳定电压公式为一般形式，适用于第一次和第二次差分采样。对于第一次采样（“A”），将 0 代入  $V_{base}$ ， $V_{DD}$  代入  $V_{ADC}$  来计算实际稳定电压。对于第二次采样（“B”），将  $V_{DD}$  代入  $V_{base}$ ，0 代入  $V_{ADC}$  来计算实际稳定电压。

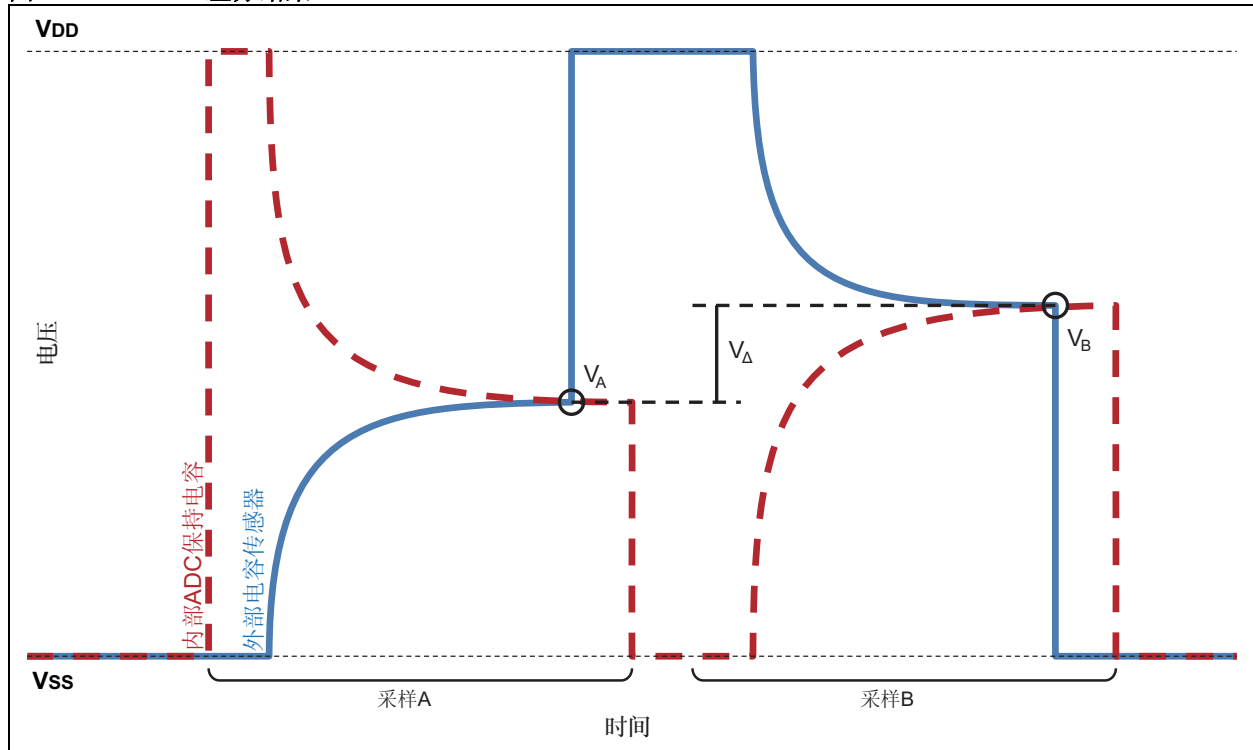
$$V_A = V_{settle}\{V_{hold} = V_{DD}, V_{base} = 0\}$$

$$V_B = V_{settle}\{V_{hold} = 0, V_{base} = V_{DD}\}$$

然后计算这两个电压的差值来获得传感器的读数。实际上， $V_B$  的值通常比  $V_A$  大，因此应调整减法的顺序以生成正结果。

$$V_{\Delta} = V_B - V_A$$

图 6: CVD 差分结果



**注：** 在软件中，建议将  $V_B$  的值偏移  $2^N$ （其中  $N$  为 ADC 的位数），以进一步确保不会得到负结果。对于上述计算，已忽略此偏移以简化数学公式。

$$V_{\Delta released} = V_{settle}(V_{hold} = 0, V_{base} = V_{DD}) - V_{settle}(V_{hold} = V_{DD}, V_{base} = 0)$$

$$\frac{V_{\Delta released}}{V_{DD}} = \frac{C_{base}}{C_{hold} + C_{base}} + \frac{C_{hold}}{C_{hold} + C_{base}}$$

#### 公式 4： 差分结果

两个 CVD 稳定点之间的电压差

$$\frac{V_{\Delta released}}{V_{DD}} = \frac{C_{base} - C_{hold}}{C_{hold} + C_{base}}$$

#### 添加手指电容

现在，执行相同的分析，但在电路中增加一个电容：用户的手指。这会更改外部电容和总电容。

$$C_{external, pressed} = C_{base} + C_{finger}$$

$$C_{total, pressed} = C_{hold} + C_{base} + C_{finger}$$

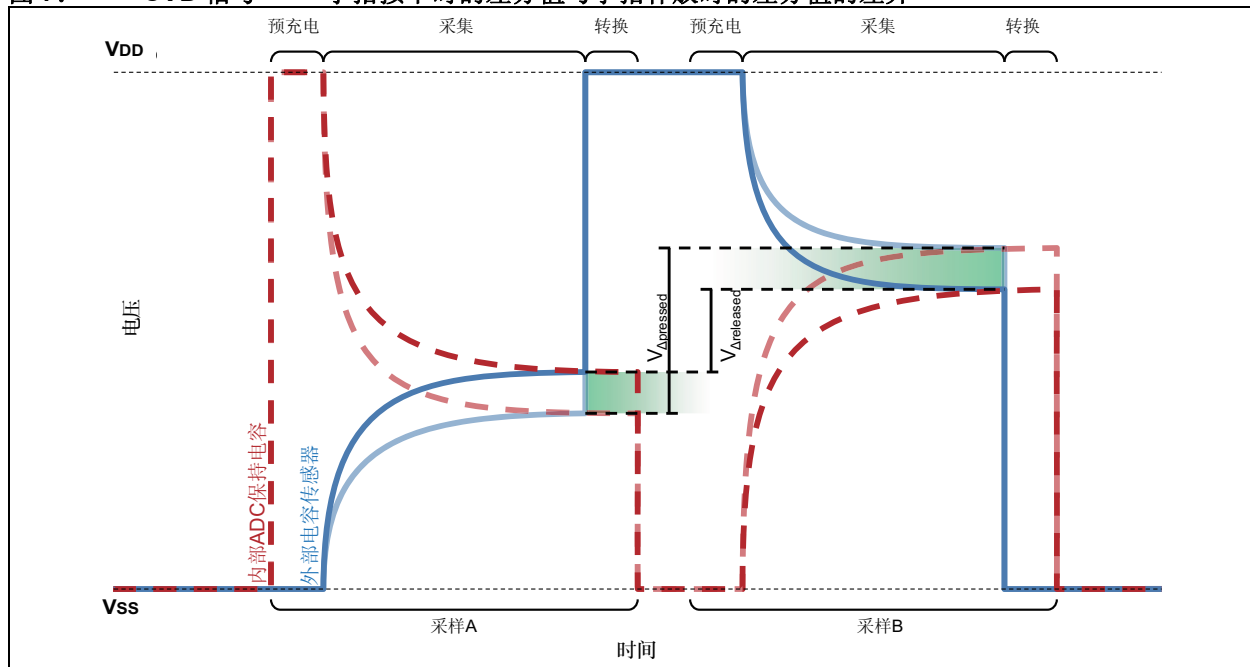
计算当手指触摸传感器时 CVD 稳定电压的一般形式公式：

$$V_{settle, pressed} = \frac{(C_{base} + C_{finger})V_{base} + C_{hold}V_{hold}}{C_{hold} + C_{base} + C_{finger}}$$

使用以下适用于  $V_{\Delta}$  的公式计算当手指触摸传感器时 CVD 波形的稳定点差分值：

$$\frac{V_{\Delta pressed}}{V_{DD}} = \frac{(C_{base} + C_{finger}) - C_{hold}}{C_{hold} + C_{base} + C_{finger}}$$

**图 7： CVD 信号 —— 手指按下时的差分值与手指释放时的差分值的差异**



最后，从手指按下时的差分值中减去手指未按下时的差分值来计算总 CVD 信号。**公式 5** 为向电路中添加手指电容所引起的传感器读数变化量。这是我们应该以最大化方式设计的值。

$$CVD = V_{\Delta pressed} - V_{\Delta released}$$

$$\frac{CVD}{V_{DD}} = \frac{(C_{base} + C_{finger}) - C_{ADC}}{C_{ADC} + C_{base} + C_{finger}} - \frac{C_{base} - C_{ADC}}{C_{ADC} + C_{base}}$$

#### 公式 5： CVD 信号

$$\frac{CVD}{V_{DD}} = \frac{2C_{ADC}C_{finger}}{(C_{ADC} + C_{base})(C_{ADC} + C_{base} + C_{finger})}$$

## 时序注意事项

### 预充电延时

定义：对内部电容和外部电容充电所花费的时间量。

此延时对系统的抗噪声性能影响不大。但是，如果 CVD 实现使用其他传感器作为内部保持电容的参考电压源，并且传感器或其参考源的时间常数较大，则在退出预充电阶段之前，默认延时所提供的时间可能不足以使外部参考源完全充电至 VDD。

如果两个电容均未完全充电，则传感器信号将被破坏。在进入采集阶段之前，电路上不再有已知的电荷。相反，电荷此时取决于电容。这对于 CVD 扫描方法来说是不可操作的模式，应加以避免。应该增加预充电延时，以确保每次采样前，电容均已完全充电。

示波器上观察到波形应表示传感器增加的电容相当于重重按下手指增加的电容。如果示波器指示传感器在进入采集阶段之前未完全充电，则应该延长预充电时间。

要延长本应用笔记提供的代码示例中的预充电时间，请在“Optional additional and/or variable delay”（可选的附加延时和 / 或可变延时）注释之后、采集阶段之前的代码行中添加更多 NOP。

要延长其他代码库或框架中的预充电时间，可搜索高级波形设置并找到预充电（有时称为 C<sub>hold</sub> 充电延时）设置。

### 采集 / 稳定延时

定义：在电容预充电至相反状态之后，使电容电压等于中值电压所花费的时间量。

在 CVD 波形的整个采集阶段中，在 ADC 转换开始前，传感器将设置为呈高阻态的输入（TRIS = 1）。传感器周围的任意低阻态源都能够对其进一步放电或充电，进而对稳定电荷产生影响。换句话说，这是波形的噪声敏感期。因此，因尽量缩短此阶段允许的时间。

在决定稳定延时量时需要考虑一个权衡问题。如果稳定延时过短，两个电容的电荷将无法完全稳定到一个均衡值。在电路中增加附加的外部电容时，这会降低灵敏度。但是，如果稳定延时过长，噪声会在传感器中耦合，并破坏最终电压。通常情况下，稳定延时应设置为能提供至少 90-95% 的完全稳定灵敏度的最短时间量。特定应用可能需要更长的稳定延时（例如，存在大外部电容/电阻时）或更短的稳定延时（例如，设计中的噪声是已知问题时）。

### 差分延时

采样 A 和采样 B 之间的时间量。

差分延时应尽可能短，以最大程度地提高波形的低频噪声抑制能力。

在两次采样的时间间隔中添加一个小的随机值将有助于衰减低 kHz 范围的噪声，但这不是必需的。

### 采样延时

定义：相邻两次采样 A 之间的时间量。换句话说，相邻两个 CVD 波形之间的时间量。

采样延时应随机化，以衰减传感器采样率的谐波噪声频率。

## 具有两个采集阶段的 CVD

采样 A 和采样 B 的标准 CVD 波形的理想稳定电压均为  $\frac{1}{2} \cdot V_{DD}$ 。这可提供稳定电压与电压轨之间的最大分离，从而使其应对噪声产生的削波的鲁棒性达到最大。为使稳定电压接近  $\frac{1}{2} \cdot V_{DD}$ ，内部电容和外部电容必须大致相等。

如果内部电容远大于外部电容（或反过来），则稳定电压将按比例由较大电容的起始值决定。这会使灵敏度不太理想。

要解决此问题，可将两个采集阶段连在一起以强制最终电压接近于  $\frac{1}{2} \cdot V_{DD}$ 。

如果外部电容远大于内部电容：

- 执行标准 CVD 预充电和采集阶段（公式 3）。
- 保持外部电容上的电压，同时再次对内部电容进行充电。
- 执行另一个采集阶段（公式 6）。

在示波器上观察传感器的波形时，可以看到此过程的稳定电压是第一个采样阶段的初始稳定电压的二倍，如图 8 所示。因此，将其称为“双 CVD 波形”。

$$V_{total}^{double} = C_{base} V_{settle}^{normal} + C_{hold} V_{hold}$$

$$V_{total}^{double} = \frac{C_{base} V_{settle}^{normal} + C_{hold} V_{hold}}{C_{base} + C_{hold}}$$

图 8: 双 CVD 波形

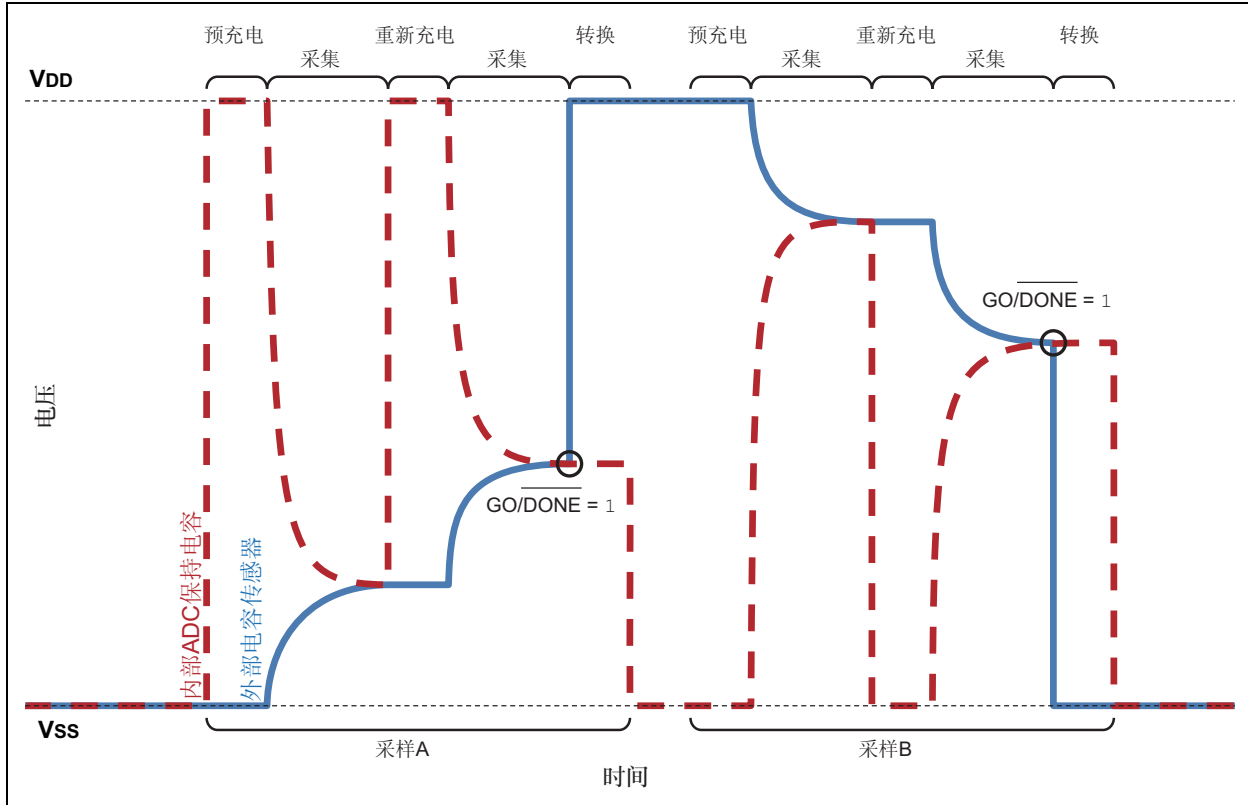
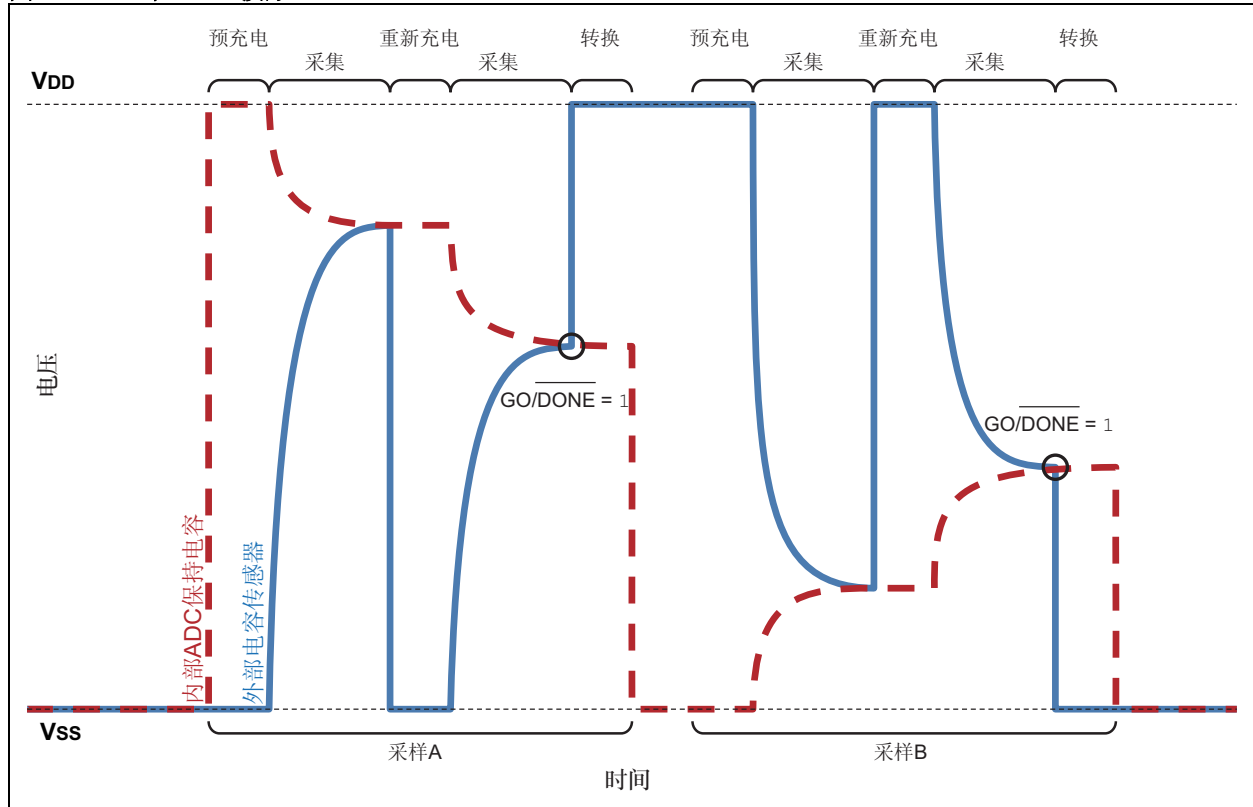




图 9: 半 CVD 波形



如果内部电容远大于外部电容：

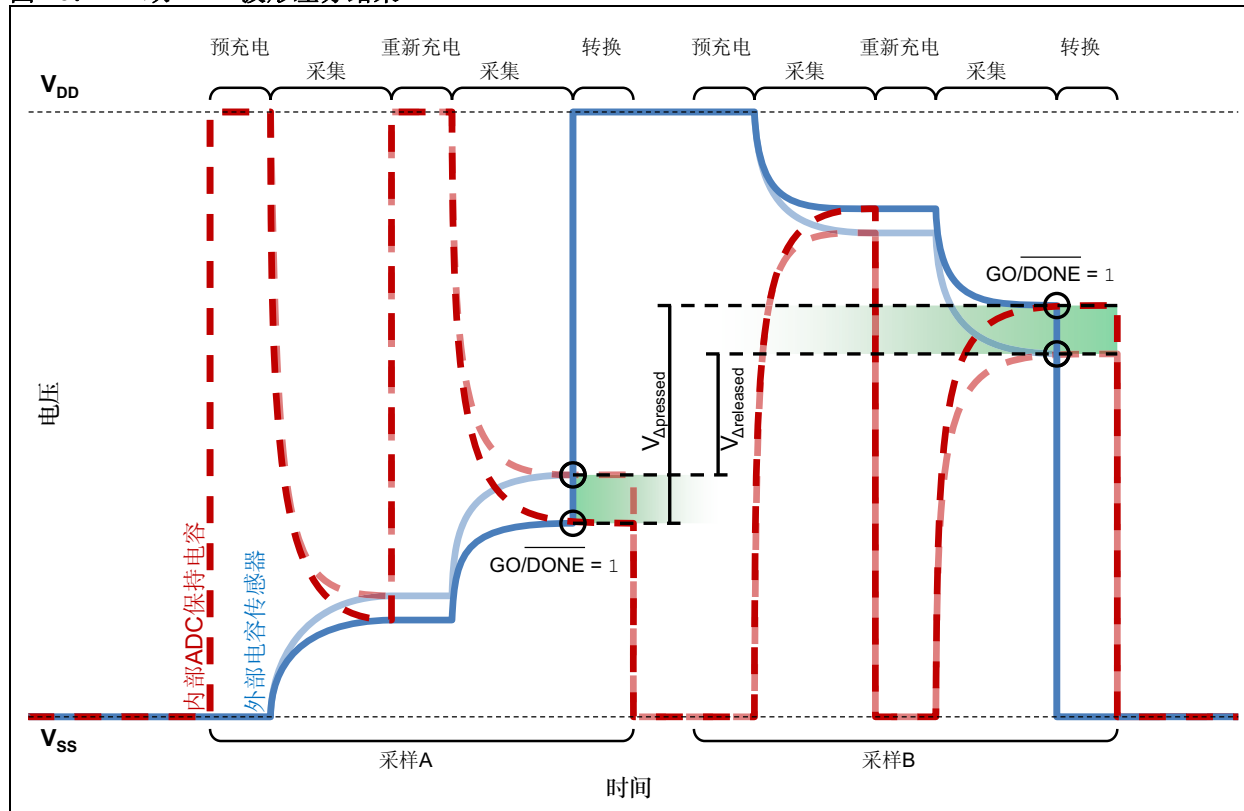
- 执行标准 CVD 预充电和采集阶段（公式 3）。
- 保持内部电容上的电压，同时再次对外部电容进行充电。
- 执行另一个采集阶段（公式 7）。

在示波器上观察传感器的波形时，可以看到此过程的稳定电压是第一个采样阶段的初始稳定电压的一半，如图 9 所示。因此，将其称为“半 CVD 波形”。

$$Q_{total}^{half} = C_{base}V_{base} + C_{hold}V_{settle}^{normal}$$

$$V_{total}^{half} = \frac{C_{base}V_{sensor} + C_{hold}V_{settle}^{normal}}{C_{base} + C_{hold}}$$

图 10: 双 CVD 波形差分结果



用公式 3 替代“标准”稳定电压并进行简化，双波形和半波形的最终稳定电压分别为：

公式 6: 双 CVD 波形的稳定电压

$$V_{total}^{double} = \frac{C_{hold}^2(V_{base} - V_{hold})}{(C_{base} + C_{hold})^2} + \frac{2C_{hold}(V_{hold} - V_{base})}{C_{base} + C_{hold}} + V_{base}$$

$V_{total}^{double}$  是双 CVD 波形的最终稳定电压。

公式 7: 半 CVD 波形的稳定电压

$$V_{total}^{half} = \frac{C_{hold}^2(V_{hold} - V_{base})}{(C_{base} + C_{hold})^2} + V_{base}$$

$V_{total}^{half}$  是半 CVD 波形的最终稳定电压。

## 双采集阶段差分结果

上面的稳定电压公式为一般形式，适用于第一次和第二次差分采样。对于第一次采样（“A”），将 0 代入  $V_{base}$ ， $V_{DD}$  代入  $V_{ADC}$  来计算实际稳定电压。对于第二次采样（“B”），将  $V_{DD}$  代入  $V_{base}$ ，0 代入  $V_{ADC}$  来计算实际稳定电压。

$$V_A = V_{settle}\{V_{hold} = V_{DD}, V_{base} = 0\}$$

$$V_B = V_{settle}\{V_{hold} = 0, V_{base} = V_{DD}\}$$

然后计算这两个电压的差值来获得传感器的读数。实际上， $V_B$  的值通常比  $V_A$  大，因此应调整减法的顺序以生成正结果。此数学计算如图 10 所示。

$$V_{\Delta} = V_B - V_A$$

## 公式 8： 两个双 CVD 稳定点之间的电压差

$$\frac{V_{\Delta released}^{double}}{V_{DD}} = \frac{C_{base}^2 - 2C_{hold}C_{base} - C_{hold}^2}{(C_{base} + C_{hold})^2}$$

## 公式 9： 两个半 CVD 稳定点之间的电压差

$$\frac{V_{\Delta released}^{half}}{V_{DD}} = \frac{C_{base}^2 + 2C_{hold}C_{base} - C_{hold}^2}{(C_{base} + C_{hold})^2}$$

## 添加手指电容

现在，执行相同的分析，但在电路中增加一个电容：用户的手指。这会更改外部电容和总电容。

$$C_{external, pressed} = C_{base} + C_{finger}$$

$$C_{total, pressed} = C_{hold} + C_{base} + C_{finger}$$

手指按下时两个双 CVD 稳定点之间的电压差：

$$\frac{V_{\Delta pressed}^{double}}{V_{DD}} = \frac{(C_{base} + C_{finger})^2 - 2C_{hold}(C_{base} + C_{finger}) - C_{hold}^2}{(C_{hold} + C_{base} + C_{finger})^2}$$

手指按下时两个半 CVD 稳定点之间的电压差：

$$\frac{V_{\Delta pressed}^{half}}{V_{DD}} = \frac{(C_{base} + C_{finger})^2 + 2C_{hold}(C_{base} + C_{finger}) - C_{hold}^2}{(C_{hold} + C_{base} + C_{finger})^2}$$

最后，对于两种类型，均从手指按下时的差分值中减去手指未按下时的差分值来计算总 CVD 信号。这是向电路中添加手指电容所引起的传感器读数变化量。公式 10 和公式 11 包含的值是我们应该以最大化方式设计的值。

## 公式 10： 对于两种双 CVD 扫描方法，手指按下时的差分值减去手指未按下时的差分值

$$\frac{CVD_{double}}{V_{DD}} = \frac{2C_{hold}C_{finger}[2C_{base}(C_{hold} + C_{base} + C_{finger}) + C_{hold}C_{finger}]}{(C_{base} + C_{hold})^2(C_{finger} + C_{base} + C_{hold})^2}$$

## 公式 11： 对于两种半 CVD 扫描方法，手指按下时的差分值减去手指未按下时的差分值

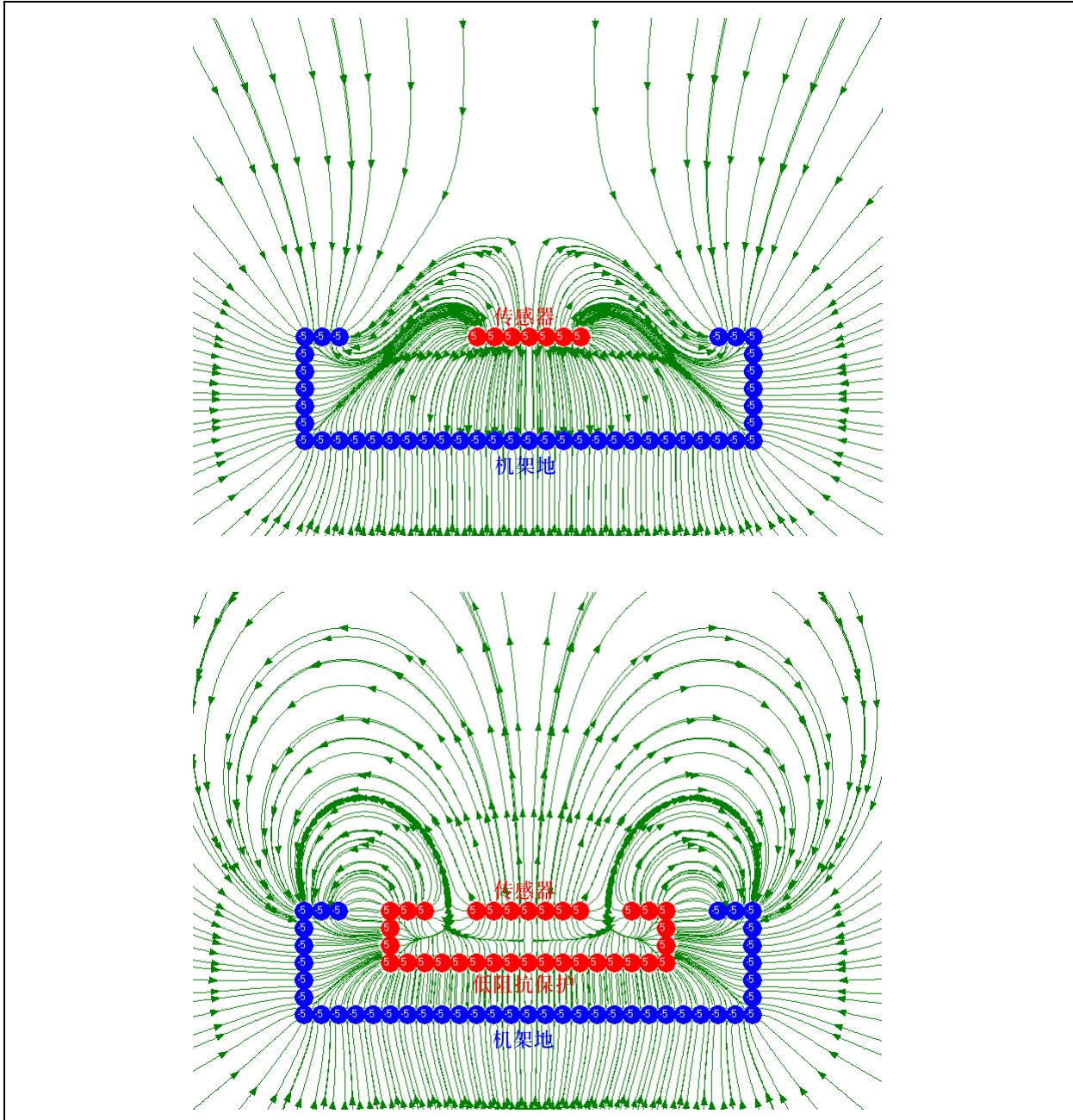
$$\frac{CVD_{half}}{V_{DD}} = \frac{2C_{hold}^2C_{finger}(2C_{base} + 2C_{hold} + C_{finger})}{(C_{base} + C_{hold})^2(C_{finger} + C_{base} + C_{hold})^2}$$

## 具有主动保护的 CVD 扫描

CVD 扫描测量总电容，因此当传感器的基础电容减小时，由用户手指引起的信号变化将增大。主动保护是一种可最大程度地减小基础电容的方法，它通过降低传感器与其周围环境之间的电势来实现。

为应用设计保护时，最佳的解决方案是完全环绕传感器及其走线。如果无法实现，则当低阻抗源接近传感器时应该特别注意。应当远离电容传感器的走线示例包括通信线路、电机驱动线路、接地层和电源层。保护走线应放置在传感器与这些源之间。但是，不要在用户与传感器之间放置保护，因为这样会将手指接近传感器的效果屏蔽。

图 11: 具有 / 不具有保护走线时的电场线



## 保护走线设计准则

设计保护时，需记住以下准则：

一条保护走线可用于所有传感器。传感器按顺序进行扫描，因此可以为当前正在扫描的传感器驱动保护，而不会影响其他传感器。

任何电源层或低阻抗走线都应远离传感器。

在传感器焊盘周围，保护走线的厚度应该约为 1 mm 并且距离传感器 2-3 mm。

沿着返回至 PIC 器件引脚的传感器走线，保护走线的厚度可以与传感器走线的厚度相同：0.1-0.3 mm。保护走线与传感器走线之间的距离可短至 0.5 mm。

标准保护方法包括使用单位增益放大器来缓冲传感器波形，并输出至周围走线上。此方法会产生不必要的成本，因为有两种可用于驱动保护信号的选项并不需要外部元件。

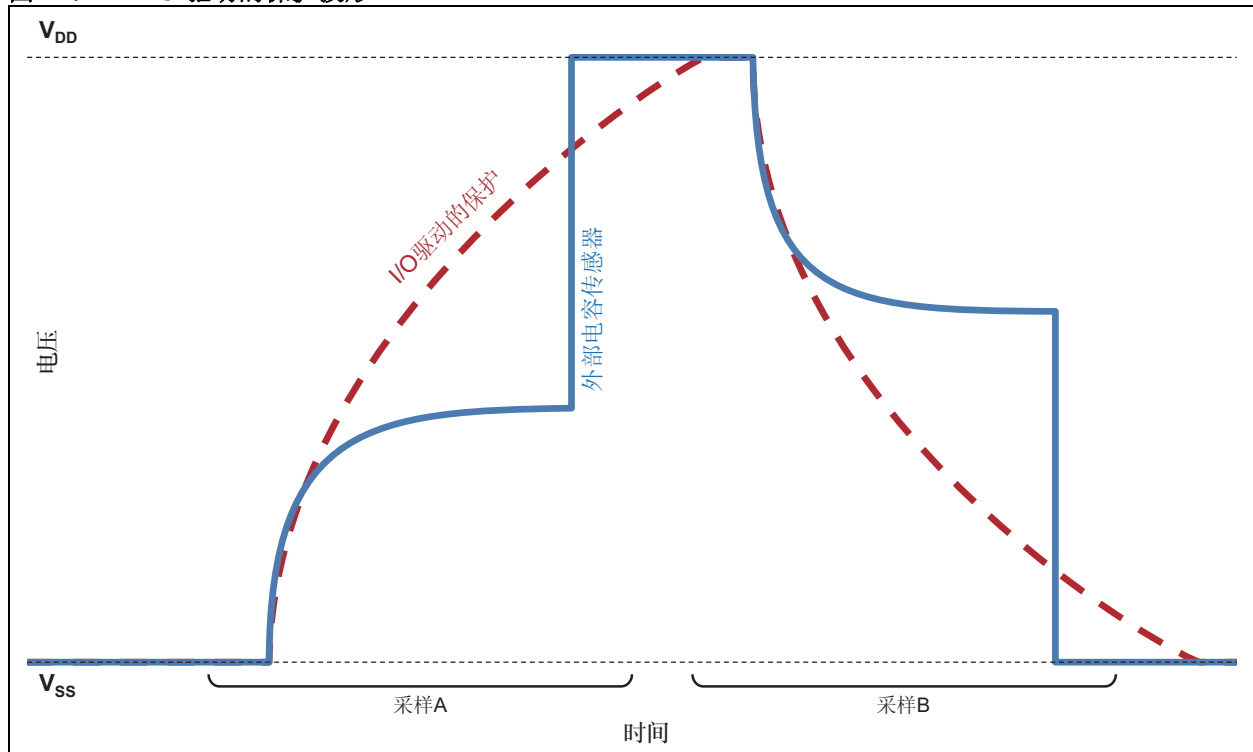
## 使用任意 I/O 引脚实现保护

可以使用任意 I/O 引脚驱动 CVD 保护信号，如图 12 所示。波形并非完全匹配，因此保护效率将降低。但测试已表明，该方法可提供完全匹配保护的效益的 50-70%，绝对值得占用一个引脚。

可以在保护中添加一个串联电阻来增大其在充电和放电时的时间常数。保护波形设计得与传感器波形越匹配，传感器可感知的电势就越小，其灵敏度就越高。

由于保护的输出与两个电容的连接之间存在一个指令周期的时序差，因此这种实现的效率稍低。（请参见附录中的代码实现示例。）可使用硬件定时器和 PWM 输出将保护的电势变化与两个电容的连接进行同步，来消除此时间差。由于这种基于 Fosc 的配置需要调整，因此不提供示例代码。

图 12: I/O 驱动的保护波形

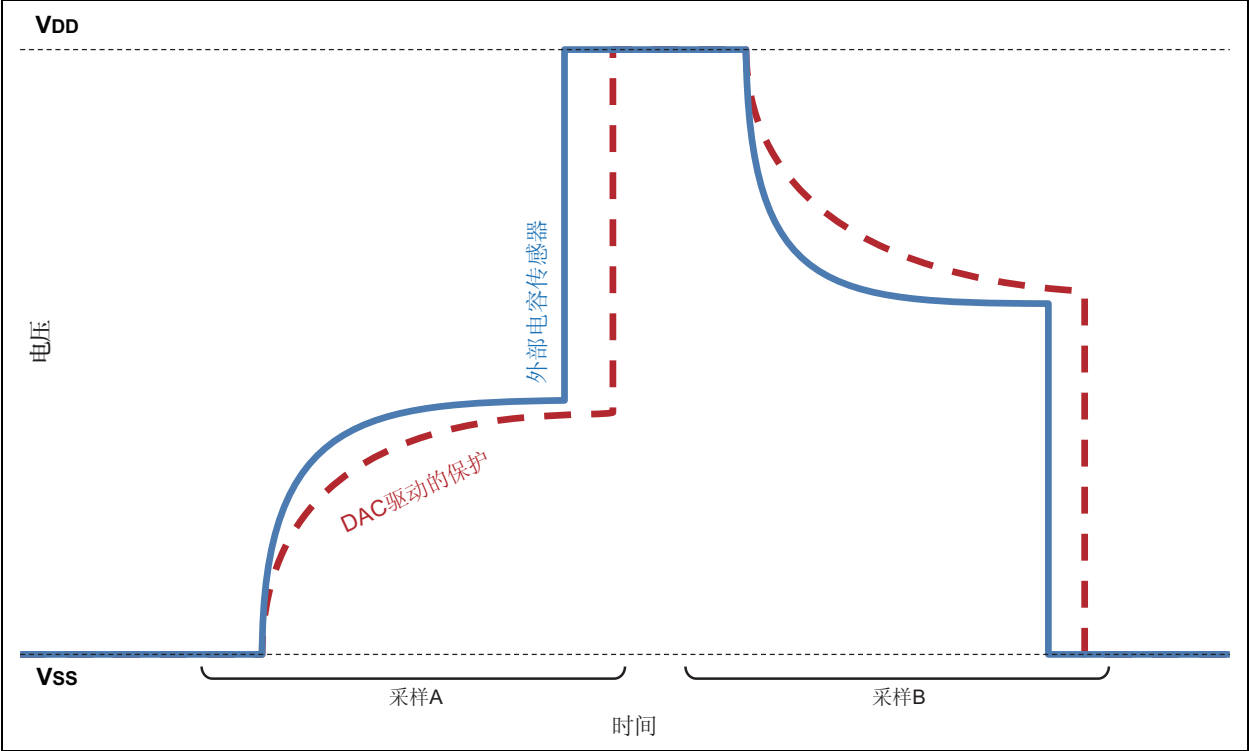


使用 DACOUT 引脚实现保护

如果使用 DACOUT 引脚驱动保护信号（如图 13 所示），则通过选择 DAC 在两个采集阶段中的稳定值可以使保护波形与传感器波形更加匹配。此方法可提供完全匹配保护的效益的 70-90%。如果 DACOUT 引脚可用，建议使用这种方法。如果 DACOUT 不可用，则使用 I/O 驱动的保护是第二选择。

可通过对采样 A 和采样 B 使用单独的 CVD 稳定值，不断按照每种采样下最匹配的输出电压来调整 DAC，从而进一步增强保护的效益。对于 10 位 ADC 和 5 位 DAC，仅需将 ADC 结果右移 5 位并将其用作下一次采样的 DACOUT 稳定电压，即可实现此效果。

图 13: DACOUT 驱动的保护波形



## 采用相互驱动器的 CVD 扫描

CVD 检测方法设计为测量电容的相对变化；但是，高阻抗走线（如电容传感器）还受到附近低阻抗源（如接地 / 电源层、天线和高频数字走线）的影响。

主动保护走线的目的是尽可能降低传感器可感知到的环境电势。正如本应用笔记的保护部分所介绍的，这可以提高传感器对电容变化的灵敏度。

但是，相互驱动器走线的目的是检测高阻抗电容传感器与低阻抗相互信号之间的耦合变化。

### 相互驱动器应用范围

相互驱动信号对设计有利的情况主要有两种：

- 如果有金属可能物理接触到传感器，使用相互信号驱动金属可在发生短接时消除传感器读数中的毛刺。（例如，金属表面触摸系统上的金属层。这不是必需的，但如果金属层可能与传感器短接，这是有好处的。）
- 如果检测的目标与传感器的参考地隔离，在传感器附近放置一个相互驱动器将使应用能够检测到传感器与相互驱动器之间的电容率变化。

**注：** 有关Microchip的金属表面触摸设计的更多信息，请参见应用笔记 AN1325, “*mTouch™ Metal-Over-Cap Technology*”，可在我们的网站（[www.microchip.com/mTouch](http://www.microchip.com/mTouch)）上找到该应用笔记。

## 工作原理

相互驱动信号是一个方波，在所有采样 A 期间驱动为低电平，在所有采样 B 期间驱动为高电平。（因此对于每次采样，它都与外部传感器的起始电压同相。）如图 14 所示。

执行任何 CVD 波形时，电容传感器都将测量到两种结果：

1. 根据典型 CVD 波形的物理特性，将正常测量传感器的电容。

电容增大时，**采样 A** 的值将减小。

电容增大时，**采样 B** 的值将增大。

2. 还将测量传感器与相互驱动器的耦合。

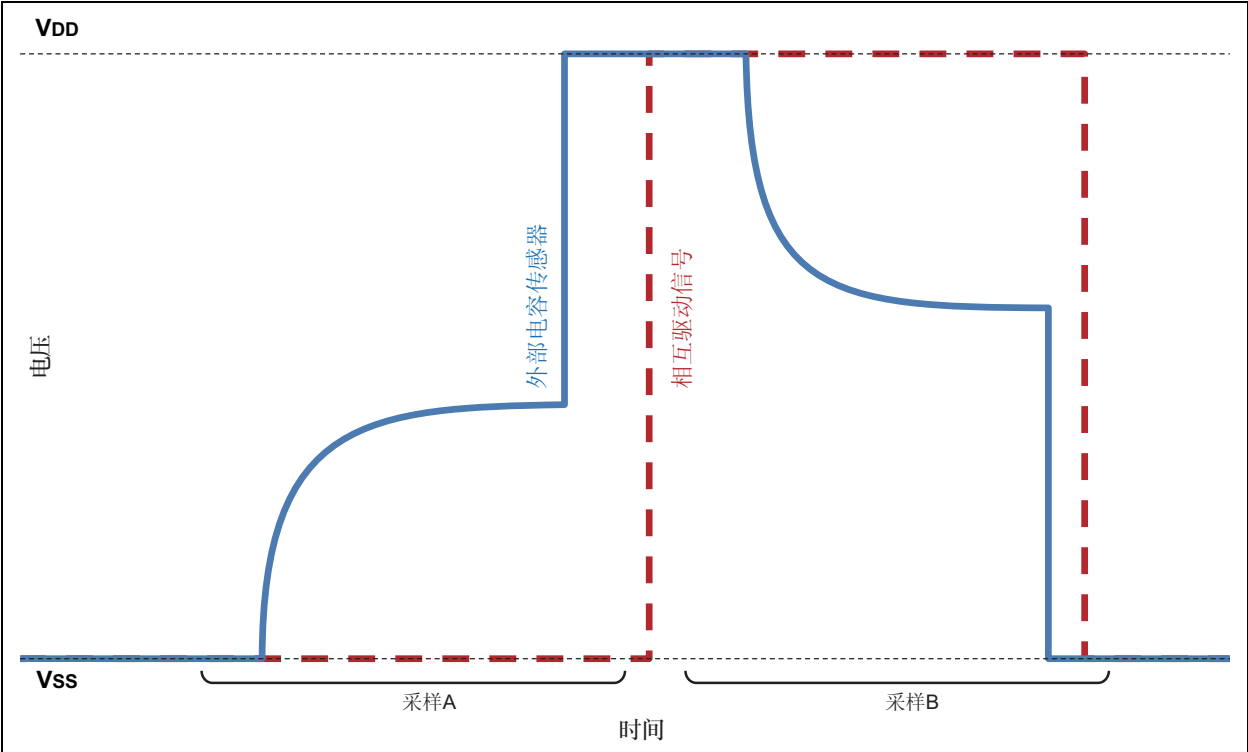
耦合增大时，**采样 A** 的值将减小。

耦合增大时，**采样 B** 的值将增大。

如果没有相互驱动器，当走线与传感器之间的耦合增大时，将导致两种结果之间发生冲突。

例如，如果走线始终驱动为 VSS，则当耦合增大时，将导致采样 B 的稳定电压降低。同时，电容增大时，将导致采样 B 的稳定电压升高。由于这两种结果常常同时发生，因此在移位方向和失去相应的灵敏度之间存在冲突。当一种结果压倒另一种结果时，传感器信号还会出现异常行为。

图 14: I/O 驱动的相互耦合波形



随着相互驱动器走线和电容传感器之间的耦合的增大，耦合增大导致的移位方向将与电容增大导致的移位方向匹配。这会最大程度地提高传感器的灵敏度，避免冲突带来的影响。



## 硬件 CVD

一些精选的 PIC® 器件包含能够在硬件中生成 CVD 波形的高级 ADC 模块。

示例包括：

- PIC16(L)F1512/3
  - 17 路模拟通道
- PIC12LF1552
  - 4 路模拟通道

硬件模块允许在应用中保存程序空间，不必为每个传感器都实现专用的扫描程序。它还可缩短 mTouch 触摸传感服务程序所需的执行时间，从而使应用可执行更多其他功能。由于在硬件中执行扫描，因此 CPU 可以自由执行其他算法，同时由硬件管理扫描方法。

## 特性

硬件 CVD 模块为 PIC16 增强型内核器件上的典型 10 位 ADC 增加了功能。

- 可在任何模拟通道上生成高度可配置的自动差分 CVD 波形
- 两组用于存储采样 A 和采样 B 结果的 ADC 结果寄存器
- 软件可调的内部 ADC 电容，从 +0 pF 至 +28 pF，增量为 4 pF
- 基于定时器和 / 或 CCP 模块的自动转换触发
- 引脚与内部 ADC 总线的可选连接，实现外部可视性
- 可配置的预充电和采集波形时序
- 通过一个或两个引脚实现同步保护环驱动输出

## 模块配置

AADCON0 必须使能 ADC 模块。

AADCON1 必须根据当前的 Fosc 值设置转换时钟。有关如何对应用执行此操作的更多信息，请参见器件的数据手册。始终建议将 VDD 用作正参考电压。

AADCON2 确定是否有任何触发信号使 ADC 的 GO/DONE 位自动置 1。如果有，建议使能 GIE、PEIE 和 ADIE，并在中断服务程序中使用 ADIF 标志位来处理结果。ADIF 标志位必须手动清零。

AADCON3 用于控制 CVD 波形。ADEPPOL 和 ADIPPOL 应设置为相反值，以确保预充电阶段正常工作。

本应用笔记中的波形对应于 ADEPPOL = 0 和 ADIPPOL = 1，但反过来同样有效。ADDSEN 应设置为高电平以使能第二次采样。ADIPEN 应设置为高电平以反转第二次采样，并实现差分波形，而不是“采样 A”的两个副本。

AADPRE 的值必须大于 0。应将其设置为确保有充足的时间使外部电容和内部电容完全充电。更多信息，请参见本应用笔记的“时序注意事项”部分。

AADACQ 的值必须大于 0。应将其设置为允许两个电容完全稳定下来的最小值。更多信息，请参见本应用笔记的“时序注意事项”部分。

AADGRD 可选择性地用于使能一个或两个引脚上的同步保护。有关如何对应用执行此操作的更多信息，请参见器件的数据手册。

AADCAP 可用于增大内部 ADC 保持电容。仅当外部传感器的电容大于内部电容时，此操作才会带来好处。（当采样 A 的稳定电压小于 VDD/2 时，外部电容大于内部电容。）理想情况下，应调节 AADCAP 的值，直到稳定电压与 VDD/2 相匹配。

## 工作原理

由于此过程自动完成，使用该模块时可采用 C 语言代码来驱动波形。对于手动实现 CVD，汇编语言是唯一建议的编程方法。

模块配置完成并且 GO/DONE 位置 1 后，ADC 模块将执行 CVD 状态机并进行两次转换。只有在两次转换完成后，才会将 GO/DONE 位清零，并将 ADIF 位置 1。

两次转换的结果存储在匹配的两组结果寄存器中：AADRES0 和 AADRES1。（共有 4 个 8 位寄存器用于存放值：AADRES0L、AADRES0H、AADRES1L 和 AADRES1H。）

## 例 1: 硬件 CVD 模块的初始化和使用

```
void InitHardwareADC(void)
{
    AADCON0bits.ADON      = 1;

    // 右对齐, Vref = Vdd
    AADCON1bits.ADFM      = 1;
    AADCON1bits.ADCS      = 0b101;
    AADCON1bits.ADPREF     = 0b00;

    // 未选择任何触发信号
    ADCON2bits.TRIGSEL     = 0b000;

    // 采样 A: 外部 Vss, 内部 Vdd
    AADCON3bits.ADEPPOL    = 0;
    AADCON3bits.ADIPPOL    = 1;
    AADCON3bits.ADDSEN     = 1;
    AADCON3bits.ADIPEN     = 1;
    AADCON3bits.ADOLEN     = 0;
    AADCON3bits.ADOEN      = 0;
    AADCON3bits.ADOOEN     = 0;

    AADPRE      = PRECHARGE_DELAY;
    AADACQ      = SETTTLING_DELAY;

    // 使能单个保护引脚
    AADGRDbits.GRDAOE      = 1;
    AADGRDbits.GRDBOE      = 0;
    AADGRDbits.GRDPOL      = 0;

    // 4 pf 附加 ADC 电容
    ADCAPbits.ADDCAP       = 0b000001;
}

void ServiceHardwareADC(void)
{
    AADCON0bits.CHS = 0b00000;
    AADCON0bits.GO  = 1;

    while(AADCON0bits.GO);
    differentialResult = AADRES1 | 0x0400;
    differentialResult -= AADRES0;
}
```

## 总结

电容分压 (CVD) 检测方法是一项用于检测电容相对变化的技术, 其功能强大且成本低。

基本扫描配置适用于大多数应用, 是市场上所有解决方案中需要元件数最少的一个。只要作出明智的设计决策, 还可以实现保护走线和相互耦合驱动等高级功能, 使任何系统都可以获得优良的信噪比。

有关 Microchip 的 mTouch™ 触摸传感技术以及为实现最大抗噪稳定性所建议的硬件设计准则的更多信息, 请访问我们的网站 [www.microchip.com/mTouch](http://www.microchip.com/mTouch)。

## 附录 A——代码示例

### 示例：DAC 作为参考

本代码示例使用 DAC（而不是单独的模拟通道）作为内部保持电容的参考电压源来实现 CVD 波形。在单传感器应用中，这可以防止扫描指定某个引脚作为参考。

采样 A 与采样 B 之间的差异已加粗并用红色显示。

#### 采样 A:

```
#define PIC_DACCON0_VDD 0xC0
#define PIC_DACCON1_VDD 0x1F
#define PIC_ADCON0_SELECT_DAC 0x79
#define PIC_ADCON0_SELECT_AN0 0x01

#define SENSOR_LAT LATA
#define SENSOR_TRIS TRISA
#define SENSOR_PIN 0

; 将 DAC 初始化为“VDD”参考
BANKSEL DACCON0
movlw PIC_DACCON0_VDD
movwf DACCON0
movlw PIC_DACCON1_VDD
movwf DACCON1

; 预充电 ADC 电容
BANKSEL ADCON0
movlw PIC_ADCON0_SELECT_DAC
movwf ADCON0

movlw LOW SENSOR_LAT
movwf FSR1L
movlw HIGH SENSOR_LAT
movwf FSR1H
movlw LOW SENSOR_TRIS
movwf FSR0L
movlw HIGH SENSOR_TRIS
movwf FSR0H

; 可选的附加和 / 或可变延时
NOP

; 采集阶段
movlw PIC_ADCON0_SELECT_AN0
bsf INDF0, SENSOR_PIN ; (TRIS)
movwf ADCON0

; 采集 / 稳定延时
NOP
NOP
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bsf ADCON0, 1

; (建议) 0.5 TAD 延时
NOP

; 为采样 B 预充电传感器
bsf INDF1, SENSOR_PIN ; (LAT)
bcf INDF0, SENSOR_PIN ; (TRIS)

btfsc ADCON0, 1
goto $-1

; 结果存储在 ADRESL 和 ADRESH 中
```

#### 采样 B:

```
#define PIC_DACCON0_VSS 0x00
#define PIC_DACCON1_VSS 0x00
#define PIC_ADCON0_SELECT_DAC 0x79
#define PIC_ADCON0_SELECT_AN0 0x01

#define SENSOR_LAT LATA
#define SENSOR_TRIS TRISA
#define SENSOR_PIN 0

; 将 DAC 初始化为“VSS”参考
BANKSEL DACCON0
movlw PIC_DACCON0_VSS
movwf DACCON0
movlw PIC_DACCON1_VSS
movwf DACCON1

; 预充电 ADC 电容
BANKSEL ADCON0
movlw PIC_ADCON0_SELECT_DAC
movwf ADCON0

movlw LOW SENSOR_LAT
movwf FSR1L
movlw HIGH SENSOR_LAT
movwf FSR1H
movlw LOW SENSOR_TRIS
movwf FSR0L
movlw HIGH SENSOR_TRIS
movwf FSR0H

; 可选的附加和 / 或可变延时
NOP

; 采集阶段
movlw PIC_ADCON0_SELECT_AN0
bsf INDF0, SENSOR_PIN ; (TRIS)
movwf ADCON0

; 采集 / 稳定延时
NOP
NOP
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bsf ADCON0, 1

; (建议) 0.5 TAD 延时
NOP

; 为下一个采样 A 预充电传感器
bcf INDF1, SENSOR_PIN ; (LAT)
bcf INDF0, SENSOR_PIN ; (TRIS)

btfsc ADCON0, 1
goto $-1

; 结果存储在 ADRESL 和 ADRESH 中
```

**示例：传感器作为参考**

本代码示例使用另一个模拟通道作为内部保持电容的参考电压源来实现 CVD 波形。由于不需要 DAC，因此这是功耗最低的扫描选项。

**采样 A:**

```
#define SENSOR_LAT      LATA
#define SENSOR_TRIS     TRISA
#define SENSOR_PIN      0
#define REFERENCE_LAT   LATA
#define REFERENCE_PIN   1

#define PIC_ADCON0_SELECT_AN0  0x01
#define PIC_ADCON0_SELECT_AN1  0x05

; 将 AN1 初始化为参考源
BANKSEL    REFERENCE_LAT
bsf       REFERENCE_LAT, REFERENCE_PIN

; 预充电 ADC 电容
BANKSEL    ADCON0
movlw     PIC_ADCON0_SELECT_AN1
movwf     ADCON0

movlw     LOW    SENSOR_LAT
movwf     FSR1L
movlw     HIGH   SENSOR_LAT
movwf     FSR1H
movlw     LOW    SENSOR_TRIS
movwf     FSR0L
movlw     HIGH   SENSOR_TRIS
movwf     FSR0H

; 可选的附加和 / 或可变延时
NOP

; 采集阶段
movlw     PIC_ADCON0_SELECT_AN0
bsf      INDF0, SENSOR_PIN    ; (TRIS)
movwf     ADCON0

; 采集 / 稳定延时
NOP
NOP
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bsf      ADCON0, 1

; (建议) 0.5 TAD 延时
NOP

; 为采样 B 预充电传感器
bsf      INDF1, SENSOR_PIN    ; (LAT)
bcf      INDF0, SENSOR_PIN    ; (TRIS)

btfsc     ADCON0, 1
goto      $-1

; 结果存储在 ADRESL 和 ADRESH 中
```

采样 A 与采样 B 之间的差异已加粗并用红色显示。

**采样 B:**

```
#define SENSOR_LAT      LATA
#define SENSOR_TRIS     TRISA
#define SENSOR_PIN      0
#define REFERENCE_LAT   LATA
#define REFERENCE_PIN   1

#define PIC_ADCON0_SELECT_AN0  0x01
#define PIC_ADCON0_SELECT_AN1  0x05

; 将 AN1 初始化为参考源
BANKSEL    REFERENCE_LAT
bcf       REFERENCE_LAT, REFERENCE_PIN

; 预充电 ADC 电容
BANKSEL    ADCON0
movlw     PIC_ADCON0_SELECT_AN1
movwf     ADCON0

movlw     LOW    SENSOR_LAT
movwf     FSR1L
movlw     HIGH   SENSOR_LAT
movwf     FSR1H
movlw     LOW    SENSOR_TRIS
movwf     FSR0L
movlw     HIGH   SENSOR_TRIS
movwf     FSR0H

; 可选的附加和 / 或可变延时
NOP

; 采集阶段
movlw     PIC_ADCON0_SELECT_AN0
bcf      INDF0, SENSOR_PIN    ; (TRIS)
movwf     ADCON0

; 采集 / 稳定延时
NOP
NOP
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bcf      ADCON0, 1

; (建议) 0.5 TAD 延时
NOP

; 为采样 A 预充电传感器
bcf      INDF1, SENSOR_PIN    ; (LAT)
bcf      INDF0, SENSOR_PIN    ; (TRIS)

btfsc     ADCON0, 1
goto      $-1

; 结果存储在 ADRESL 和 ADRESH 中
```

## 示例：DAC 作为参考时的双 CVD 波形

本代码示例使用 DAC 作为内部 ADC 保持电容的参考电压源来实现双 CVD 波形。

### 采样 A:

```
#define PIC_DACCON0_VDD 0xC0
#define PIC_DACCON1_VDD 0x1F
#define PIC_ADCON0_SELECT_DAC 0x79
#define PIC_ADCON0_SELECT_AN0 0x01
#define SENSOR_LAT LATA
#define SENSOR_TRIS TRISA
#define SENSOR_PIN 0

; 将 DAC 初始化为“VDD”参考
BANKSEL DACCON0
movlw PIC_DACCON0_VDD
movwf DACCON0
movlw PIC_DACCON1_VDD
movwf DACCON1

; 预充电 ADC 电容
BANKSEL ADCON0
movlw PIC_ADCON0_SELECT_DAC
movwf ADCON0

movlw LOW SENSOR_LAT
movwf FSR1L
movlw HIGH SENSOR_LAT
movwf FSR1H
movlw LOW SENSOR_TRIS
movwf FSR0L
movlw HIGH SENSOR_TRIS
movwf FSR0H

; 第一个采集阶段
movlw PIC_ADCON0_SELECT_AN0
bsf INDF0, SENSOR_PIN ; (TRIS)
movwf ADCON0
NOP
NOP ; 采集 / 稳定延时
NOP

; 重新为内部 ADC 电容充电
movlw PIC_ADCON0_SELECT_DAC
movwf ADCON0
NOP ; 可选, 可变延时
; 第二个采集阶段
movlw PIC_ADCON0_SELECT_AN0
movwf ADCON0
NOP
NOP ; 采集 / 稳定延时
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bsf ADCON0, 1

; (建议) 0.5 TAD 延时
NOP

; 为采样 B 预充电传感器
bsf INDF1, SENSOR_PIN ; (LAT)
bcf INDF0, SENSOR_PIN ; (TRIS)

btfsc ADCON0, 1
goto $-1

; 结果存储在 ADRESL 和 ADRESH 中
```

### 采样 B:

```
#define PIC_DACCON0_VSS 0x00
#define PIC_DACCON1_VSS 0x00
#define PIC_ADCON0_SELECT_DAC 0x79
#define PIC_ADCON0_SELECT_AN0 0x01
#define SENSOR_LAT LATA
#define SENSOR_TRIS TRISA
#define SENSOR_PIN 0

; 将 DAC 初始化为“VSS”参考
BANKSEL DACCON0
movlw PIC_DACCON0_VSS
movwf DACCON0
movlw PIC_DACCON1_VSS
movwf DACCON1

; 预充电 ADC 电容
BANKSEL ADCON0
movlw PIC_ADCON0_SELECT_DAC
movwf ADCON0

movlw LOW SENSOR_LAT
movwf FSR1L
movlw HIGH SENSOR_LAT
movwf FSR1H
movlw LOW SENSOR_TRIS
movwf FSR0L
movlw HIGH SENSOR_TRIS
movwf FSR0H

; 第一个采集阶段
movlw PIC_ADCON0_SELECT_AN0
bsf INDF0, SENSOR_PIN ; (TRIS)
movwf ADCON0
NOP
NOP ; 采集 / 稳定延时
NOP

; 重新为内部 ADC 电容充电
movlw PIC_ADCON0_SELECT_DAC
movwf ADCON0
NOP ; 可选, 可变延时
; 第二个采集阶段
movlw PIC_ADCON0_SELECT_AN0
movwf ADCON0
NOP
NOP ; 采集 / 稳定延时
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bsf ADCON0, 1

; (建议) 0.5 TAD 延时
NOP

; 为采样 A 预充电传感器
bcf INDF1, SENSOR_PIN ; (LAT)
bcf INDF0, SENSOR_PIN ; (TRIS)

btfsc ADCON0, 1
goto $-1

; 结果存储在 ADRESL 和 ADRESH 中
```

**示例：DAC 作为参考时的半 CVD 波形**

此方法需要使用未实现的ADC通道来存储ADC的电荷。

**采样 A:**

```
#define PIC_DACCON0_VDD 0xC0
#define PIC_DACCON1_VDD 0x1F
#define PIC_ADCON0_SELECT_DAC 0x79
#define PIC_ADCON0_SELECT_AN0 0x01
#define PIC_ADCON0_UNIMP_CH 0xF1
#define SENSOR_LAT LATA
#define SENSOR_TRIS TRISA
#define SENSOR_PIN 0

; 将 DAC 初始化为“VDD”参考
BANKSEL DACCON0
movlw PIC_DACCON0_VDD
movwf DACCON0
movlw PIC_DACCON1_VDD
movwf DACCON1

; 预充电 ADC 电容
BANKSEL ADCON0
movlw PIC_ADCON0_SELECT_DAC
movwf ADCON0

movlw LOW SENSOR_LAT
movwf FSR1L
movlw HIGH SENSOR_LAT
movwf FSR1H
movlw LOW SENSOR_TRIS
movwf FSR0L
movlw HIGH SENSOR_TRIS
movwf FSR0H

; 第一个采集阶段
movlw PIC_ADCON0_SELECT_AN0
bsf INDF0, SENSOR_PIN ; (TRIS)
movwf ADCON0
NOP
NOP ; 采集 / 稳定延时
NOP

; 重新为外部电容充电
movlw PIC_ADCON0_UNIMP_CH
movwf ADCON0
bcf INDF0, SENSOR_PIN ; (TRIS)
NOP ; 可选, 可变延时

; 第二个采集阶段
movlw PIC_ADCON0_SELECT_AN0
bsf INDF0, SENSOR_PIN ; (TRIS)
movwf ADCON0
NOP
NOP ; 采集 / 稳定延时
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bsf ADCON0, 1
NOP ; (建议) 0.5 TAD 延时

; 为采样 B 预充电传感器
bsf INDF1, SENSOR_PIN ; (LAT)
bcf INDF0, SENSOR_PIN ; (TRIS)

btfsc ADCON0, 1
goto $-1
; 结果存储在 ADRESL 和 ADRESH 中
```

本代码示例使用 DAC 作为内部 ADC 保持电容的参考电压源来实现半 CVD 波形。

**采样 B:**

```
#define PIC_DACCON0_VSS 0x00
#define PIC_DACCON1_VSS 0x00
#define PIC_ADCON0_SELECT_DAC 0x79
#define PIC_ADCON0_SELECT_AN0 0x01
#define PIC_ADCON0_UNIMP_CH 0xF1
#define SENSOR_LAT LATA
#define SENSOR_TRIS TRISA
#define SENSOR_PIN 0

; 将 DAC 初始化为“VSS”参考
BANKSEL DACCON0
movlw PIC_DACCON0_VSS
movwf DACCON0
movlw PIC_DACCON1_VSS
movwf DACCON1

; 预充电 ADC 电容
BANKSEL ADCON0
movlw PIC_ADCON0_SELECT_DAC
movwf ADCON0

movlw LOW SENSOR_LAT
movwf FSR1L
movlw HIGH SENSOR_LAT
movwf FSR1H
movlw LOW SENSOR_TRIS
movwf FSR0L
movlw HIGH SENSOR_TRIS
movwf FSR0H

; 第一个采集阶段
movlw PIC_ADCON0_SELECT_AN0
bsf INDF0, SENSOR_PIN ; (TRIS)
movwf ADCON0
NOP
NOP ; 采集 / 稳定延时
NOP

; 重新为外部电容充电
movlw PIC_ADCON0_UNIMP_CH
movwf ADCON0
bcf INDF0, SENSOR_PIN ; (TRIS)
NOP ; 可选, 可变延时

; 第二个采集阶段
movlw PIC_ADCON0_SELECT_AN0
bsf INDF0, SENSOR_PIN ; (TRIS)
movwf ADCON0
NOP
NOP ; 采集 / 稳定延时
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bsf ADCON0, 1
NOP ; (建议) 0.5 TAD 延时

; 为采样 A 预充电传感器
bcf INDF1, SENSOR_PIN ; (LAT)
bcf INDF0, SENSOR_PIN ; (TRIS)

btfsc ADCON0, 1
goto $-1
; 结果存储在 ADRESL 和 ADRESH 中
```

## 示例：I/O 驱动的保护

本代码示例使用另一个传感器作为内部保持电容的参考电压源，并将一个专用 I/O 引脚用于保护信号来实现 CVD 波形。

采样 A 与采样 B 之间的差异已加粗并用红色显示。请注意，除电压参考和保护驱动器的初始化外，所有步骤都相同。

### 采样 A:

```
#define SENSOR_LAT      LATA
#define SENSOR_TRIS     TRISA
#define SENSOR_PIN      0
#define REFERENCE_LAT   LATA
#define REFERENCE_PIN   1
#define GUARD_LAT       LATA
#define GUARD_PIN       2
#define PIC_ADCON0_SELECT_AN0 0x01
#define PIC_ADCON0_SELECT_AN1 0x05

; 将 AN1 初始化为参考源
BANKSEL    REFERENCE_LAT
bsf       REFERENCE_LAT, REFERENCE_PIN

; 预充电 ADC 电容
BANKSEL    ADCON0
movlw     PIC_ADCON0_SELECT_AN1
movwf     ADCON0

movlw     LOW    GUARD_LAT
movwf     FSR1L
movlw     HIGH   GUARD_LAT
movwf     FSR1H
movlw     LOW    SENSOR_TRIS
movwf     FSR0L
movlw     HIGH   SENSOR_TRIS
movwf     FSR0H

; 可选的附加和 / 或可变延时
NOP

; 采集阶段
movlw     PIC_ADCON0_SELECT_AN0
bsf       INDF1, GUARD_PIN ; 保护
bsf       INDF0, SENSOR_PIN ; TRIS
movwf     ADCON0

; 采集 / 稳定延时
NOP
NOP
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bsf       ADCON0, 1

; (建议) 0.5 TAD 延时
NOP

; 为采样 B 预充电传感器
BANKSEL    SENSOR_LAT
bsf       SENSOR_LAT, SENSOR_PIN
bcf       INDF0, SENSOR_PIN ; (TRIS)

BANKSEL    ADCON0
btfsc     ADCON0, 1
goto      $-1

; 结果存储在 ADRESL 和 ADRESH 中
```

### 采样 B:

```
#define SENSOR_LAT      LATA
#define SENSOR_TRIS     TRISA
#define SENSOR_PIN      0
#define REFERENCE_LAT   LATA
#define REFERENCE_PIN   1
#define GUARD_LAT       LATA
#define GUARD_PIN       2
#define PIC_ADCON0_SELECT_AN0 0x01
#define PIC_ADCON0_SELECT_AN1 0x05

; 将 AN1 初始化为参考源
BANKSEL    REFERENCE_LAT
bcf       REFERENCE_LAT, REFERENCE_PIN

; 预充电 ADC 电容
BANKSEL    ADCON0
movlw     PIC_ADCON0_SELECT_AN1
movwf     ADCON0

movlw     LOW    GUARD_LAT
movwf     FSR1L
movlw     HIGH   GUARD_LAT
movwf     FSR1H
movlw     LOW    SENSOR_TRIS
movwf     FSR0L
movlw     HIGH   SENSOR_TRIS
movwf     FSR0H

; 可选的附加和 / 或可变延时
NOP

; 采集阶段
movlw     PIC_ADCON0_SELECT_AN0
bcf       INDF1, GUARD_PIN ; 保护
bsf       INDF0, SENSOR_PIN ; (TRIS)
movwf     ADCON0

; 采集 / 稳定延时
NOP
NOP
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bsf       ADCON0, 1

; (建议) 0.5 TAD 延时
NOP

; 为采样 A 预充电传感器
BANKSEL    SENSOR_LAT
bcf       SENSOR_LAT, SENSOR_PIN
bcf       INDF0, SENSOR_PIN ; (TRIS)

BANKSEL    ADCON0
btfsc     ADCON0, 1
goto      $-1

; 结果存储在 ADRESL 和 ADRESH 中
```

## 示例：DACOUT 驱动的保护

## 采样 A:

```

#define PIC_DACCON0_VDD      0xC0
#define PIC_DACCON1_VDD      0x1F
#define PIC_DACCON1_SETTLE_A 0x0E
#define PIC_ADCON0_SELECT_DAC 0x79
#define PIC_ADCON0_SELECT_AN1 0x05
#define SENSOR_LAT           LATA
#define SENSOR_TRIS           TRISA
#define SENSOR_PIN           1
#define DACOUT_LAT           LATA
#define DACOUT_PIN           0

; 将 DAC 初始化为“VDD”参考
BANKSEL    DACCON0
movlw      PIC_DACCON0_VDD
movwf      DACCON0
movlw      PIC_DACCON1_VDD
movwf      DACCON1

; 预充电 ADC 电容
movlw      LOW    ADCON0
movwf      FSR1L
movlw      HIGH   ADCON0
movwf      FSR1H

movlw      PIC_ADCON0_SELECT_DAC
movwf      INDF1

movlw      LOW    SENSOR_TRIS
movwf      FSR0L
movlw      HIGH   SENSOR_TRIS
movwf      FSR0H

NOP        ; 可选延时

; 采集阶段
BANKSEL    DACCON0
movlw      PIC_ADCON0_SELECT_AN0
bsf        INDF0, SENSOR_PIN ; (TRIS)
movwf      INDF1
movlw      PIC_DACCON1_SETTLE_A
movwf      DACCON1          ; 设置值
bsf        DACCON0, DACOE   ; 使能 DACOUT

NOP        ; 采集 / 稳定延时
NOP
NOP

; 执行 ADC 转换 (GO/NDONE = 1)
bsf        INDF1, 1

; (建议) 0.5 TAD 延时
NOP

; 为采样 B 预充电传感器
BANKSEL    SENSOR_LAT
bsf        SENSOR_LAT, SENSOR_PIN
bcf        INDF0, SENSOR_PIN ; (TRIS)
BANKSEL    DACCON0
bcf        DACCON0, DACOE ; 禁止 DACOUT
BANKSEL    DACOUT_LAT
bsf        DACOUT_LAT, DACOUT_PIN

btfsc      INDF1, 1
goto       $-1

; 结果存储在 ADRESL 和 ADRESH 中

```

本代码示例将 DAC 同时用作内部保持电容的参考电压源和 DACOUT 保护驱动器来实现 CVD 波形。

## 采样 B:

```

#define PIC_DACCON0_VSS      0x00
#define PIC_DACCON1_VSS      0x00
#define PIC_DACCON1_SETTLE_B 0x10

; 将 DAC 初始化为“VSS”参考
BANKSEL    DACCON0
movlw      PIC_DACCON0_VSS
movwf      DACCON0
movlw      PIC_DACCON1_VSS
movwf      DACCON1

; 预充电 ADC 电容
movlw      LOW    ADCON0
movwf      FSR1L
movlw      HIGH   ADCON0
movwf      FSR1H

movlw      PIC_ADCON0_SELECT_DAC
movwf      INDF1

movlw      LOW    SENSOR_TRIS
movwf      FSR0L
movlw      HIGH   SENSOR_TRIS
movwf      FSR0H

NOP        ; 可选延时

; 采集阶段
BANKSEL    DACCON0
movlw      PIC_ADCON0_SELECT_AN0
bsf        INDF0, SENSOR_PIN ; (TRIS)
movwf      INDF1
movlw      PIC_DACCON1_SETTLE_B
movwf      DACCON1          ; 设置值
bsf        DACCON0, DACOE   ; 使能 DACOUT

NOP        ; 采集 / 稳定延时
NOP
NOP

; 执行 ADC 转换 (GO/NDONE = 1)
bsf        INDF1, 1

; (建议) 0.5 TAD 延时
NOP

; 为采样 A 预充电传感器
BANKSEL    SENSOR_LAT
bcf        SENSOR_LAT, SENSOR_PIN
bcf        INDF0, SENSOR_PIN ; (TRIS)
BANKSEL    DACCON0
bcf        DACCON0, DACOE ; 禁止 DACOUT
BANKSEL    DACOUT_LAT
bcf        DACOUT_LAT, DACOUT_PIN

btfsc      INDF1, 1
goto       $-1

; 结果存储在 ADRESL 和 ADRESH 中

```



## 示例：使用传感器作为参考和相互驱动器的CVD 采样

采样A与采样B之间的差异已加粗并用红色显示。请注意，除电压参考和相互驱动方向的初始化外，所有步骤都相同。

### 采样 A:

```
#define SENSOR_LAT      LATA
#define SENSOR_TRIS     TRISA
#define SENSOR_PIN      0
#define REFERENCE_LAT   LATA
#define REFERENCE_PIN   1
#define MUTUAL_LAT      LATA
#define MUTUAL_PIN      2
#define PIC_ADCON0_SELECT_AN0 0x01
#define PIC_ADCON0_SELECT_AN1 0x05

; 将 AN1 初始化为参考源
BANKSEL    REFERENCE_LAT
bsf       REFERENCE_LAT, REFERENCE_PIN

; 预充电 ADC 电容
BANKSEL    ADCON0
movlw     PIC_ADCON0_SELECT_AN1
movwf     ADCON0

movlw     LOW    SENSOR_LAT
movwf     FSR1L
movlw     HIGH   SENSOR_LAT
movwf     FSR1H
movlw     LOW    SENSOR_TRIS
movwf     FSR0L
movlw     HIGH   SENSOR_TRIS
movwf     FSR0H

; 可选的附加和 / 或可变延时
NOP

; 采集阶段
movlw     PIC_ADCON0_SELECT_AN0
bsf      INDF0, SENSOR_PIN    ; (TRIS)
movwf     ADCON0

; 采集 / 稳定延时
NOP
NOP
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bsf      ADCON0, 1

; (建议) 0.5 TAD 延时
NOP

; 为采样 B 预充电传感器
bsf      INDF1, SENSOR_PIN    ; (LAT)
bcf      INDF0, SENSOR_PIN    ; (TRIS)

; 为采样 B 预充电相互驱动器
BANKSEL    MUTUAL_LAT
bsf      MUTUAL_LAT, MUTUAL_PIN

BANKSEL    ADCON0
btfsc     ADCON0, 1
goto      $-1
; 结果存储在 ADRESL 和 ADRESH 中
```

### 采样 B:

```
#define SENSOR_LAT      LATA
#define SENSOR_TRIS     TRISA
#define SENSOR_PIN      0
#define REFERENCE_LAT   LATA
#define REFERENCE_PIN   1
#define MUTUAL_LAT      LATA
#define MUTUAL_PIN      2
#define PIC_ADCON0_SELECT_AN0 0x01
#define PIC_ADCON0_SELECT_AN1 0x05

; 将 AN1 初始化为参考源
BANKSEL    REFERENCE_LAT
bcf       REFERENCE_LAT, REFERENCE_PIN

; 预充电 ADC 电容
BANKSEL    ADCON0
movlw     PIC_ADCON0_SELECT_AN1
movwf     ADCON0

movlw     LOW    SENSOR_LAT
movwf     FSR1L
movlw     HIGH   SENSOR_LAT
movwf     FSR1H
movlw     LOW    SENSOR_TRIS
movwf     FSR0L
movlw     HIGH   SENSOR_TRIS
movwf     FSR0H

; 可选的附加和 / 或可变延时
NOP

; 采集阶段
movlw     PIC_ADCON0_SELECT_AN0
bsf      INDF0, SENSOR_PIN    ; (TRIS)
movwf     ADCON0

; 采集 / 稳定延时
NOP
NOP
NOP

; 执行 ADC 转换 (GO/nDONE = 1)
bsf      ADCON0, 1

; (建议) 0.5 TAD 延时
NOP

; 为采样 A 预充电传感器
bcf      INDF1, SENSOR_PIN    ; (LAT)
bcf      INDF0, SENSOR_PIN    ; (TRIS)

; 为采样 A 预充电相互驱动器
BANKSEL    MUTUAL_LAT
bcf      MUTUAL_LAT, MUTUAL_PIN

BANKSEL    ADCON0
btfsc     ADCON0, 1
goto      $-1
; 结果存储在 ADRESL 和 ADRESH 中
```

注:

---

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

---

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。在 Microchip 知识产权保护下，不得暗中以其他方式转让任何许可证。

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949 ==**

## 商标

Microchip 的名称和徽标组合、Microchip 徽标、dsPIC、FlashFlex、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC<sup>32</sup> 徽标、rfPIC、SST、SST 徽标、SuperFlash 和 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MTP、SEEVAL 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

Analog-for-the-Digital Age、Application Maestro、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICKIT、PICtail、REAL ICE、rfLAB、Select Mode、SQI、Serial Quad I/O、Total Endurance、TSHARC、UniWinDriver、WiperLock、ZENA 和 Z-Scale 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Inc. 在美国的服务标记。

GestIC 和 ULPP 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2013, Microchip Technology Inc. 版权所有。

ISBN: 978-1-62077-303-1

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC<sup>®</sup> MCU 与 dsPIC<sup>®</sup> DSC、KEELOQ<sup>®</sup> 跳码器件、串行 EEPROM、单片机外设、非易失性存储器及模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

## 全球销售及服务网点

### 美洲

公司总部 **Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 1-480-792-7200  
Fax: 1-480-792-7277

技术支持:

<http://www.microchip.com/support>

网址: [www.microchip.com](http://www.microchip.com)

**亚特兰大 Atlanta**  
Duluth, GA  
Tel: 1-678-957-9614  
Fax: 1-678-957-1455

**波士顿 Boston**  
Westborough, MA  
Tel: 1-774-760-0087  
Fax: 1-774-760-0088

**芝加哥 Chicago**  
Itasca, IL  
Tel: 1-630-285-0071  
Fax: 1-630-285-0075

**克里夫兰 Cleveland**  
Independence, OH  
Tel: 1-216-447-0464  
Fax: 1-216-447-0643

**达拉斯 Dallas**  
Addison, TX  
Tel: 1-972-818-7423  
Fax: 1-972-818-2924

**底特律 Detroit**  
Farmington Hills, MI  
Tel: 1-248-538-2250  
Fax: 1-248-538-2260

**印第安纳波利斯 Indianapolis**  
Noblesville, IN  
Tel: 1-317-773-8323  
Fax: 1-317-773-5453

**洛杉矶 Los Angeles**  
Mission Viejo, CA  
Tel: 1-949-462-9523  
Fax: 1-949-462-9608

**圣克拉拉 Santa Clara**  
Santa Clara, CA  
Tel: 1-408-961-6444  
Fax: 1-408-961-6445

**加拿大多伦多 Toronto**  
Mississauga, Ontario, Canada  
Tel: 1-905-673-0699  
Fax: 1-905-673-6509

### 亚太地区

亚太总部 **Asia Pacific Office**  
Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
Fax: 852-2401-3431

**中国 - 北京**  
Tel: 86-10-8569-7000  
Fax: 86-10-8528-2104

**中国 - 成都**  
Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

**中国 - 重庆**  
Tel: 86-23-8980-9588  
Fax: 86-23-8980-9500

**中国 - 杭州**  
Tel: 86-571-2819-3187  
Fax: 86-571-2819-3189

**中国 - 香港特别行政区**  
Tel: 852-2943-5100  
Fax: 852-2401-3431

**中国 - 南京**  
Tel: 86-25-8473-2460  
Fax: 86-25-8473-2470

**中国 - 青岛**  
Tel: 86-532-8502-7355  
Fax: 86-532-8502-7205

**中国 - 上海**  
Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

**中国 - 沈阳**  
Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

**中国 - 深圳**  
Tel: 86-755-8864-2200  
Fax: 86-755-8203-1760

**中国 - 武汉**  
Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

**中国 - 西安**  
Tel: 86-29-8833-7252  
Fax: 86-29-8833-7256

**中国 - 厦门**  
Tel: 86-592-238-8138  
Fax: 86-592-238-8130

**中国 - 珠海**  
Tel: 86-756-321-0040  
Fax: 86-756-321-0049

### 亚太地区

**台湾地区 - 高雄**  
Tel: 886-7-213-7828  
Fax: 886-7-330-9305

**台湾地区 - 台北**  
Tel: 886-2-2508-8600  
Fax: 886-2-2508-0102

**台湾地区 - 新竹**  
Tel: 886-3-5778-366  
Fax: 886-3-5770-955

**澳大利亚 Australia - Sydney**  
Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

**印度 India - Bangalore**  
Tel: 91-80-3090-4444  
Fax: 91-80-3090-4123

**印度 India - New Delhi**  
Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

**印度 India - Pune**  
Tel: 91-20-2566-1512  
Fax: 91-20-2566-1513

**日本 Japan - Osaka**  
Tel: 81-6-6152-7160  
Fax: 81-6-6152-9310

**日本 Japan - Tokyo**  
Tel: 81-3-6880-3770  
Fax: 81-3-6880-3771

**韩国 Korea - Daegu**  
Tel: 82-53-744-4301  
Fax: 82-53-744-4302

**韩国 Korea - Seoul**  
Tel: 82-2-554-7200  
Fax: 82-2-558-5932 或  
82-2-558-5934

**马来西亚 Malaysia - Kuala Lumpur**  
Tel: 60-3-6201-9857  
Fax: 60-3-6201-9859

**马来西亚 Malaysia - Penang**  
Tel: 60-4-227-8870  
Fax: 60-4-227-4068

**菲律宾 Philippines - Manila**  
Tel: 63-2-634-9065  
Fax: 63-2-634-9069

**新加坡 Singapore**  
Tel: 65-6334-8870  
Fax: 65-6334-8850

**泰国 Thailand - Bangkok**  
Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### 欧洲

**奥地利 Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

**丹麦 Denmark - Copenhagen**  
Tel: 45-4450-2828  
Fax: 45-4485-2829

**法国 France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**德国 Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**意大利 Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**荷兰 Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**西班牙 Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**英国 UK - Wokingham**  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820