

第 37 章 充电时间测量单元 (CTMU)

目录

本章包括下列主题：

37.1 简介	37-2
37.2 寄存器	37-3
37.3 CTMU 工作原理	37-6
37.4 CTMU 模块初始化	37-8
37.5 校准 CTMU 模块	37-9
37.6 使用 CTMU 测量电容	37-15
37.7 使用 CTMU 模块测量时间	37-18
37.8 使用 CTMU 模块产生延时	37-18
37.9 使用 CTMU 测量片上温度	37-19
37.10 休眠 / 空闲模式期间的操作	37-20
37.11 复位对 CTMU 的影响	37-20
37.12 相关应用笔记	37-21
37.13 版本历史	37-22

注： 本系列参考手册章节旨在用作对器件数据手册的补充。根据不同的器件型号，本手册章节可能并不适用于所有 PIC32 器件。

请参见当前器件数据手册中“**充电时间测量单元 (CTMU)**”章节开头部分的注释，以检查本文档是否支持您所使用的器件。

器件数据手册和系列参考手册章节可从 Microchip 网站下载：

<http://www.microchip.com>

37.1 简介

充电时间测量单元 (Charge Time Measurement Unit, CTMU) 是一种灵活的模拟模块，它具有一个可配置电流源和一个围绕它而构造的数字配置电路。CTMU 可用于脉冲源之间的时间差测量，以及异步脉冲生成。CTMU 可与其他片上模拟模块一起，用于高分辨率时间测量、测量电容、电容的相对变化，或生成具有特定延时的输出脉冲。CTMU 是连接电容式传感器的理想选择。

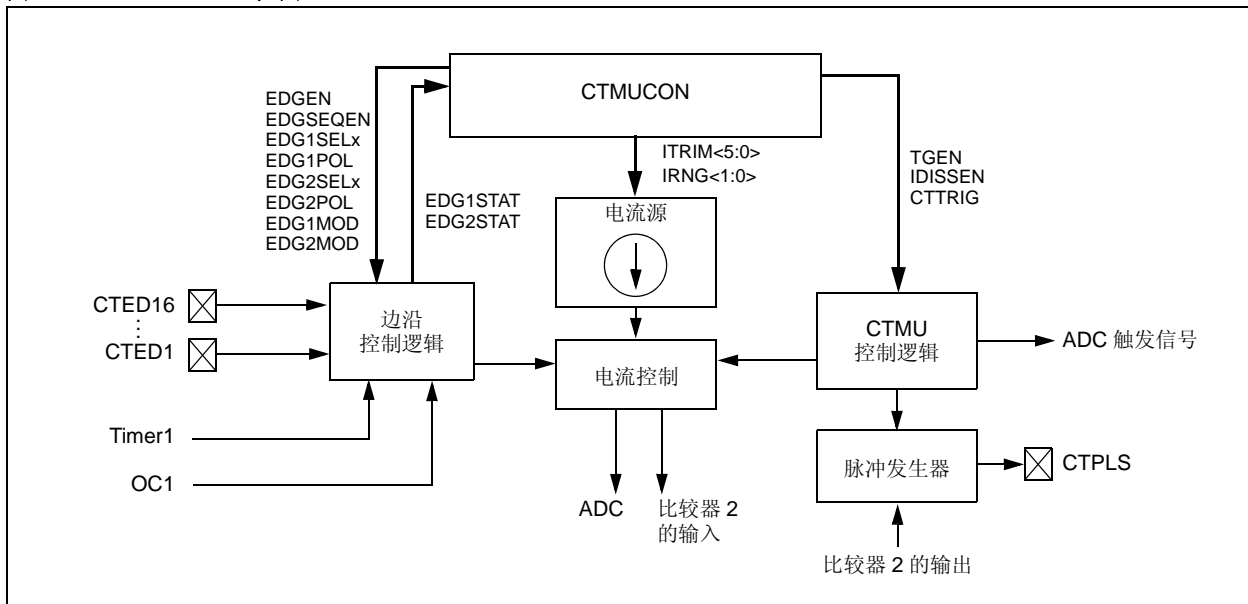
该模块具有以下主要特性：

- 最多 32 路通道，可用于电容或时间测量输入
- 片上精确电流源
- 16 个边沿输入触发源
- 边沿或电平敏感输入选择
- 每个边沿源的极性控制
- 边沿顺序控制
- 边沿响应控制
- 高精度时间测量
- 与系统时钟异步的外部或内部信号的延时
- 集成的温度检测二极管
- 自动采样期间的电流源控制
- 4 个电流源范围
- 时间测量分辨率为 1 ns

CTMU 与模数转换器 (Analog-to-Digital Converter, ADC) 配合工作，根据具体器件和可用的 ADC 通道数，最多可提供 32 路通道用于时间或电荷测量。如果配置为产生延时，那么 CTMU 连接到其中一个模拟比较器。电平敏感输入边沿源有 4 种可供选择：两个外部输入、Timer1 或输出比较模块 1。关于可用输入源的信息，请参见具体器件数据手册。

图 37-1 给出了 CTMU 的框图。

图 37-1: CTMU 框图



37.2 寄存器

CTMUCON 寄存器包含一些控制位，这些控制位用于配置 CTMU 模块边沿源选择、边沿源极性选择、边沿顺序、ADC 触发、模拟电路电容放电和使能。此外，该寄存器还包含一些用于选择电流源范围和电流源微调的位。

表 37-1 汇总了与 CTMU 相关的寄存器。该汇总表之后列出了寄存器的详细说明。

表 37-1: CTMU SFR 汇总 ⁽¹⁾

名称	位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
CTMUCON	31:24	EDG1MOD	EDG1POL	EDG1SEL<3:0>				EDG2STAT	EDG1STAT
	23:16	EDG2MOD	EDG2POL	EDG2SEL<3:0>				—	—
	15:8	ON	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
	7:0	ITRIM<5:0>						IRNG<1:0>	

图注： — = 未实现，读为 0。地址偏移值以十六进制显示。
注 1： 并非所有寄存器都具有关联的置 1、清零和取反寄存器。详情请参见具体器件数据手册。

寄存器 37-1: CTMUCON: CTMU 控制寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	EDG1MOD	EDG1POL	EDG1SEL<3:0> ⁽¹⁾				EDG2STAT	EDG1STAT
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
	EDG2MOD	EDG2POL	EDG2SEL<3:0> ⁽¹⁾				—	—
15:8	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ON	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ITRIM<5:0>						IRNG<1:0>	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31 **EDG1MOD:** 边沿 1 边沿采样选择位

1 = 输入是边沿敏感的

0 = 输入是电平敏感的

bit 30 **EDG1POL:** 边沿 1 极性选择位

1 = 边沿 1 设定为正边沿响应

0 = 边沿 1 设定为负边沿响应

bit 29-26 **EDG1SEL<3:0>:** 边沿 1 源选择位 ⁽¹⁾

1111 = 选择 CTED16

•

•

•

0000 = 选择 CTED1

bit 25 **EDG2STAT:** 边沿 2 状态位

指示边沿 2 的状态, 并且可以通过写入它来控制边沿源

1 = 已发生边沿 2 事件

0 = 未发生边沿 2 事件

bit 24 **EDG1STAT:** 边沿 1 状态位

指示边沿 1 的状态, 并且可以通过写入它来控制边沿源

1 = 已发生边沿 1 事件

0 = 未发生边沿 1 事件

bit 23 **EDG2MOD:** 边沿 2 边沿采样选择位

1 = 输入是边沿敏感的

0 = 输入是电平敏感的

bit 22 **EDG2POL:** 边沿 2 极性选择位

1 = 边沿 2 设定为正边沿响应

0 = 边沿 2 设定为负边沿响应

bit 21-18 **EDG2SEL<3:0>:** 边沿 2 源选择位 ⁽¹⁾

1111 = 选择 CTED16

•

•

•

0000 = 选择 CTED1

bit 17-16 **未实现:** 读为 0

bit 15 **ON:** ON 使能位

1 = 使能模块

0 = 禁止模块

注 1: 关于可用触发源的列表, 请参见具体器件数据手册的“CTMU”章节。

寄存器 37-1: CTMUCON: CTMU 控制寄存器 (续)

bit 14	未实现: 读为 0
bit 13	CTMUSIDL: 空闲模式停止位 1 = 当器件进入空闲模式时, 模块停止工作 0 = 在空闲模式下模块继续工作
bit 12	TGEN: 延时产生使能位 1 = 使能边沿延时产生 0 = 禁止边沿延时产生
bit 11	EDGEN: 边沿使能位 1 = 未阻止边沿 0 = 阻止边沿
bit 10	EDGSEQEN: 边沿顺序使能位 1 = 边沿 1 必须在边沿 2 之前发生 0 = 无需边沿顺序
bit 9	IDISSEN: 电流放电使能位 1 = 模拟电流源输出接地 0 = 模拟电流源输出未接地
bit 8	CTTRIG: 触发信号控制位 1 = 使能触发信号输出 0 = 禁止触发信号输出
bit 7-2	ITRIM<5:0>: 电流源微调位 111111 = 对标称电流的最小负向调整 . . . 100010 100001 = 对标称电流的最大负向调整 011111 = 对标称电流的最大正向调整 011110 . . . 000001 = 对标称电流的最小正向调整 000000 = IRNG<1:0> 指定的标称电流输出
bit 1-0	IRNG<1:0>: 电流范围选择位 11 = 基本电流的 100 倍 10 = 基本电流的 10 倍 01 = 基本电流 (标称值为 0.55 μ A) 00 = 基本电流的 1000 倍

注 1: 关于可用触发源的列表, 请参见具体器件数据手册的 “CTMU” 章节。

37.3 CTMU 工作原理

CTMU 的工作方式是使用恒流源来对电路进行充电。电路的类型取决于要进行测量的类型。在进行电容测量的情况下，电流和向电路施加电流的时间都是固定的。这样，只要通过 ADC 测得电压就可以测得电路的电容。在进行时间测量的情况下，电流和电路的电容都是固定的，充电时间是不一样的。这种情况下，由 ADC 读取的电压可以代表从电流源开始对电路进行充电到停止充电经过的时间。

如果 CTMU 用于产生延时，那么电容和电流源是固定的，向比较器电路提供的电压也是固定的。信号的延时由充电到电压达到比较器门限电压所需的时间决定。

37.3.1 工作原理

CTMU 的工作原理基于以下电荷公式，如公式 37-1 所示。

公式 37-1:

$$I = C \cdot \frac{dV}{dt}$$

简单来说，在电路中测量的电荷（以库仑为单位）定义为：电流（ I ）（以安培为单位）乘以电流流动时间（ t ）（以秒为单位）。电荷也可以定义为：电容（ C ）（以法拉为单位）乘以电路的电压（ V ），如公式 37-2 所示。

公式 37-2:

$$I \cdot t = C \cdot V$$

CTMU 模块提供了一个已知的恒流源。ADC 用于测量公式中的电压（ V ）。剩下两个未知量：电容（ C ）和时间（ t ）。公式 37-2 可用于计算电容或时间，方法是使用公式 37-3 中所示的关系和电路的已知固定电容，或通过使用公式 37-4 中所示的关系和电流源施加于电路的固定时间。

公式 37-3:

$$t = \frac{(C \cdot V)}{I}$$

公式 37-4:

$$C = \frac{(I \cdot t)}{V}$$

37.3.2 电流源

CTMU 的核心是精确电流源，旨在提供用于测量的恒定基准。用户可以从 4 个范围或总共两个数量级的值中选择电流值，并可以按 $\pm 2\%$ 的增量（标称值）对输出进行微调。电流范围通过 $\text{IRNG}\langle 1:0 \rangle$ 位（ $\text{CTMUCON}\langle 1:0 \rangle$ ）进行选择，值 01 代表最小范围。

电流微调通过 $\text{ITRIM}\langle 5:0 \rangle$ 位（ $\text{CTMUCON}\langle 7:2 \rangle$ ）进行。通过这 6 个位可以按大约每步 2% 的步阶微调电流源。请注意，其中一半的范围用于正向调整电流源，另一半用于负向调整电流源。值 000000 是中性位置（无变化）。值 100001（见注 1）代表最大负调整（大约 -62%），011111（见注 2）代表最大正调整（大约 +62%）。

注 1: 值 100001 = $-31 \& 0x3F \rightarrow \text{ITRIM} = -31 * \Delta I$ ，约为标称值的 -62%。

2: 值 011111 = $+31 \& 0x3F \rightarrow \text{ITRIM} = +31 * \Delta I$ ，约为标称值的 +62%。

37.3.3 边沿选择和控制

通过在模块的两路输入通道中发生的边沿事件控制 CTMU 测量。每路通道（称为边沿 1 和边沿 2）可以配置为接收来自 16 个边沿输入引脚之一的输入脉冲。使用 $\text{EDG1SEL}\langle 3:0 \rangle$ 和 $\text{EDG2SEL}\langle 3:0 \rangle$ 位对（ $\text{CTMUCON}\langle 29:26 \rangle$ 和 $\text{CTMUCON}\langle 21:18 \rangle$ ）选择输入。

除了电流源之外，还可以使用 EDG1POL 和 EDG2POL 位（ $\text{CTMUCON}\langle 30 \rangle$ 和 $\text{CTMUCON}\langle 22 \rangle$ ）配置每路通道的事件极性。还可以对输入通道进行过滤以选择边沿事件顺序（边沿 1 在边沿 2 之前发生），方法是将 EDGSEQEN 位（ $\text{CTMUCON}\langle 10 \rangle$ ）置 1。

37.3.4 边沿状态

CTMUCON 寄存器还包含两个状态位 EDG1STAT （ $\text{CTMUCON}\langle 24 \rangle$ ）和 EDG2STAT （ $\text{CTMUCON}\langle 25 \rangle$ ）。它们的主要功能是显示在相应的通道中是否发生了边沿响应。当在通道中检测到边沿响应时，CTMU 会自动将相应位置 1。输入通道的电平敏感或边沿敏感特性意味着，如果通道的配置发生改变，那么状态位会立即置 1，并且与通道的当前状态保持一致。

CTMU 模块使用边沿状态位来控制到外部模拟模块（如 ADC）的电流源输出。只有其中一个状态位置 1 而不是两个状态位同时置 1 时，才会向外部模块提供电流，如果两个位同时置 1 或同时清零，则会切断电流。这使 CTMU 可以仅测量两个边沿事件之间的电流。在两个状态位都置 1 后，必须先将它们清零，然后才能进行另一次测量。如果可能，两个位应同时清零，以避免重新使能 CTMU 电流源。

除了可以由 CTMU 硬件置 1 之外，边沿状态位也可以用软件置 1。也就是说可以在用户应用程序中手动使能或禁止电流源。将其中任意一位置 1（但不是同时置 1）即可使能电流源。将两位同时置 1 或清零即可立即禁止电流源。

37.3.5 中断

每当电流源先使能，然后禁止时，CTMU 就会将其中断标志（ CTMUIF ）置 1。如果未使能边沿顺序（即，边沿 1 必须在边沿 2 之前发生），则需要监视边沿状态位，确定上次发生并导致中断的是哪一个边沿事件。

37.4 CTMU 模块初始化

以下过程是用于初始化 CTMU 模块的通用指南：

1. 使用 IRNG<1:0> 位 (CTMUCON<1:0>) 选择电流源范围。
2. 使用 ITRIM<5:0> 位 (CTMUCON<7:2>) 微调电流源。
3. 通过设置 EDG1SEL 和 EDG2SEL 位 (CTMUCON<29:26> 和 CTMUCON<21:18>) 配置边沿 1 和边沿 2 的边沿输入源。
4. 使用 EDG1POL 位 (CTMUCON<30>) 和 EDG2POL 位 (CTMUCON<22>) 配置边沿输入的输入极性。默认配置是使用负边沿极性 (从高至低跳变)。
5. 使用 EDGSEQEN 位 (CTMUCON<10>) 使能边沿顺序。默认为禁止边沿顺序。
6. 使用 TGEN 位 (CTMUCON<12>) 选择工作模式 (测量或产生延时)。默认为禁止产生延时模式。
7. 使用 CTTRIG 位 (CTMUCON<8>) 将模块配置为在发生第二个边沿事件时自动触发模数转换。默认为禁止转换触发。
8. 通过将 IDISSEN 位 (CTMUCON<9>) 置 1, 对所连接电路放电。在等待足够时间, 让电路完成放电之后, 清零 IDISSEN 位。
9. 通过清零 ON 位 (CTMUCON<15>) 禁止该模块。
10. 清零边沿状态位 EDG2STAT<3:0> 和 EDG1STAT<3:0> (CTMUCON<29:26> 和 CTMUCON<21:18>)。
11. 通过将 EDGEN 位 (CTMUCON<11>) 置 1 使能两个边沿输入。
12. 通过将 ON 位 (CTMUCON<15>) 置 1 使能该模块。

根据要执行的测量或脉冲生成的类型, 可能还需要再初始化和配置一个或更多其他模块, 与 CTMU 模块配合使用:

- 边沿源生成: 除了外部边沿输入引脚之外, 其他模块 (如 ICx、OCx 和 Timer1) 也可以用作 CTMU 的边沿源。关于可用源, 请参见具体器件数据手册。
- 电容或时间测量: CTMU 模块使用 ADC 来测量连接到一路模拟输入通道的电容两端的电压。
- 脉冲生成: 在生成独立于系统时钟的输出脉冲时, CTMU 模块使用比较器 2 和关联的比较器参考电压。

关于初始化这些模块的具体信息, 请参见适用的《PIC32 系列参考手册》中相应模块的章节。

37.5 校准 CTMU 模块

要精确测量电容和时间，以及产生精确延时，需要对 CTMU 进行校准。如果应用只需要测量电容或时间的相对变化，则通常不需要校准。此类应用的示例包括电容触摸开关，在这种应用中，触摸电路具有基本电容，所增加的人体电容会改变电路的总电容。

如果需要测量实际的电容或时间，则必须进行两项硬件校准：电流源需要进行校准，以使其提供精确的电流；要测量的电路也需要进行校准，以测量和/或抵消要测量电容之外的所有其他电容。

37.5.1 电流源校准

CTMU 模块随附的电流源具有 4 种电流范围，其中每种范围都可以在其标称值 $\pm 62\%$ 的范围内进行调节。因此，要进行精确测量，可以通过在特殊的模拟通道上放置一个高精度电阻 R_{CAL} ，测量并调整该电流源。图 37-2 给出了示例电路。电流源测量使用以下步骤执行：

1. 初始化 ADC。
2. 通过将模块配置为脉冲生成 (TGEN = 1) 模式初始化 CTMU。
3. 通过将 EDG1STAT 位 (CTMUCON<24>) 置 1 使能电流源。
4. 产生稳定时间延时。
5. 执行模数转换。
6. 使用 $I = V/R_{CAL}$ 计算电流源电流；其中， R_{CAL} 是高精度电阻， V 通过执行模数转换来测量。

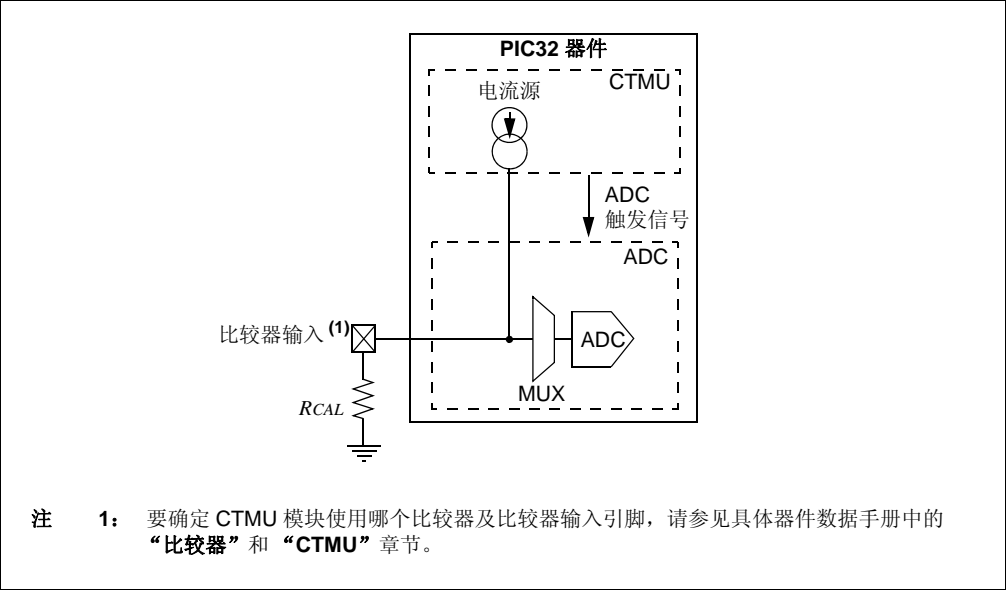
CTMU 电流源可以使用 CTMUCON 寄存器中的微调位进行微调，通过迭代过程来获取所需的精确电流。或者，也可以使用未经调整的标称值。可以由软件存储调整后的电流值，用于所有后续的电容或时间测量。

图 37-2 显示了电流源校准的外部连接，以及所需的不同的模拟模块的关系。

CTMU 电荷泵的校准只有在特殊引脚连接到与 CTMU 相关的比较器模块时才能进行。使用其他 ANx 引脚均会对测量 ADC 前的模拟多路开关增加约 $2.5\text{ k}\Omega$ 的串行电阻值。只有在最小电流设置的情况下才能忽略此影响。对于较大电流设置而言，额外串行电阻值带来的不确定性将会影响校准电阻从而破坏整体电阻值的准确性。

要计算 R_{CAL} 的值，必须选择标称电流，然后就可以计算电阻。例如，如果 ADC 参考电压为 3.3 V ，那么使用满量程的 70% (即 2.31 V) 作为由 ADC 读取的所需近似电压。如果 CTMU 电流源的范围选择为 $0.55\text{ }\mu\text{A}$ ，则所需的电阻值使用 $R_{CAL} = 2.31\text{ V}/0.55\text{ }\mu\text{A}$ 计算，得到值为 $4.2\text{ M}\Omega$ 。类似地，如果电流源选择为 $5.5\text{ }\mu\text{A}$ ， R_{CAL} 将为 $420,000\text{ }\Omega$ ；如果电流源设置为 $55\text{ }\mu\text{A}$ ，则为 $42,000\text{ }\Omega$ 。

图 37-2: CTMU 电流源校准电路



选择 70% 的满量程电压值, 以确保 ADC 处于充分高于基底噪声的范围。如果选择了某个需要结合使用 CTMUCON 寄存器的微调位的精确电流, 则必须要对 RCAL 的电阻值进行相应调整。可能需要调整 RCAL, 以符合可用的电阻值。考虑将使用 CTMU 进行测量的电路所需的精度, RCAL 应选择可用的最高精度。建议最小精度为 0.1% 的容差。

例 37-1 执行 CTMU 电流校准的一种典型方法。该方法通过手动触发 ADC, 这么做是为了逐步演示整个过程。也可以通过将 CTTRIG 位 (CTMUCON<8>) 置 1 来自动触发转换。

注: Microchip 网站上提供基于该校准程序的完整 MPLAB 项目: <http://www.microchip.com/CodeExamplesByFunc.aspx>。进入页面后, 在应用栏位的下拉菜单中选择 Touch Sense (mTouch) (触摸传感)。

例 37-1: 电流校准程序

```

// Device setup - TO DO: Set up device with #pragma's

int main(void)
{
    unsigned long int  ADC_Sum;           // For averaging multiple ADC measurements
    unsigned short int iAvg,              // Averaging index
                    Naverages = 1024,    // Number of averages < 2^22 (22=32-10 bits of ADC)
                    Log2Naverages = 10;  // Right shift equal to 1/Naverages

    short int         iTrim;              // Current trim index
    unsigned short int VmeasADC, VavgADC; // Measured Voltages, 65536 = Full Scale

    // TO DO: Disable JTAG and enable multivector interrupt table
    // Set up UART2 for transmission of button data
    // TO DO: Set up pins

    // CTMU Setup
    CTMUCONbits.TGEN = 1;                 // Enable direct output to C2INB/AN2 pin
    CTMUCONbits.IRNG = 0x3;               // Current Range

    // Turn on CTMU after setting current trim below

    // ADC Setup
    AD1CON2 = 0x0; // VR+ = AVDD, V- = AVSS, Don't scan, MUX A only

    // ADC clock derived from peripheral buss clock
    // Tadc = 4 * Tpbuss = 4 * 25 ns = 100 ns > 65 ns required
    // Tadc = 2*(1+1)*Tpbuss
    // Tadc = 2*(AD1CON3<7:0>+1)*Tpbuss
    AD1CON3 = 1;
    AD1CSSL = 0x0; // No channels scanned
    AD1CHSbits.CHOSA = 2; // Select channel AN2 (PG1D/AN2/C2INB)
    ANSELA = 0x0000; // No ADC pins
    ANSELB = 1<<0; // RB0: for AN2/C2INB, which is connected to Rcal
    ANSELC = 0x0000; // No ADC pins
    IEC0bits.AD1IE = 0; // Disable ADC interrupts
    AD1CON1bits.ON = 1; // Turn on ADC

    // Sweep over all possible current trim values
    for ( iTrim = -31; iTrim < 32; iTrim++ )
    {
        CTMUCONbits.ITRIM = iTrim & 0x3F; // Set current trim value
        CTMUCONbits.ON = 1; // Turn on CTMU
        // TO DO: Add delay of 1 ms

        ADC_Sum = 0;
        for ( iAvg = 0; iAvg < Naverages; iAvg++ )
        {
            CTMUCONCLR = 0x03000000; // Clear Status bits at same time
            AD1CON1bits.SAMP = 1; // Manual sampling start
            CTMUCONbits.IDISSEN = 1; // Ground charge pump

            // TO DO: Add delay of 500 us;
            CTMUCONbits.IDISSEN = 0; // End drain of circuit
            LATASET = 1<<8; // Turn on RA8

            CTMUCONbits.EDG1STAT = 1; // Begin charging the circuit
            LATASET = 1<<7; // Turn on RA7

            // TO DO: Wait 25 us
            AD1CON1bits.SAMP = 0; // Begin analog-to-digital conversion
            CTMUCONbits.EDG2STAT = 1; // Stop charging circuit
            while (!AD1CON1bits.DONE) // Wait for ADC conversion
            {
                // Do Nothing
            }
            AD1CON1bits.DONE = 0; // ADC conversion done, clear flag
            VmeasADC = AD1BUF0; // Get the value from the ADC
            ADC_Sum += VmeasADC; // Update averaging sum

        } //end for ( iAvg = 0; iAvg < Naverages; iAvg++ )
        CTMUCONbits.ON = 0; // Turn off CTMU

        VavgADC = ADC_Sum >> (Log2Naverages-6); // Full scale = 2^10<<6 = 65536

        // TO DO: Transmit iTrim and VavgADC using UART

    } // end for ( iTrim = -31; iTrim < 32; iTrim++ )
} // end main()

```

37.5.2 电容校准

内部 ADC 采样电容和电路板走线与焊盘的杂散电容虽然容值较小，但仍会影响电容测量的精度。在确保排除期望测量的电容的情况下，可以对杂散电容进行测量。杂散电容测量使用以下步骤执行：

1. 初始化 ADC 和 CTMU。
2. 将 EDG1STAT 位置 1 (= 1)。
3. 等待固定延时 t 。
4. 清零 EDG1STAT 位。
5. 执行模数转换。
6. 使用公式 37-5 计算杂散电容和模数转换采样电容。

公式 37-5:

$$C_{OFFSET} = C_{STRAY} + C_{AD} = \frac{(I \cdot t)}{V}$$

其中：

I 从电流源测量步骤获知
 t 是固定延时
 V 通过执行模数转换来测量

然后，可以存储该测量值，用于时间测量时的计算，或在电容测量时减去该值。要进行校准，需要大致了解 $C_{STRAY} + C_{AD}$ 的电容值。 C_{AD} 约为 4 pF。

可能需要使用一个迭代过程来调整时间 t ，该时间是对电路进行充电，以从 ADC 获得合理电压读数的时间。 t 的值可以通过将 C_{OFFSET} 设置为理论值，然后求解 t 来确定。例如，如果 C_{STRAY} 的理论计算值为 11 pF， V 预期为 VDD 的 70% (即 2.31V)，那么 t 将等于公式 37-6 (即 63 μs)。

公式 37-6:

$$t = (4pF + 11pF) \cdot \frac{2.31V}{0.55\mu A} = 63\mu s$$

当系统工作在 40 MHz 时，63 μs 的充电延时意味着 2520 条指令 (2520 = 40 MHz * 63 μs)。如果 CTMU 电荷泵设定为基本电流的 100 倍 (55 μA)，则充电时间除以 100，也就等于 25.2 条指令。这在代码中执行 25 条 NOP 指令便可轻松实现。

例 37-2 显示了在 CTMU 电荷泵校准之后测量电容的一种典型方法。

注：Microchip 网站上提供基于该校准程序的完整 MPLAB 项目：
<http://www.microchip.com/CodeExamplesByFunc.aspx>。进入页面后，在应用栏位的下拉菜单中选择 Touch Sense (mTouch) (触摸传感)。

例 37-2: 电容校准程序

```

// Device setup - TO DO: Set up device with #pragma's

int main(void)
{
    unsigned long int  ADC_Sum;                // For averaging multiple ADC measurements
    unsigned short int iAvg,                   // Averaging index
                    Naverages = 1024,         // Number of averages < 2^22 (22=32-10 bits of ADC)
                    Log2Naverages = 10;       // Right shift equal to 1/Naverages
    short int         iTrim;                   // Current trim index
    unsigned short int VmeasADC, VavgADC;       // Measured Voltages, 65536 = Full Scale

    // TO DO: Disable JTAG and enable multivector interrupt table
    // Set up UART2 for transmission of button data.
    // TO DO: Set up pins

    // CTMU Setup
    CTMUCONbits.TGEN = 1;                      // Enable direct output to C2INB/AN2 pin
    CTMUCONbits.IRNG = PLIB_CTMU_CurrentRange_100xBase; //Current Range: 100 x 0.55 = 55 uA

    // ADC Setup
    AD1CON2 = 0x0;                             // VR+ = AVDD, V- = AVSS, Don't scan, MUX A only

    // ADC clock derived from peripheral buss clock
    // Tadc = 4 * Tpbuss = 4 * 25 ns = 100 ns > 65 ns required
    // Tadc = 2*(1 +1)*Tpbuss
    // Tadc = 2*(AD1CON3<7:0>+1)*Tpbuss
    AD1CON3 = 1;

    AD1CSSL = 0x0;                             // No channels scanned
    AD1CHSbits.CH0SA = 2;                      // Select channel AN2
    ANSELA = 0x0000;                          // No ADC pins
    ANSELB = 1<<0;                            // RB0: for AN2/C2INB, which is connected to Rcal
    ANSELB = 0x0000;                          // No ADC pins
    IEC0bits.AD1IE = 0;                       // Disable ADC interrupts
    AD1CON1bits.ON = 1;                       // Turn on ADC

    // Sweep over all possible current trim values
    for ( iTrim = -31; iTrim < 32; iTrim++ )
    {
        CTMUCONbits.ITRIM = iTrim & 0x3F;      // Set current trim value
        CTMUCONbits.ON = 1;                   // Turn on CTMU
        // TO DO: Wait 1 ms for CTMU start-up

        ADC_Sum = 0;
        for ( iAvg = 0; iAvg < Naverages; iAvg++ )
        {
            AD1CON1bits.SAMP = 1;              // Manual sampling start
            CTMUCONbits.IDISSEN = 1;           // Ground charge pump
            // TO DO: Wait 1 ms for grounding
            CTMUCONbits.IDISSEN = 0;           // End drain of circuit
            CTMUCONbits.EDG1STAT = 1;          // Begin charging the circuit
            // Use FIGURE 37-6 to solve for charge time with current = 55 uA
            // Charge time = 0.63 us, or 25.2 NOPs at 40 MHz system clock
            // TO DO: Wait 25 NOPs
            AD1CON1bits.SAMP = 0;              // Begin analog-to-digital conversion
            CTMUCONbits.EDG1STAT = 0;          // Stop charging circuit
            while (!AD1CON1bits.DONE)          // Wait for ADC conversion
            {
                // Do Nothing
            }
            AD1CON1bits.DONE = 0;              // ADC conversion done, clear flag
            VmeasADC = ADC1BUF0;               // Get the value from the ADC
            ADC_Sum += VmeasADC;               // Update averaging sum
        }

        // end for ( iAvg = 0; iAvg < Naverages; iAvg++ )
    }
}

```

例 37-2: 电容校准程序 (续)

```
CTMUCONbits.ON = 0; // Turn off CTMU

VavgADC = ADC_Sum >> (Log2Naverages-6); //Full scale = 2^10<<6 = 65536

// TO DO: Transmit iTrim and VavgADC using UART

if ( VavgADC > 50000 ) // Stop, ADC voltage too high
{
    break;
}

} // end for ( iTrim = -31; iTrim < 32; iTrim++ )

return 0;

} // end main()
```

37.6 使用 CTMU 测量电容

使用 CTMU 测量电容有两种相对独立的方法。第一种是绝对方法，该方法需要测量实际电容值。第二种是相对方法，该方法不需要实际电容，只需要电容的变化量。

37.6.1 绝对电容测量

注： 为了使 CTMU 能够正常工作，必须对 ADC 进行正确配置。如有必要，请确保 ADC 指向一个未用引脚。

对于绝对电容测量，应遵循第 37.5 节“校准 CTMU 模块”中的电流和电容校准步骤。然后，电容测量使用以下步骤执行：

1. 初始化 ADC。
2. 初始化 CTMU。
3. 将 EDG1STAT 位置 1。
4. 等待固定延时 T 。
5. 清零 EDG1STAT 位。
6. 执行模数转换。
7. 计算总电容 $C_{TOTAL} = (I * T)/V$ ；其中， I 从电流源测量步骤（第 37.5.1 节“电流源校准”）获知， T 是固定延时， V 通过执行模数转换来测量。
8. 将 C_{TOTAL} 减去杂散电容和模数转换采样电容（ $COFFSET$ 来自第 37.5.2 节“电容校准”），就能得出被测电容的值。

37.6.2 相对电荷测量

注： 为了使 CTMU 能够正常工作，必须对 ADC 进行正确配置。如有必要，请确保 ADC 指向一个未用引脚。

有些应用可能并不需要精确的电容测量。例如，在检测电容式开关的有效按压时，只需要检测电容的相对变化。在此类应用中，当开关打开（未被触摸）时，总电容是电路板走线和 ADC 等的组合电容。此时 ADC 将会测量到较大的电压。当开关关闭（被触摸）时，由于以上所列电容中增加了人体的电容，总电容增大，ADC 将测量到较小的电压。

使用 CTMU 检测电容变化可以使用以下步骤实现：

1. 初始化 ADC 和 CTMU。
2. 将 EDG1STAT 位置 1。
3. 等待固定延时。
4. 清零 EDG1STAT 位。
5. 执行模数转换。

通过执行模数转换测量的电压可以反映为相对电容。请注意，在这种情况下，不需要对电流源或电路电容测量进行校准。

例 37-3 给出了电容触摸开关的软件程序示例。在本示例中，先前对于 8 键直接传感器子板的校准可用于平均通过调节每个按钮电路电容不同的充电时间而测得的电压。或者，如果软件支持每个按钮具有独立的触发电平，那么就可以使用固定的充电时间。为了简化起见，本程序中只检查了 8 键直接传感器子板上的第二个按钮（9）。

注： Microchip 网站上提供基于该校准程序的完整 MPLAB 项目：
<http://www.microchip.com/CodeExamplesByFunc.aspx>。进入页面后，在应用栏位的下拉菜单中选择 **Touch Sense (mTouch)**（触摸传感）。

例 37-3: 用于电容触摸开关的程序

```
// Taken from HardwareProfile.h
#define NUM_DIRECT_KEYS 8

static unsigned short int ButtonADCChannels[NUM_DIRECT_KEYS] = {0,1,4,5,6,7,8,9};

// Device setup
//TO DO: Setup up device with #pragma's

int main(void)
{
    unsigned long int  ADC_Sum;           // For averaging multiple ADC measurements
    unsigned short int iAvg,              // Averaging index
                    Naverages = 32,      // Number of averages < 2^22
                    Log2Naverages = 5;   // Right shift equal to 1/Naverages
    short int         iButton,           // Button Index
                    iChan,              // ADC channel index
                    CurrentButtonStatus; // Bit field of buttons that are pressed
    unsigned short int VmeasADC, VavgADC; // Measured Voltages, 65536 = Full Scale
    unsigned short int ButtonVmeasADC[NUM_DIRECT_KEYS]; // Report out all voltages at once

    // TO DO: Disable JTAG and enable multivector interrupt table

    // Set up UART2 for transmission of button data.
    // TO DO: Set up pins

    // CTMU Setup
    CTMUCONbits.IRNG = 0x3;           // Current Range
    CTMUCONbits.ON = 1;               // Turn on CTMU

    // TO DO: Wait 1 ms for CTMU to warm-up

    // ADC Setup
    AD1CON2 = 0x0;                   // VR+ = AVDD, V- = AVSS, Don't scan, MUX A only

    // ADC clock derived from peripheral buss clock
    // Tadc = 4 * Tpbuss = 4 * 25 ns = 100 ns > 65 ns required
    // Tadc = 2*(1+1)*Tpbuss
    // Tadc = 2*(AD1CON3<7:0>+1)*Tpbuss
    AD1CON3 = 1;

    AD1CSSL = 0x0;                   // No channels scanned

    // Taken from mTouchCapADC.c
    ANSELA = (1<<0) | (1<<1); //RA0,1
    ANSELB = (1<<2) | (1<<3) | (1<<13) | (1<<14) | (1<<15); //RB2,3,13,14,15
    ANSELC = (1<<0) | (1<<1) | (1<<2) | (1<<3); //RC0,1,2,3

    IEC0bits.AD1IE = 0;              // Disable ADC interrupts

    AD1CON1bits.ON = 1;              // Turn on ADC

    while ( 1 )
    {
        CurrentButtonStatus = 0;
        for ( iButton = 0; iButton < NUM_DIRECT_KEYS; iButton++ )
        {
            iChan = ButtonADCChannels[iButton];
            AD1CHSbits.CH0SA = iChan;

            ADC_Sum = 0;
            iButton = 2;
            AD1CON1bits.SAMP = 1;           // Manual sampling start
            CTMUCONbits.IDISSEN = 1;        // Ground charge pump
            DelayMs(1);                     // Wait 1 ms for grounding
            CTMUCONbits.IDISSEN = 0;        // End drain of circuit
        }
    }
}
```


例 37-3: 用于电容触摸开关的程序 (续)

```

switch (iButton)
{
    CTMUCONbits.EDG1STAT = 1;           // Begin charging the circuit
    // TO DO: Wait 33 NOPs for Button 2 charge
    AD1CON1bits.SAMP = 0;                // Begin analog-to-digital conversion
    CTMUCONbits.EDG1STAT = 0;           // Stop charging circuit

    while (!AD1CON1bits.DONE)            // Wait for ADC conversion
    {
        // Do Nothing
    }
    AD1CON1bits.DONE = 0;                // ADC conversion done, clear flag
    VmeasADC = ADC1BUF0;                 // Get the value from the ADC
    ADC_Sum += VmeasADC;                 // Update averaging sum
}

} // end for ( iAvg = 0; iAvg < Naverages; iAvg++ )

if ( Log2Naverages-6 > 0 )
{
    VavgADC = ADC_Sum >> (Log2Naverages-6); // Full scale = 2^10<<6 = 65536
}
else
{
    VavgADC = ADC_Sum << (6-Log2Naverages); // Full scale = 2^10<<6 = 65536
}

if ( VavgADC < 32768 )                  // Button is being pressed
{
    CurrentButtonStatus += 1<<iButton;
}
ButtonVmeasADC[iButton] = VavgADC;

} //end while ( 1 )

} //end main()

```

37.7 使用 CTMU 模块测量时间

通过电流和电容校准步骤测量比率 (C/I) 之后，可以使用以下步骤精确测量时间：

1. 初始化 ADC 和 CTMU。
2. 将 EDG1STAT 位置 1。
3. 将 EDG2STAT 位置 1。
4. 执行模数转换。
5. 根据 $T = (C/I) * V$ 计算边沿之间的时间；其中， I 在电流校准步骤（第 37.5.1 节“电流源校准”）中计算， C 在电容校准步骤（第 37.5.2 节“电容校准”）中计算， V 通过执行模数转换来测量。

假定所测量的时间足够小，电容 C_{OFFSET} 可以向 ADC 提供有效的电压。要进行最小的时间测量，请始终将 ADC 通道选择寄存器 (AD1CHS) 设置为悬空的 ADC 输入通道。这可以最大程度地减少杂散电容，保持总电路电容接近于 ADC 自身的电容 (4-5 pF)。要测量较长的时间间隔，可以将一个外部电容连接到 ADC 通道，并在进行时间测量时选择该通道。

37.8 使用 CTMU 模块产生延时

CTMU 模块具有一种独特功能，即它可以根据外部电容值产生独立于系统时钟的输出脉冲。这通过使用内部比较器参考电压模块、比较器 2 输入引脚和外部电容实现。脉冲输出到 CTPLS 引脚上。要使能该模式，需将 TGEN 位置 1。

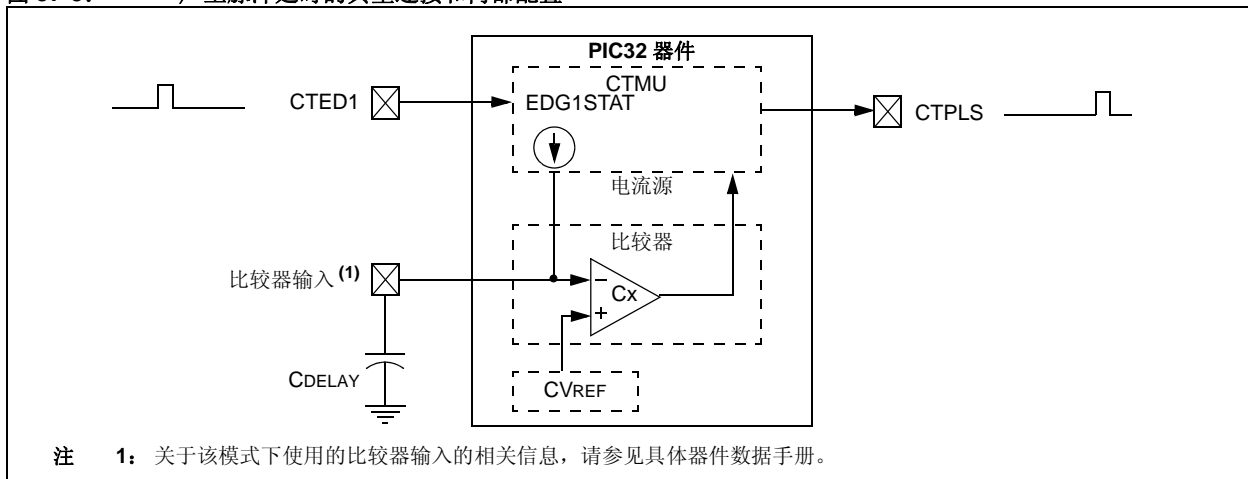
图 37-3 给出了示例电路。 C_{DELAY} 由用户选择，用于确定 CTPLS 上的输出脉冲宽度。脉冲宽度根据 $T = (C_{DELAY}/I) * V$ 计算；其中， I 从电流源测量步骤（第 37.5.1 节“电流源校准”）获知， V 是内部参考电压 (CVREF)。

该功能的使用示例是连接基于可变电容的传感器，如湿度传感器。当湿度发生变化时，CTPLS 上的脉宽输出也会变化。CTPLS 输出引脚可以连接到输入捕捉引脚，通过测量变化的脉冲宽度来确定应用环境的湿度。

执行以下步骤来使用该功能：

1. 初始化比较器 2。
2. 初始化比较器参考电压。
3. 初始化 CTMU，并通过将 TGEN 位置 1 来使能延时产生。
4. 将 EDG1STAT 位置 1。
5. 当 C_{DELAY} 充电到参考电压跳变点的值时，在 CTPLS 上会产生输出脉冲。

图 37-3：产生脉冲延时的典型连接和内部配置



37.9 使用 CTMU 测量片上温度

CTMU 模块可用于测量器件的内部温度，具体方法是通过一个可用的内部二极管。当 EDG1STAT 不等于 EDG2STAT，且 TGEN = 0 时，电流会被引导到温度检测二极管。二极管两端的电压可用作 ADC 模块的输入。

图 37-4 给出了如何使用该模块测量温度的图示。当温度上升时，二极管两端的电压将下降大约 300 mV（对应于 150°C 的温度范围）。选择较高的电流驱动能力可以使电压值上升 100 mV 左右。

37.9.1 基本原则

P-N 结（如二极管）的正向电压（ V_f ）是结点的热电压公式的扩展，如公式 37-7 所示：

公式 37-7:

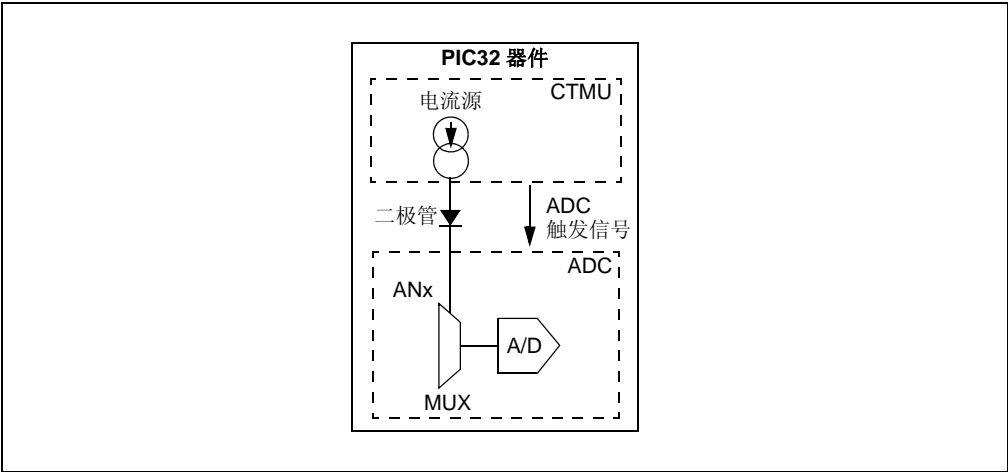
$$V_f = (kT/q) \ln (1 - I_f / I_s)$$

其中：
 k 是波尔茨曼常数（ $1.38 \times 10^{-23} \text{ J K}^{-1}$ ）
 T 是以开尔文为单位的绝对结温
 q 是电子电荷（ $1.6 \times 10^{-19} \text{ C}$ ）
 I_f 是对二极管施加的正向电流
 I_s 是二极管的特性饱和电流

由于 k 和 q 是物理常数， I_s 是器件的一个常量，所以只剩下 T 和 I_f 是独立变量。如果 I_f 保持恒定，根据公式可以推得 V_f 将为 T 的函数。由于公式的自然对数项总是为负，所以温度会与 V_f 成正比，但符号相反。

即，当温度升高时， V_f 会下降。

图 37-4: CTMU 温度测量电路



37.10 休眠 / 空闲模式期间的操作

37.10.1 休眠模式

当器件进入休眠模式时，CTMU 模块电流源将始终禁止。如果调用休眠模式时，CTMU 正在执行依赖于电流源的操作，则操作可能不会正确终止。电容和时间测量可能会返回错误值。

37.10.2 空闲模式

CTMU 在空闲模式下的行为由 CTMUSIDL 位（CTMUCON<13>）决定。如果 CTMUSIDL 位清零，在空闲模式下，模块将继续工作。如果 CTMUSIDL 位置 1，则在器件进入空闲模式时，模块的电流源会被禁止。如果调用空闲模式时，模块正在执行操作，这种情况下，结果将类似于休眠模式下的结果。

37.11 复位对 CTMU 的影响

在复位时，CTMU 的所有寄存器都会被清零。这使 CTMU 模块处于禁止状态，它的电流源被关闭，所有配置选项恢复为它们的默认设置。在任意复位之后，模块都需要重新初始化。

如果发生复位时，CTMU 正在进行测量，测量结果将丢失。正在测量的电路可能会存在部分充电的情况，在随后 CTMU 尝试进行测量之前，应正确进行放电。电路放电方法是，在 ADC 连接到相应通道的同时，先将 IDISSEN 位（CTMUCON<9>）置 1，然后再将其清零。

37.12 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC32 器件系列而编写的，但其概念是相近的，通过适当修改并受到一定限制即可使用。当前与充电时间测量单元 (CTMU) 模块相关的应用笔记有：

标题

应用笔记编号

目前没有相关的应用笔记。

N/A

注： 如需获取更多 PIC32 系列器件的应用笔记和代码示例，请访问 Microchip 网站 (www.microchip.com)。

37

充电时间测量单元
(CTMU)

37.13 版本历史

版本 A（2011 年 3 月）

这是本文档的初始版本。

版本 B（2012 年 2 月）

该版本包含以下更新：

- 向第 37.3.2 节“**电流源**”添加注 1 和注 2
- 更新第 37.3.4 节“**边沿状态**”中第一段的最后一句话
- 向 CTMU 电流源校准电路添加注 1（见图 37-2）
- 移除 CTMU 校准设置程序（之前的例 37-1）
- 更新电流校准程序的代码（见例 37-1）
- 更新电容校准程序的代码（见例 37-2）
- 更新用于电容触摸开关程序的代码（见例 37-3）
- 对全文的文本和格式做了少量更新

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。

商标

Microchip 的名称和徽标组合、Microchip 徽标、dsPIC、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC³² 徽标、rPIC 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、MXDEV、MXLAB、SEEVAL 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、Application Maestro、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICTail、REAL ICE、rLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock 和 ZENA 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。

在此提及的所有其他商标均为各持有公司所有。

© 2011-2012, Microchip Technology Inc. 版权所有。

ISBN: 978-1-62076-214-1

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

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[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和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

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: 852-2401-1200
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: 86-755-8203-2660
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: 886-7-536-4818
Fax: 886-7-330-9305

台湾地区 - 台北
Tel: 886-2-2500-6610
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: 81-66-152-7160
Fax: 81-66-152-9310

日本 Japan - Yokohama
Tel: 81-45-471-6166
Fax: 81-45-471-6122

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