



第 11 章 充电时间测量单元 (CTMU)

目录

本章包括下列主题:

11.1 简介.....	11-2
11.2 寄存器.....	11-3
11.3 CTMU 工作原理.....	11-5
11.4 CTMU 模块初始化.....	11-6
11.5 校准 CTMU 模块.....	11-7
11.6 使用 CTMU 测量电容.....	11-12
11.7 使用 CTMU 模块测量时间.....	11-14
11.8 使用 CTMU 模块产生延时.....	11-14
11.9 休眠 / 空闲模式期间的操作.....	11-16
11.10 复位对 CTMU 的影响.....	11-16
11.11 寄存器映射.....	11-17
11.12 电气规范.....	11-18
11.13 相关应用笔记.....	11-19
11.14 版本历史.....	11-20

11.1 简介

充电时间测量单元（Charge Time Measurement Unit, CTMU）是一个灵活的模拟模块，它提供脉冲源之间的精确时间差测量，以及异步脉冲生成。CTMU 可与其他片上模拟模块一起，用于精确测量时间、电容、电容的相对变化，或生成具有特定延时的输出脉冲。CTMU 是与电容式传感器接口的理想选择。

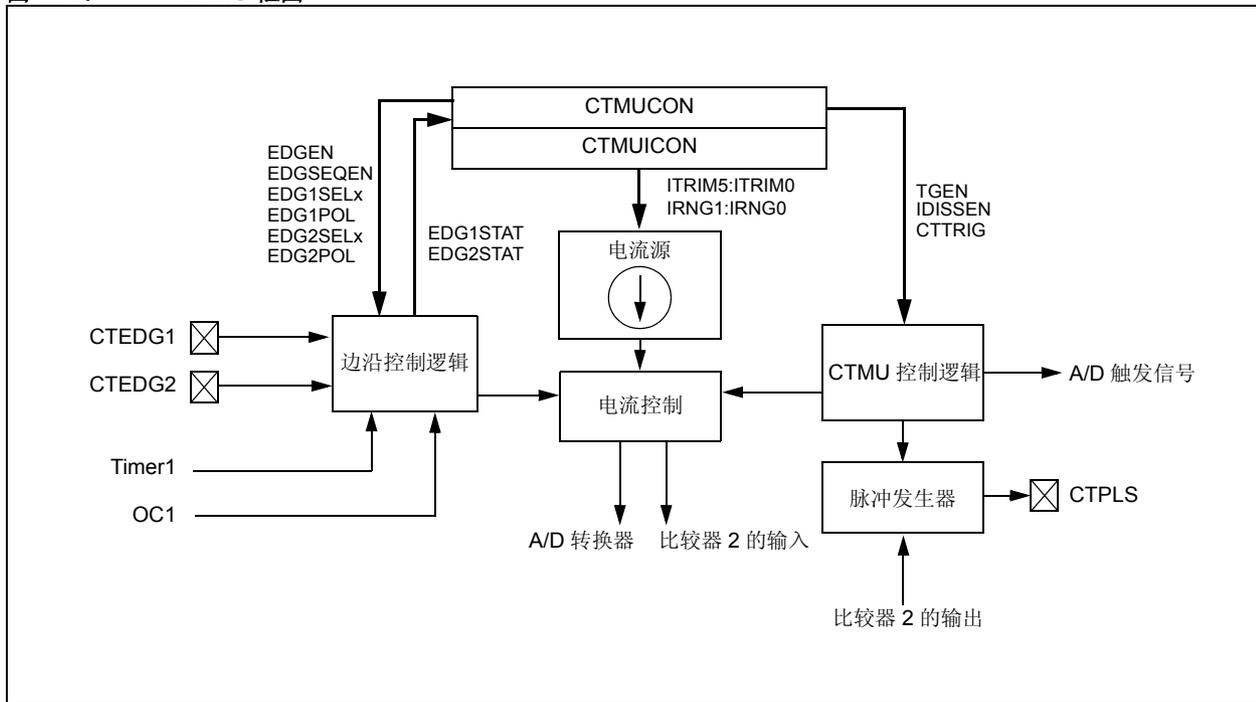
该模块具有以下主要特性：

- 最多 16 路通道，可用于电容或时间测量输入
- 片上精确电流源
- 4 个边沿输入触发源
- 每个边沿源的极性控制
- 边沿序列控制
- 控制对边沿的响应
- 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

CTMU 与 A/D 转换器配合工作，根据具体器件和可用的 A/D 通道数，最多可提供 16 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可以从 4 个源中选择：两个外部输入、Timer1 或输出比较模块 1。关于可用输入源的具体器件信息，请参见相应的 PIC24F 数据手册。

图 11-1 给出了 CTMU 的框图。

图 11-1: CTMU 框图



11.2 寄存器

有两个用于 CTMU 的控制寄存器：CTMUCON 和 CTMUICON。

CTMUCON 寄存器（寄存器 11-1）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、边沿序列、A/D 触发器、模拟电路电容放电和使能。CTMUICON 寄存器（寄存器 11-2）包含一些用于选择电流源范围和电流源微调的位。

寄存器 11-1: CTMUCON: CTMU 控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EDG2POL	EDG2SEL1 ⁽¹⁾	EDG2SEL0 ⁽¹⁾	EDG1POL	EDG1SEL1 ⁽¹⁾	EDG1SEL0 ⁽¹⁾	EDG2STAT	EDG1STAT
bit 7							bit 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 15 **CTMUEN:** CTMU 使能位
1 = 使能模块
0 = 禁止模块
- 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:** 时间生成使能位
1 = 使能边沿延时生成
0 = 禁止边沿延时生成
- bit 10 **EDGEN:** 边沿使能位
1 = 未阻止边沿
0 = 阻止边沿
- bit 10 **EDGSEQEN:** 边沿序列使能位
1 = 边沿 1 事件必须在边沿 2 事件发生前发生
0 = 无需边沿序列
- bit 9 **IDISSEN:** 模拟电流源控制位
1 = 模拟电流源输出接地
0 = 模拟电流源输出未接地
- bit 8 **CTTRIG:** 触发器控制位
1 = 使能触发器输出
0 = 禁止触发器输出
- bit 7 **EDG2POL:** 边沿 2 极性选择位
1 = 边沿 2 设定为正电平响应
0 = 边沿 2 设定为负电平响应

注 1: 关于具体边沿源类型和分配情况的信息，请参见特定的器件数据手册。

PIC24F 系列参考手册

寄存器 11-1: CTMUCON: CTMU 控制寄存器 (续)

- bit 6-5 **EDG2SEL1:EDG2SEL0:** 边沿 2 源选择位 ⁽¹⁾
 11 = 选择边沿源 3
 10 = 选择边沿源 2
 01 = 选择边沿源 1
 00 = 选择边沿源 0
- bit 4 **EDG1POL:** 边沿 1 极性选择位
 1 = 边沿 1 设定为正电平响应
 0 = 边沿 1 设定为负电平响应
- bit 3-2 **EDG1SEL1:EDG1SEL0:** 边沿 1 源选择位 ⁽¹⁾
 11 = 选择边沿源 3
 10 = 选择边沿源 2
 01 = 选择边沿源 1
 00 = 选择边沿源 0
- bit 1 **EDG2STAT:** 边沿 2 状态位
 1 = 已发生边沿 2 事件
 0 = 未发生边沿 2 事件
- bit 0 **EDG1STAT:** 边沿 1 状态位
 1 = 已发生边沿 1 事件
 0 = 未发生边沿 1 事件

注 1: 关于具体边沿源类型和分配情况的信息, 请参见特定的器件数据手册。

寄存器 11-2: CTMUICON: CTMU 电流控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
bit 15						bit 8	

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7						bit 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 15-10 **ITRIM5:ITRIM0:** 电流源微调位
 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 = IRNG1:IRNG0 指定的标称电流输出
 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
- bit 9-8 **IRNG1:IRNG0:** 电流源范围选择位
 11 = 100 × 基本电流
 10 = 10 × 基本电流
 01 = 基本电流水平 (标称值为 0.55 μA)
 00 = 禁止电流源
- bit 7-0 **未实现:** 读为 0

11.3 CTMU 工作原理

CTMU 的工作方式是使用固定电流源来对电路进行充电。电路的类型取决于要进行的测量的类型。在进行电荷测量的情况下，电流是固定的，向电路施加电流的时间也是固定的。这样，只要通过 A/D 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。这种情况下，由 A/D 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，以及向比较器电路提供的电压都是固定的。信号的延时由将电压充电到比较器门限电压所需的时间决定。

11.3.1 工作原理

CTMU 的工作原理基于以下电荷公式：

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

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

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

CTMU 模块提供了恒定、已知的电流源。A/D 转换器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 ）。以上公式可用于计算电容或时间，根据以下关系：

$$t = (C \cdot V) / I$$

使用电路的已知固定电容；或根据：

$$C = (I \cdot t) / V$$

使用电流源施加于电路的固定时间。

11.3.2 电流源

CTMU 的核心是高精度电流源，旨在提供用于测量的恒定基准。用户可以从三个范围或总共两个数量级的电平中选择电流电平，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 IRNG1:IRNG0 位（CTMUICON<9:8>）进行选择，值 00 代表最低范围。

电流微调通过 ITRIM5:ITRIM0 位（CTMUICON<15:10>）进行。这 6 个位使得可以按大约每步 2% 的步阶微调电流源。请注意，其中一半的范围用于增大电流源电平，另一半用于降低电流源电平。值 000000 是中间位置（无变化）。值 100000 代表最大负调整（大约 -62%），011111 代表最大正调整（大约 +62%）。

11.3.3 边沿选择和控制

CTMU 测量通过在模块的两路输入通道中发生的边沿事件进行控制。每路通道（称为边沿 1 和边沿 2）可以配置为接收来自一个边沿输入引脚（CTEDG1 和 CTEDG2）、Timer1 或输出比较模块 1 的输入脉冲。输入通道是电平敏感通道，响应通道中的瞬时电平，而不是电平跳变。输入使用 EDG1SEL 和 EDG2SEL 位对（CTMUICON<3:2> 和 <6:5>）选择。

除了电流源之外，还可以使用 EDGE2POL 和 EDGE1POL 位（CTMUICON<7,4>）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序列（边沿 1 在边沿 2 之前发生），方法是将 EDGSEQEN 位（CTMUICON<10>）置 1。

11.3.4 边沿状态

CTMUCON 寄存器还包含两个状态位 EDG2STAT 和 EDG1STAT (CTMUCON<1:0>)。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 会自动将特定的位置 1。输入通道的电平敏感特性也意味着, 如果通道的配置或其电流状态发生改变, 那么状态位会立即置 1。

模块使用边沿状态位来控制到外部模拟模块 (如 A/D 转换器) 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。两个位应同时清零 (如果可能), 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。

11.3.5 中断

每当电流源先使能, 然后禁止时, CTMU 就会将其中断标志 (IFS4<13>) 置 1。只有相应的中断允许位 (IEC4<13>) 也置 1 时, 才会产生中断。如果未使能边沿序列 (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。

11.4 CTMU 模块初始化

以下序列是用于初始化 CTMU 模块的通用指南:

1. 使用 IRNG 位 (CTMUICON<9:8>) 选择电流源范围。
2. 使用 ITRIM 位 (CTMUICON<15:10>) 微调电流源。
3. 通过设置 EDG1SEL 和 EDG2SEL 位 (CTMUICON<3:2 和 6:5>) 配置边沿 1 和边沿 2 的边沿输入源。
4. 使用 EDG1POL 和 EDG2POL 位 (CTMUICON<4,7>) 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 (CTMUICON<10>) 使能边沿序列。默认情况下, 将禁止边沿序列。
6. 使用 TGEN 位选择工作模式 (测量或产生延时)。默认模式是时间 / 电容测量。
7. 使用 CTTRIG 位 (CTMUICON<8>) 将模块配置为在发生第二个边沿事件时自动触发 A/D 转换。默认情况下, 会禁止转换触发器。
8. 通过将 IDISSEN 位 (CTMUICON<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。
9. 通过清零 CTMUEN 位 (CTMUICON<15>) 禁止该模块。
10. 清零边沿状态位 EDG2STAT 和 EDG1STAT (CTMUICON<1:0>)。
11. 通过将 EDGEN 位 (CTMUICON<11>) 置 1 使能两个边沿输入。
12. 通过将 CTMUEN 位置 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, Timer1 和输出比较/PWM1 模块也可以用作 CTMU 的边沿源。
- 电容或时间测量: CTMU 模块使用 A/D 转换器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 PIC24F 系列参考中相应模块的章节。

11.5 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容性触摸开关，在该应用中，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。

11.5.1 电流源校准

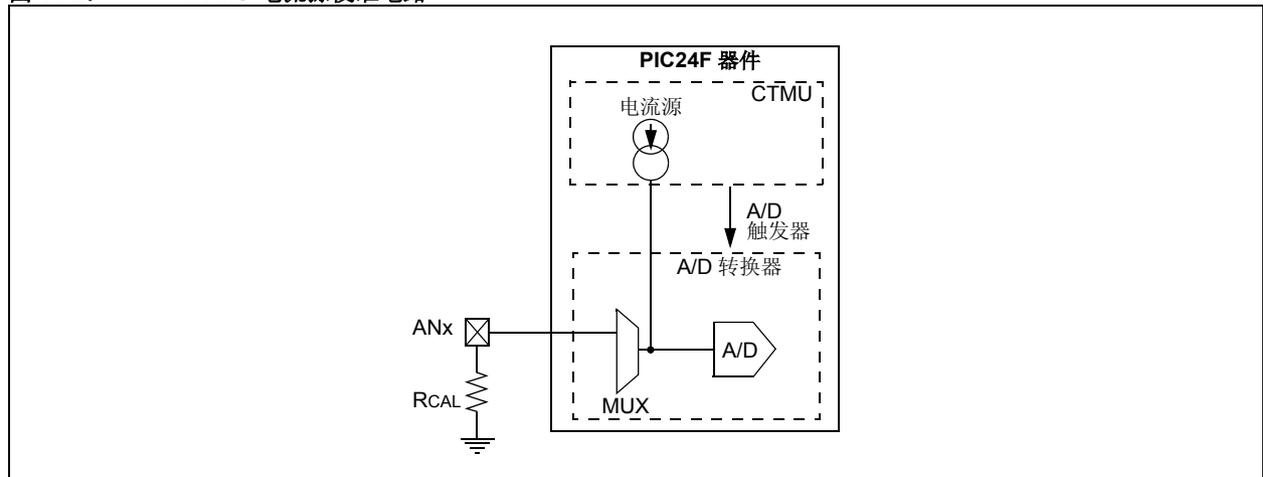
CTMU 模块随附的电流源具有三种电平范围，其中每种范围都可以在其标称值 $\pm 60\%$ 的范围内进行调节。因此，要进行精确测量，可以通过在未用模拟通道上放置一个高精度电阻 R_{CAL} ，测量并调整该电流源。图 11-2 给出了示例电路。电流源测量使用以下步骤执行：

1. 初始化 A/D 转换器。
2. 初始化 CTMU。
3. 通过将 EDG1STAT (CTMUCON<0>) 置 1 使能电流源。
4. 产生稳定时间延时。
5. 执行 A/D 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 A/D 转换来测量。

CTMU 电流源可以使用 CTMUICON 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。

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 A/D 转换器参考电压为 3.3V，使用满量程的 70% (或 2.31V) 作为要由 A/D 转换器读取的所需近似电压。如果 CTMU 电流源的范围选择为 $0.55 \mu\text{A}$ ，所需的电阻值使用 $R_{CAL} = 2.31\text{V}/0.55 \mu\text{A}$ 计算，得到值为 $4.2 \text{M}\Omega$ 。类似地，如果电流源选择为 $5.5 \mu\text{A}$ ， R_{CAL} 将为 $420,000\Omega$ ；如果电流源设置为 $55 \mu\text{A}$ ，则为 $42,000\Omega$ 。

图 11-2: CTMU 电流源校准电路



选择满量程电压 70% 的值，以确保 A/D 转换器处于充分高于基底噪声的范围。请记住，如果选择了某个需要结合使用 CTMUICON 的微调位的精确电流，则可能需要对 RCAL 的电阻值进行相应调整。可能还需要再次调整 RCAL，以允许选择可用的电阻值。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。建议最小精度是允许 0.1% 的误差。

以下示例给出了执行 CTMU 电流校准的一种典型方法。例 11-1 显示如何初始化 A/D 转换器和 CTMU；该程序是同时使用两个模块的应用的典型程序。例 11-2 给出了实际校准程序的一种方法。请注意，该方法需要手动触发 A/D 转换器，这么做是为了演示整个逐步过程。也可以通过将 CTMU 的 CTRIG 位 (CTMUCON<8>) 置 1 来自动触发转换。

例 11-1: CTMU 校准设置程序

```
#include "p24Fxxxx.h"

/*****
/*Setup CTMU
*****/

void setup(void)
{
    //CTMUCON - CTMU Control register
    CTMUCON = 0x0090; //make sure CTMU is disabled
    //CTMU continues to run when emulator is stopped,CTMU continues
    //to run in idle mode,Time Generation mode disabled, Edges are blocked
    //No edge sequence order, Analog current source not grounded, trigger
    //output disabled, Edge2 polarity = positive level, Edge2 source =
    //source 0, Edge1 polarity = positive level, Edge1 source = source 0,
    // Set Edge status bits to zero

    //CTMUICON - CTMU Current Control Register
    CTMUICON = 0x0001; //0.55uA, Nominal - No Adjustment
/*****
//setupAD converter;
*****/

    TRISB=0x0004; //set channel 2 as an input
    AD1PCFG=0x0004; //
    AD1CHS=0x002; //select the analog channel(2)
    AD1CSSL=0x0000; //

    AD1CON1 = 0x8000; //Turn On A/D Converter, continue in
    // idle mode, Unsigned fractional format, Clear
    //conversion, Sample when SAMP bit is set,
    //sampling on hold
    AD1CON2 = 0x0000; //VR+ = AVDD, V- = AVSS, Don't scan,
    //always use MUX A inputs
    AD1CON3 = 0x0000; //A/D uses system clock, conversion
    //clock = 1xTcy
}
}
```

例 11-2: 电流校准程序

```

#include "p24Fxxxx.h"

#define COUNT 500          //@ 8MHz = 125uS.
#define DELAY for(i=0;i<COUNT;i++)
#define RCAL .027          //R value is 4200000 (4.2M)
                              //scaled so that result is in
                              //1/100th of uA
#define ADSCALE 1023      //for unsigned conversion 10 sig bits
#define ADREF 3.3         //Vdd connected to A/D Vr+

int main(void)
{
    int i;
    int j = 0;              //index for loop
    unsigned int Vread = 0;
    double VTot = 0;
    float Vavg=0, Vcal=0, CTMUISrc = 0; //float values stored for calcs

    //assume CTMU and A/D have been setup correctly
    //see Example 11-1 for CTMU & A/D setup
    setup();

    CTMUCONbits.CTMUEN = 1; //Enable the CTMU

    for(j=0;j<10;j++)
    {
        AD1CON1bits.SAMP = 1; //Manual sampling start
        CTMUCONbits.IDISSEN = 1; //drain charge on the circuit
        DELAY; //wait 125us
        CTMUCONbits.IDISSEN = 0; //end drain of circuit

        CTMUCONbits.EDG1STAT = 1; //Begin charging the circuit
        //using CTMU current source
        DELAY; //wait for 125us
        CTMUCONbits.EDG1STAT = 0; //Stop charging circuit

        IFS0bits.AD1IF = 0; //make sure A/D Int not set
        AD1CON1bits.SAMP = 0; //and begin A/D conv.
        while(!IFS0bits.AD1IF); //Wait for A/D convert complete
        AD1CON1bits.DONE = 0;
        Vread = ADC1BUF0; //Get the value from the A/D
        IFS0bits.AD1IF = 0; //Clear A/D Interrupt Flag
        VTot += Vread; //Add the reading to the total
    }

    Vavg = (float)(VTot/10.000); //Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL; //CTMUISrc is in 1/100ths of uA
}

```

11.5.2 电容校准

内部 A/D 转换器采样电容和电路板走线与焊垫的杂散电容虽然容值较小但仍会影响电容测量的精度。在确保先取下期望测量的电容的情况下，可以对杂散电容进行测量。然后，测量使用以下步骤执行：

1. 初始化 A/D 转换器和 CTMU。
2. 将 EDG1STAT 置 1 (= 1)。
3. 等待固定延时 t 。
4. 清零 EDG1STAT。
5. 执行 A/D 转换。
6. 计算杂散电容和 A/D 采样电容：

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

其中， I 从电流源测量步骤获知， t 是固定延时， V 通过执行 A/D 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 $C_{\text{STRAY}} + C_{\text{AD}}$ 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 A/D 转换器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 V_{DD} 的 70%（或 2.31V），那么 t 为：

$$(4 \text{ pF} + 11 \text{ pF}) \cdot 2.31\text{V} / 0.55 \mu\text{A}$$

或 63 μs 。

例 11-3 给出了 CTMU 电容校准的典型程序。

例 11-3: 电容校准程序

```

#include "p24Fxxxx.h"

#define COUNT 25          //@ 8MHz INTFRC = 62.5 us.
#define ETIME COUNT*2.5  //time in us
#define DELAY for(i=0;i<COUNT;i++)
#define ADSCALE 1023    //for unsigned conversion 10 sig bits
#define ADREF 3.3       //Vdd connected to A/D Vr+

int main(void)
{
    int i;
    int j = 0;           //index for loop
    unsigned int Vread = 0;
    float CTMUISrc, CTMUCap, Vavg, VTot, Vcal;

                                //assume CTMU and A/D have been setup correctly
                                //see Example 11-1 for CTMU & A/D setup

    setup();

    CTMUCONbits.CTMUEN = 1;//Enable the CTMU

    for(j=0;j<10;j++)
    {
        AD1CON1bits.SAMP = 1;    //Manual sampling start
        CTMUCONbits.IDISSEN= 1;  //drain any charge on the circuit
        DELAY;                   //wait 62.5 us
        CTMUCONbits.IDISSEN = 0; //end drain of circuit
        CTMUCONbits.EDG1STAT = 1; //Begin charging the circuit
                                //using the CTMU current source
        DELAY;                   //wait for 62.5 us for circuit to charge
        CTMUCONbits.EDG1STAT = 0; //Stop charging circuit and begin A/D conv.
        AD1CON1bits.SAMP = 0;
        while(!IFS0bits.AD1IF); //Wait for A/D conversion to complete
        Vread = ADC1BUF0;        //Get the value from the A/D converter
        IFS0bits.AD1IF = 0;     //Clear AD1IF
        VTot += Vread;          //Add the reading to the total
    }

    Vavg = (VTot/10); //Average of 10 readings
    Vcal = (Vavg/ADSCALE*ADREF);
    CTMUCap = (CTMUISrc*ETIME/Vcal)/100;
    //CTMUISrc is in 1/100ths of uA,
    //calculated in Example 1-2
    //time is in us
    //CTMUCap is in pF
}

```

11.6 使用 CTMU 测量电容

使用 CTMU 测量电容有两种相互独立的方法。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。

11.6.1 绝对电容测量

对于绝对电容测量，应遵循第 11.5 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 A/D 转换器。
2. 初始化 CTMU。
3. 将 EDG1STAT 置 1。
4. 等待固定延时 T 。
5. 清零 EDG1STAT。
6. 执行 A/D 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 11.5.1 节“电流源校准”）获知， T 是固定延时， V 通过执行 A/D 转换来测量。
8. 从 C_{TOTAL} 中减去杂散电容和 A/D 采样电容（ $COFFSET$ 来自第 11.5.2 节“电容校准”），确定被测电容的值。

11.6.2 相对电荷测量

有些应用可能并不需要精确的电容测量。例如，在检测基于电容的开关的有效按压时，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、A/D 转换器等组合电容。此时 A/D 转换器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，A/D 转换器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简单实现：

1. 初始化 A/D 转换器和 CTMU。
2. 将 EDG1STAT 置 1。
3. 等待固定延时。
4. 清零 EDG1STAT。
5. 执行 A/D 转换。

通过执行 A/D 转换测量的电压可以指示相对电容。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 11-4 给出了容性触摸开关的软件程序示例。

例 11-4: 用于容性触摸开关的程序

```

#include "p24Fxxxx.h"

#define COUNT 500    //@ 8MHz = 125uS.
#define DELAY for(i=0;i<COUNT;i++)
#define OPENSW 1000 //Unpressed switch value
#define TRIP 300    //Difference between pressed
                    //and unpressed switch
#define HYST 65     //amount to change
                    //from pressed to unpressed

#define PRESSED 1
#define UNPRESSED 0

int main(void)
{
    unsigned int Vread;    //storage for reading
    unsigned int switchState;
    int i;

                                //assume CTMU and A/D have been setup correctly
                                //see Example 11-1 for CTMU & A/D setup

    setup();

    CTMUCONbits.CTMUEN = 1;    //Enable the CTMU

    AD1CON1bits.SAMP = 1;    //Manual sampling start
    CTMUCONbits.IDISSEN = 1; //drain charge on the circuit
    DELAY;                    //wait 125us
    CTMUCONbits.IDISSEN = 0; //end drain of circuit

    CTMUCONbits.EDG1STAT = 1; //Begin charging the circuit
                                //using CTMU current source
    DELAY;                    //wait for 125us
    CTMUCONbits.EDG1STAT = 0; //Stop charging circuit

    IFS0bits.AD1IF = 0;    //make sure A/D Int not set
    AD1CON1bits.SAMP = 0;    //and begin A/D conv.
    while(!IFS0bits.AD1IF); //Wait for A/D convert complete
    AD1CON1bits.DONE = 0;
    Vread = ADC1BUF0;    //Get the value from the A/D
    if(Vread < OPENSW - TRIP)
    {
        switchState = PRESSED;
    }
    else if(Vread > OPENSW - TRIP + HYST)
    {
        switchState = UNPRESSED;
    }
}

```

11.7 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 A/D 转换器和 CTMU。
2. 将 EDG1STAT 置 1。
3. 将 EDG2STAT 置 1。
4. 执行 A/D 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 11.5.1 节 “电流源校准”) 中计算, C 在电容校准步骤 (第 11.5.2 节 “电容校准”) 中计算, V 通过执行 A/D 转换来测量。

假定所测量的时间足够小, 电容 COFFSET 可以向 A/D 转换器提供有效的电压。要进行最小的时间测量, 请始终将 A/D 通道选择寄存器 (AD1CHS) 设置为未用的 A/D 通道; 该通道的相应引脚不连接到任何电路板走线。这可以最大程度降低所增加的杂散电容, 保持总电路电容接近于 A/D 转换器自身的电容 (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 A/D 通道, 并在进行时间测量时选择该通道。

11.8 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。

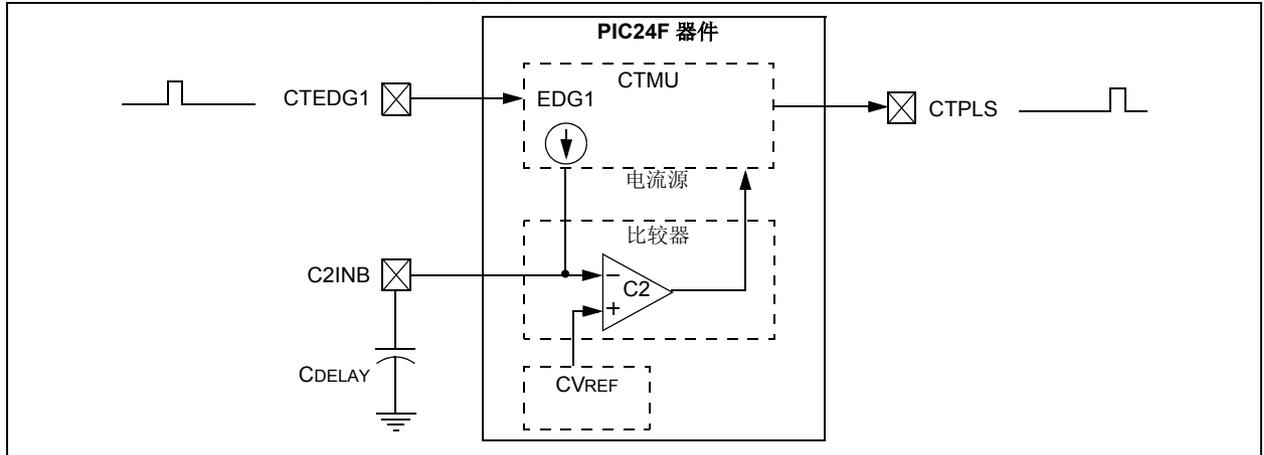
图 11-3 给出了示例电路。CPULSE 由用户选择, 用于确定 CTPLS 上的输出脉冲宽度。脉冲宽度根据 $T = (CPULSE/I) * V$ 计算; 其中, I 从电流源测量步骤 (第 11.5.1 节 “电流源校准”) 获知, V 是内部参考电压 (CVREF)。

该功能的使用示例是连接基于可变电容的传感器, 例如湿度传感器。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 来使能延时生成。
4. 将 EDG1STAT 置 1。
5. 当 CPULSE 充电到参考电压跳变点的值时, 在 CTPLS 上会产生输出脉冲。

图 11-3: 生成脉冲延时的典型连接和内部配置



11.9 休眠 / 空闲模式期间的操作

11.9.1 休眠模式和深度休眠模式

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。

11.9.2 空闲模式

CTMU 在空闲模式下的行为由 CTMUSIDL 位 (CTMUCON<13>) 决定。如果 CTMUSIDL 清零，在空闲模式下，模块将继续工作。如果 CTMUSIDL 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的结果。

11.10 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 A/D 转换器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。

11.11 寄存器映射

表 11-1 中提供了与 PIC24F CTMU 相关的寄存器汇总。

表 11-1: CTMU 寄存器映射

寄存器名称	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
CTMUCON	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	0000
CTMUICON	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	—	0000

图注: — = 未实现 (读为 0)。复位值以十六进制显示。

11.12 电气规范

表 11-2: CTMU 电流源规范

直流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
参数编号	符号	特性	最小值	典型值 (1)	最大值	单位	条件
	IOUT1	CTMU 电流源, 基本范围	—	550	—	nA	CTMUICON<1:0> = 01
	IOUT2	CTMU 电流源, 10x 范围	—	5.5	—	μA	CTMUICON<1:0> = 10
	IOUT3	CTMU 电流源, 100x 范围	—	55	—	μA	CTMUICON<1:0> = 11

注 1: 电流微调范围的中点 (CTMUICON<7:2> = 000000) 为标称值。

11.13 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 PIC24F 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 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。

注： 如需获取更多 PIC24F 系列器件的应用笔记和代码示例，请访问 Microchip 网站 (www.microchip.com)。

11.14 版本历史

版本 A（2008 年 3 月）

这是本文档的初始版本。