
Section 11. Charge Time Measurement Unit (CTMU)

HIGHLIGHTS

This section of the manual contains the following major topics:

11.1	Introduction	11-2
11.2	Registers	11-3
11.3	CTMU Operation	11-5
11.4	CTMU Module Initialization	11-6
11.5	Calibrating the CTMU Module	11-7
11.6	Measuring Capacitance with the CTMU	11-12
11.7	Measuring Time with the CTMU Module	11-14
11.8	Creating a Delay with the CTMU Module	11-14
11.9	Operation During Sleep/Idle Modes	11-16
11.10	Effects of a Reset on CTMU	11-16
11.11	Register Maps	11-17
11.12	Electrical Specifications	11-18
11.13	Related Application Notes	11-19
11.14	Revision History	11-20

11.1 INTRODUCTION

The Charge Time Measurement Unit (CTMU) is a flexible analog module that provides accurate differential time measurement between pulse sources, as well as asynchronous pulse generation. By working with other on-chip analog modules, the CTMU can be used to precisely measure time, measure capacitance, measure relative changes in capacitance or generate output pulses with a specific time delay. The CTMU is ideal for interfacing with capacitive-based sensors.

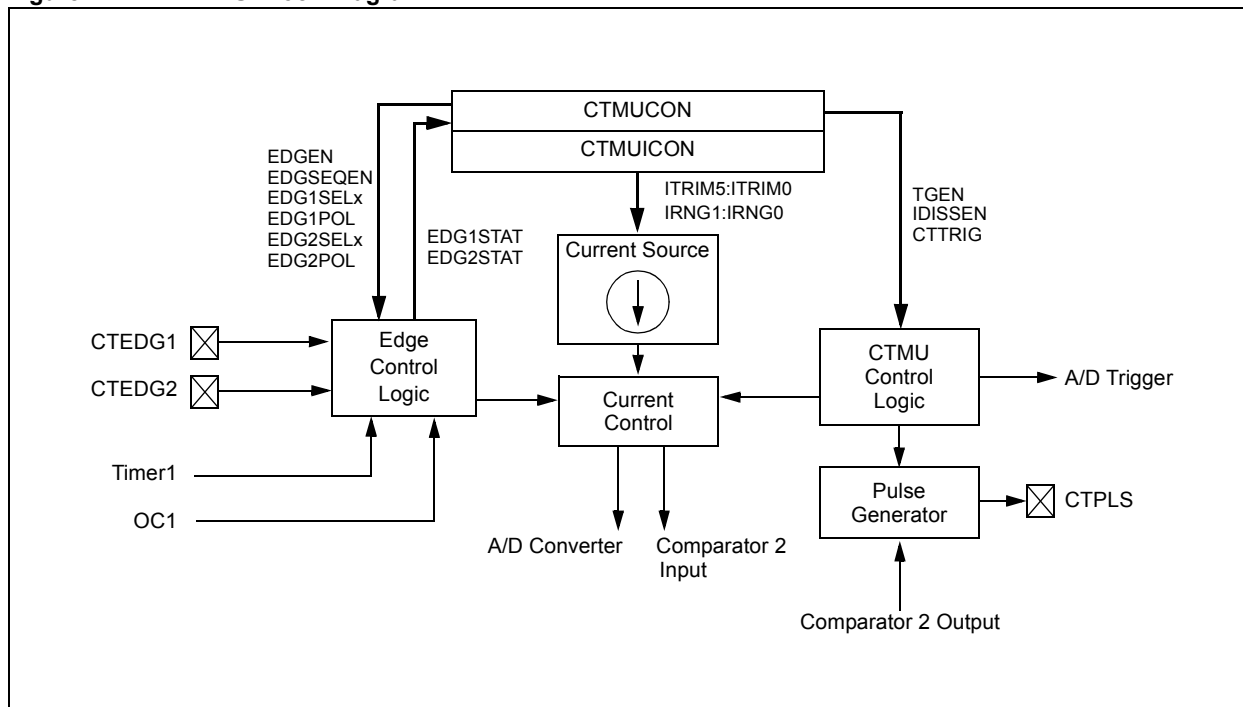
The module includes the following key features:

- Up to 16 channels available for capacitive or time measurement input
- On-chip precision current source
- Four-edge input trigger sources
- Polarity control for each edge source
- Control of edge sequence
- Control of response to edges
- High precision time measurement
- Time delay of external or internal signal asynchronous to system clock

The CTMU works in conjunction with the A/D Converter to provide up to 16 channels for time or charge measurement, depending on the specific device and the number of A/D channels available. When configured for time delay, the CTMU is connected to one of the analog comparators. The level-sensitive input edge sources can be selected from four sources: two external inputs, Timer1 or Output Compare Module 1. For device-specific information on available input sources, refer to the appropriate PIC24F data sheet.

A block diagram of the CTMU is shown in Figure 11-1.

Figure 11-1: CTMU Block Diagram



11.2 REGISTERS

There are two control registers for the CTMU: CTMUCON and CTMUICON.

The CTMUCON register (Register 11-1) contains control bits for configuring the CTMU module edge source selection, edge source polarity selection, edge sequencing, A/D trigger, analog circuit capacitor discharge and enables. The CTMUICON register (Register 11-2) has bits for selecting the current source range and current source trim.

Register 11-1: CTMUCON: CTMU Control Register

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

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

- bit 15 **CTMUEN:** CTMU Enable bit
 1 = Module is enabled
 0 = Module is disabled
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **CTMUSIDL:** Stop in Idle Mode bit
 1 = Discontinue module operation when device enters Idle mode
 0 = Continue module operation in Idle mode
- bit 12 **TGEN:** Time Generation Enable bit
 1 = Enables edge delay generation
 0 = Disables edge delay generation
- bit 10 **EDGEN:** Edge Enable bit
 1 = Edges are not blocked
 0 = Edges are blocked
- bit 10 **EDGSEQEN:** Edge Sequence Enable bit
 1 = Edge 1 event must occur before Edge 2 event can occur
 0 = No edge sequence is needed
- bit 9 **IDISSEN:** Analog Current Source Control bit
 1 = Analog current source output is grounded
 0 = Analog current source output is not grounded
- bit 8 **CTTRIG:** Trigger Control bit
 1 = Trigger output is enabled
 0 = Trigger output is disabled
- bit 7 **EDG2POL:** Edge 2 Polarity Select bit
 1 = Edge 2 programmed for a positive level response
 0 = Edge 2 programmed for a negative level response

Note 1: Refer to the particular device data sheet for specific edge source types and assignments.

PIC24F Family Reference Manual

Register 11-1: CTMUCON: CTMU Control Register (Continued)

- bit 6-5 **EDG2SEL1:EDG2SEL0:** Edge 2 Source Select bits⁽¹⁾
 11 = Edge Source 3 selected
 10 = Edge Source 2 selected
 01 = Edge Source 1 selected
 00 = Edge Source 0 selected
- bit 4 **EDG1POL:** Edge 1 Polarity Select bit
 1 = Edge 1 programmed for a positive level response
 0 = Edge 1 programmed for a negative level response
- bit 3-2 **EDG1SEL1:EDG1SEL0:** Edge 1 Source Select bits⁽¹⁾
 11 = Edge Source 3 selected
 10 = Edge Source 2 selected
 01 = Edge Source 1 selected
 00 = Edge Source 0 selected
- bit 1 **EDG2STAT:** Edge 2 Status bit
 1 = Edge 2 event has occurred
 0 = Edge 2 event has not occurred
- bit 0 **EDG1STAT:** Edge 1 Status bit
 1 = Edge 1 event has occurred
 0 = Edge 1 event has not occurred

Note 1: Refer to the particular device data sheet for specific edge source types and assignments.

Register 11-2: CTMUICON: CTMU Current Control Register

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	

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

- bit 15-10 **ITRIM5:ITRIM0:** Current Source Trim bits
 011111 = Maximum positive change from nominal current
 011110

 000001 = Minimum positive change from nominal current
 000000 = Nominal current output specified by IRNG1:IRNG0
 111111 = Minimum negative change from nominal current

 100010
 100001 = Maximum negative change from nominal current
- bit 9-8 **IRNG1:IRNG0:** Current Source Range Select bits
 11 = 100 × base current
 10 = 10 × base current
 01 = Base current level (0.55 μA nominal)
 00 = Current source disabled
- bit 7-0 **Unimplemented:** Read as '0'

11.3 CTMU OPERATION

The CTMU works by using a fixed current source to charge a circuit. The type of circuit depends on the type of measurement being made. In the case of charge measurement, the current is fixed, and the amount of time the current is applied to the circuit is fixed. The amount of voltage read by the A/D is then a measurement of the capacitance of the circuit. In the case of time measurement, the current, as well as the capacitance of the circuit, is fixed. In this case, the voltage read by the A/D is then representative of the amount of time elapsed from the time the current source starts and stops charging the circuit.

If the CTMU is being used as a time delay, both capacitance and current source are fixed, as well as the voltage supplied to the comparator circuit. The delay of a signal is determined by the amount of time it takes the voltage to charge to the comparator threshold voltage.

11.3.1 Theory of Operation

The operation of the CTMU is based on the equation for charge:

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

More simply, the amount of charge measured in coulombs in a circuit is defined as current in amperes (I) multiplied by the amount of time in seconds that the current flows (t). Charge is also defined as the capacitance in farads (C) multiplied by the voltage of the circuit (V). It follows that:

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

The CTMU module provides a constant, known current source. The A/D Converter is used to measure (V) in the equation, leaving two unknowns: capacitance (C) and time (t). The above equation can be used to calculate capacitance or time, by either the relationship:

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

using the known fixed capacitance of the circuit; or by:

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

using a fixed time that the current source is applied to the circuit.

11.3.2 Current Source

At the heart of the CTMU is a precision current source, designed to provide a constant reference for measurements. The level of current is user-selectable across three ranges or a total of two orders of magnitude, with the ability to trim the output in $\pm 2\%$ increments (nominal). The current range is selected by the IRNG1:IRNG0 bits (CTMUICON<9:8>), with a value of '00' representing the lowest range.

Current trim is provided by the ITRIM5:ITRIM0 bits (CTMUICON<15:10>). These six bits allow trimming of the current source in steps of approximately 2% per step. Note that half of the range adjusts the current source positively and the other half reduces the current source. A value of '000000' is the neutral position (no change). A value of '100000' is the maximum negative adjustment (approximately -62%), and '011111' is the maximum positive adjustment (approximately +62%).

11.3.3 Edge Selection and Control

CTMU measurements are controlled by edge events occurring on the module's two input channels. Each channel, referred to as Edge 1 and Edge 2, can be configured to receive input pulses from one of the edge input pins (CTEDG1 and CTEDG2), Timer1 or Output Compare Module 1. The input channels are level-sensitive, responding to the instantaneous level on the channel rather than a transition between levels. The inputs are selected using the EDG1SEL and EDG2SEL bit pairs (CTMUCON<3:2 and 6:5>).

In addition to source, each channel can be configured for event polarity using the EDGE2POL and EDGE1POL bits (CTMUCON<7,4>). The input channels can also be filtered for an edge event sequence (Edge 1 occurring before Edge 2) by setting the EDGSEQEN bit (CTMUCON<10>).

11.3.4 Edge Status

The CTMUCON register also contains two status bits, EDG2STAT and EDG1STAT (CTMUCON<1:0>). Their primary function is to show if an edge response has occurred on the corresponding channel. The CTMU automatically sets a particular bit when an edge response is detected on its channel. The level-sensitive nature of the input channels also means that the status bits become set immediately if the channel's configuration is changed and is the same as the channel's current state.

The module uses the edge status bits to control the current source output to external analog modules (such as the A/D Converter). Current is only supplied to external modules when only one (but not both) of the status bits is set, and shuts current off when both bits are either set or cleared. This allows the CTMU to measure current only during the interval between edges. After both status bits are set, it is necessary to clear them before another measurement is taken. Both bits should be cleared simultaneously, if possible, to avoid re-enabling the CTMU current source.

In addition to being set by the CTMU hardware, the edge status bits can also be set by software. This is also the user's application to manually enable or disable the current source. Setting either one (but not both) of the bits enables the current source. Setting or clearing both bits at once disables the source.

11.3.5 Interrupts

The CTMU sets its interrupt flag (IFS4<13>) whenever the current source is enabled, then disabled. An interrupt is generated only if the corresponding interrupt enable bit (IEC4<13>) is also set. If edge sequencing is not enabled (i.e., Edge 1 must occur before Edge 2), it is necessary to monitor the edge status bits and determine which edge occurred last and caused the interrupt.

11.4 CTMU MODULE INITIALIZATION

The following sequence is a general guideline used to initialize the CTMU module:

1. Select the current source range using the IRNG bits (CTMUICON<9:8>).
2. Adjust the current source trim using the ITRIM bits (CTMUICON<15:10>).
3. Configure the edge input sources for Edge 1 and Edge 2 by setting the EDG1SEL and EDG2SEL bits (CTMUICON<3:2 and 6:5>).
4. Configure the input polarities for the edge inputs using the EDG1POL and EDG2POL bits (CTMUICON<4,7>). The default configuration is for negative edge polarity (high-to-low transitions).
5. Enable edge sequencing using the EDGSEQEN bit (CTMUICON<10>). By default, edge sequencing is disabled.
6. Select the operating mode (Measurement or Time Delay) with the TGEN bit. The default mode is Time/Capacitance Measurement.
7. Configure the module to automatically trigger an A/D conversion when the second edge event has occurred using the CTTRIG bit (CTMUICON<8>). The conversion trigger is disabled by default.
8. Discharge the connected circuit by setting the IDISSEN bit (CTMUICON<9>); after waiting a sufficient time for the circuit to discharge, clear IDISSEN.
9. Disable the module by clearing the CTMUEN bit (CTMUICON<15>).
10. Clear the Edge Status bits, EDG2STAT and EDG1STAT (CTMUICON<1:0>).
11. Enable both edge inputs by setting the EDGEN bit (CTMUICON<11>).
12. Enable the module by setting the CTMUEN bit.

Depending on the type of measurement or pulse generation being performed, one or more additional modules may also need to be initialized and configured with the CTMU module:

- Edge Source Generation: In addition to the external edge input pins, both Timer1 and the Output Compare/PWM1 module can be used as edge sources for the CTMU.
- Capacitance or Time Measurement: The CTMU module uses the A/D Converter to measure the voltage across a capacitor that is connected to one of the analog input channels.
- Pulse Generation: When generating system clock independent output pulses, the CTMU module uses Comparator 2 and the associated comparator voltage reference.

For specific information on initializing these modules, refer to the applicable PIC24F Family Reference chapter for the appropriate module.

11.5 CALIBRATING THE CTMU MODULE

The CTMU requires calibration for precise measurements of capacitance and time, as well as for accurate time delay. If the application only requires measurement of a relative change in capacitance or time, calibration is usually not necessary. An example of this type of application would include a capacitive touch switch, in which the touch circuit has a baseline capacitance, and the added capacitance of the human body changes the overall capacitance of a circuit.

If actual capacitance or time measurement is required, two hardware calibrations must take place: the current source needs calibration to set it to a precise current, and the circuit being measured needs calibration to measure and/or nullify all other capacitance other than that to be measured.

11.5.1 Current Source Calibration

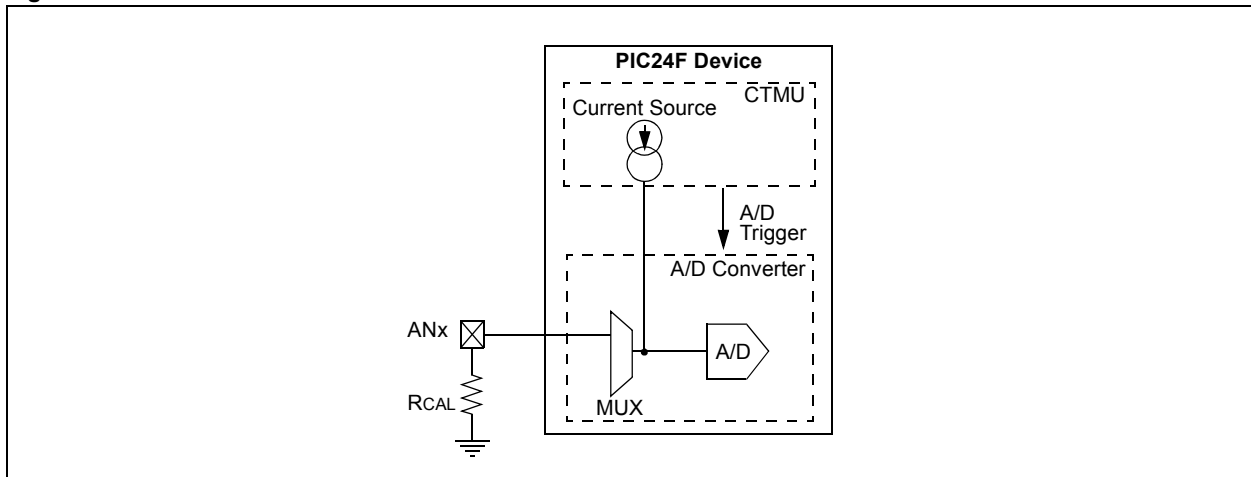
The current source onboard the CTMU module has a range of $\pm 60\%$ nominal for each of three current ranges. Therefore, for precise measurements, it is possible to measure and adjust this current source by placing a high precision resistor, R_{CAL} , onto an unused analog channel. An example circuit is shown in Figure 11-2. The current source measurement is performed using the following steps:

1. Initialize the A/D Converter.
2. Initialize the CTMU.
3. Enable the current source by setting EDG1STAT (CTMUCON<0>).
4. Issue settling time delay.
5. Perform A/D conversion.
6. Calculate the current source current using $I = V/R_{CAL}$, where R_{CAL} is a high precision resistance and V is measured by performing an A/D conversion.

The CTMU current source may be trimmed with the trim bits in CTMUICON using an iterative process to get an exact desired current. Alternatively, the nominal value without adjustment may be used; it may be stored by the software for use in all subsequent capacitive or time measurements.

To calculate the value for R_{CAL} , the nominal current must be chosen, and then the resistance can be calculated. For example, if the A/D Converter reference voltage is 3.3V, use 70% of full scale, or 2.31V as the desired approximate voltage to be read by the A/D Converter. If the range of the CTMU current source is selected to be 0.55 μA , the resistor value needed is calculated as $R_{CAL} = 2.31\text{V}/0.55 \mu\text{A}$, for a value of 4.2 M Ω . Similarly, if the current source is chosen to be 5.5 μA , R_{CAL} would be 420,000 Ω , and 42,000 Ω if the current source is set to 55 μA .

Figure 11-2: CTMU Current Source Calibration Circuit



A value of 70% of full-scale voltage is chosen to make sure that the A/D Converter was in a range that is well above the noise floor. Keep in mind that if an exact current is chosen that is to incorporate the trimming bits from CTMUICON, the resistor value of RCAL may need to be adjusted accordingly. RCAL may be also adjusted to allow for available resistor values. RCAL should be of the highest precision available, keeping in mind the amount of precision needed for the circuit that the CTMU will be used to measure. A recommended minimum would be 0.1% tolerance.

The following examples show one typical method for performing a CTMU current calibration. Example 11-1 shows how to initialize the A/D Converter and the CTMU; this routine is typical for applications using both modules. Example 11-2 shows one method for the actual calibration routine. Note that this method manually triggers the A/D Converter; this is done to demonstrate the entire stepwise process. It is also possible to automatically trigger the conversion by setting the CTMU's CTRIG bit (CTMUCON<8>).

Example 11-1: Setup for CTMU Calibration Routines

```
#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, Edg1 polarity = positive level, Edg1 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
    Samp bit to start
    //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
}
}
```


Example 11-2: Current Calibration Routine

```

#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 Capacitance Calibration

There is a small amount of capacitance from the internal A/D Converter sample capacitor as well as stray capacitance from the circuit board traces and pads that affect the precision of capacitance measurements. A measurement of the stray capacitance can be taken by making sure the desired capacitance to be measured has been removed. The measurement is then performed using the following steps:

1. Initialize the A/D Converter and the CTMU.
2. Set EDG1STAT (= 1).
3. Wait for a fixed delay of time t .
4. Clear EDG1STAT.
5. Perform an A/D conversion.
6. Calculate the stray and A/D sample capacitances:

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

where I is known from the current source measurement step, t is a fixed delay and V is measured by performing an A/D conversion.

This measured value is then stored and used for calculations of time measurement, or subtracted for capacitance measurement. For calibration, it is expected that the capacitance of $C_{\text{STRAY}} + C_{\text{AD}}$ is approximately known. C_{AD} is approximately 4 pF.

An iterative process may need to be used to adjust the time, t , that the circuit is charged to obtain a reasonable voltage reading from the A/D Converter. The value of t may be determined by setting C_{OFFSET} to a theoretical value, then solving for t . For example, if C_{STRAY} is theoretically calculated to be 11 pF, and V is expected to be 70% of V_{DD} , or 2.31V, then t would be:

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

or 63 μs .

A typical routine for CTMU capacitance calibration is shown in Example 11-3.

Example 11-3: Capacitance Calibration Routine

```

#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 MEASURING CAPACITANCE WITH THE CTMU

There are two separate methods of measuring capacitance with the CTMU. The first is the absolute method, in which the actual capacitance value is desired. The second is the relative method, in which the actual capacitance is not needed, rather an indication of a change in capacitance is required.

11.6.1 Absolute Capacitance Measurement

For absolute capacitance measurements, both the current and capacitance calibration steps found in **Section 11.5 “Calibrating the CTMU Module”** should be followed. Capacitance measurements are then performed using the following steps:

1. Initialize the A/D Converter.
2. Initialize the CTMU.
3. Set EDG1STAT.
4. Wait for a fixed delay, T .
5. Clear EDG1STAT.
6. Perform an A/D conversion.
7. Calculate the total capacitance, $C_{TOTAL} = (I * T)/V$, where I is known from the current source measurement step (**Section 11.5.1 “Current Source Calibration”**), T is a fixed delay and V is measured by performing an A/D conversion.
8. Subtract the stray and A/D capacitance (C_{OFFSET} from **Section 11.5.2 “Capacitance Calibration”**) from C_{TOTAL} to determine the measured capacitance.

11.6.2 Relative Charge Measurement

An application may not require precise capacitance measurements. For example, when detecting a valid press of a capacitance-based switch, detecting a relative change of capacitance is of interest. In this type of application, when the switch is open (or not touched), the total capacitance is the capacitance of the combination of the board traces, the A/D Converter, etc. A larger voltage will be measured by the A/D Converter. When the switch is closed (or is touched), the total capacitance is larger due to the addition of the capacitance of the human body to the above listed capacitances, and a smaller voltage will be measured by the A/D Converter.

Detecting capacitance changes is easily accomplished with the CTMU using these steps:

1. Initialize the A/D Converter and the CTMU.
2. Set EDG1STAT.
3. Wait for a fixed delay.
4. Clear EDG1STAT.
5. Perform an A/D conversion.

The voltage measured by performing the A/D conversion is an indication of the relative capacitance. Note that in this case, no calibration of the current source or circuit capacitance measurement is needed. A sample software routine for a capacitive touch switch is shown in Example 11-4.

Example 11-4: Routine for Capacitive Touch Switch

```

#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 MEASURING TIME WITH THE CTMU MODULE

Time can be precisely measured after the ratio (C/I) is measured from the current and capacitance calibration step by following these steps:

1. Initialize the A/D Converter and the CTMU.
2. Set EDG1STAT.
3. Set EDG2STAT.
4. Perform an A/D conversion.
5. Calculate the time between edges as $T = (C/I) * V$, where I is calculated in the current calibration step (**Section 11.5.1 “Current Source Calibration”**), C is calculated in the capacitance calibration step (**Section 11.5.2 “Capacitance Calibration”**) and V is measured by performing the A/D conversion.

It is assumed that the time measured is small enough that the capacitance C_{OFFSET} provides a valid voltage to the A/D Converter. For the smallest time measurement, always set the A/D Channel Select register (AD1CHS) to an unused A/D channel; the corresponding pin for which is not connected to any circuit board trace. This minimizes added stray capacitance, keeping the total circuit capacitance close to that of the A/D Converter itself (4-5 pF). To measure longer time intervals, an external capacitor may be connected to an A/D channel, and this channel selected when making a time measurement.

11.8 CREATING A DELAY WITH THE CTMU MODULE

A unique feature on board the CTMU module is its ability to generate system clock independent output pulses based on an external capacitor value. This is accomplished using the internal comparator voltage reference module, Comparator 2 input pin and an external capacitor. The pulse is output onto the CTPLS pin. To enable this mode, set the TGEN bit.

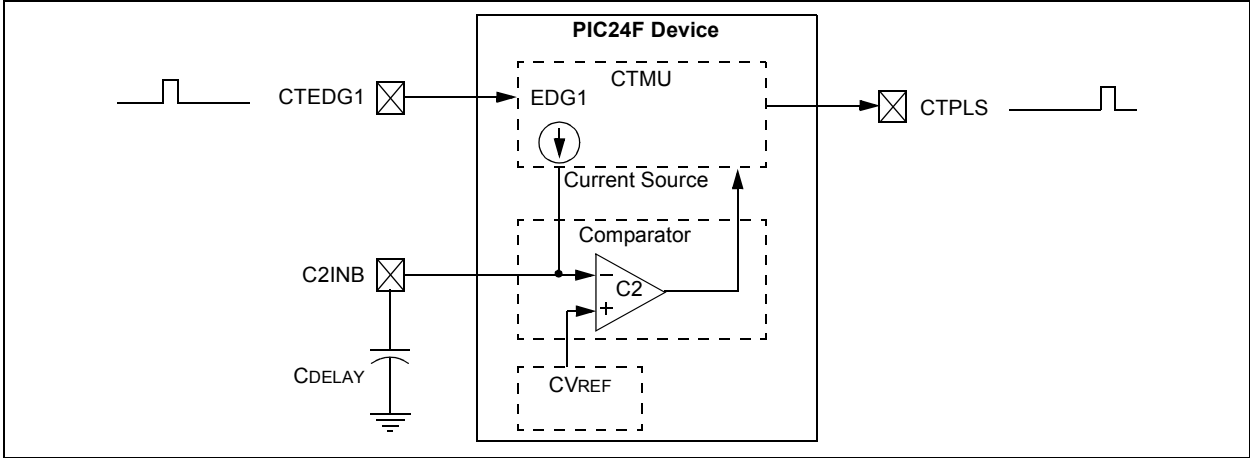
An example circuit is shown in Figure 11-3. C_{PULSE} is chosen by the user to determine the output pulse width on CTPLS. The pulse width is calculated by $T = (C_{PULSE}/I) * V$, where I is known from the current source measurement step (**Section 11.5.1 “Current Source Calibration”**) and V is the internal reference voltage (CV_{REF}).

An example use of this feature is for interfacing with variable capacitive-based sensors, such as a humidity sensor. As the humidity varies, the pulse-width output on CTPLS will vary. The CTPLS output pin can be connected to an input capture pin and the varying pulse width is measured to determine the humidity in the application.

Follow these steps to use this feature:

1. Initialize Comparator 2.
2. Initialize the comparator voltage reference.
3. Initialize the CTMU and enable time delay generation by setting the TGEN bit.
4. Set EDG1STAT.
5. When C_{PULSE} charges to the value of the voltage reference trip point, an output pulse is generated on CTPLS.

Figure 11-3: Typical Connections and Internal Configuration for Pulse Delay Generation



11.9 OPERATION DURING SLEEP/IDLE MODES

11.9.1 Sleep Mode and Deep Sleep Modes

When the device enters any Sleep mode, the CTMU module current source is always disabled. If the CTMU is performing an operation that depends on the current source when Sleep mode is invoked, the operation may not terminate correctly. Capacitance and time measurements may return erroneous values.

11.9.2 Idle Mode

The behavior of the CTMU in Idle mode is determined by the CTMUSIDL bit (CTMUCON<13>). If CTMUSIDL is cleared, the module will continue to operate in Idle mode. If CTMUSIDL is set, the module's current source is disabled when the device enters Idle mode. If the module is performing an operation when Idle mode is invoked, in this case, the results will be similar to those with Sleep mode.

11.10 EFFECTS OF A RESET ON CTMU

Upon Reset, all registers of the CTMU are cleared. This leaves the CTMU module disabled, its current source is turned off and all configuration options return to their default settings. The module needs to be re-initialized following any Reset.

If the CTMU is in the process of taking a measurement at the time of Reset, the measurement will be lost. A partial charge may exist on the circuit that was being measured, and should be properly discharged before the CTMU makes subsequent attempts to make a measurement. The circuit is discharged by setting and then clearing the IDISSEN bit (CTMUCON<9>) while the A/D Converter is connected to the appropriate channel.

11.11 REGISTER MAPS

A summary of the registers associated with the PIC24F CTMU is provided in Table 11-1.

Table 11-1: CTMU Register Map

File Name	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	All Resets
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

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

PIC24F Family Reference Manual

11.12 ELECTRICAL SPECIFICATIONS

Table 11-2: CTMU Current Source Specifications

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial				
Param No.	Sym	Characteristic	Min	Typ ⁽¹⁾	Max	Units	Conditions
	IOUT1	CTMU Current Source, Base Range	—	550	—	nA	CTMUICON<1:0> = 01
	IOUT2	CTMU Current Source, 10x Range	—	5.5	—	μA	CTMUICON<1:0> = 10
	IOUT3	CTMU Current Source, 100x Range	—	55	—	μA	CTMUICON<1:0> = 11

Note 1: Nominal value at center point of current trim range (CTMUICON<7:2> = 000000)

11.13 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC24F device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the CTMU module are:

Title**Application Note #**

No related application notes at this time.

Note: Please visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the PIC24F family of devices.

11.14 REVISION HISTORY

Revision A (March 2008)

This is the initial released revision of this document.