Section 53. Charge Time Measurement Unit (CTMU) with Threshold Detect

HIGHLIGHTS

This section of the manual contains the following major topics:

53.1 INTRODUCTION

The Charge Time Measurement Unit (CTMU) is a flexible analog module that has a configurable current source with a digital circuit built around it. The CTMU can be used for differential time measurement between pulse sources and can be used for generating an asynchronous pulse. By working with other on-chip analog modules, the CTMU can be used for high resolution time measurement, measure capacitance, resistance, inductance, temperature, used for humidity sensing, measure relative changes in capacitance or generate output pulses with a specific time delay. The CTMU is ideal for interfacing with capacitive-based sensors.

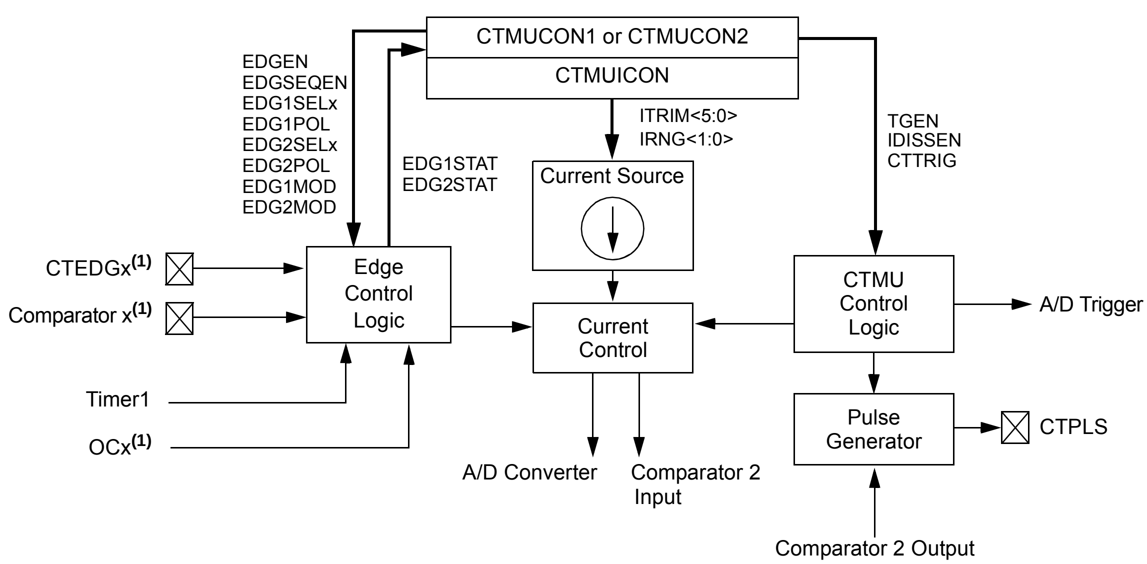
The module includes the following key features:

- On-chip precision current source
- Sixteen-edge input trigger sources
- Selection of edge or level-sensitive inputs
- 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
- Integrated temperature sensing diode
- Control of current source during auto-sampling
- Four current source ranges
- Time measurement resolution of one nanosecond or less
- CTMU operation in Sleep mode

The CTMU works in conjunction with the A/D Converter 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 input edge sources can be selected from sixteen sources for each edge. 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 53-1.

Figure 53-1: CTMU Block Diagram



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

53.2 REGISTERS

Depending on the device variant, there are up to three control registers available for the CTMU: CMTUCON1, CTMUCON2 and CTMUICON.

The CTMUCON1 and CTMUCON2 registers (Register 53-1 and Register 53-2) contain 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 53-3) has bits for selecting the current source range and current source trim.

Register 53-1: CTMUCON1: CTMU Control Register 1

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

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 **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 11 **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 7-0 **Unimplemented:** Read as ‘0’

Register 53-2: CTMUCON2: CTMU Control Register 2

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	EDG1SEL3[(1)]	EDG1SEL2[(1)]	EDG1SEL1[(1)]	EDG1SEL0[(1)]	EDG2STAT	EDG1STAT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
EDG2MOD	EDG2POL	EDG2SEL3[(1)]	EDG2SEL2[(1)]	EDG2SEL1[(1)]	EDG2SEL0[(1)]	—	—
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 **EDG1MOD:** Input mode selection bit
1 = Input is edge-sensitive
0 = Input is level-sensitive

bit 14 **EDG1POL:** Edge 1 Polarity Select bit
1 = Edge 1 is programmed for a positive level response
0 = Edge 1 is programmed for a negative level response

bit 13-10 **EDG1SEL<3:0>:** Edge 1 Source Select bits[(1)]
1111 = Edge 1 Source 15 selected
1110 = Edge 1 Source 14 selected
1101 = Edge 1 Source 13 selected
1100 = Edge 1 Source 12 selected
1011 = Edge 1 Source 11 selected
1010 = Edge 1 Source 10 selected
1001 = Edge 1 Source 9 selected
1000 = Edge 1 Source 8 selected
0111 = Edge 1 Source 7 selected
0110 = Edge 1 Source 6 selected
0101 = Edge 1 Source 5 selected
0100 = Edge 1 Source 4 selected
0011 = Edge 1 Source 3 selected
0010 = Edge 1 Source 2 selected
0001 = Edge 1 Source 1 selected
0000 = Edge 1 Source 0 selected

bit 9 **EDG2STAT:** Edge 2 Status bit
1 = Edge 2 event has occurred
0 = Edge 2 event has not occurred

bit 8 **EDG1STAT:** Edge 1 Status bit
1 = Edge 1 event has occurred
0 = Edge 1 event has not occurred

bit 7 **EDG2MOD:** Input Mode Selection bit
1 = Input is edge-sensitive
0 = Input is level-sensitive

bit 6 **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.

Register 53-2: CTMUCON2: CTMU Control Register 2 (Continued)

bit 5-2	**EDG2SEL<3:0>:** Edge 2 Source Select bits[1] 1111 = Edge 2 Source 15 selected 1110 = Edge 2 Source 14 selected 1101 = Edge 2 Source 13 selected 1100 = Edge 2 Source 12 selected 1011 = Edge 2 Source 11 selected 1010 = Edge 2 Source 10 selected 1001 = Edge 2 Source 9 selected 1000 = Edge 2 Source 8 selected 0111 = Edge 2 Source 7 selected 0110 = Edge 2 Source 6 selected 0101 = Edge 2 Source 5 selected 0100 = Edge 2 Source 4 selected 0011 = Edge 2 Source 3 selected 0010 = Edge 2 Source 2 selected 0001 = Edge 2 Source 1 selected 0000 = Edge 2 Source 0 selected
bit 1-0	**Unimplemented:** Read as ‘0’

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

Register 53-3: 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 **ITRIM<5:0>:** 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 IRNG<1:0>
111111 = Minimum negative change from nominal current
•
•
•
100010
100001 = Maximum negative change from nominal current

bit 9-8 **IRNG<1:0>:** Current Source Range Select bits

11 = 100 × base current
10 = 10 × base current
01 = Base current level (0.55 μA nominal)
00 = 1000 × base current

bit 7-0 **Unimplemented:** Read as ‘0’

53.3 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.

53.3.1 Theory of Operation

The operation of the CTMU is based on the equation for charge, as shown in Equation 1.

Equation 53-1:

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

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), as shown in Equation 53-2.

Equation 53-2:

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

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). Equation 53-2 can be used to calculate capacitance or time, by either the relationship shown in Equation 53-3 and using the known fixed capacitance of the circuit, or by Equation 53-4 using a fixed time that the current source is applied to the circuit.

Equation 53-3:

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

Equation 53-4:

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

53.3.2 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 four ranges, or a total of three orders of magnitude, with the ability to trim the output in ±2% increments (nominal). The current range is selected by the IRNG1:0> bits (CTMUICON<9:8>) with a value of ‘01’ representing the lowest range.

Current trim is provided by the ITRIM<5:0> 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%).

53.3.3 Edge/Level Selection and Control

CTMU measurements are controlled by the edge or level 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 sixteen edge input pins. The inputs are selected using the EDG1SEL and EDG2SEL bit pairs (CTMUCON2<5:2> and <13:10>). Further, the mode of the input sources to the Edge 1 and Edge 2 can either be level-sensitive or edge-sensitive, which is selected using the EDG1MOD bit (CTMUCON2<15>).

In addition to source, each channel can be configured for event polarity using the EDGE1POL and EDGE2POL bits (CTMUCON2<14> and CTMUCON2<6>). The input channels can also be filtered for an edge event sequence (Edge 1 occurring before Edge 2) by setting the EDGSEQEN bit (CTMUCON1<10>).

53.3.4 Edge Status

The CTMUCON2 register also contains two status bits: EDG1STAT and EDG2STAT (CTMUCON2<9:8>). 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, or edge-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 EDG1STAT is not equal to EDG2STAT, and shuts current off when EDG1STAT is equal to EDG2STAT. 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 allows 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.

53.3.5 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.

53.4 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 (CTMUCON2<13:10> and CTMUCON2<5:2>).
4. Configure the input polarities for the edge inputs using the EDG1POL and EDG2POL bits (CTMUCON2<14:6>). The default configuration is for negative edge polarity (high-to-low transitions).
5. Configure the Input mode for the Edge 1 and Edge 2 to select the level or edge using the EDG1MOD bit (CTMUCON2<15>).
6. Enable edge sequencing using the EDGSEQEN bit (CTMUCON1<10>). By default, edge sequencing is disabled.
7. Select the operating mode (Measurement or Time Delay) with the TGEN bit (CTMUCON1<12>). By default, the Time Delay mode is disabled.
8. Configure the module to automatically trigger an A/D conversion when the second edge event has occurred using the CTTRIG bit (CTMUCON1<8>). The conversion trigger is disabled by default.
9. Discharge the connected circuit by setting the IDISSEN bit (CTMUCON1<9>). After waiting a sufficient time for the circuit to discharge, clear IDISSEN.
10. Disable the module by clearing the CTMUEN bit (CTMUCON1<15>).
11. Clear the Edge Status bits, EDG2STAT and EDG1STAT (CTMUCON2<9:8>).
12. Enable both edge inputs by setting the EDGEN bit (CTMUCON<11> or CTMUCON1<11>).
13. Enable the module by setting the CTMUEN bit (CTMUCON1<15>).

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 Manual”* section for the appropriate module.

53.5 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.

53.5.1 Current Source Calibration

The current source on board the CTMU module has a range of ±62% nominal for each of four current ranges. Therefore, for precise measurements, it is possible to measure and adjust this current source by placing a high precision resistor, RCAL, onto the C2INB pin of the comparator. The Time Generation mode bit, TGEN bit (CTMUCON1<12>), should be enabled and then the voltage across the RCAL is measured through the ADC. An example circuit is shown in Figure 53-2. The current source measurement is performed using the following steps:

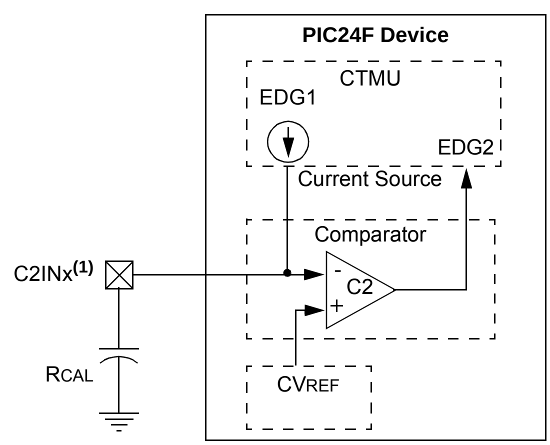
1. Initialize the A/D Converter.
2. Initialize Comparator 2.
3. Initialize the comparator voltage reference.
4. Initialize the CTMU by configuring the module for Pulse Generation (TGEN = 1) mode.
5. Enable the current source by setting the EDG1STAT bit (CTMUCON2<0>.
6. Wait for the fixed delay.
7. Disable the current source by clearing the EDG2STAT bit (CTMUCON2<0>).
8. Perform A/D conversion.
9. 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.

Figure 53-2 shows the external connections for current source calibration, as well as the relationship of the different analog modules required.

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 MΩ. Similarly, if the current source is chosen to be 5.5 µA, R_{CAL} would be 420,000Ω, 42,000Ω if the current source is set to 55 µA and 4.2 kΩ if the current source is set to 550 µA.

Figure 53-2: CTMU Current Source Calibration Circuit



Note 1: Please refer to the specific device data sheet for information related to the comparator input used in this mode.

A value of 70% of full-scale voltage is chosen to make sure that the A/D Converter is 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 R_{CAL} may need to be adjusted accordingly. R_{CAL} may also be adjusted to allow for available resistor values. R_{CAL} 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 53-1 shows how to initialize the A/D Converter and the CTMU; this routine is typical for applications using both modules. Example 53-2 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 CTTRIG bit (CTMUCON<8> or CTMUCON1<8>).

Example 53-1: Setup for CTMU Calibration Routines

```
#include "p24Fxxxx.h"

/***************************************************************************/
/ Set up CTMU *************************************************************/
/***************************************************************************/

void setup(void)
{   //CTMUCON1 and CTMUCON2 - CTMU Control register
       CTMUCON1 = 0x0000;     //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
       CTMUCON2 = 0xC0C0;
       CTMUICON = 0x0100;     // 0.55uA, Nominal - No Adjustment
/***************************************************************************/
/ Set up AD converter ******************************************************/
/***************************************************************************/
       TRISB = 0x0001;        // Set channel 2 as an input
       ANSELBbits.ANSELB0 = 1; // Make AN2 as analog
       AD1CHS = 0x002;        // Select the analog channel(2)
       AD1CSSL = 0x0000;      // Skip the analog channels for input scan
       AD1CSSH = 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,
                              // interrupts at end of conversion for each sample
       AD1CON3 = 0x0000;      // A/D uses system clock, conversion clock = 1xTcy
       AD1CON5 = 0x0000;      // Auto-Scan disabled
}
```

53

CTMU with Threshold Detect

Example 53-2: 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();

    CTMUCON1bits.CTMUEN = 1;            //Enable the CTMU

    for(j=0;j<10;j++)
    {
        AD1CON1bits.SAMP = 1;           //Manual sampling start
        CTMUCON1bits.IDISSEN = 1;       //drain charge on the circuit
        DELAY;                          //wait 125us
        CTMUCON1bits.IDISSEN = 0;       //end drain of circuit

        CTMUCON2bits.EDG1STAT = 1;      //Begin charging the circuit
                                        //using CTMU current source
        DELAY;                          //wait for 125 us

        CTMUCON2bits.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
}
```

53.5.2 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.
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 using Equation 53-5.

Equation 53-5:

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

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 $C_{STRAY} + C_{AD}$ is approximately known. Please refer to the respective device data sheet for the value of C_{AD}.

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, t would be equal to Equation 53-6 or 63 µs.

Equation 53-6: :

$$(4\ pF + 11\ pF) \bullet \frac{2.31V}{0.55\mu A}$$

A typical routine for CTMU capacitance calibration is shown in Example 53-3.

53

CTMU with Threshold Detect

Example 53-3: 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+
#define RCAL .027         //R value is 4200000 (4.2M)
                          //scaled so that result is in
                          //1/100th of uA


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();

    CTMUCON1bits.CTMUEN = 1;//Enable the CTMU

    for(j=0;j<10;j++)
    {
        AD1CON1bits.SAMP = 1;          //Manual sampling start
        CTMUCON1bits.IDISSEN= 1;       //drain any charge on the circuit
        DELAY;                         //wait 62.5 us
        CTMUCON1bits.IDISSEN = 0;      //end drain of circuit
        CTMUCON2bits.EDG1STAT = 1;     //Begin charging the circuit
                                       //using the CTMU current source
        DELAY;                         //wait for 62.5 us for circuit
                                       //to charge
        CTMUCON2bits.EDG1STAT = 0;     //Stop charging circuit and begin
                                       //A/D conversion
        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);
    CTMUISrc = Vcal/RCAL;          //CTMUISrc is in 1/100ths of uA
    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
}
```


53.6 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.

53.6.1 Absolute Capacitance Measurement

For absolute capacitance measurements, both the current and capacitance calibration steps found in **Section 53.5 "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 53.5.1 "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 53.5.2 "Capacitance Calibration"**) from C_{TOTAL} to determine the measured capacitance.

53.6.2 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 53-4.

Example 53-4: 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     UNPRESSED0

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();

    CTMUCON1bits.CTMUEN = 1;    //Enable the CTMU

    AD1CON1bits.SAMP = 1;       //Manual sampling start
    CTMUCON1bits.IDISSEN = 1;   //drain charge on the circuit
    DELAY;                      //wait 125us
    CTMUCON1bits.IDISSEN = 0;   //end drain of circuit

    CTMUCON2bits.EDG1STAT = 1;  //Begin charging the circuit
                                //using CTMU current source
    DELAY;                      //wait for 125us
    CTMUCON2bits.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;
    }

}
```


53.6.3 Relative Charge Measurement with the Auto-Threshold Scanning Method

The Auto-Threshold Scanning method is a significant extension of the Auto-Threshold Scan feature offered in the 10-bit A/D modules. In addition to being able to repeatedly sample a predefined sequence of analog channels, Threshold Detect allows the user to define match conditions based on the conversion results and generate an interrupt based on these conditions.

The procedure for calculating the relative charge with the Auto-Threshold Scanning method is similar to the legacy Auto-Scanning method, except that the threshold levels can be set for the converted values in the A/D result buffer. The Auto-Threshold Scan allows the user to define the match conditions, based on the conversion values, and generate its own interrupt based on these conditions. The operation of the Auto-Threshold Scan is controlled by the AD1CON5 register. The CSCNA bit (AD1CON2<10>), along with the ASENA bit (AD1CON5<15>), controls the overall operation of Auto-Threshold Scan by enabling the bit.

The comparison thresholds for Auto-Threshold Scan are set by writing the desired values to an appropriate location in the A/D result buffer. The location of the threshold values is determined by the comparison type which can be selected by the CM<1:0> (AD1CON5<1:0>) bits. The AD1CTMENH/L registers select the channel(s) to be used by the CTMU during conversions. The Auto-Threshold Scan can also be implemented when the device is placed in Sleep mode. The LPENA bit (AD1CON5<14>) allows the Auto-Threshold Scan to function with low-power features.

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

1. Initialize the A/D Converter (including Auto-Threshold Scanning bits) and the CTMU.
2. Set EDG1STAT.
3. Wait for a fixed delay.
4. Clear EDG1STAT.
5. Perform an A/D conversion.

Note: Refer to the 12-Bit A/D Converter with Auto-Compare FRM for further details.

Example 53-5: Setup for Auto-Threshold Sampling

```
#include "p24Fxxxx.h"
/***********************************************************************/
/ Set up CTMU *********************************************************/
/***********************************************************************/
void setup(void)
{ //CTMUCON - CTMU Control register

// 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
CTMUCON1 = 0x0000;     //make sure CTMU is disabled , CTRIG is enabled
CTMUCON2 = 0xC0C0;     // EDG1 Polarity -- positive,EDG2 Polarity -- positive;
CTMUICON = 0x0000;     // 0.55uA, Nominal - No Adjustment
/***********************************************************************/
/ Set up AD converter for Auto Threshold
*****************************************************/
/***********************************************************************/
TRISB = 0x0001;         // Set channel 2 as an input
ANSBbits.ANSB0 = 1;     // Make AN2 as analog
AD1CHS = 0x002;         // Select the analog channel(2)
AD1CSSL = 0x0004;       //Select AN2 for the input scan

AD1CHITL = 0x0000;

AD1CON1 = 0x0440;       // A/D is off, 12 bit ADC,SSRC = 4(CTMU trigger is selected)
AD1CON2 = 0x0800;       //  BUFREGN =1
AD1CON5 = 0x8204;       // A/D uses system clock, conversion clock = 1xTcy
AD1CSSL = 0x0002;       //Scan AN2
AD1CON2bits.CSCNA = 1;  //CSNA = 1 (Scan input channels)
AD1CTMUENL = 0x0002;    // An2 is selected for channel conversion
AD1CON1bits.ASAM = 1;   // A/D Auto sample bit is enabled

ADC1BUF0 = 0x04D0;      //A/D Threshold value

AD1CON1bits.ADON = 1;   //Turn on the ADC

}
```


Example 53-6: Routine for Capacitive Touch Switch with Auto-Threshold Scan

```
#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 UNPRESSED0
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();
CTMUCON1bits.CTMUEN = 1;   //Enable the CTMU
AD1CON1bits.SAMP = 1;      //Manual sampling start
CTMUCON1bits.IDISSEN = 1;  //drain charge on the circuit
DELAY;                     //wait 125us
CTMUCON1bits.IDISSEN = 0;  //end drain of circuit
CTMUCON2bits.EDG1STAT = 1; //Begin charging the circuit
                           //using CTMU current source
DELAY;                     //wait for 125us
CTMUCON2bits.EDG1STAT = 0; //Stop charging circuit
IFS0bits.AD1IF = 0;        //make sure A/D Int not set
AD1CON1bits.SAMP = 0;      //and begin A/D conv.
if(AD1CHITLbits.CHH2)
{
    while(!IFS0bits.AD1IF);//Wait for A/D convert complete
    AD1CON1bits.DONE = 0;
    Vread = ADC1BUF0;      //Get the value from the A/D
}
else
{
                           //
}

if(Vread < OPENSW - TRIP)
{
    switchState = PRESSED;
}
else if(Vread > OPENSW - TRIP + HYST)
{
    switchState = UNPRESSED;
}
}
```


53.7 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 53.5.1 “Current Source Calibration”**), C is calculated in the capacitance calibration step (**Section 53.5.2 “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 that 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 is selected when making a time measurement.

53.8 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.

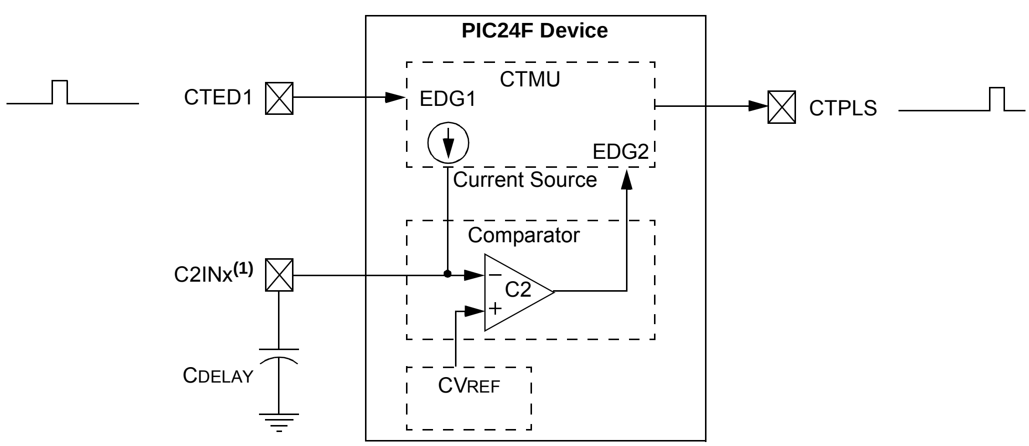
An example circuit is shown in Figure 53-3. C_{PULSE} is chosen by the user to determine the output pulse width on CTPLS. The pulse width is calculated by $T = (C_{PULSE}/I) * V$, where I is known from the current source measurement step (**Section 53.5.1 "Current Source Calibration"**) and V is the internal reference voltage (CVREF).

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 53-3: Typical Connections and Internal Configuration for Pulse Delay Generation



Note 1: Please refer to the specific device data sheet for information related to the comparator input used in this mode.

53.9 MEASURING ON-CHIP TEMPERATURE WITH THE CTMU

The CTMU module can be used to measure the internal temperature of the device through an internal diode that is available for such purposes. When EDGE 1 is not equal to EDGE 2 and TGEN = 0, the current is steered into the temperature sensing diode. The voltage across the diode is available as an input to the ADC module through AN16[(1)], which is selected using the ADCHS0.

Figure 53-4 shows how this module can be used for temperature measurement. As the temperature rises, the voltage across the diode will drop by about 300 mV over a 150ºC range. Selecting a higher current drive strength will raise the voltage value by a few 100 mV.

53.9.1 BASIC PRINCIPLE

We can show that the forward voltage (*Vf*) of a P-N junction, such as a diode, is an extension of the equation for the junction's thermal voltage:

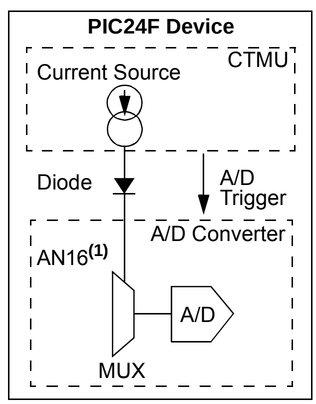
Vf = (kT/q) ln (1-If/IS)

where *k* is the Boltzmann constant (1.38 x 10-23 J K-1), *T* is the absolute junction temperature in kelvin, *q* is the electron charge (1.6 x 10-19 C), I*f* is the forward current applied to the diode and *IS* is the diode's characteristic saturation current.

Since k and q are physical constants, and *IS* is a constant for the device, this only leaves *T* and *If* as independent variables. If *If* is held constant, it follows from the equation that *Vf* will vary as a function of *T*. As the natural log term of the equation will always be negative, the temperature will be negatively proportional to V*f*.

In other words, as temperature increases, V*f* decreases.

Figure 53-4: CTMU Temperature Measurement Circuit



Note 1:	Refer to the specific device data sheet for the actual channel number associated with the temperature sensing diode.

53 CTMU with Threshold Detect

53.10 OPERATION DURING SLEEP/IDLE MODES

53.10.1 Sleep Mode and Deep Sleep Modes with the CTMUREQ Bit Disabled

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.

53.10.2 Sleep Mode and Deep Sleep Modes with the CTMUREQ Bit Enabled

When the device enters Sleep mode, the CTMU module current source can be enabled by setting the CTMUREQ bit (ADCCON5<13>). The CTMU can be triggered from the ADC, which can perform the conversion in Sleep mode when the A/D clock source is set to the internal A/D RC oscillator (ADRC = 1). This will allow the CPU to remain in the inactive state for a longer period of time, and at the same time, it can perform the conversion of the selected CTMU channel.

When the A/D interrupt (AD1IE) is enabled, the device will wake up from Sleep as soon as the A/D interrupt occurs. This will help to reduce the power consumed by the CPU, which is the prerequisite for many low-power applications.

53.10.3 Idle Mode

The behavior of the CTMU in Idle mode is determined by the CTMUSIDL bit (CTMUCON<13> or CTMUCON1<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.

53.11 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> or CTMUCON1<9>) while the A/D Converter is connected to the appropriate channel.

53.12 LOW-POWER APPLICATIONS

The CTMU module, along with the ADC with Auto-Threshold Detection technique, can be used for implementing low-power based applications. The low-power functionality can be achieved by alternatively shifting between the Normal mode and the Sleep mode, which is based on the need of the application. The CTMU module will be active in Sleep mode by enabling the CTMUREQ bit (ADCCON5<13>). This logic will significantly reduce the power consumed by the system, which is the common requirement in many of the applications.

53.13 REGISTER MAPS

A summary of the registers associated with the PIC24F CTMU is provided in Table 53-1.

Table 53-1: 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
CTMUCON1	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	—	—	—	—	—	—	—	—	0000
CTMUCON2	EDG1MOD	EDG1POL	EDG1SEL3	EDG1SEL2	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	EDG2MOD	EDG2POL	EDG2SEL3	EDG2SEL2	EDG2SEL1	EDG2SEL0	—	—	0000
CTMUICON	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	—	0000

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

53.14 ELECTRICAL SPECIFICATIONS

Table 53-2: CTMU Current Source Specifications

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial				
Param No.	Sym	Characteristic	Min	Typ[1]	Max	Units	Conditions
	I_{OUT1}	CTMU Current Source, Base Range	—	550	—	nA	CTMUICON<9:8> = 01
	I_{OUT2}	CTMU Current Source, 10x Range	—	5.5	—	µA	CTMUICON<9:8> = 10
	I_{OUT3}	CTMU Current Source, 100x Range	—	55	—	µA	CTMUICON<9:8> = 11
	I_{OUT4}	CTMU Current Source, 1000x Range	—	550	—	µA	CTMUICON<9:8> = 00

Note 1: Nominal value at the center point of the current trim range (CTMUICON<15:10> = 000000).

53.15 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.

53.16 REVISION HISTORY

Revision A (March 2011)

This is the initial released revision of this document.

ISBN: 978-1-60932-949-5

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

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

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

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

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

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

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

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

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-213-7830
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

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

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

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

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

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

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

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

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

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

02/18/11