

---

## Microchip CTMU for Capacitive Touch Applications

---

<i>Author: Bruce Bohn Microchip Technology Inc.</i>
---

### INTRODUCTION

This application note describes the use of Microchip's Charge Time Measurement Unit (CTMU) for capacitive touch applications. The CTMU is an excellent peripheral for use in touch sensing applications. The following are some of the benefits of using the CTMU for touch sensing applications:

- Easy hardware setup
- Simple to use software algorithms, available free of charge
- High speed allows for greater scan rate of capacitive touch switches and for many other processor tasks
- Low cost and low component count; no external hardware other than a connection to the copper sensor pad

The purpose of this application note is to inform prospective users of the CTMU for capacitive touch on general usage guidelines, CTMU setup and software algorithms, hardware and layout considerations, and advanced capacitive touch applications, such as matrix keys, sliders and multiple press keys.

The CTMU is currently available on selected PIC24 and PIC18F parts. Wherever possible, sample code for both has been included.

All hardware examples used in this application note directly pertain to the PICDEM™ Touch Sense 2 Demonstration Board (Part Number DM164128). The PICDEM Touch Sense 2 Demo Board uses a PIC24F device-based microcontroller (PIC24FJ256GB110). This demo board is not necessary for understanding this application note.

### GENERAL FEATURES OF THE CTMU

The CTMU is made up of a constant current source and several logic blocks for it to operate:

- The constant current source is connected to the A/D converter of the microcontroller.
- A switch connected to the CTMU constant current source (and the A/D converter) is provided to allow accumulated charge to be drained.
- Two external pins are provided to trigger the constant current source.
- An output pin is also available for CTMU use.  
For the purposes of capacitive touch, the external pins for triggering the CTMU and the CTMU output pin are not used.

The connection of the CTMU to the A/D converter is all that is needed for capacitive touch applications. For a more in-depth discussion on the CTMU module and how it is used to measure charge or time, refer to the CTMU chapter of the "PIC24F Family Reference Manual" for PIC24F microcontrollers, or the appropriate data sheet for PIC18F microcontrollers on the Microchip web site (<http://www.microchip.com>).

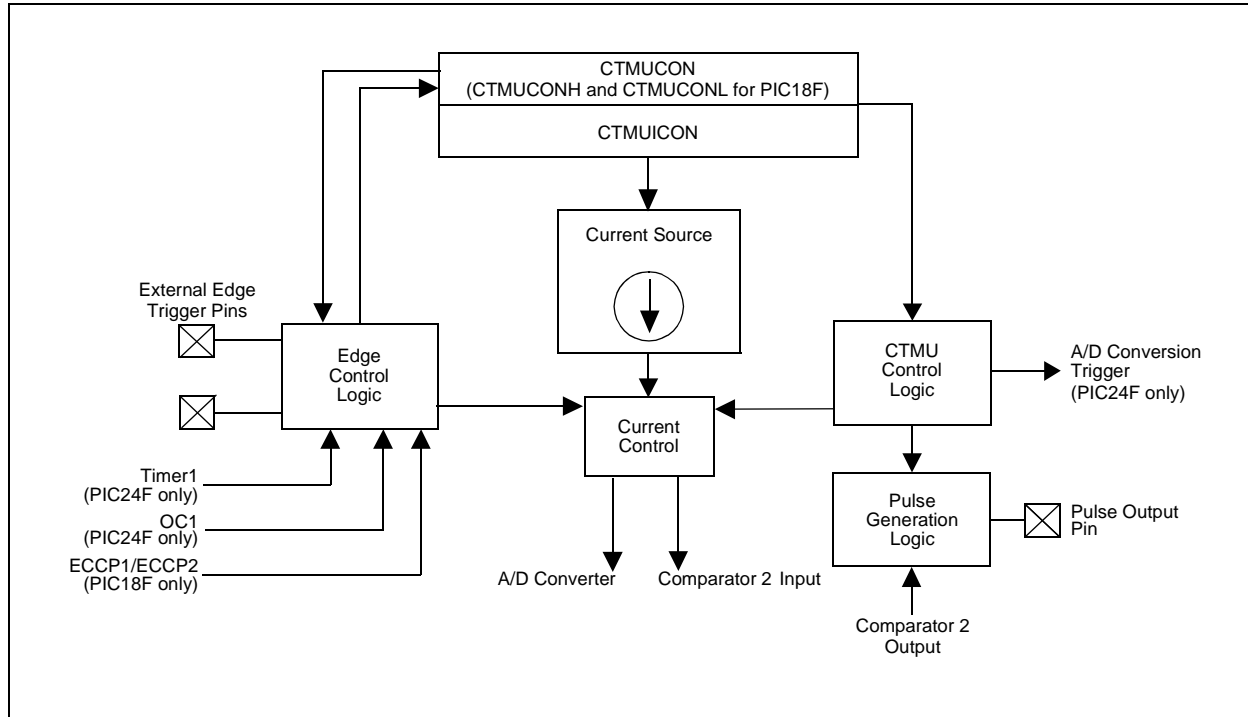
See Figure 1 for a block diagram of the CTMU hardware.

Note that there are a few slight differences in the CTMU module from the PIC18F and the PIC24F. The PIC18F CTMU module does not have an automatic ADC conversion trigger. This is of no consequence for capacitive touch applications. For clarity, the automatic ADC trigger present on the PIC24F has not been used in the sample code so that it more closely matches the PIC18F code.

Also note that the CTMU on the PIC18F has different internal trigger sources. (ECCP1 and ECCP2 for PIC18F and Timer1 and OC1 for PIC24F). This is mentioned purely for reference, and again, makes no difference for capacitive touch applications. The sample code listings in this application note manually set/clear the EDG1STAT1 and/or EDG2STAT bits in the CTMU control register for starting and stopping the CTMU current source.

# AN1250

FIGURE 1: CTMU BLOCK DIAGRAM



## OVERVIEW OF CAPACITIVE TOUCH APPLICATIONS

The CTMU is used in capacitive touch applications by applying the constant current source of the CTMU to the capacitive touch pad using the following equation:

$$I \times T = C \times V$$

Where:

- I is the constant current source of the CTMU
- T is a fixed period that the CTMU charges the capacitive touch circuit
- C is the capacitance of the touch circuit
- V is the voltage read by the A/D converter after the capacitive touch circuit is finished charging

It is possible to sense a relative shift in capacitance by observing a change in voltage. The above equation can be rearranged to:

$$V = (I \times T)/C$$

Since the CTMU current source is constant (I), the voltage present on the capacitive touch sensor (V) relies on two variables: the amount of time the touch circuit is charged (T) and the capacitive size of the touch circuit (C). If the amount of time the touch circuit is charged is held constant, then changes in the capac-

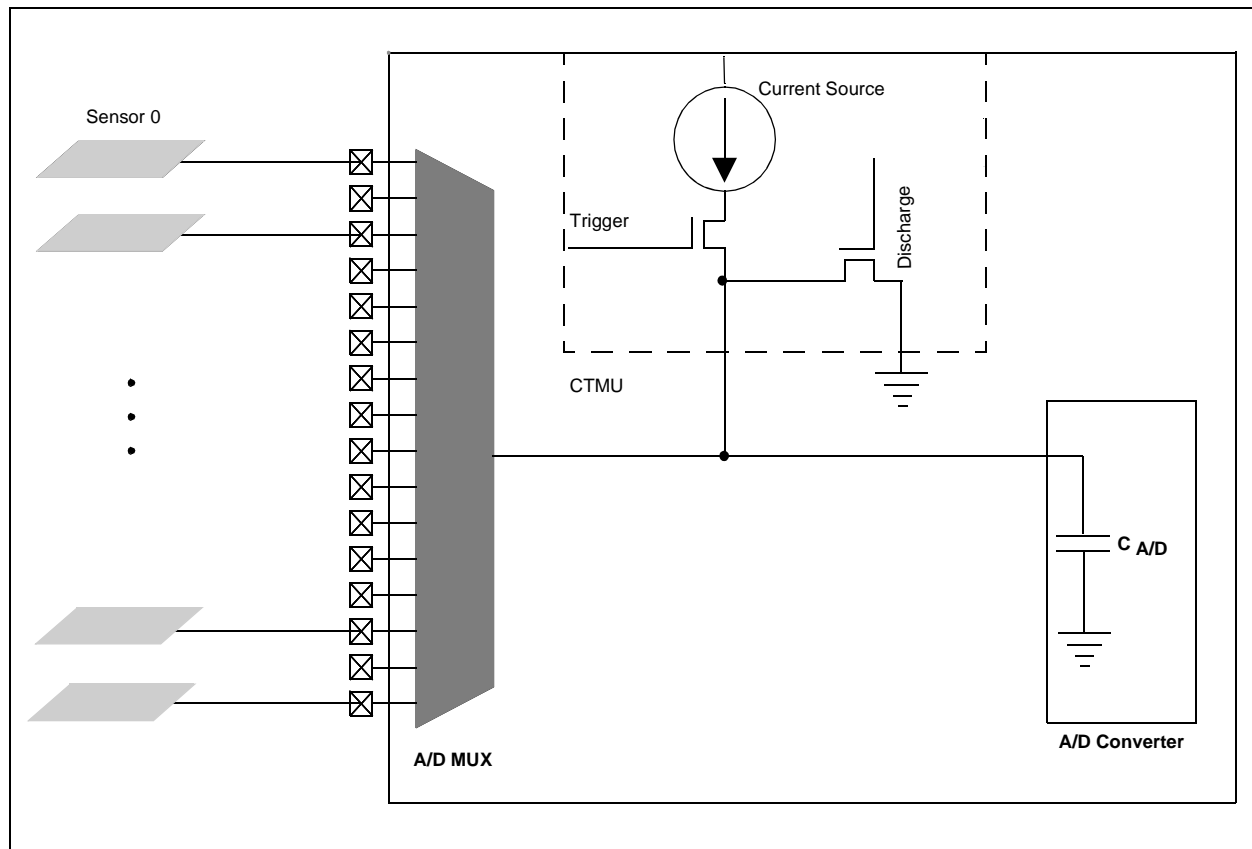
itance of the touch circuit will ultimately affect the voltage that the circuit charges to in the fixed period. The A/D converter is used to read the voltage that the touch circuit is charged to with the CTMU.

When the capacitance of a human finger is added to the touch sensor pad, the capacitance increases and the result is a lowering of the voltage seen by the A/D converter (since I and T are held constant).

### Setting Up of the CTMU

For a capacitive touch application, each sensor must be connected directly to a channel of the A/D converter. See Figure 2 for a block diagram of a Microchip microcontroller with a CTMU peripheral. This diagram illustrates that the CTMU is internally connected to the A/D converter and allows for selection of any of the A/D channels. With this configuration, a single CTMU unit can measure many capacitive touch sensors. Note that the CTMU has been simplified in order to show only the elements required for capacitive touch, namely the current source, the current source control and the current drain mechanism. These will be explained in detail in later sections. The current source control, labeled "Trigger" in Figure 2, is manually controlled in software by manipulating bits in the CTMU Control register.

FIGURE 2: CAPACITIVE TOUCH BLOCK DIAGRAM



# AN1250

The current source of the CTMU is available in three ranges: 0.55  $\mu\text{A}$ , 5.5  $\mu\text{A}$  and 55  $\mu\text{A}$ . The current range selection is made in the CTMUICON register. The current precision for each of the three ranges is  $\pm 20\%$ . For many capacitive touch applications, the highest current range setting (55  $\mu\text{A}$ ) works best. This allows for the quickest charging of the capacitive touch circuit. The CTMUICON register also has bits used to trim the current source in  $\pm 2\%$  increments up to  $\pm 62\%$  for each of the three current ranges.

The CTMU current source is enabled and disabled using software. Two control bits, EDG1STAT and EDG2STAT in the CTMU control register, determine if the current source is enabled. These bits are exclusively ORed. That is, if EDG1STAT and EDG2STAT are both set or cleared, the current source is off. If either bit is set while the other is cleared, the current source is enabled and charging the circuit.

The IDISSEN bit is enabled to drain charge from the A/D converter to insure the charging process begins at zero potential. If the bit is set, the circuit is connected to VSS (grounded). Note that the discharge of the entire circuit is not accomplished with this feature. This is because the A/D converter is not always connected to the external circuitry (i.e., touch sensor pad).

The CTMU Configuration register (CTMUCON) is set up so that the external triggers are not used (these pins may be used for general purpose I/Os). The same is true of the CTMU pulse output pin. The CTMU current

source is configured using the CTMUICON register. For detailed information on the CTMU registers on the PIC24F, refer to **Section 11. "Charge Time Measurement Unit (CTMU)"** of the "PIC24F Family Reference Manual".

The registers that control the CTMU on the PIC18F parts are identical, with the exception that they are 8-bit registers: CTMUCONH and CTMUCONL. The CTMU also has different external trigger sources available, namely the ECCP1 and ECCP2 Special Event Triggers. For detailed information on the CTMU registers for PIC18F, refer to the specific product data sheet.

See Example 1 for a typical setup of the required CTMU and A/D converter registers. Note that the CTMU has been configured so that the external pins are not enabled. For the purposes of this application note, all control of the CTMU is handled through software. The A/D converter is set up to do manual conversion.

For capacitive touch sensing, a relative change in capacitance due to the presence of a finger is required. The absolute measurement of capacitance is not required.

**Note:** All PIC18F code examples are written for the PIC18F46J11 family. For other PIC18F parts with the CTMU module, see the device data sheet for specific CTMU register configurations and CTMU usage.

## EXAMPLE 1: CTMU AND A/D CONVERTER SETUP FOR PIC24F

```
//setup CTMU
//CTMUCON
CTMUCONbits.CTMUEN = 0; //make sure CTMU is disabled
CTMUCONbits.CTMUSIDL = 0; //CTMU continues to run in idle mode
CTMUCONbits.TGEN = 0; //disable edge delay generation mode of the CTMU
CTMUCONbits.EDGEN = 0; //edges are blocked
CTMUCONbits.EDGSEQEN = 0; //edge sequence not needed
CTMUCONbits.IDISSEN = 0; //Do not ground the current source
CTMUCONbits.CTTRIG = 0; //Trigger Output is disabled
CTMUCONbits.EDG2POL = 0;
CTMUCONbits.EDG2SEL = 0x3; //Edge2 Src = OC1 (don't care)
CTMUCONbits.EDG1POL = 1;
CTMUCONbits.EDG1SEL = 0x3; //Edge1 Src = Timer1 (don't care)

//CTMUICON
CTMUICON = 0x300; //55uA
CTMUICON.ITRIM = 0; //Nominal - No Adjustment

//setup A/D converter

AD1PCFGL = 0x0000;
AD1CON1 = 0x0000;
AD1CHS = 0x0000; //select the analog channel 0
AD1CSSL=0x0000;
AD1CON1bits.FORM = 0x0; //Unsigned fractional format
AD1CON3 = 0x0000; //bits.ADRC=0;
AD1CON2 = 0x0000;
AD1CON1bits.ADON = 1; //Turn On A/D
CTMUCONbits.CTMUEN = 1; //Enable CTMU
```

**EXAMPLE 2: CTMU AND A/D CONVERTER SETUP FOR PIC18F**

```

//setup CTMU
//CTMUCON
CTMUCONHbits.CTMUEN = 0;           //make sure CTMU is disabled
CTMUCONHbits.CTMUSIDL = 0;        //CTMU continues to run in idle mode
CTMUCONHbits.TGEN = 0;           //disable edge delay generation mode of the CTMU
CTMUCONHbits.EDGEN = 0;          //edges are blocked
CTMUCONHbits.EDGSEQEN = 0;       //edge sequence not needed
CTMUCONHbits.IDISSEN = 0;        //Do not ground the current source
CTMUCONHbits.CTTRIG = 0;         //Trigger Output is disabled
CTMUCONLbits.EDG2POL = 0;
CTMUCONLbits.EDG2SEL = 0x0;      //Edge2 Src = ECCP2 (don't care)
CTMUCONLbits.EDG1POL = 1;
CTMUCONLbits.EDG1SEL = 0x1;     //Edge1 Src = ECCP1 pin (don't care)
//CTMUICON
CTMUICON = 0x03;                 //55uA
CTMUICON.ITRIM = 0;             //Nominal - No Adjustment
//setup A/D converter
ANCON0bits.PCFG = 0x00;
ANCON1 = 0x0000;
ADCON0bits.CHS = 0x00;          //select the analog channel 0
ADCON1bits.ADFM = 0x00;         //right justified result
ADCON1bits.ADON = 1;            //Turn On A/D
CTMUCONHbits.CTMUEN = 1;        //Enable CTMU

```

In the case of the PIDE<sup>M</sup> Touch Sense 2 Demo Board, all 16 A/D channels are used for capacitive touch. Reading of each of the sensor channels (A/D channels) is controlled by Timer1. Timer1 is setup to fire at a 1 ms interval. Each time that the Timer1 interrupt handler runs, it increments the channel number of the A/D and

the capacitive sensor connected to that channel is read. After all 16 touch sensors have been read, the Timer1 interrupt handler sets a flag called, "dataReadyCTMU", and the main routine uses this flag to know when to process the new data just gathered from all 16 capacitive touch sensors.

# AN1250

The software routine for reading each of the sensor channels consists of the following 13 steps:

1. Set all of the A/D converter channels to digital I/O pins and set them to logic zero ('0').

This has the effect of grounding all of the capacitive touch sensor pads, so they are at zero (0) volts potential.

2. Set the A/D channel to be read to an analog input.
3. Set the A/D converter to the selected channel.
4. Set the discharge bit, IDISSEN, of the CTMUCON register.

This step insures that the internal capacitance of the A/D converter starts at zero (0) volts potential.

5. Wait for charge to drain (a few NOPs).
6. Clear the discharge bit, IDISSEN.
7. Set the sample bit, SAMP (AD1CON1), to begin sampling the selected channel (PIC24F only).
8. Set either of the edge status bits of the CTMU.

For PIC24F:  
EDG1STAT OR EDG2STAT (CTMUCON) bits

For PIC18F:  
EDG1STAT OR EDG2STAT (CTMUCONL) bits

This step turns on the CTMU current source to the selected A/D channel and begins charging the connected capacitive touch sensor.

9. Wait for a fixed period of time. For most capacitive touch applications, this is approximately 2-10  $\mu$ S. In this case, the fixed time period is a loop set up to do nothing until the allotted time has expired.

10. Clear the edge status bit that was set above.

For PIC24F:  
EDG1STAT OR EDG2STAT (CTMUCON) bits

For PIC18F:  
EDG1STAT OR EDG2STAT (CTMUCONL) bits

This step turns off the CTMU current source and stops charging the connected capacitive touch sensor.

11. Initiate an A/D conversion on the selected channel:

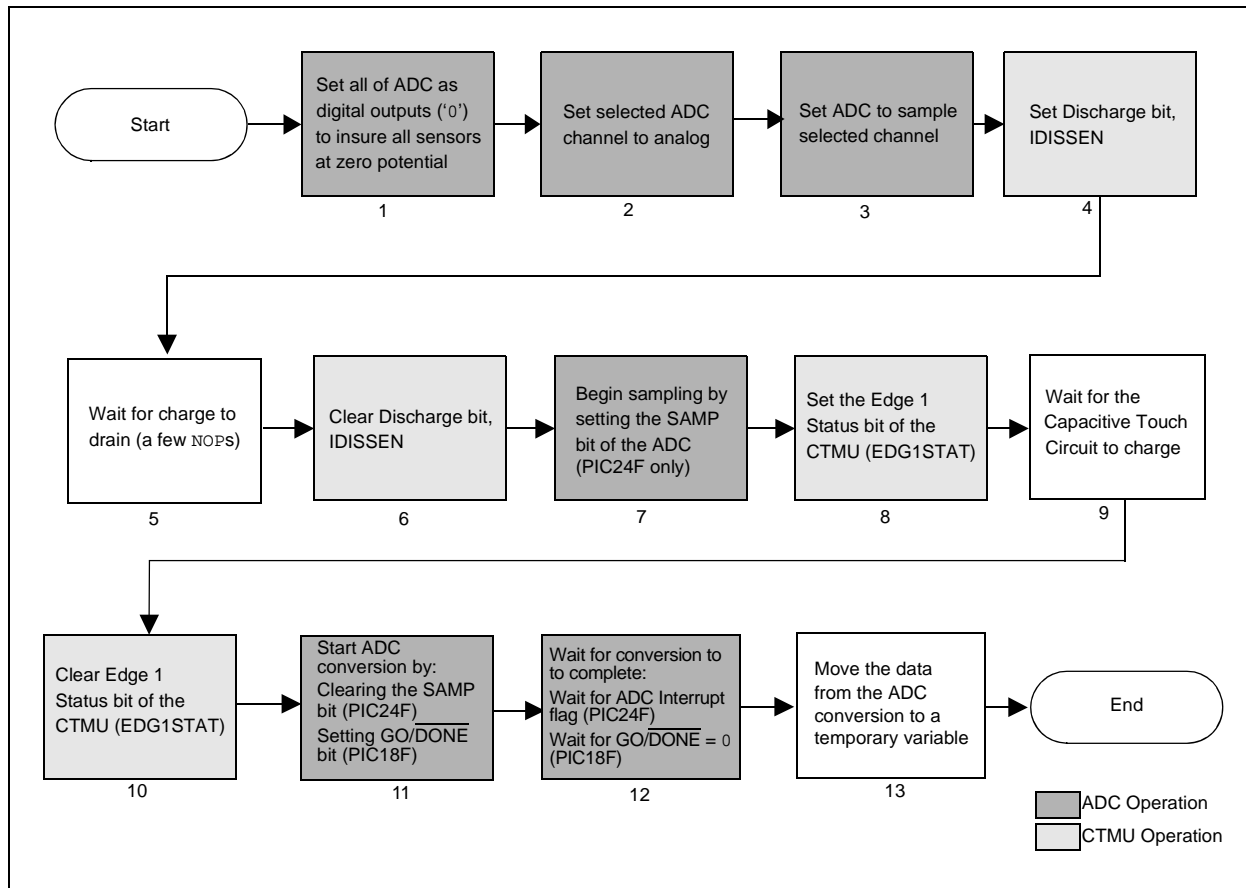
- a) On PIC24F, clear the SAMP (AD1CON1).
- b) On PIC18F, set the GO/DONE bit (ADCON0).

12. Wait for the conversion to complete.

13. Store the value read by the A/D converter.

Figure 3 provides a software flowchart of the CTMU capacitive touch sensor, reading as outlined above.

**FIGURE 3: CTMU CAPACITIVE TOUCH SENSOR READ FLOWCHART**



Example 3 provides the equivalent code for reading the capacitive touch sensors using the CTMU on a PIC24F device. Example 4 has the same algorithm for a PIC18F device. Note that the channel number being read is referenced by the `Index` variable. The `ADSELECT` array is setup to have the correct bits set for

the corresponding `Index` variable so that the correct channel is set to an analog A/D. The A/D value is temporarily stored in the variable, `immediateValue`. The `LoopCount` determines how long the CTMU current source charges the capacitive touch circuit.

### EXAMPLE 3: READING THE CAPACITIVE TOUCH SENSOR CIRCUIT SOFTWARE ALGORITHM (PIC24F)

```
//Read CTMU (Get the raw sensor reading)
AD1PCFGL= 0xFFFF; //set all A/D channels to digital I/O pins
TRISB = 0x0000;
LATB = 0x0000;
PORTB = 0x0000; //set all channels to logical 0 outputs
Nop(); Nop(); Nop(); Nop();
Nop(); Nop(); Nop(); Nop();
//Wait for charge to drain
TRISB = ADSELECT[Index]; //set selected channel to input
AD1PCFGL = ~ADSELECT[Index]; //set selected channel to analog A/D input
Nop(); Nop(); Nop(); Nop();
Nop(); Nop(); Nop(); Nop();
//wait for A/D to connect to channel
AD1CHS = Index; //select A/D channel
CTMUCONbits.IDISSEN = 1; //Drain any charge on the A/D circuit
Nop(); Nop(); Nop(); Nop(); Nop();
//Wait for charge to drain
CTMUCONbits.IDISSEN = 0; //Stop discharge of A/D circuit
IFS0bits.AD1IF = 0; //Make sure A/D interrupt flag = 0
AD1CON1bits.SAMP = 1; //Manually start sampling
CTMUCONbits.EDG2STAT = 0; // Make sure edge2 is 0
CTMUCONbits.EDG1STAT = 1; // Set edge1 - Start Charge
for (count = 0; count < loopCount; count++); // Delay for CTMU charge time
CTMUCONbits.EDG1STAT = 0; //Clear edge1 - Stop Charge
IFS0bits.AD1IF = 0;
AD1CON1bits.SAMP = 0; //Clear SAMP bit to begin manual A/D conversion
while(!IFS0bits.AD1IF); //Wait for the A/D conversion to finish
immediateValue = ADC1BUF0; //Read the value from the A/D conversion
AD1CON1bits.SAMP = 0;
IFS0bits.AD1IF = 0;
AD1CON1bits.DONE = 0; //Make sure A/D bits are cleared
//End of CTMU read
```

## EXAMPLE 4: READING THE CAPACITIVE TOUCH SENSOR CIRCUIT SOFTWARE ALGORITHM (PIC18F)

```
//Read CTMU (Get the raw sensor reading)
ANCON0bits.PCFG= 0xFF; //set all A/D channels to digital I/O pins
ANCON1bits.PCFG= 0x1F; //set all A/D channels to digital I/O pins
TRISA = 0x00;
TRISB = 0x00;
TRISC = 0x00;
TRISE = 0x00;
LATA = 0x00;
LATB = 0x00;
LATC = 0x00;
LATE = 0x00;
PORTB = 0x00; //set all channels to logical 0 outputs
Nop(); Nop(); Nop(); Nop();
Nop(); Nop(); Nop(); Nop();
//Wait for charge to drain
TRISA = ADPASELECT[Index];
TRISB = ADPBSELECT[Index];
TRISC = ADPCSELECT[Index];
TRISE = ADPESELECT[Index]; //set selected channel to input
ANCON0bits.PCFG = ~AD0SELECT[Index];
ANCON1bits.PCFG = ~AD1SELECT[Index]; //set selected channel to analog A/D input
Nop(); Nop(); Nop(); Nop();
Nop(); Nop(); Nop(); Nop();
//wait for A/D to connect to channel
ADCON0bits.CHS = Index; //select A/D channel
CTMUCONHbits.IDISSEN = 1; //Drain any charge on the A/D circuit
Nop(); Nop(); Nop(); Nop(); Nop();
//Wait for charge to drain
CTMUCONHbits.IDISSEN = 0; //Stop discharge of A/D circuit
Nop(); Nop();
CTMUCONLbits.EDG2STAT = 0; // Make sure edge2 is 0
CTMUCONLbits.EDG1STAT = 1; // Set edge1 - Start Charge
for (count = 0; count < loopCount; count++); // Delay for CTMU charge time
CTMUCONLbits.EDG1STAT = 0; //Clear edge1 - Stop Charge
AD1CON0bits.GODONE = 1; //Set Go bit to begin A/D conversion
while(ANCON0bits.GODONE); //Wait for the A/D conversion to finish
immediateValue = ADRESH;
immediateValue = immediateValue << 8;
immediateValue = immediateValue + ADRESL; //Read the value from the A/D conversion
//End of CTMU read
```

The `main()` function contains a `while(1)` loop (endless) that checks for the `dataReadyCTMU` flag (set by the Timer1 interrupt handler). If the `dataReadyCTMU` flag is not set, other tasks are performed while it waits. In the case of the PICDEM Touch Sense 2 Demo Board, communication of data over the USB is serviced. If the `dataReadyCTMU` flag is set, the `main()` function clears the flag and calls the routines necessary to process the newly acquired data. The `main()` function temporarily disables Timer1 to insure that no new data is introduced in the middle of the routines that are handling data processing. After all processing of the capacitive touch

sensor data is complete, and the LED display information is also updated, Timer1 is re-enabled, and the whole process begins again.

The structuring of the code in this manner is done to make the system deterministic. Using a fixed timer to read each capacitive touch sensor channel at a 1 ms rate fixes the update rate to approximately 16 ms (1 ms times 16 channels). The actual update rate is slightly slower, since Timer1 is disabled while the new data is processed.



## IMPLEMENTATIONS OF SPECIAL PURPOSE CAPACITIVE TOUCH

This section describes some of the special handling of the data that is available from reading the capacitive touch sensors using the CTMU. The following topics are covered:

- Software algorithms for reliable touch operation
- Software algorithms for special purpose capacitive touch sensor implementations
- Physical board properties for optimal capacitive touch sensors

### Software Algorithms for Reliable Touch Operation

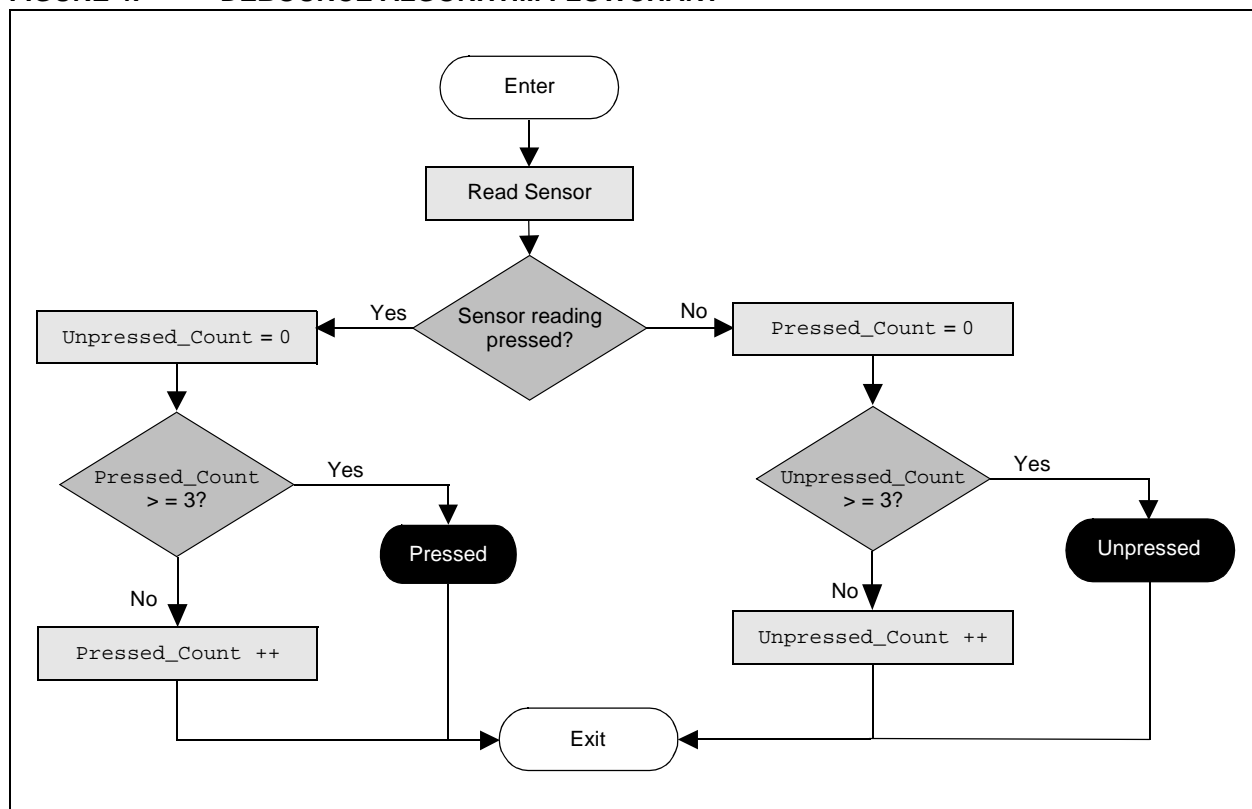
There are several software methods used to combat problems with false sensing of capacitive touch sensing circuits. The following software algorithms are used not only in the PICDEM Touch Sense 2 Demo Board, but throughout all of the capacitive touch solutions offered by Microchip.

- First method – This is to use a slow averaging routine. This is implemented by simply adding the value read for any capacitive touch sensor circuit into a running average. The key to this method is that a value is only added to the average at a rate much slower than the actual rate that the

capacitive touch sensor is being read. On the PICDEM Touch Sense 2 Demo Board, for example, the average is only updated on every 16<sup>th</sup> read of the capacitive sensor channel. Since the update rate is 16 ms, the slower moving average value is updated at a rate of 256 ms. This slower moving value is then compared to the immediate value. If the immediate value varies by more than what has been established for the trip point for that capacitive sensor, the sensor is determined to be “pressed”, or touched. For more information on the method of slow averaging, refer to the specific averaging section in AN1103, “*Software Handling for Capacitive Sensing*”.

- Second method – This is a debouncing routine. This algorithm is very similar to that used for debouncing of mechanical switches. In the case of capacitive sensors, the debouncing acts as a noise filtering mechanism. Since noise by definition is non-periodic, it should not occur repetitively in multiple sequential scans of a capacitive sensor channel. Figure 4 provides the routine setup for both a pressed and unpressed condition. It requires three consecutive values of pressed or unpressed before the actual state of the capacitive touch sensor is ultimately determined. The value can be changed to suit the application. It should be noted that if a higher number is used, a slower response time will be observed.

**FIGURE 4: DEBOUNCE ALGORITHM FLOWCHART**



- Third method – This is used to improve capacitive touch sensor reliability in oversampling. This method takes advantage of the high-speed capability of the CTMU. Using the PICDEM Touch Sense 2 Demo Board as an example, each reading of any of the 16 capacitive touch channels takes approximately 12  $\mu$ s. Since the scan rate has been determined by Timer1 to be 1 ms, there is sufficient time to reread the capacitive touch sensor channel multiple times. The PICDEM Touch Sense 2 Demo Board reads each channel 64 times. The sum of all 64 readings for the channel is then divided by 64, and this final value is used as the data used by all other software routines in the application.

To use this method, a loop is added around the routine that reads the capacitive touch sensor. The sum of the reads is tallied for each iteration through the loop, and at the end of all of the iterations, the sum is divided by the number of times through the loop (64 in this case). This routine has the effect of a low-pass filter. Since the average value of multiple readings is taken, a noise source that may cause an incorrect reading is averaged in with multiple correct readings and has a very minimal effect on the overall value.

- Fourth method – This is similar to that used for slow averaging. In this scenario, an array of previously read values is kept, and when a new value is read, it is introduced into the data array and the oldest value is removed. The average value of the array is used as the data value passed to all of the other software routines in the application. This method again causes a spurious incorrect value to have only a slight impact on the final value used. This routine is used on every value read by the capacitive touch routine.

Note that the PICDEM Touch Sense 2 Demo Board uses all of these methods combined to provide the most reliable sensing solution.

Capacitive touch sensors can go beyond a simple button that is pressed and released. In this section, we will explore some different capacitive touch topologies, such as multiple key press, paired button press, matrix keyboard implementation and the use of capacitive touch sensors as a slider.

## Multiple Keys Press

Using software algorithms, it is possible to detect when two or more buttons are simultaneously pressed. In a multiple press scenario, the buttons are specifically designed to allow for multiple presses. The physical layout of the touch pad sensors is such that an area exists for a single button press, and the button area is

extended to an area shared with an adjacent touch pad sensor. The PICDEM Touch Sense 2 Demo Board has an area designated as the “directional wheel”. This area has four capacitive touch sensors laid out as the four main ordinal compass directions (north, south, east and west). Each of these sensors is extended to an area equal to one half (1/2) of the main portion of the sensor. These “dual” touch sensor areas make up four more directions (northeast, southeast, southwest and northwest). Each of the four main sensors also extends to the center of the wheel, with each having one quarter (1/4) of the area of the main portion of the sensor. A “quad” touch sensor is then formed in the center. See **Appendix A: “PICDEM™ Touch Sense 2 Demo Board”**. Note that the cover plate is removed in Figure A-2. The software algorithm now looks first for all four sensors being pressed at approximately one quarter (1/4) of the value of a single pressed key. If all four are in range, it is determined that the center button is pressed. Next, it looks for two adjacent keys pressed at approximately one half (1/2) of a single pressed key. Lastly, single key presses at a full level of the trip value are checked. Using this multiple press setup, nine distinct keys are created from only four capacitive touch sensor inputs. The directional wheel works nicely when a finger is moved around the wheel, either clockwise or counter-clockwise. It provides a smooth fluid-like transition between the keys.

Capacitances for each of the sensors should be matched as closely as possible, especially when considering that the software algorithm is searching for multiple button presses. The trace lengths to each sensor and adjacent sensors, LEDs and other board traces become important. The CTMU current trimming capability is a great aid in achieving a “level playing field” in this respect. The PICDEM Touch Sense 2 Demo Board uses this trimming capability to match up each of the four sensors of the directional wheel.

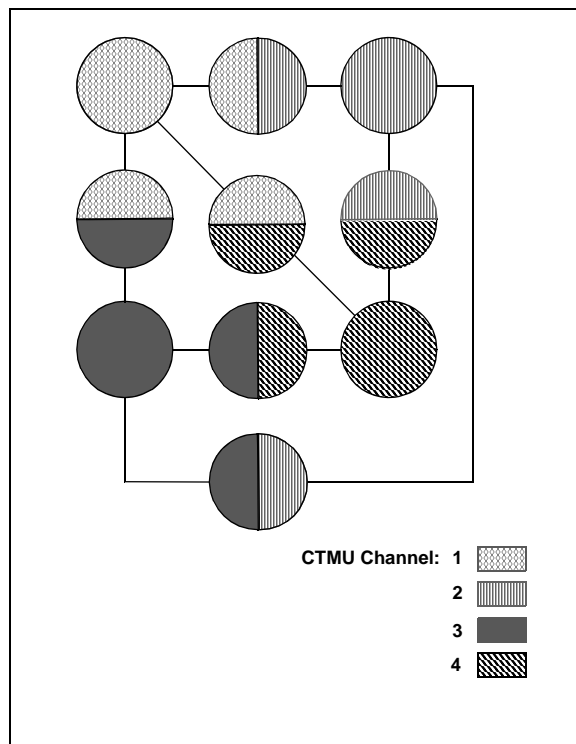
This was done using a manual process, and each of the A/D readings for the sensors was adjusted using the CTMUICON register to slightly alter the CTMU current source up or down so that the unpressed readings matched. The adjusted values required for each channel were then stored in an array, and the CTMUICON register was written to with a value that was selected from the array to match the selected channel being read. This is done for each channel and is applied to the CTMUICON register at the beginning of the CTMU channel read routine provided in Example 3.

**Note:** For clarity, the code for CTMU current adjustment is not included in the example code, but may be found in the code listing for the PICDEM Touch Sense 2 Demo Board.

## Paired Keys Press

The paired keys press is similar to the multiple key press with the exception that all of the combinations of one half (1/2) are used. In Figure 5, it can be seen that each button is made up of either a full circle single touch pad, or two half (1/2) circle touch pads. The software for the paired press is similar to that used for the multiple key press described above. The “half” pressed keys are looked for first, and then single keys are scanned for if no paired press keys are found. This implementation has the distinction of generating 10 keys from only four capacitive touch sensor inputs. It should be noted that it is somewhat difficult to differentiate between two single keys being pressed and a single paired press key.

**FIGURE 5: PAIRED KEY PRESS CAPACITIVE TOUCH SENSOR KEYS**



## Matrix Keys Implementation

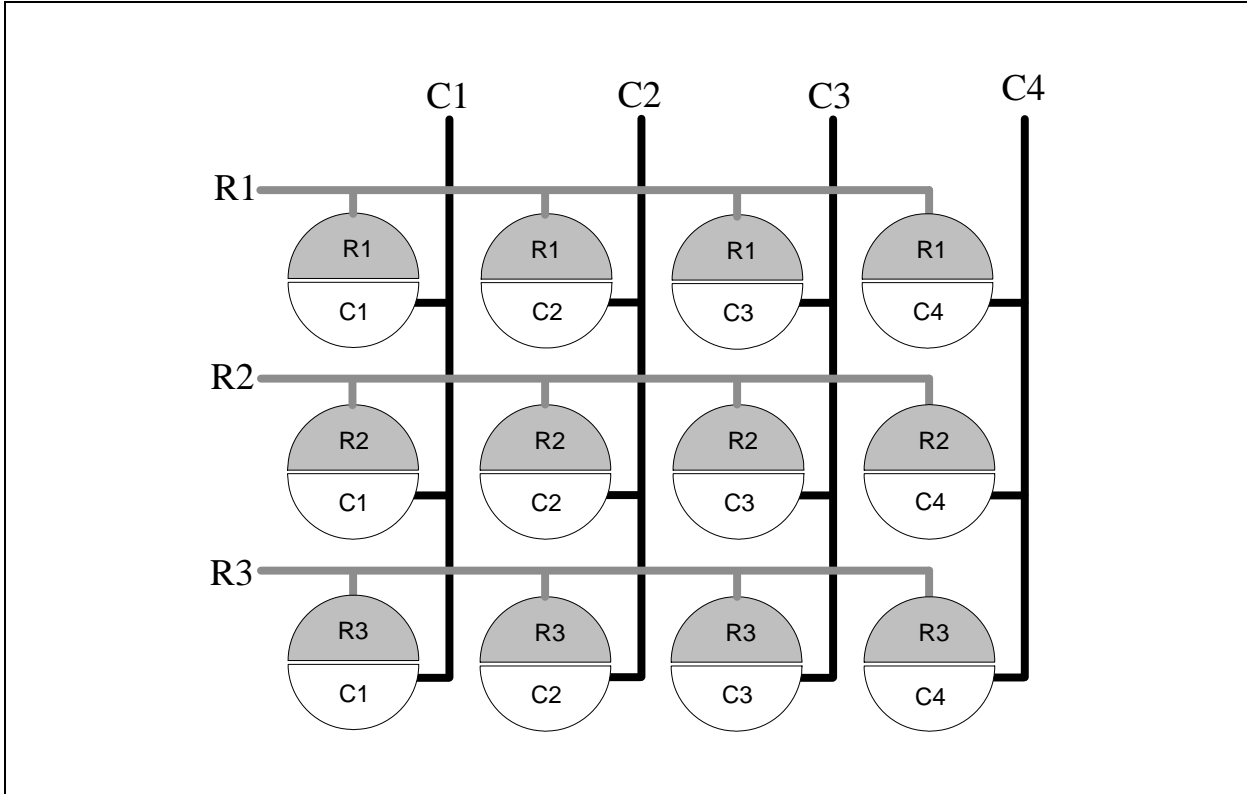
Matrix keys use a set of capacitive touch sensors arranged in rows and columns to provide a maximum number of buttons or keys with respect to the number of capacitive touch sensor inputs. The PICDEM Touch Sense 2 Demo Board makes use of a matrix of four rows by five columns (4 x 5) to implement a 20-key calculator type keypad. Notice that the number of channels (9) yields a total of 20 distinct keys. The number of possible keys is the multiplication of the number of rows and columns (4 x 5), while the number of channels required is the sum of the number of rows plus the number of columns (4 + 5). **Appendix A: “PICDEM™ Touch Sense 2 Demo Board”** depicts the PICDEM Touch Sense 2 Demo Board with the cover plate removed. The keypad matrix is in the center of the board. Each key is a unique combination of a row and a column of the matrix. Each key is made up of two touch sensors shaped as half circles, one half circle for the row and one half circle for the column. See Figure 6 for a visual example of a keypad matrix. The software algorithm for the keypad matrix looks for a row pressed and a column pressed, and decodes the key pressed. The software for this implementation also looks for the most pressed row and column. This is done since the proximity of the sensors can sometimes cause adjacent rows or columns to cross the tripped threshold level. It is important to note that using matrix keys causes the overall capacitance to increase. For example, a matrix with four (4) rows will have approximately two (2) times the capacitance of a single sensor key of equal size. This factor becomes important, since the software is looking for the relative change in capacitance due to a finger on the capacitive touch sensor key. Another key factor is that since the rows and columns are very close to each of the actual keys, capacitive coupling also increases the amount of capacitance seen by each of the sensor inputs.

It is important to keep the capacitances for each of the rows and each of the columns matched as closely as possible, especially when considering that the software algorithm is searching for the “most pressed” row and column. The trace lengths to each row/column and adjacent sensors, LEDs and other board traces become important. The CTMU current trimming capability, mentioned in the **“Multiple Keys Press”** section, is applicable to the matrix keypad as well.

The difference between paired press and matrix keys is that for the matrix, there is a key for each combination of row and column, and for the paired press, there is a key assigned to each capacitive touch channel and a key assigned to each combination of two capacitive touch channels.

# AN1250

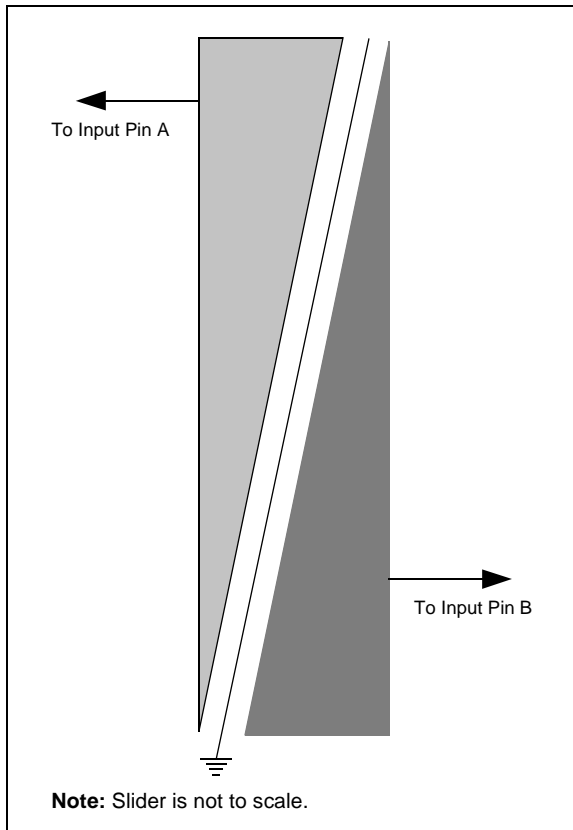
FIGURE 6: MATRIX KEY PRESS CAPACITIVE TOUCH SENSOR KEYS



## CAPACITIVE TOUCH SLIDER

The capacitive touch slider senses the position of a finger along a strip of the circuit board. The area designated as the strip has two capacitive sensor channels, each formed in the shape of a triangle (see Figure 7 and Figure A-2). The schematic of the PICDEM Touch Sense 2 Demo Board with the cover plate removed shows the slider toward the right hand side of the board.

**FIGURE 7: SLIDER IMPLEMENTATION COPPER PAD SHAPE**



Notice that the trace between the two triangular pads is grounded. This trace provides additional isolation between the two triangular capacitive touch sensors by reducing the effect of capacitive coupling.

The slider works by reading the amount of shift in capacitance from a normal untouched sensor. The slow moving average described earlier is used to track the untouched “base” capacitance of each of the two sensors. When the value is shifted from this base value to a value where the trip point is reached for either of the two slider sensors, the base value is then locked and the slow moving average is no longer used for slider position calculations until the value of both slider sensors is again above the trip threshold. The base values for each of the two sensors is captured and used for the slider position calculations. The software algorithm then looks at the change in the slider value

from the captured base value. The position of a finger on the slider is then determined by the following equations.

### EQUATION 1:

$$\begin{aligned} \text{Delta Left} &= (\text{Captured Left Base Capacitance} - \text{Current Left Value}) \\ \text{Delta Right} &= (\text{Captured Right Base Capacitance} - \text{Current Right Value}) \end{aligned}$$

Since the left triangular sensor has its point at the bottom of the slider, it is expected that the shift in capacitance from a finger on the slider at the bottom will be minimal, and maximum when a finger is positioned at the top. The position of the left sensor (scaled for a value of 0 to 100) is shown in Equation 2:

### EQUATION 2:

$$\text{Left Position} = (\text{Delta Left} * 100) / (\text{Delta Left} + \text{Delta Right})$$

Similarly, the right triangular sensor is large at the bottom of the slider and a finger positioned at the bottom would cause the largest shift in capacitance, while a finger at the top of the slider would produce a minimum amount of capacitance shift. The position of the right sensor (again scaled for a value 0 to 100) is shown in Equation 3:

### EQUATION 3:

$$\text{Right Position} = ((1 - \text{Delta Right}) * 100) / (\text{Delta Left} + \text{Delta Right})$$

Finally, the two values may be averaged for the final position calculation:

### EQUATION 4:

$$\text{Position} = (\text{Left Position} + \text{Right Position}) / 2$$

It should also be noted that when neither the left nor right slider has crossed the tripped value (base or unpressed), the position is not calculated. This prevents jitter between values that are displayed on the slider LEDs.

To achieve maximum linearity and accuracy of the slider, it is important to keep the capacitance for each of the triangular sensors matched as closely as possible since the software algorithm is using and comparing the values for two triangular sensors. The trace lengths to each triangular sensor and adjacent sensors, LEDs and other board traces become important. The CTMU current trimming capability, mentioned in the “**Multiple Keys Press**” section, is applicable to the slider sensors as well.

## APPLICATION-SPECIFIC CONSIDERATIONS

Every application will have unique needs relative to its capacitive touch features. Some of the factors include: environment, response time, number and type of capacitive touch sensors, and the number of other tasks that the microcontroller is ultimately responsible for (such as USB, graphics, I<sup>2</sup>C™, SPI, UART, etc).

There are typically three common variables that affect overall performance of capacitive touch sensors. These are:

- Printed Circuit (PC) board layout including sensor pad size, shape and proximity to other sensors
- Material and thickness of covering over the capacitive touch sensors
- Noise susceptibility, both conducted and radiated

### PC Board Layout

Basic considerations for PC Board (PCB) layout include keeping traces as short as possible. Traces to capacitive touch sensors should not run for long lengths in parallel due to capacitive coupling effects. The PICDEM Touch Sense 2 Demo Board was laid out so that the most sensitive area (i.e., the matrix keypad) was closest to the microprocessor, so that the traces run to the capacitive sensors were the shortest, and capacitance due to trace length was kept to a minimum. The subject of board layout practice can be quite intricate; refer to AN1102, "Layout and Physical Guidelines for Capacitive Sensing" for more information. The pad shape, size and proximity to other sensors is also covered in AN1102. The PICDEM Touch Sense 2 Demo Board uses a half inch (½") diameter pad with a quarter inch (¼") spacing between adjacent sensors. The matrix keypad uses half circle sensor pads that have a ½" diameter. The directional wheel sensors have a total area that is approximately double that of the ½" diameter pads. This was required to allow the double and quad button areas.

## Capacitive Touch Covering (Overlay) Considerations

The material and thickness of the covering over the capacitive touch sensors is very significant. It determines how much of the capacitance of a human finger is actually transmitted to the capacitive touch sensor. Therefore, thinner is better than thicker. If a material is too thin, it may not be mechanically sufficient to support and protect the circuit board. The dielectric constant of the material also is a determining factor. A higher dielectric constant will give better results. Two common materials used are PLEXIGLAS™ and tempered window glass. PLEXIGLAS has a dielectric constant of 2.5-3.5, while window glass has a dielectric constant in the range of 4-8. More information on the topic of covering capacitive touch sensors can be found in AN1102, "Layout and Physical Guidelines for Capacitive Sensing".

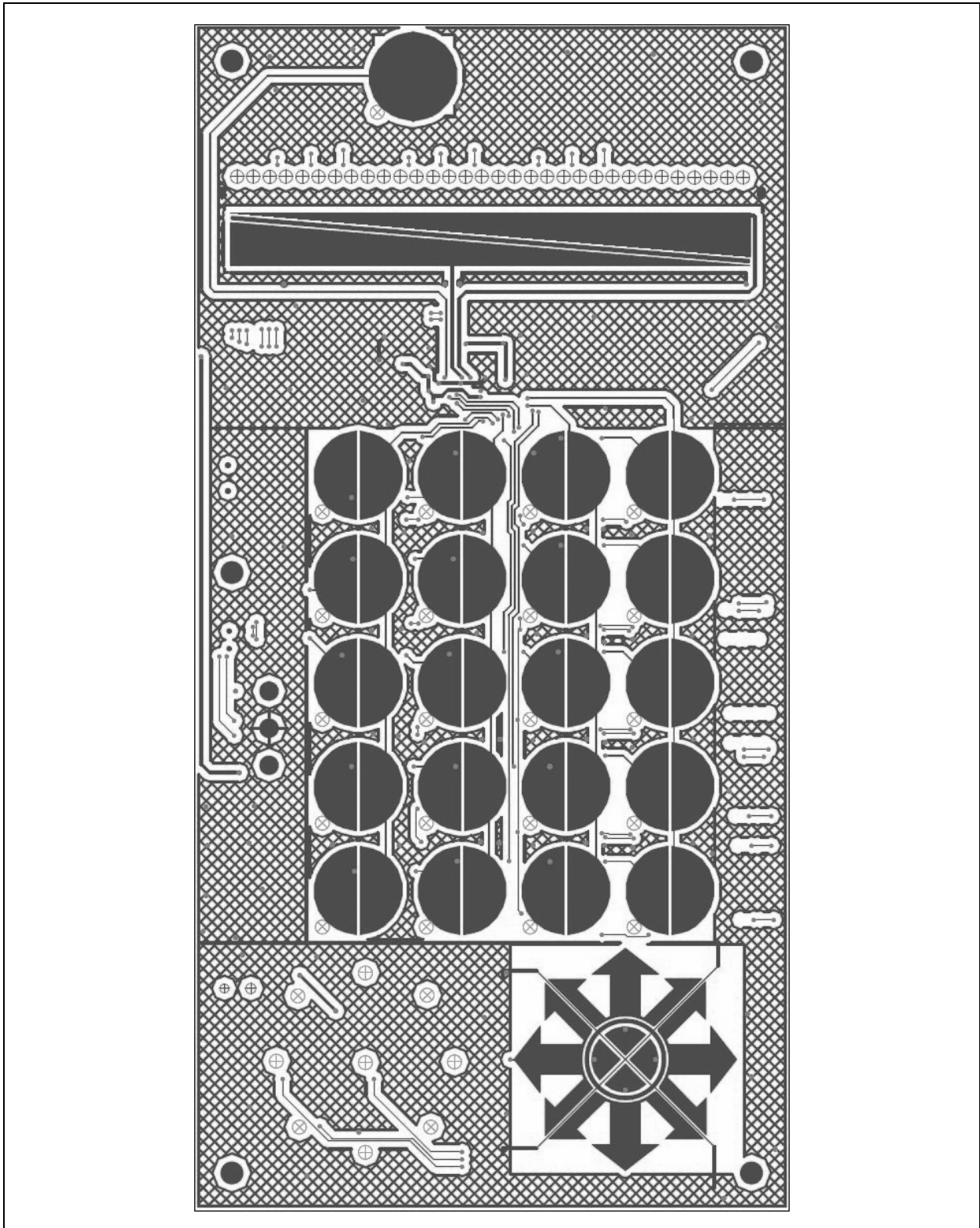
### Noise Susceptibility

The existing noise, which affects capacitive touch applications, should always be considered. The factors affecting how to protect against noise are very application-specific. The PICDEM Touch Sense 2 Demo Board, for example, is open on both the front and back. A hatched ground pattern was added to the back of the board wherever possible to minimize the effects of noise that could be introduced from the bottom of the board. The cross hatched pattern provides a good path to conduct unwanted noise to ground, and at the same time, it does not cause an appreciable increase in the capacitance of the touch sensors due to the coupling effect of the sensor pad through the PC board material to the ground plane.

Some applications may have a completely shielded enclosure which would allow for less ground plane added to the PC board. Another consideration is the proximity of the capacitive touch sensors to other noise producing circuitry on the board, such as communication busses, inverting power supplies for back lighting displays, RF radio transmitters such as those used for Wi-Fi or other radio based communications, and so on. In the case of a radio transmitter and capacitive touch sensors, it is best to read the buttons only when the RF transmitter is disabled.

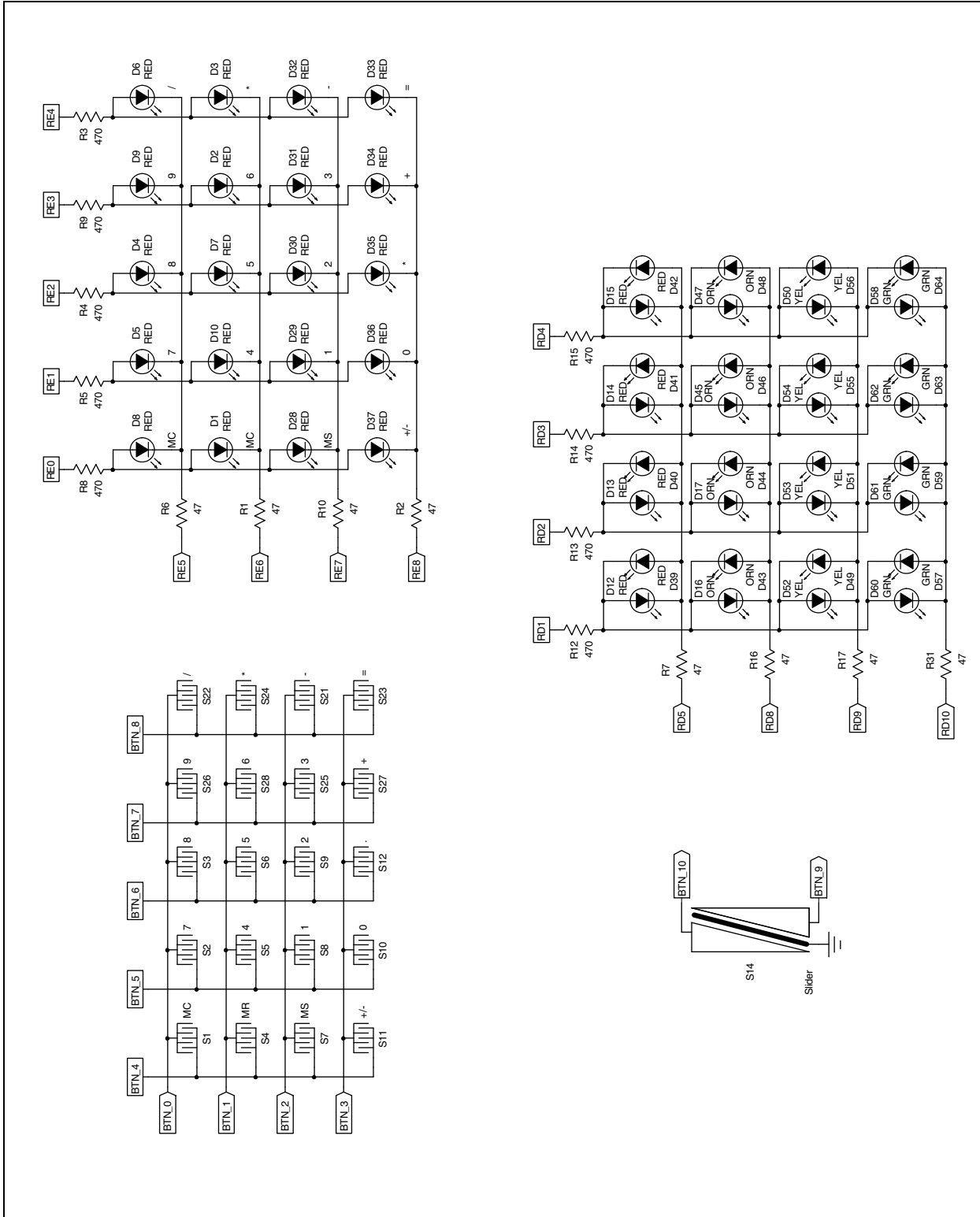
**APPENDIX A: PICDEM™ TOUCH SENSE 2 DEMO BOARD**

Figure A-2 displays the PICDEM Touch Sense 2 Demo Board schematic with the cover plate removed.

**FIGURE A-1: PICDEM™ TOUCH SENSE 2 DEMO BOARD WITH COVER REMOVED**

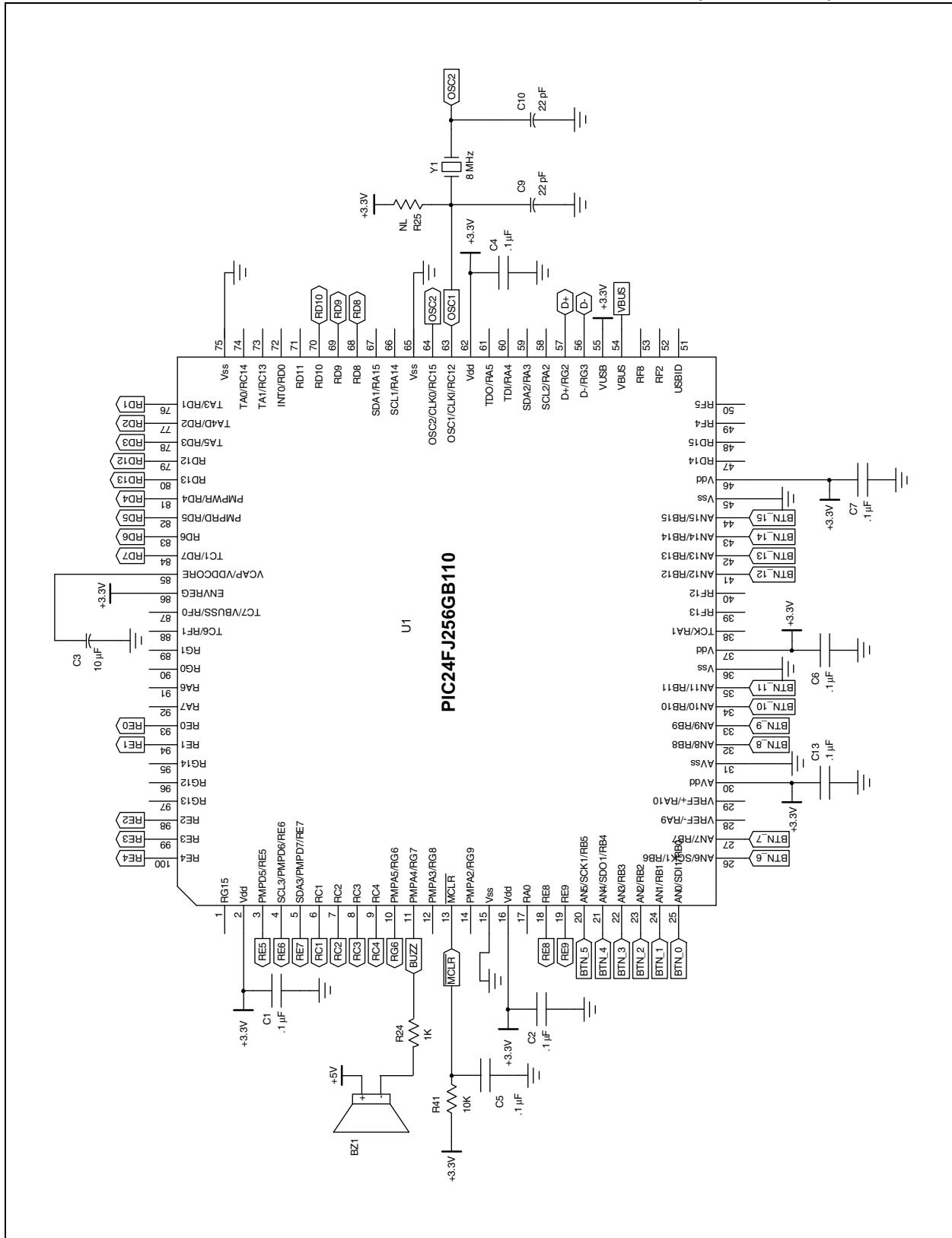
# AN1250

FIGURE A-2: PICDEM™ TOUCH SENSE 2 DEMO BOARD SCHEMATIC (PAGE 1 OF 3)



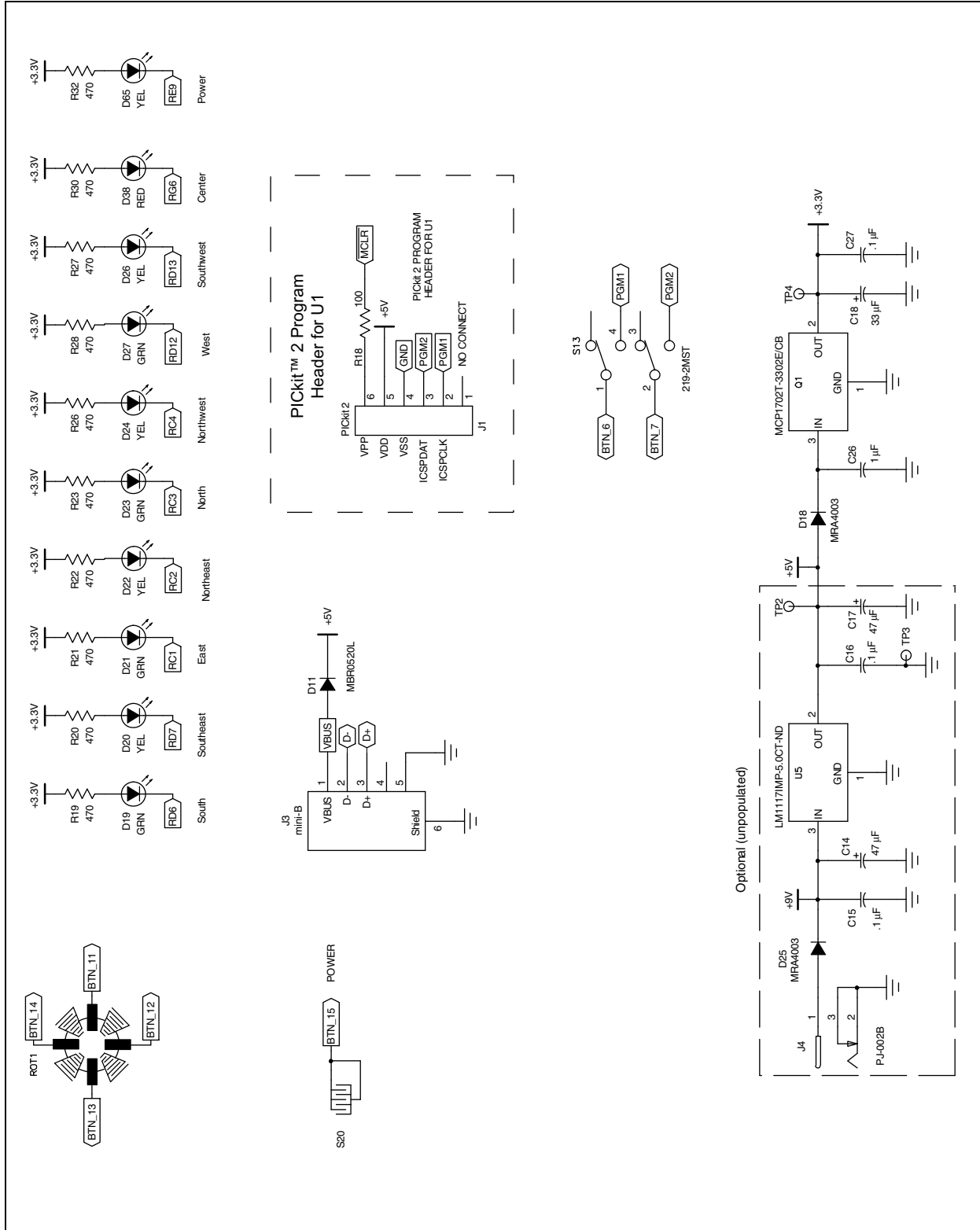


**FIGURE A-3: PICDEM™ TOUCH SENSE 2 DEMO BOARD SCHEMATIC (PAGE 2 OF 3)**



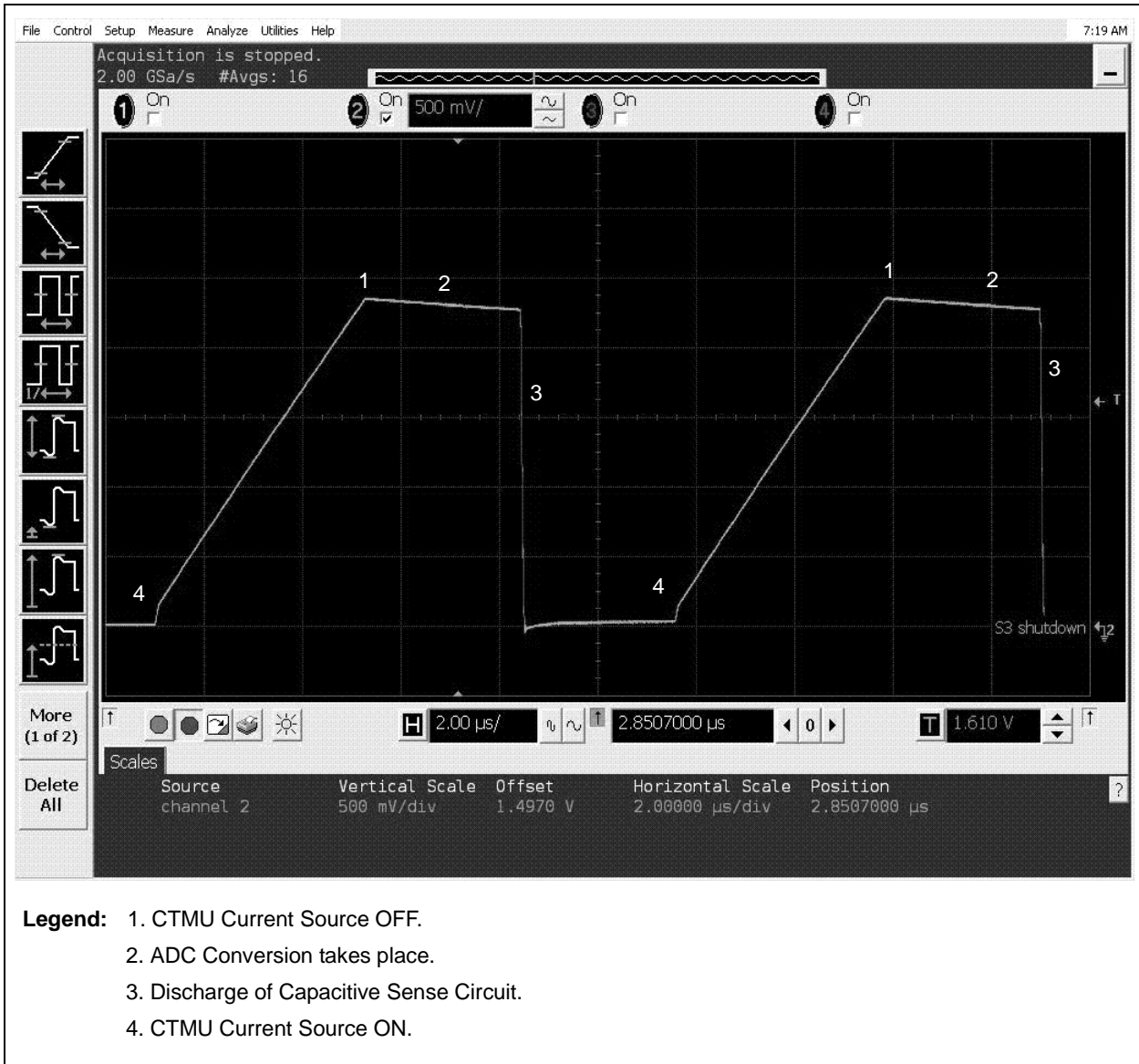
# AN1250

FIGURE A-4: PICDEM™ TOUCH SENSE 2 DEMO BOARD SCHEMATIC (PAGE 3 OF 3)



## APPENDIX B: OSCILLOSCOPE TRACE

**FIGURE B-1: OSCILLOSCOPE TRACE OF CHARGING OF CAP SENSOR CIRCUIT**



# AN1250

---

## CONCLUSION

Microchip's CTMU-based capacitive touch solutions provide a reliable solution for the implementation of capacitive touch and require minimal external hardware. The CTMU unit is fast and software needed for operation is easy to set up, configure and integrate into any application requiring capacitive touch.

## REFERENCES

- <http://www.microchip.com/mTouch>
- AN1103, "*Software Handling for Capacitive Sensing*"
- AN1102, "*Layout and Physical Guidelines for Capacitive Sensing*"
- "*PICDEM™ Touch Sense 2 Demo Board User's Guide*" (DS51748)
- "**Section 11. Charge Time Measurement Unit (CTMU)**" in the "*PIC24F Family Reference Manual*" (DS39724)

---

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, rfPIC, SmartShunt and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICtail, PIC<sup>32</sup> logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rLAB, Select Mode, Total Endurance, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

**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://support.microchip.com>  
Web Address:  
[www.microchip.com](http://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

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

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

**Kokomo**  
Kokomo, IN  
Tel: 765-864-8360  
Fax: 765-864-8387

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

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

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

### ASIA/PACIFIC

**India - Bangalore**  
Tel: 91-80-3090-4444  
Fax: 91-80-3090-4080

**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: 886-3-572-9526  
Fax: 886-3-572-6459

**Taiwan - Kaohsiung**  
Tel: 886-7-536-4818  
Fax: 886-7-536-4803

**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

## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for [Development Boards & Kits - PIC/DSPIC category](#):*

*Click to view products by [Microchip manufacturer](#):*

Other Similar products are found below :

[S-191](#) [TDGL025](#) [LSD4NBT-B208000001](#) [DV330021](#) [DM160230](#) [DM164141](#) [DM164142](#) [DM164143](#) [DM320010](#) [DM320105](#) [DM320106](#)  
[DM330028](#) [DV161001](#) [DM320008](#) [DM320008-C](#) [DM320010-C](#) [DM330026](#) [MIKROE-2653](#) [MIKROE-2644](#) [MIKROE-2657](#) [MIKROE-](#)  
[2647](#) [MIKROE-2654](#) [MIKROE-2648](#) [MIKROE-2788](#) [MIKROE-1907](#) [410-336](#) [SC70EV](#) [ECC577448EU](#) [ESP32-Audio-Kit](#) [AC103011](#)  
[AC243026](#) [AC323027](#) [ADM00333](#) [ARD00906](#) [DM160228](#) [DM163025-1](#) [DM163030](#) [DM164127-2](#) [DM164130-3](#) [DM164136](#) [DM164137](#)  
[DM164140](#) [DM180021](#) [DM182026](#) [DM183021](#) [DM240001](#) [DM240001-2](#) [DM240001-3](#) [DM240004](#) [DM240011](#)