# Introduction to Capacitive Sensing

Author: Tom Perme
Microchip Technology Inc.

## INTRODUCTION

Capacitive sensing is becoming more prevalent and in demand for consumer applications. This application note introduces one solution for capacitive sensing using several Microchip parts. The targeted parts of this application note are the PIC16F616 family, the PIC16F690 family and the PIC16F887 family, representing low-to-high pin count, 8-bit microcontrollers.

Several techniques for capacitive sensing are currently present in industry. Many are based on measuring a frequency or duty cycle which is changed by the introduction of additional capacitance from a person's finger to ground. Some other methods use charge balancing or rise and fall time measurements. This solution measures frequency using a free-running RC oscillator.

## A SHORT HISTORY

While capacitive sensing has been around for more than 50 years, it is becoming increasingly easier to implement and more popular. A classic example of a capacitive switch is the Touch Lamp. The Touch Lamp has been around for a long time, and it is a simple, capacitive switch that turns a light bulb on, off or dims it.

New technology allows much more sophisticated control of touch buttons. A key to this has been microcontrollers with mixed signal peripherals. They provide the ability to perform capacitive sensing, decision making, responsive actions and other duties pertinent to the system as well.

## CREATING A GOOD SENSOR

A quick background on the natural capacitance of a PCB pad, and the process that occurs when pressing a finger near a pad, will help clarify how to make a good sensor.

**FIGURE 1: CAPACITANCE ILLUSTRATIONS**

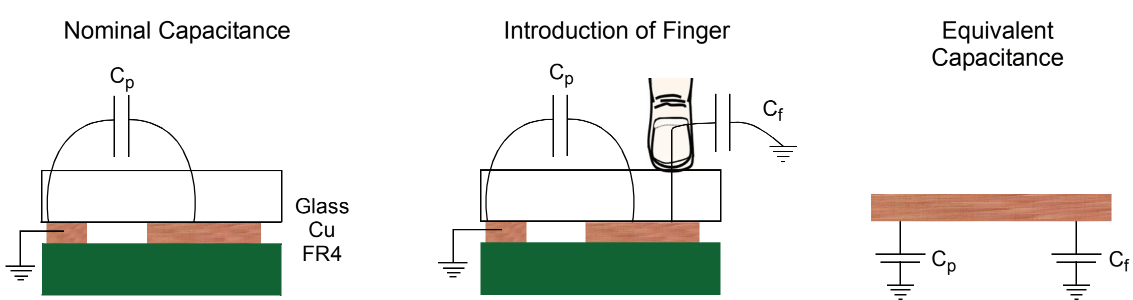
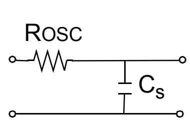



**FIGURE 2: SIMPLE RC CIRCUIT**




$\tau = R_{OSC}\, C_s \equiv$ Charge/Discharge Constant

$\tau = R_{OSC}\,(C_p + C_f)$

**TABLE 1: GLOSSARY OF TERMS**

| Acronym | Description |
|---|---|
| $\varepsilon_o$ | Permittivity of Free Space |
| $\varepsilon_r$ | Relative Dielectric Constant |
| d | Distance Between Capacitor Plates |
| A | Area of Plates |
| C | Capacitance |

**EQUATION 1: CAPACITANCE EQUATION**

$$C = \frac{\varepsilon_o \varepsilon_r A}{d}$$

A detectable press is due to the introduced capacitor from glass-finger-ground which is in parallel to the circuit's natural parasitic capacitance to ground. Capacitors in parallel are added so a finger approaching the pad increases the total capacitance. This change, as a percentage, is:

$\Delta C\% = ( (C_p + C_f) - C_p)/C_p = C_f/C_p$.

This change establishes the criteria we need to detect that a finger will introduce additive capacitance causing a shift in the RC time constant of our oscillator. Increasing the RC time constant will decrease the frequency of the oscillator, as shown later, and this is the change detectable in the microcontroller.

It is also good to note that a small $C_p$ is desirable because $C_f$ is known to be very small. A small $C_p$ results in a larger percentage change in capacitance and frequency. A finger press can vary anywhere between 5-15 pF as a guideline. Finger capacitance should not be assumed constant or a known absolute. Further detail about making $C_p$ small and other related factors are in the application note, *AN1102, "Layout and Physical Design Guidelines for Capacitive Sensing"*.

Now that we have an idea what to detect, we need an oscillator with a frequency that is dependent on our capacitor sensing plate, $C_s$. The circuit of Figure 3 will accomplish the objective. This design uses a relaxation oscillator to create a frequency dependent on the capacitor value. The resistor value of the RC oscillator is a designed parameter whose goal is to put the oscillating frequency in the 100-400 kHz range. The exact frequency is not important, but having a high frequency yields more counts in the measuring process and better resolution than a low frequency.

To detect a button press, first the system must be configured properly. Then, the key steps are as follows:

1. Oscillate signal through sensor capacitor.
2. Count positive edges using T1CKI.
3. At the end of a fixed measurement period, obtain the reading (frequency in counts).
4. Determine if current frequency is lower than the normal unpressed average value.

## HOW THE OSCILLATOR WORKS

The relaxation oscillator is a free-running RC oscillator using 2 comparators with an SR latch to change the charge direction of the sensing capacitor's voltage, up or down. It will charge and discharge the capacitor at a rate determined by the RC time constant, and it will charge between upper and lower limits set by the positive inputs to the comparators. The time required to charge from the lower limit to the upper limit and discharge back to the lower limit is the period of the oscillator.

The oscillator circuit is shown in Figure 3. The positive inputs of the comparators are the upper and lower charging limits. C1+ is internal, but C2+ must be supplied externally to set the lower limit. The 1000 pF capacitor is in place to reject high-frequency noise from the power supply and ensure a stable lower limit. Voltage, $V_-$, will charge and discharge between these limits, and is driven by logic level signals at C2OUT. Comparator 2's output, C2OUT, is configured for $\overline{Q}$ in order to get the appropriate charging and discharging behavior. Feedback resistor, R, forms the RC with the sensor plate denoted as $C_s$.

When the voltage, $V_-$, on capacitor, $C_s$, is below the lower limit, C2OUT goes high, and the system will begin charging. In between the limits, the system will retain the last state (charging or discharging). When $V_-$ is above the upper limit, C2OUT goes low, and the system will begin discharging and continue to discharge through the middle region.

An illustration of the charge and discharge cycling is shown in Figure 4. The output, $\overline{Q}$, and the charging or discharging state it represents is determined by the relative values of the negative input to the positive input of each comparator and the SR latch.

**FIGURE 3: OSCILLATOR CIRCUIT**

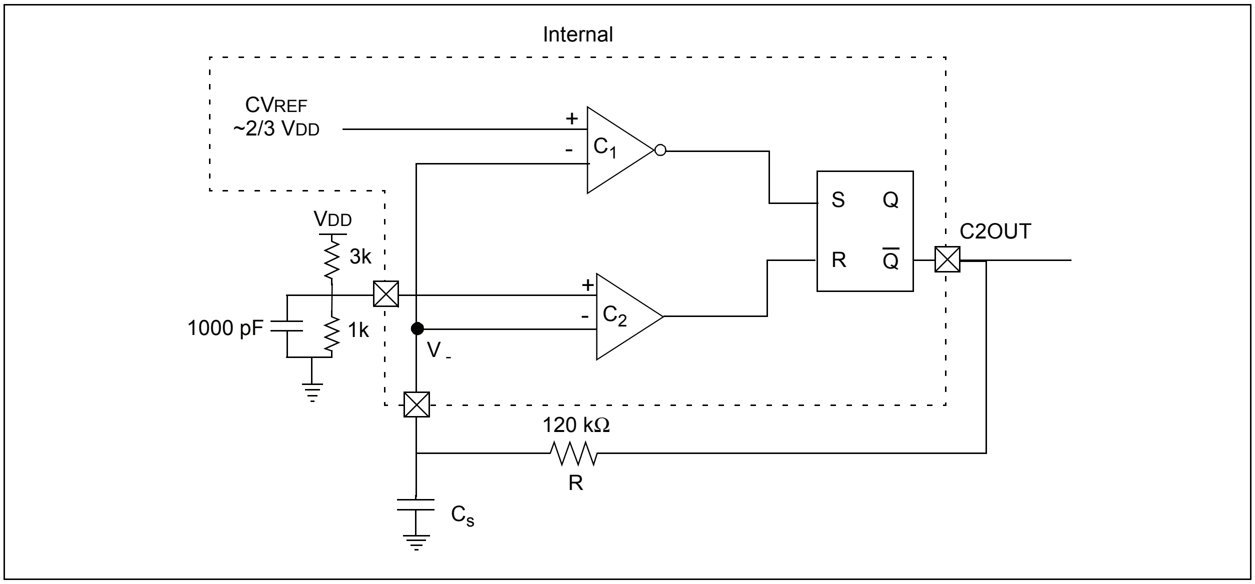



**FIGURE 4: CHARGE AND DISCHARGE CYCLES**

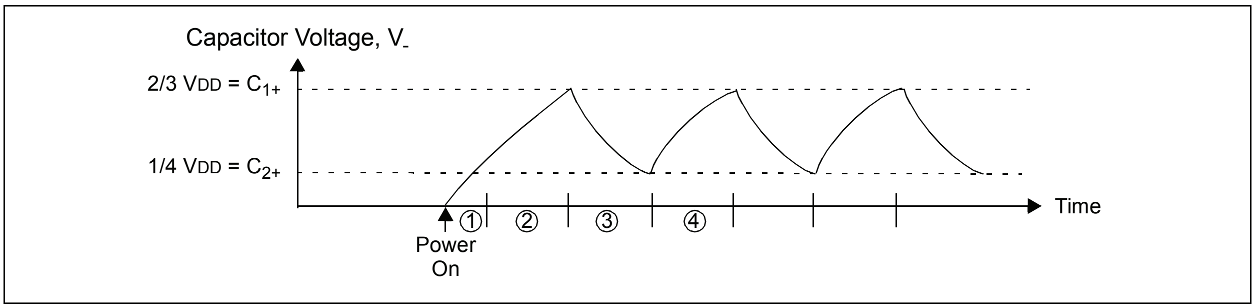



**TABLE 2: REGIONS OF OPERATION**

| Time | Set Bit | Reset Bit | C2OUT Result | Action |
|---|---|---|---|---|
| **1** | $C1+ > V_-$<br>S = 0 | $C2+ > V_-$<br>R = 1 | $\Rightarrow \overline{Q} = 1$ | "Begin Charging" |
| **2** | $C1+ > V_-$<br>S = 0 | $C2+ < V_-$<br>R = 0 | $\Rightarrow \overline{Q} = \overline{Q}_{n-1}$ | "Last State (Charging)" |
| **2 ⇒ 3** | $C1+ < V_-$<br>S = 1 | $C2+ < V_-$<br>R = 0 | $\Rightarrow \overline{Q} = 0$ | "Begin Discharging" |
| **3** | $C1+ > V_-$<br>S = 0 | $C2+ > V_-$<br>R = 0 | $\Rightarrow \overline{Q} = \overline{Q}_{n-1}$ | "Last State (Discharging)" |
| **3 ⇒ 4** | $C1+ > V_-$<br>S = 0 | $C2+ > V_-$<br>R = 1 | $\Rightarrow \overline{Q} = 1$ | "Begin Charging" |
| **4 = 2** | Charge cycle begins to repeat as region two. | | | |

## MEASURING FREQUENCY

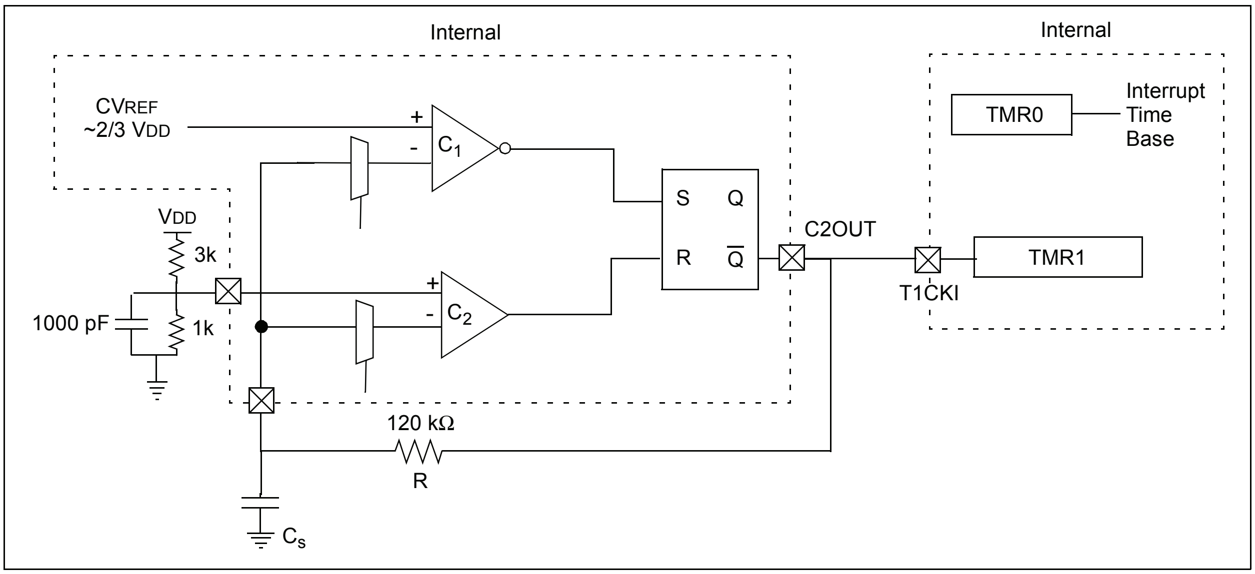
Once the oscillator is constructed, its frequency must be monitored to detect the drop in frequency caused by a finger press. Figure 5 shows a more complete schematic where C2OUT not only drives the oscillator, but also is fed into the clock input of Timer1, T1CKI. Each time C2OUT changes from '0' to '1' Timer1 will increment. Unhindered, Timer1 will increment non-stop and eventually roll over, but this is not useful for capacitive sensing.

To make it useful, a fixed time base is used to measure the frequency over a defined period. Timer0 provides this fixed period time base. At the start of a measurement, Timer0 is cleared, and it will count up to 255 and then it will overflow. On overflow, the Timer0 interrupt, T0IF, causes the program to vector to the Interrupt Service Routine. The value of TMR1 is then read and compared to previous readings. This constitutes a single scan of a button. If the current value of TMR1 is significantly lower, the capacitance has increased, the frequency has dropped and a button press is detected.

With surrounding logic, the new value may be averaged into a running average which is the base value for comparison. At the end of the Interrupt Service Routine, once all tasks determining a button press and setting appropriate flags are finished, both Timer1 and Timer0 are cleared and restarted for the next reading.

| | |
|---|---|
| **Note:** | The prescaler on Timer0 must be set such that Timer0 overflows before Timer1 can overflow. |

**FIGURE 5: MEASURING OSCILLATOR FREQUENCY**



## DETECTING A BUTTON PRESS

At this point, the system is complete except for the detection and signaling of a button press. The remaining portion is best handled in the Interrupt Service Routine on the regular calls by T0IF at the end of each scan.

A simple way to watch for the decrease is to use three unsigned integer variables. The three variables are:

```
unsigned int average;
unsigned int raw;
unsigned int trip;
```

The variable average holds a running average of the previous 16 samples; raw is the current sensor data read from Timer1, and trip is a number which specifies the distance below the average for a button press. The simplest button press algorithm would be to test if the raw is a fixed distance below the average as in the code example below.

### EXAMPLE 1: SIMPLE BUTTON DETECTION

```
ISR {
 ...

 if (raw < (average - trip))
        // Button Pressed

 else
        // Button Not Pressed


 // Perform average of current reading
 average = average*15/16 + raw/16;

 ...
}
```

To provide an illustrative example, assume the oscillator reads 10,000 without a finger pressing the button. The average and raw will both be 10,000. As the designer, assume a trip value of 1000 is a good value. When someone presses the button, the raw value immediately drops to 8500, but the average was still at 10,000. The “if statement” in Example 1 will prove to be true, because 8500 is less than 9000. The button is pressed. Then, a flag may be set or a response performed in reaction.

| **Note:** | There are better software algorithms for detecting button presses, and these are discussed in *AN1103, “Software Handling for Capacitive Sensing”*. The example above is very simplistic to demonstrate the frequency drop as the fundamental change common to all. |
|---|---|

## CONFIGURING YOUR PIC® MCU

There are currently three families of PIC microcontrollers able to use the method as shown. These are the Microchip PIC16F616 family, PIC16F690 family and PIC16F887 family. The basic settings of registers used for capacitive sensing are the same, although differences between families may cause slightly different values in each. **Appendix A: “Register Settings for the PIC16F887 Family”** shows a detailed depiction of the proper settings for the PIC16F887 family. So, **Appendix A: “Register Settings for the PIC16F887 Family”** may be used as a guideline for which bits to set in the other families of parts.

The registers that require setting for capacitive sensing are:

- CM1CON0
- CM2CON0
- CM2CON1
- SRCON (or equivalent SRCON0)
- VRCON
- ANSEL (and ANSELH if appropriate)
- TRISx (for all inputs and outputs)

The settings for these registers are found in **Appendix A: “Register Settings for the PIC16F887 Family”**.

## CONCLUSIONS

Capacitive sensing is not magic. It just takes some time and understanding of basic electrical principles. Additional application notes describe other practices for creating a successful product. Particularly important application notes are *AN1103 “Software Handling for Capacitive Sensing”*, *AN1104, “Capacitive Multibutton Configurations”*, and *AN1102, Layout And Physical Design Guidelines for Capacitive Sensing”*.

## APPENDIX A: REGISTER SETTINGS FOR THE PIC16F887 FAMILY

The following registers can be found in the *"PIC16F882/883/884/886/887 Data Sheet"* (DS41291). Detailed explanations of each bit may be found in a part's Data Sheet. These register settings provide a guideline for setting the other families' registers. TRIS registers must be set properly for inputs and outputs.

**REGISTER 8-1: CM1CON0: COMPARATOR C1 CONTROL REGISTER 0**

| R/W-0 | R-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| C1ON | C1OUT | C1OE | C1POL | — | C1R | C1CH1 | C1CH0 |
| bit 7 | | | | | | | bit 0 |
| 1 | 0 | 0 | 1 | — | 1 | 0 | 0 |

**REGISTER 8-2: CM2CON0: COMPARATOR C2 CONTROL REGISTER 0**

| R/W-0 | R-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| C2ON | C2OUT | C2OE | C2POL | — | C2R | C2CH1 | C2CH0 |
| bit 7 | | | | | | | bit 0 |
| 1 | 0 | 1 | 0 | — | 0 | 0 | 0 |

**REGISTER 8-3: CM2CON1: COMPARATOR C2 CONTROL REGISTER 1**

| R-0 | R-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-1 | R/W-0 |
|---|---|---|---|---|---|---|---|
| MC1OUT | MC2OUT | C1RSEL | C2RSEL | — | — | T1GSS | C2SYNC |
| bit 7 | | | | | | | bit 0 |
| 0 | 0 | 1 | 1 | — | — | 1 | 0 |

**REGISTER 8-4: SRCON: SR LATCH CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/S-0 | R/S-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| SR1[(2)] | SR0[(2)] | C1SEN | C2REN | PULSS | PULSR | — | FVREN |
| bit 7 | | | | | | | bit 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | — | 0 |

**2:** To enable an SR latch output to the pin, the appropriate CxOE and TRIS bits must be properly configured.

**REGISTER 8-5: VRCON: VOLTAGE REFERENCE 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 |
|---|---|---|---|---|---|---|---|
| VREN | VROE | VRR | VRSS | VR3 | VR2 | VR1 | VR0 |
| bit 7 | | | | | | | bit 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

**REGISTER 3-3: ANSEL: ANALOG SELECT REGISTER**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| ANS7[(2)] | ANS6[(2)] | ANS5[(2)] | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 |
| bit 7 | | | | | | | bit 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

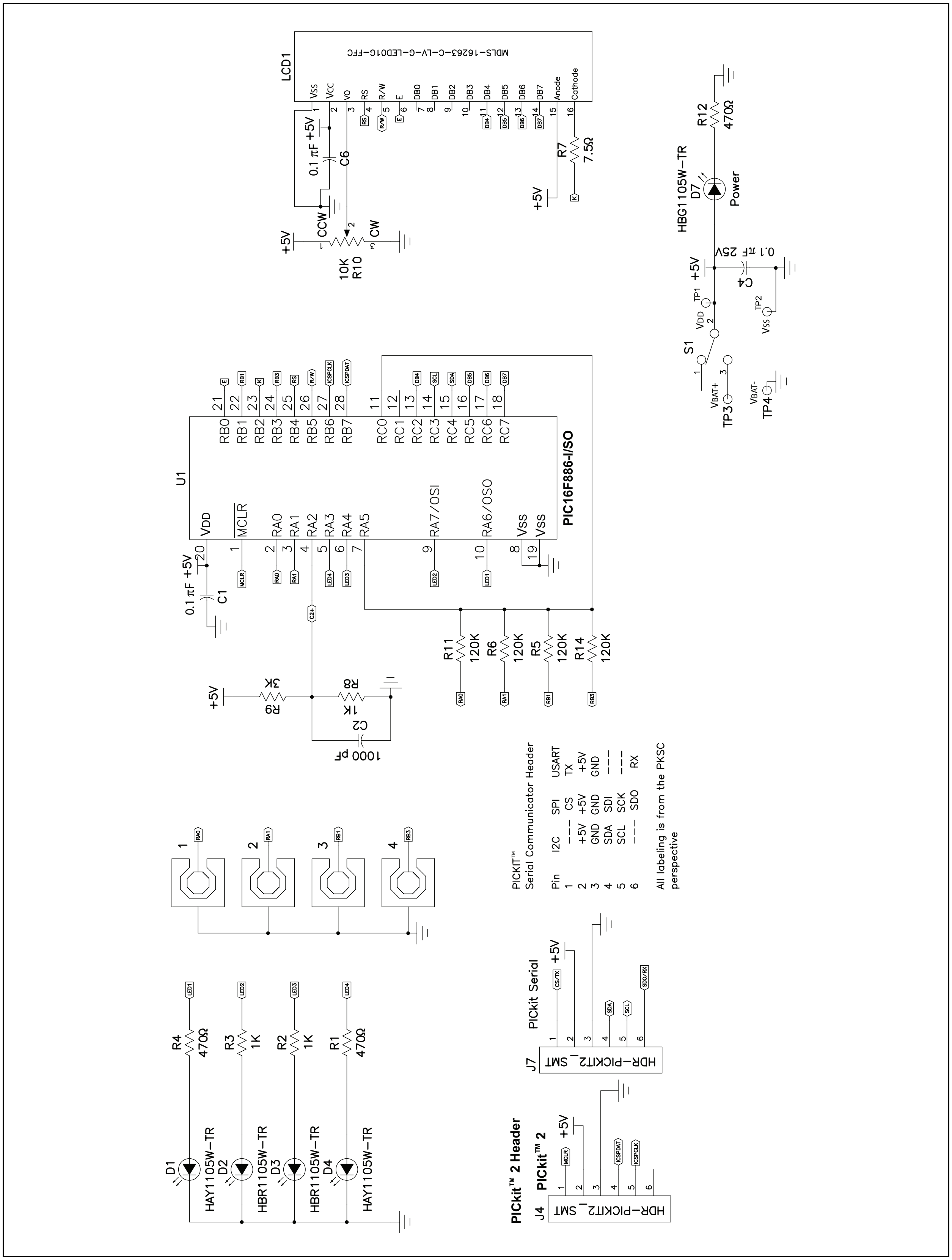
**2:** Not implemented on PIC16F883/886.

**REGISTER 3-4: ANSELH: ANALOG SELECT HIGH REGISTER**

| U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | ANS13 | ANS12 | ANS11 | ANS10 | ANS9 | ANS8 |
| bit 7 | | | | | | | bit 0 |
| — | — | 0 | 0 | 0 | 1 | 1 | 0 |

## APPENDIX B: SCHEMATIC

### FIGURE B-1: TOUCH SENSING DEMO SCHEMATIC



**NOTES:**

**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.



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

**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 - Fuzhou**
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

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

**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 - Shunde**
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

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

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

## ASIA/PACIFIC

**India - Bangalore**
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

**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 - Penang**
Tel: 60-4-646-8870
Fax: 60-4-646-5086

**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


06/25/07