

基本MSP430的电容式 触摸感应实现

王鲁克
市场应用工程师
利尔达科技有限公司

议程

- 触摸感应应用概述
- 系统级设计考虑
- MSP430实现方式
- 按键、划条、演示板和应用案例
- 总结

触摸感应

- 与机械按键相比
 - 低成本
 - 使用寿命较长
- 提供多样化的实现形式
 - 简单的按钮
 - 定位滑动按键
- 调整灵活
- 应用...
 - 消费性电子有限公司
 - 家电
 - 居家自动化
- ...只要有按键就可能应用



触摸感应应用概述

- 各种实现原理
 - 电阻、电容、光学、压力,...
- 各种原理实现方式不一
- 光学
 - 昂贵
 - 复杂的系统设计
- 电阻
 - 需要当触摸时电阻改变的材料
 - BOM的成本相对较低，但也同样增加了BOM成本
- 电容
 - 可以直接在PCB上实现
 - 按键的大小和形状可灵活设计
 - 只需一个额外功能的PCB和一些元器件

电容的应用

- 电荷转移技术
 - Quantum公司专利的解决方案
 - 通过测量感应电容间的电荷量转移来实现
 - 需要激励信号和测量单元
- 通过**ADC**测量电容
 - 激励信号对电容传感元件的影响，用ADC来测量产生的电压
 - ADI通过16位的Sigma-Delta ADC来实现C-数字的转换
- 松弛振荡器
 - 通过感应电容来影响弛张振荡器的频率
- **RC 充电/放电**
 - 使用高频率的时钟，来计数RC充放电时间

MSP430 电容测量

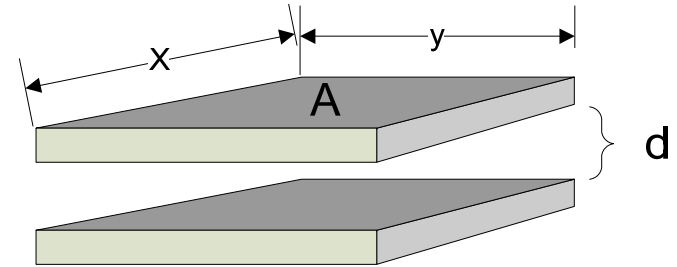
- 电容的变化是因为身体、手指或者其他导体靠近
- 方法 1:
 - 通过设计松弛振荡的振荡频率取决于感应电容
 - 通过测量振荡频率的改变来测量电容改变
- 方法 2:
 - 测量 R-C 充电/放电过程中，当R是常数时，触摸动作使电容值发生变化，从而使充放电时间发生变化

议程

- 触摸感应应用概述
- 系统级设计考虑
- MSP430实现方式
- 按键、划条、演示板和应用案例
- 总结

电容构成

- **PCB材料构成基本电容**
- 通过改变寄生电容来改变电容
 - 手指触摸 (或者其他导体)
- 尽可能小的基本电容
 - 尽量小的寄生电容
 - 限制感应点的大小
- 使电容感应影响最大化
 - 使手指和传感电容面积匹配, 使 ΔC 相对最大化
 - 传感器和手指的距离最小
- 灵敏度

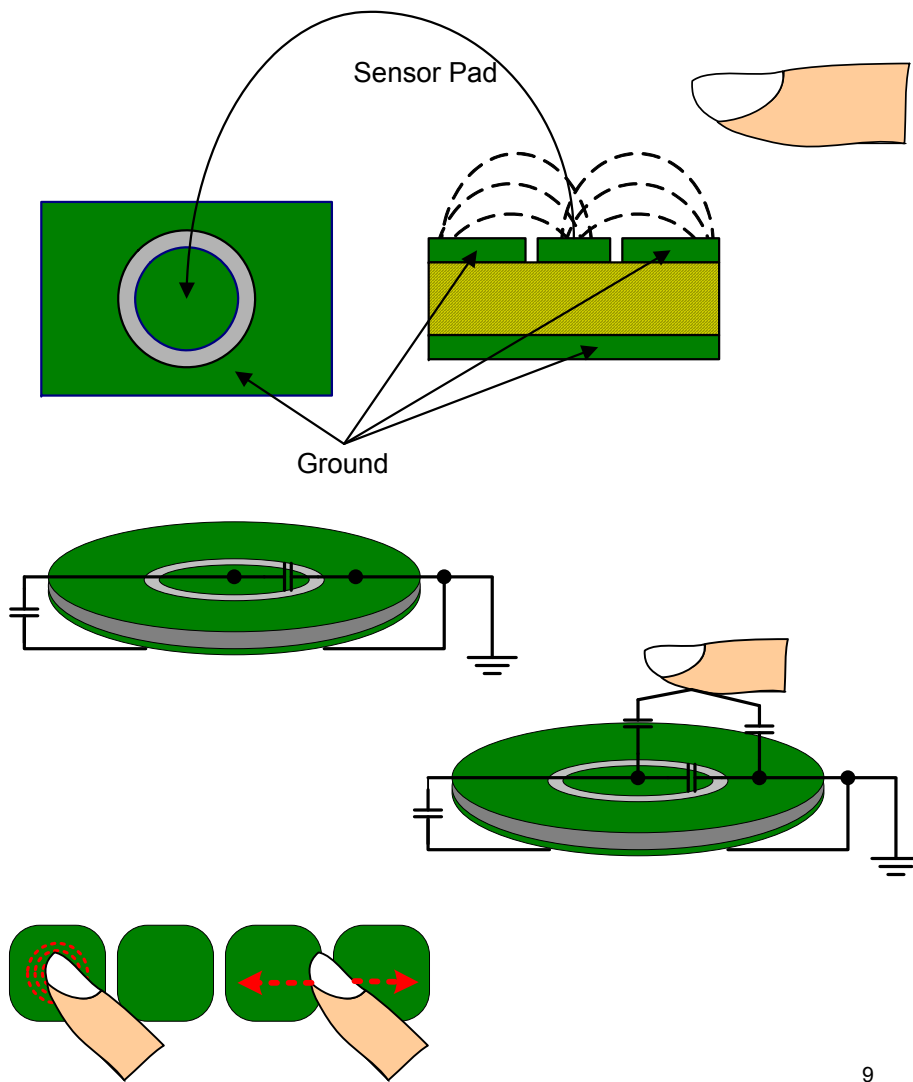


$$C = \frac{\epsilon_0 \epsilon_r A}{d}$$

<i>Material</i>	<i>Dielectric Constant (ϵ_r)</i>
Vacuum	1 (by definition)
Air	1.00054
Polyethylene	2.25
Paper	3.5
Pyrex glass	4.7
Rubber	7
Silicon	11.68

电容PCB传感器

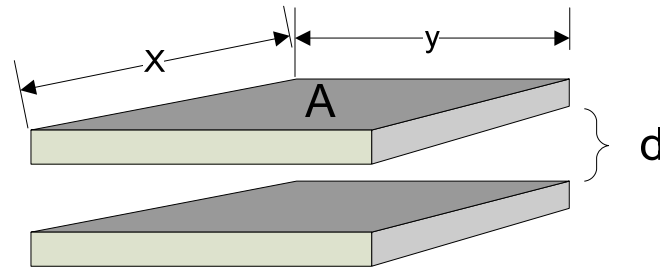
- **PCB**上的铜泊是一个良好的电容感应元件
- 传感器与传感器或邻近元件的距离为**10 - 20mil**
- 铜泊形状和大小尽量与手指匹配
- 通过感应铜泊的排列，可以做成滑条式感应
- 通过合理的布置感应铜泊，来实滑条感应定位



PCB 厚度

- 资料和厚度的事项
 - 目标 1: 小的基本电容
 - 目标 2: 稳定的基本电容

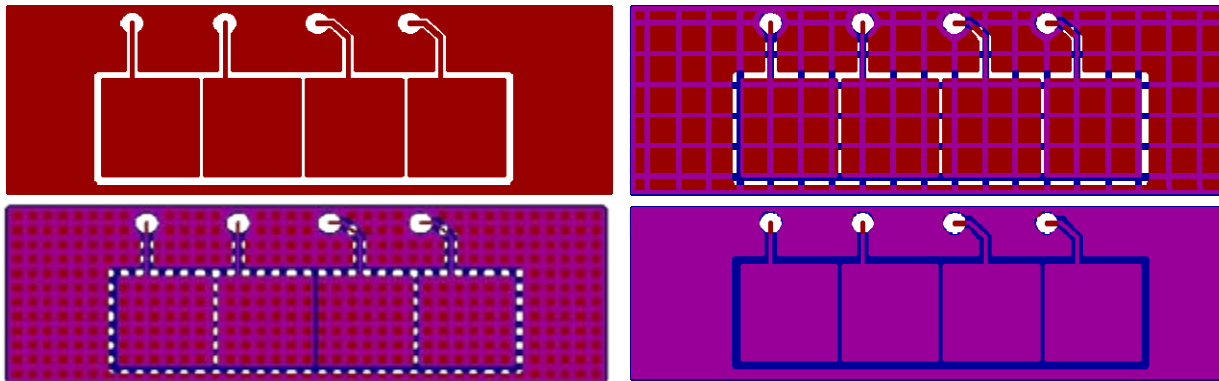
$$C = \frac{\epsilon_0 \epsilon_r A}{d}$$



- **d** 下降, 基本电容增加
- 对于一个给定的感应器大小和绝缘厚度, 电容变化量安全取决于手触摸感应动作
- 这种变化在由两极板间距离**d**引起的因素, 只占很小一部分
- 薄的**PCB**板需要更加注意对绝缘材料和绝缘层厚度的选择

布板和接地

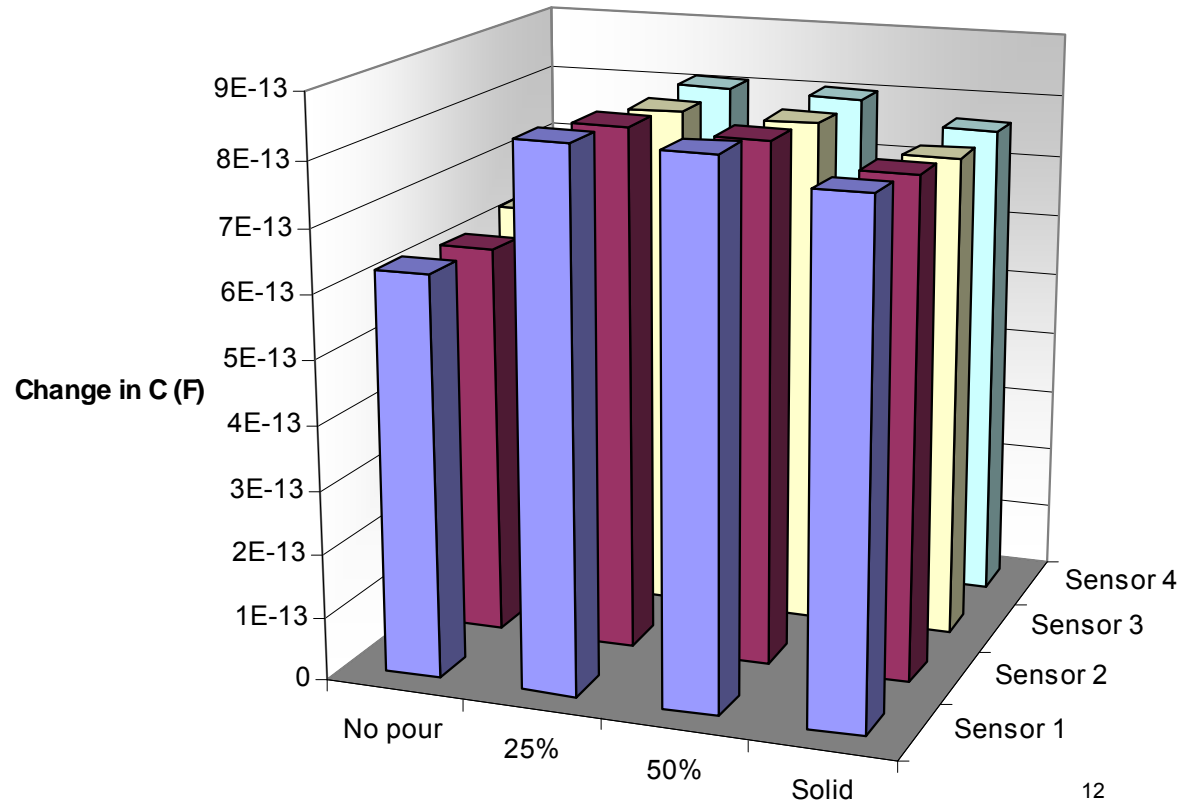
- 感应电容布板与对面铺地使噪声最小化
- 将感应电容对面的地铺成网格
 - 实地可以提高噪声抑制，但同时也提高了基本电容
 - 而对面不铺地，又会造成对噪声抑制能力降低
 - 所以50%铺地，相当比较合适



传感器和铺地影响

- 权衡**PCB**感应电容对面铺地百分比与灵敏度
- 不铺地
 - 低的基本电容
 - 低的电容变化
- **25-75%**
 - 增加基本电容
 - 增加电容变化量
- **Solid Pour**
 - 大量的基本电容
 - 相对电容变化量较小

Delta C vs. Pour
(8x8mm sensor on 1.5mm FR4)



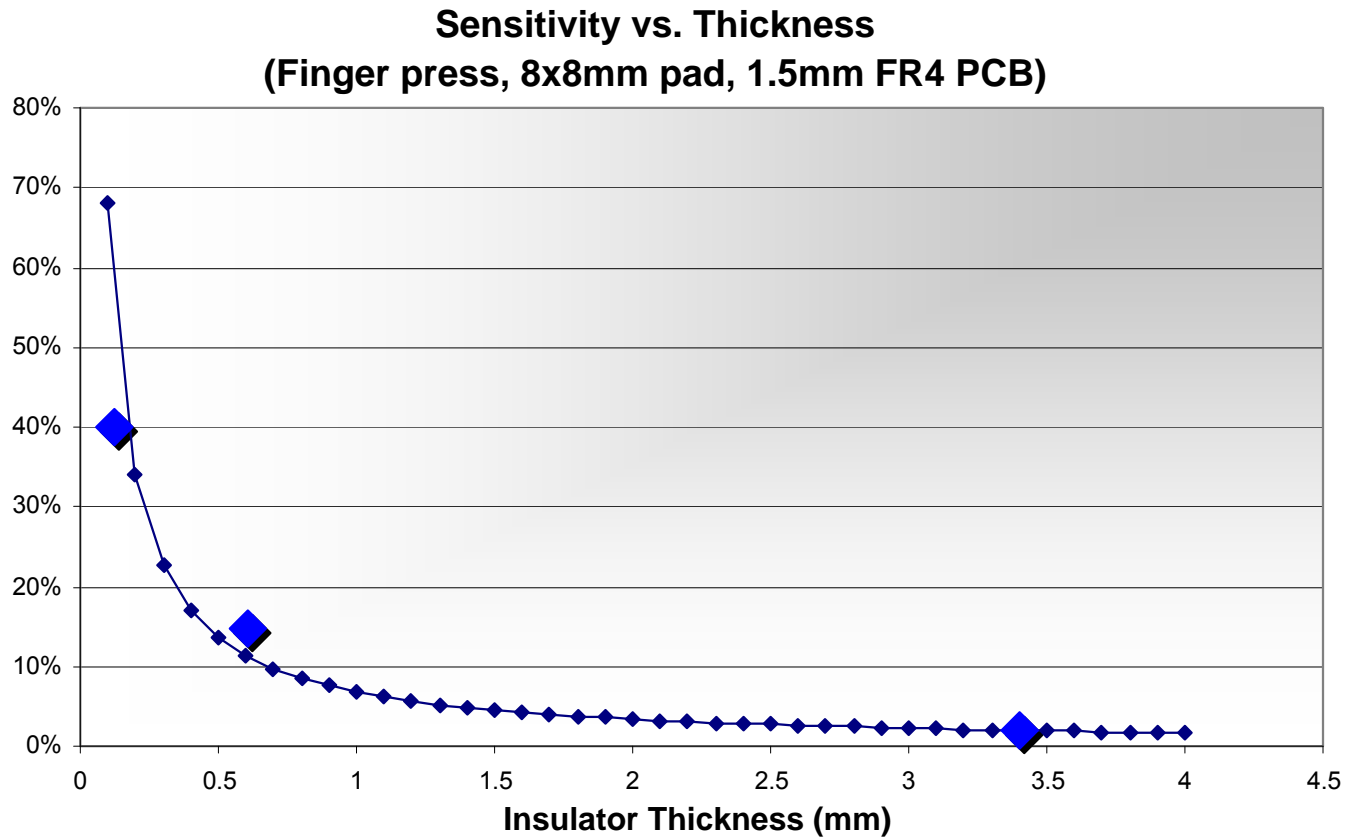
TI Proprietary information

绝缘层与安装

- **PCB**和用户之间往往需要一个绝缘体
- 绝缘材料必须是非导体
- 薄的比较好
 - 电容变化量与导体之间的距离成反比
- **PCB**上的传感器和绝缘体之间尽量不能有空气
- 使用粘合剂，以确保传感器和绝缘体稳定粘合

绝缘体厚度

- 绝缘体的厚度是于感应灵敏度成反比关系

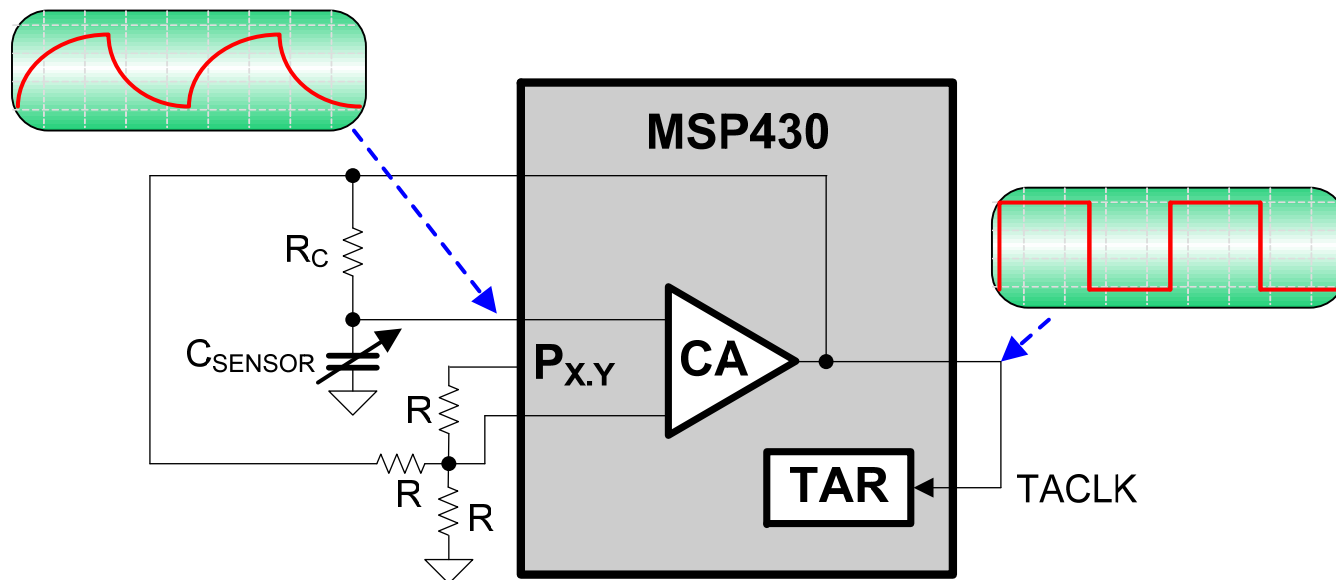


议程

- 触摸感应应用概述
- 系统级设计考虑
- **MSP430实现方式**
- 按键、划条、演示板和应用案例
- 总结

松驰振荡系统概述

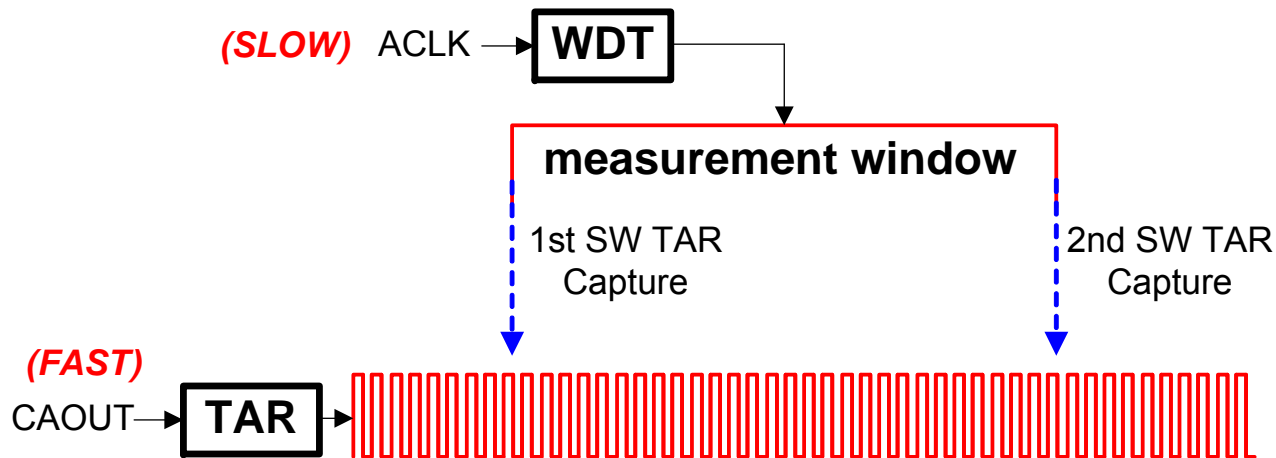
- 通过**MSP430**内部比较器和外部感应电容组成振荡电路
- 通过外部参考电阻设定充放电电压($1/3 V_{CC}$ & $2/3 V_{CC}$)
- 振荡频率= $1/[1.386 \times R_c \times C_{\text{sensor}}]$
- $\Delta C \Rightarrow \Delta f$



松弛振荡频率测量

- 用低频时钟设定测量时间窗口
- 较高的松弛振荡频率通过**MSP430**内部计数器（**Timer_A**）进步计数
- 通过**CPU**主频来计算出中断响应时间来消除捕获延时

$$ACLK < RO \text{ Freq} < CPU \text{ MCLK}$$



测量值的计算

- 测量计数值随着测量时间的增加而增加
- 如: **ACLK = VLO/64,**
SMCLK = DCO_cal/32768
 - (100K R ~ 625kHz f_{RO})

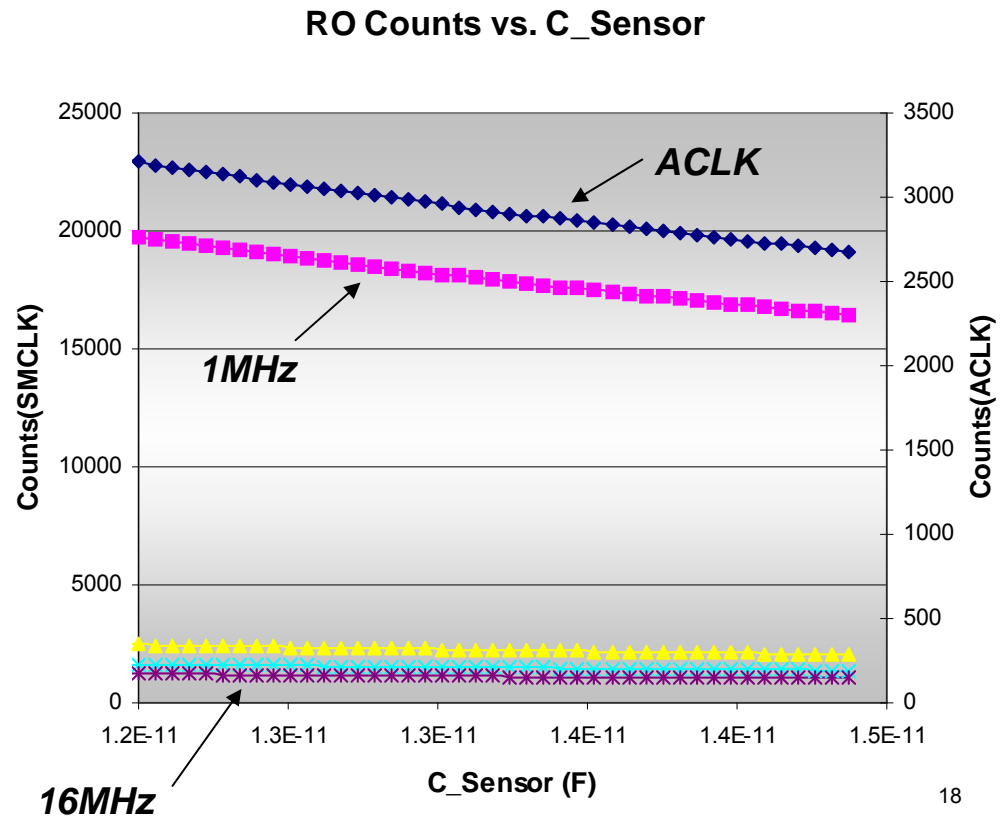
$$f_{RO} = \frac{1}{1.386 \times R \times C}, t_{RO} = \frac{1}{f_{RO}}$$

$$t_{window} = \frac{1}{f_{ACLK} / DIV_{ACLK} / DIV_{WDT}}$$

or...

$$t_{window} = \frac{1}{f_{DCO} / DIV_{SMCLK} / DIV_{WDT}}$$

$$counts = \frac{t_{window}}{t_{RO}}$$

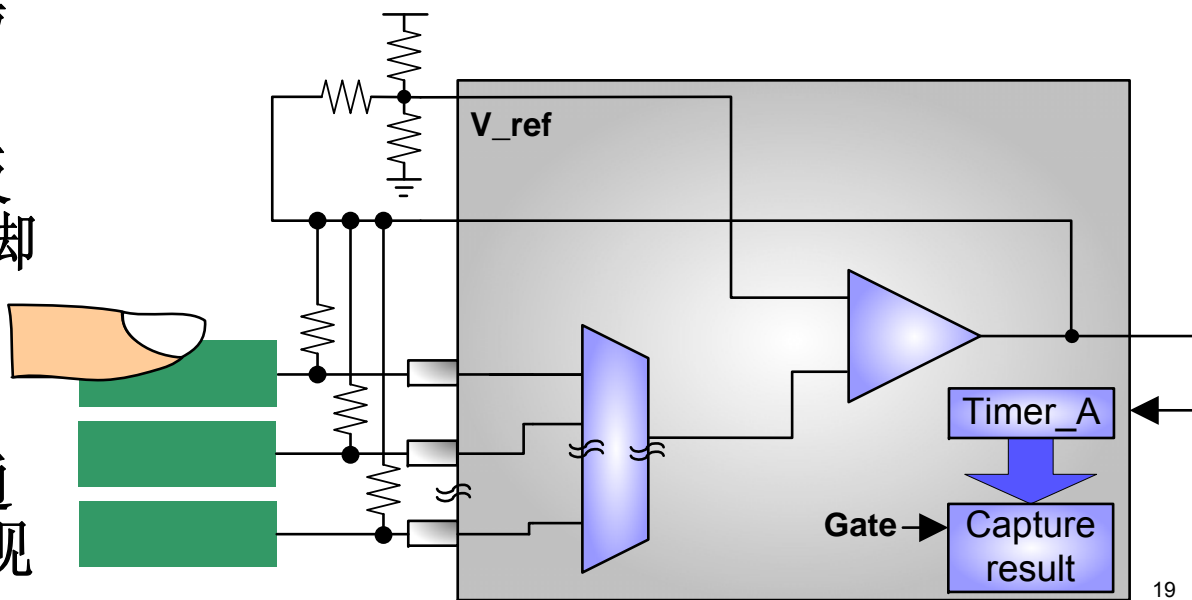
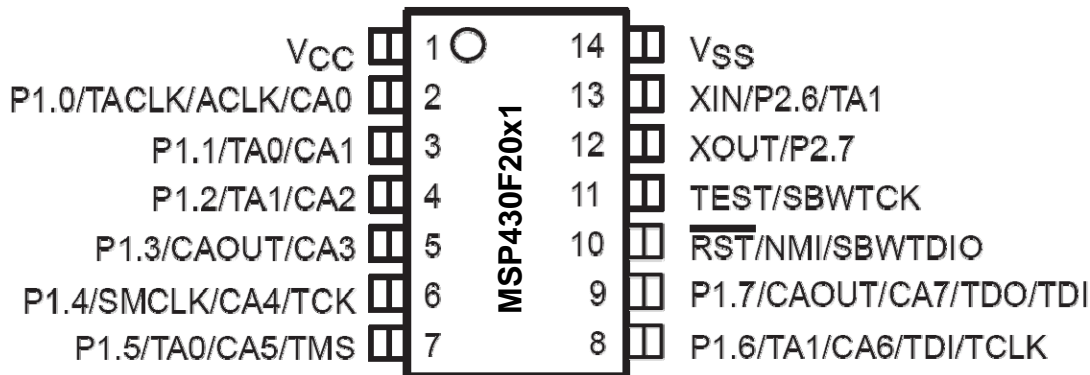


18

TI Proprietary information

完善的松弛振荡系统

- 而比较器（**Comp_A+**）带多输入口来实现多个感应点的测量
- 每个感应点需要一个电阻，另外整个系统还要三个参考电阻实现反馈
- 并将比较器输出反馈点与**TACLK**引脚连接，作为**Timer_A**时钟源
- 参考电阻电源可通过**I/O**脚控制来实现低功耗



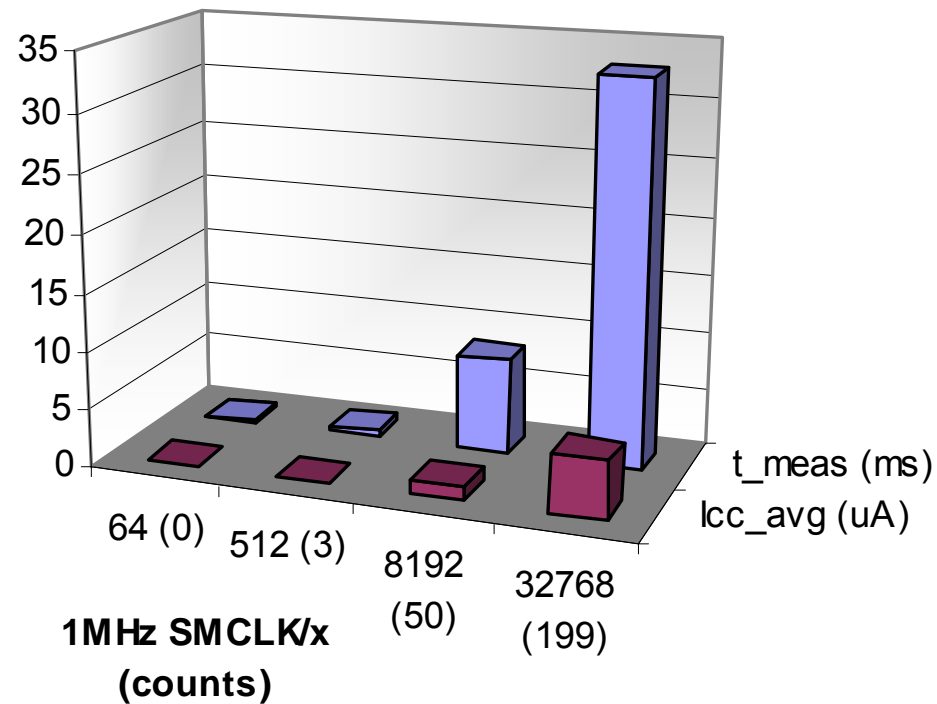
19

TI Proprietary information

松弛振荡电流消耗

- 测量窗口大意味着计数多
- 同时意味着更高的平均功耗
 - DCO: ~85uA @ 1MHz
 - Comp_A+: ~45uA
 - CA Vref: $V_{cc}/(1.5R)$ (for 100k R's, ~20uA)
- 在具体应用中要合理设定测量时间
 - 要减少测量时间，通过提高电容C的变化，
 - 从而可达到测量时间短，但又能稳定精确地测量出电容变化量

Current & Measurement Time vs. Measurement Window (1% C_delta)

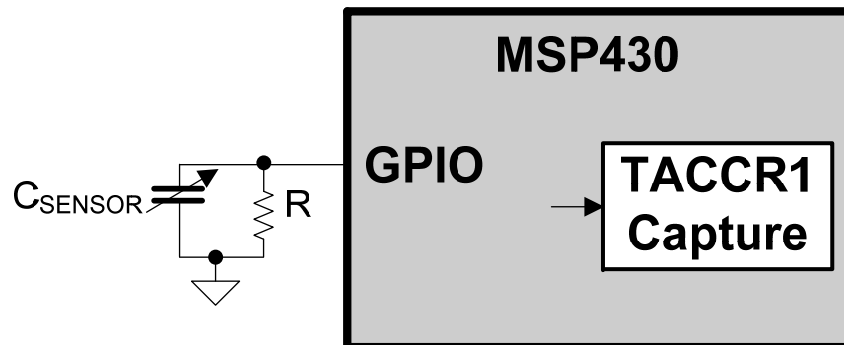


松弛振荡系统分析

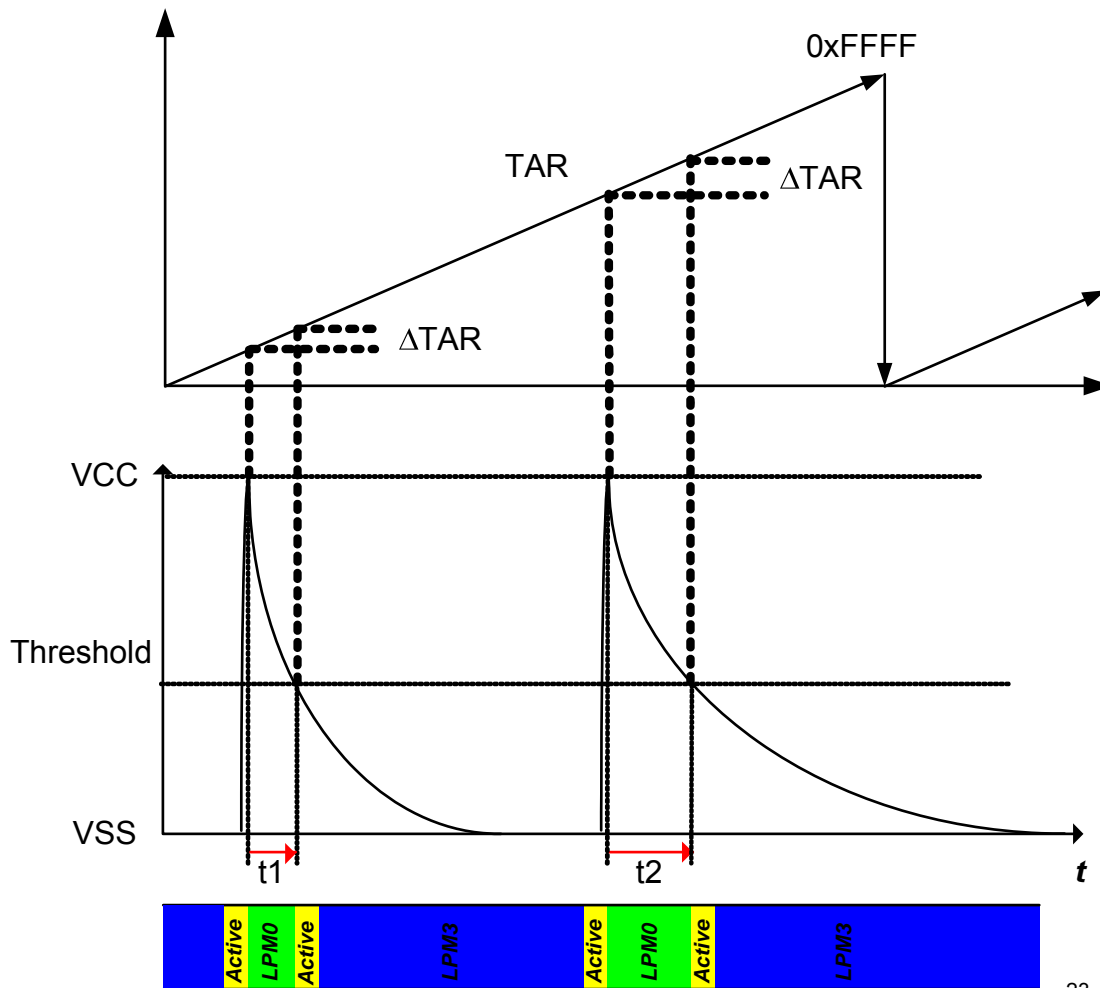
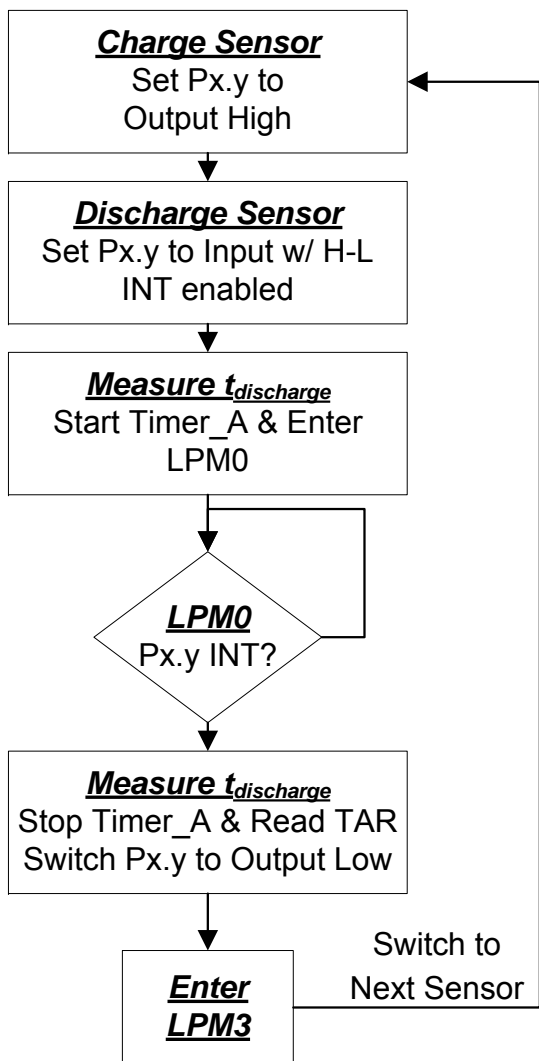
- 需要带多输入的带**Comp_A+**的**MSP430**来实现多个键的测量
- 按键受比较器输入端数量的限制
- 需要三个参考电阻分压作比较器参考
- 另外还需将比较器输出端**CAOUT**连接到**TACLK**
- 很好地抑制了电源波动对测量频率的影响
- 根据需要可编程设定测量时间
- 无需高速时钟就可实现
- 测量窗口确定后，时间受电源和温度(**VLO & DCO**)的影响

RC充放电系统概述

- **RC**放电时间通过**I/O**中断来捕获
- **MSP430 P1**和**P2**端口来实现
- 通过**P1**或**P2**对电充进行充电，然后通过中断方式来测量放电时间
 - GPIO = Output high (charge C)
 - GPIO = Input (discharge C)
 - GPIO INT on low threshold
- 通过**Timer_A**来计数放电时间



RC充放电测量



测量值关系

- 在相同的电容值，测量计数值与定时器时钟源成正比
- 如：**ACLK = VLO**；**SMCLK = DCO_cal**
 - 5.1Mohm R

$$V(t_{rc}) = V_{cc} \times e^{-\frac{t}{RC}}, V(t_{rc}) = V_{cc} \times [1 - e^{-\frac{t}{RC}}]$$

$$V_{IT-} = V_{cc} \times e^{-\frac{t_{discharge}}{RC}}, V_{IT+} = V_{cc} \times [1 - e^{-\frac{t_{charge}}{RC}}]$$

$$V_{IT-} = 0.4 \times V_{cc}, V_{IT+} = 0.6 \times V_{cc}$$

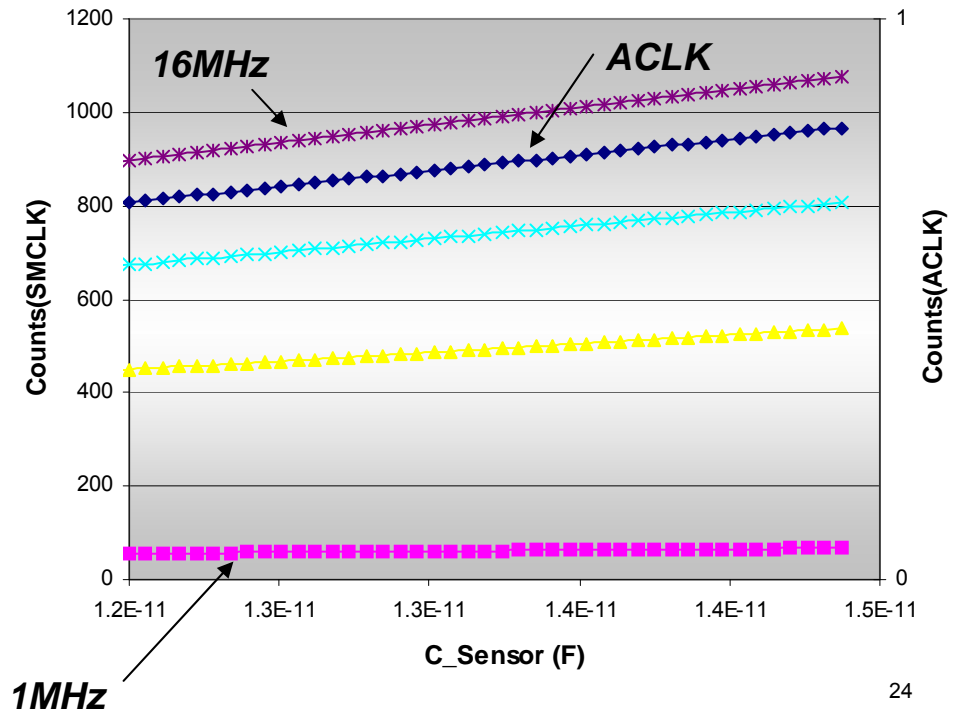
$$t_{discharge} = -RC \times \ln(0.4), t_{charge} = -RC \times \ln(1 - 0.6)$$

$$t_{CLK} = \frac{1}{f_{DCO} / DIV_{SMCLK}}$$

$$counts_{discharge} = \frac{t_{discharge}}{t_{CLK}}, counts_{charge} = \frac{t_{charge}}{t_{CLK}}$$

$$counts_{avg} = \frac{counts_{discharge} + counts_{discharge}}{2}$$

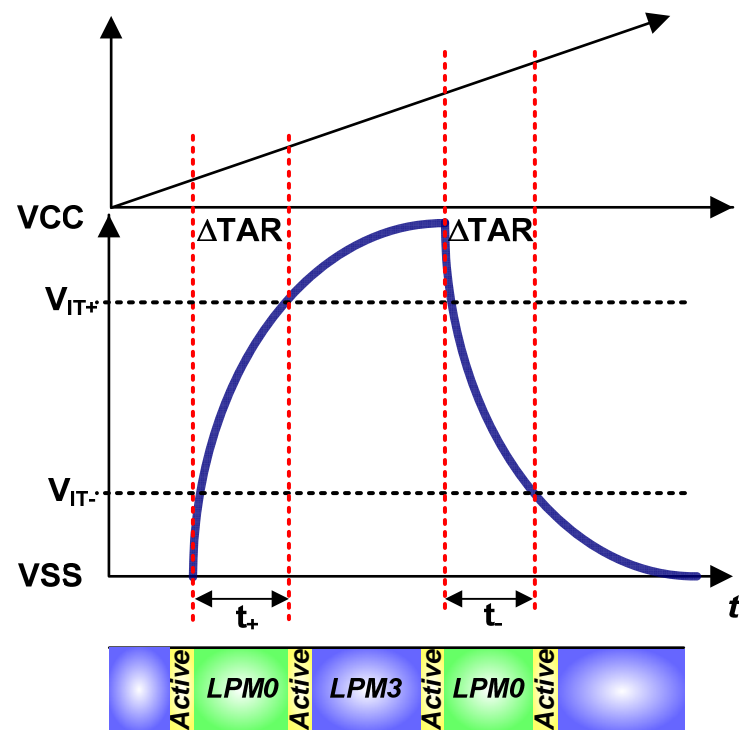
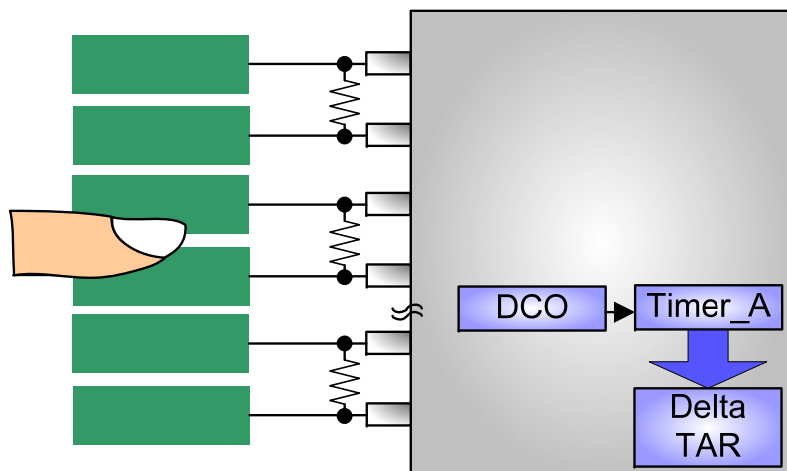
RC Counts vs. C_Sensor



TI Proprietary information

RC方式优点

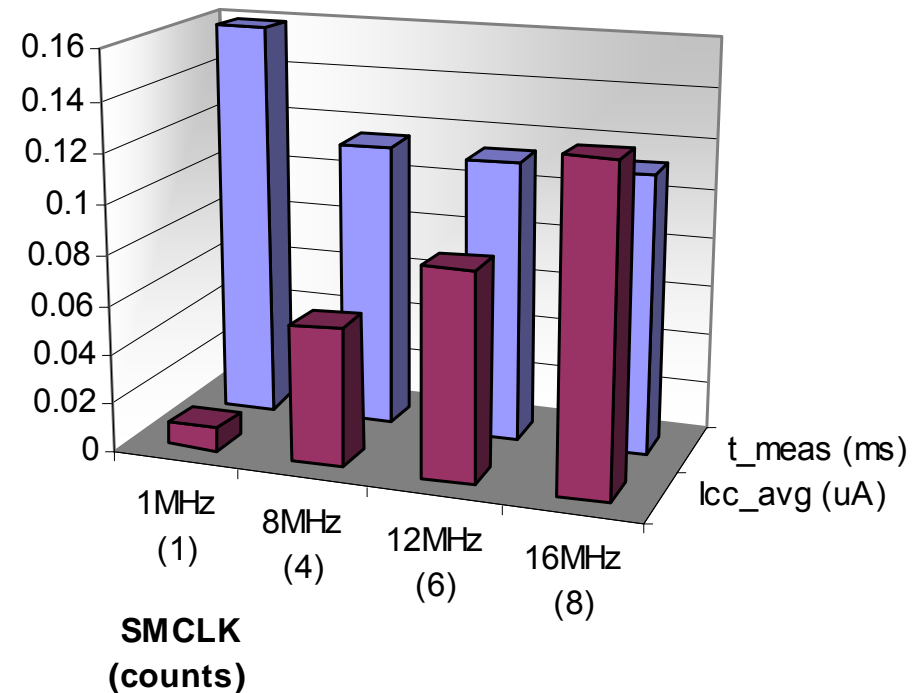
- 两个按键共用一个电阻
- 从而，可以通过测量电容充电和放电时取平均来达到更好的噪声抑制



RC充放电功耗

- 测量时间:
 $\sim 2 * t_{RC_charge}$
 - R = 5.1Mohm
 - Counts TACLK
- 平均电流决定于:
 - Tau = RC
 - DCO 电流消耗
- 根据具体应用选取**TACLK**
 - delta C越大, 可选频率相对较低的timer_A时钟源
 - 测量计数值越少, 平均功耗越低

Current & Measurement Time vs. Measurement Window (1% C_delta)



RC系统分析

- 具有中断功能的端口的才能测量
- 一个感应按键需要一个引脚，同时与另一个按键共享电阻
- **R**值一般选**5.1M**等**M**级电阻
 - With pF C, large R required for a measurable charge/discharge time
- **MSP430**的端口低漏电流，使用非常适合于该方式的应用
- 通过充放电时间平均，来实现噪声抑制
- 测量时间由**RC**常数决定
 - 测用频率较高的时钟来计数时间，从而提高测量精度
- 在其他参实确定下，测量值还跟电源和温度（**DCO**）有关

议程

- 触摸感应应用概述
- 系统级设计考虑
- MSP430实现方式
- 按键、划条、演示板和应用案例
- 总结

触摸感应考虑

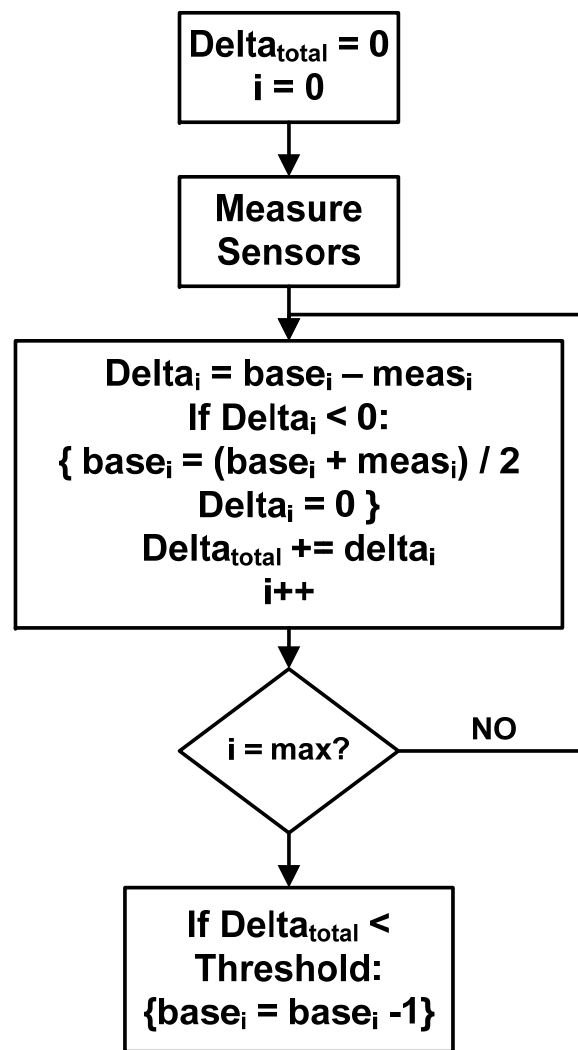
- 应用于:
 - 按键
 - 定位 (滑条)
- 极限: 设定一个可靠的限制
 - 考虑极限情况
 - 电源、电磁干扰极限
 - 批量生产差异化
- 滤波: 去除噪声
 - 在RC方式下, 加大值可以有效抑制噪声
 - 充放电多次测量取平均
 - 而在RO方式下, 本身已经多次充放电, 具很好的抑制功能
- 跟踪: 基本电容值偏移
 - 周期性的调整基本电容
 - 注意在基本电容调整时, 当心摸键动作

基本电容跟踪

- 基本电容可能会发生偏移：
 - 湿度
 - 温度
 - 器件老化
 - 电源
- 如果没有很好的对基本电容跟踪，可能造成误感应或感应无效
- 算法基础：
 - 对基电容减少要快速调整，因为触摸感应时，基电容是不是增加的
 - 对基电容增加要慢速调整，因为有可能是有效触摸
 - 停止基电容调整，当按键动作被检测到时

例程: 基电容跟踪

- 当基电容下降时, 进行快速调整
 - Ex: re-average with current result
- 当基本电容上升时, 进行慢速调整
 - Ex: adjust by 1 with each measurement
 - Only adjust if no keys are pressed
- 设定的个阈值, 当任何一个键被按下时, 所用键的电容变化量大于该值
 - Alternatively, can adjust on per key basis
- 注意:**delta**的符号在**RC**与**RO**方式下是不样的
 - RO: delta为负, 当键按下时
 - RC: delta为正, 当键按下时



数据滤波

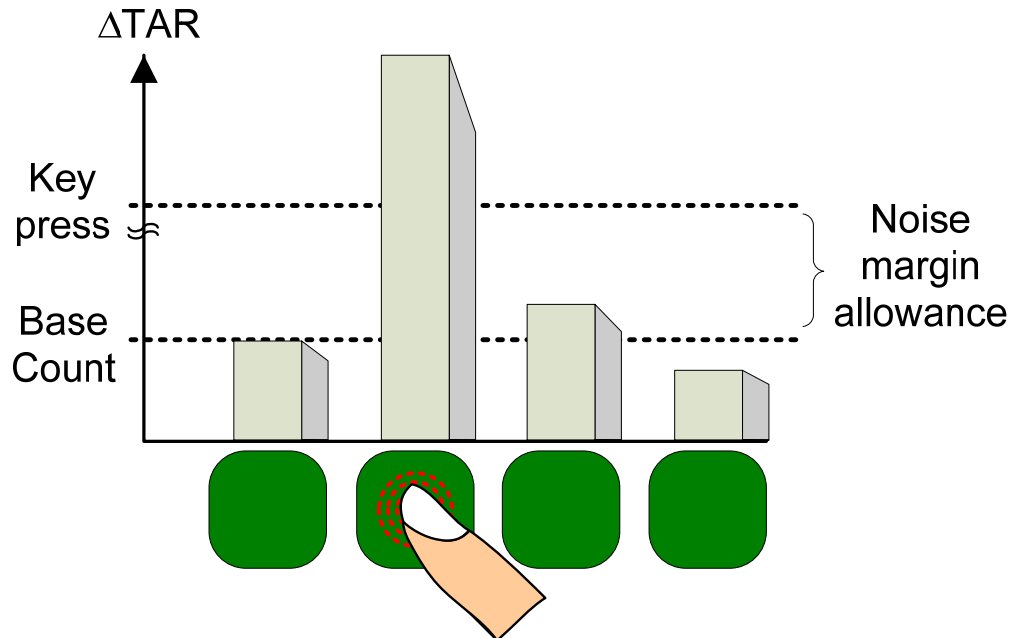
- 测量结果可能会受到电源波动的影响
- 通过多次测量，如果最低位波动，说明情况良好
 - 通过单个键检测
- 低通滤波算法可以有效地抑制干扰和噪声，使测量结果稳定，特别是对滑条位置检测的应用

按键检测

- 测量流程

- 第一步：测量基本电容
- 第二步：设定按键电容变化阈值
- 第三步：扫描按键

- 一般对这个阈值的选取，大致在感应电容最大变化量的50%



按键功耗

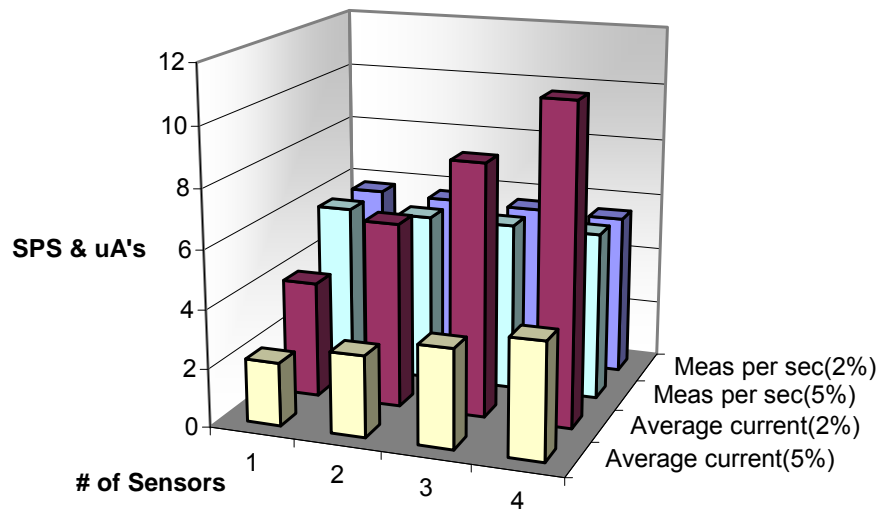
RO方式

- 猜用尽可能长的时间窗口和尽可能低的**Timer_A**时间源
 - ΔC 5% 1MHz, WDT= SMCLK/1/512
 - ΔC 2% 1MHz, WDT= SMCLK/4/512

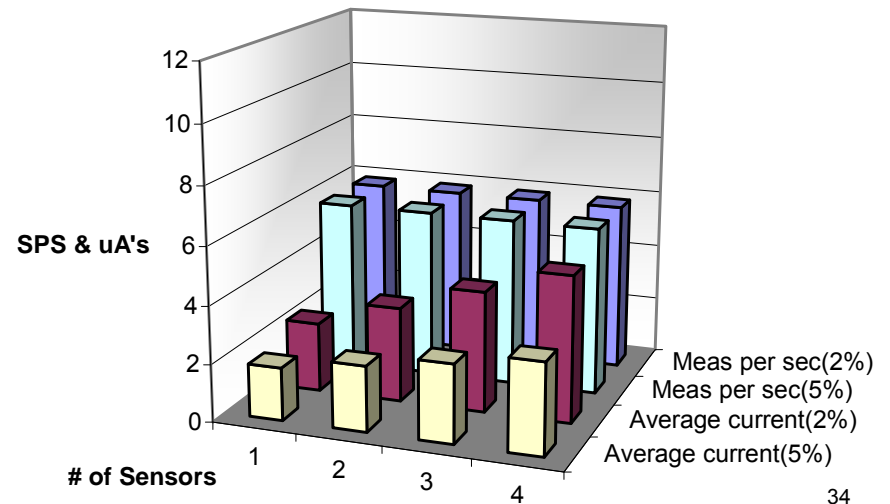
RC方式

- 选用尽量低的**TACLK**
 - ΔC 5% 8MHz TACLK
 - ΔC 2% 16MHz TACLK

Sensor Switch Application- RO
Current & SPS vs. Sensor Count (~20 counts)



Sensor Switch Application- RC
Current & SPS vs. Sensor Count (~20 counts)



TI Proprietary information

Demo: 超低功耗感应按键检测

- RC测量流程

```
// RC Method: Measurement Excerpt
...
P1OUT &=~(BIT0+BIT1+BIT2+BIT3); // everything is low
P1OUT |= active_key; // Charge the sensor
_NOP();_NOP();_NOP(); // short time for hard pull high
P1IES |= active_key; //-ve edge trigger
P1IE |= active_key;
P1DIR &= ~active_key; // set the active key to input
timer_count = TAR; // Take a snapshot of the timer
LPM0;
meas_cnt[i]= timer_count;
... // Now repeat with charging cycle and average results
```

```
// Port ISR
...
timer_count=TAR-timer_count; // Get charge/discharge time
...
```

35

TI Proprietary information

Demo: 超低功耗感应按键检测

- RO测量流程

```
// RO Method: Measurement Excerpt
TACTL = TASSEL_0+MC_2;           // TACLK, cont mode
TACCTL1 = CM_3+CCIS_2+CAP;      // Pos&Neg Capture
CACTL1 |= CAON;                 // Turn on comparator
for (i = 0; i<Num_Sen; i++)
{switch (i)
  {case 0: // Sensor 1
      CAPD = CA_Ref+S_1; // Disable I/O:CA1 ref, 1st sensor
      CACTL2 = CA_1+CA_2; // CA1 ref, CAx sensor
      break;
      ...}}
WDTCTL = WDT_meas_setting; // Set duration of sensor measurement
TACTL |= TACLK;           // Clear Timer_A TAR
LPM0;                     // Wait for WDT interrupt
meas_cnt[i] = TACCR1;     // Save result
```

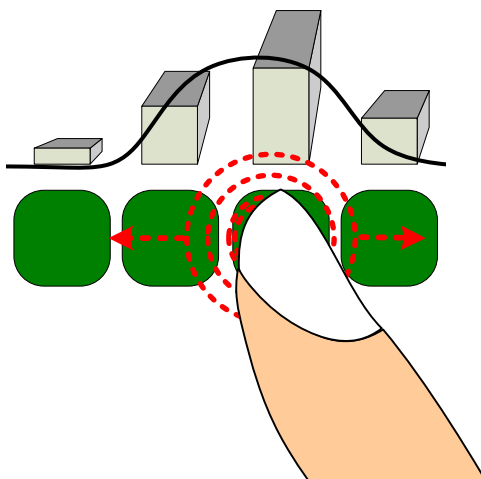
```
// WDT ISR
...
TACCTL1 ^= CCIS0; // Create SW capture of CCR1
...
```

滑条检测

- 测量流程

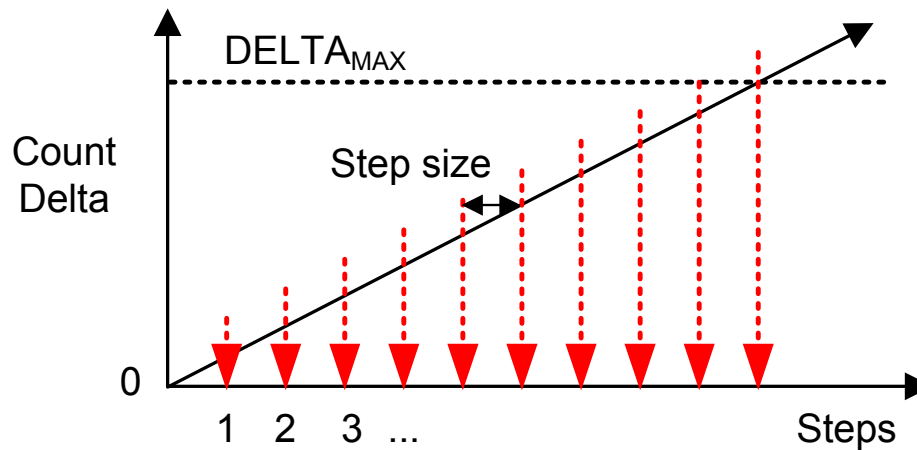
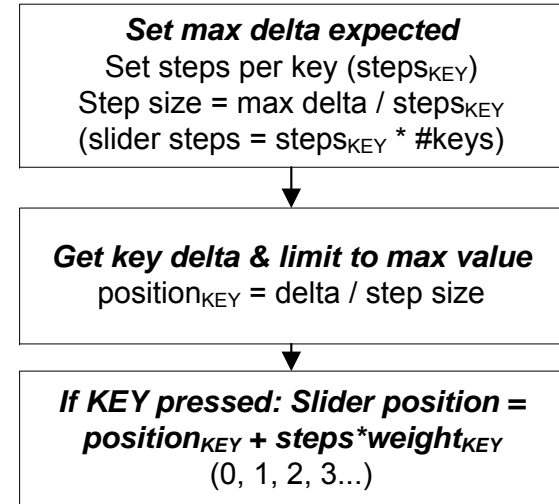
- 第一步：测量基本电容
- 第二步：设定按键电容变化阈值
- 第三步：扫描按键
- 第四步：通过各按键的电容变化值来计算出触摸滑条位置

- 采用线密度算法



定位

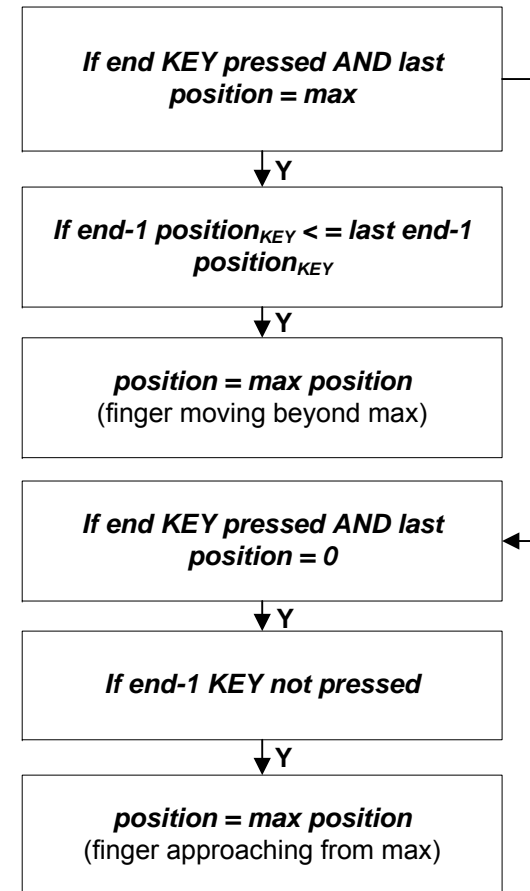
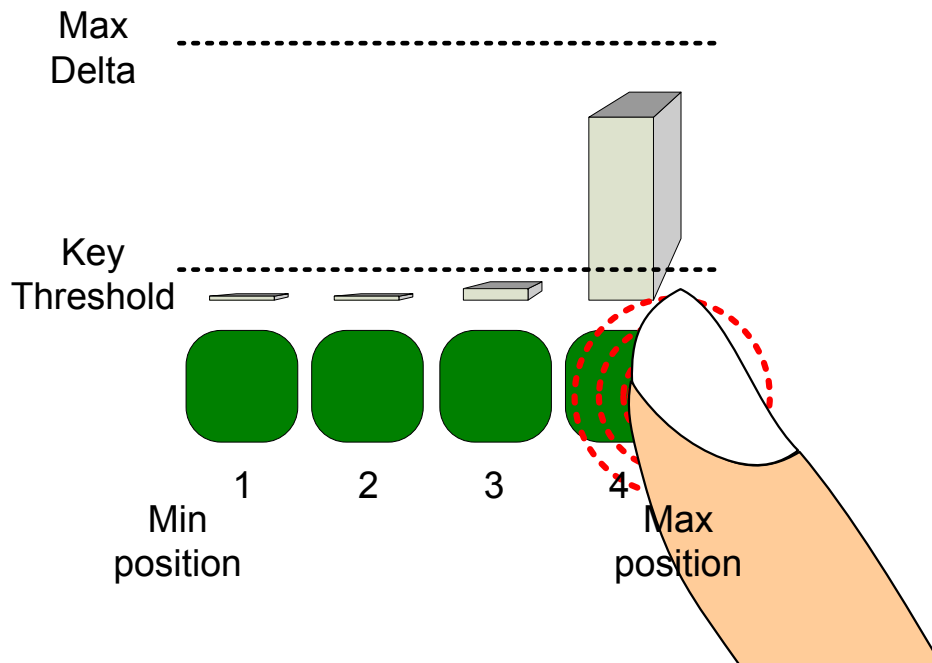
- 根据定位数和精度来设计按键
 - 感应点尺寸
 - 绝缘层厚度
- 使各按键之间均匀变化



端点

- 端点触摸处理

- Press beyond max
- Movement beyond max
- Movement from max



4个按键的滑条功耗

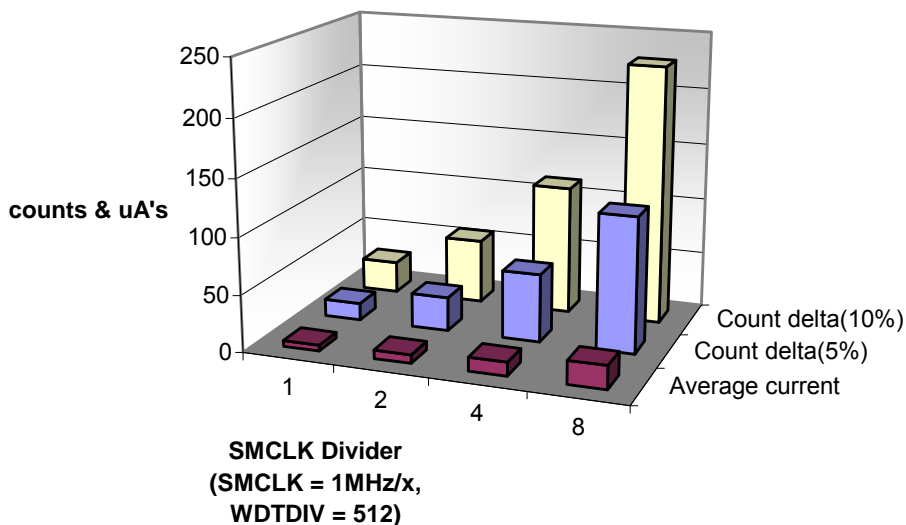
RO方式

- **t_meas**时间窗口可编程
 - Larger window = more counts
 - Define smallest window for needed counts, use lowest DCO for window

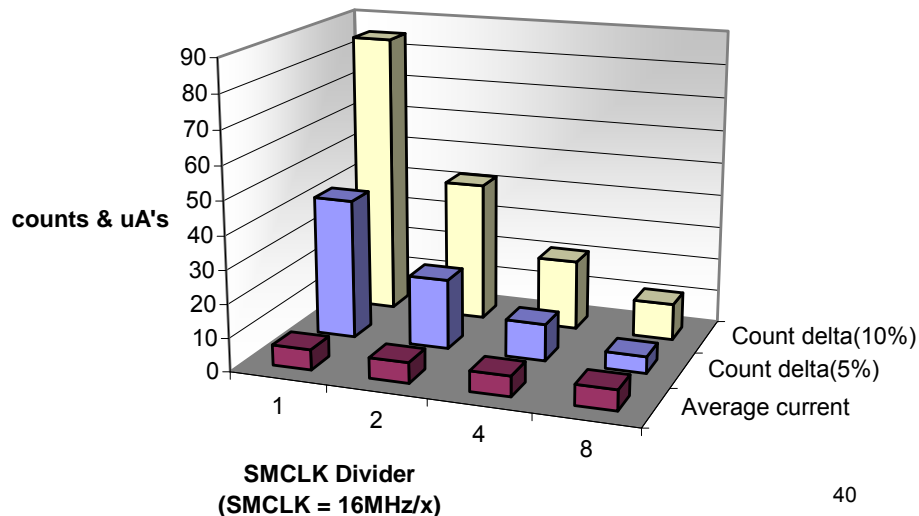
RC方式

- **t_meas**测量时间由**RC**确定
 - Faster TACLK = more counts
 - Don't divide TACLK, set = to fastest DCO required for needed counts

Sensor Slider Application- RO
Count Delta & Current Consumption vs. SMCLK (~5SPS)



Sensor Slider Application- RC
Count Delta & Current Consumption vs. SMCLK (~5SPS)



TI Proprietary information

Demo:低功耗滑条检测

```
// Sensor slider definitions
#define Num_Sen      4      // # of sensors
#define KEY_lvl      5      // min count for a "key press"
                          // Must be less than step_size

#define max_cnt      100    // Set below actual max delta expected
#define num_steps    16     // How many steps per key?
#define step_size    (max_cnt/num_steps) // Step size for position
...
if (delta_cnt[i] > max_cnt) // count exceeds preset upper delta
    delta_cnt[i] = max_cnt; // limit to set point

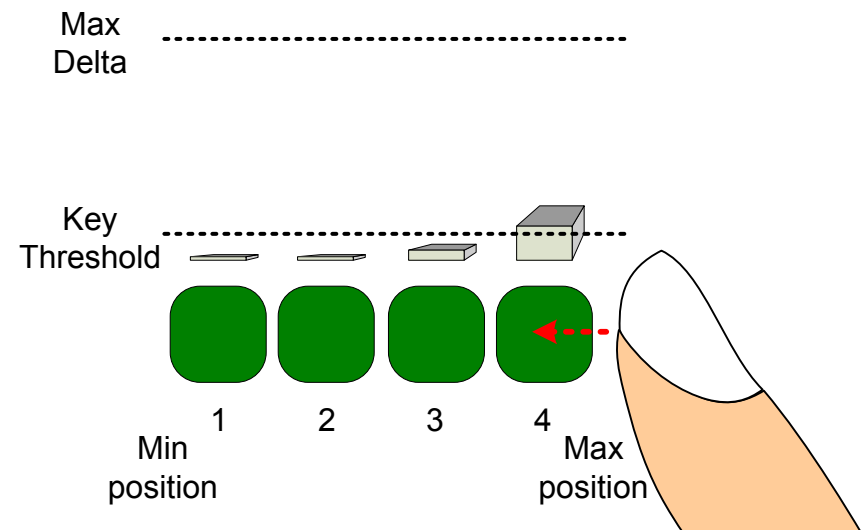
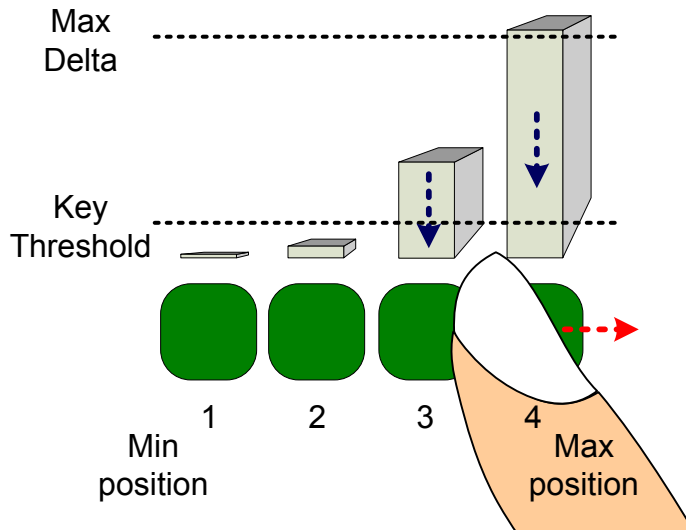
key_pos[i] = delta_cnt[i]/step_size; // individual "position"

if (key_pos[i] > 0) // If the key is "pressed",
    position = key_pos[i] + num_steps*(i); // Pos=0-16, key weight
```

- 根据应用确定滑条组成按键数
- 使按键设计尽量线性化

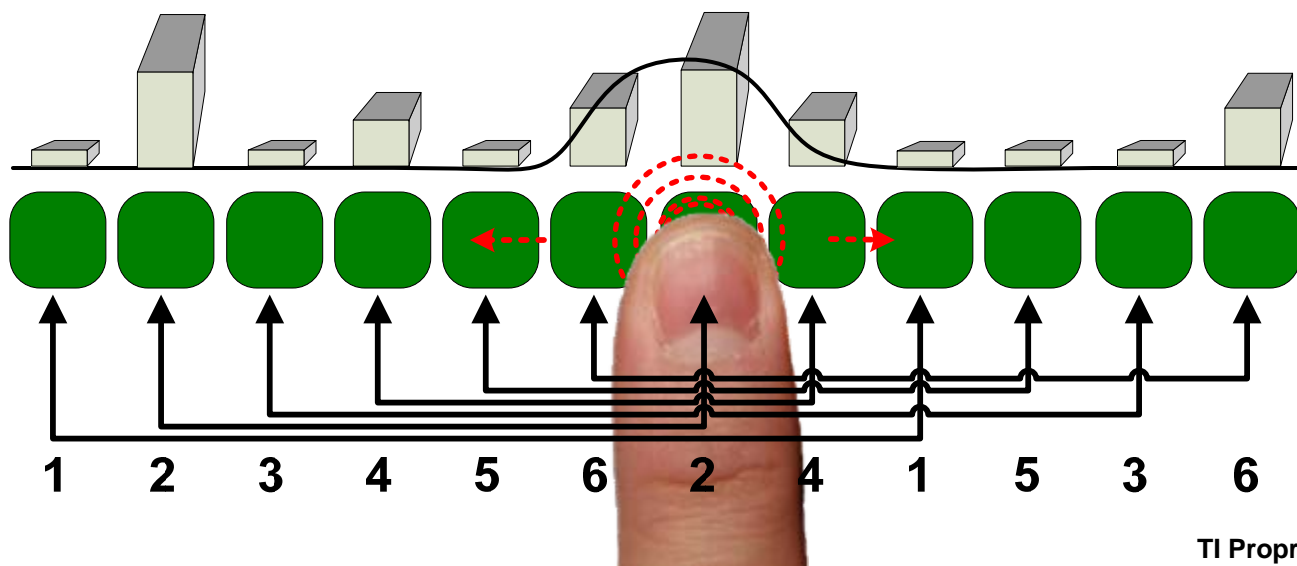
Demo:低功耗滑条端点处理

```
// Handle max end of slider
if (key_press[3] && position_old == Num_Sen*num_steps)
{if (key_pos[2]<key_pos_old[2] || key_pos[2]==key_pos_old[2])
    position = Num_Sen*num_steps; // moving beyond the max
}
else if (key_press[3] && position_old == 0 && !key_press[2])
    position = Num_Sen*num_steps; // approaching from max
```



I/O复用方式的滑条

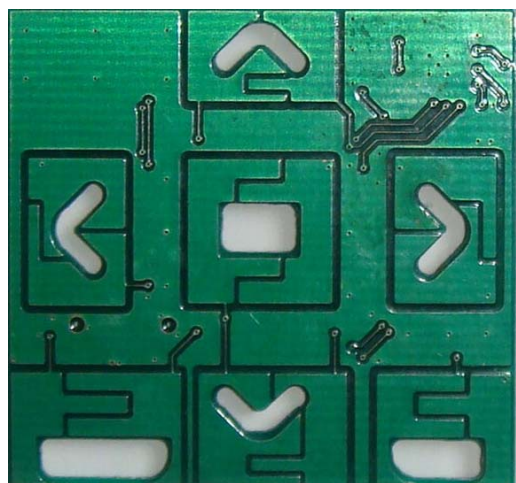
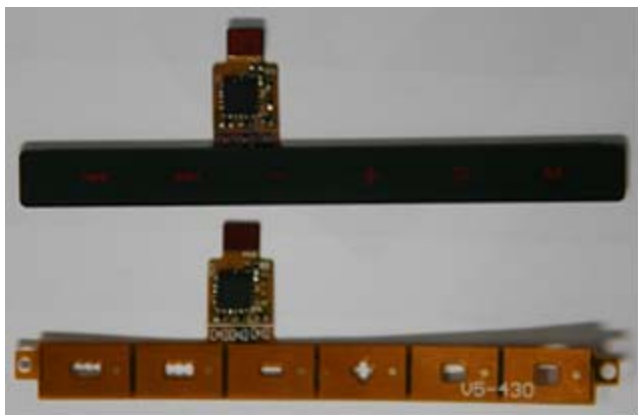
- 合理布置I/O复用来规划按键
 - 基本电容增加
 - 测量电容变化量会减少
- 使每个键感应时的状态是唯一对应的
- 通过复用可以检测感应位置和运动方向



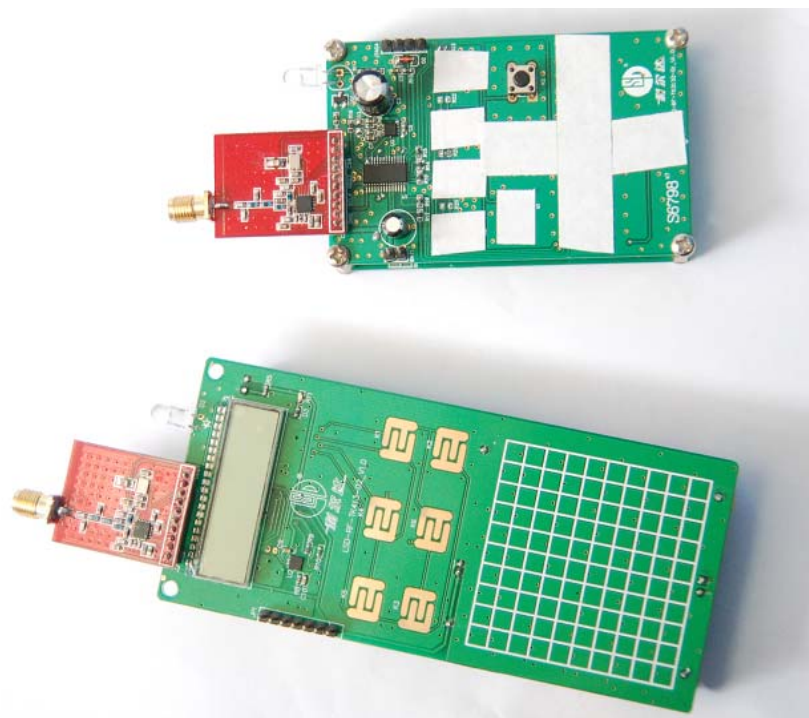
演示板



应用



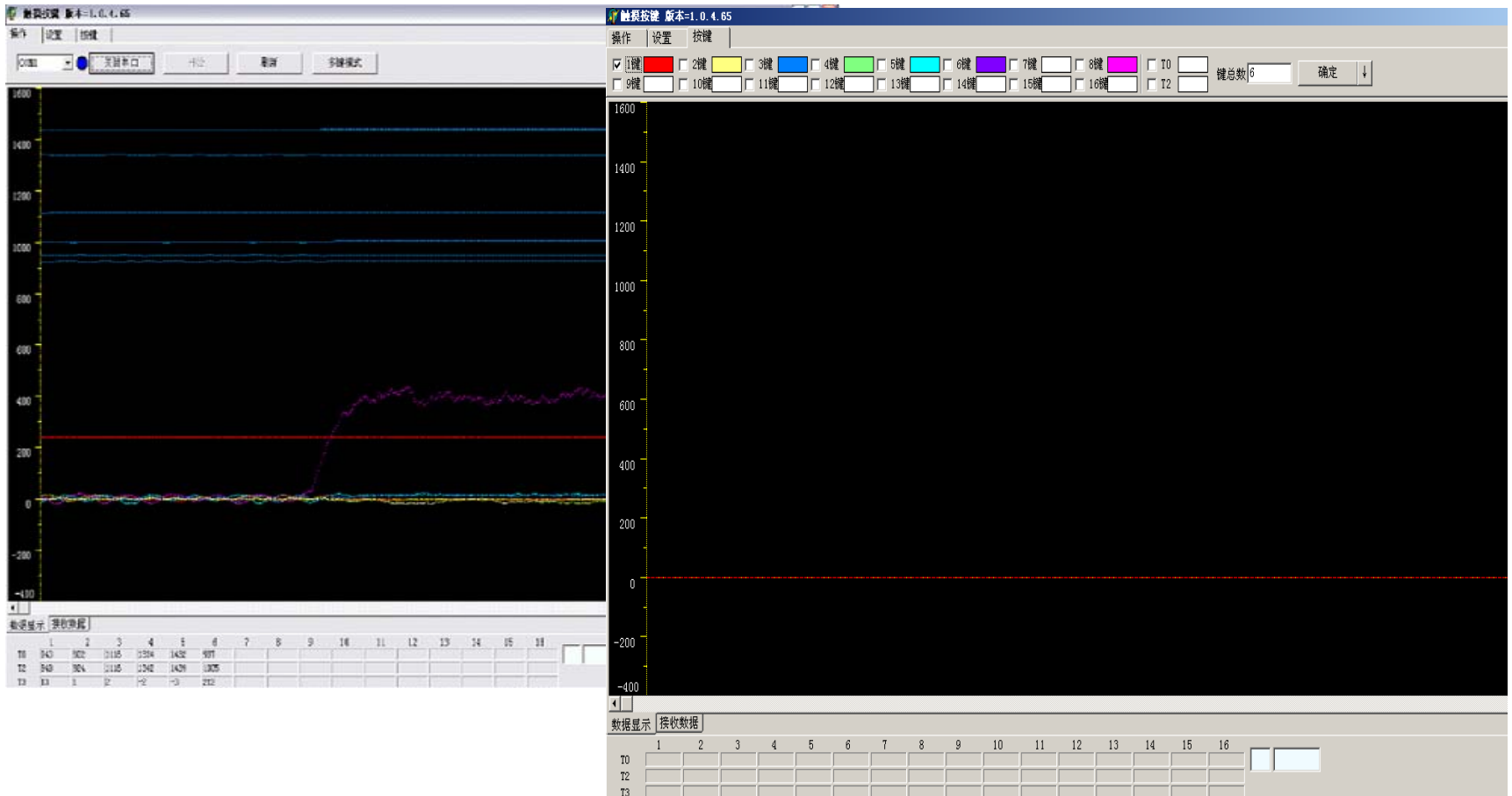
MP3
MP4



2.4G触摸式遥控器

TI Proprietary information

调试软件



议程

- 触摸感应应用概述
- 系统级设计考虑
- MSP430实现方式
- 按键、划条、演示板和应用案例
- 总结

总结

- 电容感应式触摸按键是一个具有吸引力的选择
 - ...替代现有按键
 - ...代替电位计和定位
- **MSP430 RO 方式**
 - 只能用内部带Comp_A的MSP430实现
 - 数量受Comp_A输入端的限制
 - 每个键需要一个R，并且需要三个参考电阻
 - 测量精度受功耗的制约
- **MSP430 RC 方式**
 - 可以在任何MSP430上实现
 - 最多可以实现16个独立的按键
 - 一个电阻可以与两个按键共用
 - 测量精度受系统时钟频率制约
 - 相比RO功耗可以做得更低

谢谢!

www.ti.com.cn/processors
www.lierda.com