



MICROCHIP 2010

MASTERS Conference

C11H01 CTH

Microchip mTouch™

电容触摸传感解决方案

课程目标

1. 大体了解电容传感器的基本原理
2. 理解Microchip的3种mTouch™触摸传感技术的工作原理
3. 逐一分析3种主要类别的电容触摸传感软件算法
4. 探究mTouch技术的软件架构和库；审视实现电容触摸传感解决方案的每个步骤

课程目标

**5. 本课程的学员将熟悉Microchip
mTouch™技术的软件架构和
mTouch库**

**6. 本课程的学员将克服在设计触摸传感
应用时所面临的一些实际困难**

课程安排

- 电容触摸传感器的基本物理原理
- **3种Microchip触摸传感方法**
 - CVD —— 电容式分压器
 - CTMU —— 充电时间测量单元
 - CSM —— 电容传感模块
- 选择适用于电容触摸传感应用的**Microchip**器件
- 软件处理
 - 采集
 - 滤波
 - 解码
- 入门——使用**mTouch™**技术软件架构和**mTouch**库

课程安排

- **CVD**简要回顾
- **实验1a** —— 检查硬件设置 (**CVD**)
- **CVD**库的工作原理
- **实验1b** —— 使用**CVD**软件架构
- **实验2c** —— 覆盖材料的效果

- **CTMU**简要回顾
- **实验2a** —— 检查硬件设置 (**CTMU**)
- 如何使用**CTMU**库
- **实验2b** —— 使用**CTMU**库

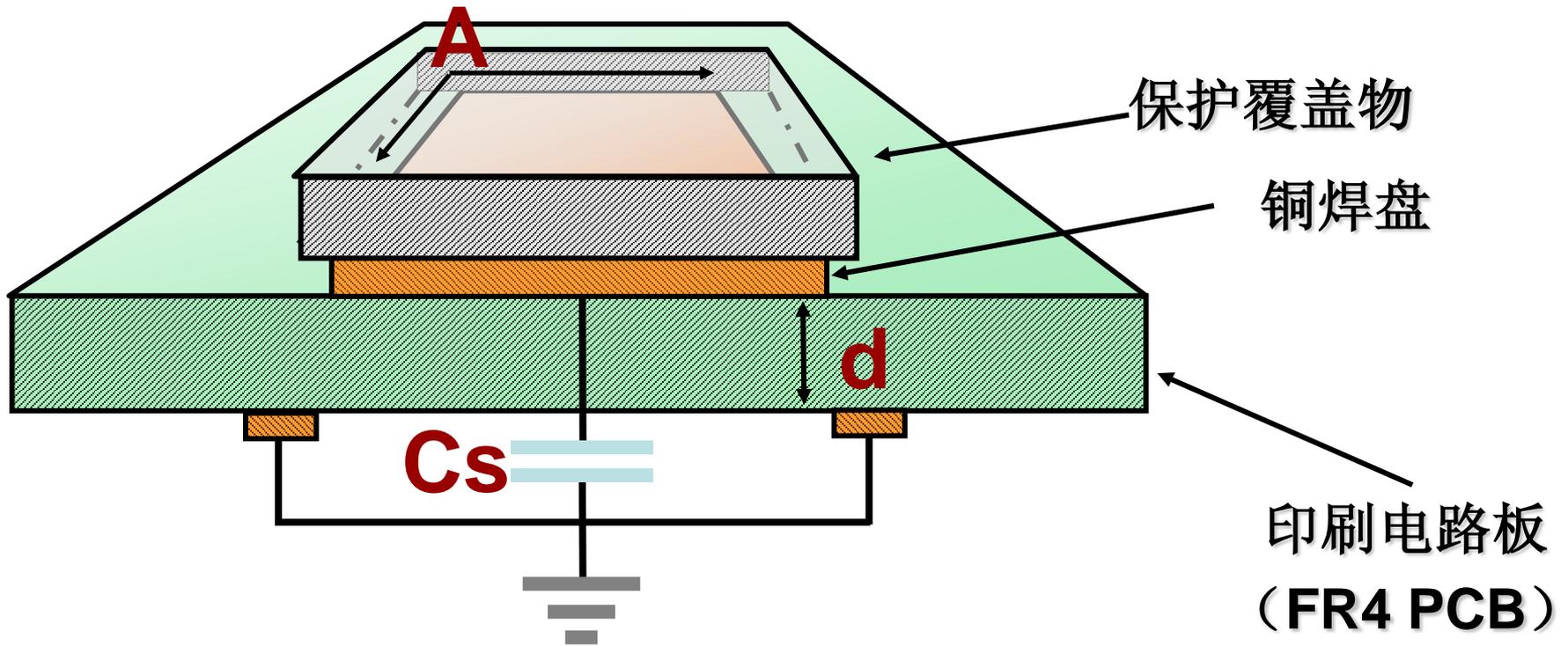


MICROCHIP 2010

MASTERS Conference

电容触摸传感器的物理原理

触摸传感器的构造



$C_s =$ 传感器基本电容

电容传感原理介绍

i = 瞬时电流

C = 电容

dv = 电压的变化

dt = 时间的变化

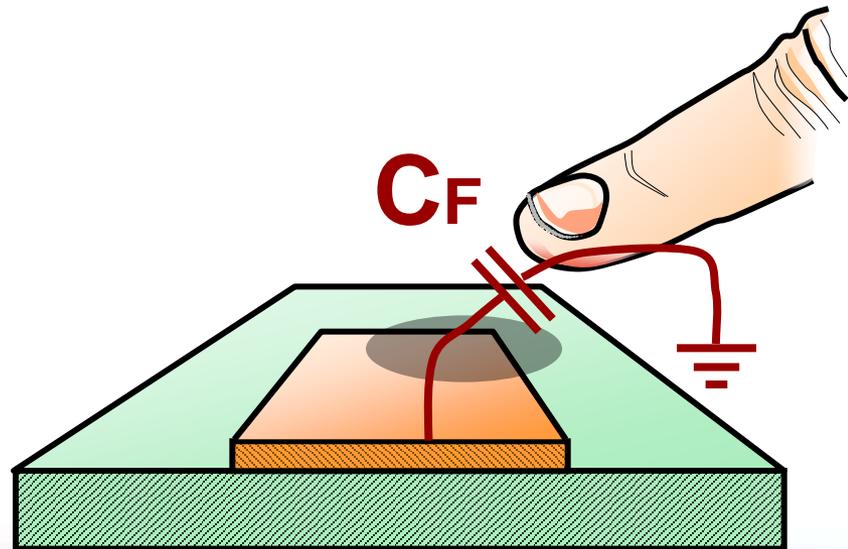
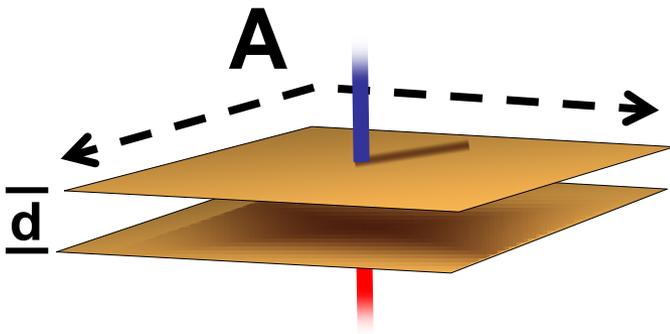
$$C = \frac{\varepsilon A}{d}$$

$$i = C \frac{dv}{dt}$$

物理原理的快速回顾

- 使用双极板电容模型
- 极板1=传感器焊盘
- 极板2=手指/导电物质或电离物质

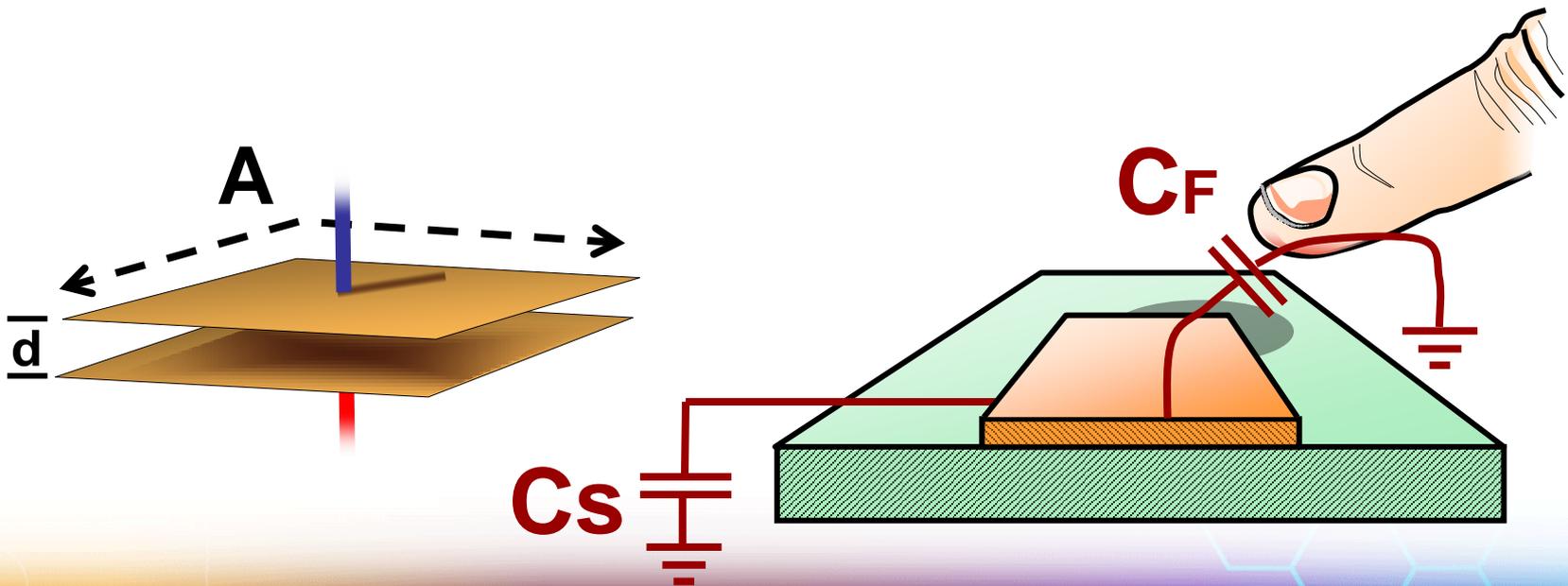
$$C = \frac{\epsilon_0 \epsilon_r A}{d}$$



物理原理的快速回顾

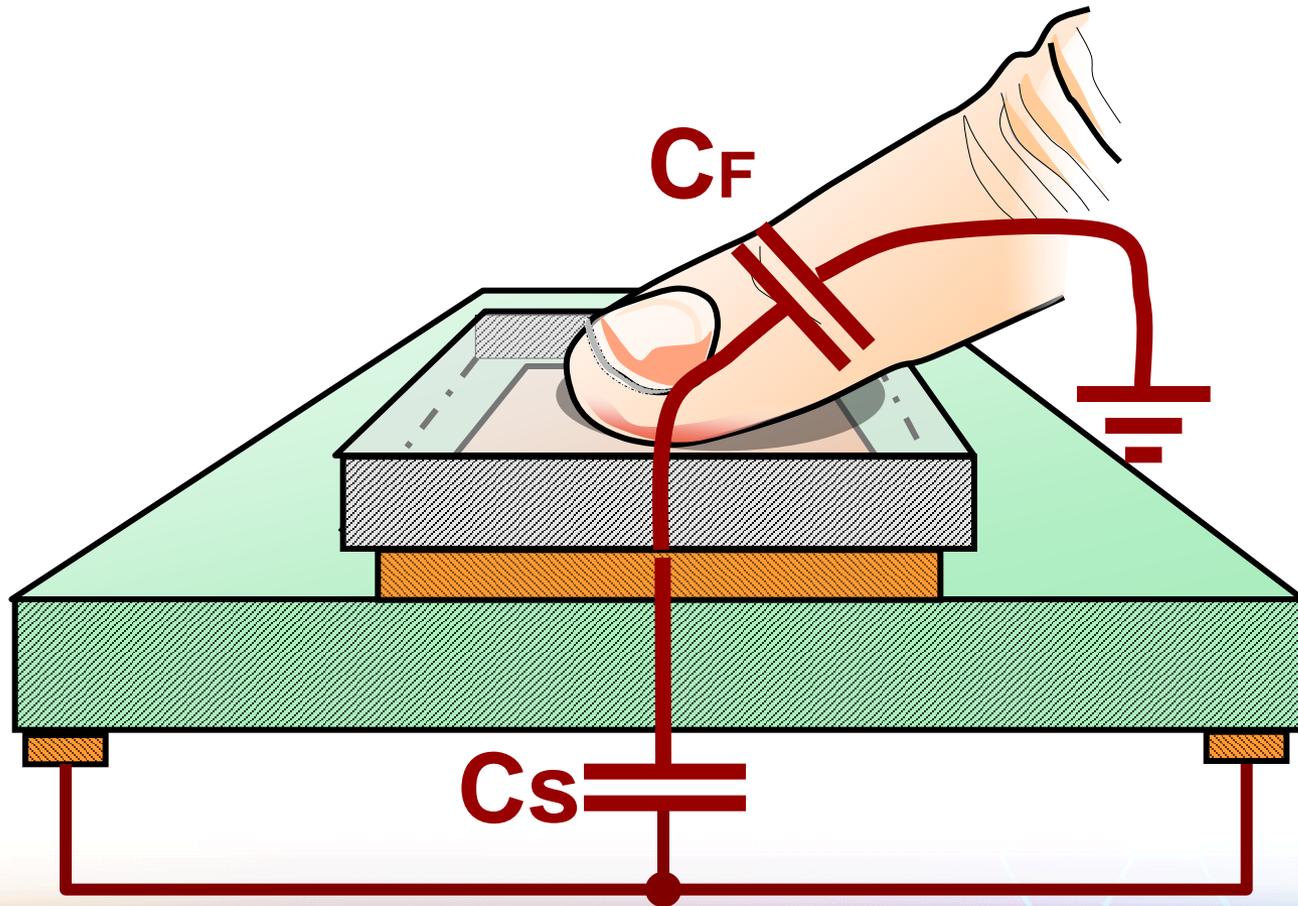
- 使用双极板电容模型
- 极板1=传感器焊盘
- 极板2=手指/导电物质或电离物质

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

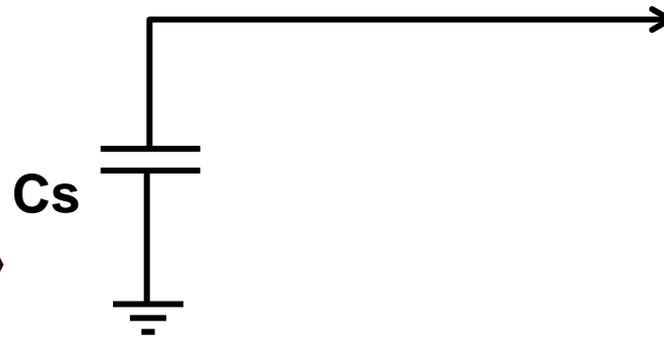
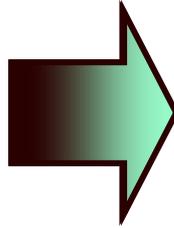
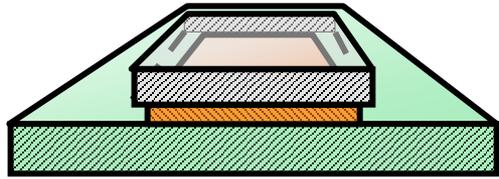


工作原理

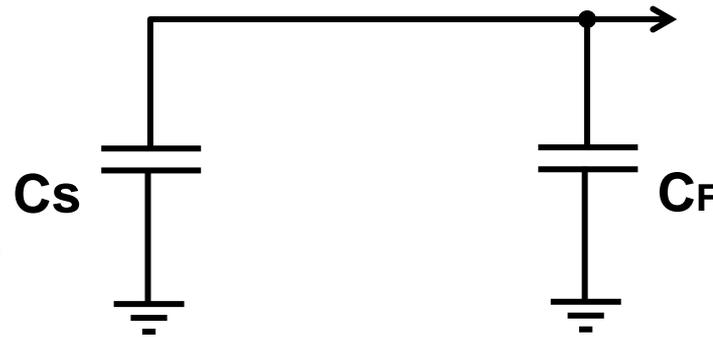
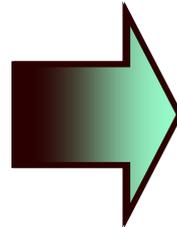
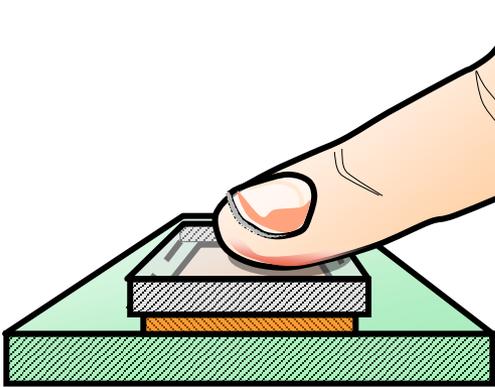
- 手指的引入产生了一个并联电容



等效电路



传感器电容 = C_S



传感器电容(C_T) = $C_S + C_F$

传感器的电容

- 随着 C_S 的改变而变化 —— 环境
- 当手指触摸时发生变化 —— 用户

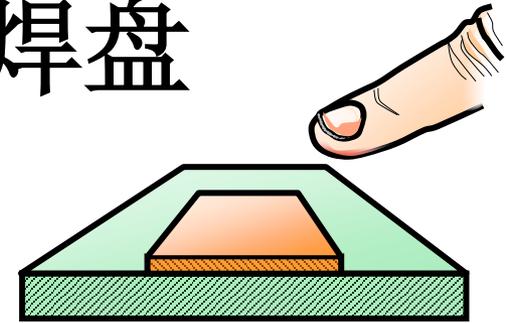
$$C_S + C_F = C_{TOT}$$

- 监测 C_{TOT} 的变化



传感器概述

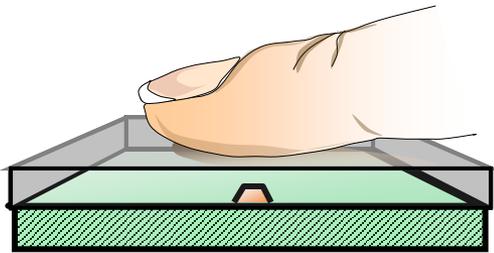
- 与手指发生耦合的导电焊盘
- 形状通常是任意的
- **A**、**d**和 ϵ_r 对性能起决定作用



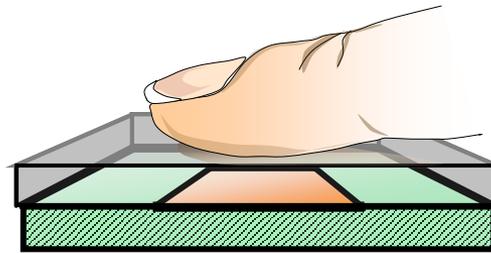
$$C = \frac{\epsilon_0 \epsilon_r A}{d}$$

传感器设计——感应区域

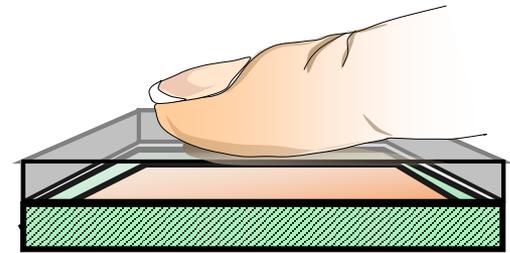
- 电容传感器的设计很简单
- 通常情况下，可以考虑将传感器设计成正方形，尺寸为一个手指大小



太小



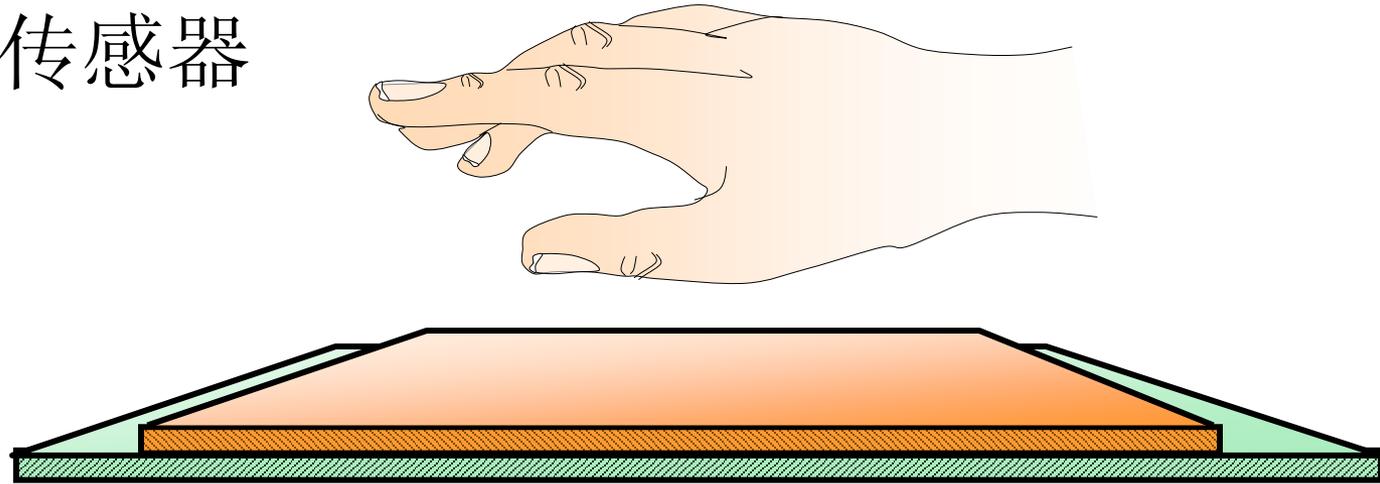
适中



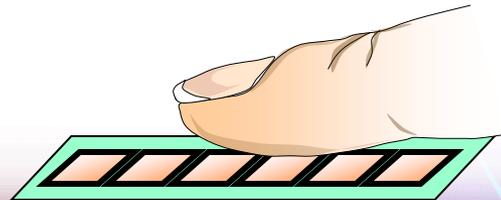
过大

传感器设计——感应区域

- 手指大小的传感器并非总是最好的!
 - 接近式传感器



- 精细传感器（需要薄的覆盖物）



覆盖材料 (距离)

- 完整的产品并非仅停留于传感器的设计
- 还包括外壳
- 覆盖材料通常为塑料

介电常数—— ϵ_r

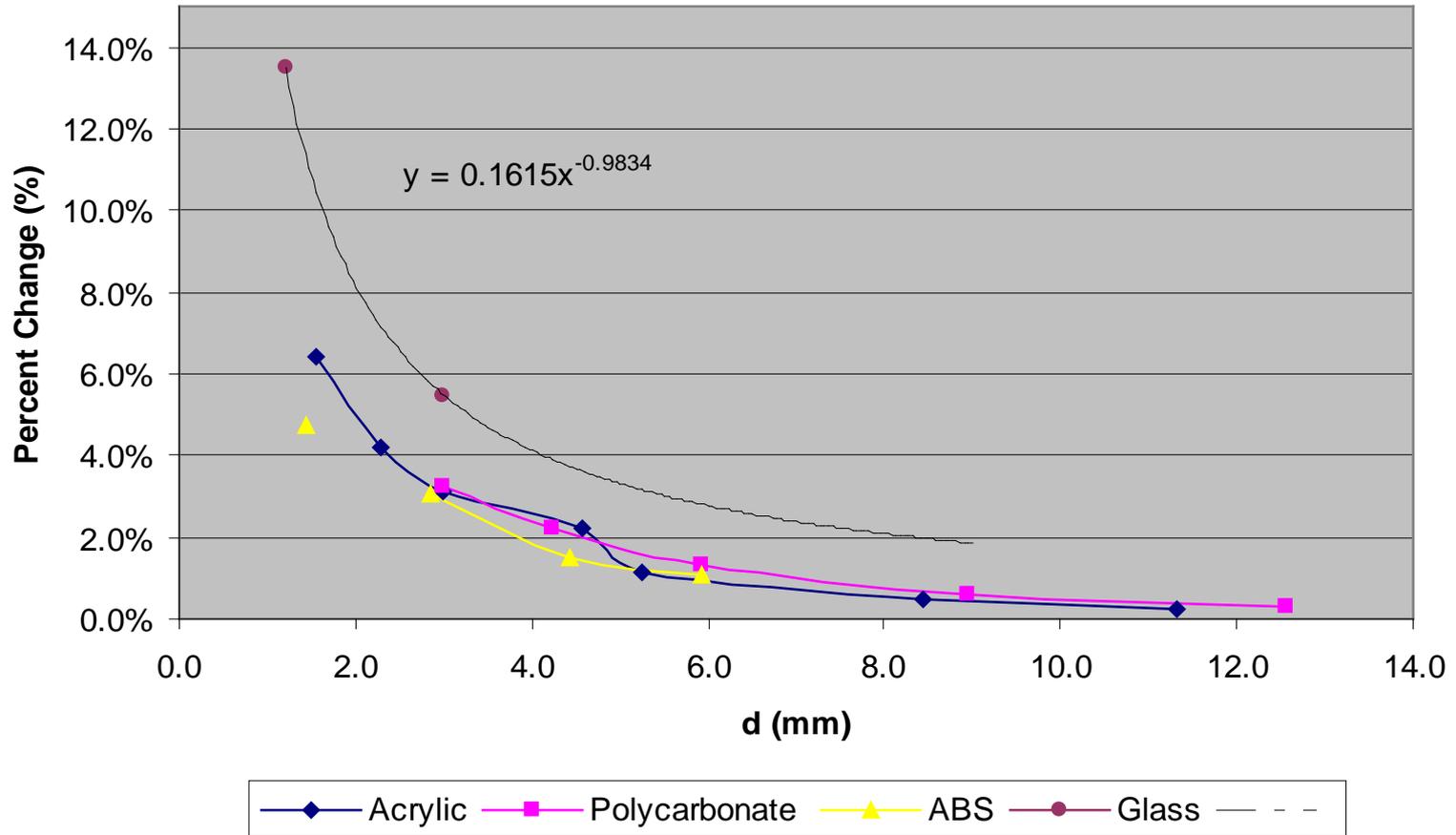
- **PCB 材料**
 - FR-4 4.20至4.70
 - CEM1 - 3 3.8至4.5
- **玻璃**
 - 铁封接玻璃 8.38至8.30
 - 钠硼硅酸盐 4.97至4.84
 - 熔凝石英 3.78
- **塑料**
 - 醋酸塑料 3.48至3.30
 - 环氧树脂 3.67至3.52
 - 聚碳酸酯 3.02至2.96
 - 聚乙烯 2.26
- **其他**
 - 液态水 78.20
 - 番茄酱 / 芥末酱 24.0
 - 冰 4.15
 - 雪 1.55

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



材料图

% Change to Finger, 0.5x0.5" pad - Differing Materials at Distance



物理原理总结

- 手指的触摸引入了一个对地的并联电容
- 传感器的尺寸要适中
- 覆盖层越薄越好
- 介电常数较大的材料较好



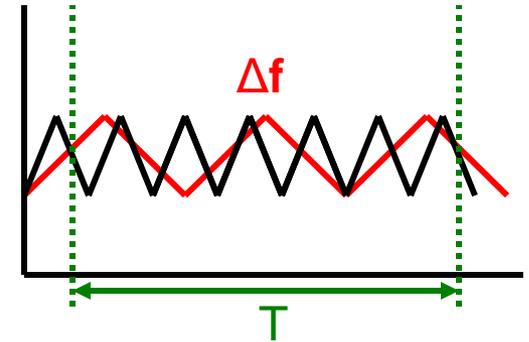
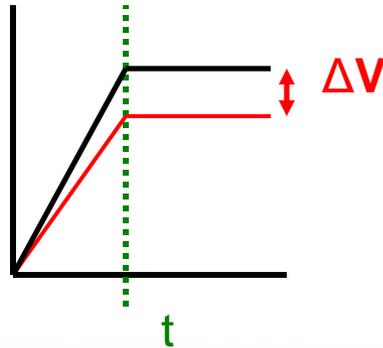
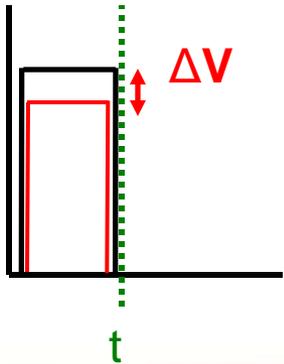
MICROCHIP 2010

MASTERS Conference

Microchip 触摸传感方法

需测量的物理量

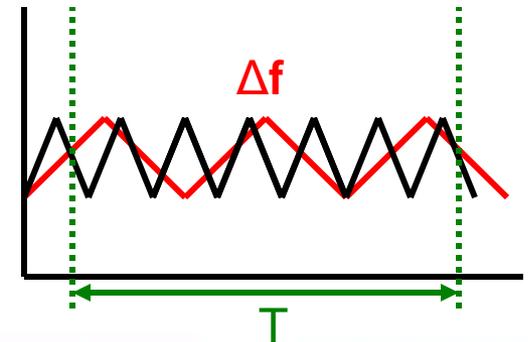
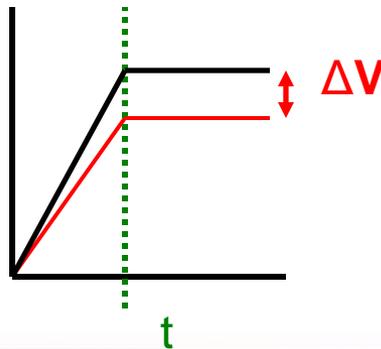
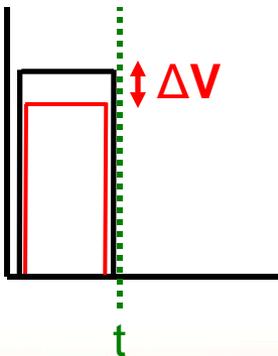
- 1. 基于电压
(固定时间 t 下的 ΔV)
- 2. 基于频率
(固定周期 T 下的 Δf)



C_F 如何改变读数

- Δf 解决方案基于等效的RC常数
- ΔV 解决方案基于固定时间内电容的充电情况
- 这两种情形下，额外引入的 C_F 均会引起 C_{TOT} 改变，从而改变待测物理量 (Δf 或 ΔV)

$$C_S + C_F = C_{TOT}$$



Microchip解决方案

Microchip采用的3种电容触摸传感方法

<u>方法</u>	<u>资源</u>
CVD	(ADC)
CTMU	(ADC + 模块)
CSM	(振荡器 + 模块)

方法术语

- **CVD =**
电容分压器
- **CTMU =**
充电时间测量单元
- **CSM =**
电容触摸传感模块

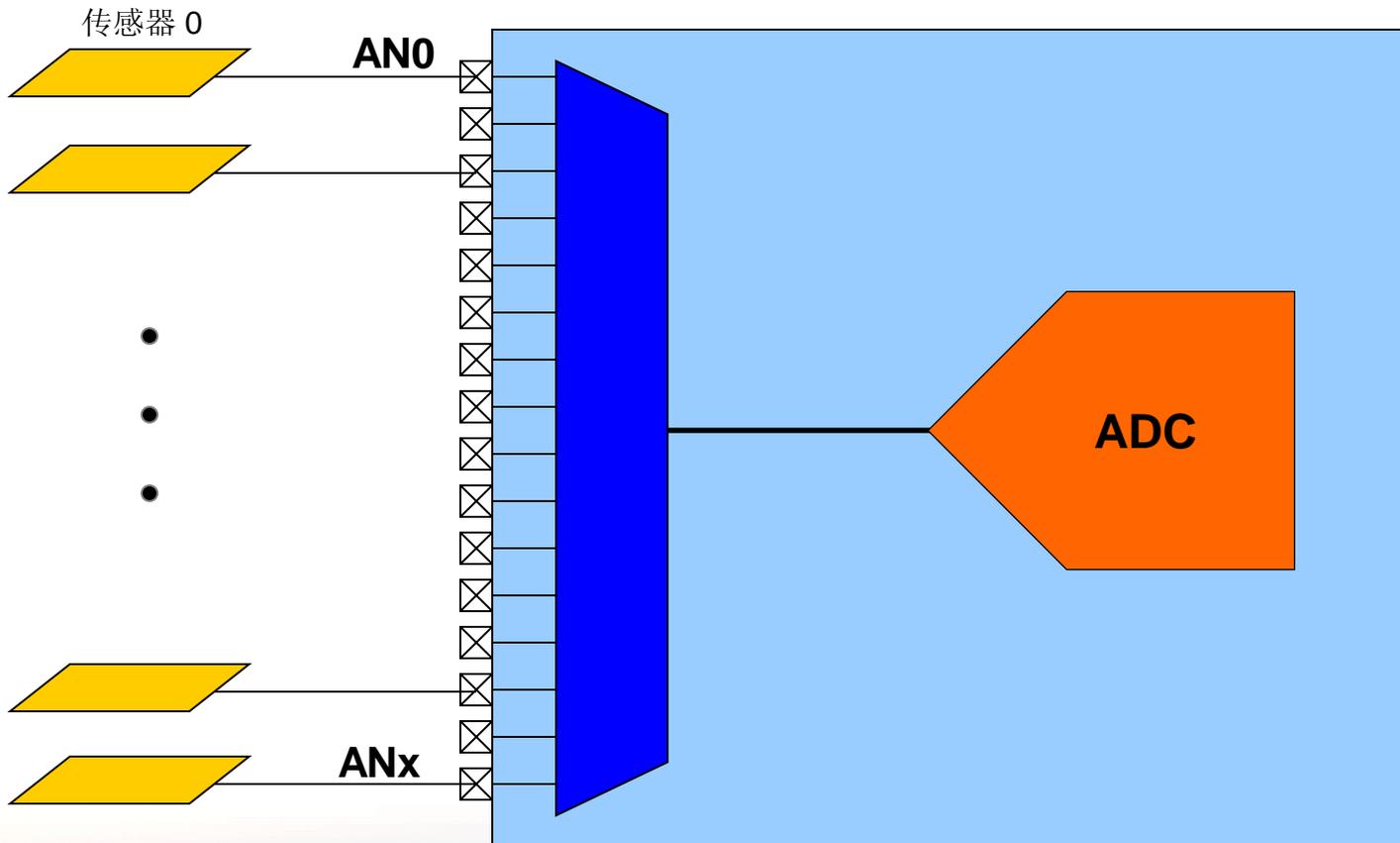


MICROCHIP 2010

MASTERS Conference

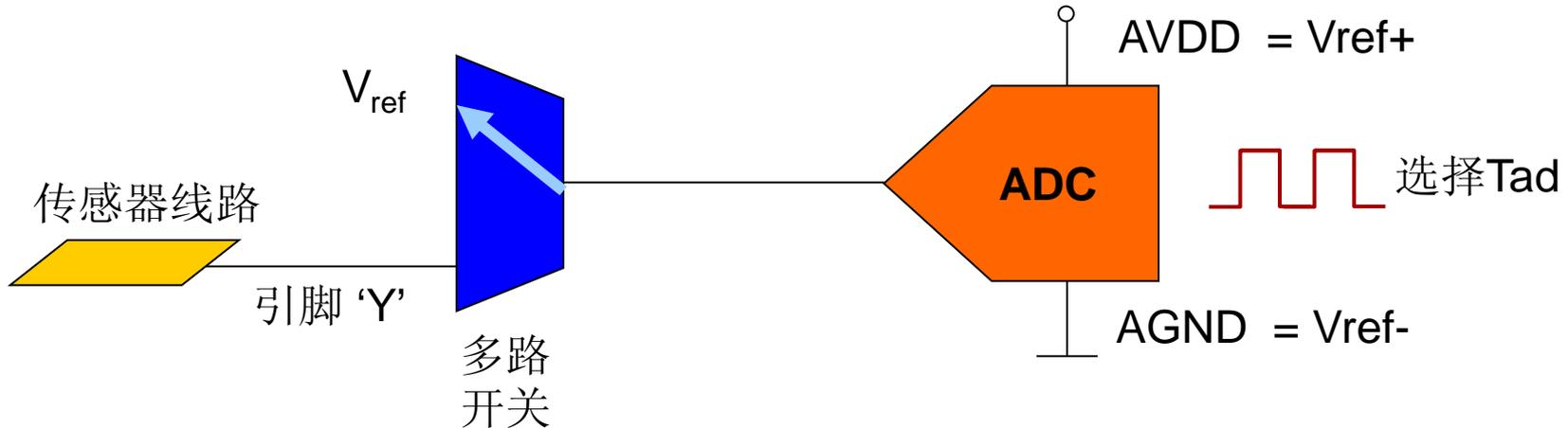
CVD的工作原理

传感器连接到ADC输入



● 初始化目标

1. 正常使能和配置ADC



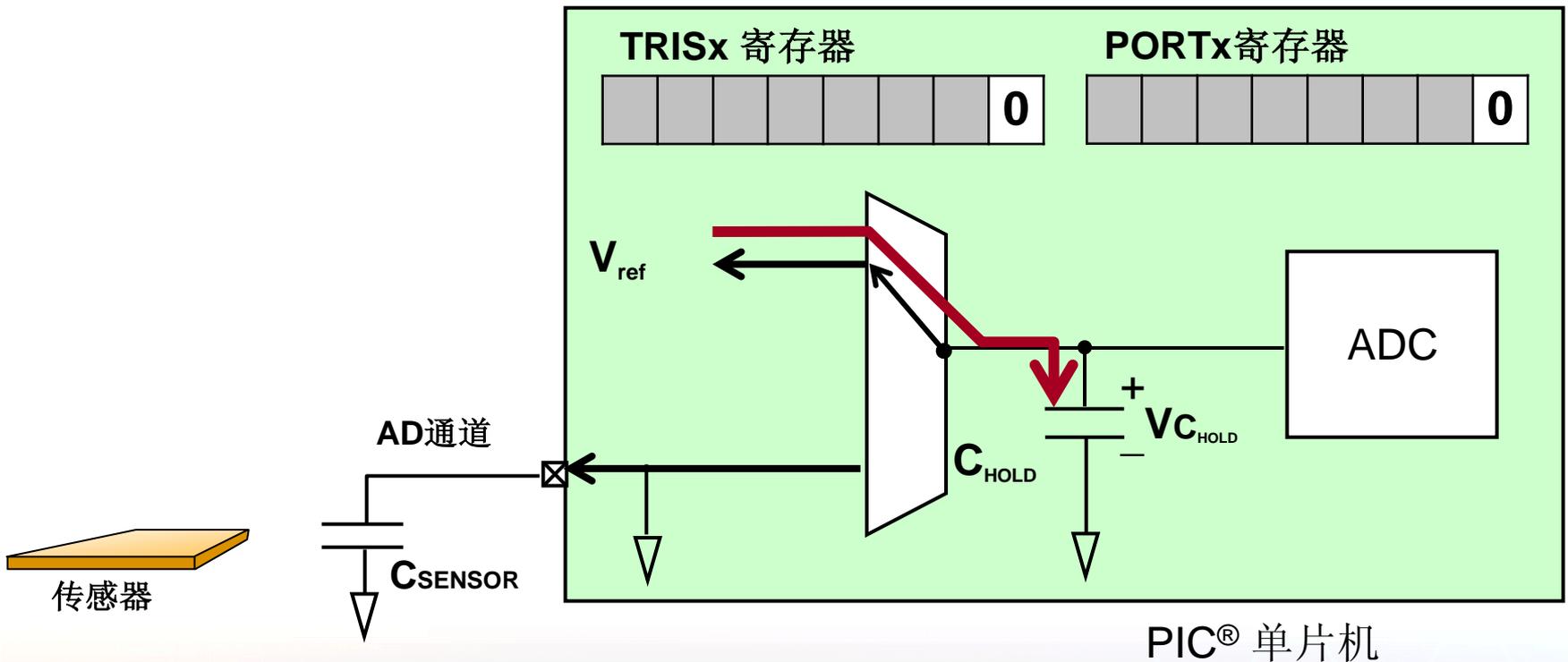
CVD扫描步骤

● 典型扫描步骤

1. 将参考I/O引脚“X”连接到VDD
2. 选择引脚“X”作为ADC通道
(将ADC的保持电容Chold充电至VDD)
3. 将传感器引脚“Y”接地
4. 将传感器引脚“Y”配置为输入
5. 选择引脚“Y”作为ADC通道
(电压取决于传感器电容与保持电容之比)
6. 读ADC结果

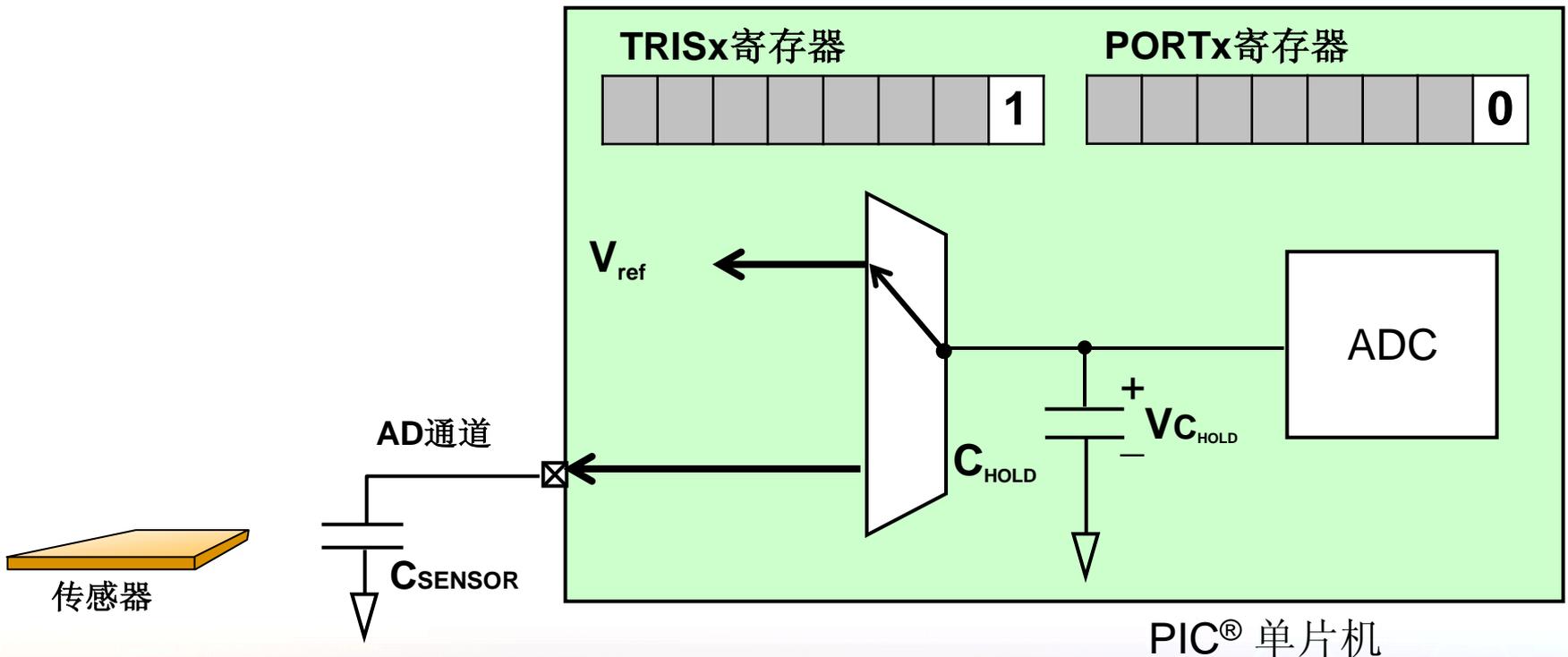
空闲状态

- C_{hold} 充电至 V_{ref}
- 传感器接地



获取传感器读数

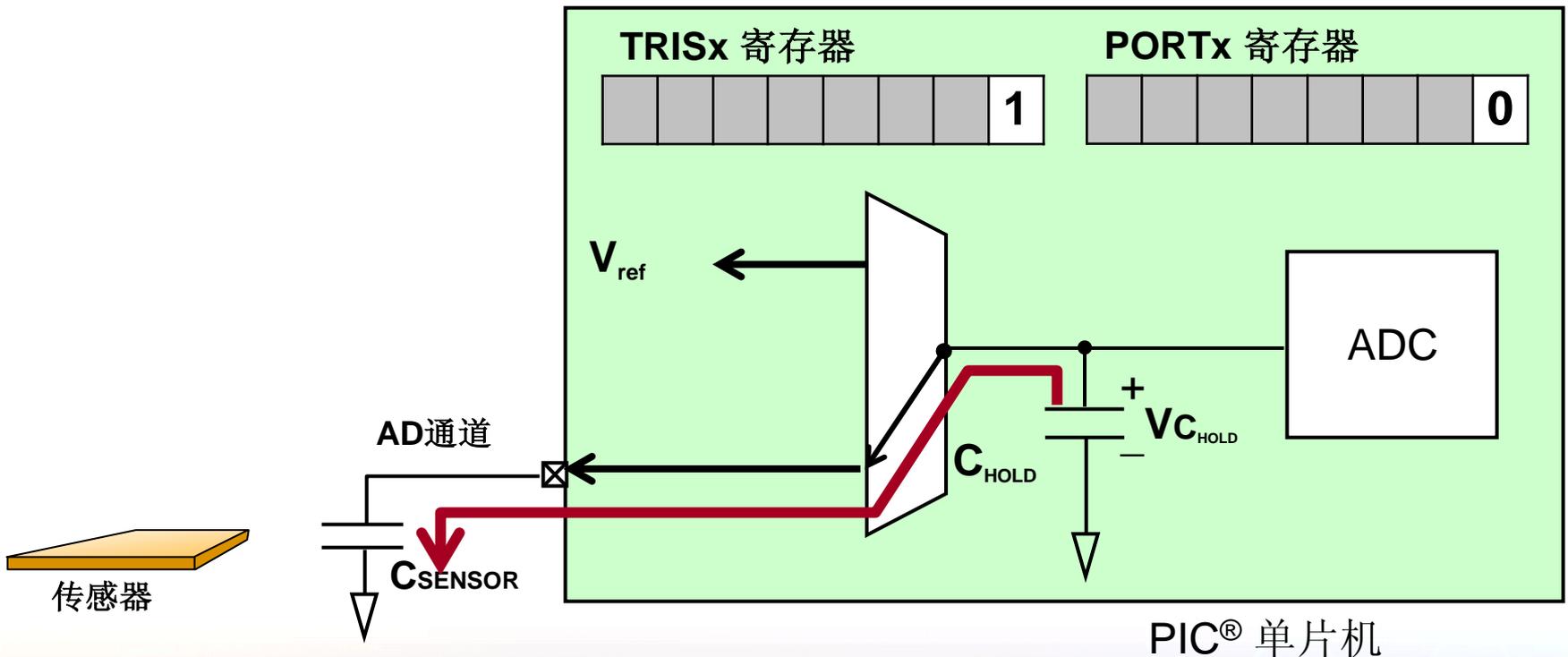
- 传感器作为输入



PIC® 单片机

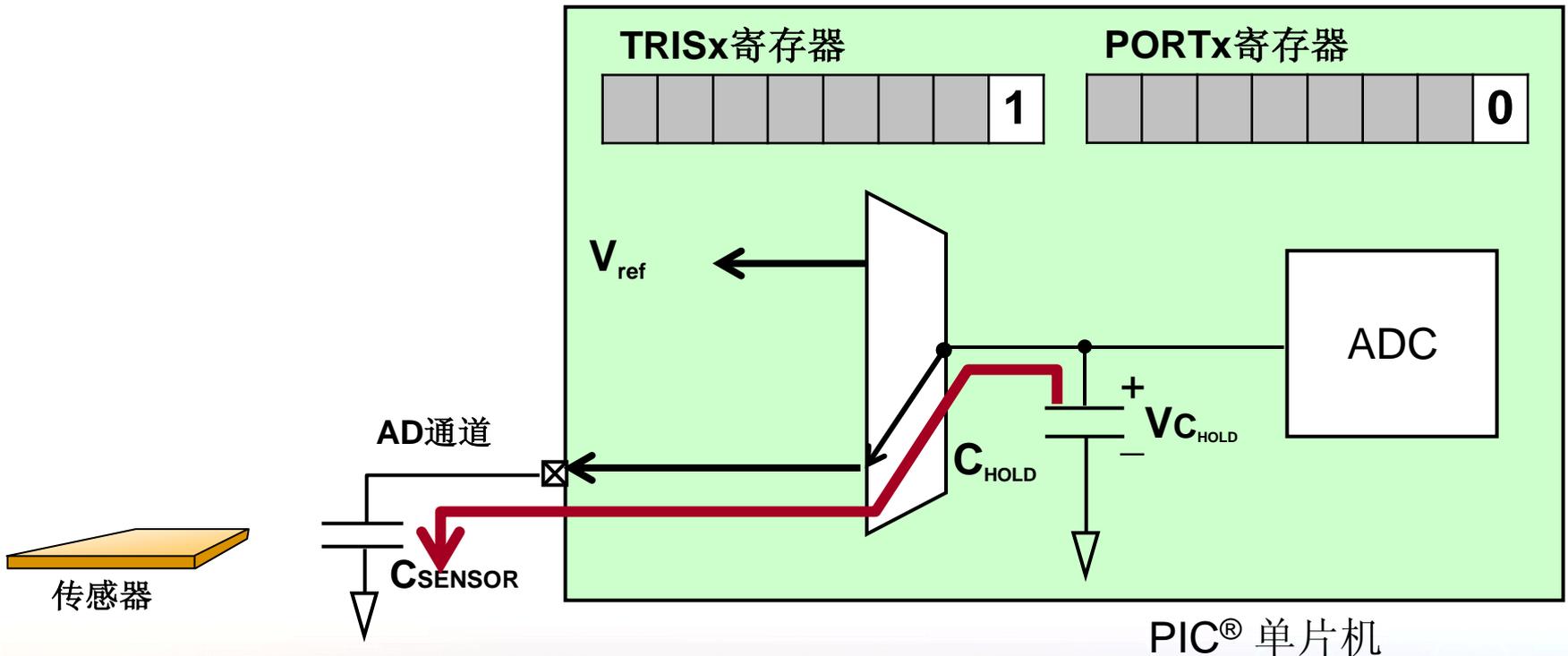
获取传感器读数

- A/D通道切换至传感器



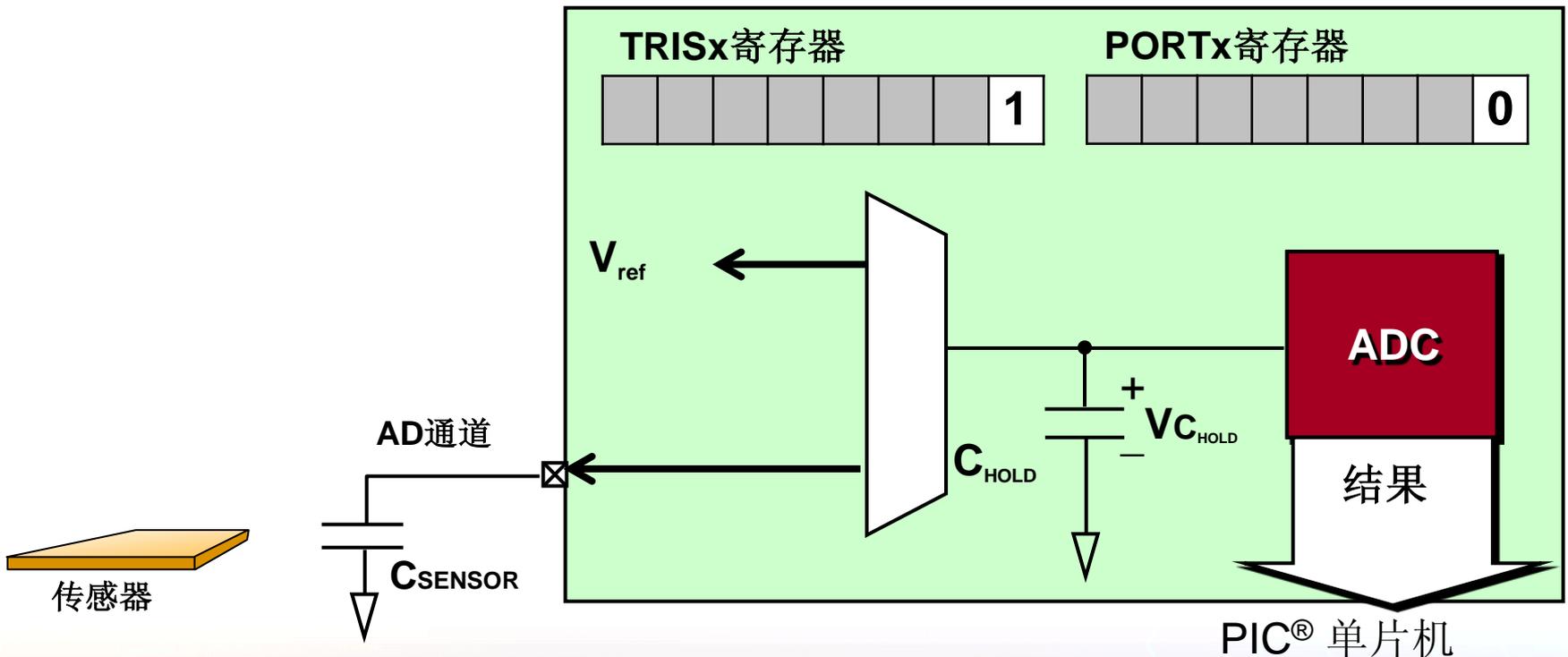
获取传感器读数

- $$VC_{\text{HOLD}} = VC_{\text{SENSOR}} = V_{\text{ref}} * C_{\text{hold}} / (C_{\text{hold}} + C_{\text{sensor}})$$



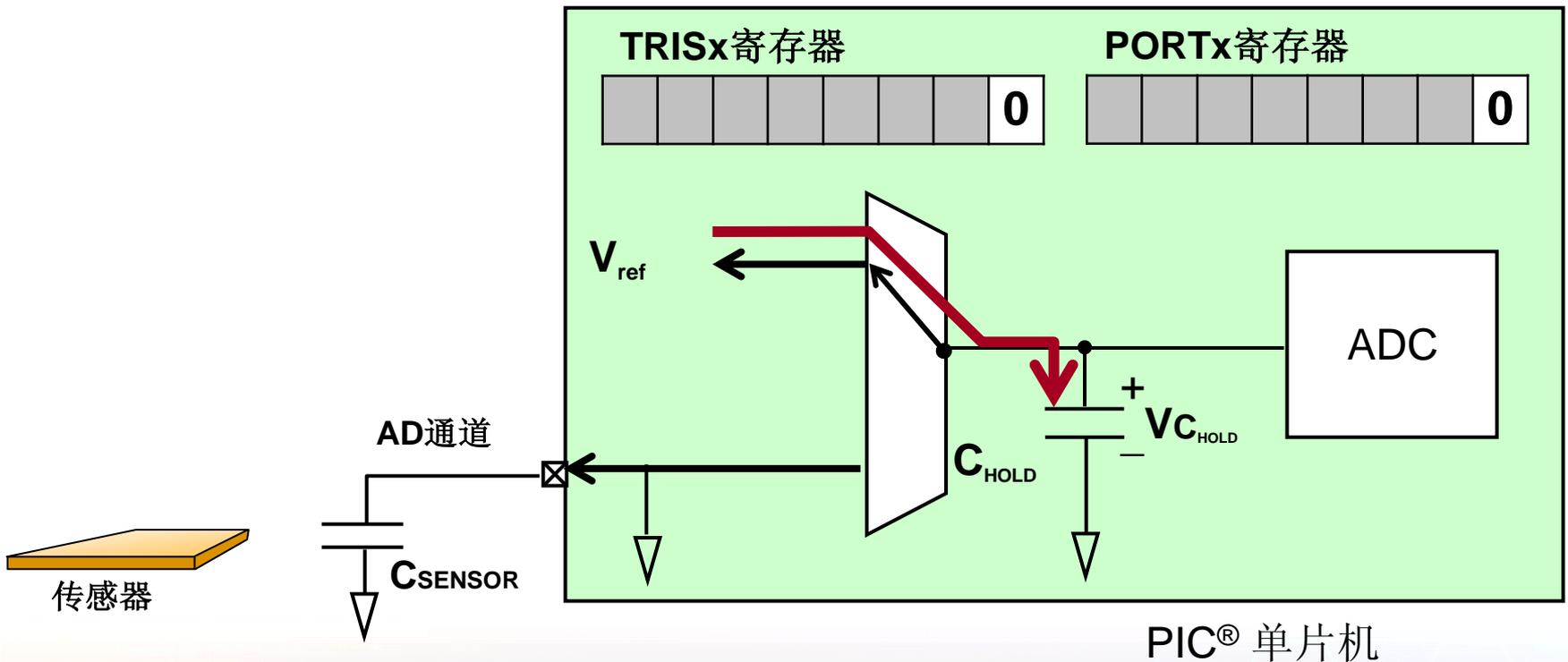
获取传感器读数

- 启动A/D转换



返回空闲状态

- C_{hold} 充电至 V_{ref}
- 传感器接地

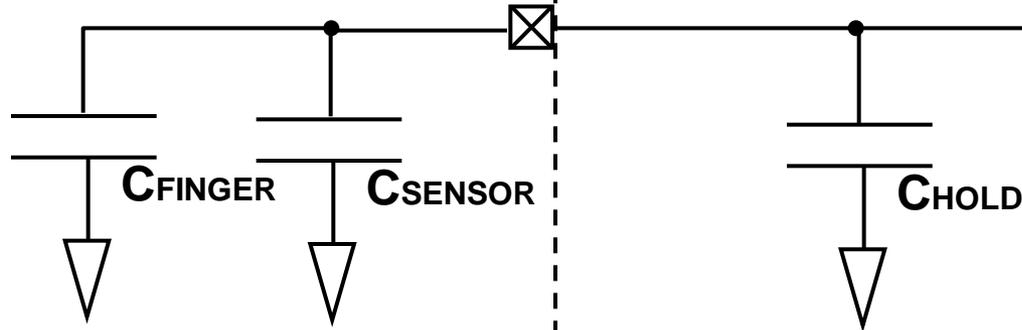
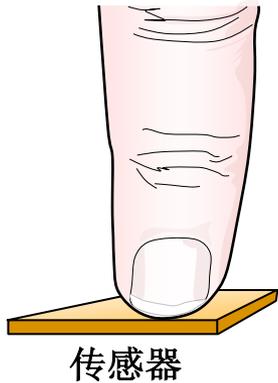
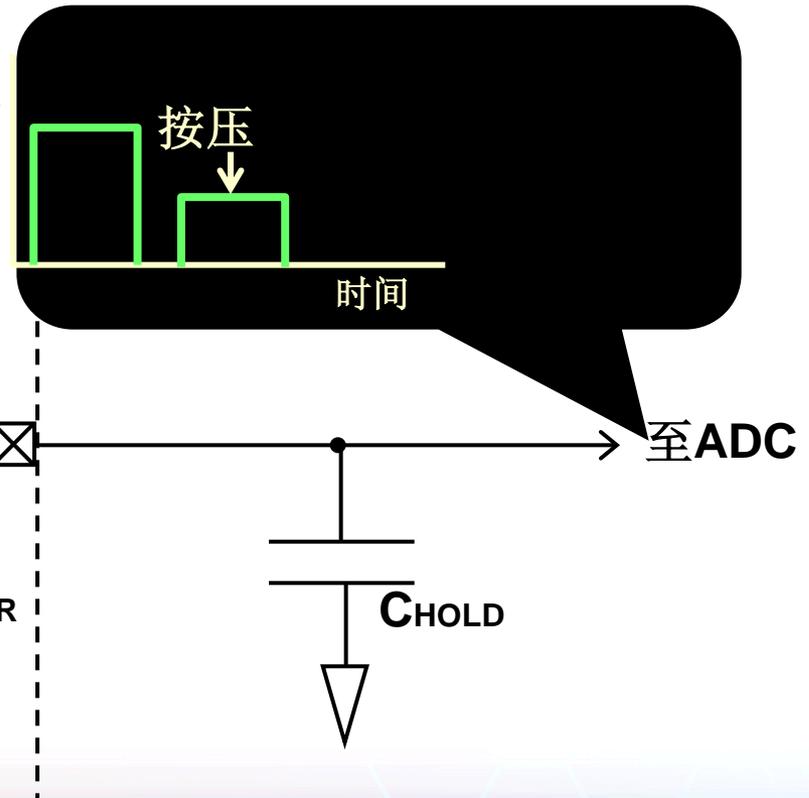


检测按压

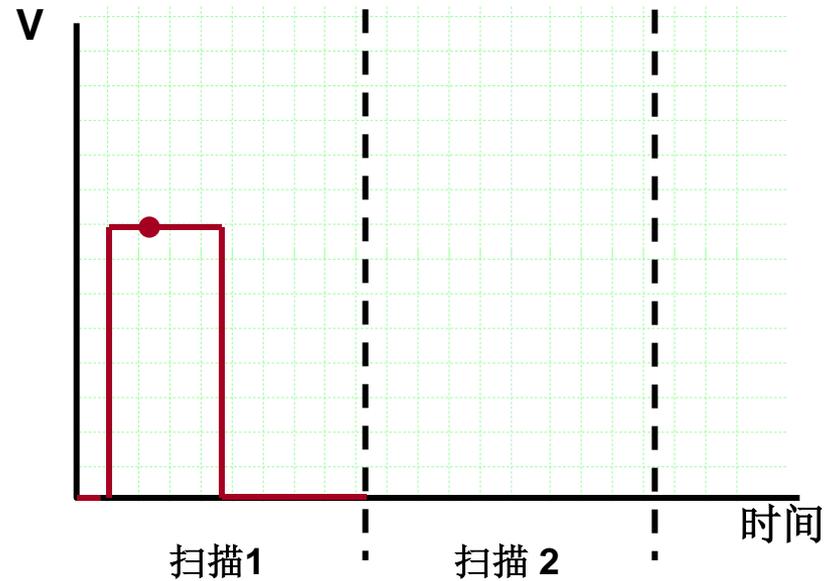
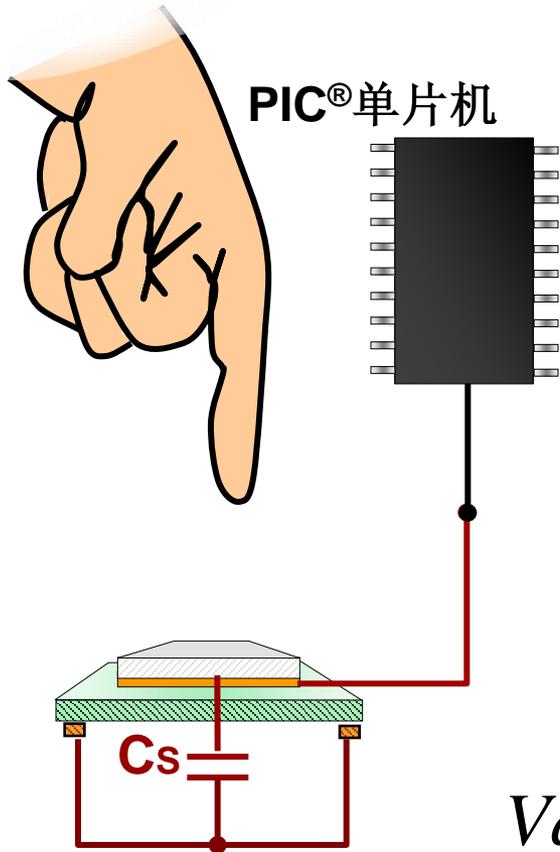
- 按压传感器增加了并联电容，从而降低了平均电压

$$V_{ADC} = \frac{V_{ref} * C_{HOLD}}{(C_{HOLD} + C_{SENSOR} + C_{FINGER})}$$

$V_{AVERAGE}$



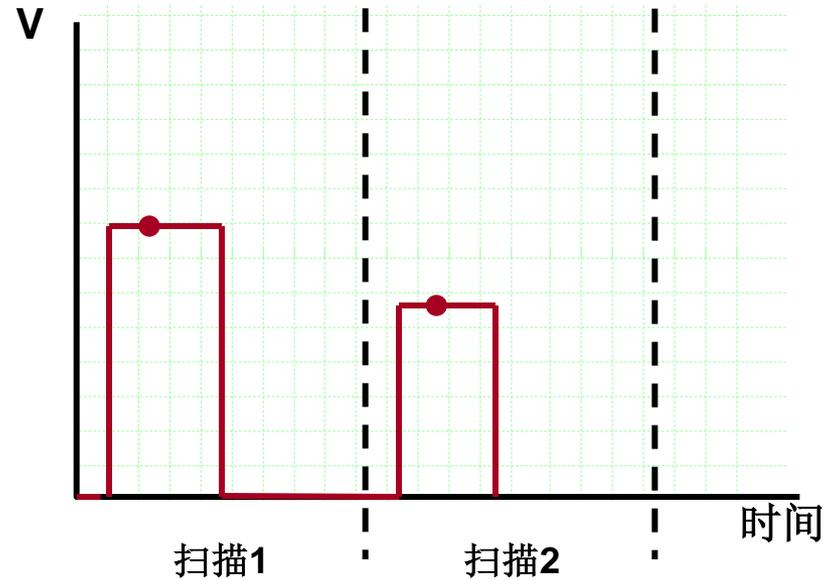
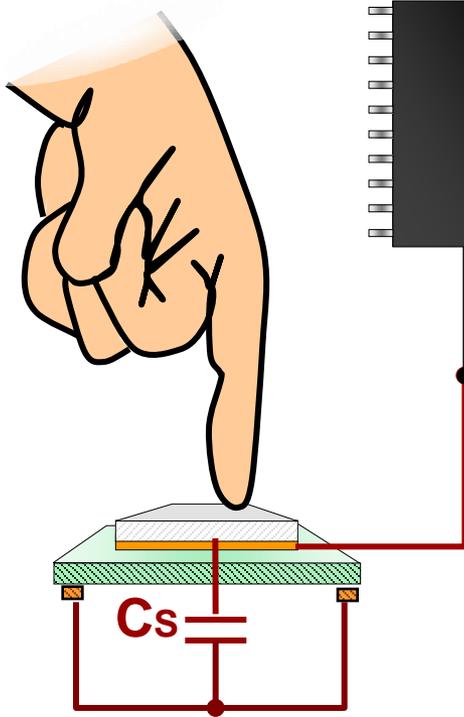
所需操作



$$V_{adc} = V_{ref} \frac{C_{hold}}{(C_{hold} + C_{sensor})}$$

所需操作

PIC® 单片机



$$V_{adc} = V_{dd} \frac{C_{hold}}{(C_{hold} + C_{sensor} + C_{finger})}$$



演示1: CVD波形



MICROCHIP 2010

MASTERS Conference

实验1a

设置并校验CVD硬件

实验1a 目标

1. 校验是否正确配置了MPLAB[®] IDE
和编译器
2. 校验CVD硬件是否能正确工作



MICROCHIP 2010

MASTERS Conference

mTouch™ 技术 软件架构概述

mTouch™ 技术 代码比较

● 软件架构

紧凑

- 代码尺寸小
- 最适合PIC12F / PIC16F器件

专用

- 分立式mTouch技术
- 对ISR的访问受到限制

● 库

灵活

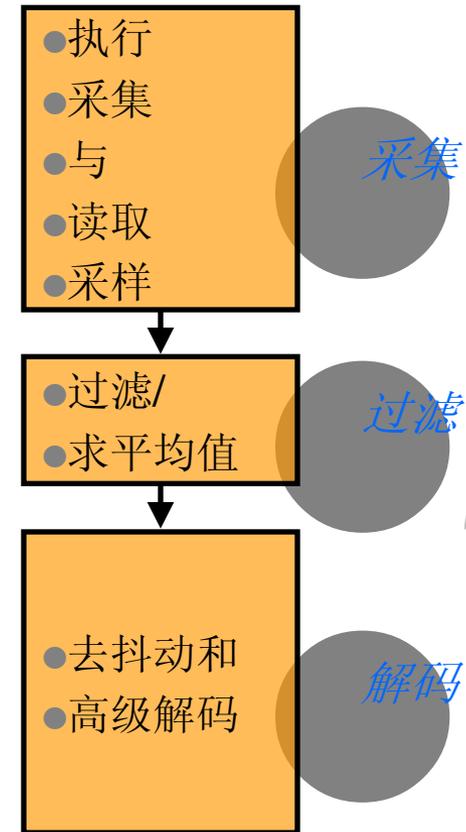
- 可定制传感器处理程序
- 最适合存储容量较大的器件系列

集成

- 与其他库和函数配合良好
- 例如：USB/图形/以太网库

mTouch™ 技术 软件架构

- 模块
 - 采集（特定于PIC® MCU）
 - 过滤
 - 解码
 - 调试通信
- 接口
 - `CVD_GetButtonState(index)`
 - `mTouchCVD_Init();`
 - `mTouchCVD_Service();`



mTouch™ 技术 软件架构

```
● void main(void)
● {
●     Init();
●
●     while(1)
●     {
●         mTouchCVD_Service();
●
●         switch(CVD_GetButtonState(0))
●         {
●             case CVD_PRESSED:    LED0 = LED_ON;    break;
●             default:             LED0 = LED_OFF;    break;
●         }
●
●         switch(CVD_GetButtonState(1))
●         {
●             case CVD_PRESSED:    LED1 = LED_ON;    break;
●             default:             LED1 = LED_OFF;    break;
●         }
●     }
● }
```

```
● void Init(void)
● {
●     mTouchCVD_Init();
● }
```

软件架构配置

mTouchCVD_Config.h

- 定义“有效传感器”通道
- 指定扫描速率
- 启用/禁用代码功能



MICROCHIP 2010

MASTERS Conference

实验1b

实验1b 目标

1. 分析软件架构
2. 使用软件架构来创建有效的电容触摸传感应用

实验1b

快速创建项目的步骤

1. 复制一个现有的项目
2. 更改config文件进行设置
3. 编程并尝试不同的代码功能

详细说明请参见“实验手册”中的“实验1b”



MICROCHIP 2010

MASTERS Conference

实验1c

实验1c目标

1. 继续用您的软件架构应用程序进行实验
2. 观察在电容触摸传感器上覆盖不同厚度材料时的效果
3. 针对系统中新的覆盖材料调节配置选项

实验1c

快速创建项目的步骤

1. 打开“Lab 1c”项目
2. 校验代码是否能正常运行
3. 在电容传感器上放置覆盖层
4. 修改代码以适应传感器的覆盖层

详细说明请参见“实验手册”中的“实验1c”



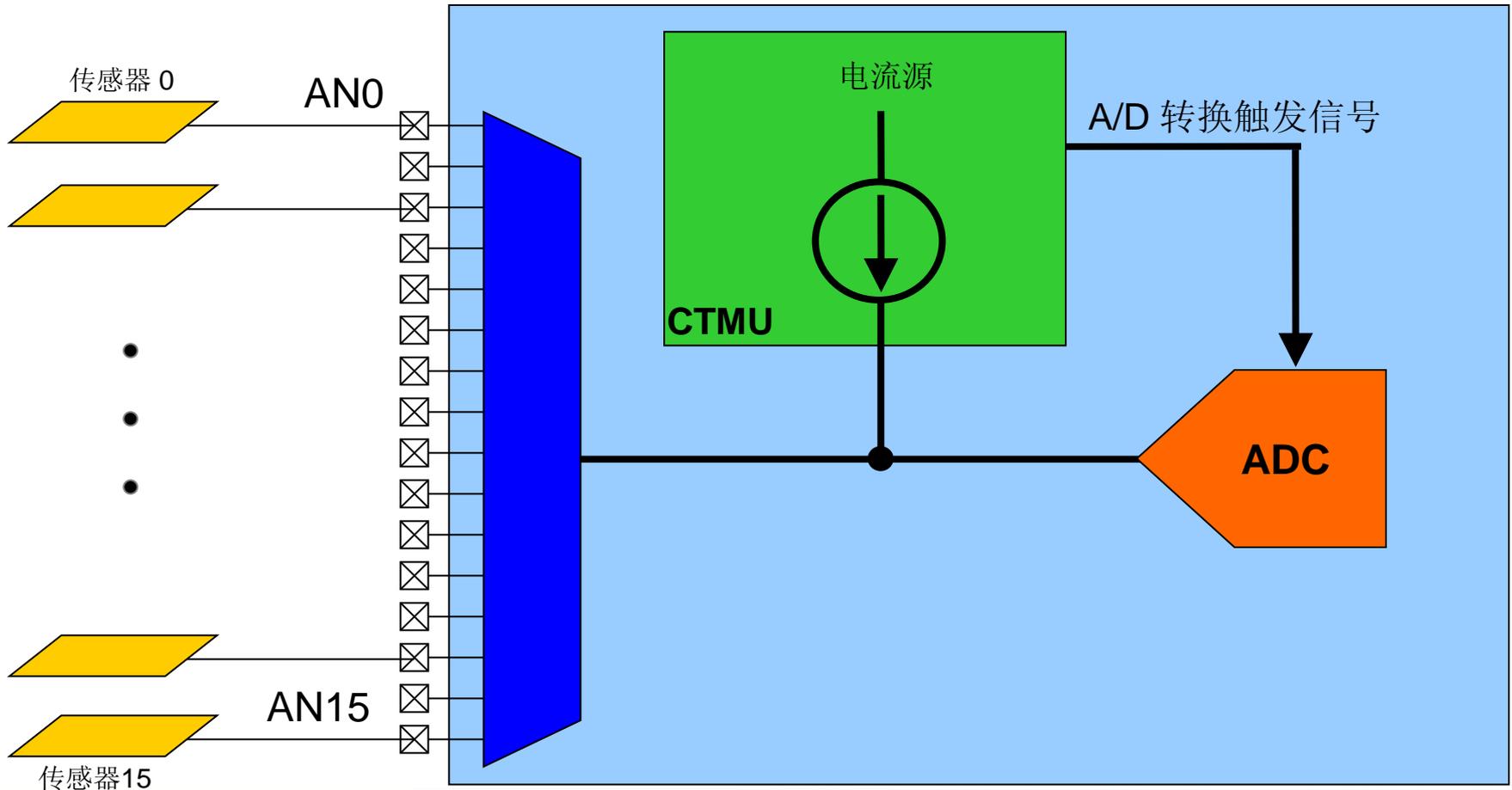
MICROCHIP 2010

MASTERS Conference

CTMU的工作原理

CTMU

- 同样地只要往传感器焊盘处引一条走线



CTMU用于电容触摸 的工作原理

EE101基本原理:

- 电容的瞬时电流:

$$i = C \cdot dV/dt$$

- 如果*i* = 恒流, 则

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

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

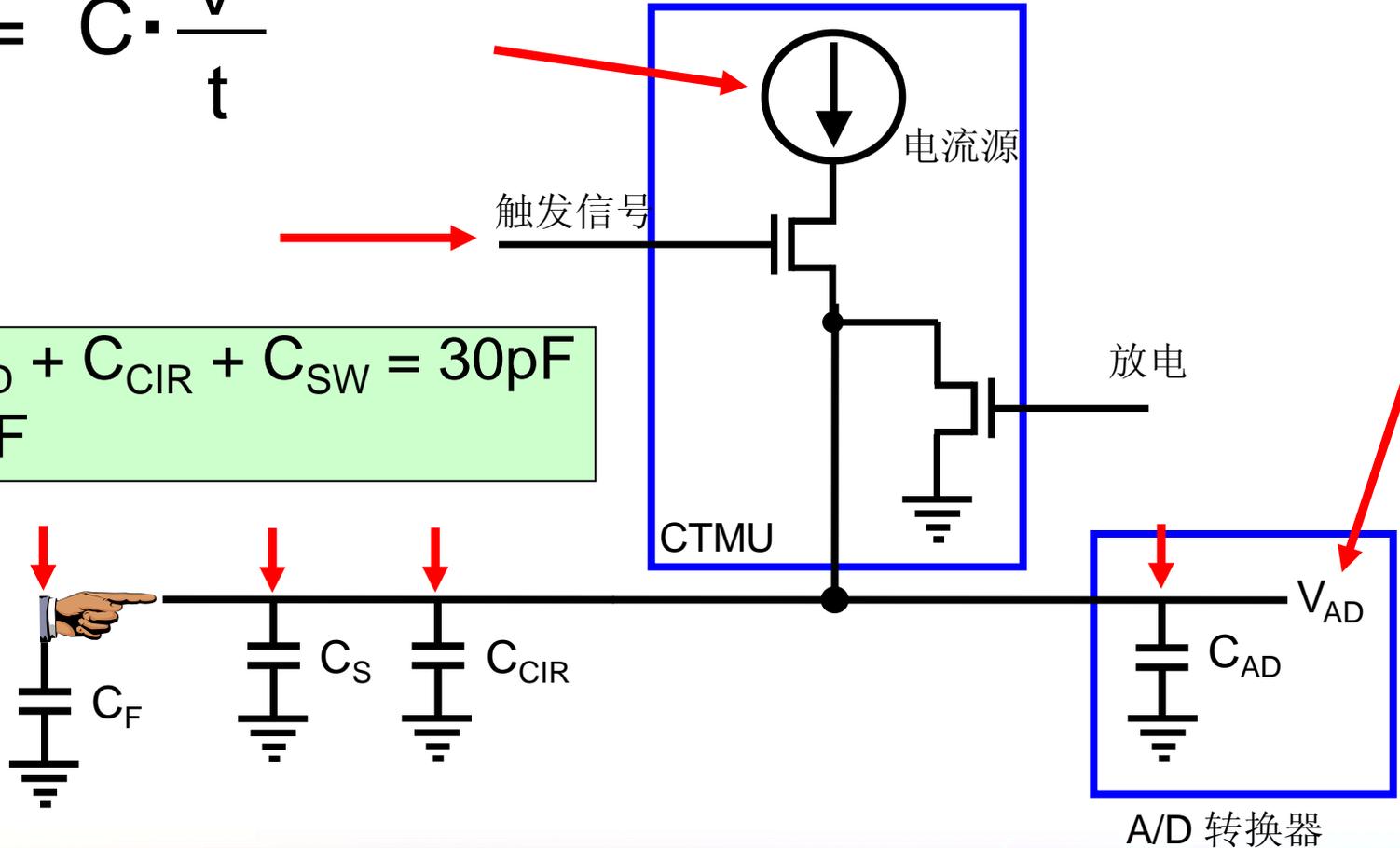
- 如果*I*和*t*保持恒定, 当 *C*增加时, *V*必然降低

CTMU触摸电路元件

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

$$C_S = C_{AD} + C_{CIR} + C_{SW} = 30\text{pF}$$

$$C_F = 7\text{pF}$$



示例计算

$$I = C \cdot \frac{V}{t} \longrightarrow \frac{I \cdot t}{C} = V$$

未触摸时

- $I = 5.5\mu\text{A}$
- $t = 10\mu\text{S}$
- $C_S = 30\text{pF}$

$$\longrightarrow V = 1.833$$

触摸时

- $I = 5.5\mu\text{A}$
- $t = 10\mu\text{S}$
- $C = C_S + C_F = 37\text{pF}$

$$\longrightarrow V = 1.486$$

$$C_S = C_{AD} + C_{CIR} + C_{SW} = 30\text{pF}$$
$$C_F = 7\text{pF}$$

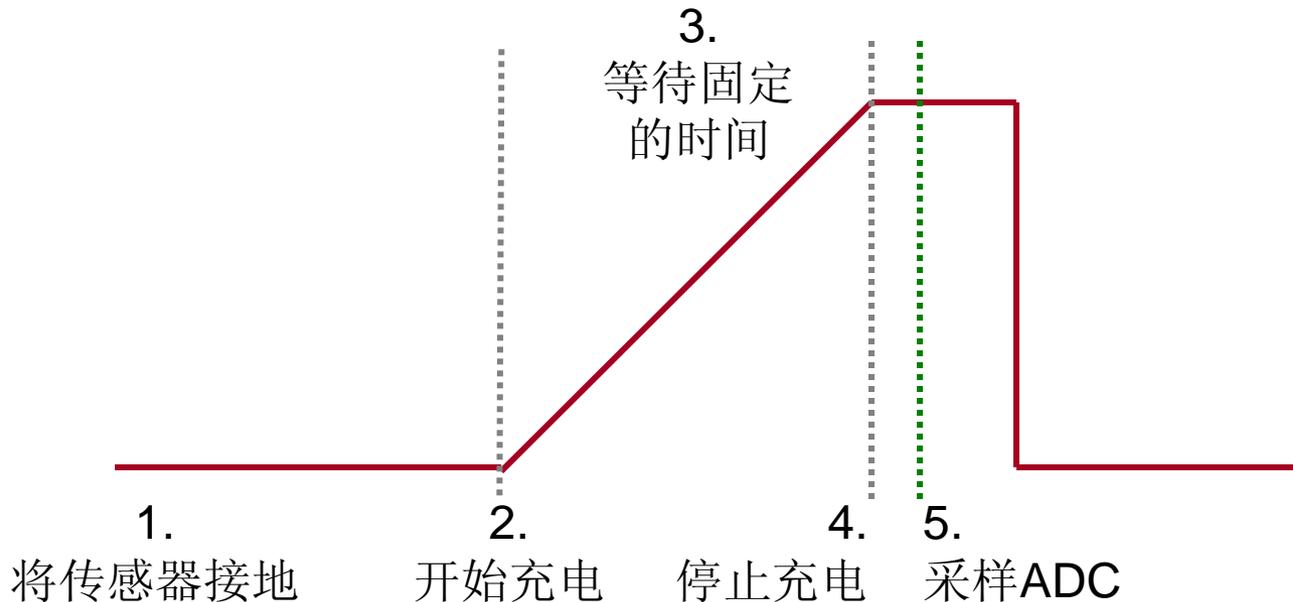
CTMU扫描

扫描步骤

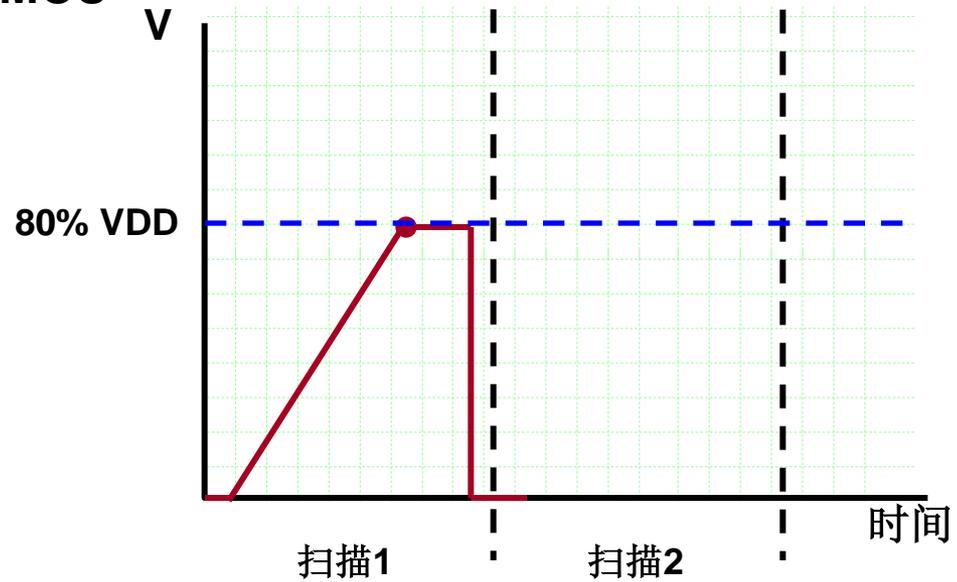
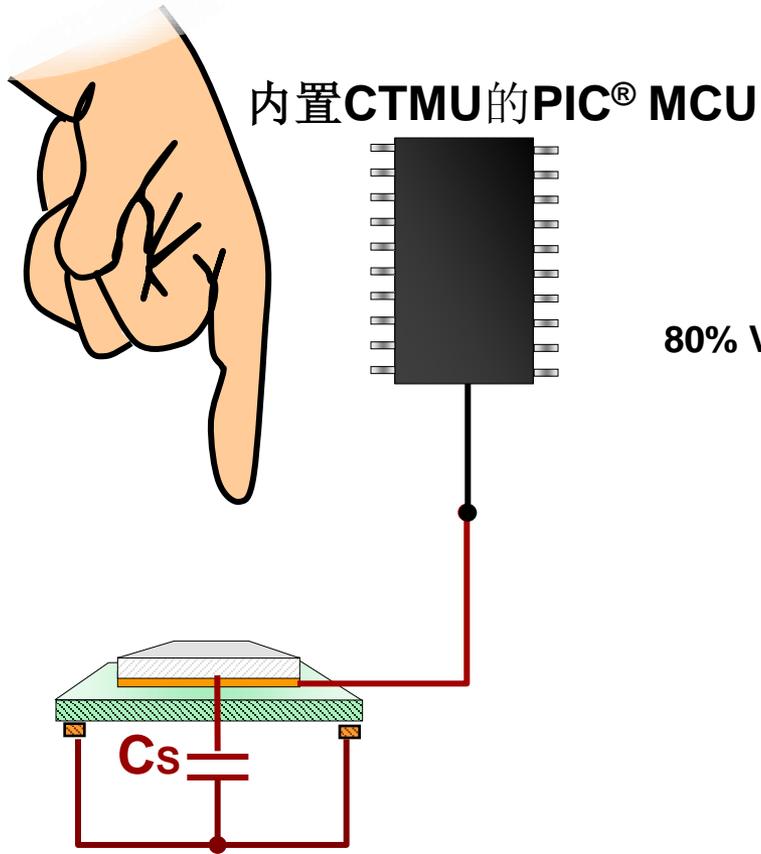
1. 使传感器放电，确保从0V开始充电
2. 启动CTMU恒流源给传感器充电
3. 等待一段固定的时间
4. 通过关断恒流源使CTMU停止对传感器充电
5. 读ADC

所需操作

利用**CTMU**对传感器进行充电，
利用**ADC**获取传感器读数



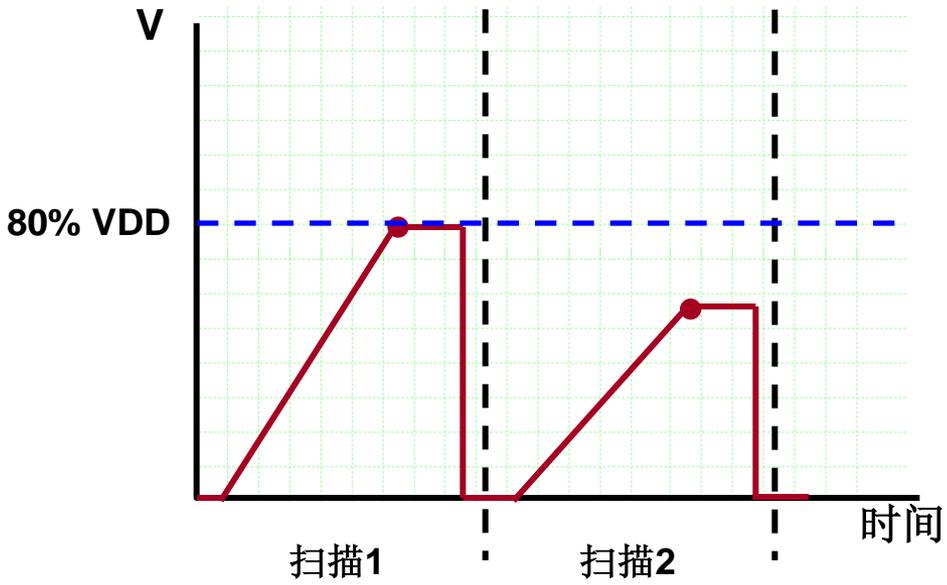
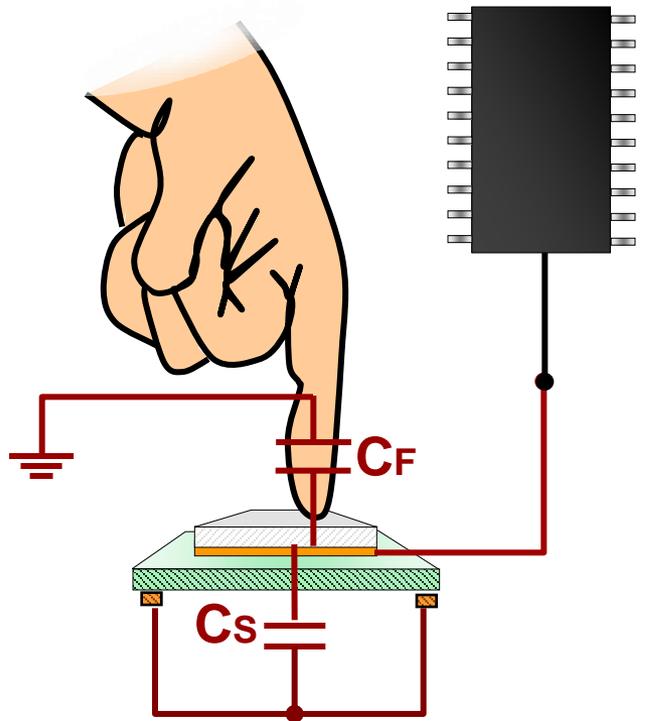
所需操作



$$V_{adc} = \frac{I * T}{(C_{sensor})}$$

所需操作

内置CTMU的PIC® MCU



$$V_{adc} = \frac{I * T}{(C_{sensor} + C_{finger})}$$

演示3： CTMU波形



MICROCHIP 2010

MASTERS Conference

实验2a 设置和校验CTMU硬件

实验2a 目标

1. 校验是否正确配置了MPLAB® IDE
和编译器
2. 校验CTMU硬件是否能正确工作

实验2a 说明

- 完成实验1a 所需的所有信息均位于
《**C11H01 CTH mTouch™ 电容触
摸传感解决方案**》实验手册中



MICROCHIP 2010

MASTERS Conference

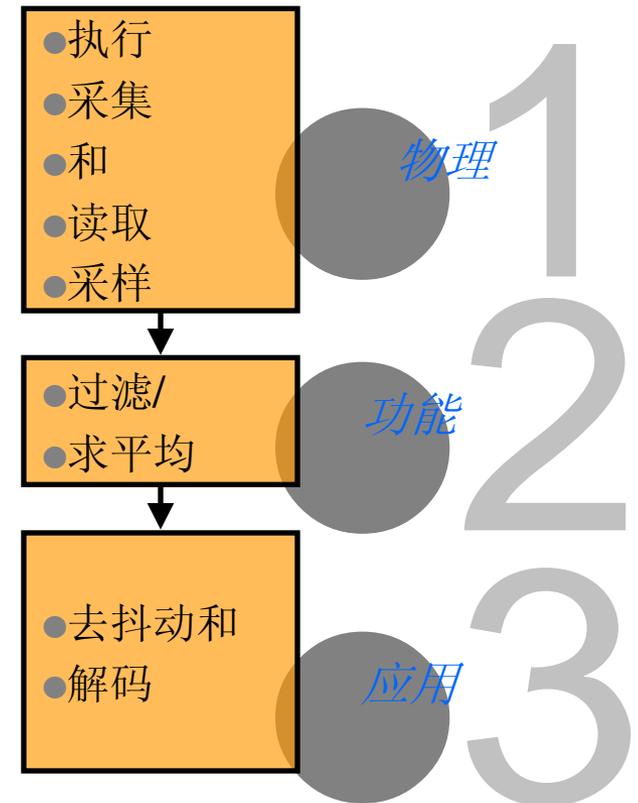
mTouch™ 技术 库概述

● 层

- 物理层（特定于PIC® MCU）
- 功能层
- 应用层

● API接口

- 14个函数调用
- 示例：
 - mTouchCapAPI_GetStatusDirectButton
 - mTouchCapAPI_SetUpChannelDirectKey
 - mTouchCapAPI_AutoAdjustChannel



```
● DirectKey myDirectKey1
● void main(void)
● {
●     mTouchCapAPI_SetUpChannelDirectKey(&myDirectKey1,
●         DIRECTKEY1, CHANNEL, DEFAULT_TRIP_VALUE, DECODE_METHOD_PRESS_ASSERT,
●         FILTER_METHOD_SLOW_AVERAGE);

●     mTouchCap_Init();

●     while(1)
●     {
●         if(dataReadyCTMU == 1)
●         {
●             dataReadyCTMU = 0;
●             disableCapTouchTimer();
●             if(mTouchCapAPI_GetStatusDirectButton(&myDirectKey1)
●             {
●                 //Key Pressed - Execute Code Here
●             }
●             enableCapTouchTimer();
●         }
●     }
● }
```



MICROCHIP 2010

MASTERS Conference

实验2b

实验2b 目标

1. 分析mTouch™技术库
2. 利用mTouch技术库使用CTMU创建有效的电容触摸传感应用

实验2b

快速创建项目的步骤

1. 打开“**Lab 2b**”项目
2. 添加所需代码以创建您的应用程序

详细信息请参见“实验手册”中的
“**实验2b**”

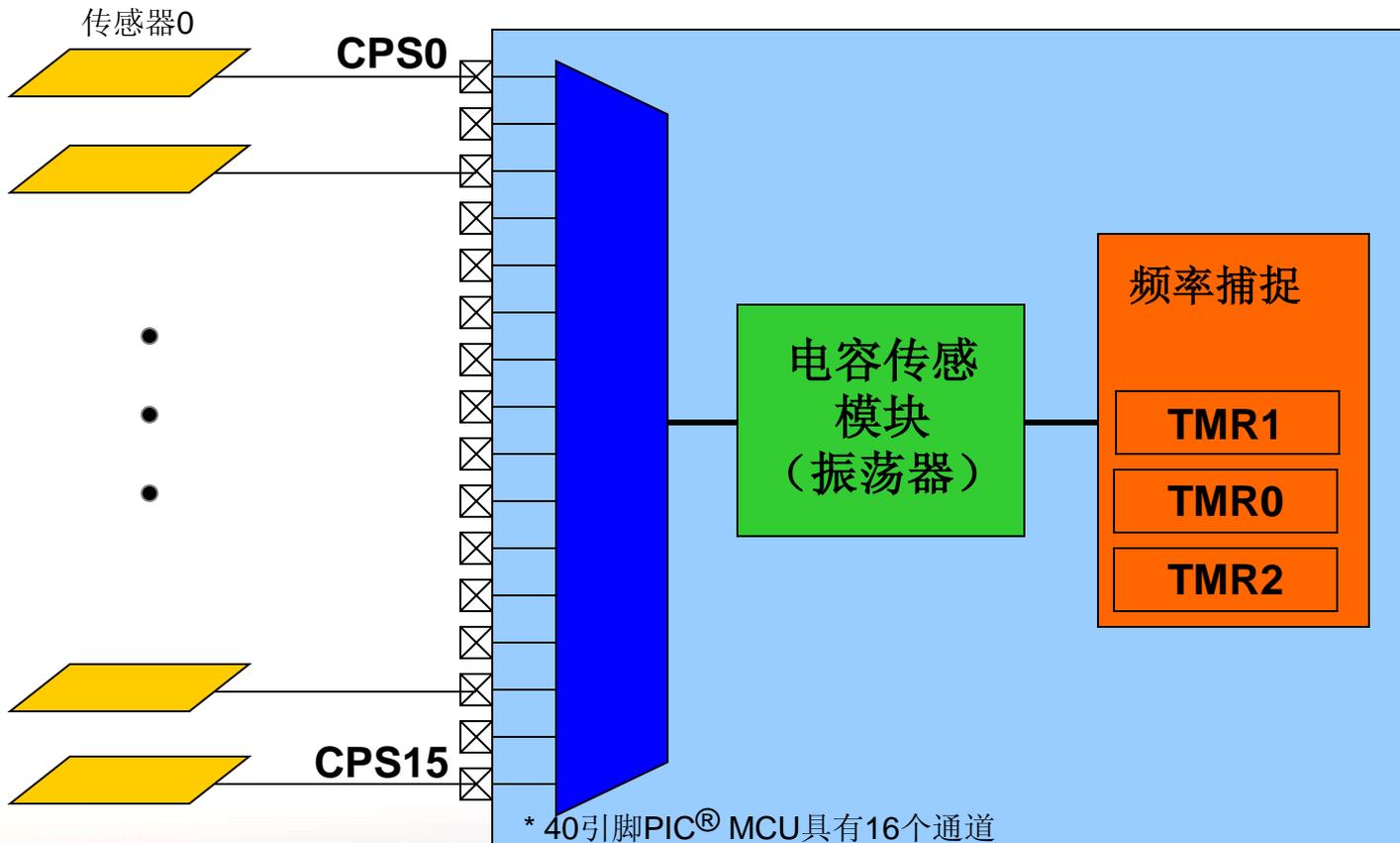


MICROCHIP 2010

MASTERS Conference

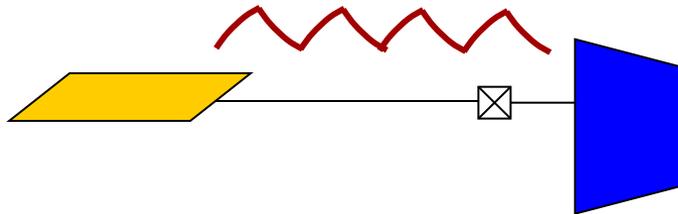
CSM的工作原理

- 又是同样，只要往传感器焊盘处引一条走线！

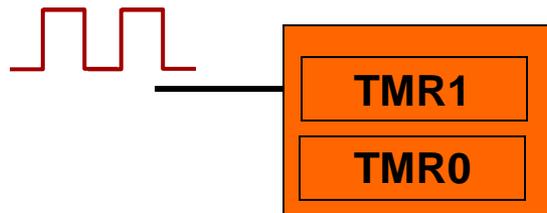


● 初始化目标

1. 创建张弛振荡器

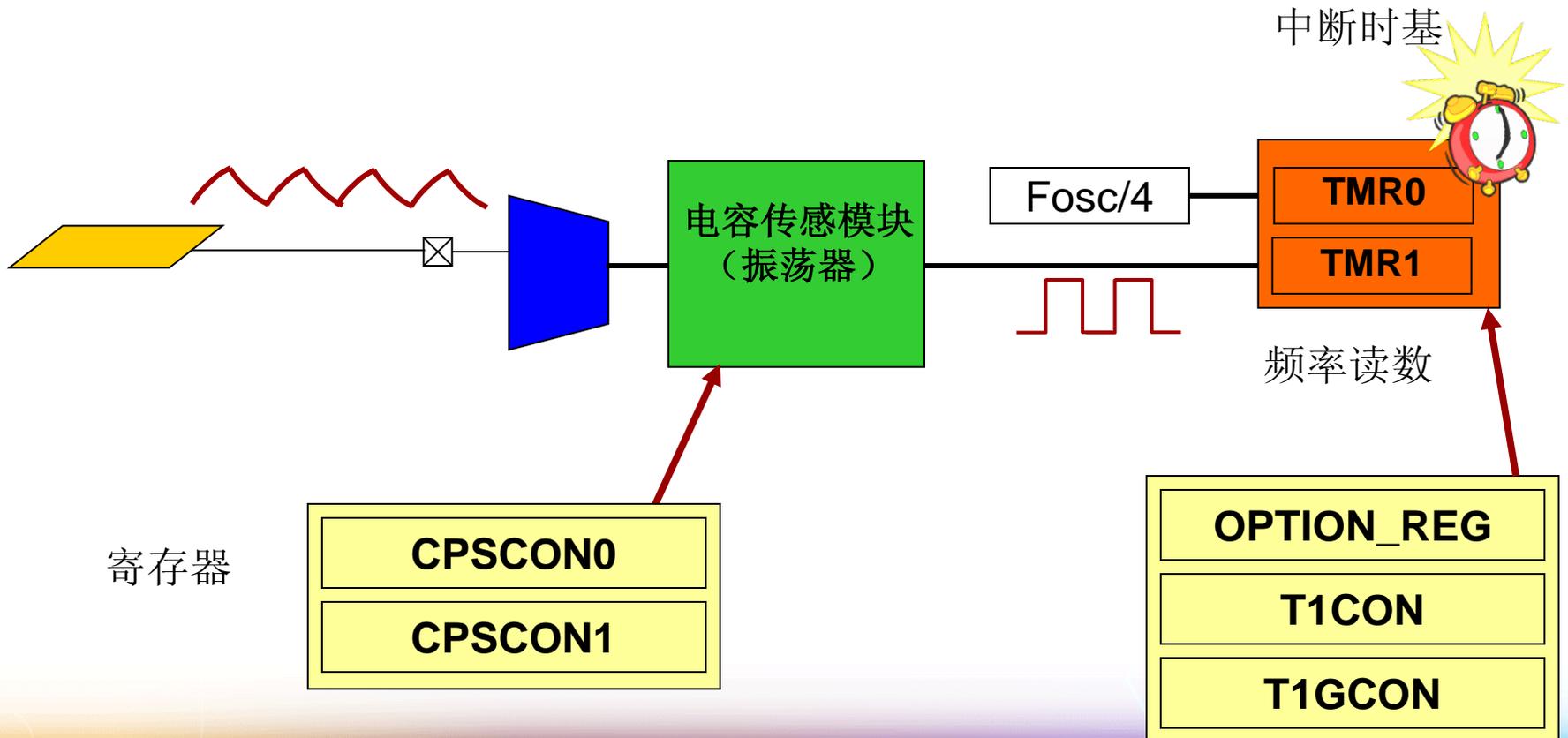


2. 配置以测量频率

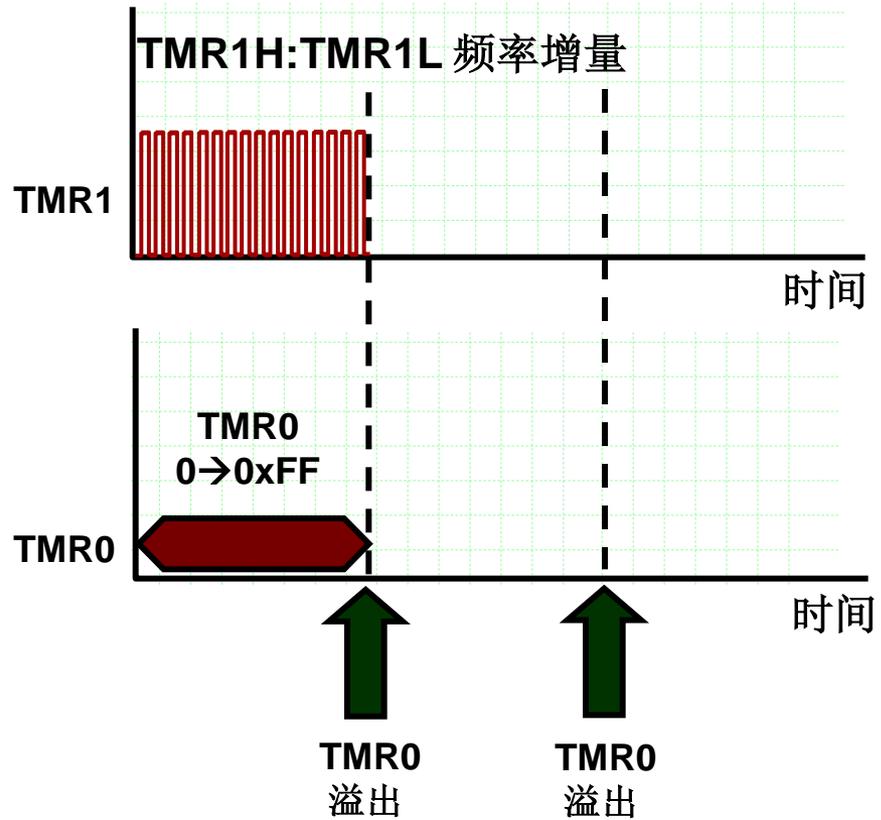
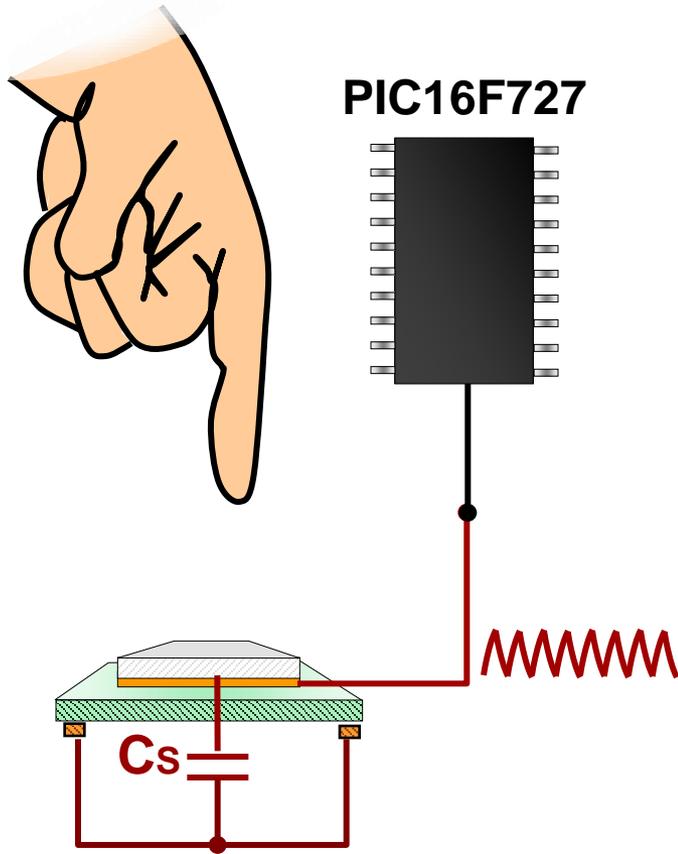


CSM设置

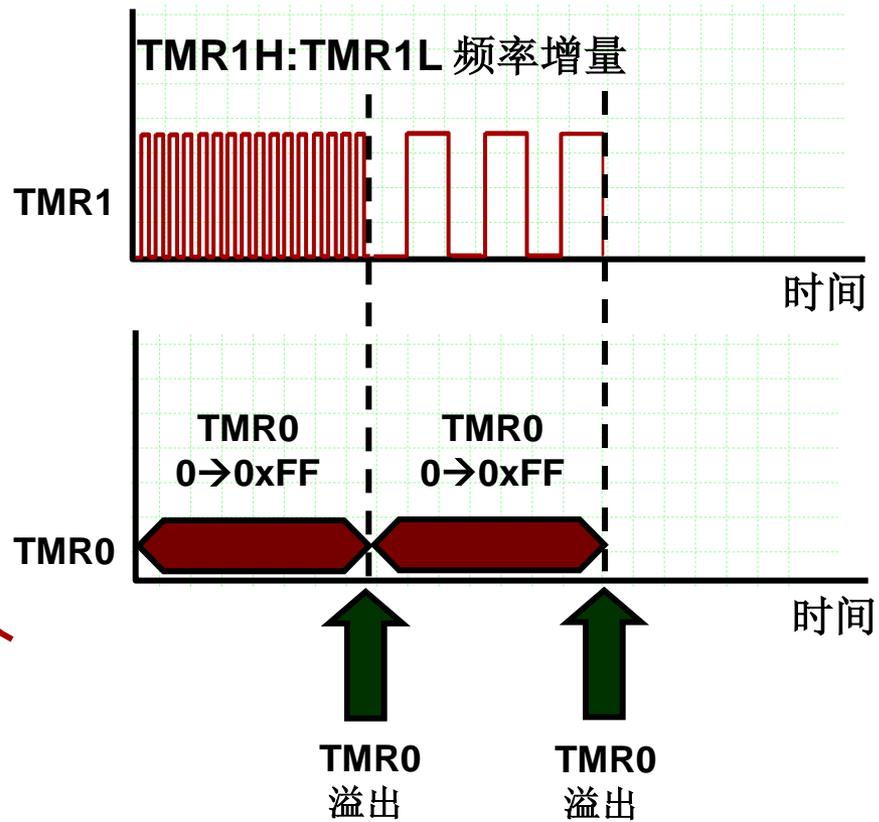
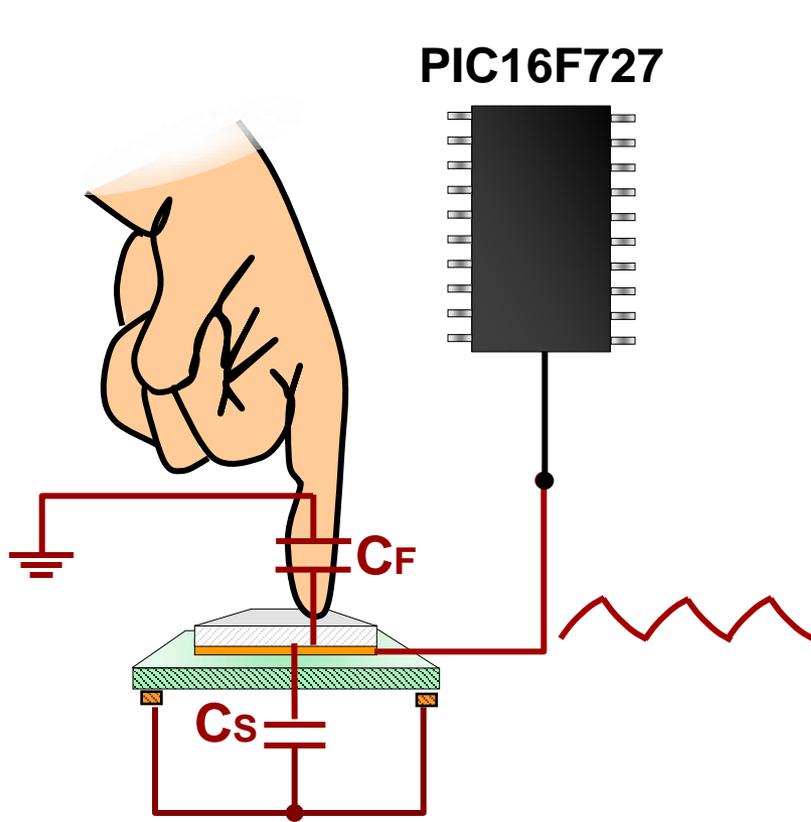
配置以测量频率



所需操作

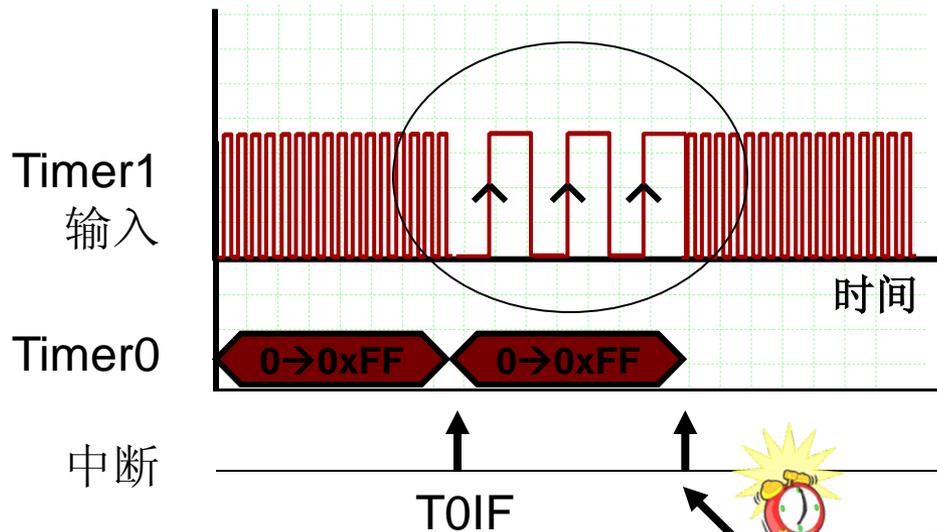


所需操作



频率慢下来

获取读数



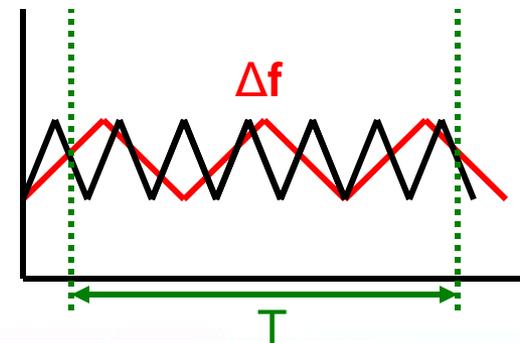
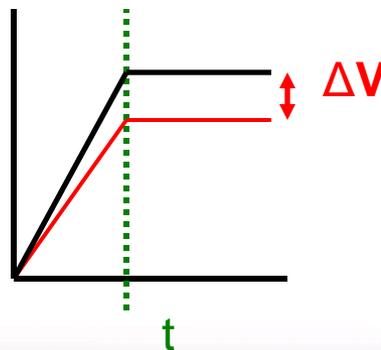
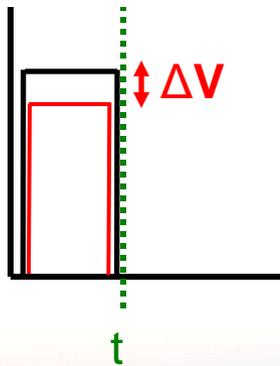
开始获取读数!

```
读数 = TMR1H:TMR1L;  
(读数 = 3)
```

Microchip电容触摸传感总结

- 所有解决方案都是基于测量固定时间段内的 Δf 或 ΔV 实现的
- 这两种情形下，额外引入的 C_F 均会引起 C_{TOT} 改变，从而改变待测物理量（ Δf 或 ΔV ）

$$C_S + C_F = C_{TOT}$$





MICROCHIP 2010

MASTERS Conference

3种方法的比较

Microchip 3种方法的比较

- 器件选择是首要因素
- 中档8位PIC[®] MCU —— 专用或小型系统的绝佳集成选择（PWM和I²C[™]/SPI等）
- 高端8位单片机或入门级16位单片机 —— 完整系统的绝佳集成选择（USB和TCP/IP等）

Microchip 3种方法的优缺点

		CVD	CTMU	CSM/ 比较器
以下各条件下，系统对噪声的稳定性：				
	应用采用不可充电电池	✓	✓	✓
	应用由交流电源供电	✓	✓	-
	应用由USB供电	✓	✓	-
	应用具有USB、CAN、以太网或RF连接功能	✓	✓	-
	响应时间	更快	最快	快
	接近传感	✓	✓	✓
	低功耗（传感方法）	✓	✓	✓



MICROCHIP 2010

MASTERs Conference

软件

软件的3个部分

- 采集 —— 这部分用于处理读取传感器的硬件（3种方法 – CVD、CTMU和CSM）
- 处理 —— 这部分用于优化采集过程获得的原始数据的性能
- 解码 —— 这部分用作与应用程序软件其他部分的软件接口



MICROCHIP 2010

MASTERS Conference

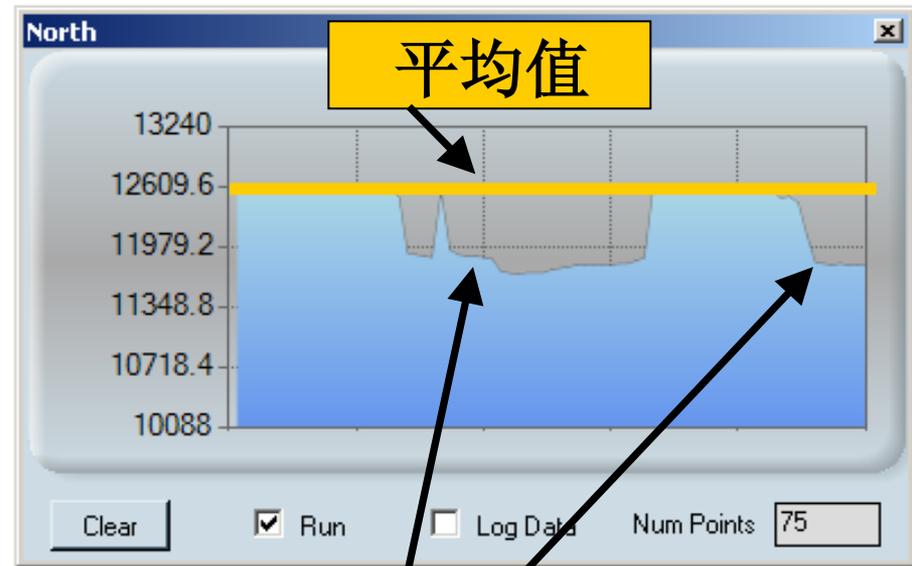
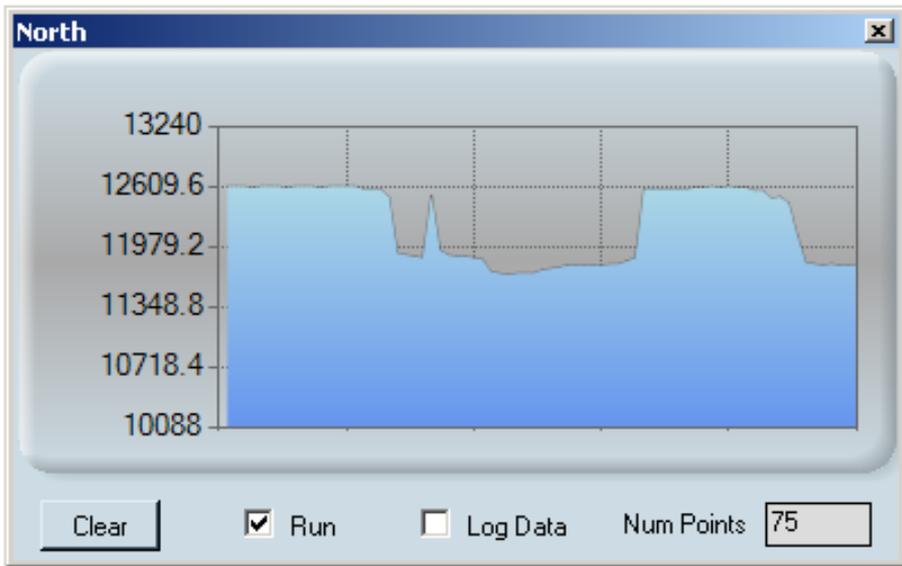
软件 处理/过滤

处理和过滤时的基本假设

- 传感器未触摸时的读数稳定
- 采用一个基准值作为参考
- 读数与基准值偏离表明发生了触摸

平均值

平均值 会跟踪环境的变化; 它可以作为因触摸产生偏离的参考值。平均值是电容触摸传感器未被触摸时的基准值



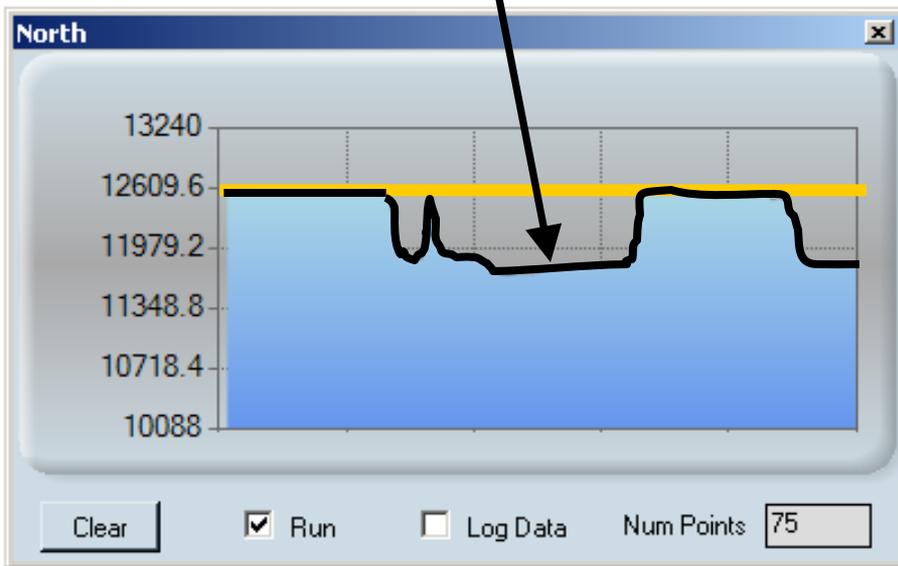
发生触摸

处理读数

- 读数 = 频率, ADC值

原始数据

阈值

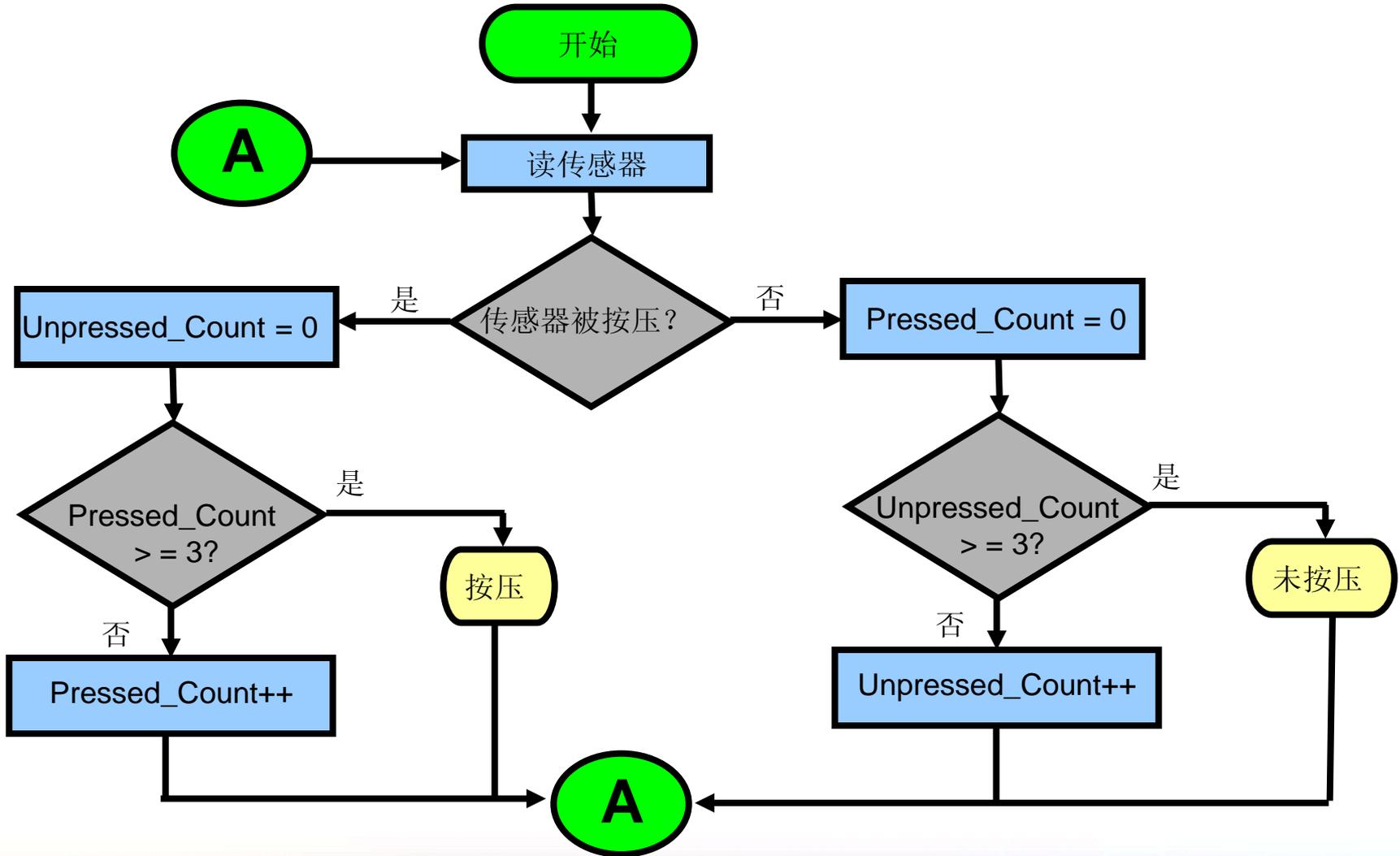


更多软件处理

- 处理原始数据的其他软件示例包括：
 - 过采样
 - 过滤
 - 去抖动

所有这些算法都可用来增加采集软件收集到的原始数据的完整性

软件去抖动程序



为什么需要更多处理？

原始假设：

传感器未被触摸时读数稳定

采用基准值作为参考

传感器读数与基准
值偏离表明发生了
触摸

实际情况：

所有电容触摸系统中均存在
一些噪声

需要进行处理来确定一
个有效的参考值

并非必然情况。有时噪声
尖峰可能会被检测为一次
电容触摸按压



MICROCHIP 2010

MASTERS Conference

软件 —— 解码

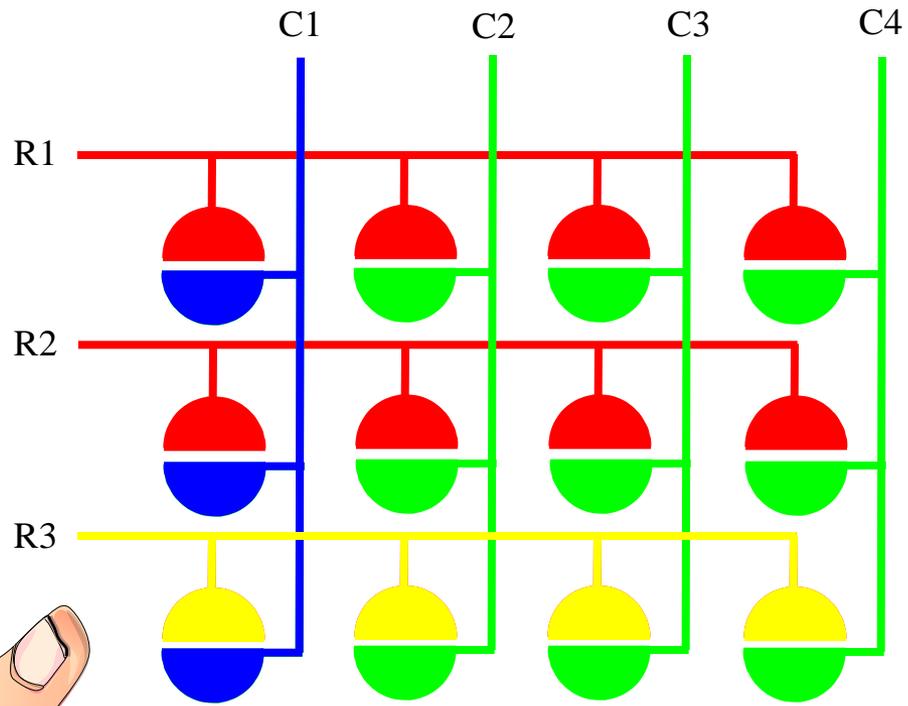
什么是软件解码？

- 包括一些算法，这些算法会根据电容触摸传感器的状态产生某些操作
 - 多点按压（诸如矩阵式按键）
 - 翻转按压（第一次按压 = 导通，第二次按压 = 关断）
 - 确定指尖（滑动条）的位置

所有解码软件的目的均是针对所处理电容触摸焊盘的“状态”确定要执行的操作

软件解码示例： 矩阵通道

- 一个 a 行 b 列的网格需要 $(a + b)$ 个通道，但可实现 $(a \times b)$ 个按键
- 在扫描了所有的行和列以后，由软件确定按键是否被按压
- 需要的高速扫描（尤其是较大的矩阵）



问：矩阵式传感器通道的最优形式是什么？

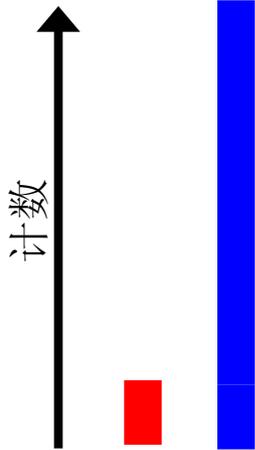
答：是行数和列数相等。



软件解码示例 滑动条

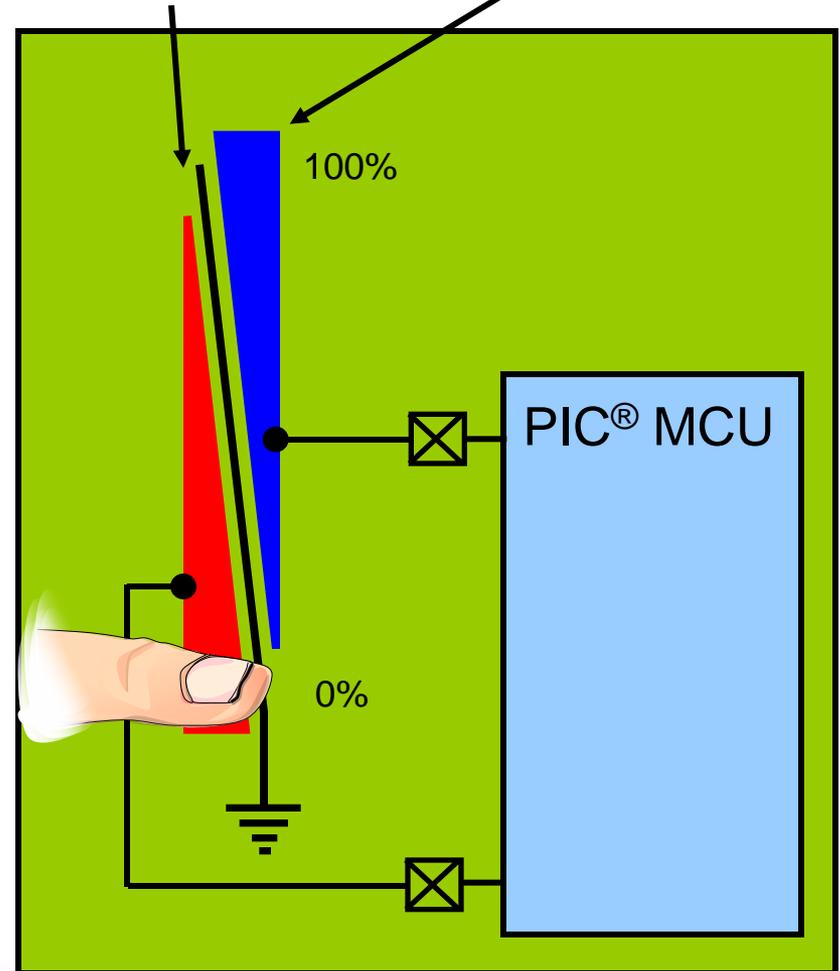
- 需要2个通道
- 基本方程:
- 右传感器
 - $\% = 100 \times (R/(L+R))$
- 左传感器
 - $\% = 100 \times (1-(L/(L+R)))$

其中 **L** 和 **R** 是相对未按压时电容的增量



左传感器 原始值 右传感器 原始值

PCB板上的三角形铜焊盘





MICROCHIP 2010

MASTERS Conference

mTouch™ 技术 软件架构和 mTouch 库

软件入手点

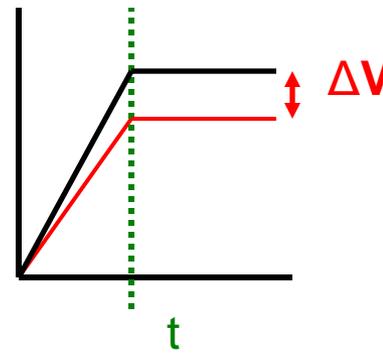
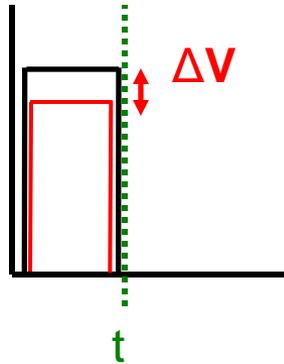
mTouch™技术 软件架构和库

- 包含执行以下操作的代码：
 - 配置
 - 扫描
 - 解码

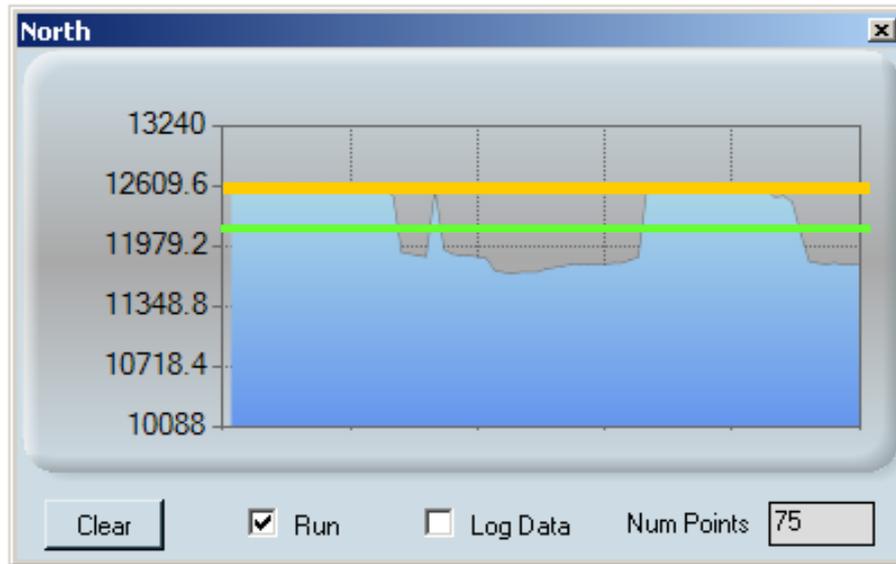
- 显著提高了开发速度！

mTouch™ 技术 软件架构和库

- 软件架构/库可助您：
 - 简化硬件的实现



- 软件架构/库可助您：
 - 进行信号处理



- 软件架构/库可助您：
对数据进行解码以判断按键是否被按压

- 简单函数调用返回1或0:

```
if(KEY_PRESSED == mTouchCapAPI_GetStatusDirectButton(&anyKey))  
{  
    // Key "anyKey" is pressed, code here determines action to take  
}
```

(example is from CTMU mTouch library)

输入文件或系统配置文件

- 配置活动通道
- 针对按键进行配置*
- 输入阈值
- 根据需要填充相应选项

(*库可以额外配置矩阵按键和滑动条)

扫描和解码

- 作为**API**提供给用户
- 用户通过检查函数以获取传感器的状态

示例：

```
mTouchCap_ScanSensor();
```

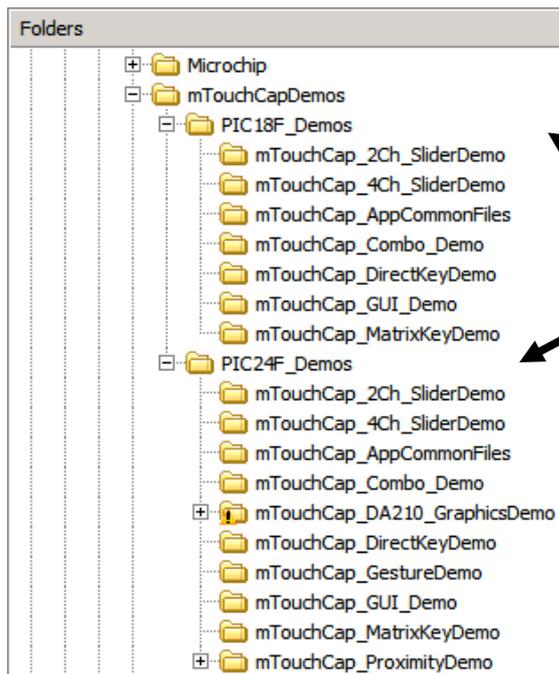
```
mTouchCap_Decode();
```

```
mTouchCap_GetStatusCh1();
```

要使工作时间最短

从现有项目着手

- Microchip提供了许多演示项目来展示软件架构和库



mTouch™技术库具有分别
演示按钮、矩阵按键和
滑动条触摸的软件示例

为什么要使用软件架构和库？

- 软件架构：

- 代码尺寸设计紧凑
- 代码在大多数情况下专用于电容触摸应用
- 需要最少的资源（RAM和外设）
- 最适合中档8位PIC[®]单片机

- 库：

- 灵活性更大
- 允许与Microchip的其他库和外设函数集成
- 需要更多资源（RAM和程序存储器）
- 最适合高端8位（PIC18F）和16位（PIC24F）PIC 单片机

针对PIC[®] MCU的 mTouch[™] 技术软件架构/库

- **2个Microchip mTouch软件资源:**

CVD软件架构 (PIC16F)

可从以下地址下载:

www.microchip.com/mtouch

CTMU库 (PIC18F/24F)

可从以下地址下载:

www.microchip.com/mal



MICROCHIP 2010

MASTERS Conference

总结

总结

- 电容触摸传感器的物理属性
- **Microchip** 电容触摸传感方法的硬件工作原理
- 用于电容触摸应用的软件
- **mTouch™** 电容触摸软件方法和 **mTouch** 库概述

问题?

有关触摸传感的其他课程

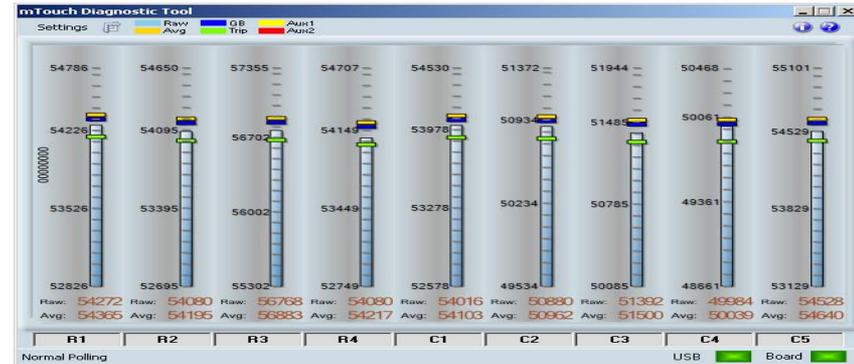
- **C11L03 PCT – Fred Fu和Tom Xue**
介绍Microchip的触摸屏技术
- **C11L09 PTM – Bob Li和Jerry Gao**
使用CTMU实现精确时间测量和传感应用

参考资料

- www.microchip.com/mtouch
- **Capacitive Sensors**
作者: Larry K. Baxter
ISBN 0-7803-5351-X
- **AN1101、AN1102、AN1103和AN1104** —— 涵盖基本的电容触摸传感
- **AN1171** —— 采用**CSM**模块实现电容触摸传感
- **AN1202** —— 采用**PIC10F206**实现电容触摸传感
- **AN1250** —— 采用**CTMU**实现电容触摸传感
- **AN1268** —— 采用**CSM**通道定期测量实现电容触摸传感
- 网上研讨会:
www.microchip.com/webinars

mTouch™ 诊断工具

- 定制电容触摸传感解决方案的**GUI**
- 用于MPLAB® IDE的一部分

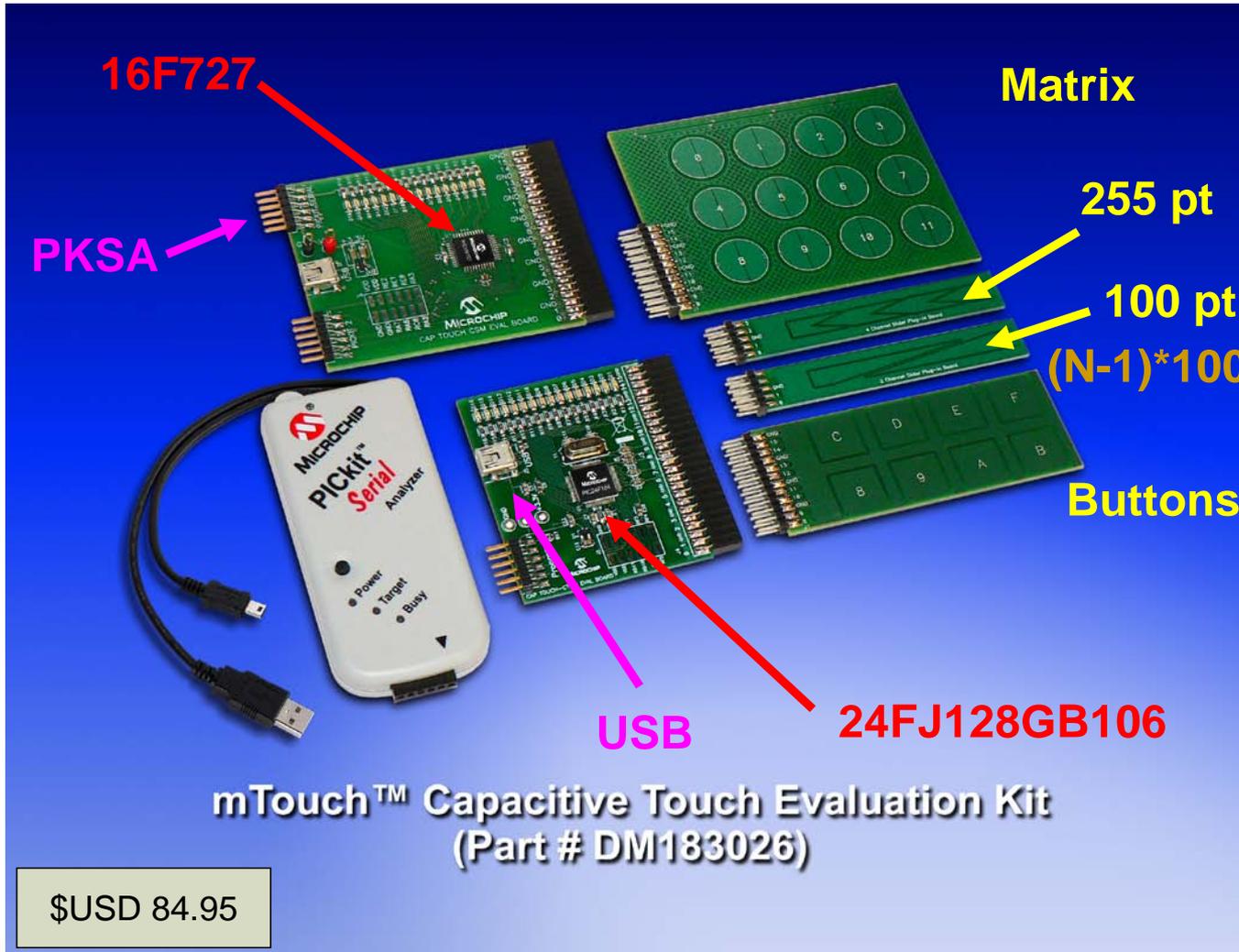


PICDEM™ 触摸传感2 开发板

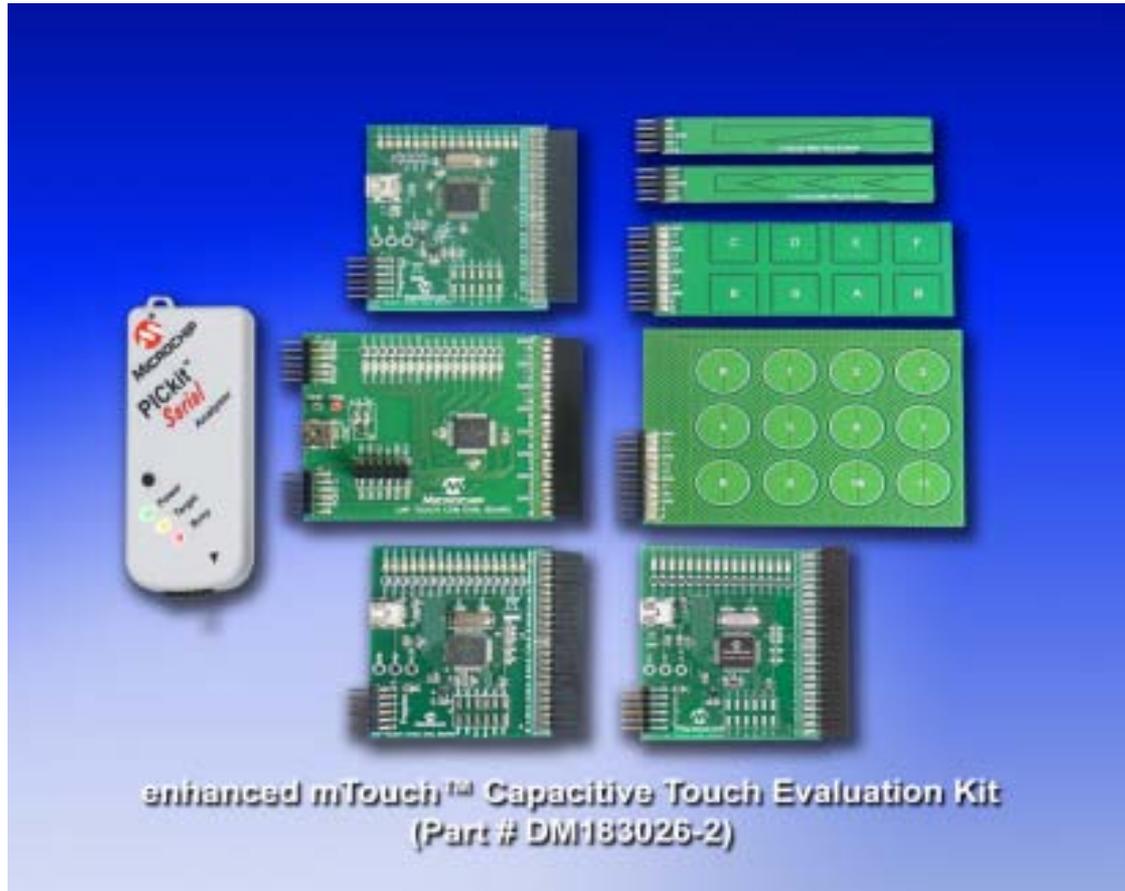
- 基于PIC24F的演示板
DM164128



DM183026: 电容触摸传感评估工具包



DM183026-2增强型 电容触摸传感评估工具包





MICROCHIP 2010

MASTERS Conference

附录

附录A —— CVD或CSM器件

器件系列	架构	通道数	XLP	EE PROM	引脚数	闪存（字数）
PIC12LF1822	8位增强型	4		256	8	2k
PIC16LF1823	8位增强型	4		256	14	2k
PIC16LF1826 / 1827	8位增强型	12		256	18	2k / 4k
PIC16F690	8位	12		256	20	4k
PIC16LF1933	8位增强型	16		256	28	4k
PIC16LF1937 / 1939	8位增强型	16		256	44	8k / 16k
PIC16LF1947	8位增强型	17		256	64	16k



“LF” 器件的功耗最低，1.8-3.3V

“F” 器件也适用，1.8-5.5V



“F1” 增强型内核器件的型号在F后有个1。

附录B —— CTMU器件

器件系列	架构	通道数	XLP	其他	引脚数	闪存 (KB)
PIC18F45K22	8位	30			28, 44	8 – 64k
PIC18F87J90	8位	12		LCD	64, 80	64 – 128k
PIC18F87K90	8位	13		LCD	28, 44	64 – 128k
PIC18F46J11 / 46J50	8位	13		USB	28, 44	16 – 64k
PIC24FJ256GA / GB110	16位	16			64至100	64 – 256k
PIC24FJ64GA104 / GB104	16位	13		USB	28, 44	16 – 64k
PIC24F16KA1xx系列	16位	9			14, 20, 28	4 – 16k
PIC24FJ256DA110 / GB210	16位	24		USB, GFX	64至121	64 – 256k

商标

Microchip的名称和徽标组合、Microchip徽标、dsPIC、KeeLoq、KeeLoq徽标、MPLAB、PIC、PICmicro、PICSTART、PIC³²徽标、rfPIC和UNI/O均为Microchip Technology Incorporated在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL和The Embedded Control Solutions Company均为Microchip Technology Incorporated在美国的注册商标。

Analog-for-the-Digital Age、Application Maestro、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified徽标、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rfLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock和ZENA均为Microchip Technology Incorporated在美国和其他国家或地区的商标。

SQTP是Microchip Technology Incorporated在美国的服务标记。
在此提及的所有其他商标均为各持有公司所有。

© 2010, Microchip Technology Incorporated. 版权所有。