



# MICROCHIP 2019 MASTERS

中国技术精英年会(第二十届)



## C20H03 TNG34

# 使用Microchip触摸库实现基于单片机的稳健触摸按键、滑条和触摸板

(MCC + mTouch®库和START + QTouch®库)

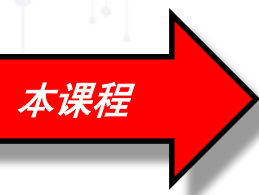




# 触摸和手势解决方案 MASTERS 2019

**C20L10 TNG2 电容式触摸原理、设计挑战和解决方案**

**C20H03 TNG34 使用Microchip触摸库实现基于单片机的稳健触摸按键、滑条和触摸板**



# 主讲人和助手

- 主任嵌入式解决方案工程师 **Alan Yao**
- 资深嵌入式解决方案工程师 **Vincent Liu**
- 资深嵌入式解决方案工程师 **Jim Li**
- 资深应用工程师 **Michael Zhu**



**MICROCHIP**  
2019  
**MASTERS**  
中国技术精英年会 (第二十届)

  
China MASTERS  
2000-2019  
20<sup>th</sup> YEAR

武汉 · 11月6 - 8日  
深圳 · 11月13 - 15日  
上海 · 11月20 - 22日

[www.microchip.com/cm](http://www.microchip.com/cm)

  
SMART | CONNECTED | SECURE

  
**MICROCHIP**



# 课程目标

完成本课程之后，您将能够...

- 能够使用**Microchip**的**PIC®**、**AVR®**和**Arm® Cortex MCU**为汽车、家电、工业和消费类等市场提供出色的触摸解决方案。
- 能够使用代码配置程序基于**Microchip Arm Cortex MCU**硬件平台轻松创建稳健的电容式触摸应用。
- 能够了解如何创建项目来满足用户的各种系统要求，包括低功耗、潮湿和噪声等使用环境。
- 通过**2D**触摸板的演示项目，能够了解**Microchip 2D**触摸板的相关设计、开发和调试等内容。



# 课程安排

- 什么是电容触摸传感
- **QTouch®**模块化库和**Data Visualizer®**实用程序简介
  - 实验1
- 低功耗设计
  - 实验2
- 防潮、防噪声设计
  - 实验3
- **2D**触摸表面
  - 实验4
  - 实验5
- 总结



# 简介和理论

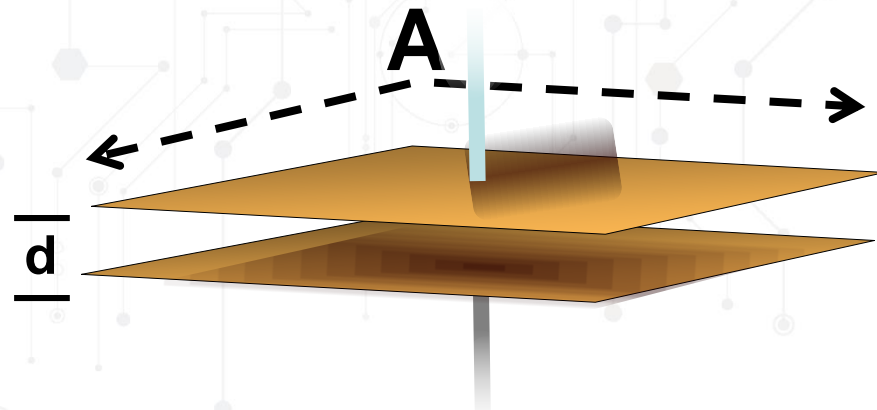
## 什么是电容触摸传感？

请参加**MASTERS**课程：

**C20L10 TNG2** 电容式触摸原理、设计挑战和解决方案

# 什么是电容触摸传感？

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



电容值取决于两个板的重叠面积以及它们之间的距离。

**C** 电容 (F)

**$\epsilon_0$**  自由空间的介电常数 (8.854 pF/m)

**$\epsilon_r$**  相对介电常数 (无单位)

**A** 板的面积 (m<sup>2</sup>)

**d** 两板间的距离 (m)

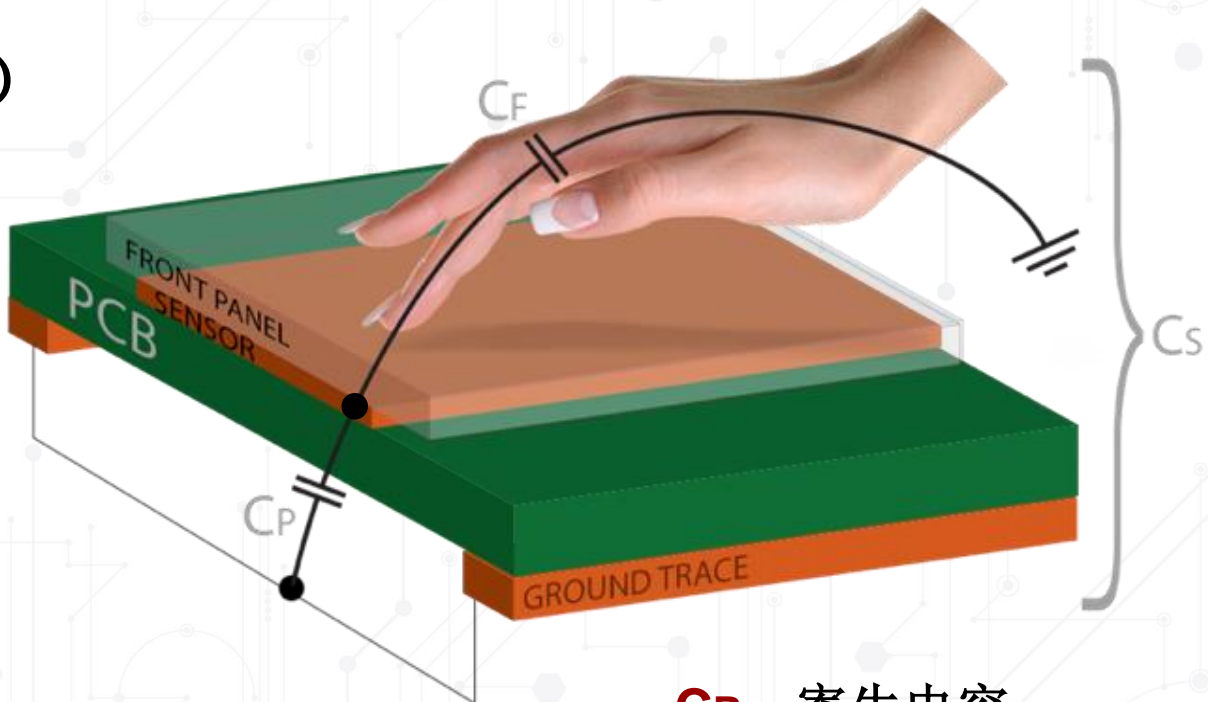
材料	$\epsilon_r$
空气	~1
聚乙烯	2.25
聚苯乙烯电容	2.4-2.7
玻璃	4-10
FR-4	4.8
水 (人类)	80



# 什么是电容触摸传感？

手指接触会产生并联电容

( $C_F$ )



$C_P$ : 寄生电容

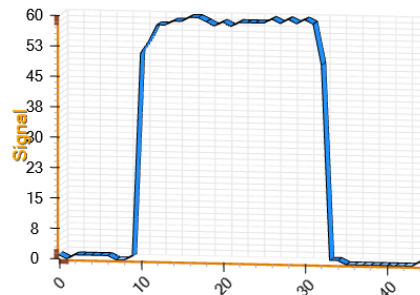
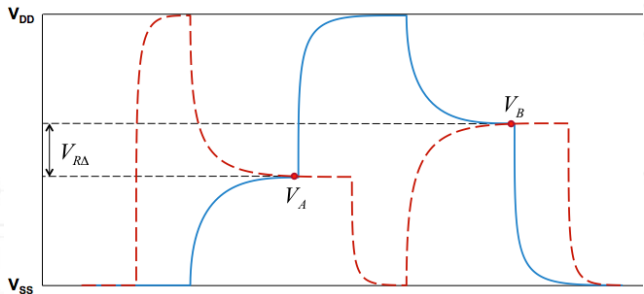
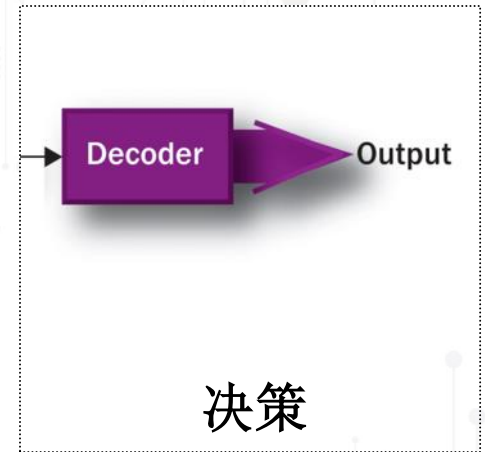
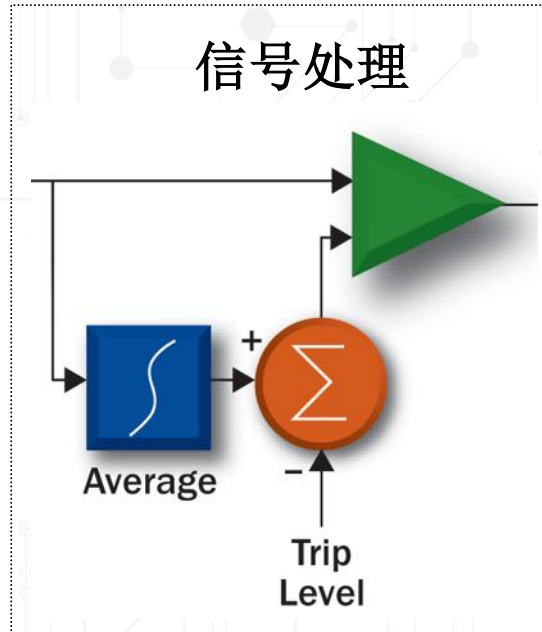
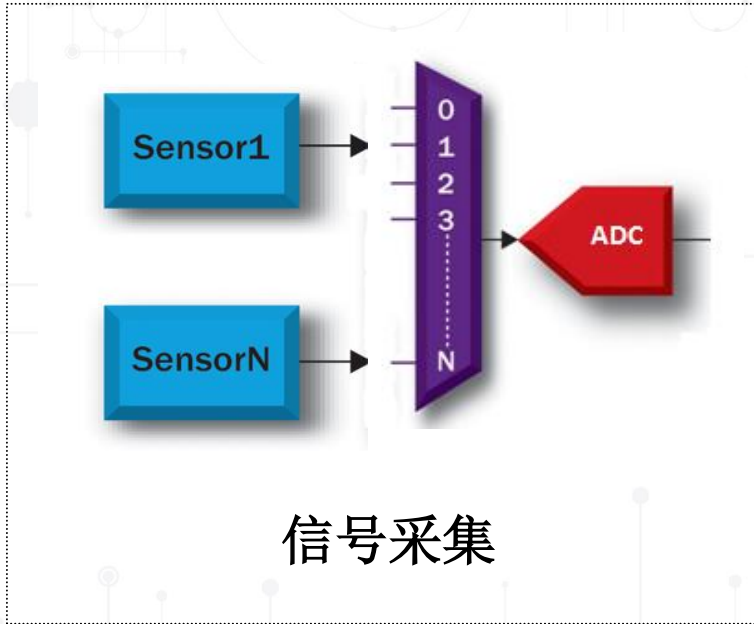
$C_F$ : 手指电容

$C_S$ : 总传感器电容

$$\text{传感器电容 } (C_S) = C_P + C_F$$



# 电容处理





# 课程安排

- 什么是电容触摸传感
- **QTouch®**模块化库和**Data Visualizer®**实用程序简介
  - 实验1
- 低功耗设计
  - 实验2
- 防潮、防噪声设计
  - 实验3
- **2D**触摸表面
  - 实验4
  - 实验5
- 总结



# START和QTouch®模块化库

## Atmel START简介

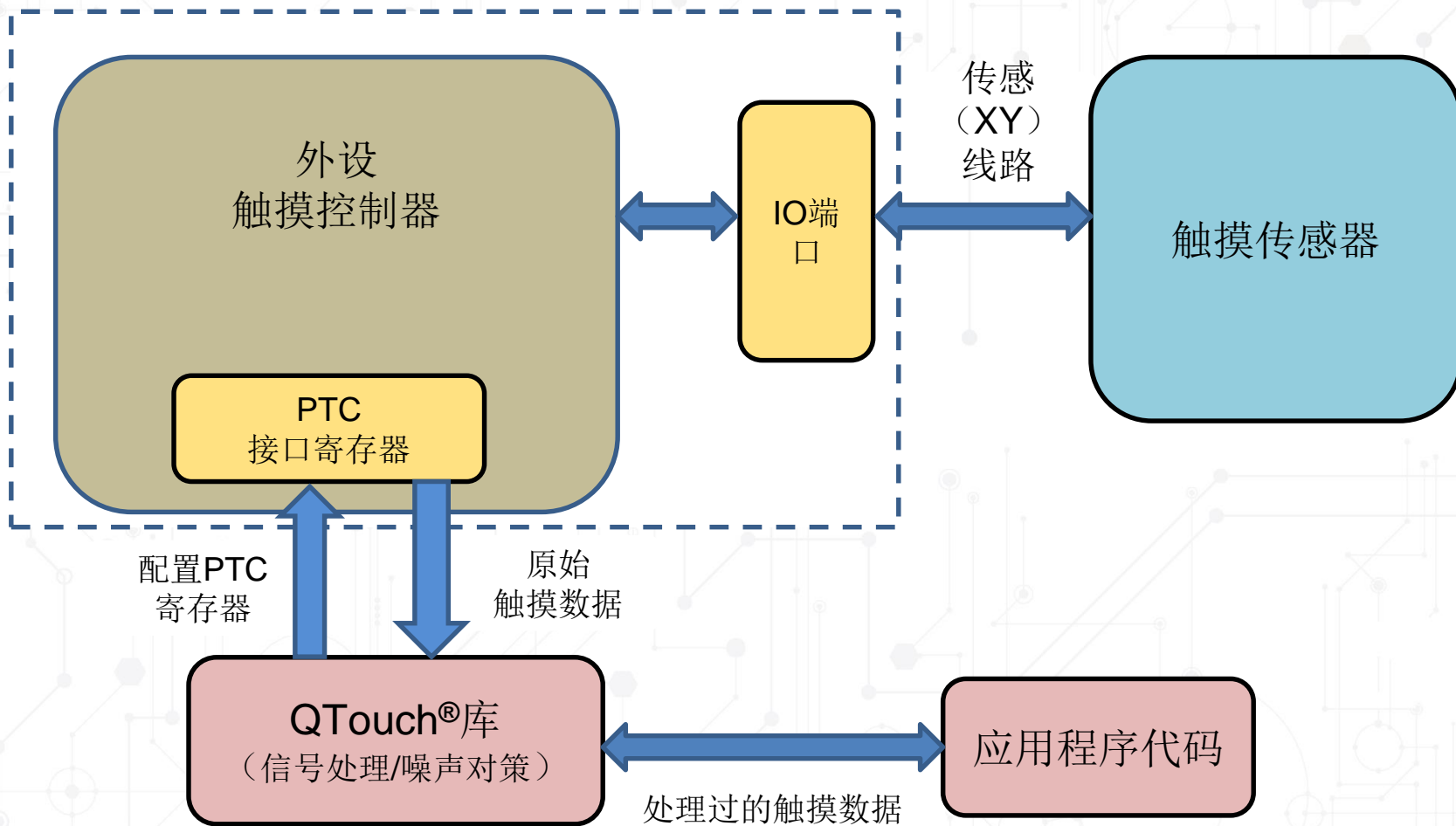
# START QTouch®解决方案

## QTouch®模块化库

- START QTouch®模块化库支持通过AVR® 8位MCU以及SAM 32位MCU和MPU中的外设触摸控制器（PTC）来实现触摸。
- 由START图形用户界面（GUI）生成代码。
- 适用于所有MCU的Web工具
  - [www.microchip.com/start](http://www.microchip.com/start)
- API函数可缩短开发时间
- 凭借触摸库和IDE，您可以将几十年累积的触摸专业知识应用于设计

[QTouch®触摸库用户指南](#)

# QTouch®库



# 模块化QTouch®库

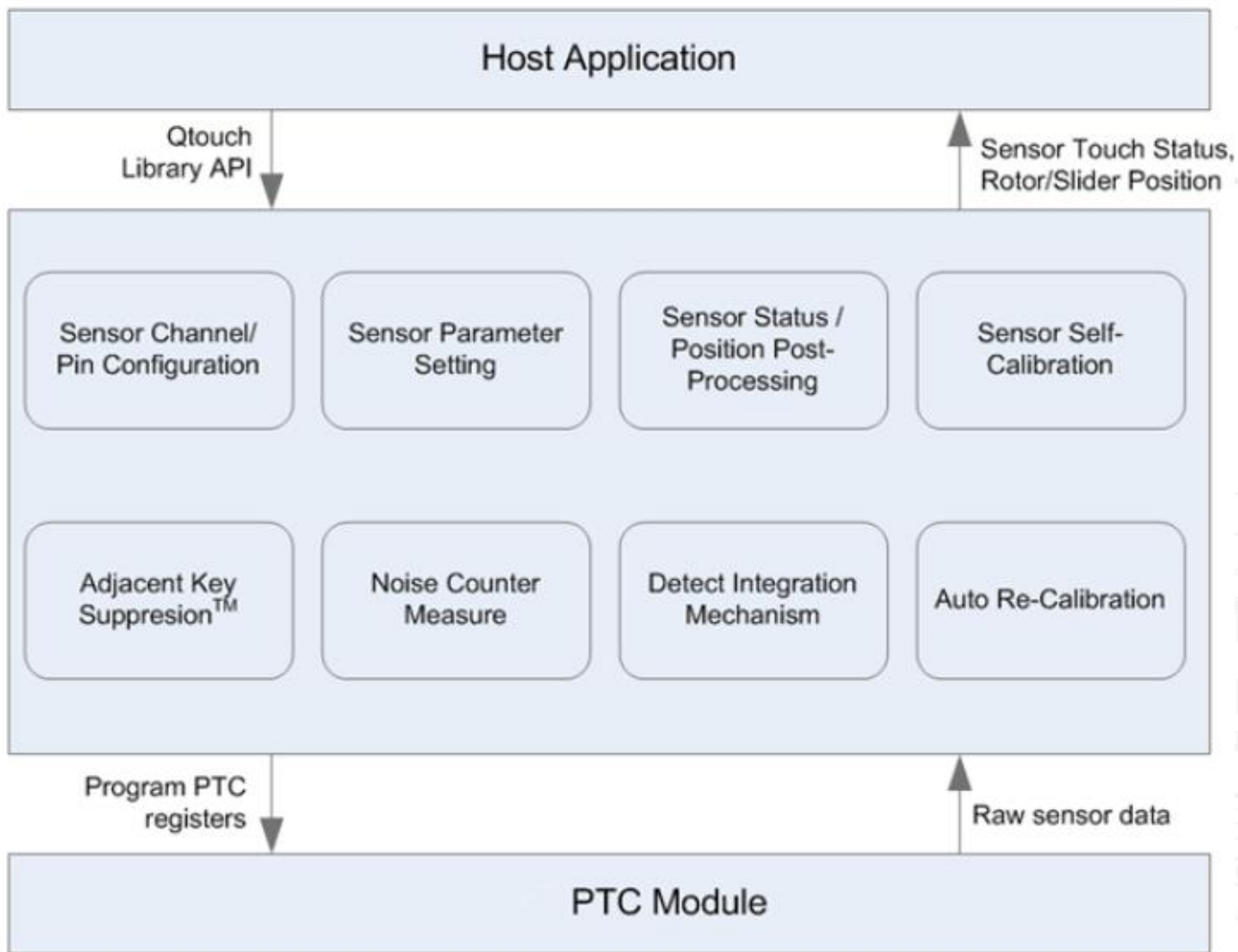
- 信号处理
  - 按键、滚动条和表面
- 高级噪声处理
- 编译器支持
  - GCC
  - IAR
- 架构支持
  - ARM
  - AVR®
- 经过全面测试
- 易于使用
- 为什么采用模块化?
  - 仅包括所需的功能
  - 减少存储空间→可在小尺寸器件中使用
  - 易于维护

# QTouch®库模块

- 可根据功能分为三类，如下所示。

Acquisition Module	Signal Conditioning module	Post processing module
<div data-bbox="127 629 606 748">Acquisition auto tune module</div> <div data-bbox="127 779 606 898">Acquisition run-time module</div>	<div data-bbox="710 629 1190 748">Frequency Hop module</div> <div data-bbox="710 779 1190 898">Frequency Hop Auto tune module</div>	<div data-bbox="1290 636 1769 755">Touch Key module</div> <div data-bbox="1290 779 1769 898">Scroller module</div>
<ul style="list-style-type: none"> <li>• Touch measurement</li> <li>• Channel sequencing</li> <li>• CC calibration</li> <li>• Auto/manual calibration of prescalar/series resistor/charge share delay</li> </ul>	<ul style="list-style-type: none"> <li>• Hop frequency</li> <li>• Median filter</li> <li>• Noise measurement</li> <li>• Change frequency based on noise</li> </ul>	<ul style="list-style-type: none"> <li>• Buttons post processing</li> <li>• Drifting, Detect integration</li> <li>• AKS groups</li> <li>• Slider/Wheel position</li> <li>• Touch active/inactive status</li> <li>• Hysteresis, IIR filtering</li> </ul>

# QTouch®模块化库 描述





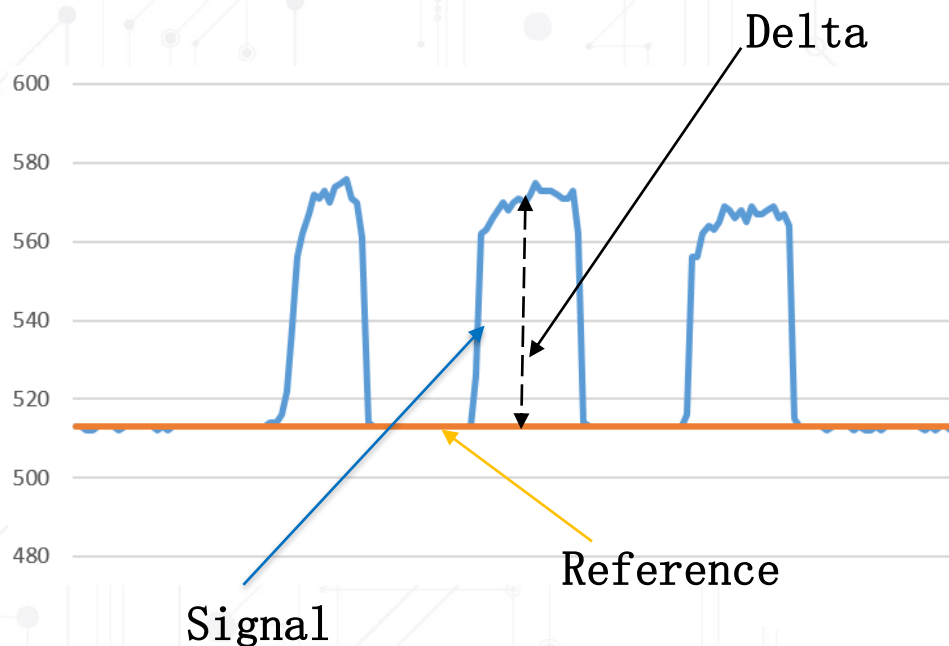


## 传感器配置参数



# 基本参数

- **信号Signal**
  - 通道的原始测量值
  - ACQ模块的输出
- **参考Reference**
  - 通道的长期平均测量值
  - 从校准过程中获得
  - 通过漂移补偿算法调节
  - 按键模块的输出
- **增量Delta = 信号 - 参考**
  - 按钮为30和80之间
  - 滑动条和滚轮为50和120之间



```
get_sensor_node_signal(n);  
get_sensor_node_reference(n);
```



# 增益Gain

- 可放大接触后的触摸增量Delta
- 增大增益设置
  - 提高传感器的灵敏度
  - 触摸测量时间不受影响
- 选择取决于：
  - 传感器设计
  - 触摸面板厚度
- 模拟和数字增益
  - $\uparrow$ 模拟增益 =  $\uparrow$ 噪声
  - $\uparrow$ 数字增益 =  $\downarrow$ 噪声 &  $\uparrow$ 采集时间

总增益 = 模拟增益 \* 数字增益

Total Gain = Analog Gain \* Digital Gain



受噪声影响，建议仅使用数字增益，模拟增益为1。



# 增益Gain

文件: touch.h

- /\*配置: X线、Y线、充电延时、NODE\_RSEL\_PRSC (串联电阻和预分频器)、NODE\_G (模拟增益和数字增益) 和滤波等级\*/

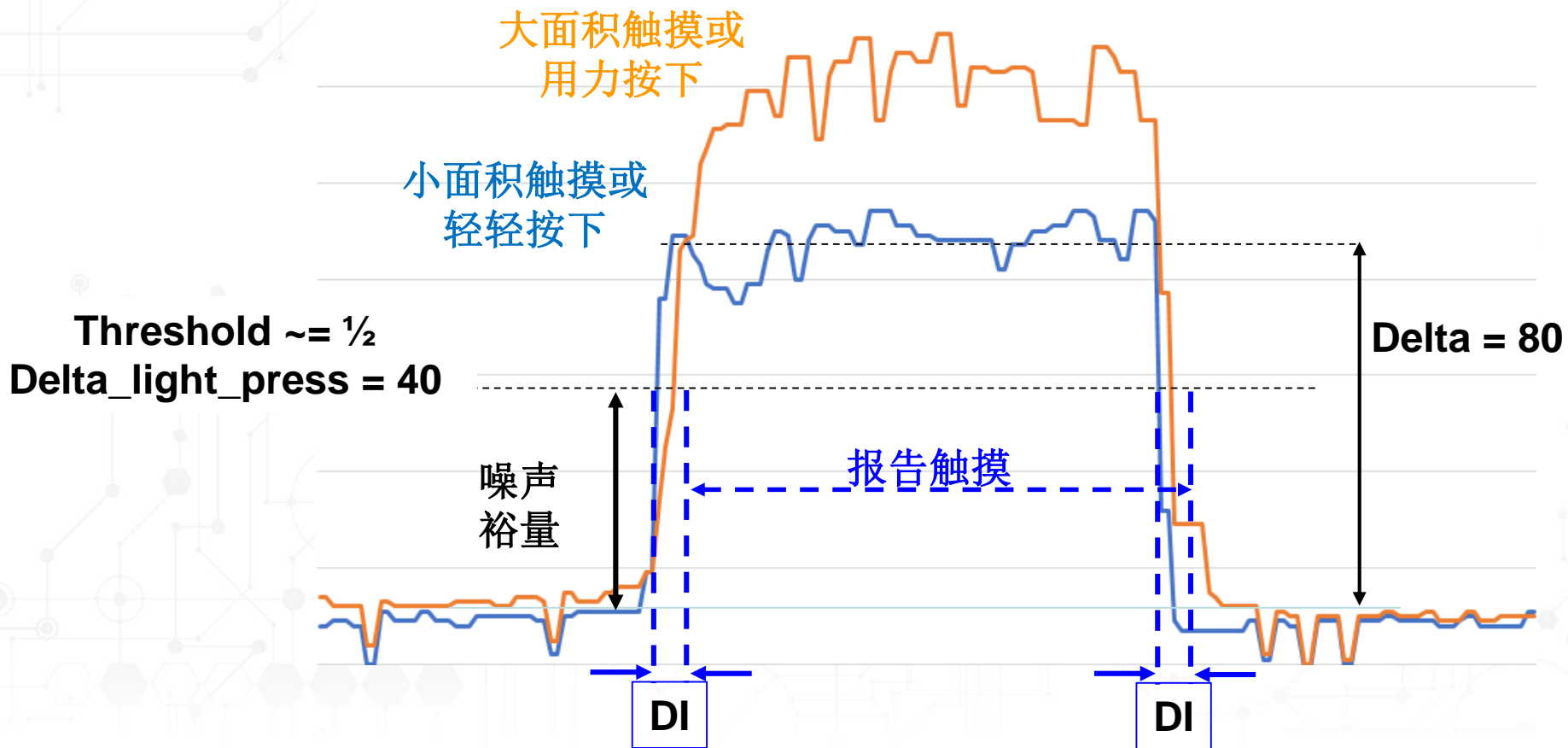
```
#define NODE_0M_PARAMS {X_NONE, Y(0), 0, NODE_RSEL_PRSC(RSEL_VAL_0,  
PRSC_DIV_SEL_1), NODE_GAIN(GAIN_1, GAIN_1), FILTER_LEVEL_16}
```

- 可基于各个通道配置
- 可配置为1至32间的值 (GAIN\_1、GAIN\_2、GAIN\_4、GAIN\_8、GAIN\_16和GAIN\_32)
- 滤波等级应  $\geq$  数字增益
- 总增益应  $\leq 32$



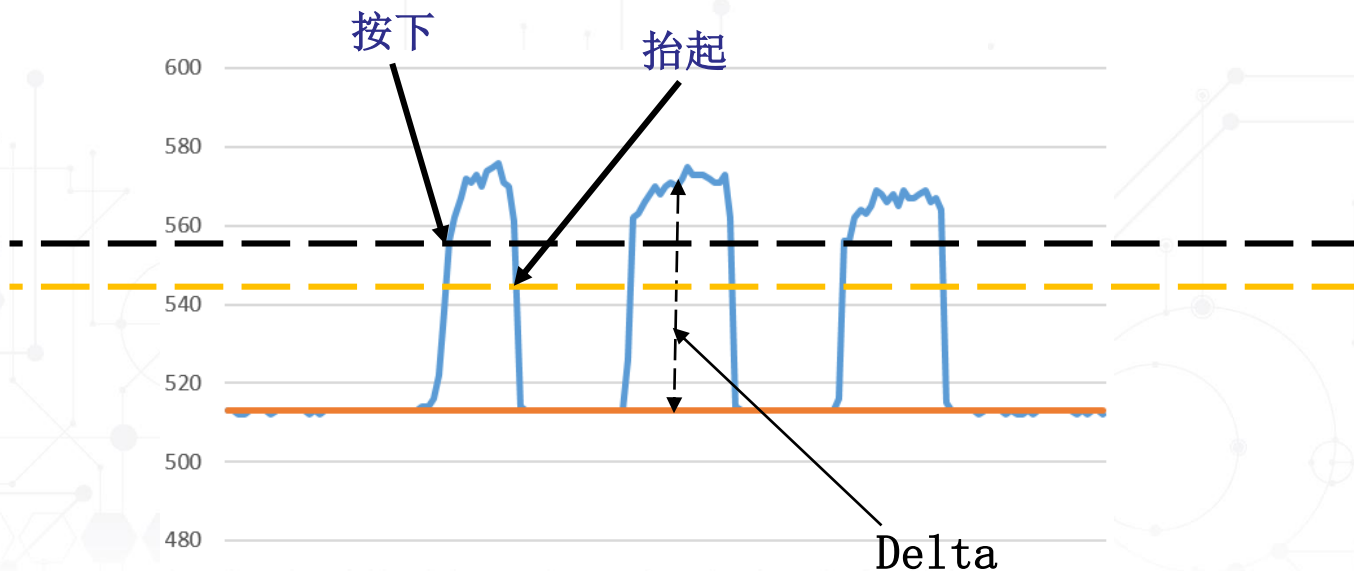
# 阈值Threshold和去抖 (DI)

- 基本调节参数
  - 阈值Threshold
  - 去抖 (DI)



# 检测阈值 Detect Threshold

- 定义信号必须比参考值增加多少才能认为检测到触摸
- 如果增量  $\Delta \geq$  检测阈值 Threshold, 则声明为触摸
- 如果增量  $\Delta < (\text{检测阈值 Threshold} - \text{滞后 Hysteresis})$ , 则声明为无触摸
- 滞后 Hysteresis 定义为检测阈值 Threshold 的百分比 (6.25%、12.5%、25%和50%)



# 检测阈值 Detect Threshold

文件: touch.h

/\*配置:

阈值Threshold、滞后Hysteresis和AKS组\*/

```
#define KEY_0_PARAMS { 50u, HYST_25, AKS_GROUP_1 }
```

- 每个按键可单独配置
- 范围: 1 ~ 255

# 充电时间 Charge Time

## 充电时间 Charge Time

- 给传感器电极充电所需的时间
- 取决于RC时间常数

传感器电容越大，需要的充电时间较长，反之亦然。

可通过两种方式调节充电时间：

- *预分频器 Prescaler*
- *CSD (电荷共用延时 Charge Share Delay)*

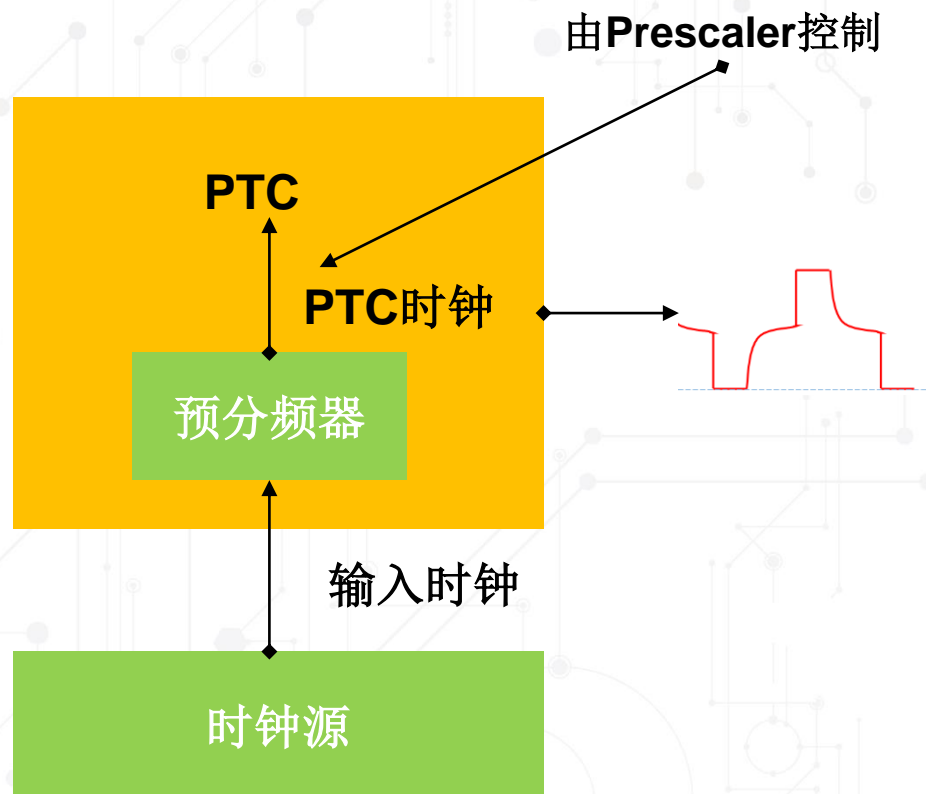


# 充电时间CT——预分频值 Prescaler

- PTC需要时钟才能运行
- 采样脉冲和ADC取决于PTC时钟
- PTC有一个预分频器，用于调节输入时钟

增大预分频比 -> 减小PTC时钟 -> 延长充电时间

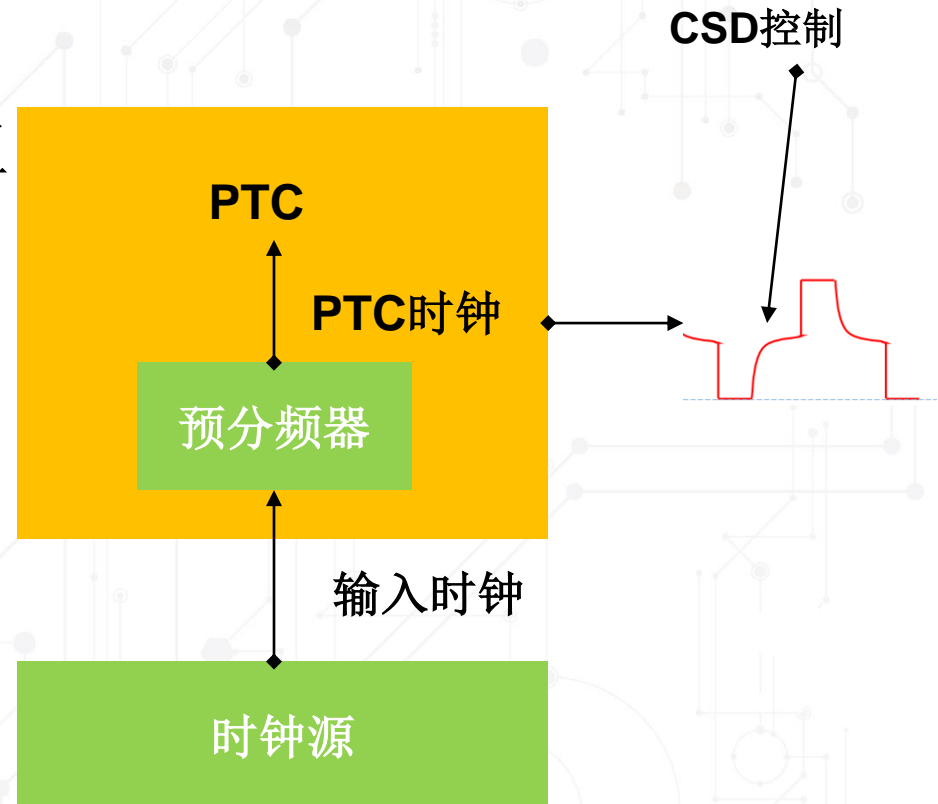
增大预分频比会延长充电时间并减慢ADC的运行速度



# 充电时间——CSD

- CSD控制充电时间，与PTC时钟无关
- 可以使用较低的预分频值，但仍可以实现较长的充电时间
- 与ADC时序无关

延长CSD -> 延长充电时间



# 预分频器Prescaler还是CSD

- 预分频器Prescaler的应用场合
  - 将输入时钟限制为允许的PTC限值以内。
    - 例如，在SAM L10中，PTC与CPU共用同一时钟。
    - PTC的最高工作频率为8 MHz。CPU的最高工作频率为20 MHz。
    - 降低PTC的时钟频率会使CPU速度变慢。
    - 因此，需要为CPU使用更高的时钟频率，并需要使用预分频器来降低PTC时钟频率
  - 在没有CSD的器件中，使用预分频器调节充电时间以确保传感器电极完全充电。
- CSD的应用场合
  - 用于确保传感器电极完全充电。


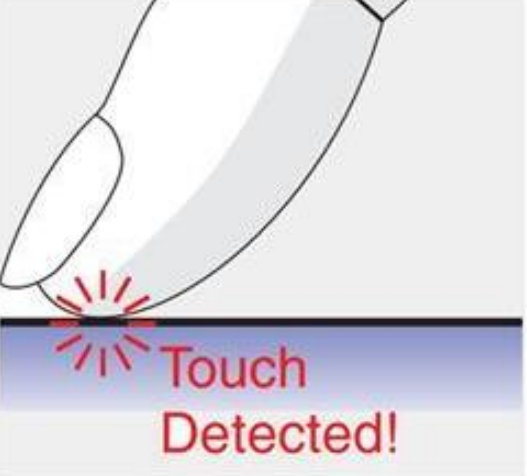
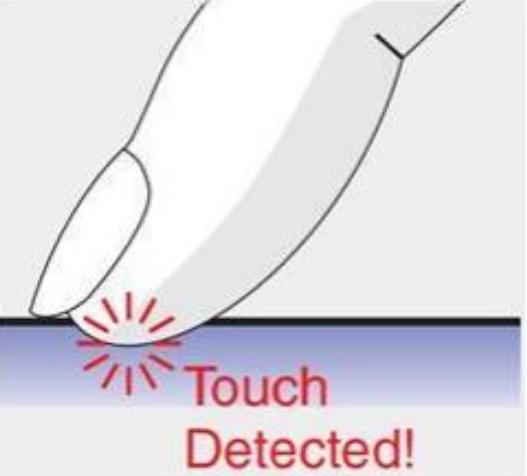
## Touch.h

*/\*配置：X线、Y线、CSD、NODE\_RSEL\_PRSC（串联电阻和预分频器）、  
NODE\_G（模拟增益和数字增益）和滤波等级\*/*

```
#define NODE_0M_PARAMS {X_NONE, Y(0), 5, NODE_RSEL_PRSC(RSEL_VAL_0,  
PRSC_DIV_SEL_1), NODE_GAIN(GAIN_1, GAIN_1), FILTER_LEVEL_16}
```



# 灵敏度 Sensitivity 调节

Over-Sensitive	Optimally Sensitive	Under-Sensitive
		
<p>If the sensor is reporting detect before contact with the touch surface, then the sensor is oversensitive.</p>	<p>A sensor is at the optimized sensitivity when it only detects the finger as it lightly contacts the touch surface.</p>	<p>If the sensor does not detect the touch or requires a heavy touch to activate, then the sensor is under-sensitive.</p> <p>Note: Capacitive Touch is not sensitive to pressure, but to the expanded area of the flattened finger.</p>
<p>Reduce Gain</p>	<p>No action required</p>	<p>Increase Gain</p>
<p>Increase Detect Threshold</p>	<p>No action required</p>	<p>Decrease Detect Threshold</p>

# 我们学到了什么？

- 基本参数
  - 信号Signal
  - 参考Reference
  - 增量Delta
- 配置参数
  - 充电时间Charge Time
  - 检测阈值Detect Threshold
  - 模拟和数字增益 Analog and Digital Gain
- 灵敏度调节
  - 欠灵敏 Under Sensitive
  - 过灵敏 Over Sensitive
  - 最佳灵敏度 Optimally Sensitive



# 实验笔记

本实验分为两部分

**第1部分：使用适用于SAML10 XPLAINED PRO + QT7 Xplained PRO硬件的START®创建兼具触摸按钮和滑动条功能的项目  
在MPLAB X中添加代码（可选择Studio 7）**

**第2部分：基于Data Visualizer®实用程序显示原始触摸信息。**



# 传感器设计准则

- **Z轴——层堆叠Layer Stack-up**
- 覆盖层中使用的材料（需要从项目开始时考虑） **Cover Lens**
- 传感器和手指之间的电介质的介电常数
- 传感器设计准则

## AN2934——电容式触摸传感器设计指南



# START的本地版本

- 在MASTERS会议期间，我们将使用本地版本的START，网址<http://localhost:8082>。
- 在培训室以外，您可从<https://start.atmel.com>或<https://www.microchip.com/start>正常访问START
- 我们已经对您的MASTERS PC进行了相应配置：  
Atmel Studio: Tools / Options / Atmel Start / Settings（工具 / 选项 / Atmel Start / 设置）：后端URL和前端URL（从<https://start.atmel.com>到<http://localhost:8082>）



# 实验1目标

## 第1部分:

- 使用**START**和**QTouch®**模块化库创建一个新的**SAML10 QTouch®**项目
- 选择传感器并配置引脚
- 添加代码以通过**LED**指示触摸

## 第2部分:

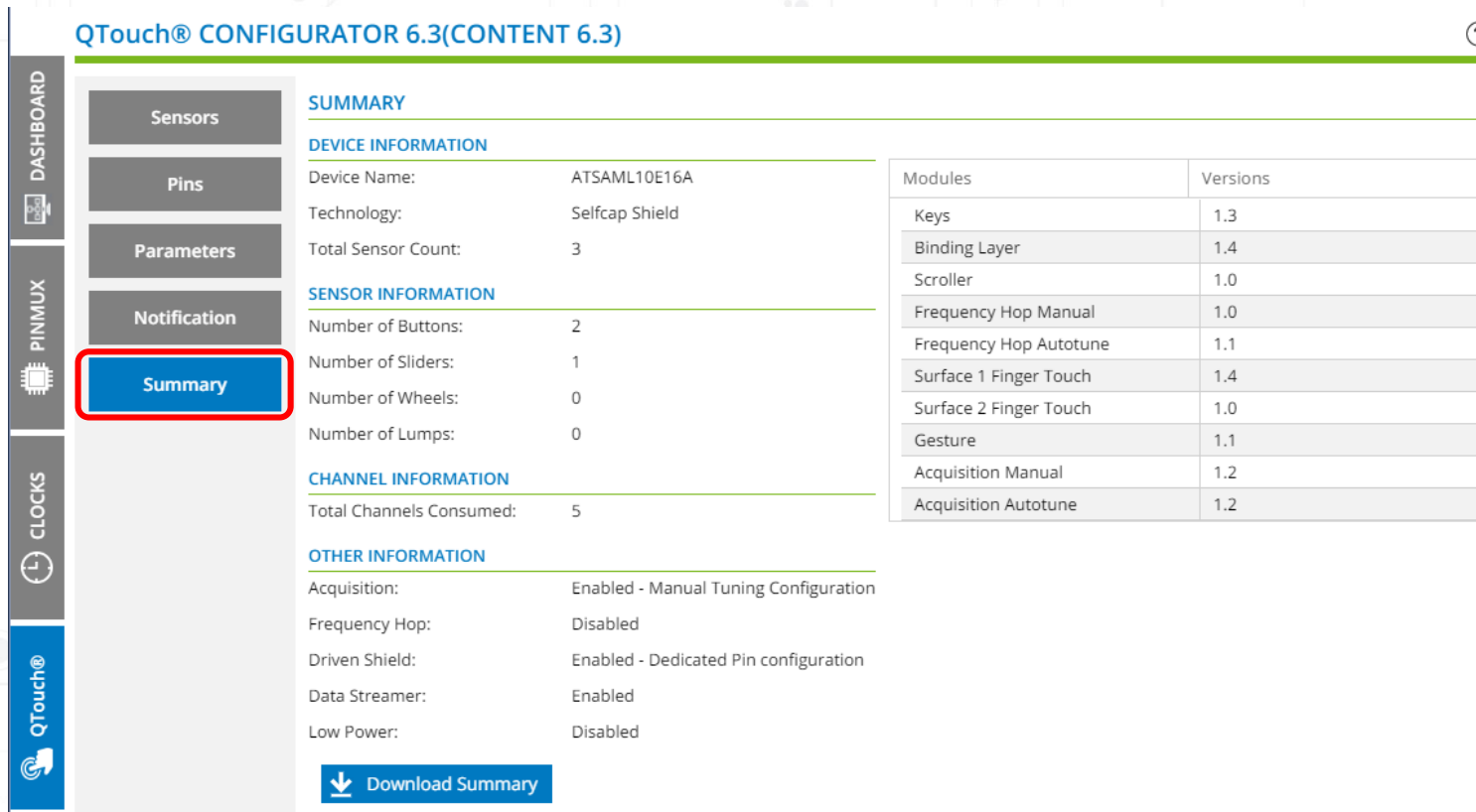
- 实现**USART**通信
- 配置**Data Visualizer**
- 显示原始触摸数据和处理过的触摸数据

请按照  
实验手册进行操作!

## 13 查看汇总

- 单击 **Summary** (汇总)
- 确认您的整体项目配置。

QTouch® CONFIGURATOR 6.3(CONTENT 6.3) ?



**SUMMARY**

**DEVICE INFORMATION**

Device Name:	ATSAML10E16A
Technology:	Selfcap Shield
Total Sensor Count:	3

**SENSOR INFORMATION**

Number of Buttons:	2
Number of Sliders:	1
Number of Wheels:	0
Number of Lumps:	0

**CHANNEL INFORMATION**

Total Channels Consumed:	5
--------------------------	---

**OTHER INFORMATION**

Acquisition:	Enabled - Manual Tuning Configuration
Frequency Hop:	Disabled
Driven Shield:	Enabled - Dedicated Pin configuration
Data Streamer:	Enabled
Low Power:	Disabled

Modules	Versions
Keys	1.3
Binding Layer	1.4
Scroller	1.0
Frequency Hop Manual	1.0
Frequency Hop Autotune	1.1
Surface 1 Finger Touch	1.4
Surface 2 Finger Touch	1.0
Gesture	1.1
Acquisition Manual	1.2
Acquisition Autotune	1.2

[Download Summary](#)



# 实验1——QTouch® START项目创建

## 15 时钟配置器

- 单击 **Clocks**（时钟），打开 **CLOCK CONFIGURATOR**（时钟配置器）
- 将PTC时钟从默认的4 MHz更改为输入8 MHz（通用时钟发生器1进行2分频而非4分频）以证明我在8 MHz下运行最新的PTC。

### CLOCK CONFIGURATOR

Zoom in Zoom out Reset

#### OSCILLATORS

- External Crystal Oscillator 0.4-32MHz
- 32kHz External Crystal Oscillator
- 16MHz Internal Oscillator (OSC16M)  
16 MHz
- Digital Frequency Locked Loop
- Digital Phase Locked Loop
- 32kHz Ultra Low Power Internal Oscillator (OSCULP32K)

#### SOURCES

- Generic clock generator 0  
16 MHz
- Generic clock generator 1  
8 MHz
- Generic clock generator 2
- Generic clock generator 3
- Generic clock generator 4
- RTC source

#### COMPONENTS

- CPU  
16 MHz
- PTC  
PTC 8 MHz
- Timer  
RTC 32.768 kHz
- USART  
Core 16 MHz  
Slow 400 kHz

### CLOCK SETTINGS

#### GENERIC CLOCK GENERATOR 1 CONFIGURATION

#### GENERIC CLOCK GENERATOR CONTROL

Generic clock generator 1 source: 16MHz Internal Oscillator (OSC16M)

Run in Standby:

Divide Selection:

Output Enable:

Output Off Value:

Improve Duty Cycle:

Generic Clock Generator Enable:

#### GENERIC CLOCK GENERATOR DIVISION

Generic clock generator 1 division: 2

#### USAGE

Used by: PTC



**MICROCHIP**  
**MASTERS 2019**

# Data Visualizer简介



# Data Visualizer说明

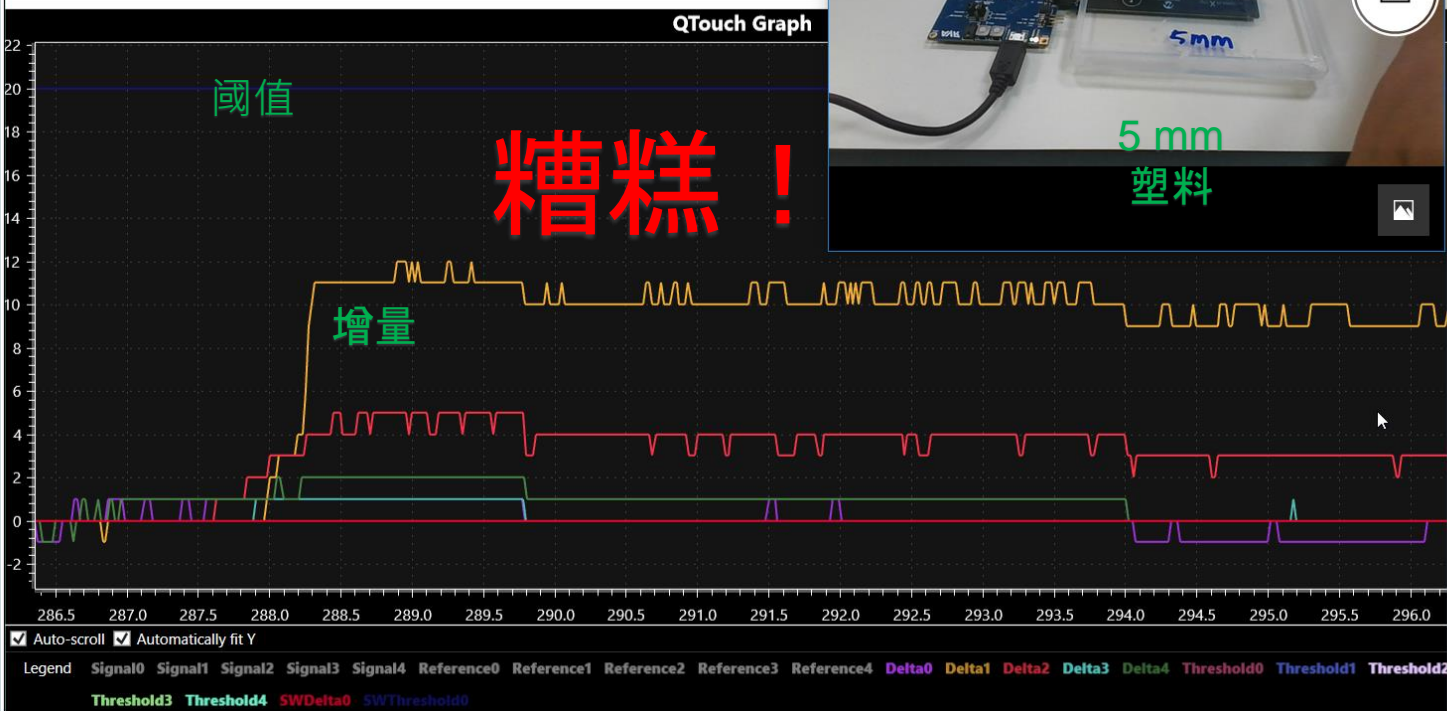
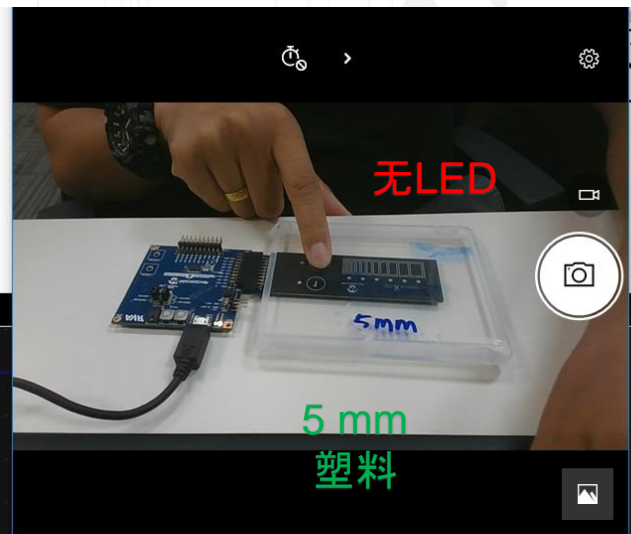
- **Data Visualizer**是用于处理和可视化数据的程序。
- 它可以从各种来源（例如，Xplained Pro板上的嵌入式调试器数据网关接口和COM端口）接收数据。
- 它具有图形功能、示波器、终端以及带有按钮、滑动条和各种指示器的可配置仪表板。
- 它可以解码协议，进行相应显示并记录到文件中。
- 我们在此使用**Data Visualizer**来查看实时触摸数据。





# 为什么选择Data Visualizer? 5 mm盖板

Channel ID	Sensor Type	State	Delta	Threshold
0	Button 0	0	0	20
1	Button 1	0	9 增量	20
2	Slider 0[0]	0	3	20
3	Slider 0[1]	0	0	20
4	Slider 0[2]	0	0	20





# 实现Data Visualizer

- 如何实现UART以及与Data Visualizer连接？
- MCC mTouch®和START QTouch®配置器均需要您选中 Debug - Enable Data Streamer（调试 - 使能Data Streamer）功能。
- 在要生成最终生产代码时，可以取消选中Debug - Enable Data Streamer功能。



需要使用最新的Data Visualizer，下载地址为：

<https://gallery.microchip.com/packages/AtmelDataVisualizerInstaller-Standalone/>.





# 实验1目标

## 第1部分:

- 使用**START**和**QTouch<sup>®</sup>**模块化库创建一个新的**SAML10 QTouch<sup>®</sup>**项目
- 选择传感器并配置引脚
- 添加代码以通过**LED**指示触摸

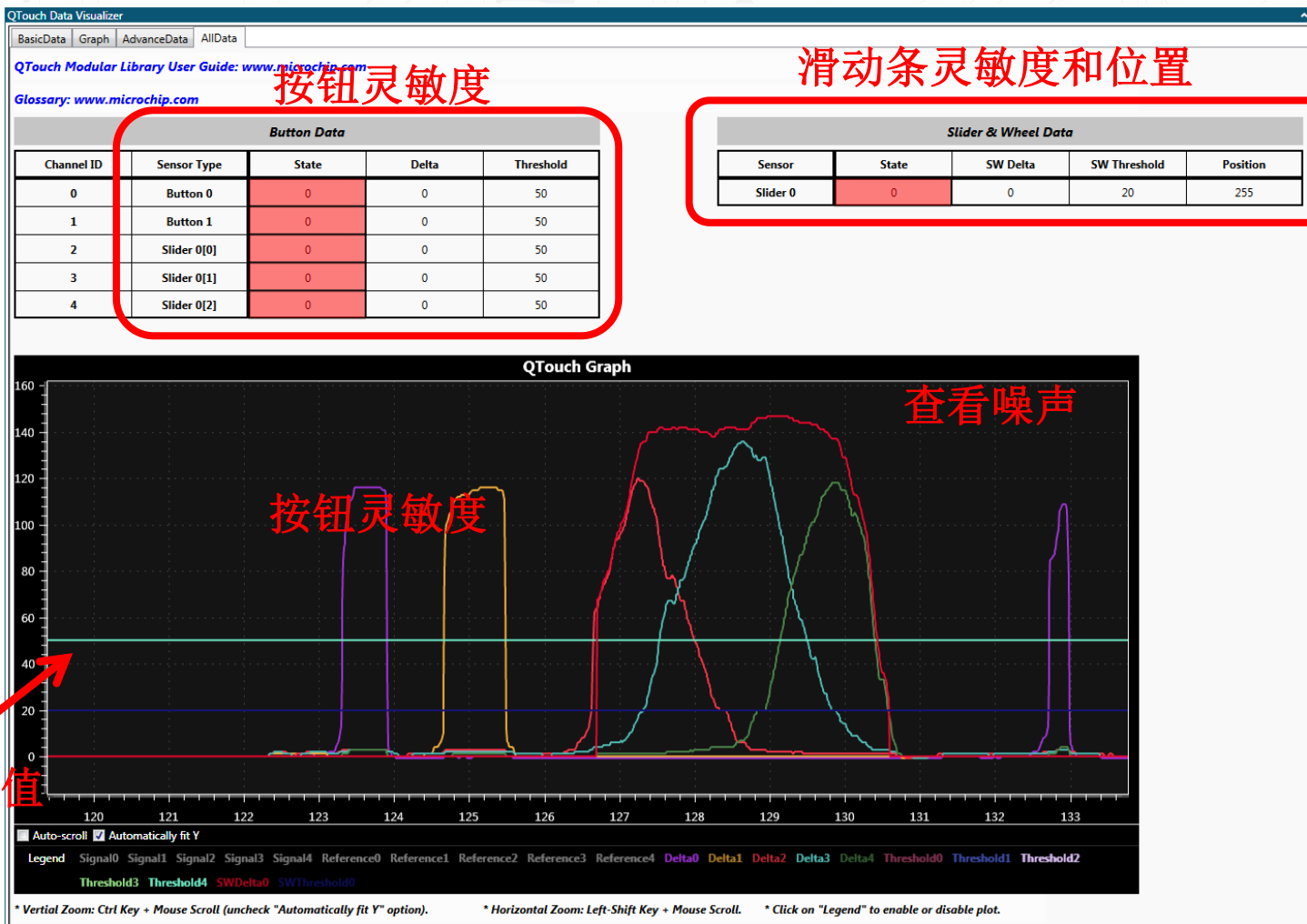
## 第2部分:

- 实现**USART**通信
- 配置**Data Visualizer**
- 显示原始触摸数据和处理过的触摸数据

**请按照  
实验手册进行操作!**

# 实验1b——使用Data Visualizer

**24** 查看按钮和滑动条灵敏度、滑动条位置和噪声等：



# 实验1b——使用Data Visualizer

使能/禁止图形数据和查看原始通道数据等:

缩放x轴和y轴



The screenshot shows the Data Visualizer interface with a waveform plot at the top and several data tables below. A red arrow points to the plot area, and a red box highlights the legend and the Sensor Data table.

**Legend:** Signal0 Signal1 Signal2 Signal3 Signal4 Reference0 Reference1 Reference2 Reference3 Reference4 Delta0 Delta1 Delta2 Delta3 Delta4 Threshold0 Threshold1 Threshold2  
Threshold3 Threshold4 SWDelta0 SWThreshold0

**Sensor Data**

Channel ID	Sensor Type	Signal	Reference	Delta	Compensation pF
0	Button 0	520	520	0	31.56975
1	Button 1	513	513	0	17.93475
2	Slider 0[0]	519	519	0	39.19725
3	Slider 0[1]	523	523	0	24.34725
4	Slider 0[2]	527	527	0	25.69725

**Frequency Hop Data**

CurrentFrequency	HopFrequency0	HopFrequency1	HopFrequency2
7	7	13	4

**Debug Data**

FrameCounter	71	Counter for datastreamer packets. Missing count indicate packet drop
QTouchLibError	0	Indicates library error state. Zero: no error. Refer "Error Code" section in User Guide

使能/禁止参数显示

原始通道数据

# 实验1总结

## 第1部分:

- 使用**START**和**QTouch®**模块化库创建一个新的**SAML10 QTouch®**项目
- 选择传感器并配置引脚
- 添加代码以通过**LED**指示触摸

## 第2部分:

- 实现**USART**通信
- 配置**Data Visualizer**
- 显示原始触摸数据和处理过的触摸数据



# 课程安排

- 什么是电容触摸传感
- **QTouch<sup>®</sup>模块化库和Data Visualizer<sup>®</sup>实用程序简介**
  - 实验1——
- **低功耗设计**
  - 实验2——
- **防潮、防噪声设计**
  - 实验3——
- **2D触摸表面**
  - 实验4——
  - 实验5——
- **总结**



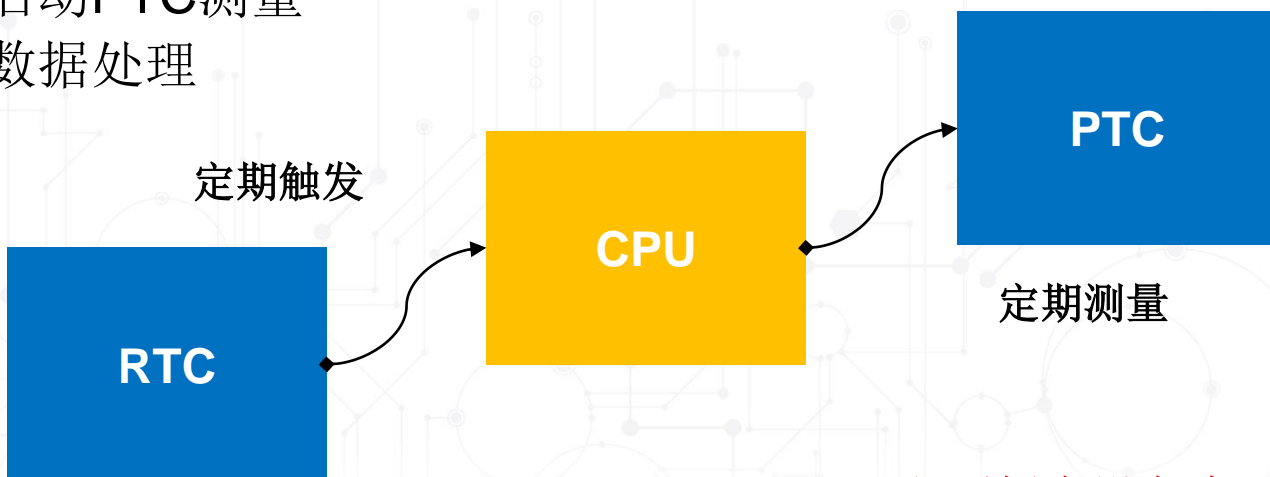
**MICROCHIP**  
**MASTERS 2019**

## 低功耗触摸测量



# 低功耗触摸测量 Low-Power Touch Measurement

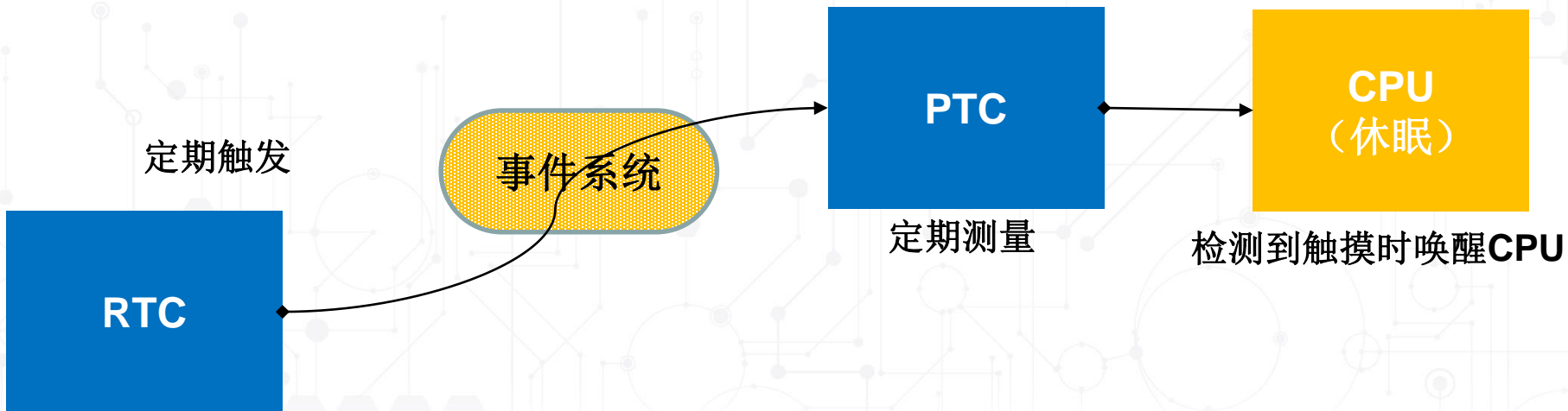
- 电池供电应用中需要低功耗
- CPU休眠→降低功耗
- 在触摸中，CPU用于：
  - 配置PTC
  - 定期启动PTC测量
  - 后置数据处理



主要耗电设备为CPU

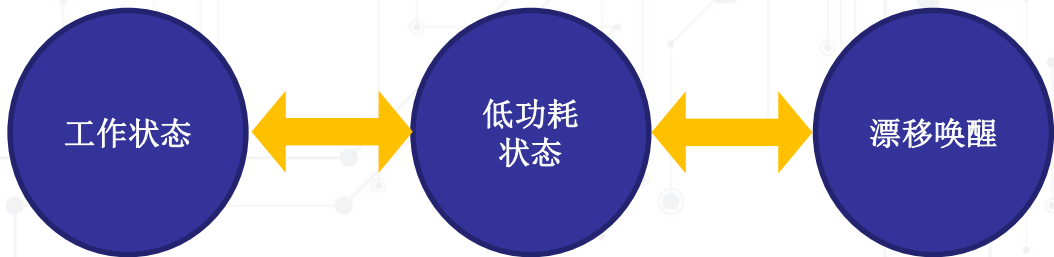
# 没有CPU的触摸测量

- RTC/任何其他周期性信号源应直接触发PTC触摸测量
- 在SAM和tinyAVR®器件中，通过事件系统完成
- 在megaAVR®器件中，通过使用128 KHz振荡器完成
  - 布置与事件系统相似——但megaAVR中不存在事件系统
- 限制
  - 只能在一个通道上执行触摸测量





# 状态机



## 工作模式

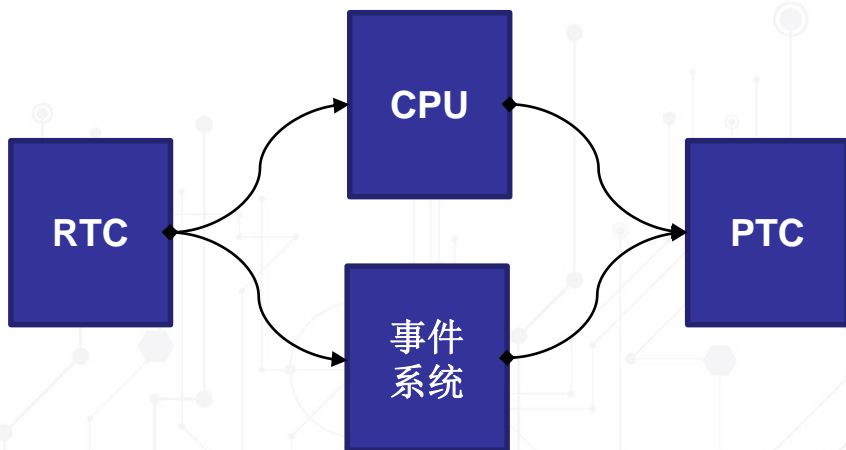
- RTC向CPU发出中断请求 (20 ms)
- CPU触发触摸测量

## 低功耗模式

- 事件系统触发触摸测量

## 漂移唤醒

- CPU触发触摸测量



- 由于环境变化导致低功率测量无法跟踪参考信号，因此需要漂移唤醒

# 状态机

切换期间会发生什么？



## 工作状态到低功耗状态

- RTC中断周期更改为2秒
- 使能事件系统
- 在PTC中使能低功耗测量

## 低功耗状态到漂移状态

- 禁止事件系统
- 在PTC中禁止低功耗测量

## 漂移唤醒状态到低功耗状态

- 在PTC中使能低功耗测量
- 使能事件系统

## 低功耗状态到工作状态

- RTC中断周期更改为20 ms
- 禁止事件系统
- 在PTC中禁止低功耗测量



# 状态机

何时切换?



工作状态到低功耗状态

- 长时间（例如5s）未检测到触摸时

低功耗状态到漂移状态

- 定期（例如2~5s）

漂移唤醒状态到低功耗状态

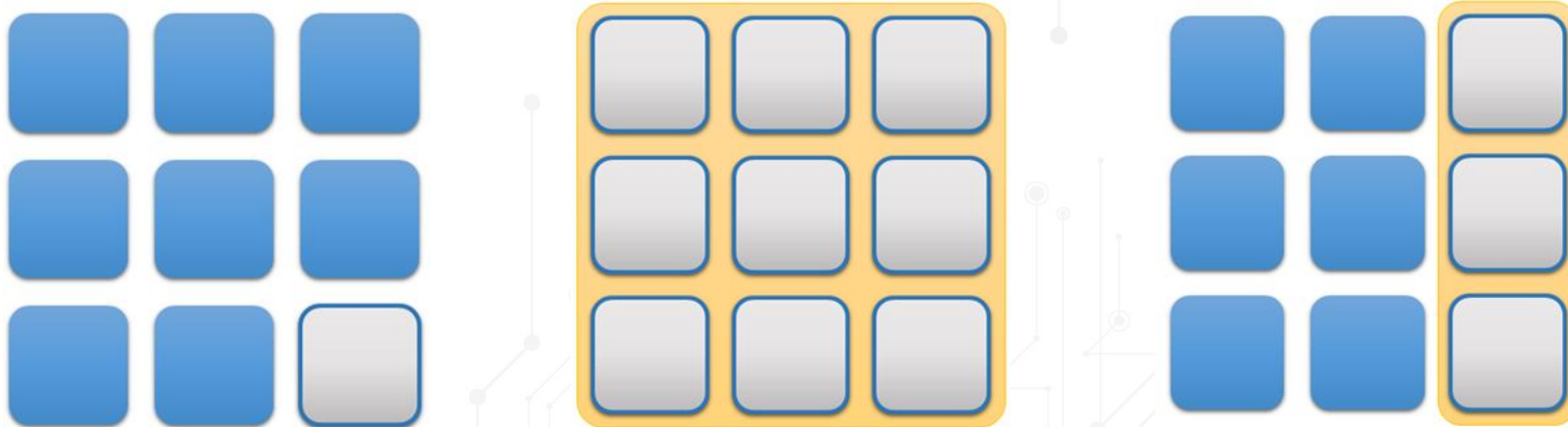
- 一次触摸测量结束

低功耗状态到工作状态

- 在低功耗测量中检测到触摸时

# 集总模式Lump Mode

- 一个按钮用作低功耗传感器
- 所有按钮集总并配置为低功耗传感器
- 部分按钮组集总并配置为集总传感器



将集总传感器配置为低功耗传感器



**实验2——低功耗设计：  
演示通过将项目配置为低功耗模式，创建“触摸唤醒”行为  
可以降低多少功耗。**



# 实验2——低功耗设计

## 目标

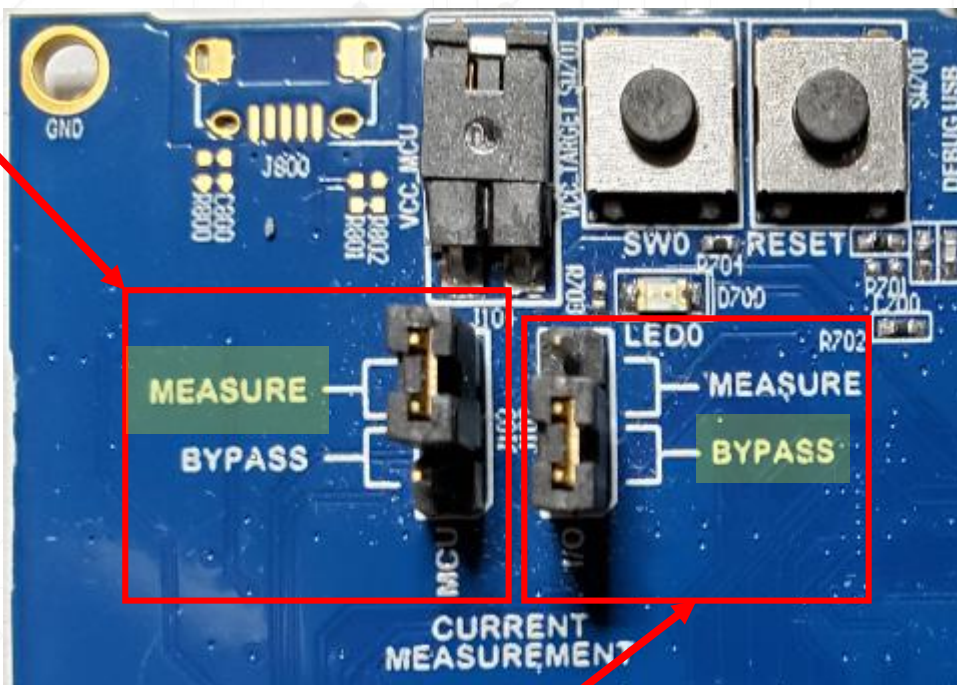
- 使用**SAML10 Xplained Pro**功耗调试器功能基于**Data Visualizer**测量实验1项目的平均电流。
- 指导如何通过修改**START**项目的方法来降低平均电流消耗，创建“触摸唤醒”行为。
- 测量“触摸唤醒”行为的平均电流消耗，并将其与实验1项目进行比较。



# 实验2——低功耗设计

**1** 准备SAML10 Xplained Pro以进行电流测量。

- 在SAML10板上，将MCU跳线连接到**MEASURE**。

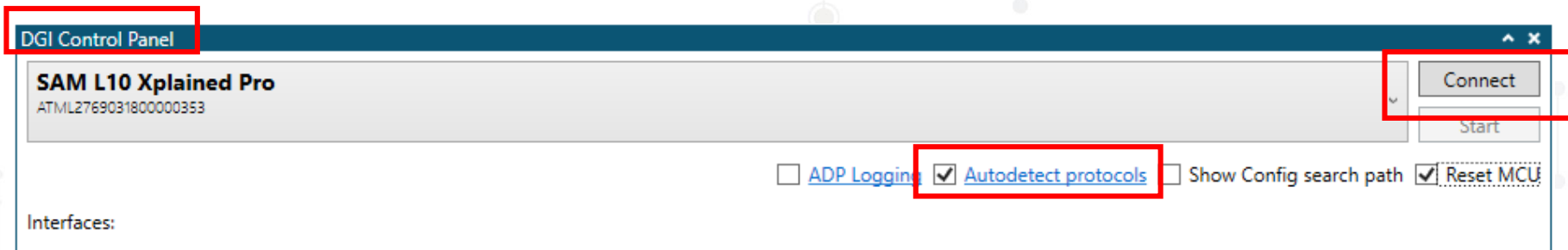


- 将I/O跳线连接到**BYPASS**，使电流绕过I/O源，例如板上ECC508A、USER\_LED和按钮SW0上拉等。

# 实验2——低功耗设计

## 2 配置Data Visualizer以进行电流测量。

- 在Data Visualizer中，打开***DGI Control Panel***（DGI控制面板）窗口。

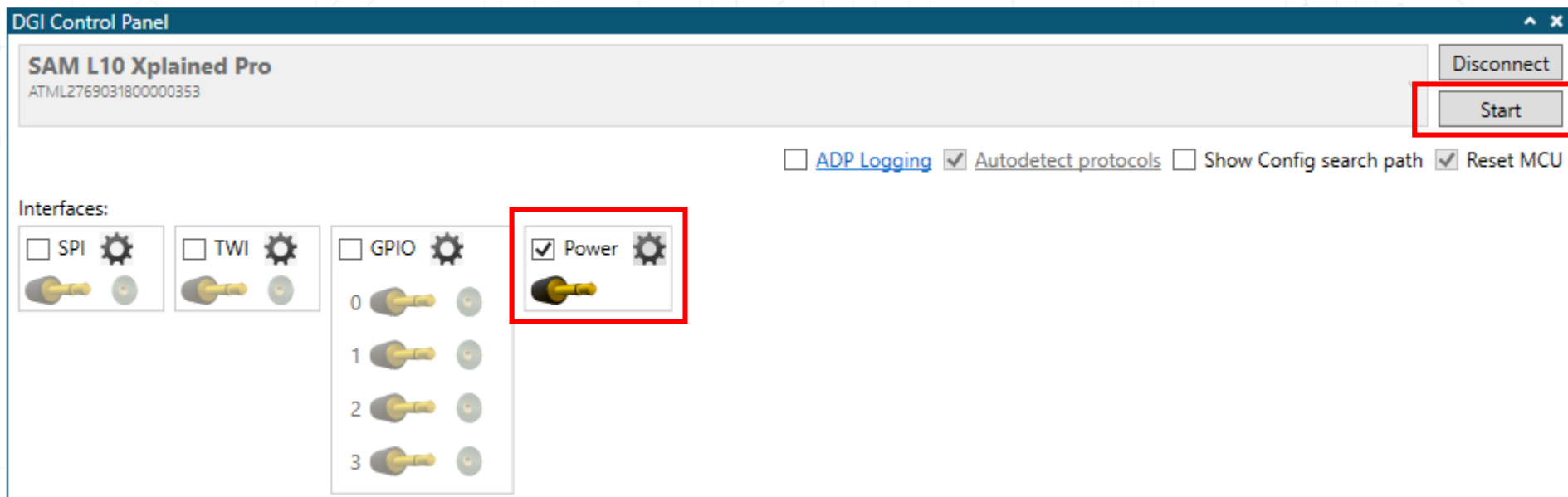


- 单击***Autodetect Protocols***（自动检测协议）的复选框。
- 单击***Connect***（连接）。
- Data Visualizer将指示正在搜索协议...



# 实验2——低功耗设计

- 接口出现在DGI Control Panel中
- 单击**Power**（功耗）的复选框。



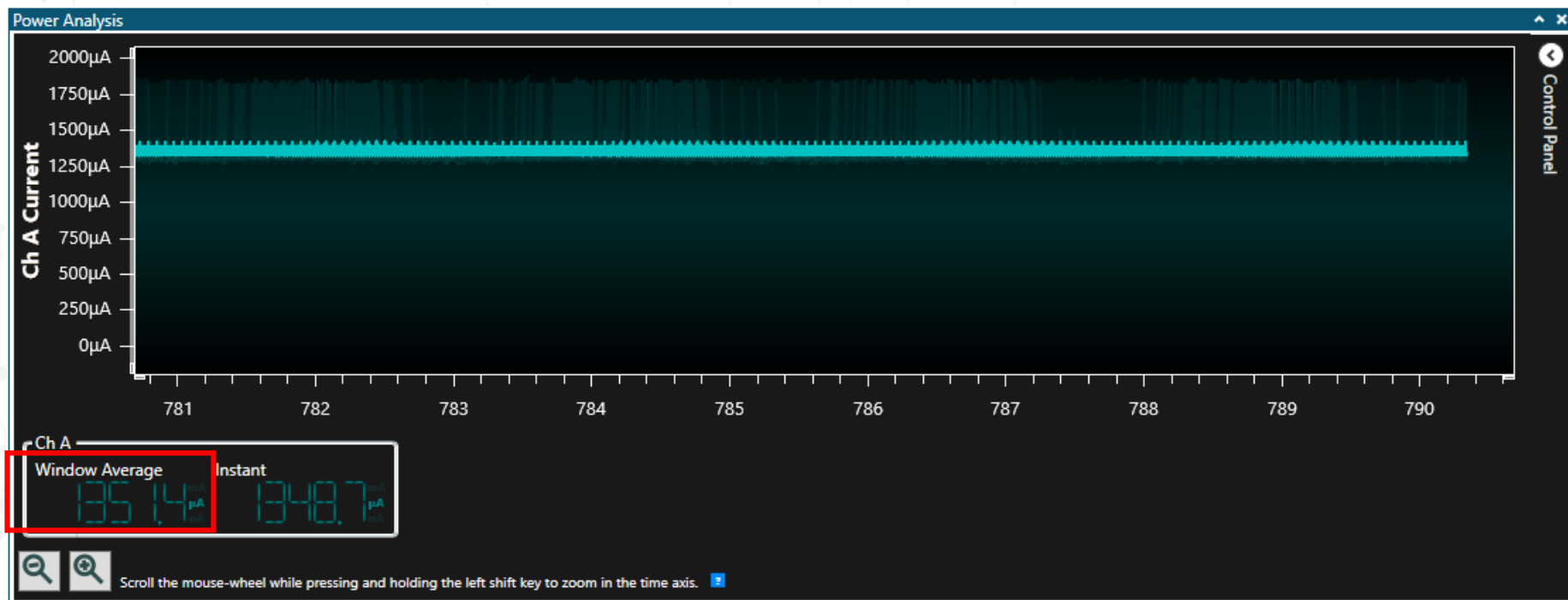
- 单击**Start**（开始）。
- Data Visualizer中将出现一个新的**Power Analysis**（功耗分析）窗口。您可能需要向下滚动来查看此窗口。



# 实验2——低功耗设计

## 3 测量基准平均电流。

- 在 **Power Analysis** 窗口中，请注意未触摸QT7-XPRO时的 **Window Average**（窗口平均）电流。我们将其称为标准触摸配置。



# 实验2——低功耗设计

## 实验笔记

- 接下来，我们将演示通过修改实验1 **START**项目配置来添加低功耗触摸唤醒功能的步骤。

请按照  
实验手册进行操作！

# 实验2——低功耗设计

## 实验笔记

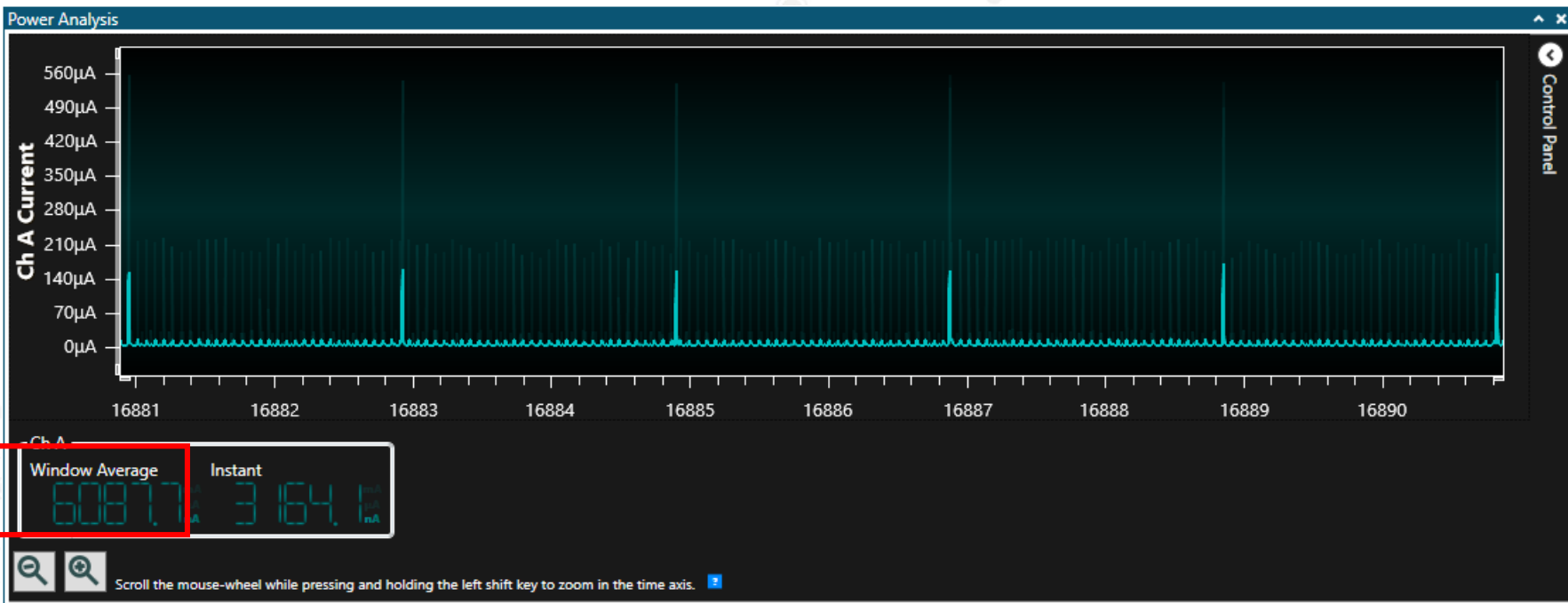
- 接下来的步骤与实验1相似：将**START MPLAB**项目导入**MPLAB X**，然后添加代码以使能触摸功能并点亮**LED**。
- 为了演示最低功耗，需要禁止**Datastreamer**功能以及**UART**引脚功能。此操作作用代码实现。
- 为了节省时间，已预先创建**Lab2**解决方案项目，并进行了必要编辑（标有“**Lab2.**”）。



# 实验2——低功耗设计

## 7 测量低功耗平均电流。

- 在Power Analysis窗口中，请注意未触摸QT7-XPRO时的 **Window Average** 电流。



# 实验2——低功耗设计

## 10 比较低功耗配置

- 继续更改touch.h中的***NODE\_SCAN\_xxMS***参数，如下表所示。
- 每次更改后均执行***Make and Program Device***（编译并编程器件），然后查看Data Visualizer上的***Power Analysis***。
- 在下表中记录**窗口平均**电流以进行比较。

模式	设置	窗口平均电流 (μA)
标准	正常	1349.9
低功耗	NODE_SCAN_64MS	6.0
低功耗	NODE_SCAN_128MS	
低功耗	NODE_SCAN_256MS	
低功耗	NODE_SCAN_512MS	

# 实验2总结



- 使用**SAML10 Xplained Pro**功耗调试器功能基于**Data Visualizer**测量了实验1项目的平均电流。
- 回顾了如何通过修改使用**START**的项目的方法来降低平均电流消耗，从而创建“触摸唤醒”行为。
- 与实验1项目标准配置相比，通过增大实验2项目中的时序参数进一步降低了平均功耗。



# 课程安排

- 什么是电容触摸传感
- **QTouch®**模块化库和**Data Visualizer®**实用程序简介
  - 实验1——
- 低功耗设计
  - 实验2——
- 防潮、防噪声设计
  - 实验3——
- 2D触摸表面
  - 实验4——
  - 实验5——
- 总结



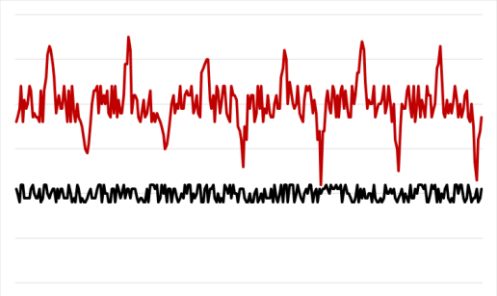
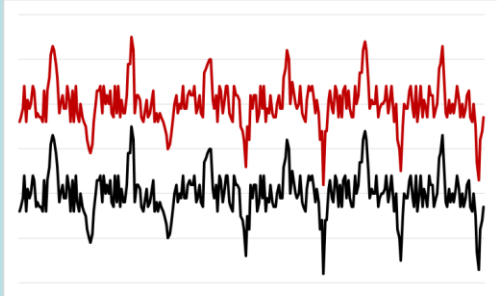


概述:

噪声类型

噪声对策

# 电磁噪声的类型

	差分	共模
		
辐射	<ul style="list-style-type: none"><li>• 荧光灯</li><li>• 靠近前面板的射频发射器</li></ul>	<ul style="list-style-type: none"><li>• 靠近电源线的射频发射器</li></ul>
传导	<ul style="list-style-type: none"><li>• 前面板上的LED</li><li>• 应用中的电机</li></ul>	<ul style="list-style-type: none"><li>• 电源噪声</li><li>• 开关模式PSU</li><li>• 与噪声设备的有线通信</li></ul>

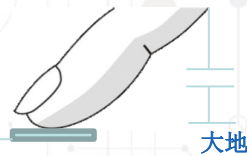


# 触摸系统中的噪声

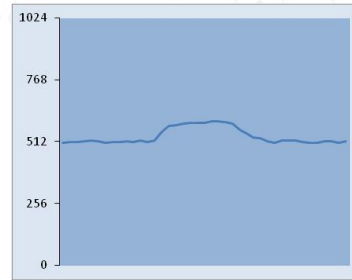
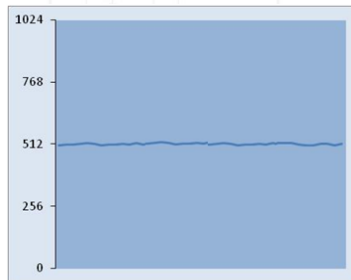
- 辐射噪声
- 传导噪声
- ESD
- 噪声会产生误检

未触摸

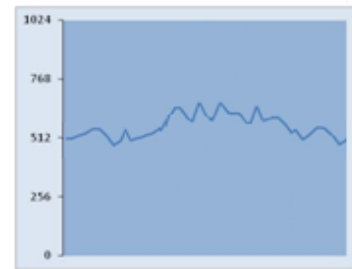
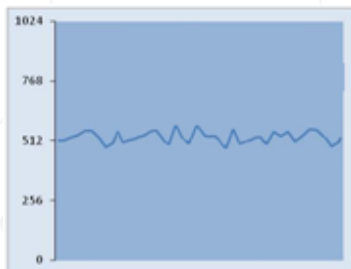
触摸



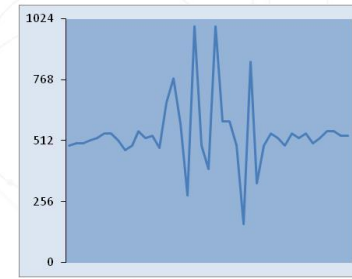
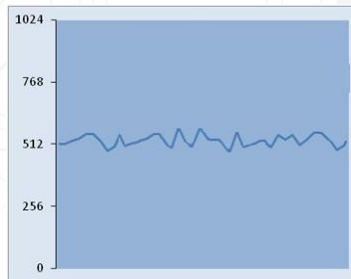
无噪声



辐射噪声



传导噪声





# 抗噪对策

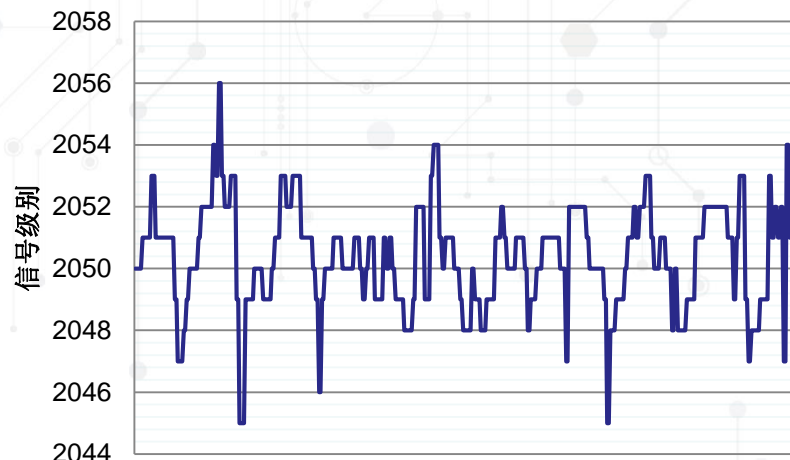
- 传感器调节不足会导致传感器性能下降和抗噪测试失败...因此需要了解应用中的噪声源!!!
- 没有单一的解决方案, 而是通过相应技术之间的折衷来抑制噪声及实现可接受的系统性能。

噪声对策	最擅长滤除...
串联电阻	传导噪声
过采样	辐射/传导噪声
跳频	辐射/传导噪声
检测阈值/滞后	辐射/传导噪声
检测积分	辐射噪声

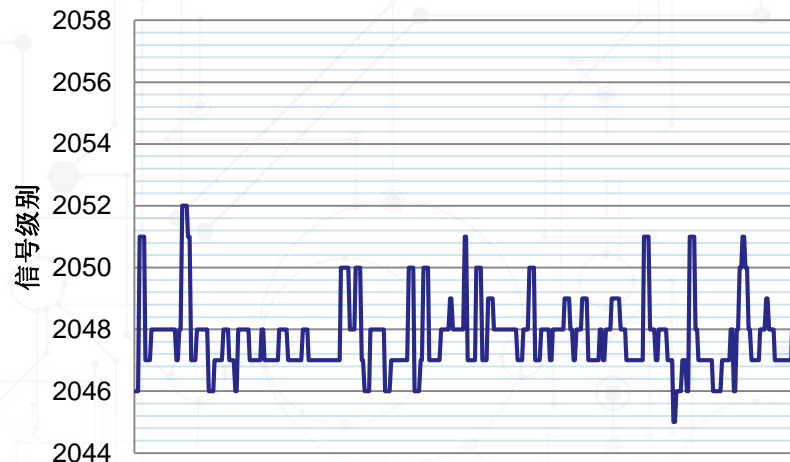


# 滤波等级

- 定义获取的采样数
- 增大滤波等级
  - 提高信噪比 (SNR)
  - 延长采集时间
- 选择取决于噪声等级
- 可基于各个通道配置
- 可配置为1至64间的值  
(`FILTER_LEVEL_1`、`FILTER_LEVEL_2`、`FILTER_LEVEL_4`、`FILTER_LEVEL_8`、`FILTER_LEVEL_16`、`FILTER_LEVEL_32`和`FILTER_LEVEL_64`)



滤波等级32

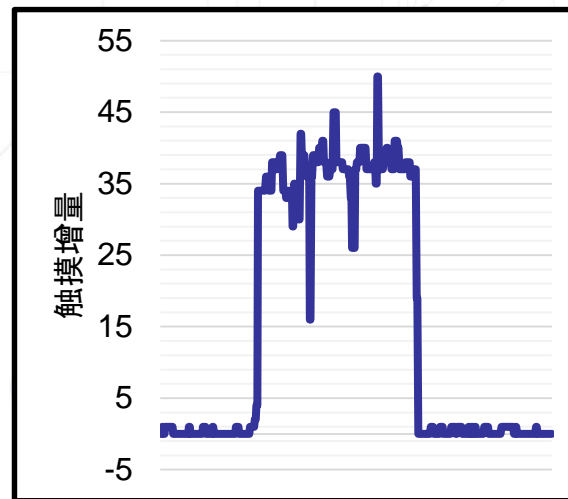


滤波等级64

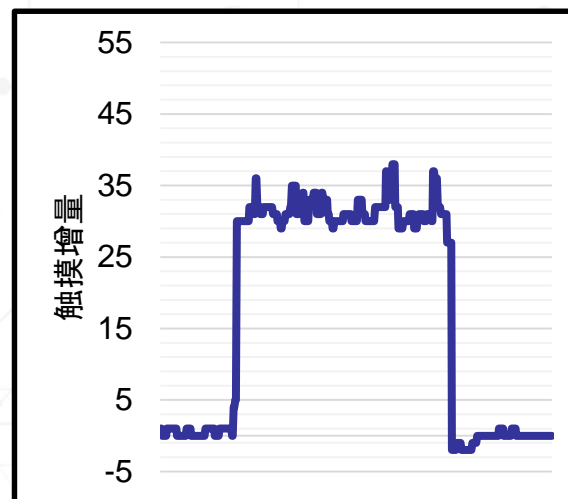


# 串联电阻

- 可选择在传感线路上使用内部串联电阻
- 增大串联电阻
  - 提高信噪比 (SNR)
  - 需要更长的充电时间
- 选择取决于噪声等级
- 可基于各个通道配置
- 可配置内部选项, 可实现互电容
  - SAM和megaAVR®: 0Ω、20 KΩ、50 KΩ和100 KΩ
  - tinyAVR®: 0Ω、20 KΩ、50 KΩ、75 KΩ、100 KΩ和200 KΩ
- 使用外部电阻实现自电容



串联电阻0Ω



串联电阻100 KΩ



**MICROCHIP**  
**MASTERS 2019**

高级噪声对策

# 跳频模块



# 跳频 Frequency Hop

- 如果噪声频率接近采集频率，则触摸信号的噪声等级会增大
- 必须更改采集频率以减少噪声

## 跳频

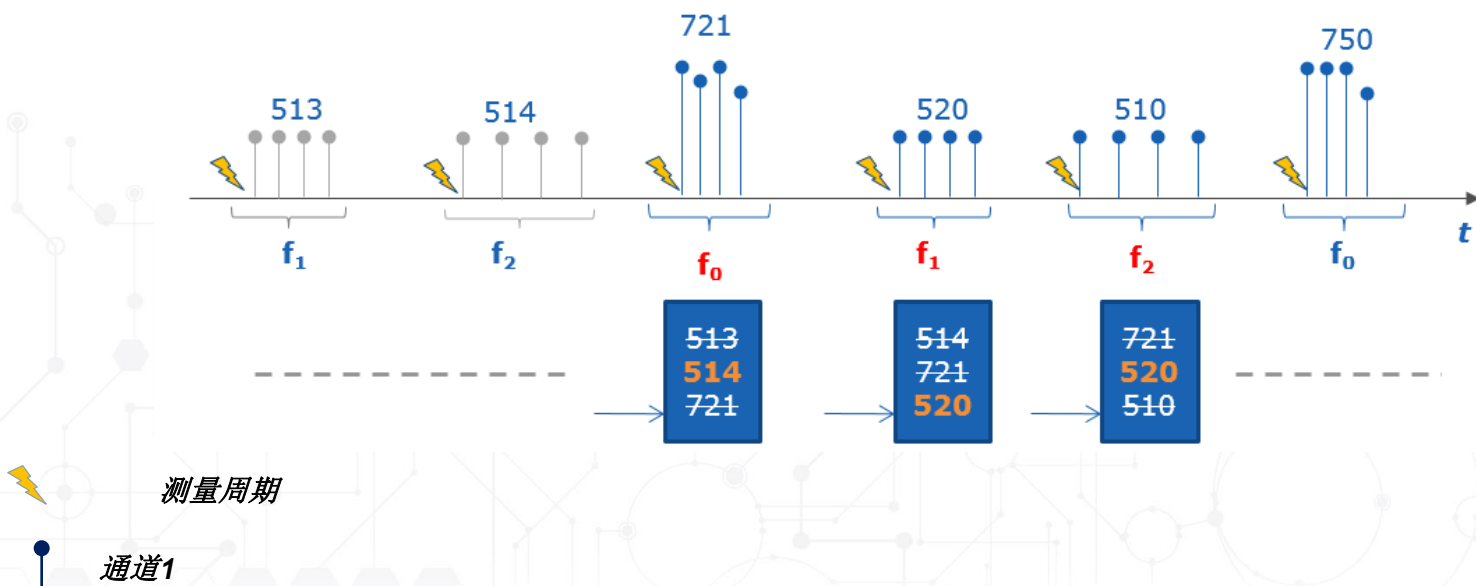
- 用户必须从包含十六个频率的列表中选择一组（N个）频率
- 触摸库可提供使用不同频率执行连续触摸测量的选项
- 对所得的N个信号值应用中值滤波器





# 跳频 Frequency Hop

- 配置示例：
  - 滤波等级 = 4，三个频率 (N = 3)
- 噪声频率接近 $f_0$ 采集频率





# 跳频 Frequency Hop

- 为了选择正确的频率，用户需要使用不同的频率组合进行迭代
- 选择可降低噪声等级的频率
- 适用于**固定的**噪声频率
- 增加频率数量（N）
  - 提高抗噪能力
  - 延长响应时间
  - 增大存储器（RAM）消耗



# 跳频自动调节 Frequency Hop Auto Tune



- 监视每个频率的噪声等级
- 噪声等级符合以下情况时动态替换噪声最多的频率：
  - 保持一致
  - 高于阈值
- 适用于变化的噪声频率



# 跳频自动调节

## Frequency Hop Auto Tune

- 跳频和自动调节之间的差异

跳频	跳频自动调节
手动调节频率	自动调节频率
适用于频率固定的噪声	适用于频率变化的噪声
较小的闪存和RAM存储器	较大的闪存和RAM存储器
用户定义的频率列表	使用所有频率



概述:

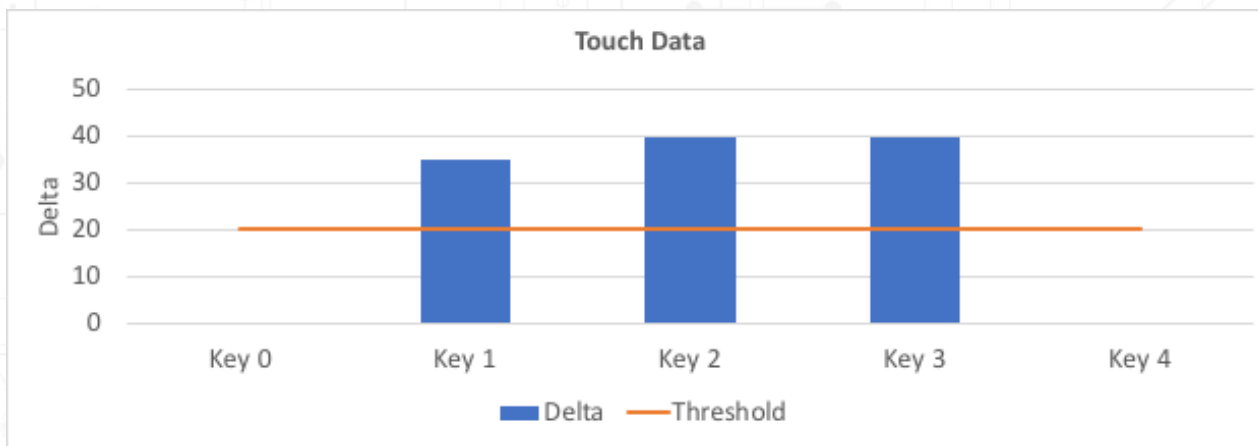
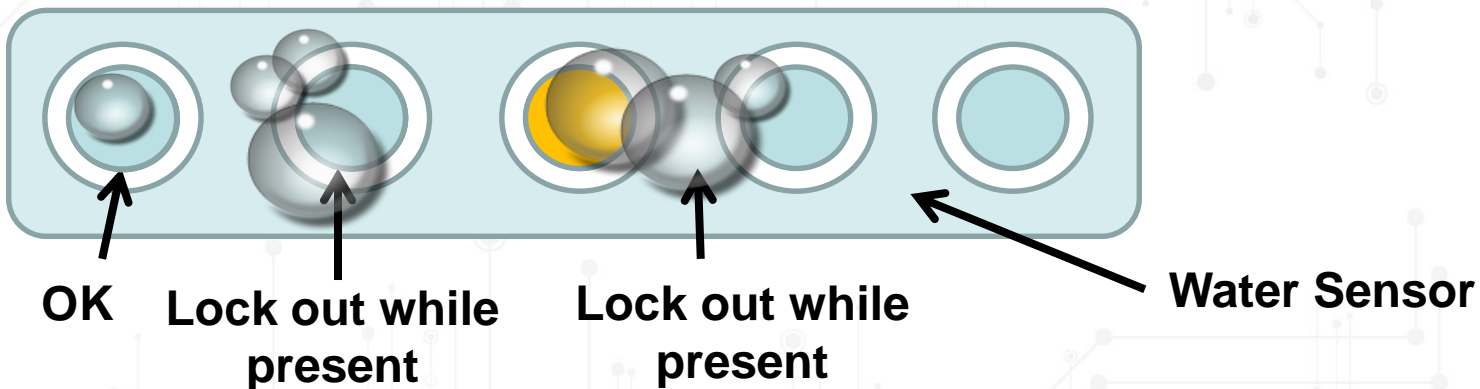
水的影响

驱动屏蔽



# Water Sensor - Lockout

- 一个额外的大电极围绕所有电极
- 所有自电容传感器和**Water Sensor**之间存在触摸





**MICROCHIP**  
**MASTERS 2019**

# 驱动屏蔽DRIVEN SHIELD

# 驱动屏蔽Driven Shield——需求是什么？

- 传感器周围的地实现了良好的抗噪性
- 传感器周围的地降低了防潮性
- 需要某种功能代替地来实现抗噪性和防潮性

驱动屏蔽可实现  
抗噪性和防潮性

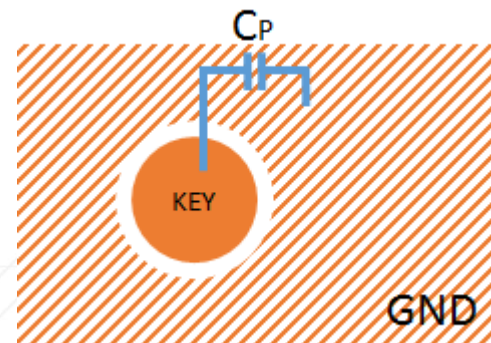
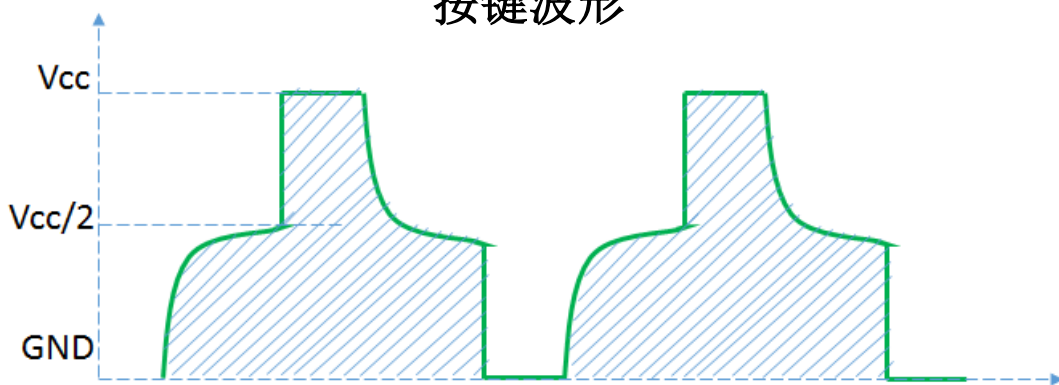




# 驱动屏蔽 Driven Shield ——工作原理是什么？

- 由于传感器电极和地之间的电势差而形成 $C_p$ 。
- $C_p$ 仅在阴影区域存在。 ( $q = C * V$ )
- 当按键和GND上存在水时， $C_p$ 会增加，因为水有助于耦合更多的电场。

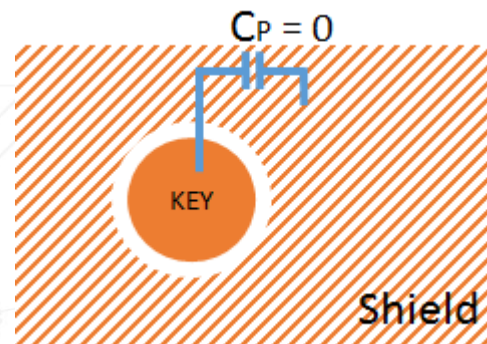
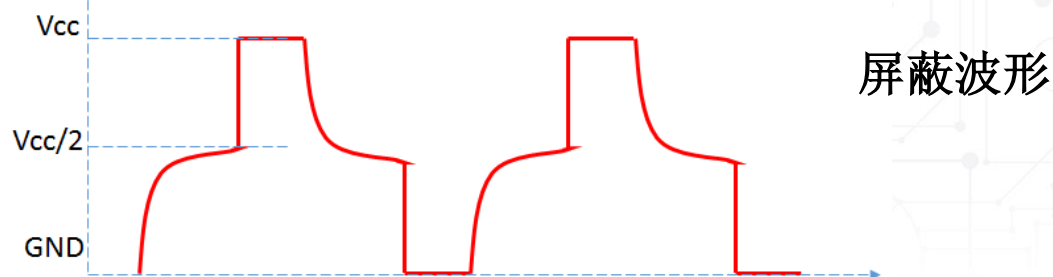
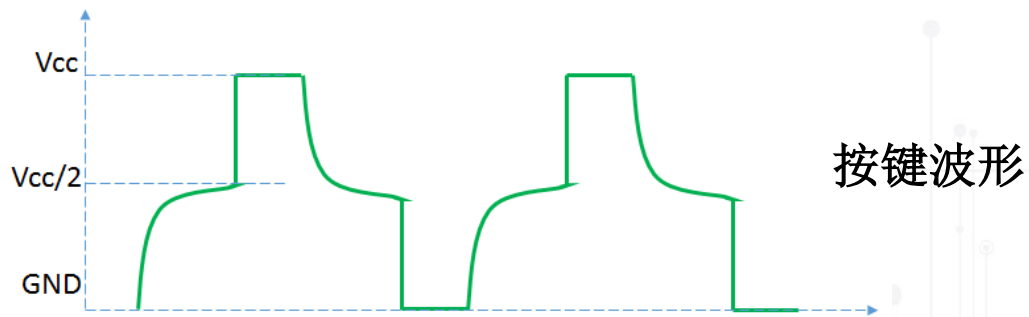
按键波形





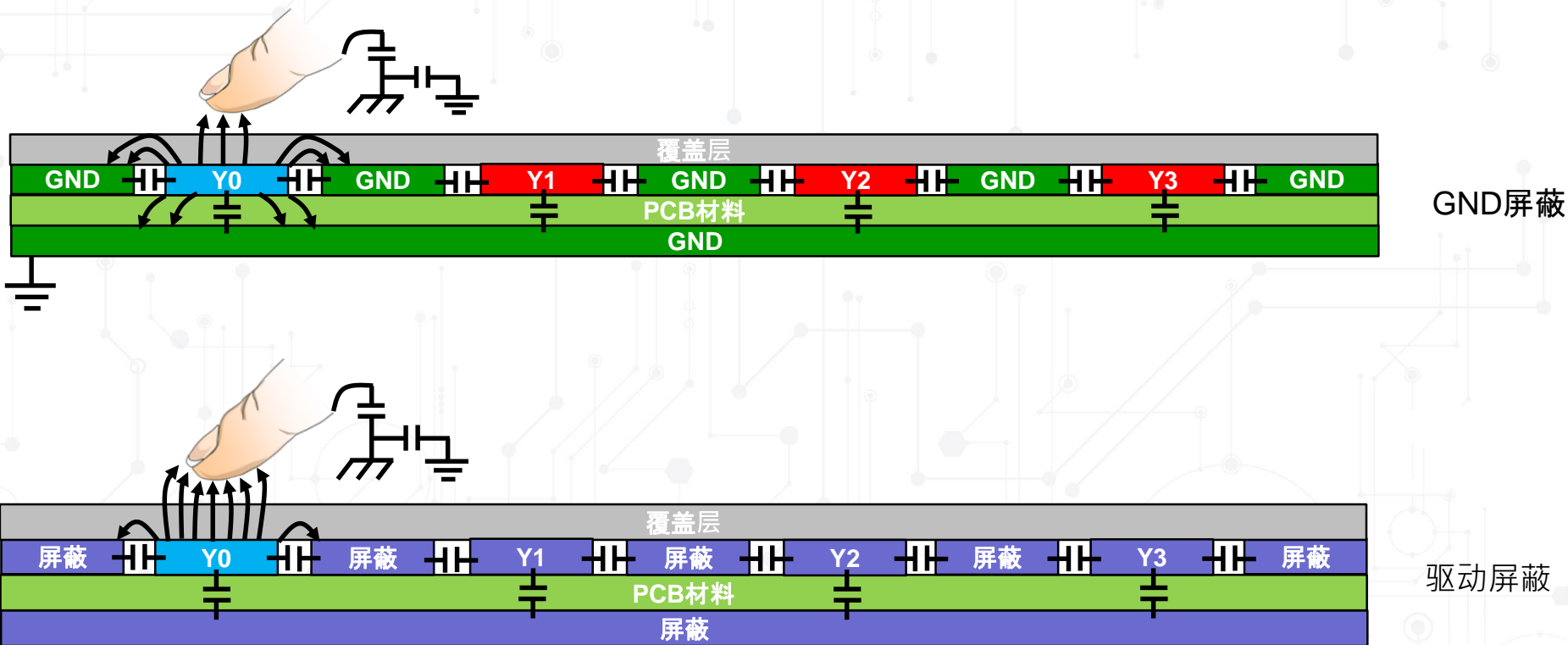
# 驱动屏蔽 Driven Shield ——工作原理是什么？

- 驱动屏蔽以与传感器电极相同的电势进行驱动。
- $C_p$ 为零，因为没有电势差。
- 传感器和屏蔽层之间的水不会导致 $C_p$ 发生任何变化，因为 $C_p$ 不存在。



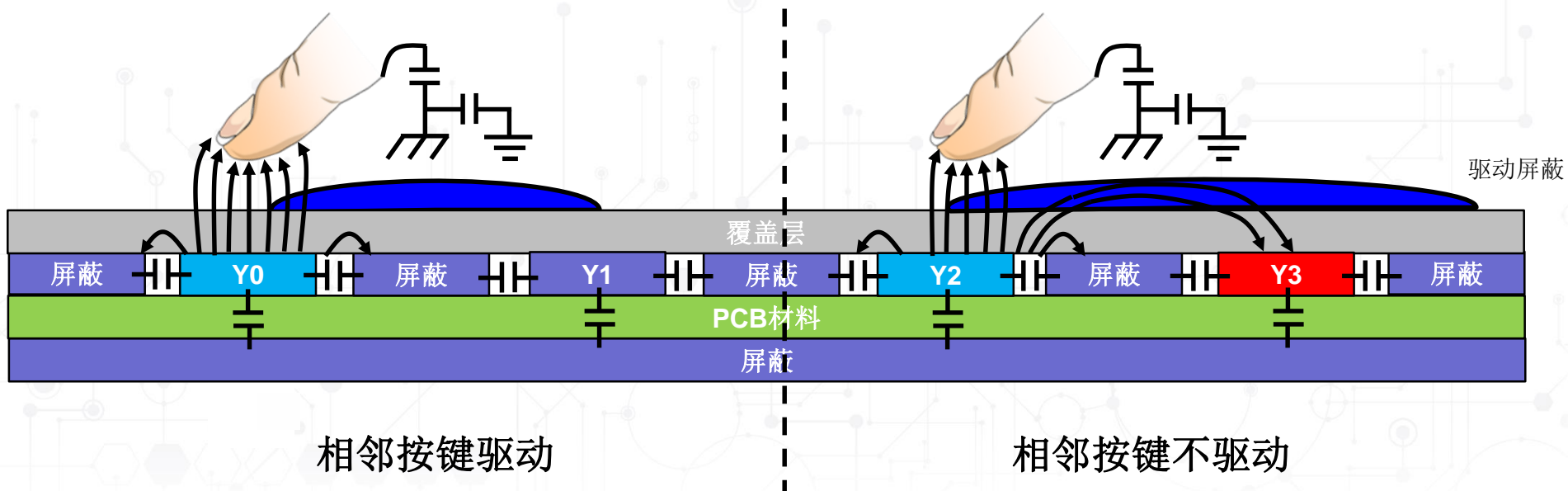
# 驱动屏蔽Driven Shield——提高SNR

- 场不会被附近的地转移（返回路径）
- 场可高效地到达表面——提高灵敏度



# 驱动屏蔽Driven Shield——防潮性能

- 由于水分没有在传感器和GND之间形成耦合，因此潮湿时不会发生误检
- 相邻按键也会随驱动屏蔽一起驱动，以提高防潮性能



# 使能驱动屏蔽 Enabling Driven Shield

## 增加驱动屏蔽

- 单击 *Enable Driven Shield*（使能驱动屏蔽）
- 确认已选中 *Enable Dedicated Shield Pin*（使能专用屏蔽引脚）
- 配置专用屏蔽引脚：Y/2（PA02）。



The screenshot shows the configuration tool interface. On the left is a sidebar with 'Pins' selected. At the top are 'Auto Assign' and 'Reset' buttons. Below are 'Pin Selection', 'Driven Shield', and 'Debug' tabs. The 'DRIVEN SHIELD PLUS' section contains the following settings:

- Enable Driven Shield
- Enable Dedicated Shield Pin
- shield pin: Y/2 (PA02)

A red box highlights the 'Pins' sidebar, the 'Driven Shield' tab, and the 'Enable Driven Shield' and 'Enable Dedicated Shield Pin' checkboxes. A red arrow points from the blue text to the 'Enable Dedicated Shield Pin' checkbox.

选中 **Enable Driven Shield**  
、 **Enable Dedicated Shield Pin**  
并配置屏蔽引脚



## 实验3——防潮性和抗噪性： 评估采用防潮和抗噪对策前后的触摸系统

# 实验3——防潮性和抗噪性

## 目标

- 演示在潮湿环境下，驱动屏蔽对触摸系统的影响。
- 演示低成本的噪声发生器。
- 演示在噪声环境下，相应对策对触摸系统的影响。

# 实验3——防潮性和抗噪性

## 实验笔记

- 为了节省时间，已预先创建**Lab3**解决方案项目，并进行了必要编辑（**Lab3.X**）。
- 使用水评估潮湿环境下的触摸。
- 具有适当固件的**AVR-P4**（**ATAVRFEB-P4**）板用于生成模拟噪声信号，以在噪声环境下评估触摸。



# 实验3——防潮性和抗噪性

## 实验笔记

- 实验3解决方案使用SW0在两个触摸配置功能之间切换：

参数	disable_robust_mode()	enable_robust_mode()
DEF_SENSOR_TYPE	NODE_SELFCAP	NODE_SELFCAP_SHIELD
PA02	GPIO输出低电平	驱动屏蔽
DEF_TOUCH_DET_INT	2	4
FILTER_LEVEL	4	32
THRESHOLD	20	50
DEF_FREQ_AUTOTUNE_ENABLE	0	1
DEF_MEDIAN_FILTER_FREQUENCIES	FREQ_SEL_0, FREQ_SEL_0, FREQ_SEL_0	FREQ_SEL_0, FREQ_SEL_1, FREQ_SEL_2

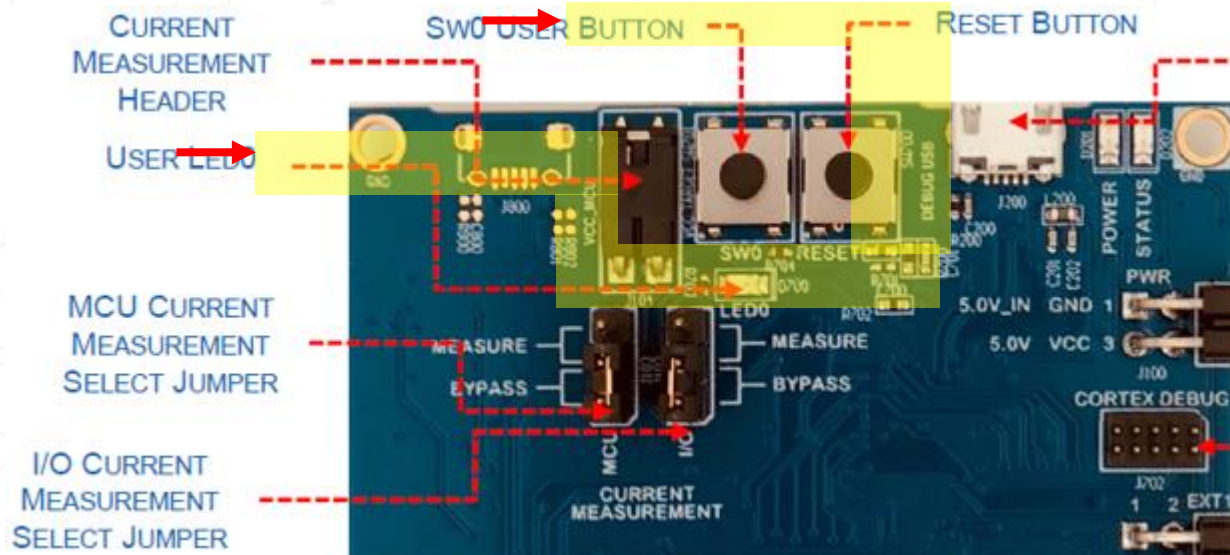
# 实验3——防潮性和抗噪性

## 硬件工具

实验3解决方案使用disable\_robust\_mode()复位，同时用户LED0熄灭。

按下SW0切换到enable\_robust\_mode()，同时用户LED0点亮。

按下SW0用户按钮  
切换回  
disable\_robust\_mode()，  
同时用户LED0熄灭。



# 实验3总结

- 评估了潮湿环境下驱动屏蔽对触摸系统的影响。
- 演示了低成本的噪声发生器。
- 评估了在噪声环境下，相应对策对触摸系统的影响。



# 课程安排

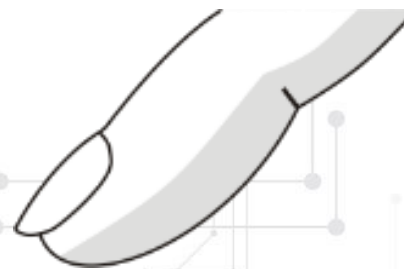
- 什么是电容触摸传感
- **QTouch®**模块化库和**Data Visualizer®**实用程序简介
  - 实验1——
- 低功耗设计
  - 实验2——
- 防潮、防噪声设计
  - 实验3——
- **2D触摸表面**
  - 实验4——
  - 实验5——
- 总结



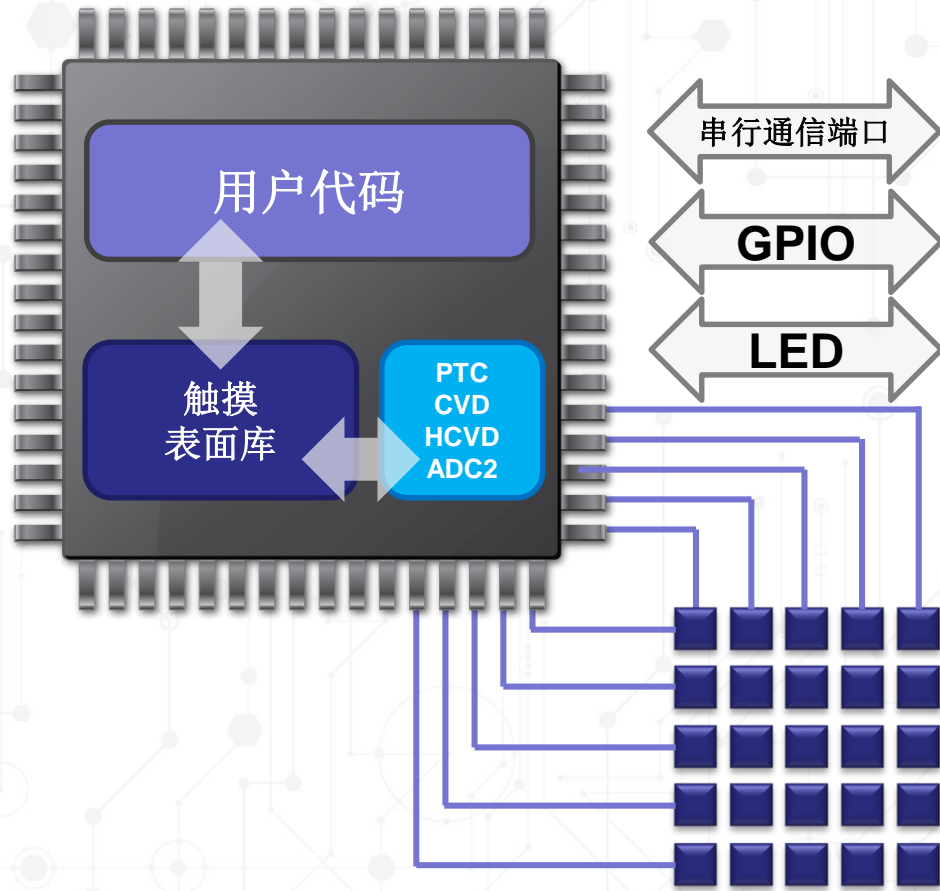
# MICROCHIP 2019 6102 MASTERS

中国技术精英年会(第二十届)

## 电容式触摸板基础知识

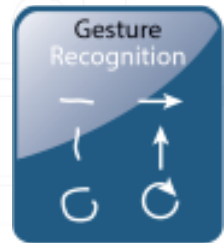


# 2D——表面触摸库——概述



将触摸表面与主机处理器结合的简单方法

- 低于**5  $\mu\text{A}$** 的待机电流
- 最高**100 Hz**的报告速率
- 最高**5.5"**对角线
- 配置实用程序可自定义大小、速度和性能
- 适用于**PCB、FPC、玻璃和薄膜**传感器
- 它可以报告单指触摸坐标或单指和双指手势



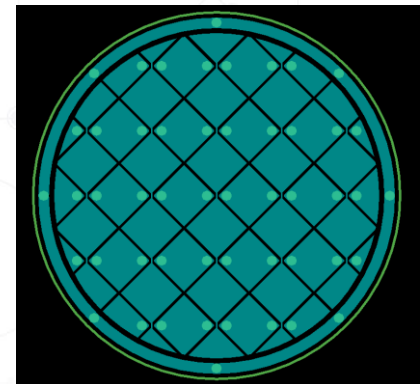
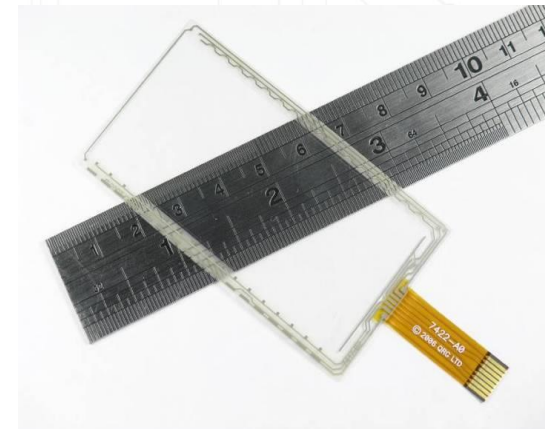
# 传感器材料

## 光学透明:

- ITO——最佳光学性能，填充开放区域，用于保持光学一致性——最普遍。
- PDOT——图案可见，但成本比ITO低约三分之一。
- 银（金属）纳米线——更好的光学性能——成本低于ITO。
- 碳纳米管——前景广阔的新技术，成本可能低于ITO，光学性能良好

## 不透明传感器:

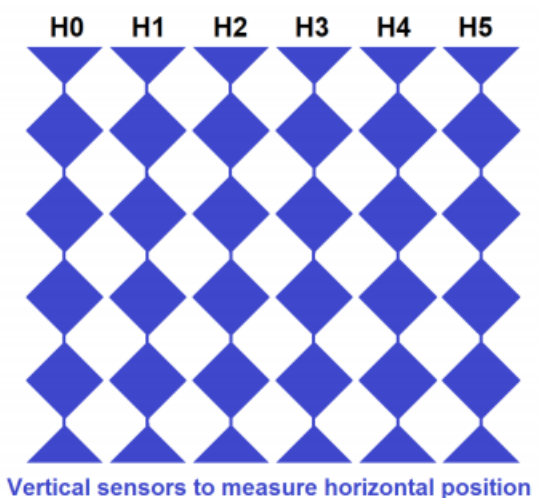
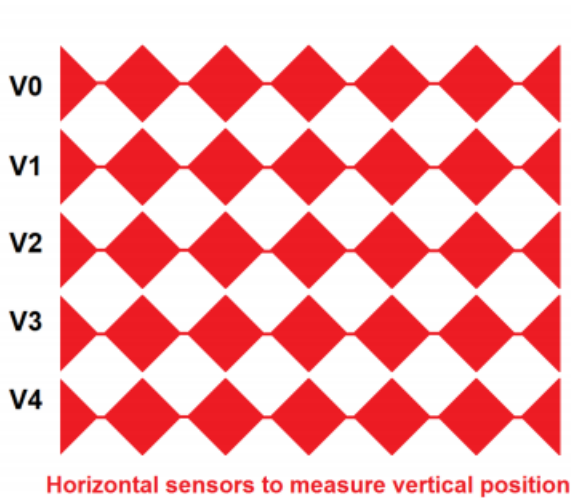
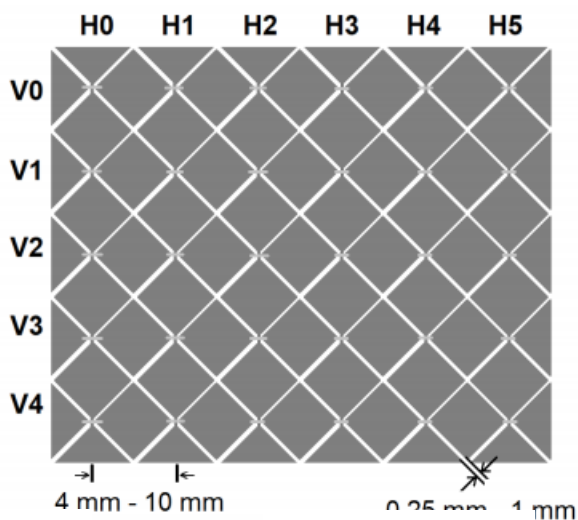
- PCB——成本最低且易于实现
- 柔性电路





# 布局示例

- 菱形图案：



- 菱形图案可位于具有交叉线的一层上，也可位于两层上。两层设计需要仔细对齐，即对准。

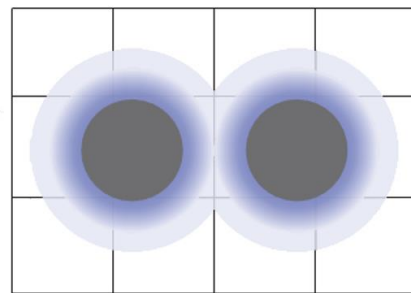




# 手指间隔

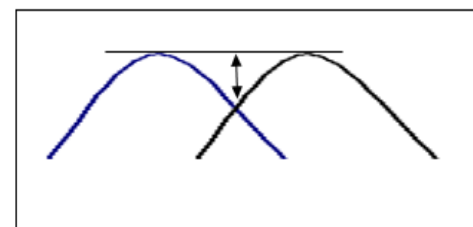
## “触摸/手指间隔”的定义

- 要使触摸IC将两根手指的触摸事件报告为单独的触摸事件，手指之间所需达到的最短距离
- 主要由传感器网格决定



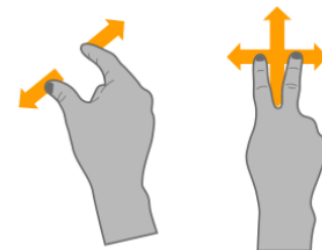
## 双指手势的关键参数

- 确定缩放手势的开始/结束



对于选择控制器和确定所需传感器的数量十分重要

- 经验法则：手指间隔 =  $2.1 \times$  节点间距



# 精度和线性度

## 优势

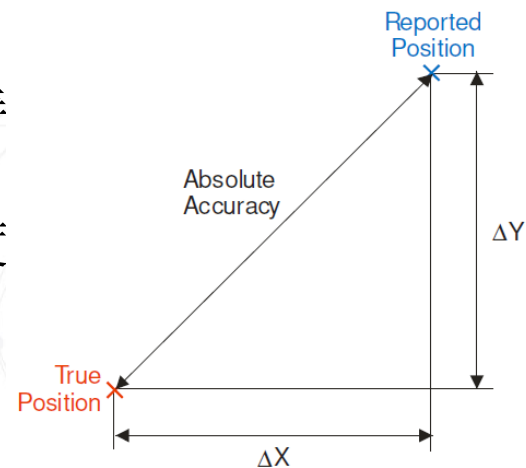
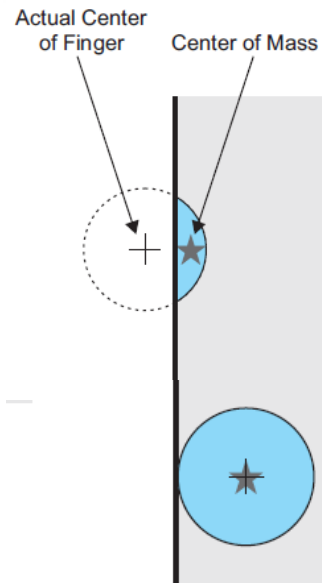
- 即使手指很小也可以正确选择选择对象
- 手指笔直移动应产生均匀线条
- 对于显示触摸位置的HMI十分重要

## 挑战

- 位置通过插值计算并受噪声影响
- 边缘上的电容分布呈非线性
- 手指只覆盖边缘上传感器的一部分

## 配置

- 增大显示活动区域上的传感器活动区域，以帮助改善边缘效应
- 如果可能，请使用较小的节点间距，以帮助提高精度和线性度



# 部件选型指南

要求	MCU或MPU (具有2D触摸表面库)	maXTouch®控制器
传感器节点数	< 100	> 100
报告速度	快	更快（每条线路都有ADC，多点触摸无复用）
手指数	1或2	1至16
应用	简单应用，例如触摸板或低成本单芯片图形用户界面（对角线最长5.5"）。	在单芯片中集成了手套触摸、手写笔、防潮功能、手势和EMI/EMC算法以减轻主机MPU负担的复杂应用



# MICROCHIP 2019 MASTERS

中国技术精英年会(第二十届)

## 开发生态系统





# 我们目前所处的位置:

- **MPLAB X**
- **Atmel Studio**

集成开发环境

- **MCC**
- **Harmony (版本3)**
- **START**

触摸代码配置工具

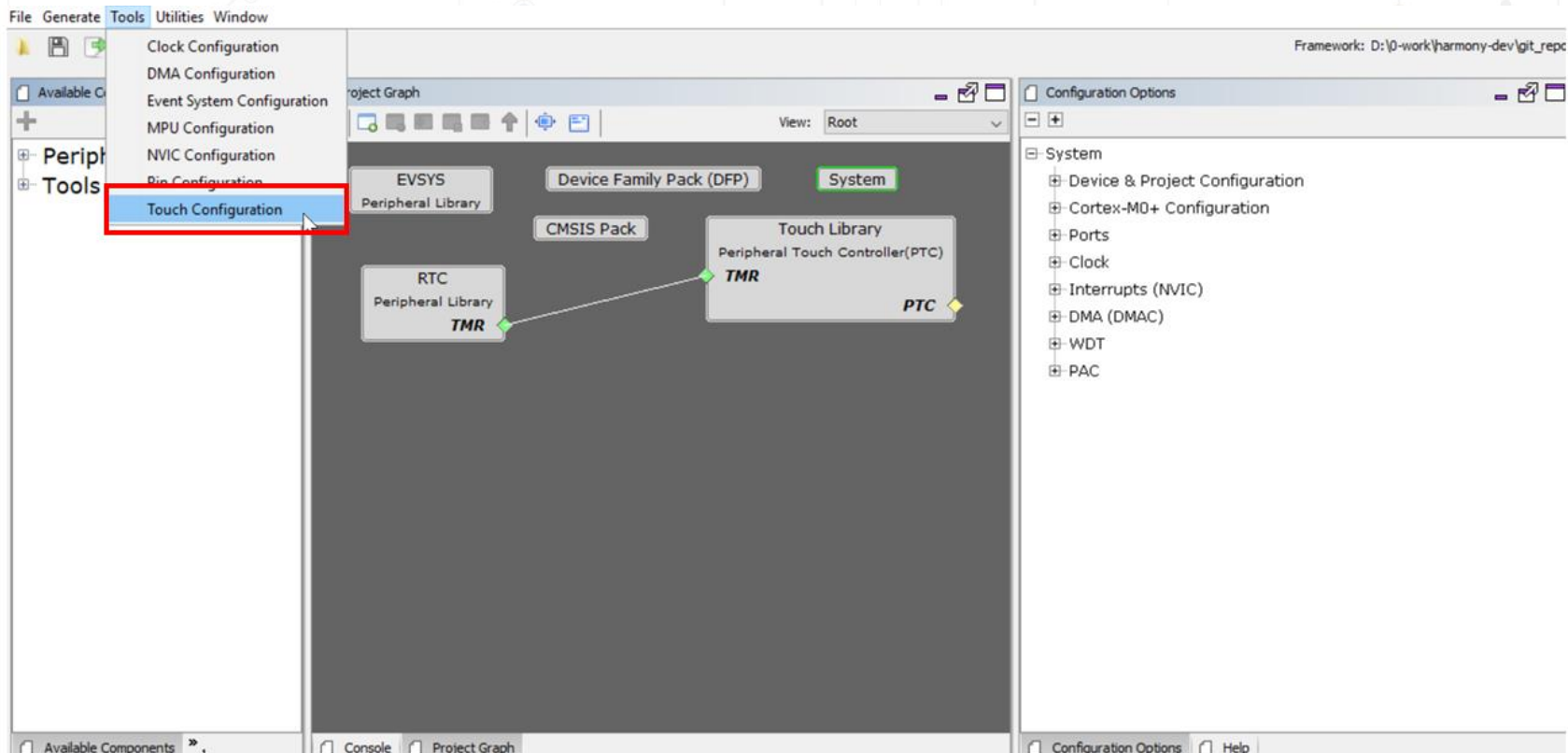
- **Data Visualizer**
- **2D Touch Surface GUI**

性能验证工具——  
适用于任何工具链



# Harmony

- 目前，**Harmony**版本3的支持有限（**SAMC**），但是会持续添加更多器件。
- 将支持**SAM**和**PIC32C**器件



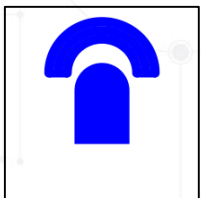


# 2DTouch Surface GUI

- 可视化2D触摸响应和手势
- 动态调节传感器
- 查看事件
- 配置手势



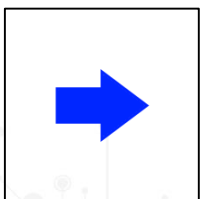
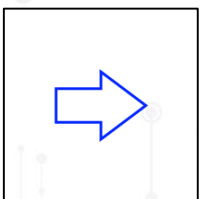
# 2D Touch Surface GUI可报告以下单 指表面手势



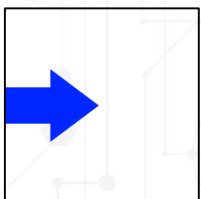
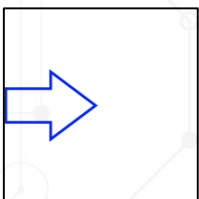
单击以及单击并按住



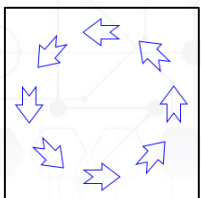
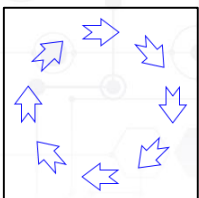
双击和手掌检测



滑动以及滑动并按住



边缘滑动以及滑动并按住

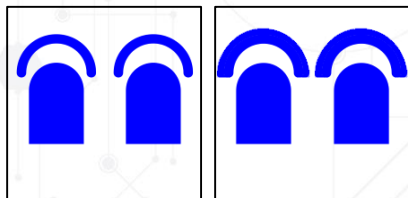


顺时针和逆时针触摸滚轮

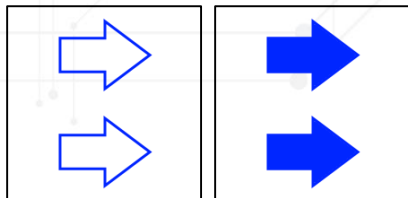




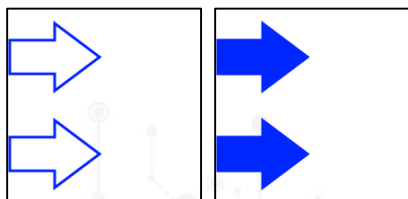
# 2D Touch Surface GUI可报告以下双指表面手势



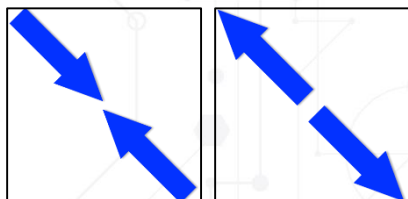
双指单击以及单击并按住



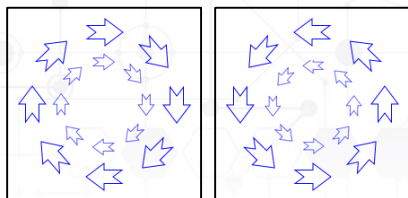
双指滑动以及滑动并按住



双指边缘滑动以及滑动并按住



放大和缩小



双指顺时针和逆时针触摸滚轮



# MICROCHIP 2019 S102 MASTERS

中国技术精英年会(第二十届)

## 2D固件库





# 触摸库

用户代码

API

传感器配置

按钮状态报告

噪声滤波

滑动条插值

噪声滤波

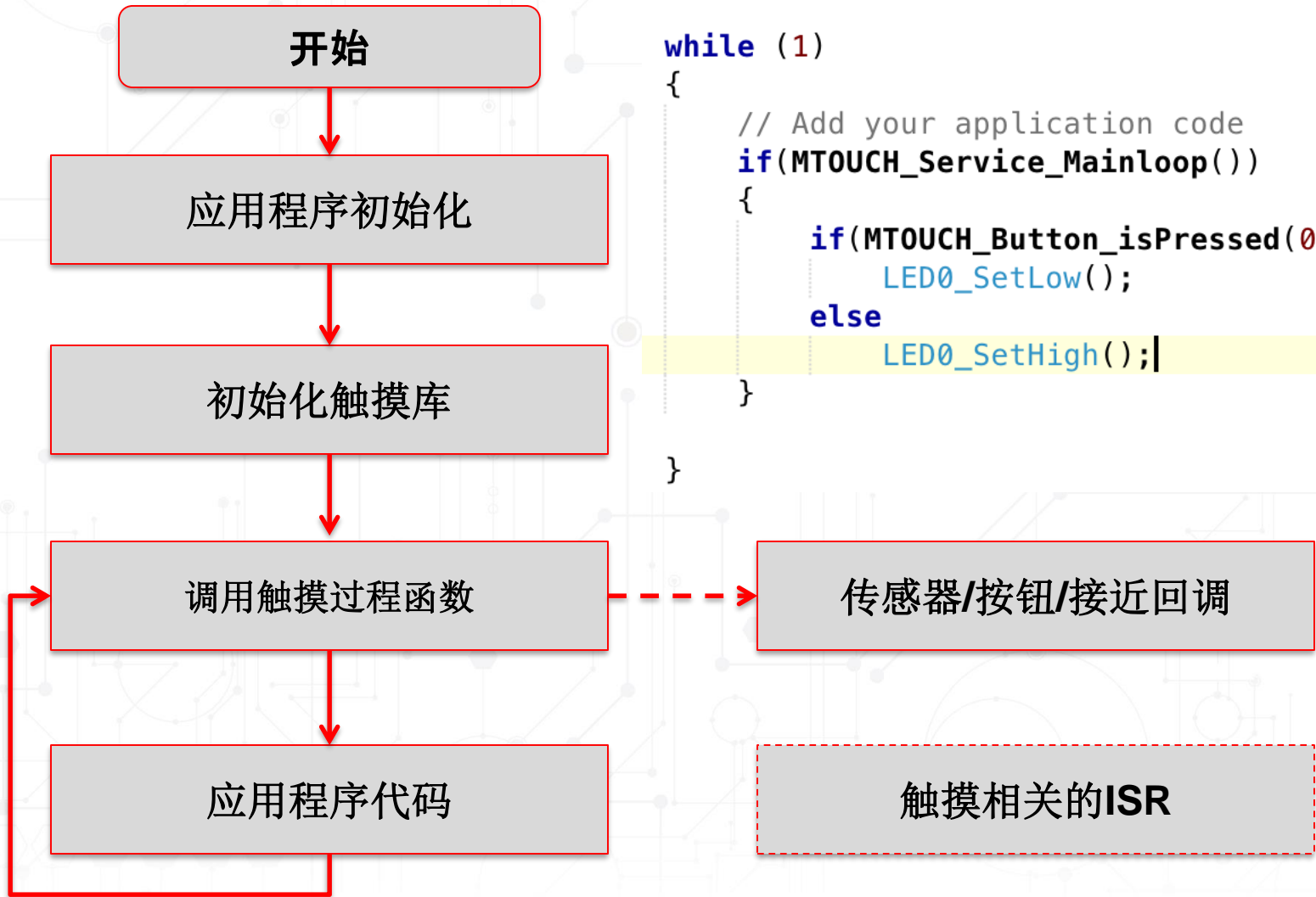
初始传感器校准

触摸去抖

运行时传感器重新校准

PIC/AVR/SAM触摸库

# 顶层流程图



```
while (1)
{
    // Add your application code
    if(MTOUCH_Service_Mainloop())
    {
        if(MTOUCH_Button_isPressed(0))
            LED0_SetLow();
        else
            LED0_SetHigh();
    }
}
```



# 2D库功能

- **主要功能:**
  - 触摸坐标检测
  - 手势检测



# 实验4: 触摸坐标API

- 了解与读取触摸坐标相关的API
- 使用触摸坐标API根据传感器上的触摸位置点亮LED



# 实验5: 手势检测API

- 了解与读取检测到的触摸手势相关的**API**
- 使用手势**API**根据检测到的手势按顺序点亮**LED**



# MICROCHIP 2019 MASTERS

中国技术精英年会(第二十届)

## 2D触摸表面开发工具





# 2D触摸表面演示

- PIC 2D防水2D触摸演示板——DM164149

- AVR 2D防水2D触摸演示板——DM080101

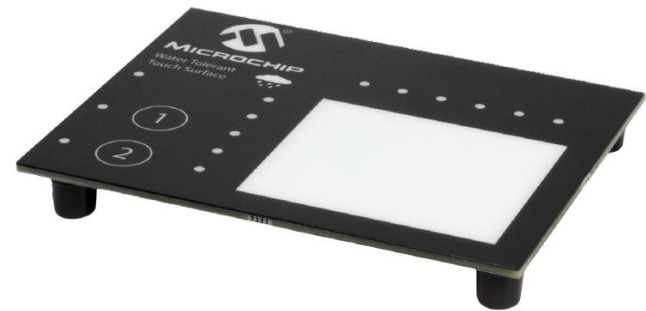
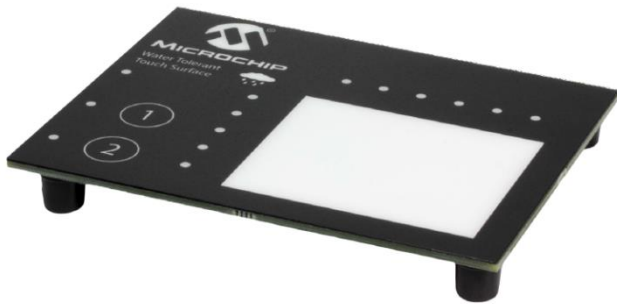
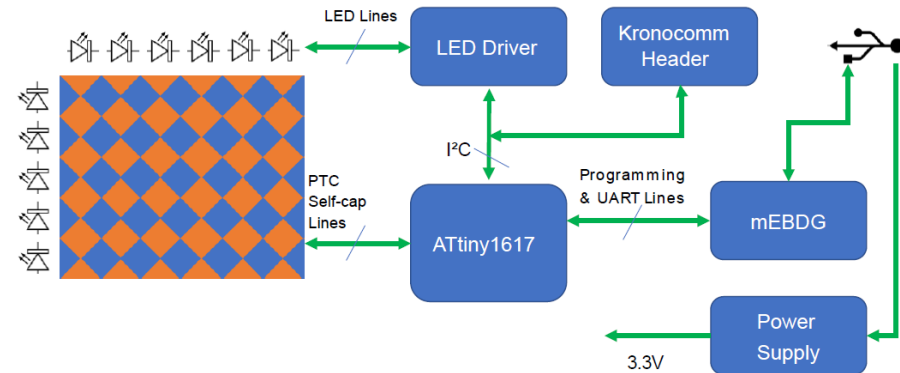
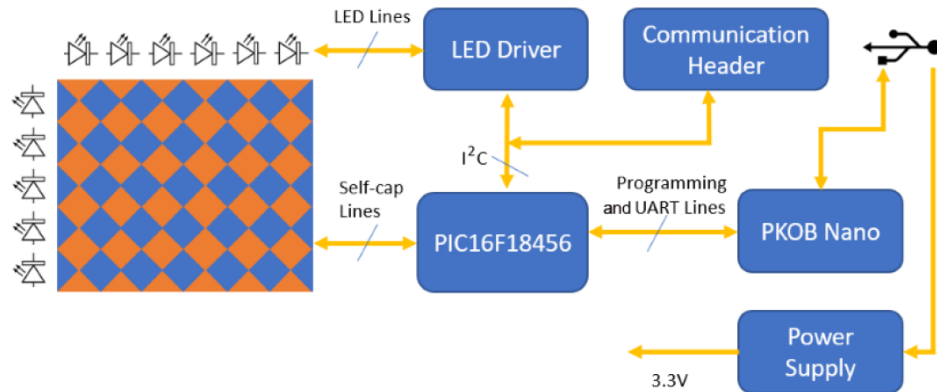


Figure 1-1. Functional Block Diagram

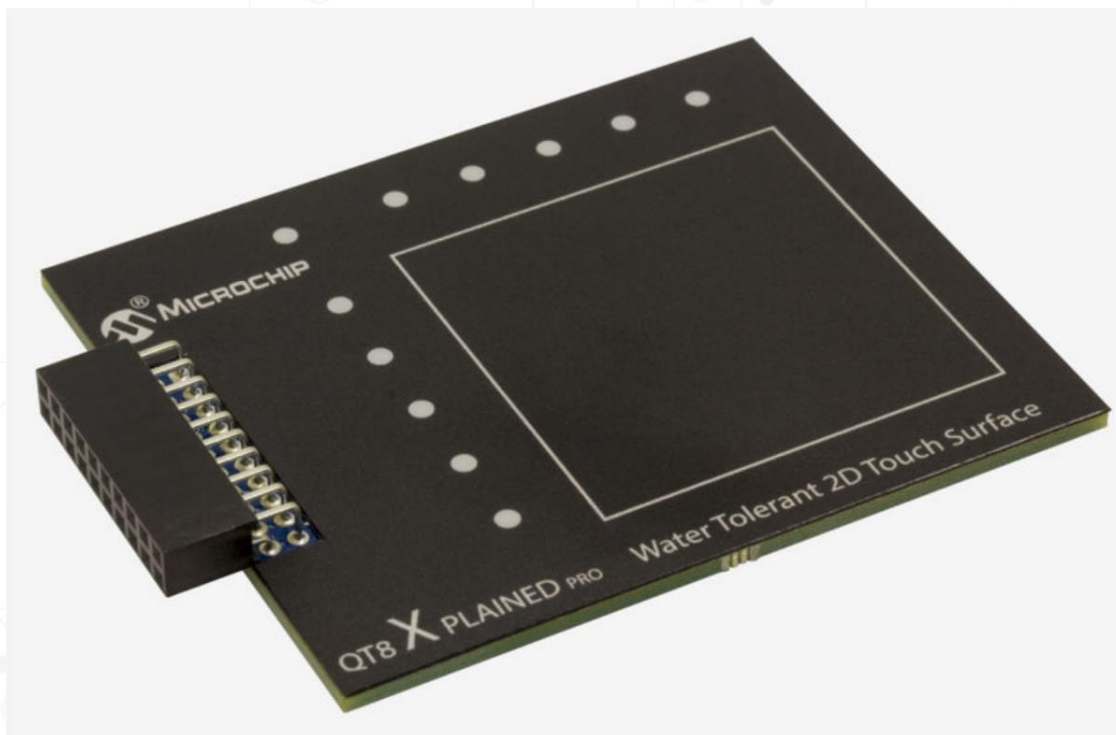
Functional Block Diagram





# Pro工具说明

- 使用**QT8**（部件编号**AC164161**）将**Xplained Pro**板连接到**2D**触摸传感器：





# 课程安排

- 什么是电容触摸传感
- **QTouch®**模块化库和**Data Visualizer®**实用程序简介
  - 实验1——
- 低功耗设计
  - 实验2——
- 防潮、防噪声设计
  - 实验3——
- **2D**触摸表面
  - 实验4——
  - 实验5——
- 总结

# 总结

## 经过今天的学习，您能够...

- 能够使用**Microchip**的**PIC®**、**AVR®**和**Arm® Cortex MCU**为汽车、家电、工业和消费类等市场提供出色的触摸解决方案。
- 能够使用代码配置程序基于**Microchip Arm Cortex MCU**硬件平台轻松创建稳健的电容式触摸应用。
- 能够了解如何创建项目来满足用户的各种系统要求，包括低功耗、潮湿和噪声等使用环境。
- 通过**2D**触摸板的演示项目，能够了解**Microchip 2D**触摸板的相关设计、开发和调试等内容。



# 其他培训链接

## mTouch®电容触摸传感库

- [触摸传感](#)
- [MCC mTouch®电容触摸传感库](#)
- [低成本mTouch®评估工具包示例](#)

## Atmel QTouch®电容触摸传感库

- [Atmel START QTouch®模块化库](#)
- [使用Data Visualizer可视化触摸调试数据](#)



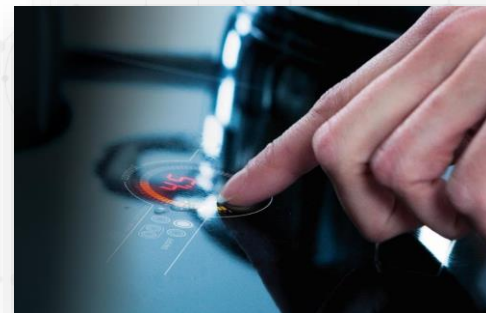
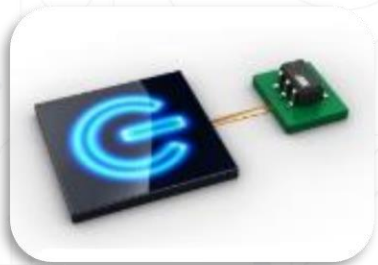
# QTouch®库 应用笔记

- **PTC相关的应用笔记**

- [AT09363](#): PTC Robustness Design Guide
- [AT12405](#): Low Power Sensor Design with PTC
- [QTAN0079](#): Buttons, Sliders and Wheels Sensor Design Guideline



**MICROCHIP**  
MASTERS 2019



# 谢谢!

*“对客户而言，  
界面即是产品”*

——人机界面专家兼Apple Macintosh项目发起人Jef Raskin

# 法律声明

## 软件:

Microchip软件仅允许用于Microchip产品。此外, Microchip软件的使用受软件附带的版权声明、免责声明以及任何授权许可条款的限制, 无论这些内容是在安装各个程序时阐明还是在头文件或文本文件中公告。

尽管有上述限制, 但Microchip和第三方提供的软件的某些组件仍可能被“开源”软件许可覆盖, 其中包括要求分发者提供软件源代码的许可。在开源软件许可要求的范围内, 许可条款将起主导作用。

## 注意事项和免责声明:

这些材料和随附信息(例如, 包括任何软件以及对第三方公司和第三方网站的引用)仅供参考, 并且按“现状”提供。Microchip对第三方公司做出的声明或第三方可能提供的材料或信息不承担任何责任。

MICROCHIP不承担任何形式的保证, 无论是明示的、暗示的或法定的, 包括有关无侵权性、适销性和特定用途的暗示保证。在任何情况下, 对于与MICROCHIP或其他第三方提供的材料或随附信息有关的任何直接或间接的、特殊的、惩罚性的、偶然的或间接的损失、损害或任何类型的开销, MICROCHIP不承担任何责任, 即使MICROCHIP已被告知可能发生损害或损害可以预见。请注意, 使用此处所述的知识产时可能需要第三方许可。

## 商标:

Microchip的名称和徽标组合、Microchip徽标、AdapteC、AnyRate、AVR、AVR徽标、AVR Freaks、BesTime、BitCloud、chipKIT、chipKIT徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi徽标、MOST、MOST徽标、MPLAB、OptoLyzr、PackeTime、PIC、picoPower、PICSTART、PIC32徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTrackr、TimeSource、tinyAVR、UNI/O、Vectron及XMEGA均为Microchip Technology Inc.在美国和其他国家或地区的注册商标。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、PrecisionEdge、ProASIC、ProASIC Plus、ProASIC Plus徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath和ZL均为Microchip Technology Inc.在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet徽标、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA和ZENA均为Microchip Technology Inc.在美国和其他国家或地区的商标。

SQTP为Microchip Technology Inc.在美国的服务标记。

AdapteC徽标、Frequency on Demand、Silicon Storage Technology和Symmcom为Microchip Technology Inc.在除美国外的国家或地区的注册商标。

GestIC为Microchip Technology Inc.的子公司Microchip Technology Germany II GmbH & Co. & KG在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2019, Microchip Technology Inc.版权所有。

AMBA、Arm、Arm7、Arm7TDMI、Arm9、Arm11、Artisan、big.LITTLE、Cordio、CoreLink、CoreSight、Cortex、DesignStart、DynaMIQ、Jazelle、Keil、Mali、Mbed、Mbed Enabled、NEON、POP、RealView、SecurCore、Socrates、Thumb、TrustZone、ULINK、ULINK2、ULINK-ME、ULINK-PLUS、ULINKpro、µVision和Versatile是Arm Limited(或其子公司)在美国和/或其他国家/地区的商标或注册商标。