

瑞萨电容触摸技术 - 低功耗应用培训 (基于RX140 MEC+AJ新功能)

MAY.2023

免责声明

本演示文稿中包含的信息不供客户使用。如果您希望重新调整此信息的用途，请重新调整幻灯片内容以满足目标受众的需求。如果您有任何疑问或疑虑，请联系相关的市场部门。

培训内容

□ Presentation session (90min)

- 一.瑞萨电容触摸技术 (15min)
- 二.瑞萨自容式触摸按键检测原理 (20min)
- 三.瑞萨电容触摸应用开发流程 (35min)
- 四.瑞萨电容触摸低功耗应用软件开发 (10min)
- 五.低功耗的测试设备以及测试方法 (10min)

□ Break (10min)

□ Lab session (90min)

- 一.实验前的准备
- 二. Lab Session 1: 基于RX140创建一个基本的含有12个自容按键的触摸应用工程
- 三. Lab Session 2 : 在Lab 1的基础上增加MEC功能
- 四. Lab Session 3 : 在Lab 2的基础上通过改变MEC电极的灵敏度增加接近传感功能
- 五. Lab Session 4 : 在Lab 3的基础上增加低功耗(Auto Judgement)功能
- 六. Lab Session 5: 在Lab 4的基础上使用DMM7510测试低功耗数据

瑞萨电容触摸技术

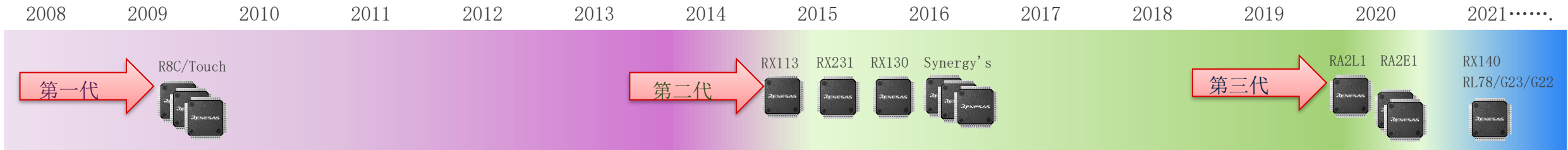


瑞萨电容触摸技术

- 瑞萨电容式触控技术
- 瑞萨电容式触控MCU产品及开发支持
- 附录：电容式触控解决方案
- 总结



瑞萨电容触控技术发展路线



第一代电容式触控IP(2008年开始, R8C-33T...)

- 基于合作伙伴的技术
- 仅支持自电容的检测方式, 基于“Workbench5”的开发支持
- 灵敏度和抗噪声较差

第二代电容式触控IP-CTSU(2014年开始,RX和Synergy产品...)

- 瑞萨自主开发的技术
- 抗噪性提升
- 支持自电容和互电容的检测方式, 从“Workbench6”到“QE for Capacitive Touch”的开发工具支持, 提供丰富的例程

第三代电容式触控IP-CTSU2(2019年开始, RA2系列,RL78和RX...)

- 抗噪性大幅提升
- 三频率采样减少同步噪声干扰
- EMC测试可通过IEC61000-4-6 level 3 (传导), IEC61000-4-3 (辐射) level 4
- 基准精度的提升
- 低功耗和并行扫描支持

技术特点

主要特点



- 支持按键，滑轮，滑条和接近感应
- 硬件自动判断实现触控超低功耗检测
- 支持电容自感应方式和电容互感应方式
- 支持硬件辅助的检测/扫描，减少CPU的负荷

高灵敏度



- 可用于厚的亚克力面板材料，木材和戴着常用的手套也能触控按键
- 在300mm的布线长度下可检测到触控电极的电容值

高抗噪声干扰能力

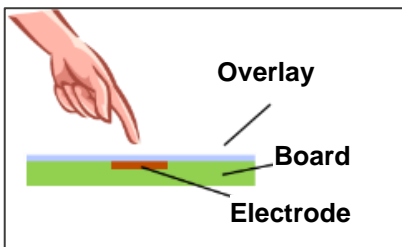


- 增强的防水能力（互感方式）
- 在硬件中实施噪声抑制措施可节省大量CPU负荷
- 抗干扰度通过了以下测试：

CTSU: IEC61000 4-3/4-6 Level 3, CTSU2: IEC61000 4-3 level4 /4-6 Level 3

高灵敏度

■ 一般的触控方案



■ 最新一代电容触控解决方案

① 可用于厚的面板材质



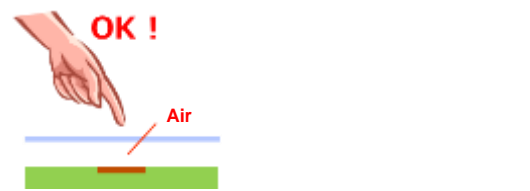
高灵敏度支持厚的面板设计!

② 自由选材



灵活的设计!

③ 节约成本



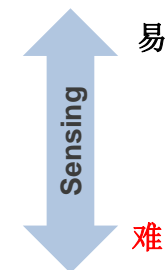
无需弹簧!

④ 非接触的应用



参考数据

覆盖物材质	介电常数
水	80
玻璃	5.4 - 9.9
亚克力	2.7 - 4.5
木材 (干燥)	2.0 - 6.0
空气	1.0



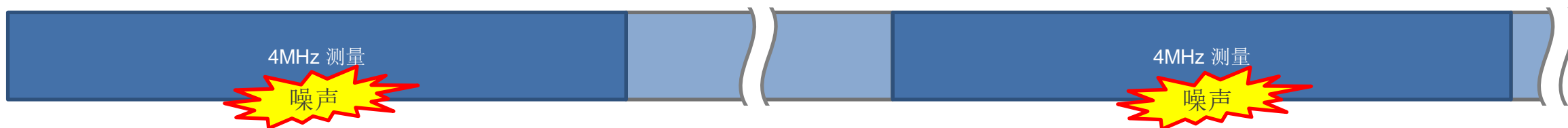
CTSU2高抗噪特点

Certified as IEC 61000 4-3 level3/4-6 level4

■ 三频率检测

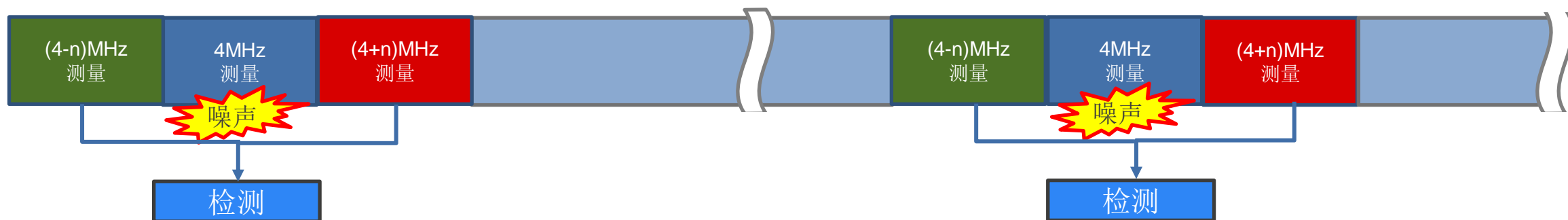
第二代(CTSU)检测方案

固定一个驱动频率测量，容易被同步噪声影响



第三代(CTSU2x)检测方案

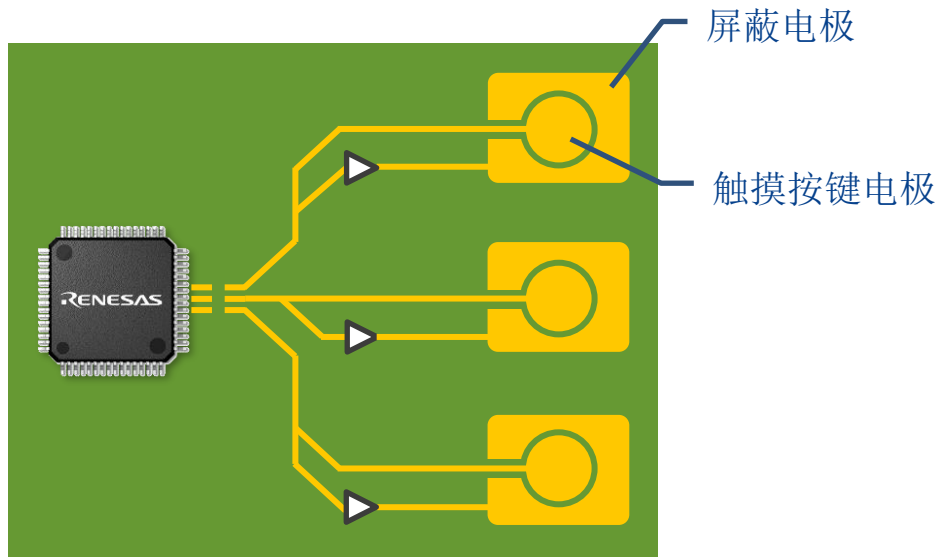
三频率测量，基于多个测量结果避免同步噪声影响



CTSU2加强了抗噪声的能力(1)

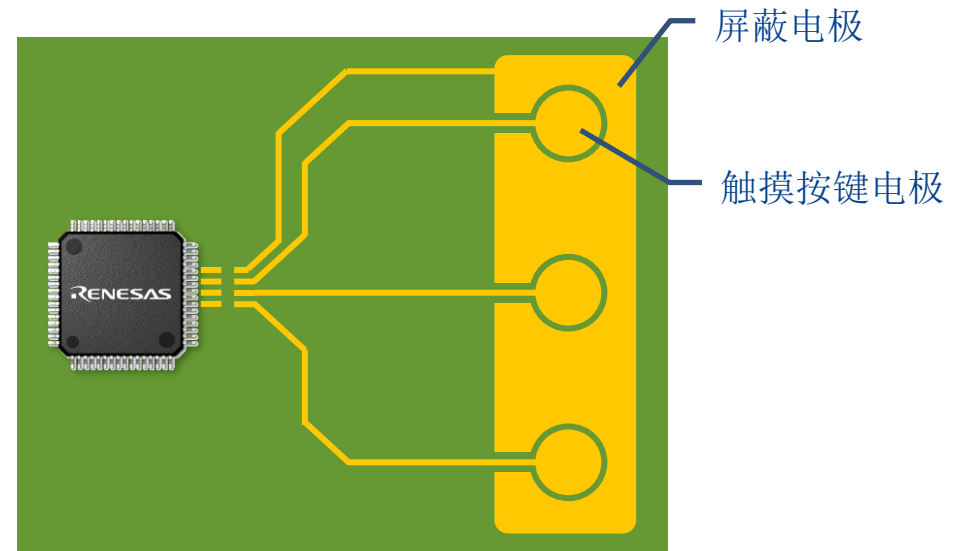
第二代触摸技术(CTSU)

每个电极Pad 需要1个屏蔽电极
每个屏蔽电极需要单独的驱动



第三代触摸技术(CTSU2)

多个电极共用一个屏蔽电极
屏蔽电极仅需一个驱动



CTSU2加强了抗噪声的能力(2)

屏蔽电极提供与目标电极脉冲具有相同相位和振幅的脉冲，目标电极旁边的屏蔽电极减少了噪声影响和滴水造成的故障。



■ 屏蔽电极的设计支持自电容的方法

无屏蔽电极的布局

如果有水滴，会产生电容的变化，相当于噪声影响目标电极。



接地的布局

接地的布局可以减少来自侧面的噪声影响，但是，如果有水滴在目标电极和接地的屏蔽电极之间，接地的布局会产生更大的电容变化，影响目标电极的灵敏度。



目标电极波形

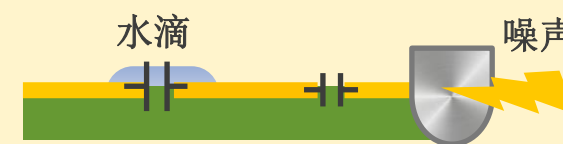


地屏蔽信号



屏蔽电极的布局

屏蔽电极可以减少来自侧面的噪声影响，如果有水滴在目标电极，屏蔽电极可以同步目标电极的变化，从而让噪声产生小的变化，达到好的抗噪效果。



目标电极波形



屏蔽电极波形



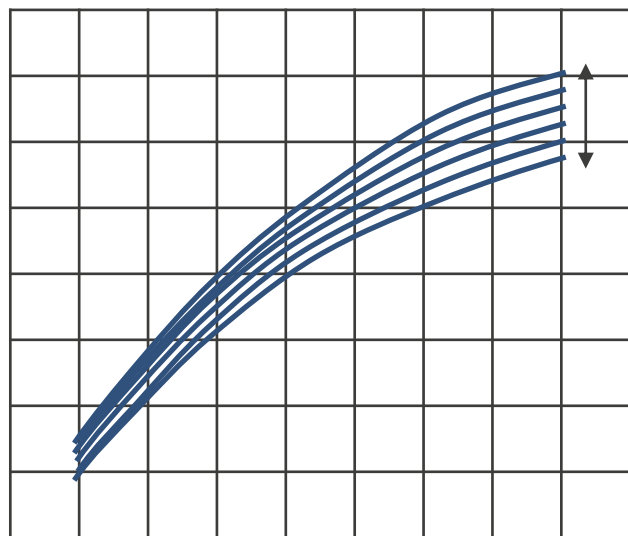
CTSU2提升了SENSOR的精度(环境温漂)

通过ICO电路提高Sensor精度，适用于需要高精度要求的应用场景

现有的触摸IP(CTSU)

电流计数器温漂值: $\pm 10\%$

Counter value



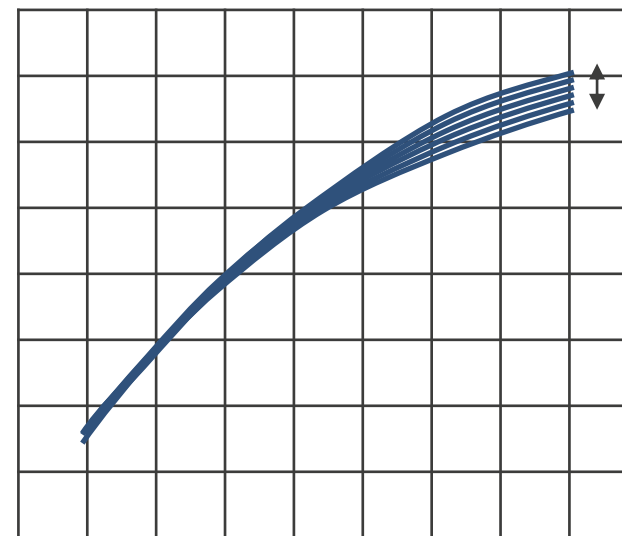
Current (capacitance)

$\pm 10\%$ variation

新的触摸IP(CTSU2)

电流计数器温漂值: $\pm 1\%$

Counter value



Current (capacitance)

$\pm 1\%$ variation Note1

CTSUS2支持高速扫描

基于CFC(Capacitance Frequency Conversion)功能支持高速并行扫描，可达20通道同步扫描

现有扫描方式

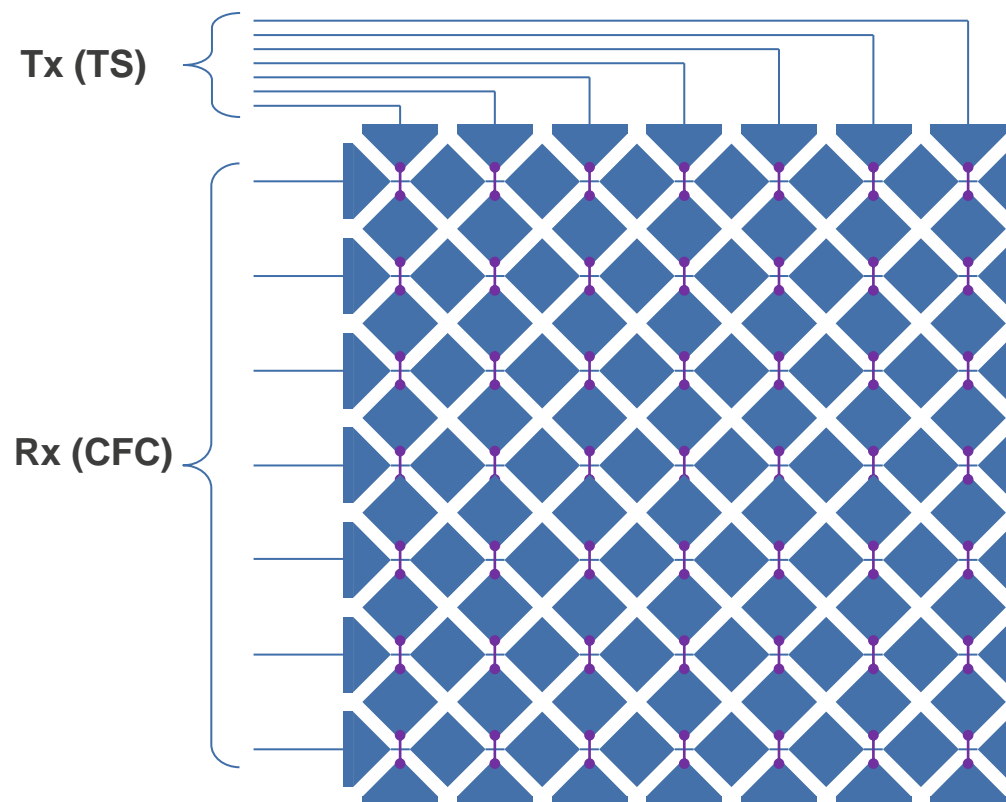
- 基于矩阵排列顺序扫描
- 7x7 矩阵需要扫描49次

CTSUS



- CFC 功能支持并行扫描
(最大20个通道同步扫描)
- 7x7 矩阵仅需要扫描7次

CTSUS2



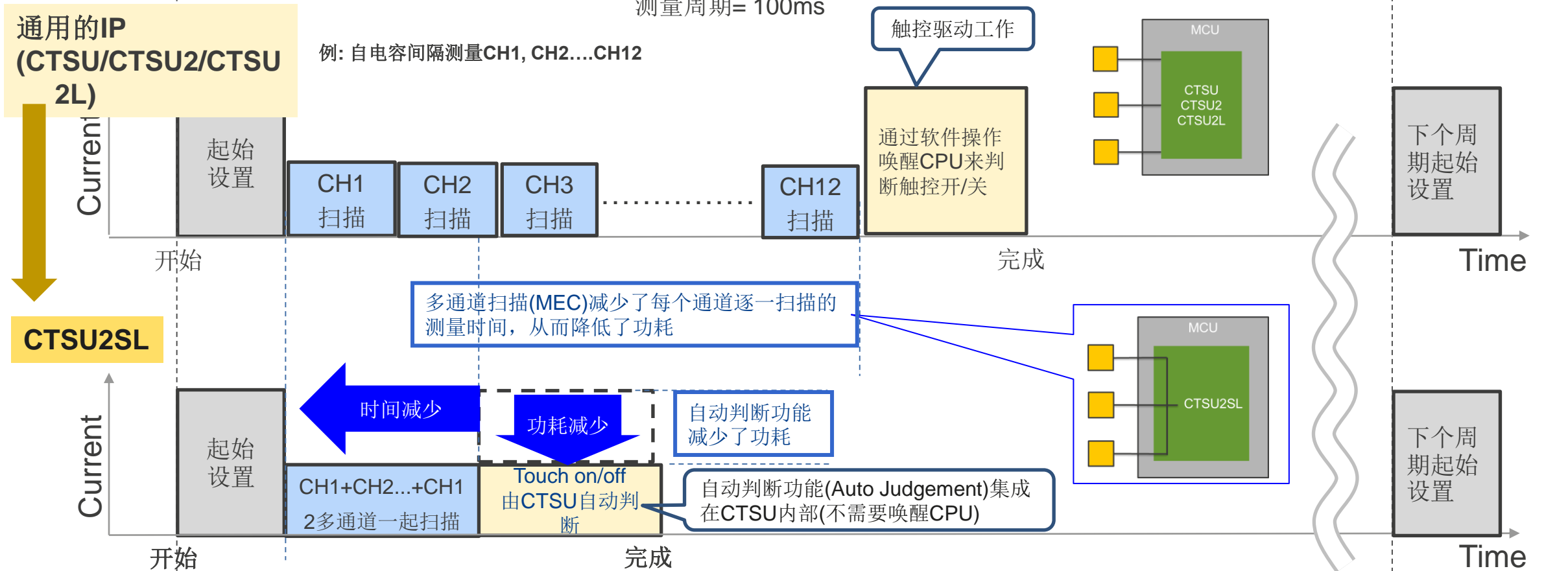
CTSUSL实现低功耗触控(AUTO JUDGMENT + MEC)

RX140

RL78/G22

自动判断功能: CTSU2SL判断触控的ON/OFF (不需要唤醒CPU)

多电极连接 (MEC->Multi Electrode Connection): 多个电极连在一起作为一个电极测量



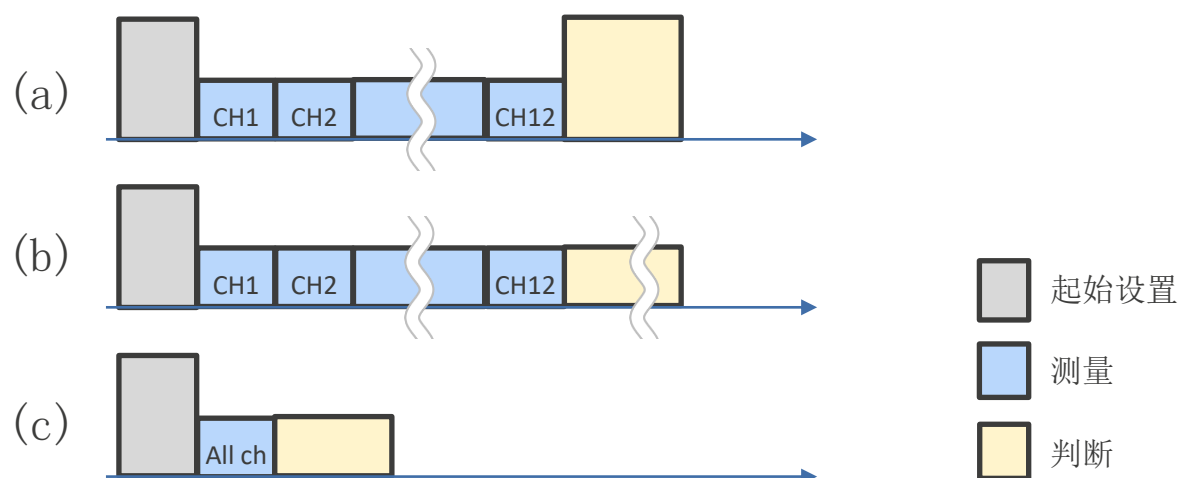
CTSU2SL低功耗测量结果

在同样的测试条件下，基于RX140的自动判断功能和多电极连接，功耗相比较RX130可减少80%。

	RX130	RX140	
	(a)	(b)	(c)
特点	-	自动判断功能	自动判断+多电极连接
功耗(μA)	120	96.3	14.3

测量条件:

- VCC: 3.3V,
- 时钟: 6MHz
- 触摸通道数: 12个
- 评估版: RX130 Touch RSSK, RX140 Touch RSSK
- 间隔扫描时间: 100ms



电容触控MCU产品群

MCU系列



Up to 240MHz			Up to 200MHz Advanced performance, connectivity, Security and scalable 2.7V – 3.6V operation	RA6		
Up to 120MHz		Up to 120MHz Outstanding real-time performance, up to 2MB Flash, connectivity, Security and scalable 2.7V – 3.6V operation	RX600	Up to 100MHz Excellent power High performance mix paired with Security	RA4	
Up to 60MHz	32MHz Low Power Intelligent Snooze Mode Security Functions	RL78	Up to 54MHz 150uA/MHz, up to 1MB Flash, Motor control / 1.62V – 5.5V operation	RX200	48MHz Low Power Higher Integration High accuracy internal oscillator	RA2
			32MHz. 100uA/MHz, Low pin count, Fast wake-up Touch, LCD, USB	RX100		

CTSU2: RL78G23,G22

CTSU: RX130/RX23x/RX671
CTSU2: RX140

CTSU: RA4Mx/RA6Mx
CTSU2: RA2L1,RA2E1

常用电容触控MCU选型指南 (闪存/封装)

RA2E1
RX140
RX130
RL78/G23
RL78/G22
RA2L1
(数字表示触摸通道数)

Flash \ Package	20	24	25	30	32	36	40	44	48	52	64	80	100	128
768 KB									16	20	22	30	32	32
512 KB									24 16	20	32 22	36 30	36 32	32
384 KB									24 16	20	32 22	36 30	36 32	32
256 KB				6	7	11	13	14	24 16	20	32 22	36 30	36 32	32
192 KB				6	7	11	13	14	16	20	22	30	32	
128 KB			9	2	3 11	5 14	6	6	8 24 20	10	12 32 30	32 36 30	32 36	32
96 KB				2	3	5	6	6	8	10	12			
64 KB	9	11	12 9	16	12 17 11	21 14	23	25	12 24 29 20		12 32 30	36	36	
32 KB	9	11	12 9	16	17 11	21 14	23	25	29 20					
Package	LSSOP (4.4x6.5)	HWQFN (4x4)	WFLGA (3x3)	LSSOP (300 mil)	LQFP (7x7) HWQFN (5x5)	WFLGA (4x4)	HWQFN (6x6)	LQFP (10x10)	LFQFP (7x7) HWQFN (7x7)	LQFP (10x10)	LFQFP (10x10) LQFP (12x12) WFLGA (5x5)	LFQFP (12x12) LQFP (14x14)	LFQFP (14x14) LQFP (14x20)	LFQFP (14x20)

常用触控MCU特性比较

CTSU1: RX130

CTSU2: RL78G23,G22,RX140,RA2E1,RA2L1

红色字体标记为差异的特性

	16bit RL78 core		32bit RX core		32bit ARM core	
	RL78/G23	RL78/G22	RX130	RX140	RA2E1	RA2L1
Pin Count	30/32/36/40/44/48/52/64/80/100/128	16/20/24/25/30/32/40/44/48	48/64/80/100	32/48/64/80	25/32/36/48/64	48/64/80/100
ROM(kByte)	96~768	32~64	64/128/256/384/512	64/128/256	32/64/96/128	128/256
RAM (kByte)	12~48	4	10/16/32/48/48	16/32/64	16	32
Data Flash(kByte)	8	2	8	8	4	8
Op. Voltage	1.6~5.5V	1.8~5.5V	1.8~5.5V	1.8~5.5V	1.6~5.5V	1.6~5.5V
Max Op. Freq.	32MHz	32MHz	32MHz	48MHz	48MHz	48MHz
Max Touch ch	2~32	5/9/11/12/16/17/21/23/25/19	24/32/36/36	12/24/32/36	10/11/14/20/30	20/30/32
DTC	Avl.	Avl.	Avl.	Avl.	Avl.	Avl.
LCD (Seg x Com)	N/A	N/A	N/A	N/A	N/A	N/A
High Current I/O	Avl.	N/A	N/A	N/A	N/A	N/A
USB	N/A	N/A	N/A	N/A	N/A	N/A
RTC	Avl.	Avl.	Avl. (※48pin: N/A)	Avl.	Avl. (※25pin: N/A)	Avl.
Comparator	2ch	2ch	2ch	2ch	2ch	2ch
A/D	12bit,8~26 ch	10bit,3~10ch	12bit, 10/14/17 ch	12bit,8/11/15/18 ch	12bit,8/10/12/9/13 ch	12bit,13/17/19 ch
D/A	8bit,1~2ch	N/A	8bit, 0/2/2 ch	8bit,0/0/2/2 ch	N/A	12bit,1ch
Security Function (TSIP/AES/RNG (Note*))	Avl.	Avl.	N/A	Avl.	Avl	Avl.
CAN	N/A	N/A	N/A	Avl.	N/A	Avl.

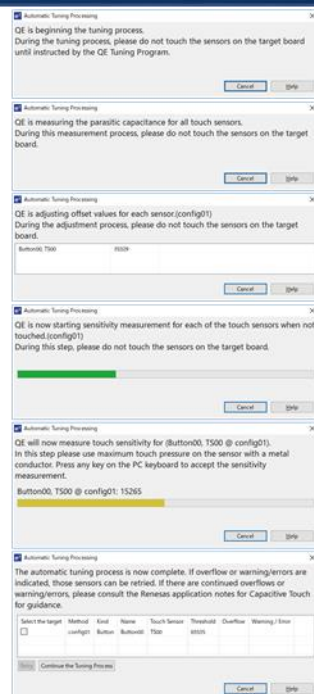
(Note*) 部分型号无此功能, 部分型号仅有部分功能, 具体参考硬件手册

易用的开发工具QE

- 开发工具对初学者简单易用
基于以下的指南轻松实现触控传感器调谐和窗口监控
- 自动调谐触控传感器灵敏度
按照以下指南调整偏移和灵敏度

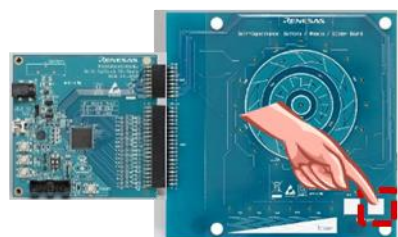


Tuning window



- Preparing for adjustment
- Measuring parasitic capacitance
- Adjusting the offset
- Measuring sensitivity (while not touched)
- Measuring sensitivity (while touched)
- Result of the tuning

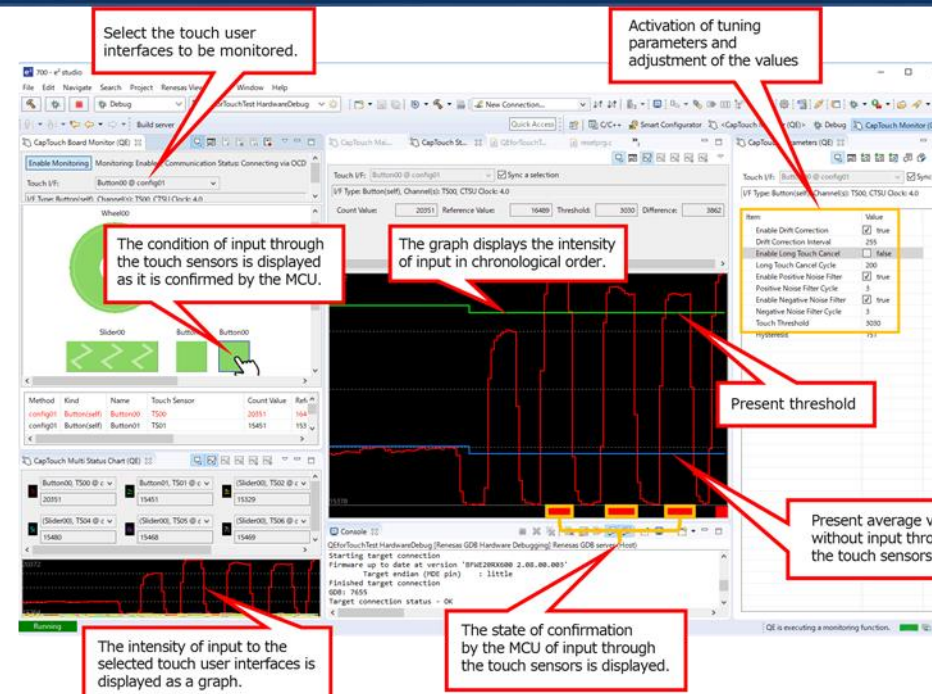
Just touch the electrode !



Touch

**Easy to develop even for beginners !
Just touch the electrode and
everything will be automatically tune by the tool.
Drastically reduce development time and efforts !**

Monitoring window



应用笔记和例程-1

类别	标题	下载链接
设计指南	Capacitive Sensor Microcontrollers CTSU Capacitive Touch Electrode Design Guide Rev.2.00	https://www.renesas.com/us/en/document/apn/capacitive-sensor-microcontrollers-ctsu-capacitive-touch-electrode-design-guide?language=en&r=1544941
驱动/中间件	RX Family QE CTSU Module Using Firmware Integration Technology Rev.2.10	https://www.renesas.com/us/en/document/apn/rx-family-qe-touch-module-using-firmware-integration-technology?language=en&r=1544941
	RL78 Family TOUCH Module Software Integration System Rev.1.20	https://www.renesas.com/us/en/document/apn/rl78-family-ctsu-module-software-integration-system?language=en&r=1496636
	Renesas Flexible Software Package (FSP) v4.0.0	https://www.renesas.com/jp/ja/document/mas/renesas-flexible-software-package-fsp-v400-users-manual
	Synergy Software Package (SSP) v2.2.0	
工具使用指南	RX Family Using QE and FIT to Develop Capacitive Touch Applications Rev.2.00	https://www.renesas.com/us/en/document/apn/rx-family-using-qe-and-fit-develop-capacitive-touch-applications-rev100?language=en&r=1170071
	RL78 Family Using QE and SIS to Develop Capacitive Touch Applications Rev.2.10	https://www.renesas.com/us/en/document/apn/rl78-family-using-qe-and-sis-develop-capacitive-touch-applications?language=en&r=1496636
	RA Family Using QE and FSP to Develop Capacitive Touch Application Rev.2.00	https://www.renesas.com/us/en/document/apn/ra-family-using-qe-and-fsp-develop-capacitive-touch-application?language=en&r=1398061
	RX/RA/RL78 Family Renesas Synergy™ Platform QE for Capacitive Touch 3D Gesture Recognition Application Development Guide Rev1.00	https://www.renesas.com/us/en/document/apn/qe-capacitive-touch-3d-gesture-recognition-application-development-guide?language=en&r=1170071
	RL78 Family Using the standalone version of QE to Develop Capacitive Touch Applications	https://www.renesas.com/us/en/document/apn/rl78-family-using-standalone-version-qe-develop-capacitive-touch-applications-rev100?r=1521766

应用笔记和例程-2

类别	标题	下载链接
低功耗例程	RA2L1 Group Capacitive Touch Low Power Guide Rev.1.00	https://www.renesas.com/us/en/document/apn/ra2l1-group-capacitive-touch-low-power-guide?language=en&r=1398061
	RL78 Family RL78/G23 Capacitive Touch Low Power Guide (SNOOZE function) Rev.1.00	https://www.renesas.com/jp/en/document/apn/rl78-family-rl78g23-capacitive-touch-low-power-guide-snooze-function-rev100?language=en&r=1496636
	RA6M2 Group Capacitive Touch Low Power Guide Rev.1.00	https://www.renesas.com/us/en/document/apn/ra6m2-group-capacitive-touch-low-power-guide?r=1403491
	RL78/G23 Capacitive Touch Low Power Guide (SMS function) Rev.1.00	https://www.renesas.com/us/en/document/apn/rl78g23-group-rl78g23-capacitive-touch-low-power-guide-sms-function-rev100?r=1521766
其他	RX113 Group CTSU Basis of Cap touch detection Rev.1.00	https://www.renesas.com/us/en/document/apn/ctsu-basis-cap-touch-detection-rev100?language=en&r=1544941
	RX113 Group CTSU Cap touch measurement by Self capacitance method Rev.1.00	https://www.renesas.com/us/en/document/apn/ctsu-cap-touch-measurement-self-capacitance-method-rev100?language=en&r=1544941
	RX113 Group CTSU Cap touch measurement by Mutual capacitance method Rev.1.00	https://www.renesas.com/us/en/document/apn/ctsu-cap-touch-measurement-mutual-capacitance-method-rev100?language=en&r=1544941
	RX Family Capacitive Touch Sensor Correction for Accuracy Enhancement Rev.1.00	https://www.renesas.com/us/en/document/apn/capacitive-touch-sensor-correction-accuracy-enhancement-rev100?language=en&r=1544941
	RX Family CTSU Noise immunity improving with Dual frequencies noise canceling process Rev.1.00	https://www.renesas.com/us/en/document/apn/ctsu-noise-immunity-improving-dual-frequencies-noise-canceling-process-rev100?language=en&r=1544941

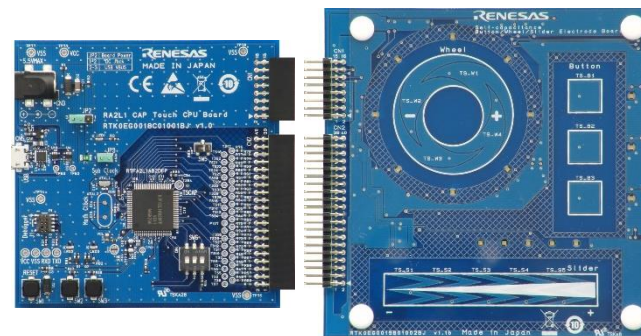
基于电容式触控技术的应用案例



电容式触控参考设计套件

- CTSU2 支持按键，滑轮，滑条设计
- 抗噪声干扰性能高
- 可参考RSSK评估套件做各种开发

RL78/G23/G22/RX130/RX140/RA2 Touch RSSK



高灵敏度

- 按键设计支持10mm厚的亚克力面板，是产品设计更加灵活
- 能够检测小的空气间隙，无需弹簧
- 可灵活支持各种面板介质材料

高抗噪特点

- 多频率扫描降低同步噪声干扰
- 自感/互感均可实现防水设计

易于开发

- “QE for Capacitive Touch” 支持源码生成和灵敏度tuning
- 可直接从官网下载各种方案资料
- 多种应用笔记可供参考



瑞萨所有
触控MCU

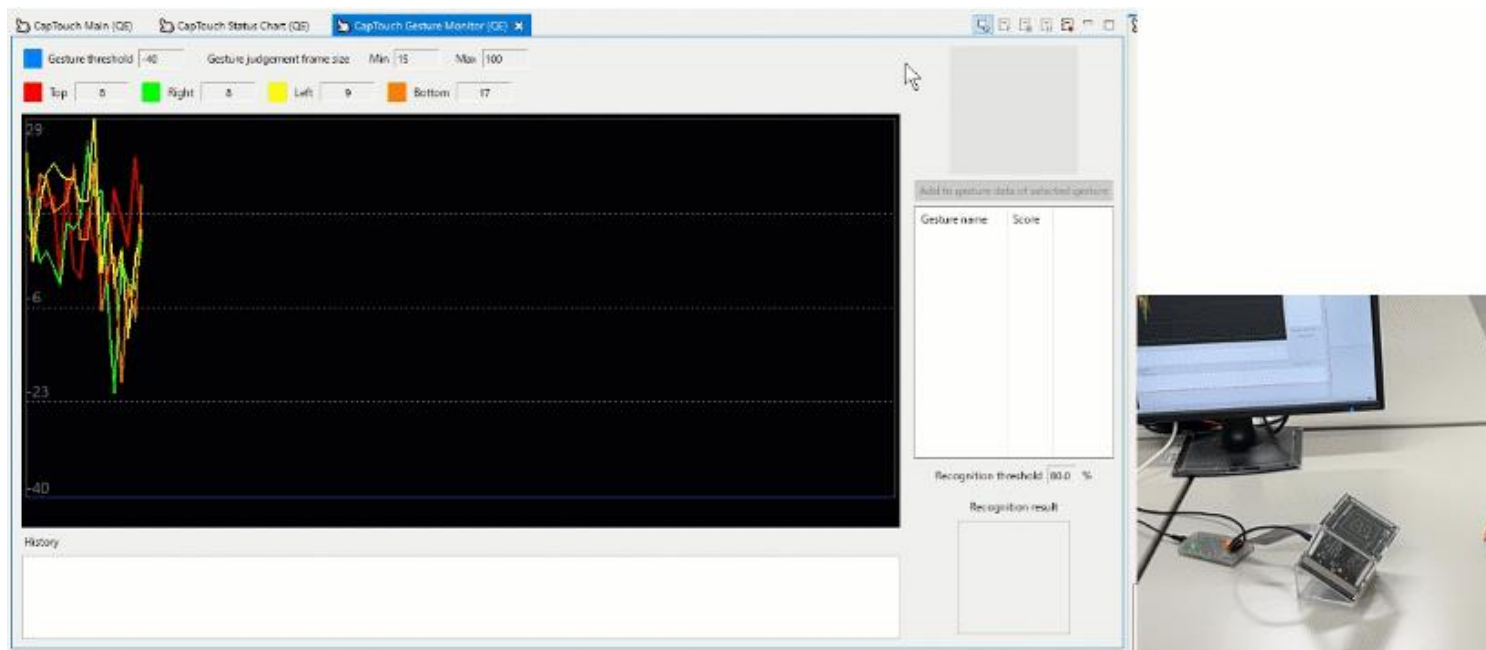
e-AI x 3D手势识别方案

3D手势标准识别方案适合较长距离的应用

电极板尺寸**160x160mm**，电极板和MCU板为分开设计，方案外形尺寸**(160x160x100mm)**和检测最大距离约**200mm** Note1

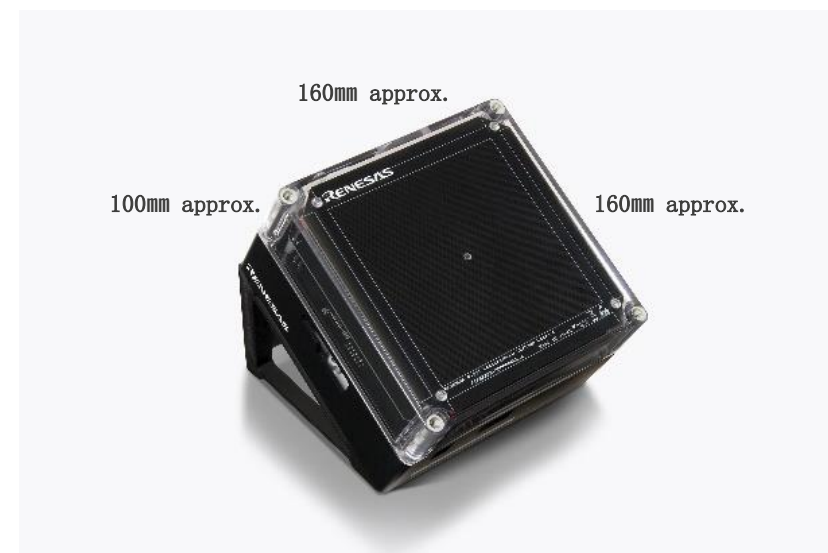
用户可通过**QE For Touch**工具窗口来直观的查看手势的波形和识别率

3D手势识别方案适用于多种智能家居应用，如智能家居面板控制和智能马桶翻盖控制等。

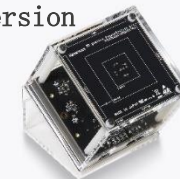


Note1: 测试结果基于瑞萨的评估版，在噪声环境或设计改变的条件下，不能确保

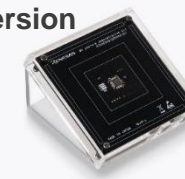
3D手势识别方案



3D Gesture Small version



3D Gesture Slim version



瑞萨所有触控MCU可用于3D手势的方案设计
(RL78/G23/G22,RX113,RX130,RX231,RX230,RA2).

2D手势识别方案

2D手势方案基于几个电极接近感应的方式来识别手靠近电极，移开电极，以及手从一个电极移动到另一个电极

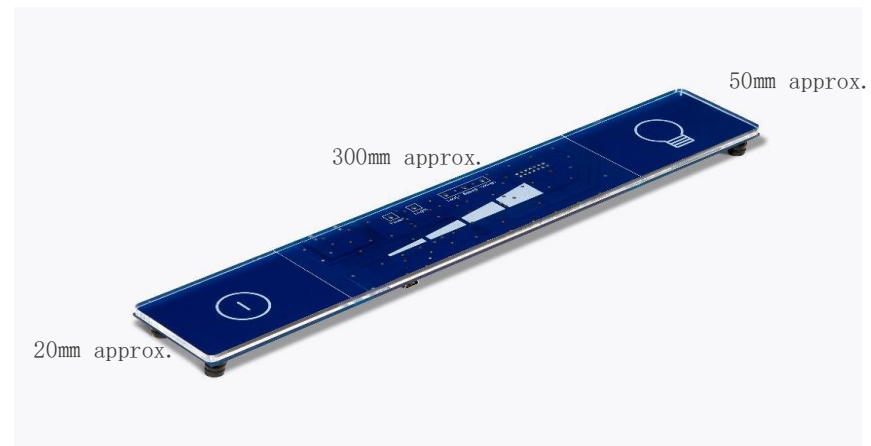
方案以长方体结构设计,尺寸为(300x50x20mm)，最大检测距离大约100mm Note1

该方案基于自电容检测方式简化为两个电极设计，双面板设计仅需要1个电容和两个电阻。

- BOM 成本低
- 手势识别更加简单
- 相对3D手势方案，结构设计轻薄

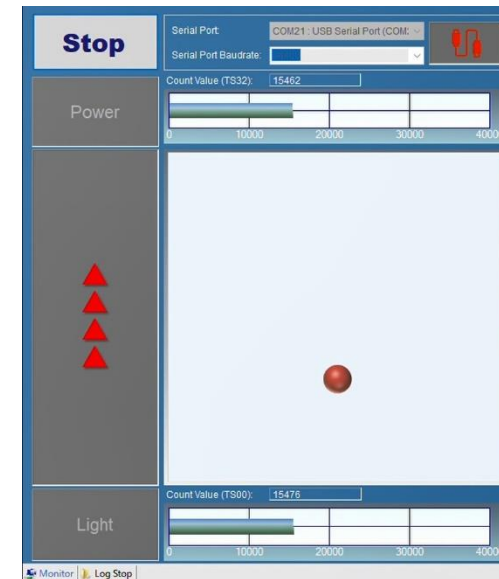
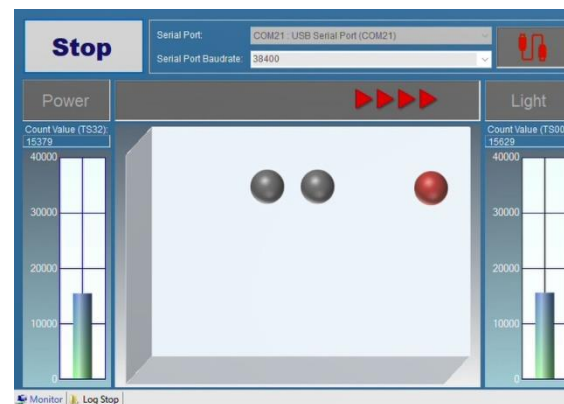
2D手势识别方案适用于多种应用，如抽油烟机风量控制，智能家居面板控制(智能窗帘开/关)等。

Note1: 测试结果基于瑞萨的评估版，在噪声环境或设计改变的条件下，不能确保



Note: 瑞萨所有触控MCU可用于2D手势的方案设计 (RL78/G23/G22,RX113,RX130,RX231,RX230,RA2).

用户可通过电脑上位机软件查看二维(X,Y)手势的位置



非接触按键解决方案

目标应用

- 家用电器：冰箱，智能马桶，烟机，等.
- 公共设施：电梯，自动售卖机，等.

方案特点

- 基于电容触控设计3*3=9 按键矩阵，实现接近感应
- 手指接近按键电极时，LED灯亮/灭
- 按键响应通过蜂鸣器输出声音

方案优势

- 相较于红外传感器，成本低，没有使用环境光强影响
- 相较于超声波传感器：，成本低，不受环境屏蔽限制
- 方案基于RX， RA触控MCU评估板， Demo搭建方便快捷，成本低

方案交付

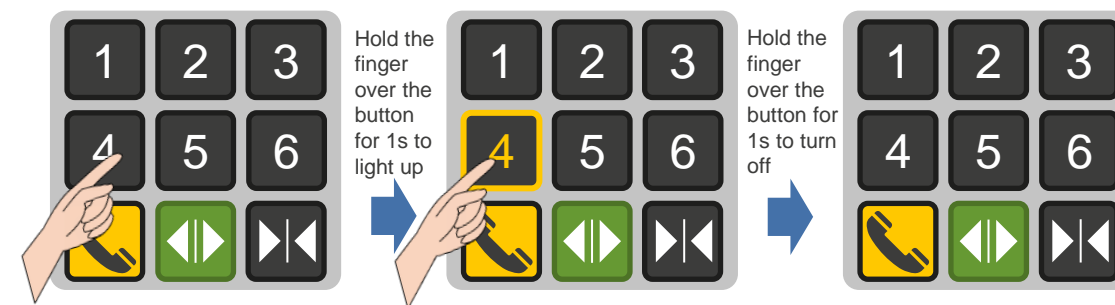
- 硬件：非接触按键硬件套件
- 软件：源代码(从瑞萨网站下载)
- 文档：应用笔记， 电路图， PCB设计文档(从瑞萨网站下载)



触控评估MCU核心板：
RX130: RTK0EG0004C01002BJ
RA2L1: RTK0EG0018C01001BJ

应用说明

例: 电梯按键



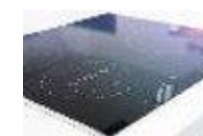
其他应用: 家电, 家用设备, 公共设施, 等.



冰箱/微波炉



智能马桶



电磁炉



自动售卖机

位置检测解决方案

集成CTSUs的MCU不仅仅用于电容触摸按键，还可以用于流体或者粉末的容积检测。

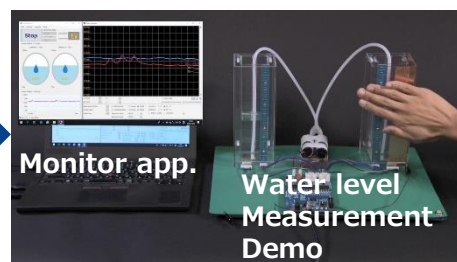
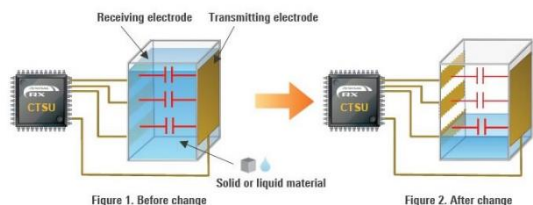
瑞萨推荐通过电容触摸的功能来实现量体的线性检测，可基于单电极+导电屏蔽的方式来实现。

粉末材料检测*

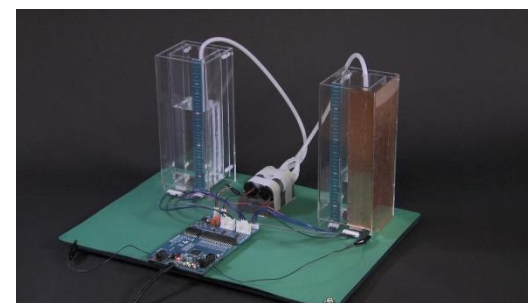
通过安装多个电极在不同位置来检测每个位置物质的存在或不存在
例：3电极实现4级检测
(100~75%， 75~50%， 50~25%， 0%)

水位检测：

基于单电极+导电屏蔽的方式来实现线性位置检测
例：单电极分100级
(100~0%)



单芯片实现人机交互控制+ 线性流体检测



Water Level Measurement
Demo



Sample program
Monitor App.



APN

轻松实现新功能添加或产品的迭代升级

*: <https://www.renesas.com/us/en/solutions/key-technology/human-interface/material-detection.html>

总结

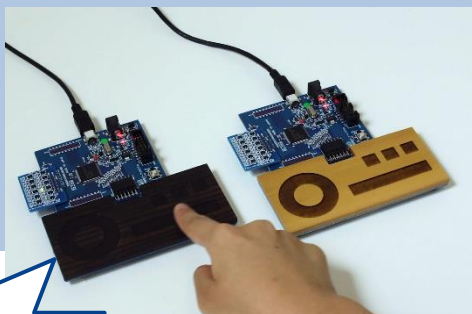
高灵敏度和抗噪声干扰能力!



增强的防水能力!



瑞萨提供多种触控MCU和解决方案，让复杂的交互设计简单化!



支持多种面板材质!



开发更加容易!

瑞萨自容式触摸按键检测原理



瑞萨自容式触摸按键检测原理

二. 瑞萨自容式触摸按键检测原理

1. 自容式触摸按键检测原理概述
2. 自容式触摸按键的电极设计通用规范
3. 自容式触摸按键的灵敏度
4. Smart lock应用中的按键电极设计举例

瑞萨自容式触摸按键检测原理概述

图 2-2 所示为电极中产生的自电容。自电容式按键中连接到电容传感器的单个电极将测量电容量 C 。 C 的值是由电极及其周围导体形成的寄生电容 C_p 和由电极及手指形成的寄生电容 C_f 的复合值。由于周围的器件是静态的，因此 C_p 是常量，但 C_f 会随着手指的靠近而增加。通过设置 C_f 增加量的阈值，可以确定触摸按键是处于“打开”还是“关闭”状态。请注意，如果手指直接接触电极，则会导致电极短路，并且无法再测量电容。通常，电极和手指之间有几毫米厚的覆盖面板。

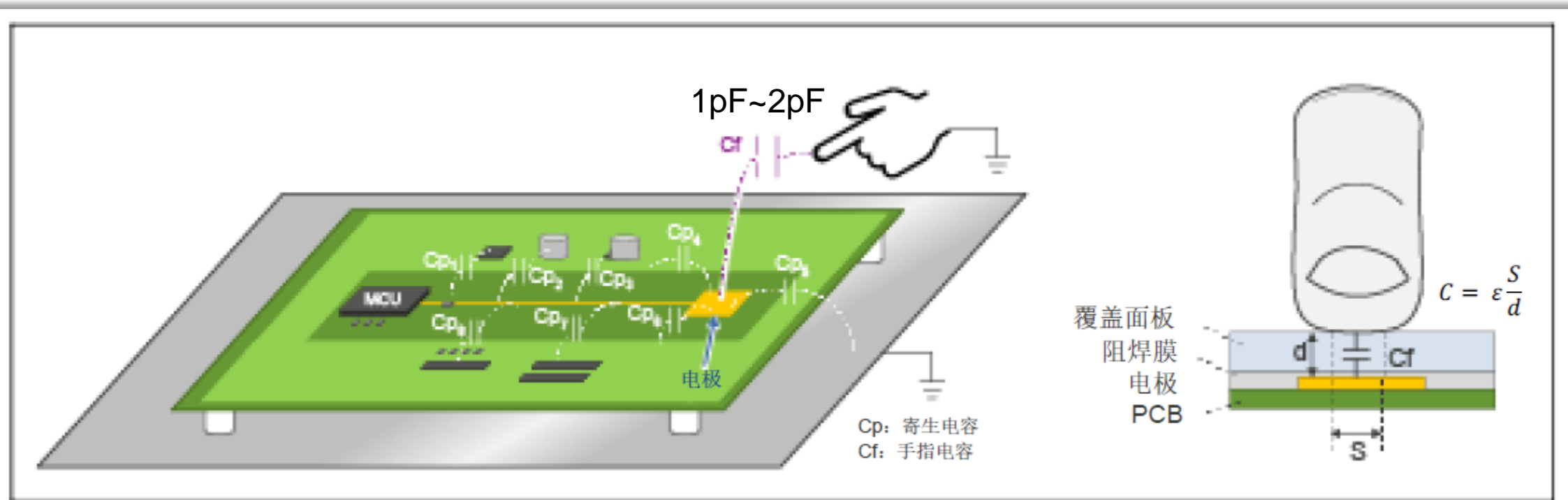


图 2-2. 电极中产生的自电容示意图

瑞萨自容式触摸按键检测原理概述

CTSUS 自电容方式检测原理

✓ 自容 CTSU 内部配置

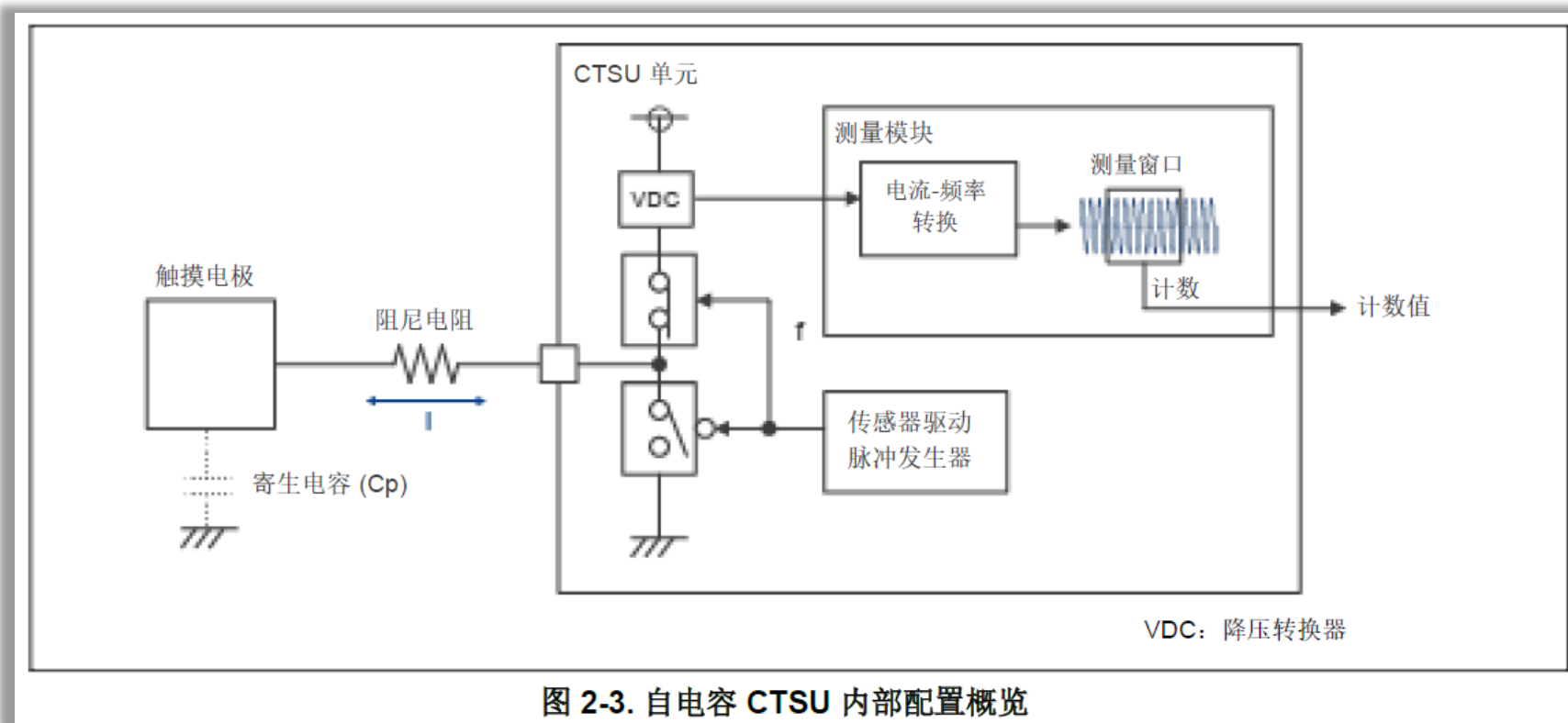
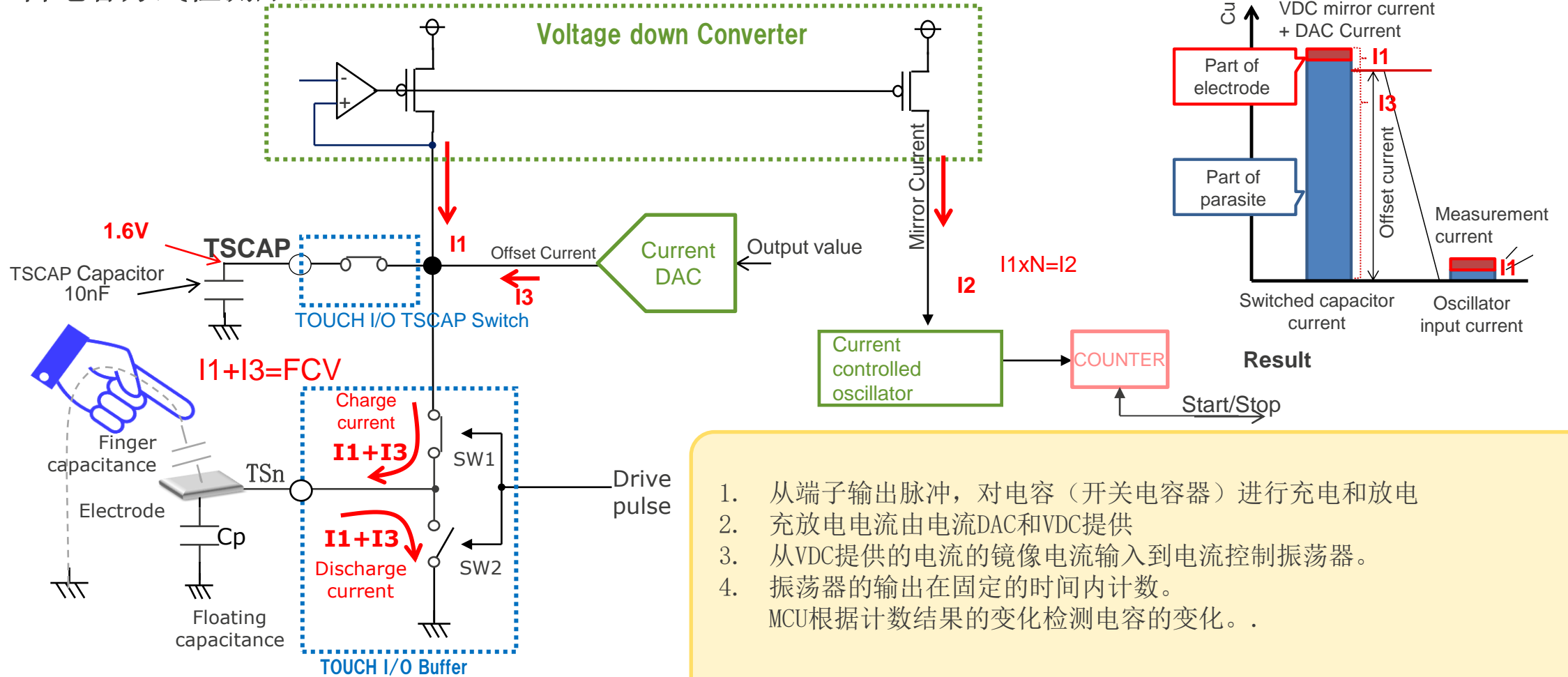


图 2-3 所示为自电容方式的 CTSU 内部配置概览。CTSUS 输出一个与所连接电极的电容量 C 成正比的数字计数值，并通过软件确定触摸按键是处于“打开”还是“关闭”状态。连接到 CTSU 时，电极会充当由传感器驱动脉冲控制的开关电容，并从充放电电流来估测 C 的电容。CTSUS 测量模块具有电流-频率转换功能，输入与充放电电流等效的电流，并输出与电流量成比例的频率。有关检测原理的详细信息，请参见应用笔记“RX113 系列 CTSU 电容式触摸检测基础”。

瑞萨自容式触摸按键检测原理概述

CTSUS 自电容方式检测原理



1. 从端子输出脉冲，对电容（开关电容器）进行充电和放电
2. 充放电电流由电流DAC和VDC提供
3. 从VDC提供的电流的镜像电流输入到电流控制振荡器。
4. 振荡器的输出在固定的时间内计数。
MCU根据计数结果的变化检测电容的变化。

瑞萨自容式触摸按键检测原理概述

CTSU 自电容方式检测原理

✓ CTSU 测量

✓ 完全充放电

✓ 最小测量频率

✓ 0.5MHz

✓ 最大测量频率

✓ 在确保完全充放电的情况下，根据寄生电容和电阻值计算得出（最大值是4MHz）

✓ 寄生电容范围：3.3pF 到 50pF（在此范围内越小灵敏度越高，10pF左右为推荐值）

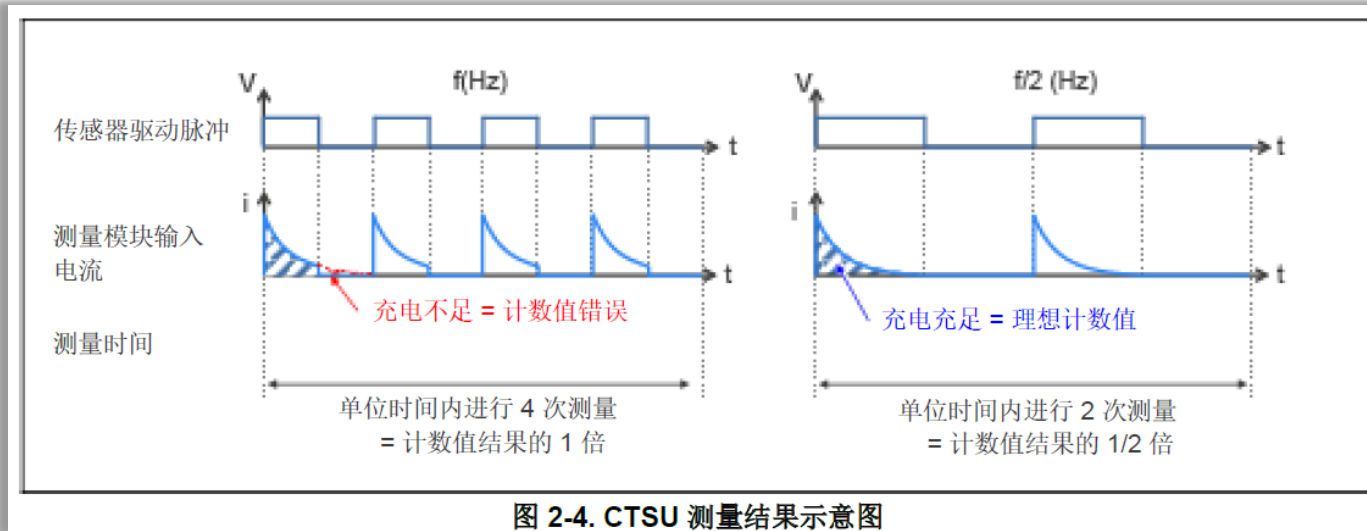


图 2-4. CTSU 测量结果示意图

图 2-4 所示为 CTSU 测量结果的示意图。当传感器驱动脉冲频率的一个周期比 C 充放电时间短且充放电不足时，则没有足够的电流流向 C，因此计数值小于理想值。当寄生电容很大时，可以通过降低传感器驱动脉冲频率来进行测量。当传感器驱动脉冲频率降低时，CTSU 可测量的最大值为 50pF。请注意，当传感器驱动脉冲频率降低时，电流-频率转换功能在单位时间内的测量次数也会减少，电极的灵敏度也可能会降低。可以通过调整 CTSU 中的寄存器设定值来增加单位时间，但完成测量所需的时间也会增加。在设计电容式电极电路时，必须满足按键灵敏度、测量时间和抗噪性等条件。

自容式触摸按键的电极设计通用规范

- 在设计硬件之前，我们强烈建议用户参考以下应用说明



应用笔记

电容式传感器微控制器

CTSU 电容式触摸电极设计指南

简介

本应用笔记介绍了如何为内置电容式触摸感应单元 (CTSU) 的 MCU 设计电极图案，并提供示例图案以供参考

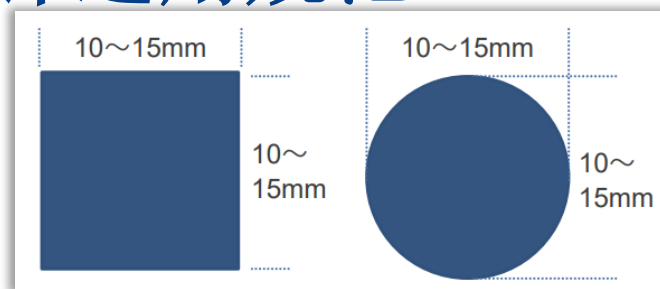
目标器件

内置 CTSU (CTSU 表示 CTSU2、CTSU2L 和 CTSU2SL 等) 的 RX 产品家族、RA 产品家族、RL78 产品家族 MCU 和 Renesas Synergy™ 平台产品

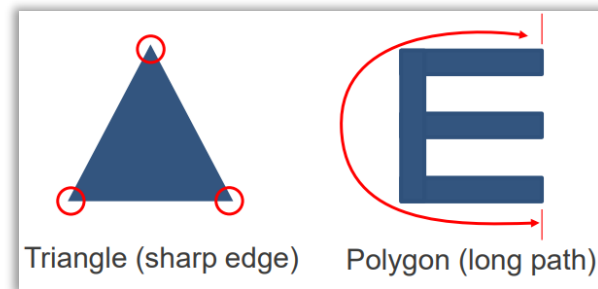
Capacitive Sensor Microcontrollers CTSU Capacitive Touch Introduction Guide

自容式触摸按键的电极设计通用规范

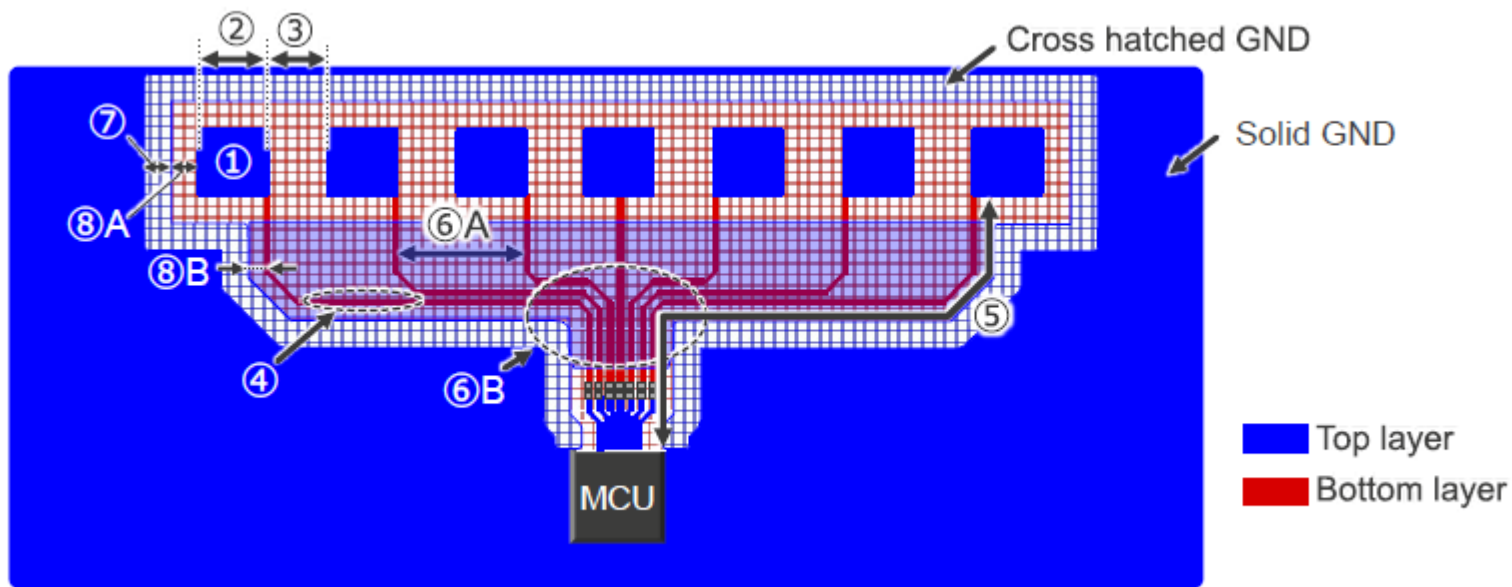
✓ 以自容式按键为例



推荐的电极形状和尺寸



不推荐的电极形状



自电容式按键的抗噪布局图案示例

- ①电极形状：方形或圆形
- ②电极尺寸：10 mm 至 15 mm
- ③电极间距：按键大小的 0.8 倍
- ④线宽：约 0.15 mm 至 0.20 mm
- ⑤导线长度：尽量缩短导线。
- ⑥导线间距：(A) (B) A: 尽量宽 B: 1.27 mm 中心距
- ⑦网状 GND 图案宽度：5 mm
- ⑧网状GND 图案和按键/导线间距 A:5mm B:3mm
- ⑨电极 + 导线电容：不超过 50 pF
- ⑩电极 + 导线电阻：不超过 2 kΩ
(含参考值为 560Ω 的阻尼电阻)

自容式触摸按键的电极设计通用规范

✓ GND 图案和寄生电容

图 2-5 所示为 GND 图案和寄生电容的示意图。使用印刷电路板时，一般的抗噪对策是在导线图案的正下方放置 GND 平面图案。在自电容按键中，由电极和 GND 平面图案产生的寄生电容 C_{pGND} 远大于 C_f ，超出了 CTSU 的测量范围。因此，在设计自电容按键时，切勿在电极正下方放置 GND 平面图案。如果需要抗噪对策，请使用交叉影线 GND 图案以减少寄生电容的增加。

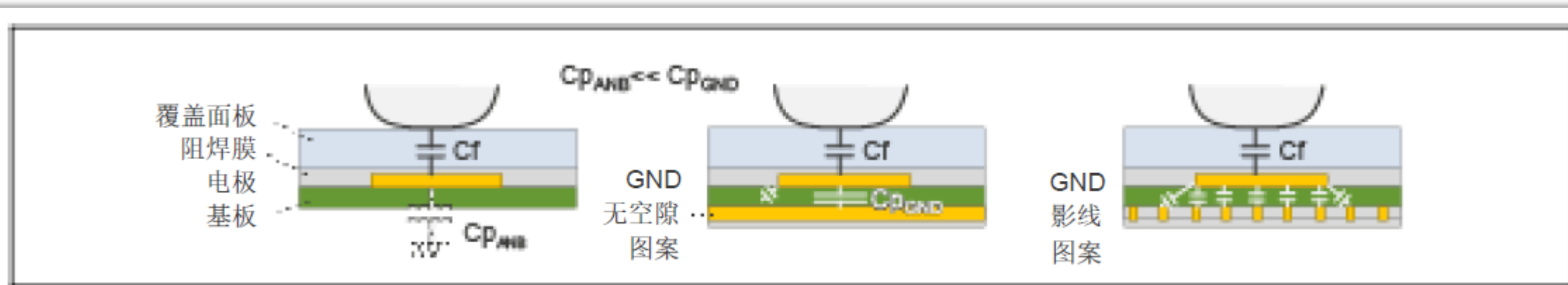
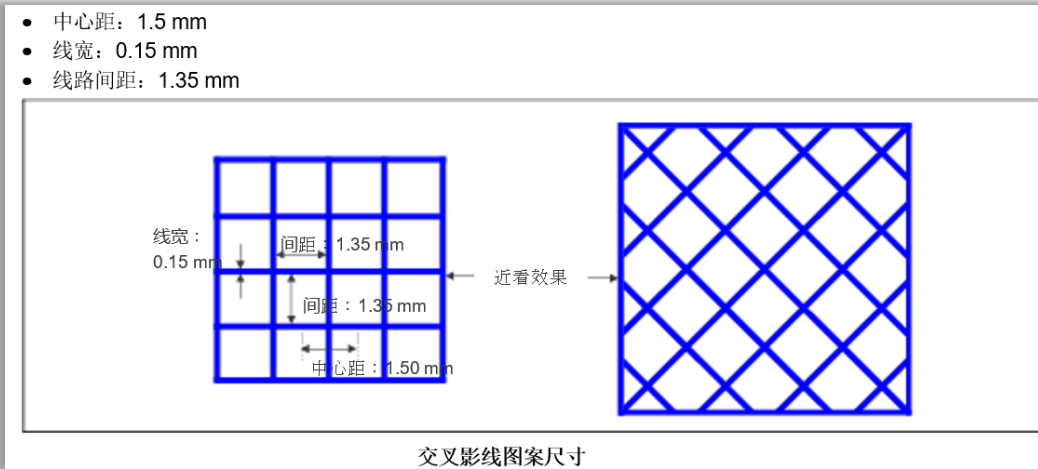
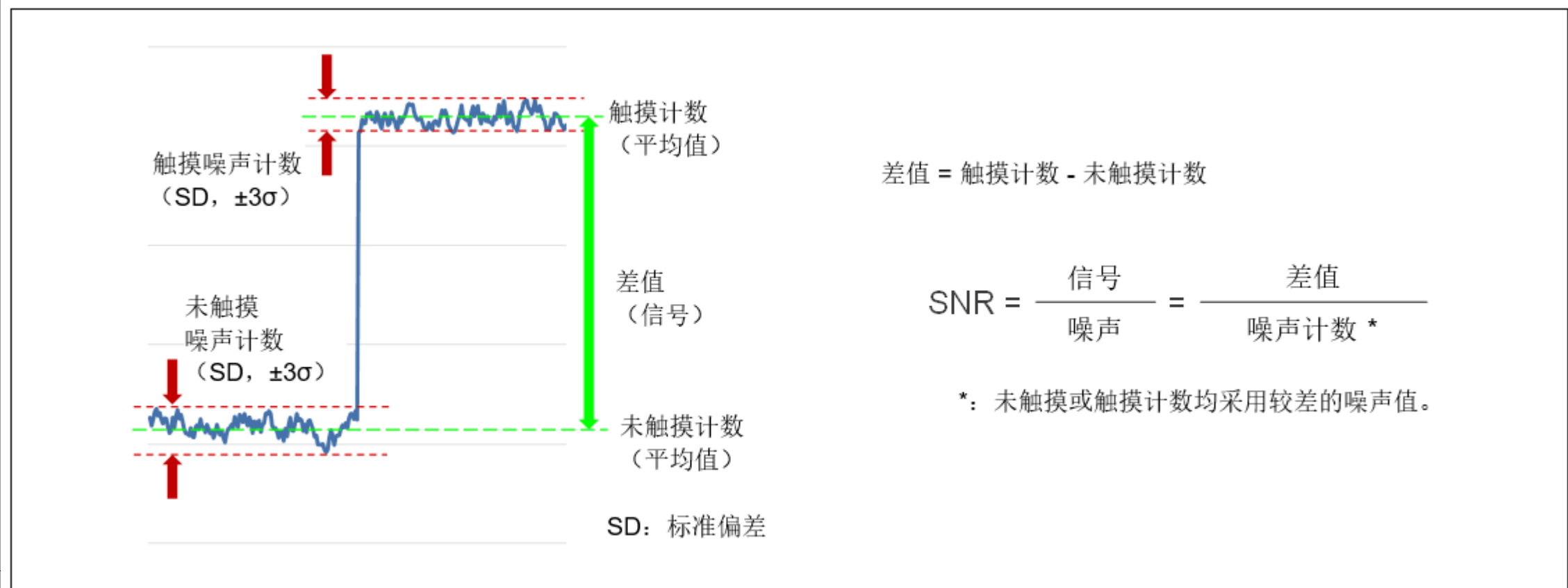


图 2-5. GND 图案和寄生电容的示意图

自容式触摸按键的灵敏度

□ 根据信噪比 (SNR) 自容式触摸按键的按钮灵敏度

下图显示了按钮灵敏度 (SNR) 推导方法。SNR 是根据触摸和未触摸按钮时的计数差值和噪声值计算得出的。测量的值为任意数量样本的平均值。噪声值为任意数量样本的标准偏差的 $\pm 3\sigma$ ；将采用触摸或未触摸测量中较差的一个值。视数据采集时间而定，噪声值可能会增加或减少，因此应使用足够的样本采集时间进行评估，以提高准确性。



自容式触摸按键的灵敏度

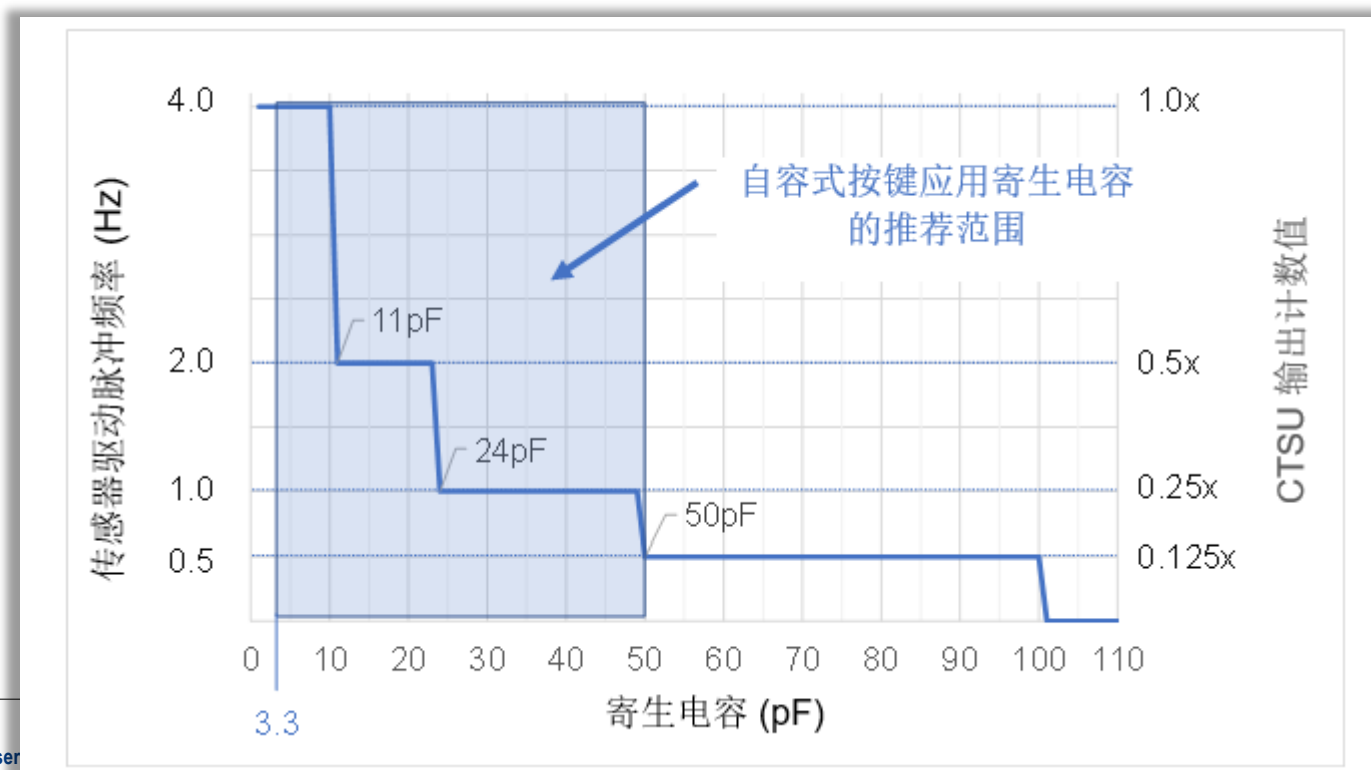
□ CTSU 的传感器驱动脉冲频率会影响电容测量范围及 SNR

下图显示了寄生电容和传感器驱动脉冲频率的设置示例。

本示例的条件如下：MCU 为 RA2L1，电压 (VCC) 为 5.0V，阻尼电阻值 $R=560\Omega$ 。

建议使用 560Ω 的阻尼电阻值，以防止引脚受到外部噪声影响，并限制引脚的输出电流。

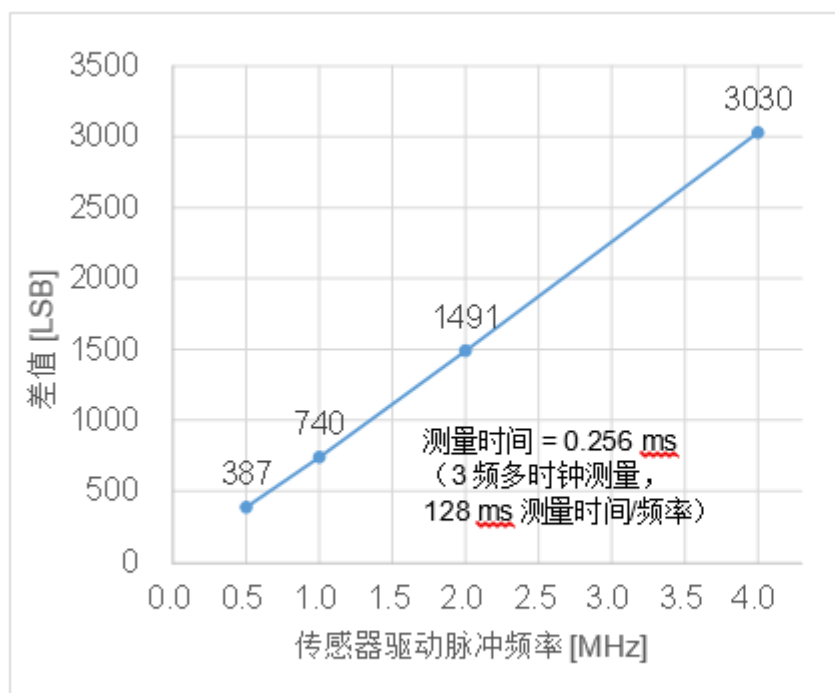
QE for Capacitive Touch 的自动调整功能 (Auto tuning process) 会选择 4.0、2.0、1.0 和 0.5 MHz 的传感器驱动器脉冲频率。该设置因所用 MCU 而异。



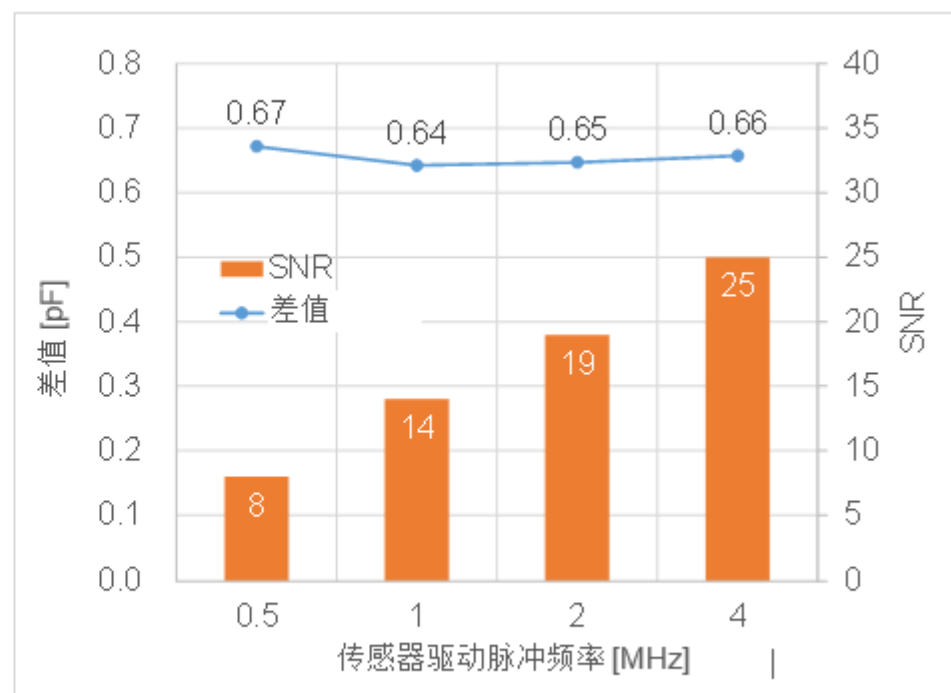
自容式触摸按键的灵敏度

□ CTSU 的传感器驱动脉冲频率与灵敏度之间的关系

- ❖ 图(a)显示每个传感器驱动脉冲频率在“触摸On”和“触摸Off”这两者之间的测量差值，该值与传感器驱动脉冲频率成正比。
- ❖ 当测量值转换为电容值时，例如，图(b)所示的触摸时检测到的电容（差值），无论传感器驱动脉冲频率如何，均可以检测到一个常量值。另一方面，SNR 与传感器驱动脉冲频率成比例地提高。传感器驱动脉冲频率越低，每次计数的噪声量越高，SNR 趋于下降。



(a) 传感器驱动脉冲频率和测量值



(b) 触摸时检测到的电容和 SNR

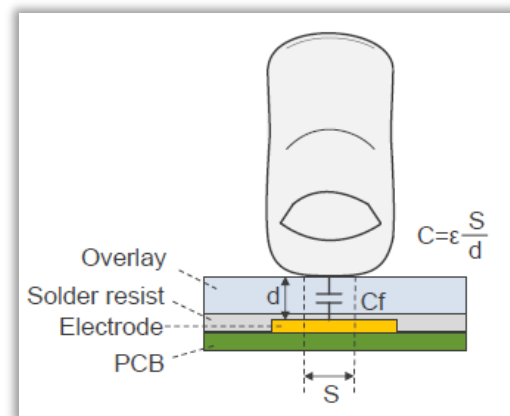
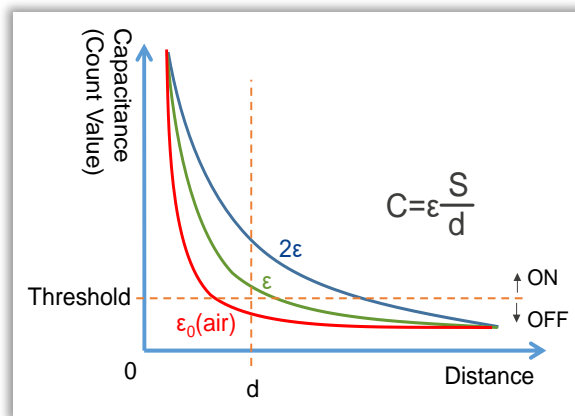
自容式触摸按键的灵敏度

❑ 电极的寄生电容值从根本上决定了灵敏度

❖ 寄生电容范围：3.3pF 到 50pF（在此范围内越小灵敏度越高，10pF左右为推荐值）

❑ 影响电极的寄生电容值的因素

- ✓ 电极尺寸
- ✓ 电极之间的距离
- ✓ 电极走线的长度
- ✓ 电极走线之间的间距
- ✓ 电极与GND之间的间距
- ✓ 电极走线与GND之间的间距
- ✓ 实心GND屏蔽与网状GND屏蔽
- ✓ 网状GND的网线间距
- ✓ **电极覆盖物的厚度**
- ✓ **电极覆盖物的介电常数**
- ✓ 电极串联的阻尼电阻的大小
- ✓ 电极并联的ESD保护二极管电容大小
- ✓ MCU供电电压的大小



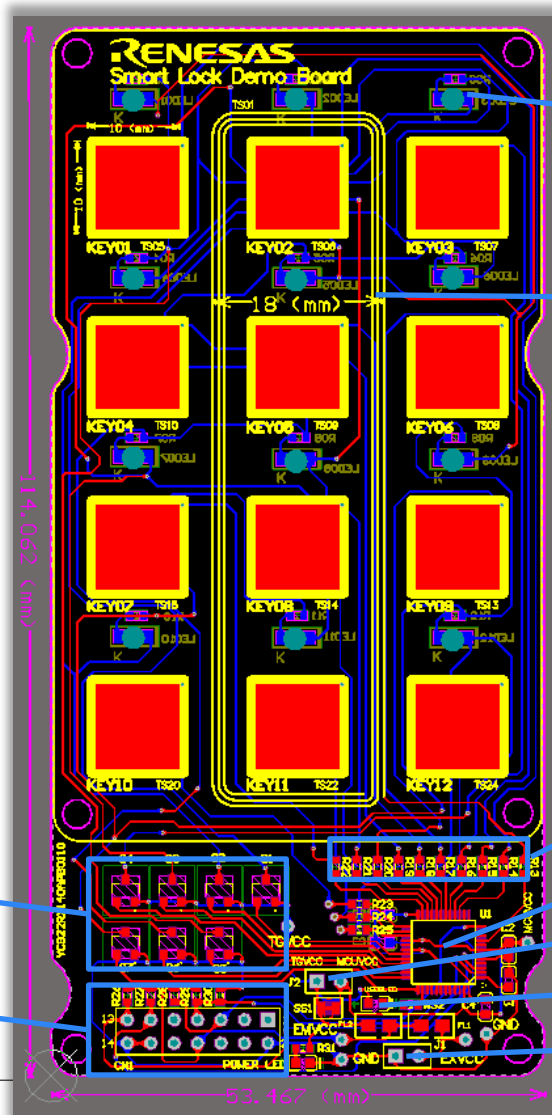
介电材料	ϵ
亚克力	2.4-4.5
玻璃	4.5-7.5
尼龙塑料	3.0-5.0
柔性乙烯基薄膜	3.2
空气	1.0
水	80

SMART LOCK应用中的按键电极设计举例

- ❑ 基于**RX140 MCU**
- ❑ 自容式按键设计示例
- ❑ 低功耗触摸评估板

3行 x 4列自容式触摸按键电极
(电极尺寸10mm x 10mm)

MEC电极(自容式)
(按键功能或者接近传感功能)



触摸按键LED指示

接近传感功能电极
高72mm x 长18mm的矩形环

LED驱动电路

仿真器接口

触摸通道阻尼电阻

RX140 48pin 256KB

电流测量端子

用户LED

外部电源输入接口

瑞萨电容触摸应用开发流程



瑞萨电容触摸应用开发流程

使用QE For Cap Touch进行CTSU开发

QE For Cap Touch 的使用说明

- 1.帮助文件

CapTouch Main (QE) View

- 1.Preparing the target project
- 2.Creating a touch-interface configuration
- 3.Tuning **【重点】**
- 4.Generating parameter files **【重点】**
- 5.Sample Code **【重点】**
- 6.Monitoring Connection **【重点】**
- 7.Software trigger / External trigger **【重点】**

CapTouch Board Monitor (QE) View

1. Recording / Playing Log

CapTouch Status Chart (QE) View

1. Displaying the Measured Values
2. Standard Deviation and SNR **【重点】**

CapTouch Multi Status Chart (QE) View

1. Display Multi TS Channel
2. Display Capacitance Converted from Count Value.

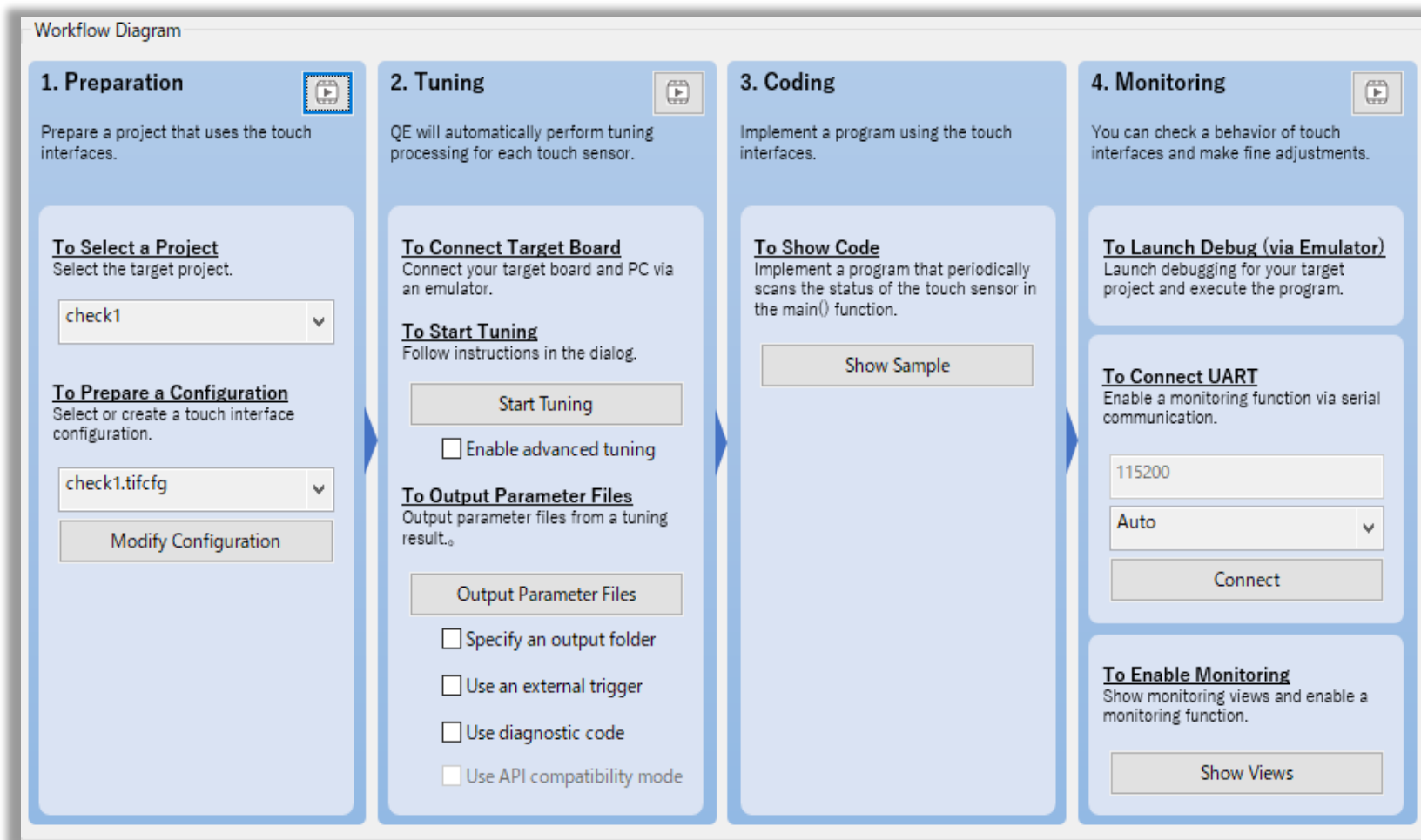
CapTouch Parameters (QE) View **【重点】**

1. Viewing and editing the Touch API parameters

瑞萨电容触摸应用开发流程

❖ 本节按照CTSUI触摸应用的开发步骤、QE For Cap Touch 工具的使用顺序，说明每个步骤的要点

❖ v3.10版本为基础说明



QE FOR CAP TOUCH 的使用说明

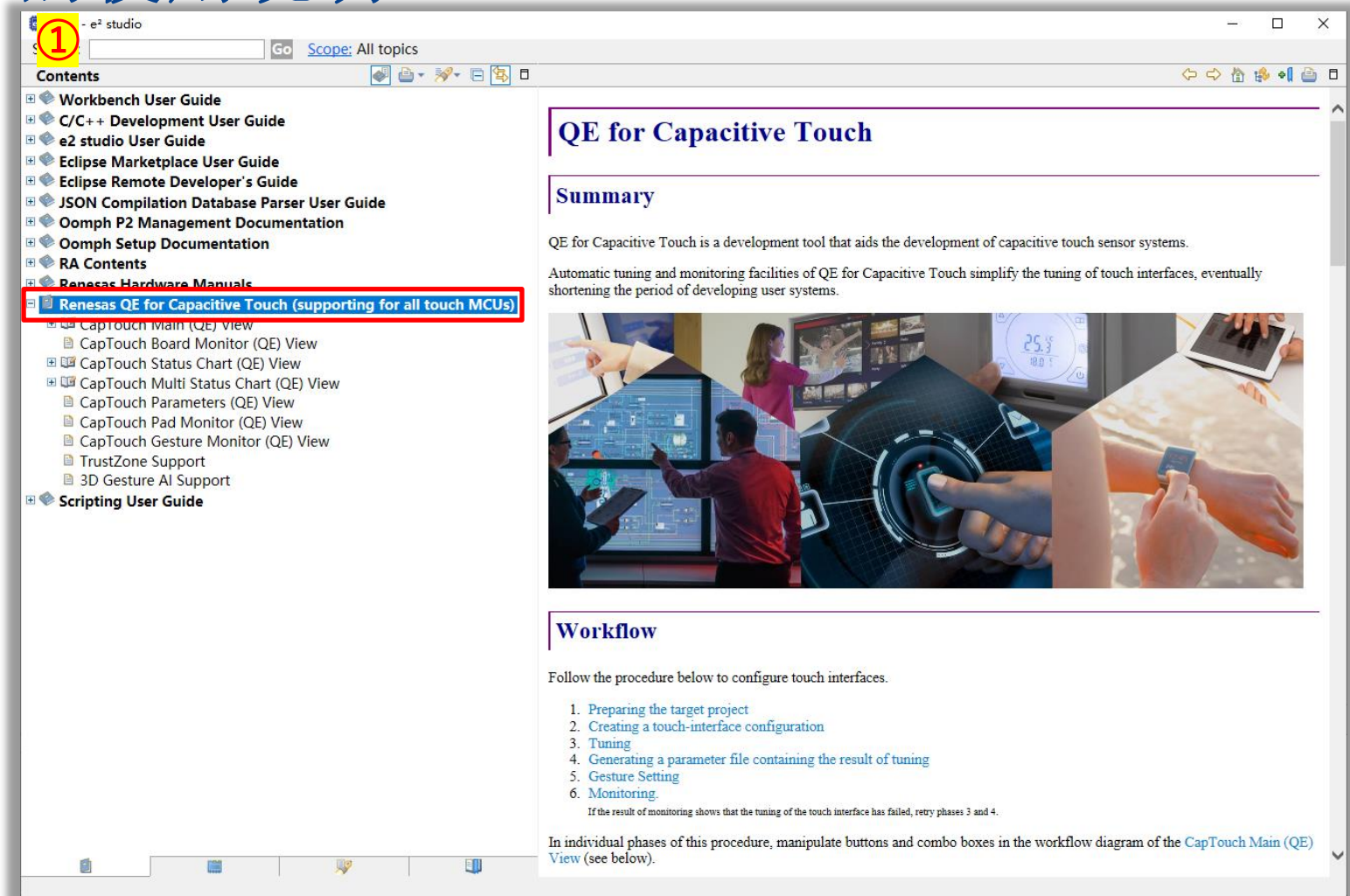
1. 帮助文件

① e2 studio Help

② [RL78 Family Using the standalone version of QE to Develop Capacitive Touch Applications](#)

②	Setup of QE for Capacitive Touch.....
8.1	Launching QE for Capacitive Touch.....
8.2	Preparation
8.3	Configuration
8.4	Tuning.....
8.5	Coding and Monitoring
8.5.1	Monitoring.....
8.6	Sample Code.....
8.7	Flowcharts

② Application note



① e2 studio → [Help] → [Help Contents]

CAPTOUCH MAIN (QE) VIEW

1.Preparing the target project准备目标工程

①在目标工程中添加r_ctsu, rm_touch驱动程序以及进行正确配置

步骤参考以下文档:

RA FSP : [Using QE and FSP to Develop Capacitive Touch Applications](#)

RX FIT : [Using QE and FIT to Develop Capacitive Touch Applications Rev.2.00](#)

RL78 SIS : [Using QE and SIS to Develop Capacitive Touch Applications Rev.2.10](#)

② r_ctsu的配置
以RX140为例

- ✓ 数据传输方式
- ✓ AJ功能使能
- ✓ Tscap使能
- ✓ 通道使能

Property	Value
Configurations	
# Parameter check	Use system default
# Data transfer of INTCTSUWR and INTCTSURD	Interrupt handler
# Select automatic judgement code	Disable
# Interrupt level for INTCTSUWR	Level 2
# Interrupt level for INTCTSURD	Level 2
# Interrupt level for INTCTSUFN	Level 2
Resources	
CTSUSUB	
~ TSCAP Pin	<input checked="" type="checkbox"/> Used
~ TS0 Pin	<input checked="" type="checkbox"/> Used
~ TS1 Pin	<input checked="" type="checkbox"/> Used
~ TS2 Pin	<input checked="" type="checkbox"/> Used
~ TS3 Pin	<input checked="" type="checkbox"/> Used
~ TS4 Pin	<input checked="" type="checkbox"/> Used
~ TS5 Pin	<input checked="" type="checkbox"/> Used
~ TS6 Pin	<input checked="" type="checkbox"/> Used
~ TS7 Pin	<input checked="" type="checkbox"/> Used
~ TS8 Pin	<input checked="" type="checkbox"/> Used
~ TS9 Pin	<input checked="" type="checkbox"/> Used
~ TS10 Pin	<input type="checkbox"/> Used
~ TS11 Pin	<input type="checkbox"/> Used
~ TS12 Pin	<input type="checkbox"/> Used

③ rm_touch的配置
以RX140为例

- ✓ 通过UART监控
- ✓ 通过UART tuning
- ✓ UART配置

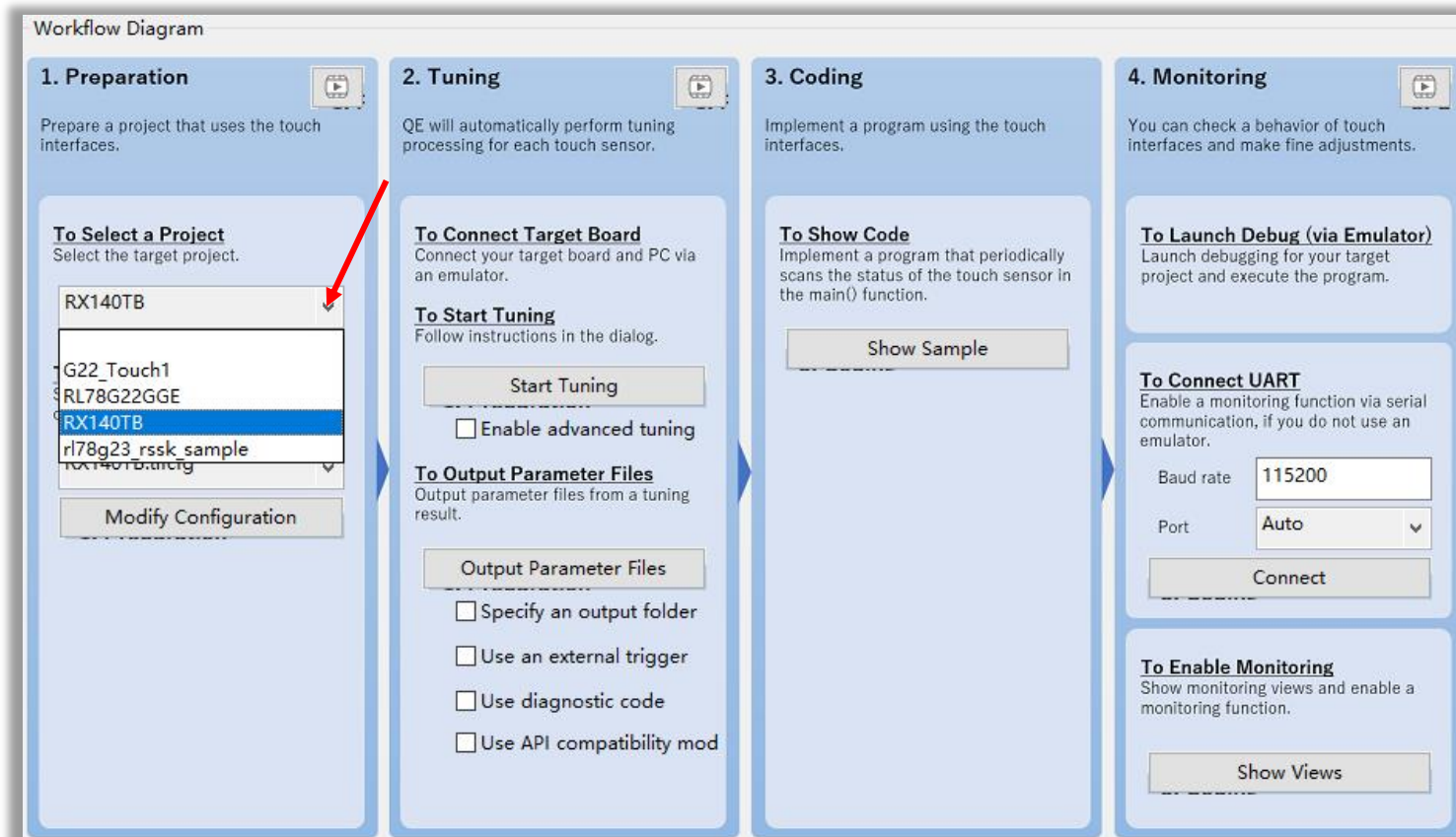
Property	Value
Configurations	
# Parameter check	Use system default
# Support for QE monitoring using UART	Sensor monitor not used
# Support for Serial tuning using UART	Serial tuning not used
# UART channel	UART0
# UART boadrate	115200
# UART priority	Level 1

CAPTOUCH MAIN (QE) VIEW

1. Preparing the target project 准备目标工程

④ 编译成功后

在 To select a Project 里选择目标工程



CAPTOUCH MAIN (QE) VIEW

Auto Sensing
Auto Judgement
说法不同意思相同

2. Creating a touch-interface configuration 创建触摸接口

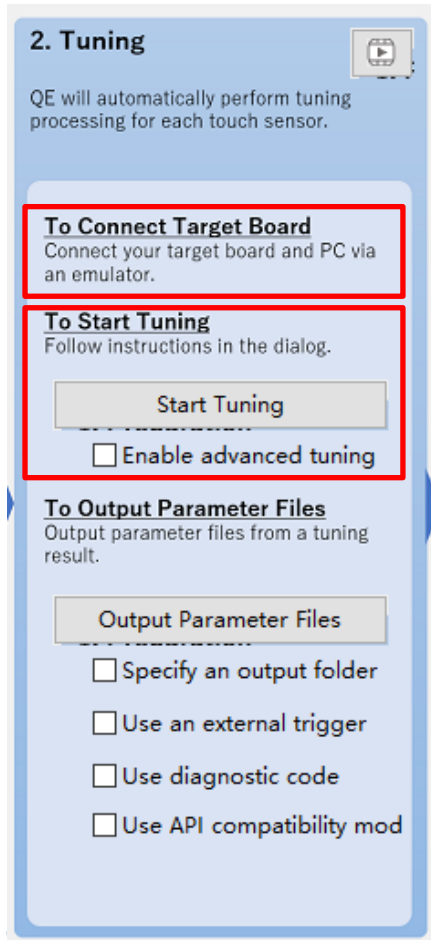
- ①可以在Setup configuration为按键分组建立不同的configuration，每个configuration可以为按键组设定独立的运行参数
- ②Auto sensing by hardware(RX140 256KB)和MEC需要在Setup configuration里为每个configuration独立设定

The screenshot displays the 'Create Configuration of Touch Interfaces' window. The 'File Name of Touch I/F' is 'RX140TB'. The 'Setup Configuration' button is highlighted with a red arrow pointing to the 'Setup Configurations (Methods)' dialog box. This dialog box shows two configurations: 'config01' and 'config02', both with 'Available' checked. Below the configurations, there are checkboxes for 'Auto Sensing by Hardware' and 'Multiple Electrode Connection' (MEC), both currently unchecked. Red arrows point from the 'Auto Sensing' and 'MEC' labels on the right to these checkboxes. The '1. Preparation' sidebar on the left shows the project selection and configuration preparation steps. Green boxes on the right label 'config01', 'config02', 'Auto Sensing', and 'MEC'.

CAPTOUCH MAIN (QE) VIEW

3.Tuning 【重点】

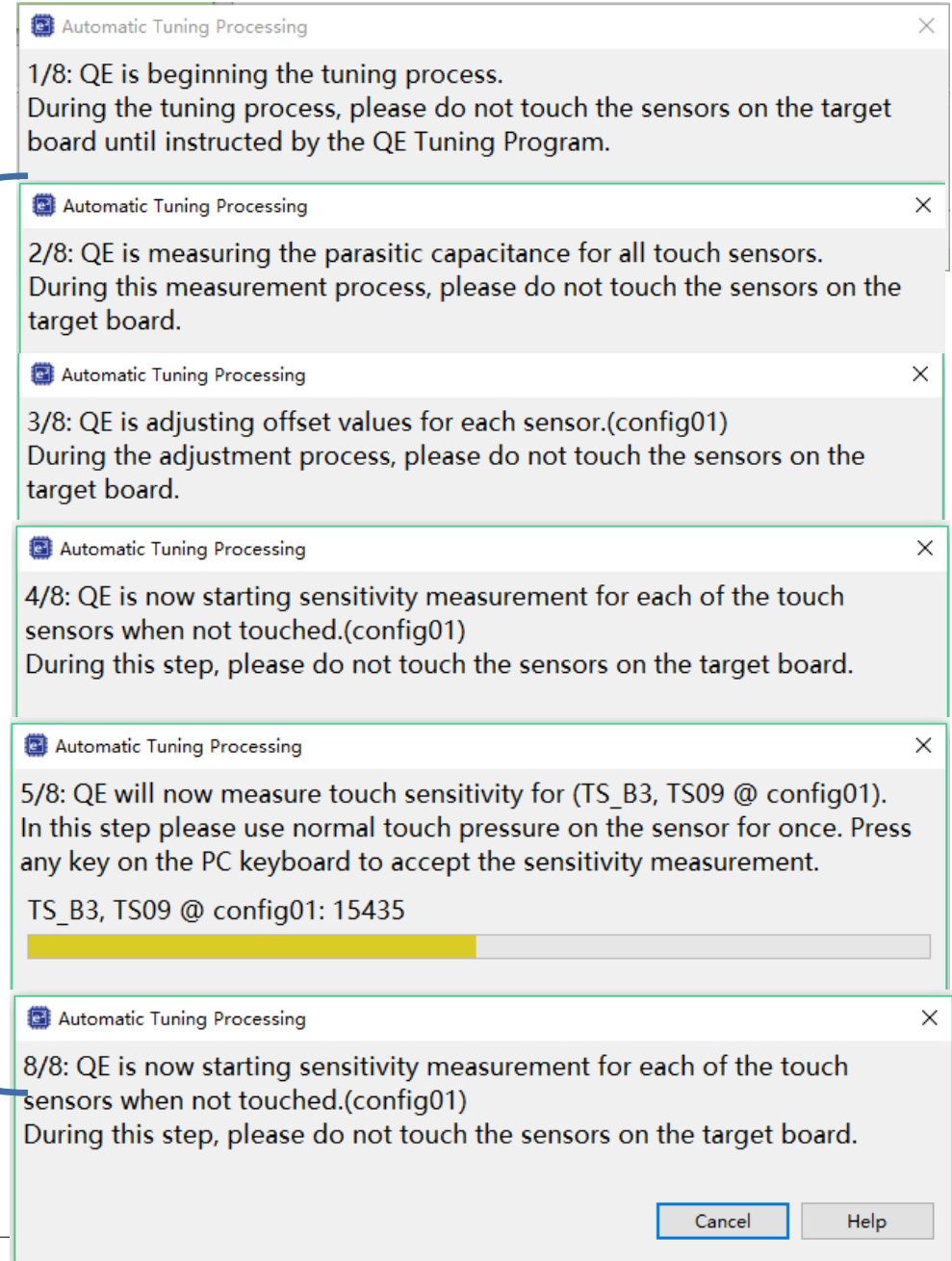
① 仿真器设定：仿真器供电or目标板供电



② Start Tuning开始自动调整

- 1.开始自动调整过程[1/8]
2. **Tuning Processing Phase 1:**
测量寄生电容[2/8]
3. **Tuning Processing Phase 2:**
偏置电流调整[3/8]
4. **Tuning Processing Phase 3:**
测量灵敏度[4/8~ 5/8 ~ 7/8]
- 5.结束自动调整过程 [8/8]

③ 显示Auto tuning的结果

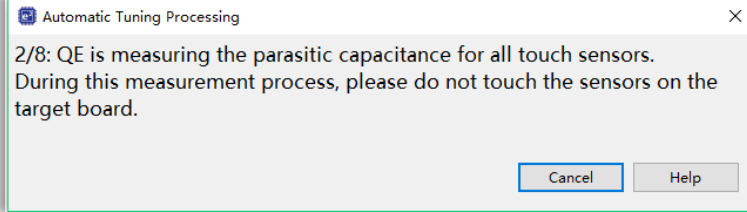


CAPTOUCH MAIN (QE) VIEW

3.Tuning 【重点】

Tuning Processing Phase 1: 测量寄生电容

- ❖ 测量寄生电容
- ❖ 决定最佳的Sensor驱动频率



以CTS1为例 (CTS2未发布)

如果电阻值可知,那么可以粗略地测量出寄生电容的容量值, 测量时, CTSU 1各寄存器的设定值如下所示, 各个触摸键上的寄生电容的粗略容值, 可以根据Sensor ICO counter 的测量值在表4.2 中查得。

- IO driver base clock 频率: **0.5MHz**
- CTSUATUNE0: Vcc≥2.4V 时设定为“0”, Vcc<2.4V 时设定为“1”
- CTSUATUNE1: 1 高输出
- CTSUPRRATIO: 3 推荐设定值
- CTSUPRMODE: 10B 基本脉冲数 = 62
- CTSUSOFF: 1 禁用高频噪声抑制功能 (OFF)
- CTSUSO: 00000000B 禁用 **Sensor Offset** 功能
- CTSUSNUM: 0 测量次数 = 1

表 4.2 Sensor ICO 测量值和寄生电容的大致对应关系

Sensor ICO count value	Capacitance (pF)
9801 or under	Under 9
9800-10200	Approximately 10
10201-11000	Approximately 12
11001-11900	Approximately 15
11901-13000	Approximately 18
13001-14600	Approximately 22
14601-16200	Approximately 27
16201-18000	Approximately 33
18001-19600	Approximately 39
19601-20300	Approximately 47
20301 or over	Over 50

利用测量所得的寄生电容值和已知的电阻值, 参考表 4.3 可以获得 IO diver 的驱动频率数。

表 4.3 寄生电容、电阻值和 IO Diver 驱动频率的关系

最佳的Sensor驱动频率



CTSU Basis of Cap touch detection Rev.1.00 Dec 25, 2014

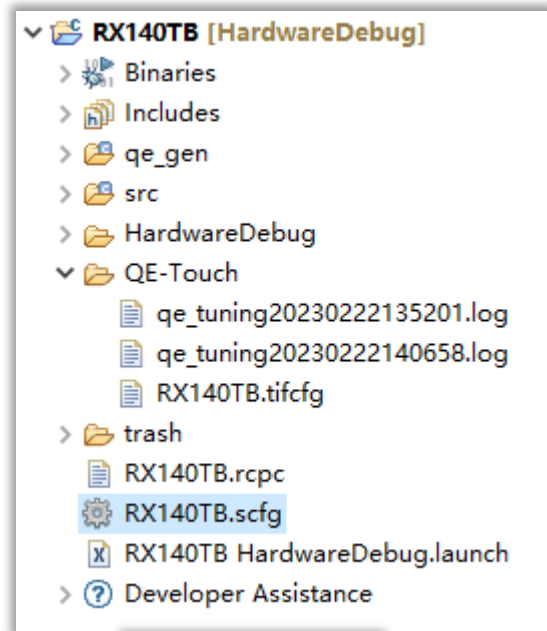
CAPTOUCH MAIN (QE) VIEW

3.Tuning 【重点】

Tuning Processing Phase 1:测量寄生电容

- ❖ 测量寄生电容
- ❖ 决定最佳的Sensor驱动频率

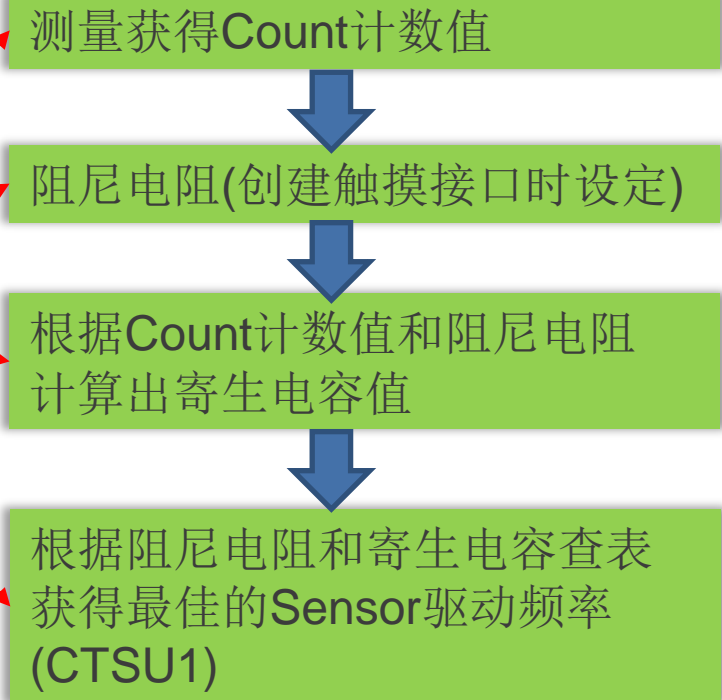
自动调整结束后，会在工程目录的QE-Touch文件夹生成*.log文件



工程目录

```
----- Phase1 -----  
<< METHOD 0 >>  
[TS28]  
Count Value : 2540  
Resistance Value (ohm) : 560  
Parasitic Capacitance (pF) : 17.63889  
Sensor Drive Pulse Frequency (MHz) : 2.0
```

*.log文件(部分截取)



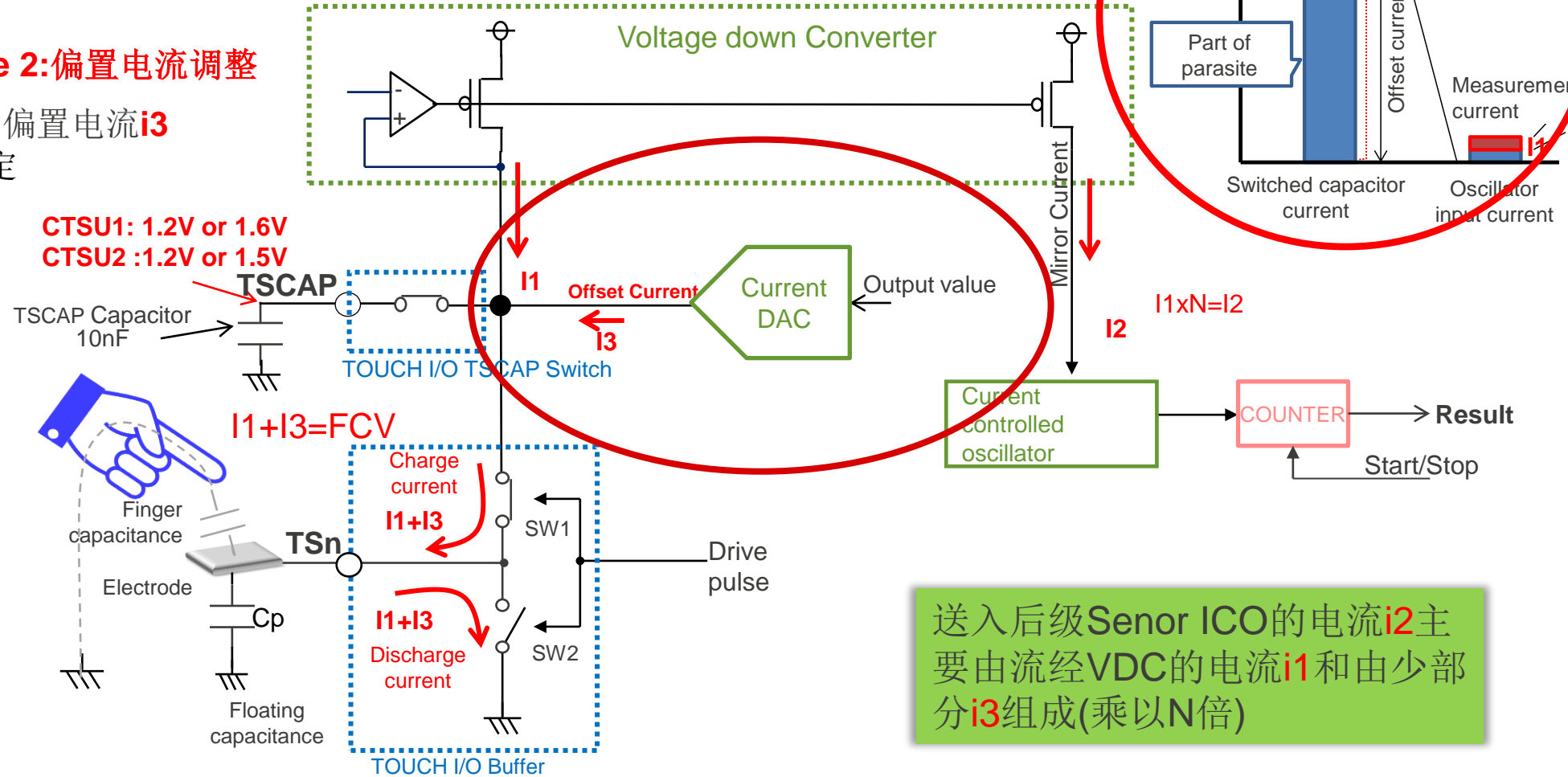
CAPTOUCH MAIN (QE) VIEW

3.Tuning 【重点】

Tuning Processing Phase 2: 偏置电流调整

❖ 右图中DAC Current为偏置电流*i3*
由CTSUSO寄存器设定

❖ 右图的右上角标出了偏置电流调整的
第一个意义：提供
寄生电容的充放电
电流(3.3pF-50pF)



CAPTOUCH MAIN (QE) VIEW

3.Tuning【重点】

Tuning Processing Phase 2: 偏置电流调整

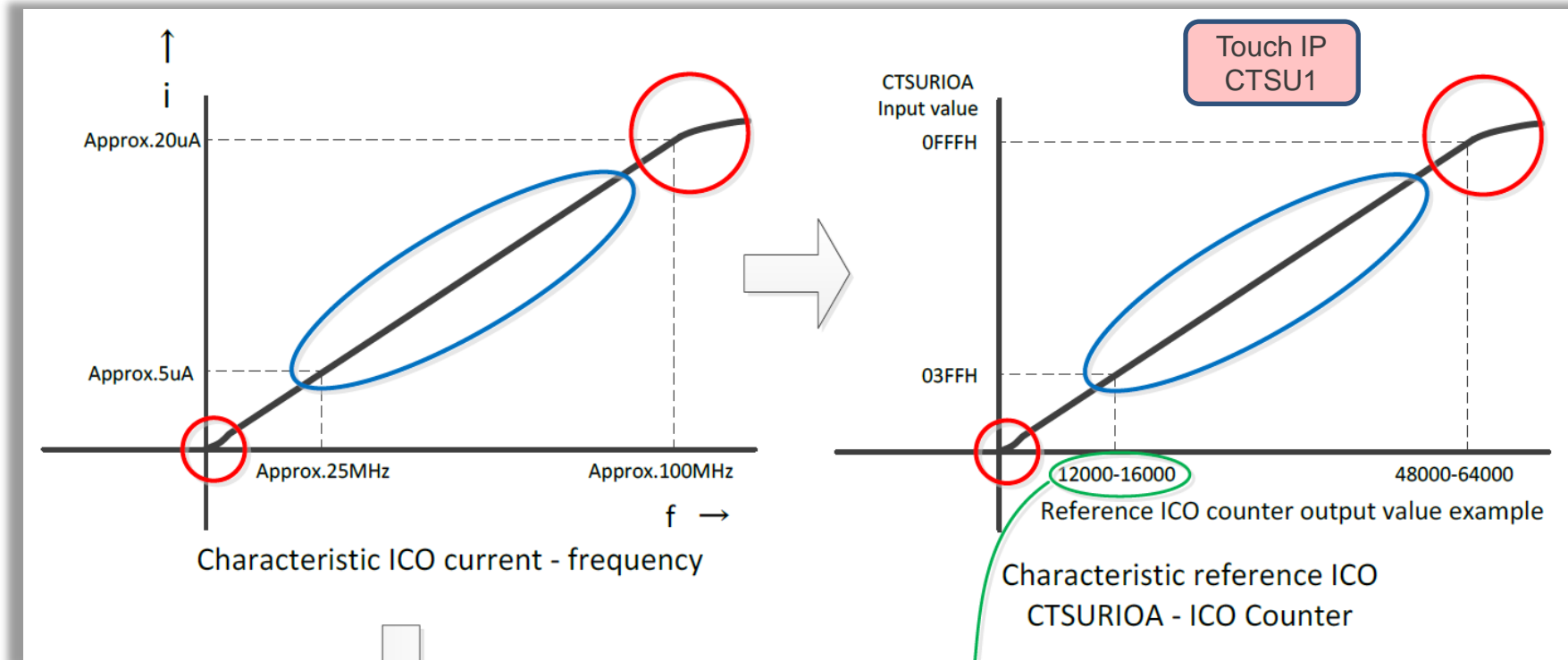
偏置电流调整的第二个意义:

❖ 以计数值**15360**[自容]/**10240**[互容]为目标, 调整偏置电流, 设定统一的**线性**工作起始点

Sensor ICO 是一种电流—频率转换器, 可以将输入的电流转换为成比例的振荡频率。

但是, 由于电流—频率呈线性关系的频域范围是有限的, 因此动态范围的上限约为100MHz。

如下图中的红色圆圈所示, 当频率高于100MHz 或很低时, 电流—频率不再保持线性关系。



Sensor ICO的线性区域

CAPTOUCH MAIN (QE) VIEW

3.Tuning 【重点】

Tuning Processing Phase 2:偏置电流调整

- ❖ 偏置电流调整在CTSUSO初始化时还会进行一次
- ❖ 此调整对QE工具生成的CTSUSO寄存器值进行轻微调整(主要是CTSUSO寄存器),以解决由于周围环境(如温度、湿度变化以及与其他电子部件或设备的接近度)生产线安装公差等等而产生的细微电容差异。此过程完成后,系统即被视为已调整并准备好运行。

补充:

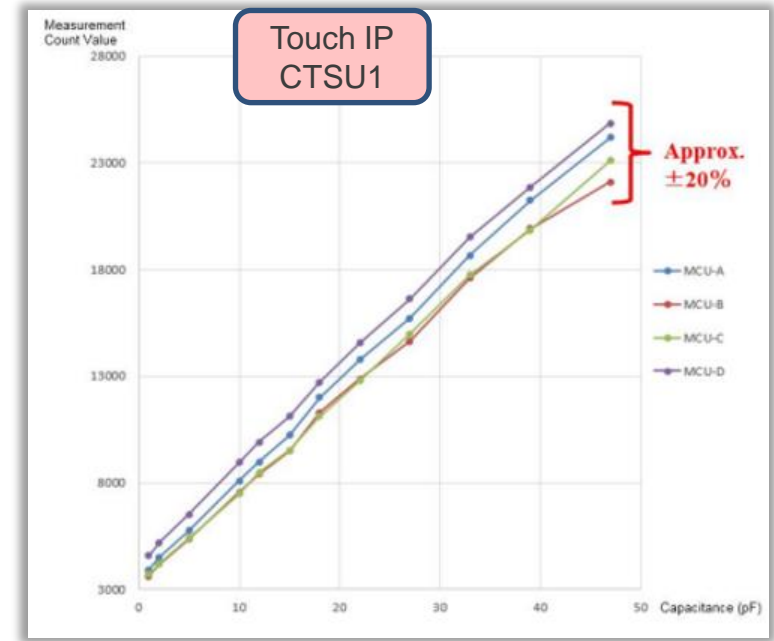
在CTSUSO初始化时,在初始化偏置电流调整之前,还有另一个校正过程。

ICO Correction

当CTSUSO外围设备首次初始化时,它暂时进入校正模式,模拟各种触摸输入值,并与理想的传感器计数进行比较。这些计数应沿直线下降,但实际上需要轻微调整。校正过程会改变内部CTSUSO寄存器值,并为触摸层生成校正系数,以确保尽可能准确的传感器读数。这解决了MCU制造过程中潜在的细微差异。

以下文档,对CTSUSO1的初始化偏置电流调整和ICO校正进行了详细解释

[Capacitive Touch Sensor Correction for Accuracy Enhancement Rev.1.00](#) Jan 18, 2017



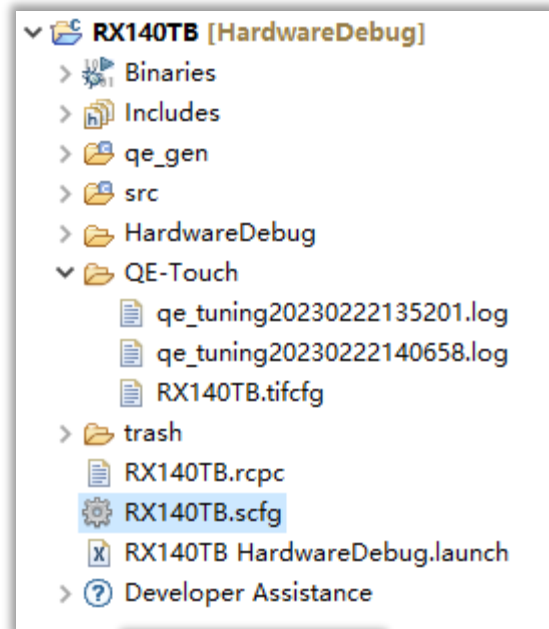
ICO Correction(Approx ±20%)

CAPTOUCH MAIN (QE) VIEW

3.Tuning 【重点】

Tuning Processing Phase 2:偏置电流调整

- ❖ 自动调整结束后，会在工程目录的QE-Touch文件夹生成*.log文件



```
----- Phase2 -----  
<< METHOD 0 >>  
[TS28]  
Sensor Drive Pulse Frequency (MHz) : 2.0  
CTSUSNUM : 15  
CTSUSDPA : 7  
CTSUSO : 206
```

*.log文件(部分截取)

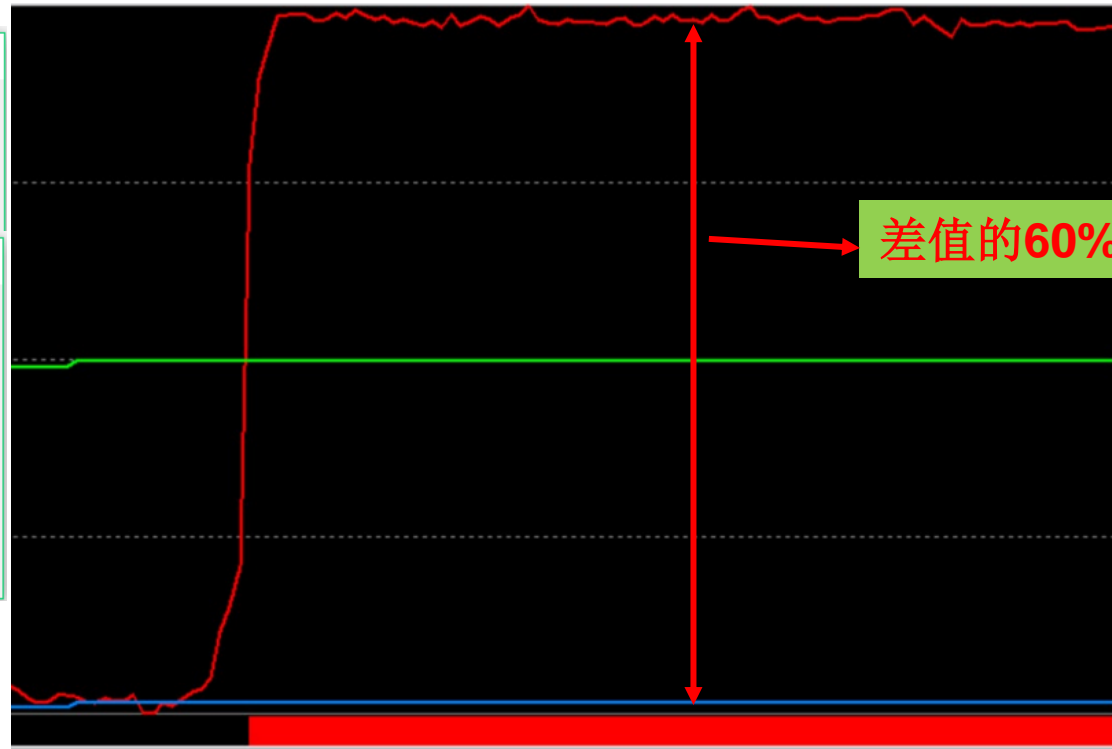
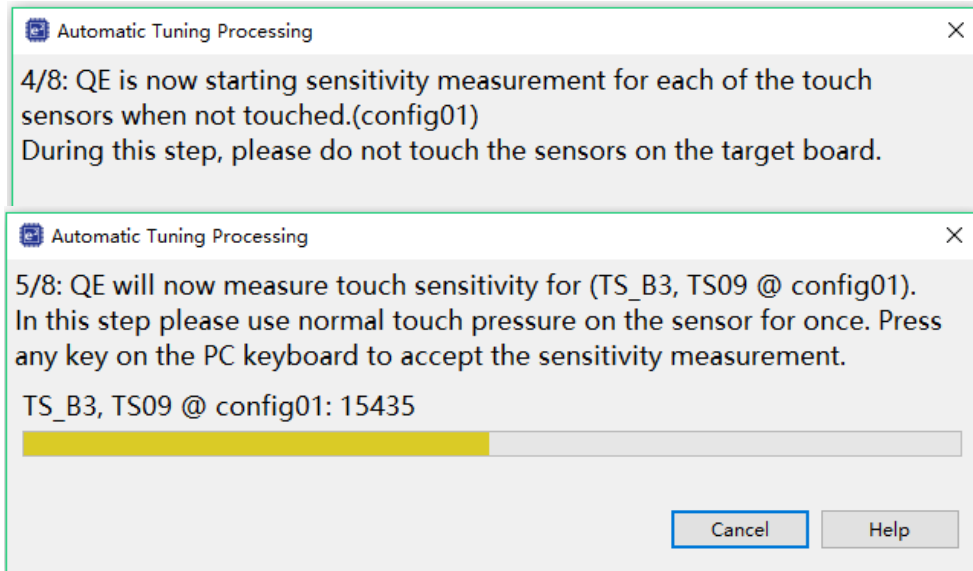
得到CTSUSO寄存器的设定值

CAPTOUCH MAIN (QE) VIEW

3.Tuning 【重点】

Tuning Processing Phase 3:测量灵敏度

- ❖ 灵敏度指的是**Threshold**阈值
- ❖ $(\text{Touch-ON计数值} - \text{Touch-OFF计数值}) \times 60\% = \text{Threshold}$ 阈值



CAPTOUCH MAIN (QE) VIEW

3.Tuning 【重点】

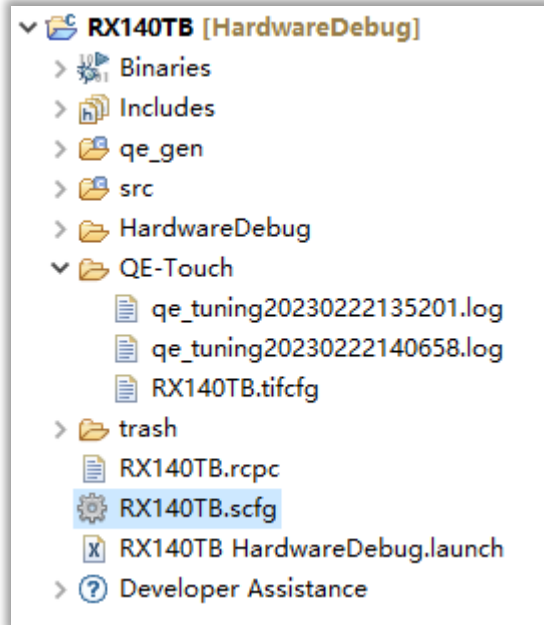
$\text{Threshold} = (\text{Count Value (On)} - \text{Count Value (Off)}) \times 0.6 = (17218 - 15086) \times 0.6 = 2132 \times 0.6 = 1279$

如果(Count Value (On) -Count Value (Off)) 等于负值, 那么Threshold等于65535

$\text{Hysteresis} = \text{Threshold} \times 0.05 = 1279 \times 0.05 = 63$

Tuning Processing Phase 3:测量灵敏度

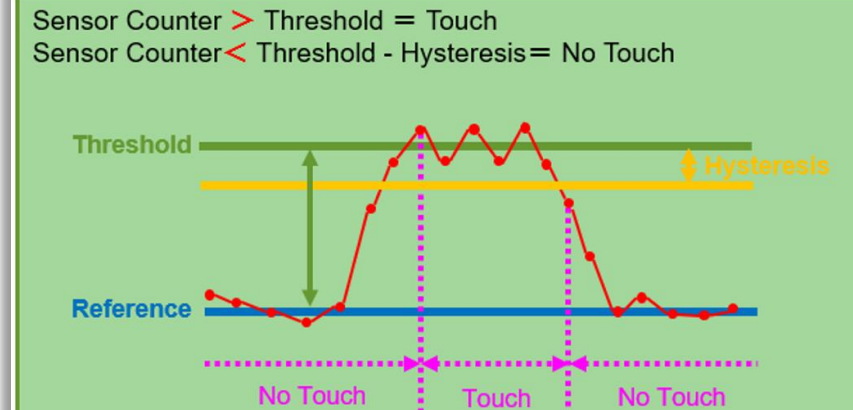
❖ 自动调整结束后, 会在工程目录的QE-Touch文件夹生成*.log文件



工程目录

```
----- Phase3 -----
<< METHOD 0 >>
[Button(self) 0]
<< TS28 >>
Count Value (Off)      : 15086
Count Value (On)      : 17218
Threshold              : 1279
Hysteresis             : 63
```

*.log文件(部分截取)



Hysteresis的意义

CAPTOUCH MAIN (QE) VIEW

3.Tuning 【重点】

自动调整全部完成后
在QE主页面下方显示结果

- ✓ 寄生电容【重点关注】
 - ❖ 寄生电容越小、驱动频率越高
 - ❖ 驱动频率越高、相同计数窗口内的计数值越高
 - ❖ 相同按键按压力度下，灵敏度越高
- ✓ Sensor驱动频率
- ✓ Threshold阈值
- ✓ 扫描时间

Prepare a project that uses the touch interfaces.

QE will automatically perform tuning processing for each touch sensor.

Implement a program using the touch interfaces.

You can check a behavior of touch interfaces and make fine adjustments.

To Select a Project
Select the target project.
RX140TB

To Prepare a Configuration
Select or create a touch interface configuration.
RX140TB.tifcfg
Modify Configuration

To Connect Target Board
Connect your target board and PC via an emulator.
To Start Tuning
Follow instructions in the dialog.
Start Tuning
 Enable advanced tuning

To Show Code
Implement a program that periodically scans the status of the touch sensor in the main() function.
Show Sample

To Launch Debug (via Emulator)
Launch debugging for your target project and execute the program.
Baud rate: 115200
Port: Auto
Connect

To Output Parameter Files
Output parameter files from a tuning result.
Output Parameter Files
 Specify an output folder
 Use an external trigger
 Use diagnostic code
 Use API compatibility mod

To Connect UART
Enable a monitoring function via serial communication, if you do not use an emulator.

To Enable Monitoring
Show monitoring views and enable a monitoring function.
Show Views

Tuning | Gesture

Touch I/F Configuration: RX140TB

Method	Kind	Name	Touch Sensor	Parasitic Capacitance[pF]	Sensor Drive Pulse Frequency[MHz]	Threshold	Scan Time[ms]	Overflow
config01	Button(self)	Button00	TS21	8.972	4.0	2349	0.576	None
config01	Button(self)	Button01	TS23	8.576	4.0	2865	0.576	None
config01	Button(self)	Button02	TS27	8.875	4.0	2475	0.576	None
config01	Button(self)	Button03	TS30	9.285	4.0	2592	0.576	None

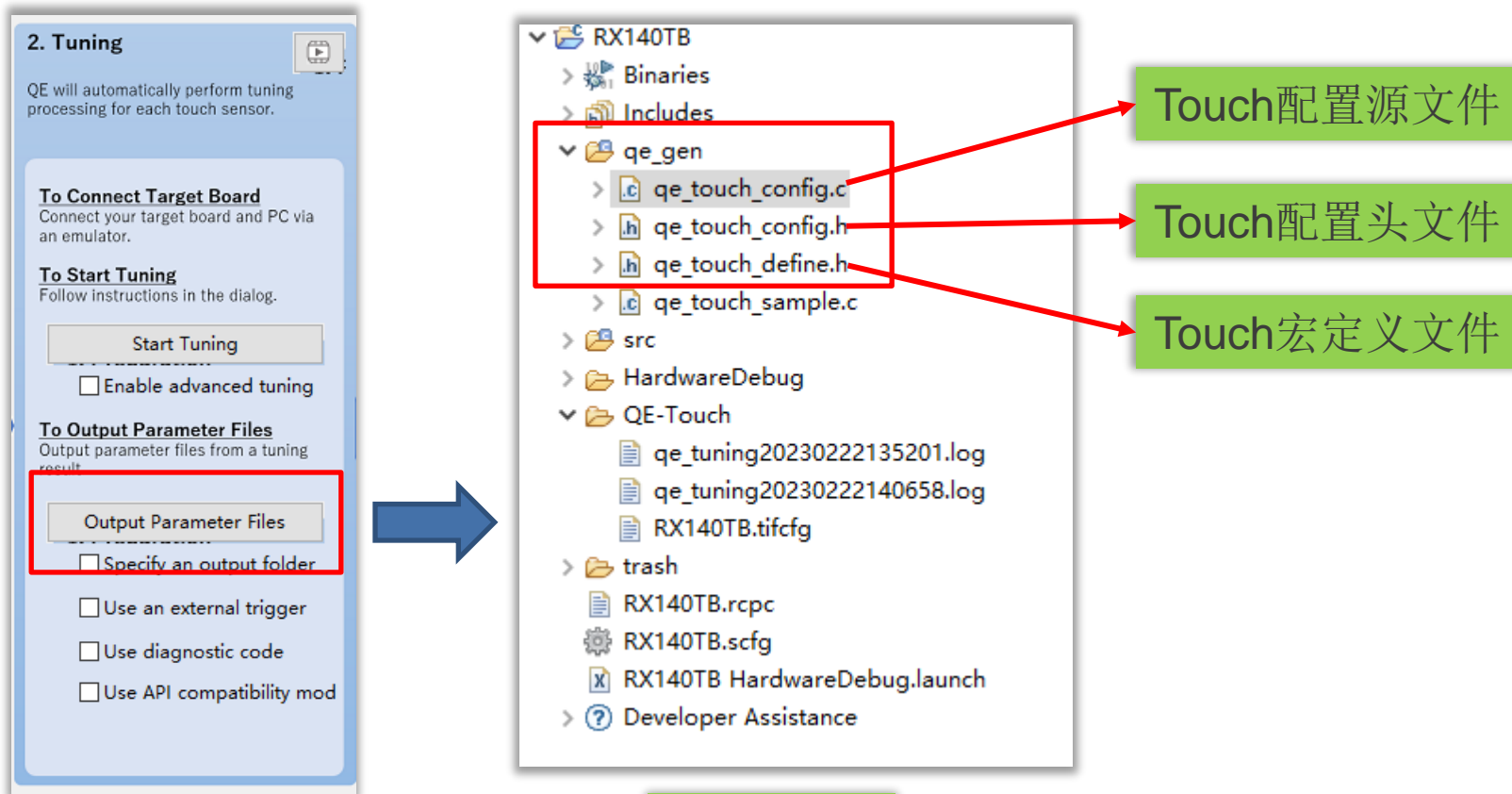
Auto Tuning Process结果

CAPTOUCH MAIN (QE) VIEW

4. Generating parameter files 【重点】

点击**Output parameter files**后，在工程目录下，生成**qe_gen**文件夹，并生成三个文件

qe_touch_config.c
qe_touch_config.h
qe_touch_define.h



工程目录

CAPTOUCH MAIN (QE) VIEW

4. Generating parameter files 【重点】

qe_touch_config.c文件，Touch配置源文件

CTSUS Related Information for [CONFIG01] configuration.

```
const ctsu_element_cfg_t g_qe_ctsu_element_cfg_config01[] =  
{  
  { .ssdiv = CTSU_SSDIV_4000, .so = 0x0CF, .snum = 0x07, .sdpa = 0x03 },  
  { .ssdiv = CTSU_SSDIV_4000, .so = 0x0C8, .snum = 0x07, .sdpa = 0x03 },  
  { .ssdiv = CTSU_SSDIV_4000, .so = 0x0D2, .snum = 0x07, .sdpa = 0x03 },  
  { .ssdiv = CTSU_SSDIV_4000, .so = 0x0DD, .snum = 0x07, .sdpa = 0x03 },  
};
```

g_qe_ctsu_element_cfg_config01[] =

ssdiv: CTSUS Spectrum Diffusion Frequency Division Setting (CTSUS1)

so : 偏置电流设定值

snum: 测量窗口设定值

sdpa: 驱动频率设定值

g_qe_ctsu_element_cfg_config01[] =



CAPTOUCH MAIN (QE) VIEW

4. Generating parameter files 【重点】

qe_touch_config.c文件，Touch配置源文件

g_qe_ctsu_cfg_config01 =

.cap = CTSU_CAP_SOFTWARE,
 CTSU的启动方式分为两种

```
typedef enum e_ctsu_cap
{
    CTSU_CAP_SOFTWARE, //< Scan start by software trigger API启动
    CTSU_CAP_EXTERNAL //< Scan start by external trigger 定时器启动
} ctstu_cap_t;
```

.tuning_self_target_value = 15360,
 线性工作起始点的计数值

.num_moving_average = 4,
 移动平均滤波的深度

.tunning_enable = true,
 初始化偏置电流调整使能

CTSUS Related Information for [CONFIG01] configuration.

```
const ctstu_cfg_t g_qe_ctsu_cfg_config01 =
{
    .cap = CTSU_CAP_SOFTWARE,

    .txvsel = CTSU_TXVSEL_VCC,
    .txvsel2= CTSU_TXVSEL_MODE,

    .atune12= CTSU_ATUNE12_40UA,
    .md      = CTSU_MODE_SELF_MULTI_SCAN,
    .pose1   = CTSU_POSEL_LOW_GPIO,

    .ctsuchac0 = 0x00, /* ch0-ch7 enable mask */
    .ctsuchac1 = 0x00, /* ch8-ch15 enable mask */
    .ctsuchac2 = 0xA0, /* ch16-ch23 enable mask */
    .ctsuchac3 = 0x48, /* ch24-ch31 enable mask */
    .ctsuchac4 = 0x00, /* ch32-ch39 enable mask */
    .ctsuchtrc0 = 0x00, /* ch0-ch7 mutual tx mask */
    .ctsuchtrc1 = 0x00, /* ch8-ch15 mutual tx mask */
    .ctsuchtrc2 = 0x00, /* ch16-ch23 mutual tx mask */
    .ctsuchtrc3 = 0x00, /* ch24-ch31 mutual tx mask */
    .ctsuchtrc4 = 0x00, /* ch32-ch39 mutual tx mask */
    .num_rx     = 4,
    .num_tx     = 0,
    .p_elements = g_qe_ctsu_element_cfg_config01,

    #if (CTSUS_TARGET_VALUE_CONFIG_SUPPORT == 1)
    .tuning_self_target_value = 15360,
    .tuning_mutual_target_value = 10240,
    #endif

    .num_moving_average = 4,
    .tunning_enable     = true,
    .p_callback         = &qe_touch_callback,
};
```

二. CAPTOUCH MAIN (QE) VIEW

4. Generating parameter files 【重点】

qe_touch_config.c文件，Touch配置源文件

```
g_qe_touch_button_cfg_config01[] =
```

```
.threshold = 2349,  
  button的阈值设定值
```

```
.hysteresis = 143,  
  button的hysteresis设定值
```

```
/* Button configurations */  
const touch_button_cfg_t g_qe_touch_button_cfg_config01[] =  
{  
    /* button00 */  
    {  
        .elem_index = 0,  
        .threshold = 2349,  
        .hysteresis = 117,  
    },  
    /* button01 */  
    {  
        .elem_index = 1,  
        .threshold = 2865,  
        .hysteresis = 143,  
    },  
    /* button02 */  
    {  
        .elem_index = 2,  
        .threshold = 2475,  
        .hysteresis = 123,  
    },  
    /* button03 */  
    {  
        .elem_index = 3,  
        .threshold = 2592,  
        .hysteresis = 129,  
    },  
};
```

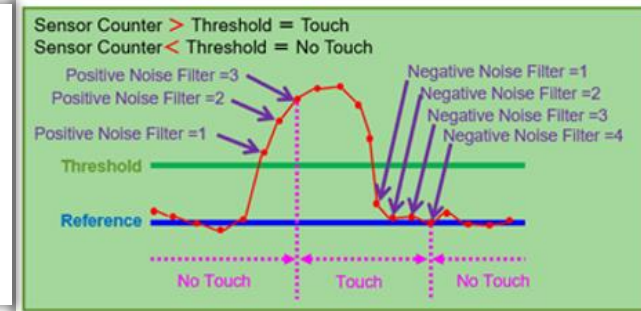
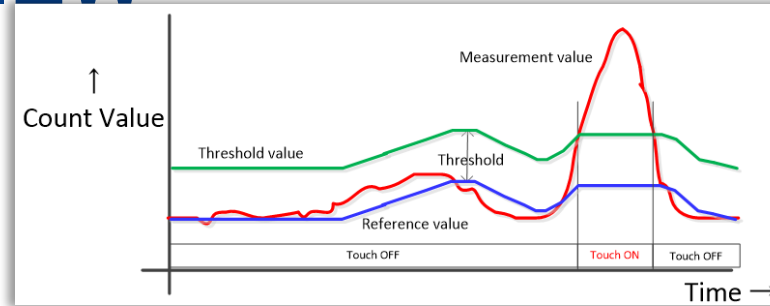
CAPTOUCH MAIN (QE) VIEW

4. Generating parameter files 【重点】

qe_touch_config.c文件，Touch配置源文件

```
g_qe_touch_button_cfg_config01[] =
```

```
.on_freq      = 3,
Positive Noise Filter Cycle 设定值
单位是控制周期
.off_freq     = 3,
Negative Noise Filter Cycle 的设定值
单位是控制周期
.drift_freq   = 255,
Drift Correction Interval 的设定值
单位是控制周期
.cancel_freq  = 0
Long Touch Cancel Cycle 的设定值
```



```
/* Touch configurations */
const touch_cfg_t g_qe_touch_cfg_config01 =
{
    .p_buttons      = g_qe_touch_button_cfg_config01,
    .p_sliders      = NULL,
    .p_wheels       = NULL,
    .num_buttons    = QE_TOUCH_CONFIG01_NUM_BUTTONS,
    .num_sliders    = QE_TOUCH_CONFIG01_NUM_SLIDERS,
    .num_wheels     = QE_TOUCH_CONFIG01_NUM_WHEELS,

    .number         = 0,

    .on_freq        = 3,
    .off_freq       = 3,
    .drift_freq     = 255,
    .cancel_freq    = 0,

    .p_ctsu_instance = &g_qe_ctsu_instance_config01,
};
```


CAPTOUCH MAIN (QE) VIEW

5. Sample Code 【重点】

3. Coding

Implement a program using the touch interfaces.

To Show Code
Implement a program that periodically scans the status of the touch sensor in the main() function.

Show Sample

Show Sample Code

Sample code of main() function:

```
/*  
 * FILE : qe_sample_main.c  
 * DATE : 2022-03-09  
 * DESCRIPTION : Main Program for RX  
 *  
 * NOTE:THIS IS A TYPICAL EXAMPLE.  
 */  
*****/  
#include "qe_touch_config.h"  
#if ((TOUCH_CFG_UART_MONITOR_SUPPORT == 1) || (TOUCH_CFG_UART_TUNING_SUPPORT ==  
#include "r_sci_rx_pinset.h"  
#endif  
#define TOUCH_SCAN_INTERVAL_EXAMPLE (20) /* milliseconds */  
  
void R_CTSU_PinSetInit(void);  
void qe_touch_main(void);  
  
#if ((TOUCH_CFG_UART_MONITOR_SUPPORT == 1) || (TOUCH_CFG_UART_TUNING_SUPPORT ==  
#if (TOUCH_CFG_UART_NUMBER == 0)  
#endif  
*****
```

Copy to the Clipboard Output to a File Show the Application Note

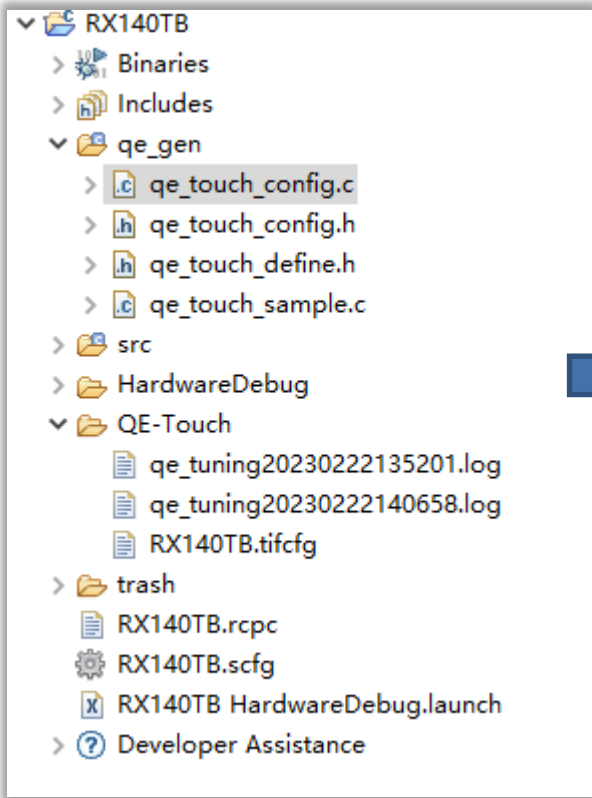
OK Help

- RX140TB
 - Binaries
 - Includes
 - qe_gen
 - qe_touch_config.c
 - qe_touch_config.h
 - qe_touch_define.h
 - qe_touch_sample.c
 - src
 - HardwareDebug
 - QE-Touch
 - qe_tuning20230222135201.log
 - qe_tuning20230222140658.log
 - RX140TB.tifcfg
 - trash
 - RX140TB.rcpc
 - RX140TB.scfg
 - RX140TB HardwareDebug.launch
 - Developer Assistance

工程目录

CAPTOUCH MAIN (QE) VIEW

5. Sample Code 【重点】



工程目录

```
void qe_touch_main(void)
{
    fsp_err_t err;
    /* Open Touch middleware */
    err = RM_TOUCH_Open(g_qe_touch_instance_config01.p_ctrl, g_qe_touch_instance_config01.p_cfg);
    if (FSP_SUCCESS != err)
    {
        while (true) {}
    }

    /* Main loop */
    while (true)
    {
        /* for [CONFIG01] configuration */
        err = RM_TOUCH_ScanStart(g_qe_touch_instance_config01.p_ctrl);
        if (FSP_SUCCESS != err)
        {
            while (true) {}
        }
        while (0 == g_qe_touch_flag) {}
        g_qe_touch_flag = 0;

        err = RM_TOUCH_DataGet(g_qe_touch_instance_config01.p_ctrl, &button_status, NULL, NULL);
        if (FSP_SUCCESS == err)
        {
            /* TODO: Add your own code here. */
        }
        /* FIXME: Since this is a temporary process, so re-create a waiting process yourself. */
        R_BSP_SoftwareDelay(TOUCH_SCAN_INTERVAL_EXAMPLE, BSP_DELAY_UNITS_MILLISECONDS);
    }
}
```

qe_touch_main

- ❖ Sample Code中使用**While(1)**死循环等待测量完成和**R_BSP_SoftwareDelay()**实现CTS API的20ms左右定时间隔的循环调用
- ❖ Captouch触摸按键对于实时性的要求并不高，但是涉及CTS运行的很多参数都与循环调用的定时间隔有关，因此较大程度决定了触摸按键的灵敏度，

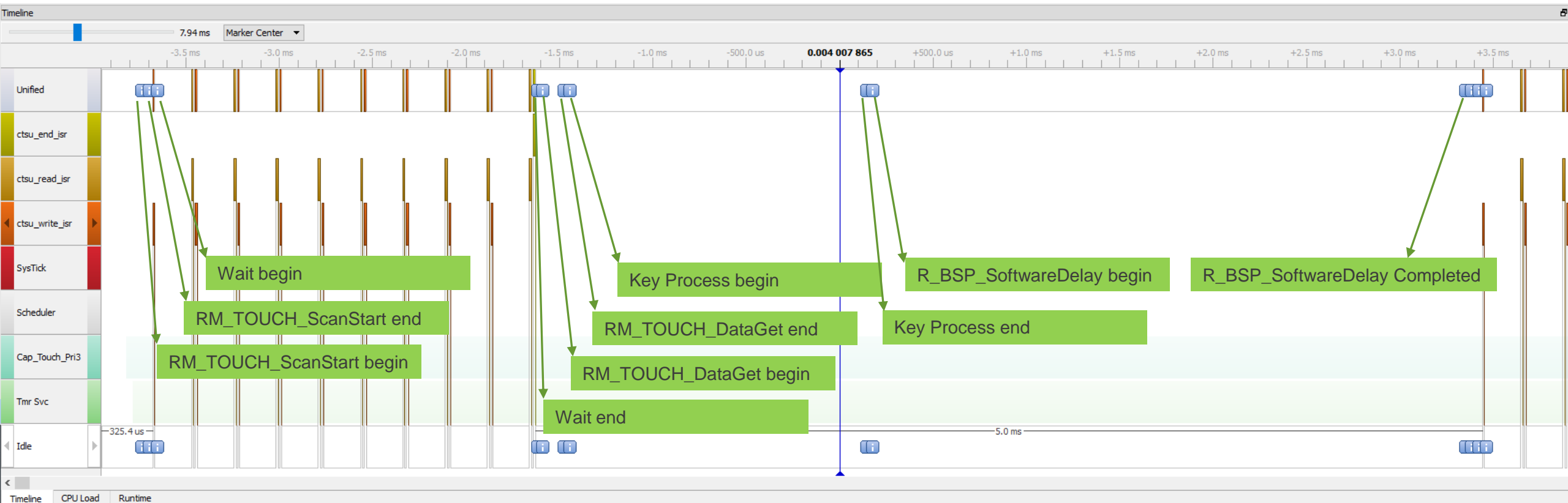
Item	Value
Drift Correction Interval	255
Long Touch Cancel Cycle	0
Positive Noise Filter Cycle	3
Negative Noise Filter Cycle	3
Moving Average Filter Depth	4
Touch Threshold	1244
Hysteresis	62

与定时调用
间隔有关

CAPTOUCH MAIN (QE) VIEW

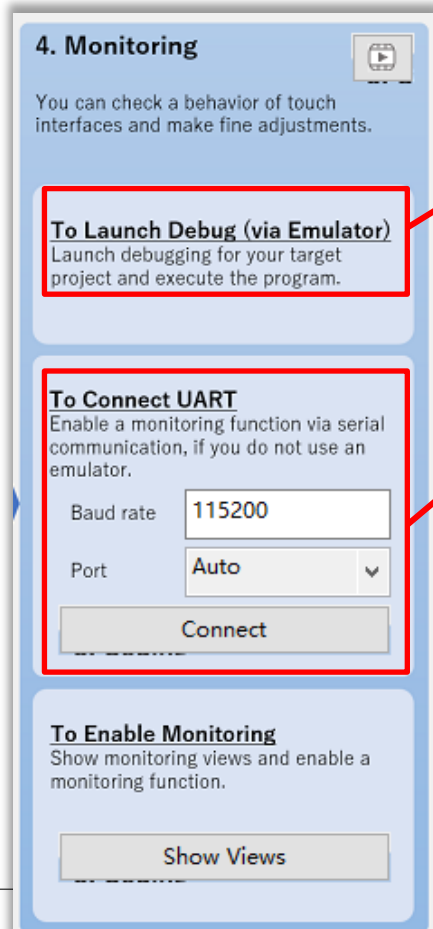
5.Sample Code 【重点】

CTSU在RTOS应用下，推荐使用辅助工具来分析task间的运行关系
借助RTOS和Segger Systemview对CTSU的运行进行分析



CAPTOUCH MAIN (QE) VIEW

6. Monitoring Connection 【重点】



通过仿真器连接，查看触摸按键的行为以及调整CTSUS运行参数
此时可以仿真器供电，也可以目标板供电
此时不需要设定串口驱动

通过串口连接，查看触摸按键的行为以及调整CTSUS运行参数
此时需要设定串口驱动
此时目标板脱离仿真器、单独供电运行

设定串口驱动的步骤详见以下文档

操作步骤

RA FSP : [Using QE and FSP to Develop Capacitive Touch Applications](#)

RX FIT : [Using QE and FIT to Develop Capacitive Touch Applications Rev.2.00](#)

RL78 SIS : [Using QE and SIS to Develop Capacitive Touch Applications Rev.2.10](#)

CAPTOUCH MAIN (QE) VIEW

7. Software trigger / External trigger 【重点】

❖ Software trigger

通过以下API启动测量

`RM_TOUCH_ScanStart`

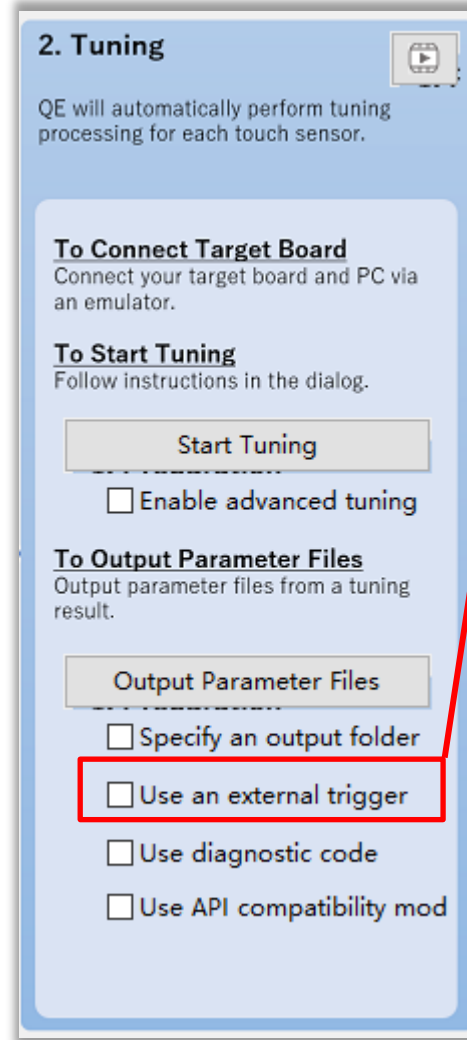
`(g_qe_touch_instance_config01.p_ctrl);`

❖ External trigger

通常由定时器启动

以RX140为例

```
(void)R_LPT_SetCMT(LPT_CH1, (uint32_t)WAKEUP_LPT_PERIOD_NORMAL);
do
{
    err = RM_TOUCH_ScanStart (g_qe_touch_instance_config02.p_ctrl);
    if (FSP_SUCCESS != err)
    {
        while (true) {}
    }
    (void)R_LPT_Control(LPT_CMD_START);
    while (0 == g_qe_touch_flag) {}
    g_qe_touch_flag = 0;
    err =
RM_TOUCH_DataGet(g_qe_touch_instance_config02.p_ctrl,&button_status02,NULL,NULL);
    if (FSP_SUCCESS == err)
    {
        /* TODO: Add your own code here. */
    }
}while(err != FSP_SUCCESS);
```



`.cap = CTSU_CAP_SOFTWARE,`
默认设定

`.cap = CTSU_CAP_EXTERNAL,`
勾选Use an external trigger后改为
`CTSU_CAP_EXTERNAL`

CAPTOUCH BOARD MONITOR (QE) VIEW

1. Recording / Playing Log



Play Monitoring Log of Touch Sensors with File



Stop Playing/Recording Log



Suspend/Resume Playing Log

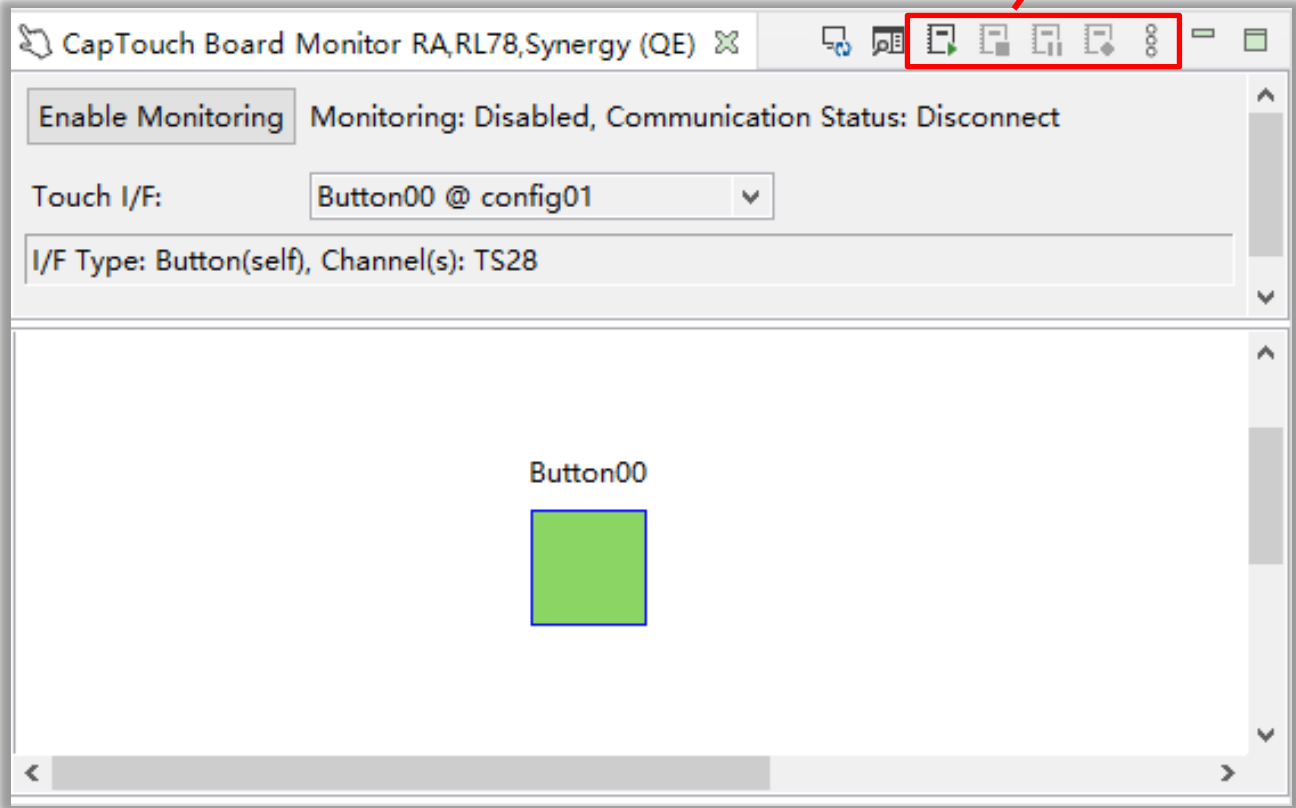


Record Monitoring Log of Touch Sensors to File



Save Contents of the Monitoring Table

Recording / Playing Log



CapTouch Board Monitor (QE) View

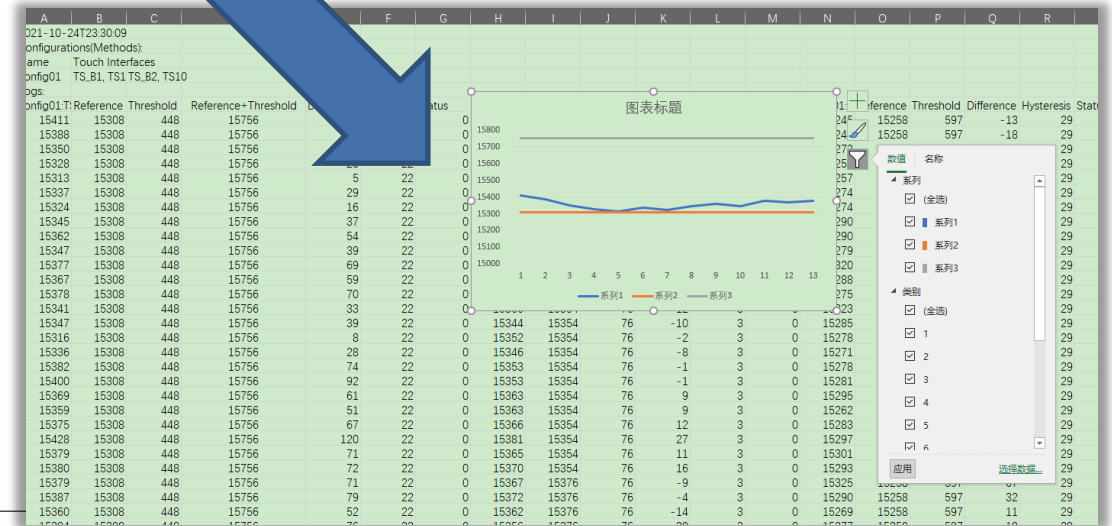
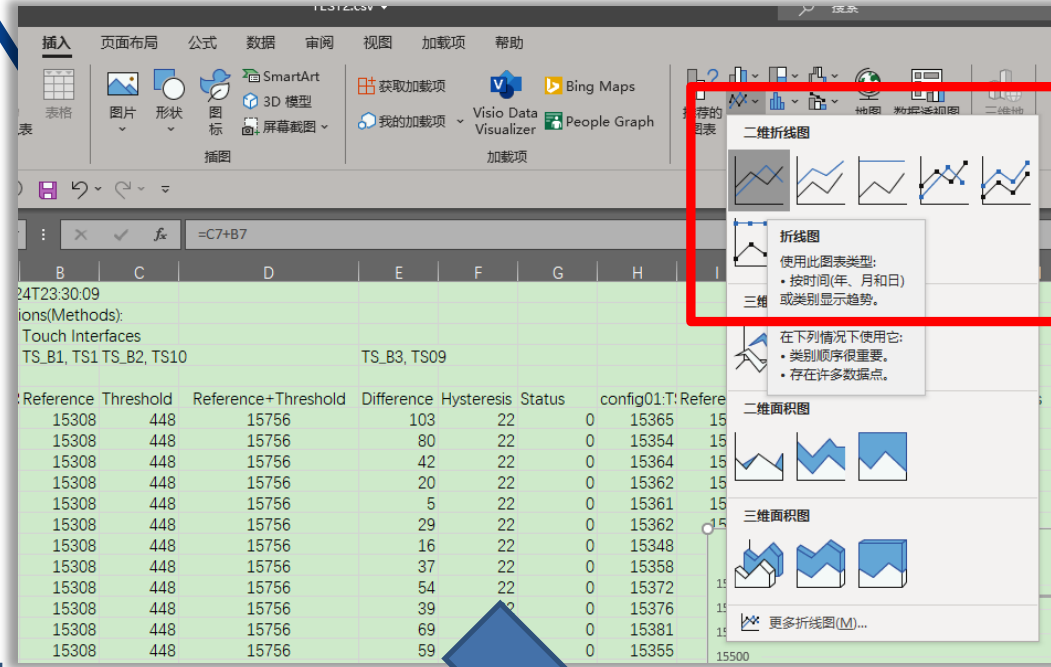
CAPTOUCH BOARD MON

1. Recording / Playing Log

Log文件以*CSV格式保存

可以使用excel的图表功能进行分析

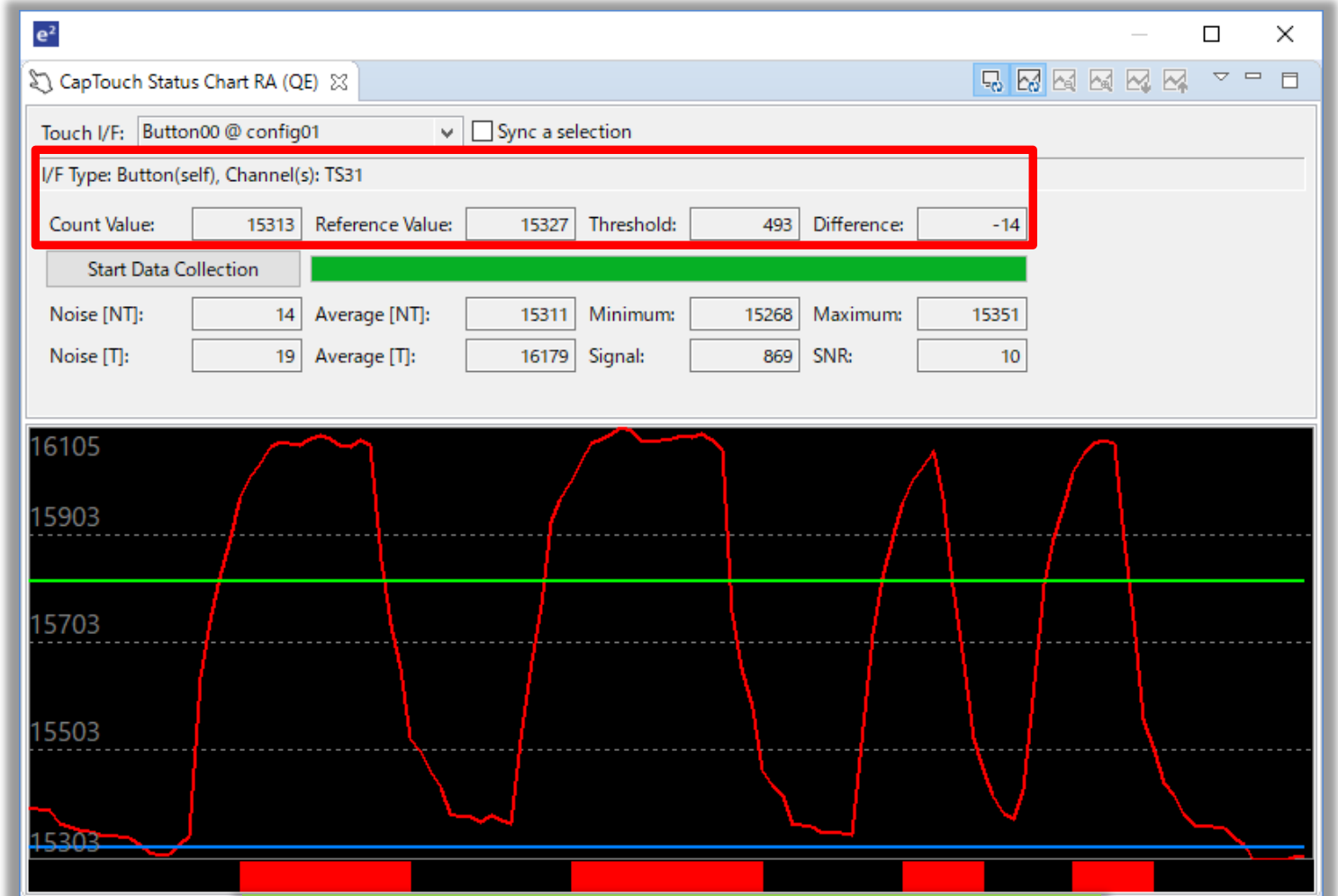
	A	B	C	D	E
1	2021-10-24T23:30:09				
2	Configurations(Methods):				
3	Name	Touch Interfaces			
4	config01	TS_B1, TS1	TS_B2, TS10		TS_B3, TS09
5	Logs:				
6	config01:T:Reference	Threshold	Reference+Threshold	Difference	
7	15411	15308	448	15756	103
8	15388	15308	448	15756	80
9	15350	15308	448	15756	42
10	15328	15308	448	15756	20
11	15313	15308	448	15756	5
12	15337	15308	448	15756	29
13	15324	15308	448	15756	16
14	15345	15308	448	15756	37
15	15362	15308	448	15756	54
16	15347	15308	448	15756	39
17	15377	15308	448	15756	69
18	15367	15308	448	15756	59
19	15378	15308	448	15756	70
20	15341	15308	448	15756	33
21	15347	15308	448	15756	39
22	15316	15308	448	15756	8
23	15326	15308	448	15756	20



CAPTOUCH STATUS CHART (QE) VIEW

1. Displaying the Measured Values

Count Value: 实时测量值【Touch-On】
Reference Value: 基准值【Touch-Off】
Threshold: 阈值(灵敏度)
Difference: Touch-On/Off差值



CapTouch Status Chart (QE) View

CAPTOUCH STATUS CHART (QE) VIEW

$$\text{SNR} = \frac{869}{15351 - 15268} = 10$$

2. Standard Deviation and SNR 【重点】

SNR的计算过程

•未接触测量[NT]

- ✓ 当未触摸目标触摸键时采集数据。
- ✓ 在数据采集期间，将根据采集的数据更新“最小值”和“最大值”并显示。
- ✓ 此外，还将计算并显示“噪声（未触摸）”（=标准偏差）和“平均值（未触碰到）”。
- ✓ 如果在采集期间触摸键状态从未触摸变为触摸，收集被中断并且数据被丢弃。

•接触测量[T]

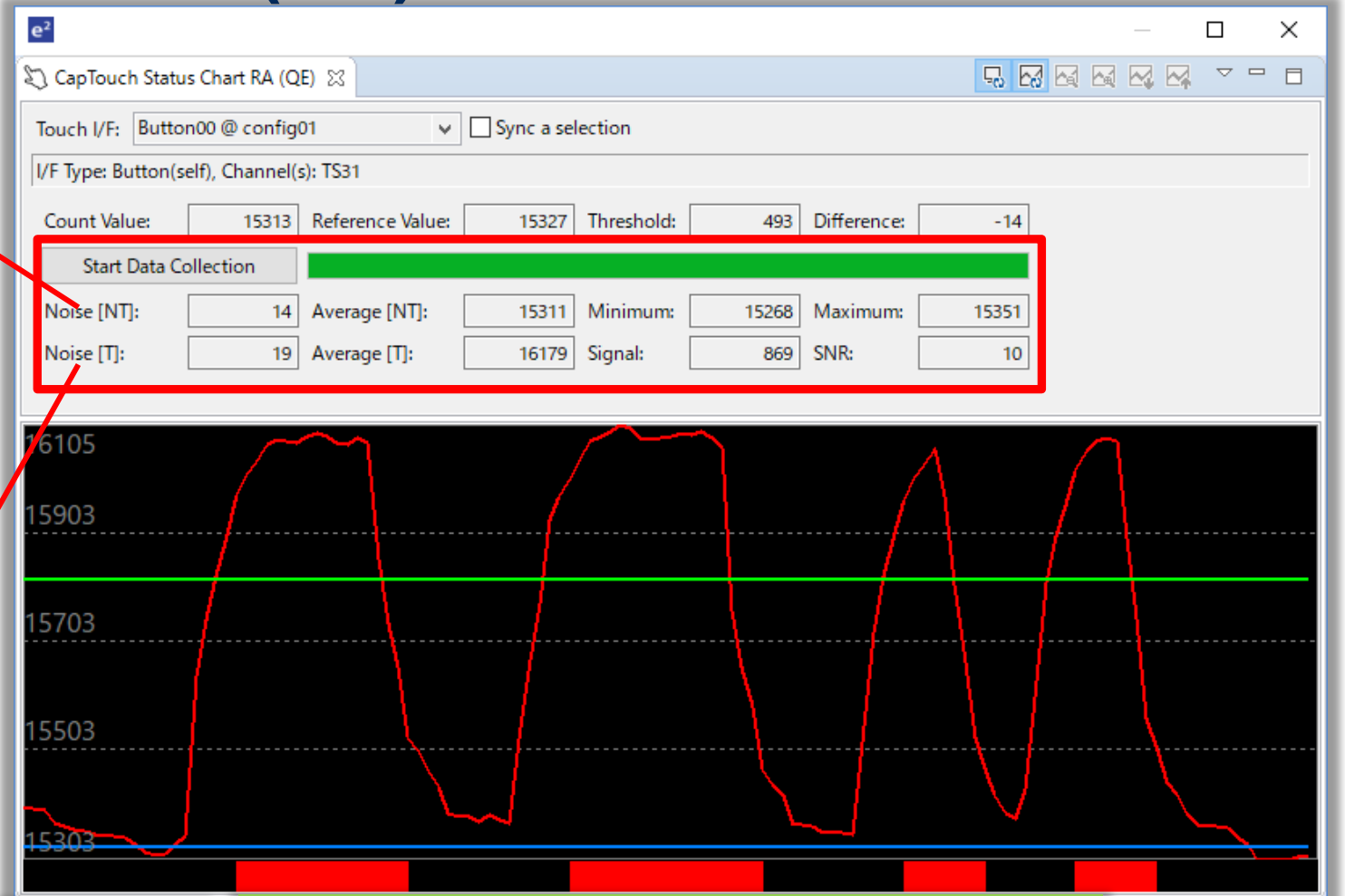
- ✓ 当触摸目标触摸键时采集数据。
- ✓ 在数据采集期间，将计算并显示“噪声（触摸）”（=标准偏差）和“平均值（接触）”。
- ✓ 如果在采集期间触摸键状态从触摸变为未触摸，采集将中断，数据将被丢弃。

操作步骤

RA FSP : [Using QE and FSP to Develop Capacitive Touch Applications](#)

RX FIT : [Using QE and FIT to Develop Capacitive Touch Applications Rev.2.00](#)

RL78 SIS : [Using QE and SIS to Develop Capacitive Touch Applications Rev.2.10](#)



CapTouch Status Chart (QE) View

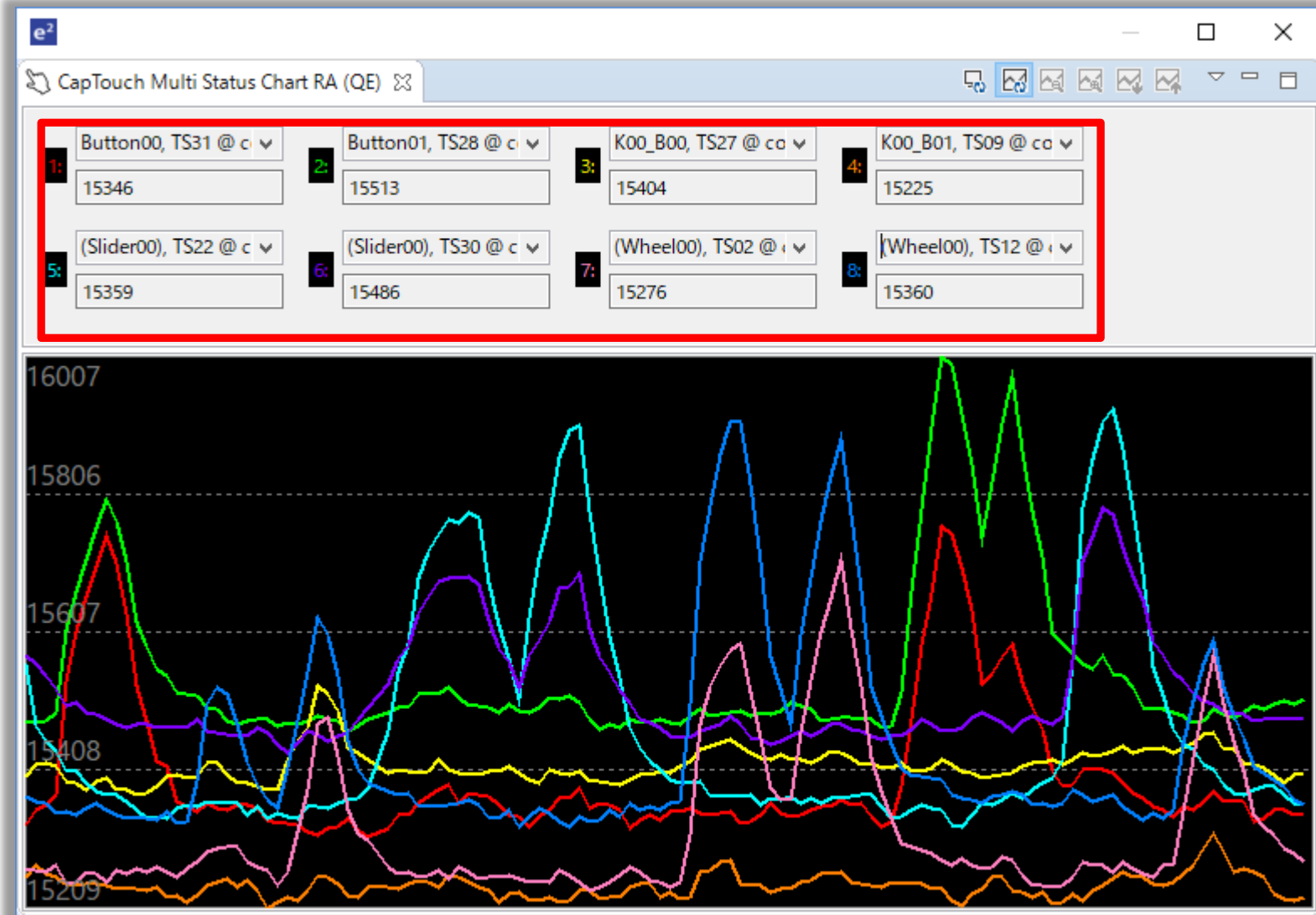
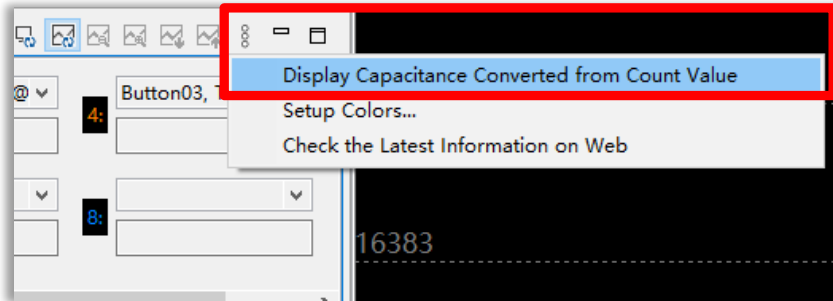
CAPTOUCH MULTI STATUS CHART (QE) VIEW

1. Display Multi TS Channel

可以显示多通道的实时测量值的波形和数值

2. Display Capacitance Converted from Count Value

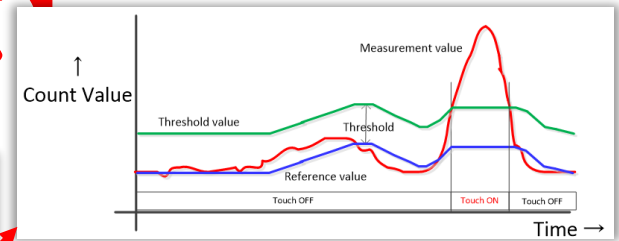
可以显示多通道的实时寄生电容值



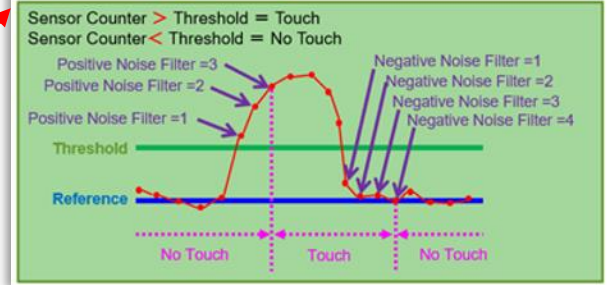
CAPTOUCH PARAMETERS (QE) VIEW 【重点】

1. Viewing and editing the Touch API parameters

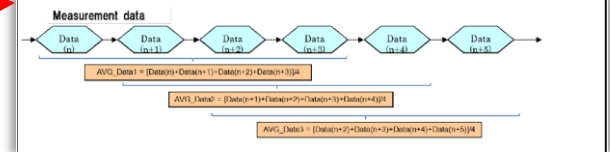
- ✓ 仿真状态下修改后点击  写入目标板立即生效
- ✓ 调整完成后点击  写入配置文件重新编译后生效



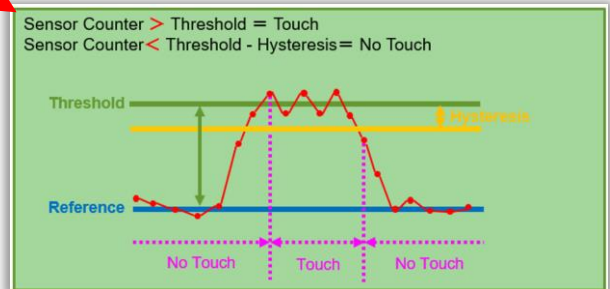
Drift Correction Interval



Positive/Negative Noise Filter Cycle



Moving Average Filter



Touch Threshold and Hysteresis

CapTouch Parameters RA (QE)

Touch I/F: Sync a selection

I/F Type: Button(self), Channel(s): TS31

Item	Value
Drift Correction Interval	255
Long Touch Cancel Cycle	0
Positive Noise Filter Cycle	3
Negative Noise Filter Cycle	3
Moving Average Filter Depth	0
Touch Threshold	493
Hysteresis	24
CTSUSO	51
CTSUSNUM	0
CTSUSDPA	Operating clock divided by 64

Set a drift correction interval. Drift Correction is a function to make the reference value follow the surrounding environment. Input a value between 0 and 65535.

- The value is 1 or more: The reference value will be corrected every cycle specified in the [Drift Correction Interval] item.
- The value is 0: No correction.

This setting item will be applied for each method.

Display in Advanced Mode

CTSUSO	51
CTSUSNUM	0
CTSUSDPA	Operating clock divided by 64

Read Value from the Target Board

Write Value to the Target Board

Output Parameter File

修改qe_touch_config.c文件

瑞萨电容触摸低功耗应用软件开发



瑞萨电容触摸低功耗应用软件开发

四.瑞萨电容触摸低功耗应用软件开发

1. 瑞萨MCU低功耗工作模式
2. CTSU低功耗应用的软件工作流程
3. RX140 MCU的MEC+AJ新功能
4. 影响功耗的主要因素

瑞萨MCU低功耗工作模式

□瑞萨MCU低功耗模式简介

MCU系列		睡眠模式				
		轻度睡眠模式	深度睡眠模式	小睡/打盹模式	待机模式	深度待机模式
RL78	RL78G23	HALT mode	X	SNOOZE mode	STOP mode	X
RA	RA2	Sleep mode	X	SNOOZE mode	Software Standby mode	X
	RA4、RA6	Sleep mode	X	SNOOZE mode	Software Standby mode	Deep Software Standby mode
RX	RX140	Sleep mode	Deep sleep mode	SNOOZE mode	Software Standby mode	X

低功耗模式 (功耗由高到低)	Sleep mode	轻度睡眠模式: CPU停止工作, 但其内部寄存器的内容被保留。其他片上外设功能不停止。在此模式下, 所有的中断源都可以退出轻度睡眠模式。
	HALT mode	
	Deep sleep mode	深度睡眠模式: CPU、RAM、ROM停止工作, 但其内容被保留。其他片上外设功能不停止。
	SNOOZE mode	小睡/打盹模式: 是待机模式的扩展。在此模式下, 有限的外设模块可以在不唤醒CPU的情况下运行。通过配置请求和结束条件, 可以在待机模式下进入或者退出小睡/打盹模式。通过配置取消条件, 小睡/打盹模式也可以返回到正常模式。
	STOP mode	待机模式: CPU, 大部分片上外设功能和振荡器停止工作。但是, CPU内部寄存器的内容和SRAM数据, 片上外设功能的状态和I/O端口状态都被保留。该模式可以显著降低功耗。
	Software Standby mode	
	Deep Software Standby mode	深度待机模式: CPU, 大部分片上外设功能, SRAM(standby RAM除外)和大部分振荡器停止工作。此外, 由于停止了对这些模块的内部电源供应, 因此功耗显著降低。所有CPU寄存器和大部分外设模块的内容变为未定义。

CTSU低功耗应用的软件工作流程

□ CTSU低功耗应用的软件工作流程

❖ CTSU配合不同的MCU低功耗模式，可实现不同的低功耗要求

□ CTSU/CTS2/CTS2L

✓ STOP

✓ STOP + Snooze

□ CTS2SL

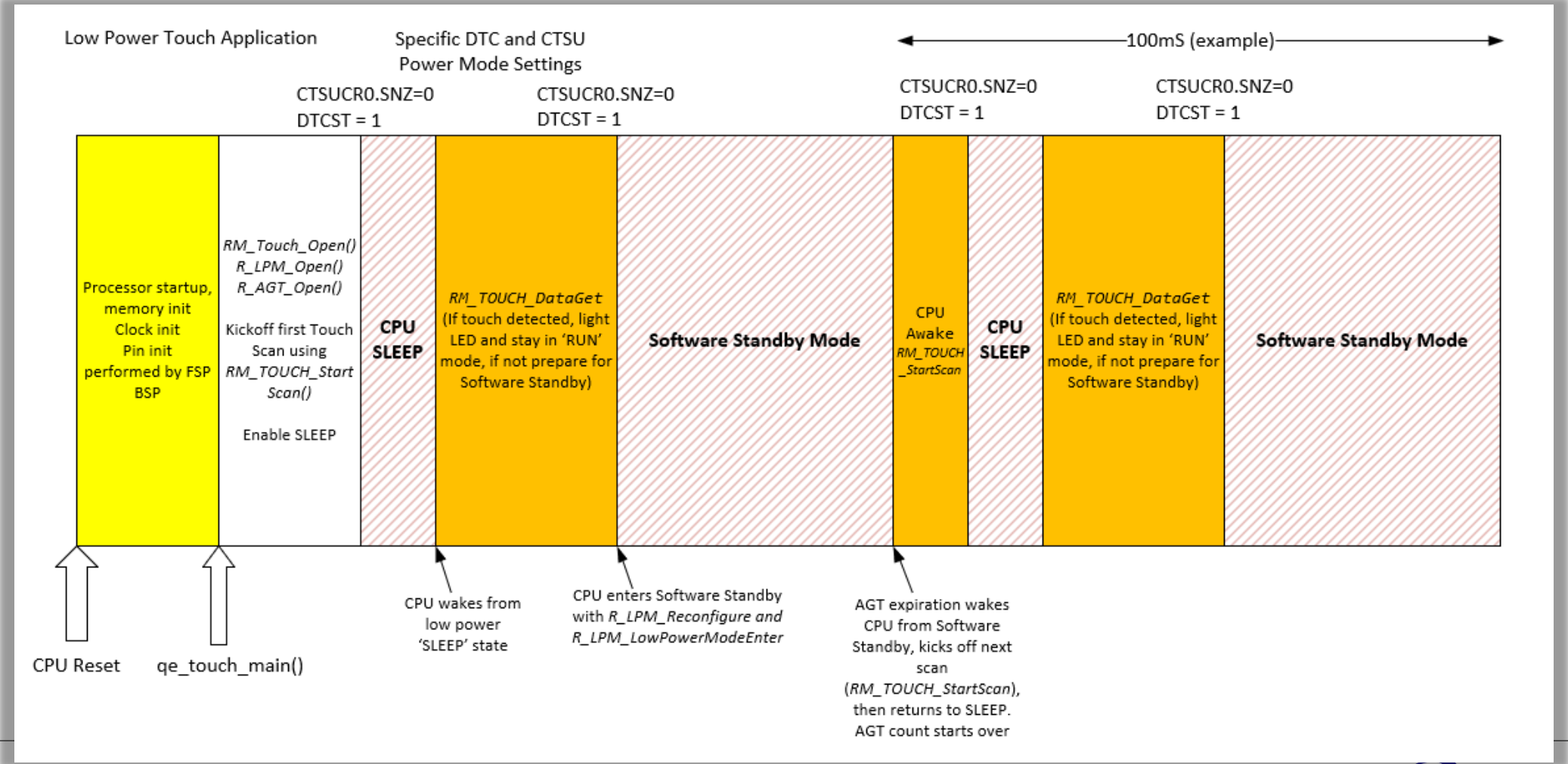
✓ STOP

✓ STOP + Snooze

✓ STOP + Snooze + MEC + AJ

CTSUSU低功耗应用的软件工作流程

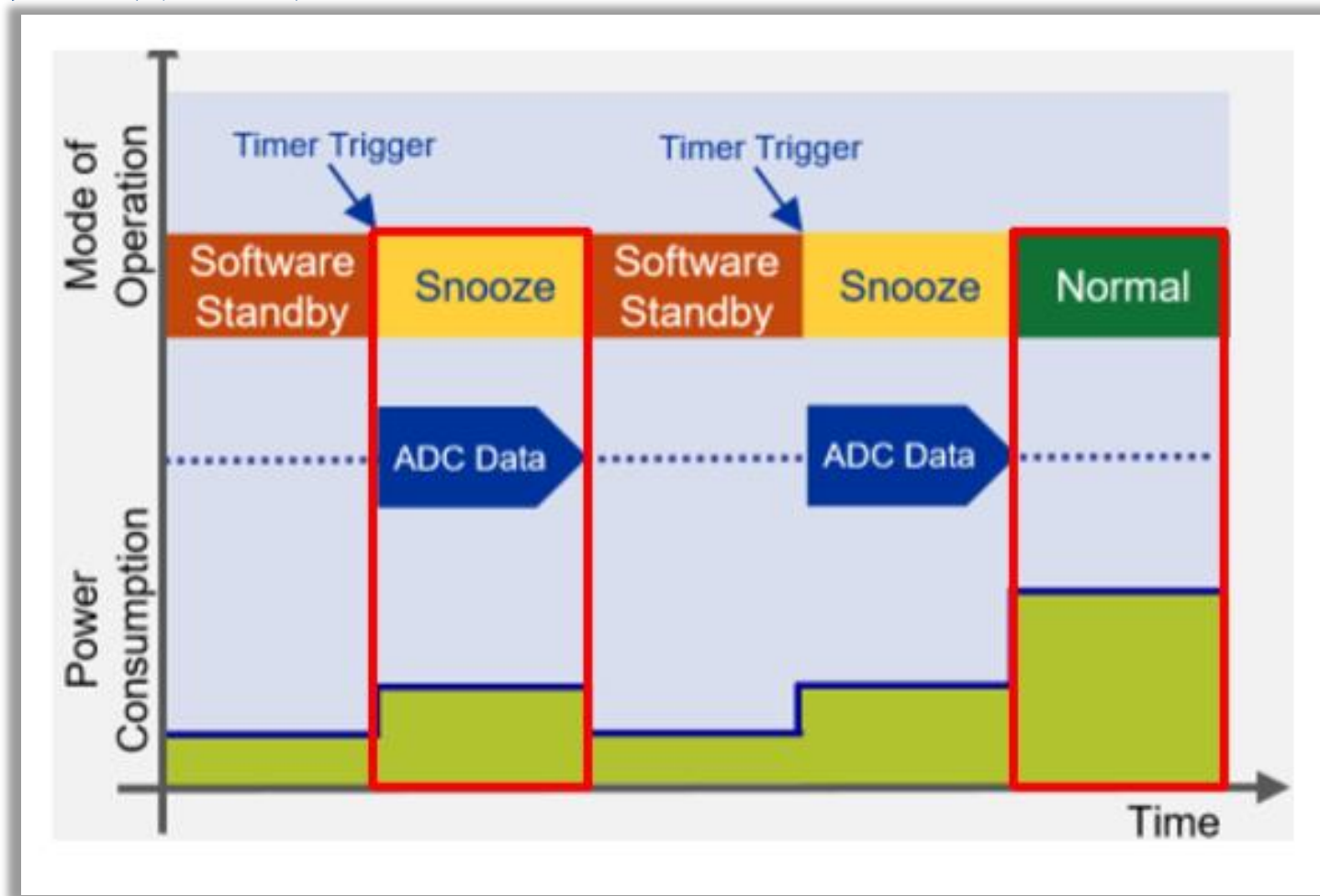
✓ STOP模式下的CTSUSU操作



CTSU低功耗应用的软件工作流程

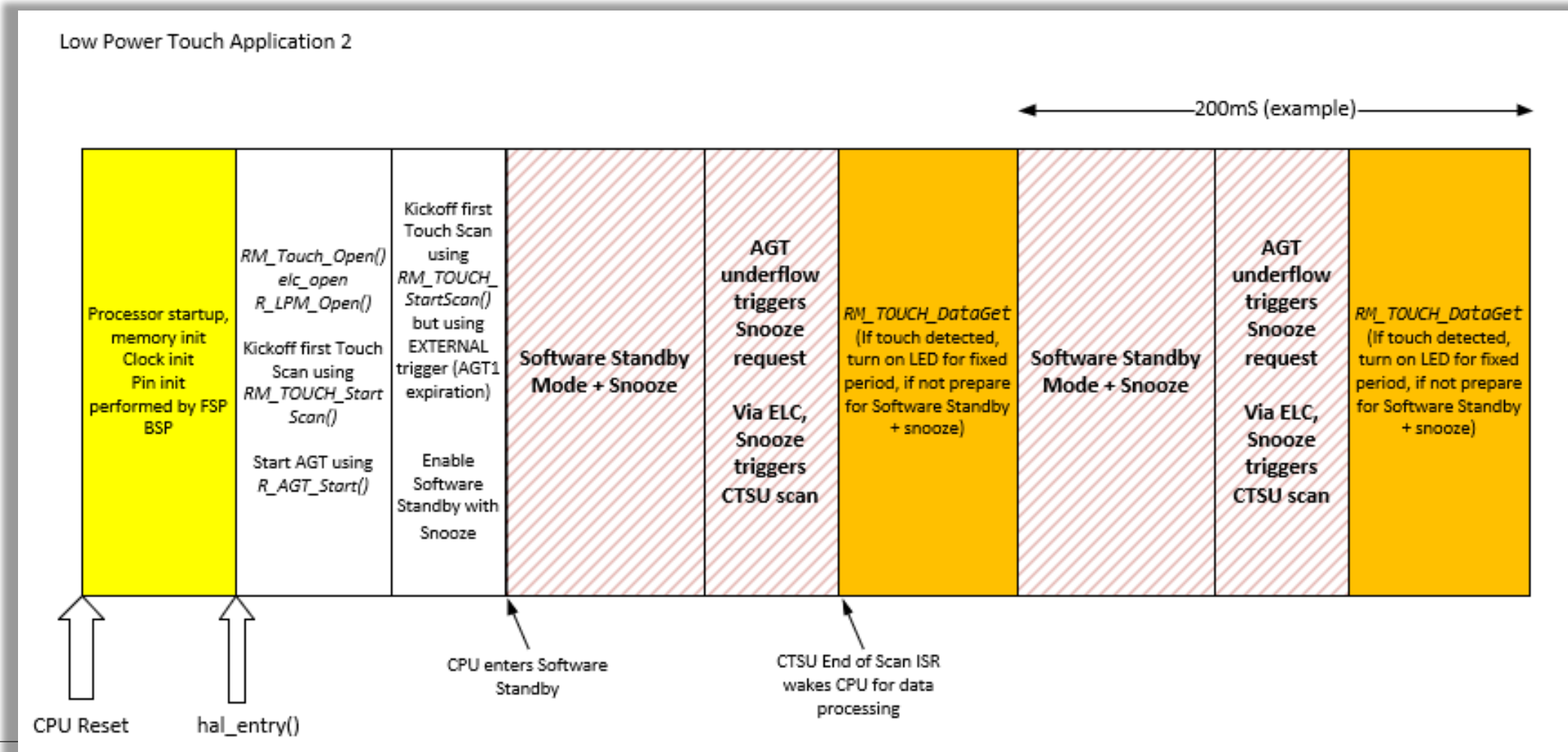
✓ Snooze简介

- Snooze功能提供了操作灵活性，可显著降低电流消耗
- 允许在保留CPU和其他外围设备的同时操作某些外围设备，例如：ADC、DAC、CTSU、SCI
- 可与某些设备上的软件待机功能集成，以进一步增强低功耗灵活性



CTSU低功耗应用的软件工作流程

✓ STOP + Snooze模式下的CTSU操作



CTSU低功耗应用的软件工作流程

✓ STOP + Snooze模式下的CTSU操作

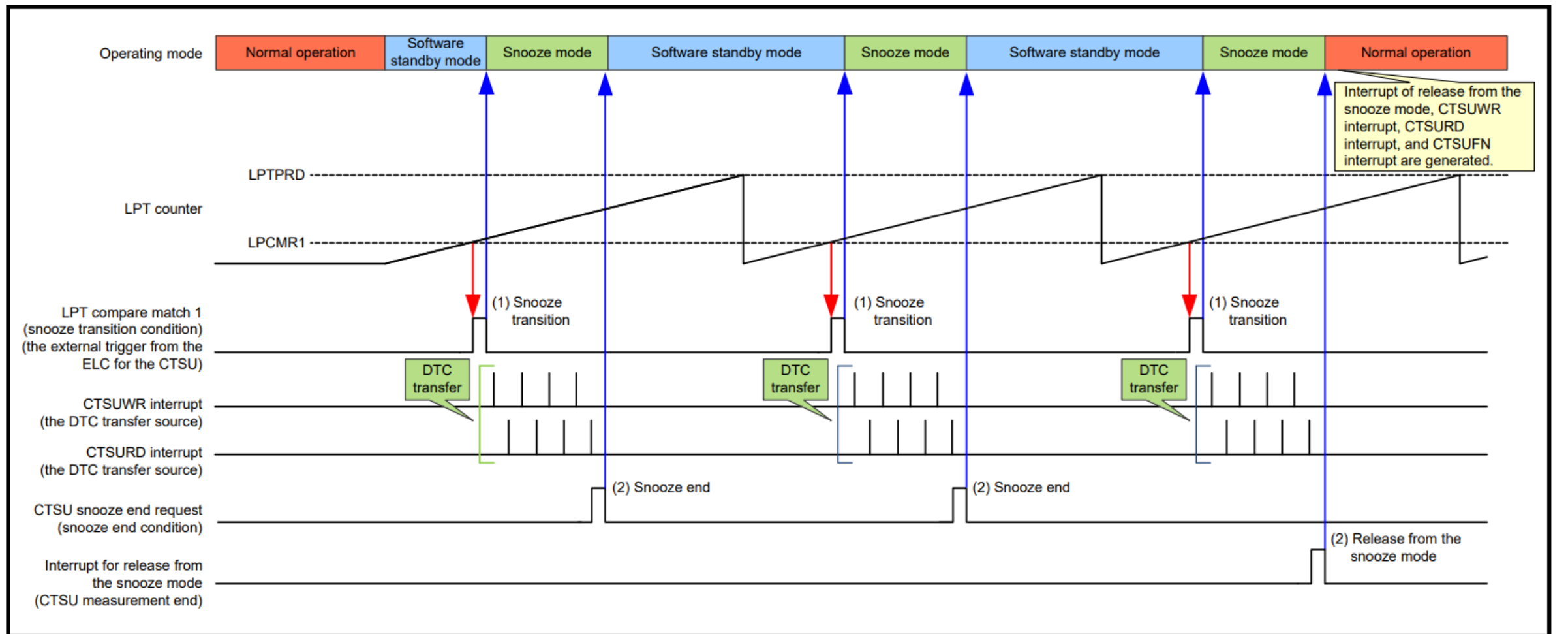
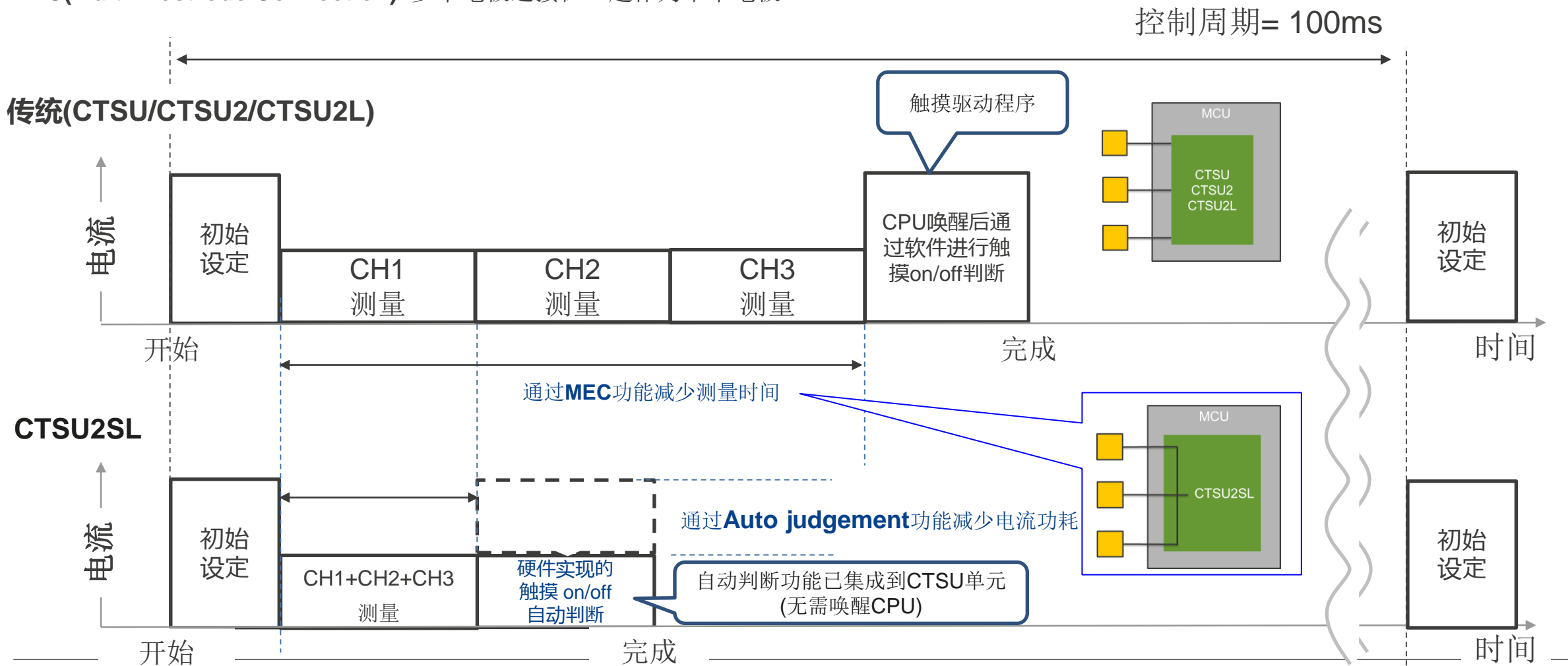


Figure 11.16 Timing of Measuring by the CTSU in the Snooze Mode

RX140的MEC+AJ新功能

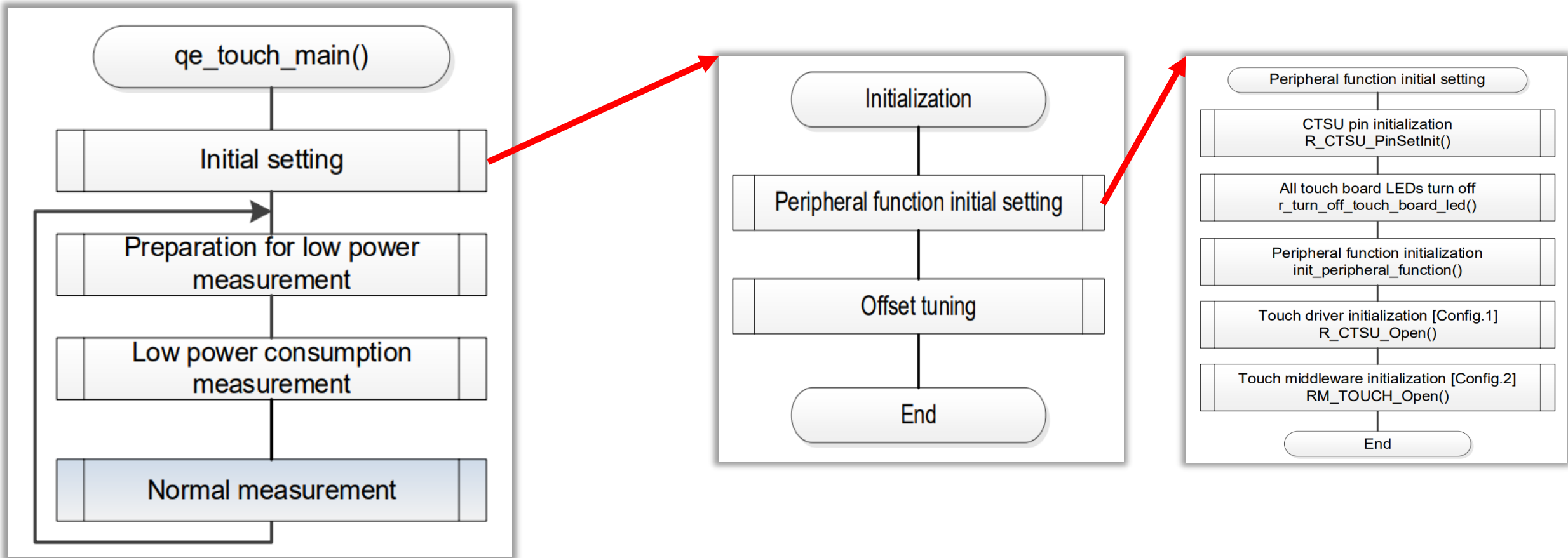
Auto judgment: 不需要启动CPU即可以进行Touch ON/OFF的自动判断(仅CTSUSL支持)

MEC(Multi Electrode Connection): 多个电极连接在一起作为单个电极



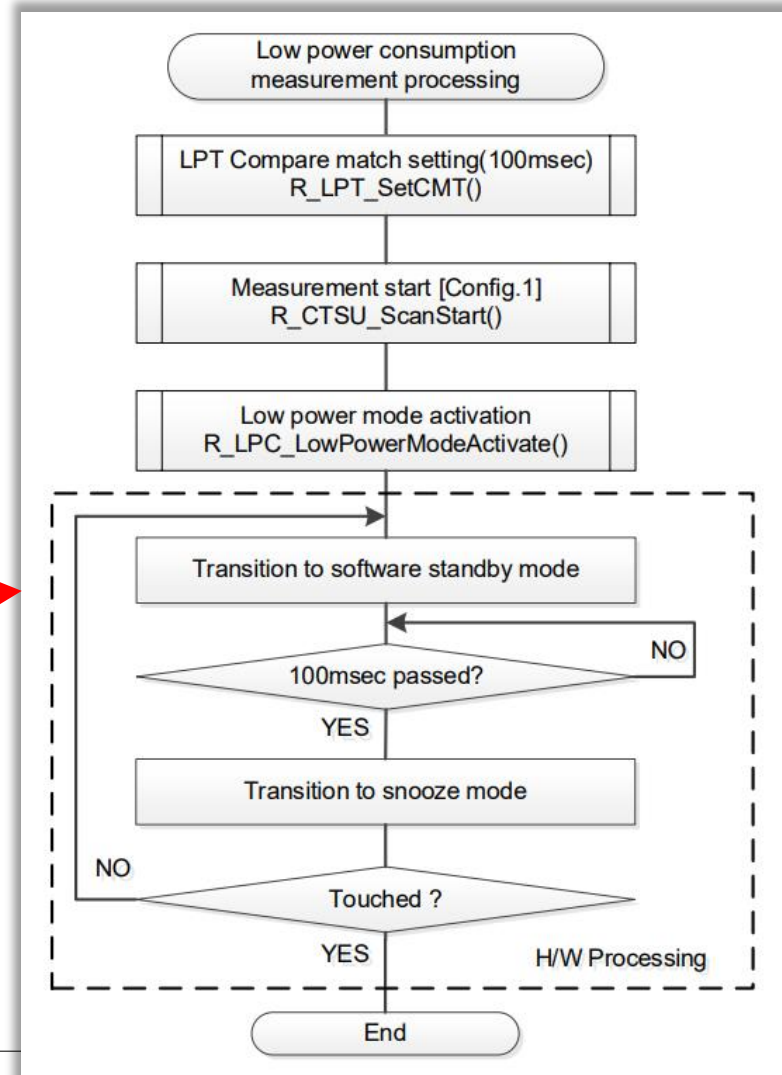
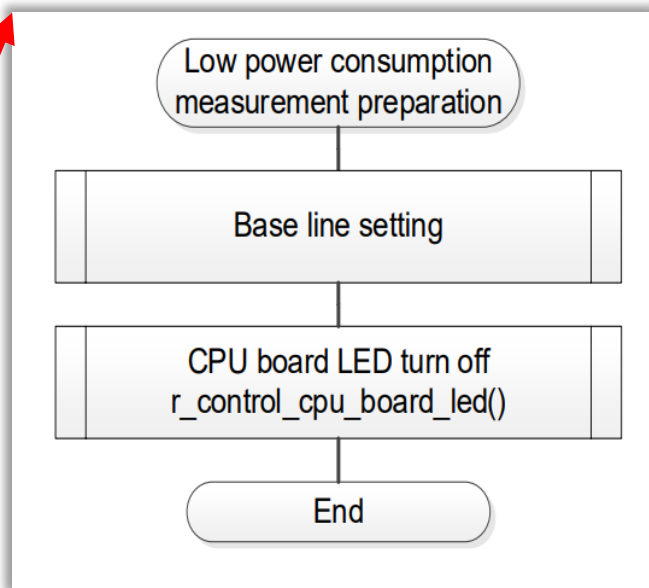
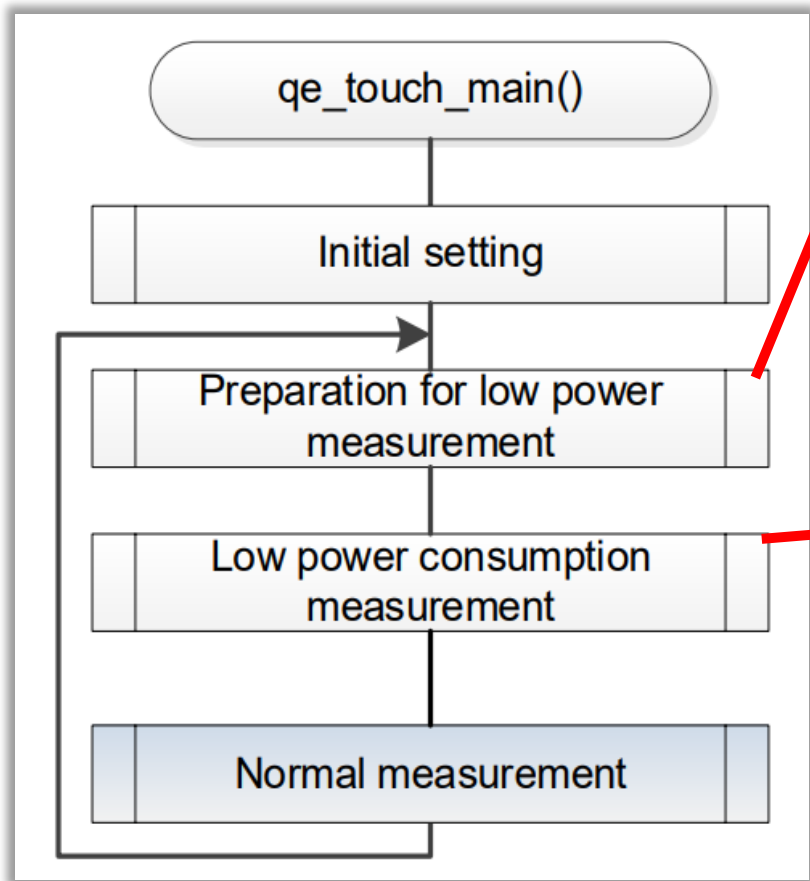
RX140的MEC+AJ新功能

RX140的CTSU低功耗应用软件工作流程
(使用MEC+AJ新功能)



RX140的MEC+AJ新功能

RX140的CTSU低功耗应用软件工作流程
(使用MEC+AJ新功能)



影响功耗的主要因素

- 1.MCU的片上振荡器
- 2.MCU的低功耗模式
- 3.CTSU的控制周期
- 4.CTSU的Sensor驱动脉冲频率
- 5.CTSU的MEC多电极连接+AJ自动判断功能

低功耗的测试设备以及测试方法



低功耗的测试设备以及测试方法

五. 低功耗的测试设备以及测试方法

1. 低功耗产品的直流功耗分析
2. 开发和分析低功耗产品时面临的挑战
3. 低功耗测试设备介绍：Keithley DMM7510
4. 低功耗测试示例

开发和分析低功耗产品时面临的挑战

◆ 在分析直流低功耗低功率设备时，有几个挑战，包括：

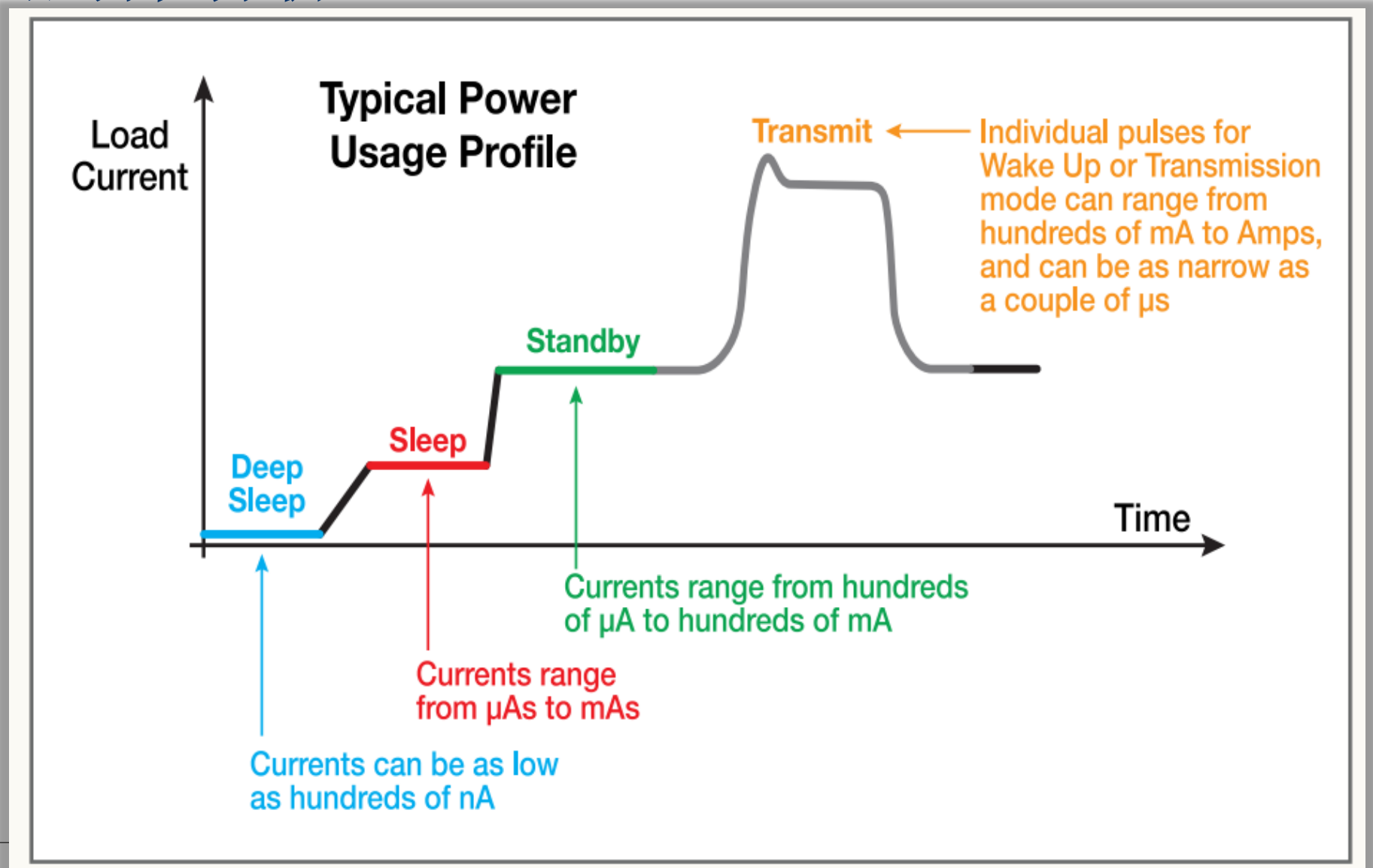
- ✓ 在动态大范围内实现高精度负载电流测量——低功耗产品的典型直流功耗曲线具有从微安到毫安，甚至到安培的大范围负载电流。大多数仪器并非在所有范围内都具有高精度/灵敏度。
- ✓ 当产品工作模式转换到活动状态(非低功耗模式)时，捕捉窄负载电流脉冲——大多数以单个脉冲或者脉冲群的形式出现，持续时间窄至微秒。大多数仪器没有捕获负载电流脉冲所需的高采样率。
- ✓ 大量的数据记录——这些测试通常运行很长时间，需要具有足够内存的测试仪器来支持长期监控。
- ✓ 测试仪器引入的误差——大多数仪器的较大内阻将误差引入系统，作为负载电压。由于低功耗产品的总功耗较低，系统中引入的高负载电压会导致较大的误差
- ✓ 清洁稳定的电源——可能会导致输出噪声错误。

◆ 在调试直流低功耗低功率设备时

- ✓ 不在于准确测试出各个低功耗阶段的电流，而在于要按照用户设计的低功耗工作模式以及时序，进行准确的模式切换。
- ✓ 要求具有足够高的分辨率，足够多的内存记录数据，可以图形的形式显示数据

低功耗产品的直流功耗分析

- ❑ 对低功耗设备（如物联网（IoT）、便携式无线设备、可穿戴设备、便携式和植入式医疗设备以及低功耗工业产品）而言，描述电源使用情况至关重要。
- ❑ 电力使用与电池消耗直接相关，并影响产品的实用性。
- ❑ 直流电源分析是一项非常重要的任务，因为这些设备通常会**根据工作模式从纳安、微安、毫安到安培的大范围内吸收电流，并且通常具有很短的唤醒时间，这可能只持续几微秒，并且需要大量的数据记录周期来捕获完整的电源使用情况。**



低功耗测试设备介绍

KEITHLEY DMM7510

☐ DMM 7510 7½数字多功能表

- ✓ 目前瑞萨官网关于低功耗测试的APPN均采用DMM7510

☐ 对于低功耗的

☐ 万用表功能

- ✓ DC电流

- ❖ 稳态电流

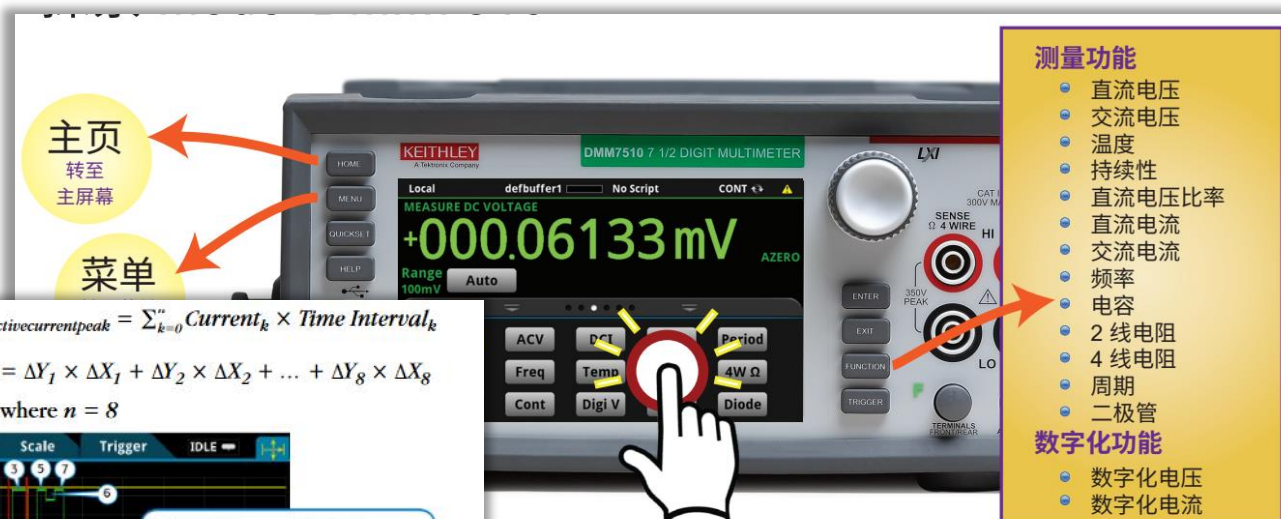
☐ 数字化功能

- ✓ 数字化电流

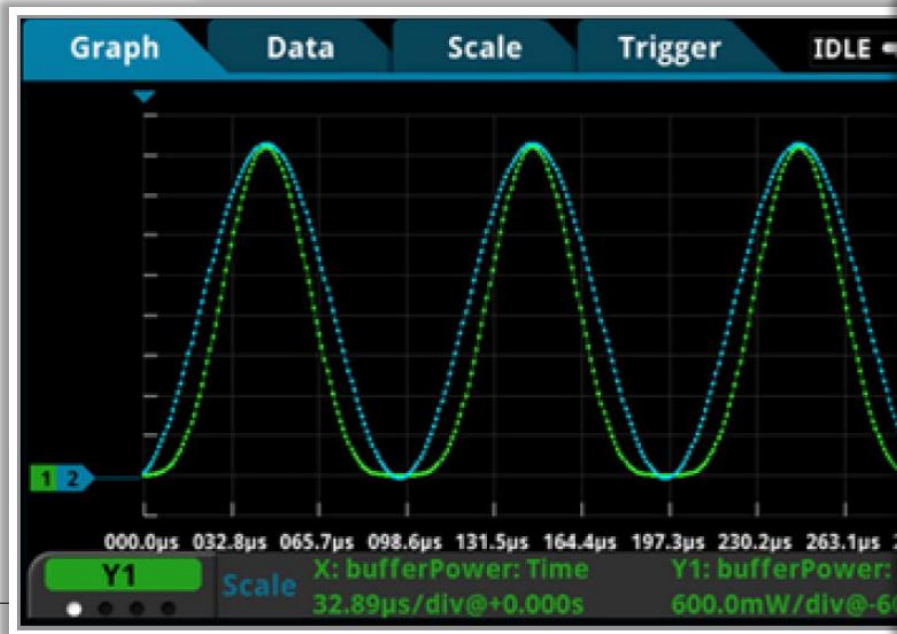
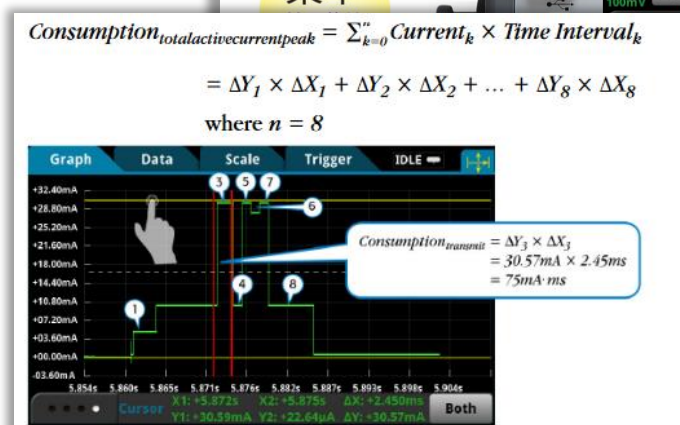
- ❖ 瞬态电流

- ❖ 1M点/秒

☐ 平均电流/功耗



- 测量功能**
- 直流电压
 - 交流电压
 - 温度
 - 持续性
 - 直流电压比率
 - 直流电流
 - 交流电流
 - 频率
 - 电容
 - 2线电阻
 - 4线电阻
 - 周期
 - 二极管
- 数字化功能**
- 数字化电压
 - 数字化电流



Item	Dmm1_Time(s)(1)(1)	Dmm1_DigitizerCurrent(1)(1)
1	0.00000000	0.000011446
2	0.000001000	0.000006525
3	0.000002000	0.000001998
4	0.000003000	-0.000004892
5	0.000004000	0.000001407
6	0.000005000	-0.000003809
7	0.000006000	-0.000000168
8	0.000007000	-0.000004203
9	0.000008000	0.000000718
10	0.000009000	-0.000001447
11	0.000010000	0.000000817
12	0.000011000	-0.000003022
13	0.000012000	-0.000002333
14	0.000013000	-0.000005876
15	0.000014000	-0.000007451

低功耗测试示例

❑ **测试示例1:** Software Standby mode模式下的电流

✓ Spec中的典型值为0.30uA($T_a = 25^\circ\text{C}$)

❑ 使用RA2L1 EK Board

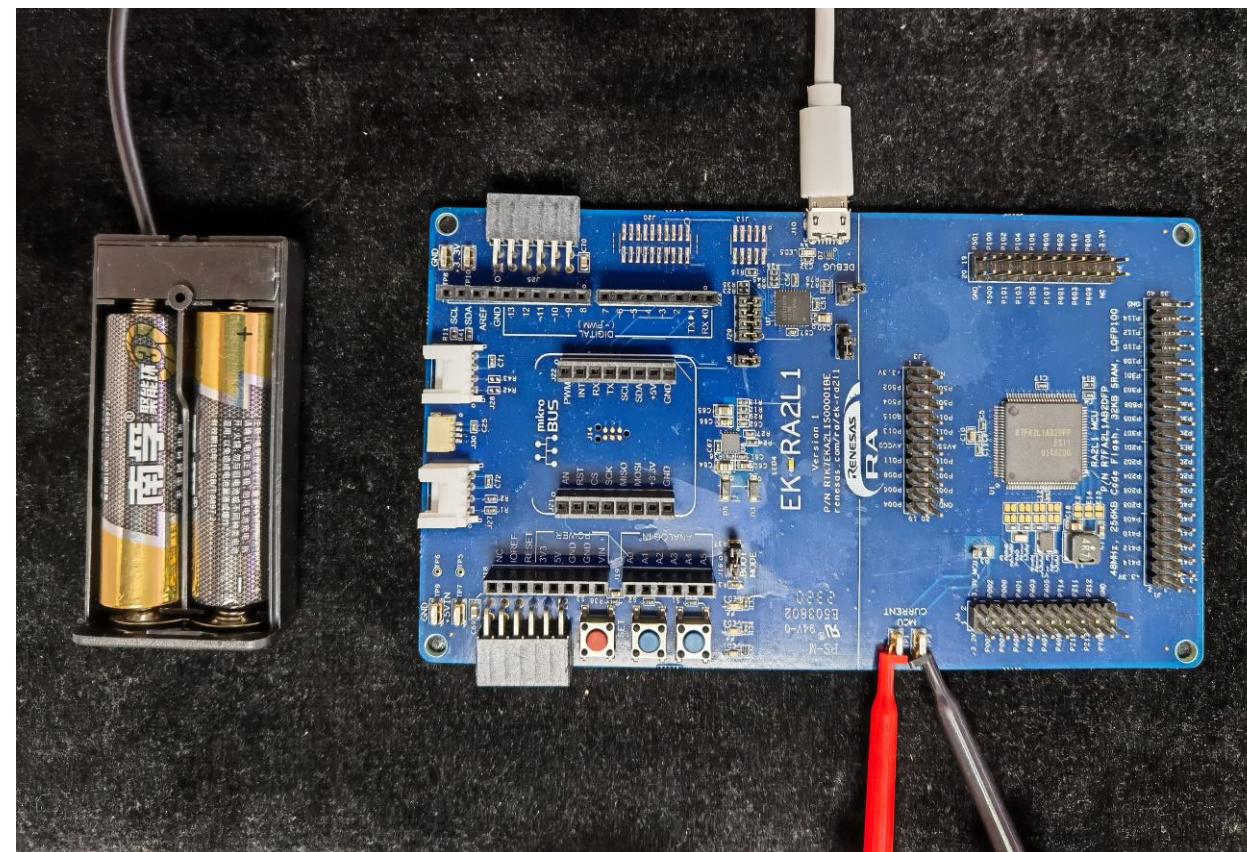
❑ 使用两节1.5V AAA电池供电

RA2L1 User's Manual 41. Electrical Characteristics

Table 41.11 Operating and standby current (2)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter	Symbol	Typ ³	Max	Unit	Test conditions	
Supply current ¹ Software Standby mode ²	All SRAMs(0x2000_4000 to 0x2000_7FFF) are on	$T_a = 25^\circ\text{C}$	0.30	2.2	μA	—
		$T_a = 55^\circ\text{C}$	0.65	5.3		
		$T_a = 85^\circ\text{C}$	2.0	20		
		$T_a = 105^\circ\text{C}$	4.0	70		
	Only 8KB SRAM (0x2000_4000 to 0x2000_5FFF) is on	$T_a = 25^\circ\text{C}$	0.25	2.2		
		$T_a = 55^\circ\text{C}$	0.6	5.3		
		$T_a = 85^\circ\text{C}$	1.8	20		
		$T_a = 105^\circ\text{C}$	3.65	70		



低功耗测试示例

- DMM 7510 : DC Current mode
- Measurement Settings in DMM7510
 - Function: DC current
 - Range: 10uA
 - NPLC: 0.0005 (Sample Rate: 24000点/s)
 - Display Digits: 7.5
- Acquisition setting
 - Measure Count: 10000
- Others
 - Auto Zero: On
 - Line Sync: Checked
 - Auto Delay: Checked

The screenshot displays the measurement settings for the DMM7510, divided into Primary Function and Secondary Function sections.

Primary Function Settings:

- Function: DC Current
- Range: 10μA
- NPLC: 0.0005
- Display Digits: 7.5
- Auto Zero: On
- Line Sync:
- Auto Delay:
- Rel: Rel Value: 0, Acquire Rel button
- Filter: Type: Repeat, Count: 10, Window (%): 0.1
- Math: Format: Percent, Reference: 1

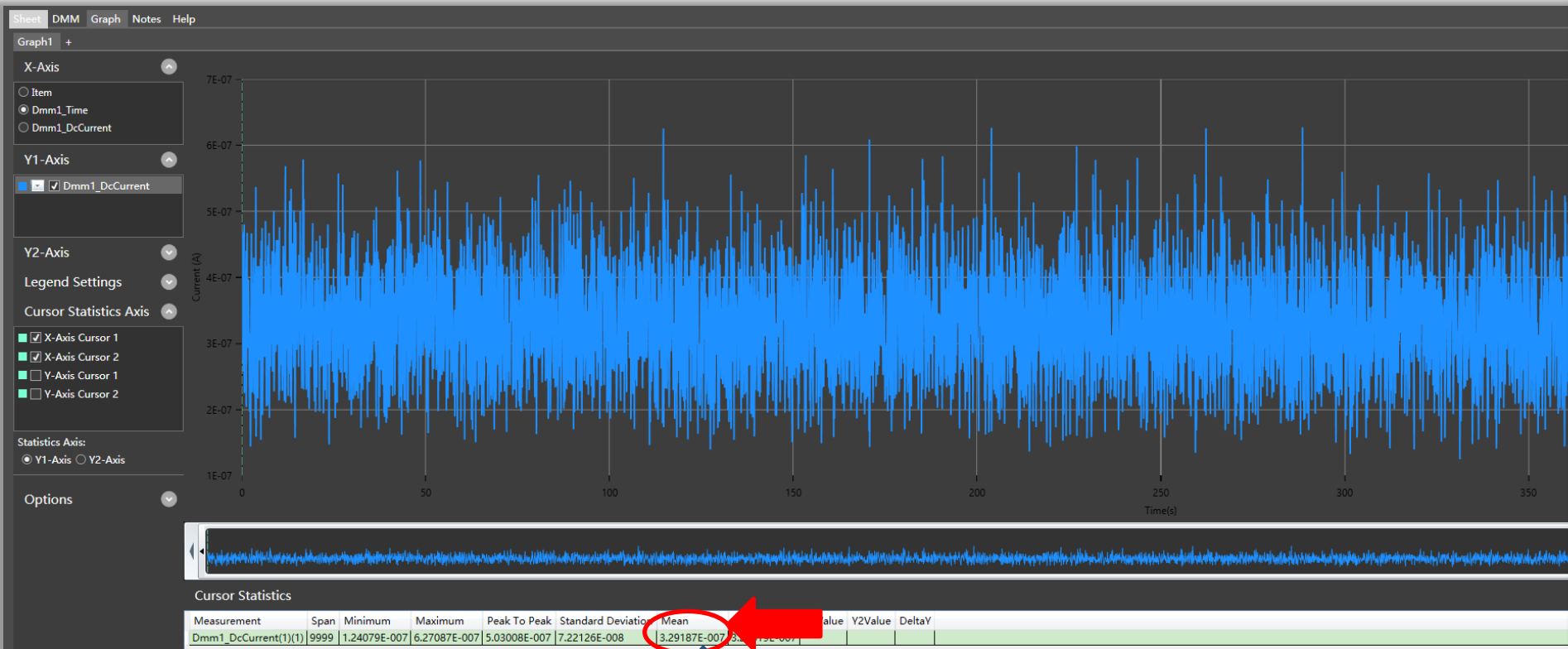
Secondary Function Settings:

- Trigger: Trigger Mode: Immediate
- Acquisition: Measure Delay (s): 0, Measure Count: 10000, Start at HH:MM: 2022/05/05 14:16:4, Timestamp Format: Relative
- Limit 1: Auto Clear: , Upper Limit: 1, Lower Limit: -1, Audible: None
- Limit 2: Auto Clear: , Upper Limit: 1, Lower Limit: -1, Audible: None

低功耗测试示例

■ 测试结果

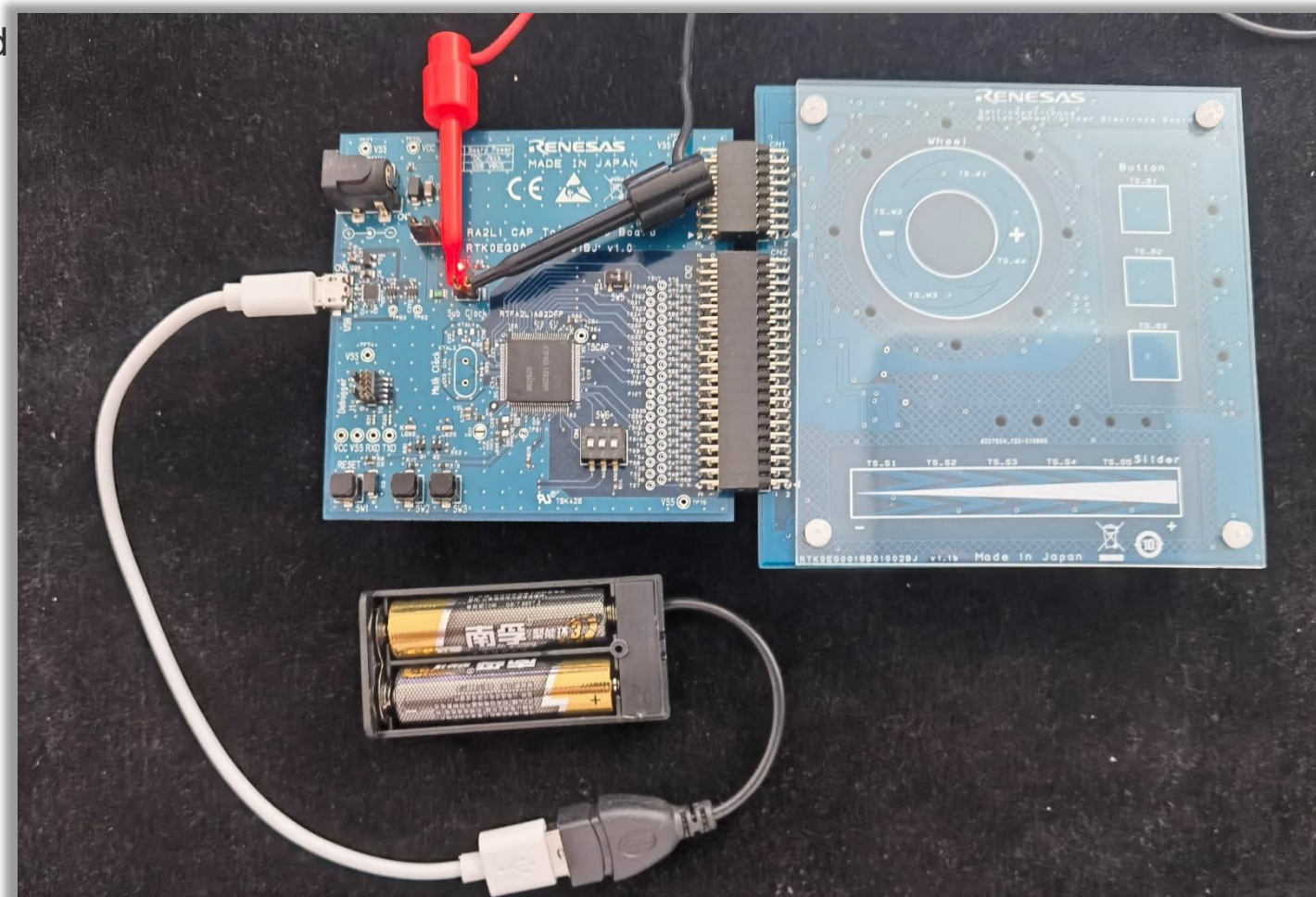
- 测试周期: 479.817s, 包含 10000 个测试数据
- 测试间隔: 4ms
- 平均电流: 0.329187uA



Exponential	Decimal	Decimal Display Precision
Item	Dmm1_Time(s)(1)(1)	Dmm1_DcCurrent(1)(1)
9969	478.3307844	0.0000003
9970	478.3707880	0.0000002
9971	478.4507989	0.0000003
9972	478.4908034	0.0000004
9973	478.5308067	0.0000003
9974	478.5708182	0.0000002
9975	478.6108229	0.0000005
9976	478.6908303	0.0000003
9977	478.7308368	0.0000003
9978	478.7708421	0.0000002
9979	478.8108485	0.0000003
9980	478.8508512	0.0000003
9981	478.9308657	0.0000004
9982	478.9708682	0.0000004
9983	479.0108737	0.0000003
9984	479.0508798	0.0000002
9985	479.0908842	0.0000003
9986	479.1708970	0.0000003
9987	479.2108994	0.0000002
9988	479.2509046	0.0000003
9989	479.2909083	0.0000003
9990	479.3309198	0.0000003
9991	479.4109276	0.0000002
9992	479.4509334	0.0000002
9993	479.4909396	0.0000002
9994	479.5309449	0.0000004
9995	479.5709455	0.0000004
9996	479.6509566	0.0000003
9997	479.6909659	0.0000003
9998	479.7309688	0.0000003
9999	479.7709734	0.0000003
10000	479.8109788	0.0000003

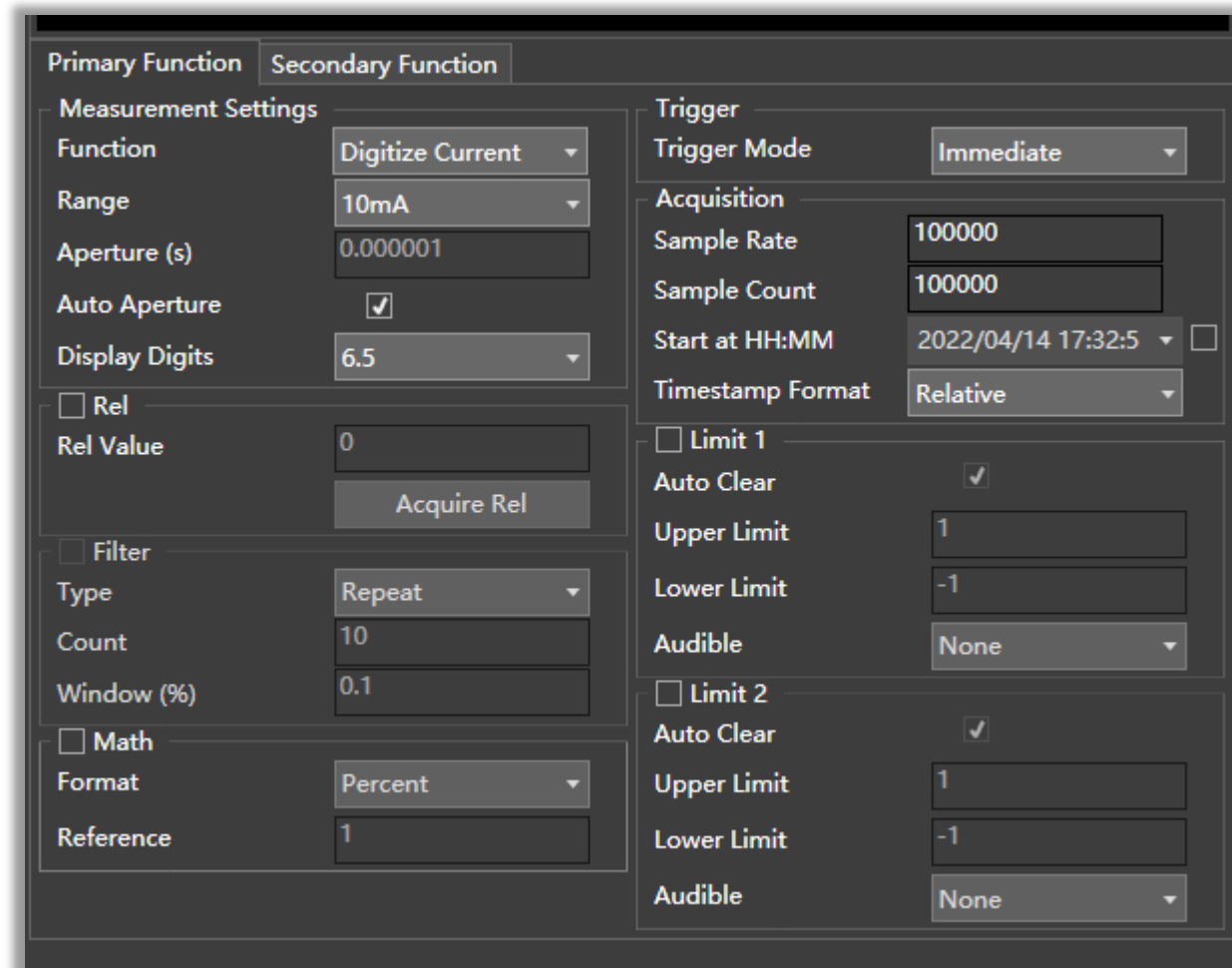
低功耗测试示例

- ❑ **测试示例2**: Software Standby mode + Snooze mode(CTSU 应用)的电流
- ❑ RA2L1 RSSK Board + Touch Application Board
- ❑ 使用两节1.5V AAA电池供电



低功耗测试示例

- DMM 7510 : Digitize Current mode
- Measurement Settings in DMM7510
 - Function: Digitize Current
 - Range: 10mA
 - Auto Aperture: checked
 - Display Digits: 6.5
- Acquisition setting
 - Sample Rate: 100000 (The interval is 10us)
 - Sample Count: 100000
- Others
 - Default



Primary Function Secondary Function

Measurement Settings

Function Digitize Current

Range 10mA

Aperture (s) 0.000001

Auto Aperture

Display Digits 6.5

Rel

Rel Value 0

Acquire Rel

Filter

Type Repeat

Count 10

Window (%) 0.1

Math

Format Percent

Reference 1

Trigger

Trigger Mode Immediate

Acquisition

Sample Rate 100000

Sample Count 100000

Start at HH:MM 2022/04/14 17:32:5

Timestamp Format Relative

Limit 1

Auto Clear

Upper Limit 1

Lower Limit -1

Audible None

Limit 2

Auto Clear

Upper Limit 1

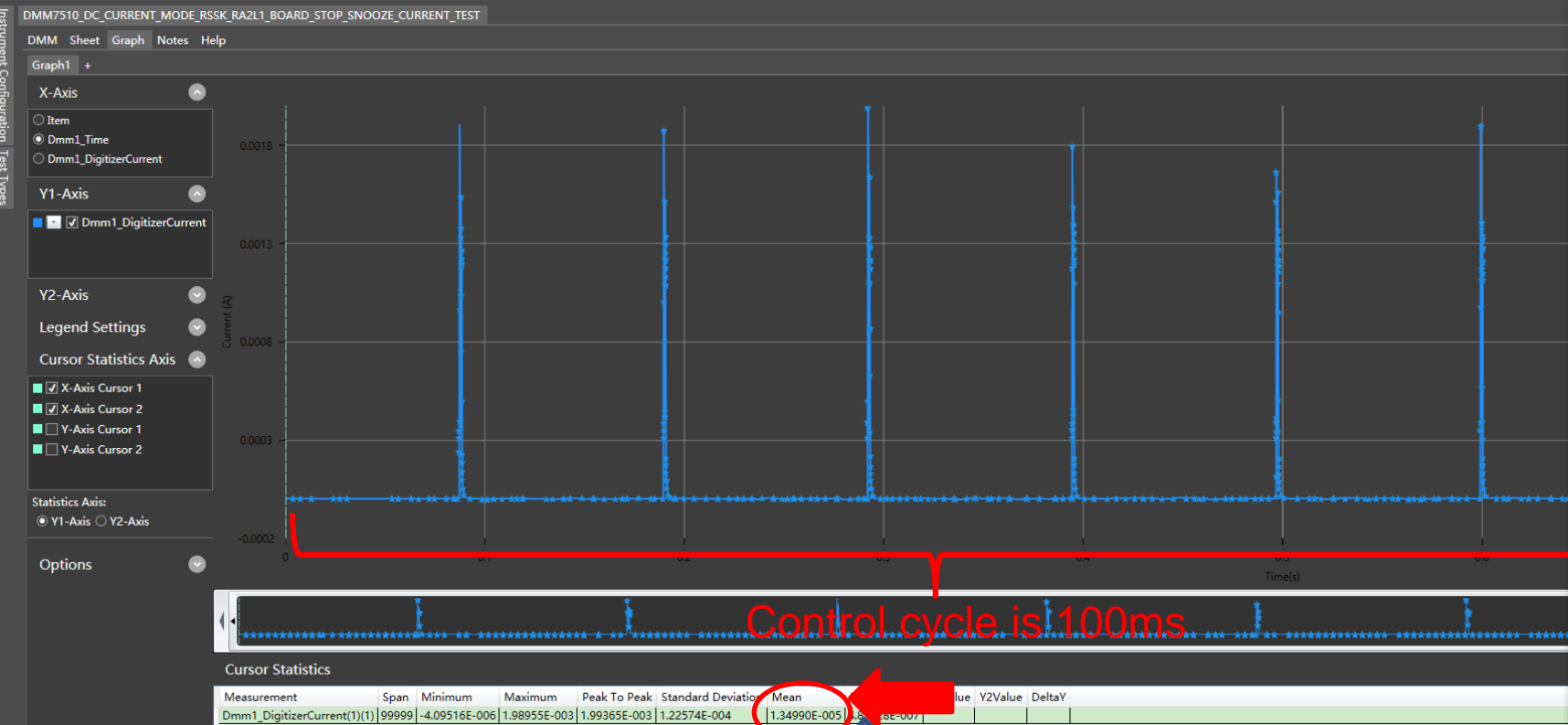
Lower Limit -1

Audible None

低功耗测试示例

测试结果

- 测试周期: 0.9999s, 包含 10000 个测试数据,
- 测试间隔: 10us
- 平均电流: 13.499uA



DMM7510_Digitize Current Mode_RL78G23_32pin_TS_1ch_STOP_SNOOZE

DMM Sheet Graph Notes Help

Exponential Decimal Decimal Display Precision: 7

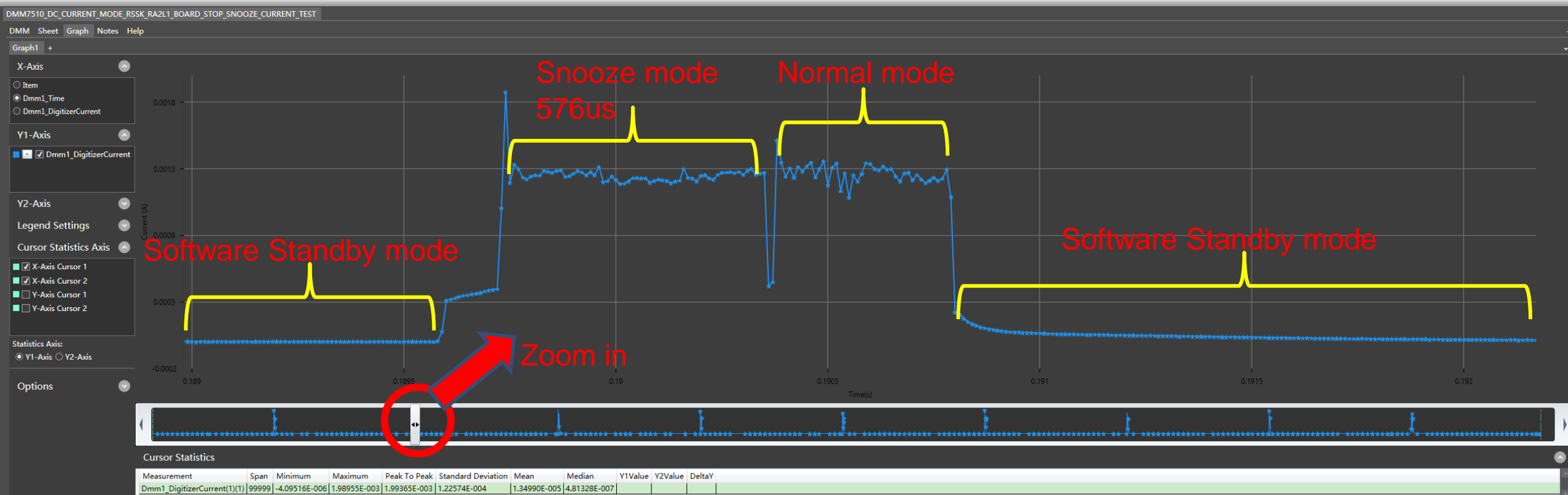
Item	Dmm1_Time(s)(1)(1)	Dmm1_DigitizerCurrent(1)(1)
99969	0.9996808	0.0000005
99970	0.9996908	0.0000010
99971	0.9997008	0.0000005
99972	0.9997108	0.0000006
99973	0.9997208	0.0000004
99974	0.9997308	0.0000001
99975	0.9997408	0.0000006
99976	0.9997508	0.0000002
99977	0.9997608	0.0000002
99978	0.9997708	0.0000001
99979	0.9997808	0.0000001
99980	0.9997908	0.0000018
99981	0.9998008	-0.0000005
99982	0.9998108	0.0000003
99983	0.9998208	0.0000004
99984	0.9998308	0.0000000
99985	0.9998408	0.0000014
99986	0.9998508	-0.0000010
99987	0.9998608	0.0000005
99988	0.9998708	0.0000004
99989	0.9998808	0.0000002
99990	0.9998908	0.0000004
99991	0.9999008	0.0000004
99992	0.9999108	0.0000002
99993	0.9999208	0.0000016
99994	0.9999308	-0.0000005
99995	0.9999408	0.0000003
99996	0.9999508	-0.0000001
99997	0.9999608	-0.0000001
99998	0.9999708	0.0000005
99999	0.9999808	0.0000005
10000	0.9999908	0.0000002

低功耗测试示例

■ 测试结果

■ 放大

■ CTSU2的一个TS通道的典型测量时间为576us



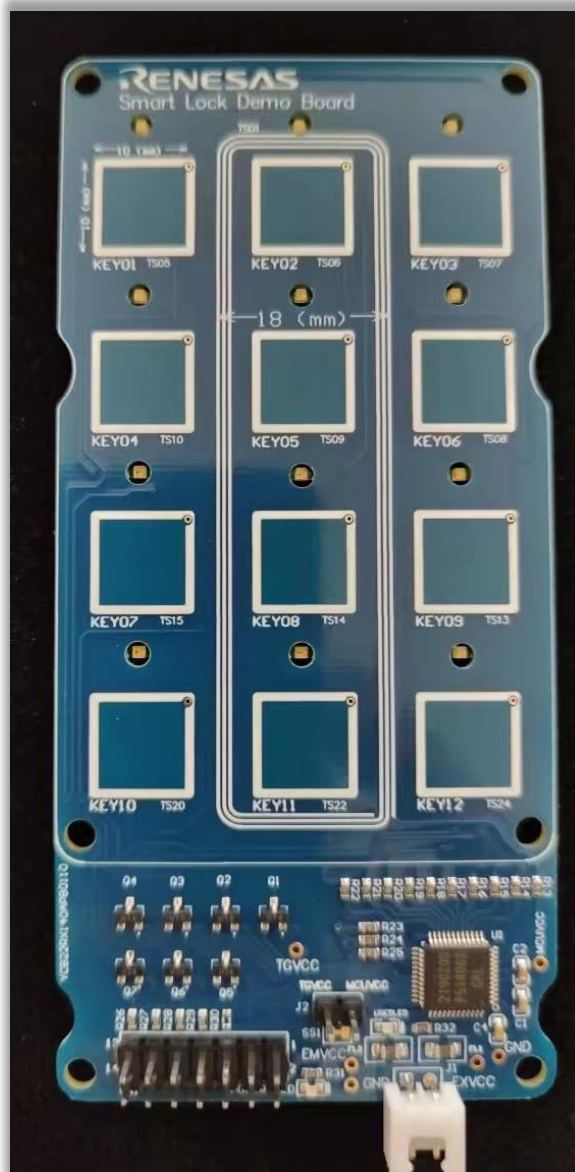
LAB SESSION



RX140低功耗触摸评估板套件

■ RX140低功耗触摸评估板套件

- 评估板 x 1
- Ez-cube2仿真器 x 1
- 亚克力板: 1mm厚度 x1 , 2mm厚度 x1
- 塑料柱、塑料螺丝若干
- 电池盒 x 1
- 1.5V AAA电池 x 2



评估板



Ez-cube2仿真器及说明书



1mm厚度亚克力



2mm厚度亚克力



电池盒及电池



塑料柱、塑料螺丝

LAB SESSION

■ **LAB Session**将帮助用户了解和掌握以下内容

- 从基本的触摸应用工程创建、使用**QE for Cap touch**监控触摸数据和调试运行参数开始，循序渐进的增加并调试以下功能
 - ✓ 12个按键矩阵功能(自容式)
 - ✓ **RX140 MEC**多电极连接功能
 - ✓ 低功耗功能(**RX140 Auto Judgement**功能)
 - ✓ 接近传感功能(改变**MEC**灵敏度)
 - ✓ 低功耗数据的测试

LAB SESSION

■ LAB Session将帮助用户了解和掌握以下内容

❖ 1. 触摸工程的创建

- ✓ 设定触摸接口
 - 触摸通道的分组(Configuration)
 - MEC功能使能
 - Auto Judgement功能使能
- ✓ 通过Smart Configurator添加驱动程序
 - R_CTSU, RM_Touch
 - LPC, LPT, ELC, DTC, PORT
- ✓ 触摸运行参数的修改
 - 常用运行参数
- ✓ 添加应用程序
 - 触摸应用程序
 - LED驱动程序

❖ 2. 触摸应用的监控和调试

- ✓ 使用QE监测触摸底层数据
 - 各个QE调试窗口的使用
- ✓ 使用QE调试触摸相关参数
 - 通用运行参数
 - MEC运行参数
 - 低功耗(Auto Judgement)运行参数

❖ 3. 触摸应用软件开发

- 通用的触摸功能应用软件
- MEC功能应用软件
- 低功耗(Auto Judgement)功能软件

❖ 4. 接近传感功能的实现方法

- 使用MEC功能增加接近传感功能(改变灵敏度)
- 使用线圈形状的电极增加接近传感功能(设计专用的电极)

❖ 5. 使用DMM7510测量功耗

LAB SESSION

□ 开发环境准备

- ✓ e2 studio 2023 04
- ✓ RX FIT Drivers
- ✓ Lab sessions
- ✓ QE for Cap Touch

```
+--- e22023_04
|   +--- setup_e2_studio_2023-04.zip
+--- FIT_drivers
|   +--- rm_touch_qe_v2.10.xml
|   +--- rm_touch_qe_v2.10.zip
|   +--- rm_touch_qe_v2.10_extend.mdf
|   +--- r_ctsu_qe_v2.10.xml
|   +--- r_ctsu_qe_v2.10.zip
|   +--- r_ctsu_qe_v2.10_extend.mdf
|   +--- r_dtc_rx_v4.10.xml
|   +--- r_dtc_rx_v4.10.zip
|   +--- r_dtc_rx_v4.10_extend.mdf
|   +--- r_elc_rx_v2.01.xml
|   +--- r_elc_rx_v2.01.zip
|   +--- r_elc_rx_v2.01_extend.mdf
|   +--- r_lpc_rx_v2.04.xml
|   +--- r_lpc_rx_v2.04.zip
|   +--- r_lpc_rx_v2.04_extend.mdf
|   +--- r_lpt_rx_v3.01.xml
|   +--- r_lpt_rx_v3.01.zip
|   +--- r_lpt_rx_v3.01_extend.mdf
+--- lab_sessions
|   +--- labsessions.zip
+--- QE_for_captouch
|   +--- RenesasQE_cap-touch_V320.zip
```


LAB SESSION

e2 studio version 2023-04

QE for Cap Touch 3.2.0

Current Configuration

Selected board/device: R5F51406BxFL (ROM size: 256KB, RAM size: 64KB, Pin count: 48)

Generated location (PROJECT_LOC\):

Selected components:

Component	Version	Configuration
<input checked="" type="checkbox"/> Board Support Packages. (r_bsp)	7.40	r_bsp(used)

Renesas e² studio 2023-04 Setup

Select the device families you wish to install support for

RENEASAS

Welcome

Device Families

- RA** Build, Debug & Code Generation support for Renesas RA device
- RZ** Build, Debug & Code Generation support for Renesas RZ device
- RL78** Build, Debug & Code Generation support for Renesas RL78 device
- RX** Build, Debug & Code Generation support for Renesas RX device
- RH850** Debug support for Renesas RH850 devices
- RE** Build & Debug support for Renesas RE devices
- Synergy** Build, Debug & Code Generation support for Renesas Synergy devices
- DA** Build & Debug support for Renesas DA devices

Select All

v202304061529 User: All Users < Back Next > Install Cancel

Renesas e² studio 2023-04 Setup

Select the additional software you wish to install

RENEASAS

Welcome

Device Families

Extra Features

Customise Features

Additional Software

Licenses

Shortcuts

Summary

Installing...

Results

Renesas QE (1) Renesas Toolchains & Ut... GCC Toolchains & Utiliti... Renesas FSP (1)

QE for AFE[RA] 2.0.0
Tools to assist in development work involving AFE.

QE for Motor 1.1.0
Tools to assist in the configuration of motor middleware and related drivers.

QE for Capacitive Touch 3.2.0
Tools to assist in the configuration, tuning, and monitoring of touch interfaces. (supports all touch MCUs: RX, RA, RL78, and Renesas Synergy)

QE for BLE[RA,RE,RX] 1.6.0
Tools to assist in development work involving Bluetooth® Low Energy.

QE for Display[RX,RA] 3.1.1
Tools to assist with timing settings or the image quality of displays. (for use with display controllers from Renesas)

QE for OTA (Technical Preview Edition) 1.0.0
Tools to assist in development work involving OTA update via Cloud.

406.2 MB download required

v202304061529 User: All Users < Back Next > Install Cancel

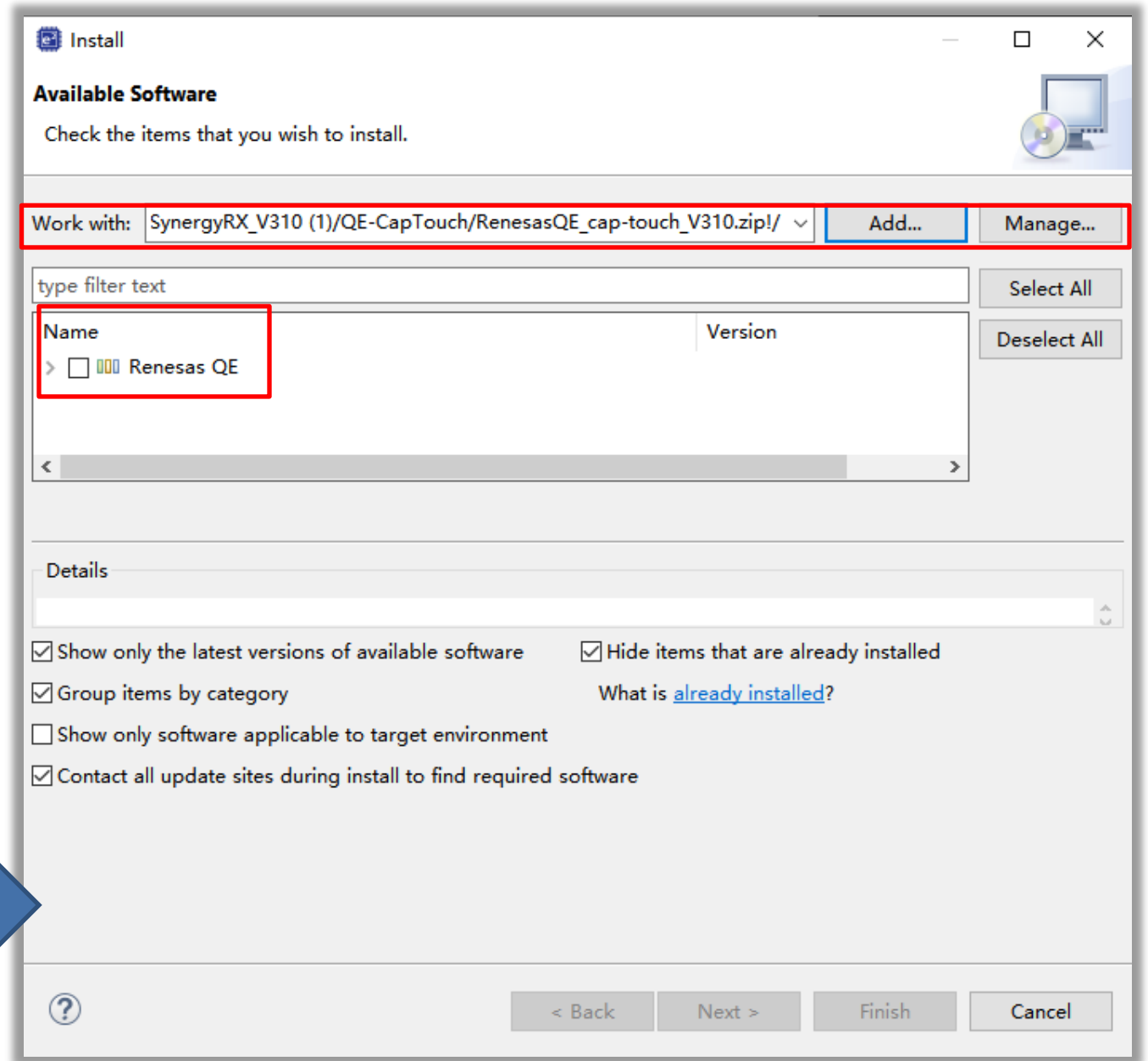
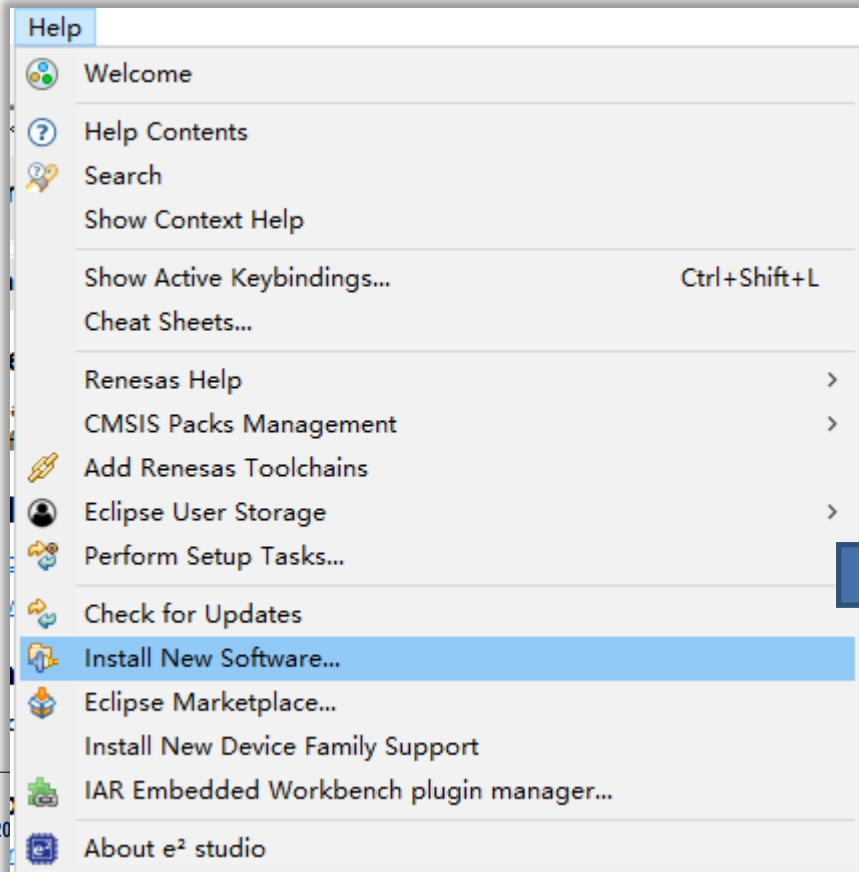
LAB SESSION

❑ e2 studio version 2023-04

❑ QE for Cap Touch 3.2.0

✓ 单独安装

e2 studio → [Help] → [Install New Software...]



LAB SESSION

RX FIT Driver

Overview information

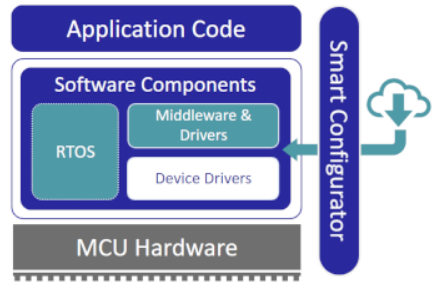
General Information

Overview
Get an [overview](#) of the features provided by Smart Configurator.

Videos
[Introduction to Smart Configurator](#)
[Browse related videos](#)

What's New
Check out [what's new](#) in the latest release.

Product Documentation
[User manual and release notes](#)
[Application Notes](#)
[Tool news](#)



Current Configuration
Selected board/device: R5F51406BxFL (ROM size: 256KB, RAM size: 64KB, Pin count: 48)
Generated location (PROJECT_LOC):

Selected components:

Component	Version	Configuration
Board Support Packages. (r_bsp)	7.40	r_bsp(used)
Control Low Power States. (r_lpc_rx)	2.04	r_lpc_rx(used)
CTSU QE API (r_cts_uqe)	2.10	r_cts_uqe(used)
DTC driver (r_dtc_rx)	4.10	r_dtc_rx(used)
ELC Driver. (r_elc_rx)	2.01	r_elc_rx(used)
Low-Power Timer Driver. (r_lpt_rx)	3.01	r_lpt_rx(used)
Ports	2.4.1	Config_PORT(PORT: used)
Touch QE API (rm_touch_qe)	2.10	rm_touch_qe(used)

Overview Board Clocks System Components Pins Interrupts

Lab_session_... X qe_touch_c...

Software component config

Components

type filter text

- Startup
 - Generic
 - r_bsp
- Drivers
 - DMA
 - r_dtc_rx
 - Event link controller
 - r_elc_rx
 - Capacitive Touch
 - r_cts_uqe
 - I/O Ports
 - Config_PORT
 - Low power consumption
 - r_lpc_rx
 - Timers
 - r_lpt_rx
 - Middleware
 - Capacitive Touch
 - rm_touch_qe

Overview Board Clocks System Co

New Component

Software Component Selection

Select component from those available in list

Category: All
Function: All
Filter:

Components	Short Name	Type	Versi...
8-Bit Timer		Code Generator	1.10.0
Board Support Packages.	r_bsp	Firmware Inte...	7.40
Buses		Code Generator	1.11.0
Clock Frequency Accuracy Measure...		Code Generator	1.11.0
Comparator		Code Generator	1.9.0
Compare Match Timer		Code Generator	2.3.0
Complementary PWM Mode Timer		Code Generator	1.11.0
Continuous Scan Mode S12AD		Code Generator	1.13.0
Control Low Power States.	r_lpc_rx	Firmware Inte...	2.04
CRC Calculator		Code Generator	1.11.0
CTSU QE API	r_cts_uqe	Firmware Inte...	2.10
Data Operation Circuit		Code Generator	1.11.0
Data Transfer Controller		Code Generator	1.11.0
Dead-time Compensation Counter		Code Generator	1.11.0
DTC driver	r_dtc_rx	Firmware Inte...	4.10

Show only latest version
 Hide items that have duplicated functionality

Description

This software component generates two units (unit 0, unit 1) of an on-chip 8-bit timer (TMR) module that comprise two 8-bit counter channels, totaling four channels.

[Download the latest FIT drivers and middleware](#)
[Configure general settings...](#)

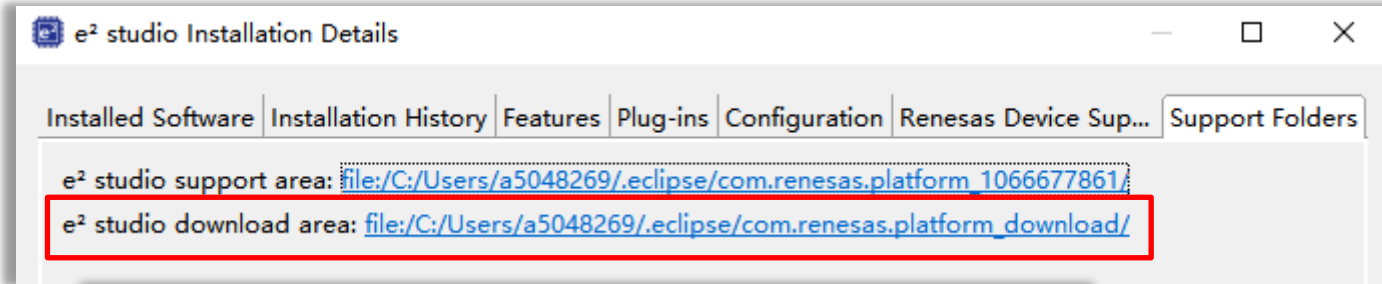
Overview Board Clocks System Co

< Back Next > Finish Cancel

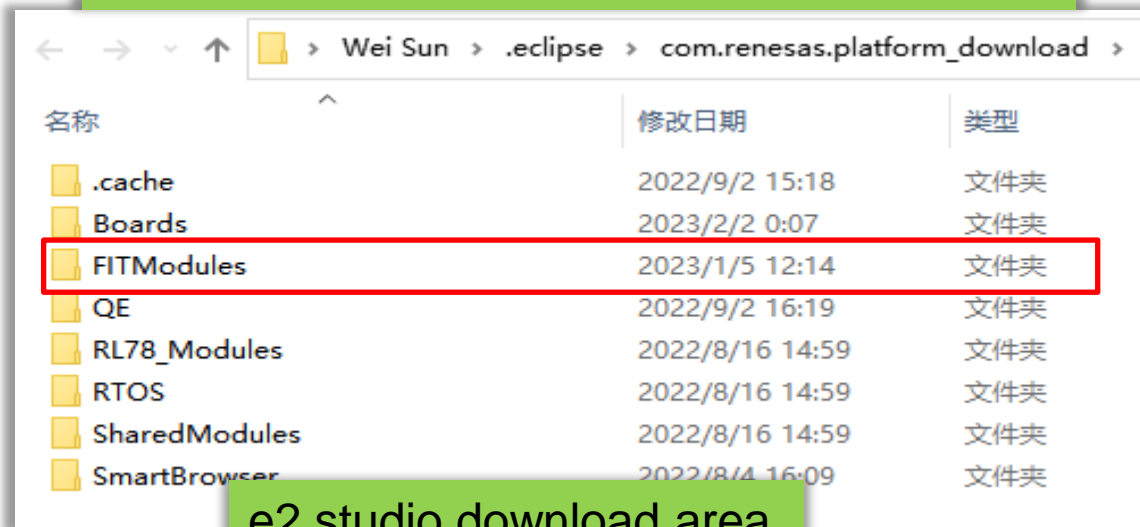
LAB SESSION

❑ RX FIT Driver

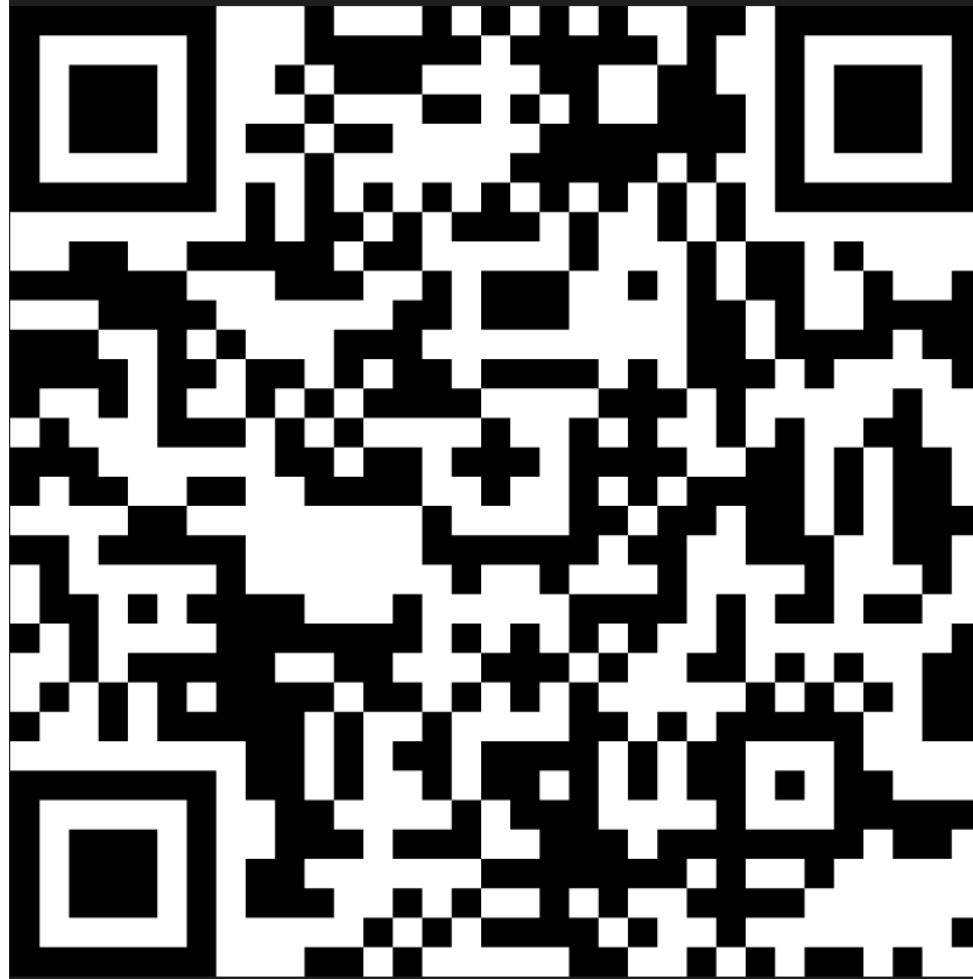
将驱动文件直接Copy到“ Support Folders ”的“ download area ”路径下的“**FITModules**”文件夹



e2 studio → [Help] → [about e2 studio] → [Installation Details]



问卷调查



[Renesas.com](https://www.renesas.com)

Thank you