

SDM670 传感器概述



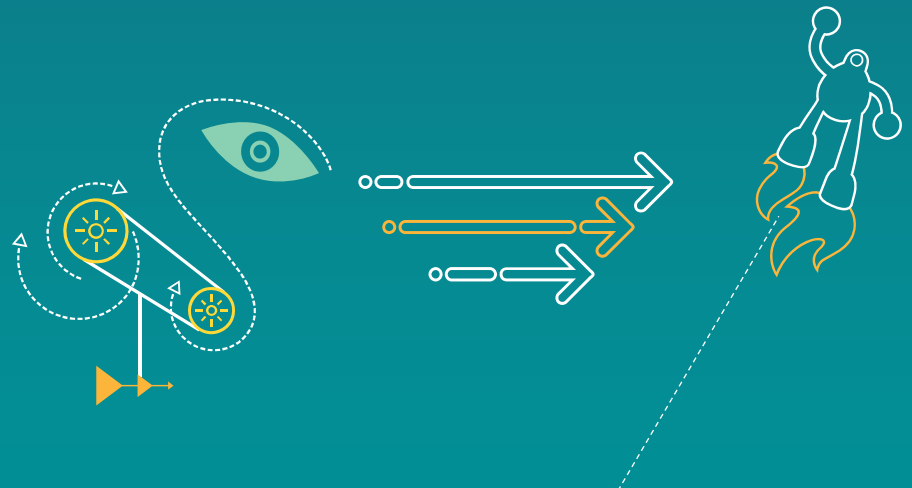
Qualcomm Technologies, Inc.

80-PD126-9SC 版本 A

机密和专有信息 – Qualcomm Technologies, Inc.

禁止公开披露：如若发现本文档在公共服务器或网站上发布，请报告至：
DocCtrlAgent@qualcomm.com。

限制分发：未经 Qualcomm 配置管理部门的明确批准，不得向 Qualcomm Technologies, Inc. 或其关联公司的员工之外的任何人分发。





机密和专有信息 – Qualcomm Technologies, Inc.

禁止公开披露：如若发现本文档在公共服务器或网站上发布，请报告至：DocCtrlAgent@qualcomm.com。

限制分发：未经 Qualcomm 配置管理部门的明确批准，不得向 Qualcomm Technologies, Inc. 或其关联公司的员工之外的任何人分发。

未经 Qualcomm Technologies, Inc. 的明确书面许可，不得使用、复印、复制或修改其全部或部分内容，或以任何方式向其他人泄露其内容。

Qualcomm Hexagon、Qualcomm ChipCode、MSM、QuRT、Qualcomm All-Ways Aware 和 QXDM Professional 是 Qualcomm Technologies, Inc. 的产品。作为 Qualcomm Snapdragon 处理器的功能，Qualcomm improveTouch 技术是 Qualcomm Technologies, Inc. 的产品。本文提到的其他 Qualcomm 产品是 Qualcomm Technologies, Inc. 或其子公司的产品。

Qualcomm、Hexagon、improveTouch、MSM、Qualcomm All-Ways Aware 和 QXDM Professional 是 Qualcomm Incorporated 在美国及其他国家/地区所注册的商标。QuRT、Qualcomm ChipCode 和 Qualcomm All-Ways Aware 是 Qualcomm Incorporated 的商标。其他产品和品牌名称可能是其各自所有者的商标或注册商标。

本技术资料可能受美国和国际出口、再出口或转让（统称“出口”）法律的约束。严禁违反美国和国际法律。

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

© 2017 Qualcomm Technologies, Inc. 和/或其关联公司。保留所有权利。

修订记录

版本	日期	说明
A	2017 年 9 月	初始版本

Qualcomm

2018-07-29 20:40:12 PDT

songpeng2@huaqin.com

议程

文档结构图	<u>5</u>
系统架构	<u>7</u>
产品特性	<u>13</u>
设计指南	<u>15</u>
软件概述	<u>20</u>
迁移摘要	<u>31</u>
软件定制	<u>34</u>
SDM670 传感器代码结构	<u>37</u>
ADSP 工具链和编译	<u>41</u>
调试概述	<u>45</u>
传感器供应商生态系统	<u>47</u>
支持	<u>50</u>
参考资料	<u>52</u>



文档结构图

OEM 项目生命周期

客户项目阶段的文档结构图

评估 (QTI 功能)

- 审核支持的传感器功能 (80-NH058-1)。
- 参见发布说明和本文档。

定义 (客户产品要求)

- 审核传感器部件/厂商信息、驱动程序支持 (80-NB925-2)，分析要求，并与厂商交互。
- 完成客户的驱动程序/功能支持计划，包括需要从传感器厂商获取的驱动程序支持 (OpenSSC 框架)。

设计/集成/ 编译

- 与传感器厂商交互，从其处获取驱动程序 (QTI PoR 驱动程序由 QTI 提供)。
- 安装所需 Hexagon 工具，在 SSC 编译版本 (TBD) 中添加新传感器驱动程序，然后进行编译。

调通/调试

- 有关调通所有物理传感器驱动程序的信息，参见 TBD。
- 参见 TBD 中的调试方法。

验证/调试

- 以不同速率的流传输验证所有物理和虚拟传感器。
- 根据需要，按照要求使用 QTI 调试算法。

定制/ 工厂测试

- 实施定制算法 (80-P9301-67)。
- 实施定制的工厂测试要求 (80-P9301-36)。

功耗调试

- 测量传感器用例的功耗，并根据需要进行调试。

认证

- 为传感器执行 Android 兼容性测试套件 (CTS)。



系统架构

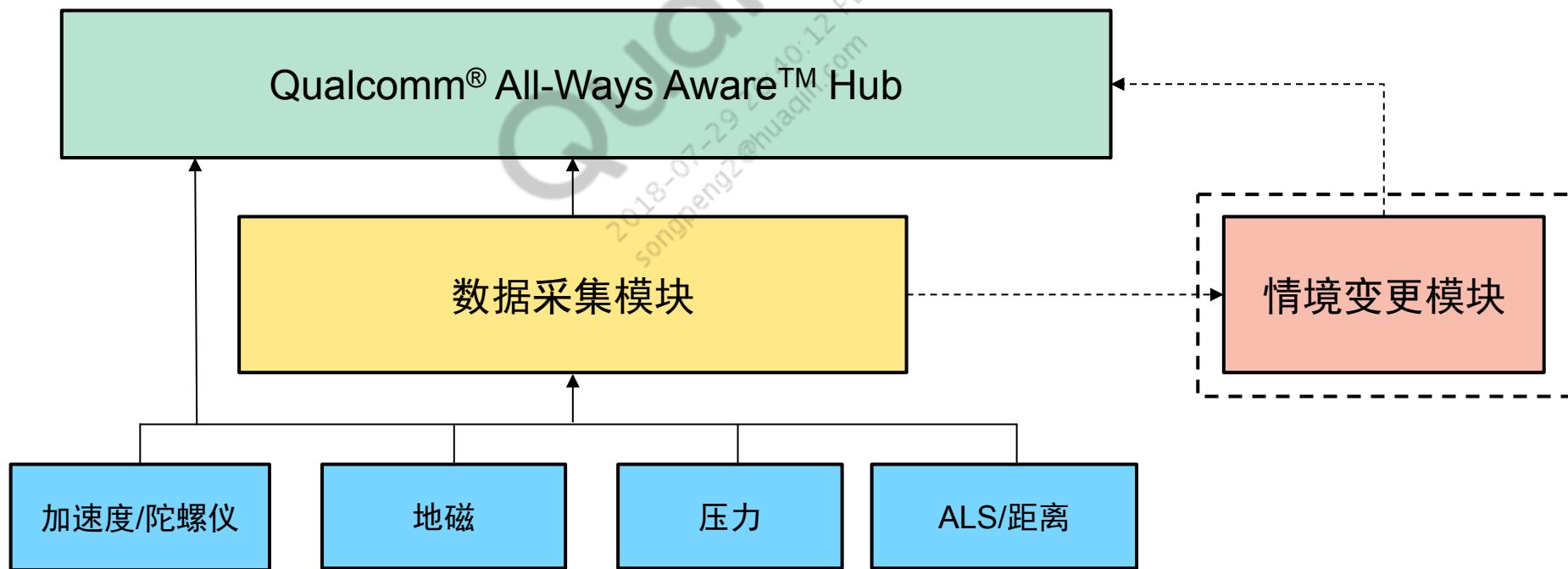
硬件概述

硬件规范

特性	SDM660	SDM670
支持的传感器	<ul style="list-style-type: none"> • 加速度/地磁/陀螺仪 • 距离 • 环境光/RGB/手势 • 压力 • 湿度 	<ul style="list-style-type: none"> • 加速度/地磁/陀螺仪 • 距离 • 环境光/RGB/手势 • 压力 • 湿度 • 霍尔传感器 <div>SDM670 支持 所有 I2C/SPI 传感器</div>
处理器	<ul style="list-style-type: none"> • Qualcomm® Hexagon™ V62C – 高达 900 MHz • 两路硬件线程 • 与音频共享 • 低功耗岛 	<ul style="list-style-type: none"> • Hexagon V65B – 850 MHz（标称），1.2 GHz（Turbo） • 两路硬件线程 • 与音频共享 • 低功耗岛
缓存	<ul style="list-style-type: none"> • L1 – 16I/16D • 岛内存 – 512 KB L2 	<ul style="list-style-type: none"> • L1 16I/16D • 岛内存 – 1 MB、512 KB TCM + 512 KB L2
QUP FIFO 总线接口	<ul style="list-style-type: none"> • BLSP (QUPv2) • 5 个（1 个专用 I2C、1 个专用 SPI、1 个 SPI 或 UART、1 个 SPI 或 I2C、1 个用于调试的 UART） • 32 个 LPI GPIO（由传感器和音频共享） • 18 个传感器专用 GPIO 	<ul style="list-style-type: none"> • QUPv3 • 6 个专用接口（1 个 I2C、2 个 SPI、3 个 UART（1 个用于调试） • 32 个 LPI GPIO（由传感器和音频共享） • 18 个传感器专用 GPIO
特性	<ul style="list-style-type: none"> • 支持以前的所有功能 	<ul style="list-style-type: none"> • 除支持以前的所有功能外，另外支持以下功能： <ul style="list-style-type: none"> • 数据采集模块（新硬件模块） • 情境变更检测器模块（新硬件模块）

数据采集模块

- 可以在不唤醒 DSP 的情况下启用针对传感器数据进行批处理的缓冲器（功耗更低）
- 支持加速度/陀螺仪、地磁（可能支持压力和 ALS/距离传感器）
- 可通过 API 访问



情境变更检测器 (CCD) 模块

- 检测情境（已在软件中验证）中潜在变更的硬件模块
- 检测到情境变化后，唤醒 DSP（功耗降低）
- 可通过 Qualcomm API 访问



SDM670 中的传感器数据调节器 (SDC)、CCD 和 ADSP

- CCD 主要应用于 ADSP LPI 中的门控算法
 - CCD 模块由 DSP 编程和配置
 - AMD – 检测运动状态的硬件模块；与 SDM660 AMD 软件算法相同
 - 行走检测器 – 检测类似行走的运动；为了门控计步器，减少 DSP 唤醒
 - 倾斜检测器 – 检测倾斜
- SDC
 - 由 Hexagon 处理器完全控制
 - QUPv3 GSI 将传感器数据放入 PRAM
 - 将特定样本水印放入 PRAM 后，或 PRAM 即将满载时，SDC 中断 Hexagon 处理器
 - 启用 CCD 模块后，SDC 读取 PRAM 中的加速度样本并将其推入 CCD

SDM670 中的 QUPv3

- QUPv3 是一种高度灵活的可编程模块，支持各种串行接口 – I2C、SPI 和 UART。
- SDM670 中的 QUPv3 取代了 SDM660 和早期芯片组中的 BLSP。
- ADSP LPI 提供六个 QUPv3 – 更多相关信息，参见 [Qualcomm 参考设计](#)。
- 每个 QUPv3 实例最多具有六个通道/GPIO，可以配置为 UART、I2C 和 SPI。

I/O	UART	I2C	SPI
QUP_L0	CTS	SDA	MISO
QUP_L1	RFR	SCL	MOSI
QUP_L2	Tx	–	SCLK
QUP_L3	Rx	–	CS_0
QUP_L4	–	–	CS_1
QUP_L5	–	–	CS_2



产品特性

SDM670 关键特性和增强功能

SDM670 支持低功耗模式

- 数据采集模块
 - 加速度
 - 陀螺仪
 - 地磁
- CCD
 - AMD
 - 行走检测器
 - 倾斜检测器
- 岛模式下支持的传感器驱动程序
 - 加速度/陀螺仪
 - 地磁
 - 压力
 - ALS/距离/RGB/手势
 - 霍尔传感器
 - 湿度
- 岛模式下的算法
 - AMD
 - RMD
 - SMD
 - 计步器
 - 倾斜
 - 旋转矩阵
 - 重力/线性加速度
 - 旋转矢量
 - Geo-mag RV
 - 游戏 RV
 - CMC（活动识别）
 - 距离界限
 - 基本手势
 - 朝向
 - 连续两次晃动
 - PMD
 - 设备方向



设计指南

SDM670 传感器 Qualcomm 参考设计

Qualcomm 参考平台中的传感器

- Qualcomm 参考平台中的传感器（CDP/MTP/手机），默认启用

传感器类型	部件（厂商）	接口
加速度*	LSM6DSOQ (ST)	SPI
陀螺仪*	LSM6DSOQ (ST)	SPI
地磁	AK09917D (AKM)	I2C
压力	BMP285 (Bosch)	I2C
光/距离/RGB/手势	TMG49033 (AMS)	I2C
湿度	SHTW2 (Sensirion)	I2C
霍尔	BU52053NVX (ROHM)	不适用（基于中断）

*LSM6DSOQ – ST LSM6DSOQ 加速度/陀螺仪部件（支持 S4S）

推荐使用 SSC GPIO 和 QUP

- 推荐 OEM 将 SPI 接口应用于加速度和陀螺仪，将 I2C 接口应用于地磁。

	SSC GPIO 编号	GPIO 使用	连接的传感器	
QUP0	SSC_0	SSC_I2C1_SDA	• 地磁 • 压力 • ALS/距离/RGB/手势	• 湿度
	SSC_1	SSC_I2C1_SCL		
QUP1	SSC_2	SSC_SPI1_MISO_TS	• 加速度/陀螺仪	
	SSC_3	SSC_SPI1_MOSI_TS		
	SSC_4	SSC_SPI1_CLK_TS		
	SSC_5	SSC_SPI1_CS0_TS		
	SSC_6	SSC_SPI1_CS1_TS		
	SSC_7	SSC_SPI1_CS2_TS		
QUP2	SSC_8	SSC_SPI2_MISO_TS	Qualcomm® improveTouch™ 解决方案	
	SSC_9	SSC_SPI2_MOSI_TS		
	SSC_10	SSC_SPI2_CLK_TS		
	SSC_11	SSC_SPI2_CS_N_TS		
QUP3	SSC_12	SSC_UART1_TX_BLE	BLE（软件目前不支持）	
	SSC_13	SSC_UART1_RX_BLE		
QUP4	SSC_14	SSC_UART2_TX_DBG	用于调试	
	SSC_15	SSC_UART2_RX_DBG		
QUP5	SSC_16	SSC_UART3_TX_EXT	不使用	
	SSC_17	SSC_UART3_RX_EXT		

传感器控制和中断 GPIO

- 该表显示了 Qualcomm 参考平台中不同传感器控制和中断的 GPIO 编号。
- 提供的所有（MPM 可唤醒）MSM™ GPIO 均可作为中断发送到 SSC 子系统。
- Qualcomm 建议客户使用与所示相同的 GPIO 配置。

MSM GPIO 编号	在 Qualcomm 参考平台中应用
117	ACC_DRDY_INT
118	GYRO_INT
119	MAG_INT_N
120	ALSPG_INT_N
123	PRESS_INT
124	HALL_INT_N
125	TS_INT_N

传感器电源轨使用

- 岛 MX/CX 电源轨
 - SSC_CX: L3B
 - SSC_MX: L14A
- 在 Qualcomm 参考平台中，传感器连接以下电源轨：

传感器类型	VDD	VDDIO
加速度	LDO14A (1.8 V)	LDO14A (1.8 V)
陀螺仪	LDO14A (1.8 V)	LDO14A (1.8 V)
地磁	LDO14A (1.8 V)	LDO14A (1.8 V)
压力	LDO14A (1.8 V)	LDO14A (1.8 V)
光/距离/RGB/手势	LDO14A (1.8 V)	LDO3B (VLED – 3 V)
湿度	LDO14A (1.8 V)	LDO14A (1.8 V)
霍尔传感器	LDO14A (1.8 V)	LDO14A (1.8 V)

注：Qualcomm 强烈建议 OEM 使用相同电源轨。如果 OEM 希望使用不同电源，应在设计硬件平台前由 Qualcomm 硬件、PMIC 和传感器团队审核并获得其批准。



软件概述

传感器执行环境

目标

- 可扩展的简单界面，便于传感器集成
- 稳定性是关键
- 自动化，分层测试
- 更全面的传感器验收测试
- 经过增强的调试性能
- 扩展第三方生态系统以包含算法


为什么会这样？

使用传统框架：

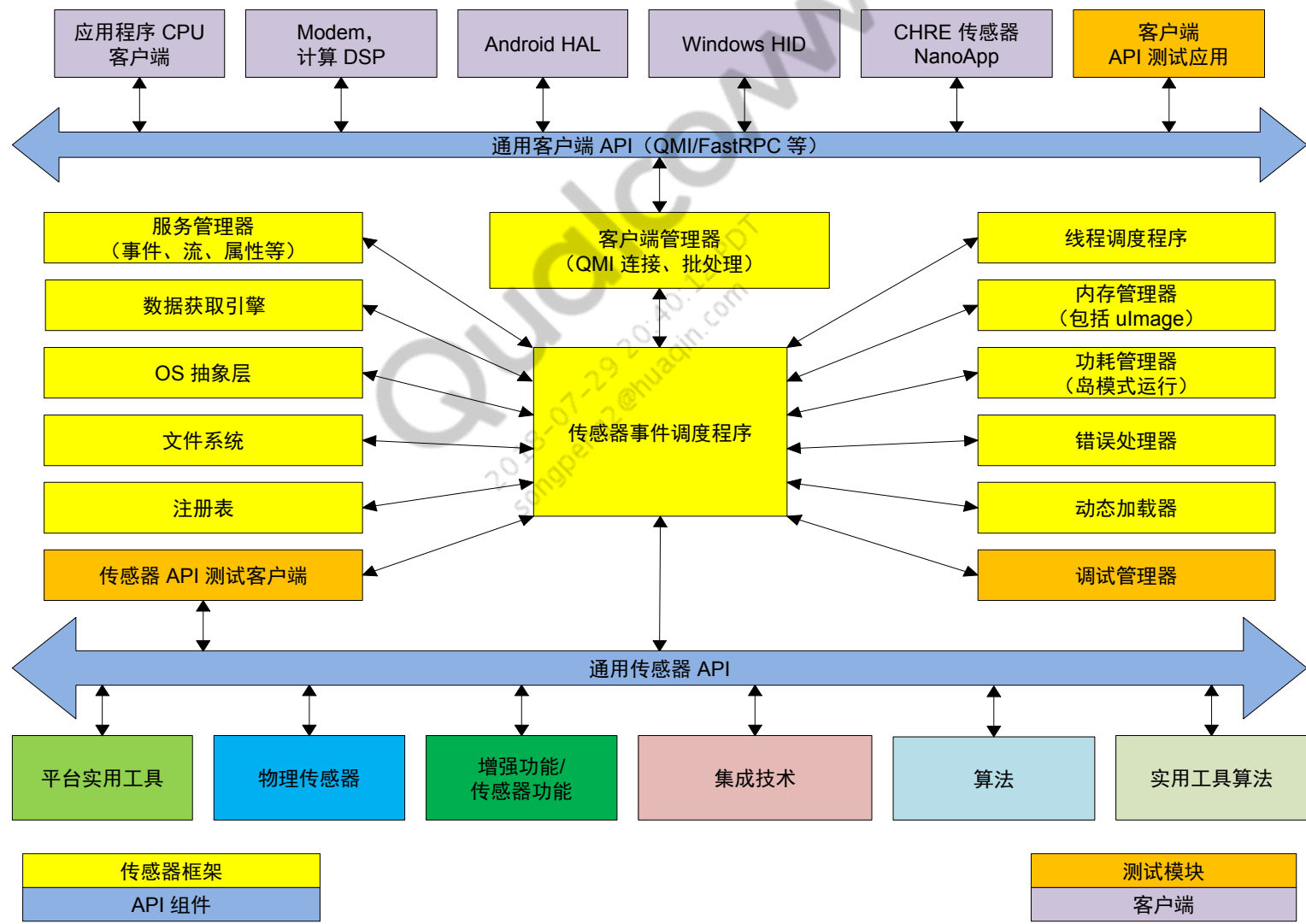
- 功能重叠的多个框架
- 多个 API 使集成工作更为复杂
- 系统功能难以调试和稳定
- IHV 无法按 Qualcomm 标准测试驱动程序
- 新功能需要在框架中更改
- 频繁的功能补丁需要 OEM 集成
- 不支持第三方算法开发



传感器执行环境 (SEE)

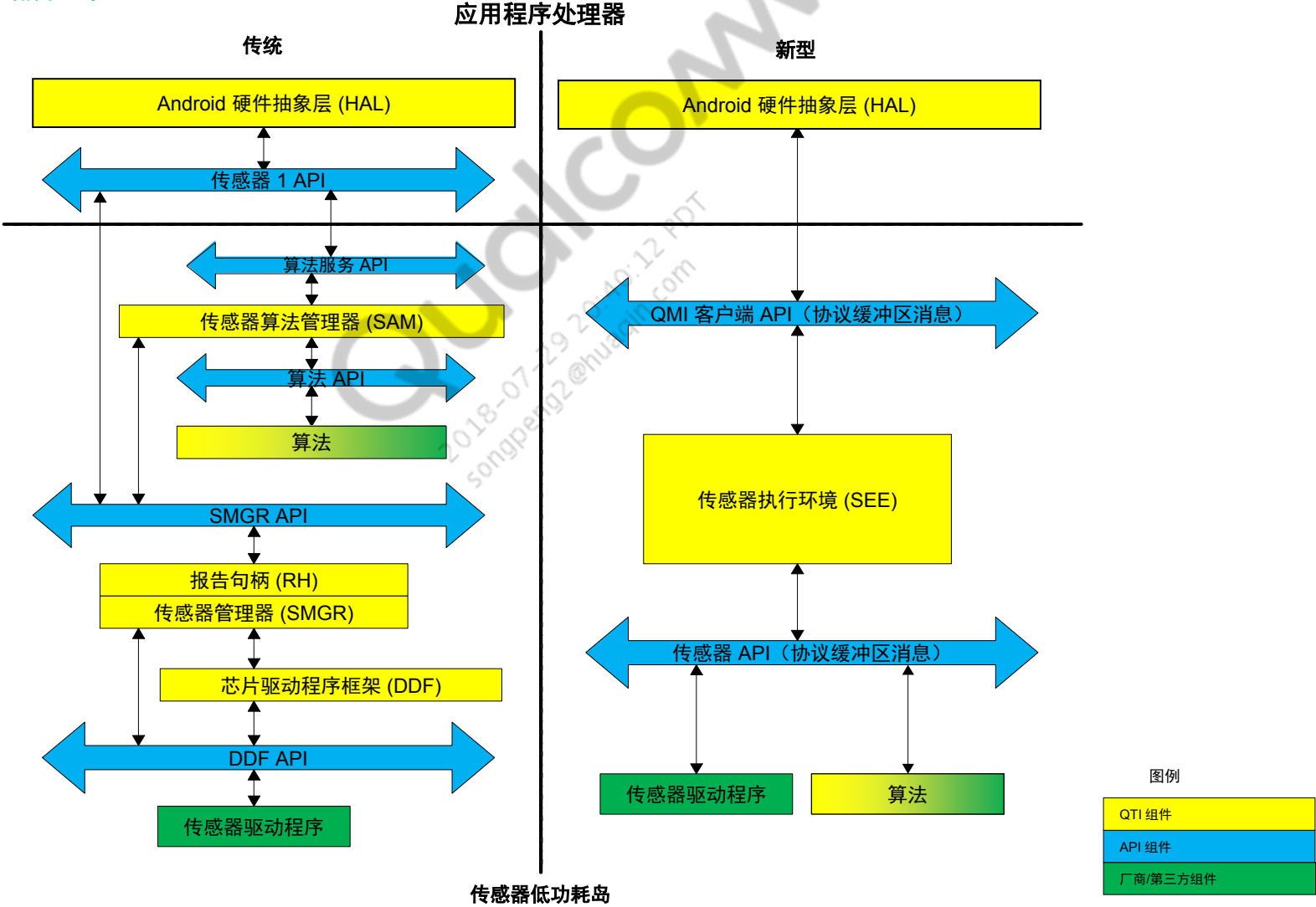
- 适用于算法和驱动程序的单一 API
 - 统一的事件驱动框架
 - 可测试的传感器和客户端接口
 - 第三方算法和驱动程序支持
 - 适用于所有芯片组层级的通用软件
 - 离线仿真环境
- 

SEE 软件框图



SEE 与传统框架高级设计对比

简化的传感器框架



SEE 的框架改进

特性	传统	SEE
稳定性	变化（框架根据新功能发生变化）	可预测 – 框架代码覆盖率高
可测试性	困难（模块间紧密耦合）	清晰的界面和仿真平台提供更好的支持
可调试性	困难（多个框架和 API 的调试支持不同）	统一软件架构提供更好的支持
播放	部分支持限制了可用性	支持完整的 SEE
框架数量	4（DDF、SMGR、RH、SAM）	1 (SEE)
API 的数量	5（DDF、SMGR、SAM 算法、算法服务、传感器 1）	2（传感器和客户端）
坐标系	SAE 坐标系	Android 坐标系
传感器样本数据类型	定点 (q16)	浮点（单精度）
API 消息定义	Qualcomm 消息传送接口	协议缓冲区

设计概念

- 框架组件按功能模块化
- 集成至框架的所有实体均以传感器或服务建模
 - 传感器提供异步数据和事件
 - 服务是用于同步函数调用的统一 API
- 消息格式根据数据类型定义
 - 所有加速度均支持在 `sns_accel.proto` 中定义的统一接口
- 各传感器的唯一标识符 (SUID)
 - 例如, BMA150 为 SUID X; LIS3DH 为 SUID Y; Qualcomm Gravity 为 SUID Z
 - 支持复制传感器硬件
- 传感器决定并发布所有属性
 - 供客户端使用/信息; ODR、操作模式、功耗等。
- 未来的灵活性
 - 动态加载传感器驱动程序和算法

传感器和传感器实例

- 每个传感器可实例化为一个或多个传感器实例。
 - 每个实例运行在特定的配置环境。
 - 对传感器数据的任何请求，都将创建一个传感器实例或共享现有实例。
- 传感器实例根据传感器确定的结果按需创建。
 - 传感器完全管理其相应实例的生命周期和配置，并负责向其客户端发送配置更新和初始状态事件。
 - 强烈建议厂商使用尽可能少的实例为所有客户端请求提供服务。
 - 由实例生成的数据流会发送至所有活动客户端。
- 单个传感器实例可由多个传感器共享和配置。
 - 这种操作模式通常适用于硬件传感器组合驱动程序，其中传感器表示支持的数据类型，传感器实例是硬件通信和配置的唯一模块。

传感器	传感器实例
<ul style="list-style-type: none">• 生成单一类型数据的实体，例如加速度计、陀螺仪、定时器、中断、旋转矢量等。• SUID – 各传感器唯一的 128 位数字• 发布属性（强制和自定义）• 管理其实例	<ul style="list-style-type: none">• 发布输出数据事件的传感器活动实例• 传感器可以根据客户端请求创建实例，或在多个客户端请求之间共享实例• 物理传感器通常共享单一实例

传感器间的通信

- 传感器之间各个方向的通信均通过请求和事件消息执行。
 - 这些消息使用 nanoPB 生成器、编码器和解码器在协议缓冲区格式中定义。
 - 缓冲区长度、消息 ID 和时间戳（位于事件中）在由 SEE 框架管理的元数据内进行通信。



- 发送的请求消息用于启用、禁用和/或配置数据流。
 - 请求消息始终针对特定的 SUID。
 - 当目标传感器接收到请求消息后，就会将其发送至传感器实例进行适当处理。
- 事件消息由传感器实例异步发送至其注册的客户端，这些客户端可以是其他传感器或传感器实例。

相关 API 文件（第 1 页，共 2 页）

- API 在以下文件中定义：
 - `adsp_proc\ssclinc\sns_sensor.h`
 - `adsp_proc\ssclinc\sns_sensor_instance.h`
 - `adsp_proc\ssclinc\sns_register.h`
- proto 文件包含协议缓冲区消息定义和文档：
 - 标准消息定义位于 `adsp_proc\ssc\sensors\pb\sns_std_*.proto`
 - **sns_std.proto** – 包含框架定义的消息 ID、标准请求消息、批处理规范、属性请求和事件以及错误事件消息
 - **sns_std_sensor.proto** – 包含请求的消息 ID、标准传感器的事件 API、流传输请求和事件消息、传感器样本状态类型、标准属性 ID、通用属性类型和物理传感器配置事件消息
 - **sns_std_type.proto** – 包含通用 API 类型定义，如传感器 UID 消息、属性事件和值消息以及通用错误类型
 - **sns_std_event_gated_sensor.proto** – 包含事件门控传感器的 API，如配置消息和 API 文档
 - 物理传感器特定的 API 定义和文档位于传感器特定的 .proto 文件中，例如 `sns_accel.proto`、`sns_proximity.proto`、`sns_motion_detect.proto` 等

相关 API 文件（第 2 页，共 2 页）

- 平台传感器 API 定义和文档位于 \<root>\ssc\sensors\pb\
 - adsp_proc\ssc\sensors\pb\sns_timer.proto
 - adsp_proc\ssc\sensors\pb\sns_interrupt.proto
 - adsp_proc\ssc\sensors\pb\sns_async_com_port.proto
- SUID、注册表、诊断和 DAE 的框架相关 API 在以下文件中定义：
 - adsp_proc\ssc\framework\suid_sensor\pb\sns_suid.proto
 - adsp_proc\ssc\framework\registry\pb\sns_registry.proto
 - adsp_proc\ssc\framework\pb\sns_diag.proto
 - adsp_proc\ssc\framework\dae_sensor\pb\sns_dae.proto
- 测试传感器
 - 标准传感器数据流传输测试位于 adsp_proc\ssc\sensors\test
 - 运动检测和事件门控测试传感器位于 adsp_proc\ssc\sensors\md_test
 - FIFO 刷新测试传感器位于 <root>\ssc\sensors\flush_test

注：API 文档后续可供使用。在此期间，参考 proto 文件。

SDC/DAE 的相关 API

- DAE 驱动程序 API 的文档。该 API 需要通过加速度、陀螺仪和地磁驱动程序实现，以支持 SDC：
 - `adsp_proc\sdg_sensors\inc\sns_dd_if.h`



迁移摘要

从 SDM660 迁移到 SDM670

从 SDM660 到 SDM670 的传感器软件兼容性

组件	与 SDM660 相比发生变化	说明
HAL	是	针对 SDM670 重写了传感器 HAL 代码
传感器客户端 API	是	传感器客户端 API 替换了传感器 1 API
设备驱动程序	是	传感器/传感器实例 API 替换了 DDF、SMGR
算法	是	传感器/传感器实例 API 替换了 SAM API
编译系统	是	ADSP LPI DSP 编译不变的基础上增加了 SDC 编译
消息传送接口	是	使用协议缓冲区替代 QMI 编码/解码
功耗管理	是	依然是专用功耗管理，包括使用 SEE 的新传感器功耗管理器，也因 SDC、CCD 发生变化
操作系统	是	对于 SDC 支持的变更，QuRT™ 软件仍然用于 ADSP LPI 并在 OSA 层抽象化
CoreBSP	是	支持岛模式；因 RPMh（硬件固化型 RPM）和 QUPv3 引入发生变化，支持 SDC；因 improveTouch 采用多 PD

主要影响领域

无法向后兼容 SDM660 或更早版本的芯片组

- 驱动程序
 - 厂商必须开发和测试与 SEE API 兼容的驱动程序。
 - 厂商为早期芯片组开发的 DDF 驱动程序与 SEE 不兼容。
 - OEM 必须使用 SEE API 开发所有自定义自测试代码。
- 算法
 - OEM 必须开发与 SEE API 兼容的自定义算法。
- 工厂代码
 - OEM 必须针对工厂测试使用传感器客户端 API 开发自定义自测试代码。
- HAL
 - OEM 必须针对所有自定义算法开发与 SEE 兼容的 HAL 客户端。



软件定制

自定义驱动程序和算法

SDM670 驱动程序定制

- 记录计划 (PoR)
 - Qualcomm 在参考设计中使用
- 非 PoR 传感器驱动程序
 - 由传感器厂商开发
 - 由传感器厂商根据 Qualcomm 驱动程序验收清单进行测试
 - 由传感器厂商直接分发给 OEM
- 如果传感器未在 SDM670 Qualcomm 参考设计中列出，可要求传感器厂商使用 SEE 开发驱动程序
 - 多数厂商均有权使用 OpenSSC 5.x 工具
 - 有关详细信息，参见 [SSC 供应商生态系统](#)
 - 要求传感器厂商使用 SEE 进行开发
- 要将传感器驱动程序在 SSC 中集成，参见 *New Sensor Driver Integration Guide*（文档号待定）

SDM670 算法定制

- 使用位于 **adsp_proc\ssc\sensors\oem1** 的样本模板算法 (OEM1) 进行算法开发。
- 更多详细信息, 参见 *Adding a Custom Sensors Algorithm with Sensors Execution Environment (SEE)* (80-P9301-67)。



SDM670 传感器代码结构

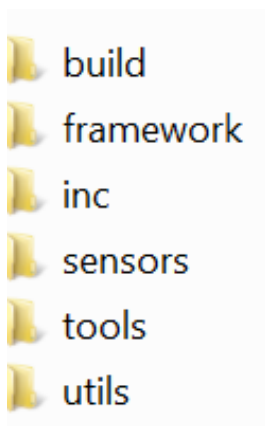
ADSP LPI 代码结构

- 从 Qualcomm ChipCode™ 下载 ADSP LPI 编译版本；命名约定是 ADSP.VT.5.0-00XXX-SDM670-y，其中 XXX-y 代表版本号（仅从传感器角度而言）
- ADSP.VT.5.0-00XXX-SDM670-1\adsp_proc\ 如下例所示：

 build	编译工具
 core	所有 CoreBSP 驱动程序，如内核、总线、系统驱动程序、诊断等
 performance	DSP 分析工具，如 sysmon
 platform	平台实用工具，如 FastRPC、性能等
 qdsp6	DSP 实用工具，包括工具、脚本（如 crashman）
 qmimsgs	QMI 定义
 sdc_core	SDC CoreBSP 组件
 sdc_sensors	SDC 传感器组件
 sectools	安全工具
 ssc	传感器执行环境 (SEE)
 tools	工具目录
 touch	improveTouch 解决方案

ADSP LPI SEE 代码结构

- SEE 代码结构
 - ADSP.VT.5.0-00XXX-SDM670-1 \adsp_proc\ssc 如下例所示：



编译版本

SEE 框架代码

SEE 通用头文件

SEE 传感器，如驱动程序、算法等

工具

SEE 实用工具功能

应用程序处理器传感器代码结构

- 应用程序处理器端上的传感器代码位于 **linux\android\vendor\qcom\proprietary\sensors-see**。

 nanopb	Nanopb
 QSensorTest	QSensorTest 应用
 sensors-diag-log	日志数据包库
 sensors-hal	传感器 HAL 代码
 sensors-log	日志记录库
 ssc	注册表文件、proto 文件、QMI 辅助函数和实用工具
 sscrpcd	Fastrpc 监听器
 test	测试脚本
 USTA	统一传感器测试应用 (USTA)
 Android.mk	



ADSP 工具链和编译版本

SDM670 ADSP LPI 所需工具（待定）

- Python 2.7.6
- Hexagon 编译器工具
 - Linux: Hexagon.LNX.8.1 安装程序（具体版本待定）
 - Windows: Hexagon.WIN.8.1 安装程序（具体版本待定）
- ARM LLVM 3.9.3
 - Linux: Snapdragon_SD_LLVM_ARM.LNX.3.9 安装程序（具体版本待定）
 - Windows: Snapdragon_SD_LLVM_ARM.WIN.3.9 安装程序（具体版本待定）
- NanoPB 生成器: <https://jpa.kapsi.fi/nanopb/download/>
 - Linux: [nanopb-0.3.6-linux-x86.tar.gz](#)
 - Windows: [nanopb-0.3.6-windows-x86.zip](#)

在 Windows 中编译 SDM670 ADSP LPI 版本（待定）

1. 安装所需工具。

2. 使用 ADSP LPI 编译版本（ADSP LPI 映像的一次性需求）集成并设置 nanopb 相关性。

- a. 从 <https://jpa.kapsi.fi/nanopb/download/> 下载 nanopb-0.3.6-windows-x86.zip。
- b. 将 nanopb-0.3.6-windows-x86.zip 文件复制至 adsp_proc\ssc\tools。
- c. 打开命令终端，转至目录：adsp_proc\。
- d. 运行以下命令：
 - “python ssc\build\config_nanopb_dependency.py -f <nanopb_gen_filename>”
 - 其中，nanopb_gen_filename 是所下载 .zip 文件的名称（不含 .zip 扩展名），例如 nanopb-0.3.6-windows-x86。
 - “python ssc\build\config_nanopb_dependency.py -f nanopb-0.3.6-windows-x86”
 - 该命令的输出如下所示：

```
protobuf-2.6.1-py2.7.egg
ssc/tools/nanopb/generator-win/protobuf-2.6.1-py2.7.egg
setuptools-0.6c11-py2.7.egg
ssc/tools/nanopb/generator-win/setuptools-0.6c11-py2.7.egg
ssc\tools\nanopb-0.3.6-windows-x86
ssc\inc\utils\nanopb
pb.h
pb_common.h
pb_decode.h
pb_encode.h
```

3. 运行以下命令：

```
“python build\build.py -c sdm670 -o all”
```

编译日志保存在 adsp_proc\build\ms\build-log.txt 中。

注：有关最新编译说明，参见版本说明。

在 Linux 中编译 SDM670 ADSP LPI 版本（待定）

1. 安装所需工具。
2. 使用 ADSP LPI 编译版本（ADSP LPI 映像的一次性需求）集成并设置 nanopb 相关性。
 - a. 从 <https://jpa.kapsi.fi/nanopb/download/> 下载 nanopb-0.3.6-linux-x86.tar.gz。
 - b. 将 nanopb-0.3.6-linux-x86.tar.gz 文件复制至 adsp_proc\ssc\tools。
 - c. 打开命令终端，转至目录：adsp_proc\。
 - d. 运行以下命令：

```
“python ssc\build\config_nanopb_dependency.py -f <nanopb_gen_filename>”
```

 - 其中 nanopb_gen_filename 是所下载文件的名称（不含 .tar.gz 扩展名），例如 nanopb-0.3.6-linux-x86。
 - “python ssc\build\config_nanopb_dependency.py -f nanopb-0.3.6-linux-x86”
 - 该命令的输出如下所示：

```
ssc/tools/nanopb-0.3.6-linux-x86.tar.gz
protobuf-2.6.1-py2.7.egg
ssc/tools/nanopb/generator/protobuf-2.6.1-py2.7.egg
ssc/tools/nanopb-0.3.6-linux-x86
ssc/inc/utils/nanopb
pb_encode.h
pb_common.h
pb.h
pb_decode.h
```
3. 运行以下命令：

```
python build\build.py -c sdm670 -o all
```

编译日志保存在 adsp_proc\build\ms\build-log.txt 中。

注：有关最新编译说明，参见版本说明。



调试概述

SDM670 调试

SDM670 传感器调试

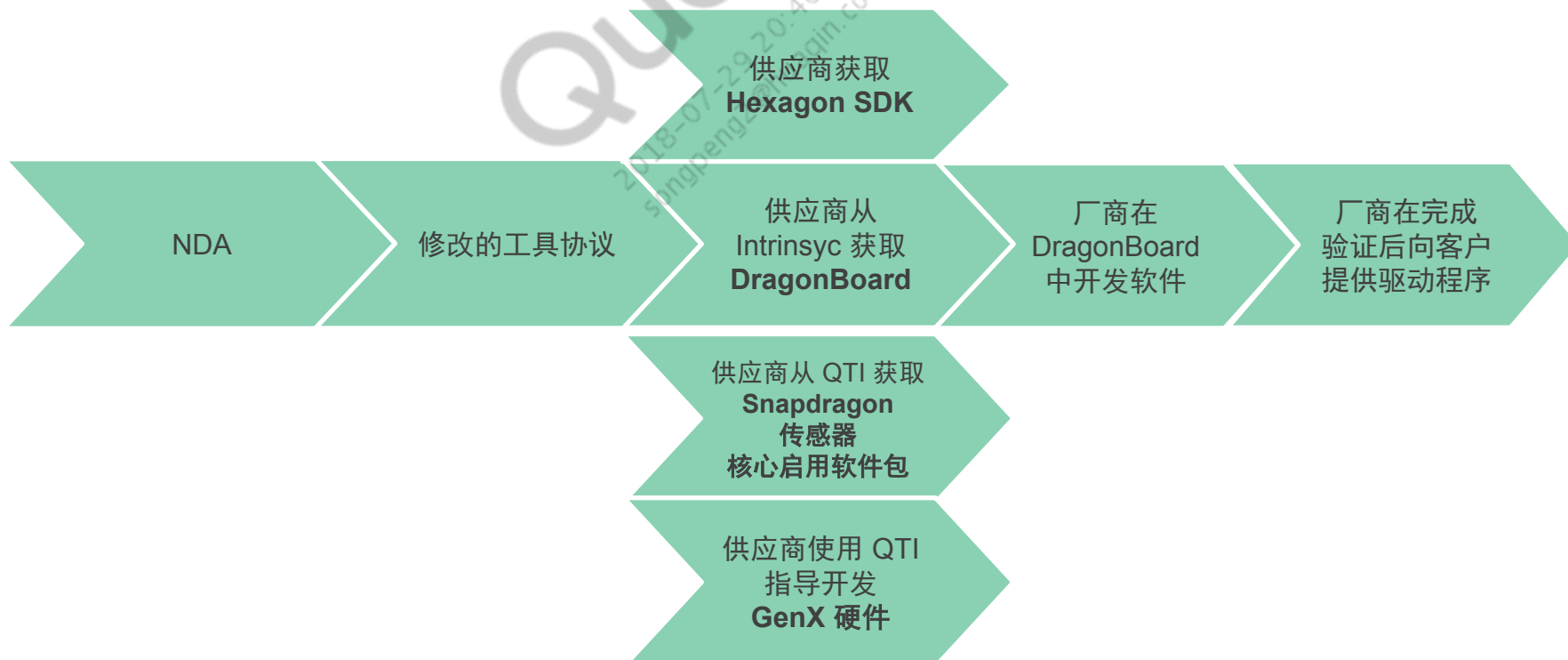
- ADSP LPI 中的传感器调试信息类型
 - 打印字符串
 - 日志数据包（格式和大小在编译时定义并固定的消息）
- 调试信息查看工具
 - ADB Logcat
 - QXDM Professional™ (QXDM Pro)
- ADB Logcat 和 QXDM Pro（QXDM v4.0.187 及更高版本）
 - 这两种工具均需 USB 连接；为传感器模块提供宏和 API，从而将打印字符串和日志数据包发送到工具
 - 应用程序处理器直接连接 USB 端口；应用程序处理器中的传感器模块可通过直接调用宏和 API 发送调试信息。ADSP LPI 未直接连接 USB 端口
 - 多数应用程序处理器端的调试通过 ADB Logcat 完成
 - ADSP LPI 端的调试主要通过 QXDM 日志完成



传感器厂商生态系统

传感器厂商生态系统

- Qualcomm Technologies, Inc. (QTI) 为传感器厂商提供生态系统，以便开发 SSC 设备驱动程序。
 - 硬件 – 供应商可以从 [Intrinsyc](#) 直接订购 Open-Q 820 开发套件。
 - 软件 – 厂商签署必要的法律协议后，QTI 为其提供所需软件（OpenSSC 5.x 包）和工具。
 - 传感器厂商可以在完成验证后直接向客户提供驱动程序。



提供 DAE 支持的传感器厂商生态系统

- 使用 OpenSSC 5.x 包和 Open-Q 820 平台，厂商可以开发和测试 SDM670 设备驱动程序。由于 Open-Q 820 平台的硬件限制，还不能验证驱动程序的 DAE 部分。
- Qualcomm 正在协助传感器厂商重新设计驱动程序以支持 DAE；以下是高级验证计划：
 - 对于 SDM670 平台中的 PoR 传感器，Qualcomm 验证 DAE 驱动程序（与预期相同）。
 - 对于非 PoR 传感器，Qualcomm 为厂商提供了一种验证支持 DAE 的传感器的机制（与 Qualcomm 团队合作）。该过程将继续，直至基于 SDM670 的硬件开发套件 (HDK) 可供使用。
- 建议 OEM 尽早联系 Qualcomm 传感器团队，提供需要 DAE 提供支持的加速度、陀螺仪和地磁传感器部件的传感器选择。



支持

传感器客户支持

- 客户工程支持通过 QTI Salesforce 门户网站提供，可访问 <https://support.cdmatech.com>
- 选择问题领域，将您的案例提交给传感器 CE 团队：
 - 问题领域 1
 - 板卡支持包 (BSP)
 - 问题领域 2
 - 驱动程序 – 外设
 - 问题领域 3
 - 传感器 – 传感器核心 – 解决 SSC 端的问题
 - 传感器 – AP – 解决 AP 端的问题



参考资料

参考资料（第 1 页，共 2 页）

标题	文档号
Qualcomm Technologies, Inc.	
<i>New Sensor Driver Integration Guide</i>	待定
<i>Adding a Custom Sensors Algorithm with Sensors Execution Environment (SEE)</i>	80-P9301-67
<i>Sensors Execution Environment Client API Reference</i>	80-P9301-36
<i>Qualcomm® Snapdragon™ Sensors Core (SSC) Features for Linux Android</i>	80-NH058-1
<i>Qualcomm® AAH Compatible Driver List for SEE</i>	80-NB925-2

参考资料（第 2 页，共 2 页）

缩略词或术语	定义
ADSP	音频 DSP (Audio DSP)
AMD	绝对运动检测器 (Absolute motion detector)
CCD	情境变更检测器 (Context change detector)
CTS	兼容性测试集 (Compatibility test suite)
DDF	芯片驱动程序框架 (Device Driver Framework)
ODR	输出数据速率 (Output data rate)
PoR	记录计划 (Plan of record)
PMD	持久运动检测器 (Persistent motion detector)
RH	报告句柄 (Report handler)
RMD	相对运动检测器 (Relative motion detector)
SAM	传感器算法管理器 (Sensors Algorithm Manager)
SDC	传感器数据调节器 (Sensors data conditioner)
SEE	传感器执行环境 (Sensors Execution Environment)
SMD	显著运动检测 (Significant motion detection)
SMGR	传感器管理器 (Sensors Manager)
SPI	串行外围设备接口 (Serial Peripheral Interface)
SUID	传感器唯一标识符 (Sensor Unique Identifier)

问题？

欲了解更多信息或者存在技术问题，可访问 <https://createpoint.qti.qualcomm.com>

