AN5332

从 Kinetis KE0x 到 KE1x 的移植指南

版本 0 — 2016 年 9 月

应用笔记

目录

1 导言

Kinetis E 系列家族提供了一个具有高度扩展性的产品线，包含了从 20 MHz 的 Arm® Cortex®-M0+ MCU 到 168 MHz 的 Arm Cortex-M4 MCU。Kinetis E 系列具有 2.7 V - 5.5 V 的供电范围，并专注于出色的 EMC/ESD 性能，非常适合在恶劣电气环境中的各种应用，并针对成本敏感型应用进行了优化。Kinetis E 系列提供了广泛的内存、外设和封装选项。

本应用笔记描述了 KE0x 和 KE1x 产品系列之间的关键差异和改进，提供了如何从一个系列移植到另一个系列的指导方针，并有助于缩短学习曲线。

2 Kinetis E 系列设备概述

2.1 Kinetis KE0xZ 子系列概述

Kinetis KE0xZ MCU 基于 Cortex-M0+ 内核，运行速度可达 48 MHz，提供最高可达 128 KB 的闪存、16 KB RAM 和 256 B EEPROM。它们包括强大的模拟、通信、定时和控制外设阵列。该系列由低功耗、高稳定性和具有成本效益的 MCU 组成，这些 MCU 提供了一个合适的 32 位入门级解决方案。

- KE02Z — 广泛提供混合信号集成、ADC、DAC、ACMP 和 Flex 定时器，并添加 256-B EEPROM。
- KE04Z — 扩展了 KE02Z 系列，增加了 BME 模块，但去掉了 EEPROM。
- KE06Z — 通过添加 MSCAN 模块扩展了 KE04Z 系列。

2.2 Kinetis KE1xZ 子系列概述

Kinetis KE1xZ MCU 基于 Cortex-M0+ 内核，运行速度可达 72 MHz，提供最高可达 256 KB 的闪存、32 KB RAM 和一套完整的模拟/数字特性。KE1xZ 将 Kinetis E 族扩展到更高的性能和更广泛的可伸缩性。稳定的 TSI 为您的 HMI 系统提供了高水平的稳定性和准确性。1MSPS ADC 和 Flex 定时器非常适合于 BLDC 电机控制系统。

- KE14Z — 广泛提供混合信号集成、ADC、CMP 和 Flex 定时器。
- KE15Z — 通过添加 TSI 模块扩展了 KE14Z 系列。

2.3 Kinetis KE1xF 子系列概述

Kinetis KE1xF MCU 是 Kinetis E 系列的高端 MCU，通过运行频率高达 168 MHz 的高性能 Cortex-M4 内核，提供了强大的 5 V 解决方案。 KE1xF 提供了多个 ADC 和 FlexTimers，符合 CAN 2.0B 的 FlexCAN 模块，以及丰富的通信接口套件，包括 UART，I2C，SPI 和 FlexIO，为串行通信仿真提供了灵活性。 设备从 64LQFP 软件包中的 256 KB 闪存开始，到 100LQFP 软件包中的 512 KB 闪存开始。

- KE14F — 广泛提供混合信号集成、ADC、DAC、ACMP 和 Flex 定时器。
- KE16F — 通过添加 Flex CAN 模块扩展了 KE14F 系列。
- KE18F — 通过添加两个 Flex CAN 模块扩展了 KE14F 系列。

3 软件比较

3.1 KExx_drivers KE0x 库

KExx_drivers 库是在 KE0x 系列的 FRDM 板上工作的恩智浦软件驱动程序。这些软件驱动程序以源代码的形式提供。所有源文件都应直接包含在应用程序项目中，或单独构建到静态链接库中。

KExx_drivers 库被视为恩智浦 KE0x 系列外设驱动程序和示例代码的历史产物。它具有通用编码风格的优点，并且易于使用。该软件结构也易于理解，您可以轻松自定义驱动程序和应用程序。缺点是编码风格与 SDK 不兼容，并且很难移植到其他 Kinetis 系列。

3.2 KE1x 的 Kinetis SDK v2.0

Kinetis 软件开发套件（KSDK）v2.0 是 KE1x 设备的软件支持的集合，其中包括外围驱动程序以及对 FreeRTOS OS 和 µC/ OS 的集成 RTOS 支持。除了基本支持之外，还通过演示应用程序，驱动程序示例项目和 API 文档对 KSDK 进行了增强，以帮助快速利用 Kinetis SDK 的支持。图 1 为架构图。

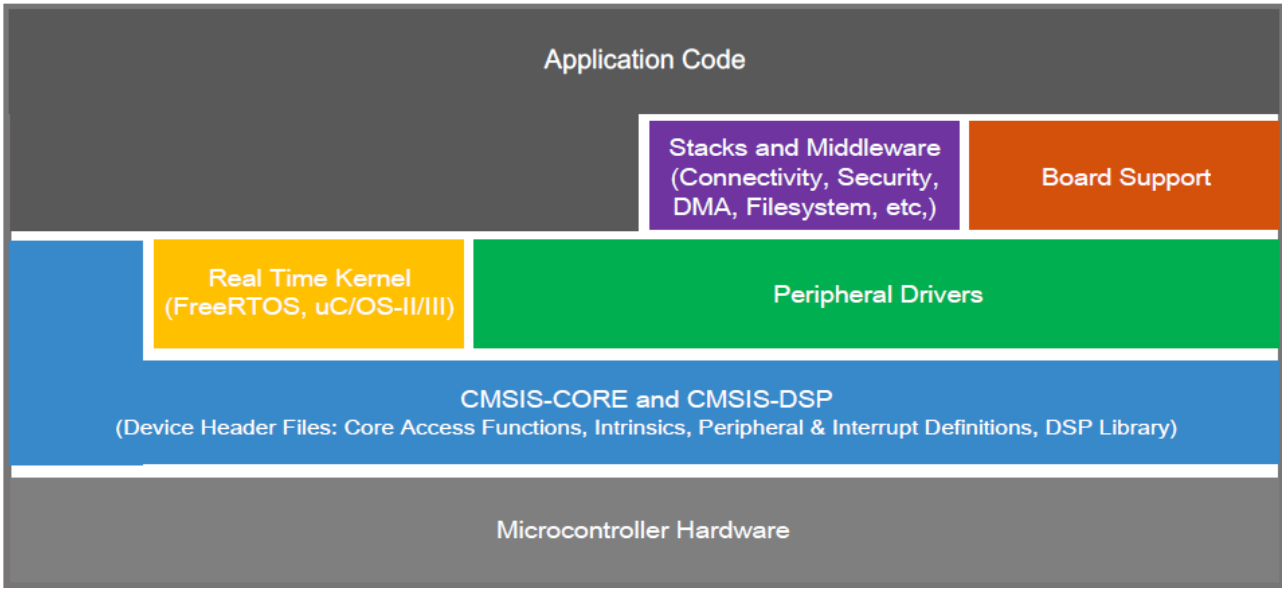

图 1. KSDK v2.0 架构图

3.2.1 KSDK 板支持文件夹

KSDK 开发板支持为 Kinetis 开发和评估板（塔式系统模块化开发平台/NXP Freedom 等）提供了示例应用程序。板支持程序包位于顶层 *boards* 文件夹中，每个受支持的板都有自己的文件夹（一个 KSDK 程序包可以支持多个板）。每个 *<board_name>* 文件夹中的各个子文件夹都将其包含的示例类型分类。这些文件夹包括（但不限于）：

- *demo_apps* — 功能齐全的应用程序，旨在突出目标 MCU 的关键功能和用例。这些应用程序通常使用多个 MCU 外设，并可能利用堆栈和中间件。
- *driver_examples* — 简单的应用程序,旨在简洁地说明如何将 KSDK 的外围驱动程序用于单用途情况。这些应用程序通常只使用一个外设，但也有使用更多外设的情况(例如，使用 DMA 的 ADC 转换)。
- *rtos_examples* — 基本的 FreeRTOS 操作系统示例展示了各种 RTOS 对象（信号量、队列和其他）的使用，并与 KSDK 的 RTOS 驱动程序进行了接口。

3.2.2 应用范例结构

boards 文件夹中的每个*<board_name>*子文件夹包含一组与特定硬件相关的全面示例。以 *hello_world* 文件夹为例（*demo_apps* 文件夹的一部分），但同样的一般规则适用于*<board_name>* 文件夹中的任何类型的示例。

图 2 显示了 *hello_world* 应用程序文件夹的内容。

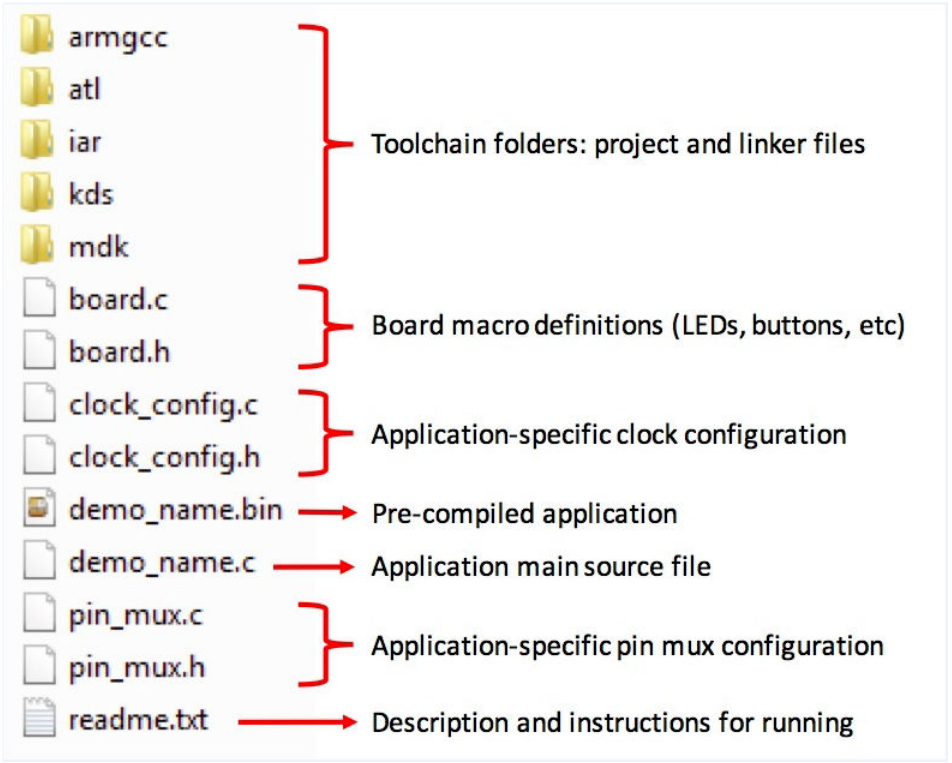

图 2. 应用程序文件夹结构

应用程序文件夹中的所有文件都是特定于示例的，因此非常容易复制和粘贴现有示例，以便根据 KSDK 中提供的项目开始开发自定义应用程序。

3.2.3 应用示例源文件的位置

在任何受支持的 IDE 中打开示例应用程序时，程序都会引用各种源文件。KSDK 的 *devices* 文件夹是应用程序的核心组件，也是所有示例应用程序的中心组件。由于它是一个核心组件，所有示例都引用相同的源文件，如果修改其中一个文件，其他示例的行为可能会受到影响。

所有示例应用程序中使用的 KSDK 树的主要区域是：

- *devices/<device_name>* — 设备 CMSIS 头文件、KSDK 功能文件和其他一些项。
- *devices/<device_name>/drivers* — 特定 MCU 的所有外围驱动程序。
- *devices/<device_name>/<tool_name<device_name>* — 工具链特定的启动代码。向量表定义位于这里。
- *devices/<device_name>/utilities* — 许多示例应用程序（如调试控制台）使用的项目。

对于包含中间件/堆栈和/或 RTOS 的示例，用户可以引用适当的源代码。中间件源文件在 *middleware* 文件夹中，RTOS 在 *rtos* 文件夹中。 同样，每个文件夹的核心文件都是共享的，因此修改它们可能会对依赖它们的其他项目产生潜在影响。

详细信息，请参阅 Kinetis SDK。本手册以下部分中的提及的所有代码片段都基于 KSDK v2.0。

4 硬件资源比较

4.1 系统级别的差异

由于 Kinetis E 产品系列是建立在不同的处理器核心上的，并且用于不同的目的和应用领域，所以 KE0x 和 KE1x MCU 之间存在着显著的差异。 表 1 概述了系统级别差异。

表 1. 系统级别的差异

特色	KE0x Z	KE1x Z	KE1x F
处理器核心	Cortex-M0+	Cortex-M0+	Cortex-M4
最大 CPU 频率	48 MHz	72 MHz	168 MHz
DSP 和 FPU	—	YES	YES
TRGMUX	—	YES	YES
调试	SWD	SWD	JTAG + SWD
闪存大小	高达 128 KB	高达 256 KB	ECC 高达 512 KB
SRAM	高达 16 KB	高达 32 KB	ECC 高达 64 KB
EEPROM 或 Flex RAM	高达 256 KB	最多 2 KB	最多 4 KB
Flex NVM	—	高达 32 KB	高达 64 KB
BOOTROM	—	YES	YES
GPIO	高达 71 I/O	高达 89 I/O	高达 89 I/O

4.1.1 时钟模式

注意

时钟模式的差异会显著影响时钟启动和应用程序的配置。

KE0x 包含这些片上时钟源：

- 内部时钟源（ICS）模块 — 主时钟源生成器，为外围设备提供总线时钟和其他参考时钟。
- 系统振荡器（OSC）模块 — 向 ICS 提供参考时钟的系统振荡器、实时时钟（RTC）计数器时钟模块和其他 MCU 子系统。
- 低功耗振荡器（LPO）模块 — 片上低功耗振荡器，为 RTC 和看门狗（WDOG）提供 1 kHz 的参考时钟。

KE0x 时钟如 图 3 所示。每个模块的时钟可以使用 SIM_SCGC 寄存器单独开关。在初始化模块之前，在 SIM_SCGC 寄存器中设置相应的位以启用时钟。在关闭时钟之前，请确保先禁用模块。对具有时钟禁用功能的外围设备的任何总线访问都会产生错误终止。

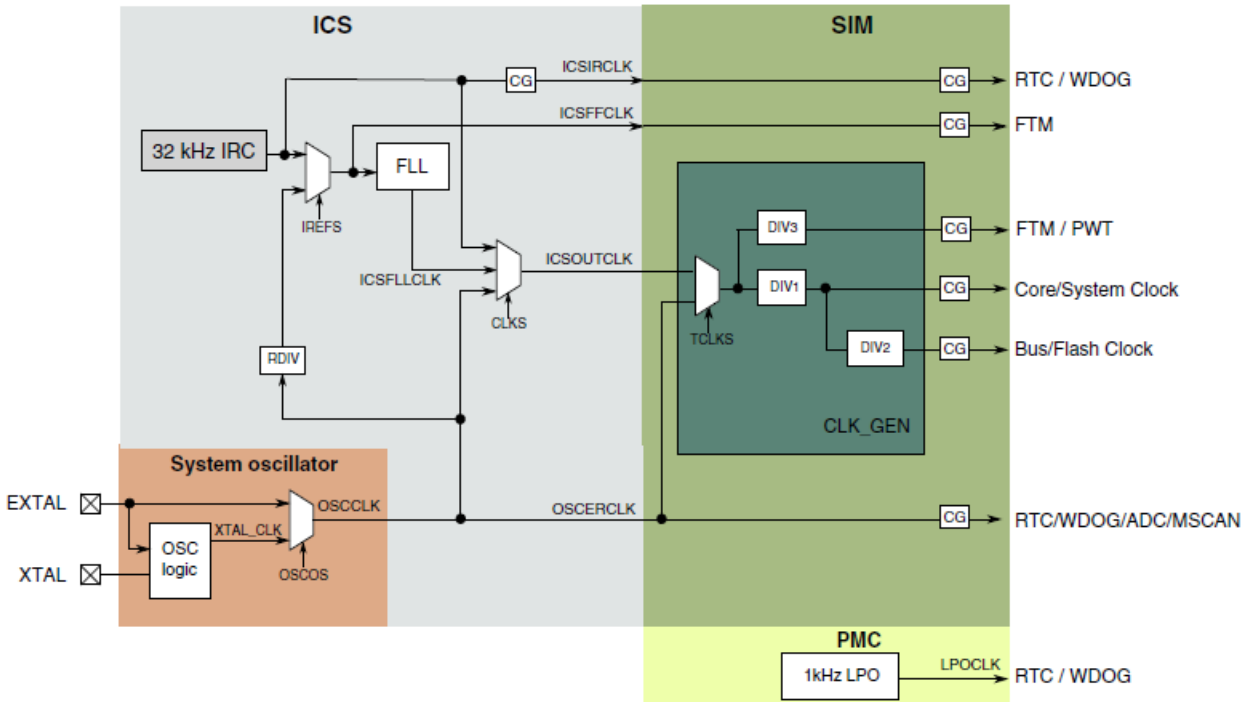

图 3. KE0x 时钟图

在 KE1x 系列中，我们部署了一个新的修改后的系统时钟生成模块（SCG）来创建各种时钟树。该装置有几个时钟源，即外部系统时钟/晶体、外部 RTC 时钟/晶体、内部快速 IRC（FIRC）振荡器、内部慢速 IRC（SIRC）振荡器和 128 kHz 低功耗振荡器（LPO）。 SCG 模块包括外部系统时钟/晶体电路、FIRC 和 SIRC。

外部 RTC 时钟/晶体用作 RTC 模块的时钟源。LPO 时钟源仅用于某些模块（例如 WDOG、LPTMR 和其他模块）作为可选的外围时钟源，SCG 模块不用于创建替代时钟频率。其他三个时钟源由 SCG 提供 CPU/平台时钟、总线/闪存时钟和替代外围时钟。包括 PLL（在 KE1xF 中）或 LPFLL（在 KE1xZ 中），以生成 CPU/平台频率到最大频率。

图 4 为 KE1xF 的时钟图。 与 KE1xZ 的图相比，有一些差异，主要是 PLL 块被 LPFLL 所取代。在 KE1x 系列中，每个模块的时钟门控制、时钟源选择和时钟分配器使用专用 PCC 模块而不是 SIM 模块。更多有关 SCG 和 PCC 模块的详细信息，请参见 *Clock management and distribution in KL28*（文档 AN5231）。

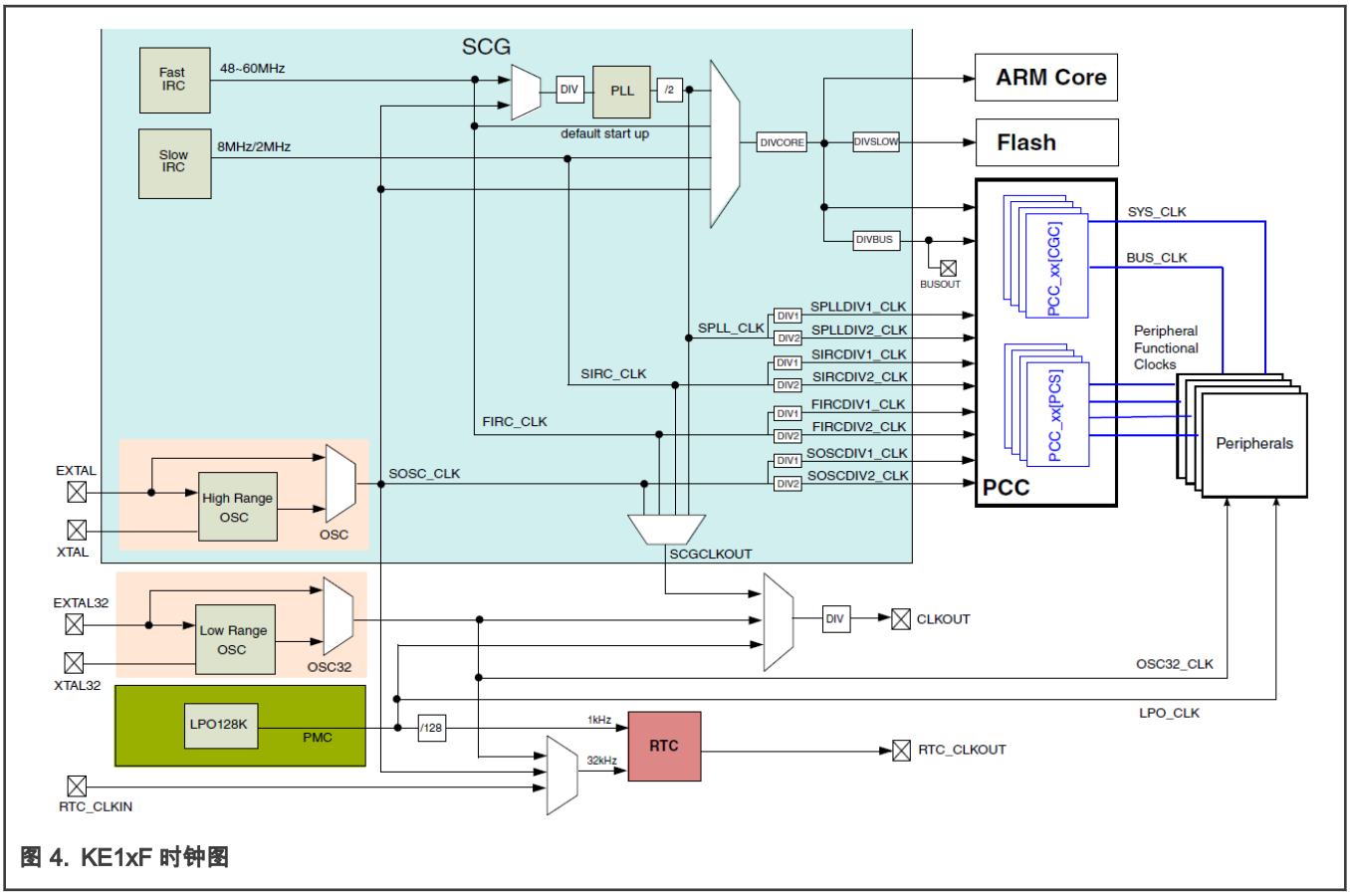

图 4. KE1xF 时钟图

用于 KE1x 时钟的 SDKv2.0 驱动程序/示例代码包括：

- 板启动，使 MCU 进入 RUN/HSRUN 模式。启用了 FIRC、SIRC 和 SYSOSC 时钟。您可以手动配置内核/总线时钟频率和分配器。默认情况下，内核时钟在 HSRUN 模式下运行到最大速度。

```
void BOARD_BootClockRUN(void);
void BOARD_BootClockHSRUN(void);
```

- 模块时钟门控制。 参数 `name` 是外设的时钟名称定义。

```
void CLOCK_EnableClock(clock_ip_name_t name);
void CLOCK_DisableClock(clock_ip_name_t name);
```

- 模块时钟源选项和分频器设置。参数 `name` 是外设的时钟名称定义，参数 `src` 是时钟源定义。它可能来自 SIRC、FIRC、SYSOSC、LPFLL/PLL 和其他。其中参数 `divValue` 为除法值，参数 `fracValue` 为分数乘法值。

```
void CLOCK_SetIpSrc(clock_ip_name_t name, clock_ip_src_t src);
void CLOCK_SetIpSrcDiv(clock_ip_name_t name, clock_ip_src_t src, uint8_t divValue, uint8_t
fracValue);
```

4.1.2 功耗模式

KE0x 电源管理控制器（PMC）提供了多种电源选项。不同的操作模式使您可以针对所需功能级别优化功耗。它支持“运行”，“等待”和“停止”模式，根据不同的功耗水平和功能要求，这些模式易于使用。I/O 状态在所有模式下均保持不变。

- 运行模式（Run） — CPU 时钟可以全速运行，内部供应是完全调节的。
- 等待模式 （Wait）— CPU 关闭以节省能源。系统时钟和总线时钟运行，并保持完整的调节。

- 停止模式 （Stop）— LVD 可选启用，电压调节器处于待机模式。

表 2. KE0x 功耗模式

功耗模式	说明	内核模式	正常恢复方法
运行模式（Run）	允许芯片的最大性能。这是重置后的默认模式。片上电压调节器导通。	Run	—
等待模式（Wait）	允许外设在核心处于睡眠模式时工作，降低功耗。NVIC 对中断敏感。外设被时钟控制。	Sleep	中断
停止模式（Stop）	将芯片置于静态状态。它是最低功率模式，保留所有寄存器，同时可选地保持 LVD 保护。NVIC 被禁用。AWIC 用于从中断中唤醒。 外围时钟停止。	Deep sleep	中断

与 KE0x 相比，KE1xPMC 具有正常调节和低功率调节两种调节模式。为节能和不同的应用使用提供了更多的功耗模式。正常调节模式包括高速运行（HSRUN）模式的用户可配置过驱动选项。

KE1x 中的功耗模式如下：

- 正常调节：
  - — HSRUN（仅 KE1xF）— 核心时钟最大 168 MHz。
  - — RUN — 核心时钟最大 120 MHz（KE1xF）/核心时钟最大 72 MHz（KE1xZ）。
  - — WAIT — CPU 睡眠模式。
  - — STOP — CPU 深度睡眠模式；低功耗外设可选打开。
- 低功率调节：
  - — VLPR — 非常低功耗运行模式，核心时钟最大 4 MHz。
  - — VLPW — 非常低功耗等待模式，总线时钟最大 1 MHz。
  - — VLPS — 非常低功耗停止模式，低功耗外设可选打开。

KE1x 上提供三种低功耗（VLPx）模式。调节处于低功耗模式以节省功率，并且来自 SIRC 或外部振荡器的时钟最大限制为 4 MHz。RTC，LPTimer，ADC，DAC 和 CMP 在 VLPx 模式下可选地处于活动状态。ADC 可以选择以 VLPx 模式开启，但性能有限，因为可用的时钟源是 SIRC 或 OSC（最大 16 MHz 晶振）。

表 3. KE1x 功耗模式

功耗模式	说明	内核模式	恢复到正常模式途径
运行模式（Run）	复位后的默认模式；片上电压调节器打开。	Run	—
高速运行模式（HSRUN（仅 KE1xF））	允许芯片发挥最大性能。片上稳压器已打开，但电压输出略有升高。在这种状态下，与正常运行模式相比，MCU 能够以更高的频率运行。	Run	—
等待模式（Wait）	在内核处于睡眠模式时允许外围设备工作，从而降低了功耗。NVIC 对中断很敏感。外设时钟。	Sleep	中断
停止模式（Stop）	将芯片置于静态。这是最低功耗模式，可在保持 LVD 保护的同时保留所有寄存器。 NVIC 被禁用。AWIC 用于从中断中唤醒。外围时钟停止。	Deep Sleep	中断

下页继续...

表 3. KE1x 功耗模式（续上页）

功耗模式	说明	内核模式	恢复到正常模式途径
非常低功耗运行模式（VLPR）	片上稳压器处于低功耗模式，仅提供以降低的频率运行芯片所需的功率。这是降频闪存访问模式（1 MHz）。LVD 已关闭。内部振荡器为内核，总线和外设时钟提供了一个低功耗的 4MHz 源。	Run	—
非常低功耗等待模式（VLPW）	与 VLPR 相同，但内核处于睡眠模式，以进一步降低功耗。NVIC 对中断保持敏感（FCLK = ON）。片上稳压器处于低功耗模式，仅提供以降低的频率运行芯片所需的功率。	Sleep	中断
非常低功耗停止模式（VLPS）	LVD 操作关闭时，将芯片置于静态。这是具有 ADC 的最低功耗模式，并且引脚中断功能正常运行。 外设时钟停止，但是可以使用 LPTMR，RTC，CMP 和 DAC。 NVIC 被禁用（FCLK = OFF）。AWIC 用于从中断中唤醒。片上稳压器处于低功耗模式，仅提供以降低的频率运行芯片所需的功率。整个 SRAM 都将运行（保留内容并保留 I / O 状态）。	Deep Sleep	中断

用于 KE1x 功耗模式配置的 SDKv2.0 驱动程序/示例代码如下：

- 将 MCU 配置为不同的功耗模式。对于正常的停止模式设置，扩展的 `option` 参数可以选择部分 Stop 模式。

```
status_t SMC_SetPowerModeRun(SMC_Type *base);
status_t SMC_SetPowerModeHsrun(SMC_Type *base);
status_t SMC_SetPowerModeWait(SMC_Type *base);
status_t SMC_SetPowerModeStop(SMC_Type *base, smc_partial_stop_option_t option);
status_t SMC_SetPowerModeVlpr(SMC_Type *base);
status_t SMC_SetPowerModeVlpw(SMC_Type *base);
status_t SMC_SetPowerModeVlps(SMC_Type *base);
```

4.1.3 模块互连

在 KE0x 系列中，模块到模块的互连是固定的，不能重新布线。您只能使用 SIM 寄存器进行有限的配置。

KE1x 方案中的模块互连基于 TRGMUX（如 图 5 所示），TRGMUX 是与现有 Kinetis 模块不同的新设计模块。 TRGMUX 引入了一种极其灵活的方法，可以将各种触发源连接到多个引脚/外围设备。

TRGMUX 最初是为多核体系结构设计的，但它也支持单核体系结构。与以前的基于 SIM 的解决方案相比，它提供了一种非常灵活的解决方案。TRGMUX 设计为 TRGMUX0 添加了 TRGMUX1 电平的预触发源。TRGMUX1 支持多达 32 个触发源，并具有八个输出。这八个输出是 TRGMUX0 的触发输入。TRGMUX0 最多支持 32 个输入源，其输出是目标模块。

对于 TRGMUX，每个接受外部触发的外设通常都有一个特定的 32 位触发控制寄存器。每个控制寄存器最多支持四个触发器，每个触发器可以从多达 32 个输入中选择。

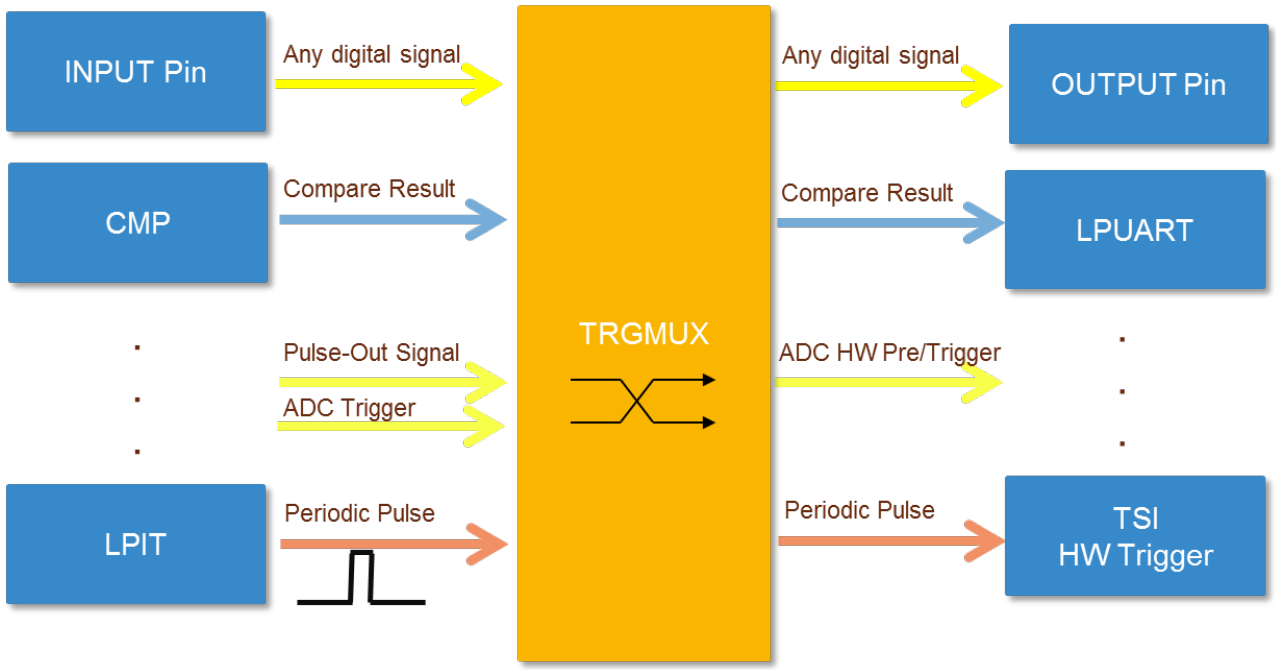

图 5. KE1xTRGMUX 触发方案

用于 KE1x TRGMUX 配置的 SDKv2.0 驱动程序/示例代码如下：

- 配置 MCU 外设的触发源输入。`index` 参数选择 TRGMUX 设备集合。`input` 参数选择触发输入通道。`trigger_src` 参数包含所有外围触发源选项；

```
status_t TRGMUX_SetTriggerSource(TRGMUX_Type *base, trgmux_device_t index,
                                                    trgmux_trigger_input_t input,
trgmux_source_t trigger_src);
```

4.2 系统级增强

为了提高安全性、性能和效率，KE1x 系列提供了比 KE0x 更先进的模块和特性，如闪存/SRAM ECC、I/D 缓存、MPU 和 FAC、MMDVSQ 和 eDMA，如 表 4 所示：

表 4. 系统级增强

特性	KE0xZ	KE1xZ	KE1xF
I/D cache	—	—	8 KB
Flash ECC	—	—	Yes
SRAM ECC	—	—	Yes
MPU	—	—	Yes
FAC	—	Yes	Yes
eDMA	—	8-channel	16-channel
MMDVSQ	—	Yes	—

4.2.1 Flash/SRAM ECC

- SRAM ECC — 具有 5 位 ECC 的 8 位数据，最多 1 位错误的检测和纠正，最多 2 位错误的检测以及支持 ECC 位的错误自检。
- Flash ECC — 具有 8 位 ECC 的 64 位数据，最多可检测和纠正 1 位错误，并支持 ECC 位的错误自检。

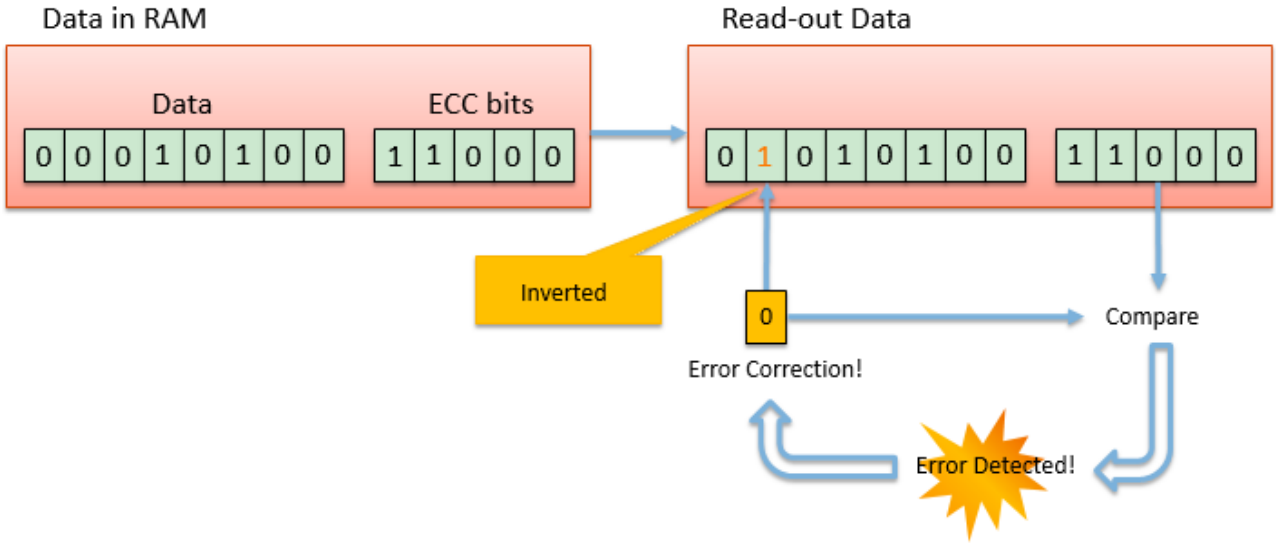

图 6. SRAMECC 检测与校正示例

4.2.2 缓存

KE1xF 包含一个 8 KB 代码高速缓存，以最大程度地减少存储器访问延迟对性能的影响。代码缓存位于 I/D 总线上，而系统总线上没有缓存。

缓存的特点是：

- 2 路设定关联。
- 4 字线。
- 线路可以单独更新。
- 可以一次刷新整个缓存。
- 通过奇偶校验缓存内存。

尽管 KE1xF 核心时钟可以运行到 168 MHz，但缓存不能以高于 25 MHz 的频率运行。系统带来一个 8 KB 的代码缓存，可以预取 CPU 的指令和数据，加速了 P-flash 数据的传输，提高了 CPU 的处理效率。

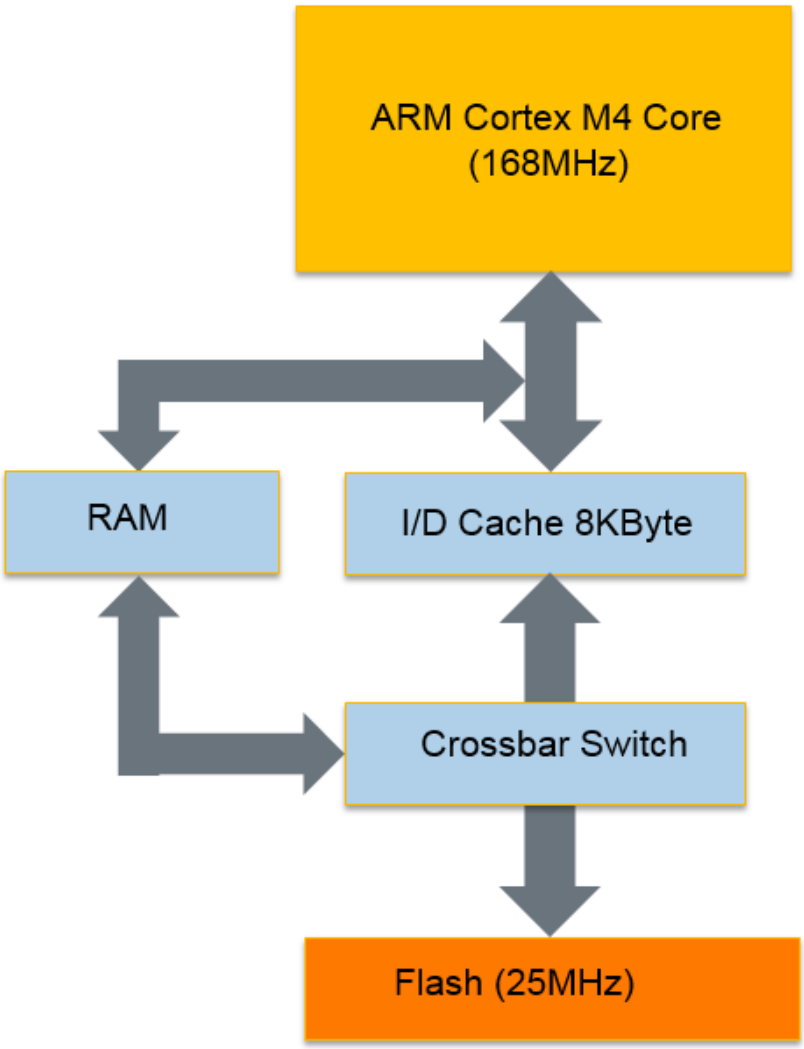

图 7. 缓存访问闪存的示例

4.2.3 MPU 和 FAC

KE1xF 上的 MPU 同时监视所有系统总线事务，并使用预定义的区域描述符（定义内存空间及其访问权限）评估它们的适用性。允许完成具有足够访问控制权限的内存引用，而未映射到任何区域描述符或权限不足的引用将以保护错误响应终止。MPU 的功能包括：

- 支持多达 8 个内存区域。
- 读取/写入/执行权限仲裁。
- 区域大小可能从 32 B 到 4 GB 不等。

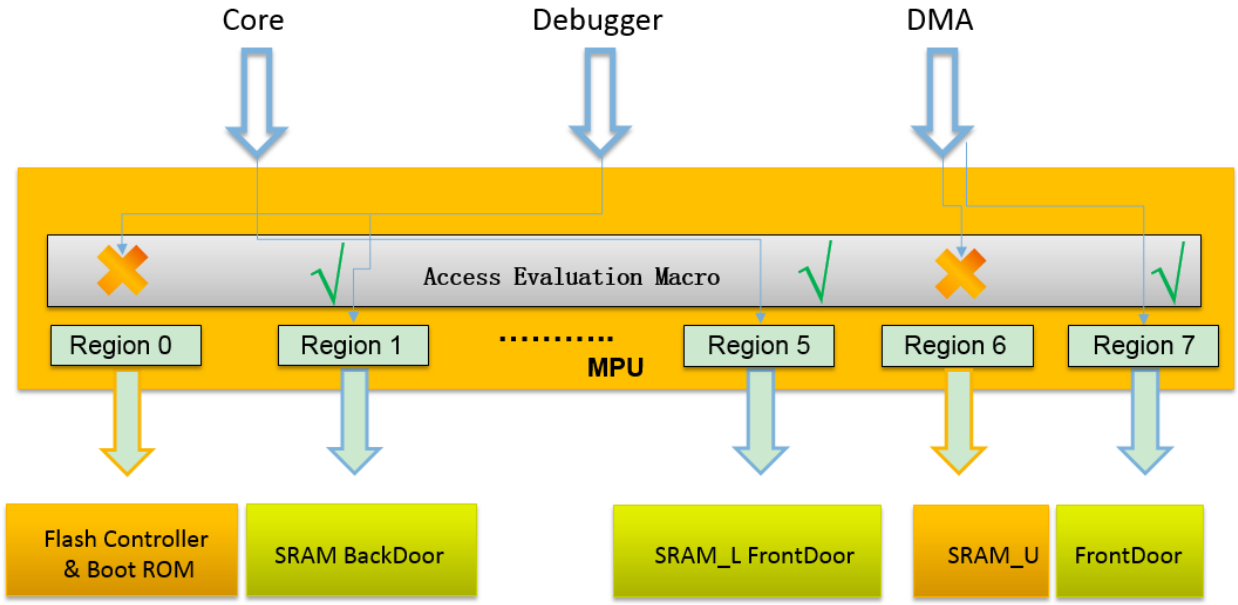

图 8. MPU 访问内存的示例

闪存访问控制（FAC）是一种本机或第三方可配置的内存保护方案，已优化为利用软件库，同时为这些库提供可编程限制。FAC 同时位于 KE1xF 和 KE1xZ 上，其主要功能包括：

- 可编程闪存分为相同大小；支持多达 64 段。
- 对访问情况进行逐周期评估。
- 不同的安全状态：
  - — 主管/特权安全状态 — 执行和修改。
  - — 中级状态 — 仅执行。
  - — 不安全状态 — 没有访问权。
- 您可以在 Program Once 区域中实现访问控制逻辑。
- 由 FXACC 和 FSACC 寄存器进行配置。

4.2.4 MMDVSQ 模块

实现 ARMv6-M 指令集架构的 Cortex-M 系列中的 Arm 处理器核心不包括对整数驱动操作的硬件支持。受影响的处理器包括 Cortex-M0+ 核心。然而，在某些深度嵌入的应用空间中，这类算术运算的硬件支持（以及无符号平方根函数）对于最大限度地提高系统性能和最小化设备功耗至关重要。

MMDVSQ 模块包含在选定的 MCU 中，作为一个内存映射的协同处理器，位于一个特殊的地址空间（在系统内存映射中），只有处理器核心才能访问。MMDVSQ 模块支持执行 ARMv7-M 指令集架构中定义的整数除法运算和无符号整数平方根运算。支持的整数除法操作包括 32/32 有符号（SDIV）和无符号（UDIV）计算。

MMDVSQ 模块仅包含在 KE1xZMCU 中，其关键特性包括：

- 支持 32/32 有符号和无符号除法（或余数）计算。
- 支持 32 位无符号平方根计算。
- 在运行数学密集型应用程序（如无传感器 PMSMFOC 算法）方面，拥有超过 25% 的性能改进。
- 简单的编程模型包括输入数据和结果寄存器以及控制/状态寄存器。

这类低端 MCU 的处理器核心和平台的通用框图如 图 9 所示。MMDVSQ 模块的位置（作为内存映射的协同处理器）以黄色突出显示。

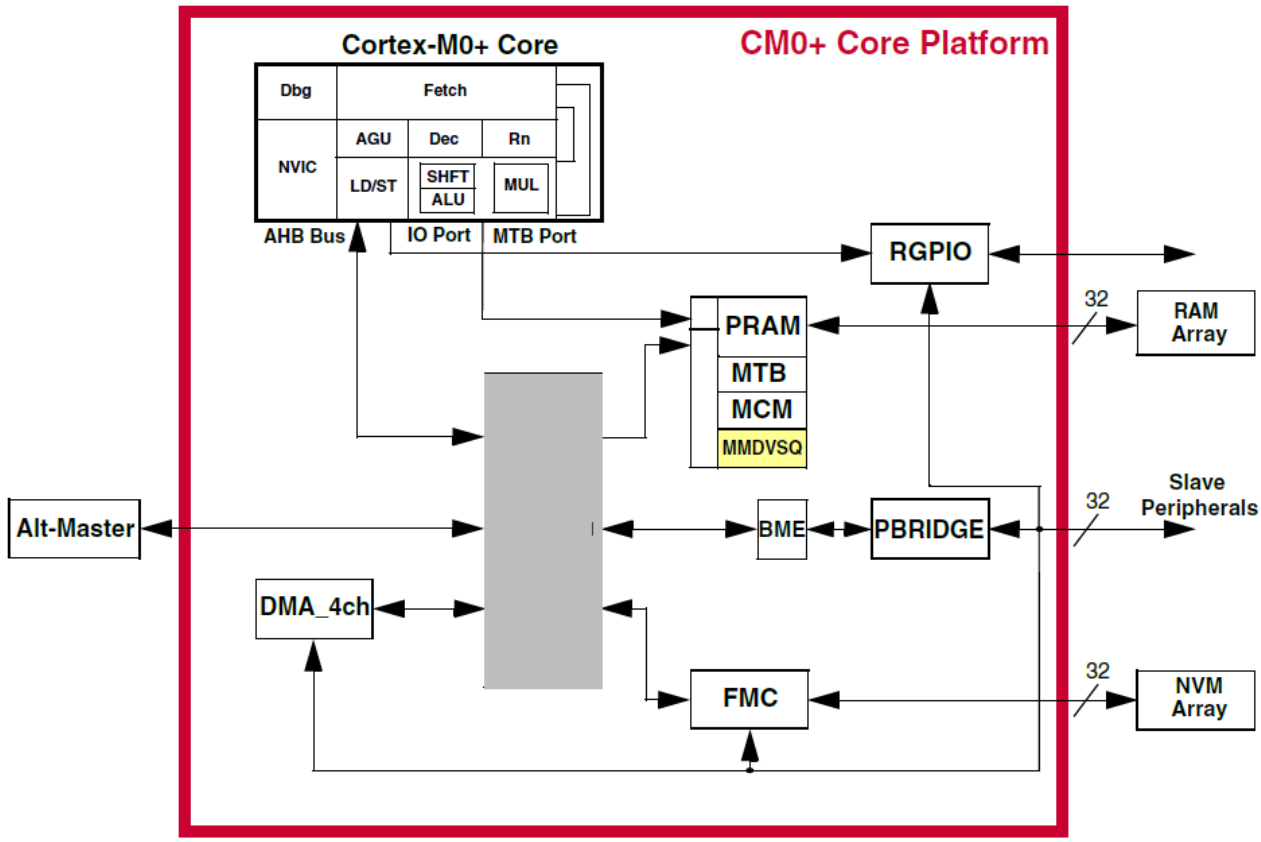

图 9. 通用 Cortex-M0+核心平台框图

4.2.5 eDMA

eDMA 是高度可编程的数据传输引擎，经过优化，可最大程度地减少主机处理器所需的任何干预。它用于静态地知道要传输的数据大小并且未在传输的数据本身中定义的应用中。eDMA 模块具有以下关键功能：

- 整个数据移动是通过双地址传输来完成的 — 从源读取，写入目的地。
  - — 可编程源和目的地地址和传输大小。
  - — 支持增强寻址模式。
- 16 通道（KE1xF）或 8 通道（KE1xZ）实现，在主机处理器的最小干预下执行复杂的数据传输。
- 传输控制描述符（TCD）的组织结构，可支持两个深度的嵌套传输操作。
  - — 存储在每个通道的本地内存中的 32 字节 TCD。
  - — 由小字节传输计数定义的内部数据传输循环。
  - — 由主要迭代计数定义的外部数据传输循环。
- 固定优先级和循环通道仲裁。
- 通过可选中断请求报告信道完成。
- 支持复杂的数据结构。

图 10 为 KE1xFDMAMUX/eDMA 方案。它包括 DMA 请求 mux，它允许多达 63 个 DMA 请求信号映射到 16 个 DMA 信道中的任何一个。 KE1xZ 有类似的方案，但 DMAMUX 通道数仅限于 8 个。

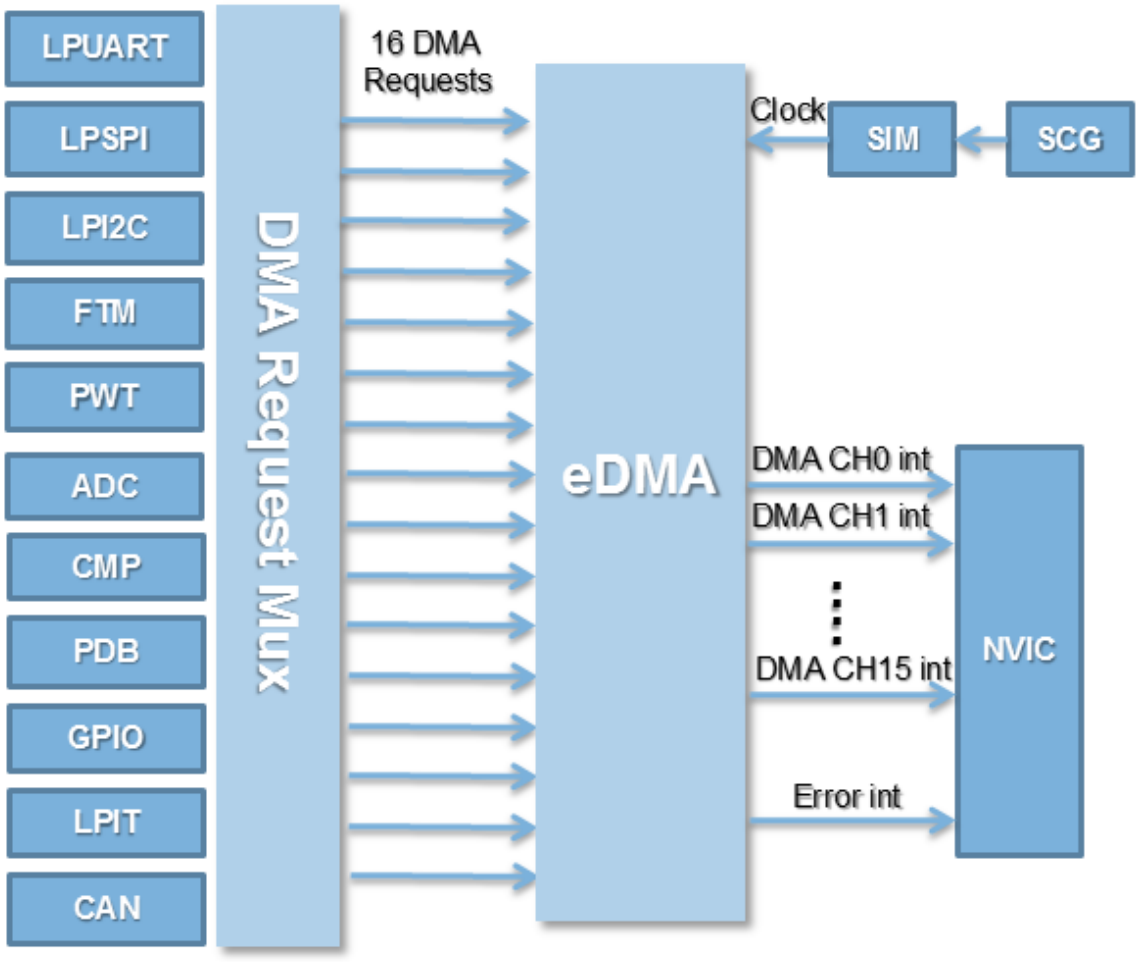

图 10. KE1xF DMAMUX/e DMA 方案

用于 KE1x DMAMUX 和 eDMA 特性的 SDKv2.0 驱动程序/示例代码：

- DMAMUX 初始化和启用/禁用。通道参数为 DMAMUX 通道号。

```
void DMAMUX_Init(DMAMUX_Type *base);
void DMAMUX_EnableChannel(DMAMUX_Type *base, uint32_t channel);
void DMAMUX_DisableChannel(DMAMUX_Type *base, uint32_t channel);
```

- DMAMUX 通道源配置。channel 参数为 DMAMUX 通道数字，source 参数指定用于触发 DMA 传输的外围源。

```
void DMAMUX_SetSource(DMAMUX_Type *base, uint32_t channel, uint8_t source);
```

- eDMA 初始化和传输配置。`config` 参数为 eDMA 全局配置结构。有时您可只使用它的默认设置。在调用 `EDMA_SetTransferConfig()`函数之前，准备好 `edma_transfer_config_t` 配置结构，并将其填充到所需的源和目标传输属性中。

```
void EDMA_Init(DMA_Type *base, const edma_config_t *config);
void EDMA_SetTransferConfig(DMA_Type *base, uint32_t channel, const edma_transfer_config_t
*config, edma_tcd_t *nextTcd);
```

- 电子 DMA 硬件通道请求启用/禁用。`channel` 参数为 DMAMUX 通道号。

```
void EDMA_EnableChannelRequest(DMA_Type *base, uint32_t channel);
void EDMA_DisableChannelRequest(DMA_Type *base, uint32_t channel);
```

4.3 外围设备改进

4.3.1 CAN 模块

KE1xF 最多集成两个片上 FlexCAN 模块。它是一种通信控制器，可根据 ISO 11898-1 标准和 CAN 2.0 B 协议规范实施 CAN 协议。图 11 所示的一般框图描述了 FlexCAN 模块中实现的主要子块，包括一个用于存储消息缓冲区的关联存储器，接收全局掩码寄存器，接收单个掩码寄存器，接收 FIFO 过滤器和接收 FIFO ID 过滤器。

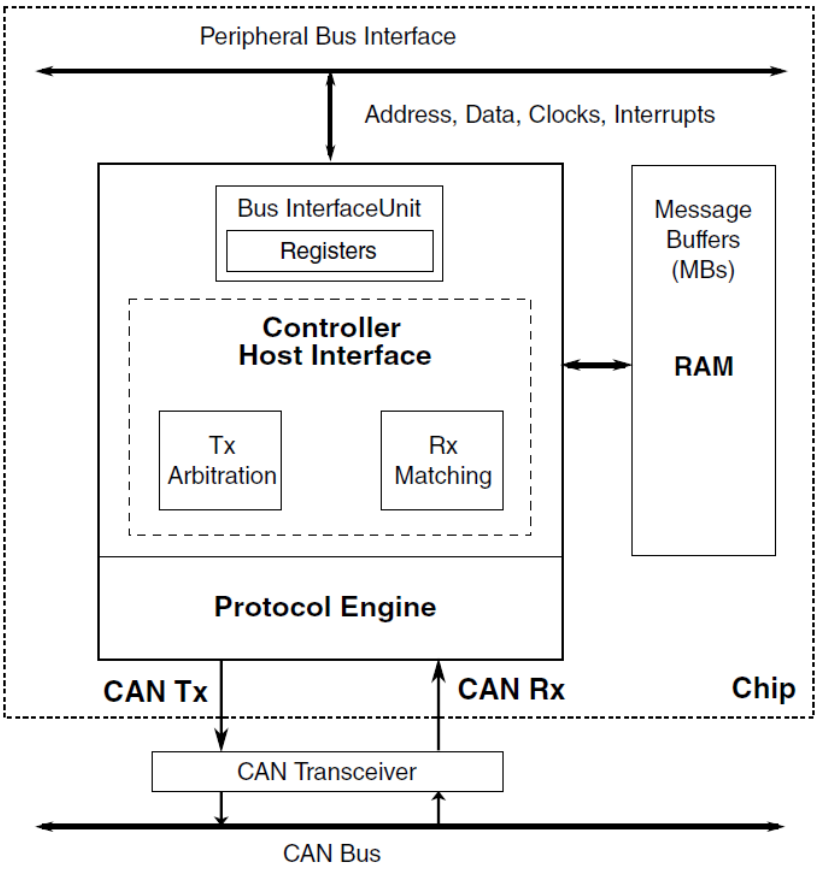

图 11. KE1xF Flex CAN 框图

Flex CAN 模块具有以下关键特性：

- 灵活的消息缓冲区（MB），总共有 16 个消息缓冲区（每个缓冲区长 8 个字节），它们可以配置为 RX 或 TX。
- 每个邮箱都可配置为 RX 或 TX，并支持标准消息和扩展消息。
- 灵活的邮箱（零到 8 字节长）。
- 支持在 VLPR 和 VLPW 模式下运行，并在总线活动上进行可编程唤醒。
- 基于带有可选外部时间刻度的 16 位自由运行计时器的时间戳。
- 比特时间计数。
- 支持只听模式。

- 非常好的编程模型。

在 KE06Z 系列中，MSCAN 被用作实现 CAN2.0A/B 协议的控制器，如 Bosch 规范（1991 年 9 月）所定义的那样）。MSCAN使用先进的缓冲区安排，导致可预测的实时行为和简化的应用软件。图 12 为 MSCAN 框图。

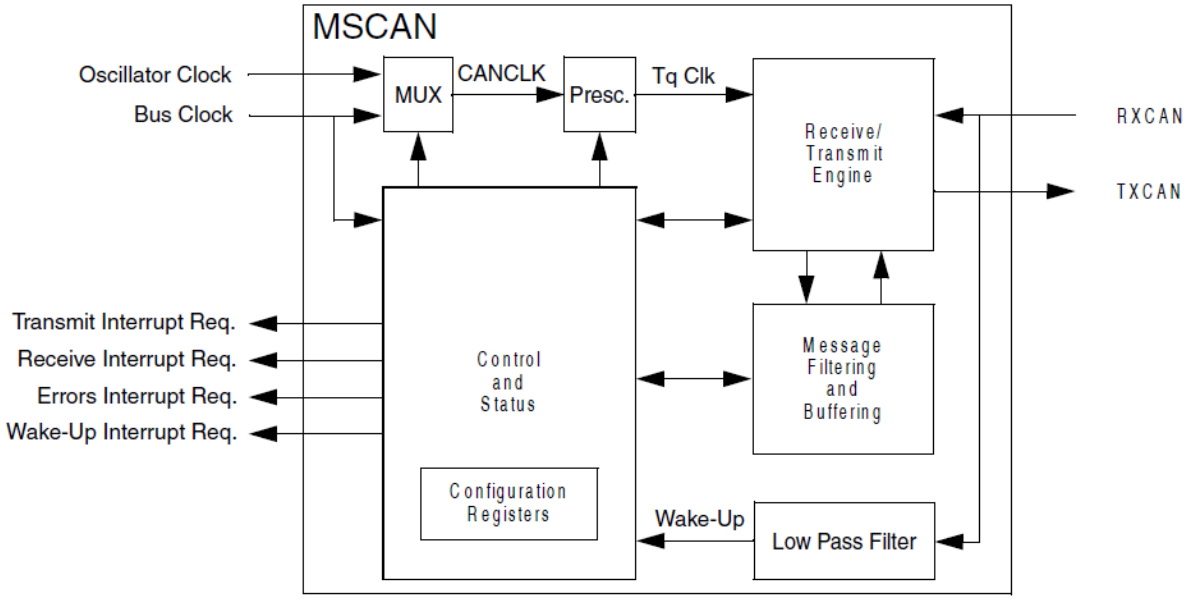

图 12. KE0xZ MSCAN 框图

FlexCAN 和 MSCAN 都完全实现了 CAN 协议规范，其结构相似、兼容。FlexCAN 的优点是它比 MSCAN 更灵活，包含更复杂的状态，并且可以自动响应远程帧（以及更多）。

4.3.2 人机界面（HMI）

除了一般的 I/O 端口之外，KE0x 系列还使用键盘中断（KBI）模块作为 HMI 接口。这个片上外围模块被称为键盘中断模块，因为它最初的设计是为了简化键盘开关的行列矩阵的连接和使用。这些输入还可用作额外的外部中断输入，以及将 MCU 从停止或等待低功耗模式唤醒的外部方法。

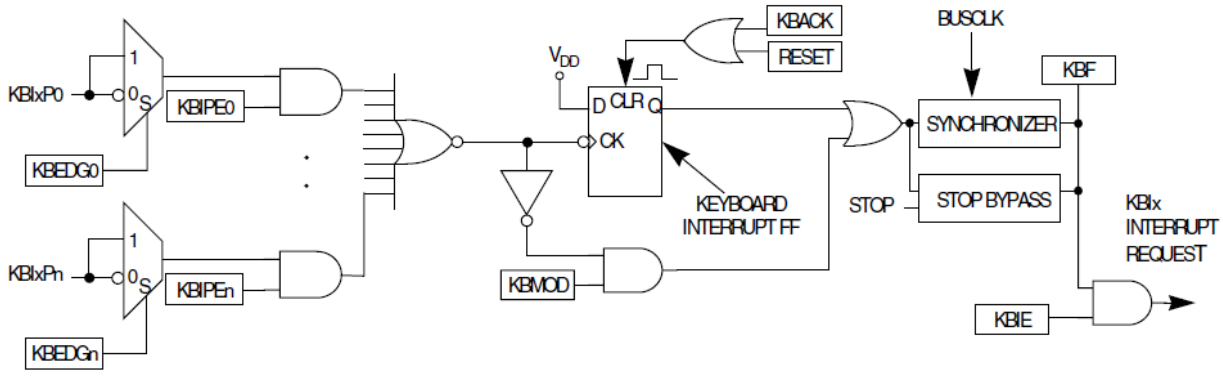

图 13. KE0x KBI 框图

KE1xZ 使用触摸感应接口（TSI）代替 KBI 作为人机界面。TSI 是许多消费者和工业领域的下一代人机界面。该 TSI 取代了传统的机械按钮/开关，这有助于使产品更可靠和时尚。由于触摸市场越来越大，NXP 在满足市场需求方面做了很多工作 — 更多的按键、更强的鲁棒性、更高的灵敏度、更少的软件干扰、更容易设计，而且没有额外的 BOM 成本。

该 TSI 提供电容式触摸传感器上的触摸传感检测。外部电容触摸传感器通常形成在 PCB 上，传感器电极通过设备中的 I/O 引脚连接到 TSI 输入通道。该 TSI 在开关集成模式下工作，以实现低功耗、高灵敏度和先进的 EMC 鲁棒性。它支持两种自电容（图 14）和互电容（图 15）传感器。在自电容模式下，TSI 只需要每个触摸传感器一个引脚。在互电容模式下，在各种 TX-RX配置中使用电容触摸矩阵进行传感。TSI 要求每个 TX 线一个引脚，每个 RX 线一个引脚。

该 TSI 完全支持 NXP 触摸传感软件（TSS）库，该库提供了一个强大的电容测量模块来实现触摸键盘、旋转和滑块功能。

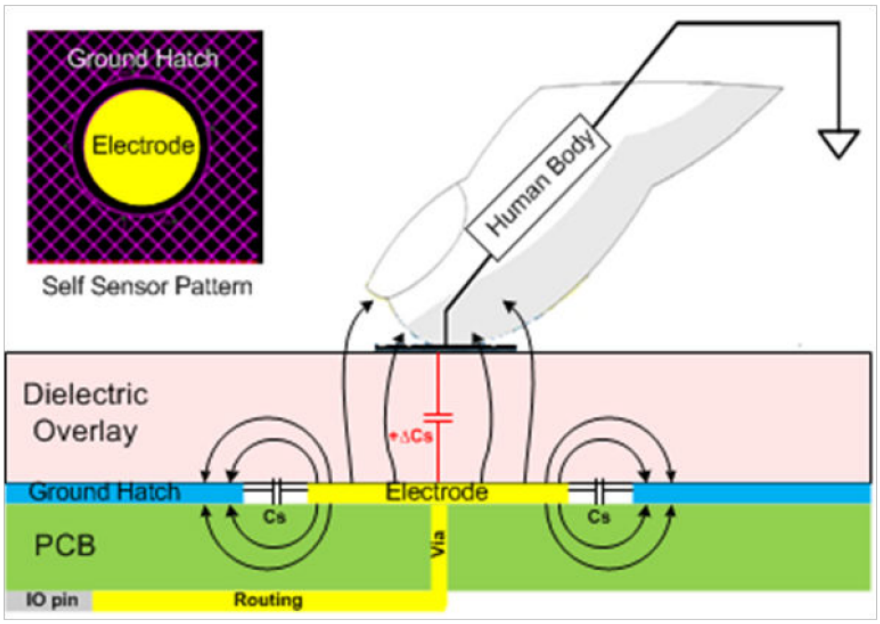

图 14. 自电容触摸传感器结构和电场

自电容式传感器结构如下：

- Cs：按键本身自电容。通常为 10 pF - 50 pF。
- ΔCs：触摸产生的自电容。通常为 0.3 pF - 2 pF。
- 传感器灵敏度：ΔCs/CS。通常为 1 % - 10 %。

本征性能取决于电极图案设计、覆盖层厚度/介电以及 PCB 走线。

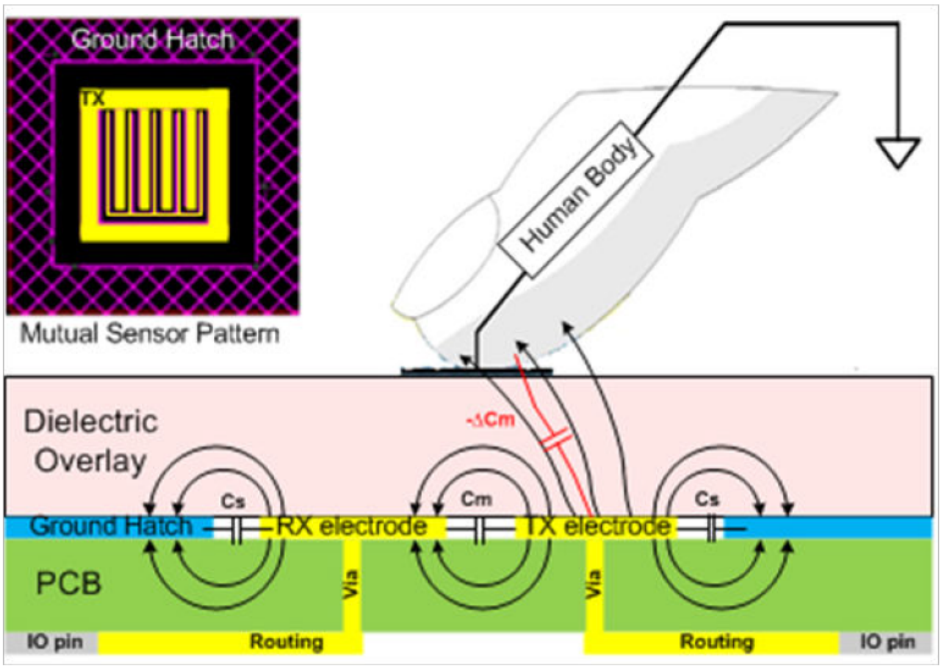

图 15. 互电容触摸传感器结构和电场

互电容传感器结构如下：

- Cm：按键本身的互电容。 通常为 2 pF - 10 pF。
- ΔCm：手指触摸导致的互电容变化量。通常为 0.3 pF - 2 pF。
- Cs：寄生自电容。通常为 10 pF - 50 pF。
- 传感器灵敏度：ΔCm/cm。 通常为 1 % - 20 %。

本征性能取决于电极图案设计、覆盖层厚度/介电以及 PCB 走线。

对于软件部分，SDKv2.0 提供全面的 TSI 模块驱动程序。NXP Touch 软件库的设计是为了加快 Touch 应用程序的开发，它可以作为二进制和源代码使用。

- TSI 自电容模式初始化函数。`config` 参数是自电容模式的配置结构。在调用此函数之前，用所需的电荷电流/电压、振荡器频率、灵敏度配置和其他属性填充它。

```
void TSI_InitSelfCapMode(TSI_Type *base, const tsi_selfCap_config_t *config)
```

- TSI 互电容模式初始化函数。`config` 参数是互电容模式的配置结构。在调用此函数之前，用所需的电荷电流/电压、振荡器频率、灵敏度配置和其他属性填充它。

```
void TSI_InitMutualCapMode(TSI_Type *base, const tsi_mutualCap_config_t *config)
```

- 按键校准功能。用于校准 TSI 空闲值。将其保存为 TSI 基线，以便将来进行按键检测。

```
void TSI_keyCalibrate(void);
```

- 按键检测功能。它代表了不同类型的信号处理算法。关键检测器算法的主要目的是确定电极是否被触摸，并计算归一化信号以与基线进行比较。`current_key_id` 参数表示指向按键通道的指针。函数返回值是指示“触摸”、“释放”等即时操作的关键事件。

```
uint8_t TSI_keyDetect(uint8_t*current_key_id) ;
```

4.3.3 通信接口

KE0x 和 KE1x 系列都包含丰富的通信接口，如 I2C，SPI 和 UART。 在 KE1x 上，它们被改进为在低功耗模式下工作，从而避免频繁唤醒 CPU 以进一步降低功耗。通信接口重新命名为 LPI2C，LPSPI 和 LPUART。如果他们使用的时钟保持启用，它们在Stop/VLPS 模式下是可以工作的。图 16 显示了 LPUART 的模块时钟。此示例图也适用于 LPSPI、LPI2C、Flex IO 和 LPIT。

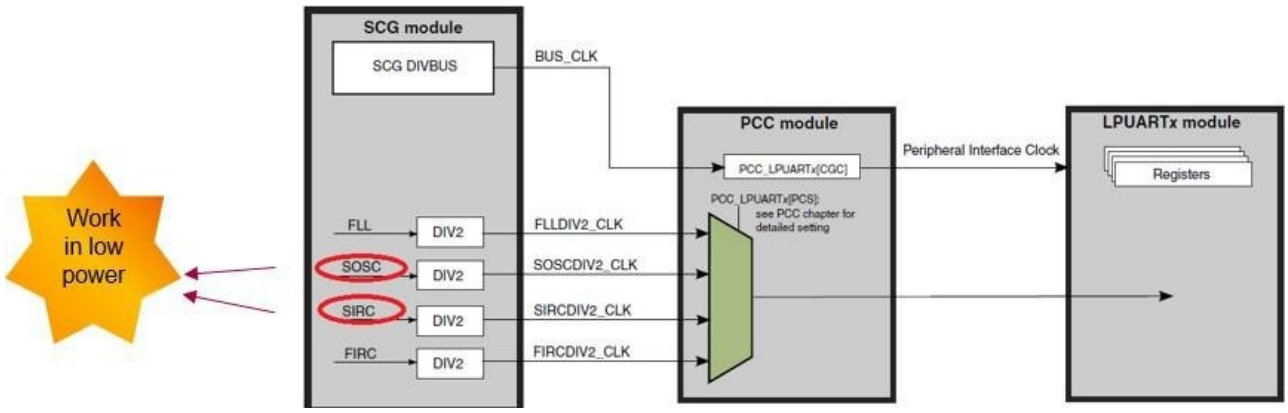

图 16. LPUART 的模块时钟

4.4 引脚兼容性

KE0x 和 KE1x 系列都具有多个封装和引脚选择，以适合不同的应用程序使用（见 图 17）。它们大多数都与不同系列引脚兼容，因此在移植时没有额外的硬件重新设计成本。

例如，KE06Z 和 KE18F 都具有 CAN 接口，因此应用领域可能相似，并且可以移植硬件和软件。由于它们之间的 64LQFP 封装不同，因此仅需注意两个引脚。KE06Z 上的引脚 9 用作 VREFL，但用作 KE18F 上的 VREFH。默认的 NMI 引脚位于 KE06Z 的引脚 19 上，但位于 KE18F 的引脚 45 上。其他 62 个引脚全部兼容，只有一些外设实例或通道发生更改，因此移植非常容易。

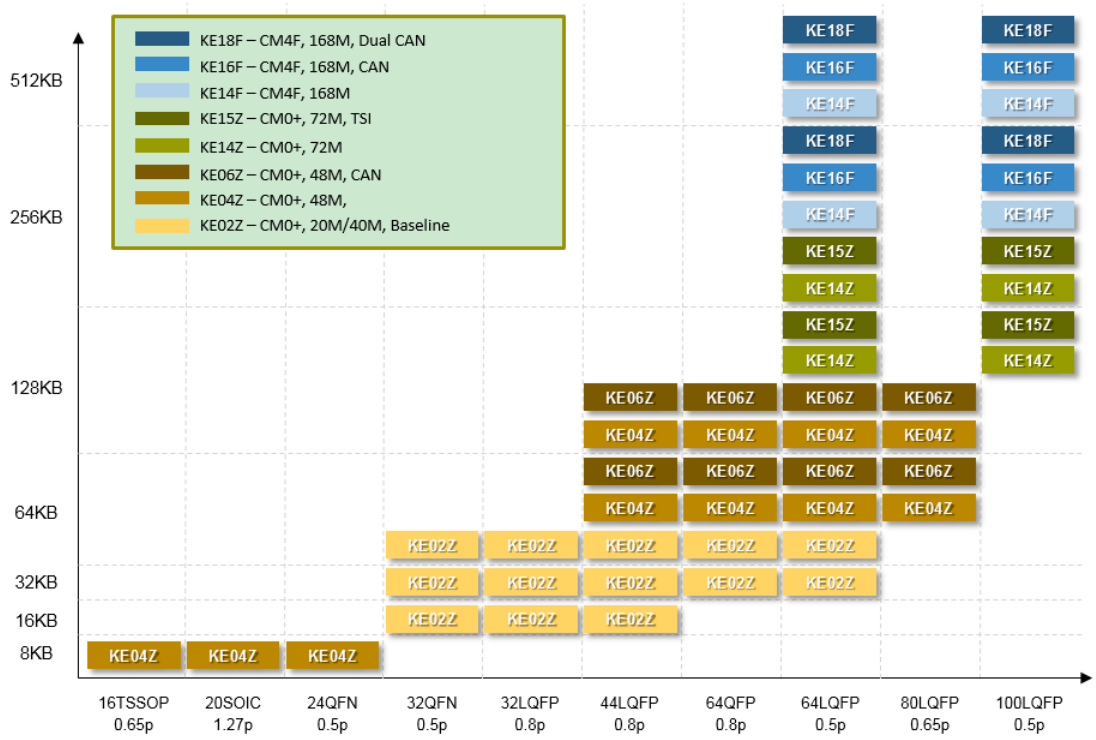

图 17. KE0x 和 KE1x 软件包选项图

5 结论

KE0x 系列是第一个采用可靠技术，5-V I / O 口和 ARM Cortex-M0 +内核的入门级 Kinetis MCU。 KE1x 是现有 KE0x MCU 系列的扩展，具有增强的 CPU 性能以及更多的存储器和外设。 本应用笔记概述了 KE0x 和 KE1x MCU 系列之间的主要区别和改进，并介绍了基本的 Kinetis SDK v2.0 软件包。 本文档可帮助您轻松地从 KE0x 系列移植到 KE1x 系列。

6 修订记录

表 5 概述了自最初发布以来对本文档所做的更改：

表 5. 修改记录

版本号	日期	说明
0	2016 年 9 月	初次版本

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 2016 年 9 月

Document identifier: AN5332

arm