

使用Kinetis K22F和KV31F MCU 的高速运行模式

本应用笔记针对最新一代Kinetis K22F和KV31F系列微控制器撰写，旨在解释该系列器件新运行模式的优点和用法，即高速运行模式 (HSRUN)。这些MCU基于120 MHz和100 MHz ARM® Cortex®-M4，具有浮点单元，HSRUN模式可发挥其最大性能，具有业界一流的CoreMark/MHz评分。

添加HSRUN模式的总体思路是，让系统设计人员可以配置K和V系列MCU，以实现最高频率，尽可能快速地执行计算任务，然后返回正常或低功率运行模式，从而节省电力。

内容

1	高速运行应用	2
2	使用高速运行模式	2
3	Kinetis SDK 示例应用	3
4	高速运行演示的关键源代码	7
5	结果和结论	9
6	参考资料	11
7	修订历史	11



1 高速运行应用

新的HSRUN模式非常适合计算密集型应用，包括传感器融合、能量计量和用于医疗保健器件的身体传感器数据。这些应用可能包括复杂的算法，需要各种函数，包括寄存器位操作、定点和浮点运算、大量的内存数据搬运等。实际上，执行压缩、信号编码/解码或数学转换的算法在HSRUN模式下可显著缩短执行时间，提高效率。

显然，这种性能提高是以电流消耗增加为代价的，因此工作频率升高将成比例地增加功耗。考虑使用HSRUN模式时，我们建议系统设计人员仔细考虑最高容许电流消耗、最高功耗和最终应用的工作温度。

本文最后将提供演示代码，基于Kinetis SDK平台。后者执行多重浮点快速傅立叶转换 (FFT)，作为CPU负载测试用例。此实际示例执行时间和功耗的比较数据将与分析技术一起共享。

2 使用高速运行模式

在HSRUN模式下，片上电压调节器保持运行调节状态，同时启用更高电流容量模式。与正常RUN模式相比，在此状态下，MCU可以更高频率运行。器件参考手册中的“电源管理”一章介绍了最高准许频率。

处于此模式时，必须遵守以下限制：

- 系统、总线、Flash 或内核时钟的最大容许频率变化限制为两倍。
- 退出 HSRUN 模式前，时钟频率应减小到 RUN 模式可接受的值。
- HSRUN 模式下，不能直接进入 STOP 模式。
- 禁止修改模块的时钟门控位。
- 不允许进行 Flash 编程/擦除。

要进入HSRUN模式，SMC_PMPROT寄存器必须首先设置为允许HSRUN模式。该保护功能旨在防止意外进入不受支持的运行模式。配置保护后，只需将SMC_PMCTRL [RUNM] 设置为HSRUN值即可。提高时钟频率前，应轮询SMC_PMSTAT寄存器以决定系统何时完成进入HSRUN模式。要重新进入正常RUN模式，只需清除SMC_PMCTRL [RUNM]。在MCU退出其复位流后，任何复位也将清除RUNM并导致系统退出正常RUN模式。这些细节旨在解释器件寄存器的工作原理，但应注意，Kinetis SDK API完全支持这些字段的设置。本文将在下一节讨论。

表1详述了RUN和HSRUN模式最常见的内部时钟设置，如器件参考手册所述。

表1. 常见内部时钟设置

时钟	支持120 MHz的器件		支持100 MHz的器件	
	RUN模式	RUN模式	RUN模式	RUN模式
内核时钟	80 MHz	120 MHz	72 MHz	100 MHz
系统时钟	80 MHz	120 MHz	72 MHz	100 MHz
总线时钟	40 MHz	60 MHz	36 MHz	50 MHz
FlexBus时钟	20 MHz	30 MHz	NA	NA

3 Kinetic SDK示例应用

Kinetic SDK (KSDK) 是一款软件开发套件，可为具有Cortex®-M内核的飞思卡尔Kinetic器件提供全面的内核与外设软件支持。KSDK包括为每个外设提供的硬件抽象层 (HAL) 以及基于HAL的外设驱动程序。演示应用和驱动程序示例旨在演示驱动程序和HAL的用途以及突出显示目标SoC的主要功能。

3.1 高速运行演示应用的说明

以下介绍的“highspeed_run_demo”是一个简单的基于shell的应用，允许在RUN与HSRUN模式间安全切换，另外包括CPU负载测试用例，它采用自定义复杂32位浮点FFT算法形式。此FFT是时间抽取算法，运算复杂性超过ARM CMSIS-DSP arm_cfft_f32() 函数，因此下文提及的运行时间不应视为任何类型的基准。演示负载测试在真实和虚拟数据数组上执行512-bin复杂FFT，每个数组包含16 Kb数据，重复N次。目的是使用具有较长运行时间的负载测试来突出HSRUN模式在相同测试用例中相对于RUN模式的优势。

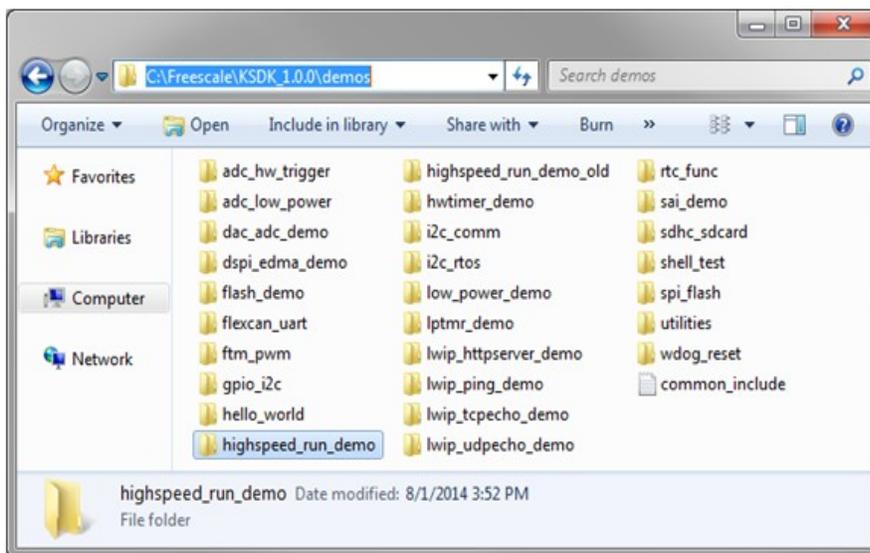
3.2 安装高速运行演示 demo

本节中提供安装和测试演示应用所需的步骤，最新KSDK V1.0 – RCS版本中称为“highspeed_run_demo”。

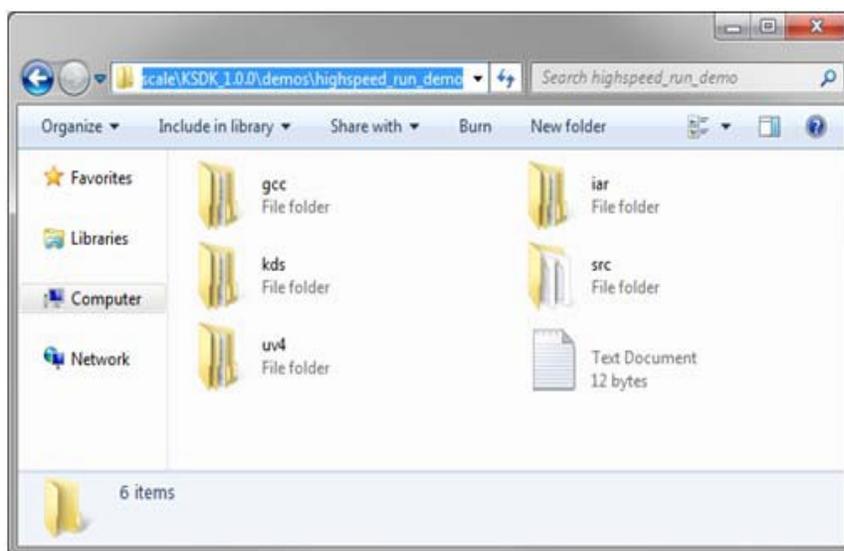
安装、构建和运行HSRUN演示的步骤：

1. 下载 KSDK V1.0 的最新版本：
http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=KINETIS_SDK
2. 启动下载的可执行文件，浏览 SDK 的安装对话。SDK 应安装在 <Install_dir>，类似于：
“C:\Freescale\KSDK_1.0.0”
3. 在以下位置下载“highspeed_run_demo.zip”文件：
http://cache.freescale.com/files/microcontrollers/doc/app_note/AN4985sw.zip?fpssp=1

4. 将提取自.zip的“highspeed_run_demo”文件夹拖放至以下位置：“C:\<Install_dir>\demos”



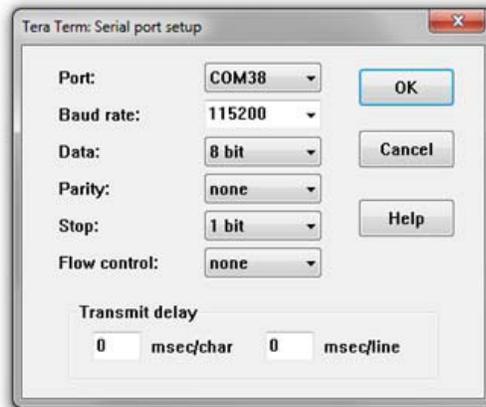
5. 在“highspeed_run_demo”文件夹内，可找到用于被支持工具链的预构建项目文件夹（IAR、Keil、ARM Gcc 或 Kinetis Design Studio）。



在每个工具链的文件夹内，可找到“frdmk22f120m”和“twrk22f120m”，其分别对应于K22F器件的FRD和TWR电路板。为所选工具链打开适当的文件夹，然后打开适合特定设置的电路板类型的文件夹，以找到IDE项目。

6. 首先，为 KSDK 库构建平台项目，然后构建对应于所用工具链的“highspeed_run_demo”项目。有关每个工具链的详细构建程序，请查看 Kinetis SDK K22 用户指南 (KSDKK22UG)。

7. 打开终端应用（本例中显示为 Tera Term）：
<http://tssh2.sourceforge.jp/index.html.en>），然后为具有以下设置的适当目标电路板配置虚拟 COM 端口连接：



8. 切换回至 IDE 窗口，启动调试器，将代码编程至 FRDM-K22F120M 或 TWR-K22F120M 目标，并运行应用。
9. 当“highspeed_run_demo”执行时，应出现如下所示的终端打印输出：

```

COM38:115200baud - Tera Term VT
File Edit Setup Control Window Help
External Pin Reset
K61
64pin
Silicon rev 1.1
512 kBytes of P-flash
128 kBytes of RAM

-----*
**          Kinetis High Speed Run DEMO          **
**          Jul 29 2014 22:12:31                  **
-----*
in HS Run Mode ? in PEE mode now at 80000000 Hz

Select the desired operation:
0 for CASE 0: Enter High Speed Run (HSRun) @ 120MHz
1 for CASE 1: Enter Normal Run @ 80MHz
2 for CASE 2: Execute CPU load test at current speed

-----*
>> □

```

10. 终端应用具有三个选项：
- 用任一按键按下“0”，应用会将器件置于 120 MHz 的 HSRUN 模式。

```

COM38:115200baud - Tera Term VT
File Edit Setup Control Window Help
-----*-----
>> Press any key to enter HSRun
    in HS Run Mode !   in PEE mode now at 120000000 Hz

Select the desired operation:
0 for CASE 0: Enter High Speed Run (HSRun) @ 120MHz
1 for CASE 1: Enter Normal Run           @ 80MHz
2 for CASE 2: Execute CPU load test at current speed

-----*-----
>> █

```

- 用任一按键按下“1”，应用会将器件置于 80 MHz 的正常 RUN 模式。

```

COM38:115200baud - Tera Term VT
File Edit Setup Control Window Help
-----*-----
>> Press any key to enter Normal Run
    in Run Mode !   in PEE mode now at 80000000 Hz

Select the desired operation:
0 for CASE 0: Enter High Speed Run (HSRun) @ 120MHz
1 for CASE 1: Enter Normal Run           @ 80MHz
2 for CASE 2: Execute CPU load test at current speed

-----*-----
>> █

```

- 按下“2”将以正弦波执行 2048 点 FFT 的 50 次迭代，该正弦波已载入器件 SRAM 存储器。FFT-CPU 负载测试的运行时间将打印至终端窗口。

```

COM38:115200baud - Tera Term VT
File Edit Setup Control Window Help
-----*-----
>>
* Start FFT *
* FFT Done *

Start Time = 00:03:36.57
Stop Time  = 00:03:38.06
FFT Runtime in Total Seconds = 1.49

    in Run Mode !   in PEE mode now at 80000000 Hz

Select the desired operation:
0 for CASE 0: Enter High Speed Run (HSRun) @ 120MHz
1 for CASE 1: Enter Normal Run           @ 80MHz
2 for CASE 2: Execute CPU load test at current speed

-----*-----
>> █

```

11. 在各运行模式（HSRUN 和 RUN）间切换，然后为各模式执行 CPU 负载测试，查看运行时间差异。HSRUN 模式 120 MHz 下，50 次 FFT 的典型结果约为 1 秒，RUN 模式 80 MHz 下，相同的 50 次 FFT 约需要 1.5 秒。完成测试比 HSRUN 模式快 50%，符合预期。

4 高速运行演示的关键源代码

在“highspeed_run_demo”的根目录下，./src目录包括该演示需要的所有特定演示源代码。配置 HSRUN 模式器件所需的数据结果和函数调用可在“highspeed_run_demo.c”的“hsrun_power_modes_test()”函数内找到。

该函数包括调用系统模式控制器平台库结构和 API，可在 KSDK 内找到。以下大部分内容定义在“fsl_smc_hal.c/h”文件内，位于：<Install_dir>/platform/hal/smc/

所有 KSDK API 的详细文档提供在 Kinetis SDK API 参考手册 (KSDKAPIRM) 中。

变量说明一节中，“hsrun_power_modes_test()”顶部（可在“highspeed_run_demo.c”内找到），定义了以下两个数据结构。这些结构例示了运行模式配置结构和功耗模式保护结构：

```
/* 说明功耗模式配置结构 */
    smc_power_mode_config_t smcConfig;

/* 说明 HSRUN 的功耗模式保护 */
    smc_power_mode_protection_config_t pmodes = {

        .vlpProt    = false,

        .llsProt    = false,

        .vllsProt   = false,

        .hsrunProt  = true

    };
```

上述结构稍后将由代码内的 SMC HAL 函数 SMC_HAL_SetRunMode() 和 SMC_HAL_SetProtection() 使用。第一步是设置 HSRUN 模式保护。此函数设置 SMC_PMPROT 寄存器内的适当位值，以支持 HSRUN 模式。

```
/* 配置应用的功耗模式保护设置*/
    SMC_HAL_SetProtection(SMC_BASE, &pmodes);
```

下一个可用的SMC HAL函数为SMC_HAL_GetStat() 函数，其用于以下if() 语句，以决定和打印出器件当前运行模式，同时测试函数的主要while() 环路为：

```
if(SMC_HAL_GetStat(SMC_BASE) == kStatRun) {
    printf(" in Run Mode ! ");
} else if(SMC_HAL_GetStat(SMC_BASE) == kStatHsrRun)
{ printf(" in HS Run Mode ! ");
}
```

在RUN与HSRUN模式间执行实际切换的源代码可在“switch(testNum){}”语句中找到。在“case 0”语句中，可发现将器件置于HSRUN模式的语句。

CLOCK_HAL_SetOutDividers() 函数同时配置所有时钟输出分频器的设置。有关支持的时钟分频器值范围，参见器件参考手册。

```
/*HSRUN 模式的设置分频器 */
/* core/system=120(0), bus=60(1), FlexBus=30(3), flash=24(4) [MHz]*/
CLOCK_HAL_SetOutDividers(SIM_BASE, 0,1,3,4);
```

SMC_HAL_SetRunMode() 函数将SMC_PMCTRL[RUNM] 寄存器设置为HSRUN模式。

```
/* 转至 HSRUN*/
/* 将功耗模式设置为 HS Run 模式 */
SMC_HAL_SetRunMode(SMC_BASE, kSmcHsrRun);
```

接着轮询SMC_PMSTAT寄存器，以确认是否成功进入HSRUN模式：

```
/* 轮询 PMSTAT，直至已进入 RUN 模式 */
while(SMC_HAL_GetStat(SMC_BASE) && kStatHsrRun != kStatHsrRun)
{ }
```

器件配置为在演示应用初始化过程中以PEE模式工作，因此必须将MCG模式状态机转换回至FBE模式。接下来，设置PLL VCO输出的PLL参考分频器 (PRDIV) 和分频比 (VDIV)，使其对应于HSRUN模式的有效设置。然后在120 MHzHSRUN模式下返回PEE模式。

```

/* 配置和启用 HSRUN 模式的时钟 */
pee_pbe (CLK0_FREQ_HZ);
pbe_fbe (CLK0_FREQ_HZ);
fbe_pbe (CLK0_FREQ_HZ, PLL0_PRDIV_HS, PLL0_VDIV_HS);
mcgClkKHz = pbe_pee (CLK0_FREQ_HZ);

```

器件现在应在120 MHz、PEE和HSRUN下工作，终端提示时将确认。或者，如果检查“case 1”语句，退出HSRUN模式和进入正常RUN模式所需的步骤和序列如下所示：

```

/* 配置和启用 RUN 模式的时钟 */
pee_pbe (CLK0_FREQ_HZ);
pbe_fbe (CLK0_FREQ_HZ);
fbe_pbe (CLK0_FREQ_HZ, PLL0_PRDIV, PLL0_VDIV);
mcgClkKHz = pbe_pee (CLK0_FREQ_HZ);

/*RUN 模式的设置分频器 */
/* core/system=80(0), bus=40(1), FlexBus=20(3), flash=26.67(2) [MHz]*/
CLOCK_HAL_SetOutDividers (SIM_BASE, 0,1,3,2);

/* 目前处于 HS Run。退出 HS Run */
SMC_HAL_SetRunMode (SMC_BASE, kSmcRun);
while (SMC_HAL_GetStat (SMC_BASE) && kStatRun != kStatRun)
{}

```

“case 2”语句包含RTC跑表功能，以测量运行时间，以及执行N次（默认代码设置为50）复杂浮点FFT的代码。

5 结果和结论

本应用笔记介绍了新Kinetis K22F和KV31F器件系列支持的HSRUN模式。分享并详细介绍了演示应用“highspeed_run_demo”。为进一步详述HSRUN模式奠定了基础，本最终章节内分享的数据采集demo可以运行在TWR-K22F120M板上，也可以在FRDM-K22F120M板上运行。

在demo中，J15被换成10欧姆的电阻，这是为了监控HSRUN演示在执行期间的电流消耗。带有差分探头的示波器用于测量10Ω寄存器两端的电压变化。此设置用于测量和记录FFT负载测试在HSRUN和正常RUN模式下的瞬态电流消耗。

图1显示简单FFT迭代在HSRUN@120MHz和RUN@80MHz下的电流分布对比。

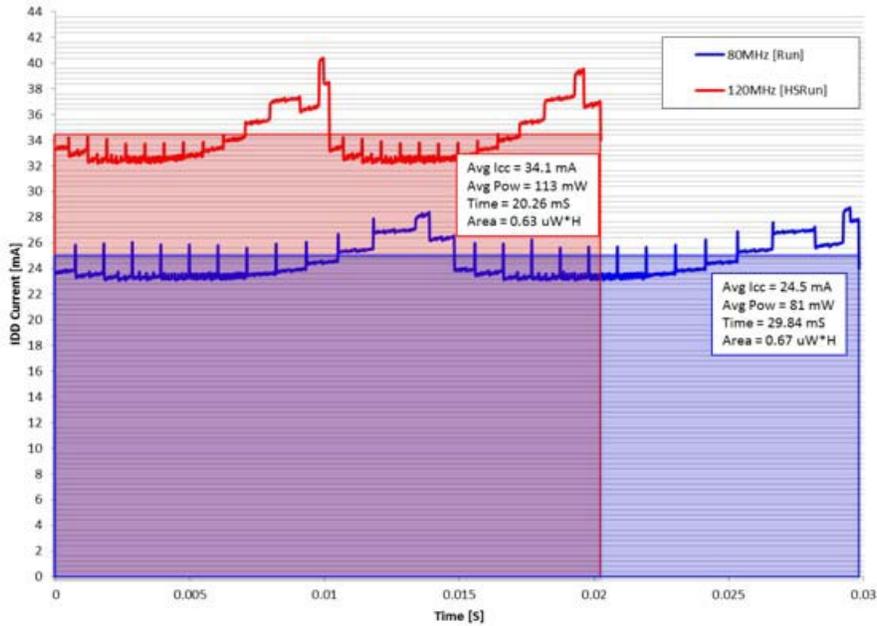


图1. 简单FFT迭代；电流与时间对比

图1中的关键信息是每个运行模式曲线下方面积。该面积对应于在每个运行模式下完成相同任务所需的能量。在HSRUN模式120 MHz下执行的FFT需要的能量比正常RUN模式少0.04 uWh。演示经修改后在HSRUN和正常RUN模式下执行200次复杂FFT迭代，并编译成Flash和SRAM方式运行。该实验的结果可在表2中找到。

表2. 以HSRUN模式执行的FFT的结果

代码	模式	时钟	平均电流 [mA]	功率 [W]	差值电流 [mA]	200 FFT			
						总运行时间 [s]	性能改进	能量 [mWH]	节能
Flash	HSRUN	120 MHz	34.1	0.11253	9.6	4.1	47%	0.0128	6%
	RUN	80 MHz	24.5	0.08085		6.04		0.0136	
SRAM	HSRUN	120 MHz	29.6	0.09768	8	3.92	50%	0.0106	9%
	RUN	80 MHz	21.6	0.07128		5.88		0.0116	

数据显示，HSRUN模式平均消耗电流多8至9.6 mA，但运行时间方面提供47%至50%的性能改进，具体取决于是从Flash还是SRAM执行。不过，能量和节能方面具有明显优势柱状图，显示以HSRUN模式运行的MCU完成FFT负载测试的效率高6%至9%。

6 参考资料

Kinetis K22F 数据手册 (K22P121M120SF8)

Kinetis K22F 参考手册 (K22P121M120SF7RM)

Kinetis SDK API 参考手册 Kinetis SDK K22 用户指南

ARM CMSIS-DSP 参考资料:

http://www.keil.com/pack/doc/cmsis/dsp/html/group__complex_f_f_t.html

Tera Term 应用链接: <http://tssh2.sourceforge.jp/index.html.en>

7 修订历史

修订号	日期	重大变更
0	09/2014	初始版本

How to Reach Us:

Home Page:

freescale.com

Web Support:

freescale.com/support

本文档中的信息仅供系统和软件实施方使用Freescale产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。Freescale的数据表和/或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括“经典值”在内，必须经由客户的技术专家对每个客户的应用程序进行验证。Freescale未转让与其专利权及其他权利相关的许可。Freescale销售产品时遵循以下网址中包含的标准销售条款和条件：

freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. ARM and Cortex are the registered trademarks of ARM Limited. All other product or service names are the property of their respective owners.

© 2014 Freescale Semiconductor, Inc.

© 2014 飞思卡尔半导体有限公司

