

Kinetis-M RTC 晶振时钟频率补偿

作者: Ashish Sharma

1 简介

Freescale Kinetis M 系列微控制器基于 ARM® Cortex®-M0+内核，包含一个运行于外部 32 kHz 晶振的 RTC 外设。该时钟源具有一定的误差。首先，每个晶振都具有一定的偏移误差，因此室温下该时钟不是准确的 32.768 kHz。此外，晶振会随着温度出现偏差。因此，需要采用一个涉及偏移误差和环境温度的修正机制。

Kinetis M 控制器中有一个可用于实现该补偿，并受 RTC 外设支持的机制。本文档将详细说明问题以及如何实现晶振误差补偿。

2 RTC 主要特性

- 完全由电池供电
- 没有电源开关
 - 仅当拆卸或重装电池时 RTC 才发生复位
 - RTC 不受 MCU 复位影响
- 日期，时，分，秒都有各自独立的计数器
- 支持日历功能 — 年，月和星期都有各自独立的计数器
 - 根据用户定义参数自动调整夏令时
 - 自动月和闰年调整
- 执行时钟补偿以修正 32.768 kHz 晶振的频移
 - 粗调补偿 — 将误差修正至 0.119 PPM 并维持精确时间 (小于 5 PPM 精度)

内容

1	简介.....	1
2	RTC 主要特性.....	1
3	如何补偿晶振时钟频率误差.....	2
3.1	粗调补偿.....	3
3.2	微调补偿.....	3
3.3	由温度产生的 PPM 误差及其修正.....	4
3.4	通过 SAR ADC 测量温度.....	4
3.5	根据温度的 PPM 误差计算.....	5
4	示例.....	6
5	修订历史记录.....	8



- 微调补偿 — 产生精确的 (5 PPM 精度) 1 Hz 时钟输出, 提供 0.88 PPM 分辨率
- 硬件补偿可在所有模式下运行 — 无需 CPU 一直处于活动状态
- 带中断的可编程闹钟
 - 闹钟信号可通过 XBAR 从 MCU 输出
- 提供寄存器写保护机制以防止代码失控
- 32 字节的通用 RAM
- 大约 1.3 μA 的电池消耗
- 篡改(Tamper)
 - 增强型篡改检测
 - 检测对于系统的非法访问
 - 主动式和被动式篡改检测
 - 带时间戳的篡改事件队列
 - 在一个队列中最多存储 4 个篡改事件信息条目
 - 对于相同的篡改事件, 在 CPU 应答前不会更新篡改状态
 - 篡改源
 - 最多 3 个外部篡改检测引脚
 - 一个针对电池拆卸的内部篡改检测事件
 - 可选极性 (高电平或低电平有效)
 - 可单独启用或禁用篡改中断
 - 无论中断是否使能, 篡改状态都会被保存下来
 - RTC 可过滤外部篡改源的噪声和毛刺
 - 可通过选择时钟源和滤波持续时间来配置滤波器宽度
 - 不会对内部篡改事件滤波
- 闹钟
 - 周期性中断
 - 1 天、1 小时、1 分钟、512 Hz 至 2 Hz 和 1 Hz 中断
- 异步中断可将 MCU 从低功耗模式唤醒

3 如何补偿晶振时钟频率误差

补偿晶振时钟频率误差需要执行一些基本步骤。首先, 必须牢记温度变化对于晶振频率特性的影响, 如图 1 所示。

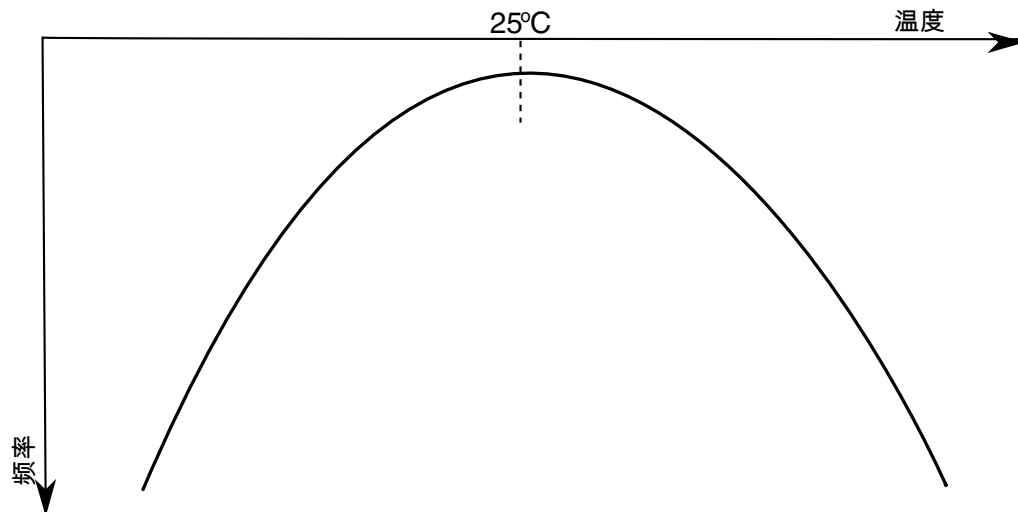


图 1. 典型温度与晶振频率特性

晶振误差修正的基本步骤:

1. 读 ADC 通道以获取环境温度。

2. 从公式（将在后文进行解释）通过插值函数获取该温度的 PPM 误差。
3. 添加偏移的 PPM 误差。
4. 获取修正系数。
5. 更新 RTC 寄存器。

Kinetis-M RTC 具有两种补偿模式：

1. 粗调
2. 微调

在粗调补偿中，可仅以一个 RTC 时钟步长进行补偿。请记住，RTC 时钟的驱动频率为 32.768 kHz。当然，可以将修正应用于多个秒计数 — 例如，在 5 秒内增加 18 个时钟，即每秒 3.6 个时钟修正。这样即可对时钟小数部分进行补偿。不过，每秒的补偿不是平均分布的；由于修正存在间隔，一些秒比另一些要“长”。

在微调补偿中，补偿以 4 MHz IRC 计数进行。4 MHz 约等于 32 kHz 乘以 128，因此会提供额外 1/128 的分辨率。

示例：

如果实际的 32k 时钟 = 32769.146 Hz (35 PPM 误差)

理想的 32k 时钟 = 32768 Hz

补偿参数：

要补偿的误差 = $(35/1000000) \times 32768 = 1.14688$

3.1 粗调补偿

误差约为 7 个计数，要在 6 秒内进行调整。这可以通过误差的小数部分(0.14688)进行计算，然后找到一个接近整数的倍数。最简单的方法是尝试计算 $1/0.14688$ 得出 6，并每次加上完整误差。

在 RTC_COMPEN 寄存器中：

设置：

RTC_COMPEN[7:0]为-7 或 0xF9 (带符号的二进制补码)

RTC_COMPEN[15:8]为 6

3.2 微调补偿

将小数部分乘以 128 得出 18.8 (四舍五入后为 19)。

在 RTC_COMPEN 寄存器中：

设置：

RTC_COMPEN[15:12]为-1 或 0x0F (带符号的二进制补码)

RTC_COMPEN[11:7]为 0 (保留)

RTC_COMPEN[6:0]为 19 (启用微调补偿后始终为正)

表 1 显示了按秒修正。当每秒向累加器计入 4 MHz 时钟计数时，将生成微调每秒输出。累加器在 128 处计满返回。

表 1. 累加器数值

累加器数值(n)	19	38	57	76	95	114	5	24	43	62
----------	----	----	----	----	----	-----	---	----	----	----

下一页继续介绍此表...

表 1. 累加器数值 (继续)

粗调加法器	0	0	0	0	0	0	1	0	0	0
粗调补偿	1	1	1	1	1	1	1	1	1	1

对于微调补偿，修正源是以 4194304 Hz 运行的 IRC 时钟。它本身也容易产生误差，比如：

- IRC 不是一个可靠的时钟源，其容差约为 10%。
- 任何环境温度的变化都会影响该时钟。

不过 RTC 中的自修正算法可以修正 IRC 的偏差。假设时钟变为 5 MHz。比较值按时钟比率进行调整。

请注意，当时钟不为 4.194 MHz 时，累加器也将加快累加速度（因为其仍为每 4.194 M 时钟周期累加一次）。在 5 MHz 下，累加器将在以下数值处计满返回：

$128 \times 5000000 / 4194304 = 152$ 。此时，微调 1 Hz 时钟将产生于：

表 2. 粗调补偿值

粗调补偿	23	45	67	90	112	135	7	29	52	74
------	----	----	----	----	-----	-----	---	----	----	----

对于 IRC 时钟偏差的补偿由 RTC 外设自身完成，用户无需进行操作。

3.3 由温度产生的 PPM 误差及其修正

让我们看一下如何产生 1 Hz 时钟和微调补偿时钟，以及如何补偿由温度产生的晶振频率偏差。

每隔数分钟（例如 15 分钟）测量一次温度并重新计算温度产生的误差。这样比较合理，因为温度不会非常频繁地变化。

3.4 通过 SAR ADC 测量温度

SAR ADC 模块有一个温度传感器，其输出与某个 ADC 模拟通道输入相连。以下公式提供了温度传感器的近似传输函数。

$$Temp = 25 - ((V_{TEMP} - V_{TEMP25}) \div m)$$

等式 1.

其中：

- V_{TEMP} 表示温度传感器通道在环境温度下的电压。
- V_{TEMP25} 表示温度传感器通道在 25 °C 下的电压。
- m 为器件数据手册中的温度传感器斜率。它表示在热或冷环境下，电压与温度的斜率，单位是 V/°C。

现在，根据所用晶振的数据手册以及在 Kinetis-M 器件上实测到的数据，计算得出：

- m (温度传感器斜率) = 1.070 mV / C
- V_{TEMP25} = 719 mV (测量值)

使用该公式和替代值可以测出温度。

3.5 根据温度的 PPM 误差计算

典型数据手册中的晶振数据如下：

表 3. 典型工作值

项目	符号	规格
标称频率	f	32.768 kHz
温度范围	储藏温度	-55 °C 至 125 °C
	工作温度	-40 °C 至 85 °C
最大驱动电平	GL	0.5 μW (最大值)
推荐驱动电平	DL	0.1 μW
频率容差 (标准)	Δf/f	±20 × 10 ⁻⁶
峰值温度 (频率)	θ T	25 °C ±5 °C
温度系数 (频率)	A	-0.04 × 10 ⁻⁶ /°C ² (最大值)
负载电容	C _L	9.0 pF, 12.5 pF
串联电阻	R ₁	70 kΩ (最大值)
动态电容	c ₁	1.7 fF (典型值)
分流电容	C ₀	1.3 pF (典型值)
绝缘电阻	IR	500 MΩ (最小值)
老化	fa	±3 × 10 ⁻⁶ /年 (最大值)
冲击电阻	S.R.	±8 × 10 ⁻⁶ (最大值)

这里的关注值为温度系数 (频率)：

温度系数 (频率)	A	-0.04 × 10 ⁻⁶ /°C ² (最大值)
-----------	---	---

在公式中替代为：

$$f = f_0 [1 - 0.04 \text{ppm}(T - T_0)^2]$$

等式 2.

其中：

- f₀ 表示晶振频率：32768 Hz
- T 表示测得的环境温度
- T₀ 为 25 °C

将最终频率 f 与 f₀ 进行比较可以得出温度(T)下晶振的 PPM 误差。该误差值必须添加到偏移 PPM 误差以获得总 PPM 误差，从而将该总和用于补偿晶振误差。

注

晶振误差修正算法在很大程度上取决于所选晶振的类型。为了能够对晶振误差进行补偿，用户必须：

- 补偿每个晶振的偏移误差，因为每个晶振的偏移误差各不相同。
- 根据温度对时钟的实际影响，对一种类型的晶振进行补偿。每种类型的晶振都具有各自的特征误差。

RTC 时钟输出信号必须用于合适的场合，这样才能方便地在频率计数器类型的器件中使用。

4 示例

我们在 Kinetis-M 参考板上展开了一项实验以测量频率。图 2 显示了晶振存在 147 PPM 的误差。这表明频率为 1.000147636 Hz，而不是 1.000000000 Hz。其误差为 0.000147，即 147 PPM。在代码中使用该 PPM 误差来设置 PPM 偏移误差。

如果 PPM 误差为负，则频率计数器上显示的周期将小于 1.000000。例如，如果值显示为 0.999853212，则 PPM 误差为-147，因此必须进行相应的补偿。



图 2. 未修正晶振误差时的输出

在计算和编程 RTC 修正的整数与小数值并启用微调补偿后，测得的频率输出如图 3 中所示。误差小于 1 PPM。在-20 至+70 °C 的温度范围内，该误差维持在 5 PPM 以内。

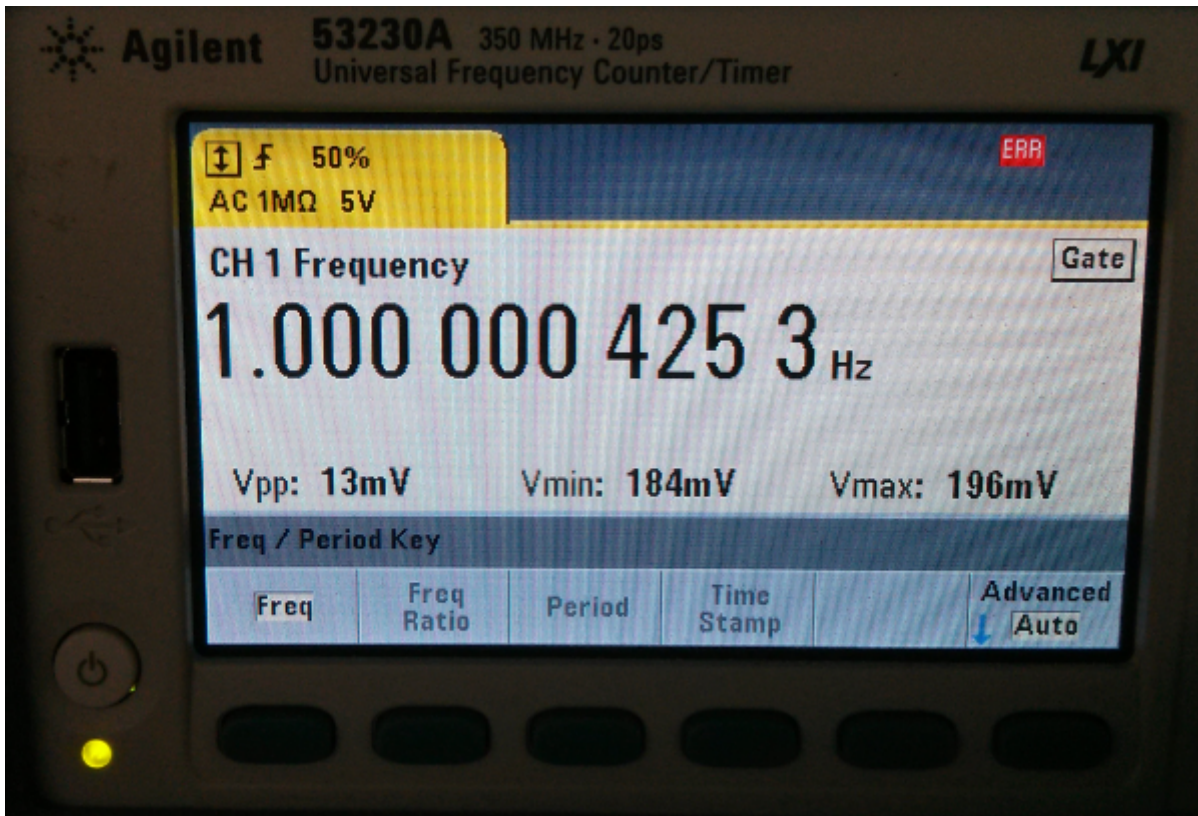


图 3. 微调补偿修正后的输出

不过，如果在计算并设定 RTC 修正的整数值后启用粗调补偿，测得的频率输出如图 4 中所示。请注意，粗调补偿输出的精度不及微调补偿输出。误差大约为 5 PPM。

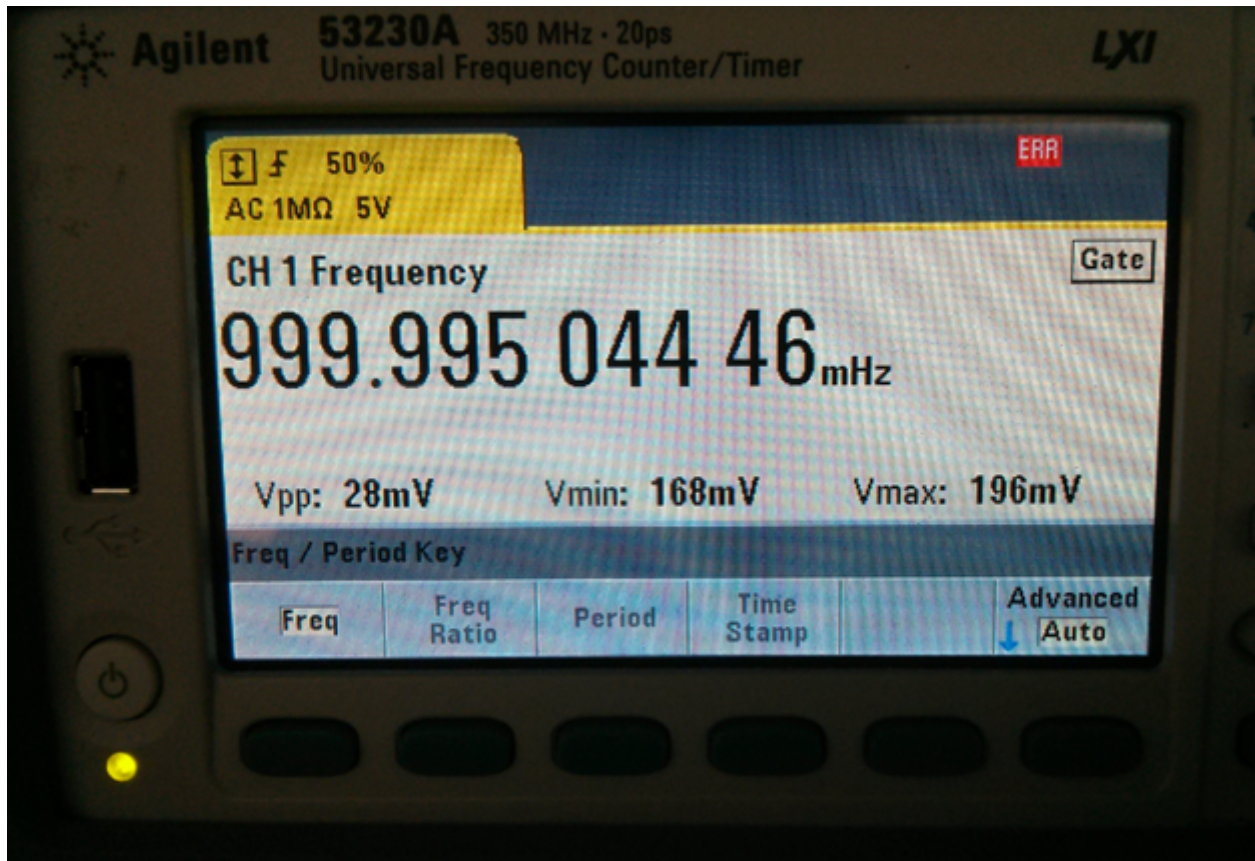


图 4. 粗调补偿修正后的输出

5 修订历史记录

表 4. 修订历史记录

修订版本号	日期	重大变更
0	07/2014	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. 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

