

如何在 STM32U5 系列上校准内部 RC 振荡器

前言

STM32U5 系列微控制器内嵌两个可选为系统时钟源的内部 RC 振荡器。它们是 HSI16（高速内部）和 MSI（多速内部）振荡器，其可驱动两个独立的输出：MSIS（用于系统时钟）和 MSIK（用于某些外设作为内核时钟）。HSI16 振荡器的典型频率为 16 MHz。MSI 振荡器为多速低功耗时钟源。

STM32U5 系列微控制器（称为 STM32U5 器件）具有三个辅助内部时钟源：

- LSI: 32kHz（内部低速）
- HSI48: 48MHz（内部高速），可直接用于 USB、RNG（真随机数发生器）和 SDMMC（SD/SDIO MMC 卡主机接口）。
- SHSI: 48MHz（+抖动）内部安全 RC 振荡器，专用于为 SAES 外设提供时钟

工作温度对 RC 振荡器的精度有影响。在 30°C 时，HSI16 精度为 $\pm 0.5\%$ ，MSI 精度为 $\pm 0.6\%$ ，HSI48 精度为 $\pm 4\%$ 。但在 -40°C 至 125°C 的温度范围内，精度会降低。为补偿温度对内部 RC 振荡器精度的影响，STM32U5 器件内置一些功能用于校准 HSI16、MSI 和 HSI48 振荡器及测量 LSI 振荡器频率。

当系统中有 32.768kHz 的 LSE（低速外部）时钟源可用时，MSI 振荡器的频率可通过硬件自动微调，从而达到优于 $\pm 0.25\%$ 的精度。这种使用 LSE 进行的自动硬件校准被称为 PLL（锁相环）模式。本应用笔记尚未对 PLL 模式进行介绍，仅侧重于用户微调。

本文档还详细介绍了如何校准 HSI16、MSI 和 HSI48 内部 RC 振荡器，方法如下：

- 基于找到具有最小误差的频率的方法
- 基于找到最大允许频率误差的方法
- 制定一个预先测量值的表然后在其中搜索相应变化的方法

LSI 振荡器的测量是通过将振荡器连接到定时器输入捕获来实现的。

本应用笔记随附的 X-CUBE-RC-CALIB 扩展包包含执行这些内部振荡器校准的源代码，以及运行示例所需的所有嵌入式软件模块。

1 STM32U5 系统时钟

STM32U5 器件具有下列时钟源可用于驱动系统时钟：

- HSI16：16 MHz 高速内部 RC 振荡器时钟
- HSE：4 至 50 MHz 高速外部振荡器时钟
- MSI (MSIS)：100 kHz 到 48 MHz 多速内部 RC 振荡器时钟
- PLL：1 MHz 到 160 MHz 锁相环 (PLL)，由 HSI16、MSI 或 HSE 振荡器提供时钟

HSI16 振荡器的典型频率为 16 MHz，功耗为 150 μ A。

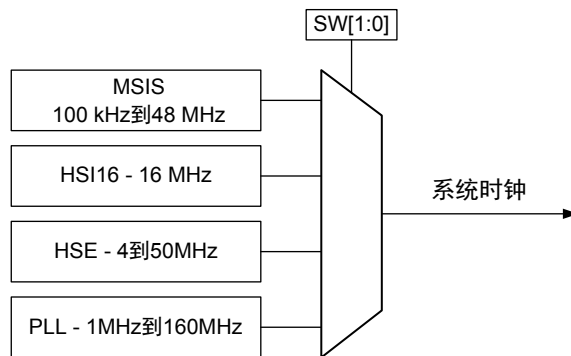
MSI RC 振荡器基于四个内部 RC 振荡器：MSIRC0 = 48 MHz、MSIRC1 = 4 MHz、MSIRC2 = 3.072 MHz、MSIRC4 = 400 kHz。每个 MSIRCx 振荡器具有四个分频器：/1、/2、/3 和 /4。MSI 共提供了 16 个频率范围，可选择用于两种输出：MSIS（用于系统时钟）和 MSIK（用于外设内核时钟）。MSI 的设计为工作电流与频率成正比（请参考产品手册以获得 MSI 功耗与所选频率范围之间关系的更详细信息），当 CPU 在低频运行时，可最小化内部振荡器功耗。在从复位重启或待机、关断低功耗模式唤醒后，MSIS 时钟被用作系统时钟。从停机模式唤醒后，可选择 MSI 时钟而不是 HSI16 作为系统时钟。

HSI48 时钟信号由内部 48 MHz RC 振荡器生成，可直接用于 USB、RNG 和 SDMMC。

内部 RC 振荡器 (HSI16、MSI 和 HSI48) 提供了一个低成本时钟源 (无需外部元件)。它们还具有比外部振荡器更快的启动时间和更低的功耗。可校准 HSI16、MSI 和 HSI48 振荡器以提高其精度。但即使校准后，内部 RC 振荡器频率精度也不如外部晶振或陶瓷谐振器的频率精度 (几十 ppm) 高。

注： 若外部振荡器发生故障，则 MSI 内部 RC 振荡器还可作为备份时钟源 (辅助时钟) 使用。

图 1. 简化的时钟树



STM32U5 器件还嵌入了以下次级时钟源 (不能用作系统时钟)：

- LSI：32kHz 低速内部 RC，可在停机和待机模式下保持运行用于 IDWG (独立看门狗)、RTC 和 LCD。LSI 振荡器无法校准，但可通过测量来评估任何频率偏差 (由于温度和电压变化)。
- LSE 晶振：32.768 kHz 低速外部晶振 RC，可选地驱动 RTC (实时时钟)
- HSI48：48 MHz 高速内部 RC，设计用于通过特制 CRS (时钟恢复系统) 电路为 USB 外设提供高精度时钟。它还可驱动 RNG 和 SDMMC。
- MSIK：多速内部 RC 振荡器时钟，用于外设内核时钟 (源自 MSIRCx 振荡器)
- SHSI：48 MHz 内部 RC 振荡器，专为 SAES 外设提供时钟

2 内部 RC 振荡器校准

由于生产过程的不同，每个器件的内部 RC 振荡器的频率都可能不同。因此，MSI 和 HSI16 RC 振荡器由意法半导体在 $T_A = 30^\circ\text{C}$ 时进行工厂校准。复位后，工厂校准值自动加载到内部校准位中。

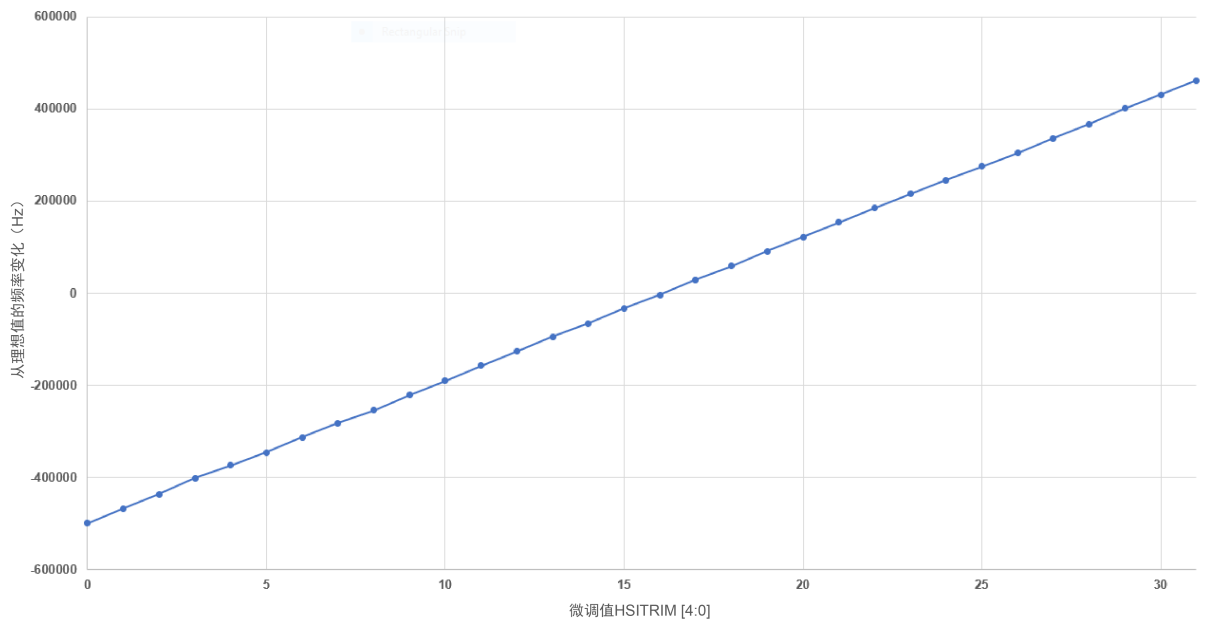
可微调内部 RC 振荡器的频率，以在更宽的温度和供电范围内达到更好的精度。这就是微调位的作用。

对于 HSI16 振荡器，复位后校准值加载于 HSICAL[11:0] 中。使用五个微调位 HSITRIM[4:0] 进行微调。默认微调值为 16。增加/降低此微调值会增加/降低 HSI16 频率。HSI16 振荡器微调步长为 0.18%（约 29 kHz），具体如下：

- 将微调值写为 17 至 31 会增加 HSI16 频率。
- 将微调值写为 0 至 15 会降低 HSI16 频率。
- 将微调值写为 16，HSI16 频率会保持为默认值。

下图显示了随校准值变化的 HSI16 振荡器行为。HSI16 振荡器频率随校准值增加（校准值 = 默认的 HSICAL[11:0] + HSITRIM[4:0]）。

图 2. HSI16 振荡器微调特性



对于 MSIRC_x 振荡器（ $x = 0..3$ ），复位后校准值加载于 MSICAL_x[4:0] 位中。五个微调位 MSITRIM[4:0] 使微调范围更宽。校准基于将默认的 MSICAL_x[4:0]（复位值）加上 MSITRIM_x[4:0]。

结果存储在 MSICAL_x[4:0] = 默认 MSICAL_x[4:0] + MSITRIM_x[4:0] 中。

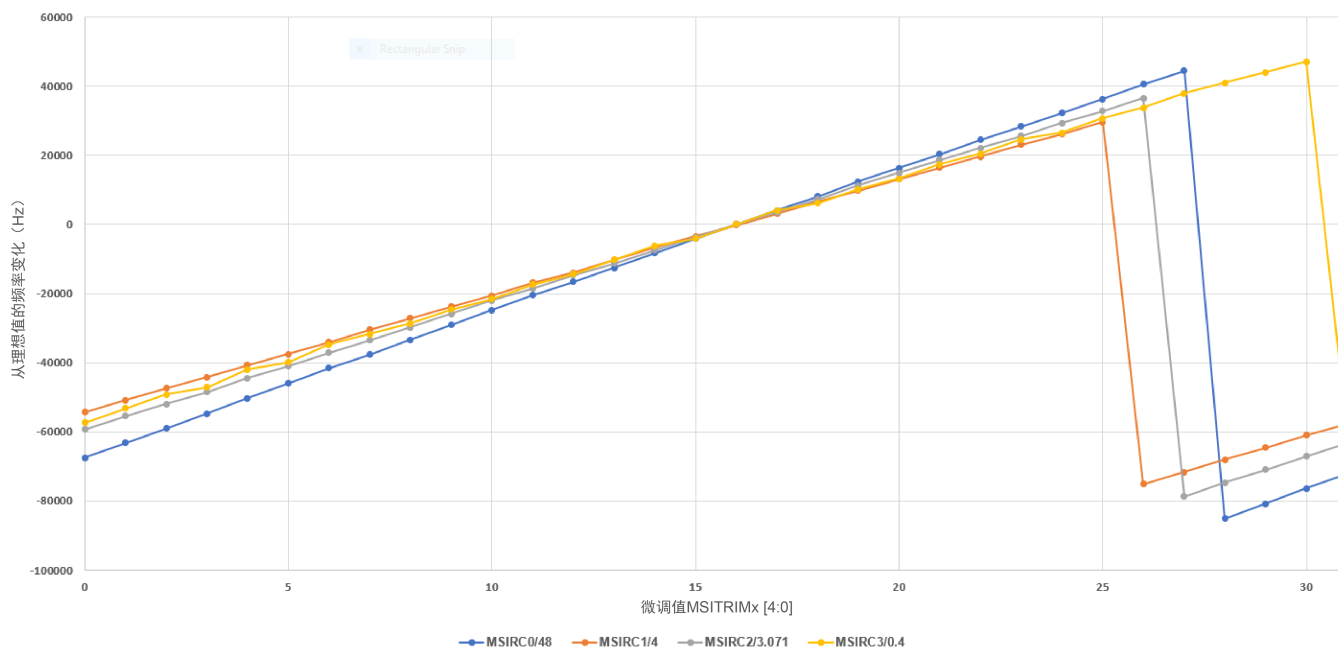
示例

假设默认的 MSI 校准值 MSICAL_x[4:0] 为 0x10：

1. 将 0x01 和 0x0F 之间的值写入 MSITRIM[4:0]，会使校准值 MSICAL_x[4:0] 处于 MSICAL_x[4:0] = 0x10 + 0x01 = 0x11 到 MSICAL_x[4:0] = 0x10 + 0x0F = 0x1F 的范围内。
这些结果大于 0x10（默认的 MSICAL_x[4:0] 值），因此 MSIRC_x 频率增加 1 步（0x11 - 0x10）至 15 步（0x1F - 0x10）。
2. 将 0x11 和 0x1F 之间的值写入 MSITRIM[4:0]，会使校准值 MSICAL_x[4:0] 处于 MSICAL_x[4:0] = 0x10 + 0x11 = 0x01 到 MSICAL_x[4:0] = 0x10 + 0x1F = 0x0F 的范围内。
这些结果小于 0x10（默认的 MSICAL_x[4:0] 值），因此 MSIRC_x 频率降低 1 步（0x01）至 15 步（0x0F）。
3. 在 MSITRIM[4:0] 中写入默认校准值 0x10 会导致校准值 MSICAL_x[4:0] 等于 MSICAL_x[4:0] = 0x10 + 0x10 = 0x00，因此 MSIRC_x 频率会降低 16 步（最低频率）。

下图显示了 MSIRCx 相比于 MSICALx[4:0] 在 4MHz 时的行为。

图 3. MSIRCx 振荡器微调行为



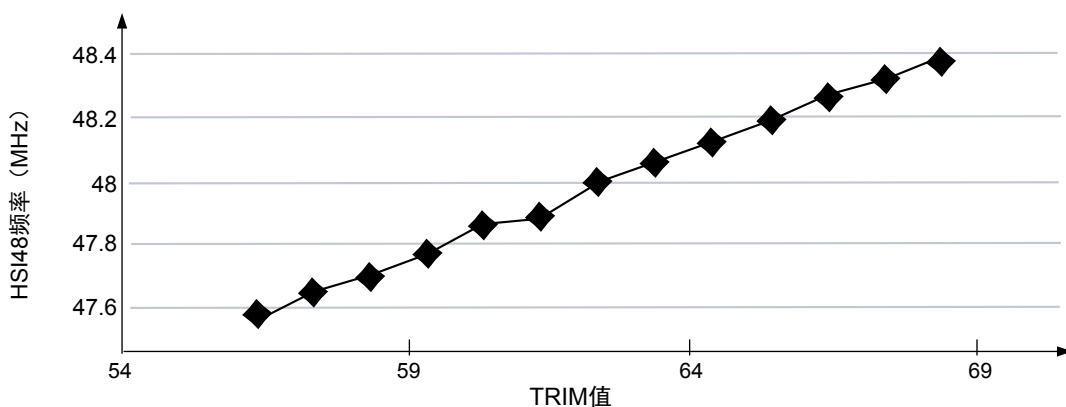
对于 HSI48 振荡器，复位后校准值加载于 HSI48CAL[8:0] 中。使用六个微调位 TRIM[6:0]（在 CRS_CR 寄存器中）进行微调。默认微调值为 64。增加/降低此微调值会增加/降低 HSI48 频率。

HSI48 振荡器微调步长为 0.12%（约 57 kHz），具体如下：

- 将微调值写为 65 至 127 会增加 HSI48 频率。
- 将微调值写为 0 至 63 会降低 HSI48 频率。
- 将微调值写为 64，HSI48 频率会保持为默认值。

下图显示了随校准值变化的 HSI48 振荡器行为。HSI48 振荡器频率随校准值增加（校准值 = 默认的 HSI48CAL[8:0] + TRIM[6:0]）。

图 4. 微调单调性



2.1 校准原理

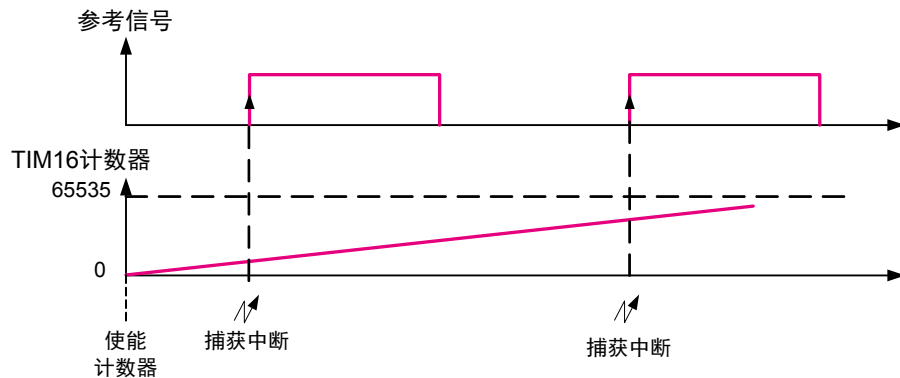
校准原理包括以下步骤：

1. 将（需要校准的）内部 RC 振荡器设置为系统时钟。
2. 测量每个微调值的内部 RC 振荡器（HSI16 或 MSIRCx）频率。
3. 计算每个微调值的频率误差。
4. 将微调位设置为最优值（对应于最低频率误差）。

内部振荡器频率并非直接测量，而是使用定时器计算时钟脉冲个数，与典型值比较。为此，必须有一个非常精确的参考频率，比如由外部 32.768 kHz 晶振或 50 Hz/60 Hz 电源频率（请参考第 2.2.2 节 用例 2：其他源用作参考频率）。

下图显示了怎样使用定时器计数个数测量参考信号周期。

图 5. 内部振荡器校准时序图



启用定时器计数后，当第一个参考信号上升沿发生时，捕捉定时器计数值，储存于 IC1ReadValue1 中。在第二个上升沿，又捕捉到定时器计数，储存于 IC1ReadValue2 中。在两个连续上升沿之间的时间（IC1ReadValue2 - IC1ReadValue1）表示了参考信号的整个周期。

因为定时器计数器的时钟由系统时钟提供（HSI16 或 MSIRCx），因此与参考信号有关的内部 RC 振荡器生成的真正频率为：

$$\text{测量频率} = (\text{IC1ReadValue2} - \text{IC1ReadValue1}) \times \text{REFERENCE_FREQUENCY}$$

误差（单位 Hz）为测量频率与典型值之差的绝对值。

因此，内部振荡器频率误差表示为：

$$\text{频率误差(Hz)} = \text{测量频率} - \text{系统时钟频率}$$

稳定时间和频率过冲/下冲

写入微调值后，振荡器频率从原始频率变为最终频率。这种频率变化需要一定的时间才能稳定到新的频率值。HSI16、HSI48 和 MSI 振荡器的稳定时间约为 15 μs （设置最终频率的时间）。

在稳定时间内，频率值可以高于（当新的频率值大于原来的频率值时）或低于（当新的频率值小于原来的频率值时）最终频率。该过冲/下冲值取决于应用于微调值的微调步长大小。最坏情况下的过冲变化（整个范围内的微调变化）约为 2.5%。中等变化约为 1.5%。

校准过程须在微调值变化和振荡器频率读取之间实施等待时间。

对每个微调值计算误差之后，算法会决定最优微调值（对应于最接近典型值的频率），编程于微调位中（请参考第 2.3 节）。

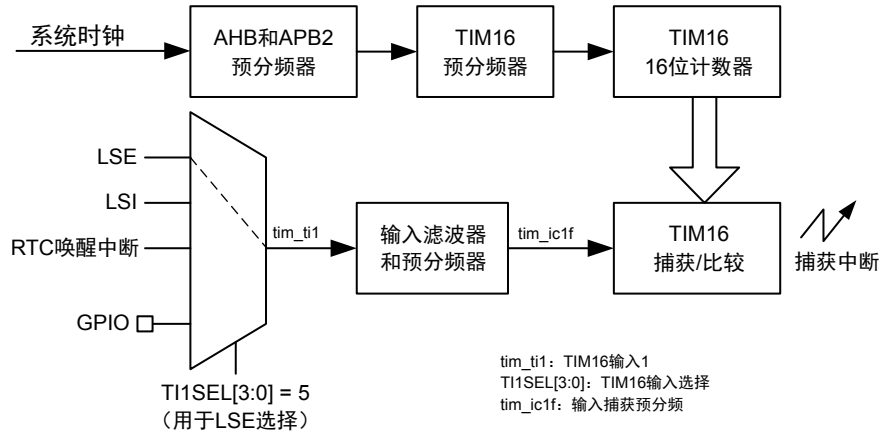
2.2 硬件实现

2.2.1 用例 1：LSE 用作参考频率

STM32U5 器件能够在内部将 LSE 连至定时器 16 通道 1。这样，LSE 时钟可用作内部振荡器校准的参考信号，而不需要额外硬件连接。仅 LSE 晶振/振荡器须连至 OSC32_IN 和 OSC32_OUT。

图中显示了使用 LSE 作为校准的精确频率源时，内部振荡器校准需要的硬件连接。

图 6. 使用 LSE 作为参考频率时的硬件连接



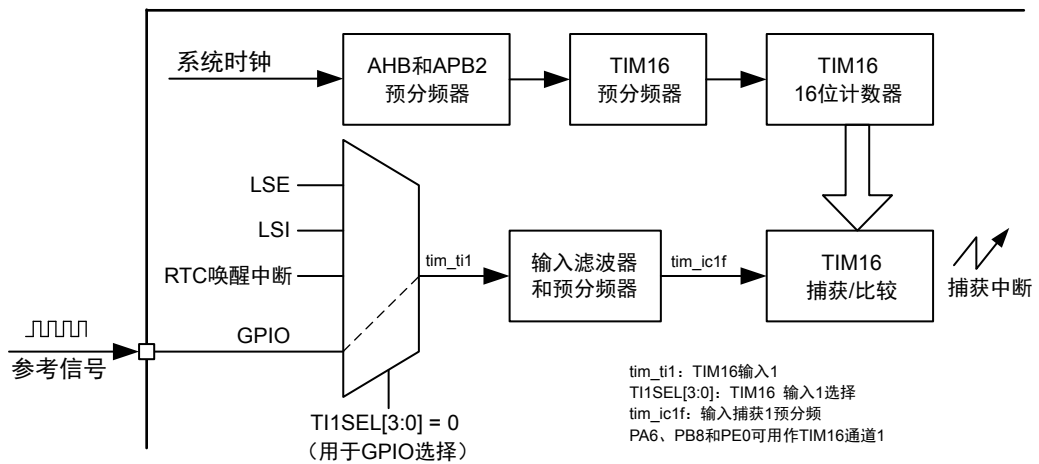
2.2.2

用例 2: 其他源用作参考频率

任何有精确频率的信号都可用于内部振荡器校准，例如主频率信号。

如下图所示，参考信号须连至定时器 16 通道 1。

图 7. 使用外部参考频率时的硬件连接



2.3

内部振荡器校准固件说明

本应用笔记提供的内部 RC 振荡器校准固件 X-CUBE-RC-CALIB 包括下列主要函数：

- uint32_t HSI16_CalibrateMinError(void)
- ErrorStatus HSI16_CalibrateFixedError(uint32_t MaxAllowedError, uint32_t* Freq)
- ErrorStatus HSI16_CalibrateCurve(uint32_t* Freq)
- uint32_t MSI_CalibrateMinError(void)
- ErrorStatus MSI_CalibrateFixedError(uint32_t MaxAllowedError, uint32_t* Freq)
- ErrorStatus MSI_CalibrateCurve(uint32_t* Freq)
- void HSI16_GetCurve(void);
- void MSI_GetCurve(void);

2.3.1 具有最小误差的 HSI16/MSI 校准

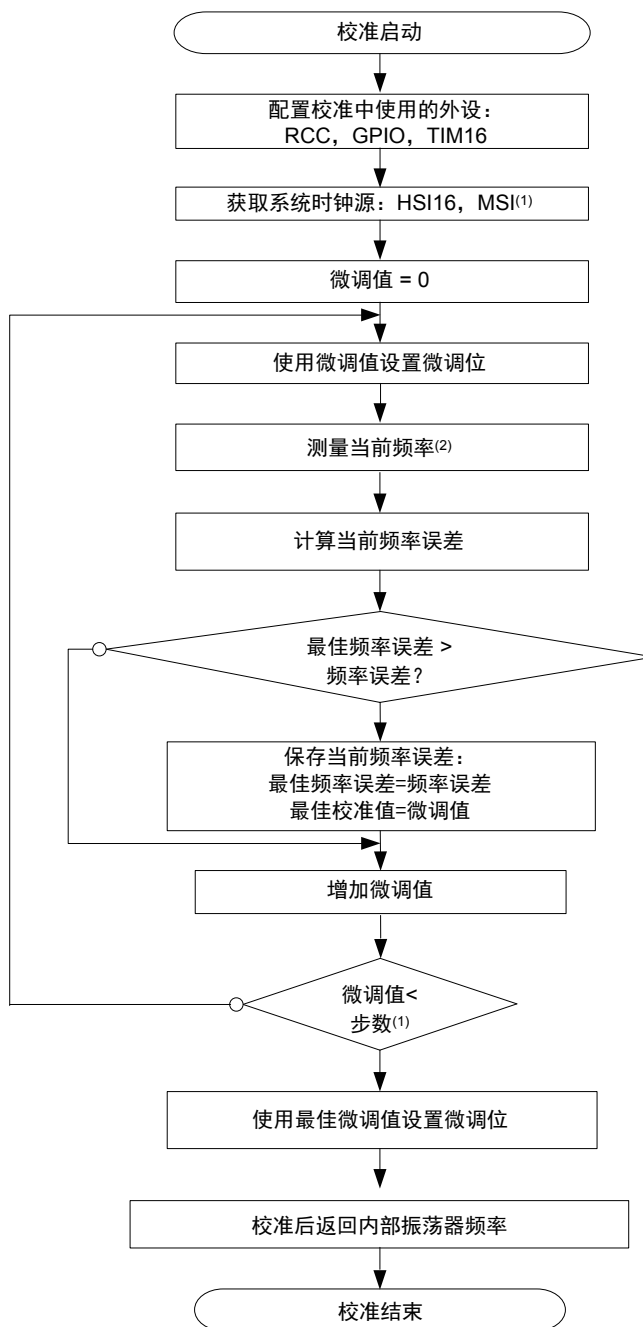
HSI16_CalibrateMinError() 和 MSI_CalibrateMinError() 函数将内部振荡器校准至最接近典型值的频率。这些函数测量不同微调值的所有频率，并提供对应于最小误差频率的微调值。所获得的微调值编程于微调位中。

校准后，HSI16_CalibrateMinError() 和 MSI_CalibrateMinError 函数会返回内部振荡器频率值，类型为无符号 32 位整数 (uint32_t)。

下图中的流程图详细说明了以下函数示例的算法。

```
uint32_t InternOscAfterCalib = 0;
{
.....
/* 得到校准后的内部振荡器 (HSI16) 值 */
InternOscAfterCalib = HSI16_CalibrateMinError();
}
```

图 8. 内部振荡器校准：找到最小频率



1. 若系统时钟源为 HSI16 或 MSI，则微调位长度为 5 位，步数为 32。
2. 第 2.3.4 节 中详细说明了频率测量。

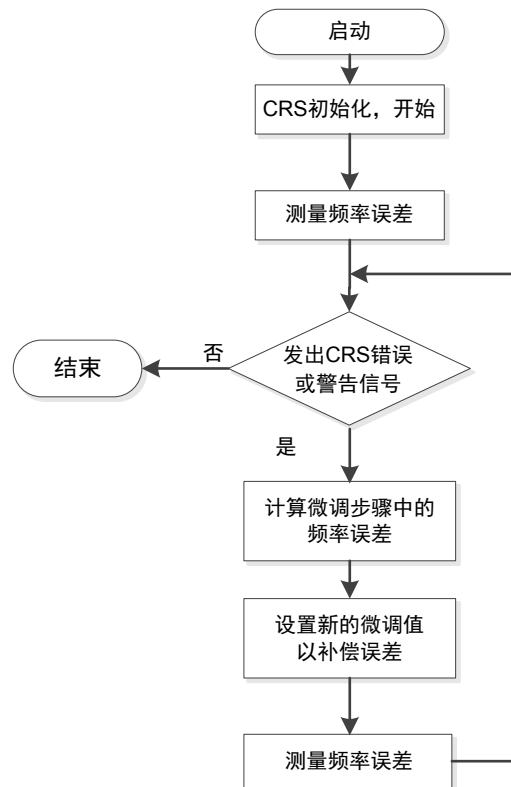
2.3.3 使用 CRS 进行 HSI48 校准

HSI48 的测量方法可以和 HSI16 或 MSI 的相同。STM32U5 器件采用 CRS（时钟恢复系统），能够基于与可选同步信号的比较自动调整振荡器微调。

在内部，HSI48 采用 16 位递增/递减计数器，该计数器逐步递增或递减微调值，直到达到预期频率值。

使用 CRS 进行 HSI48 校准可以在全自动模式下运行。为了加快校准速度，可以使用 CRS 测量实际误差，并根据预先计算好的值设置微调值。该校准过程可以重复一到两次，因为曲线可能不是线性的。当达到所要求的频率时，可以激活自动校准进而执行平滑校准（例如补偿温度变化）。

图 10. HSI48 微调算法



2.3.4 HSI16/MSI 频率测量

通过 TIM16 捕获中断执行内部振荡器频率测量。在 TIM16 ISR 中，计算了内部振荡器频率的完整周期。每个微调值测量的周期数可由用户在 hsi16.c 或 msi.c 文件中配置，如下所示：

```
#define HSI16_NUMBER_OF_LOOPS 10/* 测量的周期数 = 10 */
```

用户可以轻松配置参考源的频率（在头文件 hsi16.c 或 msi.c 中定义），如下所示：

- 若 LSE 时钟用作参考频率，则请去掉下面一行的注释，以确保配置了 LSE，并内部连接至 TIM16 通道 1。

```
#define USE_REFERENCE_LSE
```

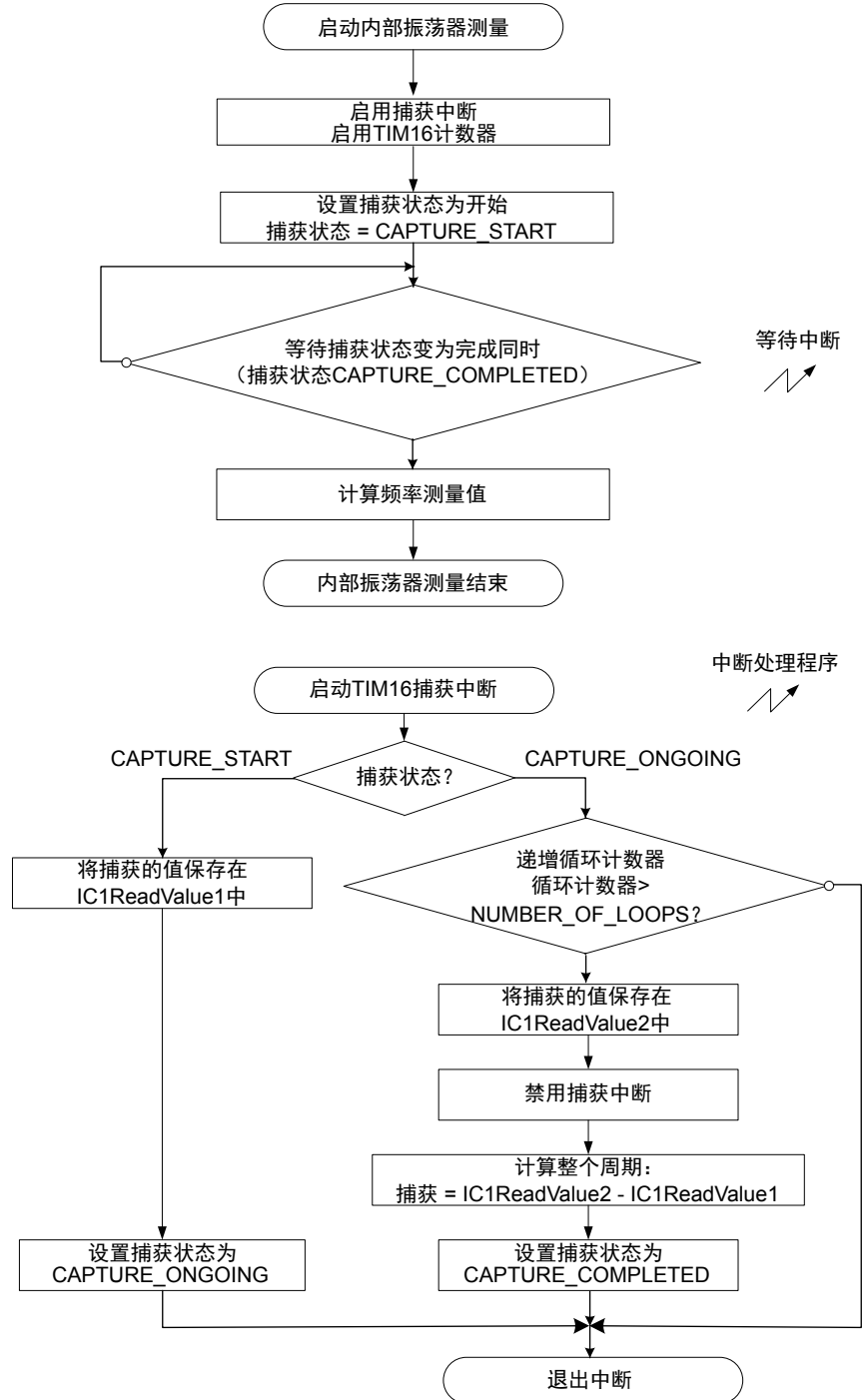
- 若参考频率为 50 Hz 的另一源频率，则请注释上面一行，并对参考频率进行如下定义。

```
#define REFERENCE_FREQUENCY (uint32_t)50/* 参考频率值，单位 Hz */
```

频率测量的计算不依赖于源参考信号的占空比，而是取决于源参考信号频率，因为捕获 1 中断已被配置为在每个参考信号的上升沿上发生（请参考第 2.1 节）。

下图详细介绍了频率测量算法。

图 11. HSI16/MSI 振荡器频率测量流程图



2.3.5 使用校准曲线进行 HSI16/MSI 校准

之前的两种方法（最小误差和固定误差）可能都需要很长时间，因为需要运行多次测量（例如，当使用最小误差法时，HSI16 需要进行 32 次测量）。对于所有微调值，测量对应频率和请求频率之间的差值并存储在表格中。校准曲线法使用与最小误差法相同的原理。当需要校准 HSI16_CalibrateCurve() 函数时，校准曲线法只测量一次实际频率，并在表中查找合适的值对差值进行补偿。

对于 MSI 校准，须使用 `MSI_GetCurve()` 和 `MSI_CalibrateCurve()` 函数，而不是 `HSI16_GetCurve()` 和 `HSI16_calibrateCurve()`。

2.4 校准库使用建议

下面列出了有关如何使用校准库的各种建议：

- 若外部信号频率低于系统时钟/65535，则必须使用 TIM16 计数器预分频支持低频。
- 若外部信号频率高于系统时钟/100，则必须使用 TIM16 输入捕获预分频器（分频器）支持高频。在使用 LSE 作为参考频率时，还必须优化输入捕捉预分频器。
- 建议在校准过程之前停止所有应用活动，并在调用校准函数后重新启动。因此，应用必须停止通信、ADC 测量和其他过程（除了将 ADC 用于校准时的情况。请参考以下实时校准相关内容）。
- 这些过程一般使用与校准过程中不同的时钟配置。否则，应用中可能引入错误（例如读/发帧时的错误或采样时间更改后的 ADC 读错误）。
- 内部 RC 振荡器校准固件使用 RCC（复位和时钟控制）微调内部 RC 振荡器，并使用 TIM16 测量内部 RC 振荡器。建议在运行校准程序后重新配置这些外设（如果在应用中使用）。
- 若应用运行时，环境温度显著改变，则可使用相比于温度的实时校准。内部温度传感器可与具有两个门限的 ADC 看门狗共同使用：
 - `Threshold_High = CurrentTemperatureValue + TemperatureOffset`
 - `Threshold_Low = CurrentTemperatureValue - TemperatureOffset`
 每次发生 ADC 看门狗中断时，都必须执行新的校准过程，根据当前温度更新两个门限。
- 随着工作条件（例如环境温度）的变化，校准曲线可能会发生变化。建议不时地（或在条件发生变化时）再次测量频率，确保使用正确的值。

2.5 校准过程性能

校准过程的时间取决于：

- 参考信号的频率（预分频值）：`REFERENCE_FREQUENCY`
- 每个微调值的测量周期个数：`NUMBER_OF_LOOPS`
- 校准过程期间测量频率的个数：步数

外设配置好并就绪之后（主要是 LSE 振荡器），校准过程的时间约为：

时间 = $(\text{NUMBER_OF_LOOPS} + 1) \times \text{步数} / \text{REFERENCE_FREQUENCY}$

若校准过程运行的是 HSI16 振荡器最小频率误差 (`HSI16_CalibrateMinError()`)，则步数等于 32。若使用 LSE 振荡器作为参考频率 (`REFERENCE_FREQUENCY = \text{LSE 值} / \text{输入 捕获 预分频} = 32768 / 8 = 4096 \text{ Hz}`)，所选的测量周期数为 10，则校准大约耗时：

时间 = $(11 \times 32) / 4096 = 86 \text{ 毫秒}$

当运行 MSI 振荡器的校准过程时，步数等于 32。若使用 LSE 振荡器作为参考频率

(`REFERENCE_FREQUENCY = \text{LSE 值} / \text{输入 捕获 预分频} = 32768 / 8 = 4096 \text{ Hz}`)，所选的测量周期数为 10，则校准大约耗时：

时间 = $(11 \times 32) / 4096 = 86 \text{ 毫秒}$

使用最大允许误差时的校准过程时间小于等于使用最小频率误差过程时的校准时间。

3 内部 LSI 振荡器测量

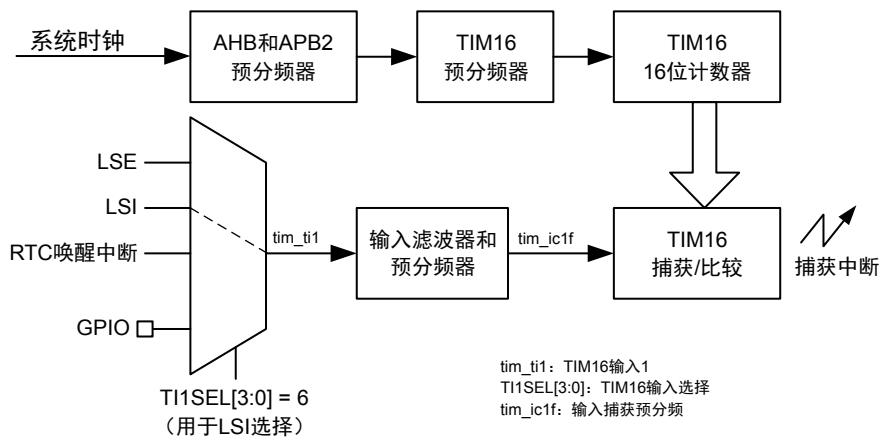
内部 LSI RC 振荡器为低功耗和低成本时钟源。在 STM32U5 器件中，在内部 RC 振荡器（LSI 或 MSI）和嵌入式定时器（TIM16）之间提供一个内部连接，以方便测量流程。

测量原理

内部 LSI RC 振荡器测量过程包括使用 HSI16 时钟运行定时器计数器、将定时器配置为输入捕获模式、将（需要测量的）内部 LSI RC 振荡器连至定时器。

下图显示了执行 LSI 测量所使用的配置。LSI 可内部连至 TIM16 输入 1（TI1）。

图 12. LSI 测量配置



启用定时器计数后，当第一个要测量的内部振荡器信号上升沿发生时，捕捉定时器计数值，储存于 IC1ReadValue1 中。在第二个上升沿，又捕捉到定时器计数，储存于 IC1ReadValue2 中。在时钟的两个连续上升沿之间的时间代表一个完整周期（参见第 2.1 节）。

内部振荡器频率值计算为 $HSI_Value / \text{捕获数}$ ，其中：

- HSI16_Value 是 HSI16 频率值（典型值为 16 MHz）。
- 捕获数表示内部 LSI RC 振荡器的整个周期：IC1ReadValue2 - IC1ReadValue1。

频率测量精度取决 HSI16 频率精度。因此，若参考信号可用，则可在执行内部 RC 振荡器测量之前，运行第 2 节中所述的内部 RC 振荡器校准程序。

可使用输入捕获预分频实现更高的测量精度。之前的公式变为：

$$LSI_Frequency = \text{InputCapturePrescaler} \times HSI16_Value / \text{Capture_Value}$$

第 2.3.4 节中所示的相同算法也用于测量 LSI 振荡器频率。

注： TIM16 ISR 用于 LSI 测量。

本应用笔记随附的 X-CUBE-RC-CALIB 扩展软件包含一个 C 源文件：

LSIMeasurement.c 使用 LSI_FreqMeasure() 函数，执行 LSI 频率测量

按照预定义的周期数测量内部 LSI RC 振荡器。

该参数（LSI 周期数）可以在 lsi_measurement.h 文件中更改：

```
#define LSI_PERIOD_NUMBERS 10
```

4 X-CUBE-RC-CALIB 演示

X-CUBE-RC-CALIB 演示展示了固件校准内部 RC 振荡器（HSI16、MSI 和 HSI48）的功能。在 STM32U5 器件上给出了一个内部 RC 振荡器测量的示例。

在运行校准程序之前，请在 `system_STM32U5xx.c` 中选择系统时钟源。如果 HSI16（或 MSI）振荡器被选为系统时钟源，则校准 HSI16（或 MSI，分别进行）。在此演示中，使用 LSE 振荡器作为参考校准内部 RC 振荡器（MSI 或 HSI16）。

默认情况下，该演示使用最小误差方法校准振荡器。

要运行提供具有固定误差或校准曲线的频率的校准过程，用户必须在 `main.h` 文件中对带有请求方法的行取消注释。

```
#define CALIBRATION_MIN_ERROR
#define FIXED_ERROR
#define ERROR_CURVE
```

对于 HSI48 振荡器，使用 CRS 用于测量和校准。要运行 CRS，可以选择以下源其中一个参考：

- 通过 GPIO 提供的信号
- LSE 振荡器

在选择 HSI16 作为系统时钟源的情况下运行演示时，将以下消息发送到 USART（在 Nucleo 板上实现为虚拟串行端口）。

```
-----
HSI before      : 16.098 MHz
HSI after       : 16.006 MHz

MSIRC0 before  : 16.055 MHz
MSIRC0 after   : 15.989 MHz
MSIRC1 before  : 16.055 MHz
MSIRC1 after   : 15.989 MHz
MSIRC2 before  : 16.055 MHz
MSIRC3 after   : 15.989 MHz
MSIRC3 before  : 16.055 MHz
MSIRC3 after   : 15.989 MHz

LSI frequency  : 31.758 kHz

HSI48 before   : 47.755 MHz
HSI48 after    : 48.008 MHz
-----
```

该消息可以通过 STLINK 虚拟串行端口上的某些终端软件（将 Nucleo 板连接到 PC）显示。校准 HSI16、MSI 和 HSI48 振荡器，然后测量 LSI。

5 结论

即使内部 RC 振荡器已经有工厂校准，若应用中需要高精度时钟，则用户还可在工作环境对其进行校准。

本应用笔记提供了以下例程：

- 多速和高速内部振荡器校准：怎样将振荡器精调至典型值
- 低速内部振荡器测量：怎样得到“精确”的 LSI/MSI 频率值

可以使用多个频率源校准内部 RC 振荡器（HSI16、HSI48 和 MSI），如 LSE 晶振或 AC 线等等。无论参考频率源是什么，内部振荡器的校准原理是一样的：必须提供一个参考信号，由定时器来测量。参考信号频率精度越高，内部振荡器频率测量的精度就越好。

对每个微调值，误差的计算为典型频率值与测量值之差的绝对值。之后，计算校准值，然后将其编程到微调位中。

本文档的第二部分介绍了 LSI 振荡器的测量。在 STM32U5 器件中，内部振荡器和嵌入式定时器之间的内部连接即用于此目的。该定时器使用系统时钟源作为时钟，配置为输入捕获模式。用内部振荡器两个连续上升沿之间捕获的时间来表示一个完整的周期。

版本历史

表 1. 文档版本历史

日期	版本	变更
2021 年 9 月 27 日	1	初始版本。

目录

1	STM32U5 系统时钟	2
2	内部 RC 振荡器校准	3
2.1	校准原理	5
2.2	硬件实现	5
2.2.1	用例 1: LSE 用作参考频率	5
2.2.2	用例 2: 其他源用作参考频率	6
2.3	内部振荡器校准固件说明	6
2.3.1	具有最小误差的 HSI16/MSI 校准	7
2.3.2	具有固定误差的 HSI16/MSI 校准	9
2.3.3	使用 CRS 进行 HSI48 校准	10
2.3.4	HSI16/MSI 频率测量	10
2.3.5	使用校准曲线进行 HSI16/MSI 校准	11
2.4	校准库使用建议	12
2.5	校准过程性能	12
3	内部 LSI 振荡器测量	13
4	X-CUBE-RC-CALIB 演示	14
5	结论	15
	Revision history	16
	目录	17
	表一览	18
	图一览	19

表一览

表 1.	文档版本历史	16
------	--------------	----

图一览

图 1.	简化的时钟树	2
图 2.	HSI16 振荡器微调特性	3
图 3.	MSIRCx 振荡器微调行为	4
图 4.	微调单调性	4
图 5.	内部振荡器校准时序图	5
图 6.	使用 LSE 作为参考频率时的硬件连接	6
图 7.	使用外部参考频率时的硬件连接	6
图 8.	内部振荡器校准：找到最小频率	8
图 9.	HSI16/MSI 校准流程图：最大允许频率误差	9
图 10.	HSI48 微调算法	10
图 11.	HSI16/MSI 振荡器频率测量流程图	11
图 12.	LSI 测量配置	13

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2021 STMicroelectronics – All rights reserved