AN2868
应用笔记

STM32F10xxx
内部 RC 振荡器 （HSI）校准

前言

STM32F10xxx 微控制器提供了适合运行的内部 RC 振荡器 （典型地，有 8 MHz 的 HSI：高速内部振荡器）。在 25 ℃时， HSI 的典型精度为 ±1%。在 –40 到 105 ℃， RC 频率精度值扩大为 ±3%。因此，温度对 RC 精度有影响。

为补偿应用中的温度影响，用户可使用运行时校准程序，进一步微调 STM32F10xxx HSI 振荡器的输出频率，提高 HSI 的频率精度。对通信外设来说，这可能是至关重要的。

本应用笔记给出了两个校准内部 RC 振荡器的方法：找到具有最小误差的频率或找到最大允许的频率误差。两者都通过提供精确的参考源，如 RTC/64 信号或主动信号实现。

这两个方法都基于相同的原理：计算 RC 频率 vs. 参考频率，计算 HSI 频率误差，设置 RCC_CR 寄存器中的 HSITRIM 位。

目录

表格索引

图片索引

1 STM32F10xxx 的内部时钟：HSI 时钟

HSI 时钟信号由内部 8 MHz RC 振荡器生成，可直接用作系统时钟，或者除以 2 用作 PLL 输入。 HSI RC 振荡器的优点是成本较低 （无需使用外部组件）。它还比 HSE 晶振具有更快的启动时间。但即使校准后，频率也不如外部晶振或陶瓷谐振器的频率精度高。 HSI 信号还可作为备份时钟源 （辅助时钟）使用，以防 HSE 晶振发生故障。

1.1 校准

由于生产过程的不同，每个芯片的 RC 振荡器的频率都可能不同。因此，每个器件都由 ST 做工厂校准，在 T_A = 25 ℃时达到 1% 精度。

复位后，工厂校准值将加载到时钟控制寄存器 RCC_CR 的 HSICAL[7:0] 位中。

通过设置 RCC_CR 寄存器中的 HSITRIM[4:0] 位进行用户校准。可对这些 位编程，以考虑电压和温度变化对内部 HSI RC 振荡器频率的影响。默认值为 16，加上 HSICAL 值，应能将 HSI 微调至 8 MHz ±1%。前后两个 HSICAL 步进之间的微调步长 （$F_{hsitrim}$）约为 40 kHz。

2 RC 校准

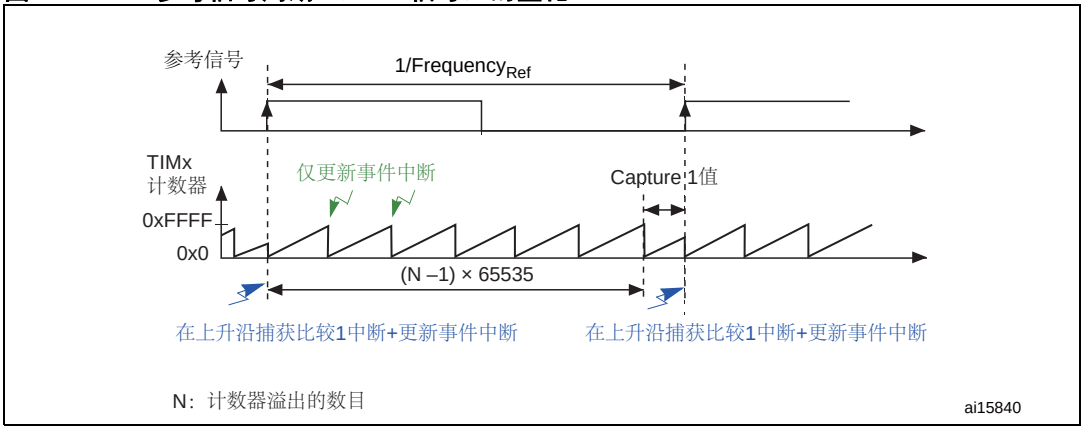
2.1 校准原理

校准的原理为首先测量 HSI 频率，然后计算频率误差，最后设置 RCC_CR 寄存器中的 HSITRIM 位。

HSI 频率并不是直接测量的，而是使用定时器对 HSI 时钟沿计数方式算出，然后与理想值 8 000 000 Hz 比较。为此，必须有一个非常精确的参考频率，比如由外部 32 kHz 晶振提供的 RTC/64 信号或 50 Hz/60 Hz 主频 （请参考*第 2.2.2 章节*）。对于 RTC 时钟源的情况，参考频率等于 512 Hz （32768 Hz/64）。

图 1 显示了怎样使用定时器计数个数测量参考信号周期。

图 1. 参考信号周期 （RTC 信号）的量化



在每个上升沿会发生两个中断：捕获比较 1 中断和更新事件中断。后者用于在参考信号周期对计数器溢出计数。因为在每个新周期开始时两个中断同时发生，所以会发生额外的溢出。这就是为什么我们必须对计数器溢出的数目减 1：N – 1。

因此 HSI 时钟沿计数数目如下：

$$TimerPeriodCount = (N-1) \times 65535 + Capture1$$ ，其中：

– N 为一个参考频率周期期间定时器溢出的数目

– Capture1 为从定时器 CCR1 寄存器中读取的值。

因为定时器由内部 RC 提供时钟，所以微控制器可计算 HSI 产生的真正频率并与参考频率比较。

$$Frequency_{RC} = TimerPeriodCount \times Frequency_{Ref}$$

误差 （单位 Hz）为 RC 频率 （$Frequency_{RC}$）与 8 000 000 Hz 之差的绝对值。

因此， RC 频率误差表示为：

$$Error(Hz) = \left| Frequency_{RC} - 8000000 \right|.$$

计算误差之后，算法会确定写入 RCC_CR 寄存器中 HSITRIM 位的校准值 （若需更详细信息，请参考*第 3 章节*）。

2.2 硬件实现

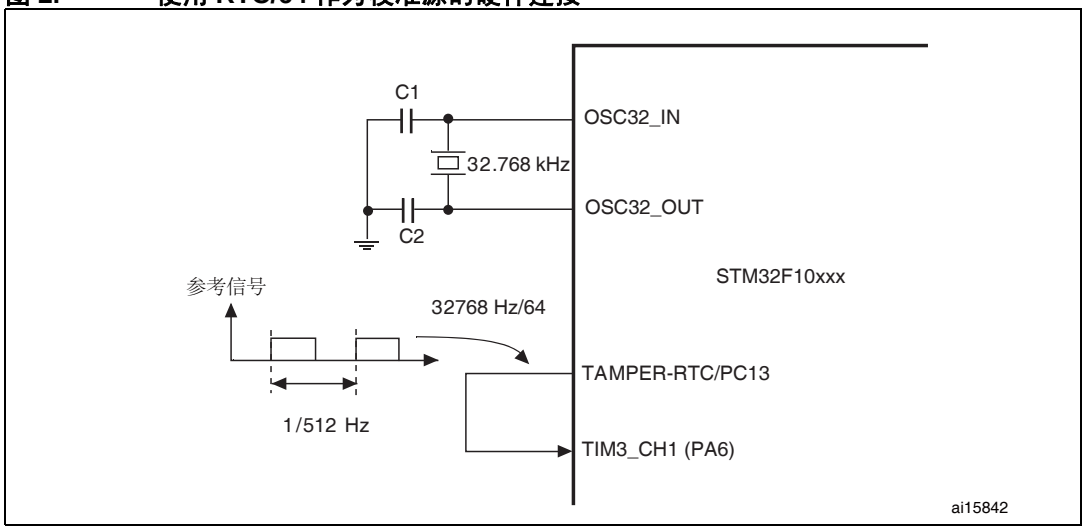
2.2.1 RTC/64 用作参考频率的情况：512 Hz

STM32F10xxx 可将 RTC 信号 64 分频输出至 GPIO PC13 （TAMPER-RTC 引脚）。TAMPER-RTC 信号可用作 RC 校准的参考频率。要达到该目的，此引脚必须配置为复用功能推挽，连至定时器通道输入。

注： *在本应用笔记的后续部分，使用的通道为定时器 3 通道 1 （TIM3_CH1）。*

图 2 显示了使用 RTC/64 作为校准的精确频率源时， RC 校准需要的硬件连接。

图 2. 使用 RTC/64 作为校准源的硬件连接



2.2.2 主频用作参考频率的情况：50 Hz/60 Hz

本节说明了使用 AC 主频作为参考频率方法的硬件需求。*图 3* 显示了向微控制器提供 DC 电源 （约 3.3 V）的电路实现：
保护定时器输入所需的唯一元件为一个电阻。因此若不需要电源，则只需在 TIM3_CH1 输入上有一个电阻以保护定时器输入的过电流。
此电路包括一些无源元件，以将 220 V/50 Hz 的 EU 电源或 110 V/60 Hz 的 US 电源转换为 3.3 V DC 电源。对于具有较高耗电电流的应用，可使用功率转换器 （请参考应用笔记 AN1357：VIPower：在非隔离应用中使用 VIPer12A 作为低成本电源）。

注意： 若电流变化太大，则参考信号调节和电源电路无法使用。

对于噪声主频的情况，建议使用输入功率线性滤波器 （请参考应用笔记 AN2326：使用主频校准 ST7ULTRALITE MCU RC 振荡器）。

图 3. AC 主频校准方法的硬件连接

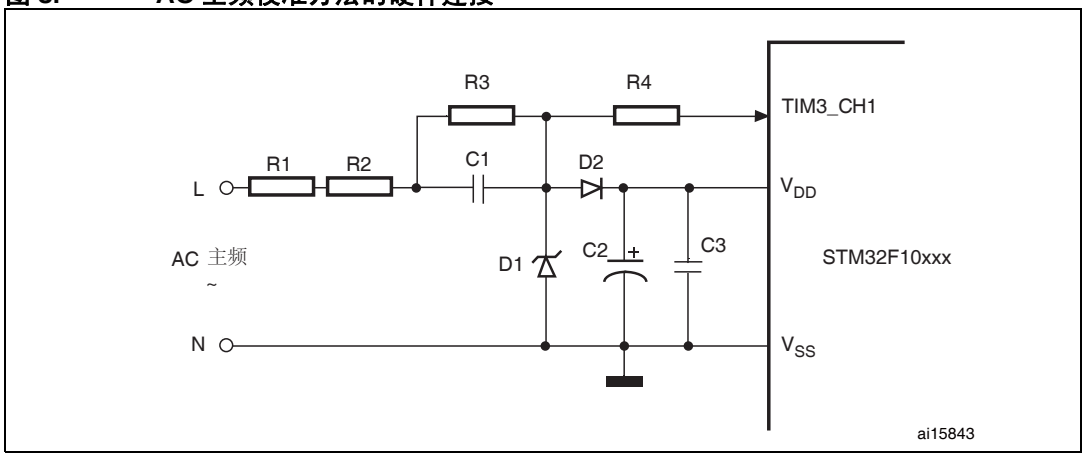


表 1. 当使用主频作为参考时的元件值

元件	230V/50 Hz 值	110V/60 Hz 值
R1	220 Ω / 0.5 W	110 Ω / 0.5 W
R2	220 Ω / 0.5 W	110 Ω / 0.5 W
R3	1 MΩ	1 MΩ
R4	5.6 kΩ	5.6 kΩ
D1	BZX85C3V9	BZX85C3V9
D2	1N4148	1N4148
C1	470 nF / ~275 V AC	330 nF / ~275 V AC
C2	100 nF	100 nF
C3	470 µF/16 V	470 µF/16 V

3 RC 校准库说明

本应用笔记提供的 HSI 振荡器校准库包括三个主要函数：

- `void HSI_FreqMeasure(void)`
- `s32 HSI_CalibrateMinError(void)`
- `ErrorStatus HSI_CalibrateFixedError(u32 AllowedErrorMax, s32* Freq)`

3.1 HSI_FreqMeasure() 函数

定时器 3 中断处理程序 （`TIM3_IRQHandler()`）中会调用此函数。
`HSI_FreqMeasure()` 函数会在每个输入信号周期后测量 RC 频率。测量周期的数目由用户在 *HSI_calibration.h* 文件中配置如下：

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

若周期数达到了 `NbOfPeriod` 个，则 `HSI_FreqMeasure()` 函数会计算所有测量频率（`NbOfPeriod` 个测量频率）的平均值。

使用平均方法使频率测量误差最小。

您可很方便地配置参考源的频率。它在文件 *HSI_calibration.h* 中如下定义：

```
#define Ref_Frequency 512 /* 参考频率值，单位 Hz */
```

若参考频率为 50 Hz 的主源频率，则请将上一行中的 512 改为 50，如下所示：

```
#define Ref_Frequency 50 /* 参考频率值，单位 Hz */
```

若使用 RTC/64 作为参考频率，则请去掉下一行的注释：

```
#define USE_Reference_RTC
```

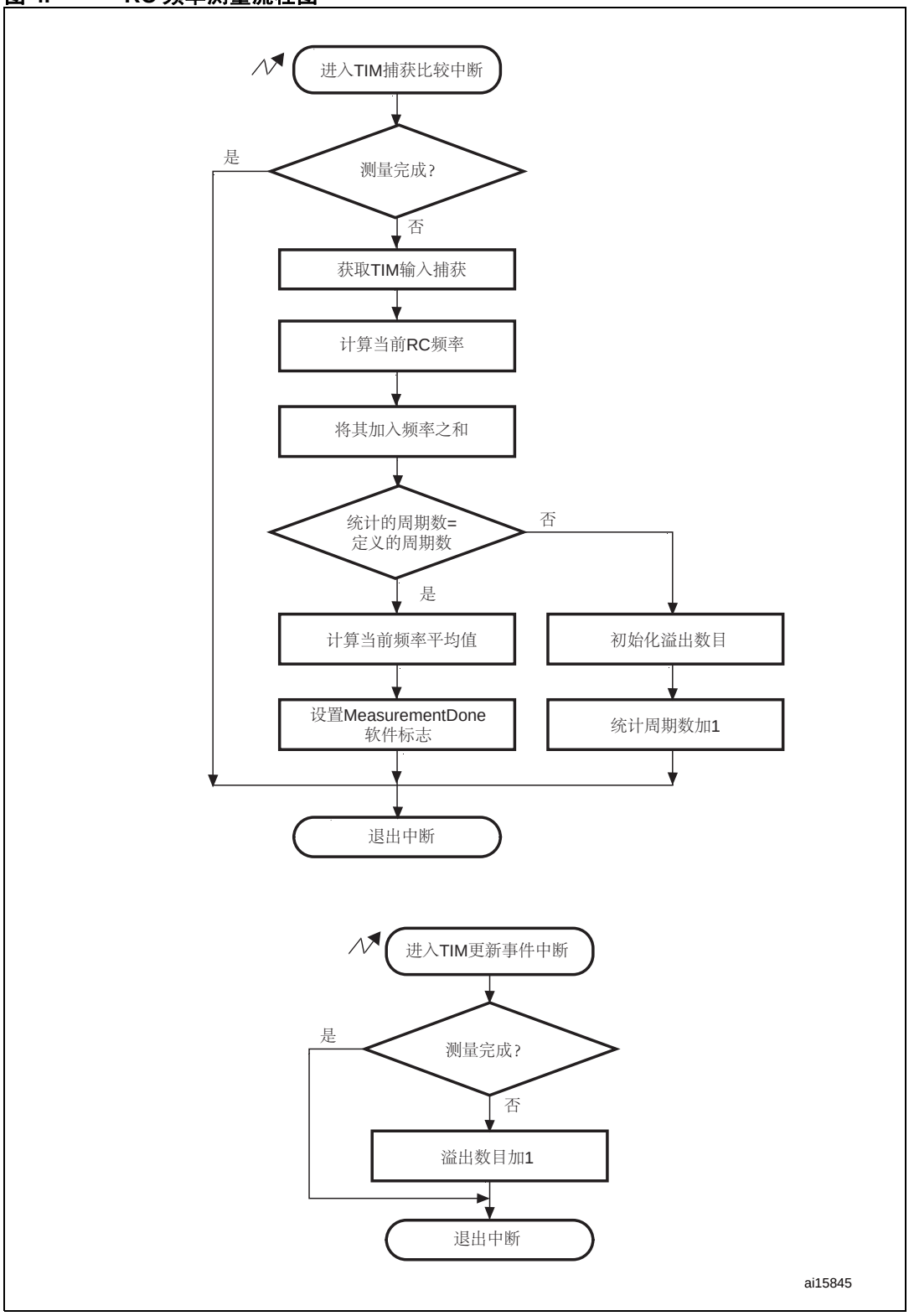
若使用的是其它参考，例如主频，则请注释掉上面一行。若这样做，则在系统校准配置期间不会配置 RTC 和 PC13 引脚。

在计算了频率平均值之后， MeasurementDone 软件标志会置为 ‘1’，表示频率测量结束，已准备好做下一次频率测量。

频率测量的计算不依赖于源参考信号的占空比。它取决于其频率，因为捕获 1 中断配置为在每个参考信号的上升沿发生 （请参考*图 1*）。

图 4 提供了频率测量算法。

图 4. RC 频率测量流程图



3.2 HSI_CalibrateMinError() 函数

此函数将 HSI 的频率校准到最接近 8 000 000 Hz。它测量所有 32 个频率 （HSITRIM 位的 32 个值），并提供对应于最小误差频率的 HSITRIM 值。以此得到的 HSITRIM 值为写入 RCC_CR 寄存器中 HSITRIM 位的校准值。
频率测量从 HSITRIM = 0 开始，到 HSITRIM = 31 结束。

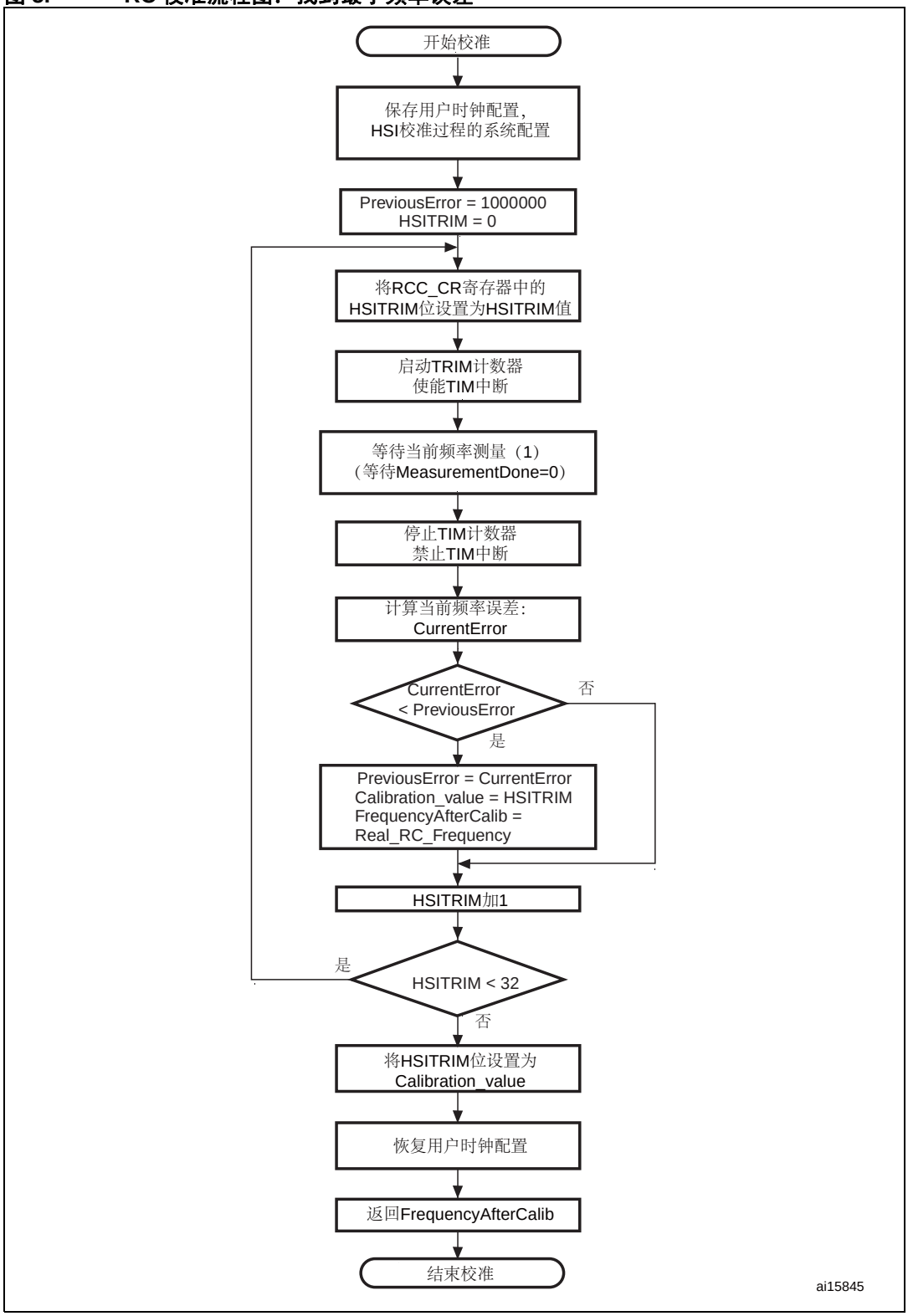
校准后， `HSI_CalibrateMinError()` 函数会将 RC 频率值作为有符号 32 位整型数（s32）返回。与用于通信外设的值一样，该值对重新配置预分频器有帮助。

图 5 中的流程图提供了此函数的算法。

示例

```
s32 FrequencyValue = 0;
int main()
{
......
/* 将内部 RC 校准至发现的最小误差频率，校准后返回单位为 Hz 的 RC 频率 */
FrequencyValue = HSI_CalibrateMinError();
.........
}
```

图 5. RC 校准流程图：找到最小频率误差



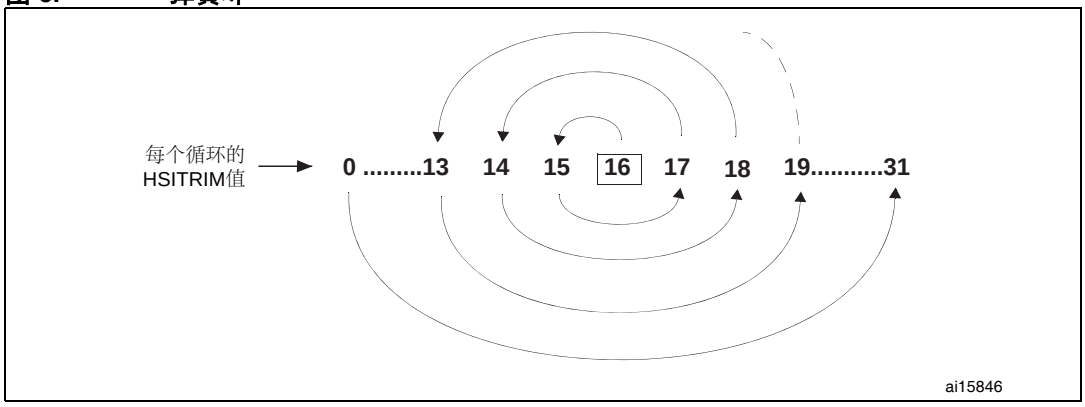
1. 请参见*第 2.1 章节*和*第 3.1 章节*。

3.3 HSI_CalibrateFixedError() 函数

此函数可将内部 RC 校准为最大允许的频率误差之内。该误差绝对值由用户配置，单位赫兹（第一个参数：*AllowedErrorMax*）。此函数与 HSI_CalibrateMinError() 相同（请参考*第 3.2 章节*），但它会搜索误差小于等于 *AllowedErrorMax* 的频率。

- 若它找到了此频率，则停止搜索，根据此频率配置 HSITRIM 位并返回 SUCCESS，意为校准操作成功。
- 否则，它会继续搜索，直到 HSITRIM 位 = 31（第 32 个频率）。然后它会将 HSITRIM 位设为 16（默认值）并返回 ERROR，意为校准失败，没有发现误差小于等于 *AllowedErrorMax* 的频率。
  频率测量从 HSTRIM = 16 开始（与 HSI_CalibrateMinError() 函数中，频率测量从 0 开始到 31 结束不同）。在循环中计算 HSITRIM 的值，以找到下一个值。即，HSITRIM 的值从 16 开始，然后变为左边的下一个值，然后变为右边的下一个值，然后变为左边的第二个值，以此类推，直到到达 31，形成一个“弹簧环”（如*图 6* 所示）。
  此算法的原理为，当 HSITRIM 位的值越接近 16 时，找到最小误差频率的概率越高；越接近 0 或 31 时，概率越低。因此，此算法的实现可最小化校准过程消耗的时间。

图 6. “弹簧环”



第二个参数是用于在校准后得到频率（单位赫兹），其类型为有符号 32 位整数（s32）。

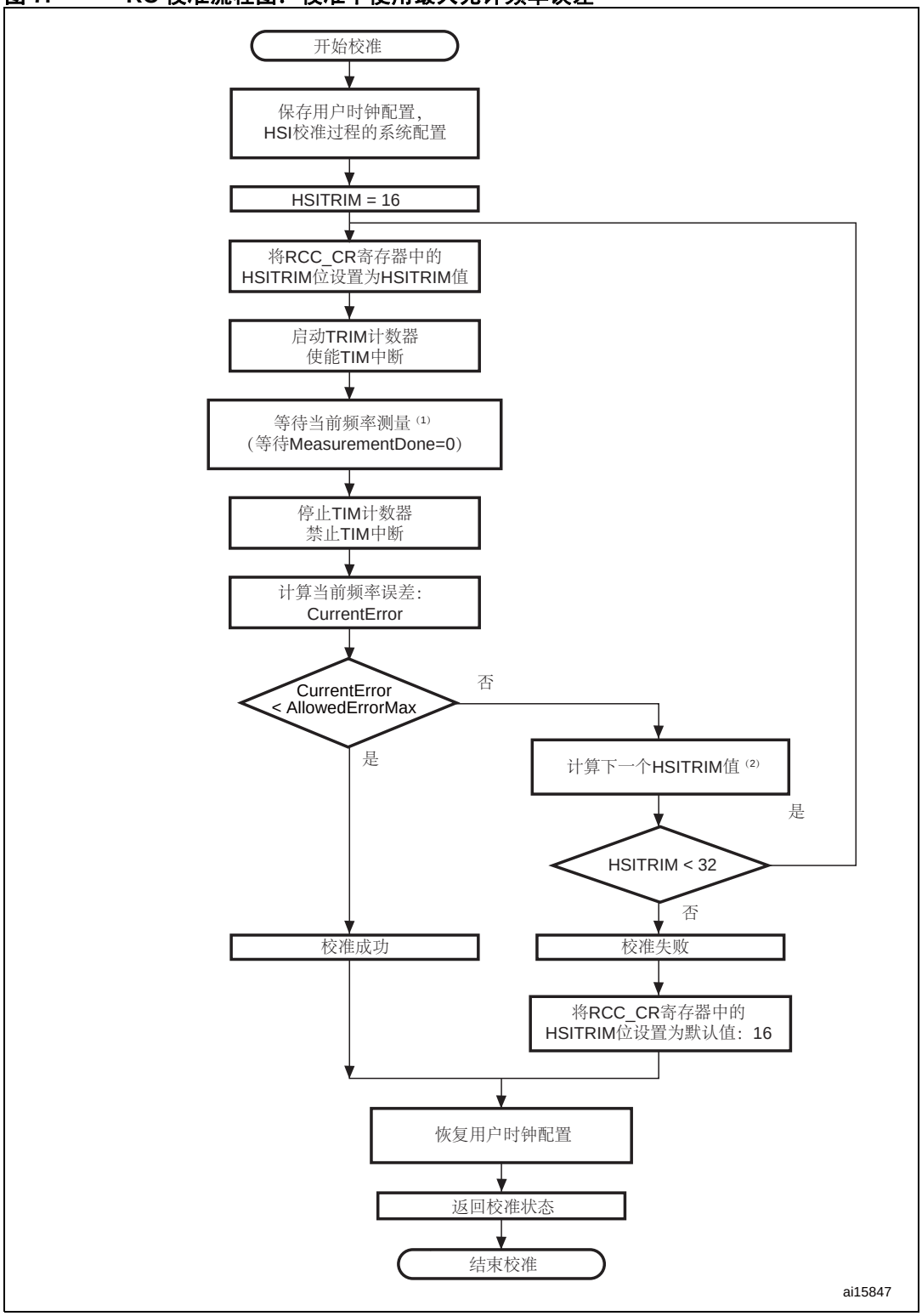
图 7 中的流程图提供了此函数的算法。

示例

```
#define LED_Green
#define LED_Red
s32 FrequencyValue = 0;
ErrorStatus CalibStatus = ERROR;
int main()
{
......
/* 允许的频率绝对值为 14000Hz */
CalibStatus = HSI_CalibrateFixedError(14000, &FrequencyValue);
if(CalibStatus = SUCCESS)
{
GPIO_SetBits(GPIOC, LED_Green);
}
else
{
GPIO_SetBits(GPIOC, LED_Red);
```

```
        }
        ......
        }
```

图 7. RC 校准流程图：校准中使用最大允许频率误差



1. 请参见*第 2.1 章节*。
2. 请参见*图 6*。

3.4 校准演示说明

本应用笔记提供的演示证明了固件校准 STM32F10xxx HSI 的能力。

绿色和红色 LED 必须连至 GPIO 端口 C 引脚，绿色 LED 连至 PC5 引脚，红色 LED 连至 PC6。

其它五个 LED 必须连至 PC0 到 PC4，以显示 RCC_CR 寄存器中 HSITRM 位的值。

默认情况下，该演示配置为使用 RTC/64 源频率校准 HSI。请记住要将 PC13 连至 PA6 （如*第 2.2.1 章节*中所示），因为校准过程会一直等待，直到在 PA6 引脚上出现可用的源信号。

默认情况下，每个频率的测量周期数目设为 10 个。

```
#define NbOfPeriod        10
```

若要使用最小误差频率方法运行校准过程，您必须在 *main.c* 文件中注释下面的定义。相反地，若要令校准过程找到具有最大允许误差的频率，您必须去掉下面一行的注释：

```
//#define USE_HSI_Fixed_Error
```

若要显示 RC 信号，请将示波器探头连至 PA8 引脚，并在 *main.c* 文件中去掉下面一行的注释：

```
#define OUTPUT_RC_ON_MCO_FOR_DEBUG
```

当演示运行于最小频率误差校准时，当测量 HSI 频率时您可在示波器上看到频率的变化。当校准过程结束后，您还可看到频率稳定下来。连至 PC0 到 PC4 的 LED 会根据 RCC_CR 寄存器中 HSITRIM 位的值 （二进制值）亮起。绿色 LED 也会亮起。

当演示运行于最大允许误差校准时，当测量 HSI 频率时您可在示波器上看到频率的变化。当校准过程结束后，您还可看到频率稳定下来。若校准过程成功，绿色 LED 会亮起，否则（若校准失败）红色 LED 会亮起。不管是哪种情况，（连至 PC0 到 PC4 的）五个 LED 会根据校准后 RCC_CR 寄存器中 HSITRIM 位的值亮起。

3.5 HSI 校准库使用建议

1. 当使用 HSI 校准库时，对于所有 TIM3 中断， NVIC 必须如下配置。此配置为 TIM3 中断预留：

   ```
   NVIC_InitStructure_IRQChannel = TIM3_IRQChannel;
   NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;
   NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
   ```

   捕获比较 1 和更新事件中断也为校准过程预留。
2. 当使用的参考频率超过 3 kHz 时，无法保证频率测量精度。
3. 校准之后，因为两个校准函数有其自身的时钟配置，因此它们会恢复用户时钟配置。因此，您应调用您的系统时钟配置函数，如下所示：

在 *HSI_calibration.c* 文件中

- 将用户时钟配置声明为外部函数：

```
/* 私有函数原型 ----------------*/
external void My_RCC_Configuration(void);
```

- 调用用户配置，如下所示：

```
void Restore_RCC_UserConfiguration(void)
{
/* 在这里调用您的默认 RCC 配置 */
```

```
My_RCC_Configuration(void);
}
```

4. 不建议在中断程序中调用校准函数，因为在这种情况下，校准过程的时间可能会长（请参考*第 4.2 章节*）。
5. 建议在校准过程之前停止所有应用程序，并在调用校准函数后重新启动它们。因此，应用必须停止通信、ADC 测量（参见*注：1*）等等，因为这些过程使用的时钟配置与校准过程中使用的不同。否则，应用中可能引入错误：读 / 发帧时的错误、采样时间更改后的 ADC 读错误等等。

注： *1 1) 除非为校准过程使用 ADC 时（请参考 7.）。*

6. 若您的应用使用 TAMPER-RTC 引脚上的 RTC 输出源，且选择的源不是 RTC/64（`BKP_RTCOutputSource_CalibClock`），则它的配置会丢失，必须根据您的应用需求重新配置引脚。
7. 若应用运行时，环境温度显著改变，则可使用实时校准 vs. 温度。内部温度传感器可与具有两个门限的 ADC 看门狗共同使用。每次发生 ADC 看门狗中断时，都必须执行新的 RC 校准过程，根据当前温度更新两个门限（本应用笔记中未实现此特性）：

   Threshold_High = CurrentTemperatureValue + TemperatureOffset

   Threshold_Low = CurrentTemperatureValue – TemperatureOffset

4 校准过程性能

4.1 频率测量的精度

HSI 频率测量的精度取决于参考频率的精度 / 稳定性及其取值。因为测量还取决于定时器的有限分辨率，所以建议使用不超过 3000 Hz 的参考频率。当使用的频率高于 3000 Hz 时，无法保证测量精度。

表 2 给出了校准效率与参考频率精度的对应关系。

表 2. RC 频率精度 vs. 参考频率精度(1) (2)

参考频率精度	RC 频率精度 （校准后）
0% 至 0.1%	0% 至 0.2%
0.2% 至 0.5%	0.3% 至 0.6%
0.6% 至 1%	0.7% 至 1.1%
1.1% 至 2%	1.2% 至 3%

1. 这些精度值仅为指示性数据，它们给出了校准后可达到的 RC 频率误差粗略范围。 RC 频率误差还取决于芯片。
2. 即使 HSI 频率被加倍，这些值仍然正确。这意味着使用 PLL 不会加倍系统时钟的频率误差。精度仍然相同。

4.2 校准过程的时间

校准过程的时间取决于：

1. 使用的参考频率
2. 每个频率的测量周期个数 （在 *HSI_calibration.h* 文件中定义的 `Ref_Frequency` 值）
3. 校准过程期间测量频率的个数

校准过程的时间由下式给出：

$$\text{CalibDuration} = \langle \frac{(\text{NPeriod}+1)\times \text{NFreq}}{F_{ref}} \rangle + \text{ErrorCompDur} + \text{ConfigRestoreDur}$$

其中：

- *NPeriod* 为同一 HSITRIM 配置 （同一频率）的频率测量次数
- *NFreq* 为所测频率的个数 （用于频率测量的 HSITRIM 值的个数）
- F_{ref} 为使用的参考频率，单位 Hz
- *ErrorCompDur* 为计算频率误差花费的时间。它需要约 14 µs。
- *ConfigResotreDur* 为校准过程将系统配置为 HSI 校准和此过程后恢复用户配置 （即 RCC、 TIM、 NVIC 等）所花费的时间。它需要约 280 µs。

注： 在每个测量频率中，为使测量频率稳定，捕获的第一个周期不会考虑。这就是为什么在上面的 CalibDuration 算式中 NPeriod 加了 1。

对于最小频率误差校准过程的情况（*HSI_CalibrateMinError()*），NFreq 的个数为 32。若使用 RTC/64 作为参考频率（512 Hz）且用户配置的测量周期数选为 10，则校准大约花费 688 ms。

$$\left\langle \frac{(10+1)\times 32}{512} \right\rangle + 300\ \mu s = 688\ ms$$

对于最大允许误差校准过程的情况（*HSI_CalibrateFixedError()*），NFreq 因芯片而异。NFreq 还取决于所选的最大允许误差。所选的允许误差越大，NFreq 就越接近于 1。所选的允许误差越小，NFreq 就越接近于 32。
因此，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。

5 结论

可使用多种频率源校准内部 RC 振荡器：RTC 晶振、 AC 线 （若需使用主频，请参考 AN2326 应用笔记）等等。不管参考频率源如何， RC 校准原理是相同的：必须提供参考信号，并由定时器测量。参考信号频率精度越高， RC 频率测量的精度就越好。误差的计算为理想 RC 频率值与测量值之差的绝对值。通过这些算出校准值，写入 RCC_CR 寄存器中的 HSITRIM 位。

若您选择使用 RTC 晶振作为参考频率源，则最大 RC 频率误差为 0.2%。

6 修订历史

表 3. 文档修订历史

日期	修订	变更
2009 年 2 月 2 日	1	初始版本。

请仔细阅读：

中文翻译仅为方便阅读之目的。该翻译也许不是对本文档最新版本的翻译，如有任何不同，以最新版本的英文原版文档为准。

本文档中信息的提供仅与 ST 产品有关。意法半导体公司及其子公司 （“ST”）保留随时对本文档及本文所述产品与服务进行变更、更正、修改或改进的权利，恕不另行通知。

所有 ST 产品均根据 ST 的销售条款出售。

买方自行负责对本文所述 ST 产品和服务的选择和使用，ST 概不承担与选择或使用本文所述 ST 产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为 ST 授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在 ST 的销售条款中另有说明，否则，ST 对 ST 产品的使用和 / 或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途 （及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

意法半导体的产品不得应用于武器。此外，意法半导体产品也不是为下列用途而设计并不得应用于下列用途：（A）对安全性有特别要求的应用，例如，生命支持、主动植入设备或对产品功能安全有要求的系统； （B）航空应用； （C）汽车应用或汽车环境，且 / 或 （D）航天应用或航天环境。如果意法半导体产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向意法半导体发出了书面通知，采购商仍将独自承担因此而导致的任何风险，意法半导体的产品设计规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML 或 JAN 正式认证产品适用于航天应用。

经销的 ST 产品如有不同于本文档中提出的声明和 / 或技术特点的规定，将立即导致 ST 针对本文所述 ST 产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大 ST 的任何责任。

ST 和 ST 徽标是 ST 在各个国家或地区的商标或注册商标。

本文档中的信息取代之前提供的所有信息。

ST 徽标是意法半导体公司的注册商标。其他所有名称是其各自所有者的财产。

© 2015 STMicroelectronics 保留所有权利

意法半导体集团公司

澳大利亚 - 比利时 - 巴西 - 加拿大 - 中国 - 捷克共和国 - 芬兰 - 法国 - 德国 - 中国香港 - 印度 - 以色列 - 意大利 - 日本 - 马来西亚 - 马耳他 - 摩洛哥 - 菲律宾 - 新加坡 - 西班牙 - 瑞典 - 瑞士 - 英国 - 美国

www.st.com