AN5126

Application note

How to calibrate internal oscillators on STM32G0 MCUs

Introduction

The STM32G0 series microcontrollers have internal RC oscillators that can be selected as the system clock source. These are known as the HSI16 (high-speed internal 16 MHz) and LSI (low-speed internal) oscillators. The STM32G0B1xx and STM32G0C1xx have a secondary internal clock source, the HSI48 (48 MHz (high-speed internal)) that can be used directly for USB and for RNG (true random number generator).

The operating temperature has an impact on the accuracy of the RC oscillators. At 25°C, the HSI16 oscillator has an accuracy of 0.25% typically, and the HSI48 accuracy is ±3%. However, in the temperature range of -40 to 105°C, the accuracy decreases.

To compensate for the influence of temperature on internal RC oscillators accuracy, the STM32G0 series microcontrollers have built-in features to allow users to calibrate the HSI16/HSI48 oscillator, and to measure the LSI (low-speed internal) oscillator frequency.

This application note focuses on how to calibrate internal RC oscillator HSI16 and HSI48. Three methods are presented:

- Method 1 consists in finding the frequency with the minimum error.
- Method 2 consists in finding the frequency within the allowed error.
- Method 3 consists in preparing the table of calibration values that is later used for fast calibration.

All three methods require an accurate reference signal.

The measurement of the LSI oscillator is performed by connecting the oscillator to a timer input capture.

This application note concerns the STM32G0 series microcontrollers and the X-CUBE-RC-CALIB software.

1 General information

This document applies to Arm®-based devices.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2 STM32G0 series system clock

The STM32G0 series microcontrollers have various clock sources that can be used to drive the system clock:

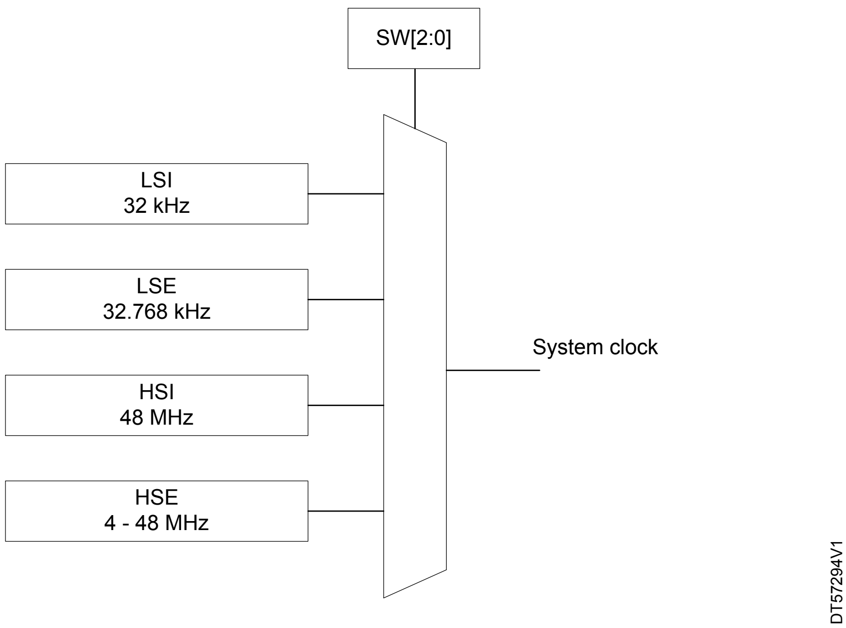
- A 16-MHz high-speed internal (HSI16) RC oscillator clock
- A 4- to 48-MHz high-speed external (HSE) oscillator clock
- A phase-locked loop (PLL) that is clocked by HSI16 or HSE oscillators
- A 32-kHz low-speed internal (LSI) RC oscillator clock
- A 32.768 kHz low-speed external (LSE) oscillator clock

Typically, the high-speed internal (HSI16) RC oscillator has a frequency of 16 MHz and consumes 155 µA.

The internal RC oscillator (HSI16) has the advantage of providing a low-cost clock source (no external components required). It also has a faster startup time and a lower power consumption than the external oscillator. The HSI16 oscillator can be calibrated to improve its accuracy. But even with calibration, the internal RC oscillator frequency is less accurate than the frequency of an external crystal oscillator or a ceramic resonator (tens of ppm).

The 32 kHz low-speed internal (LSI) RC is designed to drive the independent watchdog and optionally the real time clock (RTC). The LSI oscillator cannot be calibrated, but can be measured to evaluate frequency deviations (due to temperature and voltage changes). The 32.768 kHz low-speed external crystal (LSE crystal) optionally drives the real time clock (RTC).

Figure 1. Simplified clock tree



3 Internal RC oscillator calibration

The frequency of the internal RC oscillators may vary from one chip to another due to manufacturing process variations. For this reason, the HSI16 RC oscillator is factory-calibrated by STMicroelectronics to have a 1% accuracy at TA = 25 °C. After reset, the factory calibration value is automatically loaded in the internal calibration bits.

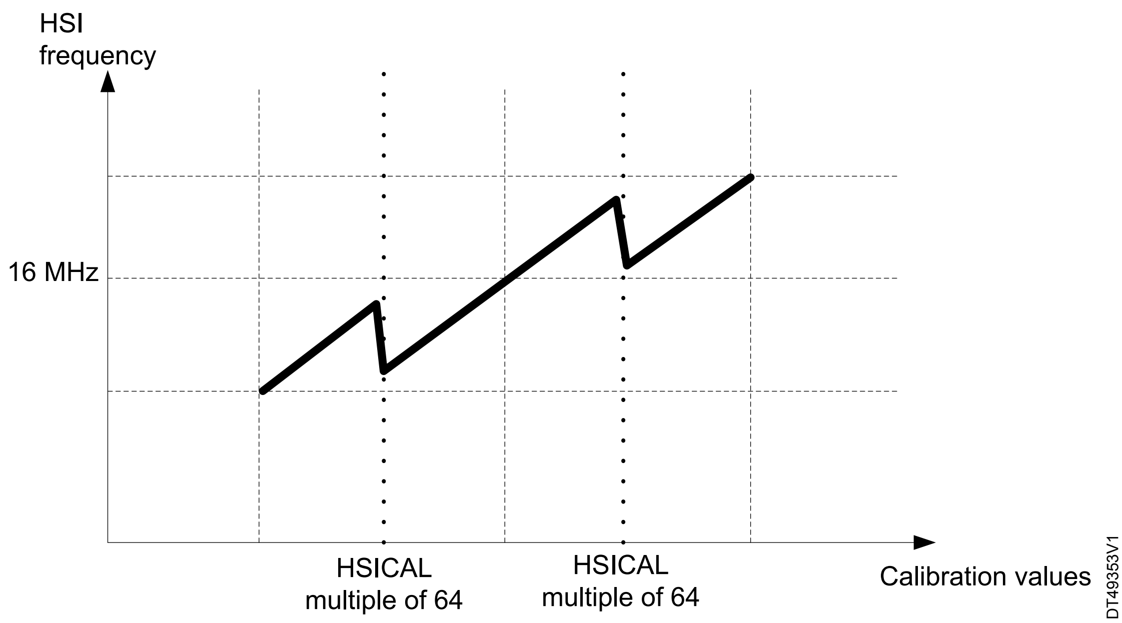
The frequency of the internal RC oscillators can be fine-tuned to achieve a better accuracy with wider temperature and supply voltage ranges. The trimming bits are used for this purpose.

For the HSI16 oscillator, the calibration value is loaded in HSICAL[7:0] bits after reset. Seven trimming bits HSITRIM[6:0] are used for fine-tuning. The default trimming value is 64. An increase/decrease in this trimming value causes an increase/decrease in HSI16 frequency. The HSI16 oscillator is fine-tuned in steps of 0.3% (around 48 kHz).

The figure below shows the dependency of HSI frequency on calibration value. The HSI16 oscillator frequency increases with the calibration value (calibration value = default HSICAL[7:0] + HSITRIM[6:0]), except at multiple of 64. At these calibration values, the negative step can reach twenty times the positive step.

During factory calibration the effort is made to place value of HSICAL just between the two negative steps. This is done so the HSITRIM can be modified to smoothly manipulate the value of final frequency without reaching the negative step.

Figure 2. HSI16 oscillator trimming characteristics



3.1 HSI16 calibration principle

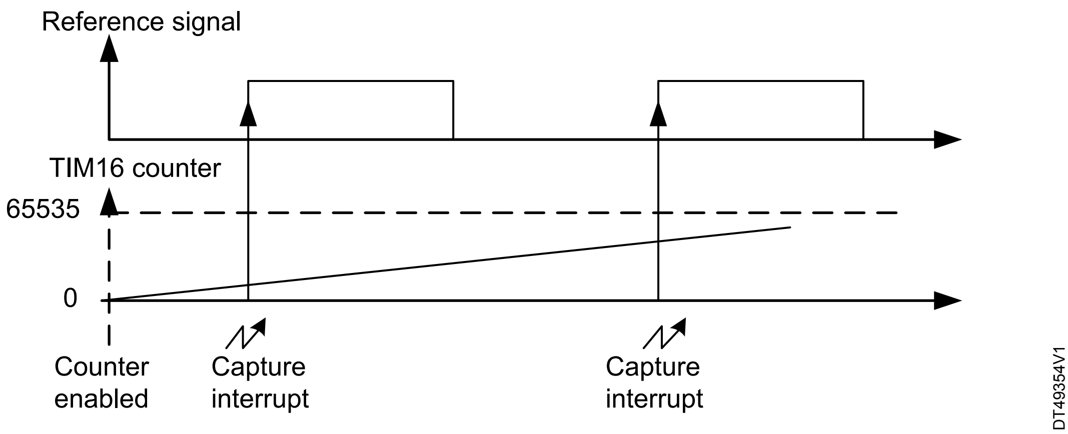
The calibration principle consists in:

1. Setting HSI16 as the system's clock source
2. Measuring the internal RC oscillator (HSI16) frequency for each trimming value
3. Computing the frequency error for each trimming value
4. Setting the trimming bits with the optimum value (corresponding to the lowest frequency error)

The internal oscillator frequency is not measured directly but it is computed from the number of clock pulses counted using a timer compared with the requested value. To do this, a very accurate reference frequency must be available such as the LSE frequency provided by the external 32.768 kHz crystal or the 50 Hz/60 Hz of the external signal (refer to Section 3.2.1 Case where LSE is used as the reference frequency to measure HSI16 and Section 3.2.2 Case where another source is used as the reference frequency to measure HSI16).

The figure below shows how the reference signal period is measured in number of timer counts.

Figure 3. Timing diagram of internal oscillator calibration



After enabling the timer counter, when the first rising edge of the reference signal occurs, the timer counter value is captured and stored in IC1ReadValue1. At the second rising edge, the timer counter is captured again and stored in IC1ReadValue2. The elapsed time between two consecutive rising edges (IC1ReadValue2 - IC1ReadValue1) represents an entire period of the reference signal.

Since the timer counter is clocked by the system clock (internal RC oscillator HSI16), the real frequency generated by the internal RC oscillator versus the reference signal is given by:

- Measuredfrequency = (IC1ReadValue2 – IC1ReadValue1) x referencefrequency

The error (in Hz) is computed as the absolute value of the difference between the measured frequency and the typical value.

Hence, the internal oscillator frequency error is expressed as:

- Error(Hz) = Measuredfrequency – requested value

After calculating the error for each trimming value, the algorithm determines the optimum trimming value (that corresponds to the nearest frequency to requested value) to be programmed in the trimming bits (refer to Section 3.3 Description of the internal oscillator calibration firmware).

3.2 Hardware implementation

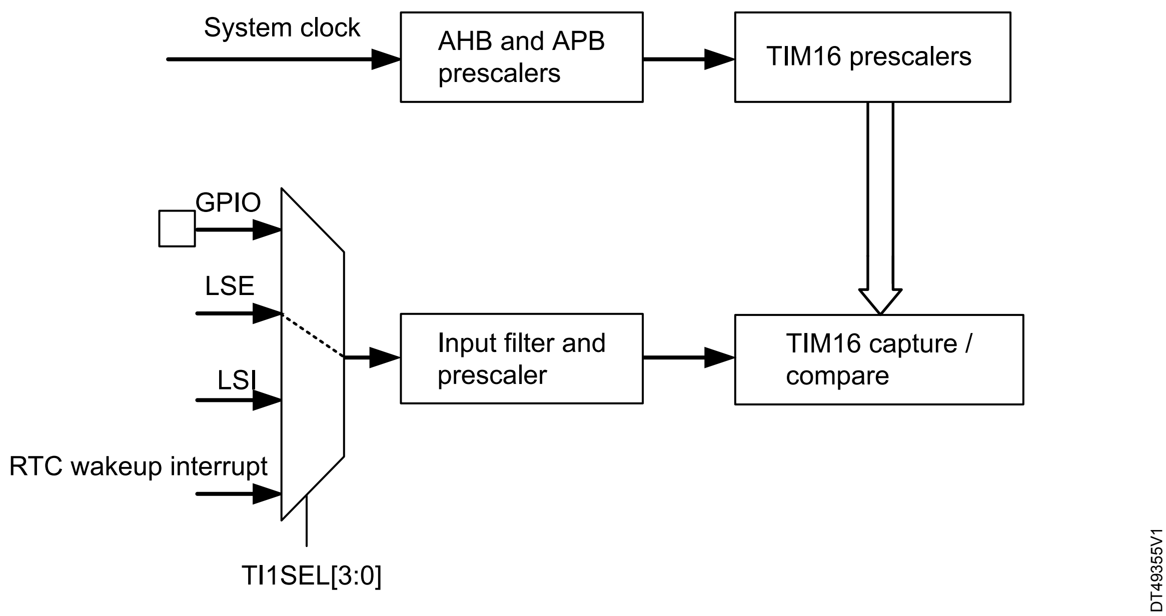
The STM32G0 series devices have the ability to connect internally and indirectly the internal or external oscillator to the dedicated timers (TIM14, TIM16 and TIM17).

3.2.1 Case where LSE is used as the reference frequency to measure HSI16

When LSE is used as the reference frequency to measure HSI16, the TIM16 channel is used. The LSE clock can be used as the reference signal for internal oscillator calibration and no additional hardware connections are required. Only the LSE oscillator should be connected to OSC32_IN and OSC32_OUT.

The figure below shows the hardware connections needed for internal oscillators calibration, using LSE as an accurate frequency source for calibration.

Figure 4. Hardware connection using LSE as the reference frequency

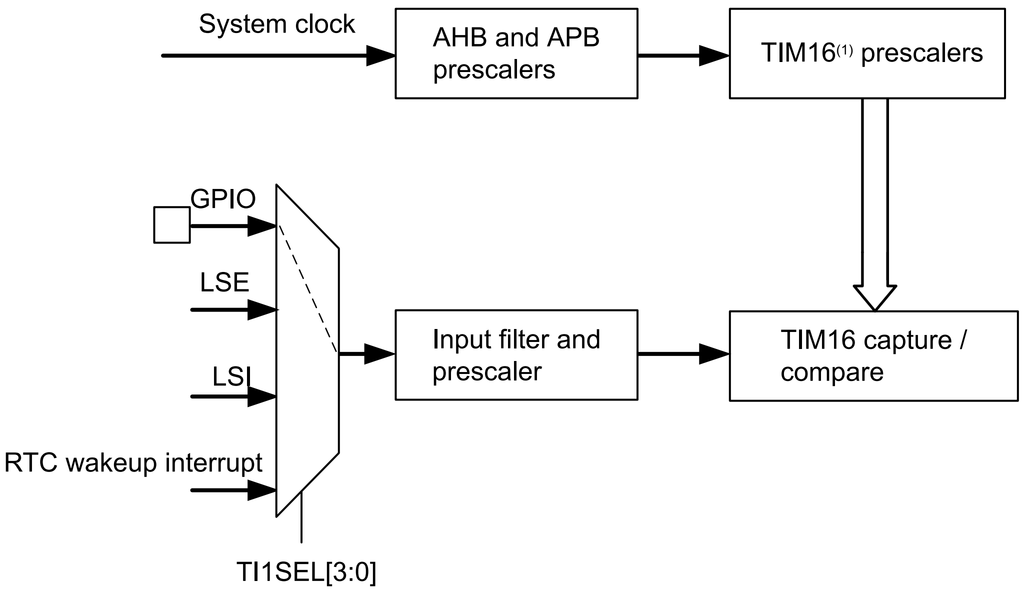


3.2.2 Case where another source is used as the reference frequency to measure HSI16

Any signal with an accurate frequency can be used for the internal oscillator calibration, and the external signal is one of the possibilities.

As shown in the figure below, the reference signal should be connected to Timer 16 channel 1.

Figure 5. Hardware connection using external reference frequency



(1) When using an external signal as a reference, another timer can be used instead of TIM16. The firmware provided with this application note uses TIM16

3.3 Description of the internal oscillator calibration firmware

The internal RC oscillator calibration firmware provided with this application note includes four major functions:

- `uint32_t HSI_CalibrateMinError(void)`
- `ErrorStatus HSI_CalibrateFixedError(uint32_t MaxAllowedError, uint32_t* Freq)`
- `ErrorStatus HSI_CalibrateCurve( uint32_t* Freq );`
- `void HSI_GetCurve( void );`

3.3.1 Internal oscillator calibration with minimum error

The HSI_CalibrateMinError() function calibrates the internal oscillator (HSI16) to have the frequency nearest to the typical value. It measures all frequencies for different trimming values and provides the trimming value that corresponds to the frequency with the minimum error. The trimming value is programmed in the trimming bits.

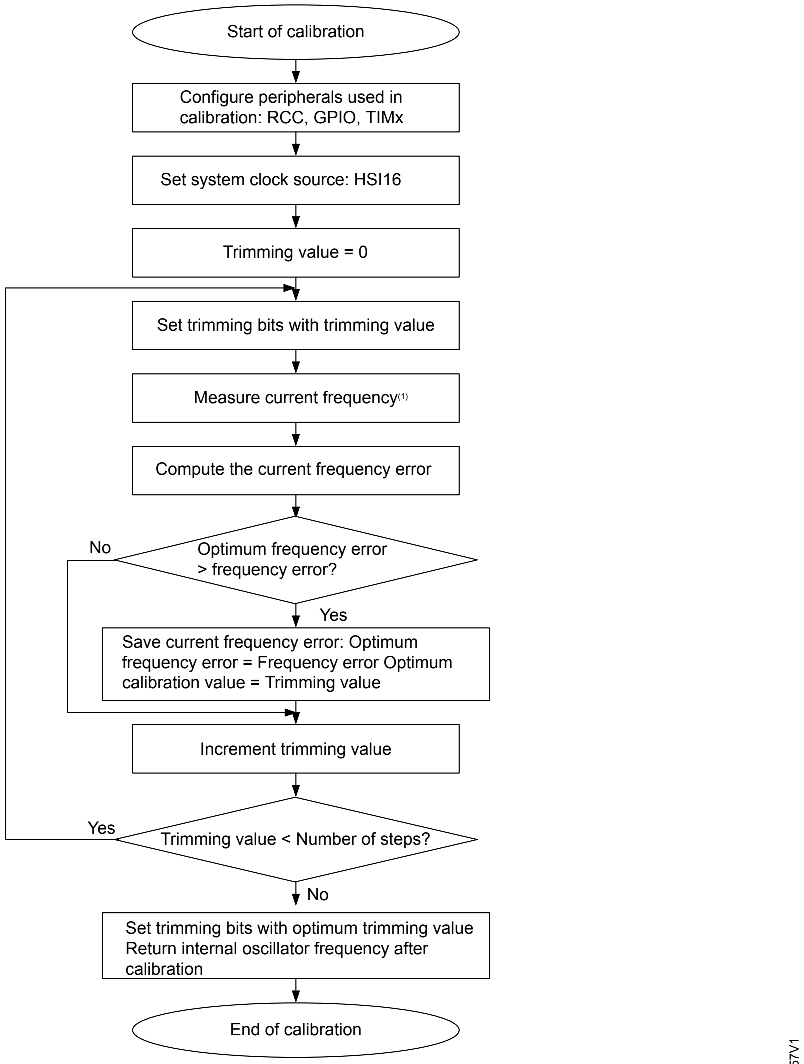
After calibration, the HSI_CalibrateMinError() function returns the internal oscillator frequency value as an unsigned 32-bit integer (uint32_t).

Example:

```
uint32_t InternOscAfterCalib = 0;
{
......
/* Get the internal oscillator (HSI) value after calibration */
InternOscAfterCalib = HSI_CalibrateMinError();
}
```

The flowchart below provides the algorithm for this function.

Figure 6. Internal oscillator calibration: finding the minimum frequency error



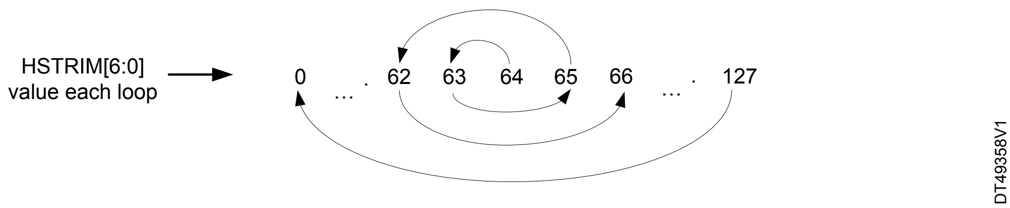
(1) The frequency measurement is detailed in Section « Internal oscillator frequency measurement »

3.3.2 HSI16 calibration with fixed error

HSI_CalibrateFixedError(uint32_t MaxAllowedError, uint32_t× Freq) function is provided to calibrate the HSI16 oscillator with a maximum allowed frequency error. It is configured by the user as an absolute value given in Hertz (the first parameter: MaxAllowedError). This function is the same as HSI_CalibrateMinError(), but it searches for the frequency that has an error (in absolute value) lower than or equal to MaxAllowedError.

- If it finds this frequency, it stops searching and configures the trimming bits HSITRIM[6:0] according to this frequency and returns SUCCESS, meaning that the calibration operation has succeeded.
- Otherwise, it continues searching for it until the HSITRIM bits = 127 (128th frequency). It then sets the trimming bits HSITRIM[6:0] to the default calibration value and returns ERROR, meaning that the calibration has failed and did not find any frequency with an error lower than or equal to MaxAllowedError. The frequency measurement starts with value of HSTRIM equal to 64. The HSITRIM value is computed in loops to find the next value. That is, the HSITRIM value starts from 64, then goes to the next value to the left, then to the next to the right, then to the second to the left and so on until it reaches 127, forming a “spring loop” (as shown in the figure below). This algorithm is based on the fact that the probability of finding the frequency that has the minimum error increases when the HSITRIM[6:0] value tends to 64. This algorithm is implemented so as to minimize the time consumed by the calibration process.

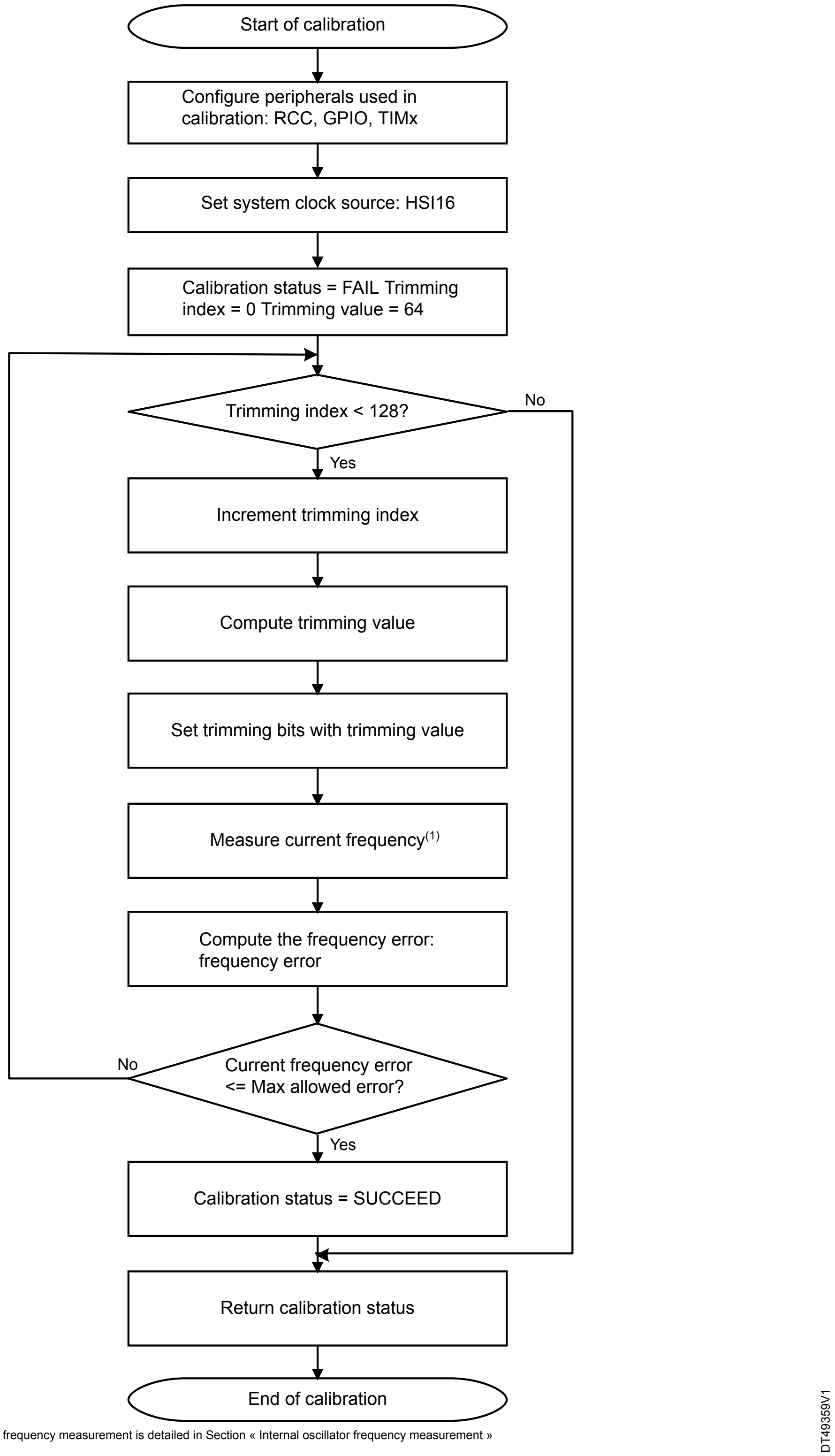
Figure 7. Spring loop



The second parameter is used to get the frequency (in Hertz) after calibration in the form of an unsigned 32-bit integer (unit32_t).

The following flowchart provides the algorithm for this function.

Figure 8. HSI16 calibration flowchart: maximum allowed frequency error



(1) The frequency measurement is detailed in Section « Internal oscillator frequency measurement »

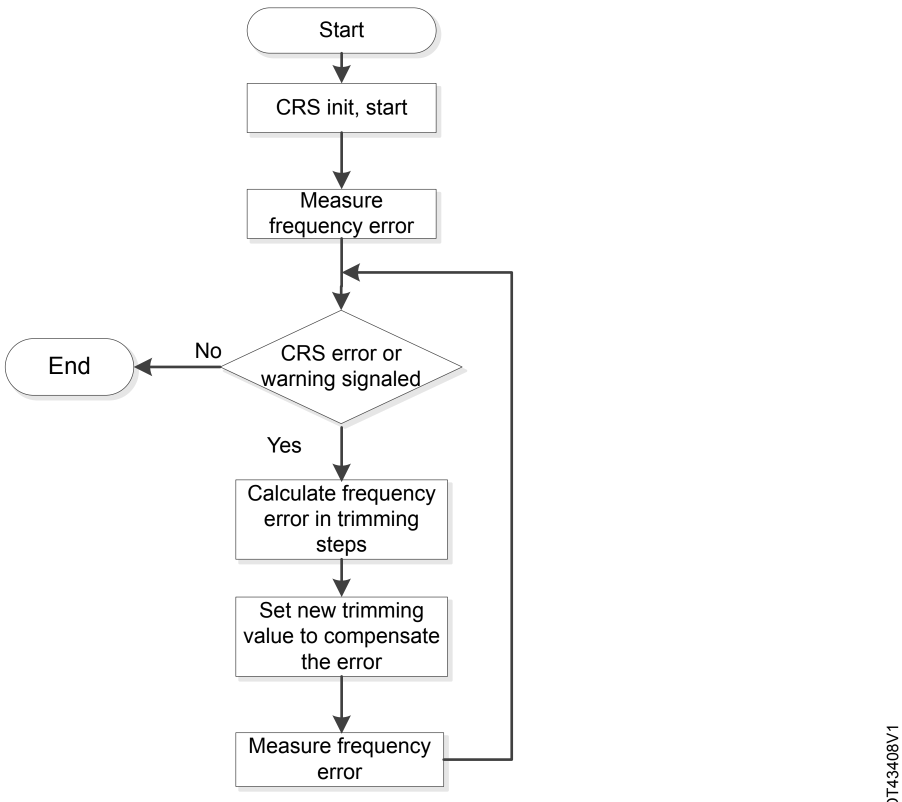
3.3.3 HSI48 calibration using CRS

The HSI48 can be measured the same way as the HSI16. The STM32G0 devices implement a CRS (clock recovery system) that is capable of doing an automatic adjustment of oscillator-trimming based on a comparison with a selectable synchronization signal.

Internally, the HSI48 implements a 16-bit down/up counter that increments or decrements step by step the trim value, until the expected frequency value is reached.

The HSI48 calibration using CRS can be run in a fully automatic way. To speed up the process, the CRS can be used to measure the actual error, and to set the trim value with a precalculated value. This process can be repeated once or twice as the curve may not be linear. When the requested frequency is reached, the automatic calibration can be activated for further smooth calibration (for example to compensate temperature changes).

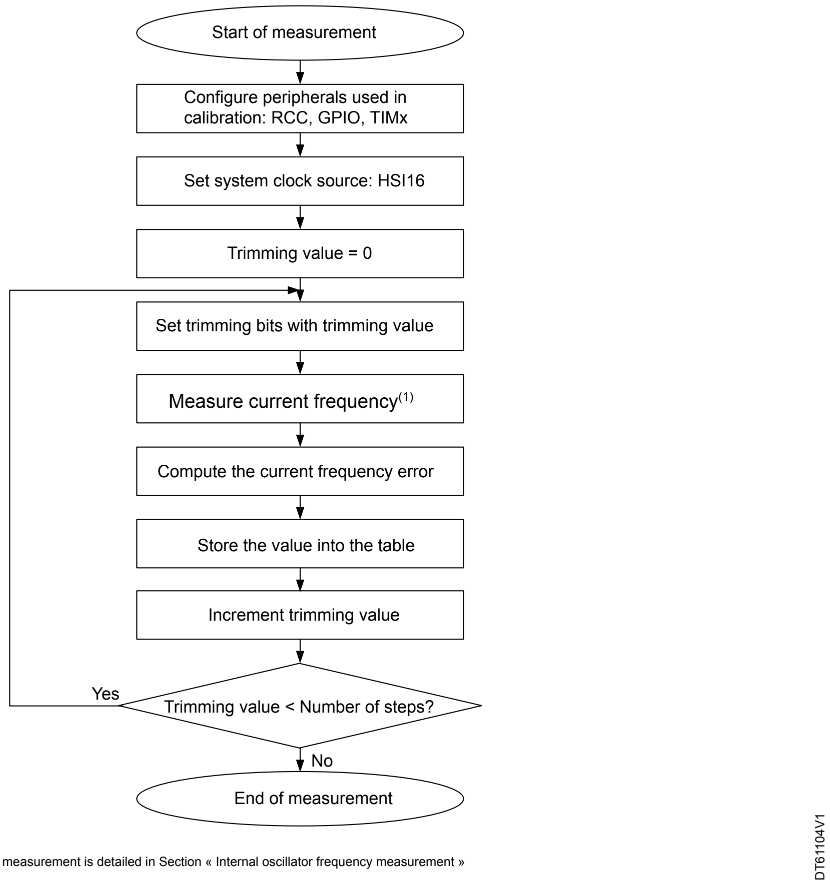
Figure 9. HSI48 trimming algorithm



3.3.4 Internal oscillator calibration using calibration curve

Both the minimal and fixed error methods can take some time due to the high number of measurements to run (for example 128 measurement for HSI16 with the minimal error method). First, for all trimming values, the difference between the corresponding and the requested frequency is measured and stored in a table. This method uses the same principle as the minimal error method. When it is necessary to calibrate the internal oscillator function HSI_CalibrateCurve(), only the actual frequency is measured and the appropriate value to compensate the difference is taken from the table.

Figure 10. Internal oscillator calibration: measuring the calibration table



(1) The frequency measurement is detailed in Section « Internal oscillator frequency measurement »

3.3.5 Internal oscillator frequency measurement

The internal oscillator frequency measurement is performed by Timer 16 capture interrupt. In the timer TIM16 ISR, an entire period of internal oscillator frequency is computed. The number of periods to be measured for each trimming value is configurable by the user in the *InternOscCalibration.h* file, as follows:

```
#define NUMBER_OF_LOOPS 50
```

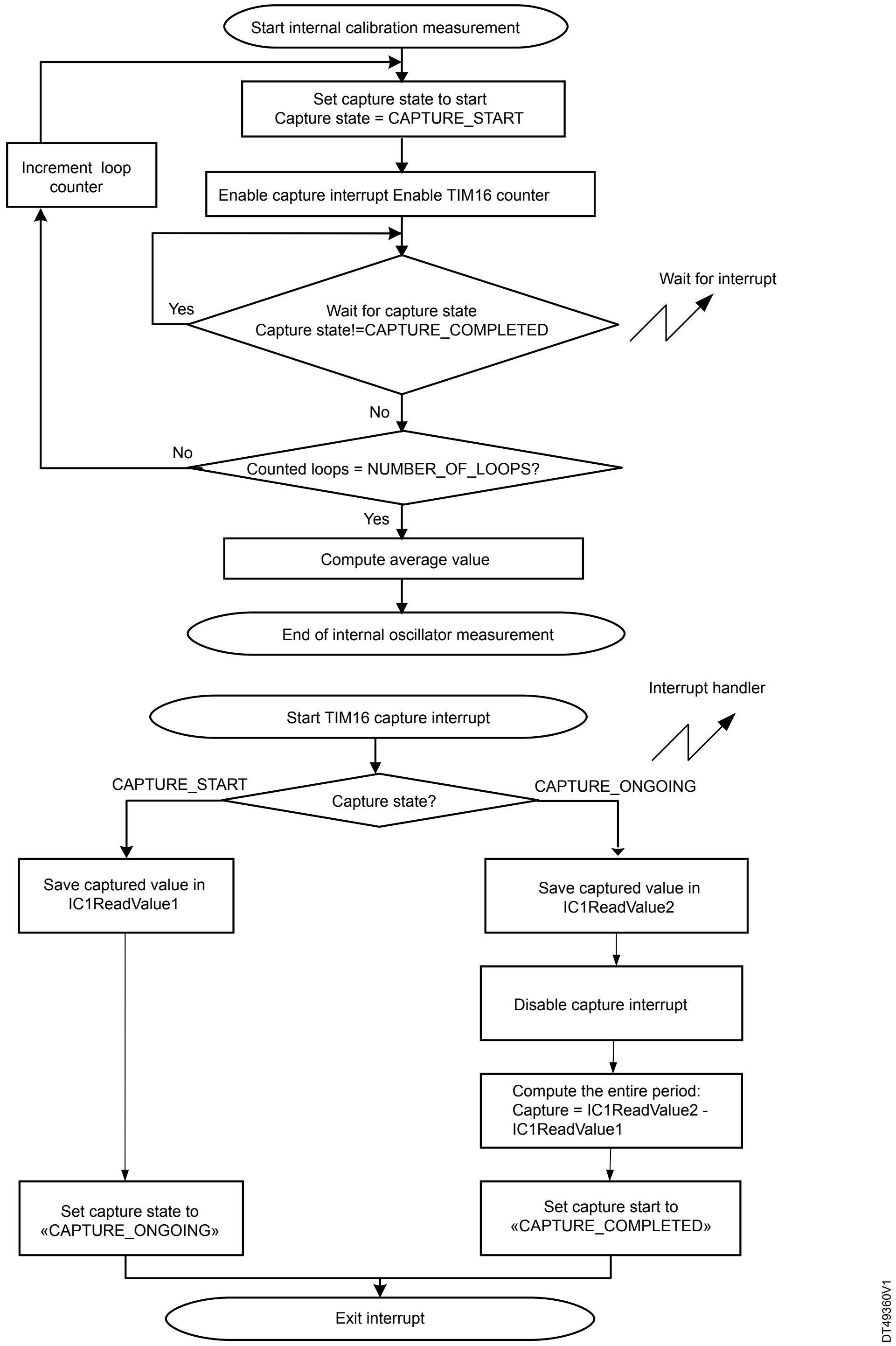
The averaging method is used to minimize frequency error measurements. So, if the counter of loops reaches NUMBER_OF_LOOPS, the average of all measured frequencies is computed.

Users can easily configure the frequency of the reference source. It is defined in the *InternOscCalibration.h* header file, as follows:

- If the LSE clock is used as the reference frequency, uncomment the line below to make sure the LSE is configured and internally connected to Timer input channel.
  ```
  #define USE_REFERENCE_LSE
  ```
- If the reference frequency is an external signal equal to 50 Hz, then comment the line above and define the reference frequency as shown below:
  ```
  #define REFERENCE_FREQUENCY (uint32_t)50 /* The reference frequency value in Hz */
  ```

The computation of the frequency measurements does not depend on the duty cycle of the source reference signal. It depends on its frequency since the capture 1 interrupt is configured to occur on every rising edge of the reference signal (refer to Figure 13. : Timing diagram of internal oscillator calibration).

Figure 11. Internal oscillator frequency measurement flowchart



3.4 Recommendations on the use of the calibration library

1. If the external signal frequency is lower than system clock / 65535, the TIM16 counter prescaler should be used to support low frequencies.
2. If the external signal frequency is higher than system clock / 100, TIM16 input capture prescaler (divider) should be used to support high frequencies.
3. It is recommended to stop all application activities before the calibration process, and to restart them after calling the calibration functions.
   Therefore, the application has to stop the communications, the ADC measurements and any other processes (except when using the ADC for the calibration, refer to Step 6 below).
   These processes normally use clock configurations that are different from those used in the calibration process. Otherwise, errors might be introduced in the application: errors while reading/sending frames, ADC reading errors since the sampling time has changed, and so on.
4. The internal RC oscillator calibration firmware uses the following peripherals: reset and clock control (for trimming internal RC oscillators), Timer 16 (for measuring internal RC oscillators). Therefore, it is recommended to reconfigure these peripherals (if used in the application) after running the calibration routine.
5. Real-time calibration versus temperature can be used when the ambient temperature changes noticeably while the application is running. The internal temperature sensor can be used with the ADC watchdog with two thresholds. Each time an ADC watchdog interrupt occurs, a new calibration process has to be performed and the two thresholds are updated according to the current temperature (this feature is not implemented in the firmware provided with this application note):
   Threshold_High = CurrentTemperatureValue + TemperatureOffset
   Threshold_Low = CurrentTemperatureValue – TemperatureOffset
6. It might happen that with change of operation conditions (such as the surrounding temperature) the calibration curve can change. From this reason it is recommended to measure from time to time again (or when the condition change) to keep working with correct values.

3.5 Calibration process performance

3.5.1 Duration of the calibration process

The duration of the calibration process depends on:

1. The frequency of the reference signal (prescaled value) “REFERENCE_FREQUENCY”
2. The number of measured periods per trimming value “NUMBER_OF_LOOPS”
3. The number of measured frequencies during the calibration process “number of steps”.

Once the peripherals are configured and ready (mainly the LSE oscillator), the duration of the calibration process is approximated by:

- Duration = (2 x (NUMBER_OF_LOOPS + 1) x number of steps) / REFERENCE_FREQUENCY

If the calibration process is run with a minimum frequency error for an HSI oscillator (`HSI_CalibrateMinError()`), the number of steps is equal to 128. If the LSE oscillator is used as the reference frequency (REFERENCE_FREQUENCY = LSE value / Input capture prescaler = 32768/8 = 4096 Hz) and the selected number of measured periods is 10, the calibration consumes approximately:

- Duration = (2 x 51 x 128) / 4096 = 3.19 s

The duration of the calibration process with a maximum allowed error is lower than or equal to the duration of calibration when using the minimum frequency error process.

Note: Multiplying by 2 in the duration formula above is due to the fact that there is no synchronization between the reference signal and the start of counting by the timer.

4 Low-speed internal oscillator measurement

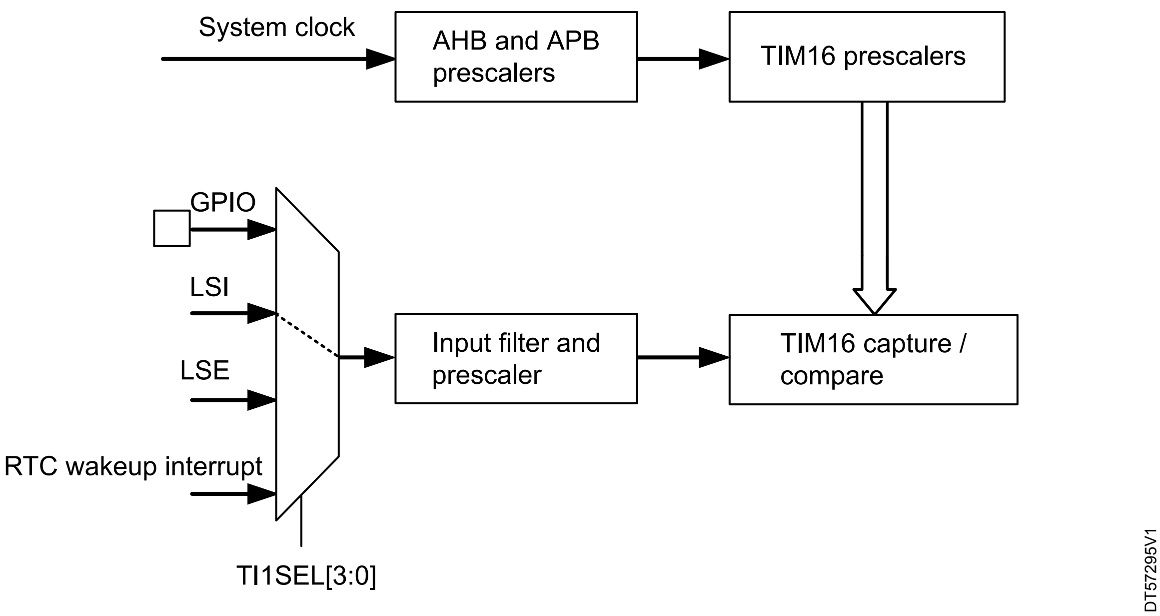
The internal LSI RC oscillator is a low-power clock source. In the STM32G0 series, an internal and indirect connection is provided between the internal RC oscillator LSI and the embedded timer TIM16 to facilitate the measurement procedure.

4.1 Measurement principle

The internal RC oscillator measurement procedure consists in running the timer counter using the HSI16 clock, configuring the internal RC oscillator LSI as the source of input capture signal of TIM16.

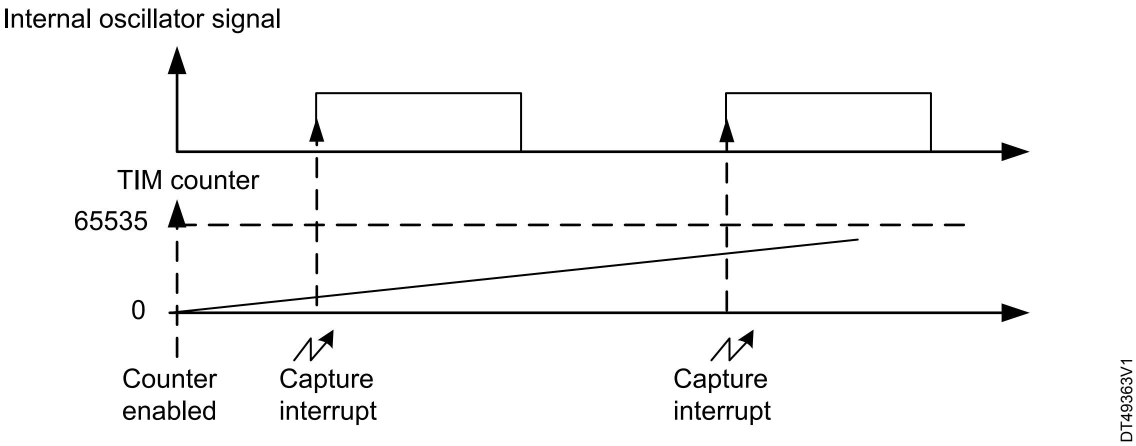
Below figure shows the configuration used to perform an LSI measurement.

Figure 12. LSI measurement configuration



After enabling the timer counter, when the first rising edge of the internal oscillator signal to be measured occurs, the timer counter value is captured and then stored in IC1ReadValue1. On the second rising edge, the timer counter is captured again and stored in IC1ReadValue2. The elapsed time between two consecutive rising edges of the clock represents an entire period. The figure below shows the timing diagram of an internal RC oscillator measurement.

Figure 13. **Timing diagram of an internal RC oscillator measurement**



The internal oscillator frequency value is computed as shown by the following formula:

- Internal oscillator frequency = HSI_Value / Capture

Where:

- HSI_Value is the HSI16 frequency value: typical value is 16 MHz
- Capture represents an entire period of internal RC oscillator LSI: IC1ReadValue2 - IC1ReadValue1.

As users can conclude from the above formula, the frequency measurement accuracy depends on the HSI16 frequency accuracy. Consequently, if a reference signal is available, users can run the internal RC oscillator calibration routine described in Section 3 Internal RC oscillator calibration before performing the internal RC oscillator measurement procedure.

The input capture prescaler can be used for better measurement accuracy so the formula above becomes:

- LSI_Frequency = InputCapturePrescaler * HSI_Value / Capture_Value

Note: If the HSE clock is available it is highly recommended to use it as the reference clock instead of HSI16 in order to benefit from its higher accuracy.

4.2 Description of the internal oscillator measurement firmware

The internal oscillator measurement firmware provided with this application note includes one C source file:

- LSIMeasurement.c performing LSI frequency measurement using LSI_FreqMeasure() function

The internal RC oscillator LSI is measured for a predefined number of periods. Then it returns the average value to minimize the error of the measured frequency.

Users can change this parameter (number of LSI periods) in the lsi_measurement.h file:

```
#define LSI_NUMBER_OF_LOOPS 10
```

5 Internal oscillator calibration/measurement example description

The X-CUBE-RC-CALIB demonstration shows the ability of the firmware to calibrate the internal RC oscillators (HSI16 and HSI48). Three examples of the internal RC oscillator measurement is given on the STM32G0 devices (NUCLEO_STM32G071, NUCLEO_STM32G0B1 and EVAL_STM32G081B). By default, the demo uses the minimum error method to calibrate the oscillators.

To run the calibration process that provides the frequency with fixed error or calibration curve, the user has to leave uncommented the line with requested method in the main.h file.

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

In the STM32G081B-EVAL example:

- After system reset, the HSI16 is selected to be used as the system clock source.
- The HSI16 is calibrated using the LSE oscillator as a reference clock.
- When the HSI16 oscillator has been calibrated, the PLL (clocked by HSI16) is configured to 64 MHz and used as the system clock source.
- After that, the LSI frequency is measured.
- Finally, the measured frequency of the different oscillators is displayed on the STM32G081B-EVAL board's LCD, as shown in the figure below.

The figure below shows the display of STM32G081B-EVAL board with results of calibration and measurement.

Figure 14. Internal oscillator calibration

Internal oscillator calibration
LSI value = 32.623 kHz
Values before calibration
HSI16 16.356 MHz
Values after calibration
HSI16 15.998 MHz

DT49364V1

In the NUCLEO_STM32G0B1 example, for the HSI48 oscillators, a CRS is used both for measurements and for calibration. To run the CRS, one of the following source references can be selected:

- a signal provided via a GPIO
- an LSE oscillator

The message below sent to the USART (implemented as virtual serial port on the Nucleo board)

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


LSI frequency : 31.758 kHz

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

This message can be displayed by some terminal software (while connecting Nucleo board to a PC) on STLINK virtual serial port. The HSI16, MSI and HSI48 oscillators are calibrated, then the LSI is measured.

6 Conclusion

Even if the internal RC oscillator is factory-calibrated, the user should calibrate them in the operating environment, when a high-accuracy clock is required in the application.

This application note provides two routines:

- **High-speed internal oscillator calibration:** how to fine-tune the oscillator to the requested value.
- **Low-speed internal oscillator (LSI) measurement:** how to get the “exact” LSI frequency value.

Several frequency sources can be used to calibrate the internal RC oscillator (HSI16/HSI48): LSE crystal, AC line, etc. Whatever the reference frequency source, the internal oscillator calibration principle is the same: a reference signal must be provided to be measured by a timer. The higher the accuracy of the reference signal frequency, the better the accuracy of the internal oscillator frequency measurement. The error is computed as the absolute value of the requested frequency value and the measured one for each trimming value. From this, the calibration value is calculated and then programmed in the trimming bits.

In order to improve the accuracy, the timer prescaler can be used to increase the ratio between the measured frequency and the reference frequency.

The second section of this application note is about the measurement of LSI oscillator. The internal connection between internal oscillators and embedded timers in the STM32G0 series family is used for this purpose. The timer is clocked using the system clock source and configured in the Input capture mode. The captured time between two consecutive rising edges of internal oscillator represents an entire period.

Revision history

Table 1. Document revision history

Date	Version	Changes
21-Nov-2018	1	Initial release.
17-Apr-2023	2	Updated: • Section Introduction, Section 3.3.4 Internal oscillator calibration using calibration curve, Section 5 Internal oscillator calibration/measurement example description, Section 6 Conclusion • Figure 1. Simplified clock tree Added Section 3.3.3 HSI48 calibration using CRS

Contents

List of figures