AN5676

Application note

How to calibrate internal RC oscillators on STM32U5 Series

Introduction

The STM32U5 Series microcontrollers embed two internal RC oscillators that can be selected as the system clock source. These are known as the HSI16 (high-speed internal) and MSI (multi-speed internal) oscillators that can drive two independent outputs: MSIS (for system clock) and MSIK (for some peripherals as kernel clock). The HSI16 oscillator has a typical frequency of 16 MHz. The MSI oscillator is a multispeed, low-power clock source.

The STM32U5 Series microcontrollers (named STM32U5 devices) have three secondary internal clock sources:

- LSI: 32 kHz (low-speed internal)
- HSI48: 48 MHz (high-speed internal) that can be used directly for USB, for RNG (true random number generator) and for SDMMC (SD/SDIO MMC card host interface).
- SHSI: 48MHz (+ jitter) internal securable RC oscillator dedicated to clock the SAES peripheral

The operating temperature has an impact on the accuracy of the RC oscillators. At 30 °C, the HSI16 accuracy is ±0.5%, the MSI accuracy is ±0.6% and the HSI48 accuracy is ±4%. In the temperature range of –40 °C to 125 °C, the accuracy decreases. To compensate the influence of temperature on internal RC oscillator accuracy, the STM32U5 devices include built-in features to calibrate the HSI16, MSI and HSI48 oscillators and measure the LSI oscillator frequency.

When a LSE (low-speed external) clock source at 32.768 kHz is available in the system, the frequency of the MSI oscillators can be automatically trimmed by hardware to reach an accuracy better than ±0.25%. This hardware auto-calibration with LSE is called PLL (phase-locked loop) mode. This application note does not describe the PLL mode and focuses only on the user trimming.

This document also details how to calibrate the HSI16, MSI and HSI48 internal RC oscillators, with the following methods:

- method based on finding the frequency with the minimum error
- method based on finding the maximum allowed frequency error
- method implementing a table of premeasured values and then searching in it for the appropriate change

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

The X-CUBE-RC-CALIB Expansion Package delivered with this application note, contains the source code to perform these internal oscillator calibrations, and all the embedded software modules required to run the examples.

1 STM32U5 system clock

The STM32U5 devices have the following clock sources that can be used to drive the system clock:

- HSI16: 16 MHz high-speed internal RC oscillator clock
- HSE: 4 to 50 MHz high-speed external oscillator clock
- MSI (MSIS): 100 kHz to 48 MHz multispeed internal RC oscillator clock
- PLL: 1 MHz to 160 MHz phase-locked loop that is clocked by HSI16, MSI or HSE oscillators

The HSI16 oscillator has a typical frequency of 16 MHz and consumes 150 µA.

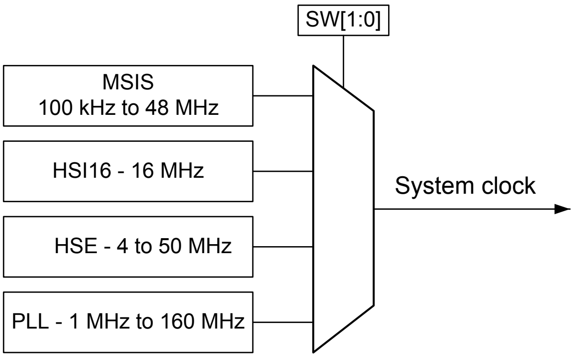
The MSI RC oscillator is based on four internal RC oscillators: MSIRC0 = 48 MHz, MSIRC1 = 4 MHz, MSIRC2 = 3.072 MHz, and MSIRC4 = 400 kHz. Each MSIRCx oscillator has four dividers: /1, /2, /3, and /4. The MSI provides together 16 frequency ranges that can be selected for the two outputs: MSIS (for system clock) and MSIK (for peripherals kernel clock). The MSI is designed to operate with a current proportional to the frequency (refer to the product datasheet for more details about the MSI power-consumption versus selected range), and to minimize the internal oscillator consumption when the CPU runs at low frequencies. The MSIS clock is used as a system clock after restart from reset or wakeup from Standby and Shutdown low-power modes. After wakeup from Stop mode, the MSI clock can be selected as system clock instead of the HSI16.

The HSI48 clock signal is generated from an internal 48 MHz RC oscillator and can be used directly for USB, RNG and SDMMC.

The internal RC oscillators (HSI16, MSI and HSI48) provide a low-cost clock source (no external components required). They also have a faster startup time and a lower power consumption than an external oscillator. The HSI16, MSI and HSI48 oscillators can be calibrated to improve their 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).

Note: The MSI internal RC oscillator can also be used as a backup clock source (auxiliary clock) if the external oscillator fails.

Figure 1. Simplified clock tree



The STM32U5 devices also embed the following secondary clock sources (that cannot be used as system clock):

- LSI: 32 kHz low-speed internal RC that can be kept running in Stop and Standby modes for the IDWG (independent watchdog), the RTC and the LCD. The LSI oscillator cannot be calibrated, but it can be measured to evaluate any frequency deviations (due to temperature and voltage changes).
- LSE crystal: 32.768 kHz low-speed external crystal RC that optionally drives the RTC (real time clock)
- HSI48: 48 MHz high-speed internal RC that is designed to provide a high-precision clock to the USB peripheral by means of a special CRS (clock recovery system) circuitry. It can also drive RNG and SDMMC.
- MSIK: multispeed internal RC oscillator clock used for peripheral kernel clocks (derived from MSIRCx oscillators)
- SHSI: 48 MHz internal RC oscillator dedicated to clock the SAES peripheral

2 Internal RC oscillator calibration

The frequency of the internal RC oscillators may vary from one device to another due to manufacturing process variations. For this reason, the MSI and the HSI16 RC oscillators are factory-calibrated by STMicroelectronics at T_A = 30 °C. After reset, the factory calibration value is automatically loaded in the internal calibration bits.

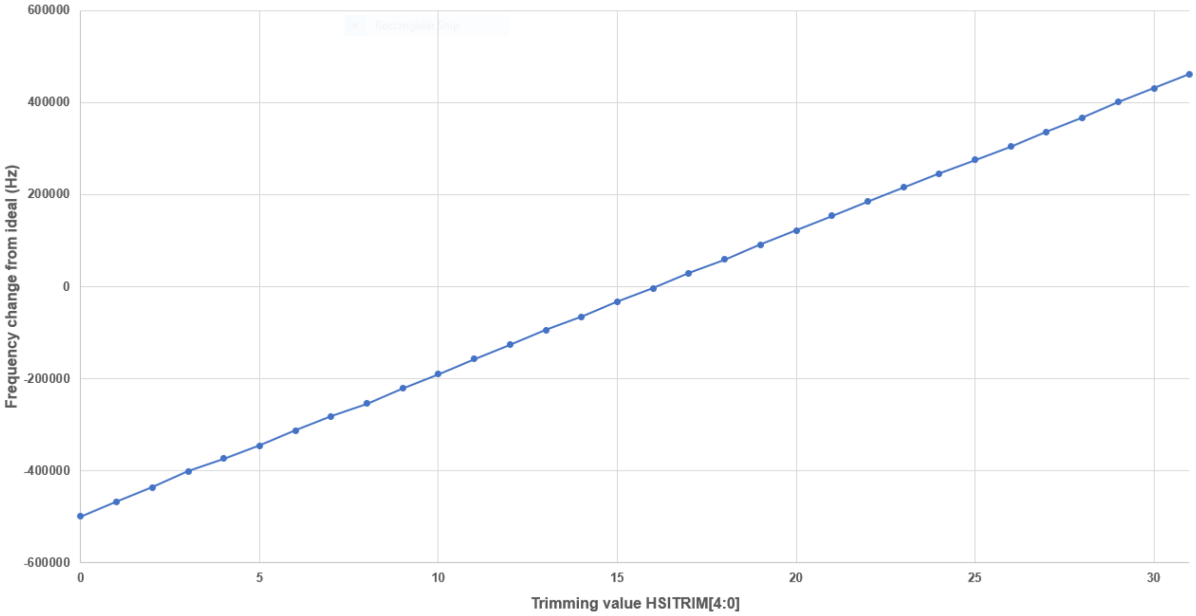
The frequency of the internal RC oscillators can be fine-tuned to achieve better accuracy with wider temperature and supply voltage ranges. The trimming bits are used for this purpose.

For the HSI16 oscillator, the calibration value is loaded in HSICAL[11:0] after reset. Five trimming bits in HSITRIM[4:0] are used for fine-tuning. The default trimming value is 16. An increase/decrease in this trimming value causes an increase/decrease in the HSI16 frequency. The HSI16 oscillator is fine-tuned by steps of 0.18% (around 29 kHz), as follows:

- Writing a trimming value in the range of 17 to 31 increases the HSI16 frequency.
- Writing a trimming value in the range of 0 to 15 decreases the HSI16 frequency.
- Writing a trimming value equal to 16 causes the HSI16 frequency to keep its default value.

The figure below shows an HSI16 oscillator behavior versus calibration value. The HSI16 oscillator frequency increases with the calibration value (calibration value = default HSICAL[11:0] + HSITRIM[4:0]).

Figure 2. HSI16 oscillator trimming behavior



For the MSIRCx oscillator (x = 0..3), the calibration value is loaded in MSICALx[4:0] after reset. Five trimming bits in MSITRIMx[4:0] give a wide tuning range. The calibration is based on adding the default MSICALx[4:0] (reset value) to MSITRIMx[4:0].

The result is stored in MSICALx[4:0] = default MSICALx[4:0] + MSITRIMx[4:0].

Example

Assuming the default MSI calibration value MSICALx[4:0] is 0x10:

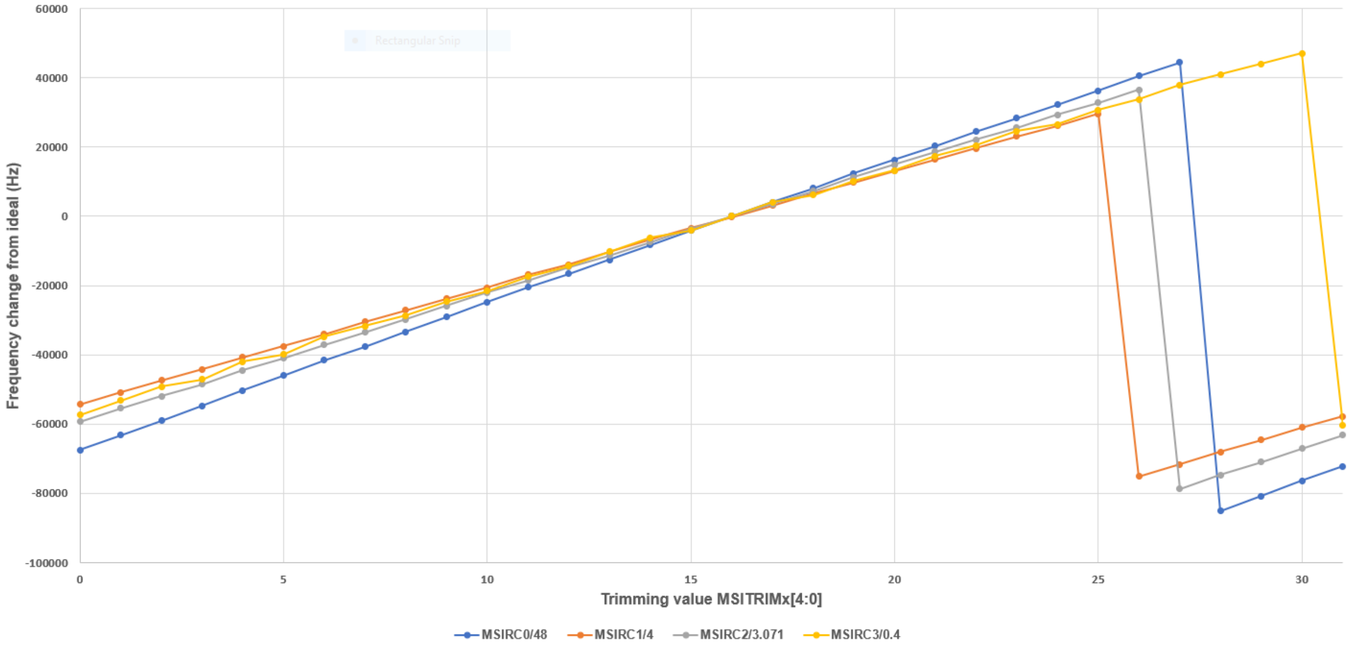
1. Writing a value between 0x01 and 0x0F in MSITRIM[4:0] leads to a calibration value MSICALx[4:0] in the range of MSICALx[4:0] = 0x10 + 0x01 = 0x11 to MSICALx[4:0] = 0x10 + 0x0F = 0x1F.

   These results are greater than 0x10 (default MSICALx[4:0] value) and consequently the MSIRCx frequency is increased by 1 step (0x11 - 0x10) to 15 steps (0x1F - 0x10).
2. Writing a value between 0x11 and 0x1F in MSITRIM[4:0] leads to calibration value MSICALx[4:0] in the range of MSICALx[4:0] = 0x10 + 0x11 = 0x01 to MSICALx[4:0] = 0x10 + 0x1F = 0x0F.

   These results are lower than 0x10 (default MSICALx[4:0] value) and consequently the MSIRCx frequency is decreased by 1 step (0x01) to 15 steps (0x0F).
3. Writing the default calibration value 0x10 in MSITRIM[4:0] leads to a calibration value MSICALx[4:0] equal to MSICALx[4:0] = 0x10 + 0x10 = 0x00, and consequently, the MSIRCx frequency is decreased by 16 steps (minimum frequency).

The figure below shows the MSIRCx behavior at 4 MHz versus MSICALx[4:0].

Figure 3. **MSIRCx oscillators trimming behavior**



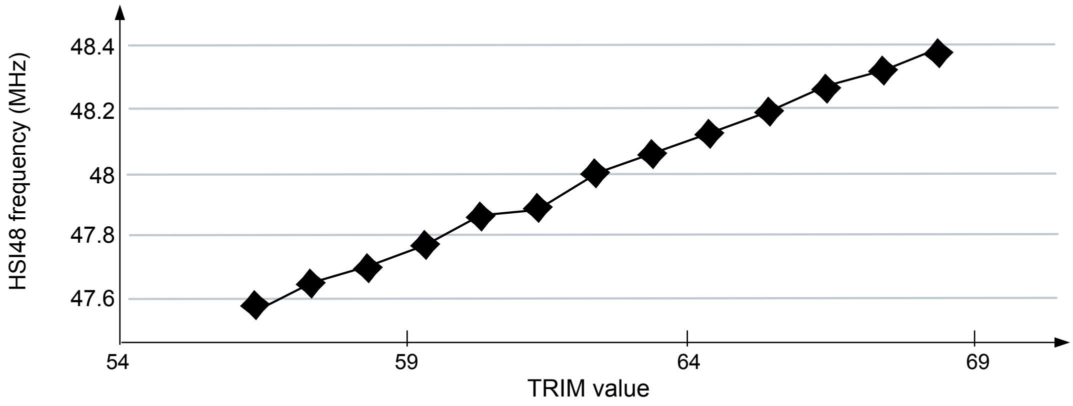
For the HSI48 oscillator, the calibration value is loaded in HSI48CAL[8:0] after reset. Six trimming bits TRIM[6:0] (in CRS_CR register) are used for fine-tuning. The default trimming value is 64. An increase/decrease in this trimming value causes an increase/decrease in the HSI48 frequency.

The HSI48 oscillator is fine-tuned in steps of 0.12% (around 57 kHz), as follows:

- Writing a trimming value, in the range of 65 to 127, increases the HSI48 frequency.
- Writing a trimming value, in the range of 0 to 63, decreases the HSI48 frequency.
- Writing a trimming value, equal to 64, causes the HSI48 frequency to keep its default value.

The figure below shows the HSI48 oscillator behavior versus the calibration value. The HSI48 oscillator frequency increases with the calibration value (calibration value = default HSI48CAL[8:0] + TRIM[6:0]).

Figure 4. **TRIM monotonicity**



2.1 Calibration principle

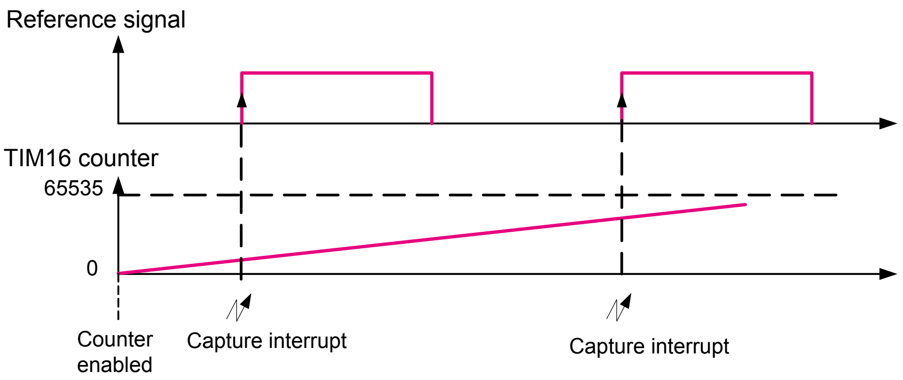
The calibration principle consists in the following steps:

1. Set the internal RC oscillator (that needs to be calibrated) as system clock.
2. Measure the internal RC oscillator (HSI16 or MSIRCx) frequency for each trimming value.
3. Compute the frequency error for each trimming value.
4. Set the trimming bits with the optimum value (corresponding to the lowest frequency error).

The internal oscillator frequency is not measured directly but is computed from the number of clock pulses counted using a timer compared with the typical 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 mains (refer to Section 2.2.2 Case 2: another source used as reference frequency).

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

Figure 5. **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 (HSI16 or MSIRCx), the real frequency generated by the internal RC oscillator versus the reference signal is given by:

measuredfrequency = (IC1ReadValue2 – IC1ReadValue1) × REFERENCE_FREQUENCY

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:

frequencyerror(Hz) = measuredfrequency – sysclockfrequency

Stabilization time and frequency overshoot/undershoot

The oscillator frequency changes from the original frequency to the final frequency after writing a trimming value. This frequency change takes some time to stabilize to new frequency value. The stabilization time for HSI16, HSI48, and MSI oscillators is approx. 15 µs (time to set the final frequency).

During the stabilization time, the frequency value can overshoot (when the new frequency value is larger than the original) or undershoot (when the new frequency value is smaller than the original) the final frequency. This overshoot/undershoot value depends on the magnitude of trimming step applied to the trimming value. The worst-case overshoot change (for full range trimming change) is approx. 2.5%. For middle-range change is approximately 1.5%.

The calibration process must implement waiting time between the trimming value change and the oscillator frequency reading.

After calculating the error for each trimming value, the algorithm determines the optimum trimming value (that corresponds to the nearest frequency to typical value) to be programmed in the trimming bits (refer to Section 2.3).

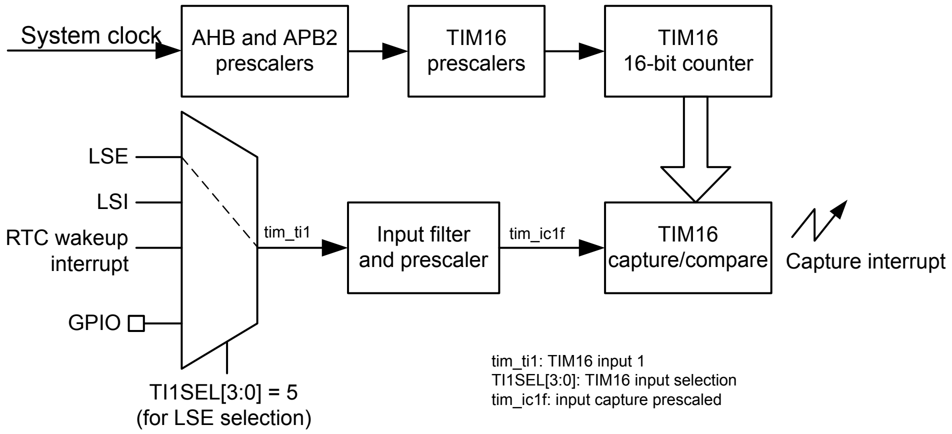
2.2 Hardware implementation

2.2.1 Case 1: LSE used as reference frequency

The STM32U5 devices offer the ability to connect internally the LSE to timer 16 channel 1. Thus, the LSE clock can be used as the reference signal for internal oscillator calibration, and no additional hardware connections are required. Only the LSE crystal/oscillator must be connected to OSC32_IN and OSC32_OUT.

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

Figure 6. **Hardware connection using LSE as the reference frequency**

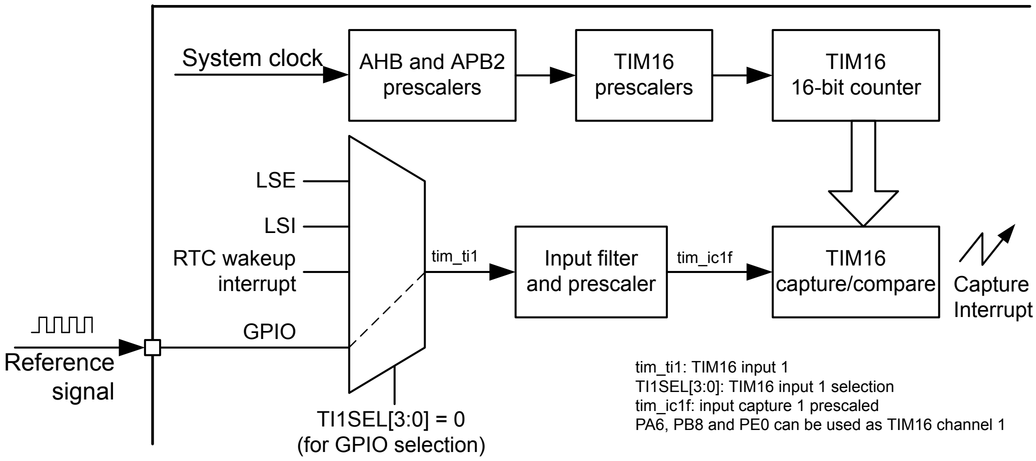


2.2.2 Case 2: another source used as reference frequency

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

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

Figure 7. **Hardware connection using external reference frequency**



tim_ti1: TIM16 input 1
TI1SEL[3:0]: TIM16 input 1 selection
tim_ic1f: input capture 1 prescaled
PA6, PB8 and PE0 can be used as TIM16 channel 1

2.3 Description of the internal oscillator calibration firmware

The internal RC oscillator calibration firmware X-CUBE-RC-CALIB provided with this application note includes the following major functions:

- `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 HSI16/MSI calibration with minimum error

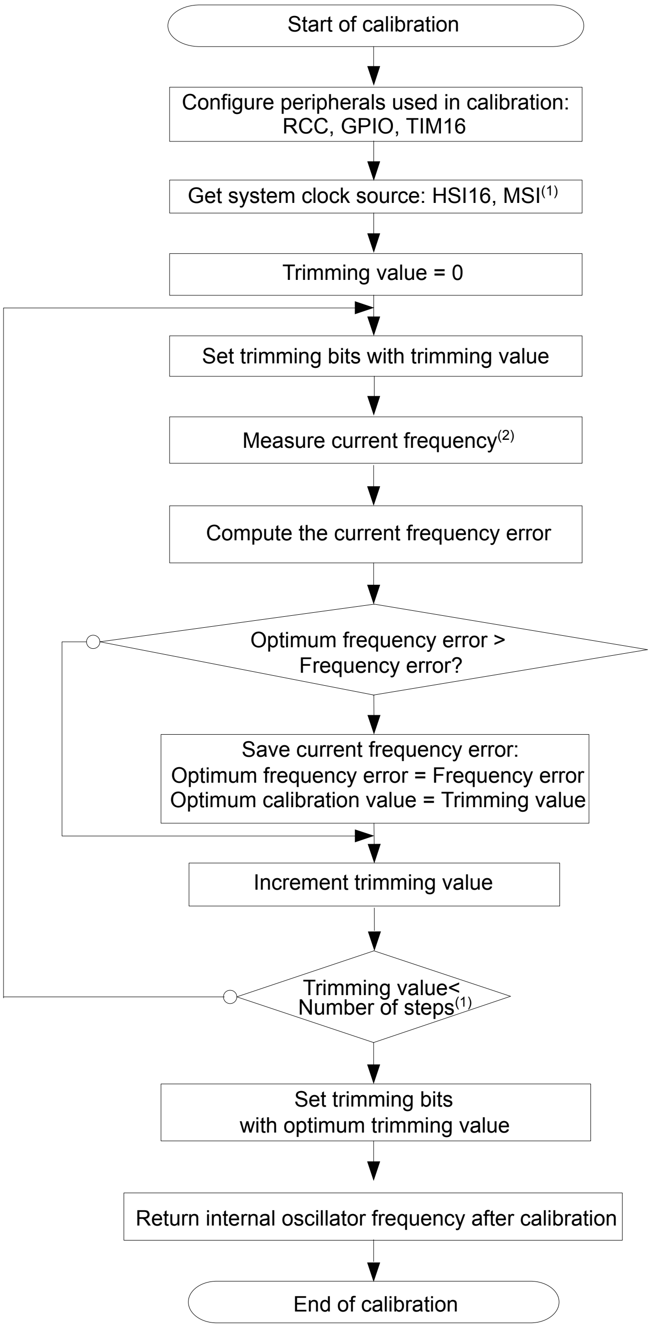
The `HSI16_CalibrateMinError()` and `MSI_CalibrateMinError()` functions calibrate the internal oscillator to have the frequency nearest to the typical value. These functions measure all frequencies for different trimming values and provide the trimming value that corresponds to the frequency with the minimum error. The trimming value obtained is programmed in the trimming bits.

After calibration, the `HSI16_CalibrateMinError()` and `MSI_CalibrateMinError` functions return the internal oscillator frequency value as an unsigned 32-bit integer `(uint32_t`).

The flowchart in the figure below details the algorithm for the following function example.

```
uint32_t InternOscAfterCalib = 0;
{
......
/* Get the internal oscillator (HSI16) value after calibration */
InternOscAfterCalib = HSI16_CalibrateMinError();
}
```

Figure 8. Internal oscillator calibration: finding the minimum frequency



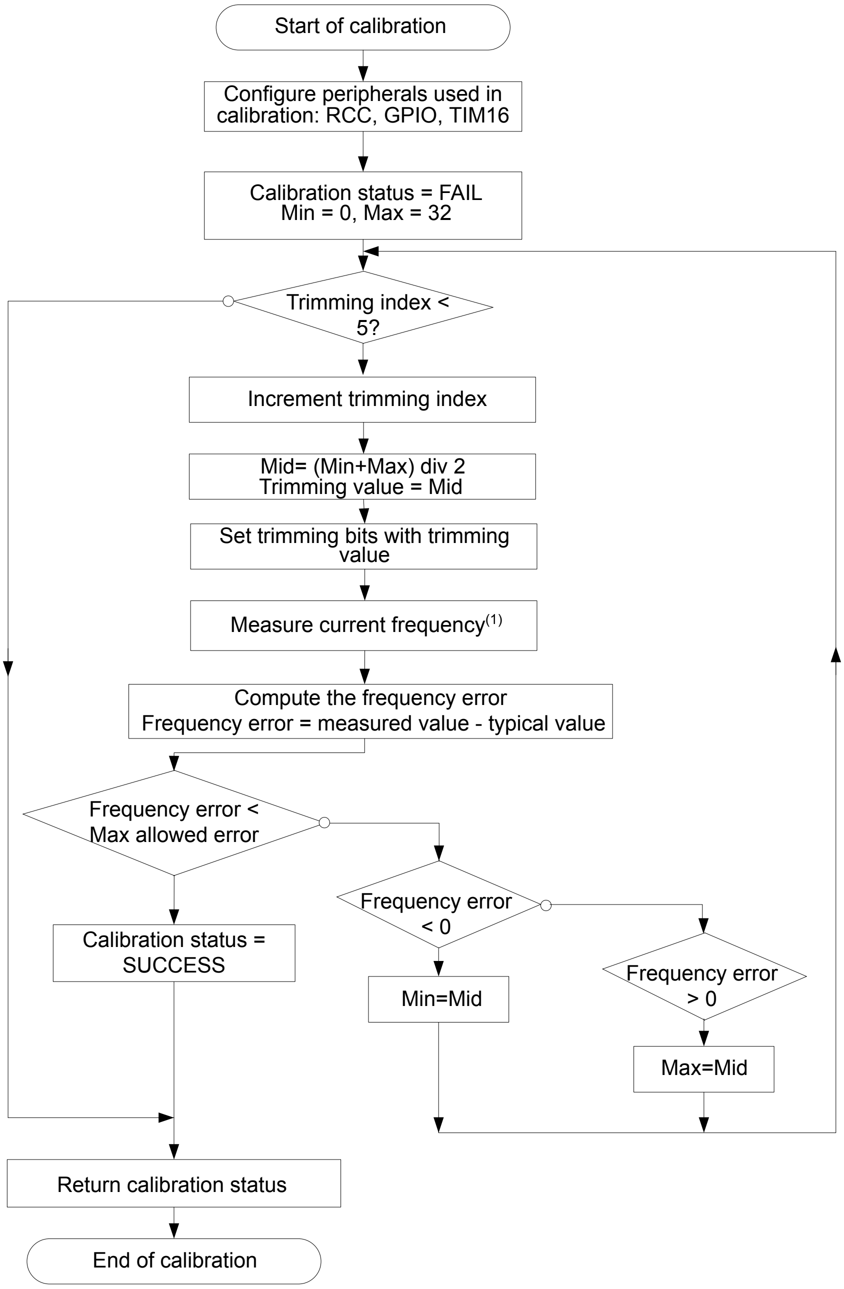
1. If the system clock source is HSI16 or MSI, the trimming bits have a 5-bit length and the number of steps is 32.
2. Frequency measurement is detailed in Section 2.3.4 .

2.3.2 HSI16/MSI calibration with fixed error

The `HSI16_CalibrateFixedError()` and `MSI_CalibrateFixedError()` functions are provided to calibrate the HSI16 and MSI oscillators with a maximum allowed frequency error in minimum time (by avoiding scan of all 32 trimming values). The allowed error is configured by the user as an absolute value given in hertz (the first parameter: `MaxAllowedError`). This function calibrates the HSI16 or MSI oscillator using a binary search algorithm (fastest way to find precise target frequency). If required fixed frequency error cannot be reached (too low error value required), the function sets the HSI16 or MSI to frequency with minimum error.

The flowchart in the figure below details the alogrithm for this function.

Figure 9. **HSI16/MSI calibration flowchart: maximum allowed frequency error**



1. Frequency measurement is detailed in Section 2.3.4 HSI16/MSI frequency measurement.

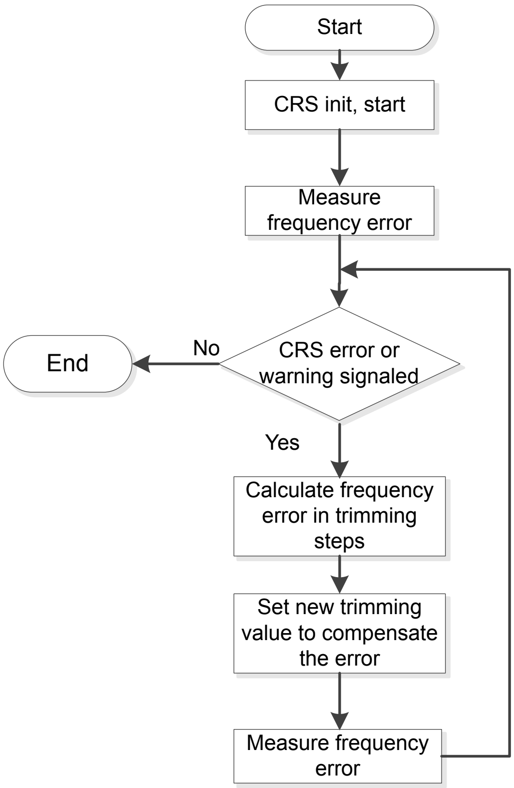
2.3.3 HSI48 calibration using CRS

The HSI48 can be measured the same way as the HSI16 or the MSI. The STM32U5 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 10. **HSI48 trimming algorithm**



2.3.4 HSI16/MSI frequency measurement

The internal oscillator frequency measurement is performed by TIM16 capture interrupt. In 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 `hsi16.c` or `msi.c` file as follows:

```
#define HSI16_NUMBER_OF_LOOPS 10/* Number of periods to be measured = 10 */
```

The user can easily configure the frequency of the reference source (defined in the header file `hsi16.c` or `msi.c`) as follows:

- If the LSE clock is used as reference frequency, uncomment the line below to make sure the LSE is configured and internally connected to TIM16 channel 1.

  ```
  #define USE_REFERENCE_LSE
  ```

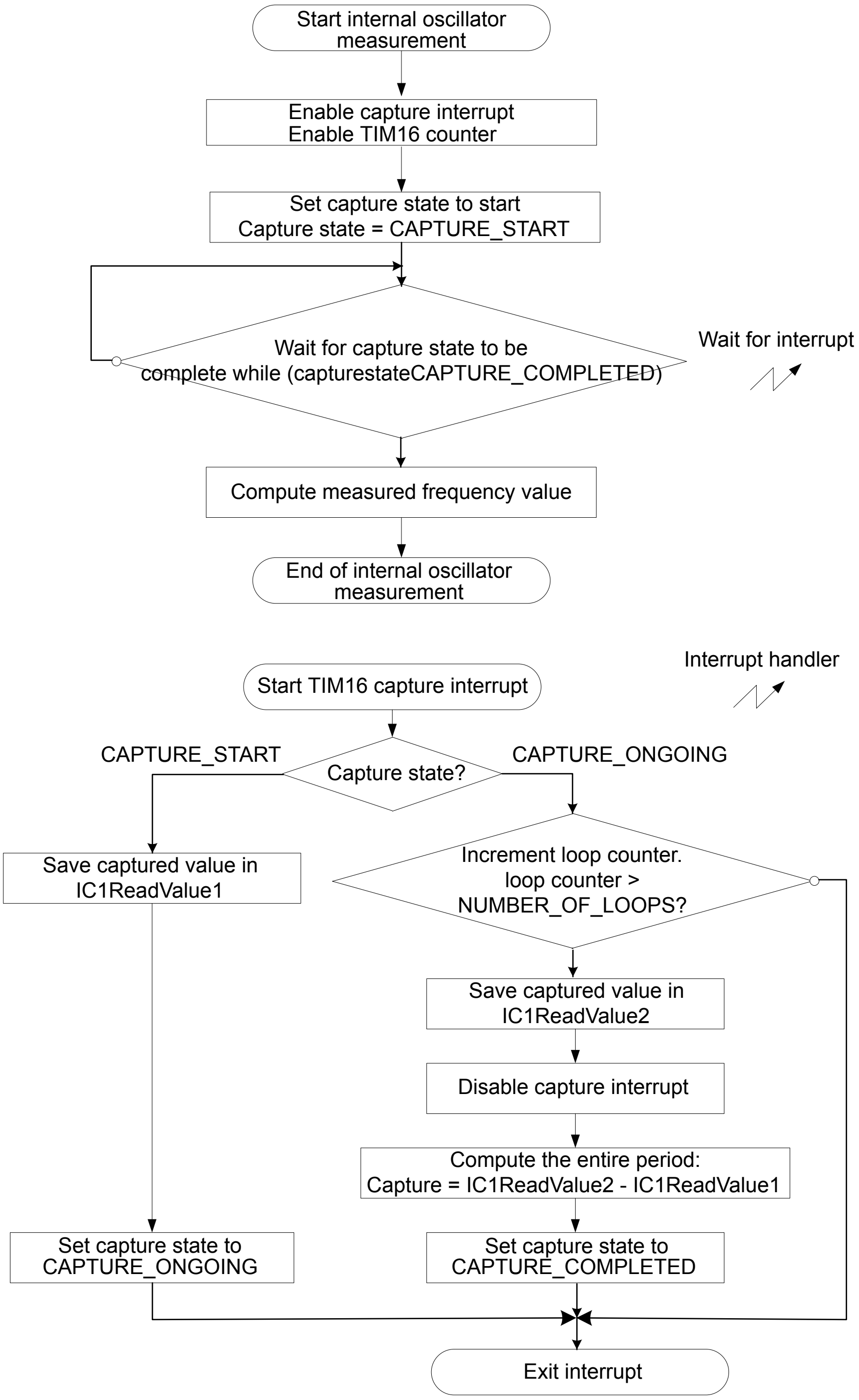
- If the reference frequency is another source frequency equal to 50 Hz, comment the line above and define the reference frequency as shown below.

  ```
  #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 the source reference signal frequency, since the capture 1 interrupt is configured to occur on every rising edge of the reference signal (refer to Section 2.1).

The figure below details the frequency measurement algorithm.

Figure 11. HSI16/MSI oscillator frequency measurement flowchart



2.3.5 HSI16/MSI calibration using calibration curve

Both previous methods (minimal and fixed error) can take a long time because many measurements need to be run (for example, 32 measurements for HSI16 when the minimal error method is used). For all trimming values, the difference between the corresponding and the requested frequency is measured and stored in a table. The calibration curve method uses the same principle as minimal error method. When it is necessary to calibrate the `HSI16_CalibrateCurve()` function,the calibration curve method measures the actual frequency only once, and searches in the table the appropriate value to compensate the difference.

For the MSI calibration, the `MSI_GetCurve()` and `MSI_CalibrateCurve()` functions must be used instead of `HSI16_GetCurve()` and `HSI16_calibrateCurve()`.

2.4 Recommendations on the use of the calibration library

Various recommendations on how to use the calibration library are listed below:

- If the external signal frequency is lower than the system clock/65535, the TIM16 counter prescaler must be used to support the low frequencies.
- If the external signal frequency is higher than system clock/100, the TIM16 input capture prescaler (divider) must be used to support the high frequencies. Input capture prescaler must be optimized also while using LSE as reference frequency.
- It is recommended to stop all the application activities before the calibration process, and to restart them after calling the calibration functions. The application therefore has to stop communications, ADC measurements and other processes (except when using the ADC for calibration, refer to the details on the real-time calibration below).
- These processes normally use clock configurations that are different from those used in the calibration process. Otherwise, errors may be introduced in the application (such as errors while reading/sending frames, or ADC reading errors since the sampling time has changed).
- The internal RC oscillator calibration firmware uses the RCC (reset and clock control) for trimming internal RC oscillators, and the TIM16 for measuring internal RC oscillators. It is recommended to reconfigure these peripherals (if used in the application) after running the calibration routine.
- A 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:
  - `Threshold_High = CurrentTemperatureValue + TemperatureOffset`
  - `Threshold_Low = CurrentTemperatureValue – TemperatureOffset`

  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.
- With a change of operating conditions (for example the surrounding temperature), the calibration curve may change. It is recommended to measure the frequency again from time to time (or when the conditions change), in order to keep working with correct values.

2.5 Calibration process performance

The duration of the calibration process depends on:

- the frequency of the reference signal (prescaled value): `REFERENCE_FREQUENCY`
- the number of measured periods per trimming value: `NUMBER_OF_LOOPS`
- 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 = (`NUMBER_OF_LOOPS` + 1) x number of steps / `REFERENCE_FREQUENCY`

If the calibration process is run with a minimum frequency error for the HSI16 oscillator (`HSI16_CalibrateMinError()`), the number of steps is equal to 32. 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 = (11 x 32) / 4096 = 86 ms

When running the calibration process for the MSI oscillator, the number of steps is equal to 32. 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 = (11 x 32) / 4096 = 86 ms

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.

3 Internal LSI oscillator measurement

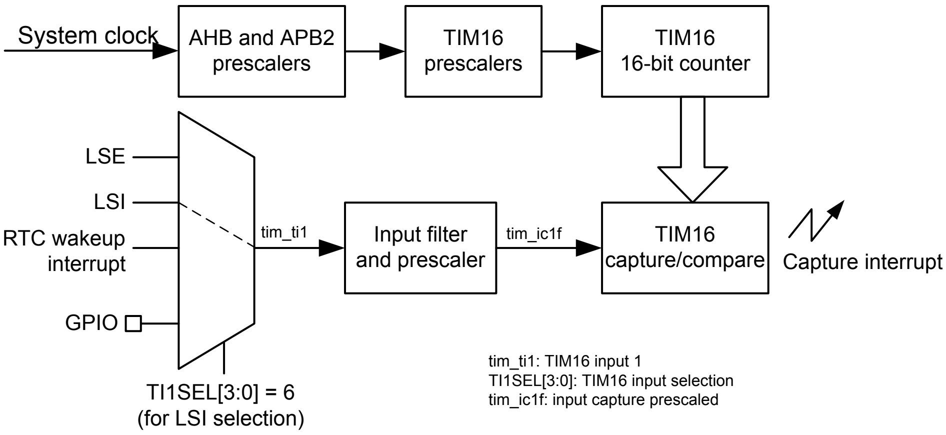
The internal LSI RC oscillator is a low-power and low-cost clock source. In the STM32U5 devices, an internal connection is provided between the internal RC oscillator (LSI or MSI) and the embedded timer (TIM16) to facilitate the measurement procedure.

Measurement principle

The internal LSI RC oscillator measurement procedure consists in running the timer counter using the HSI16 clock, configuring the timer in input-capture mode, and then connecting the internal LSI RC oscillator (that needs to be measured) to the timer.

The figure below shows the configuration used to perform the LSI measurement. The LSI can be connected internally to the TIM16 input 1 (TI1).

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 (see Section 2.1).

The internal oscillator frequency value is computed as HSI16_Value / Capture, where:

- HSI16_Value is the HSI16 frequency value (16 MHz typical).
- Capture represents an entire period of internal LSI RC oscillator: IC1ReadValue2 - IC1ReadValue1.

The frequency measurement accuracy depends on the HSI16 frequency accuracy. Consequently, if a reference signal is available, the internal RC oscillator calibration routine described in Section 2 can be run before performing the internal RC oscillator measurement procedure.

The input capture prescaler can be used for better measurement accuracy. The previous formula becomes:

LSI_Frequency = InputCapturePrescaler × HSI16_Value / Capture_Value

The same algorithm shown in Section 2.3.4 is used to measure the LSI oscillator frequency.

Note: TIM16 ISR is used for LSI measurement.

The X-CUBE-RC-CALIB expansion software provided with this application note includes one C source file: `LSIMeasurement.c` performing LSI frequency measurement using `LSI_FreqMeasure()` function

The internal LSI RC oscillator is measured for a predefined number of periods.

This parameter (number of LSI periods) can be changed in the `lsi_measurement.h` file:

```
#define LSI_PERIOD_NUMBERS 10
```

4 X-CUBE-RC-CALIB demonstration

The X-CUBE-RC-CALIB demonstration shows the ability of the firmware to calibrate the internal RC oscillators (HSI16, MSI and HSI48). An example of the internal RC oscillator measurement is given on the STM32U5 devices.

Before running the calibration routine, select the system clock source in `system_STM32U5xx.c`. If the HSI16 (or MSI) oscillator is selected as system clock source, the HSI16 (or MSI respectively) is calibrated. In this demo, the internal RC oscillator (MSI or HSI16) is calibrated using the LSE oscillator as a reference.

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
```

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) when running the demo with HSI16 selected as system clock source.

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

MSIRC0 before : 16.055 MHz
MSIRC0 after  : 15.989 MHz
MSIRC1 before : 04.043 MHz
MSIRC1 after  : 03.990 MHz
MSIRC2 before : 03.092 MHz
MSIRC3 after  : 03.078 MHz
MSIRC3 before : 00.399 MHz
MSIRC3 after  : 00.400 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.

5 Conclusion

Even if the internal RC oscillators are factory-calibrated, the user can calibrate them in the operating environment if a high-accuracy clock is required in the application.

This application note provides routines for:

- Multi and high-speed internal oscillator calibration: how to fine-tune the oscillator to the typical value
- Low-speed internal oscillator measurement: how to get the “exact” LSI/MSI frequency value

Several frequency sources can be used to calibrate the internal RC oscillators (HSI16, HSI48 and MSI) such as LSE crystal or AC line among others. Regardless of 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 typical frequency value and the measured one for each trimming value. After this, the calibration value is calculated and then programmed in the trimming bits.

The second part of this document presents the measurement of the LSI oscillator. The internal connection between internal oscillators and embedded timers in the STM32U5 devices is used for this purpose. The timer is clocked using the system clock source and configured in 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
27-Sep-2021	1	Initial release.

Contents

List of tables

List of figures

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