Using X-CUBE-RC-CALIB software to calibrate STM32WB Series internal RC oscillators

Introduction

The STM32WB 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 MSI (multi-speed internal) oscillators.

The operating temperature has an impact on the accuracy of the RC oscillators. At 25 °C, the HSI16 oscillator has a nominal accuracy of 0.25%, but 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 STM32WB Series microcontrollers have built-in features to allow users to calibrate the HSI16 oscillator and to measure the LSI (low-speed internal) oscillator frequency.

This application note focuses on how to calibrate STM32WB Series internal RC oscillators using the X-CUBE-RC-CALIB software. The HSI48 can also be trimmed even if it cannot be selected as the system clock.

The calibration of the external HSE oscillator is covered in *HSE trimming for RF applications using the STM32WB Series* application note (AN5042).

All the methods presented require an accurate reference signal.

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

1 General information

This document applies to STM32WB Series Arm®-based devices.

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

2 STM32WB Series system clock

The STM32WB Series microcontrollers have various clock sources that are used to drive the system clock:

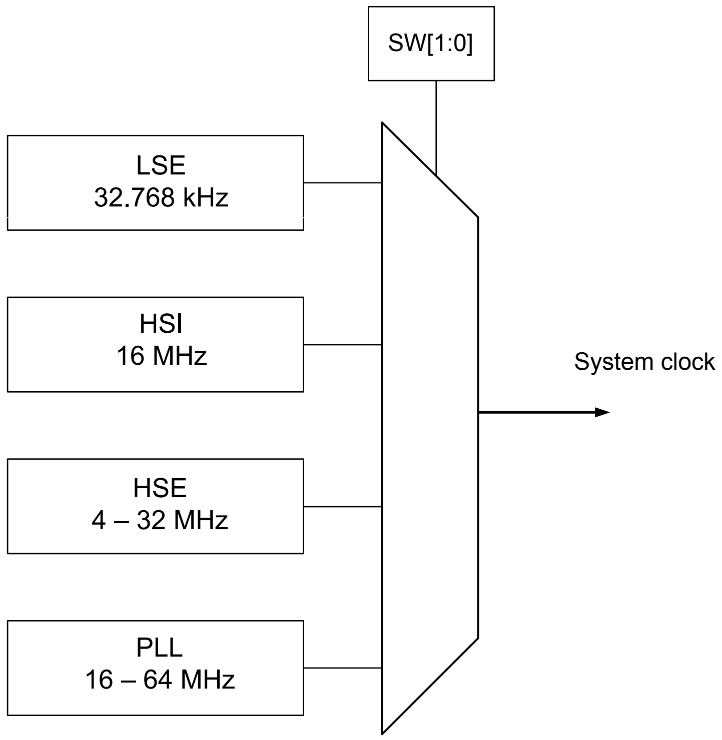
- A 16 MHz high-speed internal (HSI16) RC oscillator clock
- A 32 MHz high-speed external (HSE) oscillator clock
- A phase-locked loop (PLL) that is clocked by oscillator.
- A 32.768 kHz low-speed external (LSE) oscillator clock

Typically, the high-speed internal (HSI16) RC oscillator has a frequency of 16 MHz with a typical power consumption of 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 start-up 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 (of the order of tens of ppm).

Two 32-kHz low-speed internal (LSI1 and LSI2) RC oscillator clocks are designed as low-power clock sources which can run in stop and standby mode for the independent watchdog (IWDG), RTC, LCD, and RF wakeup. The LSI oscillators cannot be calibrated, but are measurable 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.

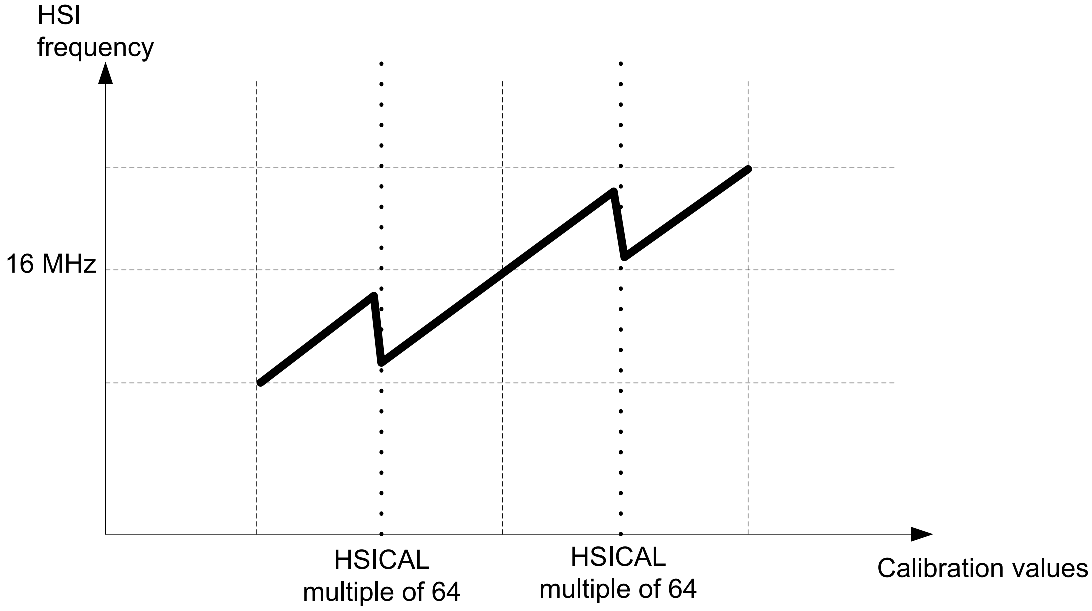
The frequency of the internal RC oscillators can be fine-tuned to achieve a better accuracy across wider temperature and supply voltage ranges. The trimming bits are used for this purpose.

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. A change in this trimming value causes a change in HSI16 frequency. The HSI16 oscillator is fine-tuned in steps of 0.3% (around 48 kHz).

Figure 2 shows the relation of the HSI frequency and the 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, an 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.

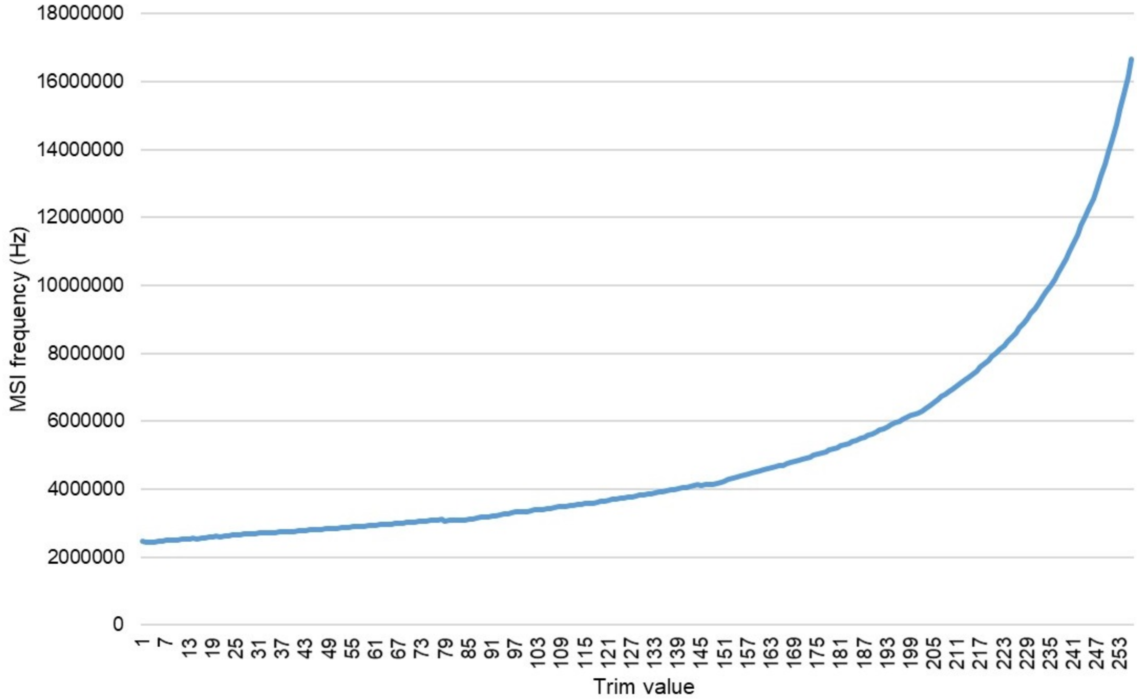
Figure 2. HSI16 oscillator trimming characteristics



For the MSI oscillator, the calibration value is loaded in the MSICAL[7:0] bits after reset. Seven trimming bits MSITRIM[6:0] are used for fine tuning. The default trimming value is 64. A change in this trimming value causes a change in MSI frequency.

Figure 3 shows the relation between the MSI frequency and the calibration value. The MSI oscillator frequency increases with the calibration value (calibration value = default MSICAL[7:0] + MSITRIM[6:0]).

Figure 3. **MSI trimming characteristics**

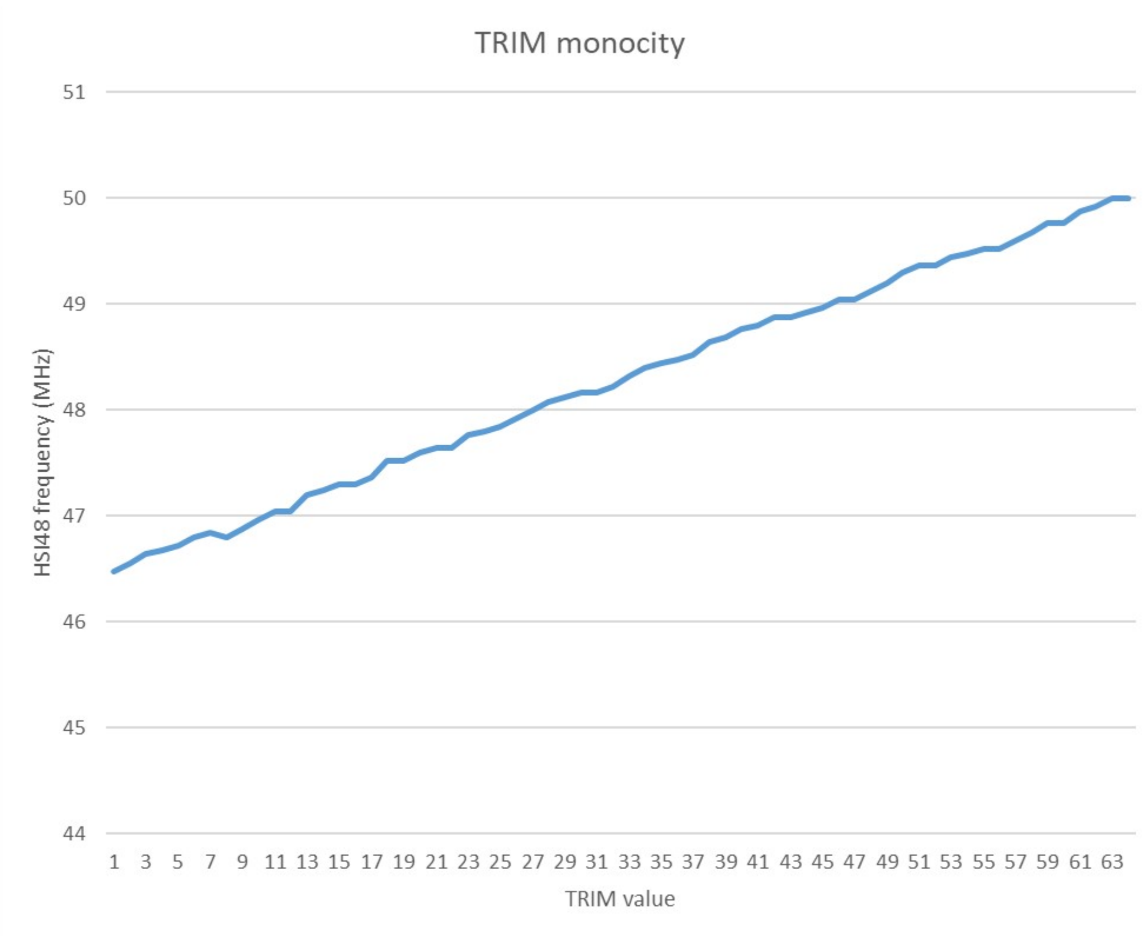


The HSI48 oscillatory is fine-tuned in steps of 1% (around 67 kHz).

- Writing a trimming value in the range 33 to 63 increases the HIS48 frequency
- Writing a trimming value in the range 0 to 31 decreases the HSI48 frequency
- Writing a trimming value equal to 32 causes the HSI48 frequency to keep its default setting.

Figure 4 shows the HSI48 oscillator behavior versus the calibration value. The HSI48 oscillator frequency increases with the calibration value (calibration value = default HSI48CAL[7:0] + TRIM[5:0]).

Figure 4. **HSI48 oscillator behavior versus the calibration value**

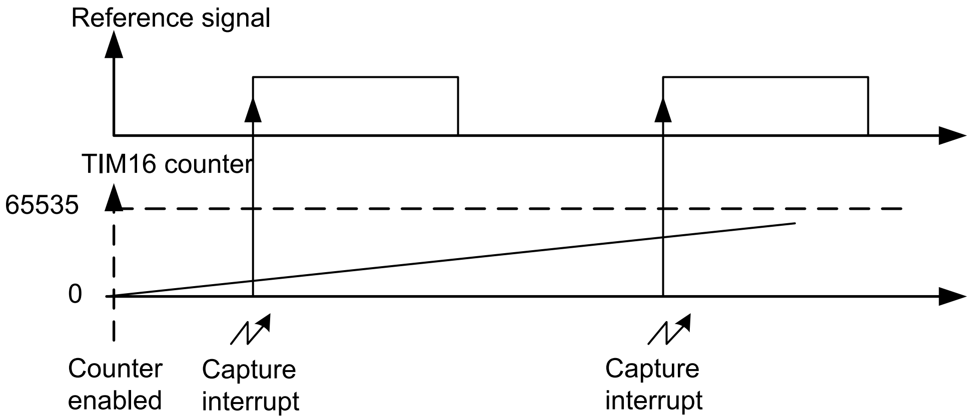


3.1 HSI16 frequency measurement principle

The internal oscillator frequency is not measured directly but it is computed from the number of clock pulses counted using a timer and comparing this count value 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 (for more information refer to Section 3.2.1 and Section 3.2.2).

Figure 5 shows how the reference signal period is measured in number of timer counts.

Figure 5. Timing diagram of internal oscillator calibration measurement



After enabling the timer counter, the timer counter period is captured when the first rising edge of the reference signal occurs 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:

- Measured frequency = (IC1ReadValue2 - IC1ReadValue1) x reference frequency

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

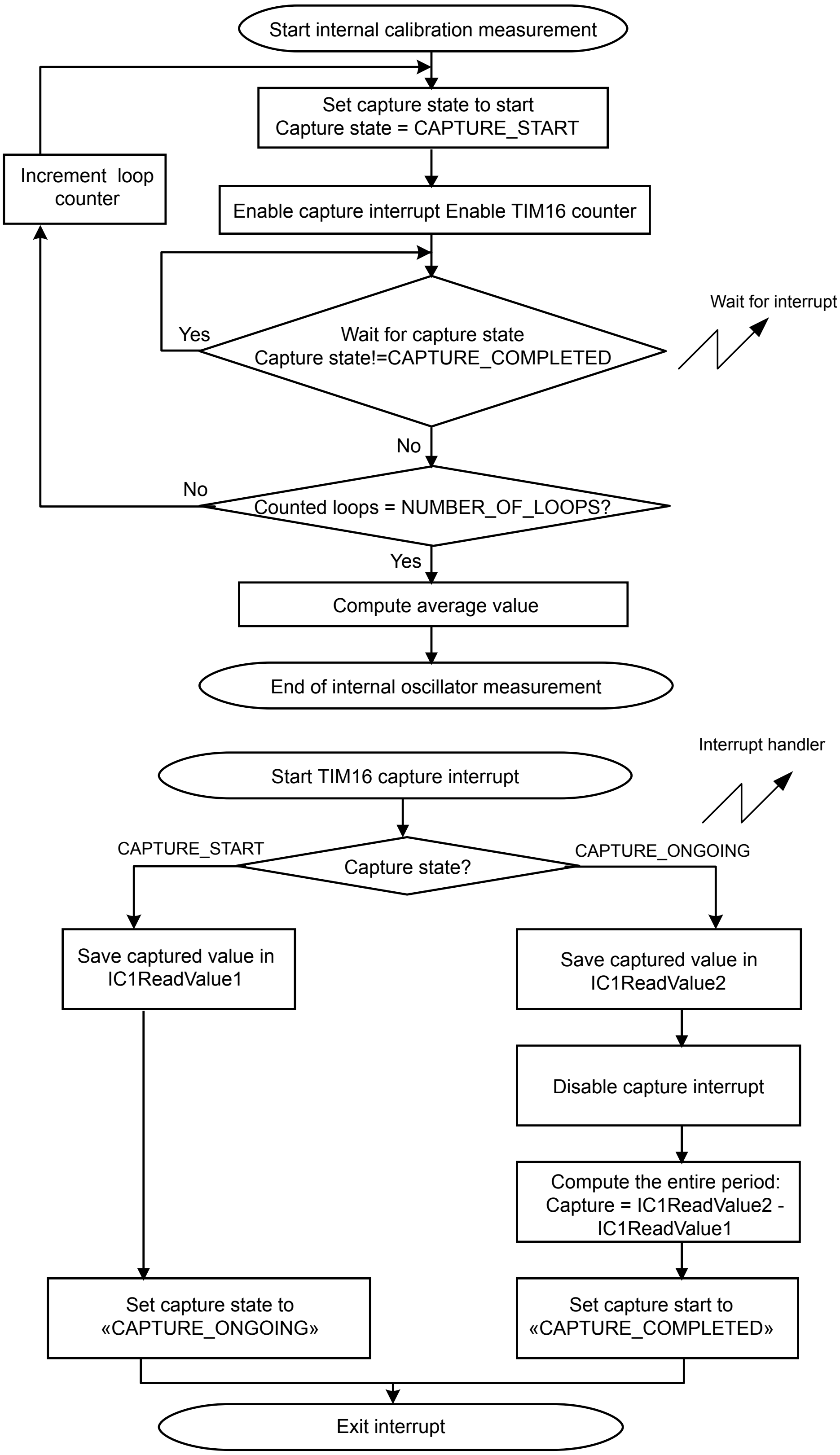
Hence, the internal oscillator frequency error is expressed as:

- Error (Hz) = Measured frequency - requested value.

The number of periods to be measured for each trimming value is configurable by the user.

The averaging method is used to minimize frequency error measurements. So, if the counter loop reaches NUMBER_OF_LOOPS, the average of all measured frequencies is computed. The internal oscillator frequency measurement is illustrated in Figure 6

Figure 6. Internal oscillator frequency measurement flowchart



3.2 Hardware implementation

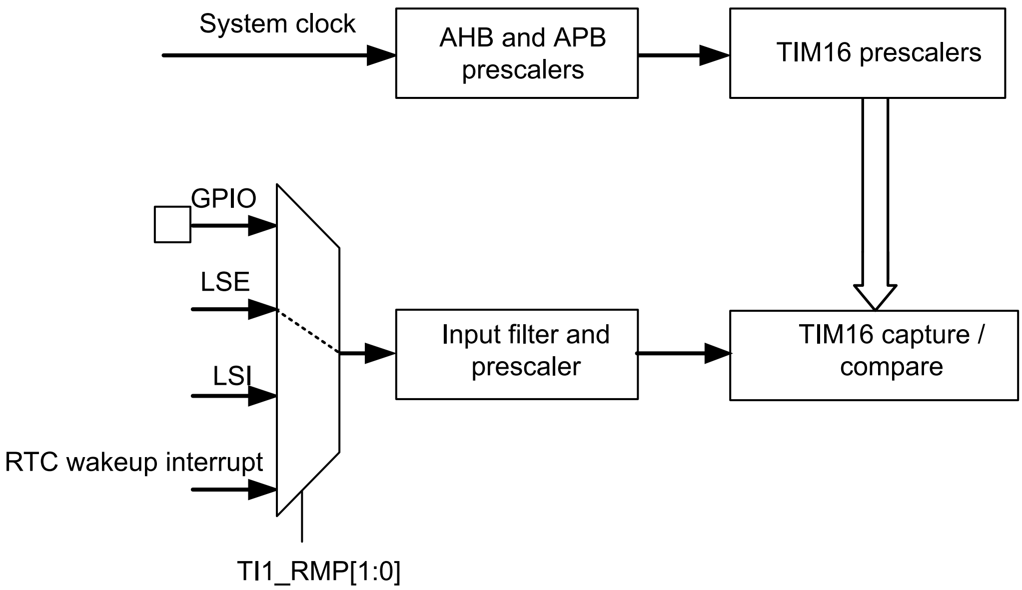
The STM32WB Series devices have the ability to connect internally and indirectly the internal or external oscillator to the dedicated timers (TIM16 and TIM17).

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

When the 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.

Figure 7 shows the hardware connections needed for the internal oscillator calibration, using LSE as an accurate frequency source for calibration.

Figure 7. Hardware connection using LSE as the reference frequency

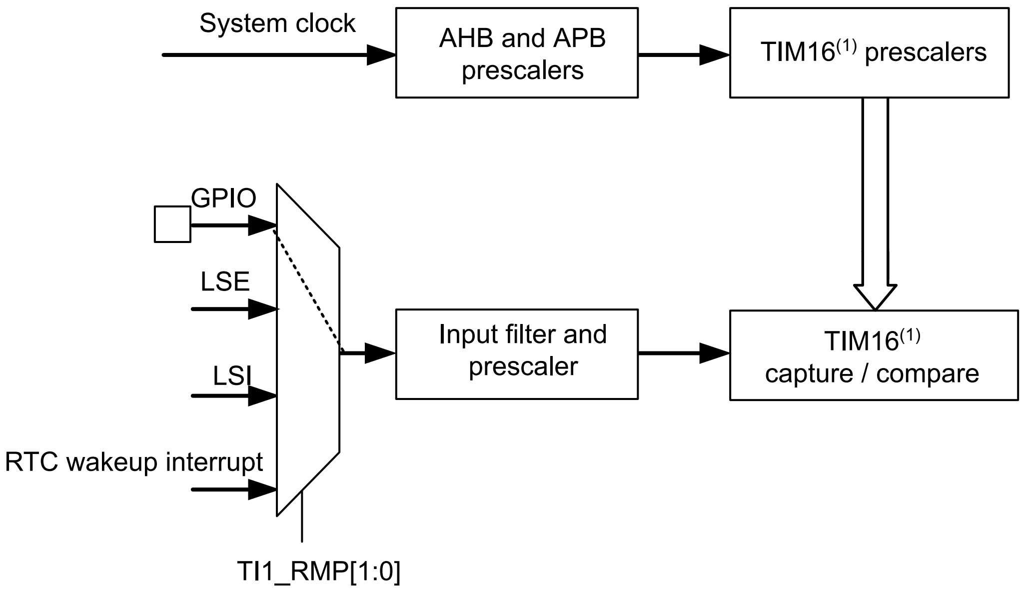


3.2.2 Case where another source is used as the reference frequency to measure HSI16

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

Figure 8 shows the reference signal connected to Timer 16 channel 1.

Figure 8. 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

3.2.3 Internal oscillator frequency measurement

The internal oscillator frequency measurement is performed by the 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
  */
  ```

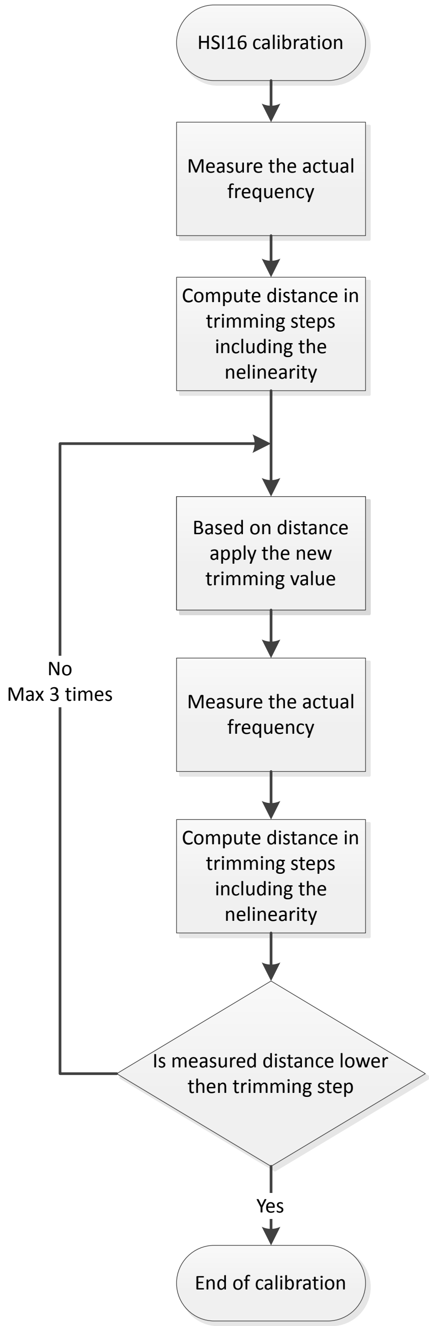
The computation of the frequency measurements does not depend on the duty cycle of the source reference signal. It depends on its frequency since the capture 1 interrupt is configured to occur on every rising edge of the reference signal (refer to Figure 6).

3.3 Description of the HSI16 internal oscillator calibration firmware

The goal of the trimming algorithm is to calibrate the oscillator in the shortest possible time. Figure 2 illustrates the calibration curve with the 64 multiple points linear exception. So that the trimming value to reach requested frequency can be easily computed. The only hurdle is to skip the non-linearities. With this approach, the final trim value is obtained with a maximum three measurement steps. If the error between two measured frequencies is lower or equal to the trimming steps, the calibration is terminated.

The flowchart Figure 9 below provides the function HSI_Calibrate(uint32_t* Frequency) algorithm.

Figure 9. HSI16 calibration process

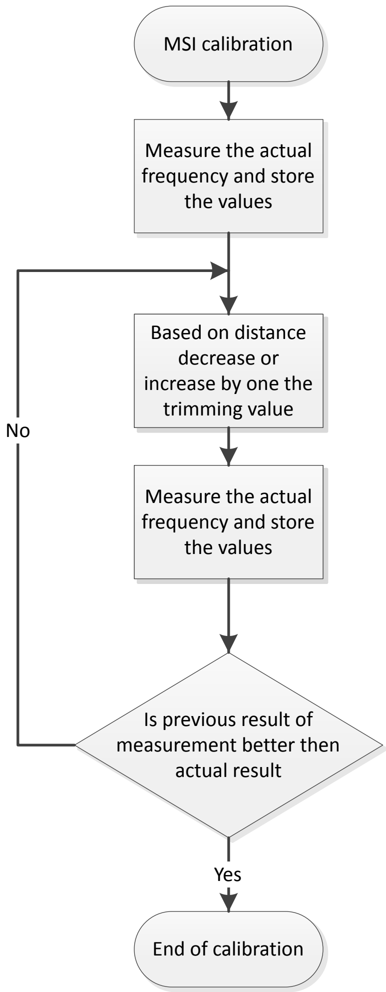


3.4 Description of the MSI internal oscillator calibration firmware

As visible from Figure 3. MSI trimming characteristics, the trimming curve is not linear. For this reason it is impossible to shorten the calibration time to determine the ideal trimming value by calculating the number trimming steps. As opposed to the HSI16 calibration, this process does not contain an negative step which simplifies the algorithm a little. The trimming value is increased or decreased by one depending the difference between actual and requested frequencies. As there is no trimming step, the decision to terminate the process is based on comparison of two sequential measures results. If the actual trim value brings worse frequency result than the previous one, the process does not continue and the previous trim value is taken as correct.

Figure 10 provides the algorithm for function MSI_Calibrate(uint32_t* Frequency).

Figure 10. MSI calibration process



3.5 Description of the HSI48 internal oscillator calibration firmware

HSI48 can be calibrated in the same way as HSI16. However, the STM32WB product family implements clock recovery system (CRS) that is able to automatically adjust the oscillator trimming based on the comparison with a selectable synchronization signal.

HSI48 implements an internal 16-bit step up / step down counter which increments the TRIM value until it reaches the required frequency value based on LSE reference clock or the USB SOF signal.

The HSI48 calibration using CRS can be run in a fully automatic mode. To speed up the process, the CRS can be used to measure the actual error and set the trim value with a pre-calculated value. This process can be repeated once or twice as the curve might not be linear. When the required frequency is reached, the automatic calibration can be activated to compensate for any changes such as compensating for temperature variations and improve the calibration accuracy, .

3.6 Recommendations on the use of the calibration library

- If the external signal frequency is lower than system clock / 65535, the TIM16 counter prescaler should be used to support low frequencies.
- If the external signal frequency is higher than system clock / 100, TIM16 input capture prescaler (divider) should be used to support high frequencies.
- Stop all application activities before the calibration process, and to restart them after calling the calibration functions.

  Therefore, the application stops 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 such as: errors while reading/sending frames, ADC reading errors since the sampling time has changed, and so on.
- 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, reconfigure these peripherals (if used in the application) after running the calibration routine.
- Use real-time calibration versus temperature 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
- The calibration curve may change over time due to changing operating conditions (such as the surrounding temperature). Environmental condition monitoring needs is to be implemented to track any changes over time to maintain correct operating values.

3.7 Calibration process performance

3.7.1 Duration of the calibration process

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 = (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

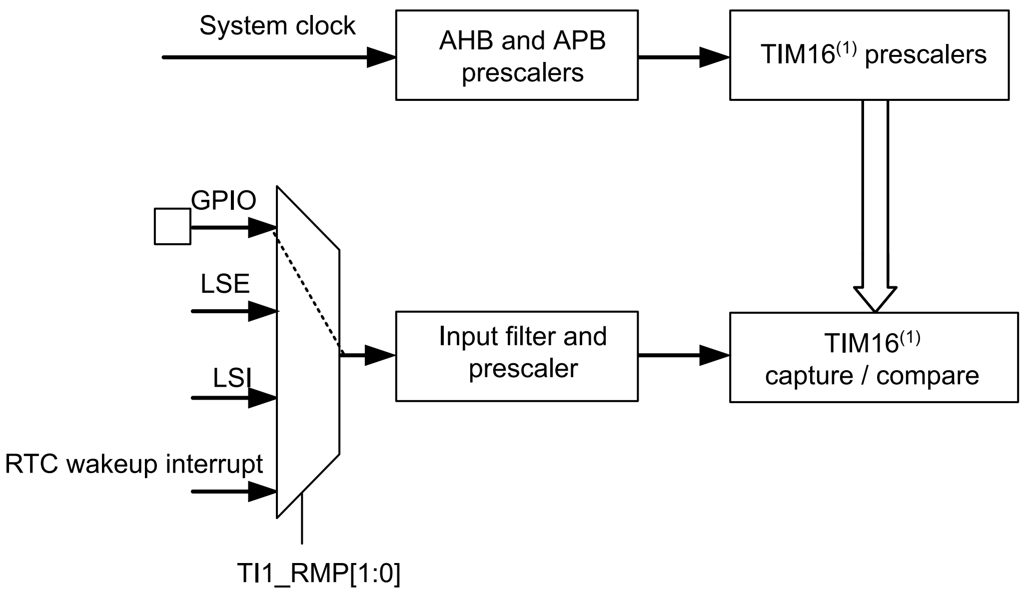
The internal LSI1 and LSI2 RC oscillators are low-power clock sources. Only one LSI source is used at any one time, resulting as LSI. The choise is done via appropriate bits in RCC_CSR register - for details see RM0434. In the STM32WB series, an internal and indirect connection is provided between the 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.

Figure 11 shows the configuration used to perform an LSI measurement.

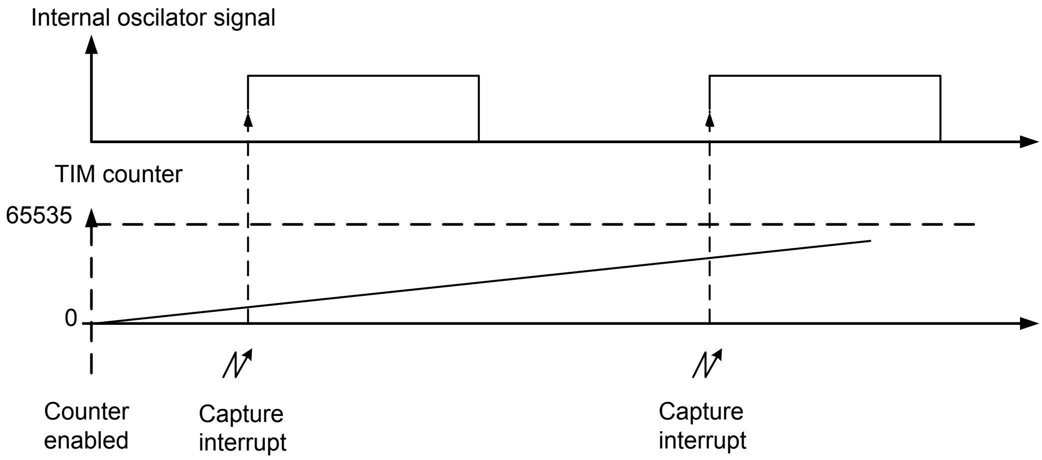
Figure 11. LSI measurement configuration



(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

After enabling the timer counter, the timer counter value is captured on the first measured rising edge of the internal oscillator signal and 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. Figure 12 shows the timing diagram of an internal RC oscillator measurement.

Figure 12. **Timing diagram of an internal RC oscillator measurement**



The internal oscillator frequency value is computed as shown in 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.

The formula states that 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 (see Section 5).

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, use this clock 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.

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

```
#define LSI_NUMBER_OF_LOOPS 10
```

5 Internal oscillator calibration / measurement example description

The example provided with this application note shows how the firmware calibrates the internal RC oscillator (HSI16) and explains how to use it to measure the internal RC oscillator (LSI) of the STM32WB Series devices.

In this example:

1. After system reset, the HSI16 is selected as the system clock source.
2. The HSI16 is calibrated using the LSE oscillator as a reference clock.
3. When the HSI16 oscillator has been calibrated, the PLL (clocked by HSI16) is configured to 64 MHz and used as the system clock source.
4. After that, the LSI frequency is measured.

6 Conclusion

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

This application note provides four routines:

- High-speed internal oscillator (HSI16) 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.
- Multi-speed internal oscillator (MSI) calibration: how to fine tune the oscillator to the requested value.
- RC 48 MHz internal clock (HSI48) calibration: how to fine tune the oscillator to the requested value.

Several frequency sources can be used to calibrate the internal RC oscillator (HSI16): LSE crystal, AC line and so on. 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 more accurately the internal oscillator frequency will be measured. 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 pre-scaler can be used to increase the ratio between the measured frequency and the reference frequency.

Section 4 Low-speed internal oscillator measurement describes the LSI oscillator measurement process followed by an implementation example in Section 5 Internal oscillator calibration / measurement example description. The internal connection between internal oscillators and embedded timers in the STM32WB Series 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
20-Jun-2019	1	Initial release.

Contents

List of figures