AN4067
Application note

Calibrating STM32F0x1, STM32F0x2 and STM32F0x8 lines internal RC oscillators

Introduction

Microcontrollers of STM32F0x1, STM32F0x2 and STM32F0x8 product lines, named STM32F0xx in this document, have internalRC oscillators that can be selected as the system clock source, and a High speed internal RC oscillator dedicated to ADC. These are known as the HSI (High-speed Internal 8 MHz), HSI14 (High-speed Internal 14 MHz) and HSI48 (High-speed Internal 48 MHz) oscillators.

The operating temperature has an impact on the accuracy of the RC oscillators. At 25 °C, the HSI, HSI14 and HSI48 oscillators have an accuracy of ±1% typically, 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 STM32F0xx microcontrollers have built-in features to allow users to calibrate the HSI and HSI14 oscillators and measure the LSI (Low-speed internal) oscillator frequency.

This application note focuses on how to calibrate internal RC oscillators: HSI, HSI14 and HSI48. Three methods are presented:

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

All three methods are implemented by providing an accurate reference signal.

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

Table 1 lists the microcontrollers and software concerned by this application note.

Table 1. Applicable products and software

Type	Part number and product line
Microcontrollers	STM32F0x1 line, STM32F0x2 line and STM32F0x8 line.
Software	STSW-STM32132

Contents

List of tables

List of figures

1 Applicable STM32F0xx product families

Table 2 provides the list of STM32F0xx product families and respective memory size to which this application note applies.

Table 2. Applicable STM32F0xx product families

STM32F0xx family	Flash memory size
STM32F030	up to 256 Kbytes
STM32F031	up to 32 Kbytes
STM32F038	up to 32 Kbytes
STM32F042	up to 32 Kbytes
STM32F048	up to 32 Kbytes
STM32F051	up to 64 Kbytes
STM32F058	up to 32 Kbytes
STM32F070	up to 128 Kbytes
STM32F071	up to 128 Kbytes
STM32F072	up to 128 Kbytes
STM32F078	up to 128 Kbytes
STM32F091	up to 256 Kbytes
STM32F098	up to 256 Kbytes

Note: All the information related to HSI48 is only applicable to the following product families: STM32F042, STM32F048, STM32F071, STM32F072, STM32F078, STM32F091 and STM32F098.

2 STM32F0xx system clock

The STM32F0xx microcontroller family has various clock sources that can be used to drive the system clock:

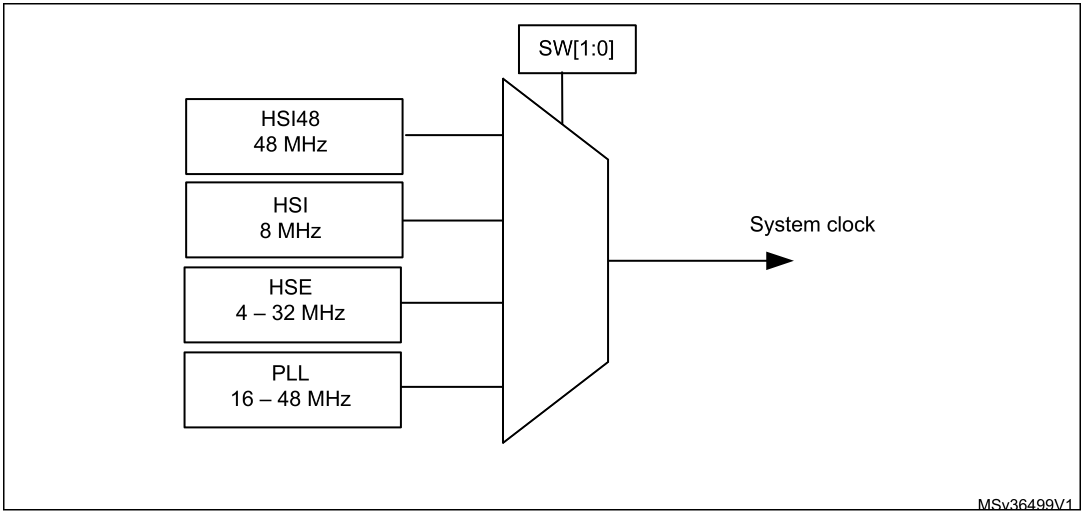
- An 8-MHz High-speed internal (HSI) RC oscillator clock
- A 4- to 32-MHz High-speed external (HSE) oscillator clock
- A Phase-locked loop (PLL) that is clocked by HSI or HSE or HSI48 oscillators
- A 48-MHz High-speed internal (HSI48) RC oscillator clock

Typically, the High-speed internal (HSI) RC oscillator has a frequency of 8 MHz or 48 MHz (HSI48) and consumes 80 µA (HSI) or 312 µA (HSI48).

The internal RC oscillator (HSI) 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 HSI 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).

Note: The internal 48 MHz RC oscillator is mainly used to provide a high precision clock to the USB peripheral by means of a special Clock Recovery System (CRS) circuitry, which uses the USB SOF signal or the LSE or an external signal to automatically adjust the oscillator frequency on-the-fly, through a very small steps.

Figure 1. Simplified clock tree



The STM32F0xx devices also have three secondary clock sources (that cannot be used as system clock sources):

- A 40 kHz Low-Speed Internal (LSI) RC which 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).
- A 32.768 kHz Low-speed external crystal (LSE crystal) which optionally drives the Real time clock (RTC)
- A 14 MHz high speed internal RC (HSI14) dedicated to ADC.

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, HSI and HSI14 RC oscillators are factory-calibrated by STMicroelectronics to have a 1% accuracy at T_A = 25 °C. HSI48 oscillator is factory calibrated to have ~3% accuracy at T_A = 25 °C.After reset, the factory calibration value is automatically loaded in the internal calibration bits.

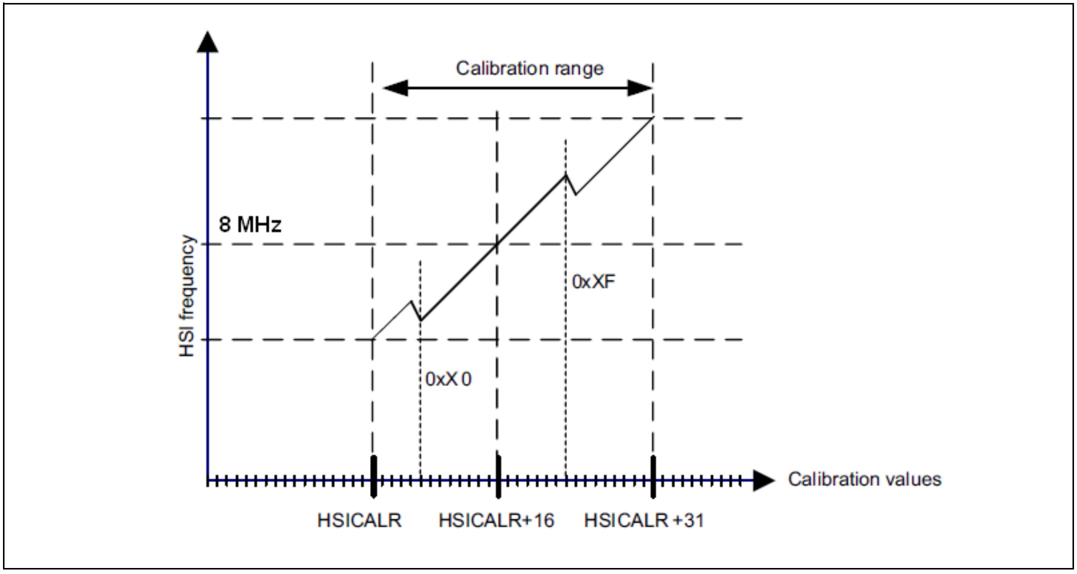
The frequency of the internal RC oscillators can be fine-tuned to achieve a better accuracy with wider temperature and supply voltage ranges. The trimming bits are used for this purpose.

For the HSI oscillator, the calibration value is loaded in HSICAL[7:0] bits after reset. Five trimming bits 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 HSI frequency. The HSI oscillator is fine-tuned in steps of 1% (around 40 kHz).

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

Figure 2 shows the HSI oscillator behavior versus the calibration value. The HSI oscillator frequency increases with the calibration value (calibration value = default HSICAL[7:0] + HSITRIM[4:0]), except at modulo 16. At these calibration values, the negative steps can reach three times the positive steps.

Figure 2. HSI oscillator trimming characteristics

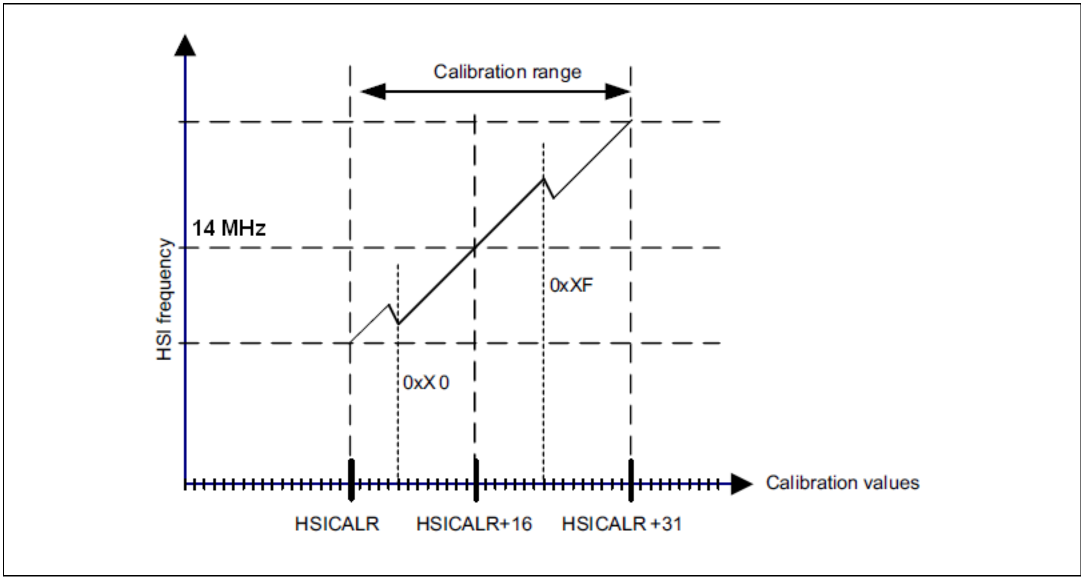


For the HSI14 (14 MHz) oscillator, The calibration value is loaded in HSI14CAL[7:0] bits after reset. Five trimming bits HSI14TRIM[4:0] are used for fine-tuning. The default trimming value is 16. An increase/decrease in this trimming value causes an increase/decrease in the HSI14 frequency. The HSI14 oscillator is fine-tuned in steps of 1% (around 50 kHz).

- Writing a trimming value, in the range of 17 to 31, increases the HSI14 frequency.
- Writing a trimming value, in the range of 0 to 15, decreases the HSI14 frequency.
- Writing a trimming value, equal to 16, causes the HSI14 frequency to keep its default value.

Figure 3 shows the HSI14 oscillator behavior versus the calibration value. The HSI14 oscillator frequency increases with the calibration value (calibration value = default HSI14CAL[7:0] + HSI14TRIM[4:0]), except at modulo 16. At these calibration values, the negative steps can reach three times the positive steps.

Figure 3. HSI14 oscillator trimming characteristics



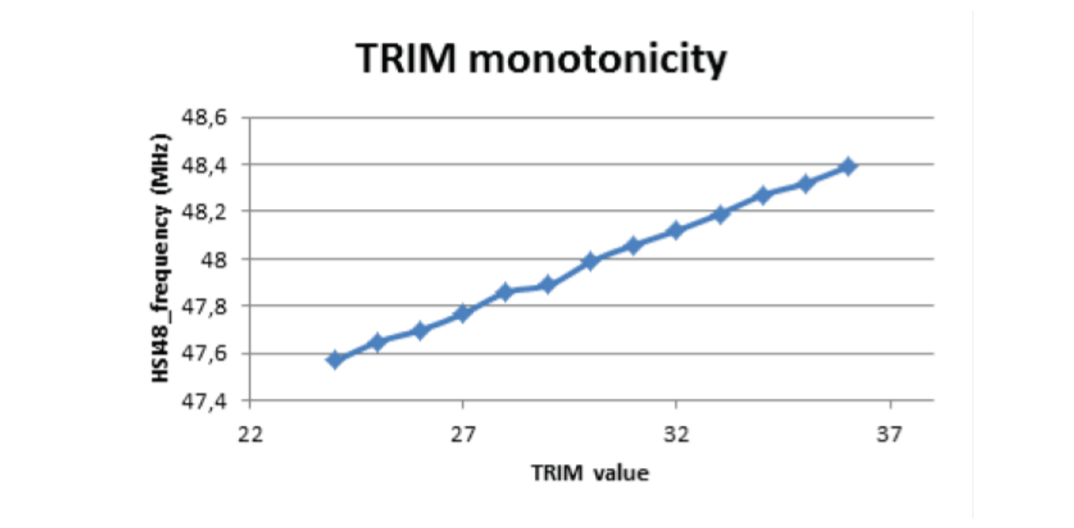
For the HSI48 oscillator, the calibration value is loaded in HSI48CAL[7:0] bits after reset. Six trimming bits TRIM [5:0] (in CRS control register) are used for fine-tuning. The default trimming value is 32. An increase/decrease in this trimming value causes an increase/decrease in HSI48 frequency.

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

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

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 trimming capability



3.1 Calibration principle

3.1.1 HSI calibration

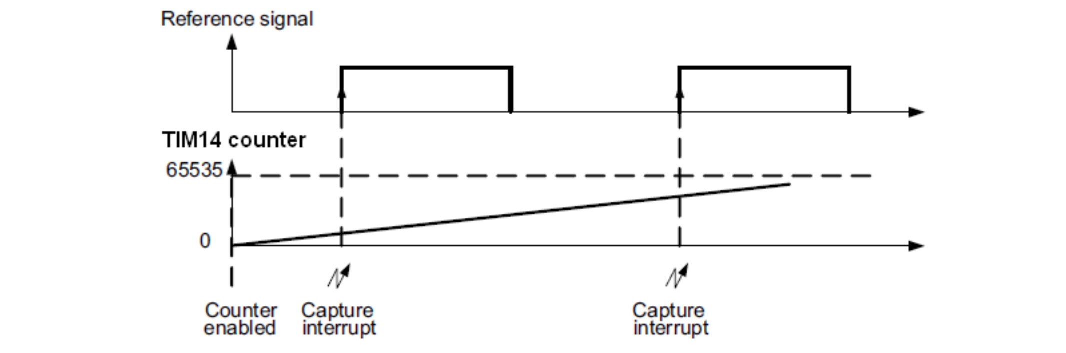
The calibration principle consists in:

1. Setting the HSI as the system clock source
2. Measuring the internal RC oscillator (HSI) 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 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 external signal (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



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 HSI), the real frequency generated by the internal RC oscillator versus the reference signal is given by:

$$\mathrm{Measuredfrequency} = (\mathrm{IC1ReadValue2} - \mathrm{IC1ReadValue1}) \times \mathrm{referencefrequency}$$

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:

$$\mathrm{Error(Hz)} = |\mathrm{Measuredfrequency} - \mathrm{typicalvalue}|$$

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 3.3: Description of the internal oscillator calibration firmware* for more details).

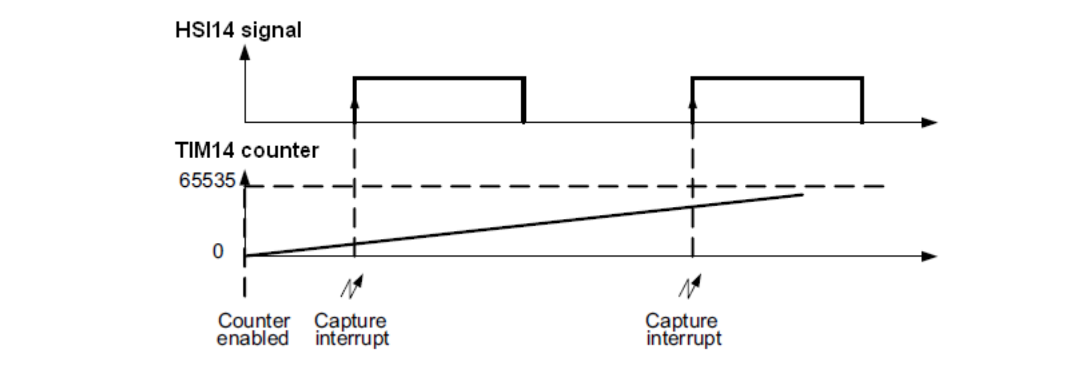
3.1.2 HSI14 calibration

The calibration principle consists in:

1. Setting the PLL, clocked by the HSI (after calibrating the HSI), as the system clock source.
2. Measuring the internal RC oscillator (HSI14) frequency for each trimming value.
3. Computing the frequency error for each trimming value.
4. Finally, setting the trimming bits with the optimum value (corresponding to the lowest frequency error).

Figure 6 shows how the reference signal period is measured in number of timer counts.

Figure 6. Timing diagram of internal oscillator calibration HSI14



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.

The real frequency generated by the internal RC oscillator HSI14 is given by:

$$\text{Measuredfrequency} = ((\text{SystemClock}) / (\text{IC1ReadValue2} - \text{IC1ReadValue1}))$$

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:

$$\text{Error}(\text{Hz}) = |\text{Measuredfrequency} - \text{typicalvalue}|$$

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

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

In order to improve the accuracy, the timer prescaler can be used to increase the ratio between HSI14 frequency and the reference frequency.

3.1.3 HSI48 calibration

HSI48 can be calibrated in the same way as HSI or HSI14. However, STM32F0xx product family implements CRS (Clock recovery system) that is capable of doing automatic adjustment of the oscillator trimming based on the comparison with a selectable synchronization signal.

Internally HSI48 implements a 16-bit down/up counter which step by step increments or decrements of TRIM value until it reaches the expected frequency value based on LSE reference clock or the USB SOF signal.

The HSI48 calibration using CRS can be run in fully automatic mode. To speed up the process, the CRS can be used for measurement of the actual error, and set trim value with pre-calculated value. This process can repeated once or twice as the curve might not be linear. When requested frequency is reached automatic calibration can be activated for further smooth calibration, e.g. compensating temperature changes.

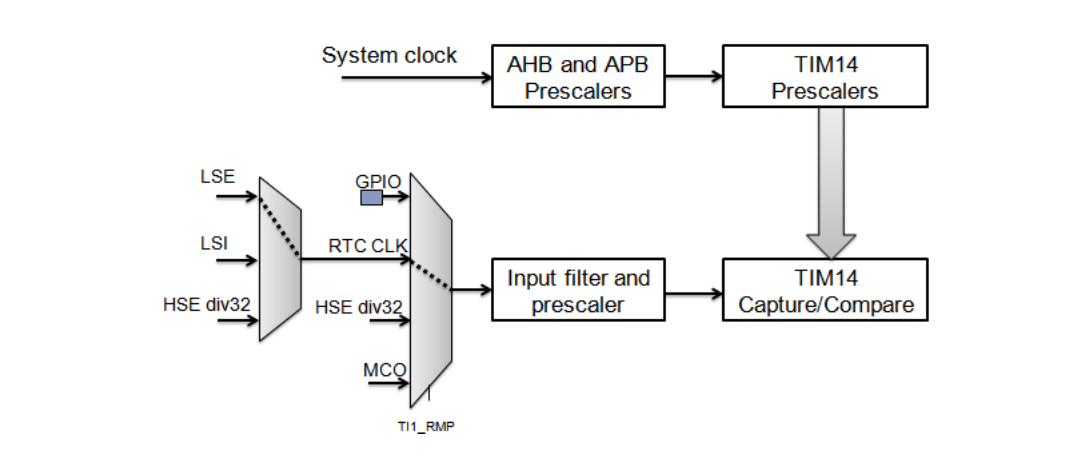
3.2 Hardware implementation

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

The STM32F0xx offers a useful feature, that is the ability to connect internally and indirectly the Low-speed external (LSE) oscillator to Timer 14 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 oscillator should be connected to OSC32_IN and OSC32_OUT.

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

Figure 7. Hardware connection using LSE as the reference frequency



1. TI1_RPM: TIM14 input 1 remapping
2. TI1_RPM: Timer input 1
3. IC1PS: Input capture 1 prescaled

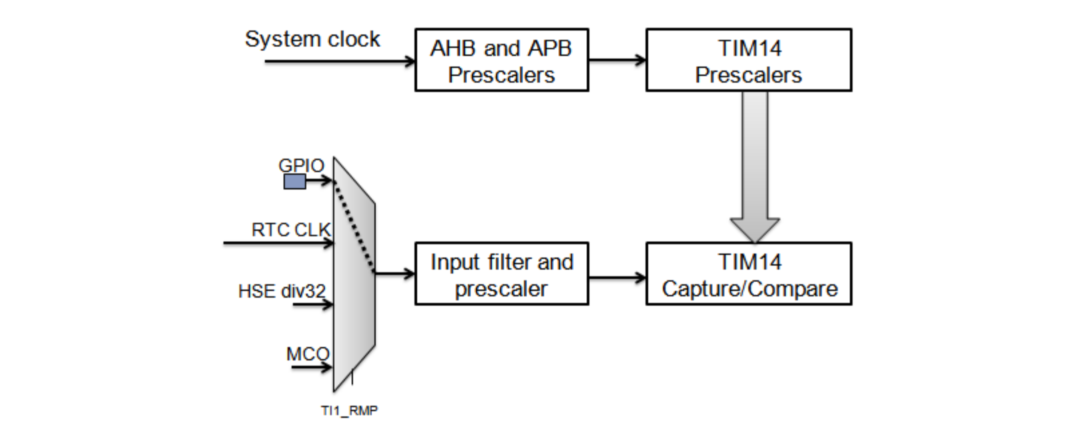
Note: It is also possible to use MCO for this purpose.

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

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 *Figure 8* below, the reference signal should be connected to Timer 14 channel 1.

Figure 8. Hardware connection using external reference frequency



1. TI1_RPM: TIM14 input 1 remapping
2. TI1: Timer input 1
3. IC1PS: Input capture 1 prescaled

Note: When using an external signal as a reference, another timer can be used instead of TIM14. The firmware provided with this application note uses TIM14.

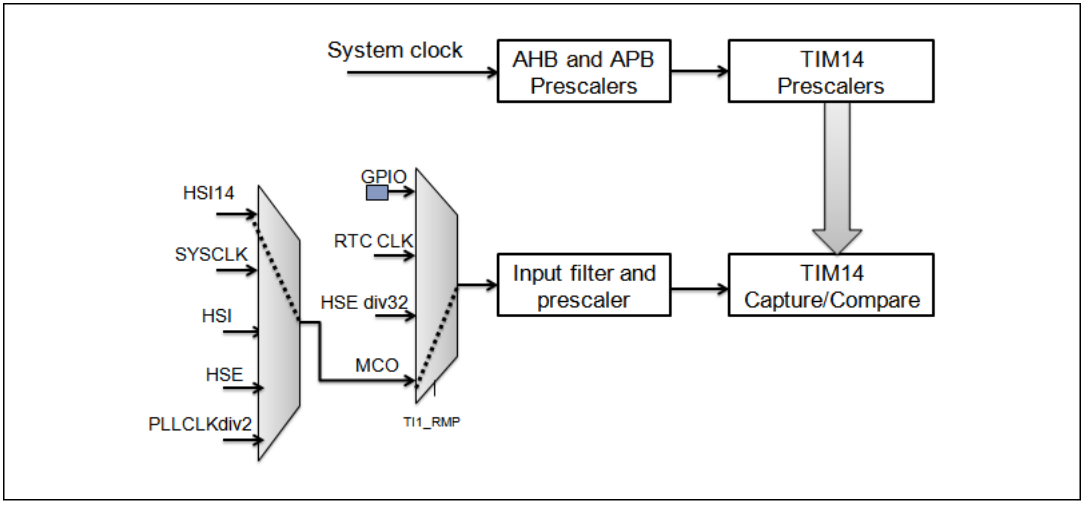
Refer to the application note AN2868 “Internal RC oscillator (HSI) calibration” for more details about using the external signal frequency for calibration.

3.2.3 Measure the HSI14 frequency

The STM32F0xx product family offers a useful feature, that is the ability to connect internally and indirectly the High-speed internal (HSI14) oscillator to Timer 14 channel 1.

Figure 7 shows the hardware connections needed for internal oscillators calibration.

Figure 9. Hardware connection For HSI14



1. TI1_RPM: TIM14 input 1 remapping
2. TI1: Timer input 1
3. IC1PS: Input capture 1 prescaled

3.3 Description of the internal oscillator calibration firmware

The internal RC oscillator calibration firmware provided with this application note includes 4 major functions:

- `uint32_t HSI_CalibrateMinError(void)`
- `ErrorStatus HSI_CalibrateFixedError(uint32_t MaxAllowedError, uint32_t* Freq)`
- `uint32_t HSI14_CalibrateMinError(void)`
- `ErrorStatus HSI14_CalibrateFixedError(uint32_t MaxAllowedError, uint32_t* Freq)`
- `ErrorStatus HSI_CalibrateCurve( uint32_t* Freq );`
- `void HSI_GetCurve( void );`
- `ErrorStatus HSI14_CalibrateCurve(uint32_t* Freq);`
- `void HSI14_GetCurve(void);`
- `uint32_t HSI48_FreqMeasure( void );`
- `uint32_t HSI48_ManualTrimming( void );`

3.3.1 Internal oscillator calibration with minimum error

The HSI_CalibrateMinError() function calibrates the internal oscillator (HSI) 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 thus obtained 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).

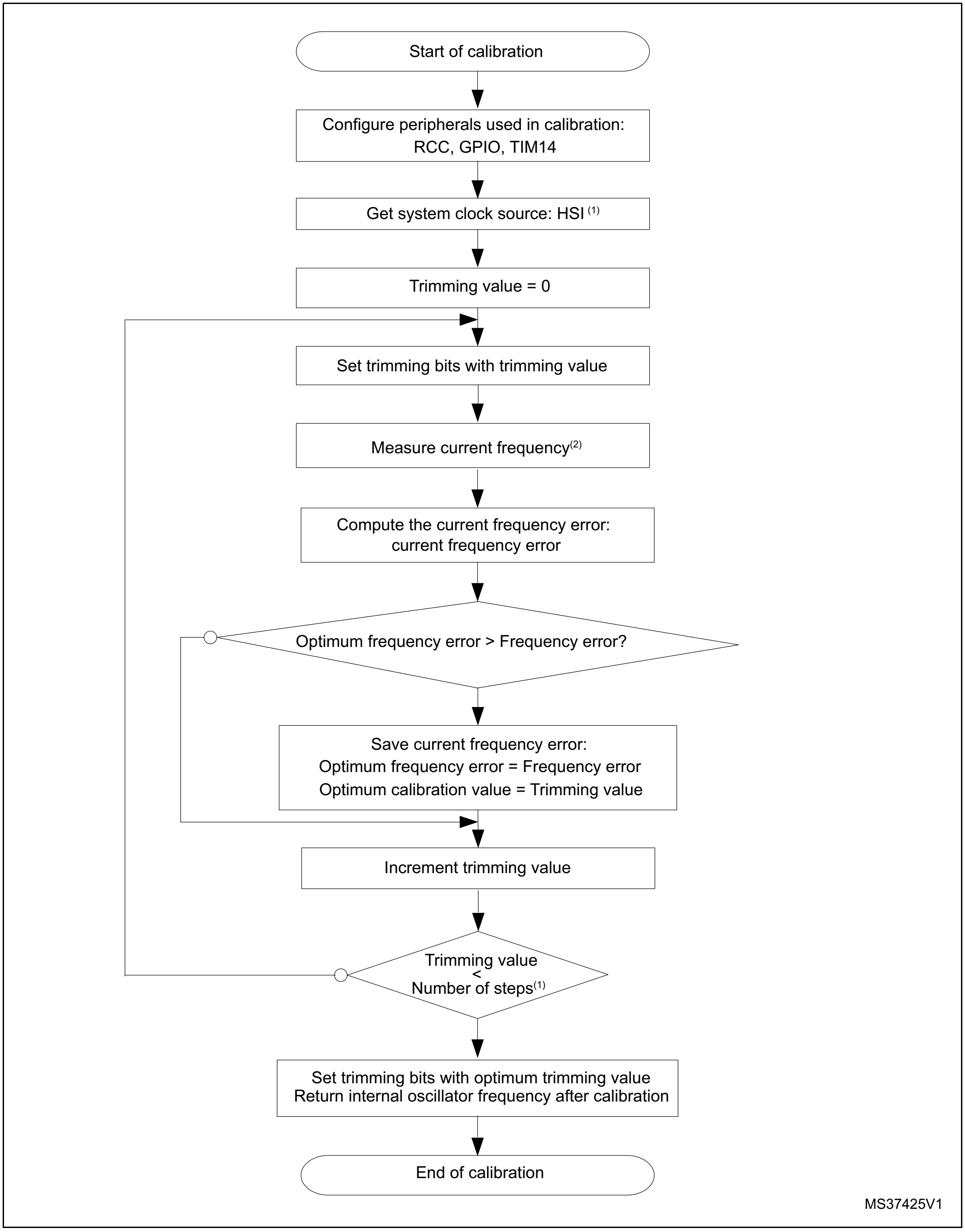
The flowchart in *Figure 10* provides the algorithm for this function.

Example

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

In case of an HSI14 calibration, the **HSI14_CalibrateMinError()** function should be used instead of HSI_CalibrateMinError**().**

Figure 10. Internal oscillator calibration: finding the minimum frequency error


Start of calibration
Configure peripherals used in calibration: RCC, GPIO, TIM14
Get system clock source: HSI (1)
Trimming value = 0
Set trimming bits with trimming value
Measure current frequency(2)
Compute the current frequency error: current frequency error
Optimum frequency error > Frequency error?
Save current frequency error: Optimum frequency error = Frequency error Optimum calibration value = Trimming value
Increment trimming value
Trimming value < Number of steps(1)
Set trimming bits with optimum trimming value Return internal oscillator frequency after calibration
End of calibration
MS37425V1

1. If the system clock source is HSI, the trimming bits have a 5-bit length and the number of steps is 32.
2. The frequency measurement is detailed in *Section 3.3.3*.

3.3.2 HSI calibration with fixed error

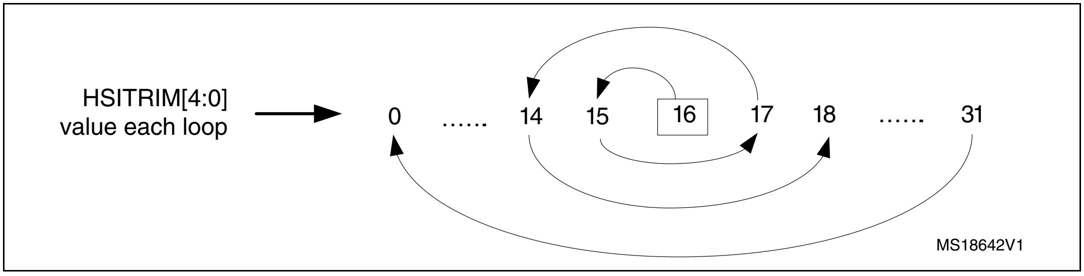
The HSI_CalibrateFixedError() function is provided to calibrate the HSI oscillator with a maximum allowed frequency error. It is configured by the user as an absolute value given in Hertz (the first parameter: `MaxAllowedError`). This function is the same as

HSI_CalibrateMinError() (refer to *ErrorStatus HSI14_CalibrateFixedError(uint32_t MaxAllowedError, uint32_t* Freq)*), 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[4:0] according to this frequency and returns SUCCESS, meaning that the calibration operation has succeeded.
- Otherwise, it continues searching for it until the HSITRIM bits = 31 (32nd frequency). It then sets the trimming bits HSITRIM[4:0] 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 HSTRIM = 16. The HSITRIM value is computed in loops to find the next value. That is, the HSITRIM value starts from 16, 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 31, forming a “spring loop” (as shown in *Figure 11*).

  This algorithm is based on the fact that the probability of finding the frequency that has the minimum error increases when the HSITRIM[4:0] value tends to 16. This algorithm is implemented so as to minimize the time consumed by the calibration process.

Figure 11. “Spring loop”

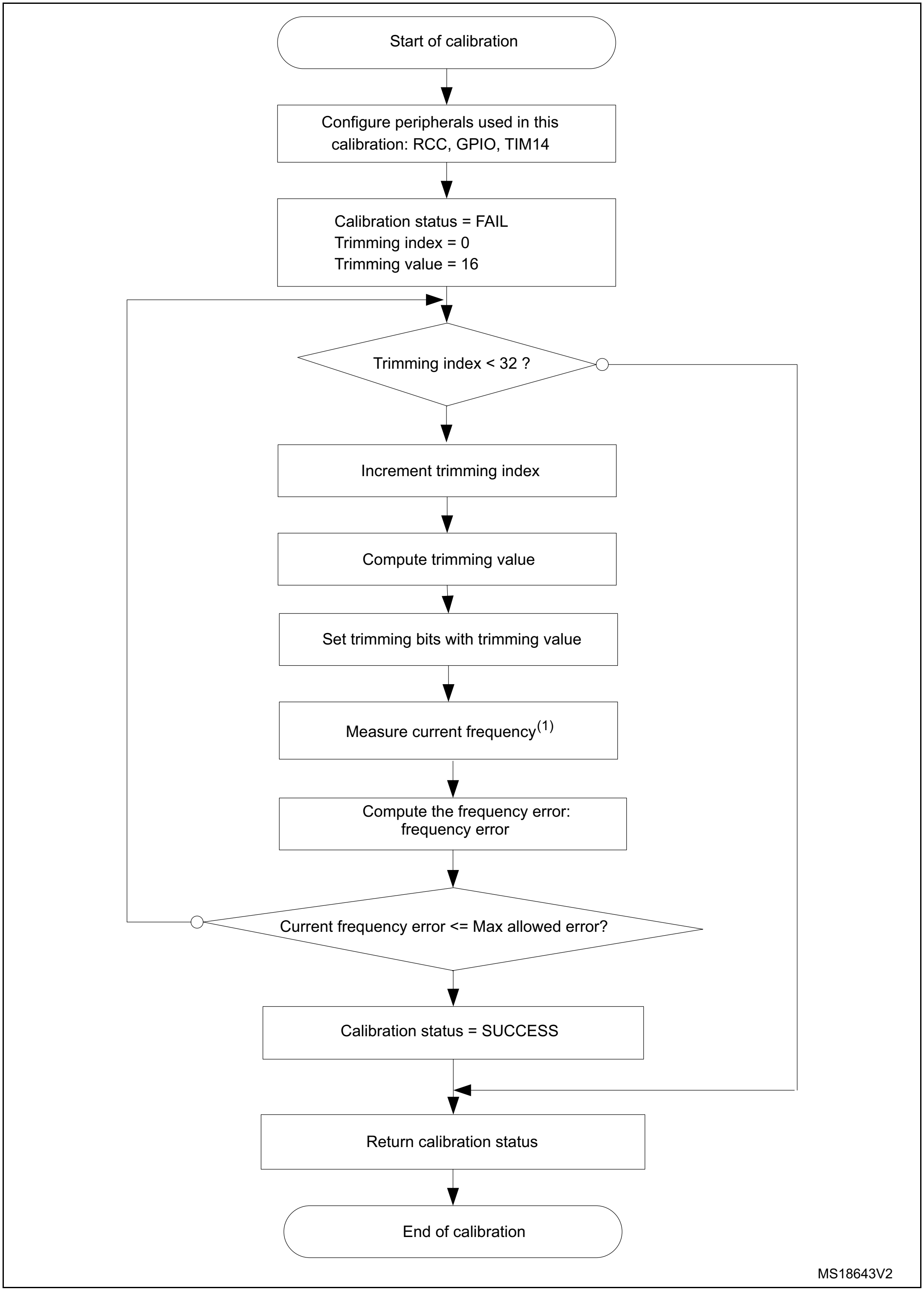


The second parameter is used to get the frequency (in Hertz) after calibration in the form of an unsigned 32-bit integer (unit32_t).

In case of an HSI14 calibration, the **HSI14_CalibrateFixedError()** function should be used instead of **HSI_CalibrateFixedError().**

The flowchart in *Figure 12* provides the algorithm for this function.

Figure 12. HSI calibration flowchart: maximum allowed frequency error


Start of calibration
Configure peripherals used in this calibration: RCC, GPIO, TIM14
Calibration status = FAIL
Trimming index = 0
Trimming value = 16
Trimming index < 32 ?
Increment trimming index
Compute trimming value
Set trimming bits with trimming value
Measure current frequency(1)
Compute the frequency error: frequency error
Current frequency error <= Max allowed error?
Calibration status = SUCCESS
Return calibration status
End of calibration
MS18643V2

1. The frequency measurement is detailed in *Section 3.3.3: Internal oscillator frequency measurement*.

3.3.3 Internal oscillator frequency measurement

The internal oscillator frequency measurement is performed by Timer 14 capture interrupt. In the timer TIM14 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 14 channel 1:

  ```
  #define USE_REFERENCE_LSE
  ```

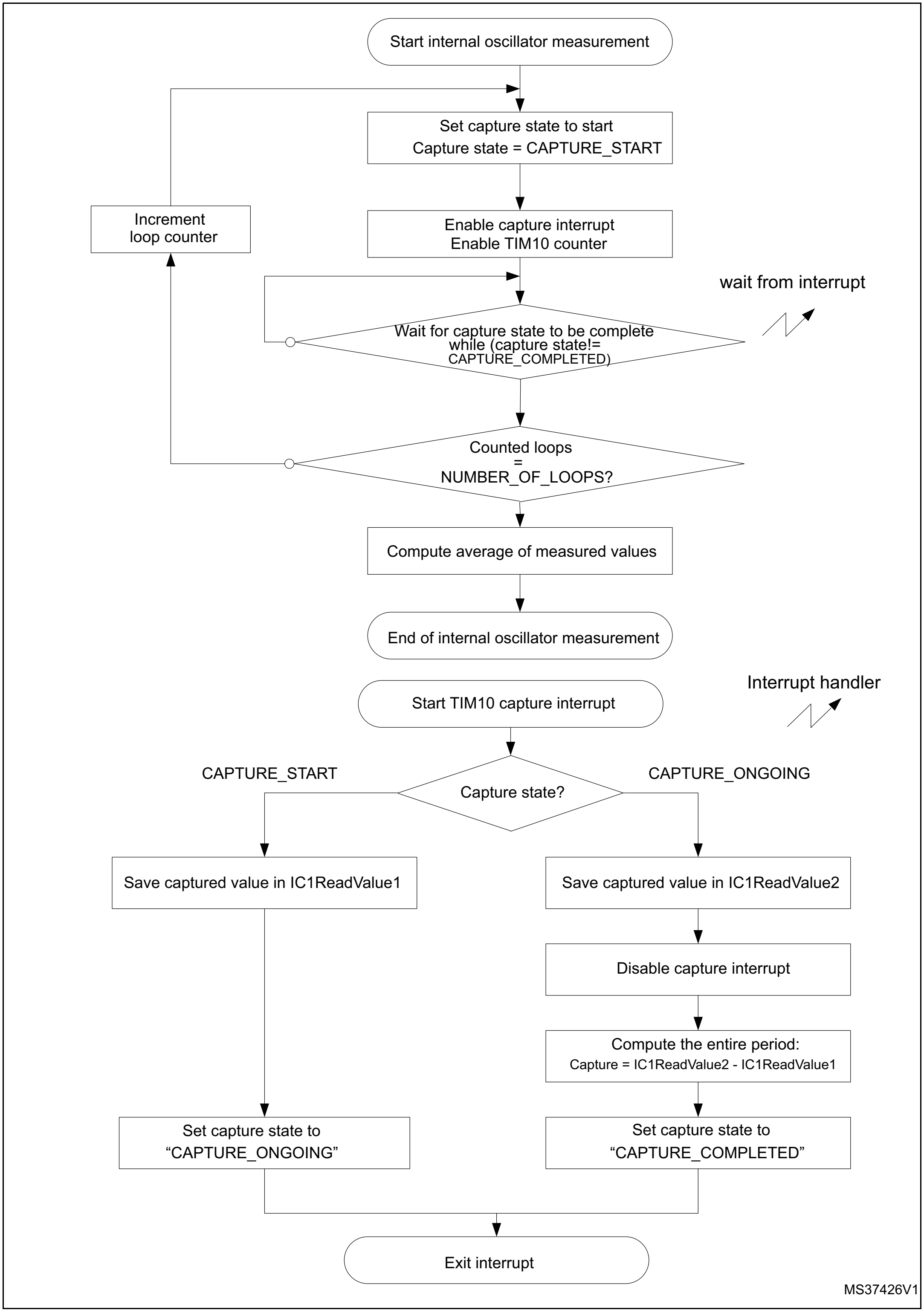
- If the reference frequency is an external signal equal to 50 Hz, then comment the line above and define the reference frequency as shown below:

  ```
  #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 5*).

Note: Figure 3.3.4 provides the frequency measurement algorithm.

Figure 13. Internal oscillator frequency measurement flowchart


Start internal oscillator measurement
Set capture state to start
Capture state = CAPTURE_START
Increment
loop counter
Enable capture interrupt
Enable TIM10 counter
wait from interrupt
Wait for capture state to be complete
while (capture state!=
CAPTURE_COMPLETED)
Counted loops
=
NUMBER_OF_LOOPS?
Compute average of measured values
End of internal oscillator measurement
Start TIM10 capture interrupt
Interrupt handler
CAPTURE_START
Capture state?
CAPTURE_ONGOING
Save captured value in IC1ReadValue1
Save captured value in IC1ReadValue2
Disable capture interrupt
Compute the entire period:
Capture = IC1ReadValue2 - IC1ReadValue1
Set capture state to
“CAPTURE_ONGOING”
Set capture state to
“CAPTURE_COMPLETED”
Exit interrupt
MS37426V1

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 32 measurement for HSI 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 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.

In case of an HSI14 calibration, the HSI14_ GetCurve () and HSI14_CalibrateCurve() functions should be used instead of HSI_GetCurve() and HSI_CalibrateCurve().

3.4 Recommendations on the use of the calibration library

1. If the external signal frequency is lower than system clock / 65535, the TIM14 counter prescaler should be used to support low frequencies.
2. If the external signal frequency is higher than system clock / 100, TIM14 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 14 (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 vs. 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 (e.g. surrounding temperature) the calibration curve can change. From this reason it's recommended to measure from time to time (or when the condition change) again 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 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 = (2 x 51 x 32) / 4096 = 797 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.

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

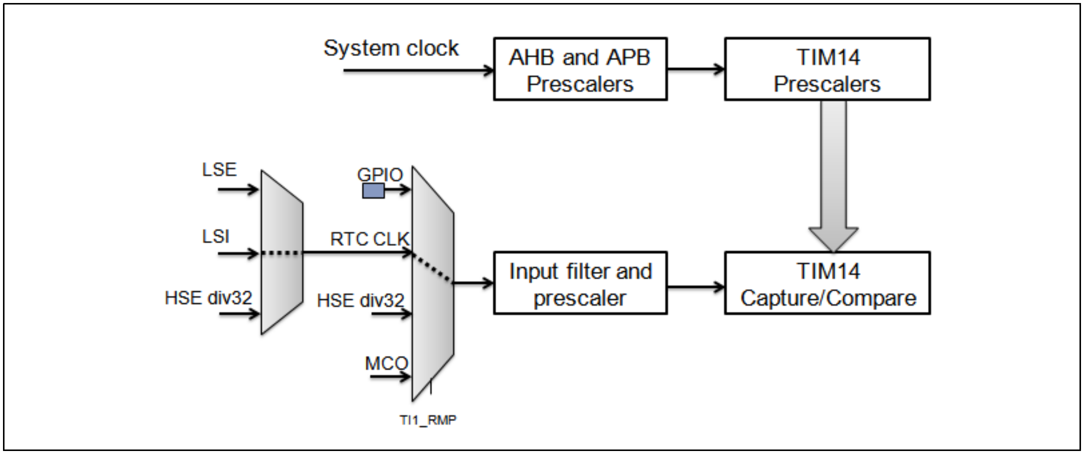
The internal LSI RC oscillator is a low-power clock source. In the STM32F0x microcontroller family, an internal and indirect connection is provided between the internal RC oscillator LSI and the embedded timer TIM14 to facilitate the measurement procedure.

4.1 Measurement principle

The internal RC oscillator measurement procedure consists in running the timer counter using the HSI clock, configuring the internal RC oscillator LSI as the clock source for RTC, configuring the timer in Input capture mode and then connecting the RTC to the timer.

Figure 14 shows the configuration used to perform an LSI measurement.

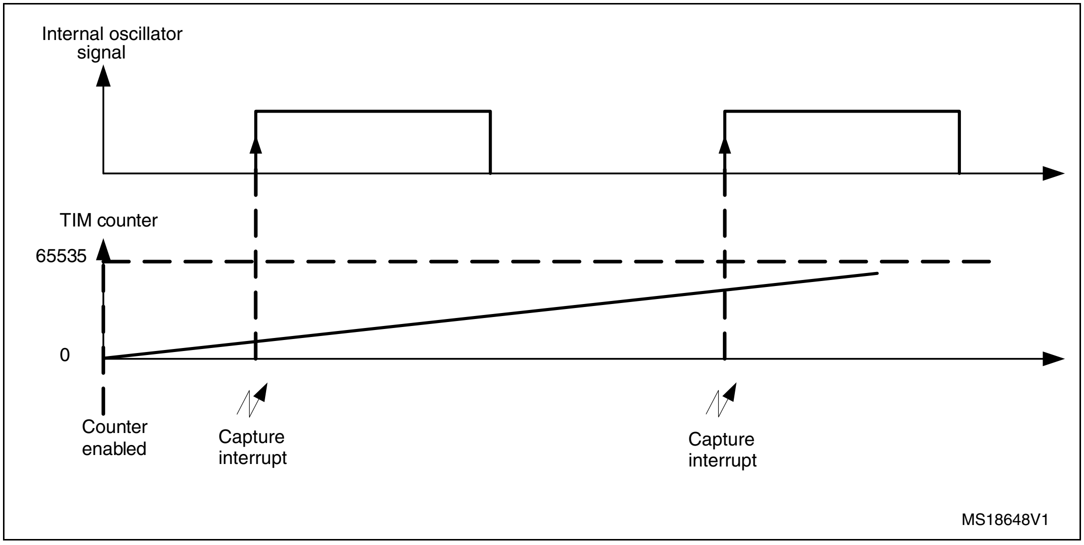
Figure 14. LSI measurement configuration



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. *Figure 15* shows the timing diagram of an internal RC oscillator measurement.

Note: MCO can be also used for this purpose.

Figure 15. 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 HSI frequency value: typical value is 8 MHz,
- Capture represents an entire period of internal RC oscillator LSI: IC1ReadValue2 - IC1ReadValue1.

As users can conclude from the formula above, the frequency measurement accuracy depends on the HSI 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 HSI, 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 files:

- 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_PERIOD_NUMBERS 10
```

5 Internal oscillator calibration/measurement example description

The example provided with this application note shows the ability of the firmware to calibrate the internal RC oscillators (HSI, HSI14 and HSI48) and explains how to use it to measure the internal RC oscillators (HSI14 and LSI) of the STM32F0xx microcontroller.

In this example:

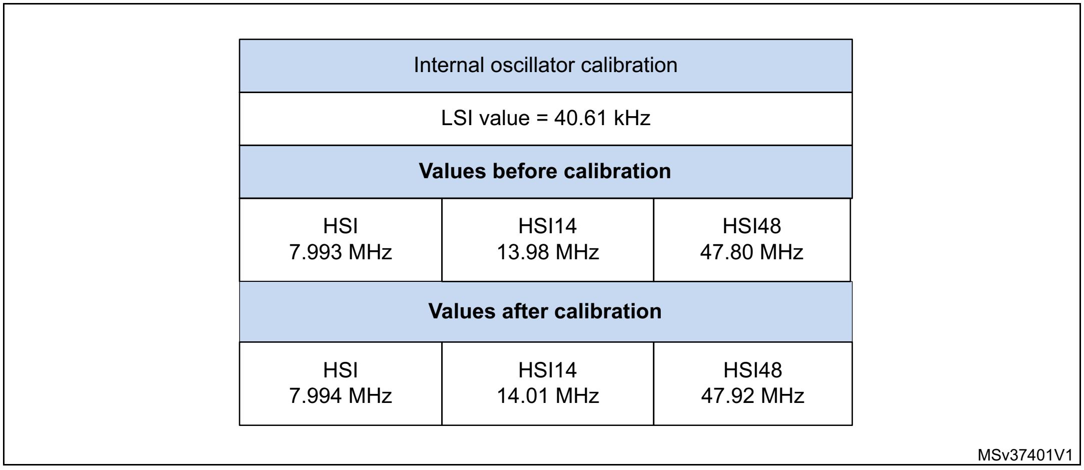
- After system reset, the HSI is selected to be used as the system clock source.
- The HSI is calibrated using the LSE oscillator as a reference clock.
- When the HSI oscillator has been calibrated, the PLL (clocked by HSI) is configured to 48 MHz and used as the system clock source.
- Then, the HSI14 is calibrated according to HSI precision.
- HSI48 is measured and calibrated.
- Then HSI48 is selected as system clock, CRS peripheral initialized and run.
- After that, the LSI frequency is measured.
- Finally, the measured frequency of the different oscillators is displayed on the STM320518-EVAL board's LCD, as shown in *Figure 16*.

By default, the example uses the minimum error method to calibrate the HSI and HSI14 oscillators.

To run the calibration process that provides the frequency with fixed error, users have to comment out the following define in the main.c file:

```
#define CALIBRATION_MIN_ERROR
```

Figure 16. Internal oscillator calibration



Internal oscillator calibration		
LSI value = 40.61 kHz		
Values before calibration		
HSI 7.993 MHz	HSI14 13.98 MHz	HSI48 47.80 MHz
Values after calibration		
HSI 7.994 MHz	HSI14 14.01 MHz	HSI48 47.92 MHz

MSv37401V1

6 Conclusion

Even if internal RC oscillators are 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 (HSI, HSI14 and HSI48) calibration:** how to fine-tune the oscillator to the typical value.
- **Low-speed internal oscillator measurement**: how to get the “exact” LSI frequency value.

Several frequency sources can be used to calibrate the internal RC oscillator (HSI): 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 typical frequency value and the measured one for each trimming value. From this, the calibration value is calculated and then programmed in the trimming bits.

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 STM32F0xx microcontroller 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.

7 Revision history

Table 3. Document revision history

Date	Revision	Changes
02-Jul-2012	1	Initial release.
05-Feb-2015	2	Extended the document applicability to all STM32F0x1, STM32F0x2 and STM32F0x8 product lines. – Added HSI48. – Added *Section 1: Applicable STM32F0xx product families*. – Updated *Section 3.3: Description of the internal oscillator calibration firmware*. – Added *Section 3.1.3: HSI48 calibration*. – Added *Section 3.3.4: Internal oscillator calibration using calibration curve*. – Updated *Section 3.4: Recommendations on the use of the calibration library*. – Updated *Figure 12: HSI calibration flowchart: maximum allowed frequency error*. – Updated *Figure 13: Internal oscillator frequency measurement flowchart*. – Updated *Figure 16: Internal oscillator calibration*.

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. 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.