

How to calibrate an STM32L0xx internal RC oscillator

Introduction

The STM32L0xx microcontrollers have 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. The HSI16 oscillator has a typical frequency of 16 MHz. The MSI oscillator is a lower speed, low-power clock source.

The STM32L0xx microcontrollers have two secondary clock sources.

- LSI: 37 kHz (Low-Speed Internal)
- HSI48: 48MHz (High-Speed Internal) that can be used directly for USB and for random number generator (RNG).

The operating temperature has an impact on the accuracy of the RC oscillators. At 25 °C, the HSI16 oscillator has an accuracy of ±1% typically, the MSI oscillator has an accuracy of ± 0.5% and the HSI48 oscillator has an accuracy of ±4%. But in the temperature range of -40°C to 105 °C, the accuracy decreases. To compensate for the influence of temperature on internal RC oscillators accuracy, the STM32L0xx microcontrollers have built-in features to allow you to calibrate the HSI16, MSI and HSI48 oscillators and measure the MSI and LSI (Low-Speed Internal) oscillator frequencies.

This application note focuses on how to calibrate internal RC oscillators: HSI16, MSI and HSI48. Three methods are presented: the first method is based on finding the frequency with the minimum error, the second one consists in finding the maximum allowed frequency error and the third one implements a table of premeasured values and then only searches in it to for the appropriate change. All three 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.

- Note:*
- 1 *STM32L0xx refers to Ultra Low Power Access line, USB line and USB and LCD line devices in this document.*
 - 2 *Ultra Low Power Access line devices are STM32L051xx microcontrollers where the Flash memory density ranges between 32 and 64 Kbytes.*
 - 3 *Ultra Low Power USB line devices are STM32L052xx and STM32L062xx microcontrollers where the Flash memory density ranges between 32 and 64 Kbytes.*
 - 4 *Ultra Low Power USB and LCD line devices are STM32L053xx and STM32L063xx microcontrollers where the Flash memory density ranges between 32 and 64 Kbytes.*

Table 1. Applicable products and software

Type	Part Number
Microcontrollers	STM32L051C6, STM32L051C8, STM32L051K6, STM32L051K8, STM32L051R6, STM32L051R8,
	STM32L052C6, STM32L052C8, STM32L052K6, STM32L052K8, STM32L052R6, STM32L052R8,
	STM32L053C6, STM32L053C8, STM32L053R6, STM32L053R8
	STM32L062K8
	STM32L063C8, STM32L063R8
Software	X-CUBE-RC-CALIB

Contents

- 1 STM32L0xx system clock 5**

- 2 Internal RC oscillator calibration 7**
 - 2.1 Calibration principle 10
 - 2.2 Hardware implementation 11
 - 2.2.1 Case where LSE is used as the reference frequency 11
 - 2.2.2 Case where another source is used as the reference frequency 12
 - 2.3 Description of the internal oscillator calibration firmware 13
 - 2.3.1 HSI16/MSI calibration with minimum error 13
 - 2.3.2 HSI16 calibration with fixed error 15
 - 2.3.3 MSI calibration with fixed error 17
 - 2.3.4 HSI48 calibration 19
 - 2.3.5 HSI16/MSI frequency measurement 19
 - 2.3.6 HSI16/MSI calibration using calibration curve 21
 - 2.4 Recommendations on the use of the calibration library 21
 - 2.5 Calibration process performance 22
 - 2.5.1 Duration of the calibration process 22

- 3 Internal oscillator measurement 24**
 - 3.1 Measurement principle 24
 - 3.2 Description of the internal oscillator measurement firmware 26
 - 3.3 Internal oscillator calibration/measurement demo description 26

- 4 Conclusion 28**

- 5 Revision history 29**

List of tables

Table 1.	Applicable products and software	1
Table 2.	Document revision history	29

List of figures

Figure 1.	Simplified clock tree	5
Figure 2.	HSI16 oscillator trimming characteristics	7
Figure 3.	MSI trimming behavior	9
Figure 4.	TRIM monotonicity	10
Figure 5.	Timing diagram of internal oscillator calibration	11
Figure 6.	Hardware connection using LSE as the reference frequency.	12
Figure 7.	Hardware connection using external reference frequency	12
Figure 8.	Internal oscillator calibration: finding the minimum frequency	14
Figure 9.	“Spring loop”	15
Figure 10.	HSI16 calibration flowchart: maximum allowed frequency error.	16
Figure 11.	MSI calibration flowchart: maximum allowed frequency error.	18
Figure 12.	HSI16/MSI oscillator frequency measurement flowchart	20
Figure 13.	MSI measurement configuration	24
Figure 14.	LSI measurement configuration	25
Figure 15.	Timing diagram of an internal RC oscillator measurement.	25
Figure 16.	HSI16 calibration	27

1 STM32L0xx system clock

The STM32L0xx microcontroller family has various clock sources that can be used to drive the system clock:

- 16 MHz High-Speed Internal (HSI16) RC oscillator clock
- 1 to 32 MHz High-Speed External (HSE) oscillator clock
- 65.5 kHz to 4.2 MHz Multi-Speed Internal (MSI) RC oscillator clock
- 2 to 32 MHz Phase-Locked Loop (PLL) that is clocked by HSI16 or HSE oscillators

The High-Speed Internal (HSI16) RC oscillator typically has a frequency of 16 MHz and consumes 100 μ A.

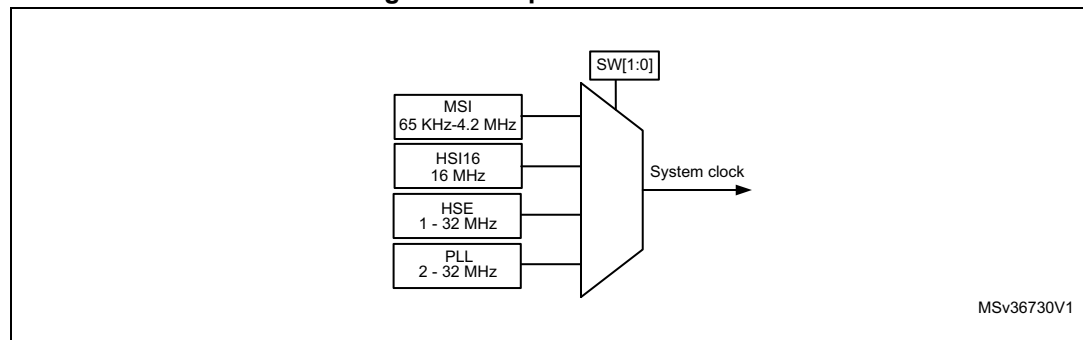
The Multi-Speed Internal (MSI) RC oscillator provides seven frequency ranges: 65.5 kHz, 131 kHz, 262 kHz, 524 kHz, 1.05 MHz, 2.1 MHz (default value) and 4.2 MHz. It is designed to operate with a current proportional to the frequency (refer to the product datasheet for more details about MSI power consumption versus selected range), so as to minimize the internal oscillator consumption when the CPU runs at low frequencies. The MSI clock is used as a system clock after restart from Reset and wake-up from standby. After wake-up from stop mode, the MSI clock can be selected as system clock instead of HSI16 (or HSI16/4).

The HSI48 clock signal is generated from an internal 48 MHz RC oscillator and can be used directly for USB and for random number generator.

The internal RC oscillators (HSI16, MSI and HSI48) have the advantage of providing a low-cost clock source (no external components required). It also has a faster startup time and lower power consumption than the external oscillator. 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 ceramic resonator (tens of ppm).

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

Figure 1. Simplified clock tree



STM32L0xx system clock

The STM32L0xx devices also have three secondary clock sources (that cannot be used as system clock sources):

- 37 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)
- 32.768 kHz Low-Speed External crystal (LSE crystal) which optionally drives the Real Time Clock (RTC)
- 48MHz High-Speed Internal (HSI48) which is designed to provide a high precision clock to the USB peripheral by means of a special Clock Recovery System (CRS) circuitry

2 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, MSI and HSI16 RC oscillators are factory-calibrated by ST to have a 1% accuracy at $T_A = 25\text{ }^\circ\text{C}$. After reset, the factory calibration value is automatically loaded in the internal calibration bits.

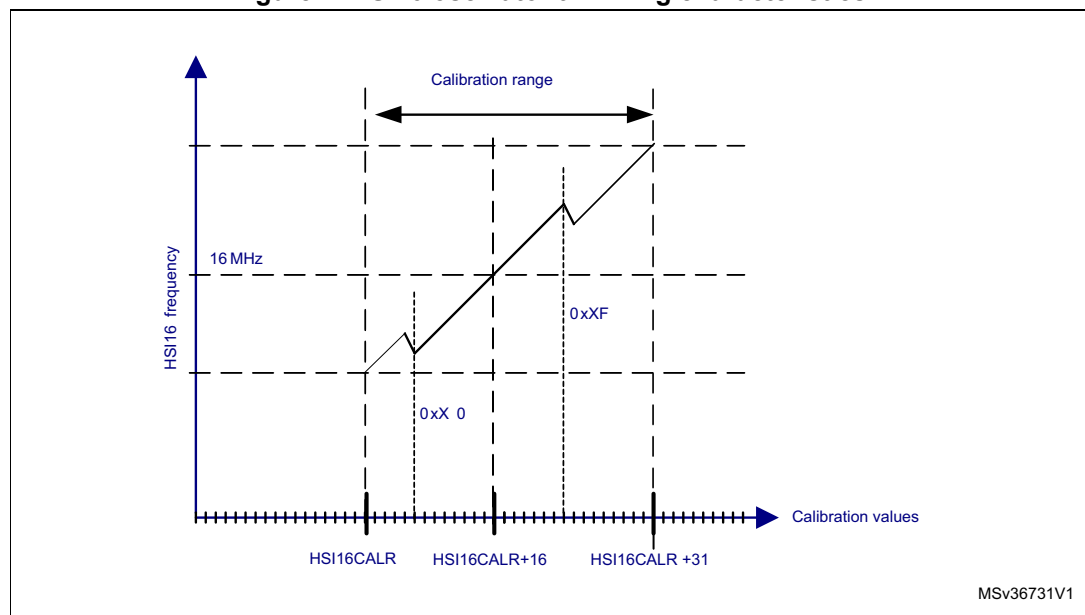
The frequency of the internal RC oscillators can be fine-tuned to achieve better accuracy with wider temperature and supply voltage ranges. The trimming bits are used for this purpose.

For the HSI16 oscillator, the calibration value is loaded in HSI16CAL[7:0] bits after reset. Five trimming bits HSI16TRIM[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 HSI16 frequency. The HSI16 oscillator is fine-tuned in steps of 0.5% (around 80 kHz).

- 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 graph below shows HSI16 oscillator behavior versus calibration value. The HSI16 oscillator frequency increases with calibration value (calibration value = default HSI16CAL[7:0] + HSI16TRIM[4:0]), except at modulo 16. At these calibration values, the negative steps can reach three times the positive steps.

Figure 2. HSI16 oscillator trimming characteristics



MSv36731V1

Internal RC oscillator calibration

For the MSI oscillator, the calibration value is loaded in the MSICAL[7:0] bits after reset. Eight trimming bits MSITRIM[7:0] are used giving a wide tuning range. The calibration is based on adding the default MSICAL[7:0] bits (reset value) to the MSITRIM[7:0] bits.

The result is stored in MSICAL[7:0]:

$$\text{MSICAL}[7:0] = \text{default MSICAL}[7:0] + \text{MSITRIM}[7:0]$$

Example:

Assuming the default MSI calibration value MSICAL[7:0] is 0x80.

1. Writing a value between 0x01 and 0x7F in MSITRIM[7:0] leads to calibration value MSICAL[7:0] in the range of:

$$\text{MSICAL}[7:0] = 0x80 + 0x01 = 0x81$$

$$\text{and MSICAL}[7:0] = 0x80 + 0x7F = 0xFF$$

These results are greater than 0x80 (default MSICAL[7:0] value) and consequently the MSI frequency is increased by 1 step (0x81 - 0x80) to 127 steps (0xFF - 0x80).

2. Writing a value between 0x81 and 0xFF in MSITRIM[7:0] leads to calibration value MSICAL[7:0] in the range of:

$$\text{MSICAL}[7:0] = 0x80 + 0x81 = 0x01$$

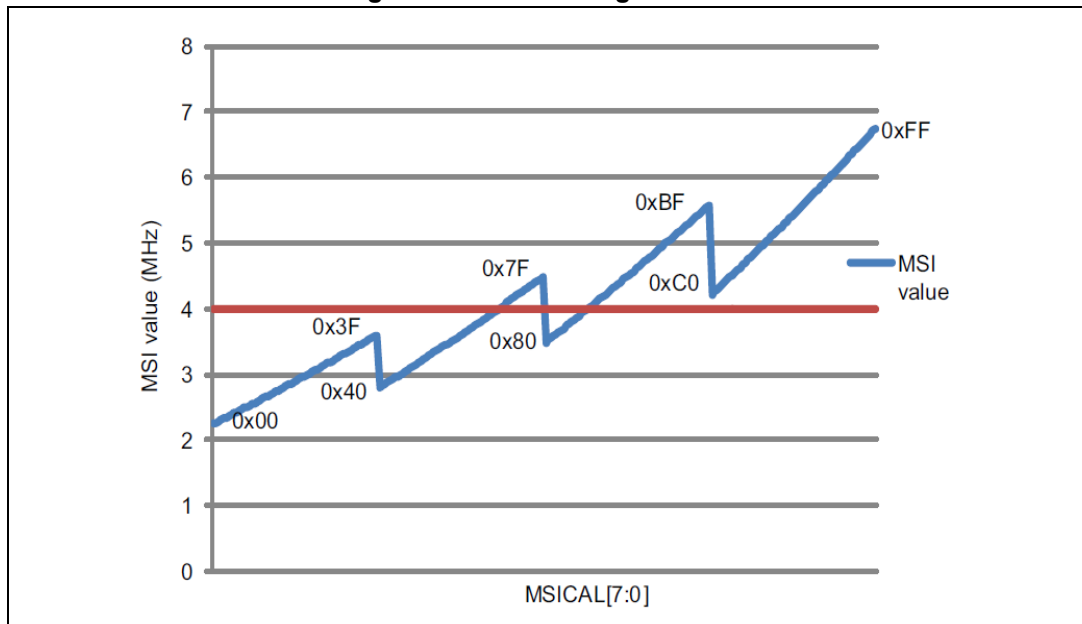
$$\text{and MSICAL}[7:0] = 0x80 + 0xFF = 0x7F$$

These results are lower than 0x80 (default MSICAL[7:0] value) and consequently the MSI frequency is decreased by 1 step (0x01) to 127 steps (0x7F).

3. Writing the default calibration value (0x80) in MSITRIM[7:0] leads to calibration value MSICAL[7:0] equal to $\text{MSICAL}[7:0] = 0x80 + 0x80 = 0x00$ and consequently the MSI frequency is decreased by 128 steps (minimum frequency).

Figure 3 below shows MSI behavior at range 6 (4194304 Hz) versus MSICAL[7:0].

Figure 3. MSI trimming behavior



Note the negative steps when the two MSB bits change from “00” to “01”, from “01” to “10” and from “10” to “11”. The change of the six LSBs leads to fine-tuning of around 0.5% per step.

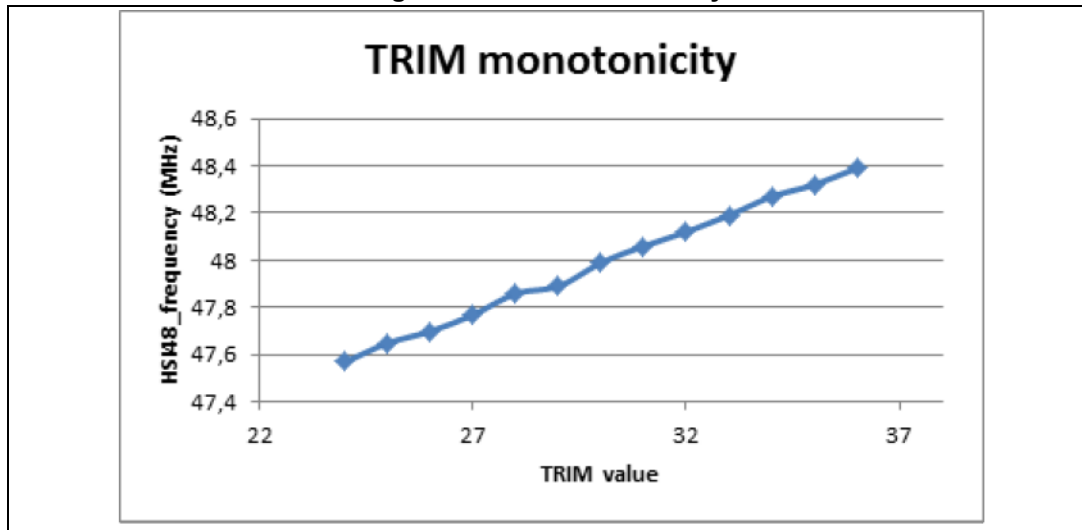
For the HSI48 oscillator, the calibration value is loaded in HSI48CAL[7:0] bits after reset. Six trimming bits TRIM [5:0] (in CRS control register) are used for fine-tuning. The default trimming value is 32. An increase/decrease in this trimming value causes an increase/decrease in HSI48 frequency.

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. TRIM monotonicity



2.1 Calibration principle

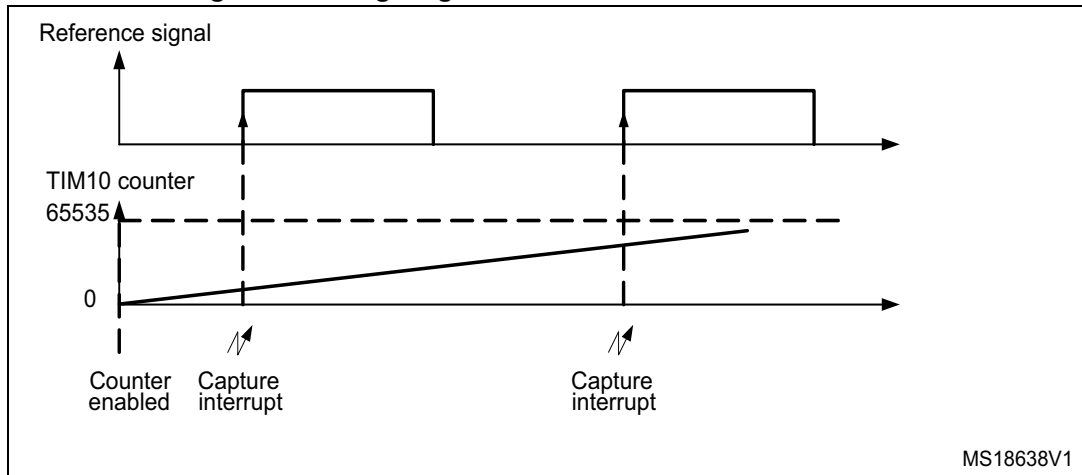
The calibration principle consists in:

1. Setting the internal RC oscillator (that needs to be calibrated) as system clock,
2. Measuring the internal RC oscillator (HSI16 or MSI) 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).

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 where another source is used as the reference frequency](#)).

The following figure 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 HSI16 or MSI), the real frequency generated by the internal RC oscillator versus the reference signal is given by:

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

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:

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

After calculating the error for each trimming value, the algorithm determines the optimum trimming value (that corresponds to the nearest frequency to typical value) to be programmed in the trimming bits (refer to [Section 2.3: Description of the internal oscillator calibration firmware](#) for more details).

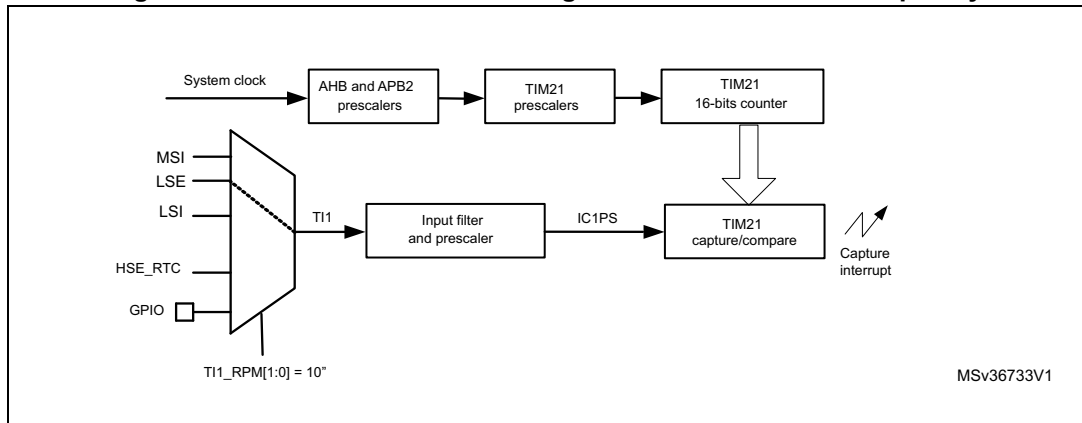
2.2 Hardware implementation

2.2.1 Case where LSE is used as the reference frequency

The STM32L0xx offers a useful feature, that is the ability to connect internally the Low-Speed External (LSE) oscillator to Timer 21 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 6](#) shows the hardware connections needed for internal oscillators calibration using LSE as an accurate frequency source for calibration.

Figure 6. Hardware connection using LSE as the reference frequency



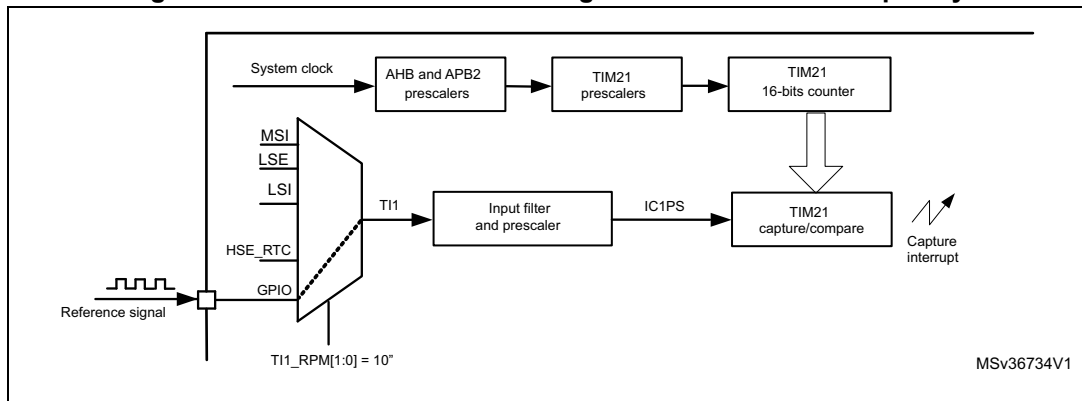
In the above figure:
 TI1_RPM: TIM21 input 1 remapping
 TI1_RPM: Timer input 1
 IC1PS: Input capture 1 prescaled

2.2.2 Case where another source is used as the reference frequency

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

As shown by *Figure 7* below, the reference signal should be connected to Timer 21 channel 1.

Figure 7. Hardware connection using external reference frequency



In the above figure:
 TI1_RPM: TIM21 input 1 remapping
 TI1: Timer input 1
 IC1PS: Input capture 1 prescaled
 *PA2 and PB13 can be used as TIM21 channel 1

2.3 Description of the internal oscillator calibration firmware

The internal RC oscillator calibration firmware provided with this application note includes three 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);`
- `uint32_t HSI48_FreqMeasure(void);`
- `uint32_t HSI48_ManualTrimming(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. 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.

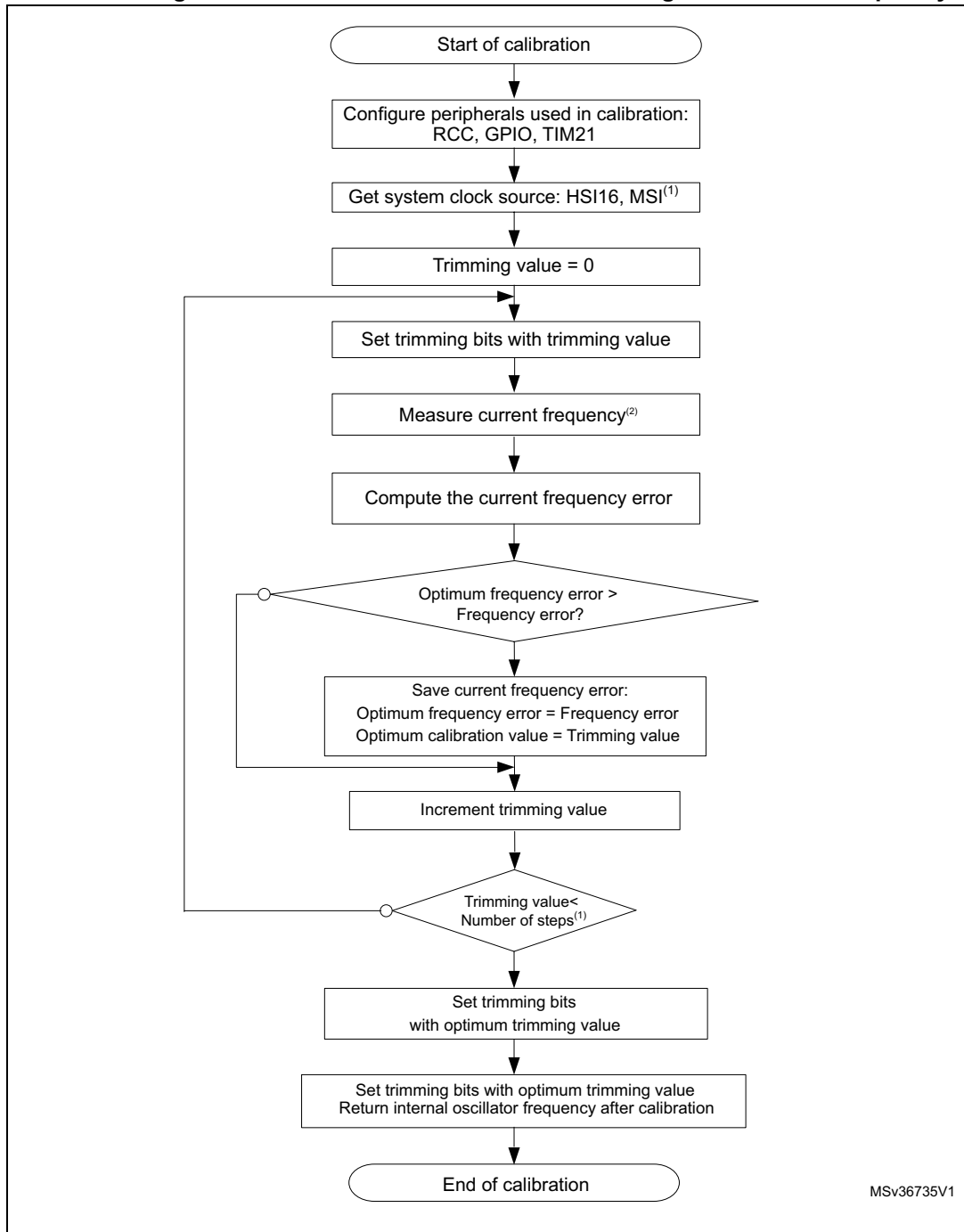
After calibration, the `HSI16_CalibrateMinError()` and `MSI_CalibrateMinError` functions return the internal oscillator frequency value as an unsigned 32-bit integer (`uint32_t`).

The flowchart in [Figure 8](#) provides the algorithm for this 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



MSV36735V1

1. If the system clock source is HSI16, the trimming bits have a 5-bit length and the number of steps is 32. If the system clock source is MSI, the trimming bits have an 8-bit length and the number of steps is 256.
2. Frequency measurement is detailed in [Section 2.3.5](#).

2.3.2 HSI16 calibration with fixed error

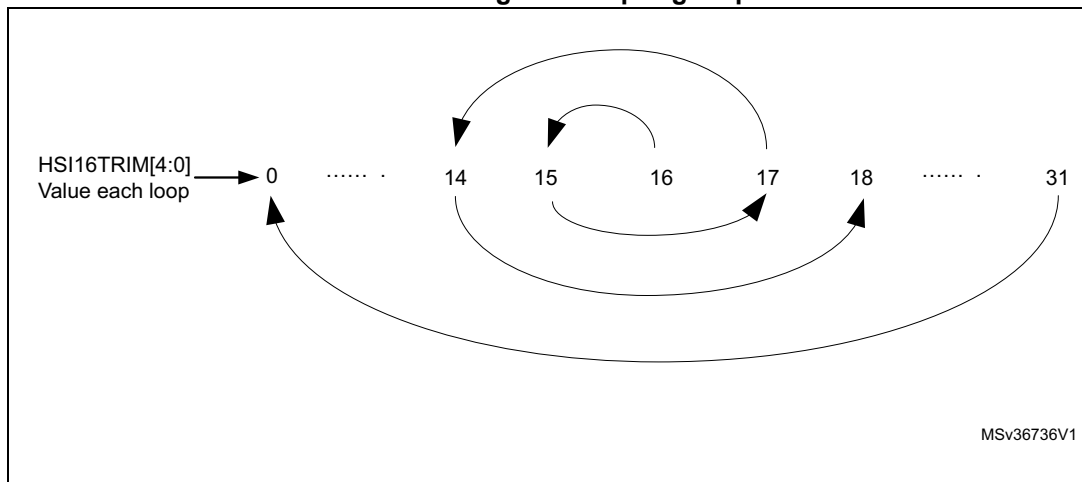
The `HSI16_CalibrateFixedError()` function is provided to calibrate the HSI16 oscillator with a maximum allowed frequency error. It is configured by the user as an absolute value given in Hertz (the first parameter: *MaxAllowedError*). This function is the same as `HSI16_CalibrateMinError()` (refer to [Section 2.3.1: HSI16/MSI calibration with minimum error](#)), 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 `HSI16TRIM[4:0]` according to this frequency and returns `SUCCESS`, meaning that the calibration operation has succeeded.
- Otherwise, it continues searching for it until the `HSI16TRIM` bits = 31 (32nd frequency). It then sets the trimming bits `HSI16TRIM[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 `HSI16TRIM = 16`. The `HSI16TRIM` value is computed in loops to find the next value. That is, the `HSI16TRIM` 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 9](#)).

This algorithm is based on the fact that the probability of finding the frequency that has the minimum error increases when the `HSI16TRIM[4:0]` value tends to 16. This algorithm is implemented so as to minimize the time consumed by the calibration process.

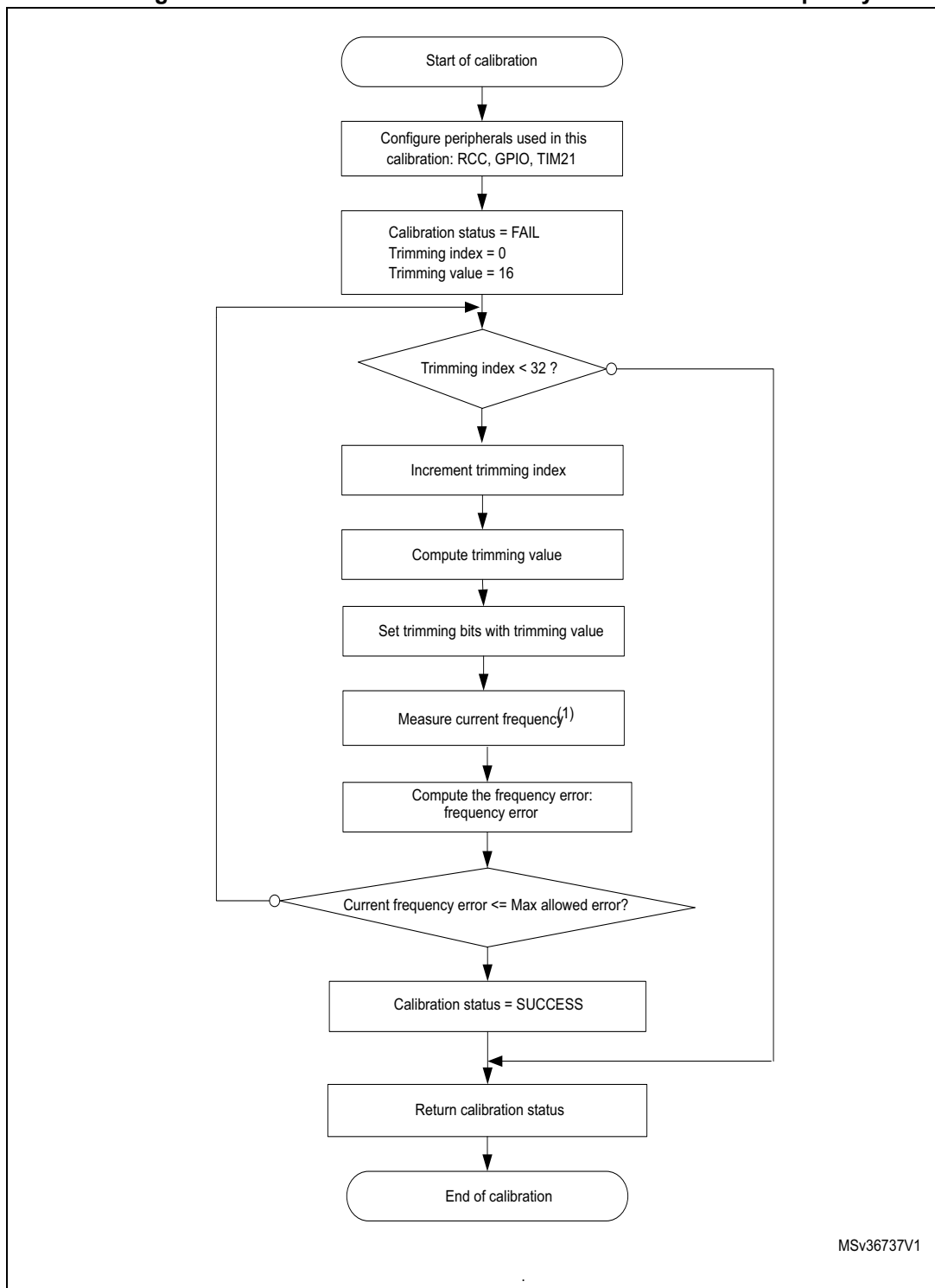
Figure 9. “Spring loop”



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

The flowchart in [Figure 10](#) provides the algorithm for this function.

Figure 10. HSI16 calibration flowchart: maximum allowed frequency error



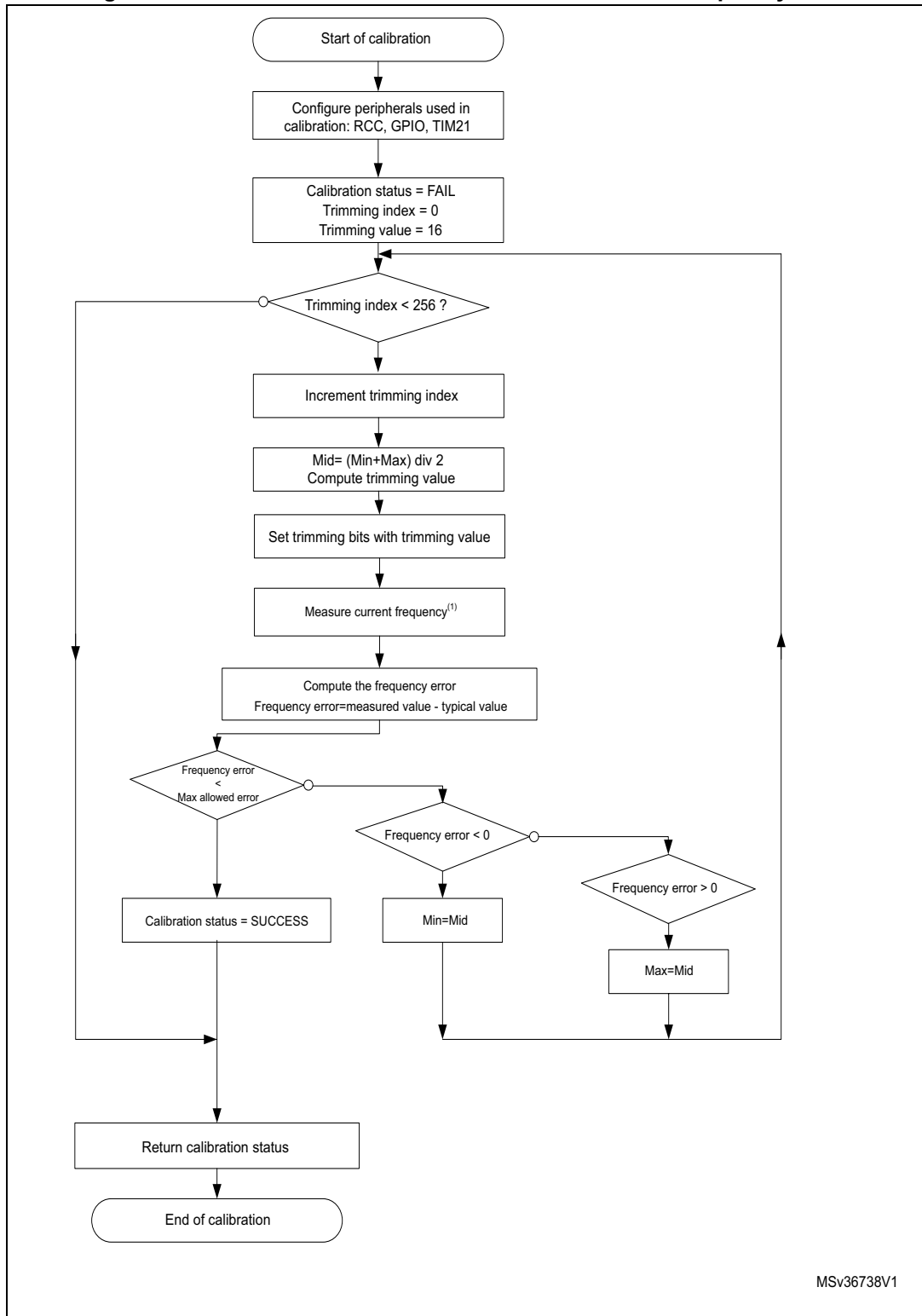
1. Frequency measurement is detailed in [Section 2.3.5: HSI16/MSI frequency measurement](#).

2.3.3 MSI calibration with fixed error

The `MSI_CalibrateFixedError()` function is provided to calibrate the MSI oscillator with a maximum allowed frequency error in minimum time (by avoiding scan of all 256 trimming values). The allowed error is configured by the user as an absolute value given in Hertz (the first parameter: *MaxAllowedError*). This function is the same as `HSI16_CalibrateFixedError()` (refer to [Section 2.3.1: HSI16/MSI calibration with minimum error](#)), with the exception that it calibrates the MSI oscillator using a binary search algorithm.

The flowchart in [Figure 11](#) provides the algorithm for this function.

Figure 11. MSI calibration flowchart: maximum allowed frequency error



1. Frequency measurement is detailed in [Section 3: Internal oscillator measurement](#).

2.3.4 HSI48 calibration

HSI48 can be calibrated the same way as HSI16 or MSI. But STM32L0 family implements CRS (Clock recovery system) that is capable of doing automatic adjustment of oscillator trimming based on comparison with a selectable synchronization signal.

Internally it implements a 16-bit down/up counter which step by step increments or decrements TRIM value until it reaches expected frequency value.

The HSI48 calibration using CRS can be run fully automatic. To speed up the process, the CRS can be used for measurement of the actual error, and set trim value with precalculated value. This process can be 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.

2.3.5 HSI16/MSI frequency measurement

Internal oscillator frequency measurement is performed by Timer 21 capture interrupt. In the timer TIM21 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* files as follows:

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

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.

You can easily configure the frequency of the reference source. It is defined in the header file *hsi16.c* or *msi.c* 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 21 channel 1:

```
#define USE_REFERENCE_LSE
```

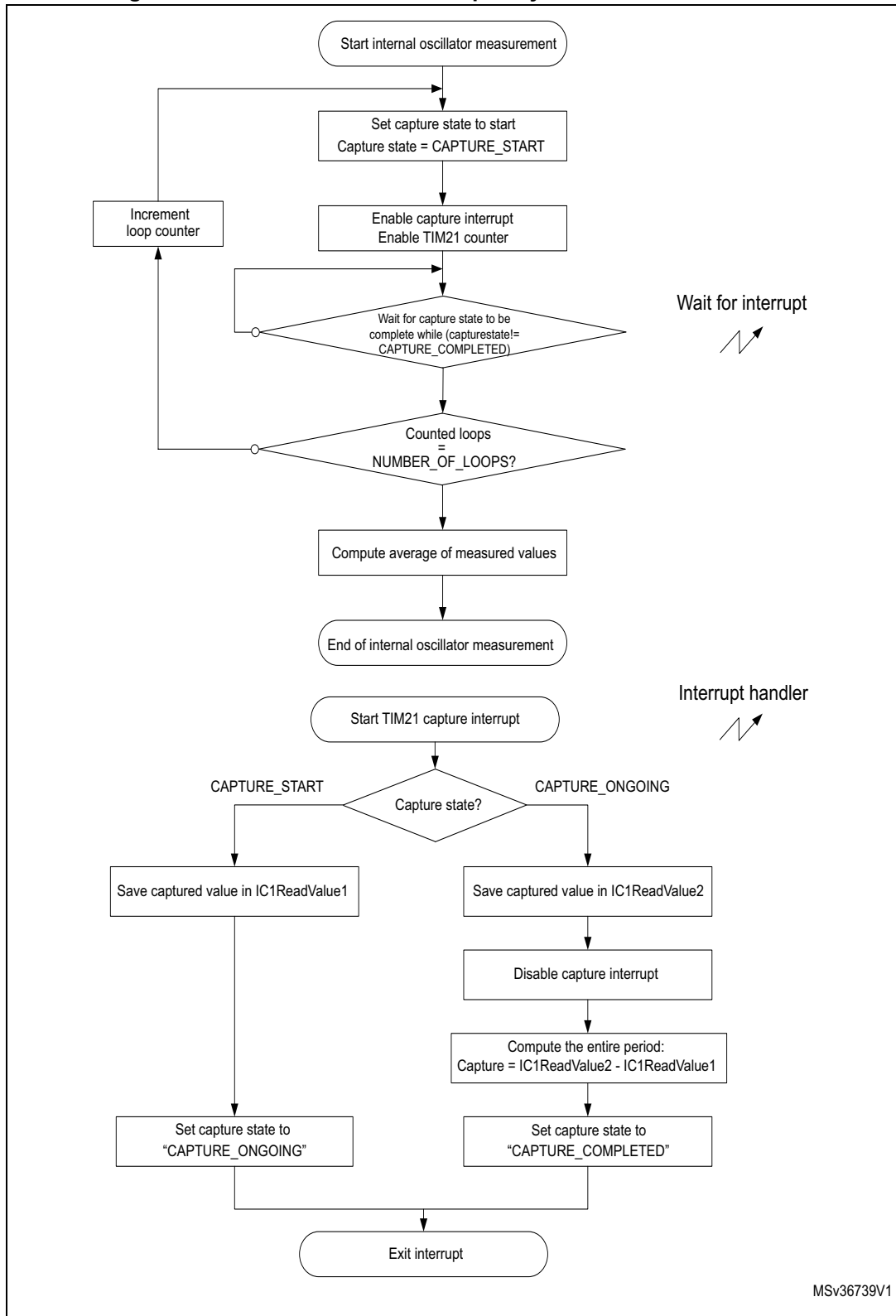
- If the reference frequency is a mains source frequency equal to 50 Hz, then comment the line above and define the reference frequency as shown below:

```
#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 12](#) provides the frequency measurement algorithm.

Figure 12. HSI16/MSI oscillator frequency measurement flowchart



2.3.6 HSI16/MSI calibration using calibration curve

Both previous methods (minimal and fixed error) can take quite long because many measurements (32 for HIS when minimal error method used) need to be run. First, for all trimming values, the difference between the corresponding and the requested frequency is measured and stored in a table. It uses in fact the same principle as minimal error method. When it's later necessary to calibrate internal oscillator function *HSI16_CalibrateCurve()* measures only once actual frequency and searches in the table appropriate value to compensate the difference.

In case of an MSI calibration, the *MSI_GetCurve()* and *MSI_CalibrateCurve()* functions should be used instead of *HSI16_GetCurve()* and *HSI16_calibrateCurve()*.

2.4 Recommendations on the use of the calibration library

1. If external signal frequency is lower than system clock / 65535, the TIM21 counter prescaler should be used to support low frequencies.
2. If external signal frequency is higher than system clock / 100, TIM21 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.
The application therefore has to stop communications, ADC measurements and other processes (except when using the ADC for the calibration, refer to Step 5. 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 21 (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):

$$\text{Threshold_High} = \text{CurrentTemperatureValue} + \text{TemperatureOffset}$$

$$\text{Threshold_Low} = \text{CurrentTemperatureValue} - \text{TemperatureOffset}$$
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.

2.5 Calibration process performance

2.5.1 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:

$$\text{duration} = (2 \times (\text{NUMBER_OF_LOOPS} + 1) \times \text{number of steps}) / \text{REFERENCE_FREQUENCY}$$

If the calibration process is run with a minimum frequency error for 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:

$$\text{duration} = (2 \times 11 \times 32) / 4096 = 172 \text{ ms}$$

When running the calibration process for the MSI oscillator, the number of steps is equal to 256. If the LSE oscillator is used as the reference frequency ($\text{REFERENCE_FREQUENCY} = \text{LSE value} / \text{Input capture prescaler} = 32768/8 = 4096 \text{ Hz}$) and the selected number of measured periods is 10, the calibration consumes approximately:

$$\text{duration} = (2 \times 11 \times 256) / 4096 = 1.4 \text{ s}$$

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.

3 Internal oscillator measurement

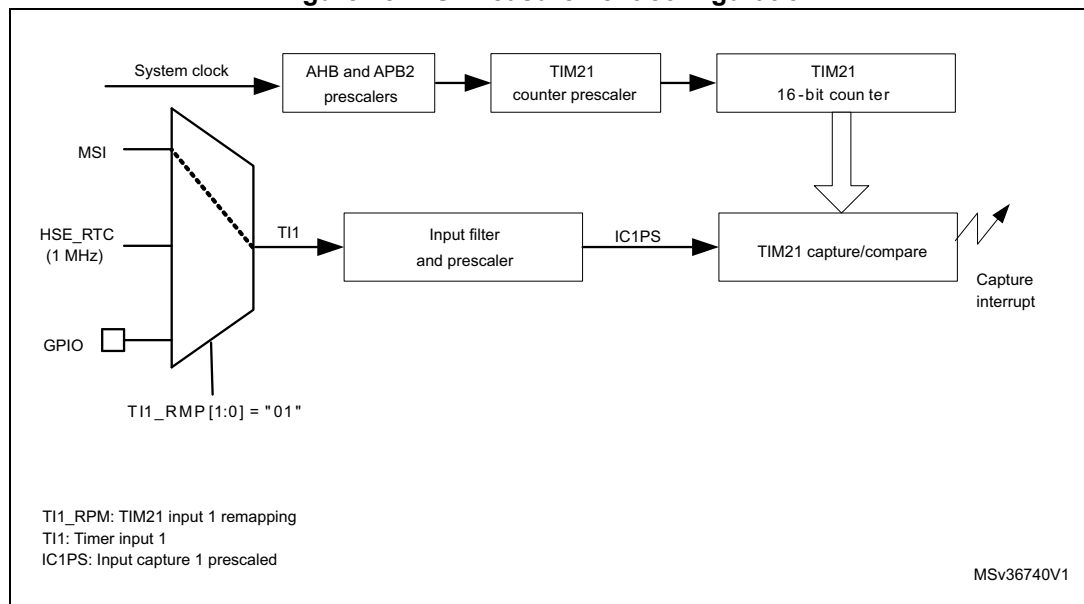
The internal MSI and LSI RC oscillators are low-power and low-cost clock sources. In the STM32L0xx microcontroller family, an internal connection is provided between the internal RC oscillators (MSI and LSI) and the embedded timer (TIM21) to facilitate the measurement procedure.

3.1 Measurement principle

The internal 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 RC oscillator (that needs to be measured) to the timer.

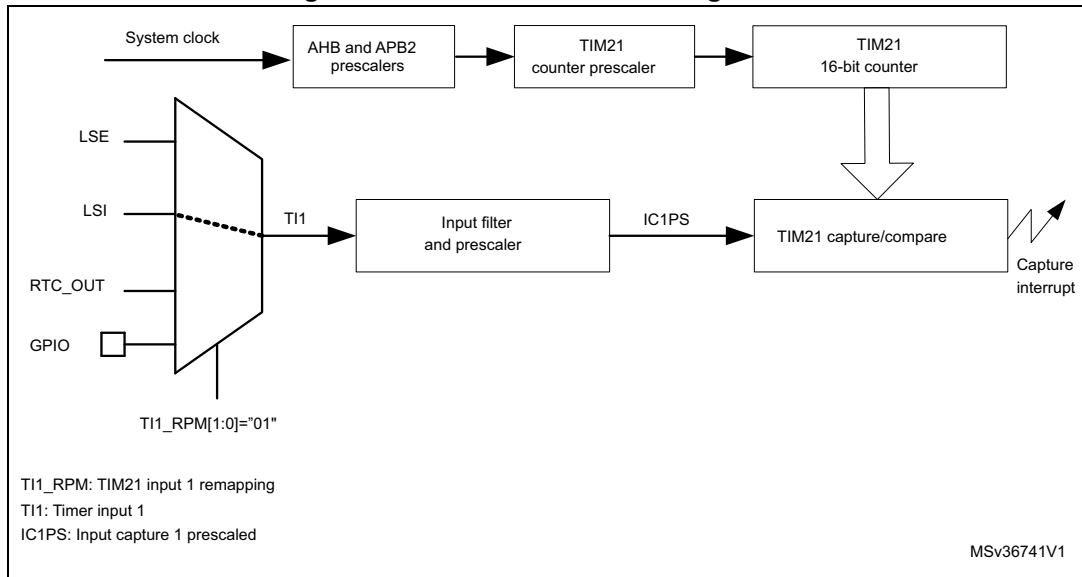
The following figure shows the configuration used to perform the MSI measurement. As shown, MSI can be connected internally to TIM21 Input 1 (TI1).

Figure 13. MSI measurement configuration



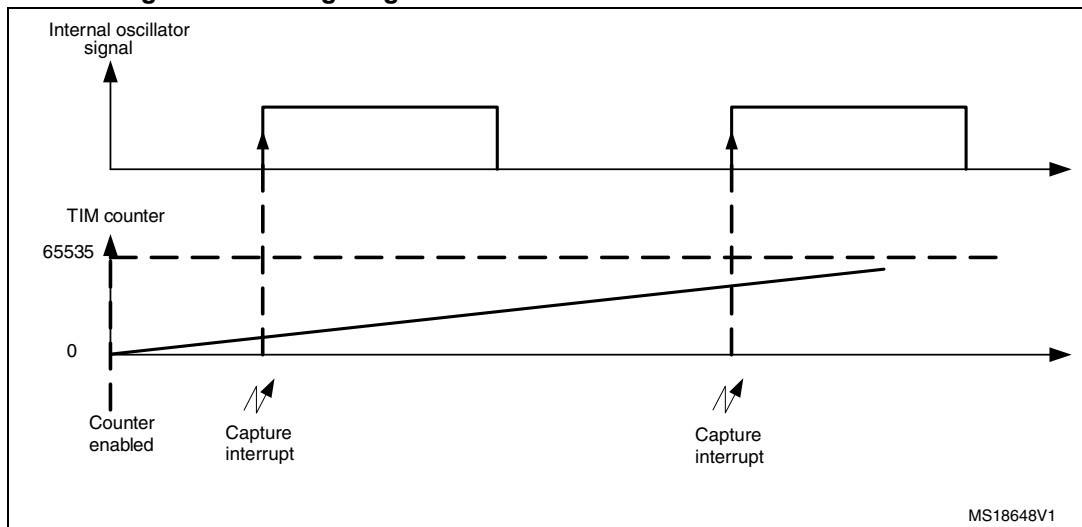
The next figure shows the configuration used to perform LSI measurement. As shown, LSI can be connected internally to TIM21 Input 1 (TI1).

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. The following figure shows the timing diagram of an internal RC oscillator measurement.

Figure 15. Timing diagram of an internal RC oscillator measurement



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

$$\text{internal oscillator frequency} = \text{HSI16_Value} / \text{Capture}$$

where:

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

As you can conclude from the formula above, the frequency measurement accuracy depends on the HSI16 frequency accuracy. Consequently, if a reference signal is available, you can run the internal RC oscillator calibration routine described in [Section 2: 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:

$$\text{LSI_Frequency} = \text{InputCapturePrescaler} * \text{HSI16_Value} / \text{Capture_Value}.$$

The same algorithm as shown in [Figure 12](#) is used to measure the LSI and MSI oscillator frequency. Note that TIM21 ISR is used for MSI and LSI measurement.

3.2 Description of the internal oscillator measurement firmware

The internal oscillator measurement firmware provided with this application note includes two C source files:

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

The internal RC oscillators (MSI or LSI) are measured for a predefined number of periods. Then it returns the average value to minimize the error of the measured frequency.

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

```
#define LSI_PERIOD_NUMBERS 10
```

In the same way, the number of periods can be changed in the msi_measurement.h:

```
#define MSI_PERIOD_NUMBERS 10
```

3.3 Internal oscillator calibration/measurement demo description

The demo provided with this application note shows the ability of the firmware to calibrate the internal RC oscillators (HSI16, MSI and HSI48) and gives an example of how to use it to measure the internal RC oscillator of the STM32L0xx microcontroller.

Before running the calibration routine, select the system clock source in system_STM32L0xx.c. If the HSI16 (rep. MSI) oscillator is selected as the system clock source, HSI16 (or MSI) is calibrated. In this demo, the internal RC oscillator (MSI or HSI16) is calibrated using the LSE oscillator as a reference.

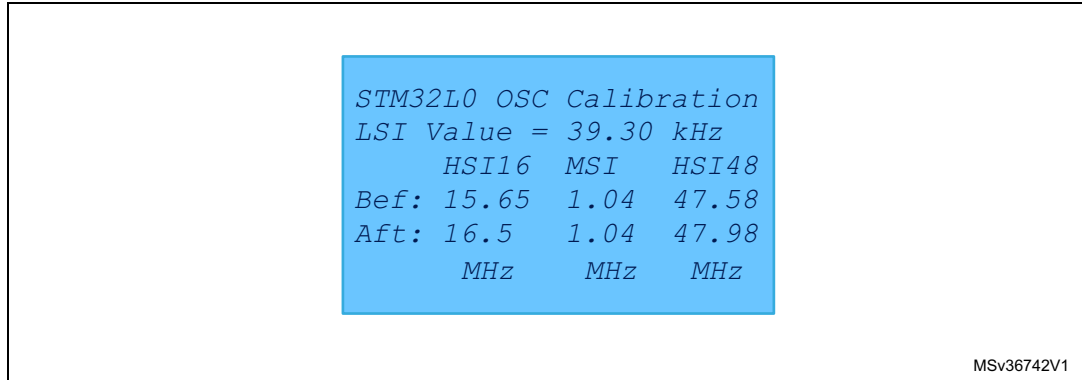
By default, the demo uses the minimum error method to calibrate HSI16 oscillators.

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

```
#define CALIBRATION_MIN_ERROR
```

Figure 16 below shows the message displayed on the E-paper display implemented on the STM32L0xx-DiscoveryL board when running the demo with HSI16 selected as the system clock source. The HSI16, MSI and HSI48 oscillators are calibrated, then both LSI is measured.

Figure 16. HSI16 calibration



4 Conclusion

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

This application note provides two routines:

- **Multi- and High-Speed Internal oscillator calibration:** how to fine-tune the oscillator to the typical value
- **Multi- and 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): 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 discussed the measurement of LSI and MSI oscillators. The internal connection between internal oscillators and embedded timers in the STM32L0xx microcontroller family 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.

5 Revision history

Table 2. Document revision history

Date	Revision	Changes
02-Mar-2015	1	Initial release.
28-Apr-2015	2	Updated RPN

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.

© 2015 STMicroelectronics – All rights reserved