Iztok BLAZINSEK[1], **Amor CHOWDHURY**[1]

Margento R&D (1)

doi:10.15199/48.2016.11.60

Enhancing the accuracy of standard embedded RTC module with random synchronization events and dynamic calibration

***Abstract**. This paper presents a three-step calibration method for improving the accuracy of the standard crystal oscillator, which is running the embedded RTC (real time clock) module in a standalone not networked sensor device. The first is the analysis and verification of the hardware design. The second step includes implementation of the static calibration procedure where special attention is given to the power use. In the last step the dynamic calibration procedure is implemented, based on the random synchronization events. The developed method should offer a mix of good stability and power efficiency, small footprint and competitive price.*

Streszczenie.** Zaprezentowano metodę kalibracji i poprawy dokładności generatora wykorzystywanego we wbudowanym module zegara RTC (real time clock).**Poprawa dokładności modułu zegara RTC z losową synchronizacją I dynamiczną kalibracją

Keywords: accuracy improvement, accuracy enhancement, calibration procedures, calibration methods.
Słowa kluczowe: zegar kwarcowy, kalibracja.

Introduction

The concept of internet of things foresees interconnection of embedded systems, which com-bine a variety of sensors to perform the defined functionality within the network. Most of such embedded systems and sensor networks need a reliable time source to operate properly [1, 2]. In this paper we will present a three step method, with which we will try to improve the accuracy and stability of the RTC module. The first step analyse the current RTC to determinate initial accuracy. Second step is the static calibration with heavy emphasis on the power consumption. The final step is the experimental dynamic calibration with which the accuracy of each individual RTC module is furthers improved.

Presentation of the secure authentication node device

The secure authentication sensor node represents a base device, on with our study is performed. It's designed with the STM32F10x series of microcontroller [3]. This microcontroller features an embedded real-time clock module, which is used to generate precise time. The real-time module for its primary operation uses one 32-bit timer, whose value is increased every second or other selected time interval defined with the module configuration by designer at design stage. If the desired interval is set to one second, the time is reported in seconds. It is up to the developer to convert this value into a usable timestamp. The accuracy of such a system is defined mostly by its external components - crystal oscillator, possible external resistor RE and load capacitors and their PCB (printed circuit board) placement strategy (Figure 1).

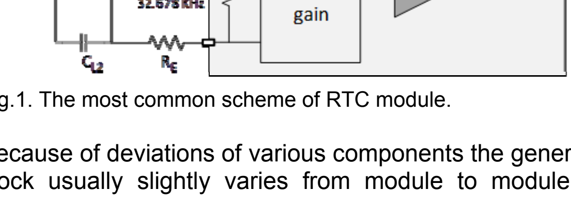

Fig.1. The most common scheme of RTC module.

Because of deviations of various components the generated clock usually slightly varies from module to module. To ensure optimal stability and accuracy, component selection is critical. In general, oscillator inaccuracy can be classified into three different areas:

a) Static inaccuracy – this type of error is usually the result of well-known factors such as crystal frequency deviation due to change of temperature.

b) Dynamic inaccuracy – this is usually the result of either individual component deviation or production process.

c) Instability – this is the result of a bad/wrong choice of components and/or bad/wrong components placement. It is the most difficult error to detect and eliminate. To eliminate this kind of error, an entire solution must be redesigned.

Accuracy and stability analysis

The secure authentication node uses the Epson PC-13F crystal oscillator [4] with load capacitors of 10 pF to govern the embedded RTC module. The ST Microelectronics offers a detailed selection and design guide which we will follow [5, 6]. To successfully apply the procedure, described in [5], the reading of references [7-11] is recommended. Only after verifying that every parameter is in accordance with these in the reference documents, further testing can be done.

To test the accuracy of RTC, five samples of secure authentication nodes were randomly selected for test samples. The test was split in three parts. First two parts will determinate initial accuracy and stability and the part three will check the environmental dependency.

The first part of the test was focused on oscillator base frequency. As the base oscillator frequency cannot be measured directly in the STM32F10x, the RTC module features a special pin on which base the frequency divided by 64 is outputted. On this pin, the output frequency could be measured and analysed. Results are in Table 1:

Table 1. Results of first part of the test.

Sample	S1	S2	S3	S4	S5
measured frequency	512,0036 Hz	512,0030 Hz	512,0026 Hz	512,0024 Hz	512,0016 Hz
measured frequency error	7,0313 ppm	5,8594 ppm	5,0781 ppm	4,6875 ppm	3,1250 ppm

In the first part of the test it appeared that the frequency was slightly higher than expected. Frequency deviation is very small and well inside the manufacturer specification, the first and the most reasonable explanation is that this is the result of normal crystal deviation.

The second part of the test was measuring tick count. This method was designed to verify the module clock stability. As the test duration is precisely known, ticks can be converted to average base frequency and compared to the base frequency measured in the first part of the test. To ensure the best test results, a large timeframe of more than two and a half days was set as the default test timeframe. Test results are presented in Table 2:

Table 2. Results of second part of the test.

Test duration: 2 days 15 h 55 min 14 s and 973 ms					
Sample	S1	S2	S3	S4	S5
ticks count	29454888	29454853	29454832	29454821	29454776
Δt [s]	-1,344	-1,056	-0,896	-0,8	-0,464
Δt [ticks]	-172	-136	116	-104	-60
Extrapolated frequency	512,003 Hz	512,0024 Hz	512,002 Hz	512,0018 Hz	512,001 Hz
Extrapolated frequency error	5,8395 ppm	4,6512 Ppm	3,9382 ppm	3,5648 ppm	2,0370 ppm

The second part of the test is intended to check the system stability. A long timeframe was chosen to get as accurate results as possible. Test results show the average real-time clock error of 4 ppm. The small difference shows that our clock was stable throughout the test time span, which confirms the clock design is not only accurate but also stable.

Third part is intended to analyse environmental impact. The biggest environmental impact is the temperature change. Although the parabolic coefficient is defined in the documentation of the crystal, it is still important to get a precise reference as the design can influence these parameters. To ensure the most accurate results, initial measurements are performed on three samples of the system simultaneously. Results are presented on figure 2:

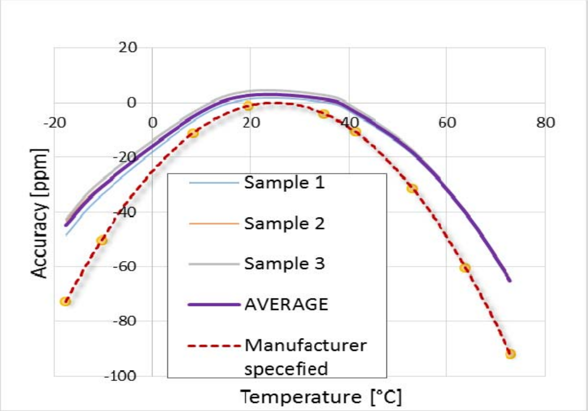

Fig.2. Temperature dependency measurements.

Static calibration – Implementation

The first part compensates the inaccuracy which is caused by the crystal temperature dependency. This problem is well-known and there are many already developed methods enabling temperature compensation [12-17]. The challenge in implementing static calibration is not in the design of the calibration method itself, as it is a simple method of applying the correct calibration equation; but to ensure the method's power consumption is as low as possible. Equation (1) is used as the core of the static calibration method. The proposed method works by sampling the temperature with an on-board temperature sensor embedded in STM32F103 microcontroller (± 2°C accuracy) in regular time intervals. When the temperature is sampled, it is stored in a non-volatile memory. The next time when the temperature is sampled, the average temperature can be calculated from the stored and the current temperature. The average temperature and the sampling interval are then used to update the local time offset value. This offset value is added to the local time value whenever this value is read.

Static calibration – Verification

To validate the method, the time interval and the temperature remain constant. To get the best results, time intervals are set to approximately one hour and each test duration is set to the same value. The temperature condition is changed from -10°C to 70°C.

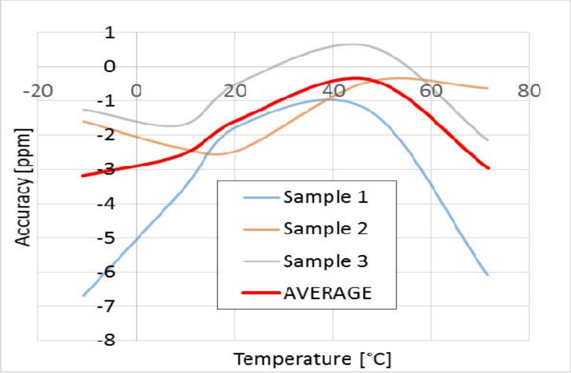

Fig.3. Compensated crystal accuracy vs. temperature.

Static calibration – Optimization for optimum power efficiency

While the proposed method results in a much more accurate local time generation, it has a significant drawback. The method needs periodic temperature measurements and this means waking the device from sleep state at regular intervals. Because the device uses much less power in sleep state (up to ten thousand times less) this represents a significant increase in power consumption. This is not problematic on systems connected to permanent power supply, but it does present a serious drawback on battery-powered systems. To get the best solution, a compromise between sampling time and the accuracy of the correction must be found. To get the best compromise, the environment in which the module will be operating needs to be examined. The best com-promise is based on the minimum timeframe in which the temperature changes for the defined accuracy mar-gin. This temperature margin is defined by the temperature sensor as an accuracy of ±2 °C. If the devices are to be used indoors, where the temperature is fairly strictly regulated, this critical timeframe can be very large or even non-existent. The problem is more apparent when the devices are used in an environment that is not temperature regulated. To define the optimal time interval, analyses of the temperature conditions of the specific location are needed. We took the city of Celje in Slove-nia [18] for our example – Figure 4:

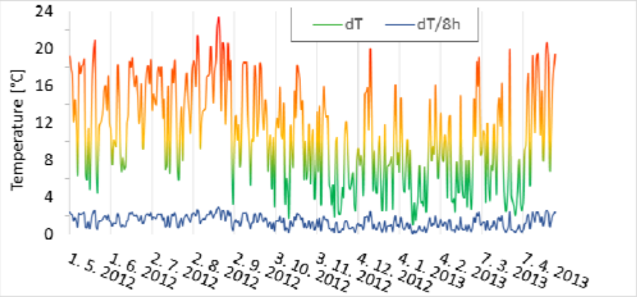

Fig.4. Temperature changes in the city of Celje, Slovenia.

Figure 4 represents yearly temperature measurements in Celje. The value of dT is the difference between maximum and minimum daily temperature. We estimate temperature changes in the worst case from lowest to highest in approximately one third of the day (8 hours). With that estimation the temperature change in one hour can be extrapolated (dT/8h). From this data the time in which the temperature would change by a minimum margin of 2 °C is extrapolated – Figure 5:

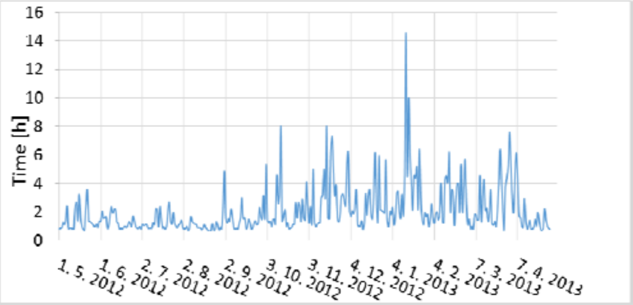

Fig.5. Extrapolation of critical timeframes.

Figure 6 shows critical timeframes extrapolated from data in figure 7. Timeframe changes are quite substantial. From this information and with the use of median and average method we try to extract the most optimized critical timeframe value tc:

Table 3. Extrapolated critical timeframe value tc.

Max.	Min.	Average	Median
14.5 h	0.7 h	2.1 h	1.5 h

Table 3 shows the critical timeframe maximum and minimum values, in which the temperature change of 2 °C is detected, and the two optimized values. One is extrapolated with the help of the average method, and one with the help of the median method. The optimized values differ significantly. To get a better idea of how this affects the secure authentication device's lifetime, each configuration is inspected. Because of the low cost, reliability, low idle discharge and long shelf life alkaline batteries are mainly used to power the secure authentication module. To get a better representation of how critical time influences power consumption, two test scenarios were chosen. The first assumes the lock module is running idle most of the life time, and the second scenario represents normal use. With normal use it is assumed that the authentication device is triggered 7 times per day, out of which fife times successfully. With consumption measurements executed, power consumption in each of the stages is presented in Table 4:

Table 4. Power consumption in various stages.

Event	Average active current	Average active time [s]	Average activations per day
Idle time	0.007 mA	/	/
Sampling temperature	20 mA	1	24h / t_c
Active	50 mA	30	7
Maximum power use	500 mA	0,5	5
Battery aging	3 % loss of capacity / year		

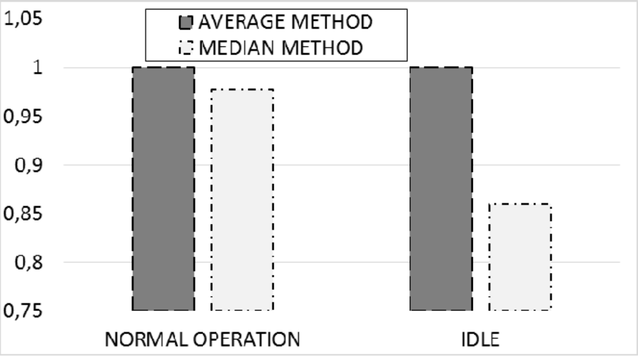

Fig.6. Difference in lifetime.

From the data in Table 4 the effect of the periodic wake up on power consumption, using average and median method, can be estimated. An ideal battery is assumed. Figure 6 shows how power consumption is affected by the change of the critical timeout. In a normal operation mode smaller sampling interval would result in 2 % and in idle mode in 14 % of decrease in the lifetime of the battery. As the battery would probably already be empty, it is clear that the critical timeout change from 2 to 1.5 hour does not have any major effect on power consumption if the module is regularly used. Because smaller critical timeout would not significantly affect the power consumption it has been decided to set the timeout to 1.5 h.

Dynamic calibration – Implementation

While the method with static calibration is designed and implemented at the design time and is the same on all modules, the dynamic method targets each of the modules individually. The method is designed to analyse the module and to try to improve the accuracy with an individual approach at run time.

Dynamic method works by acquiring the most accurate timestamp from an online source when the device is triggered. As the device is triggered by the users, these synchronization events are unpredictable. The most important requirement is that the received timestamp is really up to date. If the time is offset or even wrong, this can have catastrophic consequences. The proposed calibration procedure works by calculating and updating the time differential coefficient between online and local time. This time differential is then applied to the time every time it is read. The basic algorithm is presented in Figure 7:

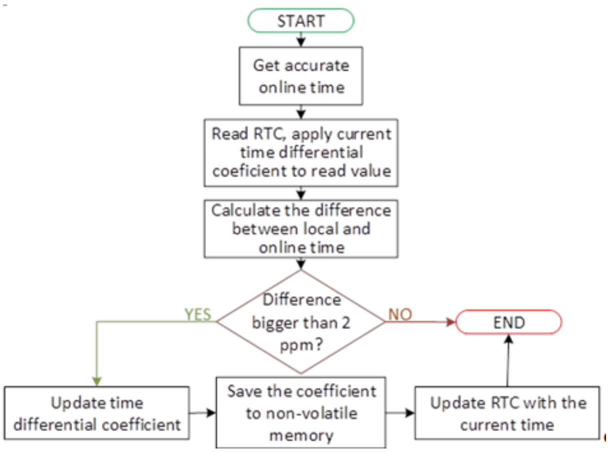
START
Get accurate online time
Read RTC, apply current time differential coeficient to read value
Calculate the difference between local and online time
Difference bigger than 2 ppm?
YES
NO
END
Update time differential coefficient
Save the coefficient to non-volatile memory
Update RTC with the current time

Fig.7. Dynamic calibration algorithm.

Immediately after the online time is successfully acquired, it is stored and compared to the current device time. If the difference is bigger than the desired maximum margin, the time differential coefficient is calculated between local and online time. The calculated value is then added or subtracted from the current differential coefficient value. The RTC is updated with the accurate time and the newly calculated time differential is saved to be used the next time the RTC module is read. This type of improvement is targeted at upgrading static calibration by adjusting the frequency on each individual module. This includes deviations, which are caused by component deviation, production process or aging of components.

Dynamic calibration – Verification

For test purposes it has been decided that dynamic calibration is to be implemented and run manually. First the initial set of measurements are performed. From these measurements the time differential coefficient for each of the test samples is calculated. Next, the second set of measurements is performed under the exact same condition. The RTC values are again read, and the previously calculated time differential coefficient is applied to each of the results. The ideal result would be total elimination of the remaining error. The test is performed at the constant temperature of 10 °C.

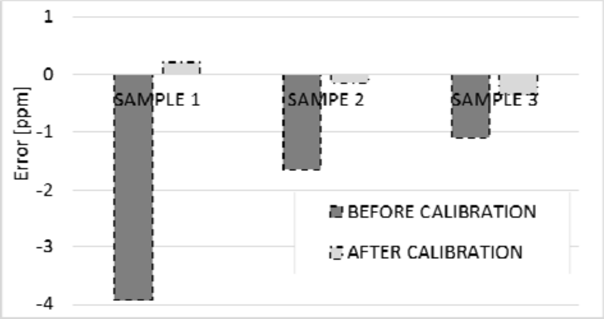

Fig.8. Results of dynamic compensation.

Figure 9 shows the results of a statically compensated system versus the combination of static and dynamic calibration. The accuracy is significantly improved. The dynamic calibration under perfect conditions seems to bring the system to its full potential achieving the maximum possible accuracy of ±0.36 ppm, which is well inside of desired ±5 ppm margin.

Dynamic calibration – Optimisation

The most important factor in the dynamic calibration method is to always ensure the online time is accurate. If the time is inaccurate this can result in catastrophic RTC failure. To try to reduce this risk, a maximum difference between local and online time is specified. If the detected time difference is bigger than the maxi-mum specification, the device tries to confirm the difference by waiting a specified time interval and requesting new online time. If the time difference is equal as the time difference calculated from the previous trans-action, the new time differential coefficient is set. There is still the risk of online time being tampered with. Be-fore the calibration is deployed, a method which will ensure the online time is properly maintained and efficiently transferred, must be implemented. As this method does not require periodic adjustments (it is executed only when the device is already n active mode) there is no need to wake the device at specific intervals – there is no significant impact on power consumption.

Conclusion

In this article the potential of one of the embedded RTC is explored, analysed and improved with the help of calibration methods with the aim to achieve higher stability, accuracy, and low consumption as primary goal. To verify the stability of the design the recommended set of calculations was executed, the hardware design was examined and tested in real time. After evaluation, two separate calibration methods were designed for calibration. The first part was intended to compensate frequency deviations caused by environment changes, and was implemented in the form of periodic temperature sampling and continuous adjustment of time. This method is well known and greatly improves the system accuracy when the system is ex-posed to an environment where temperature is not stable. The method however does not address individual modules' characteristics. With the intent to enhance the accuracy on each of the modules separately, the dynamic calibration method was proposed. The dynamic method works on the principle of comparing the current local time with accurate online time. This method proved to achieve high accuracy under ideal conditions and has many advantages over static methods; most notable of them is the ability to adapt to each module separately. It even compensates some of the future predicted crystal deficiencies such as change in frequency due to crystal aging characteristics. The downside of the dynamic method is the need for very accurate online time. The result of this study has greatly improved accuracy of the real time clock used by the secure authentication nodes. Although this solution is not the most accurate it is very interesting with large production, as it offers very good compromise between the price and the precision.

***Authors**: dr. Iztok Blazinsek, Margento R&D, Gosposvetska c. 84, 2000 Maribor, Slovenia, E-mail: iztok.blazinsek@margento.com;*
dr. Amor Chowdhury, Margento R&D, Gosposvetska c. 84, 2000 Maribor, Slovenia, E-mail: amor.chowdhury @margento.com

REFERENCES

[1] Nizhnik, O.; Higuchi, K.; Maenaka, K. Quartz Resonator Based, 0.12 μW, 32768 Hz Oscillator with ±100 ppm Frequency Accuracy. J. Low Power Electron. Appl.2011, 1, 327-333.

[2] Kumar A.; Madaan P.; Oscillators: How to generate a precise clock source. Cypress Semiconductor 2013.

[3] STM32F103xC Documentation. Available online: http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1031/LN1565/PF16448

[4] Epson Toyocom FC-13F crystal datasheet. Available online: http://www.farnell.com/datasheets/1563923.pdf

[5] Oscillator design guide for ST microcontrollers. Available online: http://www.icbase.com/File/HTML/hotic/html/docs/15287.pdf (accessed on 12 November 2013)

[6] Zhou, H.; Nicholls, C.; Kunz, T.; Schwartz, H.; Frequency accuracy & stability dependencies of crystal oscillators. Carleton University, Systems and Computer Engineering 2008.

[7] Williams J.; Sub-μA RMS Current Measurement for Quartz Crystals. Linear Technology Magazine 2007, 41-42.

[8] Best Practices for the PCB layout of Oscillators, Available online: http://www.atmel.com/Images/doc8128.pdf

[9] TPS65950/30/20 32-kHz Oscillator Schematic and PCB Layout Guide, Available online: http://www.ti.com/lit/an/swca076/swca076.pdf

[10] Crystal Considerations with Dallas Real Time Clocks, Available online: http://www.zmitac.aei.polsl.pl/Electronics_Firm_Docs/DALLAS/app58.pdf

[11] MSP430 32-kHz Crystal Oscillators, Available online: http://www.ti.com/lit/an/slaa322b/slaa322b.pdf

[12] RTC Crystal Deviation & Compensation Simulator V1.0, Available online: http://www.ricoh.com/LSI/product_rtc/info/compensation_m.pdf

[13] Huang, L.; Fu, W.; Tan F.; A temperature compensating method for crystal oscillator, Chinese Patent Application Number 200410022680.3, June 3, 2004.

[14] Li, M.-Q.; Huang, X.-H.; Tan F.; Fan, Y.-H.; Liang X.; A Novel Microcomputer Temperature-Compensating Method for an Overtone Crystal Oscillator", Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions., vol.52, no.11, pp.1919-1922, Nov, 2005.

[15] Achenbach, R.; Feuerstack-Raible, M. ; Hiller, F. ; Keller, M. ; Meier, K. ; Rudolph, H. ; Saur-Brosch, R.; A digitally temperature-compensated crystal oscillator. IEEE Journal of Solid-State Circuits 2000, 35, 1502-1506.

[16] Deno, S.; Hahnlen, C. ; Landis, D. ; Aurand, R.; A low cost microcontroller compensated crystal oscillator. Frequency Control Symposium (FCS), 1997 IEEE International, Orlando, U.S., 28-30 May 1997, pp. 954-960

[17] Esterline, J.C.; Temperature compensation of crystal oscillators using an Artificial Neural Network. Frequency Control Symposium (FCS), 2012 IEEE International, Baltimore, U.S., 21-24 May 2012, pp. 1-7

[18] Slovenian Environment Agency - National Meteorological Service of Slovenia, Temperature statistics for the city of Celje, Avalible online: http:// http://meteo.arso.gov.si/