Providing Accurate Time Reference with the STM4F32Discovery Embedded System for Purposes of Crystal Calibrations
Iztok Blazinšek
Margento R&D d.o.o. Gosposvetska cesta 84, 2000 Maribor, Slovenia

iztok.blazinsek@margento.com

ABSTRACT
Crystal oscillators are today present in almost every embedded system. They are usually used for precise frequency generation, which can then be used for generating driving processors, microcontrollers, radio transceivers, real time clock generators and other devices which rely on precise and stable frequency for proper operation. This article presents the design of a base system which enables the operator to measure precise time intervals. These intervals can be used to verify clock's stability and to adjust hardware or software parameters if necessary, so as to assure the most accurate frequency generation.

Keywords: accurate time base, crystal measuring, oscillator calibration

1. INTRODUCTION
Crystal oscillators are today used for generating clock signals in most embedded systems, as they are one of the most accurate and efficient ways to generate oscillations with stable frequency. Although crystal oscillators usually generate highly stable frequency, they are still susceptible to external factors, such as temperature, humidity and general hardware design, which, if they are not compensated for, can be the cause of frequency deviations [1], [2]. While many crystal solutions exist which already compensate for temperature deviation, design engineers still must address possible deviations caused by the design.

There are several methods available for detecting frequency deviations. With large deviations, the frequency can simply be measured with a frequency meter or an oscilloscope, whereas a spectral analyser needs to be used for tuning the output frequency when a crystal is used as the source of base frequency for radio-frequency transceivers. Although these techniques produce fairly good results, they are not accurate enough for systems which need to generate frequency with extreme accuracy, such as precise signal generators and RTC (real time clock) generators. With such systems, frequency deviation must also be measured with extreme accuracy.

The easiest way to precisely measure the frequency is to count the generated pulses over a specified timeframe and then extrapolate the accuracy. All crystal oscillators have their accuracy stated among their specifications, usually by specifying the deviation in parts per million (ppm). A mainstream oscillator normally features a 20 ppm deviation margin, which can however be improved with various techniques [3]. For example in case of oscillators that are most commonly used to drive RTC systems at 32.768 kHz a deviation of 20 ppm translates to an error margin of 0.072 s/h.

This article presents the design of an embedded system with the potential to calibrate multiple devices, such as RTC, with the accuracy margin of 1 ppm or more for 32.768 KHz oscillators.

2. THE BASIC PRINCIPLE
The most basic method for precisely tuning low frequency usually involves testing over long time periods, where error can accumulate over time and is therefore easily detectable. The longer is the test timeframe, the more accurate is the test result. An obvious disadvantage of such a method is the extremely long time period needed for error to accumulate. To get good results, the operator must run the test for at least a few days in a precisely controlled environment. Because this method takes such a long time for every test, it is more appropriate for final testing, whereas for the process of research and development a fast turnaround test time is desired. For the purpose of this research, a target test turnaround time of one hour is set, with the minimum accuracy resolution of one millisecond (one microsecond translates at the frequency of 32.768 kHz to 0.033 ppm).

3. SYSTEM DEVELOPMENT
There are two main challenges when designing a system which should enable precise time interval capture. The first is to generate and sustain precise time reference, as this defines the basic accuracy of the system. The second challenge is to implement a communication interface between the measurement control system and the test subject. The measurement system must be able to issue at least three commands, which are used to start the test, to stop it, and to capture results and transfer the current frequency count from the device to the operator.

3.1 The interface between the Control System and Test Samples
The first step is to define a communication interface between the control system and the samples which are being measured. The easiest way to trigger each test is to simply connect one switch to all test samples (trigger pin). This switch controls the duration of the test. Samples are connected via another connection to the control system with a serial interface, so that they can
generate a response to received events (UART pin) – Figure 1.

When the switch closes for the duration of more than 1 and less than 1.5 second (t1), test samples start to run the test (start event). At this point frequency counters on all the samples are reset. When the switch is closed for more than 1.5 second (t4), the test ends (stop event). Each sample saves the value at the frequency counter immediately after detecting the stop event. The saved frequency counter value is later extracted from each module independently by closing the switch for less than half a second (t7) (get event). At each start and stop event, responses are sent from test samples via a serial interface to the control system. The control system then extrapolates the time between start and stop events, and uses it as precise reference time. The basic chronological test scheme is shown in Figure 2:

The processing times of events on the samples and on the PC control system could impact the accuracy.

3.2 Prototype I

To verify the efficiency of the protocol, we have constructed a test environment, where the control system is implemented with a simple switch and the precise time base is implemented with the use of a personal computer. A basic scheme of prototype I is presented in Figure 3:

The personal computer can get precise time from online time servers. From experience with C# programming environment, it is known that the local time can be accessed with a resolution greater than one millisecond (for example C# Stopwatch class [4]), which would satisfy the minimum accuracy requirements set in section 2.

To check the accuracy a simple test has been devised. A test embedded system is set up to generate events at precise time intervals. The chosen intervals are half a second, one second, 10 seconds, one minute and 5 minutes. The control interface running on the personal computer captures the event generated by the embedded device, and generates a precise timestamp. As the precise timeout between events is known, the measurement error can be extrapolated - Figure 4.
The results show that the protocol is fast and efficient. The maximum stable accuracy achieved by this method changes with the duration of the test. When the test duration is short, the majority of timeout accuracy is better than when the test duration is long. The accuracy margin seems to settle at 50 ms.

The high accuracy deviations could be the result of the hardware and software combination, as none is built for generating time with such accuracy. We assume that a large part of the error is the result of the controlling firmware. Since the firmware runs on an application level with normal priority, it may experience delays when an event is received by UART (Universal asynchronous receiver/transmitter), as the computer is busy processing other threads. These threads delay the processing of the event, which then causes the timestamp to be offset.

While this method should be effective for most calibration processes, it is still far from the goals set in section two.

3.3 Prototype II

Tests results from section 3.2 show that a personal computer is not accurate enough to be used as a very precise time base, so it must be replaced. To achieve a better resolution, the time base for prototype II is implemented in the form of an embedded system. The embedded system, chosen to replace a personal computer as the precise time base, was STM32F4Discovery [5] in combination with SIM 18 GPS [6] module. The STM32F4Discovery module connects via a USB (universal serial bus) interface to the personal computer, on which the operator interface is running.

To take full advantage of the SMT32F4Discovery board, we have designed the trigger signal to be automated. The operator simply issues the appropriate command on the operator interface and the STM32F4Discovery control board generates the appropriate signal on the “TRIGGER” line. In addition, each sample is now connected via a separate UART interface to the control board. This gives the operator the possibility to transfer the results from one or from all samples simultaneously. Beside the standard protocol interface, which is composed of two lines (section 3.1), one more auxiliary connection is implemented. The “WAKE” line is dedicated to waking the device from sleep state if necessary, before issuing the appropriate command. The basic scheme of prototype II is presented in Figure 5:

![Prototype II scheme](image)

**Fig 5:** Prototype II scheme

The main idea is to utilize the “TIMEMARK” output of the SIM18 GPS module, which, under the condition of successful GPS connection, pulses at precise intervals of one second with the accuracy of one microsecond. These pulses can then be counted to generate the precise time base. The system is now able to generate the time base with the accuracy of one microsecond.

The remaining challenge is to provide high accuracy when the timeout is not generated by the control system, but by one of the test subjects. The “TIMEMARK” is pulsing every second and increasing the second counter value. To get a greater resolution, one of the timer modules on the microprocessor is configured in a free running mode with a tick interval of one microsecond. The timer is reset every time the “TIMEMARK” output pulses. In this configuration the timer count should directly represent the number of microseconds, elapsed from the last “TIMEMARK” pulse, thus giving us a resolution of 1 µs. When the time base is required, it is calculated as the second count plus the microseconds stored in the data register of the free running timer.

To further improve the accuracy of the system, the main 16 MHz oscillator, driving a microprocessor with ±20 ppm accuracy, is replaced with a special temperature-stabilized (TXCO) oscillator with ±1.5 ppm accuracy. To test the system accuracy, the following test has been devised. The test procedure captures the free running timer value every time “TIMEMARK” pulses. If the device is accurate, the timer value should increase by exactly one million. Figure 5 presents test results.
Results in Figure 6 show that the goal of one microsecond accuracy has been achieved, but they also show that there are some deviations in two out of ten tests. To examine these deviations further, another test has been devised with larger timeframes. The selected timeframes are 10 seconds, 1 minute and 5 minutes.

Results in Figure 7 show that the accuracy deviation seems to be linearly accumulating with test duration. This is probably due to the offset in the oscillator base frequency. An extremely precise oscillator should result in great clock stability, but does not by itself guarantee a more accurate time base, as it does not guarantee that the generated frequency will be precisely 16 MHz ± 1.5 ppm. The basic frequency can be offset because of hardware design or because of the impact of external factors.

Because error is linearly accumulated, it can be eliminated by implementing a time differential factor. This factor is calculated with equation 1:

\[ dt = 1 + \frac{AVERAGE(Deviation_{30min})}{300000000} = 1 + 1.19667E-07 \]  

The time differential factor must then be applied to every calibration result. Its effectiveness is checked with additional measurements. Figure 8 presents the results.

The time differential factor reduces error by more than 70%. Because this factor may not always be same, as it can change due to the environment, it is recommended that at least one test measurement is done at the end of each test to calculate the optimum differential factor (\( dt \)). Our system is set to perform time differential measurements three times before and three times after each test. The final time differential value is computed as an average of all six measurements.

Calibration over longer time periods should improve the system accuracy. A further test has been devised to verify if increased resolution results in better accuracy. The resolution is set to 0.25 µs. We must mention that we still use the tick from the GPS module as the basis for the test. Manufacturers guarantee the accuracy of the tick to be one microsecond. When the resolution is increased to a sub-microsecond level, the test results may not be absolute. The result of increasing the resolution is shown in Figure 9.
It seems that the system is capable of measuring time with a sub-microsecond resolution. It needs to be noted that the time differential coefficient seems to have the opposite effect, as 8 out of 10 results yielded exactly four million ticks, which translates into exactly one second. The time differential coefficient then adds about 0.5 tick to the result, which produces an error of 0.13 µs.

7. CONCLUSION

The purpose of this article was to implement a precise time base with the use of a simple embedded system, which would enable time measurement on multiple devices simultaneously with a resolution of at least one microsecond.

For the first prototype a personal computer was used. As expected, it was discovered that a personal computer cannot provide sufficient accuracy. To improve the accuracy, a special time base was designed with the use of the STM32F4Discovery embedded board in combination with a GPS module. While the GPS module provides precise ticks in precise second intervals, the STM32F4 microcontroller provides the sub-second resolution.

Preliminary measurements of the new system showed that the error margin is now down to a few milliseconds. To improve the accuracy even more, the base crystal driving the STM32F4 board was substituted with a very precise temperature-stabilized crystal. As the improved stability of the crystal only results in stable frequency and not necessarily in precise frequency, the time differential was extrapolated, to improve the accuracy even more.

With all these improvements the system is now able to provide precise time reference of almost unlimited length with the accuracy at a sub-microsecond level. We will use this time reference base to try and improve the real time clock in several of our designs.

ACKNOWLEDGEMENT

This operation is partly financed by the European Union, European Social Fund. The operation is implemented in the framework of the Operational Programme for Human Resources Development for the Period 2007-2013, Priority axis 1: Promoting entrepreneurship and adaptability, Main type of activity 1.1.: Experts and researchers for competitive enterprises.

REFERENCES


