Low-Power High-Accuracy Timing Systems for Efficient Duty Cycling

Thomas Schmid, Jonathan Friedman, Zainul Charbiwala,
Young H. Cho, Mani B. Srivastava
Department of Electrical Engineering
University of California, Los Angeles
{schmid, jf, zainul, young, mbs}@ee.ucla.edu

ABSTRACT

Time keeping and synchronization are important services for networked and embedded systems. High quality timing information allows embedded network nodes to provide accurate time-stamping, fast localization, efficient duty cycling schedules, and other basic but essential functions - all of which are required for low power operation.

In this paper we present a new type of local clock source called Crystal Compensated Crystal based Timer (XCXT) and a number of novel algorithms that effectively utilize it to achieve low power consumption in wireless sensor networks. The XCXT has timing accuracies similar to timers based on temperature compensated crystal oscillators (TCXO) but has a lower implementation cost and requires less power. Our initial 8MHz prototype unit, using the simplest algorithm, achieves an effective frequency stability of ±1.2ppm and consumes only 1.27mW. On the other hand, commercially available TCXOs with similar stability can cost over 10 times as much and consume over 20mW.

In addition to the prototype, we will present algorithms that will improve the XCXT's power consumption by at least 48%, depending on application and environmental conditions. We will also show how XCXT's power efficiency can be improved even further by employing clocks at different frequency when different time granularities are required by an application.

Categories and Subject Descriptors

B.m [**Hardware**]: Miscellaneous

General Terms

Experimentation, Measurement

Keywords

Time Synchronization, Oscillator, Clocks, Emulation, Low-power clocks

ISLPED'08, August 11–13, 2008, Bangalore, India..

1. INTRODUCTION

The quality of network-wide time information available to an embedded system is determined by three sources of variance. In order of increasing significance: (1) In finite-bandwidth systems the precise moment of the syncronization symbol is imprecisely defined. (2) In embedded applications where network time maintainence is one of several tasks on the processor, computational and time-to-service latency is not necessarily constant. (3) Last, the quality of each device's local clock source (quantization, environmental stability, frequency tolerance, aging, and drift) plays a significant role in the relative clock drift among network peers between resynchronizations (resync) and, consequently, constrains the rate at which resync must be performed.

Although our ranking is rooted in the fundamental physics of the situation, early implementations of time-sync protocols in sensor networks elevated the significance of the second source well above the third. Consequently, much of the early work on time synchronization, specifically for sensor networks, focused on the design of protocols and regression mechanisms to cope with the noisy time stamps that their implementations achieved. Improvements of the local clock source itself were dismissed under the notion that anything with drift characteristics better than a simple crystal oscillator would add too much cost and utilize too much energy. This is despite the dominance of drift as a source of clock frequency error.

Crystal oscillators which exhibit a frequency instability of a few tens of ppm or worse as environmental conditions vary are the local clock source of choice in prevalent sensor nodes, even the high end ones. While more stable clock sources do exist (GPS with PPS signal, TCXO/OCXO/MCXO), they tend to be too costly, too power hungry, or too limited otherwise (ex. GPS doesn't work indoors) to proffer a viable alternative. With ordinary crystal oscillators, the high clock drift results in a build-up of clock offset error without frequent resync with a reference node. That is not easily accomplished in battery-driven systems operating at low duty cycle nor in systems which support extended periods of disconnected/offline usage.

This paper will describe a technique that exploits a dual crystal system based on our prior work presented in [6]. We show that this system not only achieves a highly accurate, low-drift timer, with high granularity, but also consumes very little power during system sleep and, thus, is ideal for low duty-cycle embedded systems. Traditionally, the need for fine granularity drives design towards high frequency crystals which impact the energy budget of an embedded device tremendously – especially during deep sleep since the timer unit must remain active to keep track of time. We will show that our XCXT offers some interesting alternatives while maintaining granularity. After a brief introduction to *Differential Drift*, the concept behind our XCXT, this paper will expand our prior published power measurement and analysis and describes a technique to reduce the power consumption even further.

1.1 Clock Accuracy and Duty Cycling Performance

Duty-cycling is a well known technique to lower the average power consumption of high-performance embedded systems that have a bursty usage pattern. At times of inactivity, the main CPU and communication subsystems are turned off and thus consume little power. In [2], Dutta et al. make the case that the lower limit of duty cycling for current sensor network nodes with typical clock stabilities of 50ppm is 0.02%. They argue that in order to achieve even lower levels of duty-cycling, the clock system needs a smaller drift.

The MAC protocol is one of the main components that enables duty-cycling in networked systems, i.e., networked nodes need to wake up at the same time in order to communicate with each other. SCP-MAC [8] is considered to be the current state of the art for sensor networks and achieves a duty cycle as low as 0.1%. In [8], Ye et. al. give a theoretic analysis of the SCP-MAC protocol performance based on a standard 30ppm crystal oscillator. Using the same methods and code generously provided by the authors of [9], we studied what the improvements are given a better clock system. Figure 1 illustrates the improvement if we go from a 30ppm clock source to 1ppm. We can see that we have a 30% to 50% improvement in duty-cycling, which results in a 30% to 50% improvement in battery lifetime. There is only one drawback in this analysis. Since the duty-cycled node already consumes only $67\mu W$[1] in the worst case, an improvement of %50 means a gain of about $30\mu W$. Therefore, the improved clock system can not use more than $30\mu W$ or else the gain in energy would be nullified again.

This short analysis of SCP-MAC shows that the clock system plays an important role in duty cycled systems. Great care has to be given to its design, or else its possibilities of saving power are easily offset by an increased standby power consumption of the more accurate clock system. Additionally, as we will show in Section 4, if high accuracy and high granularity clocks are needed for a specific application, like localization systems or time synchronization between nodes, the standby power consumption of such clocks significantly increases and can not be ignored if the nodes need to be duty cycled. We will give an idea for an algorithm that solves this problem in a “Smart Timer Unit” in Section 5.

2. RELATED WORK

The basic idea of Differential Drift, i.e., exploiting two components to compensate its frequency drift for each other, isn't new. In [7] Schodowski introduces a temperature sensing device using a dual-harmonic-modecrystal (SC cut crystal). The two harmonics of the SC cut crystal have different temperature behavior. Mixing these two frequencies results in a beat frequency that is proportional to the temperature. Subsequently, Bloch et. al. [1] develop the Microcontroller Compensated Crystal Oscillator (MCXO) based on said SC cut crystal. This MCXO achieves the precision of an Oven Controled Crystal Oscillator (OCXO) though consumes only a fraction of its power (∼70mW instead of ∼1.5W) since it doesn't need an active heating element.

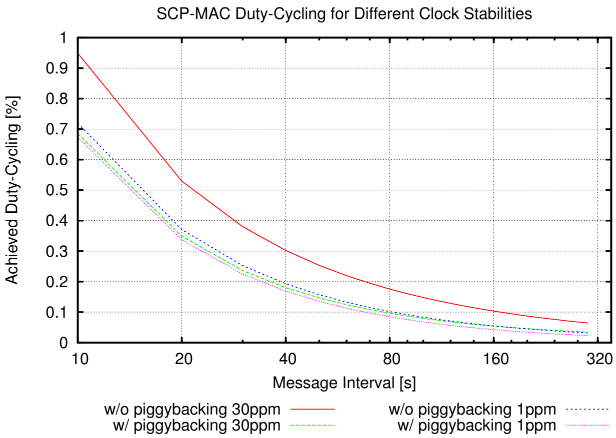

Figure 1: Duty-Cycling performance of SCP-MAC using different clock source stabilities. Note that with a 1ppm clock source, the difference between piggybacking the sync messages on regular data messages or not virtually disappears.

Measuring the temperature using two AT-cut crystals, which is similar to our basic approach of measuring drift, has been done by Satou et al [5]. Nevertheless, all the prior work focuses on ways to generate a stable frequency for the purpose of building accurate oscillators. Our research on the other hand did not begin with the sole interest of building a low drift oscillator. Instead, we sought to find a way to provide accurate time when an application requests it.

The contribution of this paper are the low-power timer algorithms for wireless sensor network applications, and a full working prototype that achieves a 5× better energy performance compared to commercially available systems, with the potential of higher gains in future hardware iterations.

3. THE CRYSTAL COMPENSATED CRYSTAL-BASED TIMER (XCXT)

We introduced the concept of Differential Drift and the algorithm for the XCXT in [6]. For the sake of completeness of this article we repeat the main principles and review the hardware implementation of the XCXT which we evaluate in subsequent sections.

It is a well known fact that crystals change their frequency with changing temperature. The AT-cut crystal, which accounts for about 75% of the current crystal market, has a cubic frequency drift vs. temperature curve, and its linear parameter changes if the crystals are cut at a slightly different angle. This difference in angle, and thus difference in temperature response, can be exploited in a two phase algorithm. Let's assume we have a system consisting of two AT-cut crystals, each one of them cut at a slightly different angle, and a timer unit with two counters. Each counter is connected to one of the crystals and is thus incremented according to each individual's crystal instantaneous frequency.

---

[1] Power consumption of a TI MSP430 and a ChipCon CC2420 radio chip as found on the TMote Sky platform [4]

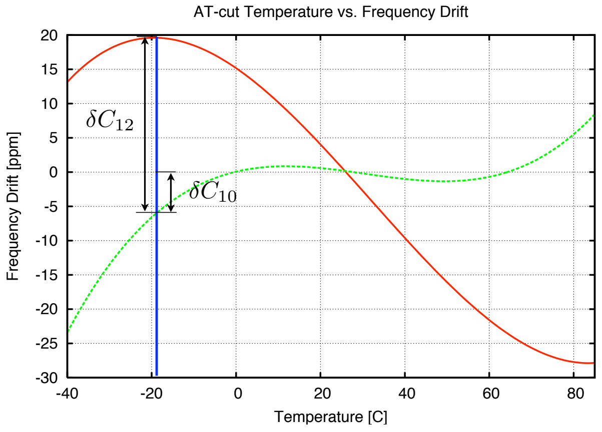

Figure 2: Differential Drift explained. For each temperature, we measure the difference between one of the clocks and the reference C_{10} (0 ppm frequency drift), and the difference between the two clocks C_{12}.

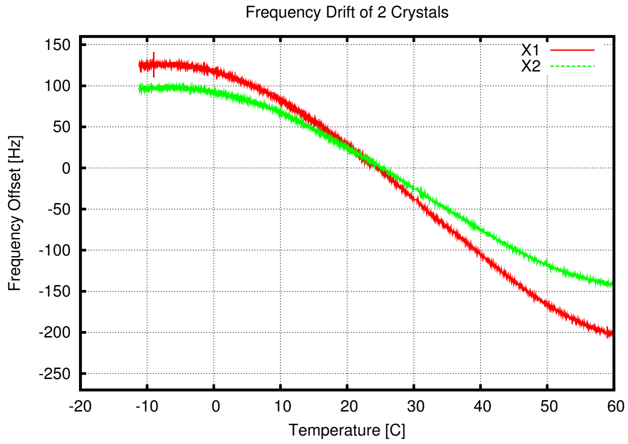

Figure 3: The two curves are the measured drift of two AT-cut 8MHz crystals. The crystals are from two different manufacturers to increase the probability of different cut angle.

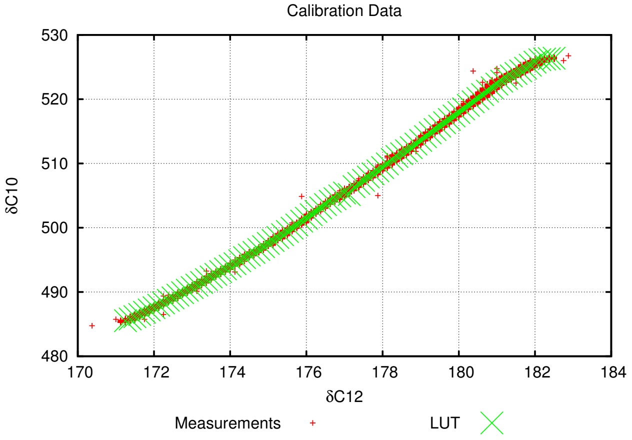

Figure 4: After the calibration phase, we generate the lookup table (LUT) from the obtained drift measurements (small red crosses).

Figure 2 shows the theoretic frequency drift vs. temperature curves, and Figure 3 depicts the frequency drift vs. temperature curves of two 8MHz AT-cut crystals from two different manufacturers. We can see that their slopes differ slightly and thus match the theoretic expected curve. These two crystals represent an example setup for our two phase algorithm.

In the first phase, the calibration phase, the two crystals with nominal frequency f_0 are put into a temperature chamber and subjected to the full operating temperature range. Over this range, the timer unit measures the drift between the two crystals, denoted as C_{12}, and the drift between one of the crystals and a stable reference clock C_{10} (see Figure 2). The measurements are taken at a frequency of $f_s << f_0$, and the result is a calibration curve. Figure 4 illustrates this for the case of our two 8MHz crystals. Converting this curve into a lookup table (LUT) greatly simplifies the implementation on an embedded system since no floating point math or curve fitting has to been done on the microprocessor which in turn reduces CPU load and thus power consumption. Note that this calibration phase is similar to that used for a temperature compensated crystal oscillator (TCXO) during manufacturing. Each TCXO needs to be calibrated individually in order to achieve a good stability before it leaves the factory.

The second phase happens during runtime. Now, a microprocessor initially estimates the sampling interval f_s to measure the drift between the two clocks $\tilde{C}_{12}$. It then uses this estimated drift measurement and the LUT from the calibration phase to find the drift between one of the clocks and the reference clock and thus successfully compensates for any temperature induced clock drift.

3.1 XCXT Implementation and Hardware

After successfully simulating and emulating the differential drift algorithm (presented in [6]) we continued and implemented the whole algorithm on the TMote Sky [4], a Texas Instrument MSP430F1611 based sensor network platform. We chose this platform because the MSP430 has two crystal inputs and two timer units, two of the prerequisites to implement our algorithm. Our implementation is written in C and currently occupies 3292 bytes. We expect this number to shrink further as we optimize the code. Additionally, the implementation makes heavy use of the two timer units and its input capture and timer output capabilities. This is necessary in order to minimize unpredictable software interrupt latencies that could introduce large errors in timer measurements.

We subjected our hardware implementation to the same environmental conditions as used for the calibration phase, and measured it's capability to self compensate. Figure 5 shows the compensation results and compares the uncompensated clock to the compensated clock output. The achieved mean stability is 0.47ppm and the standard sample deviation is 0.31ppm over the temperature range of -10°C to 60°C, resulting in an effective frequency stability at a 95% confidence interval of ±1.2ppm. This is very close to the simulated theoretic accuracy of 0.14ppm standard deviation that we measured by extracting the drift model of the two 8MHz crystals and using them as an input to our simulation framework.

Our hardware implementation showed us that the concept

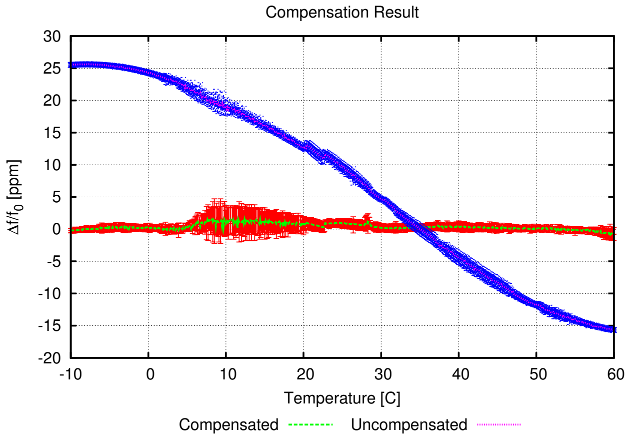

Figure 5: Using the LUT from the calibration data, we can successfully compensate one of the crystals for its temperature drift. Over the full temperature range of -10°C to 60°C we measured a standard deviation of 0.31ppm.

of Differential Drift not only works in simulation, but that we actually can build such a system in an XCXT and achieve clock stabilities close to what the simulations predicted. It is now of interest to investigate the power consumption of the XCXT since this is not something our simulation framework can predict.

4. XCXT POWER MEASUREMENTS

When we first measured the average power consumption of our XCXT implementation at 1.27mW, we were already impressed since we didn't do any special optimizations to decrease the power consumption. Additionally it is to note that a commercially available TCXO that achieves about the same precision already consumes 5mW [3][2] without the drive and oscillatory circuitry, which are included in our measurement. This lead us to investigate which parts of our hardware implementation spend how much energy, in order to optimize our hardware even further.

4.1 Microprocessor Power Consumption

The MSP430 has several low power states. In the second lowest one, LPM3, all the peripherals, except one timer unit and one clock signal, are turned off. In that state, the microprocessor consumes as low as $7\mu A$ if it uses a standard 32kHz crystal as its timer input. Unfortunately, we can not use this low power mode in our XCXT implementation since we need two active timers and two input clock signals in order to measure the drift between the two of them. Thus, the lowest possible mode is LPM1, where the CPU and the internal digitally controlled oscillator (DCO) are turned off, but both timer units are kept on. The microprocessor wakes up from sleep only if one of the timers overflows, or the timer unit interrupts the microprocessor to measure the difference between the two frequencies. At that point, the power consumption sharply rises as depicted in Figure 6. However, the computations are small and thus the length of active time is very short. As we will show in the next subsection, the average power consumption is dominated by the power consumption of the oscillatory circuitry and timer units that need to keep track of the counters. In Section 5 we will go deeper into techniques and algorithm improvements that exploit this fact. But first, let's look at the power consumption for one oscillator and its timer unit.

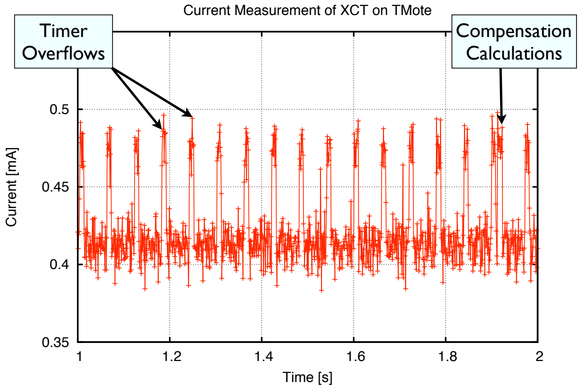

Figure 6: This is a typical current consumption plot of the XCXT. The measurements were taken at 3V. The spikes represent instances when the microprocessor woke up because of a timer overflow which has to be treated.

4.2 Oscillator and Timer Power Consumption

Provided with the power measurements from last section, we can see that most of the energy of the XCXT is spent while the CPU is off. Thus, it is very important to be able to attribute that energy to the specific components in order to optimize the XCXT power efficiency. In a series of experiments we tested the different sub-components of the microprocessor and the TMote platform. We found that the 8 MHz oscillators and the according clocking and timer subsystem consumes most of the remaining power budget at sleep. In order to get a more detailed view, we conducted several different experiments with different combinations of number of clocks, timing units, clock dividers, and even timer dividers.

To better understand the different settings we present a quick overview of the basic clock system of the used MSP430 platform. The MSP430 has three different clock signals called ACLK, SCLK, and MCLK. Each one of them can be configered to be clocked from different sources, like the crystal 1 (XT1), crystal 2(XT2), or the internal DCO. Additionally, each clock signal can use a divider that divides its clock source by a factor of 1, 2, 4, or 8 respectively. The MCLK is the master clock and drives the CPU. Thus, this signal can not be used to feed the two timer units. But either ACLK or SCLK can be used as a source for each of the timer units. Additionally, each timer has again a divider that can be set to divide the input signal by 1, 2, 4, or 8. In our configuration, both XT1 and XT2 are a 8MHz crystal. XT1 is the source for the ACLK clock signal, and XT2 for SCLK. Additionally, Timer A is fed by ACLK and Timer B by SCLK. Thus, using the different dividers, different clock

---

[2] The Maxim DS4026 TCXO datasheet also gives numbers for the drive and oscillatory circuit. They are 3mA and 2mA respectively (typical). This gives a grand total of 21.4mW at 3.3V.

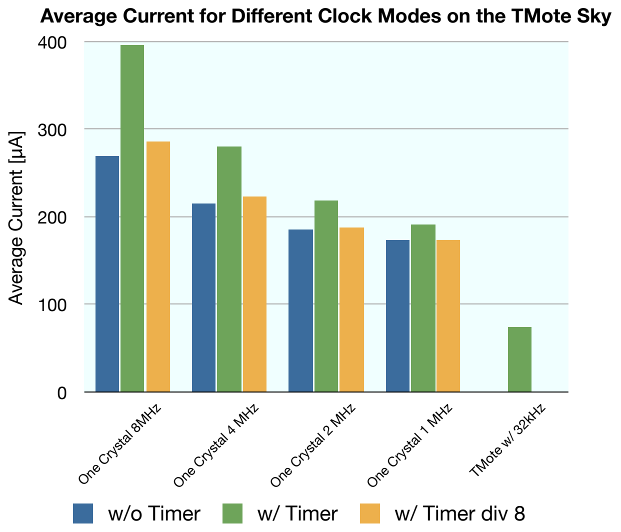

Figure 7: These are the typical current consumptions of the clock and timer subsystem of the MSP430. The clock source was an 8MHz crystal, and the different clock speeds were achieved by dividing that signal using an internal clock divider. As comparison, we also measured the MSP430 power consumption when it uses a 32kHz crystal, one timer active, CPU turned off.

speeds can be achieved at the input to the timers, as well as within the timer.

Using the different divider combinations, and by turning on and off the timer units, we were able to measure the power consumption of one oscillator using different clock dividers, as well as the power consumption of the oscillator including a timer unit. Figure 7 shows a summary of the result. For each possible clock divider that results in a 8 MHz, 4 MHz, 2 MHz, and 1 MHz clock signal, we made measurements with the timer unit disabled, enabled, and enabled with an additional division by 8. Additionally, we measured the power consumption of the TMote platform when the CPU is at sleep, and one timer unit is clocked from a 32kHz crystal. This will help us to identify the impact to the power budget when using a high frequency crystal, instead of a low frequency one.

From the obtained measurements, we can establish some general and intuitive rules:

1. The higher a clock divider, the less power the clocking subsystem consumes.
2. Enabling a timer unit adds a considerable power overhead at high speeds.
3. Dividing at the timer unit is less efficient than dividing the clock signal ahead of the timer.
4. A high frequency clock system has a considerable impact on the power budget and should thus be avoided if high granularity time is not needed.

Using these simple rules, we can come up with new algorithms and techniques to improve the power efficiency of the XCXT and make it into a Smart Timer Unit. In the next section, we will analyze some of these possibilities and show how they can be implemented on our current prototype. But before that, let's look again at the power consumption we measured for our XCXT prototype. We measured the consumption while running the compensation phase, where the two clock sources run at 1 MHz each, at 1.27mW. From our other measurements we know that one crystal with a divider by 8 and a timer unit consumes 191 μW. Thus, the two crystals with their timers alone consume a total of 1.15mW at 3V. This confirms our suspicion that the crystals, and not our compensation algorithm or the microprocessor, account for the majority of the power consumption of the XCXT.

5. THE SMART TIMER UNIT

As we showed in the last section, the oscillator circuitry, high speed crystals, and the timer unit consume most of the power of the XCXT. Thus, duty cycling one of the two crystals of the XCXT seems to be an efficient method in order to cut down on the energy consumption. Indeed, it is not necessary to have both timers on all the time. For example, if the XCXT experiences a constant environment temperature, the measured drift is always the same and doesn't add any new information for the compensation phase. This is similar in the event where the environment temperature changes only slowly. By this, we mean where the environment doesn't change more than $\pm 1^\circ$C per minute. In such an environment, the second crystal could be turned off for 58 seconds. Then, the crystal is turned on for 2 seconds. The first second is there to let the crystal stabilize itself. In general, this should not take more than a couple of tens of milliseconds. In the second second we perform the drift measurement and thus compensate for any change in temperature. At the same time, we saved approximately 48% of the total power, since one of the crystals is turned off most of the time.

A second improvement can be done by using one slow crystal, like a standard low frequency 32kHz crystal as it is found on many real time clocks and embedded systems, and one high frequency one, like the ones we use in the XCXT. The advantage in a duty-cycled system is evident, since the 32 kHz crystal can be used to bridge the time while the node is asleep, and doesn't use a high granularity clock. Once the microprocessor wakes up, the fast clock is turned on and provides a high granularity clock source to measure precise time. At the same time, the two crystals compensate each others drift using the differential drift algorithms. The problem is that the current XCXT algorithm does not work in such a scenario since it is based on the fact that both crystals have the same speed and thus time resolution (minus their drift, which we calculate in the algorithm). Nevertheless, we developed a technique that extends the XCXT algorithm and allows the usage of a slow and a fast crystal. We will now explain the basics behind the extension, although we cannot provide a prototype yet, since there are still some technical problems left we need to solve. We will mention them at the end of this section.

The main idea behind the extension is to interpolate the slow frequency and measure its phase by using the high frequency clock signal. Figure 8 illustrates this concept. The blue line (solid) represents the fast clock signal, and the red line (dashed) is the slower clock. Let's assume that the sampling interval is f_s, timer A uses the slow clock signal, timer B the fast, and the two timers are started at the exact same time. At the sampling time t_s we read both timer values as $C_A(t_s) = C_{A0}$ and $C_B(t_s) = C_{B0}$. Now, in order to

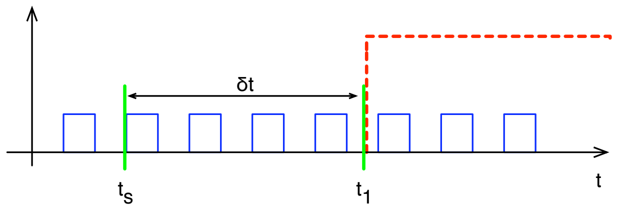

Figure 8: This is the timeline of the smart timer unit. We can find the phase of the slower timer (red, dashed) since the time instance t_s by counting the number of ticks of the fast clock (blue, solid)

find the phase of the timer A we wait until timer A's clock signal rises again. Let's call this time t_1. We know that $C_A(t_1) = C_{A0} + 1$ and $C_B(t_1) = C_{B0} + \lfloor \delta t_1 \cdot f_B \rfloor$ where the floor operator comes from the digital nature of the timers. Thus, we can calculate the phase of timer A as

$$\phi_A(C_B) = \frac{f_B/f_A - (C_B(t_1) - C_B(t_s))}{f_B/f_A} = 1 - \frac{\lfloor \delta t \cdot f_B \rfloor}{f_B/f_A}$$

This phase information can now be used to calculate the drift between the two crystals that source the timer A and B, and the same differential drift algorithm as for the XCXT can be applied.

There are still some technical problems left that need to be solved, in order to implement the smart timer unit in hardware:

- Commercial available 32kHz crystals are all tuning-fork based. A tuning-fork crystal has a quadratic temperature vs. frequency drift curve. Combining this with a cubic curve from a high frequency AT-cut crystal does not yield a calibration function and can thus not be used for calibration, i.e., there is no unique mapping from $C_{12} \rightarrow C_{10}$.

- The time between the high frequency timer interrupts, and the time the slow timer signal rises (δt in Figure 8) can be very short. We will see if the timer hardware is fast enough to rearm the input capture of the fast timer such that an earlier result doesn't get overwritten. A possible solution is to use multiple timer capture registers.

6. CONCLUSION

We discussed the problems arising in low-power, duty-cycled systems if they need precise time estimation. We showed that current technology is not adapted for low-power embedded systems since they are too power hungry. We described the XCXT, a new way of compensating a pair of crystals and thus achieved a $\pm 1.2ppm$ precision over a temperature range of -10 to 60°C. We showed that we can achieve this precision while using only 1.27mW. We then benchmarked the XCXT to pinpoint the components that consume most of the energy. We found that the two 8MHz crystals and their timer units consume more than 90% of the XCXT's power budget. Therefore, we discussed two different ways on improving the algorithm such that we can minimize the crystals power consumption. The first way is to simply duty cycle one of the crystals. This already gains us 48%. An even better way is to use two crystals, one fast, and one slow. The fast crystal can be used if high granularity time is needed, and the slow while the system is in sleep. Still, both crystals are used to compensate each other's drift and thus provide a highly stable timer unit.

Our next XCXT prototype will investigate our proposed algorithms further, and we will explore ways of integrating the XCXT into embedded systems. We expect that if we combine the XCXT with a good time synchronization protocol, that we can even achieve an online calibration and thus remove the whole calibration phase itself. This is possible if the embedded system has the capability to communicate with a reference clock system, such as an embedded system with access to a GPS receiver.

7. ACKNOWLEDGMENT

This material is supported in part by the U.S. ARL and the U.K. MOD under Agreement Number W911NF-06-3-0001, by the NSF under award CNS-0614853, and by the Center for Embedded Networked Sensing at UCLA. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the listed funding agencies. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

8. REFERENCES

[1] Bloch, M., Meirs, M., and Ho, J. The microcomputer compensated crystal oscillator (mcxo). *Frequency Control, 1989., Proceedings of the 43rd Annual Symposium on* (31 May-2 Jun 1989), 16–19.

[2] Dutta, P., Culler, D., and Shenker, S. Procrastination might lead to a longer and more useful life. *HotNets-VI* (2007).

[3] Maxim. DS4026-10MHz to 51.84MHz TCXO, 2007.

[4] Moteiv. Tmote sky datasheet.

[5] Satou, K., and Hara, T. Temperature detector and a temperature compensated oscillator using the temperature detector, May 1993. US Patent 5,214,668.

[6] Schmid, T., Charbiwala, Z., Friedman, J., Cho, Y. H., and Srivastava, M. B. Exploiting manufacturing variations for compensating environment-induced clock drift in time synchronization. *Sigmetrics* (2008).

[7] Schodowski, S. Resonator self-temperature-sensing using a dual-harmonic-modecrystal oscillator. *Proceedings of the 43rd Annual Symposium on Frequency Control 1989*, 2–7.

[8] Ye, W., Silva, F., and Heidemann, J. Ultra-low duty cycle MAC with scheduled channel polling. *SenSys* (2006), 321–334.

[9] S. Yoon and M. L. Sichitiu Power Consumption Models for Wireless Sensor Networks MAC Protocols *CACC Technical Report, NCSU TR-07/01.*