

The True Cost of Accurate Time

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

Young H. Cho
The Computer Network Division
University of Southern California
Information Sciences Institute
youngcho@isi.edu

1. INTRODUCTION

The scientific community has long envisioned vast networks of sensors each attached to a small battery operated micro-processor replete with a radio communications transceiver. For these scientists, each of the sensor terminals is called a node and the entirety of nodes is termed a Wireless Sensor Network (WSN). WSN's have proven effective in a plethora of application domains spanning climate science [2] to military operations [10]. In applications where remote long-term operation, broad geographic coverage, low-cost (e.g. expendable hardware), and rapid deployment are vital, the WSN offers a practical and efficient solution. To achieve remote distributed sensing and computation the WSN must achieve synchrony among its component nodes.

But the nature of this problem is more fundamental. Wireless communication among terminals of any creed carries with it inherent synchrony requirements. Data-rate, channel capacity, and communication range are ultimately tied back to synchrony through preamble length, chip-rate, and ISI. Further, the application objectives layer additional synchrony requirements. Even the primordial one-to-one link between a piece of factory automation equipment (such as a bottle filler, conveyor, or robotic arm) and its controller becomes substantially more involved when the link to that controller is wireless (as to a laptop or PDA carried by a technician or plant manager). Despite the substantial performance benefits from increased factory floor space (no cable path planning reserves, restrictions in foot-paths from cable-in-the-way, equipment cabinets to house routers and repeaters, etc) and cost benefits from reduced installation, maintenance, and refitting expenditures, wireless deployments in mass production facilities remain rather limited. In large part, this reflects the industry's lack of confidence in wireless real-time control.

A recently introduced standard, IEEE 1588 [5], has experienced wide spread adoption by industrial automation applications on wired Ethernet networks. Although 1588 is ideal for high-performance real-time control applications, as it specifies precision below 100ns, it has yet to be applied successfully over wireless links – let alone the *low-power* wireless links fundamental to WSN and mobile remote control.

1.1 Timing Guard Bands

The single biggest impediment to a node's battery-powered lifetime is the energy spent during radio communication, and secondary to that, the time spent in its awake (as op-



Figure 1: Minimizing guard bands reduces wasted active time and increases channel capacity.

posed to its low-power shutdown sleep state). Reduce these times, and lifetime improves substantially. However, as soon as nodes in the network begin sleeping and are correspondingly offline, other nodes in the network that are still awake can no longer use them as a communications hub to route sensor data back to a command-and-control station (referred to as a gateway node). To optimize sleep time and network performance simultaneously all of the nodes must synchronize their internal clocks and sleep and wake at the same time¹.

The ratio between a node's total lifetime and the time spent in its awake state is its *application duty cycle*. Many monitoring and control applications allow exceptionally low duty cycles. Outdoor temperature, for example, may not need to be sampled faster than every two minutes [2] and rare-event detection systems (ex. [10]) may even allow nodes near-indefinite sleep sans event occurrence.

Though the node is shut down during sleep, a timer must still remain active to maintain synchrony with the rest of the network. This maintains the synchronized wake-up capability required by the network to resume operation. Yet, as duty cycle decreases, sleep power dominates average power consumption and timer power, in turn, comprises the bulk of the sleep power. Consequently, minimizing that timer's power consumption is key to assuring low average power consumption.

Conversely, the longer a node stays asleep, the more drift accumulates in the timer from clock frequency errors. To insure that nodes attempt communication only when other nodes are available for correspondence, the node must transmit a communication-request signal long-enough to over-

¹We are aware that this statement is somewhat of a generality and numerous works on WSN scheduling exist, but our axiom to follow - that better synchrony yields better lifetime - still holds in these cases.

come the uncertainty between their local notion of time and that of their worst network peer. This extra time required to ensure that connections can be reestablished is the *temporal guard band* (Figure 1). An accurate clock can save energy by reducing guard times, or in other words, for equal guard times, *a node with a more accurate clock can sleep longer*. However, we must ensure that this accuracy does not come at a substantial energy cost or the gains made through coordinated time will disappear.

1.2 Time Synchronization Protocols

Time synchronization protocols can significantly increase the accuracy and stability of local clocks. In [6], a linear regression applied over multiple clock offsets measurements taken between a pair of nodes was used to achieve an accuracy of $\pm 0.04ppm$ over short time intervals.

In general, researchers benchmark time synchronization protocols by the achieved accuracy and by how many messages these protocols transmit/receive [3]. Although these are illustrative metrics, the performance of a time synchronization protocol depends greatly on the local clock in residence. A simple ring oscillator, as can be found on the popular TI MSP430 MCU, is unsuitable due to its extreme instability with respect to both ambient temperature and supplied voltage (on the order of $\pm 10\%$).

Intuitively, the research of low power clocks is orthogonal to the investigation of low power time synchronization protocols. Nevertheless, there is an interesting intersection, where local crystals get calibrated by the use of high precision time synchronization, and thus increase the time between resynchronization attempts through improved stability. However, in this paper we will first concentrate on the development of a low-power, high accuracy, fine granularity local clock source – leaving the interplay between local clock and synchronization protocol to future research.

2. BACKGROUND

2.1 Frequency, Resolution, and Power Consumption

To measure its own clock frequency, a digital system would typically feed its clock to a counter and periodically compare its output with that of another counter clocked by an accurate reference. The digital nature of the counting process, however, limits the resolution of the measurement due to temporal quantization. Thus, to obtain a high resolution frequency measurement, one must either use a high nominal frequency or sample the counter over long intervals.

More formally, the resolution error ϵ of a frequency measurement is bounded by

$$\epsilon \geq \frac{1}{F_0 \cdot \tau},$$

where, F_0 is the nominal frequency of the clock and τ is the interval over which the counter is sampled. In other words, if a system wants to know the frequency error with a resolution of ϵ_0 , then it needs to wait for at least

$$\tau \geq \frac{1}{F_0 \cdot \epsilon_0}$$

Table 1: Balance

Device	Granularity	Stability	Power
Tuning Fork XO	Coarse	100ppm	$< 50\mu W$
Fox F254 32k Osc	Coarse	100ppm	$56\mu W$
AT-Cut Quartz XO	Fine	25ppm	$200\mu W$
DS32KHz 32K TCXO	Coarse	7.5ppm	$750\mu W$
DS3232 32KHz TCXO	Coarse	2ppm	$1mW$
DS4026 10MHz TCXO	Fine	1ppm	$21mW$
ACHL 10MHz Osc	Fine	30ppm	$13.8mW$
Crystek C3392 10MHz	Fine	30ppm	$10mW$
8MHz XCXT	Fine	1ppm	$1.4mW$
Smart Timer Unit	Fine	1ppm	$\sim 300\mu W$

seconds. Numerically, a nominal clock frequency of 1MHz estimated to a resolution of 0.04ppm will require at least 25 seconds of contiguous counting. Power consumption is also related to nominal frequency through the linear relationship:

$$P = P_0 + (C \cdot V^2 \cdot F_0), \quad (1)$$

where P_0 is the power lost due to leakage, C is the effective load capacitance and V is the driving voltage. Further, as energy consumed is $P \cdot \tau$, there exists a strict tradeoff between the energy consumption and the achievable timing resolution. Or in other words, if you lower the nominal frequency, you lower the power consumption, but at the same time decrease the achievable time resolution.

2.2 Frequency and Temperature

Assume that the frequency of a clock as a function of temperature is denoted by the function $f(T)$. Its nominal frequency is designated as F_0 , and thus the frequency error is $\Delta f(T) = f(T) - F_0$. The normalized frequency error is calculated as

$$\delta f(T) = \frac{\Delta f(T)}{F_0}$$

and is usually expressed in the unitless quantity called *ppm*, or Parts Per Million, computed as

$$\delta f(T) = \frac{f(T) - F_0}{F_0} \times 10^6$$

The multiplication by 10^6 is a measure of convenience since the frequency error of most commercially available clock sources ranges from tens to hundreds of *ppm*.

Thermal variation is the single most significant contributor to clock frequency error. Thus, changes in the ambient temperature are reflected in the frequency of a node’s clock. The most common way to counterbalance this effect in quartz-crystal based oscillators is to measure the temperature and adjust the crystal oscillator (by changing its load capacitance) using a set of predetermined corrections. Commercial Temperature Compensated Crystal Oscillators (TCXO) embody this approach, but adding thermal measurement and analysis logic not only increases the production cost, but power consumption as well. TCXO’s are often unfeasible for the strict low power requirements of WSN’s. In Table 1, a representative set of oscillators (including several TCXO’s) are compared in terms of stability and power.

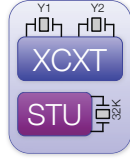


Figure 2: Block diagram of the Smart Timer Unit. The XCXT uses two Crystal Oscillators (Y1 and Y2) to achieve high accuracy and fine granularity, whereas the STU uses a 32kHz clock to keep time during sleep periods.

3. DIFFERENTIAL FREQUENCY ERROR AND THE CRYSTAL COMPENSATED CRYSTAL BASED TIMER (XCXT)

We introduced differential frequency error and the XCXT in [8, 9]. For completeness, we will summarize these works here. Every crystal oscillator has a unique frequency error versus temperature curve that depends strongly upon the specifics of its manufacture (e.g. the cut angle of its crystal blank). The greater the difference between the cut angles of two different crystals, the greater the difference in their behavior with respect to temperature will appear to be. Manufacturers necessarily control the cut angle variance as much as possible to minimize product performance variation. In general, that angle can be controlled to a precision of up to 1'.

Differential frequency error exploits the fact that we may choose to operate two crystals, with different cut angles (from implicit manufacturing – variance, or by design – AT-cut vs. tuning fork vs. SC-cut, etc.), simultaneously. If designed properly, the differential behavior of the two oscillators with respect to temperature will allow the temperature and the frequency offset to be precisely determined.

More specifically, during manufacturing, one of the individual crystals (f_1 , f_2) of the XCXT is calibrated against a reference frequency source at different temperatures resulting in frequency error curve $f_1(T)$. Then the differential frequency error term

$$\delta f_{12}(T) = \delta f_1(T) - \delta f_2(T)$$

is calibrated over temperature. If T vs δf_{12} is a bijection, then we know that there exists a unique mapping from differential frequency error to temperature, and thus we can construct a

$$\delta f_{12} \rightarrow \delta f_1$$

lookup table or calibration curve.

At runtime, the XCXT measures δf_{12} and can thus estimate δf_1 using its predetermined calibration curve. Compensating by this amount, we can achieve the same stability as the factory reference. Our initial prototype was based on a popular sensor network platform, the TMote Sky [7] which has, at its core, a TI MSP430F1611 MCU. The prototype consumed a total of 1.4mW on average, from which about 1.2mW came from the power consumption of the counters and the crystals alone [9].

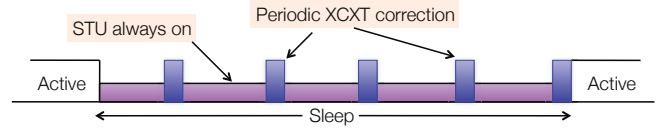


Figure 3: The Smart Timer Unit exploits the higher stability of the XCXT to compensate the 32kHz clock, and thus achieving a high precision, low power clock during sleep times.

Already a 5x improvement over alternative commercially available temperature compensated oscillators in the same frequency range, it still was insufficient to reach the realm of sensor network sleep powers (tens of μW). Because the timers and oscillators are an integral part of the XCXT, lowering their power consumption was not possible.

In this work we propose the Smart Timer Unit (STU), a new system and algorithmic approach which proffers further dramatic power reduction beyond prior XCXT achievements. The remainder of this paper describes the architecture of the STU and its design process.

4. THE SMART TIMER UNIT (STU)

The main consumers in the XCXT are the oscillator circuits and counters. Thus, lowering the total power consumption requires lowering the power of these two parts of the design. From Equation 1, we see that an effective way of doing this is lowering the frequency. However, lowering the frequency also reduces time resolution. This is counter productive, since it will lead to higher guard bands than necessary. Instead, some mechanism needs to be sought that provides high temporal resolution at low power.

One solution comes in the form of a combination of a fast, and a slow clock in the same device. Figure 2 depicts this in a small block diagram. It shows that while the XCXT uses two high frequency crystals, the STU has in addition a 32kHz clock that can be used during sleep time. The trick is that just before going to sleep, the XCXT is used to calibrate the 32kHz clock, such that we can turn off the XCXT, but maintain accurate time while consuming little power during sleep. This recalibration happens independent of the MCU sleep schedule. Figure 3 visualizes this, where at the far left, and right, the MCU is active, and in-between, the XCXT is periodically turned on and off in order to compensate the 32kHz clock for changes in temperature. Note that shortly before the MCU wakes up, the XCXT is turned on in order to exploit the higher granularity of the fast crystals available to the MCU.

At first, it was thought that a slow 32kHz and a fast ($> 1MHz$) clock within the XCXT itself would suffice. However, because of inherent crystal resonance properties, there are no combinations of crystals with frequencies two decades apart such that the T vs δf_{12} curve is bijective. This led us to the design choice of implementing the STU with three inexpensive crystals.

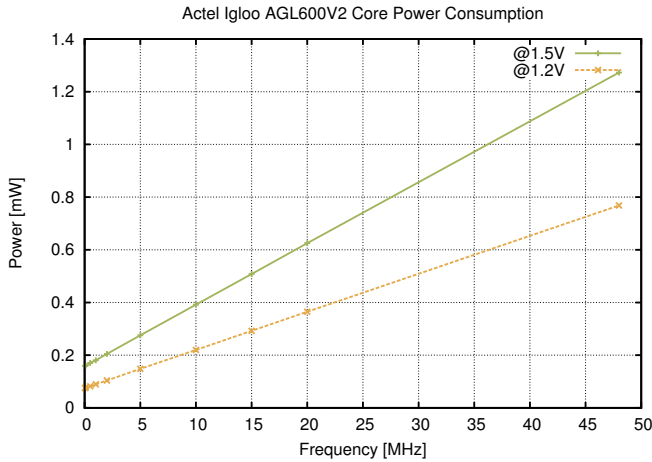


Figure 4: Power consumption of the Actel Igloo AGL600V2 core having a simple 32-bit counter instantiated. We can clearly see the linear power behavior as expected from theory. Note that this FPGA is slightly bigger than the one used in our prototype, and thus consumes about $30\mu W$ more static idle power.

4.1 Prototype Design

The TI MSP430F1611 was a good fit for a first prototype. The MCU has two flexible timer units and the possibility to connect them to two individual crystals. Additionally, it has very low power consumption during sleep ($< 10\mu A$). However, for faster clocks, the 16-bit wide counters of the timer units are too small and overflow far too often. This leads to MCU wake-ups just to treat the overflows, increasing the average power consumption substantially. In addition, it inhibits the possibility of having other tasks running on the MCU directly, like sensing, control applications, or communication. For this reason, we based the next version of the STU on an FPGA. This allows us to implement the timer features we need and optimize the design for low power. We retained low power MCU for rapid prototyping and algorithm testing.

The FPGA needs to work at a very low-power active mode, have a small form factor, but at the same time be cheap in order to be viable for a large deployment. One such candidate is the Actel IGLOO Family [1] of FPGAs. They come in a variety of configurations, ranging from a 15k gate, $12\mu W$ static idle to a 1M gates, $59\mu W$ static idle power consumption, ARM-Enabled chip. For our prototype, we chose a mid-range sized FPGA, the AGL125V2, which is a good compromise between number of system gates (125k), to power consumption ($20\mu W$ static idle). In addition, it can operate at 1.2V, which allows an even smaller power consumption during run-time (see Figure 4).

The only drawback of an FPGA is the longer turn-around time for prototyping compared to a MCU. Therefore, we decided to keep in addition to the FPGA a TI MSP430F1611 which can handle different parts of the algorithm, and provide a quick prototyping platform. Figure 5 depicts the interconnections between the FPGA and the TI MSP430. In addition to the two large chips, the STU needs three crystal oscillators, two high frequency, and one low frequency one.

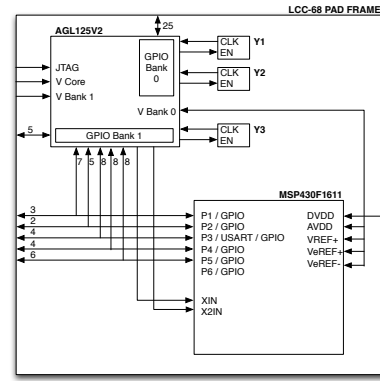


Figure 5: STU prototype board in the LCC-68 form factor. It contains 3 oscillators (two fast, one 32kHz), one TI MSP430F1611 MCU, and an Actel Igloo AGL125V2 Low Power FPGA. The board will have a size of 24x24mm.

The problem with the TI MSP430, as with many MCUs with similar capabilities, is that it only provides two crystal inputs. Therefore, the FPGA will work as an intermediary that can dynamically provide the different clock signals to the MSP430, two at a time. In addition, the FPGA can implement timers and counters by itself, and thus eliminate the restrictions of the 16-bit timers found on the MSP430.

The decision of the form factor chosen for the STU prototype was inspired by [4]. The industry-standard LCC-68 (68 pin leadless chip carrier) provides an easy transition from prototyping to production systems, because the modules can be socketed, hand-soldered, or machine-assembled, without loosing any space for a connector, since the side of the board would else be unused. The only drawback would be that the LCC-68 form factor can not be stacked, though that is a minor concern for the current prototype.

5. PRELIMINARY RESULTS

Parallel to our prototype design, we also began to evaluate the effects of the combination of a fast (8 MHz) XCXT with a slow (32 KHz) oscillator. Intuitively, one could view the periodic re-calibration (shown in Figure 3) of the slow clock by the high accuracy XCXT as a two-staged duty cycled system itself. To analyze the overall power consumption of the STU, one would need to compute the duty cycle of this subsystem. Since the frequency of the slow clock drifts primarily due to changes in temperature, it would be natural to presume that the duty cycle and, in effect, the time interval between compensations depends on the change in the environmental temperature. Thus, any meaningful evaluation of STU performance requires to be done on real temperature data.

Figure 6 illustrates the result of a simulation setup of the STU using a 3 year temperature trace from a wildlife reserve in California. Though this setting does not represent the harshest of temperature variations, it emulates a reasonable set of outdoor conditions suitable for early evaluation. The top trace depicts the temperature variation and clearly shows seasonal changes (and less clearly, diurnal changes).

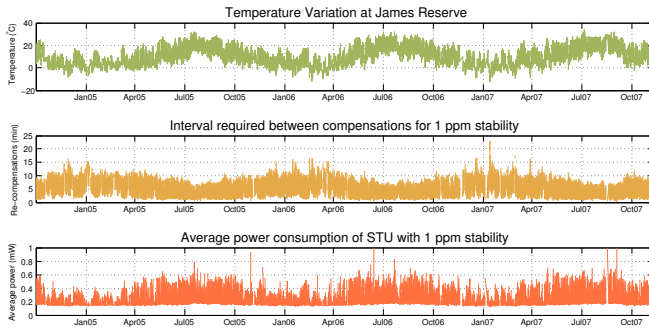


Figure 6: In a simulation of 3 years of temperature data, the Smart Timer Unit achieves $< 400\mu W$.

The middle trace is an estimate of the re-compensation interval based on the rate of the change of temperature and the accuracy of the XCXT at that temperature. (The re-compensation attempts to maintain the stability of the slow clock to below 1 *ppm*.) This estimate is fairly optimistic since it assumes knowledge of future temperature gradients. In an actual implementation, the XCXT would use its δf_{12} sensing as a proxy to track changes in environmental temperature. Using measurements gleaned from development boards, a state model of the STU power consumption was constructed, utilizing which the bottom trace shows the average power consumption for this temperature trace.

To show that duty-cycling fast clocks has a vast effect on the power consumption, we instantiated two counters on an Igloo FPGA. The first counter was connected to a low frequency oscillator, whereas the second one was connected to a 48 MHz clock. In regular intervals, based on the slow counter, we enabled and disabled the fast oscillator and measured the power consumption of the FPGA core. Figure 7 shows the result. We can clearly see that the average power consumption drops drastically while with the STU concept, a stable clock can still be achieved.

6. CONCLUSION

The true cost of accurate time has so far often been ignored in the design of low-power embedded systems. The development of the STU will for the first time enable true low-power time synchronization protocols by exploiting the fact that we don't always need high precision, and can thus duty-cycle high frequency clocks. By adding a low-power FPGA to our design, we created a flexible platform that allows us to implement, and test new algorithms for timer units that could be included in future low-power microprocessors directly.

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.

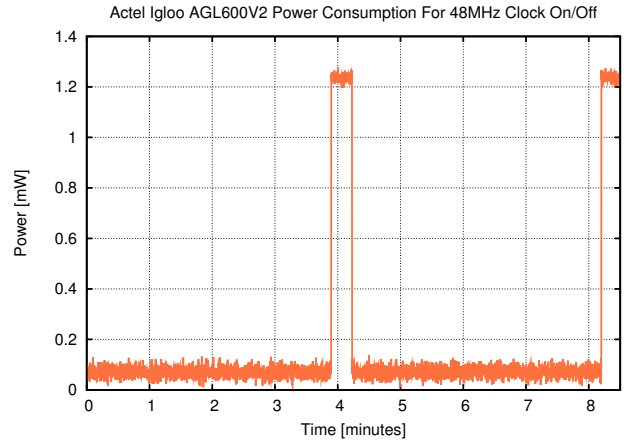


Figure 7: Power consumption of the Igloo AGL600V2 while duty-cycling the fast 48MHz clock.

7. REFERENCES

- [1] ACTEL. Igloo Datasheets. <http://www.actel.com/products/IGLOO/default.aspx>. Oct. 2008
- [2] BARRENETXEA, G., INGELREST, F., SCHAEFER, G., AND VETTERLI, M. The Hitchhiker's Guide to Successful Wireless Sensor Network Deployments. *SenSys* (2008).
- [3] DAI, H., AND HAN, R. TSync: a lightweight bidirectional time synchronization service for wireless sensor networks. *ACM SIGMOBILE*, (2004).
- [4] DUTTA, P., TANEJA, J., JEONG, J., JIANG, X., AND CULLER, D. A Building Block Approach to Sensor Network Systems. In *SenSys* (2008).
- [5] EIDSON, J. Measurement, Control, and Communication Using IEEE 1588. Springer, 2006.
- [6] MARÓTI, M., KUSY, B., SIMON, G., AND LÉDECZI, Á. The flooding time synchronization protocol. *SenSys* (2004).
- [7] MOTEIV. Tmote sky datasheet.
- [8] 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).
- [9] SCHMID, T., FRIEDMAN, J., CHARBIWALA, Z., CHO, Y., AND SRIVASTAVA, M. Low-power high-accuracy timing systems for efficient duty cycling. In *ISLPED* (2008).
- [10] VOLGYESI, P., BALOGH, G., NADAS, A., NASH, C., AND LEDECZI, A. Shooter Localization and Weapon Classification with Soldier-Wearable Networked Sensors. *Mobisys* (2007).