# DEMO: The Need for Wireless Clock Drift Estimation and Its Acceleration on a Heterogeneous Sensor Node

Andreas Engel, Andreas Koch
TU Darmstadt
Embedded Systems and Applications Group
Hochschulstraße 10, 64289 Darmstadt, Germany
engel,koch@esa.cs.tu-darmstadt.de

*Abstract*—**Time synchronization is an essential service for nearly all Wireless Sensor Network (WSN) applications as it permits simultaneous sensor sampling and tightly scheduled communication. With increasing sampling rates, as required, e.g., for monitoring vibration or acoustic phenomena, a synchronization accuracy of a few microseconds is required. Without proper drift compensation, clocks have to be synchronized once per second to achieve this accuracy. Drift estimation allows less frequent synchronization (and thus saves wireless transmission energy), but does tax the (possibly already constrained) node processor in terms of memory and compute overhead. The demo shows both the effect as well as the extent of the clock drift while attempting to simultaneously blink node LEDs in a distributed wireless network. The exact synchronization error for different configurations of the synchronization protocol is analyzed by an oscilloscope. In addition, the speedup of the clock drift compensation computation using the FPGA-based Hardware-Accelerated Low Power Mote (HaLoMote) is demonstrated.**

## I. Scope and Significance

Time synchronization is required in WSN applications mainly for two reasons. First, data sampled from spatially distributed sensors cannot be properly interpreted without knowledge of the exact sampling time. The acceptable uncertainty is typically a small percentage of the application's sampling period, which may range from several seconds in environmental monitoring down to several milliseconds in vibration-based structural health monitoring [2] (or even shorter in acoustic localization applications). Thus, synchronization protocols with an accuracy of a few microseconds are required for the more demanding applications.

A second reason for precise synchronization derives from the large power consumption of the radio transceiver idly waiting for incoming messages. If sender and receiver are synchronized, the radio protocol can define short periods of time in which transmissions can be initiated and received, thus limiting idle listening.

The local time at a sensor node is represented as a timestamp, which is the value of an oscillator driven counter with a certain oscillation frequency and certain start-up time. Even if two nodes run at the same nominal frequency, the temperature and voltage dependency of the oscillators result in small relative frequency deviations, typically in the range of a few part per million (ppm). Manufacturers specify the maximum frequency uncertainty over the whole operating temperature range with even higher values (e.g., ±40 ppm for the TI CC2530). Without explicit drift compensation, the timestamps used for the time synchronization have to be exchanged every few seconds to achieve a synchronization accuracy of a few microseconds.

To estimate the clock drift between two nodes, a list of subsequently exchanged timestamp pairs is collected to compute a linear regression. The slope and the offset resulting from this regression can be used for a precise time synchronization achieving a microsecond accuracy with a 20 s timestamp exchange interval. The Flooding Time Synchronization Procotol (FTSP) [3] is one of the most popular wireless synchronization protocols relying on linear regression based clock drift compensation. A performance-optimized linear regression implementation is presented in [1].

The main purpose of this demonstration is to show the

- effects of the clock drift on the synchronization accuracy of a small WSN,
- improvement of the synchronization accuracy by clock drift estimation,
- temperature dependency of the clock drift,
- static offset between the transmission and the reception of the IEEE 802.15.4 Start of Frame Delimiter (SFD) as captured by the automated MAC timestamping,
- acceleration of the linear regression computation by the Field Programmable Gate Array (FPGA) of the heterogeneous HaLoMote

## II. Demonstration Setup and Procedure

As shown in Figure 1, the demonstration is based on a small network of five commercial off-the-shelf (COTS) WSN motes (TI CC2531 USB dongles) and an additional HaLoMote organized in a star topology. All nodes toggle an LED at 1 Hz and the toggle events are displayed on an oscilloscope. One of these nodes acts as time reference. All other nodes are synchronized to the reference with different configurations for the synchronization protocol (i.e., with or without drift compensation and differently sized regression tables). The resulting synchronization accuracy can be observed at the

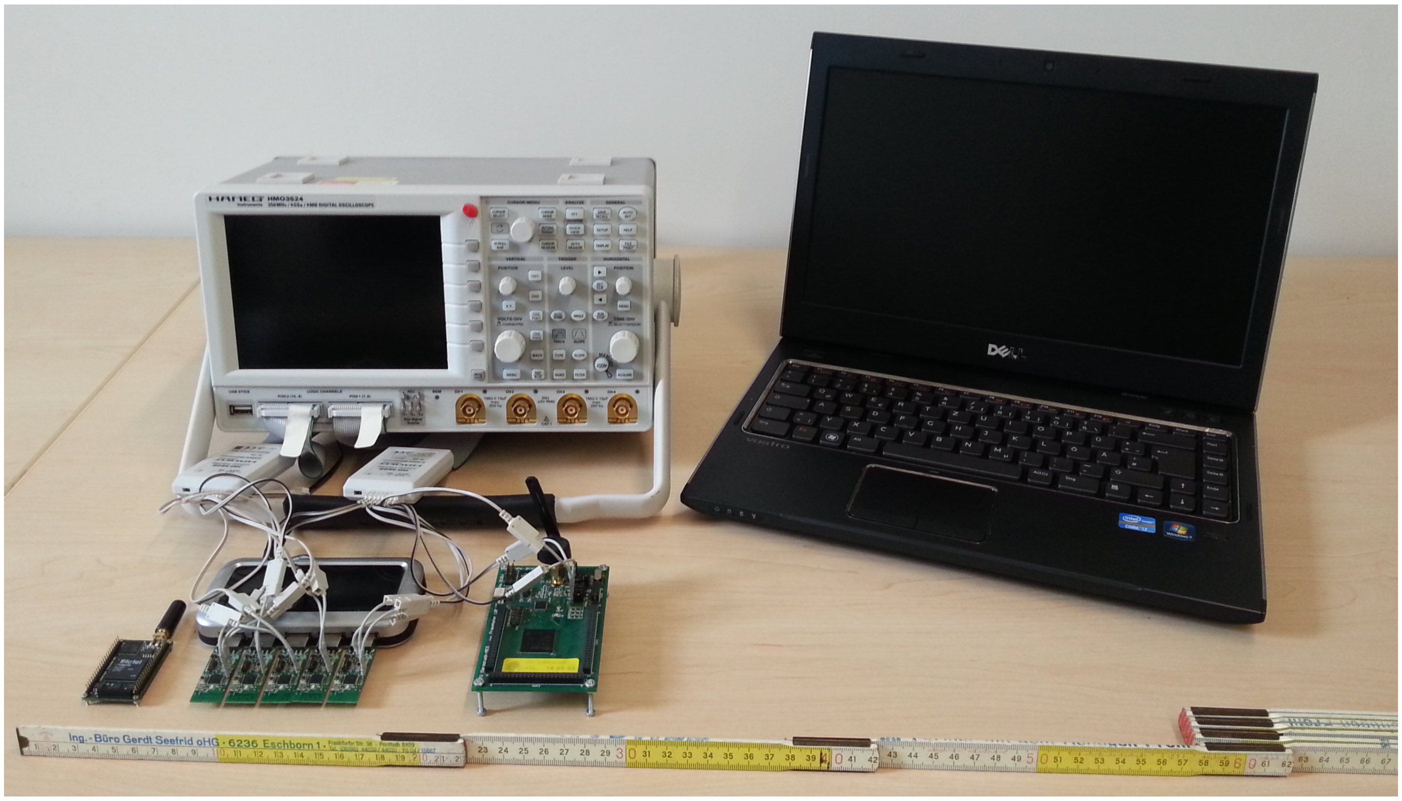

Fig. 1. Demonstrator setup consisting of a network of COTS WSN motes, a HaLoMote, an oscilloscope and an optional laptop

oscilloscope. By warming up individual processors (e.g., by touching the RF-SoC package), the temperature-dependency of the synchronization accuracy can be observed.

In addition to the achieved synchronization accuracy, more detailed timing information is displayed on the oscilloscope. This includes the delay between SFD transmission and reception, as well as the time required to process the clock drift compensation by one of the COTS software processors and the heterogeneous HaLoMote. The latest implementation of the HaLoMote is available as additional exhibit as shown in Figure 2.

About 15 min are required to setup (and demount) the demonstrator. It occupies about 60 cm × 40 cm of table area and requires two power outlets for the WSN and the oscilloscope. If an additional table area of 60 cm × 30 cm is available, a laptop could be provided to discuss implementation details. WiFi access is not required.

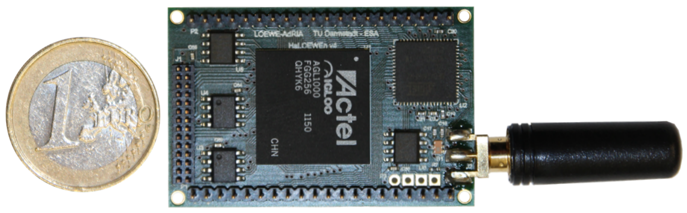

Fig. 2. Current implementation of the HaLoMote architecture

## III. Demonstrated Results

The delay between the transmission and the reception of an SFD, as it is registered by the automatic MAC timestamping, amounts to 3.51 µs with a standard deviation of only 40 ns, as shown in Figure 3. As the propagation delay is smaller than 30 ns, the major part of the SFD delay must be caused by the

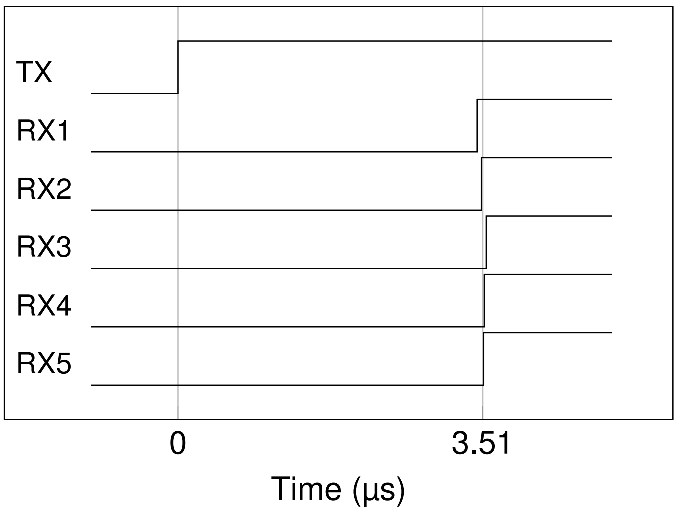

Fig. 3. Delay between SFD transmission and reception of the CC2531 automated MAC timestamping

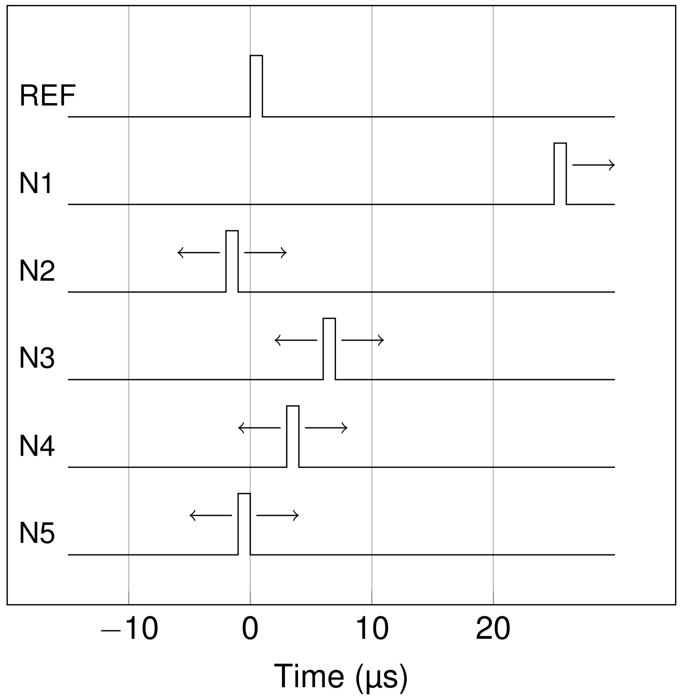

Fig. 4. Accuracy of synchronized LED toggling. N1 does not compensate the clock drift and slowly drifts away from the reference. N2 to N5 compensate the clock drift and are kept close to the reference.

capturing hardware.

Looking at the achievable synchronization accuracy shown in Figure 4, the LED toggle events of all nodes with activated clock drift compensation jump around the reference event in a range of a few microseconds. In contrast, the node without clock drift compensation (N1 in Figure 4) slightly drifts away from the reference. Its inaccuracy reaches up to several tens of microseconds, until it is reset by a newly exchanged synchronization point once per 10 s. The speed of the clock drift clearly changes when the corresponding processor is heated up.

Finally, the time spent for calculating the linear regression required for the clock drift compensation is shown to be 320 µs for the software processor of the COTS WSN mote, but just 90 µs for the Hardware-Accelerated Low Power Mote.

## References

[1] A. Engel and A. Koch, "Accelerated clock drift estimation for High-Precision wireless Time-Synchronization," in *Tenth IEEE Workshop on Practical Issues in Building Sensor Network Applications 2015 (IEEE SenseApp 2015)*, Clearwater Beach, USA, 2015.

[2] A. Engel, T. Siebel, and A. Koch, "A heterogeneous system architecture for Low-Power wireless sensor nodes in compute-intensive distributed applications," in *Tenth IEEE Workshop on Practical Issues in Building Sensor Network Applications 2015 (IEEE SenseApp 2015)*, Clearwater Beach, USA, 2015.

[3] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04. ACM, 2004, pp. 39–49.