Temperature Calibration on a Crystal-Free Mote

Titan Yuan*, Filip Maksimovic*, David C. Burnett*, Brad Wheeler*, Lydia Lee* and Kristofer S.J. Pister*
*Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA, USA

***Abstract*—We present a method to calibrate a crystal-free wireless system on chip for use as an IoT temperature sensor between $0\,^{\circ}\mathrm{C}$ and $100\,^{\circ}\mathrm{C}$. Using two clocks on the mote that would already be running during radio operation as a regular IoT node—a timer similar to a $32\,\mathrm{kHz}$ crystal and another $2\,\mathrm{MHz}$ chipping clock for the radio transmitter—we create a temperature sensor by finding a linear relationship between the ambient temperature and the ratio of these two clock frequencies. We show that a simple two-point temperature calibration is sufficient to find this linear model, and by averaging over a few temperature measurements, we observe a temperature error of less than $2\,^{\circ}\mathrm{C}$. We can then use the mote as a tiny wireless temperature sensor by using the temperature estimate to calibrate the radio frequency oscillator.**

***Index Terms*—system-on-chip, temperature sensor, internet of things**

I. Introduction

Obtaining reliable temperature data is important for many IoT applications because of the dependence of many devices on the ambient temperature. Even a rough temperature estimate allows IoT devices to compensate for temperature, thus permitting them to be deployed in various environments. Most importantly, a small, low-cost IoT temperature sensor enables many exciting applications, such as tracking the temperature on different parts of the body using small, wearable temperature sensors. However, it is challenging to keep both the size and the cost of such temperature sensors low.

The single-chip micro mote as presented in [1] has an ARM Cortex M0 microprocessor and can operate as a standards compatible 802.15.4 or Bluetooth Low-Energy (BLE) radio without any external frequency reference. However, a one-time frequency calibration at room temperature is insufficient for crystal-free operation across temperature, hence the need for periodic network compensation. We report a new method to estimate the ambient temperature using two existing running oscillators on the chip during radio operation similar to the method proposed in [2] and thus eliminate the need for periodic compensation after calibration. We use a $32\,\mathrm{kHz}$ free-running oscillator similar to a $32\,\mathrm{kHz}$ crystal and a $2\,\mathrm{MHz}$ oscillator used as a chipping clock for 802.15.4 and BLE transmission. By computing the ratio of these two frequencies, we can linearly estimate the temperature within an error of $2\,^{\circ}\mathrm{C}$ after calibration.

II. Single-Chip Micro Mote Chip

The single-chip micro mote is a low-power wireless chip developed to operate in OpenWSN networks implementing the 802.15.4 time synchronized channel hopping (TSCH) standard. In particular, it has no off-chip frequency reference, so it is prone to local oscillator (LO) frequency drift [1]. However, after a one-time calibration of the frequency during initial programming of the mote, the mote can be used to transmit packets compliant with 802.15.4 standards or Bluetooth Low-Energy (BLE).

Programming is performed optically using an infrared diode connected to a Teensy 3.6 microcontroller that first transmits an executable binary with a CRC check followed by repeated $100\,\mathrm{ms}$ start frame delimiters (SFD) to an optical receiver on the chip. These $100\,\mathrm{ms}$ SFD interrupts are used to calibrate the frequencies of the on-chip oscillators. However, due to the lack of external frequency references, the frequencies drift with temperature, so some temperature compensation is needed after calibration to maintain radio operation within the 802.15.4 or BLE standards over varying temperature.

In [3], previous temperature compensation was done using network-based calibration that tracks the intermediate frequency (IF). An external OpenMote beacon was programmed to periodically transmit 802.15.4 frames every $125\,\mathrm{ms}$, and when the mote received each frame, the IF frequency offset was determined in the clock and data recovery module and was then used to finely tune the RF frequency. Another method described in [4] to allow channel hopping over varying temperature was by using a recursive least-squares (RLS) model that predicts the tuning codes based on the current environment's temperature and on the 802.15.4 channel as demanded by the channel hopping schedule. Initial calibration for the model was done by using the IF frequency offset to ensure the radio frequency accuracy was within $40\,\mathrm{ppm}$ per the 802.15.4 standards. The RLS model was then updated during normal operation when the channel or the environment temperature changed. However, when there is no external beacon providing a constant stream of 802.15.4 frames, such a network-based compensation is infeasible, so we propose a new method to estimate the ambient temperature using on-chip components in order to tune the LO frequency.

III. $2\,\mathrm{MHz}$ and $32\,\mathrm{kHz}$ Oscillators

The $2\,\mathrm{MHz}$ and $32\,\mathrm{kHz}$ oscillators both use a circuit topology similar to the circuit in [5]. Notably, the $2\,\mathrm{MHz}$ circuit, as described in [6] and shown in Fig.1, has a finely tunable resistor as it is the chipping clock for the chip's transmitter, so its frequency needs to be accurate within $\pm 50\,\mathrm{ppm}$ for BLE. It also does not include the additional supply rejection circuitry

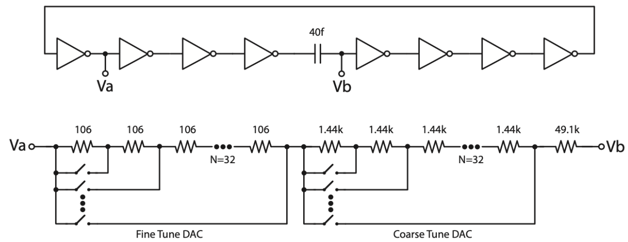


Fig. 1: The $2\,\mathrm{MHz}$ RC oscillator circuit schematic as presented in [7]. The $32\,\mathrm{kHz}$ RC oscillator circuit does not have a tunable resistor DAC.

from [5]. The frequency is tuned by adjusting a 10-bit resistor DAC containing 5 bits of coarse and 5 bits of fine adjustment.

The $32\,\mathrm{kHz}$, on the other hand, is intended to be a low-frequency timer and can be used as the on-chip Cortex M0 clock to conserve energy between higher-power radio operations [6]. Moreover, it does not have a tunable resistor DAC and was originally designed to have a temperature coefficient of zero.

Because the two oscillators were not designed exactly as described to meet this chip's unique requirements, their temperature coefficients are high. In addition, they are sensitive to supply voltage which, while locally regulated, also varies from chip to chip. This requires each chip to be calibrated individually. Results here are presented from one chip.

The frequencies of these two oscillators can be estimated using two internal memory-mapped counter registers in the Cortex M0 microprocessor that increment on the positive edge of the respective clocks. However, only the $2\,\mathrm{MHz}$ RC oscillator frequency is calibrated during initial programming of the chip using the $100\,\mathrm{ms}$ optical SFD interrupts. During each SFD interrupt, we tune the bits controlling the resistor DACs depending on the number of counts within the last $100\,\mathrm{ms}$. If the difference between the expected and actual number of counts is greater than the number of counts per fine or coarse LSB, we adjust the 5-bit fine or coarse tuning codes, respectively. 25 optical SFDs must be received by the mote before calibration is complete.

IV. Linear Relationship between Temperature and $2\,\mathrm{MHz}$ / $32\,\mathrm{kHz}$ Frequency Ratio

We use the $2\,\mathrm{MHz}$ and the $32\,\mathrm{kHz}$ oscillators on the mote to calibrate the mote with respect to temperature. The frequencies of both of these RC oscillators have a non-linear relationship with temperature as plotted in Fig.2, but we observe that the ratio of their two frequencies is linear with respect to temperature and therefore propose a linear model. Calibrating the mote to its ambient temperature is then equivalent to determining this linear relationship.

Using a separate on-chip oscillator, we programmed the mote to trigger an interrupt approximately every $100\,\mathrm{ms}$. This is similar to optical calibration, but the interrupts originate from an on-chip oscillator with around $500\,\mathrm{ppm}$ of noise instead of a Teensy 3.6 microcontroller. We then read the two counts from their respective registers, calculated their ratio, and reset the counters. Since the chip has no floating point unit, a fixed point library was implemented on the chip for floating point operations, such as the ratio calculation. The counts and their ratio were then transmitted to a computer over UART.

Temperature calibration was performed in a TestEquity Model 107 temperature chamber that contained the mote and a reference temperature sensor. The mote was programmed as described previously, and we used a SparkFun TMP102 digital temperature sensor connected to a Teensy 3.6 microcontroller as the reference temperature sensor. Starting from an initial temperature of $25\,^{\circ}\mathrm{C}$, we programmed the temperature chamber to linearly decrease the temperature to $5\,^{\circ}\mathrm{C}$, then increase it to $80\,^{\circ}\mathrm{C}$, and finally decrease it back down to room temperature at a rate of $\pm 1.5\,\frac{^{\circ}\mathrm{C}}{\mathrm{min}}$. Since the mote was placed on a development PCB with a large metal ground plane, the board's large thermal mass caused some hysteresis in the ratio measurements if the temperature ramp rate was too high. The temperature and the ratio of the $2\,\mathrm{MHz}$ and the $32\,\mathrm{kHz}$ frequencies were logged every $100\,\mathrm{ms}$, the results of which are shown in Fig.3 overlaid with a linear regression line.

We observed that the resulting graph can be very accurately approximated as a linear one although the coefficients of the linear model are chip-dependent. For the particular mote we used, we found using a linear least-squares regression:

$$\text{temp} = -30.715 \cdot \text{ratio} + 1915.142 \tag{1}$$

The maximum error between the linear model and the actual temperature was less than $3\,^{\circ}\mathrm{C}$, but we did notice some measurement inaccuracies due to hysteresis caused by the board's thermal mass.

V. Temperature Averaging

After the initial temperature calibration described above, we observed that the mote's estimated temperature at a given fixed temperature would vary by up to $1\,^{\circ}\mathrm{C}$. Compared to the Allan deviation plots in [6] for the $2\,\mathrm{MHz}$ and $32\,\mathrm{kHz}$ oscillators, the data we observed had more jitter in the $100\,\mathrm{ms}$ range.

To mitigate variance in the temperature estimate, we implemented a simple averaging method. We recorded the $2\,\mathrm{MHz}$ and the $32\,\mathrm{kHz}$ frequency ratios at room temperature over an hour and calculated the standard deviations of the estimated temperature as a function of how many samples we average. The results are shown in Fig.4.

We found that averaging over any more than five temperature samples did not significantly decrease the variance of the temperature estimates, so we chose to average over five samples. This results in a duration of around $500\,\mathrm{ms}$ for each temperature measurement.

Using the coefficients given in (1) and averaging over 5 samples, we verified the accuracy of the mote's estimated temperature at intervals of $5\,^{\circ}\mathrm{C}$ between $5\,^{\circ}\mathrm{C}$ and $80\,^{\circ}\mathrm{C}$. At each temperature, we waited for the mote's estimated temperature to stabilize and then recorded around 50 consecutive temperature measurements before increasing the temperature by another

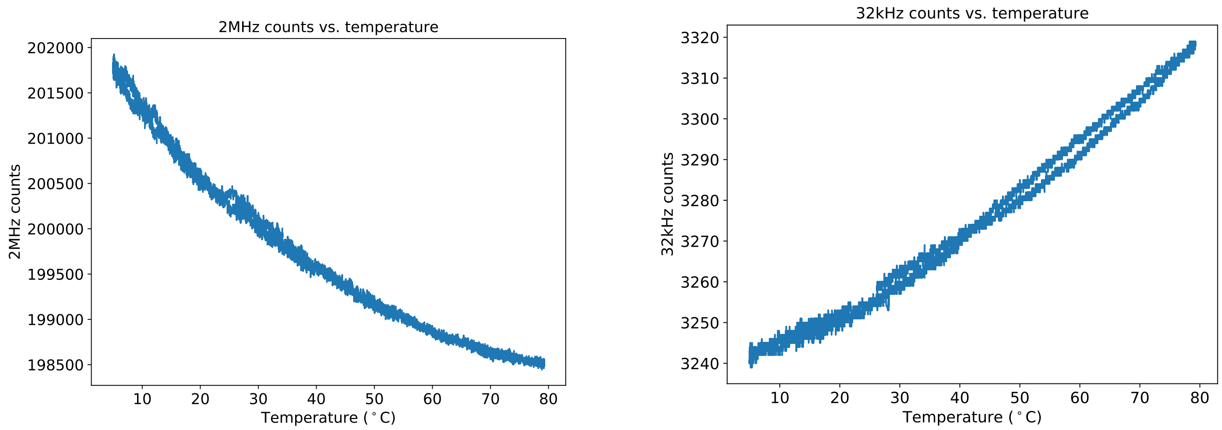


Fig. 2: The mote's $2\,\mathrm{MHz}$ (left) and $32\,\mathrm{kHz}$ (right) frequency counts vs. temperature. The counts were recorded every $100\,\mathrm{ms}$.

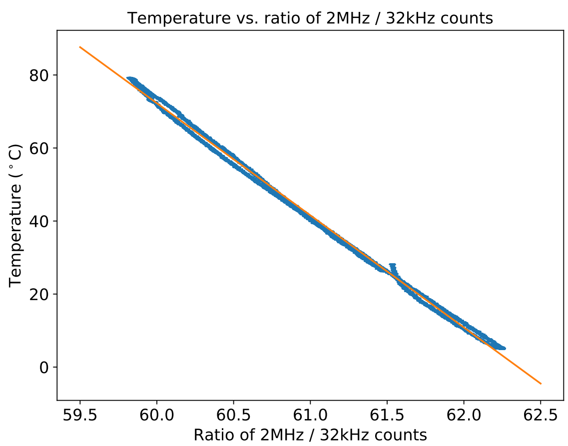


Fig. 3: Temperature vs. ratio of the $2\,\mathrm{MHz}$ and $32\,\mathrm{kHz}$ counts (blue) and the linear model given by linear least-squares regression (orange). The temperature was varied at a rate of $\pm 1.5\,^{\circ}\mathrm{C}$, and ratio measurements were taken every $100\,\mathrm{ms}$.

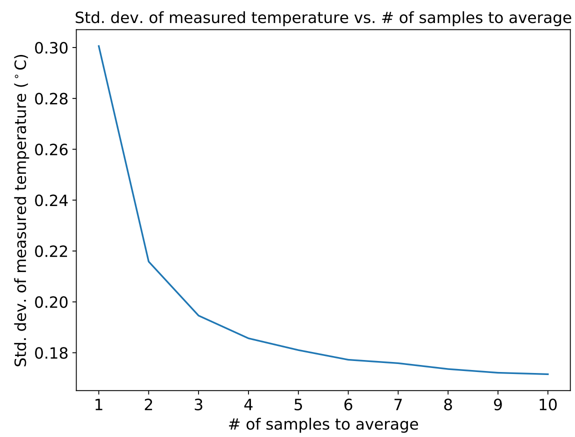


Fig. 4: Standard deviation of the measured temperature vs. number of samples to average over.

$5\,^{\circ}\mathrm{C}$. In Fig.5a, we used the means of these estimated temperatures at each $5\,^{\circ}\mathrm{C}$ interval to plot the mote's estimated temperature between $5\,^{\circ}\mathrm{C}$ and $80\,^{\circ}\mathrm{C}$, and in Fig.5b, we show the corresponding temperature errors and their distributions. While the measured temperature is fairly accurate around room temperature ($20\,^{\circ}\mathrm{C}$), the maximum error of around $2\,^{\circ}\mathrm{C}$ occurs at higher temperatures.

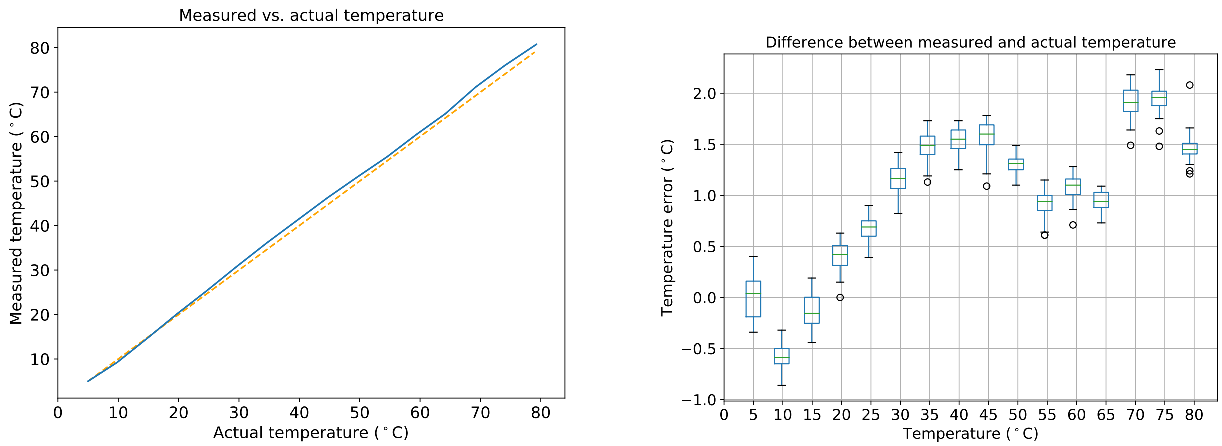
VI. Two-Point Calibration

Due to the linear relationship between the temperature and the ratio of the $2\,\mathrm{MHz}$ and the $32\,\mathrm{kHz}$ frequencies, a lengthy temperature sweep to find this relationship is rather unnecessary. Instead, we propose a two-point calibration at two different temperatures. Using the same mote, we performed the temperature calibration as described above again, but we measured the ratio of the two frequencies only at $20\,^{\circ}\mathrm{C}$ and at $30\,^{\circ}\mathrm{C}$, both after the temperature in the chamber stabilized. Notably, we did not choose to calibrate at the extremes of the temperature range in order to reduce the cost and time of the two-point calibration.

After calculating a linear model for the measured temperature and adjusting the bias term to account for observed hysteresis, we reprogrammed the mote and verified the accuracy of this model by sweeping the chamber temperature from $0\,^{\circ}\mathrm{C}$ to $100\,^{\circ}\mathrm{C}$ while recording the mote's estimated temperature and using the TMP102 temperature sensor as the ground truth. The measured temperature and its corresponding error between $0\,^{\circ}\mathrm{C}$ and $100\,^{\circ}\mathrm{C}$ are shown in Fig.6. At low temperatures, we noticed sporadic overflow errors occurring in the embedded software due to the implemented fixed point division algorithm. Ignoring the chip's erroneous estimated temperatures, we found that the difference between the measured and the actual temperature was within $1\,^{\circ}\mathrm{C}$.

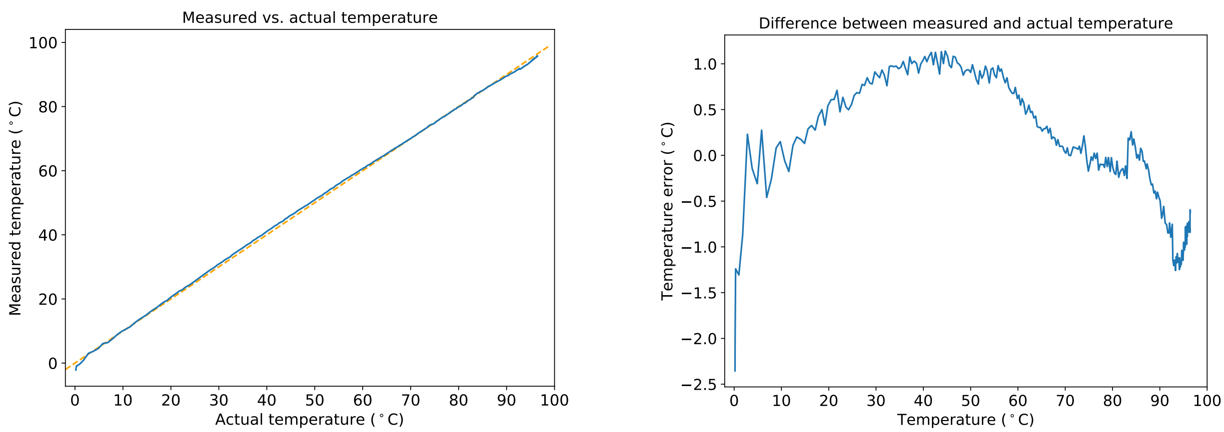
VII. Conclusion

We showed a method to find a linear relationship between the temperature and the ratio of two fully on-chip timers with other designed functions, a $32\,\mathrm{kHz}$ oscillator similar to a



(a) (b)

Fig. 5: The mote's measured temperature (solid blue) vs. the reference temperature (dashed orange) after a temperature sweep at a ramp rate of $1.5\,\frac{^\circ\mathrm{C}}{\mathrm{min}}$ (left) and the corresponding distribution of the temperature error between $5\,^\circ\mathrm{C}$ and $80\,^\circ\mathrm{C}$ (right).



(a) (b)

Fig. 6: The mote's measured temperature (solid blue) vs. the reference temperature (dashed orange) after a two-point calibration (left) and the corresponding temperature error between $0\,^\circ\mathrm{C}$ and $100\,^\circ\mathrm{C}$ (right).

crystal and a $2\,\mathrm{MHz}$ chipping clock for the chip's transmitter. Although these two RC oscillators have different temperature coefficients, their ratio is roughly linear over temperature. Using a two-point calibration, the coefficients of the linear model can be easily determined, and after averaging over 5 temperature samples, we showed that the error of the mote's measured temperature is less than $2\,^\circ\mathrm{C}$ within a temperature range from $0\,^\circ\mathrm{C}$ to $100\,^\circ\mathrm{C}$. This allows the crystal-free mote to generate temperature estimates and perform temperature compensation using these estimates, which eliminates the need for network-based compensation requiring an external beacon. The next step is to tune the radio frequency oscillator using this temperature estimate, so that it can operate within the 802.15.4 or BLE standards over varying temperature without needing an external frequency reference. We can also transmit this temperature, so that the mote can operate as a tiny wireless temperature sensor.

Acknowledgment

The authors would like to thank members of the Autonomous Microsystems Lab as well as the Siebel Scholars Foundation for funding this work.

References

[1] F. Maksimovic, B. Wheeler, D. C. Burnett, O. Khan, S. Mesri, I. Suciu, L. Lee, A. Moreno, A. Sundararajan, B. Zhou, R. Zoll, A. Ng, T. Chang, X. Villajosana, T. Watteyne, A. Niknejad, and K. S. J. Pister, "A Crystal-Free Single-Chip Micro Mote with Integrated 802.15.4 Compatible Transceiver, sub-mW BLE Compatible Beacon Transmitter, and Cortex M0," in *2019 Symposium on VLSI Circuits*, June 2019, pp. C88–C89.

[2] S. Park, C. Min, and S. Cho, "A 95nW ring oscillator-based temperature sensor for RFID tags in 0.13μm CMOS," in *2009 IEEE International Symposium on Circuits and Systems*, May 2009, pp. 1153–1156.

[3] I. Suciu, F. Maksimovic, D. Burnett, O. Khan, B. Wheeler, A. Sundararajan, T. Watteyne, X. Vilajosana, and K. Pister, "Experimental Clock Calibration on a Crystal-Free Mote-on-a-Chip," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2019, pp. 608–613.

[4] I. Suciu, F. Maksimovic, B. Wheeler, D. C. Burnett, O. Khan, T. Watteyne, X. Vilajosana, and K. S. J. Pister, “Dynamic Channel Calibration on a Crystal-Free Mote-on-a-Chip,” *IEEE Access*, vol. 7, pp. 120 884–120 900, 2019.

[5] D. Griffith, P. T. Røine, J. Murdock, and R. Smith, “A 190nW 33kHz RC oscillator with $\pm 0.21\%$ temperature stability and 4ppm long-term stability,” in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, Feb 2014, pp. 300–301.

[6] D. C. Burnett, B. Wheeler, L. Lee, F. Maksimovic, A. Sundararajan, O. Khan, and K. S. J. Pister, “CMOS oscillators to satisfy 802.15.4 and Bluetooth LE PHY specifications without a crystal reference,” in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2019, pp. 0218–0223.

[7] B. Wheeler, “Low Power, Crystal-Free Design for Monolithic Receivers,” Ph.D. dissertation, EECS Department, University of California, Berkeley, May 2019. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-36.html