RefCal: Regression-based multi-Frequencies Calibration for Crystal-Free micro-Chip

Yiming Yuan
yyuan861@connect.hkust-gz.edu.cn
HKUST(GZ),IoT Thrust
Guangzhou, China

Yuanming Luo
yluo334@connect.hkust-gz.edu.cn
HKUST(GZ),IoT Thrust
Guangzhou, China

David Burnett
davidburnett@pdx.edu
Portlant State University
Portland, USA

Filip Maksimovic
filip.maksimovic@inria.fr
Inria
Paris, France

Tengfei Chang
tengfeichang@hkust-gz.edu.cn
HKUST(GZ),IoT Thrust
Guangzhou, China

ABSTRACT

SCμM (single-chip micro mote) is a $2 \times 3 \times 0.3\text{mm}^3$-sized crystal-free mote-on-chip. The synthesized on-chip 2.4 GHz radio is compatible with the Bluetooth Low Energy (BLE) and IEEE 802.15.4 standards. While the crystal-less design offers a size advantage, it also introduces challenges, such as frequency calibration for radio communication. In this work, we propose RefCal, a regression-based fitting approach to determine the correct frequency setting. Compared to existing methods that require blind sweeping across a large frequency setting range, RefCal can quickly identify the correct frequency settings within a narrow range based on a set of fitting models. The results indicate improvements in calibration delay and over a 83% reduce in number of packets SCμM needs to transmit, which induce to save energy and time consumption compared to existing methods.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**.

KEYWORDS

SCμM, Crystal-free, Bluetooth Low Energy (BLE), Regression, Channel-Hopping

1 INTRODUCTION

In the era of the Internet of Things (IoT), the number of connected devices has experienced exponential growth. The deployment of IoT devices is expanding across diverse application scenarios, including personal health monitoring [10], factory automation [5], smart cities [9], and micro-robotics [6, 8]. Key design considerations for these applications, particularly in large-scale deployments, include small footprint, cost efficiency, and low power consumption. To address these challenges, the integration of components at the chiplet level has emerged as a prominent trend.

The RF chip and the crystal for communication are usually the two largest components in the communication module. Tab. 1 lists the sizes of common BLE System-on-Chips (SoCs) and the corresponding crystal footprint in the market for the SoCs to communicate. The minimal-footprint crystals are investigated through a tool provided on the website of Mouser. As shown in Tab. 1, the ratio of crystal and SoC size ranges from 14% to 49%.. A crystal-free SoC could significantly reduce the footprint of an IoT device.

SoCs	Min. size of SoC	Min. size of required crystal
nRF52840	3.5 mm x 3.6 mm	2.5 mm x 2.0 mm for 32 MHz
nRF52833	3.2 mm x 3.2 mm	2.5 mm x 2.0 mm for 32 MHz
EFR32MG24	5 mm x 5 mm	2 mm x 2.5 mm for 39 MHz
MKW41Z512	3.9 mm x 3.8 mm	2.5 mm x 2.0 mm for 32 MHz
CC2650	4 mm x 4 mm	2 mm x 1.6 mm for 24 Mhz

Table 1: Package comparison of 2.4 GHz low-power SoCs and required crystals.

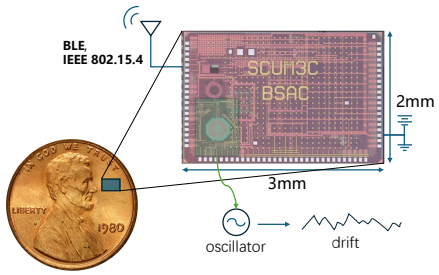

Figure 1: SCμM size on a penny. The 2.4GHz transceiver is clocked with on-chip oscillator which needs to be calibrated.

The Single-Chip micro-Mote, SCμM [4], is the first crystal-free Cortex-M0-based chip with dimensions of 2 mm×3 mm×0.3 mm. It is equipped with a transceiver compatible for BLE and IEEE 802.15.4 standards [12]. The transceiver is clocked using two main internal on-chip oscillators: a 2.4 GHz LC oscillator for generating the electromagnetic wave, and a 2 MHz RC oscillator to clock the chip rate. The current version of SCμM operates with just an external antenna and battery, as shown in Fig. 1.

With the elimination of the external crystal, SCμM provides an ultra-small footprint to support applications where size is critical. However, this also introduces challenges in frequency calibration. The internal oscillators are less accurate compared to crystal-based ones. Due to variations in the manufacturing process, voltage, or temperature changes, the oscillators on different chips may operate

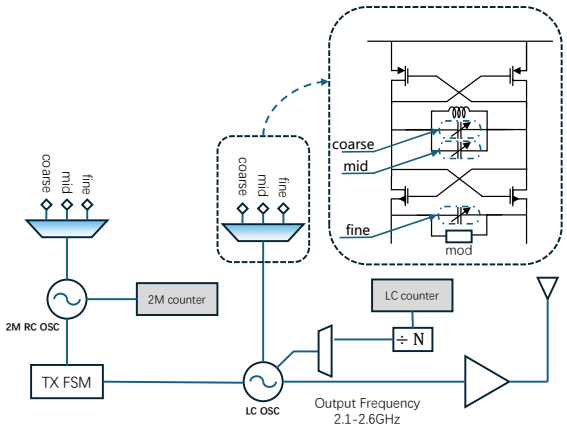

Figure 2: SCμM oscillator structure. Both the 2M RC oscillator and the LC oscillator can be tuned in three different resolutions: (< `coarse` >, < `mid` >, < `fine` >). For the LC oscillator, these three variables are 5-bit registers that control the value of the capacitors, which in turn impact the oscillator frequency. The LC counter logs the number of ticks of the divided LC oscillator frequency.

at different frequencies [7]. SCμM provides three adjustable capacitors for tuning the frequency at `coarse`, `mid`, and `fine` resolutions. Each capacitor is controlled by a 5-bit Digital-to-Analog Converter (DAC), resulting in a total 15-bit tuning range, as shown in Fig. 2. Each of the 15-bit values can be represented as a tuple (`<coarse>`, `<mid>`, `<fine>`), referred to as **frequency settings**. Due to engineering challenges, it is extremely difficult to map frequency settings evenly across the entire communication frequency band. To avoid missing any target channel frequency, an overlapping design is adopted, resulting in saw-like shape mapping between frequency settings and the corresponding frequency (as shown in Fig. 3). This mapping is chip-dependent and temperature varying.

This article proposes RefCal, a regression-based multi-frequency calibration approach for SCμM to determine the frequency settings for all channels simultaneously, without manually sweeping through tens of thousands of settings. RefCal dramatically reduces the latency and energy consumption required to find all channel frequency settings compared to existing calibration methods.

The rest of this article is organized as follows: Sec. 2 introduces the existing calibration approaches for SCμM. Sec. 3 details the proposed RefCal calibration approach. Sec. 4 validates the correctness of RefCal through experiments and evaluates its performance. Sec. 5 concludes this article.

2 RELATED WORK

The calibration for SCμM involves finding the correct frequency settings for each target channel, i.e., 40 channels defined in BLE or 16 channels defined in IEEE 802.15.4. The existing calibration approaches mainly focus on three aspects: 1) single-channel frequency calibration, 2) multiple-channel frequency calibration, and 3) frequency calibration over temperature.

In the early stages, [11] utilized the optical receiver embedded in SCμM to calibrate the LC oscillator. The optical receiver is initially designed for programming SCμM. By pulsing a fixed interval of optical interrupts after programming, SCμM can self-calibrate the internal oscillators based on that interval. This optical interrupt-based calibration only supports single-channel calibration. The CoCa calibration in [2] addresses the issue of calibration over different temperatures. The approach relies on the intermediate frequency (IF) counter of SCμM and the frequency offset (FO) of CC2538 to monitor frequency errors. CoCa calibrates the frequency by tuning the frequency settings to keep the IF counter and FO at target values. CoCa has proven that SCμM can maintain communication with a CC2538 device under a temperature change rate of 3°C/min. However, CoCa also supports only single-channel calibration. QuickCal [1] introduces a calibration approach using 16 OpenMotes (CC2538-based IEEE 802.15.4 devices) and can calibrate the frequencies of all 16 IEEE 802.15.4 channels. In QuickCal, SCμM finds the frequency settings for each channel by having each OpenMote send packets over 16 channels. SCμM listens for packets by cycling through each frequency setting within a certain range and records the setting where a packet is received, indicating the correct setting for listening. Similarly, by sending packets at each frequency setting within the same range, SCμM can find the correct setting for transmitting. [3] proposed an improved version to find all channel frequencies using only one OpenMote. With the proposed method, a multi-hop synchronized mesh network can be established with channel hopping enabled. The drawback of sweeping-based calibration approaches is that they blindly try all the frequency settings without any prior knowledge of the LC oscillator of SCμM, resulting in a large overhead in traffic bandwidth cost and energy consumption.

This paper proposes RefCal, a multiple-channel calibration approach for SCμM. RefCal utilizes a second-order linear regression model to find the fitting curve for the relationship between frequency settings and their corresponding frequencies. The proposed regression fitting model is achieved through pure on-chip calculation. Once the model is established, the target frequency settings corresponding to all channel frequencies can be calculated directly, without any radio activity involved. Compared to existing multiple-channel calibration approaches, RefCal is faster and more energy-efficient as it does not require sending or receiving packets over a range of frequency settings.

3 REFCAL: REGRESSION-BASED FREQUENCY CALIBRATION

The goal of RefCal is simple: find the function between frequency and tuning settings and provide the corresponding setting when specific channel communication is need.

To know the function of frequency over settings, `LC count` is utilized. With a fixed interval, i.e. 1ms, armed by the HF clock, which is calibrated at the initial optical interval, the `LC count` provides the ticks of LC oscillator passed during the interval. Since there is a divider, here 960 was selected as a multiple of 2.4 GHz, applied from the LC oscillator to the LC counter, as indicated in Fig. 2, the `LC count` can be represented with Eq. 1.

$$LC_{count} = \frac{frequency(Hz) \times 10^{-3}}{960} \tag{1}$$

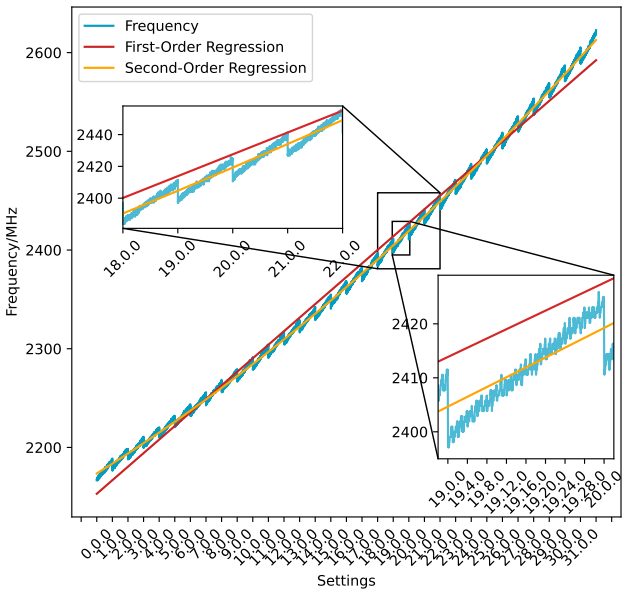

Figure 3: The frequency shows a saw-like curve over all frequency settings. Second-order polynomial regression fits the curve comparing to linear regression.

The equation requires the LC oscillator to be calibrated first. Theoretically, the initial calibration by optical or wired during the programming phase can calibrate all SCμM on-chip oscillators. However, the 1 ms interval is also generated by the on-chip unstable oscillator, which can lead to inaccuracies in the LC count measured within this 1 ms period, thereby introducing offset in frequency. Therefore, we also use a 1 ms duration to monitor the 2M RC oscillator count as a control compensation, reading and use this count, which is called $2MHz_{real}$ to calibrate the LC oscillator and 2000 is the ideal count of 2M RC oscillator within 1 ms. The updated equation is shown in Eq. 2.

$$LC_{count} = \frac{frequency(Hz) \times 10^{-3} \times \frac{2MHz_{real}}{2000}}{960} \tag{2}$$

The saw-like shape plot is the required function of frequency settings and corresponding frequency, but it takes more than 30 seconds to obtain one by one and contains 32,768 sample points. If the environment changes, this process must be repeated. The idea is to use a relatively simple function that is accurate enough to replace this saw-like shape function shown in Fig. 3.

3.1 Choosing the Orders of Polynomial Regression

The first step is to determine the order of polynomial curves to be used for fitting the saw-like function in Fig. 3. We experimented with first-order, second-order, and third-order curves and calculated the coefficient of determination, R^2, correspondingly. R^2 is an indicator that measures the degree of fit between the fitted model and the data. An R^2 value of 1 indicates a perfect match. The R^2 equation and the fitting results are provided in Eq. 3 and Tab. 2. The corresponding curves are also shown in Fig. 3.

Table 2: R^2 value (degree of fit) of different fitting curves

	1st-Order Regression	2nd-Order Regression	3rd-Order Regression	2ns-Order by Selected Points
R^2	0.99402	0.99923	0.99926	0.99917

$$R^2 = 1 - \frac{\sum(y_i - y_{regression})^2}{\sum(y_i - \bar{y})^2} \tag{3}$$

The second and third-order regressions have a higher R^2 value compared to the first order. The zoomed-in section in Fig. 3 clearly shows that the first-order fitting line deviates from the main trend, while the second-order line intersects the trend in the middle where the <coarse> value changes. Regarding the R^2 value, the third-order regression shows only a slight improvement compared to the second-order. To maintain lower calculation complexity, the second-order polynomial function is chosen for fitting the full curve.

This second-order function helps to select the coarse setting, as shown in Fig. 4. To facilitate a better understanding, we provide two examples. Assuming the target frequency is 2350 MHz (shown as the green line in Fig. 4) , the green line intersects with the second-order function curve at (15.3.X)(we don't consider <fine> here), as shown in the upper plot of Fig. 4. The actual frequency setting is off from the intersects (15.3.X) due to the overlap of frequency. To identify the correct <mid> setting, two first-order fitting lines are applied to the data from (14.16.0) to (14.31.31) and (15.0.0) to (15.15.31) (A2 and A1 in above Fig. 4). We indicate the area where (15.3.X) located as A1, from (15.0.0) to (15.15.31) in this example. As in the smaller or larger <coarse> setting may also have viable frequency setting, either smaller or larger <coarse> will also be chosen depending on the <mid> setting calculated based on the second-order function curve. In this case, it is 3 in (15.3.X), which is less than $32/2 = 16$, leading to the selection of a smaller <coarse>, 14, setting as A2. Otherwise, the next <coarse> setting would be chosen as A2, as shown in the bottom plot of Fig. 3.

The two first-order fitting lines intersect with the target frequency, resulting in two settings (14.23.25) and (15.7.20). The final target settings will be chosen around these two settings.

The next section details how the parameters of the second-order function curves and the two linear fitting lines are calculated.

3.2 Function Parameters Calculation

3.2.1 Parameters of Second-Order Function. A second-order function presenting frequency setting and corresponding frequency can be represented as shown in Eq. 4. The variable x_c represents the <coarse> settings, which range from 0 to 31. The variable y_f is the corresponding frequency of the <coarse> setting. Since there are three parameters in the equation, at least three points are required.

$$y_f = ax_c^2 + bx_c + c \tag{4}$$

Based on the second-order regression mentioned previously, we found that the curve mostly passes through the middle points between the last <coarse> setting ((x.31.31)) and the first following <coarse> setting ((x.0.0)), as shown in Fig. 3. Hence, we

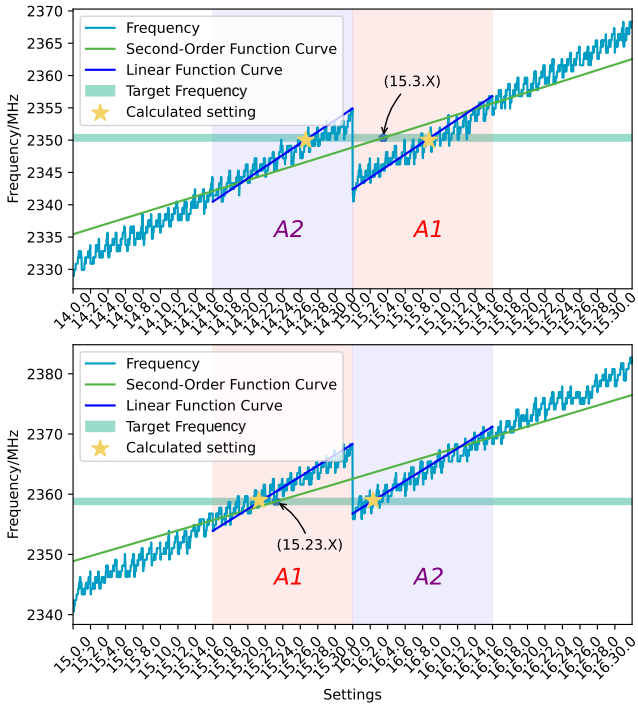

Figure 4: Second-order function is used for identifying the `coarse` setting. First-order fitting lines within single `coarse` settings are used for identifying the `mid` setting.

separately pick the three points $P1(x, y)$, $P2(x, y)$, and $P3(x, y)$ as shown in Eq. 5, 6, and 7 to describe the whole trend of function.

$$P1(x,y), \quad P1_x = 2, \quad P1_y = \frac{f_{(1.31.31)} + f_{(2.0.0)}}{2} \tag{5}$$

$$P2(x,y), \quad P2_x = 16, \quad P2_y = \frac{f_{(15.31.31)} + f_{(16.0.0)}}{2} \tag{6}$$

$$P3(x,y), \quad P3_x = 31, \quad P3_y = \frac{f_{(30.31.31)} + f_{(31.0.0)}}{2} \tag{7}$$

The parameter a and b are calculated based on the matrix equation Eq. 8 and the determinant of the matrix Eq. 9 is not equal to 0.

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} P2_x^2 - P1_x^2 & P2_x - P1_x \\ P3_x^2 - P1_x^2 & P3_x - P1_x \end{bmatrix}^{-1} \times \begin{bmatrix} P2_y - P1_y \\ P3_y - P1_y \end{bmatrix} \tag{8}$$

$$\begin{vmatrix} P2_x^2 - P1_x^2 & P2_x - P1_x \\ P3_x^2 - P1_x^2 & P3_x - P1_x \end{vmatrix} \neq 0 \tag{9}$$

The parameter c is calculate based on Eq. 10.

$$c = P1_y - a * P1_x^2 - b * P1_x \tag{10}$$

After obtaining the parameters a, b, and c, the target LC frequency <coarse> setting can be calculated using Eq. 11.

$$setting_{\texttt{<coarse>}} = \frac{-b \pm \sqrt{b^2 - 4a(c - f_{target}(MHz))}}{2a} \tag{11}$$

The $setting_{\texttt{<coarse>}}$ can be a floating-point number, e.g., 15.718. The fractional part is used to calculate the <mid> setting. For 15.718, the <mid> setting is rounded to 0.718×32, which is 23. According to the <mid> selection explained in Sec. 3.1, the final setting will be within one of two ranges: A1 from `(15.16.0)` to `(15.31.31)` and A2 from `(16.0.0)` to `(16.15.31)`. The first-order function will be applied to these two ranges.

3.2.2 Parameters of First-Order Function. The linear function can be represented as Eq. 16. The variable x_s represents the frequency settings within either of the two ranges. The variable y_f is the corresponding frequency converted from the LC count read. The equation has two parameters, k and d, requiring two points to calculate these parameters. Using $setting_{\texttt{coarse}} = 15.718$ as an example, we pick two points for each of the linear functions as shown in Eq. 12, 13, 14, and 15.

$(Pl1_{xs1}, Pl1_{yf1})$ and $(Pl1_{xs2}, Pl1_{yf2})$ are used for the linear function $l1$ for settings from `(15.16.0)` to `(15.31.31)`. $(Pl2_{xs1}, Pl2_{yf1})$ and $(Pl2_{xs2}, Pl2_{yf2})$ are used for the linear function $l2$ for settings from `(16.0.0)` to `(16.15.31)`.

$$Pl1_{xs1} = (15.16.0), \quad Pl1_{yf1} = f_{(15.16.0)} \tag{12}$$

$$Pl1_{xs2} = (15.31.31), \quad Pl1_{yf2} = f_{(15.31.31)} \tag{13}$$

$$Pl2_{xs1} = (16.0.0), \quad Pl2_{yf1} = f_{(16.0.0)} \tag{14}$$

$$Pl2_{xs2} = (16.15.31), \quad Pl2_{yf2} = f_{(16.15.31)} \tag{15}$$

The parameters k and d for the linear function $l1$ are calculated based on Eq. 17. The parameters for function $l2$ can be calculated in the same way.

$$y_f = kx_s + d \tag{16}$$

$$\begin{cases} k = \frac{Pl1_{yf2} - Pl1_{yf1}}{Pl1_{xs2} - Pl1_{xs1}} \\ d = Pl1_{yf1} - k \times Pl1_{xs1} \end{cases} \tag{17}$$

3.3 Final Setting Selection

Once the parameters are determined, the corresponding frequency settings ($setting_{target}$) for a given target frequency can be identified based on Eq. 16 in reverse. Due to the LC count resolution limits and initial calibration accuracy, the $setting_{target}$ may deviate slightly from the calculated linear curve. Instead of relying on a single frequency setting, a range of settings within $[setting_{target} - 8, setting_{target} + 8]$ is selected as a backup for communication. After monitoring the reliability of each of the 17 settings,the half within a <mid> setting, one setting can be chosen as the final target setting for communication with best performance. Fig. 5 shows the full workflow of the RefCal approach.

4 EVALUATION

To evaluate the performance of RefCal, we conduct an experiment by having SCμM send BLE packets to a nRF52840-dk board. All the calculation will run on board. The nRF52840-dk is configured to listen on channel 0. SCμM continuously sends 100 21-byte long packets (1 byte preamble, 4 byte address, 13 byte PDU and 3 byte CRC) at each of the settings. The corresponding setting is written in the payload of the packet. If the settings for transmitting are correct,

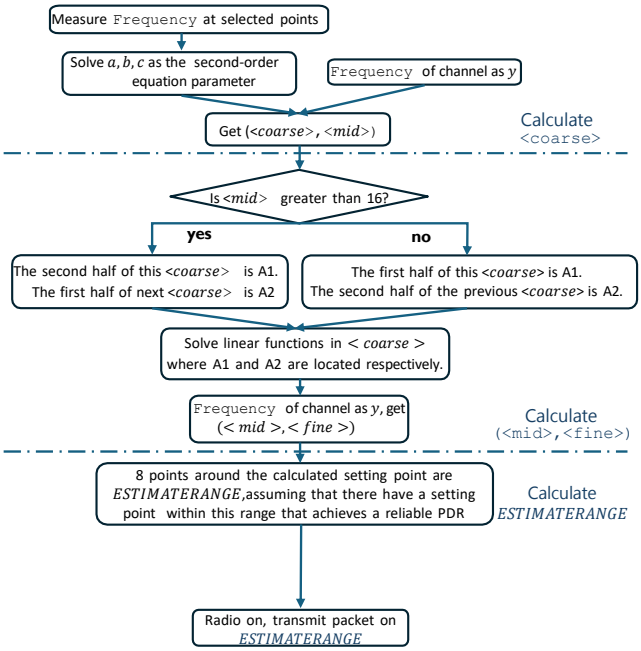

Figure 5: Workflow of the algorithm.

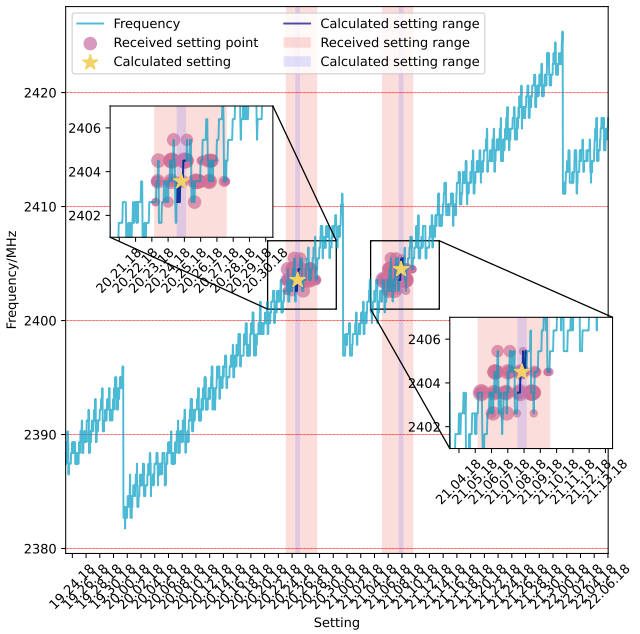

Figure 6: Packet received at different settings. The RefCal calculated setting ranges are within the range of settings with packet received.

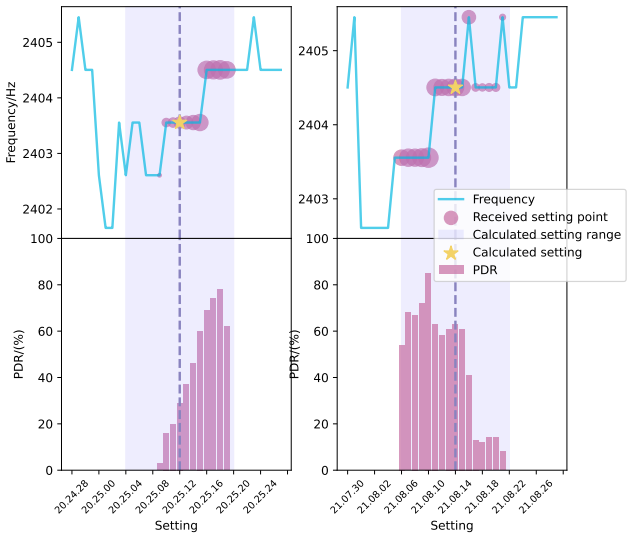

Figure 7: PDRs of the settings within RefCal calculated setting ranges. The highest PDR of the settings in the range (i.e. (21.8.10)) is greater than 88% .

the nRF52840-dk will receive the packet and log the corresponding settings at 15cm distance . The settings with received packets will be compared with the calculated frequency settings by RefCal.

The frequency of BLE channel 0 is 2.404 GHz. As indicated in [12], the LC frequency needs to be 250 KHz lower than the target TX frequency, making the target frequency to tune 2.40375 GHz. With this frequency as input, RefCal calculates the target settings as (20.25.11) and (21.8.14).

The correctness of RefCal validate result is shown in Fig. 6. The background plot of the figure shows the frequency over settings from (19.20.18) to (22.08.18). The RefCal calculated settings are marked with yellow stars. The settings with received packets are marked with purple dots. The larger dot size indicates more packets were received and logged at that setting. It can be seen that the calculated setting ranges are within the settings ranges with received packets.

The packet delivery ratio (PDR) for each setting in the calculated setting range is shown in Fig. 7, which is marked by the pink shadow. Within this range, the highest PDR among the settings (i.e., (21.8.10)) is greater than 85%. This is also the highest PDR among all the settings with received packets.

To certify the stability RefCal, we repeated the RefCal algorithm 100 times and calculated 100 settings with best PDR for channel 0. The upper plot in Fig. 8 indicates the mean value of the PDRs is 80% among the 100 best-PDR settings. The bottom plot shows the cumulative distribution function (CDF) of these PDRs, which indicates over 98% settings have a PDR greater than 70%.

4.1 Multi-Channel Calibration Performance

The strength of RefCal is that it avoids the long calibration process and the transmission of a large number of calibration packets by blindly probing each of the settings, as seen in the sweeping method used in [1]. To evaluate the performance of multi-channel frequency calibration with RefCal, the PDRs on different channel frequencies were also measured 30 times, and the results distribution are shown in Fig. 9. Seven different channel frequencies (0, 6, 12, 18, 24, 30, and 36) with even spacing across the full Bluetooth 2.4 GHz band were

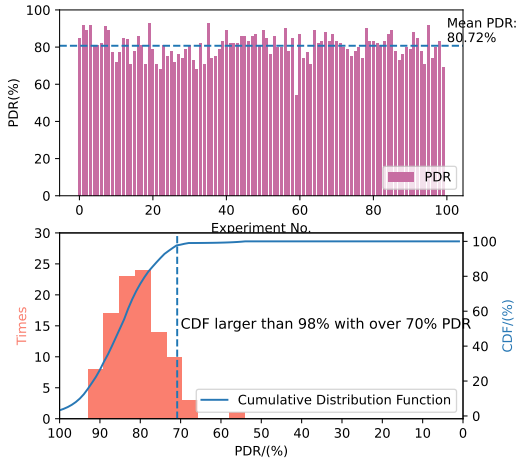

Figure 8: PDRs distribution for 100 settings calculated by RefCal for Channel 0. The mean value of PDR is 80.72%. (*upper plot*) 98% of the calculated settings have a PDR greater than 70%. (*bottom plot*)

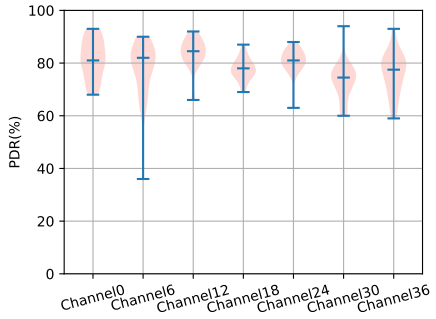

Figure 9: PDRs distribution for 30 settings calculated using the RefCal on 7 different channels.

evaluated. Fig. 9 indicates that, by using RefCal, the average PDRs at each channel are around 80%. Our experiments with the same configuration using the sweep method resulted in a maximum PDR of 95% and an average PDR of 81%, which is approximately same as the performace in RefCal.

To the calculation time, RefCal takes 1.2s in total, including the LC count measuring, the second-order function and two linear functions establishing and the setting calculation. The final best setting selection takes 340 packets per channel by sending 10 packets at 1ms interval on each of the settings in the two 17 setting ranges. Comparing to the sweeping-based methods which takes roughly 2048 packets for per channel, RefCal reduces 83% of packets transmission. Since the number packets sent by RefCal is reduced, the energy consumption is reduced as well.

5 CONCLUSION AND FUTURE WORK

SCμM is a $2 \times 3\text{mm}^2$ crystal-free mote-on-chip. The small footprint of the mote enables numerous applications where size matters. While being crystal-free offers advantages in size and cost, it also introduces challenges in frequency calibration.

This article introduced RefCal, a multi-channel frequency calibration approach for SCμM. Compared to other existing multi-channel frequency calibration approaches, which require probing each frequency setting, RefCal utilizes one second-order and two first-order polynomial functions to model the relationship between frequency and settings. Using this model, the frequency settings for multiple channel frequencies can be calculated directly. RefCal saves thousands of packet transmissions for calibrating multiple channels, thereby dramatically reducing both time and energy consumption.

The performance of RefCal was evaluated using an nRF52840-dk acting as the receiver. One SCμM was configured to send packets over all settings, and the settings of the packets received by the nRF52840-dk were recorded. According to the results, over 80 packets were received within the RefCal calculated setting ranges, resulting in a PDR greater than 80%. The settings for seven different channel frequencies calculated by RefCal were also evaluated, achieving an overall PDR greater than 80%.

In the future, the performance of RefCal in a temperature-varying environment needs to be evaluated. Under different temperatures, the frequency-over-setting model will change. The second-order and first-order functions will need to be recalculated. Determining whether the frequency settings calculated with models under different temperatures remain valid is the next step in this work.

REFERENCES

[1] Chang, T., Watteyne, T., Maksimovic, F., Wheeler, B., Burnett, D. C., Yuan, T., Vilajosana, X., and Pister, K. S. Quickcal: Assisted calibration for crystal-free micromotes. *IEEE Internet of Things Journal 8*, 3 (August 2020), 1846–1858.

[2] Chang, T., Watteyne, T., Wheeler, B., Maksimovic, F., Burnett, D. C., and Pister, K. Surviving the hair dryer: Continuous calibration of a crystal-free mote-on-chip. *IEEE Internet of Things Journal 9*, 6 (August 2021), 4737–4747.

[3] Maksimovic, F., Patel, A., Burnett, D., Watteyne, T., and Pister, K. S. A time synchronized multi-hop mesh network with crystal-free nodes. In *GLOBECOM 2023-2023 IEEE Global Communications Conference* (Kuala Lumpur, Malaysia, February 2023), IEEE, IEEE, pp. 4158–4163.

[4] Maksimovic, F., Wheeler, B., Burnett, D. C., Khan, O., Mesri, S., Suciu, I., Lee, L., Moreno, A., Sundararajan, A., Zhou, B., et al. 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* (July 2019), IEEE, pp. C88–C89.

[5] Mondal, D. The internet of thing (iot) and industrial automation: A future perspective. *World J. Model. Simul 15*, 2 (2019), 140–149.

[6] Moreno, A., Maksimovic, F., Lee, L., Kilberg, B., Schindler, C., Gomez, H., Teal, D., Acker-James, D., Fearing, A., Rentmeister, J. S., et al. Single-chip micro-mote for microrobotic platforms. In *Government Microcircuit Applications & Critical Technology Conference. GOMACTech* (2020).

[7] Suciu, I., Maksimovic, F., Burnett, D., Khan, O., Wheeler, B., Sundararajan, A., Watteyne, T., Vilajosana, X., and Pister, K. Experimental clock calibration on a crystal-free mote-on-a-chip. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (Paris, France, April 2019), IEEE, IEEE, pp. 608–613.

[8] Suciu, I., Maksimovic, F., Wheeler, B., Burnett, D. C., Khan, O., Watteyne, T., Vilajosana, X., and Pister, K. S. Dynamic channel calibration on a crystal-free mote-on-a-chip. *IEEE Access 7* (August 2019), 120884–120900.

[9] Syed, A. S., Sierra-Sosa, D., Kumar, A., and Elmaghraby, A. Iot in smart cities: A survey of technologies, practices and challenges. *Smart Cities 4*, 2 (2021), 429–475.

[10] Wan, J., AAH Al-awlaqi, M., Li, M., O'Grady, M., Gu, X., Wang, J., and Cao, N. Wearable iot enabled real-time health monitoring system. *EURASIP Journal on Wireless Communications and Networking 2018*, 1 (2018), 1–10.

[11] Yuan, T., Maksimovic, F., Burnett, D. C., Wheeler, B., Lee, L., and Pister, K. S. Temperature calibration on a crystal-free mote. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)* (2020), IEEE, pp. 1–5.

[12] Yuan, T., Maksimovic, F., Wheeler, B., Burnett, D. C., Lee, L., Watteyne, T., and Pister, K. S. A temperature-compensated ble beacon and 802.15. 4-to-ble translator on a crystal-free mote. In *2021 51st European Microwave Conference (EuMC)* (London, United Kingdom, June 2022), IEEE, IEEE, pp. 765–768.