Excalibur: An Accurate, Scalable, and Low-Cost Calibration Tool for Sensing Devices

Chia-Chi (Chelsey) Li and Behnam Dezfouli

***Abstract*—Calibration of an analog-to-digital converter is an essential step to compensate for static errors and ensure accurate digital output. In addition, ad-hoc deployments and operations require fault-tolerant IoT devices capable of adapting to unpredictable environments. In this paper, we present the design and implementation of Excalibur – a low-cost, accurate, and scalable calibration tool. Excalibur is a programmable platform, which provides linear current output and rational function voltage output with a dynamic range. The basic idea is to use a set of digital switches to connect with a parallel resistor network and program the digital switches to change the total resistance of the circuit. The total resistance and output frequency of Excalibur is controlled by a program communicating through the GPIO and I^2C interfaces. The software provides two salient features to improve accuracy and reliability: time synchronization and self-calibration. Furthermore, Excalibur is equipped with a temperature sensor to measure the temperature before calibration, and a current sensor which enables current calibration without using a digital multimeter. We present the mathematical model and a solution to compensate for thermal and wire resistance effects and validate scalability by incorporating the concept of the Fibonacci sequence. Our extensive experimental studies show that Excalibur can significantly improve measurement accuracy. For example, for ATMega2560, the ADC error reduces from 0.2% to 0.01%, for ADS8353, the error reduces from 0.17% to 0.0014%, for INA219, the error reduces from 0.42% to 0.02%, and for MCP3208, the error reduces from 5.29% to 0.01%.**

***Index Terms*—Sensor; Accuracy; Measurement; ADC; Internet of Things (IoT); Power Emulation**

I. Introduction

Data conversion accuracy is one of the main requirements for building reliable wireless sensor networks and IoT systems. The fundamental architecture of an intelligent sensor node consists of an *analog-to-digital converter* (ADC), a processor, and a short-range radio. Ad-hoc deployments and operations of the IoT systems require fault-tolerant sensors adaptable to unpredictable environments [1], [2]. For example, a smart temperature sensor with an integrated ADC needs to ensure the error is limited to a range from -0.25 °C to 0.35 °C when the sensor operates between 0 °C to 90 °C [3]. Confidence in the accuracy is crucial for mission-critical applications such as monitoring a tank's temperature, a pipe's pressure in a chemical factory, or vital human signs [4].

Without calibration, for example, a 12-bit ADC results in an absolute error of around six to seven bits of accuracy [5]. Accuracy is dominated by the uncalibrated gain and uncalibrated offset parameters of the ADC. Although most devices are well within their data sheet limits, they do not center around zero. Therefore, ADC always requires calibration to offer a full-range accuracy. The key static specifications that define the accuracy of an ADC are *differential non-linearity* (DNL) and *integral non-linearity* (INL) [6]. DNL describes deviations from the ideal transition voltage in the converter's transfer functions. The deviation from ideal transition is the differential linearity error for that unique code, which translates to *least significant bit* (LSB). INL is the overall shape of the transfer function of ADC, also referred to as *static non-linearity* or *absolute non-linearity* [7]. Non-linearity causes data distortion while the signals are being digitized. By thorough characterization of an ADC's non-linearity and transfer function, we can minimize its distortion. There are a few different approximations of ADC non-linearity commonly used for calibration, such as ordinary polynomials, Chebyshev polynomials, and Fourier series [8]. Therefore, ADC calibration requires characterization of both increasing and decreasing input levels, which require a wide range of samples [9]. Since all components, including ADCs in an analog chain, demonstrate deviations from their reasonable values, a full calibration process is essential.

In addition to non-linearity, another critical element that affects ADC inaccuracy and requires a full calibration process is temperature variations. Furthermore, a common sensing device consists of resistors in the circuit, where the resistance of the resistors varies based on temperature conditions. However, ad-hoc deployments and operations require IoT systems with the capability of adapting to unpredictable environments and offering fault-tolerance under different temperature conditions. Therefore, the calibration process must provide results that are accurate across a wide temperature range.

The common approaches for calibrating an analog input are: (i) using various resistor values [10], [11], (ii) using a potentiometer [12], [13] to generate variable loads, (iii)

Chia-Chi Li is with the Intel Corporation, Santa Clara, CA, and the Internet of Things Research Lab, Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA, USA. E-mail: cli1@scu.edu.

Behnam Dezfouli is with the Internet of Things Research Lab, Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA, USA. E-mail: bdezfouli@scu.edu.

An earlier version of this paper was presented at ICIOT'18 and was published in its proceedings: https://link.springer.com/chapter/10.1007/978-3-319-94370-1_7

creating an extensive simulation environment [3], [14], [15], and (iv) using a commercial power analyzer [16], [17]. However, these calibration mechanisms pose several limitations. Resistors and potentiometers cannot provide quality calibrations due to their limited range and unreliable resistance. Developing extensive simulation environments for each type of purpose require a sophisticated design to achieve the desired accuracy. Furthermore, these simulation environments are designed for specific applications and cannot be reused for various scenarios. Finally, despite providing high accuracy, commercial power analyzers are very costly.

To overcome the above challenges, we proposed in [18] a programmable calibration tool, which provides a wide range of discrete voltage and current outputs. The tool uses a digital potentiometer to generate small changes in the resistance, and a resistor network controlled by the digital switches to enable large resistance jumps. However, since variations of the digital potentiometer's resistance are linear, the circuit generates non-linear current and voltage outputs. Because correction of the ADC non-linearity requires well-characterized curve-fitting, it is essential to provide the linear output over a wide range to achieve the desired accuracy. Furthermore, digital potentiometer has a large temperature coefficient, which results in significant resistance variations versus temperature [19]. Resistance variation is one of the main factors that decreases calibration accuracy. Since the majority of IoT devices operate in non-controlled temperature conditions, it is critical for the calibration tool to sustain high accuracy under various temperatures. Therefore, we need a calibration tool which not only can provide linear current and voltage outputs, but is also low cost and can maintain high accuracy against temperature variations.

In this paper, we present *Excalibur* – a scalable tool for calibrating various analog-to-digital interfaces. Excalibur is a low-cost and programmable tool that provides a wide range of linear current output and rational function voltage output. In particular, the contributions of this work are as follows:

– The basic idea of Excalibur is to use a set of digital switches connected to a parallel resistor network and program the digital switches to change the total resistance of the circuit. We reduce resistance variations by introducing low-temperature coefficient resistors instead of digital potentiometers, thus increasing calibration accuracy. Besides, by using digital switches with assigned resistor values, we ensure the step size of the current output is fixed and uniformed.
– The scalability of Excalibur's outputs supports a broad range of calibration requirements. In other words, the range of the digital output matches the required scale of analog input. For example, assume that an ADC can measure between 0 V to 5 V, and the light sensor connected to the ADC has a sensing range between 0 V to 3 V. This creates a void from 4 V to 5 V. Therefore, the digital output range needs to map to the analog input range between 0 V to 3 V to maximize calibration resolution. In addition, Excalibur's output range can be scaled up, and the resolution can be reduced by adding more switches and resistors.
For example, adding a digital switch with four 50 Ω resistors increases the total output current by 400 mA, and adding a digital switch with four 25 kΩ improves the resolution to 0.2 mA. To show the scalability of Excalibur, we incorporate the Fibonacci sequence to formulate a mathematical model of scalability.
– Excalibur offers self-calibration, which is essential to compensate for wire resistance and total resistance differences under various temperature conditions. Excalibur includes a set of pre-calibrated measurement arrays and a set of temperature error correction functions. The arrays are used to compensate for the impact of wire resistance on the system, and the temperature correction functions are used to correct the influence of temperature on output performance. When the calibration process starts, the software first measures the ambient temperature using a temperature sensor. Based on the sensor's reading, the corresponding temperature error correction function is selected and applied to the pre-calibrated measurement array to adjust the output.
– Excalibur provides a time synchronization mechanism which eliminates the need for accurate time stamp matching. This feature notably simplifies the calibration process because various IoT devices present different warm-up times and sampling frequencies, depending on factors such as ADC type, processor speed, and available memory. For example, the sampling rate of an ADC depends on its serial interface, device driver, and programming techniques used. Therefore, when Excalibur simultaneously triggers measurement equipment and an IoT device to sample their analog inputs, we do not need to worry about the exact start times and sampling rates when matching the pairwise values.
– We have implemented a prototype of Excalibur and conducted an extensive set of experimental evaluations to measure its performances. Our results show that Excalibur achieves quality measurement fidelity under a wide range of operations.
In particular, our prototype produces a dynamic current range from 0 mA to 600 mA and dynamic voltage range from 0.6 mV to 5 V. The minimum resolutions of our prototype in terms of current and voltage are 1 mA and 0.1 mV, respectively. We also present case studies for calibrating high-resolution ADCs. Our calibrations show improvements across different devices. For example, the error reduction of various devices are as follows: ATMega2560: from 0.2% to 0.01%, ADS8353: from 0.17% to 0.0014%, MCP3208: from 5.29% to 0.01%, and INA219: from 0.42% to 0.02%. The completed Excalibur board only costs about $120, which is less than 2% of the cost compared to current commercial solutions.

The rest of the paper is organized as follows: Section II presents the design challenges of scalable calibration, as well as the design, mathematical modeling, and implementation of Excalibur. We present the evaluations of Excalibur in Section III. Section IV discusses the related work. We conclude the paper in Section V.

II. Design Challenges, Considerations, and Implementation

The primary objective of designing a calibration tool is to calibrate the analog inputs and outputs of IoT devices such as the ADC or a digital-to-analog converter (DAC). Designing a scalable calibration tool imposes the following challenges and requirements: **(i) Non-linear distortion:** The power calibration tool must support high-resolution current and voltage change steps to provide full coverage for data converter characterization. **(ii) Large dynamic range:** The power calibration tool must provide current and voltage samples over a dynamic range to span the input range of various types of IoT devices. **(iii) Timing synchronization:** A time synchronization mechanism is necessary to correlate each current and voltage value generated by the calibration tool and the value sampled by the target device. **(iv) Temperature variations:** The total resistance of the circuitry varies based on the device and ambient temperature. Ambient temperature is the air temperature that surrounds the circuitry, and device temperature is the temperature of the circuit's components. The power consumption of the circuitry generates heat and results in a device temperature higher than the surrounding ambient temperature. Temperature changes result in resistance variation, thus affecting the output accuracy of the calibration tool. **(v) Wire resistance:** Wire resistance is the total amount of resistance on the circuit path. The wire resistance in the circuit adds extra resistance and impacts the output performance. Some of the main factors that contribute towards path resistance are the total length of the wires, temperature, number of cross-section paths, and wire material. **(vi) Portability and cost:** Compared to high-cost commercial power analyzers, the tool must be easy to integrate, both electronically and mechanically. Also, it should be low cost and easy to use.

A. Design Considerations and Solutions

In this section, we propose solutions to the above challenges and present our design. We formulate mathematical models to show the impact of temperature and wire resistance on calibration outputs. We also prove that the operating range of Excalibur is scalable. We then present the hardware and software implementation of this platform. Table I summarizes the key notations.

1) Calibration Output Linearity: The essential objective of a calibration tool is to generate variable loads to offer pre-determined current and voltage outputs. Digital potentiometers can satisfy this objective and provide a range of resistance to generate various loads [20]. The resistance value for a digital potentiometer is $R_p(\sigma) = \frac{\sigma}{2^b} \times R_{max} + R_w, \ \ 0 \leq \sigma \leq 2^b$, where b is the number of bits supported by the potentiometer, σ is the value programmed into the potentiometer, and R_w is the wiper resistance. Output resistance $R_p(\sigma)$ is proportional to the digital potentiometer's end-to-end resistance R_{max}. Base on Ohm's law, when the resistance variations of a circuit is linear, it generates non-linear current and voltage outputs, which also implies the step size of the output is not constant.

Table I
Key Notations

Notation	Description
T_a	Ambient temperature
I_{total}	Target current output value for each iteration
$I_{50}[\psi]$	1-D current value array when 50 Ω resistors are enabled
ψ	Number of 50 Ω resistors enabled
$I_{250}[\psi][\zeta]$	2-D current value array when 250 Ω and 50 Ω resistors enabled
ζ	Number of 250 Ω resistors enabled
$I_{1k\Omega}[\psi][\xi]$	2-D current value array when 1 kΩ and 50 Ω resistors are enabled
ξ	Number of 1 kΩ resistors enabled
$I_{5k\Omega}[\psi][\nu]$	2-D current value array when 5 kΩ and 50 Ω resistors are enabled
ν	Number of 5 kΩ resistors enabled
$f_{T_a}(\theta)$	Temperature error correction function
ρ	Period of output changing events
pt	Resistor pointer that identifies which resistor to connect or disconnect
pt_{250}	Resistor pointer that identifies which 250 Ω resistor to connect or disconnect
pt_{50}	Resistor pointer that identifies which 50 Ω resistor to connect or disconnect
pt_{1k}	Resistor pointer that identifies which 1 kΩ resistor to connect or disconnect
pt_{5k}	Resistor pointer that identifies which 5 kΩ resistor to connect or disconnect
pts	Array including the 50 Ω, 250 Ω, 1 kΩ, and 5 kΩ pointers
R_t	Array of 50 Ω, 250 Ω, 1 kΩ, and 5 kΩ resistor's GPIO pin numbers
I	Current output value
I_{max}	Maximum target current value
I_{pre}	Current output value of previous iteration

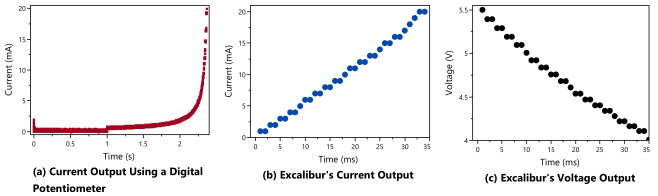


Figure 1. Linearity comparison between a digital potentiometer and a digital switch. (a) current output using a digital potentiometer AD5200. (b) and (c) show current and voltage, respectively, when using ADG1612 digital switches.

Figure 1 (a) shows the experimental results of current output using a digital potentiometer AD5200 [19]. The non-constant step size means the potentiometer generates smaller output changes when it starts at the maximum resistance. It demonstrates more substantial output changes once it reaches its minimum resistance. Because correcting ADC non-linearity requires well-characterized curve-fitting, it is essential to provide a linear output over a wide range to achieve the desired accuracy. Besides, calibration quality significantly drops for those sensors that operate within the ranges that have a reduced number of steps.

To address the above issues, we replace the digital potentiometer with two ADG1612 digital switches [21]. One is connected to four 5 kΩ high-precision resistors, and the other is connected to four 1 kΩ high-precision resistors.

These high-precision resistors are part of the variable resistor network, which can provide 1 mA and 5 mA change in current. Figures 1 (b) and (c) illustrate the experimental results of replacing the digital potentiometer with two digital switches in terms of current and voltage, respectively. The current output becomes linear and the step sizes are evenly distributed throughout the entire range, which dramatically improves calibration accuracy.

2) Volume Steps with Dynamic Output Range: The standard analog to digital conversion has an analog input $V_{in}(t)$ and a digital output $D_{out}(t)$, where t represents time. The purpose of voltage calibration is to find a function $f(d)$ to implement in the correction module, where d is a digital output. Function $f(d)$ is to minimize the error between the corrected voltage output $f(D_{out}(t))$ and ground truth measurement $V_{in}(t)$. Equation $LSB = \Delta = \frac{V_{FS}}{2^r}$ defines the minimum resolution of the ADC per digital bit (LSB), where V_{FS} is the full-scale range of the ADC determined by the reference voltage, and r is the maximum number of bits the input value can be translated to. Equation 1 defines the final digital output value of the ADC,

$$D_{out}(t) = \left\lfloor \frac{V_{in}(t)}{\Delta} \right\rfloor \Bigg|_{t=r\times\rho_s} = \left\lfloor 2^r \times \frac{V_{in}(r \times \rho_s)}{V_{FS}} \right\rfloor \quad (1)$$

where ρ_s is the sampling period. Based on this equation, normalization and truncation of the analog inputs are involved in the ADC quantization process. The maximum instantaneous value of distortion is $\frac{LSB}{2}$, and the total variation range is $-\frac{LSB}{2}$ to $+\frac{LSB}{2}$. In addition to quantization errors, there are two static specifications that define the accuracy of conversion: *differential non-linearity* (DNL), and *integral non-linearity* (INL). DNL error is the difference between an actual step width and the ideal value of 1 LSB. INL error is the deviation, in LSB or percentage of the full-scale range, of an actual transfer function from ideal, straight conversion line. More importantly, the magnitude of INL directly depends on the correlation position chosen for the ideal straight line. Since $f(d)$ is non-linear, it is crucial to characterize ADC's full scale instead of only a few positions.

To overcome this challenge, we connect a resistor network in parallel with the digital switches to increase the total current flow. The total current output of the parallel resistor network can support is

$$I_{total}(t) = \sum_{j=1}^{n} I_j(t) \quad \forall n \in N \quad (2)$$

where $I_j(t)$ is the current that a resistor R_j belonging to the resistor network carries. For any positive integer n, we can expand the current output range $I_{total}(t)$ to satisfy any calibration range requirement as long as the given input voltage is within the operational range of the circuit components. The capability to expand the output range proves the scalability of this platform in terms of current. Similarly, for voltage scalability, we connect a resistor R_v with the resistor network in serial. Based on Ohm's law, Excalibur's voltage output range is,

$$V_{total}(t) = I_{total}(t) \times R_v \quad (3)$$

With the capability to expand the $I_{total}(t)$ range, we prove the voltage scalability of Excalibur.

In addition to scalability, we want Excalibur's current output to be linear. Because ADC non-linearity requires curve-fitting to achieve accuracy, we can improve calibration accuracy by providing a linear output over a wide range. To generate a linear output, $I_j(t)$ should have a linear relationship with $I_{j-1}(t)$. The linear relationship between $I_j(t)$ and $I_{j-1}(t)$ is,

$$I_j(t) = k \times I_{j-1}(t) \quad \forall j, k \in N \quad (4)$$

To support gradual increase of output current, $I_j(t)$ can be expressed as follows,

$$I_j(t) = I_{j-1}(t) + I_{j-2}(t) \quad (5)$$

also known as the Fibonacci sequence. Therefore, we use induction to prove the maximum current is $\sum_{j=1}^{n} I_j(t) = I_{n+2}(t) - 1$, where $\forall n \geq 2$ and $n, j \in N$.

With the mathematical model proven, we implement our prototype with $n = 3$. However, based on Equation 2, scalability is not necessarily limited to $n = 3$. To increase the number of current jumps, we use ADG1612 digital switches to control the number of resistors in the resistor network that are connected to the circuit.

One of the digital switches connects with four 5 kΩ resistors to generate a 1 mA minimum current. The reason behind selecting the 5 kΩ resistor is to support 1 mA current steps. We choose $I_2(t) = 20$, $I_3(t) = 80$, and $I_4(t) = 100$ to implement our prototype. In this case, $I_4(t) = I_3(t) + I_2(t)$ satisfies Equation 5, $I_3(t) = 4 \times I_2(t)$, and $I_4(t) = 5 \times I_2(t)$ satisfies the linearity relationship of Equation 4. After reaching the maximum current of the first digital switch, a 1 kΩ resistor on the resistor network (controlled by the second digital switch) is enabled to provide additional current. Repeating this process results in gradually increasing the current by enabling additional resistors until reaching Excalibur's maximum current output. Based on Equation 3, a similar analysis can be employed for voltage.

3) Time Synchronization: The differences in the sampling offset and sampling rates of the measurement device and the target device make correlating the readings a challenging task. For example, when configuring an output value, the ADC might miss that value due to its lower sampling rate compared to measurement devices such as DMM and current sensor. If the design does not provide enough duration during two consecutive output changes, the ADC and the measurement device may collect samples belonging to different output values. Therefore, we need to ensure that both the measurement device and the target device sample the output at least once during a time interval. The next challenge is

Table II
RESISTANCE VALUES BASED ON TEMPERATURE COEFFICIENT α UNDER DIFFERENT TEMPERATURES. THE CURRENT OUTPUT AT 55 °C IS LOWER THAN THE OUTPUT AT 15 °C.

Resistor Value	Material	α	R @ 15 °C	R @ 55 °C
5 kΩ	Thin Film	± 5 ppm/°C	4.999 kΩ	5.001 kΩ
1 kΩ	Thin Film	± 5 ppm/°C	1.002 kΩ	0.9998 kΩ
250 Ω	Thin Film	± 5 ppm/°C	249.95 Ω	250.05 kΩ
50 Ω	Wire-wound	± 20 ppm/°C	49.96 Ω	50.04 kΩ

to ensure that when we compare two sampled values, the samples are collected when the output is stable and at the same time interval. To overcome these challenges, Excalibur stores the voltage or current settling time information after applying each configuration that results in an output change. For example, assuming the minimum sampling rate of the ADC and target device is r_{min}, we first ensure that $\rho > \frac{1}{r_{min}}$, where ρ is the period of the output change. Second, assuming that a new output value is configured at time t_i, to correlate the values collected by DMM and target IoT device, we correlate the readings with values that are closest to time $t_i + \frac{\rho}{2}$. This synchronization approach ensures the two values belong to the same output value.

4) Accuracy Impact on Thermal Effect: Since Excalibur consists of resistors and capacitors, its accuracy is limited by static and dynamic power dissipation. Dynamic power is the main contributor to the power dissipation [22]. Specifically, dynamic power consumption is proportional to switching frequency: $P_{dyn} \propto freq$. Excalibur uses digital switches frequently to switch resistor terminals, which causes the resistor path to connect and disconnect to the power source. The switching activities result in the charging and discharging of the capacitor connected to the resistor. Therefore, the frequency of charging and discharging is the primary source of Excalibur's dynamic power dissipation.

To charge a capacitance C, we applied a voltage step V as a source of energy. The total amount of energy $E_{0\to1}$ to charge a capacitance is,

$$E_{0\to1} = \int_0^\infty VC\frac{dV_C}{dt}dt = CV\int_0^V dV_C = CV^2 \qquad (6)$$

When charging the capacitor, half of that energy (i.e., $\frac{1}{2}CV^2$) is stored in the capacitor, while the other half dissipates as heat by the resistance of the RC circuit. While discharging, the stored energy dissipates as heat as well. Heat dissipation during charging and discharging increases the resistor temperature, making it higher than the ambient temperature. The amount of temperature increase from ambient temperature is,

$$T_\Delta = P_{dyn} \times R_\lambda \qquad (7)$$

where R_λ is thermal resistance of resistor. Besides, the resistance of the resistors changes based on temperature [23]. Resistor values (R_τ) at a particular temperature T can be expressed as follows,

$$R_\tau = R_{ref}[1 + \alpha(T - T_{ref})] \qquad (8)$$

where R_{ref} is resistance at reference temperature, α is temperature coefficient, T is current resistor temperature, and T_{ref} is reference temperature usually at 20 °C. Based on Equation 7 and Equation 8, we can express the relationship between the resistor value and temperature as follows,

$$\begin{aligned} R_\tau &= R_{ref}[1 + \alpha((T_a + T_\Delta) - T_{ref})] \\ &= R_{ref}\{1 + \alpha[T_a + (\frac{1}{2}CV^2 \times R_\lambda)] - T_{ref}\} \end{aligned} \qquad (9)$$

where T_a is the ambient temperature. Similarly, for current input analysis, we apply a fixed current source as a source of energy to charge a capacitor. The total amount of energy taken from supply to charge a capacitance is $E_{0\to1} = E_C + E_R$, where E_C is the energy stored at capacitor, and E_R is energy dissipated by the resistor. The total energy stored in capacitor can be expressed as follows,

$$E_C = \int_0^\infty I_s(RI_s)dt = RI^2t_c = (\frac{RC}{t_c})CV^2 \qquad (10)$$

where t_c is the charging time of the capacitor, and $t_c = \frac{CV}{I}$. Therefore, the voltage output rises linearly with time. Equation 10 shows that energy dissipation in the resistor can be reduced by increasing charging time t_c or decreasing the supply current I_s. Therefore, accuracy is affected by the level of supply current and the size of the capacitor.

Pure metal with a positive α indicates that resistance increases when temperature increases. The resistance decreases with increasing temperature with other materials such as carbon, silicon, and germanium. The relationship of resistance and temperature for a metal resistor fits tightly to a quadratic polynomial equation [24]. Most resistor data sheets indicate two temperature coefficients α: a positive value for coefficient above reference temperature T_{ref}, and a negative value for under T_{ref}. Table II lists all the high-precision resistors we used in the prototype design. Excalibur uses metal thin film resistors [25] as part of the small current generation process and wire-wound resistors [26] to provide stability for larger current. Table II uses Equation 8 to calculate the resistance value under different temperatures. We expect to see noticeable current variations on the 50 Ω resistor path since the temperature coefficient is ± 20 ppm/°C [27].

Equation 9 indicates that accuracy can be affected by the level of the voltage supply, temperature coefficient, and material thermal resistance. Specifically, increasing the supply voltage level and test temperature reduce accuracy. To provide a high accuracy output, it is crucial to calculate the accuracy impact of the temperature. Failure to consider this impact causes the outputs to demonstrate gaps between output steps, thus providing lower than desired output range and unexpected current or voltage drop.

Table III
OUTPUT CURRENT AT 25 °C IS USED AS A BASELINE TO CALCULATE TEMPERATURE ERROR CORRECTION FUNCTIONS. A POSITIVE ABSOLUTE ERROR PERCENTAGE INDICATES THE CURRENT IS HIGHER THAN 25 °C. ABSOLUTE ERROR WAS MEASURED IN MA. θ IS THE CURRENT VALUE AT 25 °C.

T_a	Abs. Error (%)	Temperature Error Correction Function
15	0.63%	$f_{15}(\theta) = -0.003125 + 0.0000329 \times \theta + 2.7027e^{-8} \times (\theta - 312.886)^2$
35	-1.45%	$f_{35}(\theta) = 0.0056552 - 8.0728e^{-5} \times \theta - 3.8828e^{-8} \times (\theta - 312.886)^2$
45	-5.54%	$f_{45}(\theta) = -0.02355 - 0.0001225 \times \theta + 4.6587e^{-7} \times (\theta - 266.002)^2$
55	-6.3%	$f_{55}(\theta) = -0.014299 - 0.0001126 \times \theta + 2.917e^{-7} \times (\theta - 266.959)^2$

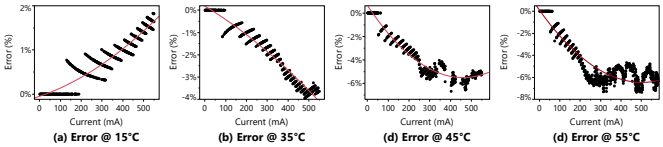


Figure 2. Output current error compared with output current measured at 25 °C Output current error at (a) 15 °C, (b) 35 °C, (c) 45 °C and (d) 55 °C.

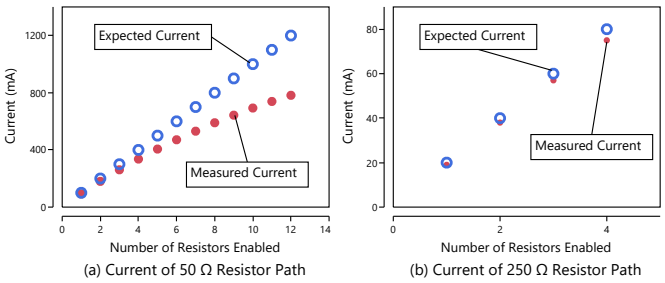


Figure 3. Theoretical and empirical current output values. We can observe that output current degrades when more resistor paths are enabled. (a) Theoretical and empirical output current values versus measured current value on 50 Ω, and (b) 250 Ω resistor paths.

To overcome this challenge, we use an ultra high-precision DMM (DMM7510 [28]) to measure Excalibur's current output under different temperature conditions and compare the results with the current output measured at 25 °C. The current output is measured after Excalibur is placed into a stable temperature chamber with the temperature set for 30 minutes. Figure 2 shows the current output error under four different temperatures: 15 °C, 35 °C, 45 °C, and 55 °C. If the absolute error percentage is positive, it indicates the output at that temperature is higher than the output at 25 °C, and a negative value indicates that the output at that temperature is lower than the output at 25 °C.

Table III shows the absolute error increases versus temperature. Since the relationship of resistance and temperature fits closely to the quadratic function [24], we use polynomial regression to model the temperature error correction function as shown in Table III. The reason we use the polynomial regression instead of Equation 10 is that the system has variable wire resistance under different output configurations, which increases the complexity of calculating the temperature effect. The difference in wire resistance also impacts the final current performance. The simplest way to model the effect of temperature on current is to measure current output (while the device is inside a stable temperature chamber) and use polynomial regression. Excalibur collects the ambient temperature before the calibration and finds the closest predefined temperature condition. It then applies the temperature error correction function corresponding to the predefined temperature condition to the output calculation. By applying the temperature error correction equation we can offer a stable and linear output under different temperature conditions. The same process applies to voltage output calibration.

5) Accuracy Impact on Wire Resistance: The resistance of the circuit path is one of the critical parameters when determining calibration system performance. Wire resistance depends on four main factors: material resistivity, length of the wire, cross-sectional area, and wire temperature [29]. Wire resistance can be represented as $R_{wire} = \frac{R_\rho l}{A}$, where l is the length of the wire, A is the area of the wire, and R_ρ is the material resistivity of the wire. Based on Ohm's law, a single resistor path's current is $I(t) = \frac{V(t)}{R+R_{wire}}$.

When the resistor value is large, for example, 1 kΩ, the impact of R_{wire} on current is low. On the other hand, when the resistor value is small, for example, 50 Ω, the impact of R_{wire} on current is significant. Since we use digital switches to connect and disconnect the resistor path to a power source, the total wire length varies in each step, which in turn results in frequent wire resistance changes.

Figure 3 presents a comparison between expected and measured current values versus the number of enabled resistor paths. As we can see, when the number of enabled resistors increases, the gap between the theoretical and empirical output current increases significantly on the 50 Ω resistor path. This is because the increase in wire resistance causes a bigger impact on the low resistive path. As shown in Figure 3, the impact of the 250 Ω resistor path is smaller compared to the 50 Ω.

To minimize the complexity of calculating R_{wire}, we use a high accuracy DMM to measure the wire resistance impact of different resistor path combinations. The Excalibur prototype includes twelve 50 Ω, four 250 Ω, four 1 kΩ, and four 5 kΩ resistors. We first measure the current output value with only one 50 Ω resistor enabled, then with two 50 Ω resistors enabled, and repeat the process until all the twelve 50 Ω resistors are enabled. All measurement values are saved into a one-dimensional (1-D) array $I_{50}[\psi]$, where ψ represents the number of 50 Ω resistors enabled for that current value. $\psi \in N$ and $0 \leq \psi \leq 12$, where $\psi = 0$ represents no 50 Ω resistor is enabled. We repeat the same process for measuring current value by enabling the 250 Ω resistors with the 50 Ω resistors enabled simultaneously. The measured current values are entered into a two-dimensional (2-D) array $I_{250}[\psi][\zeta]$, where ζ represents the number of the 250 Ω resistors enabled, and ψ represents the number of 50 Ω resistors enabled for that current value, $\zeta \in N$ and

$0 \leq \zeta \leq 4$. For example, $I_{250}[2][3]$ represents the current value when two 50 Ω and three 250 Ω resistor paths are enabled. The same process repeats for creating 2-D arrays $I_{1\text{k}}[\psi][\xi]$ and $I_{5\text{k}}[\psi][\nu]$, where ξ represents the number of 1 kΩ, and ν represents the number of 5 kΩ resistors enabled for that current value, $\xi, \nu \in N$ and $0 \leq \xi, \nu \leq 4$.

After calibrating R_{wire}, we use these calibration arrays to accurately calculate the expected current output. Instead of using a linear algorithm to traverse through resistor combinations [18], we combine I_{50} and I_{250} to generate a custom loop based on the calibrated current value. The custom loop reduces the computation time and provides a linear performance for the current output. Section III compares the results between the linear algorithm and our proposed methodology. Current output using the linear algorithm shows large jumps and gaps between steps, while the proposed algorithm with pre-calibrated arrays shows smooth and linear results. The same process of generating pre-calibrated arrays can also be used for voltage calibration.

B. Hardware Implementation

Excalibur's hardware consists of seven main units: (i) micro-controller unit (MCU), (ii) resistance control unit (RCU), (iii) variable resistor network, (iv) output selection jumpers, (v) digital temperature sensor (MCP9808) [30], (vi) current sensor (INA219 [31]), and (vii) measurement selection jumper. MCU is used to control the RCU, synchronize the operation of the target and measurement device, read and collect ambient temperature, and record the settling time of each output value generated. RCU is responsible for controlling the variable resistor network. The MCP9808 is used to the collect ambient temperature before calibration. The INA219 is used to collect current measurement during calibration. The measurement selection jumpers allow the user to enable the temperature and current sensor. The output selection jumpers enable the user to select between current or voltage outputs.

Figure 4 presents the block diagram of Excalibur. We implement MCU functions by developing a Python configuration code running on a Raspberry Pi 3 (RPi) board. RPi acts as a micro-controller unit which connects to the RCU through the GPIO interface and controls the RCU to perform the desired changes. To control the start and stop of sampling, MCU also provides trigger signals through the GPIO to the target device and external DMM.

RCU contains six ADG1612 digital switches to control the total resistance of the circuit. The various resistance values provide a full calibration coverage, from low sleeping current to a high active current of IoT devices, as proved in Section II-A2.

The current leakage of ADG1612 is nearly negligible under junction temperature of 80 °C, which is crucial to the stability against temperature influence [21]. In addition, AD1612 is capable of 8 MHz switching frequency under

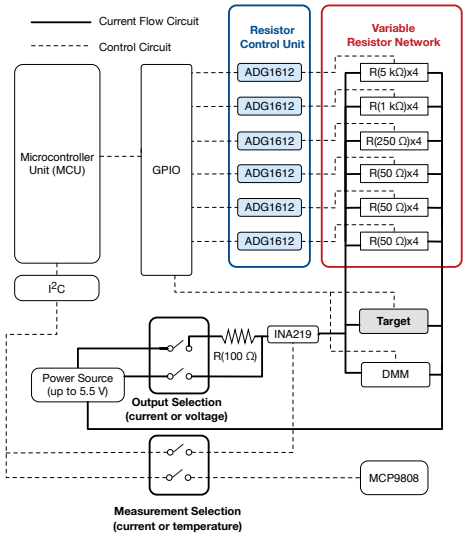


Figure 4. The block diagram of Excalibur. MCU provides synchronized trigger signals to the target ADC, DMM, and INA219. MCU also measures temperature and current. The ADG1612 digital switches control the resistor network.

80 °C and 5 V supply. These features enable us to provide stable current and voltage outputs within 1 μs period.

The variable resistor network is a circuit connecting 24 resistors in parallel, which is controlled by the RCU's ADG1612 ICs. The ADG1612 digital switch contains four terminals and each terminal is connected to a resistor. When a terminal receives a logic 1 signal, the resistor that is controlled by the terminal connects to the variable resistor network, which increases the total current flow. If the terminal receives a logic 0 signal, the resistor is disconnected from the variable resistor network, which reduces the total current flow. There are six ADG1612s used in the RCU: one is connected to four 5 kΩ, one is connected to four 1 kΩ, one is connected to four 250 Ω, and the other three are connected to four 50 Ω high-precision resistors.

These high-precision resistors connected to the ADG1612s are part of the variable resistor network, which can provide 1 mA, 5 mA, 20 mA and 100 mA changes in current. It is worth noting that the effect of the ADG1612s on the current draw is minimal, as it only shows a 1 Ω resistance when operating.

The output selection jumpers are used to switch between the voltage and current outputs. When voltage output is selected, the output selection jumpers enable a high-power 100 Ω resistor to connect in serial with the resistor network. It provides the same dynamic range for the voltage change from 0 V to 5 V as proved in Section II-A2. Excalibur generates trigger signals through the GPIO pins to start and stop simultaneous data collection of the target ADC and measurement devices (i.e., INA219 or DMM). To improve stability and remove the alternating current caused by voltage ripple, we add low pass filters to the resistor network.

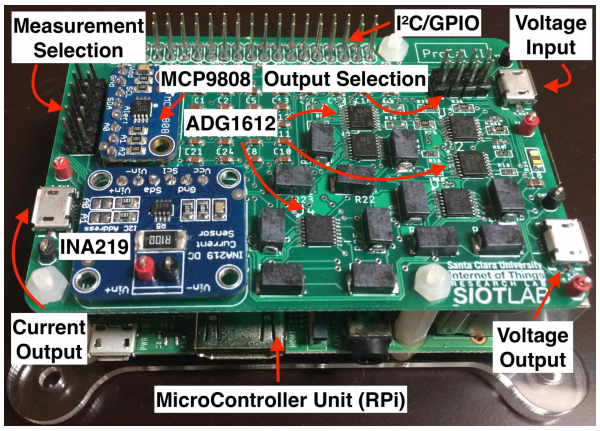


Figure 5. Prototype of Excalibur. The proposed tool is a shielded board installed on top of a RPi. The RPi communicates with Excalibur through I^2C and GPIO interfaces.

The digital temperature sensor (i.e., MCP9808) collects the ambient temperature to select the corresponding temperature error correction functions $f_{T_a}(\theta)$ to compensate the impact of temperature drift on the resistor network. Function $f_{T_a}(\theta)$ is a set of four error correction functions which have been measured under four different temperatures: 15 °C, 35 °C, 45 °C, and 55 °C. Based on the ambient temperature, Excalibur switches to the closest corresponding $f_{T_a}(\theta)$ to compensate the impact of temperature drift. To reduce the board's power consumption, the digital temperature sensor can also be turned off by using the measurement selection jumpers.

The current sensor (i.e., INA219) collects current or voltage measurements through internal connections on the board. With the internal op-amp's gain set on the INA219 board, the maximum current measurement it can support is ± 400 mA and the resolution is 0.1 mA [31]. By introducing the INA219, the proposed tool adds the capability to calibrate low power sensors when there is no DMM available. For example, for electrocardiography (ECG) sensors whose measurement range is around a few mV [32], Excalibur can be used for calibration without using a DMM. When not required, the measurement selection jumpers enable the user to deactivate the INA219 and reduce the board's power consumption.

Figure 5 shows the prototype of Excalibur. The total cost of the complete Excalibur PCB assembly is estimated to be about $120, including the manufacturing costs as well. Compared to current commercial power analyzers which cost a minimum of $14,000 [33], our solution provides substantial cost reduction, scalability, portability, improved accuracy, and ease of integration.

C. Software Implementation

Excalibur's software runs on a flavor of Linux operating system called Raspbian and uses the BCM library [34] to control the GPIO pins and I^2C bus. The software controls the operations of the resistor network which includes setting up the I^2C and GPIO drivers, accurate time stamping of resistance change events, collecting temperature and measurement results, and sending synchronization signals. It uses a collection of Python scripts that are used by the MCU to trigger scheduled actions to start and stop the calibration process. A set of current arrays I_{50}, I_{250}, $I_{1\text{k}}$, and $I_{5\text{k}}$ are stored in the software, which are pre-calibrated with the system's wire resistance and the temperature error correction functions. When more and more resistors are enabled on the parallel resistor network, the wire resistance increases, as shown in Section II-A5. Also, as we proved in Section II-A4, the resistance value of circuitry varies under different temperature conditions. The software applies the corresponding temperature error correction functions $f_{T_a}(\theta)$ based on the ambient temperature value measured before calibration.

At a high level, the software consists of two major components: initialization functions and a nested recursive function. Initialization functions initialize the hardware connections and variables. A nested recursive function is used to enable resistors to achieve target current value recursively. To explain the operation of the software, we broke it into Algorithm 1, which is the main configuration algorithm, Algorithm 2, which shows the utility functions, and Algorithm 3, which shows the resistor control functions. We explain these algorithms as follows.

As shown in Algorithm 1, line 2, and Algorithm 2, line 2, the controller software first initializes all GPIO pins, disconnects all resistors from the resistor networks, and sets up the I^2C driver. In Algorithm 2, line 6, all pointers pt in pointer array pts are initialized to 0 and all the resistor's GPIO pins in the resistor array R_t are disconnected. Users need to provide: (i) the desired time interval between configurations ρ, and (ii) the maximum current output I_{max}. Excalibur uses MCP9808 to measure ambient temperature T_a before the calibration process as shown in Algorithm 2, line 3.

The `generate_current_array()` function first initializes the index for I_{total} in Algorithm 2, line 8. Based on the temperature T_a measured, the target current value is adjusted to compensate current based on the temperature error correction function $f_{T_a}(\theta)$. T_a rounds up to the closest of one of the four predefined temperature points to apply the corresponding $f_{T_a}(\theta)$ in Algorithm 2, line 12. The I_{total} calibration array is generated by combining both $I_{50}[\psi]$ and $I_{250}[\psi][\zeta]$ arrays as the target current value for each iteration of Algorithm 1, line 3 to be achieved.

Before the start of each T_{total} iteration, Algorithm 1, line 6, initializes $breakflag$ to *FALSE*. During each iteration, the current value starts to increase from the inner loop which enables the 5 kΩ resistors on the resistor network one at a time as shown in Algorithm 1, line 24 and Algorithm 3, line 25. In Algorithm 3, line 26, the software connects each resistor to the network by enabling the GPIO pin

Algorithm 1: Configuration Algorithm

inputs: ρ: time interval
I_{max}: maximum target current value

```
1  function main(ρ, I_max)
2      initialization();
3      I_total = generate_current_array();
       /* generate target current value for each
          iteration                                   */
4      send start trigger signals through GPIO;
       /* signal to all connected devices             */
5      for i ← 0 to size of I_total do
6          breakflag = FALSE;
7          if I == I_max then
8              break;
9          for j ← pts[pt_50] to ψ do
               /* pts[pt_50] = (the number of enabled 50 Ω
                  resistors) + 1                      */
10             if breakflag == TRUE or I == I_max then
11                 break;
12             if I_pre == (I_50[j + 1] + 1) then
13                 en_50Ω();
14             for k ← pts[pt_250] to ζ do
                   /* pts[pt_250] = (the number of enabled 250
                      Ω resistors) + 1                */
15                 if breakflag == TRUE or I == I_max then
16                     break;
17                 if k != 0 and I_total[i] > I_250[pt_50][1] then
18                     breakflag = en_250Ω();
19                 for l ← pts[pt_1k] to ξ do
                       /* pts[pt_1k] = (the number of enabled 1
                          kΩ resistors) + 1            */
20                     if breakflag == TRUE or I == I_max then
21                         break;
22                     if l != 0 and I_total[i] > I_1k[pt_50][1] then
23                         breakflag = en_1kΩ();
24                     for m ← pts[pt_5k] to ν do
                           /* pts[pt_1k] = (the number of enabled 5
                              kΩ resistors) + 1        */
25                         if breakflag == TRUE or I == I_max then
26                             break;
27                         breakflag = en_5kΩ();
28     send stop trigger signals through GPIO;
29     return;
```

Algorithm 2: Utility Functions used by the Configuration Algorithm

```
1  function initialization()
2      initialize GPIO, disconnect all resistors, and set up I²C driver;
3      connect MCP9808 to measure T_a;
4      pts = [pt_5k, pt_1k, pt_250, pt_50];
5      R_t = [5kΩ[ν], 1kΩ[ξ], 250Ω[ζ], 50Ω[ψ]];
6      initialize pts = 0, R_t = disconnect, and I = 0;
7  function generate_current_array(I_250[ψ][ζ],I_50[ψ],T_a)
8      initialize index to 0 for I_total array;
9      for i ← 0 to number of 50 Ω resistors do
10         for j ← 0 to number of 250 Ω resistors do
11             θ = I_50[i] + I_250[i][j] -1;
               /* θ is target current number for each
                  iteration                           */
12             T_a rounds up to the closest measured temperature point;
13             I_total[index] = f_{T_a}(θ);
14             index = index +1;
15     return I_total;
16 function time_stamps(ρ, I)
17     delay for ρ/2;
18     Time stamps ρ/2 and I into a log file;
19     delay for ρ/2;
20 function reset_pointer(pt,resistor,pts,R_t)
21     index_1 = pt's index in pts;
22     pts[:index_1] = 0;
23     index_2 = resistor's index in R_t;
24     R_t[:index_2] = disconnect;
25 function increment_pointer(pt,pts)
26     index_1 = pt's index in pts;
27     index_2 = pt_250 index in pts;
28     for q ← index_1 to index_2 do
           /* increment nested loop's pointers        */
29         if pts[q] does not point to the last resistor then
30             pts[q] ++;
31         else
32             pts[q] = 0;
```

indicated by $R_t[5\text{k}\Omega[m]]$, where m is the mth 5 kΩ resistor GPIO pin number. Current value I is the sum of currents introduced by the enabled resistors during that iteration as shown in Algorithm 3, line 27. The `time_stamps()` function in Algorithm 2, line 18 and Algorithm 3, line 28, time stamps the current output values I and saves into a log file. This log file is used for time synchronization between the target device and measurement device. In Algorithm 3, line 33, if current value I reaches the iteration's target current value $I_{total}[i]$, the algorithm increments the pointers, sets $breakflag$ to *TRUE*, and saves I as I_{pre}. In Algorithm 3, line 29, if $I + 1$ has the same value as the next iteration of I_{50} current value, the algorithm sets $breakflag$ to *TRUE*, saves I as I_{pre}, and resets the $pt_{5\text{k}}$ pointer. In Algorithm 2, line 25 and Algorithm 3, line 33, the `increment_pointer_function()` is used to increment the identified pointer number by one and reset to zero when it reaches the last resistor position. If the current value in the next iteration is enough to enable one extra 50 Ω resistor, the algorithm will break the loop and set $pt[pt_{5\text{k}}]$ to 0 as shown in Algorithm 3, line 30. The `reset_pointer()` function in Algorithm 3, line 2, 8, and 17, and Algorithm 2, line 20 is for resetting the pointers and disconnecting the resistor terminals from the network. After enabling all the 5 kΩ resistors, the algorithm enables the 1 kΩ, 250 Ω, and 50 Ω resistors as shown in Algorithm 1, line 23, 18, and 13, until the target current $I_{total}[i]$ in each iteration is reached.

The software configuration for generating current output is the same as generating voltage output. Excalibur can support output frequency (i.e., changes of current/voltage values) up

Algorithm 3: Resistor Control Functions used by the Configuration Algorithm

```
function en_50Ω(pts, R_t, ρ, I_50, j)
    reset_pointer(pts[pt_250], R_t[250Ω[ζ]], pts, R_t);
    connect R_t[50Ω[j]] terminal to the resistor network;
    I = I_50[j + 1];
    time_stamps(ρ, I);
    pts[pt_50] ++;
function en_250Ω(pts, R_t, ρ, I_50, I_250, I_total[i], k)
    reset_pointer(pts[pt_1k], R_t[1kΩ[ξ]], pts, R_t);
    connect R_t[250Ω[k]] terminal to the resistor network;
    I = I_250[pts[pt_50]][k] + I_50[pts[pt_50]];
    time_stamps(ρ, I);
    if I == I_total[i] then
        set breakflag = TRUE and I_pre = I_total[i] ;
        increment_pointer(pts[pt_250], pts);
    return breakflag;
function en_1kΩ(pts, R_t, ρ, I_50, I_250, I_1k, I_total[i], k, l)
    reset_pointer(pts[pt_5k], R_t[5kΩ[ν]], pts, R_t) ;
    connect R_t[1kΩ[l]] terminal to the resistor network;
    I = I_1k[pts[pt_50]][l] + I_250[pts[pt_50]][k] + I_50[pts[pt_50]];
    time_stamps(ρ, I);
    if I == I_total[i] then
        set breakflag = TRUE and I_pre = I_total[i];
        increment_pointer(pts[pt_1k], pts);
    return breakflag;
function en_5kΩ(pts, R_t, ρ, I_50, I_250, I_1k, I_5k, I_total[i], k, l, m)
    connect R_t[5kΩ[m]] terminal to the resistor network;
    I= I_5k[pts[pt_50]][m] + I_1k[pts[pt_50]][l] + I_250[pts[pt_50]][k] +
      I_50[pts[pt_50]];
    time_stamps(ρ, I);
    if (I + 1) == I_50[pts[pt_50] + 1] then
        /* (I + 1) is enough to enable one extra 50
           Ω resistor                                  */
        set breakflag = TRUE, I_pre = I_total[i] and pts[pt_5k] = 0;
    if I == I_total[i] then
        set breakflag = TRUE and I_pre = I_total[i];
        increment_pointer(pts[pt_5k], pts);
    return breakflag;
```

to 50 MHz. After the MCU sends the trigger signals to the target device and measurement device through the GPIO pins, it starts to change resistance until it reaches the desired target current or voltage.

III. Performance Evaluation and Case Studies

In this section, we evaluate Excalibur's linearity, dynamic range, resolution, stability, and temperature.

A. Evaluation of Current Output Linearity

ADC non-linearity requires curve-fitting to achieve high accuracy. Excalibur improves calibration accuracy by providing a linear output over a wide range. Therefore, linearity is a crucial performance metric for a high accuracy calibration platform. To measure linearity performance, we use a high-accuracy DMM7510 [28] to validate Excalibur's current output at 25 °C. When validating the linearity, we connect a DMM to Excalibur's output to measure the ground-truth current.

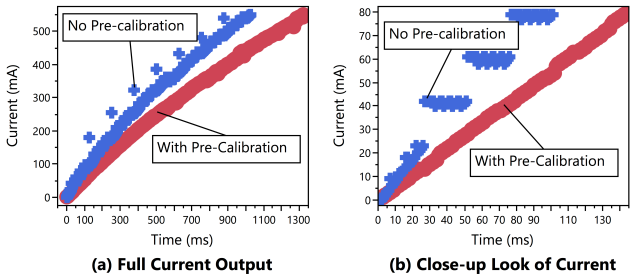


Figure 6. Experimental results of current output performance between the linear algorithm of [18] and the algorithm proposed in this paper. (a) Full current output range. (b) A close-up look at the current output.

For this experiment, the current output changes every 1ms from 0 mA to 500 mA. As we shown in Figure 1, the current output is linear. Figure 1(b) shows that the current output step size is 1 mA while keeping the linear performance to 20 mA .

We also validate the current output by using the linear algorithm of [18] and our proposed algorithm under the same conditions. Figure 6 shows the comparison between the linear algorithm without pre-calibration and our proposed algorithm with a pre-calibration array. On the full current output range, we can see the output without pre-calibration is not linear and introduces jumps between steps. The non-linearity is because the target current value is mismatched due to the lack of wire resistance calculation when the board switches from a higher resistance path to a lower one. We can see that our proposed algorithm offers a linear behavior while the output without pre-calibration includes large gaps.

B. Evaluation of Dynamic Range

Dynamic range and resolution are important performance metrics for a calibration platform. Excalibur's output range and resolution can be customized to match with the target ADC's range of interest. To measure dynamic range, resolution, and static accuracy, a high-accuracy DMM (DMM7510) is used to validate the results. Figure 7 shows the experimental results pertaining to dynamic range validation. The current output spans from 0 mA to 600 mA and its voltage output spans from 0.6 mV to 5 V. These features make Excalibur ideal for calibrating various ADCs.

C. Evaluation of Temperature Stability

To measure Excalibur's performance under different temperatures, we place it into a high precision temperature chamber for 30 minutes before performing the measurement. Then, we connect the DMM7510 to Excalibur to perform current and voltage measurements. Figure 8 illustrates the experimental results under five different temperature conditions: 15 °C, 25 °C, 35 °C, 45 °C, and 55 °C. We can see

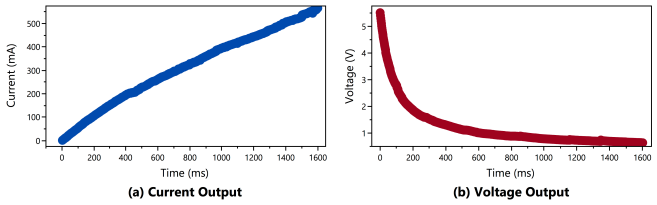


Figure 7. Empirical evaluation of (a) current and (b) voltage output.

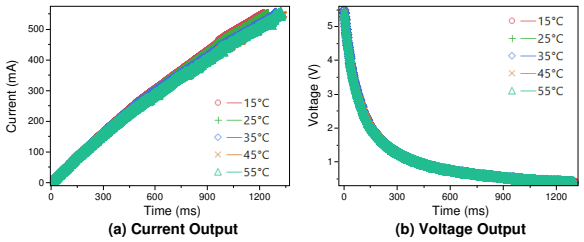
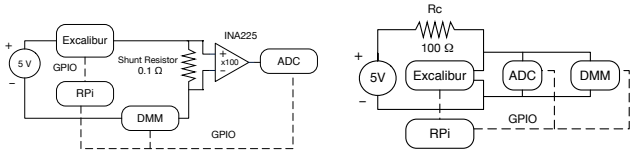


Figure 8. This figure shows the output linearity under different temperature conditions for (a) current and (b) voltage.



(a) Current Calibration Setup (b) Voltage Calibration Setup

Figure 9. Experimental setup used for (a) current, and (b) voltage calibration.

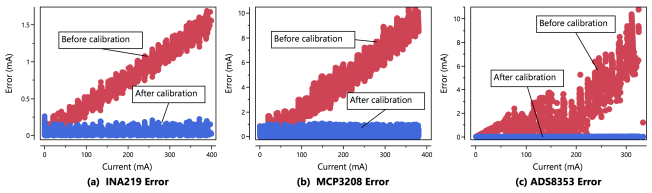


Figure 10. Experimental calibration results for (a) INA219, (b) MCP3208, and (c) ADS8353. These results show current measurement errors before and after calibration. Errors are significantly reduced after calibration.

the current performance remains linear and the voltage performance is stable under various temperature conditions. This feature not only enhances reliability and accuracy, but also enhances the ability to adapt to unpredictable environments.

D. Case Studies

In this section, we present case studies to demonstrate the benefits of calibration using Excalibur. We perform real-world calibrations of the most popular COTS ADCs supporting various resolutions and sampling rates: ATMega2560 (10 bit, 125 kHz), MCP3208 (12 bit, 100 kHz) [35], ADS8353 (16 bit, 600 kHz) [36], and INA219 (12 bit, 500 kHz) [37]. We use a DMM7510 to obtain ground truth. We use the target ADC and DMM to measure the outputs of Excalibur simultaneously. After completing the measurements, we use the saved timestamp log to correlate the measurements between ADC and DMM. For each experiment, we generate a calibration function $f(d)$ or a calibration table to correlate ADC and DMM measurements. Since the log file reflects the stability time stamp of each configuration, we can safely compare the two closest entries collected by the ADC and DMM without requiring precise time synchronization of the two devices.

1) Current Calibration: Some of the IoT energy measurement platforms provide simplified solutions by ignoring the effect of voltage variations during the measurement [10], [12], [13], [16], [38], [39]. They assume voltage is constant during sensor activities; thus, measuring current represents power consumption. A common solution for current measurement is to sample voltage drop across a precision current shunt resistor connecting to the power supply of the device under observation, as shown in Figure 9 (a). According to Ohm's law, the voltage across the shunt resistor is proportional to the current and can be multiplied by supply voltage to obtain the device's power consumption. We use a 0.1 Ω shunt resistor with 0.5% accuracy. More importantly, we placed the shunt resistor close to the ground leg of the circuit to increase the measurement accuracy. Ignoring this would cause the common-mode voltage produced by the shunt resistor to be outside of the specification of op-amp or ADC, which could lead to inaccurate readings or damaging the device [40].

Our experimental results on current performance concentrate on the range from 0 mA to 400 mA. Since the magnitude of sleep state current for a typical sensor node is below a typical ADC's resolution, we use an amplifier to magnify the voltage across the shunt resistor before performing the analog to digital conversion. We use the INA225 *operational amplifier* (OP-AMP) [41] to magnify voltage across the shunt resistor and set the gain at 100. To perform the calibration, we connect the ADC and DMM to the current output of Excalibur and collect measurements using both devices simultaneously. We use the settling time information provided by Excalibur to calibrate between DMM and ADC measurements. As timing data reflects the current change stability interval, we can safely compare the two closest entries collected by the ADC and DMM. Figure 10 shows the current measurement error of INA219, MCP3208, and ADS8353 conducted in a normal indoor temperature at 25 °C. Since the errors of all three ADCs increase linearly versus current, we perform a polynomial fitting to calibrate the errors.

2) Voltage Calibration: Figure 9 (b) illustrates the common calibration setup for voltage. High-resolution measurements of ultra-low voltage are essential due to most of the

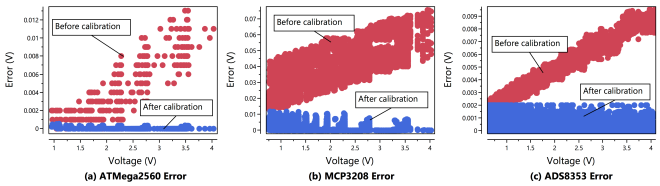


Figure 11. Experimental calibration results for (a) ATMega2560, (b) MCP3208, and (c) ADS8353. These results show voltage measurement errors before and after calibration.

Table IV
COMPARISON OF MEASUREMENT ERRORS BEFORE AND AFTER CALIBRATION.

ADC	Measurement Type	%Error (before)	%Error (after)
INA219	Current	0.42%	0.02%
MCP3208	Current	2.58%	0.01%
ADS8353	Current	3.15%	0.01%
ATMega2560	Voltage	0.2%	0.01%
MCP3208	Voltage	5.29%	0.01%
ADS8353	Voltage	0.17%	0.0014%

sensor devices operate under low voltage.

For example, the ECG sensor's operating range is below 3 mV [42]. To validate Excalibur's performance for sensing applications, our experimental results on voltage performance concentrate on the range from 0.6 mV to 5 V. We connect the ADC and DMM to the voltage output of Excalibur and collect measurements on both devices simultaneously to perform calibration. Figure 11 shows the voltage measurement errors of ATMega2560, MCP3208, and ADS8353. The errors of all three ADCs increase non-linearly versus voltage. For these cases, we use a calibration table to find the best fit to calibrate the ADC measurement errors.

3) Results Summary: Table IV presents the calibration results and confirms the significant effect of Excalibur's calibration on measurement accuracy. These results demonstrate that by providing an extensive calibration range, the errors reduce to nearly 0.01%. For example, in the MCP voltage case, the ADC error reduces from 5.29% to 0.01%.

IV. RELATED WORK

ADC plays a pivotal role to provide optimal data conversion accuracy of IoT devices even under various environmental conditions. However, various factors are affecting the precision of data conversion. For example, device characteristics, process technology, the resistance of circuit paths, environmental temperature variations, and finite gains are among the parameters affecting accuracy.

Standard methods to minimize the effects of these parameters are *trimming* and *calibration* [43]. Trimming is performed during the production phase [44]. After measuring and correcting the parameters of the device in a controlled condition, the trimmed values are programmed into the device. Trimming values, however, cannot be changed after manufacturing. Therefore, subsequent drift due to device aging, environmental temperature, or system-level noise cannot be corrected by trimming [45].

Calibration can be performed multiple times after fabrication. Therefore, it can be used to compensate for changes that occur over time and those influenced by environmental conditions. There are two types of calibrations: *self-calibration* (a.k.a., *foreground calibration* and *power up calibration*) and *external calibration* (a.k.a., *background calibration* and *run-time calibration*). Self-calibration performs the measurement-correction process within the device itself and does not require an external device. This calibration process, which interrupts the ordinary data conversion process, is performed during the device power-up process [46]–[49]. Although this technique significantly reduces errors, it cannot cope with temperature drift and other sources of inaccuracy such as sensor-to-ADC path loss. It is also possible to calibrate a data converter without affecting the conversion process. This is referred to as external calibration [50], [51]. External calibration encompasses temperature effects and offers a higher absolute accuracy than foreground calibration.

Self-calibration, however, is only as accurate as the on-board reference voltage. This voltage can drift slightly over time, and thereby, external calibration is still essential. For example, although the 12-bit *successive-approximation-register* (SAR)[1] ADC of STM32F205 includes a background calibration feature, Hartung et al. [53] used two resistors to perform external calibration. They showed that this process reduces voltage and current errors from 2.36% and 2.21% to 0.87% and 1.56%, respectively. Zhou et al. [12] relied on external calibration, although TI MSP430F2618 has a background calibration capability.

The common approaches of external calibration include using: (i) fixed value resistors [10], [11], (ii) mechanical potentiometer [12], [13], (iii) extensive simulation environment [3], [14], [15], and (iv) commercial power analyzer [16], [17]. However, as we mentioned in Section I, these calibration mechanisms pose several limitations. Although using fixed value resistors provides an accurate and stable load, it requires a large number of fixed-value resistors and a great deal of effort to collect enough samples for a quality calibration. Due to these challenges, researchers use only a few resistors to calibrate the full range of an ADC [10], [11]. This insufficient use of resistors provides a limited calibration range and does not eliminate data distortion across a wide range. Some researchers use mechanical potentiometers to generate a wide-range load [12], [13]. However, mechanical potentiometers raise concerns regarding long-term reliability because they are sensitive to shock and vibration. Furthermore, mechanical potentiometers have a limited

[1]SAR ADCs are common choices for medium to high resolution applications with sample rates under 5 Msps. These ADCs perform a binary search to convert analog input to digital output [52]. The resolution of SAR ADCs ranges from 8 to 16 bits and they offer low power consumption.

resistance range and they are unable to withstand high voltage and current values. It is also possible to create extensive simulation environments to calibrate ADCs [3], [14], [15]. These environments often require a sophisticated design to achieve the desired accuracy and cannot be reused for other applications. The last solution is to use a commercial DC power analyzer [16], [17], such as the Agilent N6705X [33], which costs more than $14,000 for a complete solution.

V. Conclusion

Calibration is one of the key requirements when building reliable sensing systems. In this paper, we presented Excalibur, a highly-accurate, low-cost, portable, programmable, and scalable calibration tool. The key novelties of this tool are as follows: First, it provides linear current and rational function voltage outputs. Second, it offers a scalable design where both the output range and resolution can be adjusted to match with the input range of various ADCs. Third, this tool employs a set of pre-calibrated measurement arrays and a set of temperature error correction functions to perform self-calibration. Fourth, Excalibur offers a time synchronization mechanism to eliminate the impact of warm-up time and sampling rate on matching pairwise values during calibration. We implemented a prototype of Excalibur and showed its impact on reducing the measurement error of various ADC types.

Instead of using the Raspberry Pi board to program and control the operation of Excalibur, it is possible to employ a simpler, more energy-efficient processor. In addition to reducing cost, employing a low power module reduces heat dissipation, which would also lower the temperature impact of the measurement and increase calibration accuracy. Another enhancement is adding an adjustable gain circuit to the INA219 module to measure a broader current and voltage range. By adding a measurement selection block, the INA219 module would be able to measure both current and voltage directly.

Currently, Excalibur uses capacitor coupling to reduce electrical noise. Using a multi-layer PCB design (instead of two layers) provides isolation to the sensitive analog components and reduce circuit noise. Also, adding suppressors as close to the noise source as possible reduces the suppressed electrical noise caused by switching reactive loads. Using power planes instead of traces enhances current carrying capability and reduces wire resistance.

Besides calibration, the proposed tool can be employed in the following applications. Excalibur is suitable for stress testing the voltage regulator logic of IoT devices. Power management has been a challenge for IoT devices due to decreasing die size, lower decoupling capacitance, multiple chips, platform power states, increasing the number of power grids, and migrating hot-spots. The workloads being executed on IoT devices demonstrate a large variation in terms of both voltage and current [54]. It is critical for the IoT device's voltage regulator to be able to compensate such a dynamic range with the ability to stabilize power state transition in a few clock cycles [55]. When placed between the IoT device's voltage regulator and IoT logic, Excalibur can perform stress test by rapidly changing its resistance, causing the input voltage to the IoT logic to transient overshoot or undershoot. By doing so, Excalibur can test the health of the onboard voltage regulator's compensation logic. Excalibur can also be employed for power emulation and power measurement. Given the programmability of Excalibur, various operations of an IoT device could be modeled as functions generating current draws corresponding to the real behavior of the device. This system can be used for software and hardware enhancement, as well as lifetime analysis. For example, it enables testing the resilience of a power harvesting system under various types of loads.

References

[1] N. Verma and A. P. Chandrakasan, "An ultra low energy 12-bit rate-resolution scalable SAR ADC for wireless sensor nodes," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 6, pp. 1196–1205, 2007.

[2] K. L. Kraver, M. R. Guthaus, T. D. Strong, P. L. Bird, G. S. Cha, W. Höld, and R. B. Brown, "A mixed-signal sensor interface microinstrument," *Sensors and Actuators A: Physical*, vol. 91, no. 3, pp. 266–277, 2001.

[3] P. Chen, K.-M. Wang, Y.-H. Peng, Y.-S. Wang, and C.-C. Chen, "A time-domain SAR smart temperature sensor with -0.25 $\sim$ +0.35 °C inaccuracy for on-chip monitoring," in *34th European Solid-State Circuits Conference*. IEEE, 2008, pp. 70–73.

[4] B. Dezfouli, M. Radi, and O. Chipara, "REWIMO: A Real-Time and Reliable Low-Power Wireless Mobile Network," *ACM Transactions on Sensor Networks (TOSN)*, vol. 13, no. 3, p. 17, 2017.

[5] M. Garrard and P. Ryan, "Design, Accuracy, and Calibration of Analog to Digital Converters on the MPC5500 Family," *Freescale Semiconductor, Application Note AN2989*, 2005.

[6] M. G. Flores, M. Negreiros, L. Carro, and A. A. Susin, "INL and DNL estimation based on noise for ADC test," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 5, pp. 1391–1395, 2004.

[7] Texas Instruments, "Selecting an A/D Converter," http://www.ti.com/lit/an/sbaa004a/sbaa004a.pdf, 2015.

[8] P. Suchanek, V. Haasz, and D. Slepicka, "ADC nonlinearity correction based on INL (n) approximations," in *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. IEEE, 2009, pp. 137–140.

[9] IEEE collaboration and others, "IEEE Standard for Terminology and Test Methods for Analog-To-Digital Converters," *IEEE Std*, pp. 1241–2000, 2011.

[10] X. Jiang, P. Dutta, D. Culler, and I. Stoica, "Micro power meter for energy monitoring of wireless sensor networks at scale," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 186–195.

[11] A. Milenkovic, M. Milenkovic, E. Jovanov, D. Hite, and D. Raskovic, "An environment for runtime power monitoring of wireless sensor network platforms," in *Proceedings of the Thirty-Seventh Southeastern Symposium on System Theory (SSST)*. IEEE, 2005, pp. 406–410.

[12] R. Zhou and G. Xing, "Nemo: A high-fidelity noninvasive power meter system for wireless sensor networks," in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2013, pp. 141–152.

[13] I. Haratcherev, G. Halkes, T. Parker, O. Visser, and K. Langendoen, "PowerBench: A scalable testbed infrastructure for benchmarking power consumption," in *Int. Workshop on Sensor Network Engineering (IWSNE)*, 2008, pp. 37–44.

[14] J. Rivera, M. Carrillo, M. Chacón, G. Herrera, and G. Bojorquez, "Self-calibration and optimal response in intelligent sensors design based on artificial neural networks," *Sensors*, vol. 7, no. 8, pp. 1509–1529, 2007.

[15] M. Demierre, S. Pesenti, J. Frounchi, P.-A. Besse, and R. S. Popović, "Reference magnetic actuator for self-calibration of a very small hall sensor array," *Sensors and Actuators A: Physical*, vol. 97, pp. 39–46, 2002.

[16] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel, "Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems," in *Proceedings of the 12th international conference on Information processing in sensor networks*. ACM, 2013, pp. 153–166.

[17] A. Pötsch, A. Berger, and A. Springer, "Efficient analysis of power consumption behaviour of embedded wireless IoT systems," in *IEEE International on Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, 2017, pp. 1–6.

[18] C.-C. Li and B. Dezfouli, "ProCal: A Low-Cost and Programmable Calibration Tool for IoT Devices," in *International Conference on Internet of Things*. Springer, 2018, pp. 88–105.

[19] Analog Devices Inc, "AD5200/AD5201 256-Position and 33-Position Digital Potentiometers," https://www.analog.com/media/en/technical-documentation/data-sheets/ad5200_5201.pdf, 2012.

[20] J. Creech and D. Rice, "Digital potentiometers vs. mechanical potentiometers: Important design considerations to maximize system performance," *Analog Devices, MA, USA, Technical article*, 2015.

[21] Analog Devices Inc, "ADG1611/ADG1612/ADG1613 1 Ω Typical On Resistance, ±5 V, +12 V, +5 V, and +3.3 V Quad SPST Switches," http://www.analog.com/media/en/technical-documentation/data-sheets/ADG1611_1612_1613.pdf, 2015.

[22] J. Rabaey, *Low power design essentials*. Springer Science & Business Media, 2009.

[23] J. N. Fox, "Temperature coefficient of resistance," *Physics Education*, vol. 25, no. 3, pp. 167–69, 1990.

[24] F. Zandman and J. Szwarc, "Non-Linearity of Resistance/Temperature Characteristic: Its Influence on Performance of Precision Resistors," *Vishay Precision Group, Tech. Rep*, vol. 108, 2013.

[25] K. Seshan and D. Schepis, *Handbook of thin film deposition*. William Andrew, 2018.

[26] R. L. Frey, "Wire-wound resistor," 1980, US Patent 4,185,263.

[27] Vishay Intertechnology, Inc, "Wirewound Resistors, Precision Power, Surface Mount," http://www.vishay.com/docs/30102/wscwsn.pdf, 2017.

[28] Tektronix, Inc, "DMM7510 $7\frac{1}{2}$-Digit Graphical Sampling Multimeter," https://www.tek.com/tektronix-and-keithley-digital-multimeter/dmm7510, 2017.

[29] M. Al-Asadi, A. Duffy, A. Willis, K. Hodge, and T. Benson, "A simple formula for calculating the frequency-dependent resistance of a round wire," *Microwave and Optical Technology Letters*, vol. 19, no. 2, pp. 84–87, 1998.

[30] Microchip Technology Inc., "$\pm 0.5\,^{\circ}\mathrm{C}$ Maximum Accuracy Digital Temperature Sensor," http://ww1.microchip.com/downloads/en/DeviceDoc/25095A.pdf, 2011.

[31] L. Ada, "Adafruit INA219 Current Sensor Breakout," *Adafruit Industries*, 2016.

[32] F. R. Parente, M. Santonico, A. Zompanti, M. Benassai, G. Ferri, A. D'Amico, and G. Pennazza, "An electronic system for the contactless reading of ECG signals," *Sensors*, vol. 17, no. 11, p. 2474, 2017.

[33] Keysight Technologies, "Keysight N6700," https://www.keysight.com/en/pc-851482/n6700-modular-power-system?pm=SC, 2017.

[34] N. P. Pawar and M. M. Patil, "Driver Assistance System based on Raspberry Pi," *International Journal of Computer Applications*, vol. 95, no. 16, 2014.

[35] Microchip Inc, "MCP3204/3208 2.7 V 4-Channel/8-Channel 12-Bit A/D Converters with SPI™ Serial Interface," http://ww1.microchip.com/downloads/en/DeviceDoc/21298c.pdf, 2017.

[36] Texas Instruments Inc., "ADSxx53 Dual, High-Speed, 16-, 14-, and 12-Bit,Simultaneous-Sampling, Analog-to-Digital Converters," http://www.ti.com/lit/ds/symlink/ads7853.pdf, 2014.

[37] Texas Instruments, "INA219 Zerø-Drift, Bidirectional Current/Power Monitor With I^2C Interface," http://www.ti.com/lit/ds/symlink/ina219.pdf, 2015.

[38] S. Naderiparizi, A. N. Parks, F. S. Parizi, and J. R. Smith, "μMonitor: In-situ energy monitoring with microwatt power consumption," in *IEEE International Conference on RFID (RFID)*. IEEE, 2016, pp. 1–8.

[39] J. Andersen and M. T. Hansen, "Energy bucket: A tool for power profiling and debugging of sensor nodes," in *Third International Conference on Sensor Technologies and Applications (SENSORCOMM)*. IEEE, 2009, pp. 132–138.

[40] National Instruments, "Current Measurements: How-To Guide," http://www.ni.com/tutorial/7114/en/, 2018.

[41] Texas Instruments, "INA225 36 V, Programmable-Gain, Voltage-Output, Bidirectional, Zero-Drift Series, Current-Shunt Monitor," http://www.ti.com/lit/ds/symlink/ina225.pdf, 2015.

[42] E. Nemati, M. J. Deen, and T. Mondal, "A wireless wearable ECG sensor for long-term applications," *IEEE Communications Magazine*, vol. 50, no. 1, pp. 36–43, 2012.

[43] G. Manganaro, *Advanced data converters*. Cambridge University Press, 2011.

[44] W. A. Kester, *Data conversion handbook*. Newnes, 2005.

[45] S.-T. Ryu, S. Ray, B.-S. Song, G.-H. Cho, and K. Bacrania, "A 14-b linear capacitor self-trimming pipelined ADC," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 11, pp. 2046–2051, 2004.

[46] R. C. Taft, C. A. Menkus, M. R. Tursi, O. Hidri, and V. Pons, "A 1.8 V 1.6 GSample/s 8 b self-calibrating folding ADC with 7.26 ENOB at Nyquist frequency," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 12, pp. 2107–2115, 2004.

[47] H.-S. Lee, D. A. Hodges, and P. R. Gray, "A self-calibrating 15 bit CMOS A/D converter," *IEEE Journal of Solid-State Circuits*, vol. 19, no. 6, pp. 813–819, 1984.

[48] A. N. Karanicolas, H.-S. Lee, and K. L. Barcrania, "A 15-b 1-Msample/s digitally self-calibrated pipeline ADC," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 12, pp. 1207–1215, 12 1993.

[49] J. Cao, X. Meng, G. C. Temes, and W. Yu, "Power-on digital calibration method for delta-Sigma ADCs," in *IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016, pp. 2002–2005.

[50] C. Tsang, Y. Chiu, J. Vanderhaegen, S. Hoyos, C. Chen, R. Brodersen, and B. Nikolic, "Background ADC calibration in digital domain," in *IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2008, pp. 301–304.

[51] U.-K. Moon and B.-S. Song, "Background digital calibration techniques for pipelined ADCs," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 2, pp. 102–109, 1997.

[52] S.-S. Wong, U.-F. Chio, Y. Zhu, S.-W. Sin, U. Seng-Pan, and R. P. Martins, "A 2.3 mW 10-bit 170 MS/s two-step binary-search assisted time-interleaved SAR ADC," *IEEE journal of solid-state circuits*, vol. 48, no. 8, pp. 1783–1794, 2013.

[53] R. Hartung, U. Kulau, and L. Wolf, "Distributed energy measurement in WSNS for outdoor applications," in *13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2016, pp. 1–9.

[54] B. Dezfouli, I. Amirtharaj, and C.-C. Li, "EMPIOT: An energy measurement platform for wireless IoT devices," *Journal of Network and Computer Applications*, vol. 121, pp. 135–148, 2018.

[55] S. Ray, Y. Jin, and A. Raychowdhury, "The Changing Computing Paradigm With Internet of Things: A Tutorial Introduction," *IEEE Design & Test*, vol. 33, no. 2, pp. 76–96, 2016.