Analog Considerations for Designing a Potentiostat in a PSoC: Sources of Errors and Compensation Techniques

Prattana Lopin¹ and Kyle V Lopin²*

¹Department of Biology, Naresuan University, Faculty of Science, Phitsanulok, Thailand
²Department of Physics, Naresuan University, Faculty of Science, Phitsanulok, Thailand

*Email: kylel@nu.ac.th

Abstract. Designing with a Programmable System on a Chip (PSoC) allows for entire circuit designs to be implemented with a single commercially available chip and eliminates the need to physically assemble electronic components. This is possible as the PSoC incorporates a microcontroller with digital and analog components into a single package. While this design allows for much easier implementation of circuits, there are some drawbacks. In this paper we will demonstrate one of those drawbacks, high routing resistance in parts of the analog mesh that connects the analog parts and external pins. We show how this resistance can cause measurement errors when the PSoC is implemented as a single chip potentiostat. As the internal analog routing resistance is in the kΩ range, measuring currents in the µA range can cause mV errors, leading to lose of voltage control during electrochemical experiments. We also demonstrate a calibration routine to compensate for this voltage error that reduced the error by over 90%.

1. Introduction

Electrochemistry is a branch of chemistry that studies chemical reactions that involve the movement of electrons and is the process underlying a broad range of phenomenon from lithium ion batteries in electric vehicles, to handheld glucose meter for health monitoring, and virus detection methods [1]. To study electrochemistry, an electrode is placed into solution while the voltage of the electrode is controlled and the current through the electrode is measured. As chemical reactions occur on the electrode, called the working electrode, when electrons are added or removed from a chemical in solution to the electrode, the electrons can be measured to indicate the speed of the chemical reaction. The device used to perform these experiments is called a potentiostat. To control the potential of the solution, another electrode is used that can pass or remove current into the solution, called the counter electrode. To accurately measure the potential of the solution, a third electrode, the reference electrode, is often used to control the counter electrode current [2]. The circuit used to control the electrodes is called a potentiostat and a basic layout is shown in figure 1B.

There has been much research into developing new implementations of potentiostats. Both general purpose potentiostats [3–5] and potentiostats for special purposes have been proposed over the last few years. As electrochemistry has a diverse set of applications, there has also been a diverse set of potentiostats developed for special purposes such as for wireless applications [6], for health monitoring with a smart phone [7], non-invasively measuring blood alcohol levels [8], to water

Content from this work may be used under the terms of the CreativeCommons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Published under licence by IOP Publishing Ltd
monitoring sensors [9] to name a few. One of the benefits of the many developed open source potentiostats is that it allows for new research groups to integrate the potentiostats into new electrochemical devices.

The Programmable System on a Chip (PSoC) is a family of integrated circuits (IC) that integrates an ARM microcontroller with integrated programmable digital and analog components. This allows new electronics devices to be developed with fewer components. The PSoC lends itself well to researchers as it allows them to make new research prototypes that are easier and cheaper to develop. The versatile PSoC has been used in many research areas, include: educational purposes [10], to study animal psychological experiments [11], measuring human health [12], acoustic physiological experiments [13], and for an early flood warning system [14].

While the PSoC does have many benefits, there are drawbacks that come with the platform. Some of these are, increased noise through the analog components as the resistors are implemented as switch capacitors to be compact and programmable. Also, the resistors have large errors in their value as the switch capacitors are implemented with CMOS technology. To compensate for this, the analog components have to be calibrated before use. Many of the analog connections that connect the analog components and pins also have high resistances. Every analog pin can be programmed to attach to any analog component. Analog switches are used to make these programmable connections, but because they have to implemented in CMOS technology, they have relatively high resistances (50 - 500 Ω) per switch and 2-5 switches are usually needed to connect components together. While some special connections are available for the digital to analog converters and the operational amplifiers, most components will have high resistance connectors between them. The component where this resistance will have the most affect is on the transimpedance amplifier. This is the component we will focus on in this research report to characterize the problem and show a compensation method for it.

2. Methods and Results

2.1. Analysis of the Problem
In this research we highlight design considerations for using analog components in a PSoC. We have previously built a potentiostat using a Programmable System on a Chip 5LP (PSoC) [15]. To demonstrate the problem with the analog routing, we first port the previous version of the PSoC-Stat (available at: https://github.com/KyleLopin/PSoC-Potentiostat) to the CY8CKIT-050 development board. Figure 1A shows a basic potentiostat layout in PSoC Creator, the PSoC programming environment, along with an ideal potentiostat layout in figure 1B. While at first it appears that the

![Figure 1](image-url)
implementation is straightforward, we next run a cyclic voltammetry experiment from -1 V to +1 V over a 10 kΩ resistor, with the results measured by the potentiostat shown in figure 2A.

An ideal resistor should have a linear I-V relation with a slope of 100 (100 µA / 1 V). This ideal relationship is shown in figure 2A as the black dotted line. What we actually measure from the device is the solid red line in figure 2A. To show why the measurements are non-ideal, we recorded the potential (using an Analog Discovery 2) between the reference electrode (RE) and the working electrode (WE) shown in figure 2B. Again, the ideal situation is shown in the black dotted line as the potential should sweep from -1 V to +1 V and back down at a rate of 1 V/s. Instead, we measured the grey solid line, which goes from just -0.9 V to +0.9 V. To find the source of the 100 mV of error, we next recorded the voltage from the signal ground (a virtual ground held at 2.048 V above the USB bus ground) to the WE. This signal is shown in the solid red line in figure 2B. Ideally the WE should maintain a constant voltage at the virtual ground which is shown as the dotted red line in figure 2B. Unlike the ideal transimpedance amplifier (TIA) shown in figure 1B where it holds the voltage of the WE at the virtual ground potential, the PSoC potentiostat has a voltage error on the WE. The cause of this error is from the analog routing in the PSoC. Because the analog components are programmable, any component can be connected to any other analog or pin. To achieve this, the PSoC has a series of analog buses and switches internally to make the connections. Because all of these connections and switches have to be build in the integrated circuit (IC) they are optimized to be compact and configurable and not for analog performance. Because of this the analog routing has relatively high resistances, especially in the analog switches. To route the WE connection into the TIA implementation of the PSoC, a routing resistance of 800-2,500 kΩ is between the electrode and the operational amplifier (opamp) input, shown in figure 1C.

The analog routing can take many different internal paths, and the analog placement goes through an iterative improvement process, so the exact resistance of the routing can change between different builds. On average, the analog routing will produce a resistance of ~1 kΩ. This is the build shown in the solid red line in figure 2A and 2B. As the current across the 10 kΩ resistor for 1 V is 100 µA, this current across the 1 kΩ of routing error causes the 100 mV of error. This is the typical error, but some

![Figure 2](image-url)
builds can have even higher error. The highest error seen from the automatic analog placement routine was 2.5 kΩ. This can cause errors of up to 250 mV for currents of 100 µA. The dashed green line in figure 2A shows the current measured by the potentiostat across the 10 kΩ resistor with the 2.5 kΩ routing resistance. From this it can be seen that the potentiostat has a 25% error in the measurements. The RE to the WE potential is shown in figure 2B in the dashed grey line. The voltage difference now goes from ~750 to +750 mV instead of the full -1 to +1 V. The dashed red line in figure 2B shows that this ~250 mV of voltage error is from the WE not maintaining virtual ground caused by the analog routing resistance.

2.2. Solution to Mitigate Analog Routing Resistance Error

For currents less than 10 µA, the analog routing error will be less than 10 mV, which is relatively small for most electrochemical experiments. Larger currents though can start to cause more significant errors, and in general we would like to eliminate or reduce this error to make the potentiostat as accurate as possible. To do this we can compensate for the WE voltage error.

If we put a known resistor between the WE and the counter electrode (CE) (connected to the RE) we get the circuit shown in figure 3. We then use the routine shown in figure 4 to calculate the routing resistance. After the hardware is started, a known calibration current from a current digital to analog current (IDAC) is put through the TIA. As the TIA implements the resistor through a switch capacitor topology and the resistance can be off by up to 20%, the TIA has to be calibrated before being used. The output voltage is then put into the PSoC’s analog to digital converter (ADC) (not shown in schematic) and the resistance of the TIA can be calculated as \( R_{TIA} = \frac{V_{out}}{i_{calibrate}} \). After this the current through the TIA can be calculated as \( i_{unknown} = \frac{V_{out}}{R_{TIA}} \). Running a cyclic voltammetry experiment, similar to the protocol shown in figure 2A, allows the total resistance between the CE and WE to be calculated as \( R_{known} + R_{routing} = \frac{V_{applied}}{i_{measured}} \). \( R_{known} \) and \( V_{applied} \) are set before the calibration and \( i_{measured} \) is calculated as \( V_{out} \) divided by \( R_{TIA} \). By taking the inverse of the slope of the cyclic voltammetry results and subtracting \( R_{known} \), the routing error can be calculated. After the routing resistance is calculated, it can be put into the firmware so we can fix the voltage error it causes.
Figure 3. Schematic of calibration setup to measure the routing resistance.

Figure 4. Flowchart for how to calculate analog routing resistance.

On Initialization: Put known current through TIA

On Initialization: Calculate µA per ADC counts

Every ADC ISR routine calculate voltage error

Add voltage error to DAC value on next voltage update

Finally, we are ready to remove the voltage error between the WE and CE. To do this, every time the potentiostat measures the current through the WE, the firmware will calculate the current and multiply the current by the analog routing error to calculate the voltage error. The voltage error can then be added to the $V_{\text{applied}}$ (applied by a voltage digital to analog converter (VDAC) not shown) driving the CE to maintain the proper WE-RE potential through the top opamp shown in figure 1B. Figure 5 shows the flowchart to perform this action. During the TIA calibration, a new variable has to save the conversion factor to take ADC counts and get the current in µA. Then during the potentiostat reading, every time the ADC records data (during an interrupt service routine (ISR)), the ADC counts are multiplied by the conversion factor to get the current (in µA) and multiplied by the analog routing resistance (in kΩ) to get the voltage error (in mV). This voltage error is then added to the voltage the RE is set to maintain to offset the WE offset. Figure 6 shows the large voltage error recorded before the compensation in red and the voltage error after implementing the compensation procedure outlined.
above in black. Figure 6B shows the errors zoomed in on the y axis. From figure 6 it can be seen that the original voltage errors range over 160 mV (-80 to +80 mV) while after adding the compensation, the voltage error ranges over 11 mV (-3.5 to 7.5 mV), though the error is mostly within 2.5 mV of what the WE to RE voltage should be.

All of the code needed to do this is in the updated PSoC-Potentiostat github repository and is saved in the version 1.2 release found at https://github.com/KyleLopin/PSoC-Potentiostat/releases/tag/1.2.

![Figure 6](image)

**Figure 6.** A) Voltage error with no compensation (V-shaped line) and after compensation (almost horizontal line close to 0 V). B) Same as A with the Y-axis zoomed in.

3. **Discussion and Conclusions**

The PSoC is a very versatile integrated circuit that allows for entire circuit designs to be programmed into a single chip. This eliminates the need for assembling electronic components which can take specialized tools and skills. This versatility comes at a cost though, as the programmable analog components cannot be optimized for any single application. To demonstrate that we looked at a potentiostat build with the PSoC and saw that the analog routing causes an error in how the potentiostat controls the electrode voltage. We then demonstrated a way to calibrate and compensate for this analog routing error. The full code used for this can be found at https://github.com/KyleLopin/PSoC-Potentiostat/releases/tag/1.2.

References
[1] Ozer T, Geiss B J and Henry C S 2019 Review—Chemical and Biological Sensors for Viral Detection *J. Electrochem. Soc.* 167 037523
[2] Søpstad S, Johannessen E A and Imenes K 2020 Analytical errors in biosensors employing combined counter/pseudo-reference electrodes *Results Chem.* 2 100028
[3] Glasscott M W, Verber M D, Hall J R, Pendergast A D, McKinney C J and Dick J E 2020 SweepStat: A Build-It-Yourself, Two-Electrode Potentiostat for Macroelectrode and Ultramicroelectrode Studies *J. Chem. Educ.* 97 265–70
[4] Nordin N A, Jamil A J, Som A S C M, Abdullah W F H, Zain Z M, Hamid U M A and Rani S 2016 Potentiostat readout circuit design for a 3-electrode electrochemical biosensing measurement system 2016 7th IEEE Control and System Graduate Research Colloquium (ICSGRC) pp 159–63
[5] Arévalo-Ramírez T, Torres C C, Cela Rosero A and Espinoza-Montero P 2016 Low cost potentiostat: Criteria and considerations for its design and construction 2016 IEEE ANDESCON pp 1–4

[6] Jenkins D M, Lee B E, Jun S, Reyes-De-Corcuera J and McLamore E S 2019 ABE-Stat, a Fully Open-Source and Versatile Wireless Potentiostat Project Including Electrochemical Impedance Spectroscopy J. Electrochem. Soc. 166

[7] Shen X, Ju F, Li G and Ma L 2020 Smartphone-Based Electrochemical Potentiostat Detection System Using PEDOT: PSS/Chitosan/Graphene Modified Screen-Printed Electrodes for Dopamine Detection Sensors 20 2781

[8] Gamella M, Campuzano S, Manso J, Rivera G G de, López-Colino F, Reviejo A J and Pingarrón J M 2014 A novel non-invasive electrochemical biosensing device for in situ determination of the alcohol content in blood by monitoring ethanol in sweat Anal. Chim. Acta 806 1–7

[9] Soøstad S, Imenes K and Johannessen E A 2020 Chloride and pH Determination on a Wireless, Flexible Electrochemical Sensor Platform IEEE Sens. J. 20 599–609

[10] Felgueiras M, Fidalgo A, Alves G, Motta G, Schlichting L and Ferreira G 2015 A remote lab to support e-learning on Programmable System-on-Chip (PSoC) 2015 3rd Experiment International Conference (exp.at’15) pp 167–8

[11] Shan Q, Bullock D, Sumner C J and Shackleton T M 2014 Monitoring Lick Responses in Animal Behavioral Experiments Using a PSoC 2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS) pp 633–40

[12] Patil P N and Sawant S D 2016 Estimating multiple health parameters using PSoC controller 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) (Chennai, India: IEEE) pp 384–7

[13] Shan Q, Bullock D, Palmer A R and Shackleton T M 2015 Control of Acoustic Signal Processing in Physiological Experiments Using PSoCs 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems pp 1135–8

[14] Magpantay Mercado R J 2016 Design of wireless sensor networks using embedded Programmable System-on-Chip (PSoC) as applied to community-based flood early warning systems (CBF-EWS) 2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAES) (Putrajaya: IEEE) pp 214–23

[15] Lopin P and Lopin K V 2018 PSoC-Stat: A single chip open source potentiostat based on a Programmable System on a Chip PLOS ONE 13 e0201353