Assembly Parts and Instructions for the PUDELstat and PUDELstat App

Joseph S. Elias*

Chemistry & Physics Department, Simmons University, 300 Fenway, Boston, Massachusetts 02115, United States

* joseph.elias@simmons.edu

www.eNanoLab.com

Assembly Instructions

All parts required for the assembly of the PUDELstat are listed in **Table 1** and can be procured from most electronics suppliers. We have included part numbers that refer to the Digi-Key online catalog. The microcontroller board interfaces with the PUDELstat circuitry (**Figure 1**) through a prototyping breadboard “shield” that is inserted directly over the Arduino board. Step-by-step assembly instructions are included in **Figure 2**.

The assembled PUDELstat can be used without further modification in the potentiostatic configuration, or modified to be capable of performing galvanostatic experiments (**Figure 3**). If desired, an enclosure for the workstation can be laser cut from 3/32” thickness cast acrylic, and fastened together with a pair of screws (M3 x 0.50 mm thread, 45 mm long, McMaster-Carr # 90116A186) and nuts (McMaster-Carr # 94000A330). The enclosure design is available as a scalable vector graphics (.svg) at www.eNanoLab.com/PUDELstat along with the Gerber files for custom-designed printed circuit boards (PCBs) for hard-wired versions of the PUDELstat.

Table 1. Parts list for the assembly of one PUDELstat. Parts were quoted from Digi-Key Electronics on 12/22/2023. The custom PCB was quoted from OSH Park on 11/11/23.

Component	Digi-Key Part Number	Units	Unit Price (USD)
DFRduino Uno V3.0	1738-1228-ND	1	16.9
10 kΩ trimmer potentiometer	1993-1104-ND	1	0.70
Potentiometer knob	1993-JPEPL5116GR-ND	1	0.44
Alligator clip with pigtail (4 pack)	CAB-13191-ND	1	3.50
LMC6484 Quad Operational Amplifier	LMC6484IN/NOPB-ND	1	2.83
Fixed resistors, through-hole			
4.7 kΩ	CF14JT4K70CT-ND	1	0.10
100 kΩ	CF14JT100KCT-ND	3	0.10
68 Ω	CF14JT68R0CT-ND	4	0.10
1 kΩ	CF14JT1K00CT-ND	2	0.10
10 kΩ	CF14JT10K0CT-ND	4	0.10
510 Ω*	CF14JT510RCT-ND	1	0.10
* variable depending on desired current range: 220 Ω (± 10 mA), 510 Ω (± 5 mA), 1 kΩ (± 2.75 mA), 10 kΩ (± 0.5 mA)			
Capacitors, through-hole			
1 μF, film	BC1622-ND	1	1.80
10 μF, electrolytic	1189-3733-1-ND	2	0.23
1000 pF, ceramic	BC5256CT-ND	1	0.47
100 pF, ceramic	445-175498-ND	1	0.30
Breadboard Configuration			
DFRobot prototype shield	1738-1076-ND	1	11.00
2-Port terminal block	277-1734-ND	1	0.65
Assorted jumper wires (70 pieces)	BKWK-3-ND	1	4.90
Jumper wires, female/male (20 pieces)	1528-2233-ND	1	1.95
		Total	47.40

PCB Configuration			
Custom PCB	N/A	1	9.45
3-Port terminal block	277-1735-ND	1	1.03
Pin headers	2057-PH1-13-UA-ND	3	0.17
Boardmount socket	929974E-01-01-ND	2	0.21
		Total	40.31

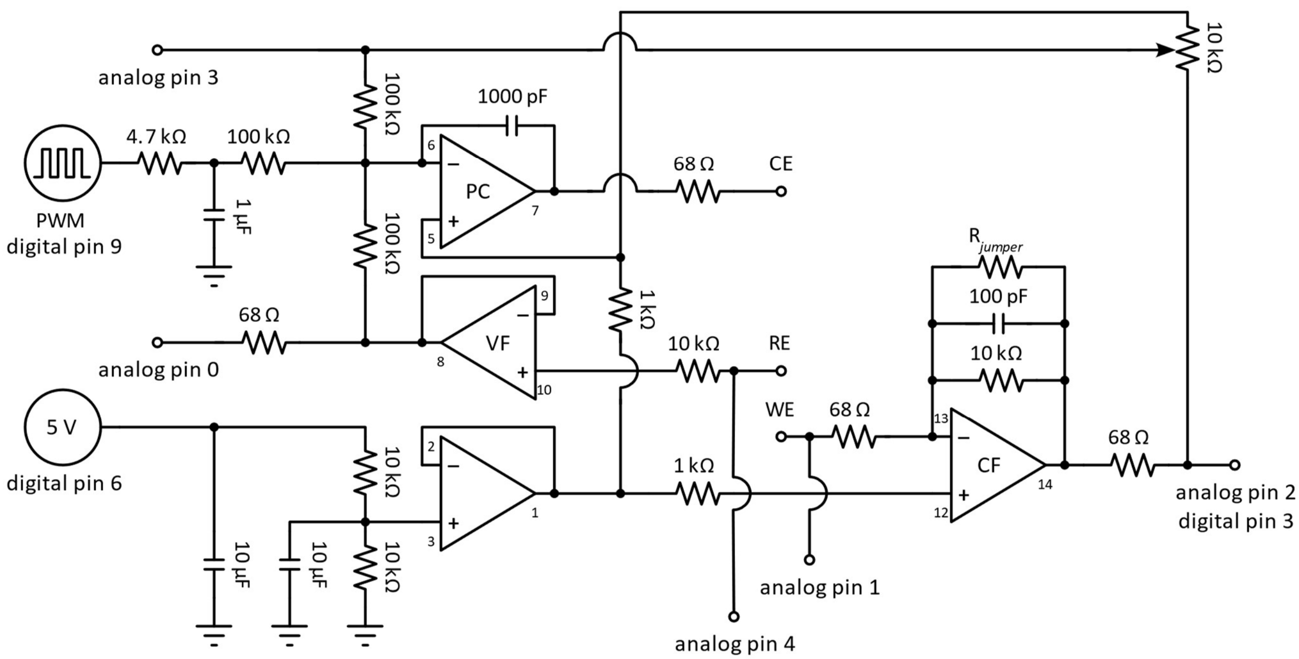


Figure 1. Schematic circuit diagram for the PUDELstat in potentiostat mode. The numbers next to the op amp inputs and outputs correspond to the LMC6484 pins, which is powered by the +5 V supplied by the Arduino microcontroller. The jumper resistor (R_{jumper}) across the current follower is variable, depending on the current sensitivity; R_{jumper} = 220 Ω (± 10 mA), 510 Ω (± 5 mA), 1 kΩ (± 2.75 mA), 10 kΩ (± 0.5 mA). Analog and digital pins refer to the inputs and outputs of the Arduino microcontroller board.

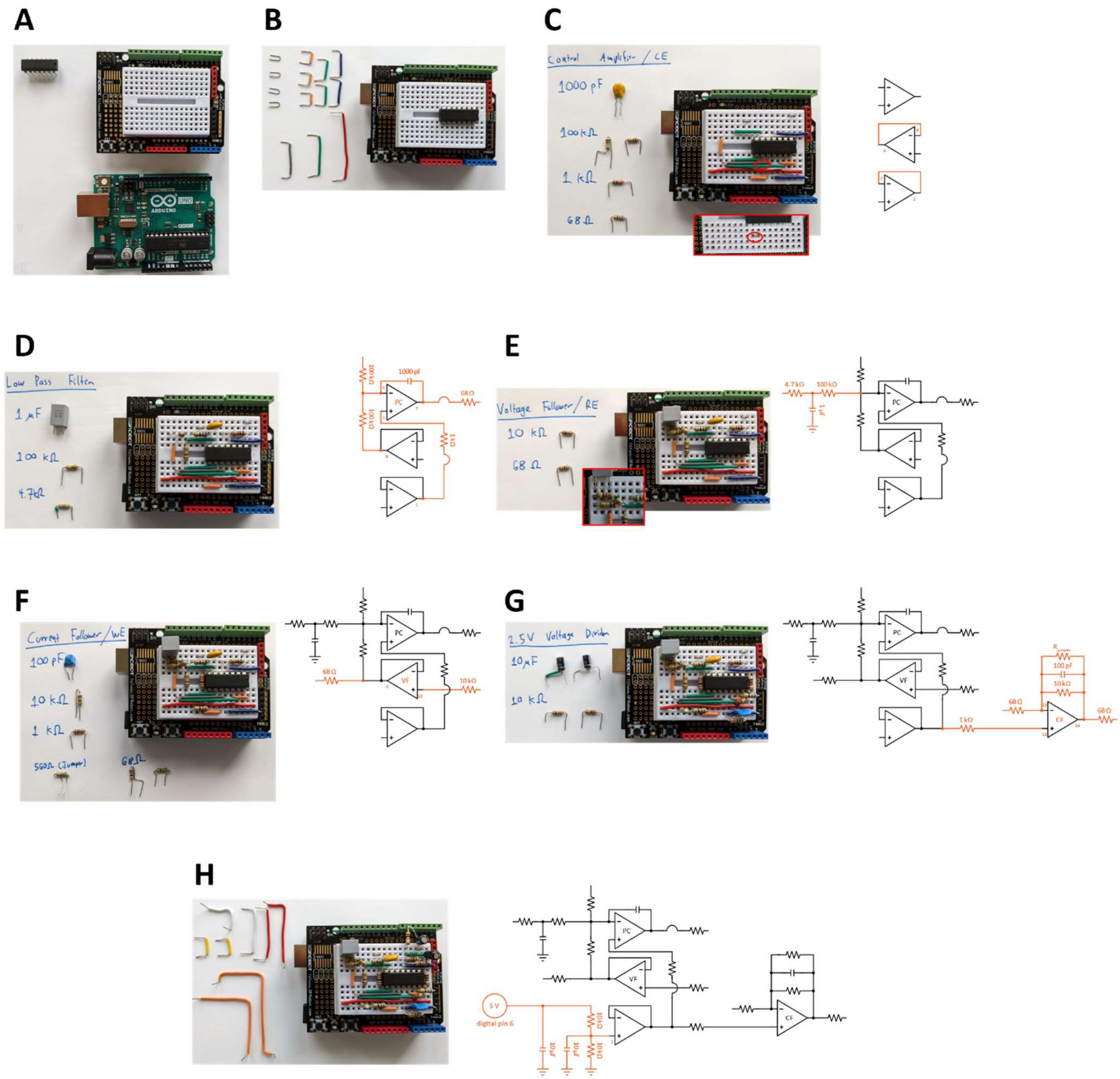
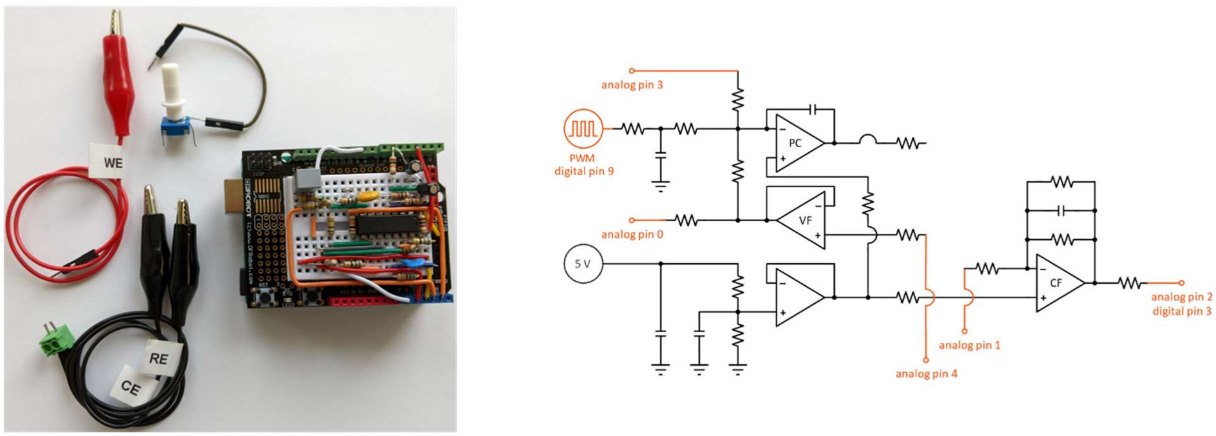


Figure 2. Assembly of the PUDELstat; (A – B) initial components, (C) voltage followers, (D) potential control amplifier, (E) low-pass filter, (F) refinement of the reference electrode voltage follower, (G) current follower, and (H) voltage divider circuitry.

I



J

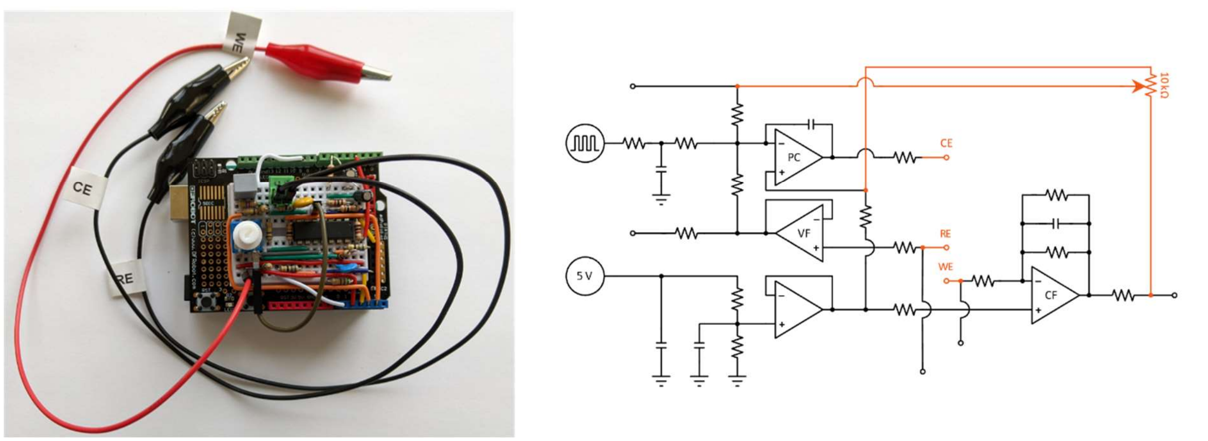


Figure 2, continued. Assembly of the PUDELstat; (I) connections to Arduino pins, and (J) positive-feedback ohmic drop compensation circuitry.

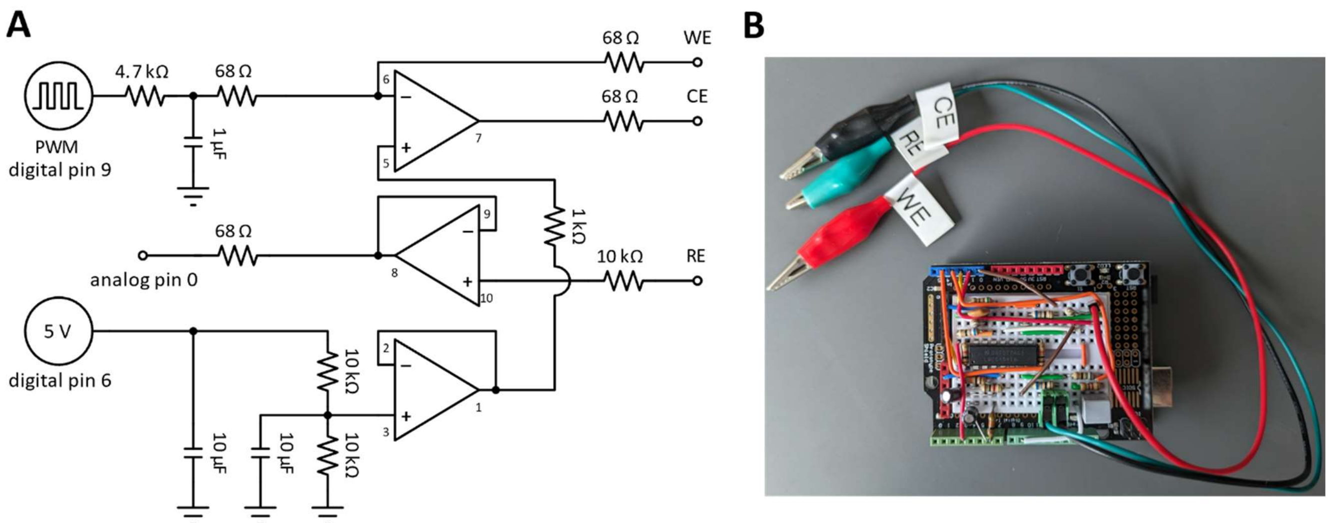


Figure 3. Galvanostat configuration of the PUDELstat; (A) schematic circuit diagram and (B) the corresponding prototyping board.

Software Installation

The standalone graphical user interface PUDELstat App is available for Windows machines as an installer package at www.eNanoLab.com/PUDELstat; simply download and unzip *PUDELstat App Installer.zip* and run the *installer.exe* executable. All LabVIEW VIs and the most up-to-date version of PUDELstat App can be found on our homepage (www.eNanoLab.com/PUDELstat). Included in the installer are assembly instructions and a parts list, as well as all necessary files to run PUDELstat App. The *README.txt file*—reproduced below—walks through the installation of the application on Windows machines and includes all the necessary firmware upgrades.

System Requirements:

Windows 7 or later operating systems

Program Files:

1. PUDELstat App.exe (main application)
2. README.txt
3. Assembly Parts and Instructions.pdf (detailed instructions on how to build the PUDELstat)
4. PWM_31kHz.ino (updates Arduino's PWM frequency)
5. PUDELstat App.aliases
6. PUDELstat App.ini

Required Software:

This software requires the NI LabVIEW Runtime 2021 SP1, NI-VISA Runtime 25.1, as well as the drivers from the Arduino IDE.

This software is available free of charge at the following links:

http://www.ni.com/en-us.html

https://www.arduino.cc/en/Main/Software

Software Installation:

1. Install LabVIEW firmware to Arduino:

    a. Download and install LabVIEW (will only be used once to install the necessary firmware; you may install the Community Edition of LabVIEW or an evaluation copy):

    https://www.ni.com/en-us/support/downloads/software-products/download.labview.html#443310

    b. Download and install the MakerHub Hobbyist Toolkit:

    https://www.ni.com/en-us/support/downloads/tools-network/download.labview-hobbyist-toolkit.html#443255

    c. Connect the fully-assembled PUDELstat to the computer and open LabVIEW

    d. From the LabVIEW menu bar run the firmware wizard: *Tools>Hobbyist>Firmware Wizard*

    c. Follow the on-screen instructions to upload the firmware, when prompted, select

        Device Family: Arduino

        Device Type: Arduino Uno

        Firmware Upload Method: Serial/USB

2. Update the pulse-width modulation (PWM) of the Arduino to 31 kHz; the default frequency (490 Hz) is too low for PWM to be useful with this instrument

    a. Connect the fully-assembled PUDELstat to the computer and open Arduino IDE

    b. From the Arduino IDE menu (*File>Open*) navigate to and open the necessary PWM_31kHz.ino sketch, included in the installer

    c. Within the PWM_31kHz sketch, upload the code to the Arduino by clicking the right-pointing arrow on the top left-hand side

3. Open the PUDELstat App.exe executable and start experimenting!

Software Instructions

After successful installation of PUDELstat App, electrochemical experiments can be performed.

Software Configuration

1. Connect the PUDELstat to your computer via USB.
2. Open PUDELstat App and configure your experiment as needed by clicking on the following buttons on the top portion of the application window (**Figure 4**):

    Save Path (): navigate to a folder you wish to save your data in, then click "Current Folder."

Serial Port ( ): select the serial port that is communicating with the PUDELstat. This will be "COM" followed by a number (*not* necessarily COM3). You may need to click "Refresh" for it to properly display.

Sensitivity (Jumper): from the drop-down, select the current sensitivity of the instrument that matches your choice of R_{jumper}.

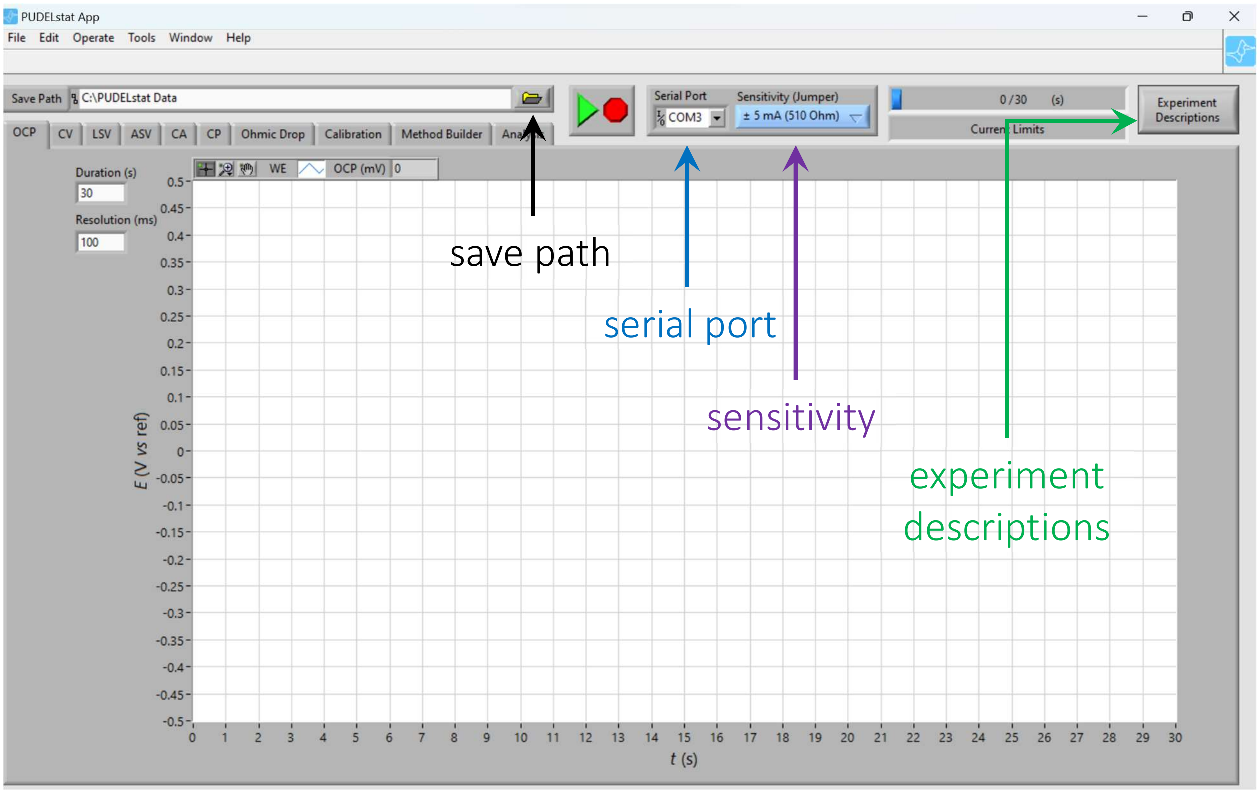


Figure 4. Configuring PUDELstat App.

Device Calibration

The current and voltage measured by the PUDELstat is calibrated by measuring the cyclic voltammetry of a resistor of known resistance and fitting the resulting voltammogram to Ohm's Law. This routine is performed in the *Calibration* tab (**Figure 5**), and calibration data is automatically saved into the *conversion factors.ini* initialization file which can be found in the PUDELstat App Program Files. Whenever PUDELstat App is called, the calibration data is automatically loaded from this initialization file.

1. For best results, be sure *not* to correct for solution resistance; turn the potentiometer all the way clockwise, or remove it from the breadboard temporarily.
2. Connect the resistor between the working electrode and the counter/reference electrode. Adjust the parameters on the left-hand side, being sure to input the resistance (in Ohms) of the resistor into *Sample Cell Resistance*.
3. Click the green "play" button ( ). Enter a suitable file name and click "OK."
4. Three plots will display; Measured Voltage and Current vs. Time (bottom left), the Measured Voltage vs. Control Voltage (center), and Voltage Response vs. Potential (right). The potential is calibrated from the linear regression of Measured Voltage vs. Control Voltage, and the current is calibrated from the linear

regression of Voltage Response vs. Potential. Results from linear regression (slope, intercept, and R^2) are displayed on the top of these plots.

5. To update the conversion factors, copy and paste the results from the linear regression analyses to the respective fields under *Potential Calibration* and *Current Calibration* on the bottom of the screen.
6. Click the green "play" button ( ) once again to update the conversion factors saved in the *conversion factors.ini* file.

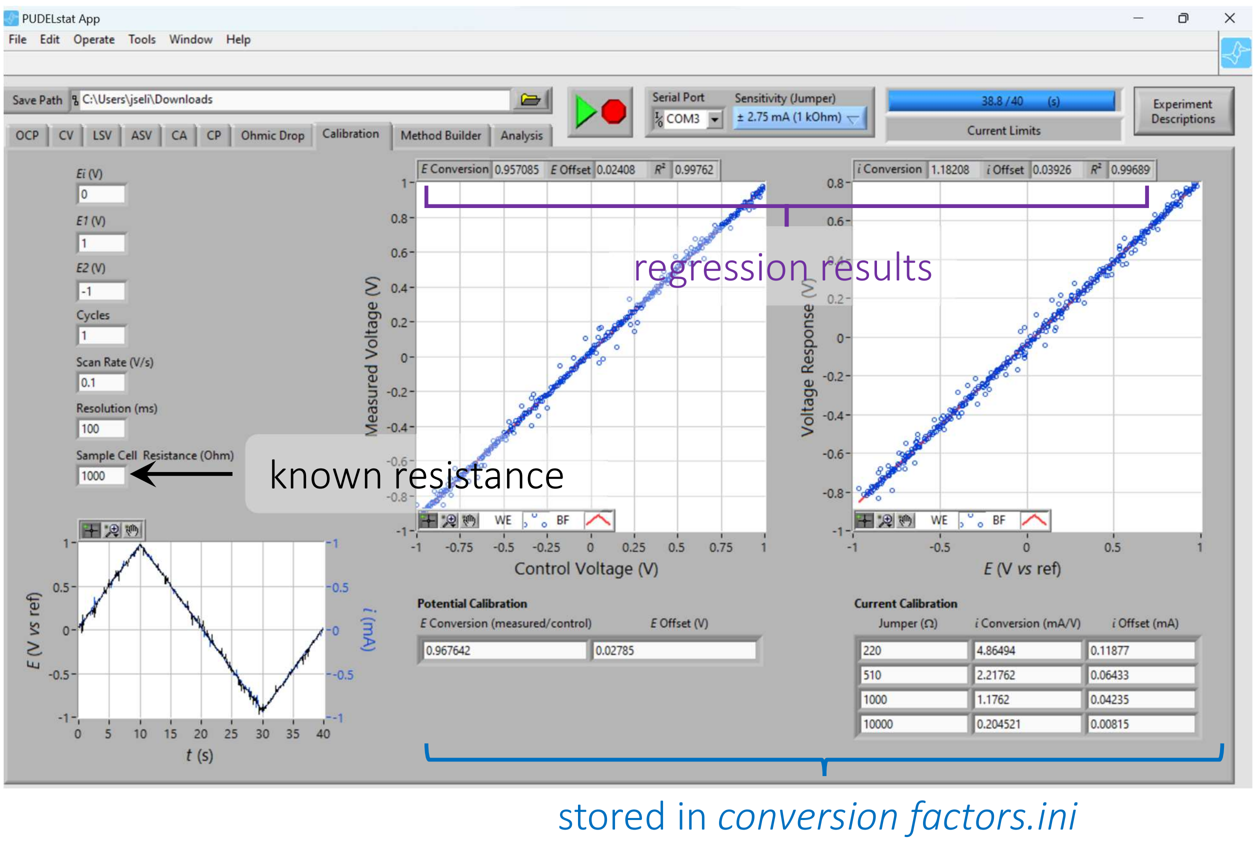


Figure 5. Current and potential calibration in PUDELstat App.

Correcting for Solution Resistance

Before voltammetry, the solution resistance can be compensated for by performing the following routine in the *Ohmic Drop* tab (**Figure 6**).

1. Adjust any parameters as needed. Compensation of 85% of R_u is customary.
2. Click the green "play" button ( ). Enter a suitable file name and click "OK."
3. Slowly turn the Ohmic drop compensation dial on the top of the PUDELstat in the counter-clockwise direction until high-frequency oscillations appear in the data. Slowly turn the dial clockwise until the oscillations just disappear.
4. Click "*Measure Ru*" on the left. Two values appear; R_u (Ohm), which is your measured *uncompensated* solution resistance, and R_c (Ohm), which is the how much of R_u you will need to *compensate* for.
5. To compensate by R_c, slowly turn the dial clockwise until the R_u approximately matches the value for R_c you noted above.
6. The solution resistance has been compensated. Click the red "stop" button ( ) to stop acquisition.

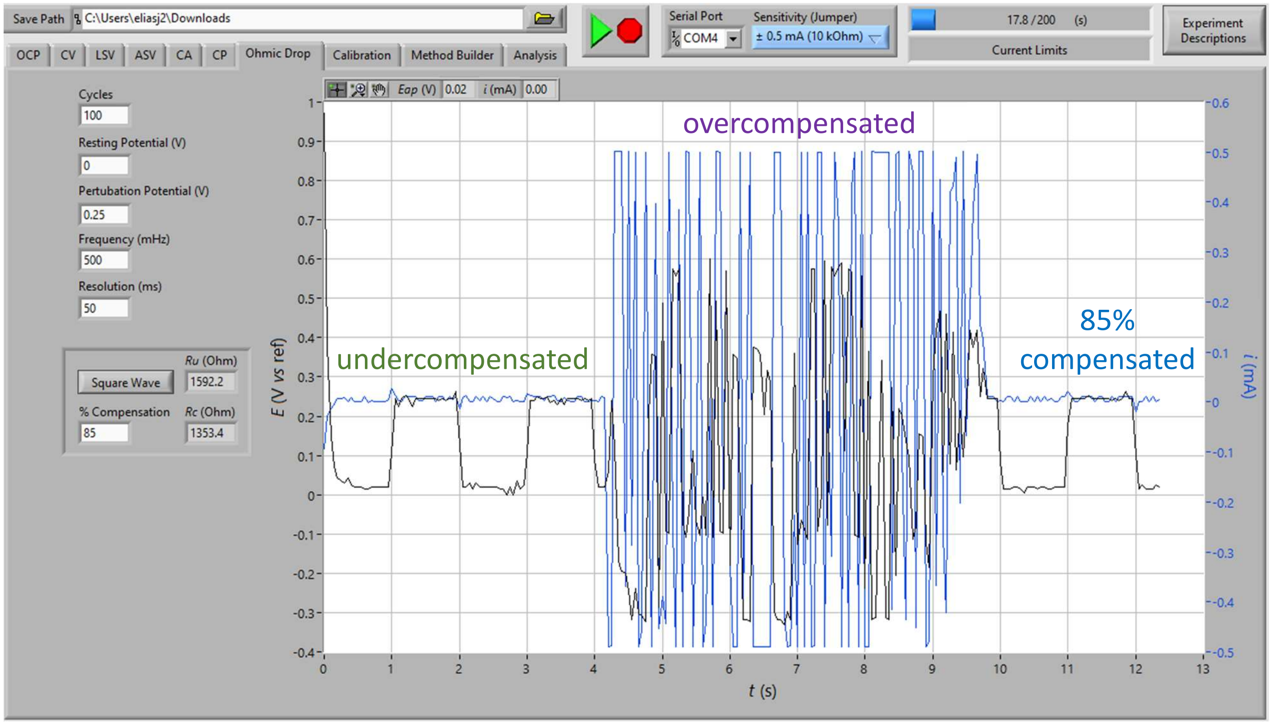


Figure 6. Compensating for solution resistance with the "Ohmic Drop" method.

Electrochemical Measurement

Once the solution resistance has been compensated for, electrochemical measurements can be acquired by clicking on the methods tab of your choice (*OCV, CV, LSV, ASV, CA, CP, Galvanostatic Cycling, Method Builder*), adjusting the experimental parameters as needed, and pressing the green "play" icon. Brief descriptions of each of the electrochemical measurements can be found by clicking the *Experiment Descriptions* button (**Figure 7**), and saved data can be analyzed graphically using the *Analysis* tab.

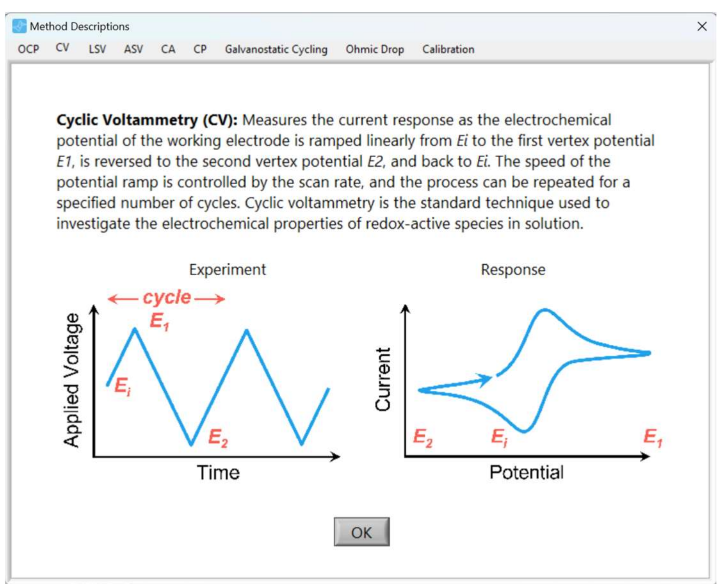


Figure 7. The *Experiment Descriptions* popup.