

An Improved Galvanostat for the Characterization of Commercial Electrochemical Cells

D. González-Arjona*, E. Roldán González, G. López-Pérez, M. M. Domínguez Pérez

Department of Physical Chemistry, Faculty of Chemistry, University of Seville, E-41012, Spain

Abstract An improvement of both the hardware and software of a previous design developed by the authors for the study of commercial electrochemical cells is reported. The galvanostat now features the automatic selection of the current scale, with a maximum full scale of 250 mA. An external reference voltage is also used to increase the precision (ca. 0.1%) of the measured cell voltage. In the new design, the acquisition software is linked to Excel spreadsheet. In this way, the nominal cell parameters can be read from the spreadsheet via the serial port and can be set without reprogramming the Arduino interface. Furthermore, the data are gathered directly into the spreadsheet, and can even be plotted in real time. The cost of the whole system remains under \$150, not including the controlling laptop

Keywords Electrochemistry, Laboratory Instrumentation, Commercial Batteries, Arduino Platform

1. Introduction

Electrochemical cells are used to store electrical energy in the form of free energy of chemicals, as they can draw direct-current from the electrodes, sustained by a combination of electron transfer reactions (Red-Ox). These reactions are quite complex, involving solid-state diffusion which may result in phase changes and re-crystallization. The processes, in some of these electrochemical cells, can be reversed by applying a direct-current to restore the electrical energy. Cells of this type are categorized as secondary or rechargeable batteries. Nowadays, miniaturized primary and/or rechargeable (secondary) batteries are employed in a vast number of devices: mobile telephones, watches and clocks, cameras, laptops, portable tools, remote controllers, cardiac pacemakers, etc.

The study and characterization of these devices provides a very interesting and instructive learning experience as it comprehensively illustrates concepts of thermodynamics and kinetics at the undergraduate level within the context of practical electrochemistry. At the same time, it can help students to deepen their understanding of such common concepts as energy conversion, efficiency, cell capacity, etc.

Quantities which are employed to characterize the operation and performance of an electrochemical cell can be obtained by simply monitoring the current supplied to or produced by the cell, and its terminal voltage. So, the measuring tools required to perform the experimental

characterization are simple: a voltmeter, a chronometer and a device that can supply a constant current, i.e. a galvanostat.

Studies of this kind have been performed in our laboratory for more than ten years using a manually operated homemade galvanostat. This device has been recently modified to allow digital control using an Arduino microcontroller board[1].

There are many other microcontrollers and microcontroller platforms. All of them simplify the challenging microcontrollers' programming and provide an easy-to-use package. The Arduino platform offers some additional advantages: it is relatively inexpensive (ca. \$30), runs on Windows, Macintosh OSX, and Linux operating systems, presents a transparent, user-friendly programming environment, and is accessible as open source software-hardware. It is a very versatile device with a wide range of applications in many areas like process control and data acquisition, robotics, communications, sensing, etc[2].

The galvanostat operations (on/off and charging/discharging) are controlled by different Arduino digital lines which activate the associated relays. The current value within the full scale is set by a digital line capable of generating a pulse with modulated signal. The PWM pulsed signal is applied to a classical low pass-filter (gain=1, freq=1Hz) to be transformed into a dc signal proportional to the mark-space ratio. Moreover, the Arduino board is capable of timing operation that allows a very precise control of the charging/discharging processes. Communication with the computer is maintained via a conventional RS232 interface. It is used both for reading and transmitting the acquired data to the computer as well as setting the program code in the internal Arduino memory.

* Corresponding author:

dgonza@us.es (D. González-Arjona)

Published online at <http://journal.sapub.org/jlce>

Copyright © 2013 Scientific & Academic Publishing. All Rights Reserved

A complete description of the earlier design of the digital galvanostat and the attendant programs can be found in reference[1] and supplemental material.

In the course of using this device for about one year, considerable improvements were made of both hardware and software while keeping costs low. The main hardware upgrades are: a) Use of a stabilized external reference voltage to increase precision, b) Automatic selection of the current full scale and c) Enhanced full-scale current up to 250 mA. The software has been updated mainly by using a free Data Acquisition tool (DAQ)[3], associated with an Excel spreadsheet. The DAQ enables communication between the Arduino code and an Excel spreadsheet via the RS232 interface. Thus, initial parameters can be sent to the Arduino during the acquisition process and the data acquired saved, plotted and analyzed with the Excel spreadsheet in real time. The software has been adapted to the new hardware characteristics (full-scale current and the external voltage reference) as well as the status information displayed onto the spreadsheet control module.

The new upgrades offer a more versatile and user friendly assemblage. Thus, students can focus their attention on data analysis and discussion of results, rather than on operating the instrument and the physically demanding and tedious timing control during the acquisition process. A more detailed description of these improvements will be described in the following sections.

2. Hardware Upgrades

The galvanostat is based on the well known scale/inverter circuit. The former digital galvanostat had four full-scale current levels (5mA, 10mA, 50mA and 100mA). The

maximum current obtained from this design has been found to be insufficient even for low capacity secondary batteries (up to 500 mAh).

Moreover, the full scale was chosen manually by selecting a combination of resistances. For these reasons, the system has been redesigned so as to include three resistances (2k Ω , 220 Ω and 20 Ω). A combination of these resistances can be set using two extra Arduino digital outputs (#3 for 250 mA and #5 for 25 mA full-scale).

Each digital output controls a switch relay that connects or disconnects the resistance to the circuit, as a function of the current selected in the experiment, see Figure 1. Thus, the new galvanostat has enhanced the current range by a factor of 2.5. Larger heat sinks, aided by air blown by a small fan, are now needed to prevent overheating.

The reference voltage in the common Arduino design is 5 volts by default. This is the maximum value that can be set for the analog inputs. These analog inputs have associated A/D 10 bits converters returning integers from 0 to 1023. Thus a 3.3 V electrochemical cell gives 675 counts on the converter, one bit up or down implies an error of ca. 0.3% which can be considered small enough for most types of measurement in an undergraduate lab. For nominal voltage electrochemical cells around 1 V the error starts to become significant (ca. 1%).

A way to increase the precision of measured voltage is by using the Arduino directive “analogReference(External)”. In this way, the voltage applied to the AREF pin in the Arduino board is used as the maximum voltage value of the converter. Using an external reference voltage somewhat larger than the electrochemical cell nominal voltage, the error can be lowered to ca. 0.1%.

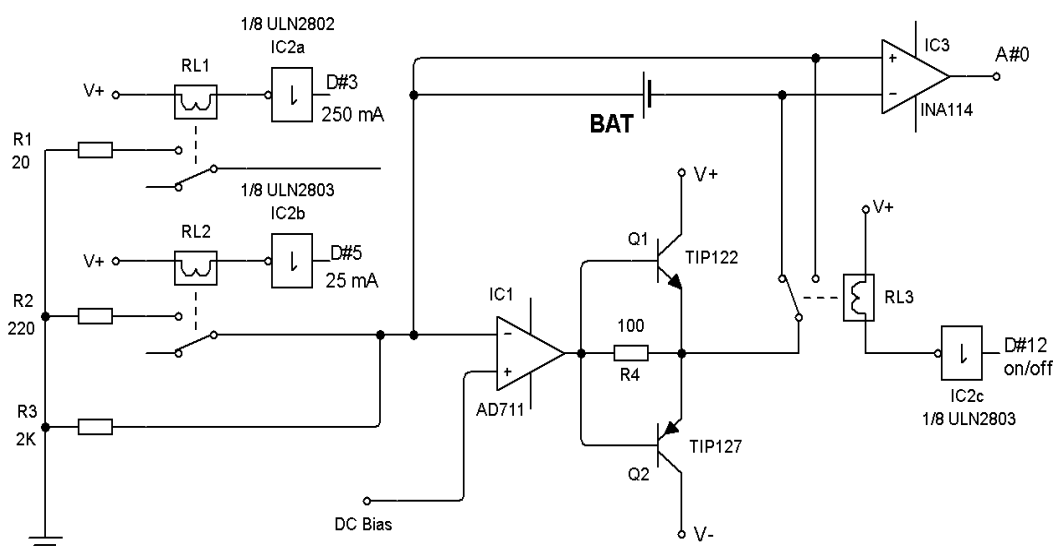


Figure 1. Galvanostat with automatic full-scale current selection

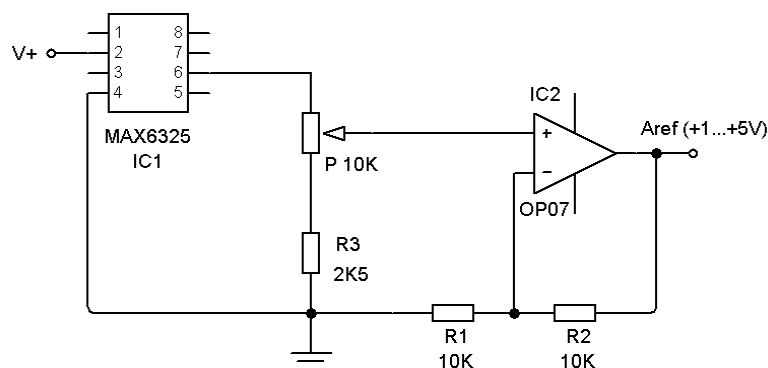


Figure 2. Variable precision voltage circuit used as external voltage reference

Thus, a circuit having a MAX6325 precision reference voltage[4], has been included in the new design, see Figure 2. The voltage can be set between 1.0 V to 5.0 V by an external multi-turn potentiometer (P10K in Figure 2). This reference voltage level is set before the start of the measurement by using a common external voltmeter, according to the electrochemical cell nominal voltage.

Additional hardware modifications have been implemented in the galvanostat PCB: a power supply (+15V/-15V) and two rows of male pins that allow the Arduino PCB to be directly plugged in. Figure 3 shows a picture of the galvanostat PCB which includes descriptions of the main components.

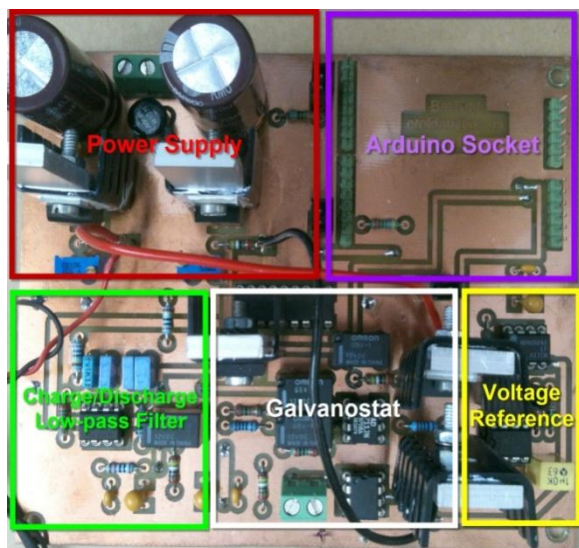


Figure 3. Galvanostat PCB board showing its different components: power supply, Arduino socket, charge/discharge module, voltage reference module and the galvanostat circuit itself

3. Software Upgrades

The Arduino interface is easily programmed to perform control and data acquisition tasks. A complete instruction manual with examples, language reference, forums, purchasing information, and so on, can be found on the official website[5]. Programming is carried out by a “front-end” environment used to write, check, compile and

upload the final code to the Arduino microcontroller. The programming language is called “Processing”[6], with syntax quite similar to C++. Communication between the Arduino and the computer is maintained via the USB interface to which the Arduino is attached. The USB interface is recognized as a COM or Serial port by the computer. This interface is employed to transmit all information between the computer and the Arduino. For debugging purposes, the data communication through this interface can be displayed on the laptop screen using the COM monitor included in the Arduino environment front-end.

Nevertheless, the acquired data should be saved in a text file for future processing. This task can be accomplished by using the data logger software that writes all information[7].

An interesting and instructive feature of the new design is the option of visualizing the experimental data as they are being acquired. The main advantage of this option is clear; the user has direct visual information about the whole process, at a glance (“A picture is worth more than a thousand words”). On the other hand, including this feature introduces some additional complications linked to the timing for plotting, graphical interface programming, etc.

For the purposes of these experiments, Excel spreadsheet is an excellent tool that enables students to perform data analysis. Moreover, it has the ability to manage digital data via RS232[8, 9]. This option is very interesting, though a bit restrictive as it is limited to an OS where Excel can be fully installed (Windows, Mac, Wine ...). Under these circumstances, Excel spreadsheet controls the acquisition, the graphical representation and the subsequent treatment of the data. The main drawback arises from the fact that it is necessary to understand the Excel programming tools. Other spreadsheet types can be used instead but the difficulties are more severe, because adding a serial communications port requires rather specialized programming knowledge[10]. This strategy is therefore quite complicated, so it does not seem worthwhile to pursue it.

Nevertheless, a rule of thumb in programming will guide us here: don’t waste time “reinventing the wheel”. In this regard, a very nice and useful data acquisition tool connecting RS232 and Excel spreadsheet is provided freely by Parallax Inc., the PLX-DAQ Data Acquisition module[3].

This software was originally designed for their own microcontrollers, but since it uses the RS232 interface, and is made freely available, this could provide a solution. This PLX-DAQ control has very useful features: It records up to 26 columns of data and plots them as they arrive in real-time, reads from and records data into any cell on a worksheet, marks data within real-time, reads/sets any of 4 checkboxes, and has baud rates up to 128K.

The PLX-DAQ control uses a set of commands to indicate the kind of action to be performed. These commands are in the form of a string of characters which are sent through the COM interface. The syntax is simple and straightforward.

All commands are given in CAPITAL letters, and entered by a carriage return. For example, the "DATA, var1, var2, var3" directive enters the values of the var1, var2 and var3 variables in the first 3 cells of the next row. In order for this command to be carried out, the Arduino program code has to be: "Serial.println("DATA, var1, var2, var3");".

Information about the use of PLX-DAQ control and its relation to the Arduino code, along with instructive examples can be found on the World Wide Web[11]. Figure 4, shows an example of an Arduino program for reading and checking values of Excel cells by using the Parallax controller.

```

/* Head to write in a Excel sheet using the galvanostat
BatTes2 using the Parallax DAQ Excel controller
DGA Feb. 2013 */
unsigned long bat_nominal=0; //nominal capacity mAh; Read from Excel
double volt_nominal=0.0; // nominal voltage; Read from Excel
double int_act=0.0; // mA ; Read from Excel
double Ext_ref=0.0; // Volts ; Read from Excel
//
void setup() {
  pinMode (ledPin, OUTPUT); //
  //Initialize serial port
  Serial.begin(9600);
  // Send status info message
  Serial.println("MSG,Starting up");
  Setup_head();
  delay(1000);
  // Check that battery parameters can't be zero
  while ((bat_nominal==0.0) || (volt_nominal==0.0) || (int_act==0.0) || (Ext_ref==0.0))
  {
    Serial.println("MSG,Battery parameters can't be zero");
    //Read cells from Excell until no null values entered
    Serial.println("CELL,GET,G2");
    bat_nominal=Serial.parseFloat();
    Serial.println("CELL,GET,G3");
    volt_nominal=Serial.parseFloat();
    Serial.println("CELL,GET,G4");
    int_act=Serial.parseFloat();
    Serial.println("CELL,GET,G5");
    Ext_ref=Serial.parseFloat();
    Serial.println("MSG, ");
  }
  Serial.println("MSG,Read battery parameters");
}
//
void loop() {
  Serial.println("MSG,Process terminated normally.");
  while (HIGH==HIGH)
  {
    // Nothing to do: Process stopped
  }
}
// Subroutines
void Setup_head() {
  Serial.println("LABEL,Timer,t(s),E(V)");
  Serial.println("CELL,SET,F1, Primary Battery");
  Serial.println("CELL,SET,F2, Nominal Capacity(mAh)");
  Serial.println("CELL,SET,F3, Nominal Voltage(V)");
  Serial.println("CELL,SET,F4, Discharge current(mA)");
  Serial.println("CELL,SET,F5, External Reference(V)");
}

```

Figure 4. Example of Arduino code using a PLX-DAQ controller. The code reads and checks the values of Excel cells and writes the information on their status

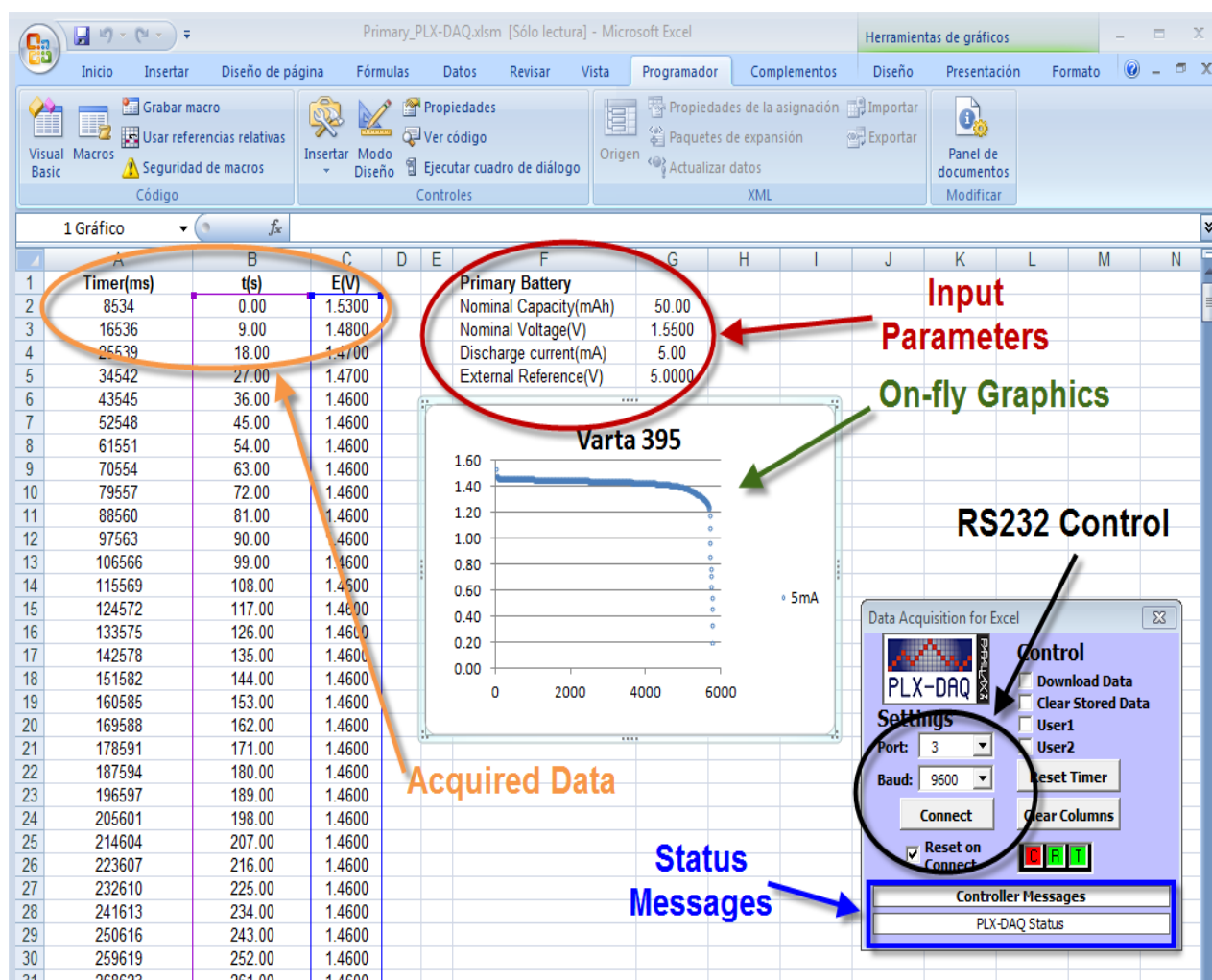


Figure 5. Snapshot of an Excel spreadsheet for the study of a V395 primary electrochemical cell showing the input/output data, plot and PLX_DAQ controller

Another interesting point is that the PLX-DAQ module is able to read/write Excel cell values “on the fly”. Thus, the acquisition program can read the characteristic experimental parameters (electrochemical cell characteristics, current to be used) and it can produce the adequate output without recompiling the entire program code.

PLX-DAQ control has to be set to the number and bauds rate of the serial COM port used by the Arduino board, see Figure 5. The character transmission rate can be selected within the range 1 ms to 0.08 ms (9600 to 128000 bauds).

The computer used has to be fast enough to be able to support real-time data acquisition and processing. Some factors that negatively affect this requirement are: computer speed, baud rate, data processing by Excel (data entering, plotting, and/or performing mathematical calculations).

Our developed characterization programs for electrochemical cells sample the voltage roughly every second. For this reason, they do not place much demand on the computer. Nevertheless, displaying a graph will increase the processing time considerably.

Figure 5 shows a snapshot of an Excel spreadsheet for discharging a V395 primary electrochemical cell [12]. After

acquisition, the spreadsheet should be saved for further analysis.

The earlier versions of the program that had been developed for primary and secondary electrochemical cells have been updated to take into account the Excel controller and the hardware upgrades. However, the working philosophy of the previously developed programs has been adhered to in cases where Excel spreadsheet is not available. Thus, the versions that store all the information to a text file have been updated to take into account the changes in hardware, different current scales and the use of external reference voltage. These versions are still useful when an automatic experimental procedure is employed, to safeguard against loss of data due to power outages.

The program for primary electrochemical cells reads a time/voltage combination every second, but only records arithmetic means at selected time intervals via RS232. This interval is a function of the electrochemical cell capacity and the discharge current selected. The stored mean values improve the signal/noise ratio achieving a better accuracy in the experimental measurements. The program terminates when a single voltage reading is below a defined threshold

value (ca. 0.2V). For a comparison of the behavior of secondary electrochemical cells it is required to set a fixed charge level. Thus, the program for secondary electrochemical cells starts by discharging it. Next, after a short time interval to allow for stabilization, it is subsequently charged at a selected percentage of its nominal value. The program estimates the current to be set as a function of nominal capacity and a charging time between 15min to 60min to reach that charge level. Again, the program ceases during a short time interval. Next, a current level is set to perform a number of pre-selected charging/discharging cycles. The nominal percentage of charge and discharge for each step is a parameter chosen at the beginning of the program. As in the case of primary electrochemical cells, the data are acquired every second, but the stored data is calculated as the mean voltage during the time interval.

4. Galvanostat Set up and Teaching Lab Examples

4.1. Setting up the Galvanostat

The new galvanostat version showed in Figure 3 can be assembled and checked by the teachers using the build list of electronic components, the general electronics schematics and the circuit picture board. The Arduino board is plugged upside-down in the upper right socket, see Figure 3. The complete assembly should be placed inside a plastic box for a safer use in the laboratory.

Arduino driver and the PLX-DAQ Data Acquisition module software have to be installed and checked in the computer intended to be used. Arduino installation software contains basic examples to check the correct operation. The program code depicted in Figure 4 can be loaded into Arduino to check the connection between the Arduino and PLX_DAQ module with Excel.

The supplementary material provides Arduino source programs and the Excel spreadsheets for the electrochemical cells characterization experiments: primary, secondary and power measurement. For teachers, all information, tips and tricks to build up the system, can be provided by the authors upon request.

4.2. Primary Electrochemical Cell Characterization Experiment

The commercial cell should be connected properly to **BAT** terminals + (plus) and – (minus), see Figure 1. The manufacturer cell nominal capacity, in mAh, and nominal voltage, in volts, should be entered in the spreadsheet cells G2 and G3 respectively, see Figure 5. The selected discharging current, mA, to be used during the test should be entered in the cell G4. Finally, the reference voltage level is set by the multi-turn potentiometer, Figure 2, and its value in volts entered in the cell G5. Next, these parameters are read and checked to avoid null values, when the button “Connect” is pressed in the RS232 Control. Then, the system waits until

the manual Start/Stop switch is activated. The measuring process starts storing values in Excel columns A, B and C corresponding to the Arduino Timer, in ms, the elapsed time, in seconds, and the cell measured voltage, in volts. The process finishes automatically when the voltage measured is negative or the user activates the manual switch Start/Stop.

From the measured voltage, V , the elapsed time during the experiment, t , the constant current used during the discharging process, i , and the manufacturer parameters: nominal capacity, Q_T , and nominal voltage, V_{Eq} , some interesting electrochemical cell parameters can be estimated. Thus, the theoretical maximum energy stored in the cell, E_T , can be calculated according to:

$$E_T = Q_T \cdot V_{Eq} \quad (1)$$

The practical available energy, E_p , can be calculated from:

$$E_p = i \int_0^t V dt \quad (2)$$

This integration can be easily performed by using the trapezoidal rule from the proper spreadsheet columns. Additionally, the cell efficiency, η , can be also estimated by using the following relationship: $\eta = E_p / E_T$.

It is also interesting to estimate the cell power, P , which can be calculated from: $P = i \cdot V$. The maximum value can be easily obtained from the P vs i plot.

4.3. Secondary Electrochemical Cell Characterization Experiment

The study of these kinds of cells has more complex setup. In a first step the cell has to be completely discharged. Then, it has to be charged to a predetermined level (i.e., 30% of its nominal value). Next, a number of discharging/charging cycles can be performed by using different current values. From these experiments the cycle efficiency, η_{cycle} , can be studied:

$$\eta_{cycle} = \frac{E_{\text{discharge}}}{E_{\text{charge}}} = \frac{i_{\text{discharge}} \int_{t_1}^{t_2} V_{\text{discharge}} dt}{i_{\text{charge}} \int_{t_1}^{t_2} V_{\text{charge}} dt} \quad (3)$$

This parameter is usually a function of the electrochemical cell, its nominal values and the performed cycles during the measurements.

As can be noted, the system provides great versatility in the preparation of the teaching lab. Moreover, the depth of the analysis of the results is an additional feature to be considered in the design of experiments for characterization of electrochemical cells. These characteristics make the system suitable for different educational levels (from undergraduate to graduate).

5. Results

The obtained results for the tested electrochemical cells with the modified galvanostat are very similar to those ones obtained with the previous design[1], but with higher

precision of the results, as stated before.

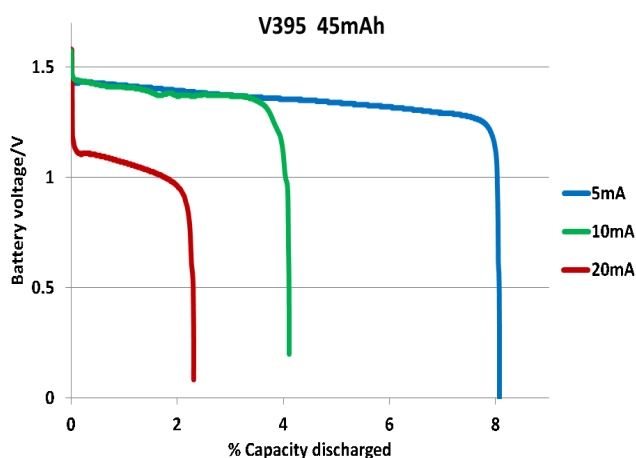


Figure 6. Discharge plots for a primary electrochemical cell, V395

Figure 6 shows the behavior of a primary electrochemical cell with low capacity ($Q_T = 45$ mAh), when relatively high discharge currents are used: 5 mA, 10 mA and 20 mA. The time elapsed for the complete discharge is an exponential function of the applied current. To permit the analysis of the behavior of the electrochemical cell at a glance, Figure 6 plots the electrochemical cells voltage vs. the percentage of discharge. For the commercial V395 ($V_{eq} = 1.55$ V), the theoretical maximum energy stored can be calculated according to Eq. (1), that is: $E_T = 250$ J. The practical available energy depends on the discharge current and it can be calculated from Eq.(2). For the worked example, $E_P = 15$ J, 5 J and 1 J for 5mA, 10 mA and 20 mA, respectively. The

cell efficiency, η , can be easily obtained. From these experiments, it can be noted that higher currents imply a lower percentage use of the nominal electrochemical cell capacity, as should be expected.

Figure 7 shows the typical behavior for a couple of charging/discharging cycles of a secondary Ni-Cd electrochemical cell for different values of currents (10 mA and 25 mA). The cycle efficiency, η_{cycle} , can be calculated from Eq. (3), being for this experiment very close to 95%.

It can be also interesting to explore the power plot behavior for different kinds of commercial cells, from the acquired voltage and current data with the experimental system. Figure 8 shows the obtained results for the V395 and the NiCd140 electrochemical cells. These experiments have been performed integrating the electrochemical cell voltage along a specific time interval while the current is increased until the electrochemical cell voltage suddenly drops to zero. The current steps and time interval used are 1mA and 10 seconds for V395 and 5mA and 60 seconds for NiCd 140mA. It can be noticed that V395 cell has a maximum P about 40 mW, while the NiCd140 offers its maximum P value around 100 mW. It can be pointed out that the optimal performance recommended by the electrochemical cell manufacturer is located under the 50% of the maximum P .

Importantly, the combination of the PLX-DAQ controller and the Arduino system could also provide a basis for data acquisition with other kinds of sensors (gases, temperature, humidity, pH, ...) which can be easily interfaced with it, at low cost.

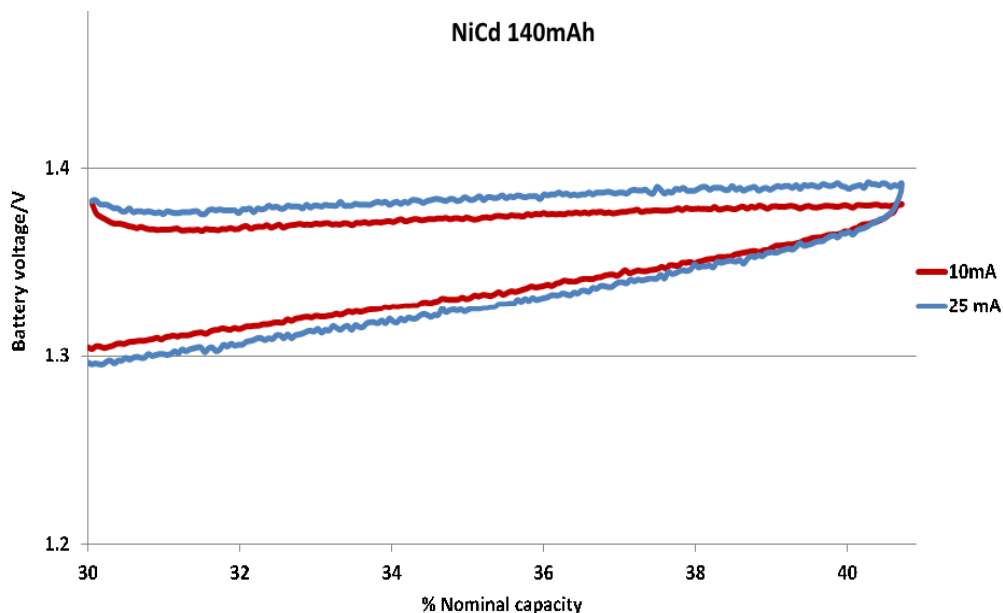


Figure 7. Response of a Ni-Cd electrochemical cell (1.2 V, 140 mA) for charging/discharging cycle at different current values

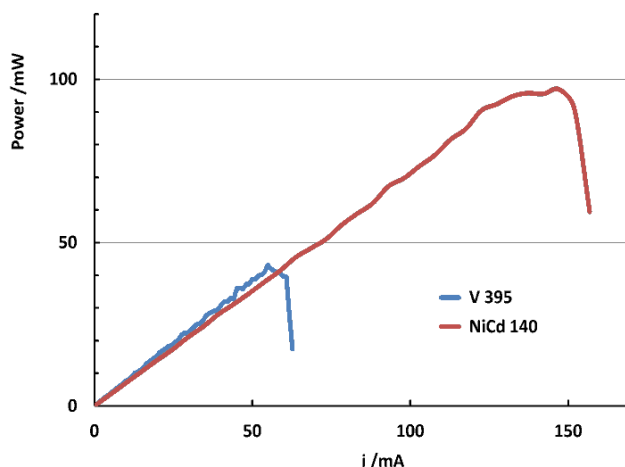


Figure 8. Power plots for V395 and NiCd 140mAh electrochemical cells

6. Conclusions

Several improvements have been included in the revised design of a previously proposed galvanostat, which permits a higher precision of data by using an external voltage reference and a higher current range to investigate secondary electrochemical cells of high capacity. On the other hand, the measuring system has been linked to Microsoft Excel allowing direct data acquisition and real time plotting on the spreadsheet by using the PLX_DAQ module.

ACKNOWLEDGEMENTS

The authors thank Prof. W. H. Mulder for his helpful comments and suggestions.

REFERENCES

- [1] D. González-Arjona, E. Roldán González, G. López-Pérez and M. M. Domínguez Pérez, "Versatile instrumental assemblage for the study of commercial electrochemical cells," *Chem. Educator*, vol. 17, pp. 100-104, January, 2012.
- [2] <http://arduino.cc/en/Main/Software> (accessed March 2013)
- [3] <http://www.parallax.com/tabid/393/Default.aspx> (accessed March 2013)
- [4] <http://www.maximintegrated.com/datasheet/index.mvp/id/1700> (accessed March 2013)
- [5] <http://arduino.cc/en/Main/Software> ; <http://arduino.cc/en/Main/Buy> (accessed March 2013)
- [6] <http://processing.org/> (accessed March 2013)
- [7] <http://www.eltima.com/products/rs232-data-logger/> (accessed March 2013)
- [8] <http://www.mikmo.dk/gobetwino.html> (accessed March 2013)
- [9] <http://www.aggsoft.com/asdl-rs232-to-excel.htm> (accessed March 2013)
- [10] <http://www.activexperts.com/serial-port-component/> (accessed March 2013)
- [11] <http://robbottini.altervista.org/arduino-and-real-time-charts-in-excel> (accessed March 2013)
- [12] http://www.varta-microbattery.com/applications/mb_data/documents/data_sheets/DS20395.pdf (accessed March 2013)