PALM- BASED INTERFACE FOR PORTABLE POTENTIOSTAT

Luiz Elidio de Menezes [1] *, Juliana Lopes Cardoso* [2] *and Marcelo Bariatto Andrade Fontes* [1]

[1]Faculdade de Tecnologia de São Paulo – FATEC/SP - CEETEPS,
Praça Cel. Fernando Prestes, 30 – Bom Retiro 01124-060
[2]Laboratório de Sistemas Integráveis (LSI) –EPUSP
Av. Prof. Luciano Gualberto, 158, trav. 3 – Cidade Universitária
São Paulo, SP, Brasil, 05508-900

ABSTRACT

In this paper we present the development of a PalmOS program using the compiler OnBoardC to perform voltammetric analysis. The voltammetric analyses are based on applying specific waveforms potential and recording theirs output, through a potentiostat circuit.

The program sends the data relative to the analysis via RS-232 to a PIC that controls the potentiostat, displays the recorded data in the Palm's LCD and saves the data into a memo.

1. INTRODUCTION

This program drives the operation of a microcontroled potentiostat, giving it the analysis parameter, proportional to the user defined values in a format that optimizes the sweep, receive the results, displays it ad save for future visualization.

Potentiostats are amplifiers that control the voltage between two or three electrodes, commonly immersed in liquids, called work electrode, reference electrode, and auxiliary electrode.

Potentiostats determines experimentally polarization waveforms of elements in a solution. The potentiostat allows us to range, anodically or cathodically, the solution potential in respect to reference electrode. For each voltage value applied on input, the result is a current that the potentiostat supplies. The element concentration is proportional to the detected current[1]. Then, the potentiostat is the tool used to perform a voltammetry technique, as cyclic voltammetry.

The cyclic voltammetry is perhaps the most versatile electroanalytical technique. With cyclic voltammetry is possible to perform both quantitative and qualitative analysis. Because it needs less hardware to be performed than other qualitative and quantitative techniques (e.g.: chromatography), this technique is ideal to be performed by portable devices, and for this reason, this analysis is the very first implemented in this program.

This technique (Figure 1) consists in scanning the applied potential in the solution, in order to oxidate or to reduce the substance and measure the current through the solution.

In a first moment, the potential is applied, but no current is measured because no redox reaction is generated so far. As the potential increases, the current begin to growth because ions were generated, this happens until a point of maximum, named reduction potential. From this potential and beyond, the current begins to fall due to the reduced material in the electrode surface make difficult to the ions reach it. This potential is named reduction potential [2].

After this, the current would stabilize if the potential were not reversed. When the potential is reversed, after a determined point, the reduced material begins to oxidate and leave the electrode surface, making the current to growth until a second potential where the current begins to fall again, due to the oxidized material in the electrode. This second potential is the oxidation potential, and the pair oxidation potential-reduced potential is unique for each material. The concentration of the studied material in the solution is proportional to the peak currents, which in ideal reversible reaction are the same [2].
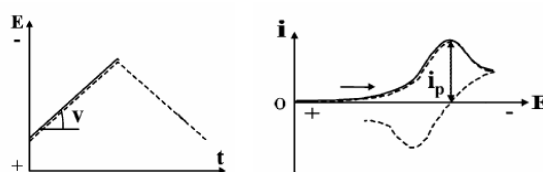

Figure 1 – Cyclic voltammetry technique [2]

2. PALM PROGRAM DEVELOPMENT

OnBoardC is a standard C-based event oriented language. The program waits the user-generated events and responds to them.[3]

2.1. Initialization and Event Loop

When the user opens the application, the program opens the main form, allowing to the user to choose the voltametric analysis that will be performed. Then it gets into the event loop, where, the following actions are performed:

- Wait for an event for 10ms;
- If the serial flag is set, wait for 7 bytes from the serial for 100ms and handle them;
- If the analysis flag is set, refresh the points count;
- Handle the event;
- If it is the AppStop event, exit the loop, and the program.

2.2. Entering the parameters

After choosing the analysis, a new form will be

opened (Figure 2) where the analysis parameters should be defined. The analysis flag is set, so between each event the number of analysis points is calculated according to the given parameters, and some default values are set into the fields in order to avoid null memory handler errors.

The user also can choose in this form how the data will be saved. The user can start the analysis by pressing the go button.
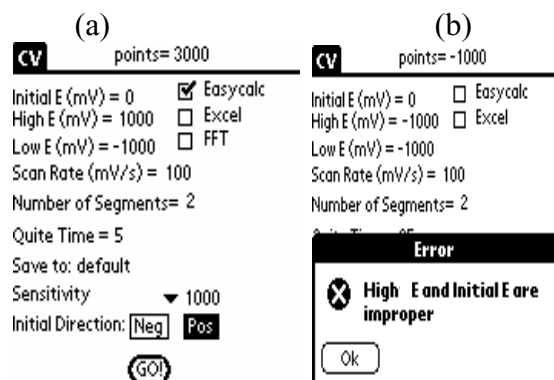

Figure 2 – (a) Parameters Screen, and (b) Error message example.

2.3. Sending data

When the go button is pressed, the parameters are converted to an optimized format for PIC, and formatted into a protocol of 21 bytes. Some tests are made to verify the validity of the parameters. If there is no errors, the serial port is opened and the data will be send, the Graph form will be loaded, the serial flag will be set, and the analysis flag unset.

2.4. Receiving the analysis results

The data is showed in the Graph form. It has, as default, axis X and Y, which correspond to potential and current respectively. The receiving is made in the event loop.

During a pre-determinate period, the PIC will send back 7 bytes containing information relative to potential, current, time and sensibility. When this information is got, a test is made to verify if it is the last point. Else, a pixel is set in the screen, in a position proportional to the potential and the current times sensibility in the axis. The information is saved into a memo, or into an array if the data should pass through a FFT filter for noise treatment.

If the potential, current and time information where set to 0xffff, indicating that the analysis is finished, the serial port and the memos must be closed, and the serial flag must be unset. This value was elected because it is a value impossible to come from the solution, because the ADC is a 12-bit converter.

The graph can be seen in the screen, and can be accessed later by the memos using EasyCalc (a common graph calculator software for PalmOS) or after the hotsync process, can be retrieved by Origin, Excel or any data editor.

3. RESULTS AND DISCUSION

The preliminary tests verified the communication between the Palm and the PIC. The results showed a stable communication speed at 57,600 kbps. This high value permits a future implementation of fast voltammetric analysis.

In earlier versions, the program was not responding during the analysis, because the Palm M100 is unable to open background connection.

Implementing the serial receiving during the event loop, alternating turns with the event waiting, solved the problem. In this approach, some points of analysis might be lost, if this data reach when the program is in awaiting events. Since the program, during the data receiving, spend 1/11 of the time waiting events, the chance of a point be lost is 9,09%. Since the lost points is uniformly spread trough the analysis; and low resolution analysis gets at least 1000 points, it might not interfere in the graph quality.

Using larger scan rates, with minus reads per second without changing the number of acquired points, and thus spreading more the lost points this effect can be minimized.

4. CONCLUSION

We presented a palm-based program that controls a potentiostat using RS-232 protocol, display the results into the screen and save them for future viewing in the palm or in a desktop.

This program was implemented to perform CV analysis and might be extended to perform any voltametric or amperometric analysis in future works.

5. AKNOWELDGEMENTS

6. REFERENCES

[1] Bank Eletronik, Inteligent Controls GmbH, www.bankic.de/support/download/doku/potstae2.pdf, 2003.

[2] Buffle, J.; Teecier-Waeber, M.-L. Voltammetric environmental trace-metal analysis and speciation: from laboratory to in situ measurements, Trends in Analitical Chemistry, 24, 3, p. 172-191, 2005.

[3] The Palm Programmer's Cookbook. V 0.7.2, http://onboardc.sourceforge.net/cookbook.html 26/08/2006