MethodSCRIPT SDK Example - Python

www.palmsens.com

1   Contents

The example *MSConsoleExample.py* found in the */MethodSCRIPTExample_Python* folder demonstrates basic communication with the EmStat Pico using Python.

The *PsEsPicoLib.py,* custom library contains some commonly used functions for communication with the EmStat Pico.

The *MSPlotCV.py* example demonstrates the common electrochemical technique, Cyclic Voltammetry and plots the resulting voltammogram.

The *MSPlotEIS.py* example demonstrates the Electrochemical Impedance Spectroscopy technique and plots the resulting Nyquist and Bode plots.

2   Examples:

2.1   Example 1: Console Example (MSConsoleExample.py)

This example opens a communication port, sends a MethodSCRIPT file, reads and parses the device responses and prints the parsed data (variable type, value, unit) to the console. The metadata (status, current range) are not parsed in this example.

The name of the com port connected to the EmStat can be looked up in the Device manager in Control Panel in Windows, as shown below.
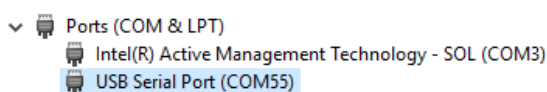

Figure 1: Available com ports in device manager

This example, uses COM55 as the EmStat Pico com port, opens the com port and checks if an EmStat Pico is connected. If an EmStat Pico is connected, the version is printed, a MethodSCRIPT file is send to the device and the returned measurement packages iare parsed into variable types with their corresponding value and unit.

Here's a sample measurement data package in response to a MethodSCRIPT, *MSExampleCV.mscr*, on a Palmsens dummy cell (10 kOhm) and its corresponding output.

Pda7F8797Du;ba7F87CFBn,10,288

Output: Applied potential=-0.493187 V
        WE current=-0.000492293 A

2.2   Example 2: Cyclic Voltammetry Plot Example (MSPlotCV.py)

This example performs a Cyclic Voltammetry (CV) on a Palmsens Dummy Cell WE A (RedOx circuit) and plots the I vs E curve.

The first part of the example connects to the EmStat Pico, sends the MethodSCRIPT file "MSExampleCV.mscr", reads the data and saves it to a resultfile with a .dat extensionin the *data* subfolder. The second part of the example reads the result file, parses the data to a value matrix and plots the I vs E curve as shown below.
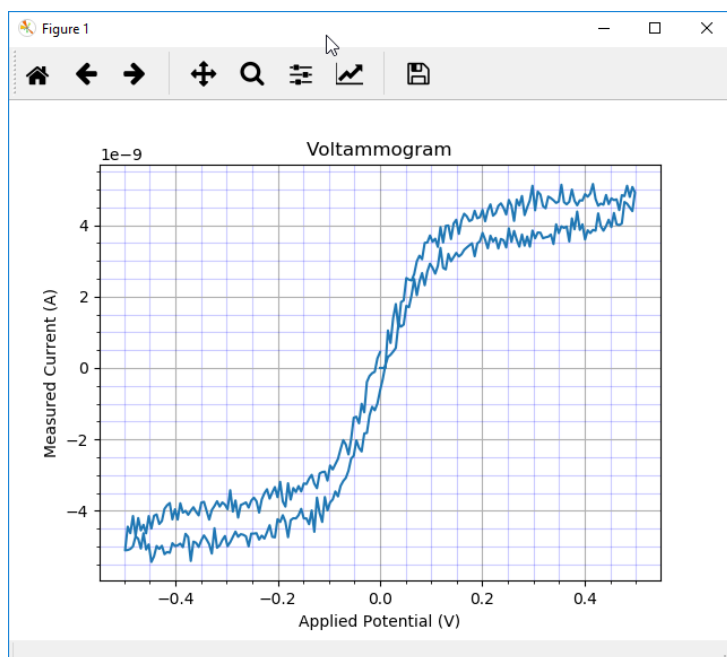
Figure 2: CV on PalmSens Dummy Cell WE A (RedOx circuit)

2.3 Example 3: EIS Plot Example (MSPlotEIS.py)

This example performs an EIS (Electrochemical Impedance Spectroscopy) scan on a PalmSens Dummy Cell WE C (Randles circuit) and generates a Nyquist plot and a Bode plot.

The parsed values are stored in a matrix, the *value_matrix,*where the first column (0) holds the applied frequencies, the second (1) holds the real part of the complex impedance and the third (2) holds the imaginary part of the complex impedance. The complex impedance is composed from the real and imaginary parts and the absolute impedance (Z) and phase are calculated from the complex impedance, *zcomplex*. A sample Nyquist plot and a Bode plot for an EIS scan on a dummy cell with Randles circuit are shown below.
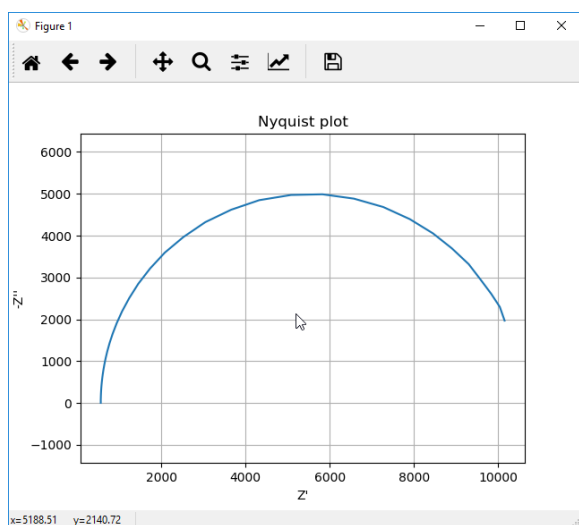

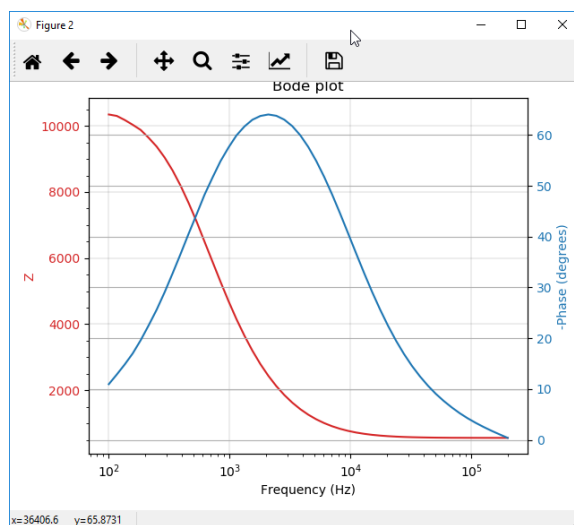*Figure 3: Sample plot EIS: Nyquist Plot*

Figure 4: Sample plot EIS: Bode Plot

2.4 Example 3: Advanced SWV Plot (MSPlotAdvancedSWV.py)

This example showcases some of the more advanced features of the PSEsPicoLib. The example runs a script that contains two SWV measurements. It can also be easily converted to parse any MethodSCRIPT output by changing the plotted columns in the configuration section. This example is meant to be run on the PalmSens RedOx dummy cell (WEA).

To convert the example to another MethodSCRIPT, simply change the following 3 lines to specify what data columns to plot on the x and y axes:

```
#column index to put on the x axis
xaxis_icol = 0
#column indices to put on the y axis, can be multiple but must be same type
yaxis_icols = [1, 2, 3]
#names for each column
yaxis_col_names = ['Potential', 'Current', 'Forward Current', 'Reverse Current']
```
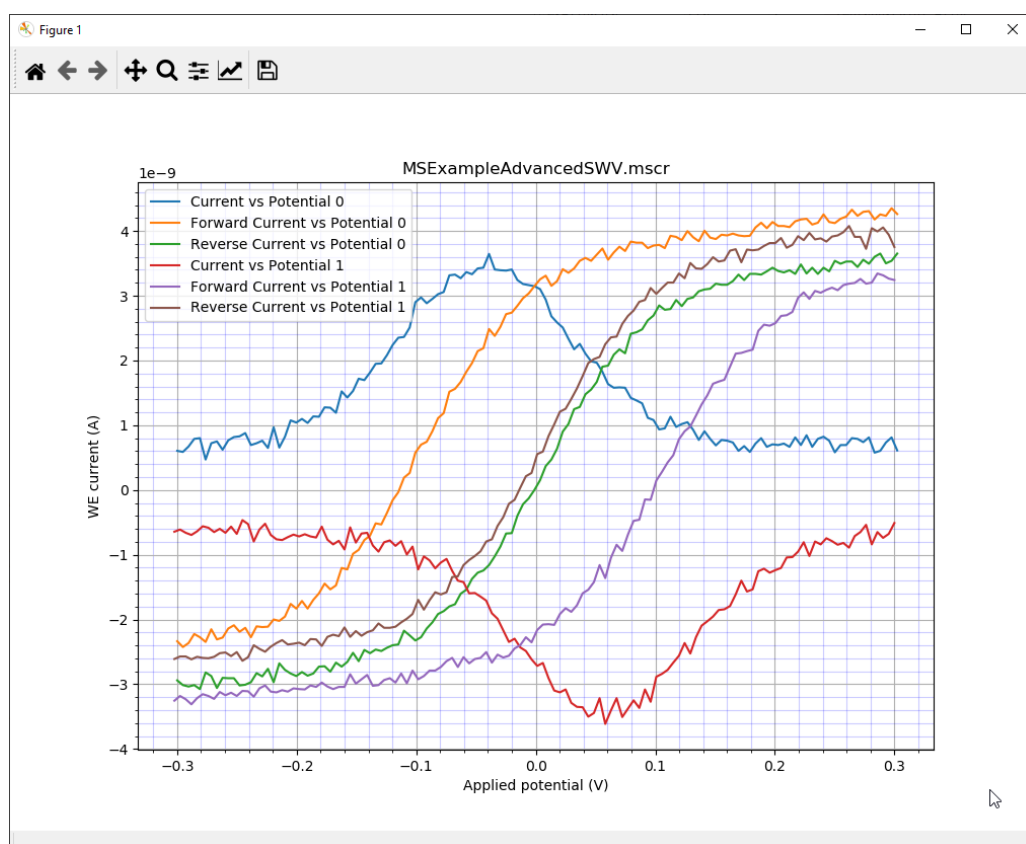

Figure 4: SWV plot example

3 Communications

The examples have been developed using the Spyder IDE as part of the Anaconda distribution.

3.1 Connecting to the device

The examples use the *serial* library for serial communication with the device.

3.2 Sending a MethodSCRIPT

The MethodSCRIPT can be read from a txt file stored in the *MScriptFile* and then sent to the device. In this example the MethodSCRIPT files are stored in the *MethodSCRIPT files* directory.

3.3 Receiving the measurement packages

Once the script file is sent to the device, the measurement packages can be read continuously, line by line from the device as shown below.

```
response = ser.readline()
```

3.4 Parsing the response

The measurement data packages returned by the method *ser.readline()*, can be parsed further to obtain the actual data values. Here's a set of data packages received from a Linear Sweep Voltammetry (LSV) measurement on a dummy cell with 10 kOhm resistance.

```
eMoooo\n
Pda7F85F3Fu;ba48D503Dp,10,288\n
Pda7F9234Bu;ba4E2C324p,10,288\n
Pda806EC24u;baAE16C6Dp,10,288\n
Pda807B031u;baB360495p,10,288\n
*\n
\n
```

While parsing a measurement package, various identifiers are used to identify the type of package. For example, In the above sample,

1. 'e' is the confirmation of the "execute MethodSCRIPT" command.
2. 'M' marks the beginning of a measurement loop.
3. 'P' marks the beginning of a measurement data package.
4. "*\n" marks the end of a measurement loop.
5. "\n" marks the end of the MethodSCRIPT.

The data values to be received from a measurement can be sent through 'pck' commands in the MethodSCRIPT. Most techniques return the data values Potential (set cell potential in V) and Current (measured current in A). These can be sent with the MethodSCRIPT.

In case of Electrochemical Impedance Spectroscopy (EIS) measurements, the following *variable types* can be sent with the MethodSCRIPT and received as measurement data values.

- Frequency (set frequency in Hz)
- Real part of complex Impedance (measured impedance Ohm)
- Imaginary part of complex Impedance (measured impedance in Ohm)

The following metadata values if present can also be obtained from the data packages.

- CurrentStatus (OK, Underload, Overload, Overload warning)
- CurrentRange (the current range in use)
- Noise