MethodSCRIPT SDK Example





Last revision: March 29 2019

© 2019 PalmSens BV

www.palmsens.com

## Contents:

* PSEsPicoLib.py
* MSConsoleExample.py
* MSPlotCV.py
* MSPlotEIS.py
* \MethodSCRIPT files\ MSExampleCV.mscr
* \MethodSCRIPT files\ MSExampleEIS.mscr

The examples are based on Python under Windows using the Spyder IDE as part of the Anaconda distribution. The example “MSConsoleExample.py” found in the “/MethodSCRIPTExample-Python” folder demonstrates basic communication with the EmStat Pico using Python. 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.

### 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 meta data (status, currentrange) is not parsed in this example.

A connected EmStat Pico will create a comport with a generated (enumerated) port number. To find this number one can look in the Windows device manager.

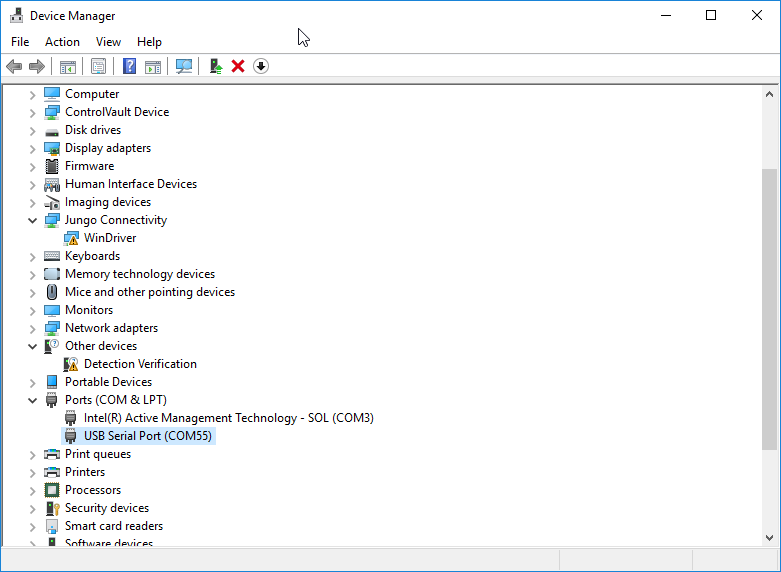


Figure 1; Available com ports in device manager

The example uses COM55 as our EmStat Pico comport and set the myport variable accordingly, opens the comport and checks is 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 data is fetched and parsed into values and the corresponding variable-type and unit. A custom library (PSEsPicoLib.py) is included containing commonly used functions. The response from the send MethodSCRIPT file "MSExampleCV.mscr" to the console looks like :

*read data: Pda7F8797Du;ba7F87CFBn,10,288*

*Applied potential=-0.493187 V*

*WE current=-0.000492293 A*

Where the string after “read\_data:” is the unparsed response from the EmStat Pico and the next 2 lines the parsed variable-type , value , unit. The next chapter explains more about the parsing of the response.

### 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.

eM0000\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 at the moment)
* Noise (Noise)

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

This example performs a Cyclic Voltammetry (CV) and plots the I vs E curve.

The shown plot is the result when the PalmSens Dummy Cell WE A (RedOx circuit) is used.

The first part of the example connects to the EmStat Pico, sends the MethodSCRIPT file "MSExampleCV.mscr" , captures the data and saves the data to a result-file in the “data” subfolder.

The resultfile has a unique filename composed from the MethodSCRIPT filename, date and time having the “.dat” extension. The second part of the example reads the resultfile, parses the data to a value matrix and plots the data as I vs E curve. Figure 2 shows the resulting plot when performed on the PalmSens Dummy Cell WE A (RedOx Circuit).

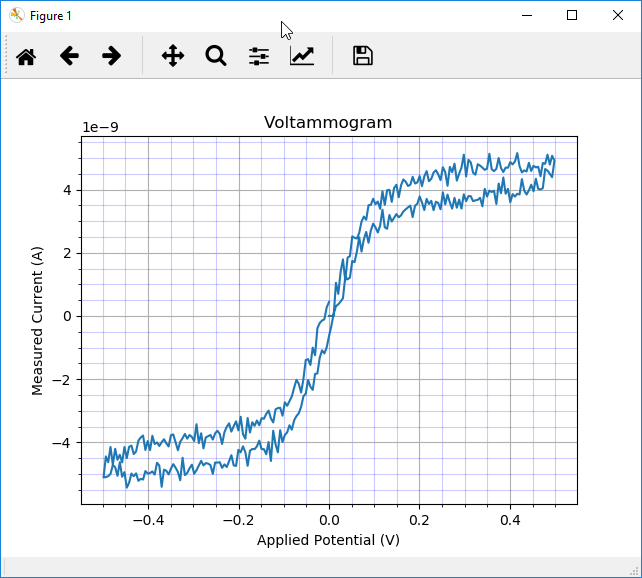


Figure 2; CV on PalmSens Dummy Cell WE A (RedOx circuit)

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

This example performs an EIS scan and generates a Nyquist plot and a Bode plot.

The shown plots are the result when the PalmSens Dummy Cell WE C (Randles circuit) is used.

The first part of the code connects to the EmStat Pico, sends the MethodSCRIPT file and saves the results to the filename given by ResultFile using the functions from PSEsPicoLib.py.

The second part parses the data stored in ResultFile to a matrix given by 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 . Nyquist plot and the Bode plot are generated accordingly. At last the results are saved to a Comma Separated File enabling the results to be imported in other applications like Excel.

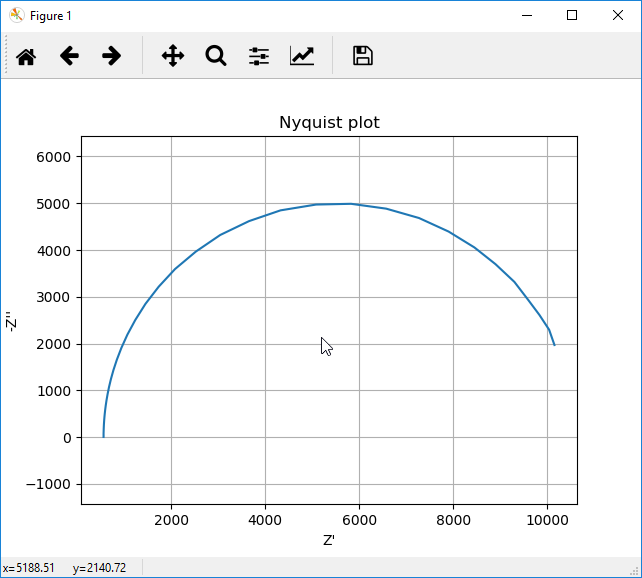


Figure 3; Nyquist plot of PalmSens Dummy Cel WE C (Randles circuit)

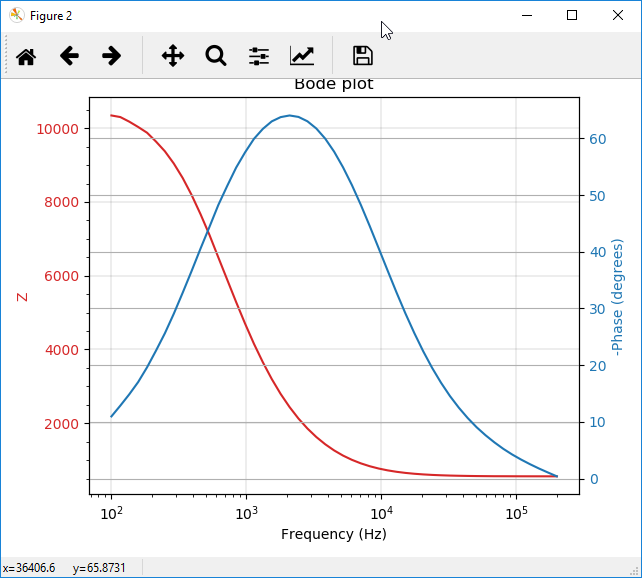


Figure 4;Bode plot of PalmSens Dummy Cel WE C (Randles circuit)