

## 1 Software support

PSTrace 5.8 does not support automatic switching of the Pico MUX16 in measurements or manual control. Instead, [MethodSCRIPT™](#) can be used to control the selected channel.

The MethodSCRIPTs found with the Python examples (.mscr files), can also be run directly in PSTrace 5.8 using the MethodSCRIPT Sandbox. However, PSTrace will not put each channel into a separate curve.

## 2 Addressing the MUX16

There are two ways of addressing the mux:

- Mux16 mode where WE and RE/CE are both switched at the same time.
- Mux256 mode where WE and RE/CE are switched separately, allowing 256 channels to be addressed as a matrix.

The mux is controlled using 10 digital IO lines on the EmStat Pico module.

The following pins of the EmStat Pico are used, in order of Most Significant Bit (MSB) to Least Significant Bit (LSB):

ADDRESS BIT	9	8	7	6	5	4	3	2	1	0
FUNCTION	MUX enable		RE and CE select				WE select			
BIT IN SUBSET	1	0	3	2	1	0	3	2	1	0
EMSTAT PICO PIN	IC2_SDA	I2C_SCL	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

Note that the two IC2 pins are not used for I2C comms, but are used as GPIO for this purpose. All 10 pins must always be configured as outputs (1) by using the MethodSCRIPT command:

```
set_gpio_cfg 0x3FFi 1
```

The two MSB of the address control the two muxes (RE/CE and WE). These bits are inverted, so 0=enable and 1=disable. The next two bits switch the RE and CE mux. The four LSB switch the WE mux. Addresses range from 0 to 15, whereas the labels of the RE, CE and WE range from 1 to 16, so there is an offset of 1 when selecting the correct address.

For example: to select RE3/CE3 and WE2 (WE1/RE1/CE1 being the first electrodes), the following address should be used:

00 0010 0001 = 0x21 (Hex) = 33 (decimal).

To select RE5/CE5 and WE5, the following address should be used:

00 0100 0100 (binary)= 0x44 (Hex) = 68 (decimal).

The following script can be used to set the channel in MethodSCRIPT (\n represents a newline):

```
e\n#configure all GPIO pins as outputs\nset_gpio_cfg 0x3FFi 1\n#Select RE2/CE2 and WE2, add "i" to indicate it is an integer\nset_gpio 0x11i\n\n
```

## 2.1 Using PStace Sandbox

The scripts used in the Python example can also be used in the PStace MethodSCRIPT sandbox.

In case of PStace 5.9 and later, make sure you have enabled the ‘Show MethodSCRIPT editor’ checkbox in the Settings window of PStace, see figure below. When enabled, a MethodSCRIPT button appears underneath the Method Editor.

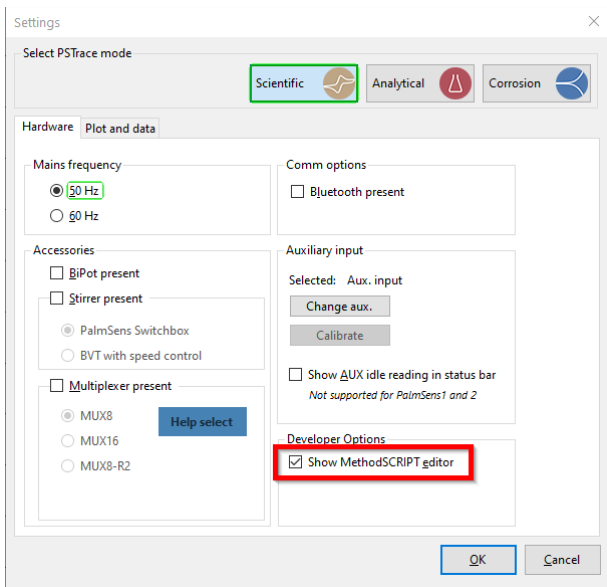


Figure 1 - Enable MethodSCRIPT editor in PStace

The MethodSCRIPT editor can also be opened via the MethodSCRIPT Sandbox.

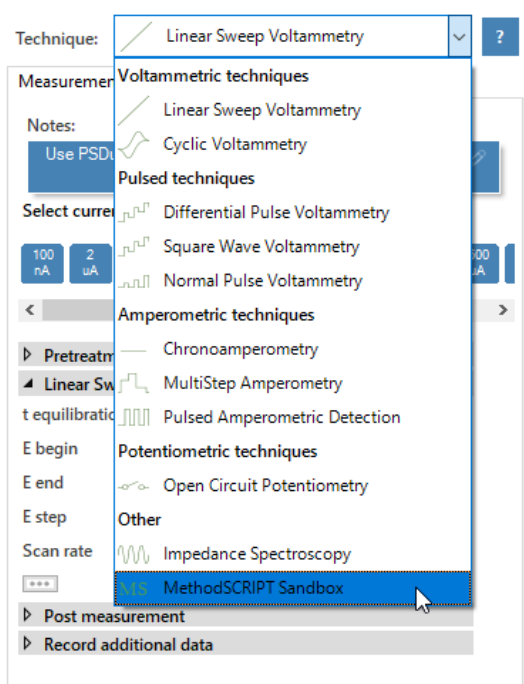


Figure 2 - MethodSCRIPT Sandbox in the Method Editor in PSTrace

The MethodSCRIPT Sandbox is listed as a separate technique in the Method Editor. It shows a single button that opens the MethodSCRIPT Editor. The script shown in the editor can be run directly in PSTrace. Using the Legend at the right of the plot, different curves can be distilled from the returned data.

Secondly, you can setup a measurement with any of the available techniques, and then click the “Show MethodSCRIPT” button below the Method Editor. Then in the MethodSCRIPT Viewer window, click the button “Open in Editor”. This will copy the script to the MethodSCRIPT Sandbox. See figure below.

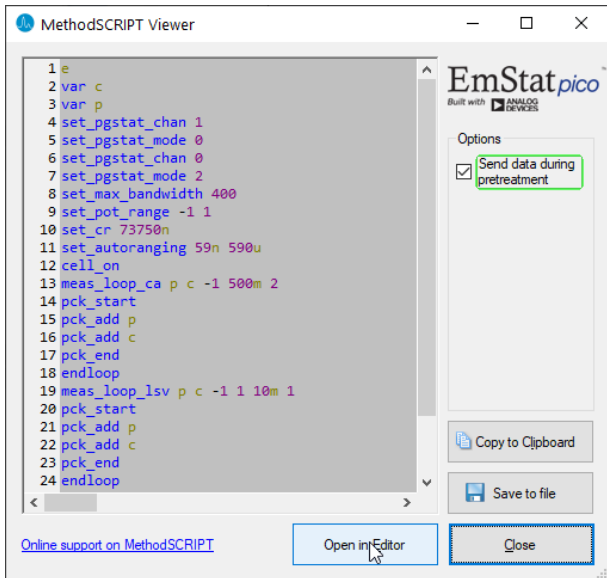


Figure 3 - Open MethodSCRIPT in Editor (Sandbox)

## 3 Example in PStace using the supplied test cell

Your EmStat Pico MUX16 came with a Test Cell with 16 resistors of different values, starting at 1K Ohm for channel 1 and 8.66K Ohm for channel 16. Connect the test cell with the three supplied flat cables.

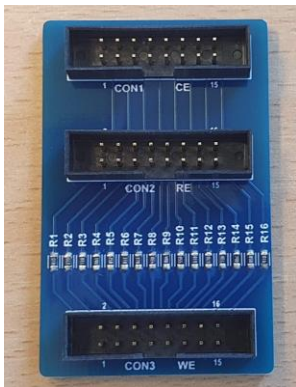


Figure 4 - Test cell for evaluating the EmStat Pico MUX16

By doing an LSV on all channels you can verify your hardware. For this example we use an LSV with the following parameters:

Linear Sweep Voltammetry Settings		
t equilibration	2	s
E begin	-1.0	V
E end	1.0	V
E step	0.01	V
Scan rate	1.0	V/s

When we generate the MethodSCRIPT and add the required lines of code to control the mux, we get the following script for use in the MethodSCRIPT Sandbox:

```
e
set_gpio_cfg 0x3FFi 1
set_gpio 0x11i
var i
var c
var p
store_var i 0i aa
set_pgstat_chan 0
set_pgstat_mode 2
set_max_bandwidth 400
set_pot_range -1 1
set_cr 1m
set_autoranging 10u 1m
cell_on
loop i <= 0xFFi
  set_gpio i
  set_e -1000m
  wait 100m
  meas_loop_lsv p c -1 1 10m 1
    pck_start
    pck_add p
    pck_add c
    pck_end
  endloop
  add_var i 0x11i
endloop
on_finished:
cell_off
```

Note that a MethodSCRIPT is always ended by means of two new lines also referred to as Line Feeds, in programming languages often represented by the character ‘\n’.

The code in blue is specifically added to control the mux, and the green part is to give 100 ms extra time to settle after switching to the next channel.

You can copy/paste the above script into the MethodSCRIPT editor, or you can open the method file (menu: Method → Load) “LSV on test cell.psmethod”, which contains the same script.

When running this script, the plot will show the following result:

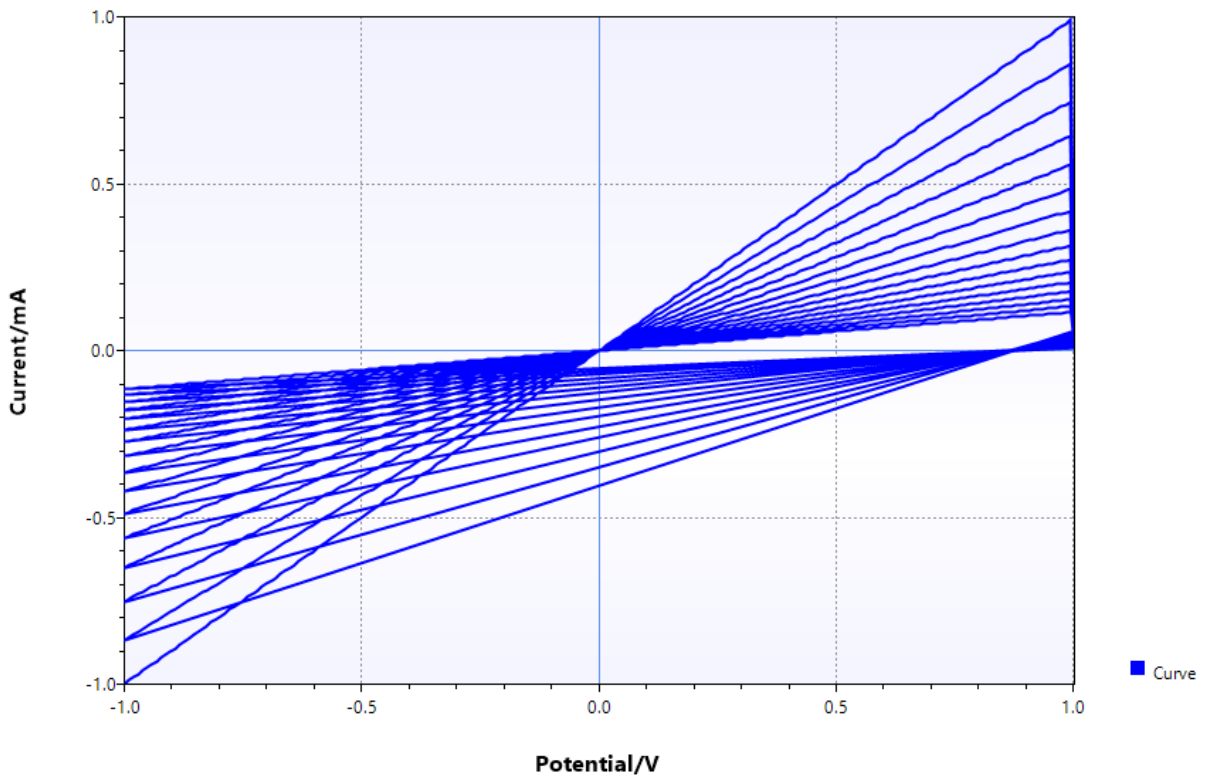


Figure 5 - Result of LSV on test cell

In case automatic peak detection is enabled, you might get the following error message in the plot area: “Unable to draw plot”. This message disappears by removing the peaks found. Using the toolbar at the top of the plot area:

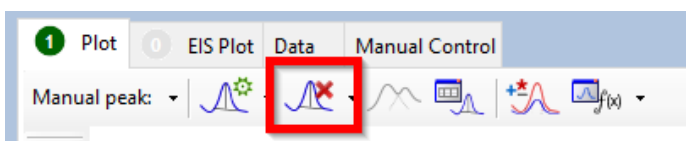


Figure 6 - Remove peaks button

## 4 Using a terminal emulator to send MethodSCRIPT

MethodSCRIPTs can be sent through a terminal emulator or with the Python example. For a terminal emulator we suggest Tera Term. Make sure line endings are set to newline (\n) as shown below:

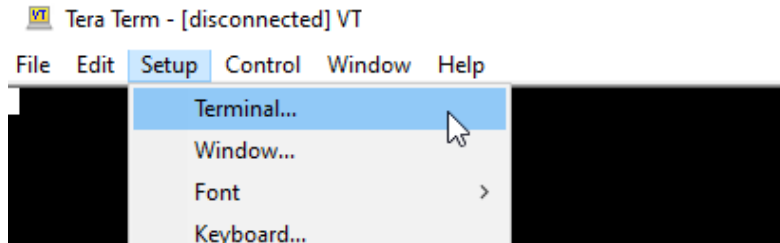


Figure 7 - Opening the terminal setup in Tera Term

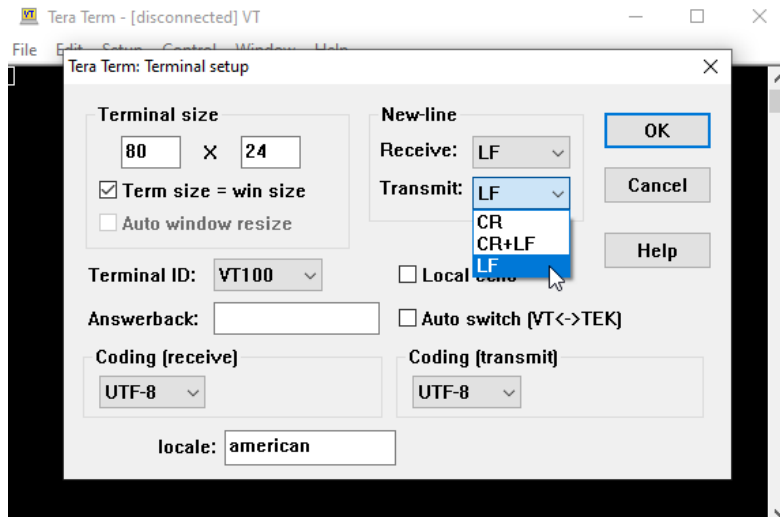


Figure 8 - Terminal setup in Tera Term

## 5 Using Python to send MethodSCRIPT

The python example uses Python 3.7. We used the Anaconda python environment with the integrated Spyder IDE. You may need to install the “pyserial” package. To install pyserial, enter the following command in a command prompt window:

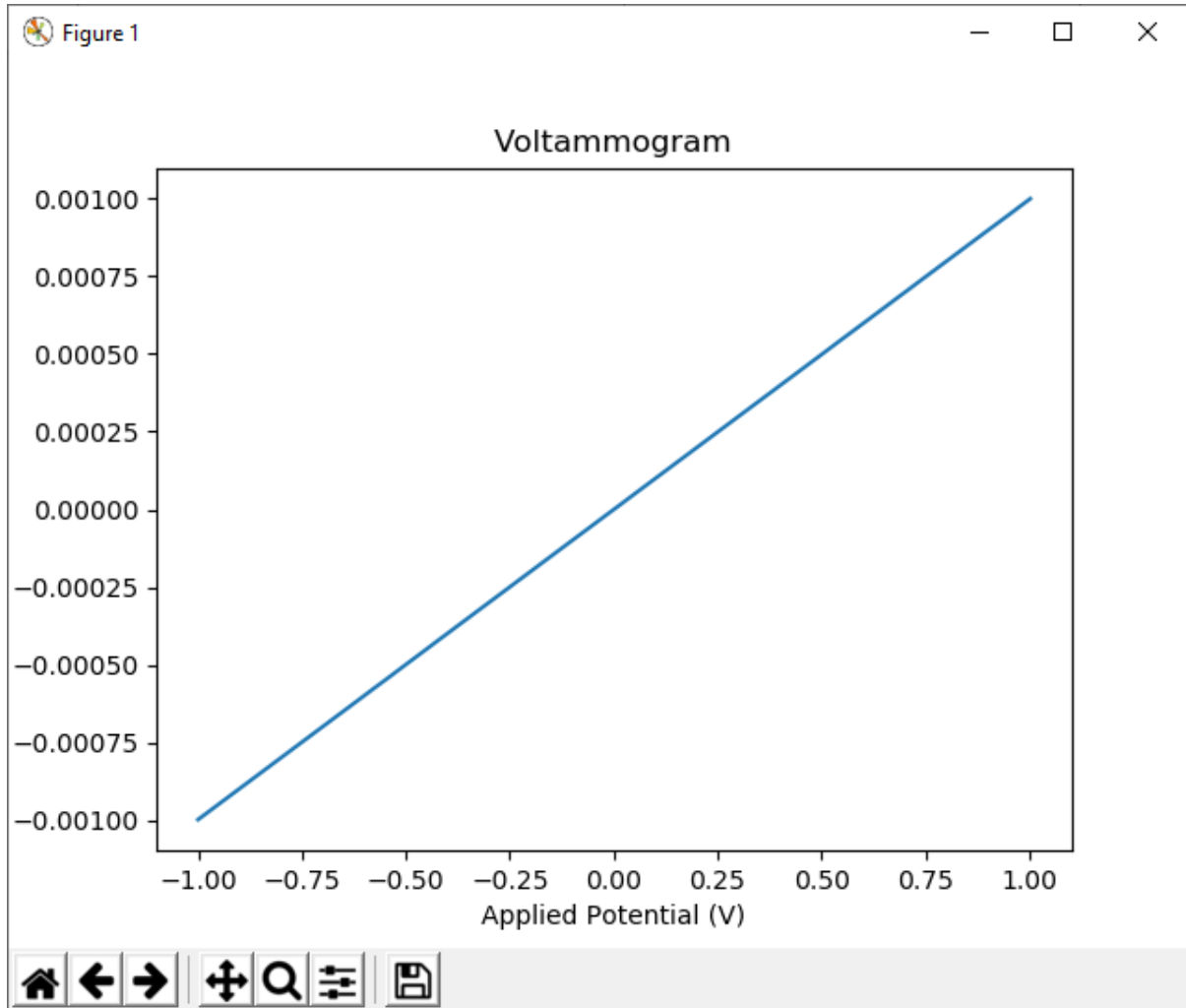
```
pip install pyserial
```

Before running the example you will need to change the COM port in the python code to the COM port of the EmStat Pico. The COM port can be found in the device manager or in Tera Term. To set the com port, edit the following line in your MSPlotMuxLSV16CH.py and MSPlotMuxLSV256CH.py file,

```
myport = "COM33" #set the comport
```

The PSEsPicoLib.py library has been modified in this example to allow the separation of multiple curves.

Running the MSPlotMuxLSV16CH.py with the correct com port will result in the 16 graph windows:





## 6 Hardware connections

The cells can be connected using pin headers or a flat cable to the three box connectors for RE, CE, and WE on the PCB.



Pins: 2 x 8  
Pitch: 0,079" (2,00mm)

Manufacturer: Würth  
Article number: 62501621621  
or

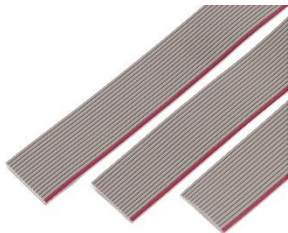
Manufacturer: Assmann WSW Components  
Article number: AWHW2 16G-0202  
Digikey number: AE11088-ND

### Crimp connectors on the flat cable:



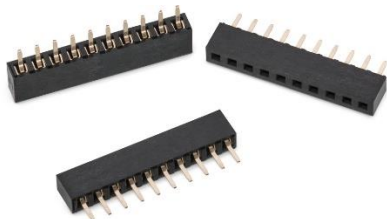
Manufacturer: Würth  
Article number: 62501623021

### Ribbon cable:



Manufacturer: Würth  
Article number: 63901615621CAB

Alternatively, socket headers can be used for soldering on directly:



Manufacturer: Würth  
Article number: 62000811821