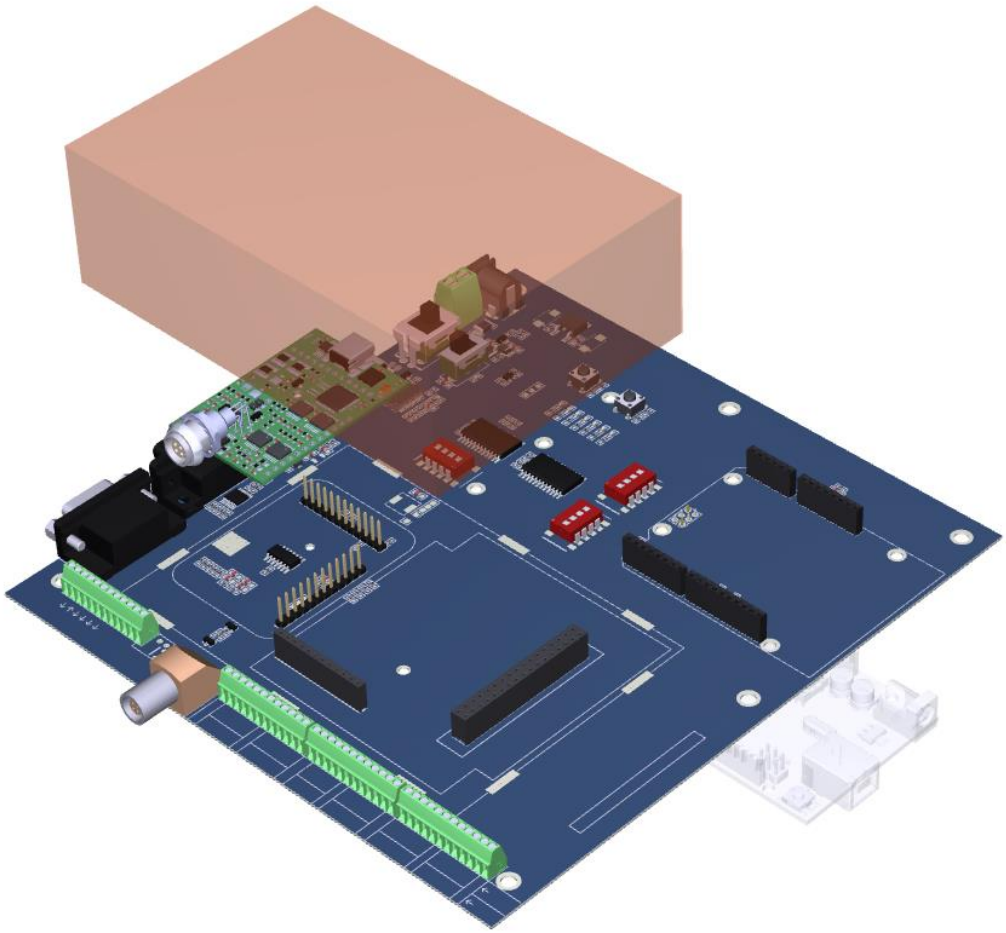
# Getting started with the EmStat Development Board

Development board for EmStat module: OEM Interface for electrochemical sensors



PalmSens
Compact Electrochemical Interfaces

Last revision: October 24, 2017


www.palmsens.com

# Important before you start unpacking

Please read this quick start guide carefully before unpacking the boards from their antistatic bags or using the kit, as it contains important information on how to avoid damage to the PCB's.

The EmStat board can be damaged by static electricity (electrostatic discharge or ESD). Please take adequate precautions against static discharge during handling.

# Customer support

Questions regarding the contents of this document or the products described in this document can be directed to PalmSens BV via:

- the contact form on the website: www.palmsens.com
- an e-mail to info@palmsens.com ,
- or by telephone: +31 30 2459211

## Contents

# 1 EmStat development board features

The EmStat-series are potentiostats with an embedded microcontroller. They provide all the major potentiostatic techniques with automatic current ranging and peripheral control.
The EmStat development board contains the following features:

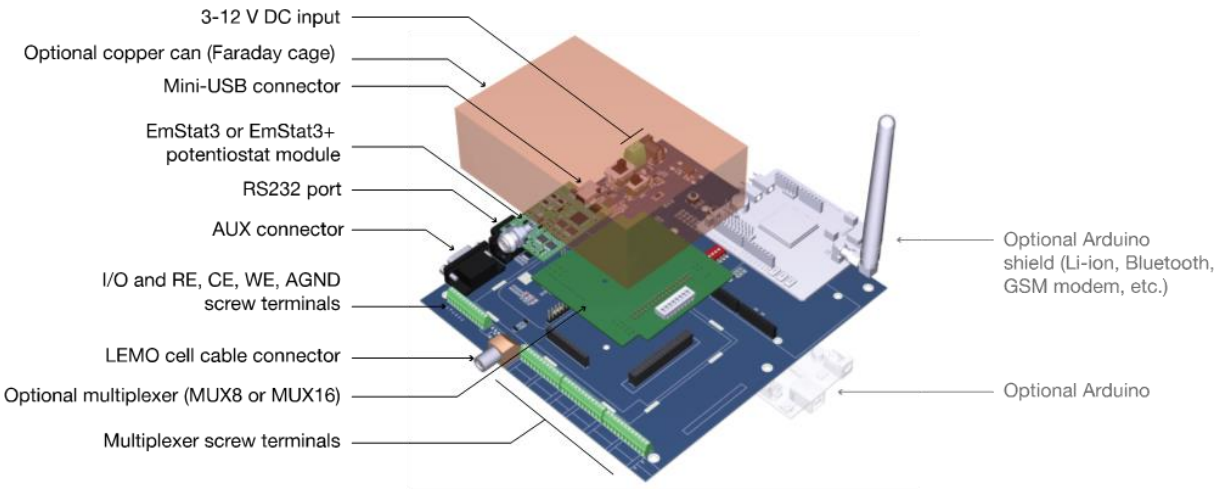



*Figure 1- EmStat Development board features*

> Schematics of the development board can be found on the CD that was supplied with the development board.

## 1.1 Circuit protection

The following circuit protection mechanisms have been implemented in the dev. board:

- All digital IO's have over-voltage, reverse-voltage, over-current, and ESD protection.
- Analog IO's are buffered on-board for impedance isolation, over-voltage and reverse-voltage protection, and ESD protection.
- Valid input supply range of 3-12VDC, with over-current and reverse-voltage protection, and a regulated +5V supply for the EmStat to ensure correct operation and preventing damage.

## 1.2 Power options

The development board can be powered by:

- DC adaptor,
- screw terminals,
- via USB,
- or via the Arduino.

See for more information section “Power supply options” on page 11.

## 1.3 Communication options

The dev. board contains the following interfaces for different forms of communication or triggering:

- USB communications via a mini-USB connector
- RS232 communications via a standard DB9 connector, allows the use of standard RS232 dongles
- An auxiliary port for use with other PalmSens modules
- On-board level shifting options for all digital IO's, between 5V, 3V3, or any externally provided logic level less than 5V.

See for more information section “Communication options” on page 11.

## 1.4 Potentiostat interfaces

The potentiostat can be connected to an electrochemical cell by means of:

- A LEMO connector for use with a PalmSens Sensor Cable;
- A Screen Printed Electrode (SPE) connector;
- On-board standard test sensor option enabled by a switch;
- Pin headers/screw terminals.

## 1.5 Arduino Integration

The EmStat development board is designed specifically for compatibility with an Arduino Zero, Uno or Intel Edison[1].
Various jumper setting available for:

- selecting different UART lines,
- EmStat-DAC to Arduino-ADC,
- EmStat-4Vref to Arduino-AREF,
- EmStat-D0 & D1 to Arduino-INT0 & INT1 interrupts,
- and EmStat-Reset control from Arduino-IO4.

The dev. board can be powered from the Arduino itself, either via sharing the Arduino-DC adaptor voltage directly (3-12V), or sharing power from any shield such as an Li-ion shield.

See also section “Power supply options” on page 11.

## 1.6 Other

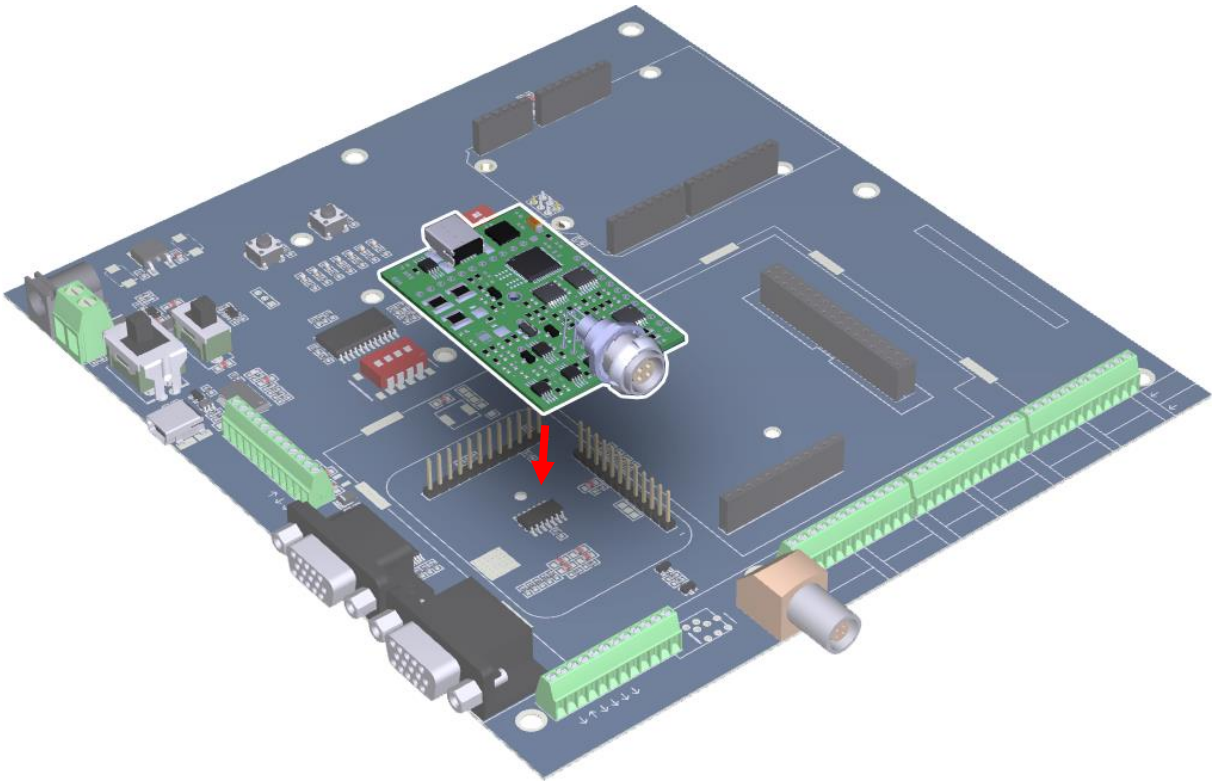
Other features the dev. board contains:

- Option to populate a MUX8 or MUX16 multiplexer, with connections pulled to pinheaders/screw terminals;
- LED indicators for the EmStats digital IO activity: D0-D3, Reset, Download, RxD and TxD lines;
- Push buttons to generate a pulse on D0 or Reset lines;
- DIP switch to break UART connections to RS232, Arduino, USB, or external headers individually as required;
- Can supply up to 200mA of the regulated +5V off-board.

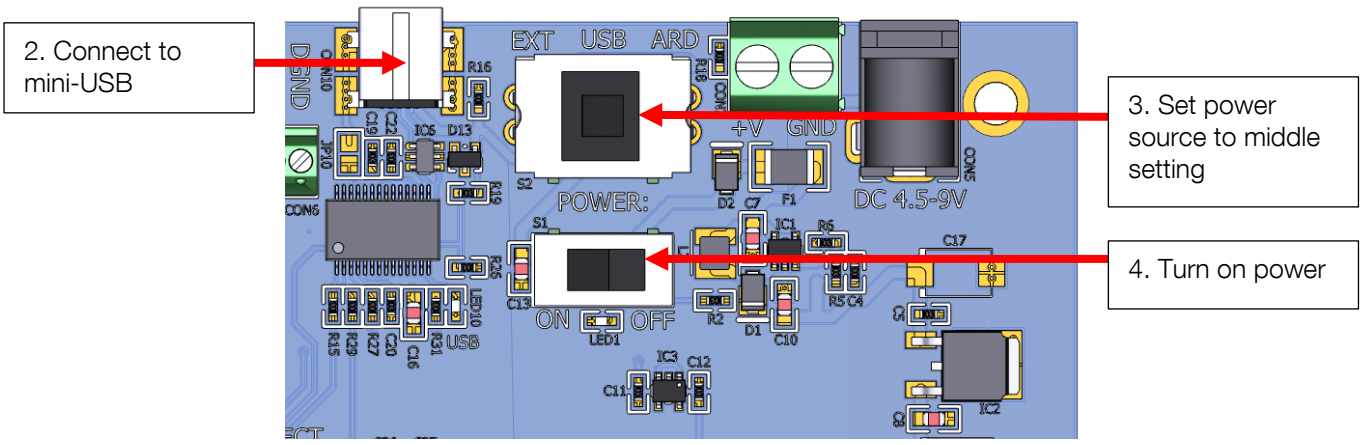
[1] Contact PalmSens BV for more information.

# 2 First measurement using PSTrace

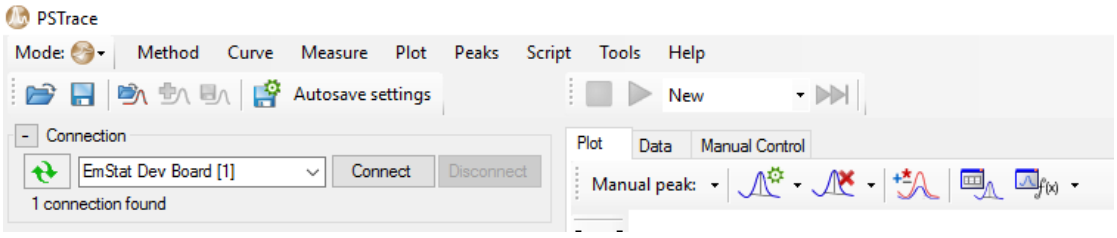
Follow the steps below to run a first measurement in PSTrace on the development board. Make sure PSTrace is installed on your PC before following these steps.



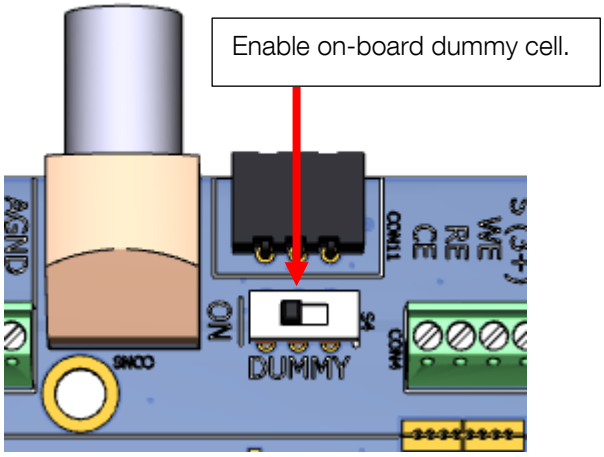
1. Mount the EmStat pin-headers to the header on the development board as shown above.



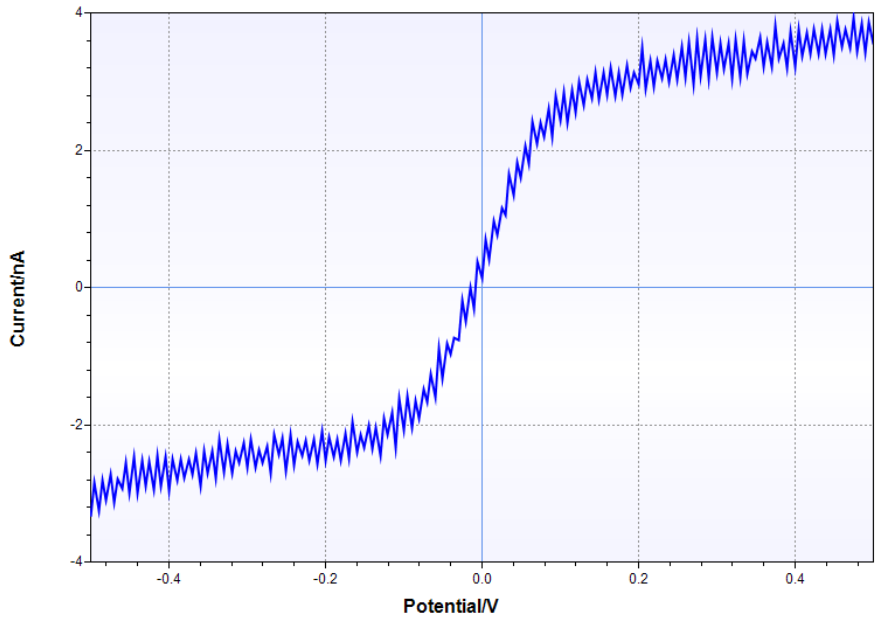

5. Open PSTrace. The Connection box at the upper left corner should show “EmStat Dev Board”. Click Connect.



6. Load the method file “PSNoiseTest.psmethod” from the “My Documents\PSData” folder.\
7. Set the small switch labelled “DUMMY” (next to the LEMO sensor cable connector) to ON.




8. Run the measurement.
9. The result will probably look similar to this:




10. Repeat the measurement but with a scan rate of 0.05 V/s. Note that the noise level is lower, due to the fact that the current sampling time is longer now. This decreases the measured noise level.

Refer to section “
Noise considerations” on page 16 if noise levels seem too excessive.

11. Make sure to switch back the on-board dummy cell to the LEMO when done.

# 3 Getting started with the PalmSens Embedded SDK libraries

The Arduino example (see next section) shows how to use the PalmSens Embedded SDK library for EmStat which can also be used as plain C library for development in C/C++.
Another example for plain C/C++ can be found in the leading text of the header file PSComm.h.

This Embedded SDK consists of the following libraries:

| | |
|---|---|
| PSComm.h | Contains routines for running measurements and receiving readings during idle mode for EmStat devices |
| PSMethod.h | Contains routines for setting up a measurement (like Cyclic Voltammetry, Chronoamperometry, Square Wave Voltammetry, etc.) and its corresponding parameters like Ebegin, Evertex1, Evertex2, scan rate, step potential, frequency, run time, etc. that can be sent to EmStat devices to initiate a measurement. |
| PSCommon.h | Contains some supporting functions, like converting from and to hexadecimal numbers. |

An example on how to use the embedded libraries in plain C/C++ can be found in the leading text of PSComm.h.

For direct communication with EmStat, please refer to the “EmStat Communication Protocol” pdf which can be found on the same CD that was supplied with the dev. board.

# 4 Getting started with Arduino/Genuino

*Arduino or Genuino?*
*Genuino is Arduino.cc's sister-brand. This brand is used for boards and products sold outside the US. Please note that the steps described in this chapter are for Arduino's from the Arduino.cc community, which is different from the Arduino.org community.*

## 4.1 Requirements

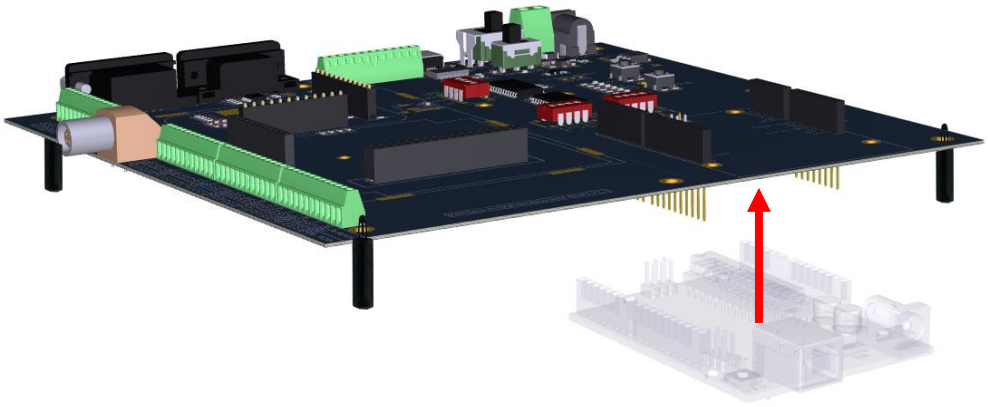
In order to follow the steps below, make sure you have the following:

- PalmSens Embedded SDK (included with EmStat Starter Kit)
- Arduino / Genuino ZERO board (other Arduino boards can also be used, but they require different settings)
- Arduino IDE (download from www.arduino.cc)

## 4.2 Preparing the EmStat dev. board for use with Arduino

Follow the steps below to prepare the EmStat development board for use with the Arduino ZERO.

1. Set all DIP switches of UART SELECT (S6) to OFF.
2. Set DIP switches of ARDUINO UART (S3) to UART1, so set only 1 and 2 are ON.
3. Set all DIP switches of ARDUINO IO'S (S5) to OFF
4. Connect the Arduino underneath the EmStat development board:

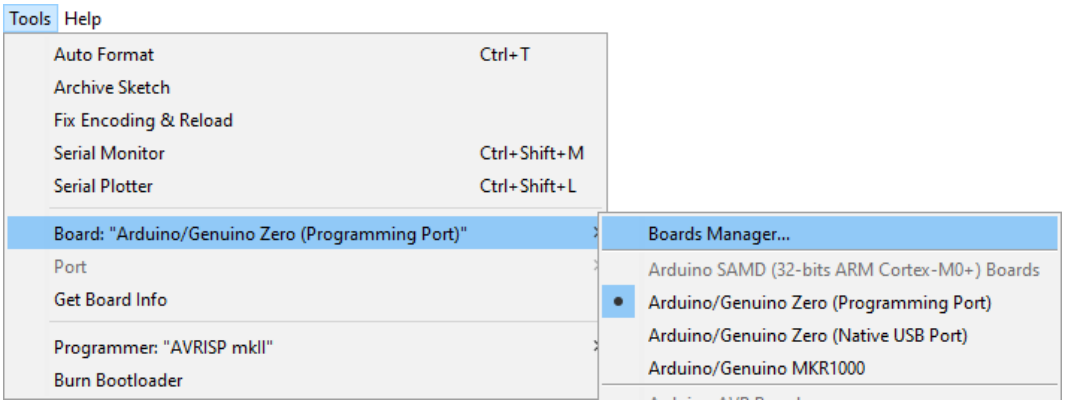


Use the stand-offs to suspend the development board to make space for the Arduino.

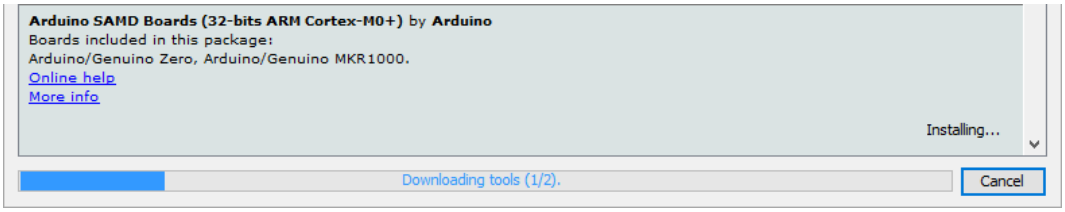
5. Make sure the development board is powered by means of any of the power supply options (see section "Power supply options" on page 11).
6. Connect the Arduino to the PC by means of two micro-USB cables. One will be used for programming and the other one for interacting with the example via a terminal.

## 4.3 Running the EmStat SDK Example for Arduino

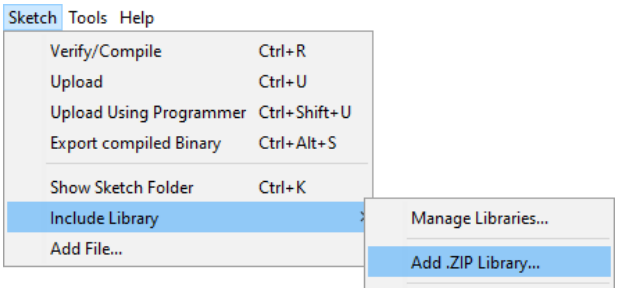
7. In case the Arduino Zero was included with your EmStat Development Board you first have to open the Boards Manager from the menu: Tools → Board: […] → Boards Manager…
**For any other Arduino, make sure to select the right board and continue with the next step.**



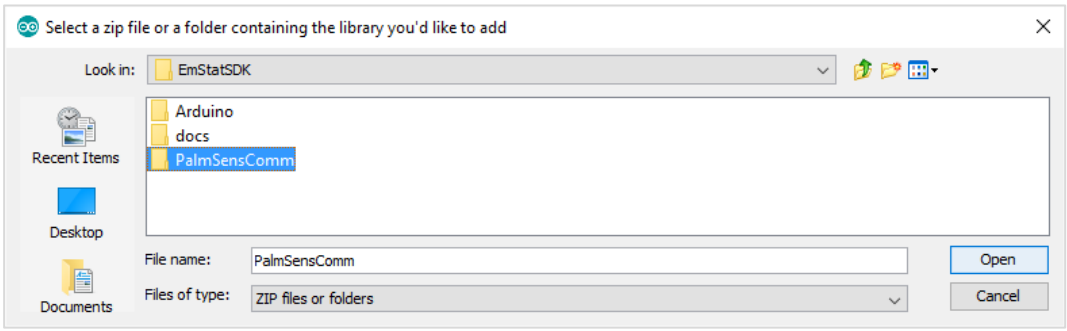

Then select and install the Arduino SAMD Boards (32-bits ARM Cortex-M0+):




8. Install the PalmSens library, using the menu: Sketch → Include Library → Add .ZIP Library:

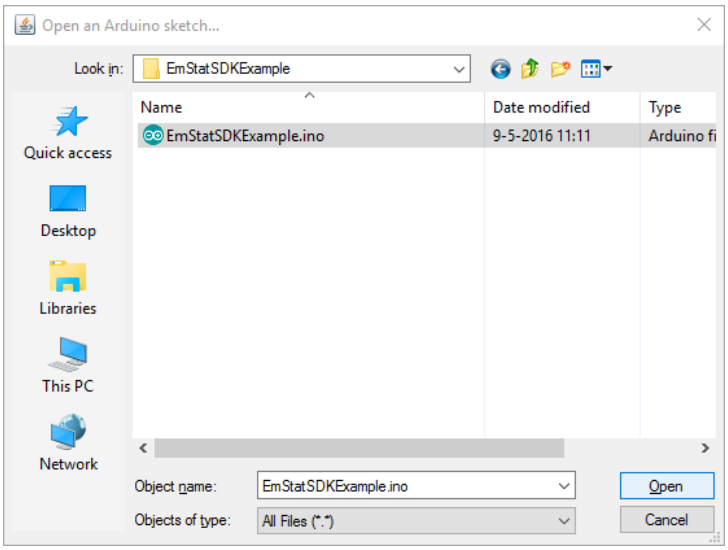



Select the PalmSensComm Library from the EmStatSDK folder.




If everything went well the Library should be available in the menu:
Sketch → Include Library under 'Contributed libraries'

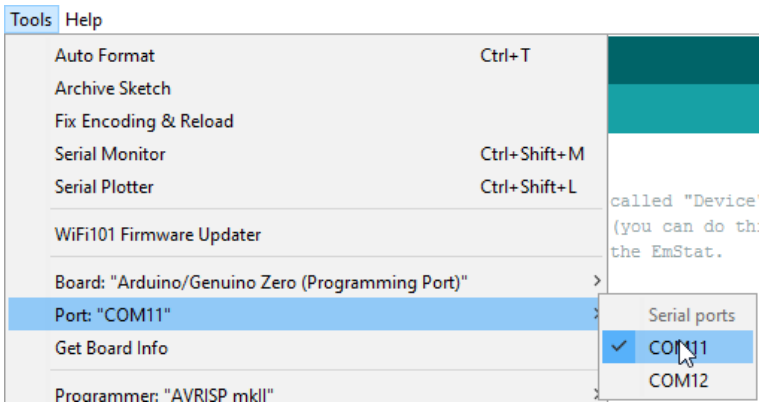
9. Use the menu File → Open… to load the example “EmStatSDKExample.ino” project found in the folder “..\Arduino\EmStatSDKExample”.




10. Upload the example to the Arduino.

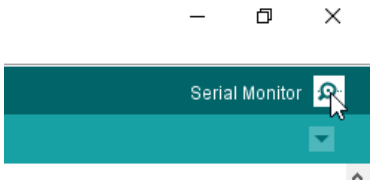
### 4.3.1 Controlling the Arduino via terminal

In the menu Tools → Port … ,you can select which port to use for the terminal:

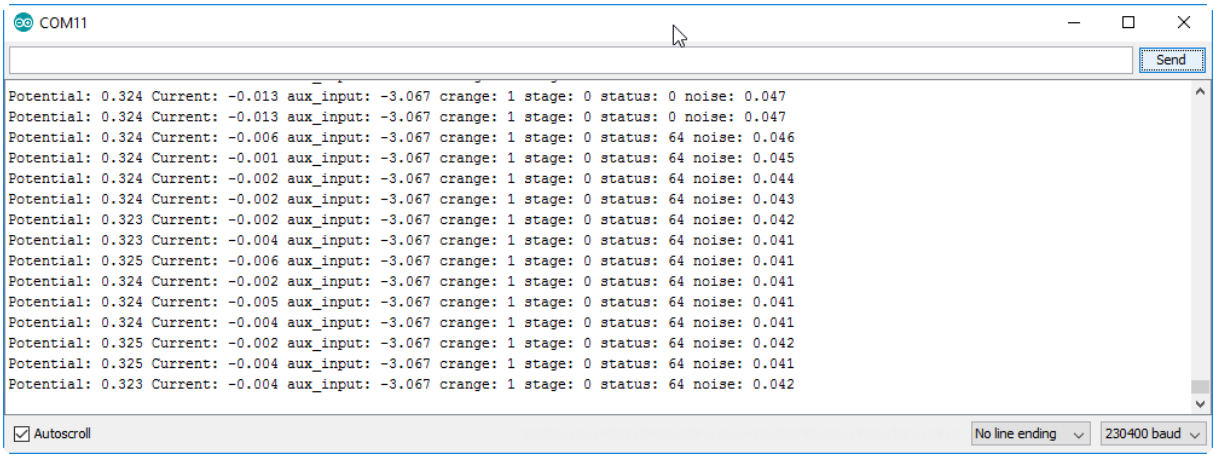



You can use the terminal window to send commands to the Arduino example and for example start measurements.

A button for the Serial Monitor can be found in the upper right corner of the Arduino IDE:



Open the terminal window and set the baud rate to 230400. If you have selected the right COM port you will see idle status messages that are sent every second by the EmStat and parsed by the Arduino:

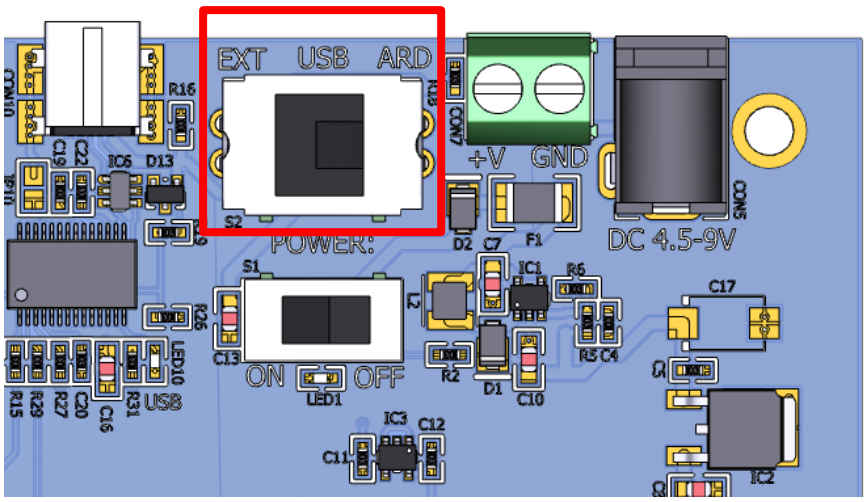



The following commands can be used with the Arduino example:

- s: Toggles the printing of all sent messages to the EmStat.
- r: Toggles the printing of all received messages from the EmStat.
- v: Prints the version of the connected EmStat.
- p: Manually set the output potential to 0.5 V.
- c: Set cell ON
- m: Set multiplexer channel / digital IO.
- 1 to 9: Start a pre-set measurement.

See comments in the Arduino source code for more information.
See PSComm.h for more information on how to address the EmStatSDK library.

# 5 Power supply options




There are three different settings for power supply selection (+5V).
The different inputs can be selected using S2

1. EXT (left): screw terminals (CON7) or DC-input barrel socket (CON5)
2. USB (middle): mini-USB (CON10)
3. ARD (right): power via Arduino, requires JP12 to be closed

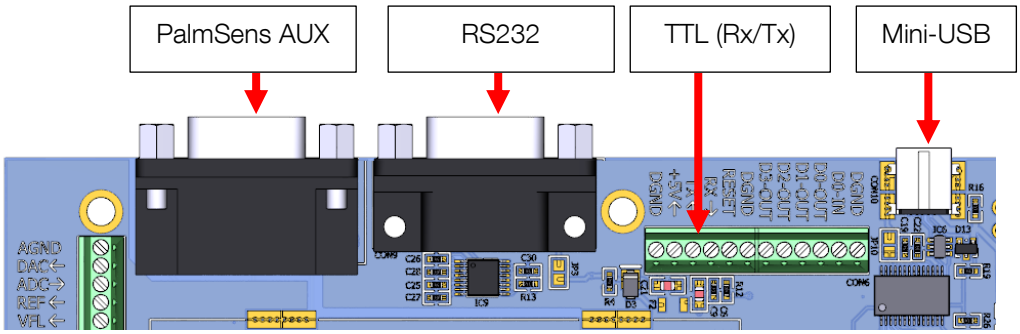
Any power source in the range of 3-12VDC can be used for external power supply. The development board has over-current and reverse-voltage protection, and a regulated +5V supply for the EmStat to ensure correct operation.

# 6 Communication options

The dev board allows for EmStat communication via

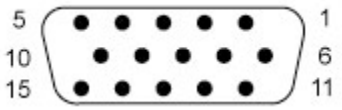
- USB (mini-USB connector)
- RS232 communications via a standard DB9 connector, allows the use of standard RS232 dongles like an off-the-shelve Bluetooth-RS232 dongle.
- An auxiliary port for use with PalmSens extension like a multiplexer or the PalmSens Bluetooth extension:

- On-board level shifting options for all digital IO's, between 5V, 3V3, or any externally provided logic level less than 5V.




## 6.1 PalmSens AUX pinout

Front view of female port:




| | |
|---|---|
| pin 1 | d0 digital output |
| pin 2 | d3 digital output |
| pin 3 | ADC (analog input, range 0 – 4.095 V) |
| pin 4 | Rx |
| pin 5 | Tx |
| pin 6 | d1 digital output |
| pin 7 | d0 digital input |
| pin 8 | NC |
| pin 9 | +5V output (max. 30 mA when powered from USB) |
| pin 10 | DGND (digital ground) |
| pin 11 | d2 digital output |
| pin 12 | NC |
| pin 13 | NC |
| pin 14 | AGND (analog ground) |
| pin 15 | DAC (analog output, range 0 – 4.095 V) |
| shield | DGND (digital ground) |

All digital I/O at 5V.

## 6.2 RS232 communications

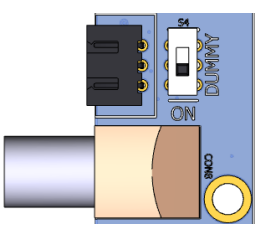
Any standard RS232 cable or dongle can be used with the DB9 port. Only the Rx and Tx lines of this port are used all other pins (RTS, CTS, DTR, DSR) are not connected.
In order use this port, make sure to set enable the corresponding DIP switches of UART SELECT (S6).

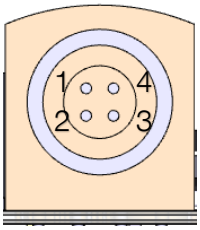
## 6.3 Serial communication at TTL level

Serial TTL communication can be done using CON6 pins 9 and 10 with digital ground pin 7.
In order use this port, make sure only switches 1 and 2 are enabled off the UART SELECT (S6) .

# 7 Pin descriptions

## 7.1 Pinout of LEMO sensor socket (CON8)



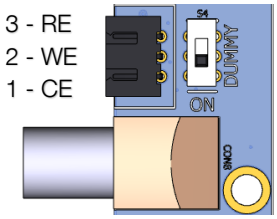




Front view of socket

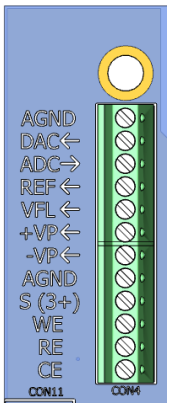
1 ● ● 4
2 ● ● 3

1. RE: blue connector or reference electrode
2. CE: black connector or counter electrode
3. Not connected
4. WE: red connector is working electrode
5. Metal housing: AGND

## 7.2 Pinout of SPE connector (CON11)

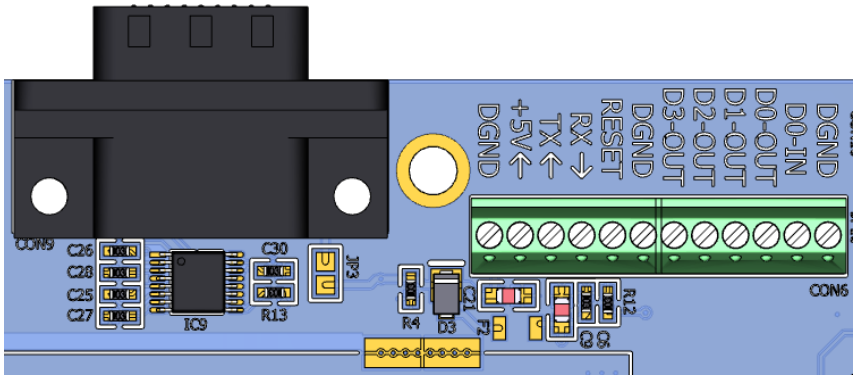



## 7.3 CON4 pinout




| Pin | Function |
|---|---|
| 1. | AGND (analog ground) |
| 2. | DAC (analog output, range 0 – 4.095 V) |
| 3. | ADC (analog input, range 0 – 4.095 V) |
| 4. | Voltage reference (4.096 V) |
| 5. | VFL - Reserved (DO NOT CONNECT) |
| 6. | +VP - Reserved |
| 7. | -VP - Reserved |
| 8. | AGND (analog ground) |
| 9. | Sense (only for use with 100 mA range of EmStat3+) |
| 10. | WE (working electrode) |
| 11. | RE (reference electrode) |
| 12. | CE (counter/auxiliary electrode) |

## 7.4 CON5 pinout

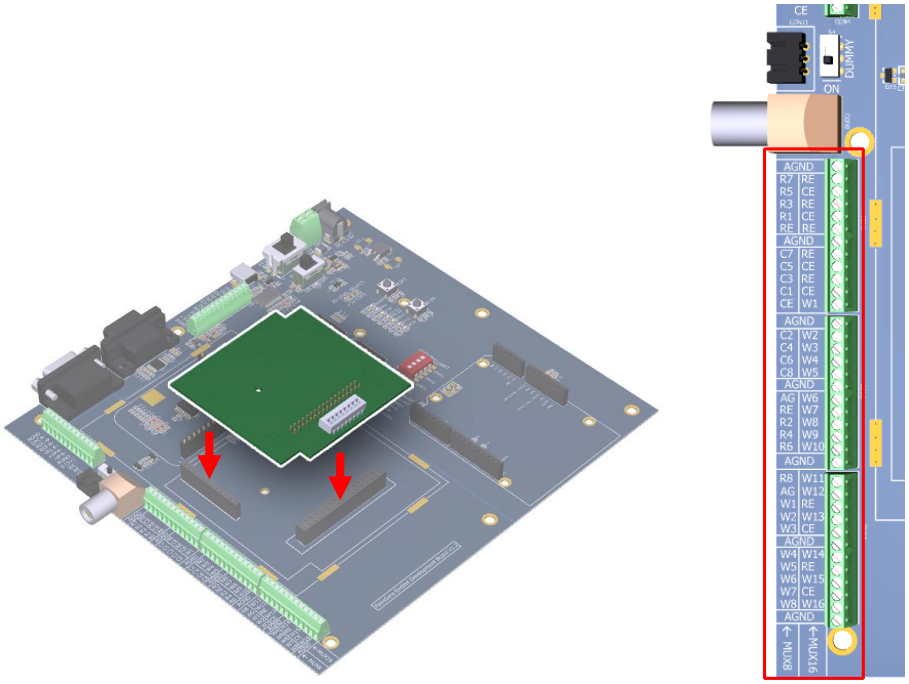



| Pin | Function |
|---|---|
| 1 | DGND |
| 2 | +5V output (max. 30 mA when powered from USB) |
| 3 | Tx (uart) at 5V |
| 4 | Rx (uart) at 5V |
| 5 | EmStat reset (active high) |
| 6 | DGND |
| 7 | Digital output d3 |
| 8 | Digital output d2 |
| 9 | Digital output d1 |
| 10 | Digital output d0 |
| 11 | Digital input line d0 |
| 12 | DGND (digital ground) |

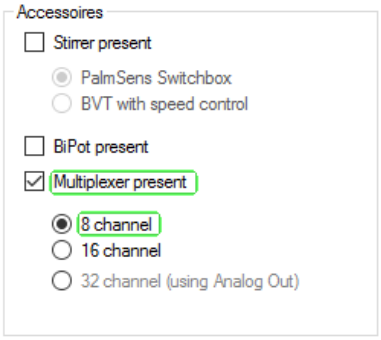
All digital I/O at 5V.

# 8 Adding a multiplexer

A MUX8 or MUX16 multiplexer can be added to the development board as shown below.



The connections for a MUX8 or MUX16 multiplexer are shown at the edge of the board.
The multiplexer can be tested and controlled in PSTrace. Make sure to enable the presence multiplexer in the menu: Tools → General settings…




A multiplexer tab will appear in the method editor and the multiplexer can be controlled in the Manual Control tab.

The multiplexer cannot be used in combination with the 100 mA range of the EmStat3+ !

# 9 Noise considerations

The development board is an open system, picking up different kinds of electrostatic noise from the environment. Sources of this noise can be other devices. Electrochemical sensors and cells are susceptible to noise.

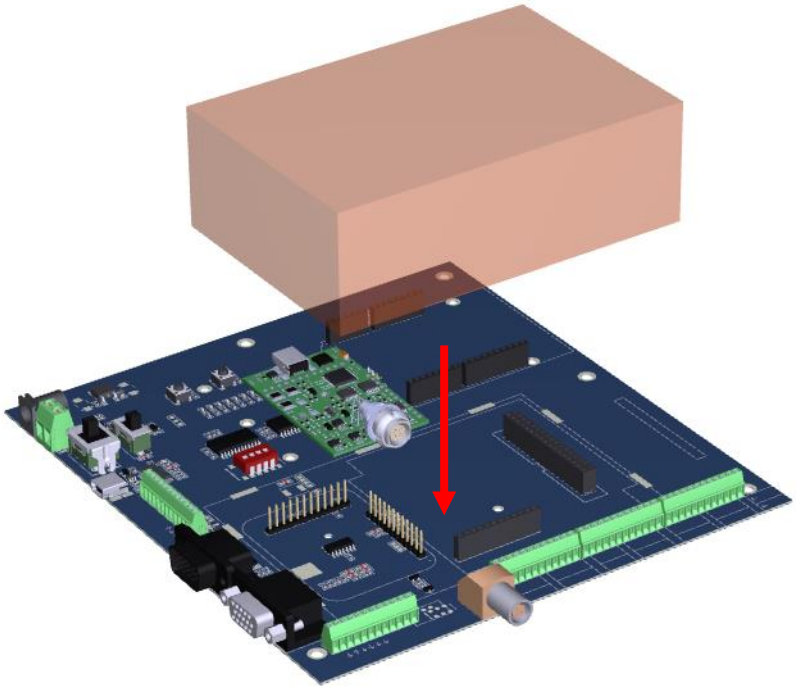
Other known sources of noise are:

- Unshielded or too long sensor- or cell cables. The connection between the dev board and the sensor or cell should not be longer than required. Especially unshielded cables used with many commercially available reference electrodes may result in high noise levels. Shorten the cables when possible.
- Bad cable connections. A bad connection for example due to corroded banana plugs or clips can be the cause of noisy or other kinds of bad readings.
- Power supply or the PC connected to the development board may induce noise. Sensitive measurements can be performed without these adapters connected.
- Electrical equipment. Measurements in the vicinity of electrical equipment might be distorted by electrical interference. This might be eliminated by placing the cell in a faraday cage which is connected to earth or ground.

In case the noise levels remain too high, the use of a faraday cage (a simple metal cage may be sufficient) is required.

## 9.1 Shielding can

An optional copper shielding can for the development board that covers the EmStat and optional the MUX module is available from PalmSens BV. This can should be used for measurements performed in the order of uA and below.



The shielding can is held by metal grounding clips on the development board.

# 10 Changing EmStat baudrate settings

The EmStat module has been set to a default baudrate of 230400 baud.
PSTrace also supports a baudrate of 57600 baud.

The baudrate has the following influence on the maximum scan rate for measurements;

| Baudrate | Max. number of data points / second | Min. interval time |
|---|---|---|
| 230400 | 1000 | 1.00 ms |
| 57600 | 197 | 5.07 ms |

It is possible to change the baudrate setting of the EmStat module in the firmware (requires firmware v7.0 or higher).
See the EmStat Communications Protocol section 4.6: *Changing the baudrate.*

**The baudrate can also be changed in PSTrace 4.8 or later. See menu: Tools → Instrument settings.**

The firmware update program will always be able to update the firmware regardless of the baudrate supported by the firmware.
See section "Updating EmStat firmware" on page 17 in this document on how to update the firmware using a serial connection.

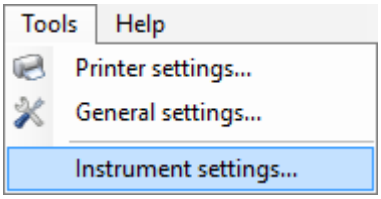
# 11 Updating EmStat firmware

The firmware of EmStat can easily be updated automatically by using PSTrace. Just open the latest version of PSTrace and connect to the development board. PSTrace will now check if the latest version is present on the EmStat module and update if needed.

## 11.1 Programming the EmStat with custom firmware

In case your EmStat module needs special firmware you can upload this firmware file (.HEX) by following these steps:

1. Make sure to connect the development board to the PC via the mini-USB port.
2. Open PSTrace (do not connect to the development board).
3. In PSTrace use the menu Tools → Instrument settings… to open the Instrument settings window.

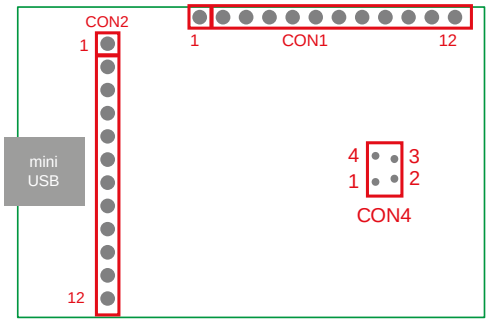



4. Click the button Update Firmware.
5. Connect to the development board.
6. Select the firmware file you want to upload.
7. Press button 'Start update'.

## 11.2 Updating the EmStat's firmware via serial connection

When updating the EmStat via USB, the USB chip takes care of toggling the download and reset pins when necessary. This needs to be done manually when updating the EmStat via serial connection.

In order to follow the steps below first solder on two small wires to both pin 6 and pin 8 of CON2 of the EmStat module directly so you can connect the two pins briefly.




Follow these steps to upload new firmware via the serial port (bypassing the USB):

- Remove the power supply from EmStat.
- Connect pin 8 (download - active low) to pin 6 (GND) of CON2 on the EmStat.
- Connect the power supply.
- Run program 'Update firmware'.
- Select the corresponding COM port and press 'Connect'.
- Load firmware file ESx##, where x is the EmStat type and ## the firmware version.
- Make sure the checkbox 'PalmSens or EmStat is on, but does not work property' is checked
- Click button 'Update Firmware'.
- Remove the power supply briefly to force a reset.
- Wait until updating has finished.
- Remove the connection between pin 8 and 6.
- Remove the power supply briefly to force a reset.

# 12 EmStat3 and EmStat3+ module specifications

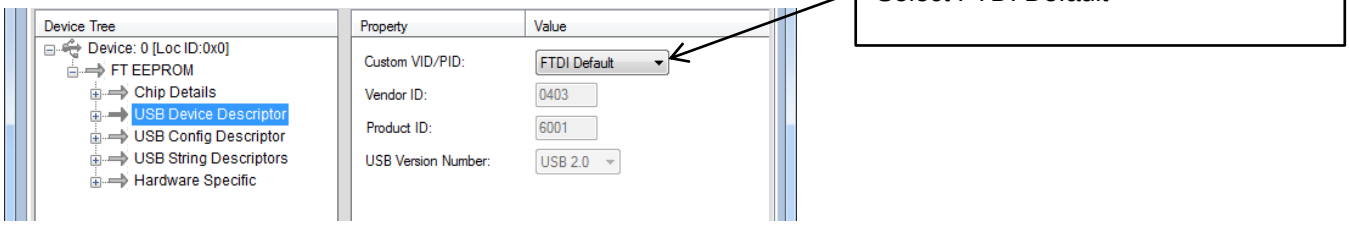
| | EmStat3 | EmStat3+ |
|---|---|---|
| ▪ dc-potential range | ± 3.000 V | ± 4.000 V |
| ▪ compliance voltage | ± 5 V | ± 8 V |
| ▪ applied dc-potential resolution | 0.1 mV | 0.125 mV |
| ▪ applied potential accuracy | ≤ 0.2 %<br>with max. 2 mV offset error | ≤ 0.3 %<br>with max. 3 mV offset error |
| ▪ current ranges | 1 nA to 10 mA (8 ranges) | 1 nA to 100 mA (9 ranges) |
| ▪ maximum measured current | ± 20 mA typical and<br>± 15 mA minimum | ± 100 mA typical |

EmStat 3 and 3+ :

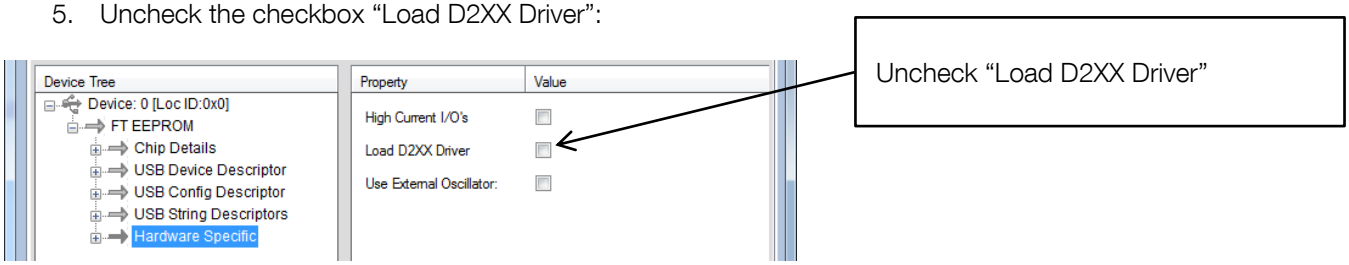
| | |
|---|---|
| ▪ current resolution | 0.1 % of current range<br>1 pA on lowest current range |
| ▪ current accuracy | ≤ 1 % of current range at 1 nA<br>≤ 0.5 % at 10 nA<br>≤ 0.2 % at 100 nA to 100 uA<br>≤ 0.5 % at 1 mA, 10 mA and 100 mA<br>all with max. 0.2 % offset error |
| ▪ electrometer amplifier input | > 100 Gohm // 4 pF |
| ▪ rise time | approx. 100 µs |

# Appendix A: Change EmStat USB connection to virtual COM port

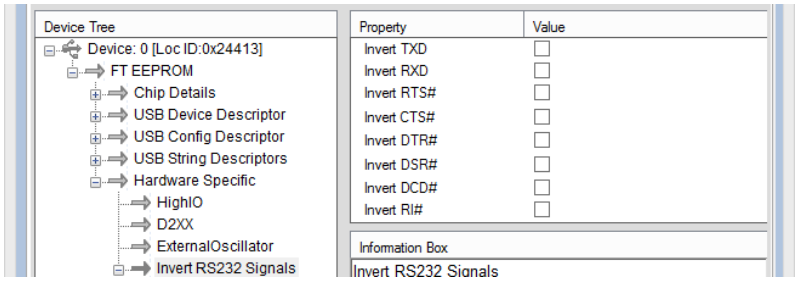
1. Connect EmStat to the USB port.
2. Open FT_PROG (see http://www.ftdichip.com/Support/Utilities.htm)
3. Click the “Scan and Parse” button 
4. Change the Custom VID/PID to FTDI Default:




5. Uncheck the checkbox “Load D2XX Driver”:




6. Make sure all “Invert RS232 Signals” settings are unchecked:




7. Click the “Program Devices” button 
8. Click the “Scan and Parse” button  again to check if the settings are written correctly, especially the invert DTR# setting. If this setting cannot be changed, please contact us. Your FT_PROG version might have a bug.
9. Remove the USB cable and insert again.
10. Your PC will now detect a virtual COM port which can be used to communicate with EmStat.
11. The com port number can be found in the Device Manager found in Windows Configuration. You can also press [windows key] + [pause] on your keyboard to find the Device Manager.

Make sure the USB chip is <u>not programmed</u> as VCP (Virtual COM port) when using the Rx/Tx lines on CON2 of the EmStat module.

# NOTES