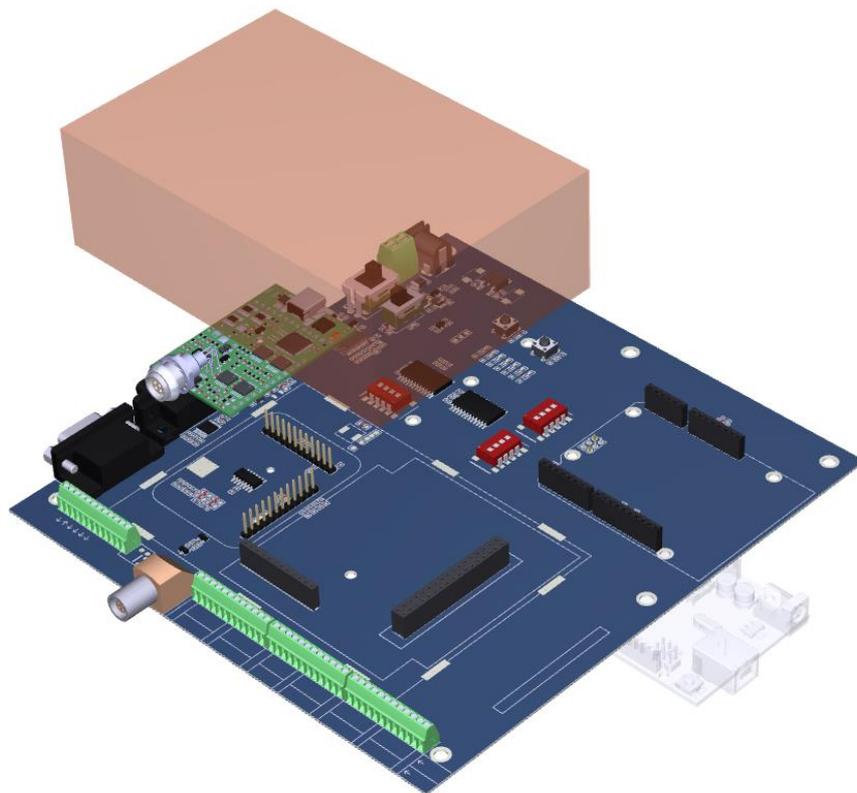


# EmStat<sup>TM</sup>

## Development Kit description

---

Development Kit for EmStat module: OEM Interface for electrochemical sensors



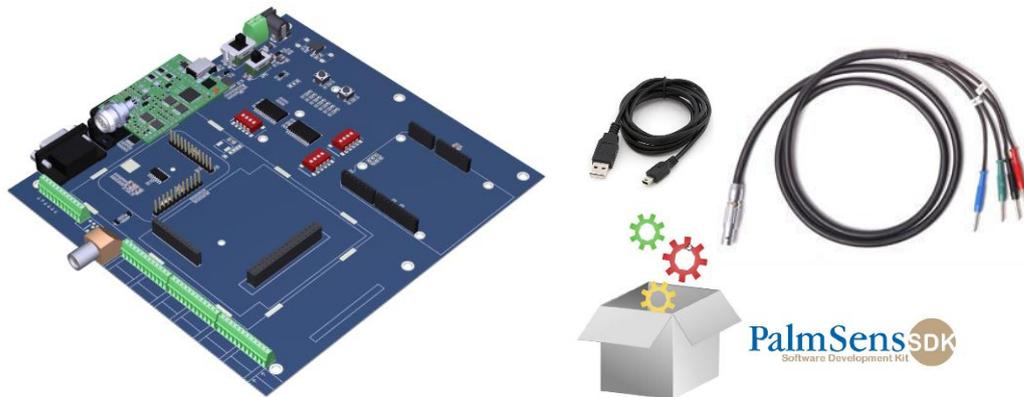
Last revision: October 31, 2016

© 2016 PalmSens BV  
[www.palmsens.com](http://www.palmsens.com)

# EmStat Development Kit contents

The standard kit comes with the following items:

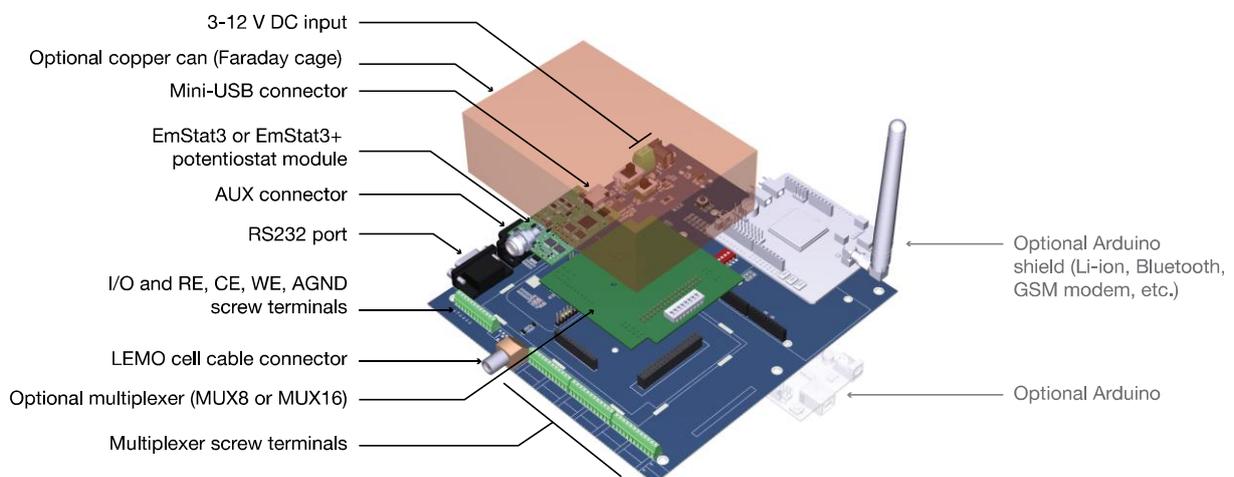
- EmStat Development Board
- EmStat3 or EmStat3+ module
- Mini-USB cable
- Standard sensor cable with 2 mm banana lead ends and croc-clips
- PStTrace software for Windows
- Software Development Kit for .NET
- Embedded Software Development Kit (ANSI C and Arduino)
- Document describing the EmStat communications protocol



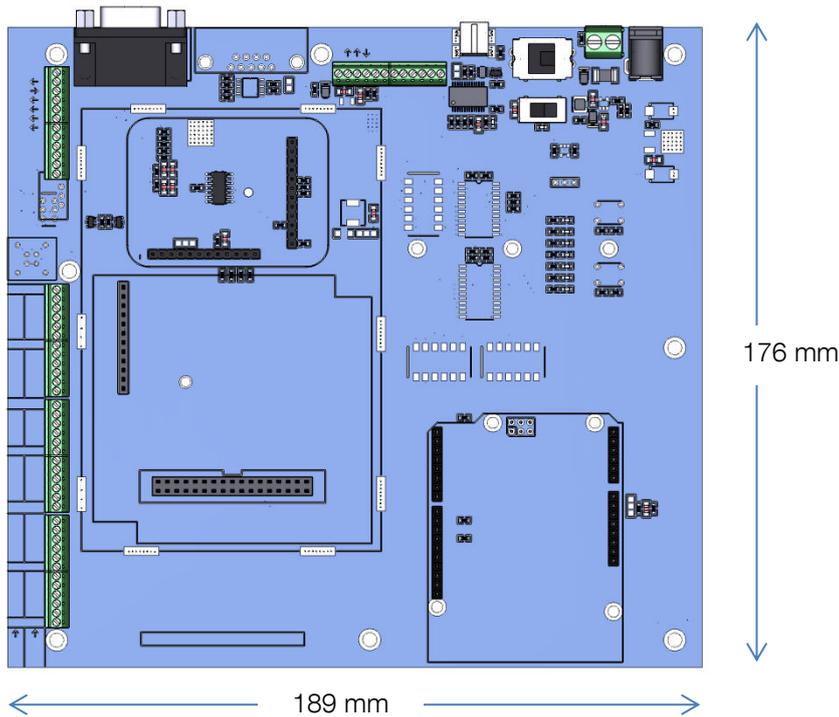
Optional:

- MUX8 or MUX16 multiplexer
- DC adapter (in case not powered by USB or own power supply)

## Development board features



## Development board dimensions



Max. height without shielding can, without standoffs:	25 mm
Max. height without shielding can on standoffs:	45 mm
Max. height with shielding can:	55 mm

### Communication options

- RS232 communications via a standard DB9 connector allows the use of standard RS232-Bluetooth dongles
- USB communications via a mini-USB connector
- DB15 AUX port (compatible with PalmSens Bluetooth extension)
- Onboard level shifting options for all digital IO's, between 5V, 3V3, or any externally provided logic level less than 5V.

### Circuit protection

- 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 operate and prevent damage

### Power options

Various power options:

- DC adaptor
- USB
- Screw terminals to connect from any power source like a battery
- Li-ion Arduino shield or via Arduino board.

### Arduino/Genuino Integration

- Designed specifically for compatibility with an Uno or Zero

Various jumper setting available:

- UART lines

- EmStat-DAC to Arduino-ADC
- EmStat-4Vref to Arduino-AREF
- EmStat-D0 & D1 to Arduino-INT0 & INT1 interrupts
- 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

## Potentiostat interfaces

- LEMO connector for use with a PalmSens Sensor Cable
- An Screen Printed Electrode (SPE) connector (through-hole version)
- On-board standard test sensor option enabled by a switch
- Via pin headers/screw terminals

## Other

- Option to populate a MUX8 or MUX16 multiplexer, with connections pulled to pin headers/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

# Software / embedded development

The EmStat Development Kit comes with extensive code libraries that allows you to develop EmStat-powered applications for the most popular (embedded) platforms:

- .NET 4.5.1 (Windows 7, 8 and 10 applications)
- Universal Windows Apps (Windows 10 / IoT)
- Android (using Xamarin)
- Raspberry Pi
- Arduino (open-source electronic prototyping platform)
- ANSI C (for virtually any microcontroller system)

All code libraries come with example code and support.

## .NET 4.5.1 (Windows applications)



### Available libraries:

PalmSens.Core.dll	Platform independent dll build against <a href="#">.NET Platform Standard 1.2</a> with all core functionalities for running measurements and doing analysis with any PalmSens or EmStat device.
PalmSens.Core.Windows.dll	Expands PalmSens.Core.dll for use with Windows. Built against .NET 4.5.1.
PalmSens.Controls.dll	Allows building complete interfaces as used in PSTrace in your applications <ul style="list-style-type: none"> <li>- selecting connected devices;</li> <li>- displaying idle instrument readings;</li> <li>- using PSTrace style components, like the PSRadioButton, PSCheckBox or PSLabel.</li> </ul>
PalmSens.Plot.dll	For visual representation of data recorded with PalmSens or EmStat, similar to the plot object in PSTrace.
PalmSens.MethodEditor.dll	Control for editing methods to setup your measurement.

## Universal Windows Apps



Windows IoT



Raspberry Pi

### Available libraries:

PalmSens.Core.dll	Platform independent dll build against <a href="#">.NET Platform Standard 1.2</a> with all core functionalities for running measurements and doing analysis with any PalmSens or EmStat device.
PalmSens.Core.UniversalWindows.dll	Expands PalmSens.Core.dll for use with Windows Universal Apps.

Windows 10 Universal Apps run on Windows 10 for desktop, for mobile and for Windows 10 IoT. This means these apps also run on a Raspberry Pi (or any other embedded development platform) running Windows 10 IoT.

## Android (using Xamarin)



Xamarin

### Available libraries:

PalmSens	Contains all platform-independent code for device communication, measurement settings and measurements. SDK users will need the classes in this assembly for creating curves, techniques, methods, saving/loading files etc, all of these are demonstrated in the SDK examples.
PalmSens.Android1.Core.Utils	A collection of helper classes that can be used in any Xamarin Android project. Contains classes that allow USB and Bluetooth functionality to be added to an app with a single line of code, this is demonstrated in the SDK examples.
PalmSens.Android1.Core.Communication.BlueTooth PalmSens.Android1.Core.Communication PalmSens.Android1.Core.Communication.USB	A collection of classes that take care of all USB and Bluetooth functionality and provide an API for SDK users to communicate with connected devices. The "Communications" class provides an API for SDK users containing all commands that can be sent to a device and events for all information that can be received. This should be the only class SDK users need to manage device communication.
PalmSens.Core.Plot.Mono PalmSens.Android1.Core.Plot	Contains all the classes needed to implement a plot object for Android.
	For visual representation of data recorded with PalmSens or EmStat, similar to the plot object in PSTrace.

<p>Xamarin.Android.Support.v4 FilePicker</p>	<p>The FilePicker assembly contains a file-browser activity that can be started as an Intent in a Xamarin.Android app. Xamarin.Android.Support.v4 is a dependency of the FilePicker project. Use of the FilePicker activity as an Intent is demonstrated in the SDK examples when loading and saving files.</p>
--	---

## Arduino (open-source electronic prototyping platform)

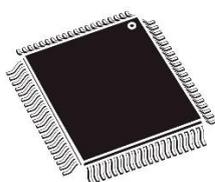


### Available libraries:

<p>PSComm.h</p>	<p>Contains routines for running measurements and receive readings during idle mode for EmStat devices</p>
<p>PSMethod.h</p>	<p>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 send to EmStat devices to initiate a measurement.</p>
<p>PSCommon.h</p>	<p>Contains some supporting functions, like converting from and to hexadecimal numbers.</p>

Includes an example for Arduino/Genuino for integration with the EmStat Development Board.

## ANSI C (for virtually any microcontroller system)



### Available library header files:

<p>PSComm.h</p>	<p>Contains routines for running measurements and receive readings during idle mode for EmStat devices</p>
<p>PSMethod.h</p>	<p>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 send to EmStat devices to initiate a measurement.</p>
<p>PSCommon.h</p>	<p>Contains some supporting functions, like converting from and to hexadecimal numbers.</p>