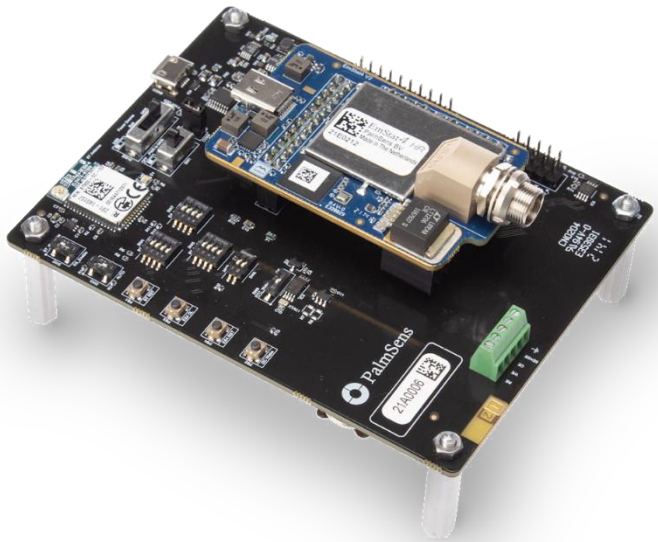
EmStat4M™

Development Kit Manual

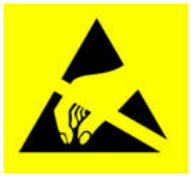


PalmSens

Rev. 4-2024-003

1 Contents

2 Important before you start unpacking



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

3 Customer support

Make sure your software and firmware are up-to-date. You can update PSTrace for Windows via the menu *Help → Check for updates.* The firmware of your instrument is updated automatically in PSTrace upon connecting to the EmStat4M Development Board, if needed. In case of questions, please do not hesitate to contact us at support@palmsens.com. Try to describe the problem as detailed as possible, photos and screenshots are often very helpful.

4 Terms and Definitions

BT	Bluetooth
EEPROM	Electrically Erasable Programmable Read-Only Memory
ES4M	EmStat4M
GPIO	General Purpose Input Output (pin)
IC	Integrated Circuit
PCB	Printed Circuit Board
RTC	Real-Time Clock
SDK	Software Development Kit
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

5 Introduction

The EmStat4M Development Board is intended as development, demonstration and testing platform for electrochemical applications using the EmStat4M module.
The EmStat4M is a module (small PCB) of 62 x 40 mm functioning as Potentiostat, Galvanostat, and optional Frequency Response Analyser (FRA) for Electrochemical Impedance Spectroscopy (EIS).
The development board can be powered and communicate directly via its USB-C port, or it can be connected using the pin headers for communication, power, and cell connections.
A high-end LEMO connector for use with our standard shielded sensor cables is available on the EmStat4M module. It is always advised to use the LEMO connector for best performance.

6 EmStat4M Development Kit contents

The box you received contains the following items:

- EmStat4M LR or HR module
- Development Board
- Micro-USB cable
- USB-C cable
- USB splitter cable for extra power (EmStat4M HR only)
- Sensor cable (1 meter with 2 mm pins)
- 4 croc clips (EmStat4M LR) or 5 croc clips (EmStat4M HR)
- 2x SPE connector
- Mini screwdriver
- Dummy Cell

- USB drive with PSTrace software for Windows and documentation
- Quick Start document
- Card with an overview of the development board
- Card with an overview of the EmStat4M module

7 Recommended software tools

The following software tools are recommended:

- **PSTrace** - software for Windows for use with EmStat4M
- **Tera Term** - open-source terminal for direct serial communication
- **Notepad++** - free and simple source code editor to replace the default Windows Notepad
- **Visual Studio Code** - free and open-source Integrated Development Environment (IDE) for use with our SDK's for .NET

8 Development board

8.1 Schematic

The following schematic represent the main connections found on the Development Board.

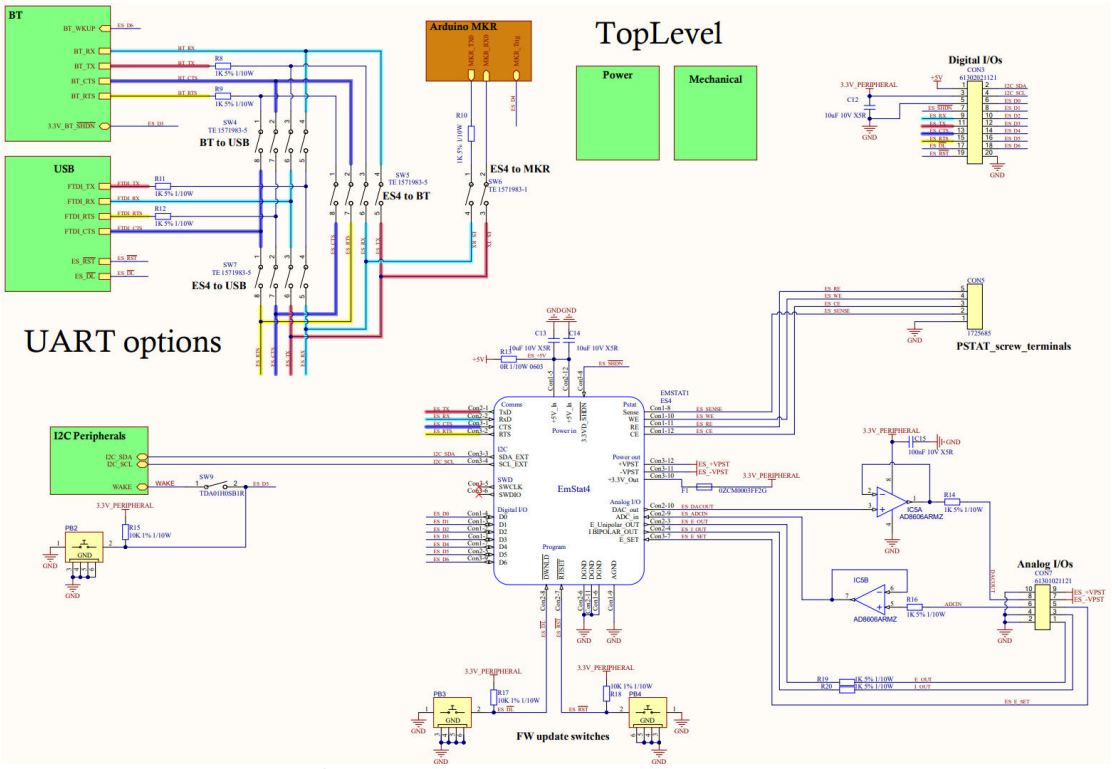
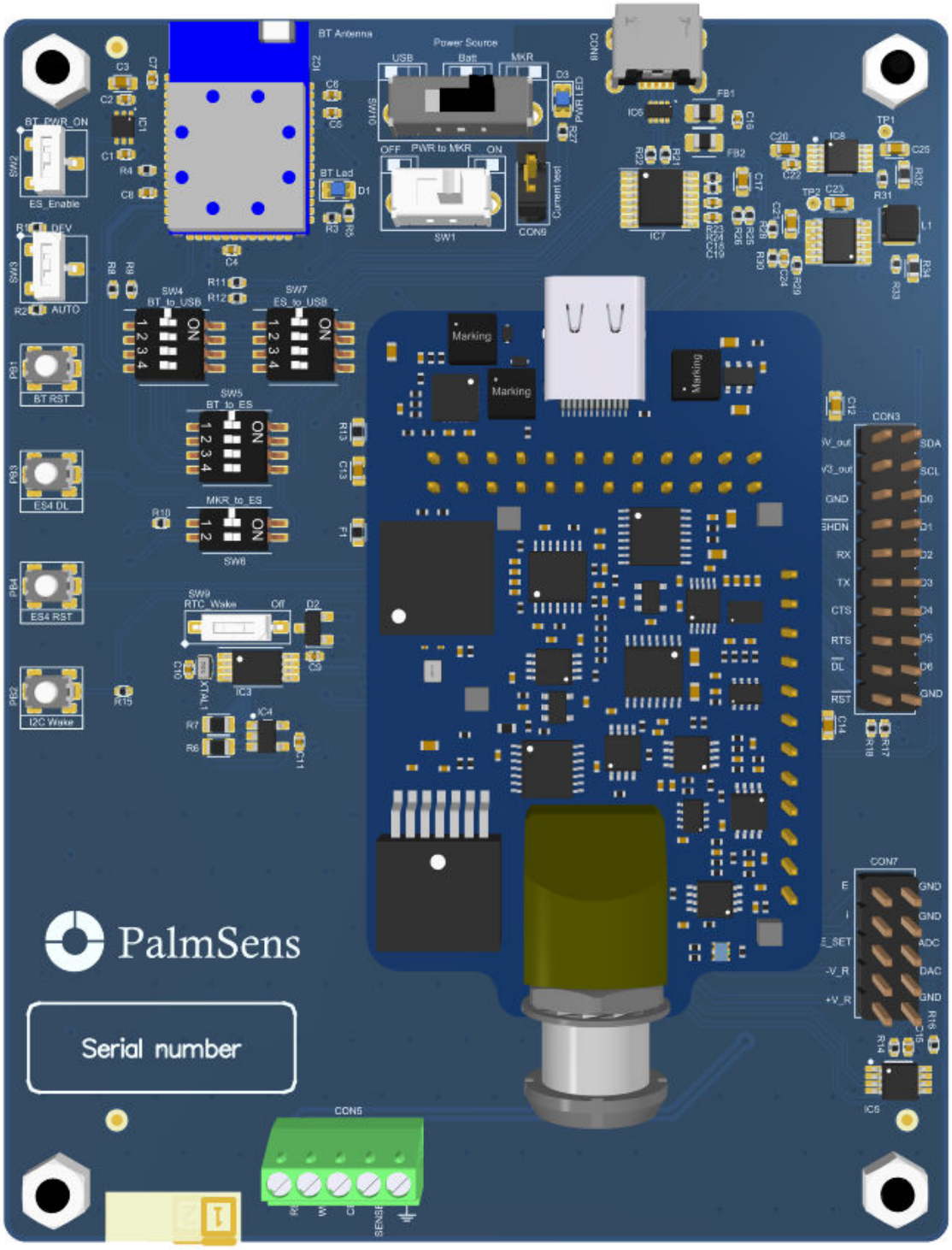


Figure 1 - Top level schematics of the Development Board

For more detailed schematics in PDF format, see the section "Downloads" on:

www.palmsens.com/es4m-dev

8.2 PCB Layout



8.3 Pin descriptions

Table 1 - CON1 Arduino MKR Socket (left row)

Pin	Arduino MKR Zero	Dev. Board	Dev. Board Function
1	AREF / AIN / PA03	*NC*	-
2	DAC0 / A0 / D15	*NC*	-
3	A1 / D16	*NC*	-
4	A2 / D17	*NC*	-
5	A3 / ~D18	*NC*	-
6	A4 / ~D19	*NC*	-
7	A5 / D20	*NC*	-
8	A6 / D21	*NC*	-
9	~D0	*NC*	-
10	~D1	*NC*	-
11	~D2	MKR_Trig	Connected to EmStat4M D4
12	~D3	*NC*	-
13	~D4	*NC*	-
14	~D5	*NC*	-

Table 2 - CON2 Arduino MKR Socket (right row)

Pin	Arduino MKR Zero	Dev. Board	Dev. Board Function
1	+5V	+5V_MKR_OUT	-
2	VIN	+5V_MKR_IN	-
3	+3V3	*NC*	-
4	GND	GND	Ground
5	RESET	*NC*	-
6	D14 / Tx	MKR_TX0	Connected to EmStat4M (ES4M) RX via SW6
7	D13 / Rx	MKR_RX0	Connected to EmStat4M (ES4M) TX via SW6
8	~D12 / SCL	*NC*	-
9	D11 / SDA	*NC*	-
10	D10 / CIPO	*NC*	-
11	D9 / SCK	*NC*	-
12	~D8 / COPI	*NC*	-
13	~D7	*NC*	-
14	~D6	*NC*	-

Table 3 - CON3 Digital I/Os

Pin	To EmStat4M	Function	Pin	To EmStat4M	Function
1	-	+5V out	11	CON2-1	TX
2	CON3-3	I2C SDA	12	CON1-1	D3
3	CON3-10	+3.3V out	13	CON3-1	CTS
4	CON3-4	I2C SCL	14	CON1-7	D4
5	-	GND	15	CON3-2	RTS
6	CON1-4	D0	16	CON2-5	D5
7	CON3-8	~SHDN (Shutdown)	17	CON2-8	~DL (Download)
8	CON1-3	D1	18	CON3-9	D6
9	CON2-2	RX	19	CON2-7	~RST (Reset)
10	CON1-2	D2	20	-	GND

For more detailed information on the EmStat4M pinout, please refer to the EmStat4M datasheet.

Table 4 - CON5 Potentiostat Screw Terminals

Pin	To EmStat4M	Function
1	GND	Ground
2	CON1-8	Sense (Working Sense Electrode)
3	CON1-12	CE (Counter Electrode)
4	CON1-10	WE (Working Electrode)
5	CON1-11	RE (Reference Electrode)

Table 5 - CON7 Analog I/Os

Pin	To EmStat4M	Function	Comments
1	CON2-3	E out	To potentiostat Voltage Follower
2	-	GND	
3	CON2-4	I Out	To potentiostat Current Follower
4	-	GND	
5	CON3-7	E Set	To input an external potential to control the ES4 potentiostat voltage set point. Can be used for IR compensation. Input impedance = 4.99k
6	CON2-9 (via IC5B)	ADC in	Unipolar ADC input: 0-3V, 16 bits
7	CON3-11	ES +VPST	Positive rail voltage output: ~+7.2V (LR) / ~+9.75V (HR) , max. 20mA
8	CON2-10 (via IC5A)	DAC out	Unipolar DAC output: 0-3V, 12 bits
9	CON3-12	ES -VPST	Positive rail voltage output: ~-7.0V (LR) / ~-9.75V (HR) , max. 20mA
10	-	GND	

8.4 Connecting to the PC via USB-C or micro-USB

You can either connect the PC directly using the USB-C port on the EmStat4M module or you can use the micro-USB connection of the development board.

Do not use the micro-USB and USB-C cables simultaneously.

The micro-USB port on the development board is meant for programming the BT module, charging, or testing the UART lines to the EmStat4M.

When using the USB-C port on the EmStat4M module, the communication on the PC side is handled by the Windows CDC drivers. The micro-USB port works using an FTDI chip for USB to serial communication. With both USB ports a connection can be made via a serial virtual COM port to use a terminal program like Tera Term for example.
The following table shows the implications for connecting either to the micro-USB of the development board or the USB-C cable of the EmStat4M module.

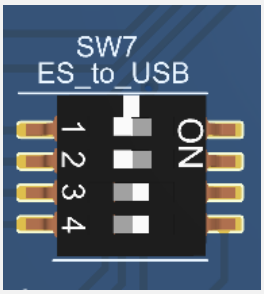
Table 6 - Differences between USB ports

	 Micro-USB cable to development board	 USB-C cable directly to EmStat4M
▪ **Speed**	230.000 baud (default) allowing for +/- 700 datapoints/second	Full-speed USB allowing for +/- 2500 datapoints/second
▪ **Connecting to PSTrace**	Shows as “ES4Dev” Enable SW7 3+4 (Rx and Tx) leave SW7 1+2 OFF[1]	Shows as “EmStat4 LR” or “EmStat4 HR” SW7 setting is ignored
▪ **Drivers**	Works standard in Windows 7 to 11 without installation of special drivers	Works out-of-the box in Windows 10, requires EmStat drivers to be installed (via PSTrace) in Windows 7
▪ **Other functions**	Allows for direct communication between PC and the Laird BT900 Bluetooth module SW7: ALL OFF SW4: ALL ON	Allows for automatic firmware updating of the EmStat4M module in PSTrace

[1] Flow control will be added with a software update. Until then, leave SW7 1+2 OFF.

To communicate to the EmStat4M via the USB port make sure all switches (1-4) of SW7 (ES_to_USB) are set to ON and for SW4 (BT_to_USB) and SW5 (BT_to_ES) all switches are set to OFF.

PSTrace 5.9 does not support flow control:
when communicating with PSTrace 5.9, make sure only to only set SW7 pin 3+4 to ON and pin 1+2 to OFF.



8.5 Using the Bluetooth module

The Laird BT900 Bluetooth module comes programmed for use in either Bluetooth 5.0 (BLE) or in Serial Port Profile (SPP, Classic Bluetooth) mode.
See for more detailed information the Documents section of the Laird BT900 module.

8.5.1 Hardware settings for connecting via Bluetooth with the EmStat4M

To allow for Bluetooth communication with the EmStat4M module, make sure to set all switches on SW4, SW6 and SW7 **OFF** and all dipswitches on SW5 **ON**.
Set SW2 to **BT_PWR_ON** (up) and SW3 to **AUTO** (down):

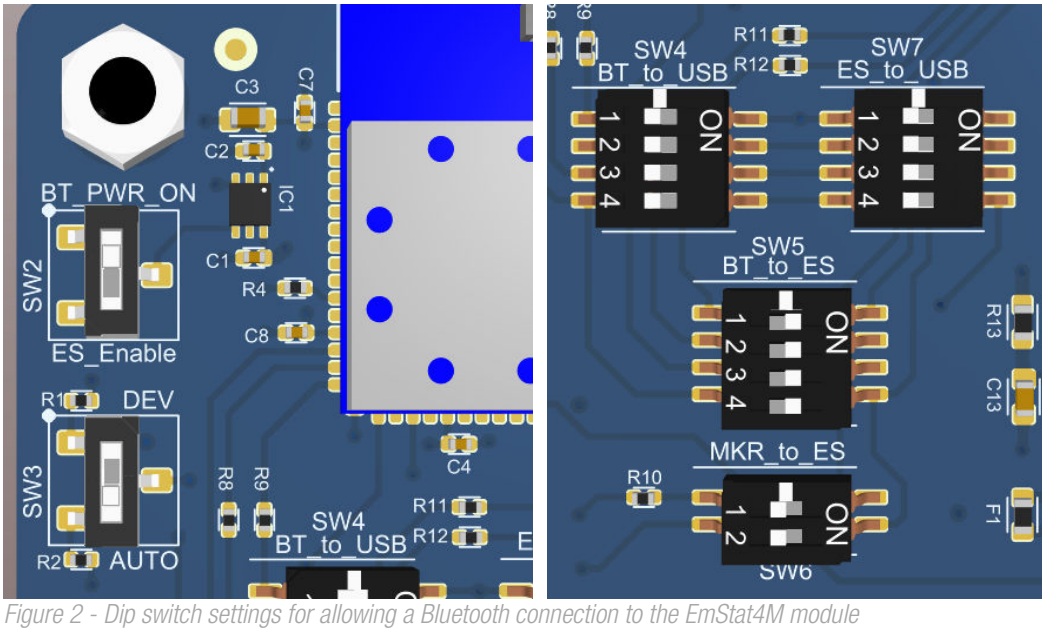


Figure 2 - Dip switch settings for allowing a Bluetooth connection to the EmStat4M module

When connecting via Bluetooth using SPP, a serial COM port for Bluetooth communications will become available on the host side (e.g. Windows) when connecting to the Bluetooth device. The Bluetooth name has the format “PS-xxxx” where xxxx is the last digits of the Bluetooth MAC address. (This MAC address can be found on the sticker of the Bluetooth module between the QR code and “CE” logo):

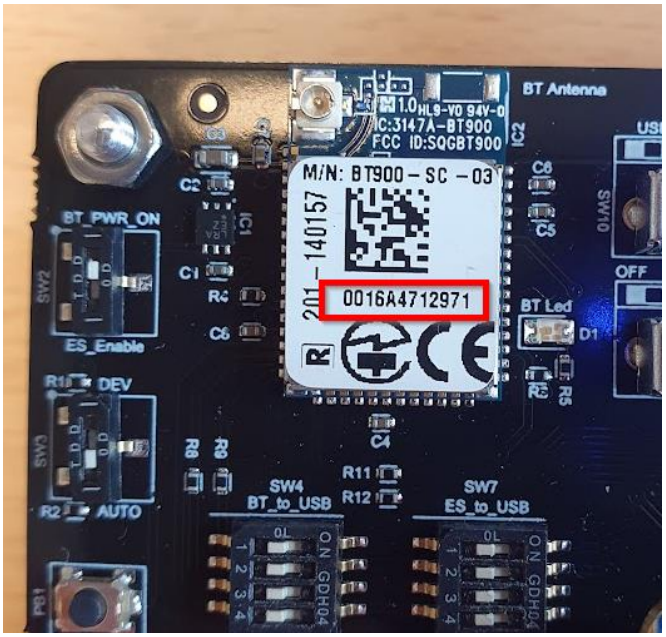


Figure 3 - The MAC address of the Bluetooth module

The Bluetooth SPP mode is not supported by iOS.
For connecting with iOS, see section 8.5.3.

8.5.2 Connecting in PSTrace using SPP (Classic Bluetooth)

In PSTrace the Bluetooth device appears after clicking the Bluetooth icon in the upper left corner:

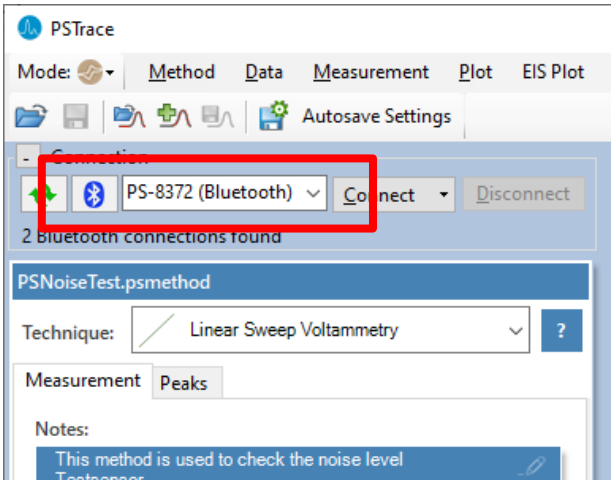


Figure 4 - The Bluetooth device appears here after clicking the blue Bluetooth icon

The first time PSTrace connects with a new Bluetooth device Windows show a pop-up message in the right bottom corner of the screen.

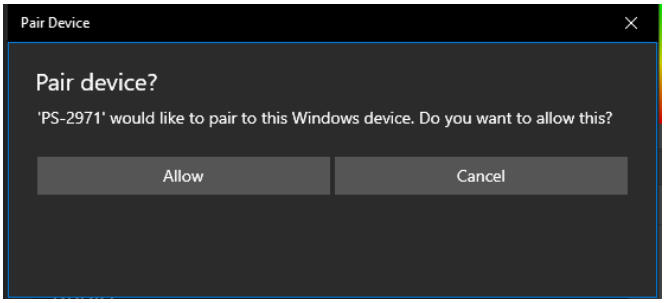


Figure 5 - Windows Pop-up asking to pair with the Bluetooth device

Click on this message and Allow Windows to pair the device.

Figure 6 - The Bluetooth Classic device shown in the Windows 10 Bluetooth devices list

Now you can click the Connect button in PSTrace to connect.

8.5.3 Connecting using vSP (for iOS)

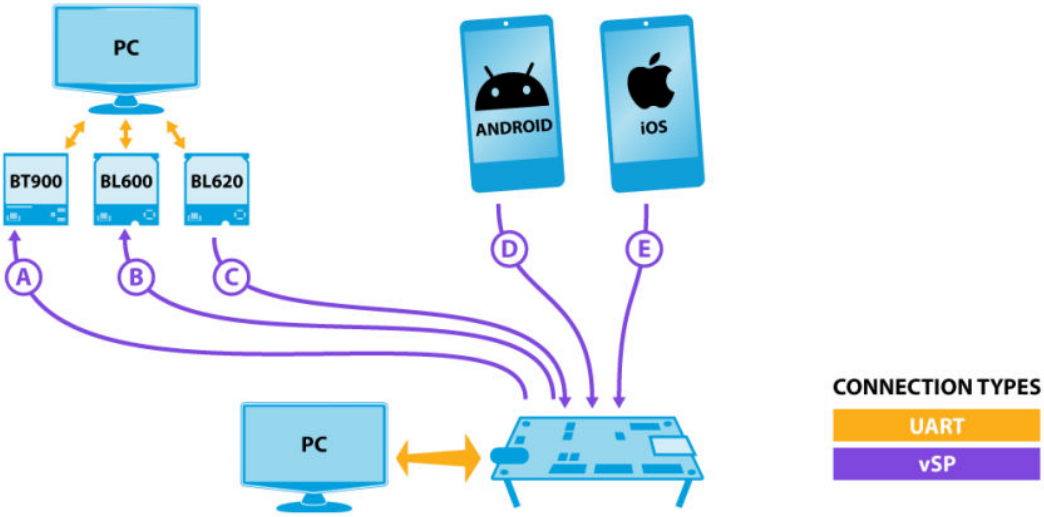


Figure 7 - Possibilities with vSP (image courtesy of Laird)

vSP (Virtual Serial Port) is a method of sending data over BLE (Bluetooth Low Energy), similar to how SPP works in Classic Bluetooth. Laird developed vSP as a proprietary feature enhancement at the request of our customers. vSP is relatively low throughput and does not take advantage of the low power features of BLE.
For further details about vSP, please see the vSP smartBASIC Application Note.

In Windows 10, the vSP device identifies as a Bluetooth Device with a heartbeat icon.

See for an iOS code example with Getting Started guide:
https://github.com/PalmSens/MethodSCRIPT_Examples

8.5.4 Hardware settings for programming the Laird BT900 module

Set SW2 to the BT_PWR_ON position to enable the BT900 module power supply.
Set SW3 to the DEV position to enter the programming mode of the BT900 module
The BT900 can be programmed via USB by setting all switches of SW4 (BT_to_USB) to ON and set all switches OFF for SW7 (ES_to_USB) and SW5 (BT_to_ES).
See https://www.palmsens.com/knowledgebase-article/change-bluetooth-settings/

8.6 Using the Real-Time Clock

Place CR1216 in BT1 battery-holder as backup supply for the S-3590A RTC.
S-35390A REAL-TIME CLOCK (ablic.com)

The RTC can be accessed via MethodScript i2c commands.
See https://www.palmsens.com/wp-content/uploads/2022/01/MethodSCRIPT-v1_3.pdf
Chapter13.5 I2C example – Real time clock

8.7 Using the EEPROM

Non-Volatile data can be stored in the 32-Kbit I2C Serial EEPROM (24LC32AT)
The EEPROM can be accessed via MethodSCRIPT I2C commands.
See https://www.palmsens.com/wp-content/uploads/2022/01/MethodSCRIPT-v1_3.pdf
Chapter 13.6 I2C example - EEPROM example

8.8 Using an Arduino MKR Zero

To access the EmStat4M from the Arduino MKR set all switches of SW6 to ON and all switches OFF for SW4, SW5, SW7.

Note: There's no hardware-handshaking (CTS/RTS) between the Arduino MKR and the EmStat4M.

For code examples see:
MethodSCRIPT_Examples/MethodSCRIPTExample_Arduino at master · PalmSens/MethodSCRIPT_Examples (github.com)

Remark for development board V2 only

This only applies to boards with the text found at the bottom of the board: "ES Dev V2".

The MKR can be used with the Arduino software using the USB connector from the MKR. The development board can be power from the MKR in this case by setting SW10 to the MKR position and make sure SW1 (Pwr to MKR) is in the OFF position.

Caution: Having SW10 in the MKR position and SW1 in the ON position may damage the MKR and/or the board.

8.9 Powering the Development Board

The EmStat4M Development board can be powered from 3 sources selectable with SW10:

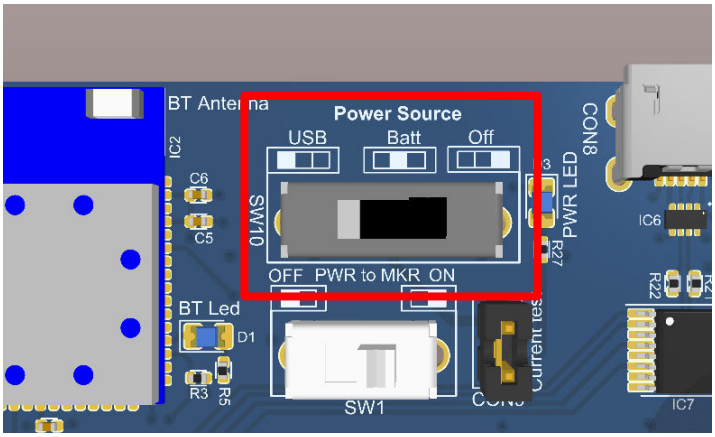


Figure 8 - SW10 on the development board

8.9.1 Powering from the micro-USB or USB-C port

When an EmStat4 HR is installed, the maximum current drawn from the USB port will exceed 500mA when using the 100 mA range. Use a Y-cable to split the total current into 2 USB ports or make sure the port can handle current up to 900mA.

EmStat4HR requires up to 900 mA in the 100 mA range. Make sure to use the USB Y-cable to ensure enough power can be drawn from the USB ports.

8.9.2 Powering from a Lithium-Ion battery

The JST PH2 connector (CON10) is suitable for Adafruit batteries like https://www.adafruit.com/product/328 or https://www.adafruit.com/product/2011
It is recommended to use a battery capacity of 2000mAh or more.

The battery is charged when the Development board is connected via CON8 (Micro USB). The charge current is set to ~150mA, a 2000mAh battery may take 10 to 14 hours to be completely charged. The charge current can be changed by changing the value of R31
See the datasheet of IC8 for detailed information:
https://www.analog.com/media/en/technical-documentation/data-sheets/4053-4.2fs.pdf

8.9.3 Powering from an Arduino MKR board

In case an Arduino MKR board is connected to the bottom side of the development board, all boards can be powered by a single USB connection to the Arduino. This can be done by setting SW10 to OFF and connecting the Arduino board via a micro-USB cable to the PC or other USB power source.

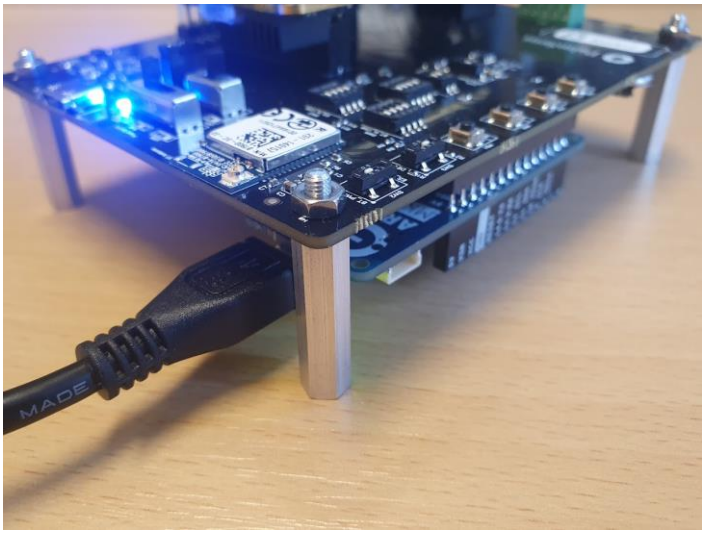


Figure 9 - Powering all stacked boards via the Arduino MKR

8.10 Heat dissipation when using EmStat4M HR

The EmStat4M HR is designed for higher currents and is protected against over-current and high temperature. To prevent entering these overload conditions make sure that the setup has good airflow. When the EmStat4M is used in a closed environment (housing) forced air-cooling (fan) is strongly recommended.

When using the maximum current range with an EmStat4M HR, proper heat dissipation around the heat sinks is essential.

Note: MethodSCRIPT error 0x0032 will be thrown in case of an overload condition
See https://www.palmsens.com/wp-content/uploads/2022/01/MethodSCRIPT-v1_3.pdf Section 14 “Error handling”

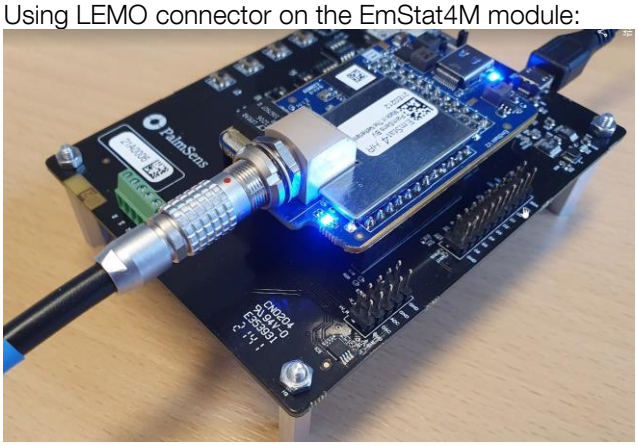
8.11 Connecting the cell (or sensor)

To do valid measurements on an electrochemical cell, the RE, WE and CE of the EmStat4M module and the Sense lead (for the EmStat4M HR) need to be connected.

Make sure to always connect the Sense lead when using the EmStat4M HR. In most common situations, the Sense lead connects directly to the WE.

There are multiple ways to connect a sensor or cell to the EmStat4M.

Using LEMO connector on the EmStat4M module:



Or using the screw terminals on the development board:

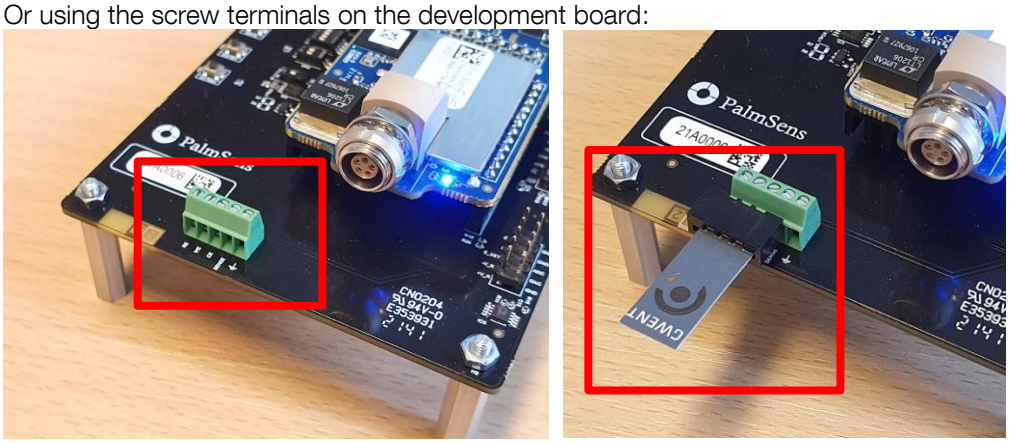


Figure 10 - Using the screw terminals

A small screwdriver is supplied for using the screw terminals.
The SPE connector for use with Screen Printed Electrodes that have the common 2.54 mm pitch can be screwed in directly to the screw terminals.

When using the EmStat4M HR:
In case of using the SPE connector with an EmStat4M HR, make sure the Sense is also connected to the WE. This can be done by using a small wire and screwing it in together with the SPE connector.

9 Software and documentation

9.1 Communication protocols

The EmStat4M module works with the human-readable MethodSCRIPT protocol for direct serial (USB or UART) communication. This allows for connecting and using the module in a terminal window or with any microcontroller, PC or operating system.
The simple script language allows for running all supported electrochemical techniques and makes it easy to combine different measurements and other tasks.

Code examples are available for:

- Android
- Arduino
- C/C++
- Python
- iOS
- and C#

For more information, documentation and MethodSCRIPT code examples, see:

www.palmsens.com/methodscript

The EmStat4M Communication Protocol can be found in the section "Downloads" on:

www.palmsens.com/es4m-dev

MethodSCRIPTs can be sent to the EmStat4M by using the EmStat4M communications protocol. This is the device-specific communication protocol and also covers other device-specific functionality like calibration and firmware updates.

9.2 Using PSTrace for Windows

The EmStat4M can be used with PSTrace for Windows to run all the supported electrochemical techniques and to some extend to use the auxiliary and GPIO pins.
PSTrace is ideal for evaluating the performance of the EmStat4M, but can also be used for generating MethodSCRIPT (see next paragraph for more information). The MethodSCRIPT snippets can again be used in any custom code project that controls the EmStat4M.

It takes only 5 minutes to read through the help section "PSTrace basic principles'. This is highly recommended for first-time users.

9.2.1 Generating MethodSCRIPT

The Method Editor in PSTrace can be used to define a method (set of measurement parameters).

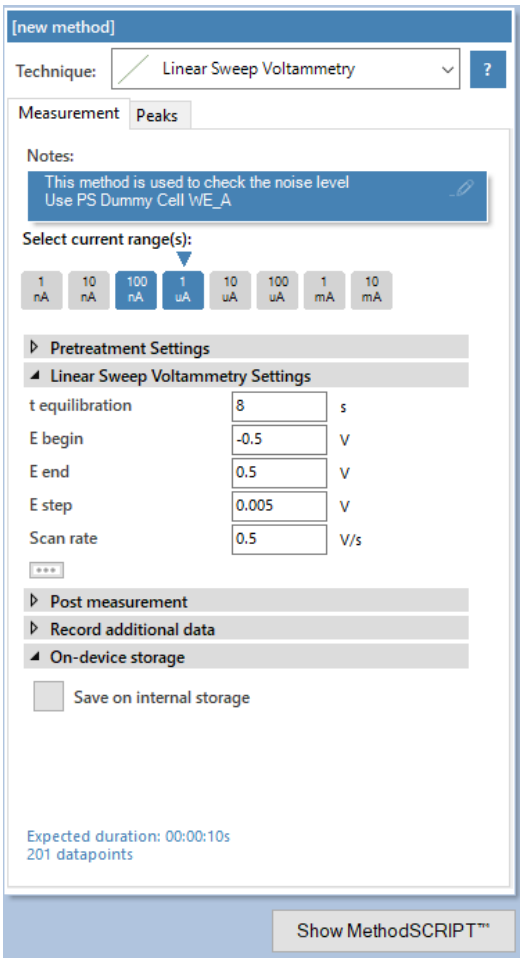


Figure 11 - Method Editor in PSTrace

When being connected to the EmStat4M, the Method Editor will automatically validate the parameters with the EmStat4M hardware any time a parameter is changed. Pressing the button “Show MethodSCRIPT” opens a window that contains the MethodSCRIPT which is generated for the method. See next section for more information about MethodSCRIPT.

9.3 Windows drivers

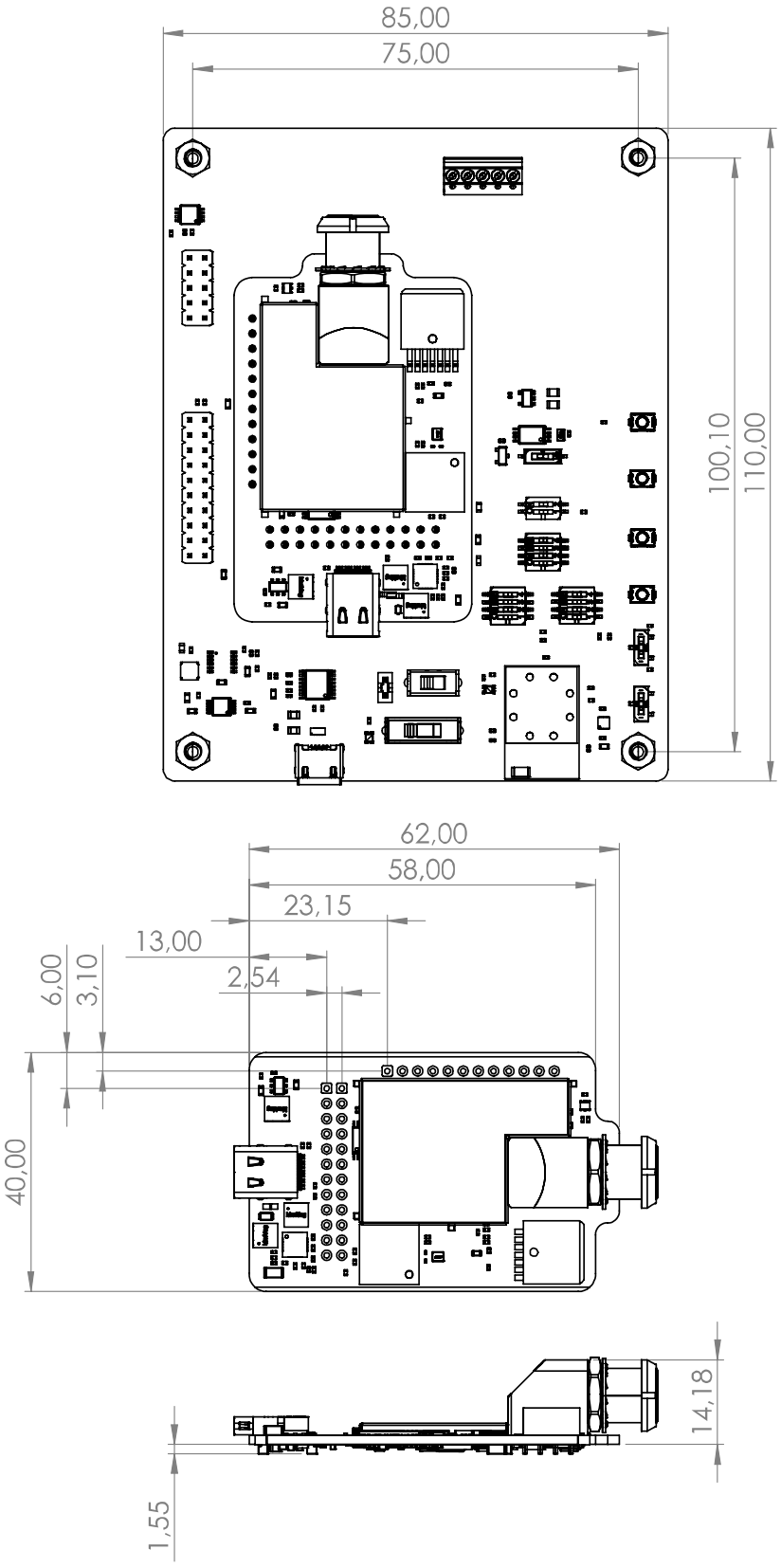
The EmStat4M USB connection with Windows uses the USB CDC device class, which emulates a RS-232 “Serial port” for ease of use. This class can be used without drivers in most operating systems, the device will show up as a “Serial port”.

Starting from Windows 10, the EmStat4M is supported without having to install drivers. The PalmSens USB drivers can be used to ensure the devices are named “EmStat4”, rather than the generic “USB Serial Device”. This driver is automatically installed during the installation of PSTrace.

9.3.1 Development Board Micro USB connector

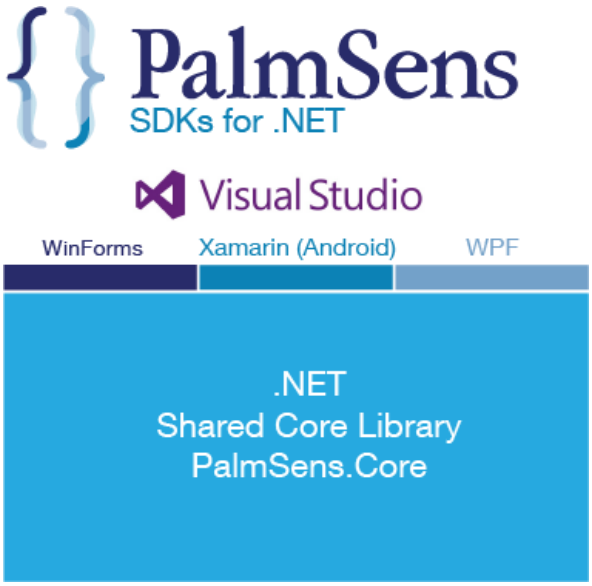
The Micro USB connector on the development board is implemented using an FTDI USB to serial converter. A USB driver is needed for this converter. However, most operating systems will install this driver automatically. If this is not the case, the drivers can be found here.

10 Development board and module dimensions



11 Software Development Kits for .NET

Develop your own application in no time for use with any PalmSens instrument or potentiostat module. Our SDKs are free of charge.



There are three PalmSens Software Development Kits (SDKs) for .NET. Each SDK can be used with any of our instruments or OEM potentiostat modules to develop your own software. The SDK's come with a set of examples that shows how to use the libraries. PalmSens SDKs with examples are available for the following .NET Frameworks:

- WinForms
- Xamarin (Android)
- WPF

Each SDK comes with code examples for:

- Connecting
- Running measurements and plotting data
- Manual control of the cell
- Accessing and processing measured data
- Analyzing and manipulating data
- Peak detection
- Equivalent Circuit Fitting on impedance data
- Saving and loading files

```
/// <summary>
/// Initializes the EIS method.
/// </summary>
1 reference
private void InitCVMethod()
{
    _methodEIS = new ImpedimetricMethod();
    _methodEIS.ScanType = ImpedimetricMethod.enumScanT
    _methodEIS.Potential = 0.0f; //0.0V DC potential
    _methodEIS.Eac = 0.01f; //0.01V RMS AC potential a
    _methodEIS.FreqType = ImpedimetricMethod.enumFrequ
    _methodEIS.MaxFrequency = 1e5f; //Max frequency is
    _methodEIS.MinFrequency = 10f; //Min frequency is
    _methodEIS.nFrequencies = 11; //Sample at 11 diffe

    _methodEIS.EquilibrationTime = 1f; //Equilabrates
    _methodEIS.Ranging.StartCurrentRange = new Current
    _methodEIS.Ranging.MinimumCurrentRange = new Curre
    _methodEIS.Ranging.MaximumCurrentRange = new Curre
}
```

More information and downloads are available here:

www.palmsens.com/sdk

12 Document Revison History

Revision	Section/Figure/Entry	Correction
3-2022-001	Document release	
4-2022-002	Terms and definitions	Added new relevant and removed irrelevant terms

13 Disclaimer

Products sold by PalmSens BV are not designed, intended or approved for use in life support, implantable medical devices, transportation, nuclear, safety or other equipment where malfunction of the Product can reasonably be expected to result in personal injury, death, severe property damage or severe environmental harm. Buyer represents and warrants that it will use appropriate safeguards to minimize potentially dangerous consequences associated with Product failure. Notwithstanding the foregoing, if Buyer uses or sells Products for in such critical applications, it does so at Buyer's own risk and Buyer agrees to defend, indemnify and hold harmless PalmSens BV from any and all damages, claims, suits or expenses resulting from such use

Information in this document is believed to be accurate and reliable. However, PalmSens BV does not give any representations or warranties, express or implied, as to the accuracy or completeness of such information. PalmSens BV furthermore takes no responsibility whatsoever for the content in this document if provided by any information source outside of PalmSens BV. PalmSens BV reserves the right to change without notice the information published in this document, including, without limitation, the specification and the design of the related semiconductor products, software and applications. All use of PalmSens BV products, software and applications referred to in this document is subject to PalmSens BV's Standard Terms and Conditions of Sale.

Please do not hesitate to contact PalmSens for more details:
info@palmsens.com

PalmSens BV
The Netherlands
www.palmsens.com