



**Bio-Stream Diagnostics Inc**

Source Measurement Unit – Amperostat - Potentiostat

User's Manual

Revision 7

## Contents

Getting Started.....	4
General Information .....	4
Contact Information.....	4
Support Information .....	4
Safety symbols and terms.....	4
Inspection.....	4
Options and accessories.....	4
Product Overview .....	5
Specifications .....	5
Features .....	6
Power .....	7
Power Up.....	7
Connections .....	7
Universal Serial Bus.....	7
Sensor Port.....	7
External Sensor Connection .....	8
Basic Operation.....	9
Overview .....	9
traxInsight .....	9
Source-Delay-Measure Mode .....	9
Speed .....	9
Continuous Measurement Mode.....	9
Serial Communication.....	10
Data Input .....	10
Data Output .....	10
Action .....	10
Message .....	11
Data .....	11
Command Reference .....	12
Command Queue .....	12
Enable/Disable DAC .....	12

Enable/Disable ADC .....	13
Abort Command.....	14
Voltammetry .....	15
Linear Sweep (Continuous Capture Mode).....	15
Linear Sweep 2 (Source-Delay-Measure Mode) .....	15
Square Wave Voltammetry.....	16
Chronoamperometry .....	19
Capture Constant Current.....	19
FindGateForAmperage.....	<b>Error! Bookmark not defined.</b>
Electrochemical Impedance Spectroscopy .....	19
Informational .....	20
Get Serial Number.....	20
Get Sensor Serial Number.....	20
Get Firmware Revision.....	20
Get Calibration Information.....	20
MaxStepsForRange .....	<b>Error! Bookmark not defined.</b>
FindBiasForAmperage.....	<b>Error! Bookmark not defined.</b>

# Getting Started

## General Information

### Contact Information

Bio-Stream Diagnostics Inc  
9403 20 Ave NW  
Edmonton, Alberta, Canada  
T6N 1E5

### Support Information

Phone: 1-780-800-1900

Email: [support@bio-stream.ca](mailto:support@bio-stream.ca)

### Safety symbols and terms

Not yet CE or UL approved.

### Inspection

The traxReader was carefully inspected both mechanically and electronically before being packaged for shipment. When unpacking the traxReader, check for any obvious damage that may have occurred during shipment. We suggest keeping the shipping container in case it needs to be returned for service.

### Options and accessories

1. Carrying Case
2. USB Cable
3. Sensor Breakout Board
4. Breakout Board Cables

## Product Overview

The traxReader was designed to characterize and commercialize biological electrochemical sensors. It has all the required circuitry to produce stable voltages for Gate/Working electrode, Source/counter electrode, and drain connections as well as reliably read the source/drain bias current. Further the gate amperage can be monitored, and a reference electrode can be utilized.

## Specifications

Gate/WE Voltage Range	-1500 mV to 1500mV (When Source is 1500mV)
Gate/WE Voltage Resolution	~45 $\mu$ V
Gate/WE Voltage Stability	+/- 0.1 %
Gate/WE Voltage Accuracy	+/- 0.1 %
Gate/WE Response Time	~5 $\mu$ s
Gate/WE Output Resistance	~2 $\Omega$
Max Gate/WE Current *	~+/-60 $\mu$ A
Gate/WE Current Resolution *	~14 pA
Source/CE Voltage	0 to 3000mV (Gate & Drain are relative to Source)
Source/CE Voltage Stability	+/- 0.1 %
Source/CE Voltage Accuracy	+/- 0.1 %
Source/CE Response Time	~5 $\mu$ s
Source/CE Output Resistance	~100 m $\Omega$
Drain Voltage	-1500 mV to 1500mV (When Source is 1500mV)
Drain Voltage Stability	+/- 0.1 %
Drain Voltage Accuracy	+/- 0.1 %
Drain Response Time	~5 $\mu$ s
Drain Output Resistance	~2 $\Omega$
Reference Voltage	0 to 3000mV
Reference Input Resistance	>50 G $\Omega$
Max Source/Drain Current Range 1	27mA
Source/Drain Current Resolution Range 1	3.6nA
Max Source/Drain Current Range 2	3mA
Source/Drain Current Resolution Range 2	0.36nA
Available Electrode Oversampling Rates	32 @ 38400Hz, 64 @ 19200Hz, 128 @ 9600Hz, 256 @ 4800Hz, 512 @ 2400Hz, 1024 @ 1200Hz, 2048 @ 600Hz, 4096 @ 300Hz, 8192 @ 150Hz , 16384 @ 75Hz, 20480 @ 60Hz, 24576 @ 50Hz, 40960 @ 30Hz, 49152 @ 25Hz, 81920 @ 15Hz, 98304 @ 12.5Hz
ADC Buffer Size	4096 for each of the electrodes

\* Note: If using gate/working electrode current, you need to operate the gate within 0 to 2.3 Volts.

#### Features

Custom command patterns	Internal calibration
Command batching and repetition	Square wave voltammetry
Chronoamperometry	Linear sweep voltammetry
USB-C port	Pulse voltammetry
USB power pass-through	Integrated sensor port
	breakout board for external sensors
traxInsight for data storage and analysis	myTrax on mobile device for field usage

## Power

### Power Up

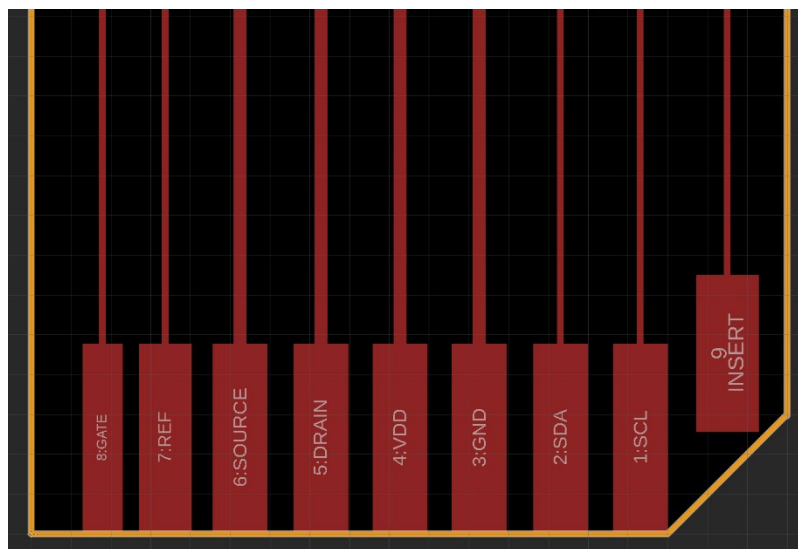
When powering up the unit, we recommend removing any sensor from the port or remote connection, as there may be undesirable voltage application while the internal components are powering up.

## Connections

### Universal Serial Bus

The traxReader can connect to any USB device if an appropriate cable is available. It is packaged with a USB-C cable. The device itself has a female USB-C for power pass through. Serial communication as well as charging is done via this port.

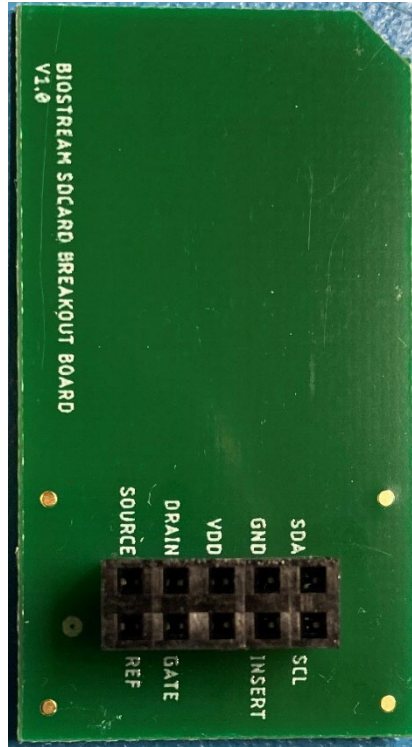
### Sensor Port



The traxReader has a 9-pin connection for the direct insertion of a bio sensor. Currently anyone can make or order a sensor with these pins and dimensions and the reader will read them.

## External Sensor Connection

The traxReader comes with a breakout board that allows for the sensor connection to be accessed by standard breadboard wires.





# Basic Operation

## Overview

### traxInsight

The traxReader is integrated with the Bio-Stream traxInsight system. traxInsight is a combination capture and analysis software system created for the express purpose of storing research related information created using the traxReader. Look for detailed instructions in the traxInsight manual.

### Source-Delay-Measure Mode

The traxReader operates on a series of source-delay-measure cycles. Each of these cycles consists of the following:

1. Set the source level.
2. Wait for a period of time.
3. Make the measurement.

### Speed

The frequency setting of each command determines how fast the reader takes measurements. For example, with a setting of 1 Hz, the reader will set the source and take measurements once per second.

The oversampling rate specifies how many samples are taken per measurement and will determine the lowest possible frequency setting and the amount of noise in each measurement. Lowering the oversampling rate can increase speed at a cost of increased noise, and increasing the oversampling rate will decrease speed but also decrease noise.

### Continuous Measurement Mode

The traxReader can operate on a continuous measurement mode. While using this mode, measurements will happen continuously on the electrodes of choice at the desired oversampling rate. Then the task of scanning, pulsing, biasing electrode etc. will be done at a separate rate.

The oversampling rate specifies how many samples are taken per measurement and will determine the lowest possible frequency setting and the amount of noise in each measurement. Lowering the oversampling rate can increase speed at a cost of increased noise, and increasing the oversampling rate will decrease speed but also decrease noise.

## Serial Communication

The traxReader has an API available via a serial interface. The following are the required settings:

Port	COM1-99
Baud rate	115200
Data bits	8
Stop bits	1
Parity	None
Flow Control	None

## Data Input

Data input comes in the form of command sets that are created by the traxInsight software but can also be made by advanced users using JSON over the serial port. All commands share at least two fields as listed below, but many of the commands contain unique fields and are documented with each of the commands.

Field Name	Data Type	Value
name	String	The name of the command you want to run
id	String	The identifier you want attached to the data returned by this command

## Data Output

The traxReader communicates with traxInsight as well as any other device via serial communication and its serial API. The data output from the serial API is in the form of JSON with the following fields:

### Action

Field Name	Data Type	Value
Type	String	Name of the JSON object type "action"
action	String	Name of the action happening, eg: "StartCommand" or "EndCommand"
name	String	Name of the command related to the action, eg: "LinearSweep"
id	String	Identifier of command the action is related to, eg: "1234567"
t	Long	The time the action was performed on the device, reported in seconds, since the device was turned on

```
{  
  "type": "action",  
  "action": "commandStart",  
  "name": "LinearSweep",
```

```

    "id": "2",
    "t": 0
  }

```

### Message

Field Name	Data Type	Value
type	String	Name of the JSON object type "message"
text	String	Message content
t	Long	The time the action was performed on the device, reported in seconds since the device was turned on

```

{
  "type": "message",
  "text": "this is a message",
  "t": 0
}

```

### Data

This type of data is returned as part of any =0data capture command. <b>Field Name</b>	Data Type	Value
type	String	Name of the JSON object type "data"
id	String	Identifier of command the data is related to, eg: "1234567"
dc	Float	Drain current in milliamps
gc	Float	Gate current in milliamps
rv	Float	Reference Voltage in millivolts
gv	Float	Gate voltage in millivolts
sv	Float	Source voltage in millivolts
dv	Float	Drain voltage in millivolts
d	Integer	Direction of sweep (0 – low gate voltage to high gate voltage or 1 – high gate voltage to low gate voltage)
r	Integer	Number of repetitions
s	Integer	Number of steps
t	Long	Time in seconds

```

{
  "type": "data",
  "id": "some identifier",
  "dc": 1,
  "gc": 1,

```

```
"rv": 1,
"gv": 1,
"sv": 1,
"dv": 1,
"d": 1,
"r": 1,
"s": 1,
"t": 0
}
```

## Command Reference

### Command Queue

The reader will accept and queue up to 10 commands at a time and run them in a first in first out order. When a command completes you can immediately send another command in order to prevent any downtime between commands. This may be important when attempting things like cyclic voltammetry.

### Enable/Disable DAC

These commands turn on the source, drain, and gate electrodes. If you run a command without first enabling the system you will receive only electrical noise. If you do not disable the electrodes then the device will remain active after the commands complete but no data will be reported. When the DAC is enabled, all values are referenced to the source voltage.

Field Name	Data Type	Value
name	String	EnableDAC
id	String	Identifier of command the action is related to, eg: "1234567"
drainVoltage	Integer	Drain voltage in mV, in relation to source Voltage.
gateVoltage	Integer	Gate voltage in mV, in relation to source Voltage.
sourceVoltage	Integer	Source voltage in mV, 0 to 3000.
isGateFeedbackAmp	Int	0 off, 1 gate/source potential being varied with ref electrode.

```
{
  "name": "EnableDAC",
  "id": "id0",
  "drainVoltage": 0,
  "gateVoltage": 0,
  "sourceVoltage": 1500,
  "isGateFeedbackAmp": 0
}
```

```
}
```

Field Name	Data Type	Value
name	String	DisableDAC
id	String	Identifier of command the action is related to, eg: "1234567"

```
{  
  "name": "DisableDAC",  
  "id": "id1"  
}
```

### Enable/Disable ADC

This will start the ADC sampling at the rate you want and on the electrodes you wish to sample. There is also an option to select the range the drain current is measured in.

Field Name	Data Type	Value
name	String	LinearSweep
id	String	Identifier of command the action is related to, eg: "1234567"
overSamplingRate	Integer	1-16, see table below
captureGateVoltage	Integer	0 – No, 1 - Yes
captureDrainVoltage	Integer	0 – No, 1 - Yes
captureSourceVoltage	Integer	0 – No, 1 - Yes
captureRefVoltage	Integer	0 – No, 1 - Yes
captureDrainAmperage	Integer	0 – No, 1 - Yes
captureGateAmperage	Integer	0 – No, 1 - Yes
drainCurrentRange	Integer	0 – 27mA to 3.6nA 1 - 3mA to 0.36nA

The following Oversampling Rates are available. Further the sampling time is the speed listed multiplied by the number of electrodes that are being sampled.

- 1) 32 @ 38400Hz
- 2) 64 @ 19200Hz
- 3) 128 @ 9600Hz
- 4) 256 @ 4800Hz
- 5) 512 @ 2400Hz
- 6) 1024 @ 1200Hz
- 7) 2048 @ 600Hz
- 8) 4096 @ 300Hz
- 9) 8192 @ 150Hz

- 10) 16384 @ 75Hz
- 11) 20480 @ 60Hz
- 12) 24576 @ 50Hz
- 13) 40960 @ 30Hz
- 14) 49152 @ 25Hz
- 15) 81920 @ 15Hz
- 16) 98304 @ 12.5Hz

```
{
  "name": "EnableADC",
  "id": "id3",
  "overSamplingRate": 16,
  "captureGateVoltage": 1,
  "captureDrainVoltage": 0,
  "captureSourceVoltage": 0,
  "captureRefVoltage": 0,
  "captureDrainAmperage": 0,
  "captureGateAmperage": 0,
  "drainCurrentRange": 0
}
```

Field Name	Data Type	Value
name	String	DisableADC
id	String	Identifier of command the action is related to, eg: "1234567"

```
{
  "name": "DisableADC",
  "id": "id1"
}
```

### Abort Command

The abort command lets you stop running any existing commands without having to wait for them to finish. Further it will clear all queued commands. This does not turn off the DAC or the ADC.

```
{
  "name": "AbortCommand",
  "id": "id3"
}
```

## Voltammetry

### Linear Sweep (Continuous Capture Mode)

This command will perform a linear sweep going from either high to low voltage or low to high voltage depending on the direction setting. This command can be run on its own, in series, or as part of a repetition (traxInsight). This command is the basis of Cyclic Voltammetry.

Field Name	Data Type	Value
name	String	LinearSweep
id	String	Identifier of command the action is related to, eg: "1234567"
lowVoltage	Integer	Gate voltage in mV
highVoltage	Integer	Gate voltage in mV
numberOfSteps	Integer	Number of steps between low and high gate voltage
Direction	Integer	0 for low to high, 1 for high to low
Frequency	Float	Speed to change step, also the sampling rate
repetitionNumber	Integer	If you are sending this as part of a repetition this will be added to the data returned
Electrode	Integer	0 – Gate, 1 - Drain

```
{  
  "name": "LinearSweep",  
  "id": "id1",  
  "lowVoltage": 0,  
  "highVoltage": 1000,  
  "numberOfSteps": 25,  
  "direction": 0,  
  "frequency": 25,  
  "drainVoltage": -300,  
  "electrode": 0  
}
```

### Linear Sweep 2 (Source-Delay-Measure Mode)

This command will perform a linear sweep going from either high to low voltage or low to high voltage depending on the direction setting. This command can be run on its own, in series, or as part of a repetition. The primary difference between this command the Linear Sweep is that the ADC sampling will be coordinated in the source delay measurement cycle.

Field Name	Data Type	Value
name	String	LinearSweep2

id	String	Identifier of command the action is related to, eg: "1234567"
lowVoltage	Integer	Drain voltage in mV
highVoltage	Integer	Drain voltage in mV
numberOfSteps	Integer	Number of steps between low and high voltage
direction	Integer	0 for low to high, 1 for high to low
Frequency	Float	Speed to change step, also the sampling rate.
electrode		0 is gate, 1 is drain
overSamplingRate	Integer	Sample as previous
captureGateVoltage	Integer	0 – No, 1 - Yes
captureDrainVoltage	Integer	0 – No, 1 - Yes
captureSourceVoltage	Integer	0 – No, 1 - Yes
captureRefVoltage	Integer	0 – No, 1 - Yes
captureDrainAmperage	Integer	0 – No, 1 - Yes
captureGateAmperage	Integer	0 – No, 1 - Yes
drainCurrentRange	Integer	0 – 27mA to 3.6nA 1 - 3mA to 0.36nA

```


{
  "name": "LinearSweep2",
  "id": "id3",
  "lowVoltage": -900,
  "highVoltage": 100,
  "numberOfSteps": 25,
  "direction": 0,
  "frequency": 25,
  "gateVoltage": 500,
  "electrode": 1,
  "overSamplingRate":8,
  "captureGateVoltage":1,
  "captureDrainVoltage":1,
  "captureSourceVoltage":1,
  "captureRefVoltage":1,
  "captureDrainAmperage":1,
  "captureGateAmperage":1,
  "drainCurrentRange": 0
}

```

### Square Wave

This command will perform a square wave going from either high to low voltage or low to high voltage depending on the direction setting.



Field Name	Data Type	Value
name	String	SquareWave
Id	String	Identifier of command the action is related to, eg: "1234567"
lowVoltage	Integer	low voltage requested as part of the ending transfer curve, the amplitude will be added and subtracted from this number
highVoltage	Integer	high voltage requested as part of the ending transfer curve, the amplitude will be added and subtracted from this number
amplitudeVoltage	Integer	number of mV higher and lower of each stepping point.
numberOfSteps	Integer	Number of steps between low and high voltage
Direction	Integer	0 for low to high, 1 for high to low
Frequency	Float	Speed to change step, also the sampling rate
electrode 	Integer	0 gate, 1 drain
repetitionNumber	Integer	Current repetition

```

{
  "name": "SquareWave",
  "id": "id6",
  "lowVoltage": 0,
  "highVoltage": 1000,
  "amplitudeVoltage": 25,
  "numberOfSteps": 25,
  "direction": 0,
  "repetitionNumber": 1,
  "frequency": 25,
  "electrode": 0
}

```

### Square Wave 2

This command will perform a square wave going from either high to low voltage or low to high voltage depending on the direction setting.

Field Name	Data Type	Value
name	String	SquareWave

Id	String	Identifier of command the action is related to, eg: "1234567"
lowVoltage	Integer	low voltage requested as part of the ending transfer curve, the amplitude will be added and subtracted from this number
highVoltage	Integer	high voltage requested as part of the ending transfer curve, the amplitude will be added and subtracted from this number
amplitudeVoltage	Integer	number of mV higher and lower of each stepping point.
numberOfSteps	Integer	Number of steps between low and high voltage
Direction	Integer	0 for low to high, 1 for high to low
Frequency	Float	Speed to change step, also the sampling rate
RepetitionNumber	Integer	
electrode	Integer	0 gate, 1 drain
overSamplingRate	Integer	Sample as previous
captureGateVoltage	Integer	0 – No, 1 - Yes
captureDrainVoltage	Integer	0 – No, 1 - Yes
captureSourceVoltage	Integer	0 – No, 1 - Yes
captureRefVoltage	Integer	0 – No, 1 - Yes
captureDrainAmperage	Integer	0 – No, 1 - Yes
captureGateAmperage	Integer	0 – No, 1 - Yes
drainCurrentRange	Integer	0 – 27mA to 3.6nA 1 - 3mA to 0.36nA

```

{
  "name": "SquareWave2",
  "id": "id6",
  "lowVoltage": 0,
  "highVoltage": 1000,
  "amplitudeVoltage": 25,
  "numberOfSteps": 25,
  "direction": 0,
  "repetitions": 1,
  "frequency": 25,
  "electrode": 0,
  "repetitionNumber": 1,
  "overSamplingRate": 8,
  "captureGateVoltage": 1,

```

```

"captureDrainVoltage":1,
"captureSourceVoltage":1,
"captureRefVoltage":1,
"captureDrainAmperage":1,
"captureGateAmperage":1,
"drainCurrentRange": 0

```

```

}

```

## Chronoamperometry

### Capture Constant Current

Once the ADC and DAC are setup, this command wait for a period of time and allow all the data captured to be tagged during that period.

Field Name	Data Type	Value
name	String	CaptureConstantCurrent
id	String	Identifier of command the action is related to, eg: "1234567"
maxTime	Long	How long to run in milliseconds

```

{
  "name": "CaptureConstantCurrent",
  "id": "id5",
  "maxTime": 1000
}

```

## Electrochemical Impedance Spectroscopy

This feature is still in development, and we welcome feedback on its implementation. At this time, you should not attempt to run faster than 1000Hz.

Field Name	Data Type	Value
name	String	EIS
id	String	Identifier of command the action is related to, eg: "1234567"
amplitudeVoltage	Float	The amplitude of the sine wave in mV.
dcOffset	Integer	The offset from source/CE in mV
frequency	Integer	The frequency of the sine wave in hertz

```
{
  "name": "EIS",
  "id": "id6",
  "amplitudeVoltage": 500,
  "dcOffset": 0,
  "frequency": 10
}
```

## Informational

### Get Serial Number

This command returns the serial number assigned to the board.

```
{
  "name": "GetSerialNumber",
  "id": "id3"
}
```

### Get Sensor Serial Number

This command returns the serial number assigned to the board.

```
{
  "name": "GetSensorSerialNumber",
  "id": "id3"
}
```

### Get Firmware Revision

This command returns the firmware revision on the given board.

```
{
  "name": "GetFirmwareVersion",
  "id": "id3"
}
```

### Get Calibration Information

The calibration stored in the traxReader can be extracted and returned using this command.

```
{
  "name": "GetCalibration",
  "id": "id3"
}
```