

EZO-EC™

Embedded Conductivity Circuit

ISO 7888 Compliant

(determination of electrical conductivity)

Reads **Conductivity = $\mu\text{S}/\text{cm}$**
Total dissolved solids = ppm
Salinity = PSU (ppt) 0.00 – 42.00
Specific gravity
(sea water only) = 1.00 – 1.300

Range **0.07 – 500,000+ $\mu\text{S}/\text{cm}$**

Accuracy **+/- 2%**

EC reading time **600ms**

Supported probes **K 0.01 – K 10.2 any brand**

Calibration **2 or 3 point**

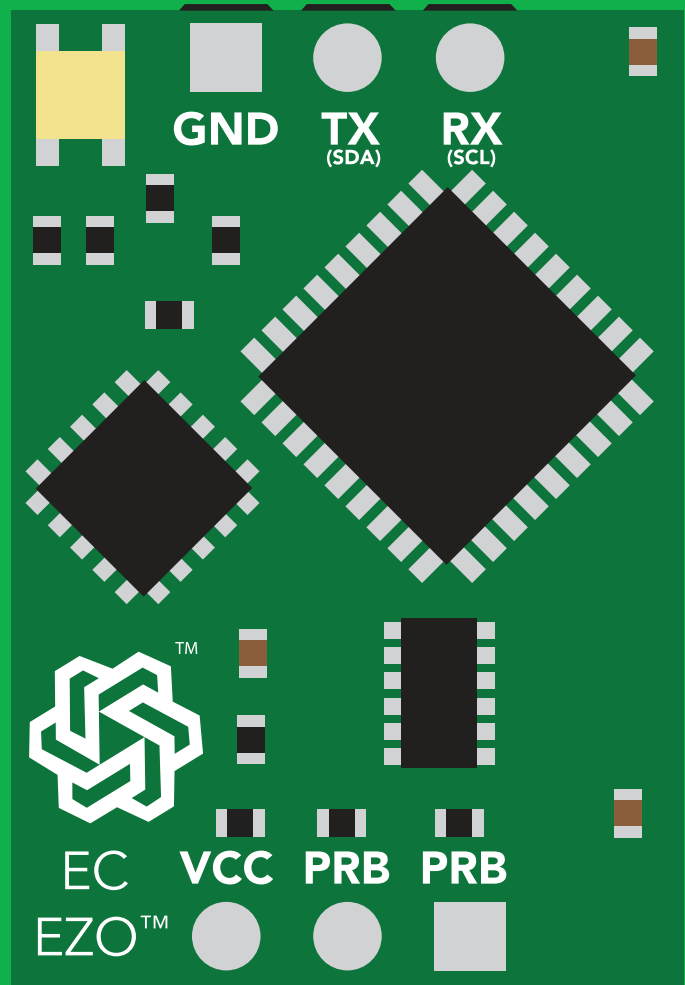
Temp compensation **Yes**

Data protocol **UART & I²C**

Default I²C address **100 (0x64)**

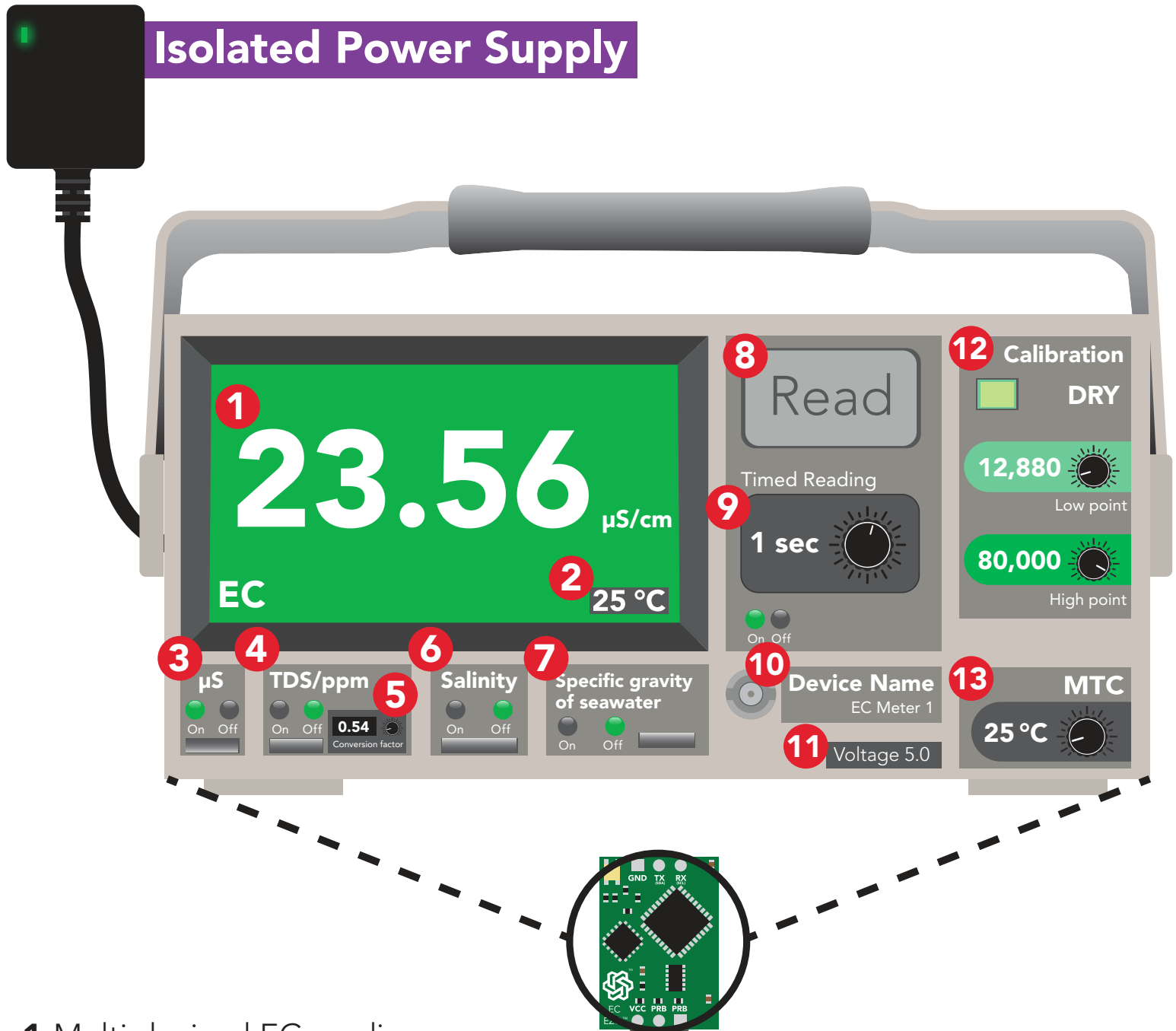
Operating voltage **3.3V – 5V**

Data format **ASCII**



PATENT PROTECTED

The EZO™ EC Circuit has all the features of this bench top meter.



The EZO Complete-EC™ is compatible with any brand of EC probe from K 0.01–K10.2

Conductivity probe range

The EZO™ Conductivity circuit is compatible with any brand of two-conductor conductivity probe, ranging from:

K 0.01



K 10.2

Atlas Scientific™ has tested three different K value probe types:

K 0.1



accurate reading range

0.07 μ S/cm – 50,000 μ S/cm

TDS (ppm) 0 – 25,000

Salinity (ppt) 0 – 33

K 1.0



accurate reading range

5 μ S/cm – 200,000+ μ S/cm

TDS (ppm) 2 – 100,000

Salinity (ppt) 0 – 42*

**salinity scale cannot go any higher*

K 10



accurate reading range

10 μ S/cm – 1S/cm

TDS (ppm) 5 – 500,000

Salinity (ppt) 0 – 42*

**salinity scale cannot go any higher*

Atlas Scientific™ does not know what the accurate reading range would be for conductivity probes, other than the above mentioned values. Determining the accurate reading range of such probes, i.e. **K 2.6**, or **K 0.66**, is the responsibility of the embedded systems engineer.

Resolution

The EZO™ Conductivity circuit, employs a method of scaling resolution. As the conductivity increases the resolution between readings decreases.

The EZO™ Conductivity circuit will output conductivity readings where the first **4 digits** are valid and the others are set to 0. This excludes conductivity readings that are less than 9.99. In that case, only 3 conductivity digits will be output.

0.07 – 99.99

Resolution = **0.01 μ S/cm**

100.1 – 999.9

Resolution = **0.1 μ S/cm**

1,000 – 9,999

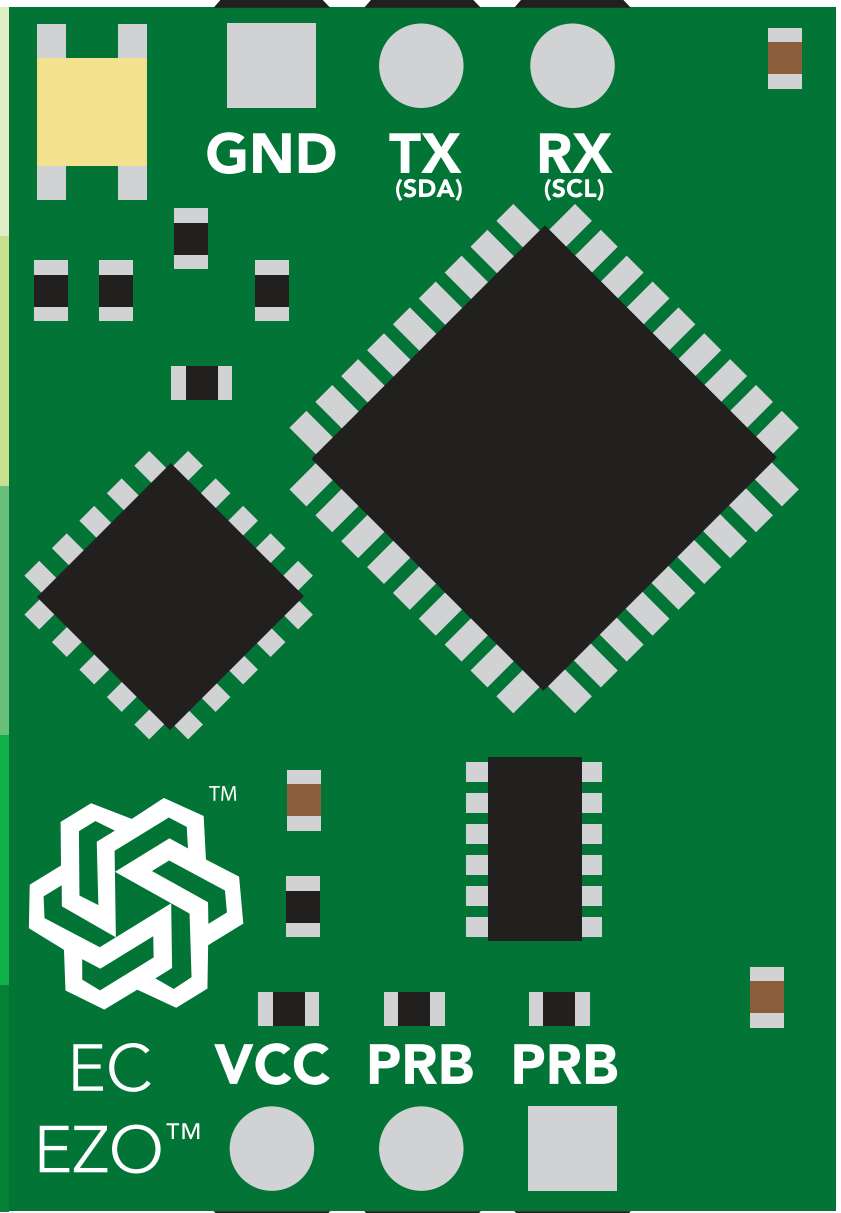
Resolution = **1.0 μ S/cm**

10,000 – 99,990

Resolution = **10 μ S/cm**

100,000 – 999,900

Resolution = **100 μ S/cm**



✓ Available data protocols

UART

Default

I²C

✗ Unavailable data protocols

SPI

Analog

RS-485

Mod Bus

4–20mA

STOP



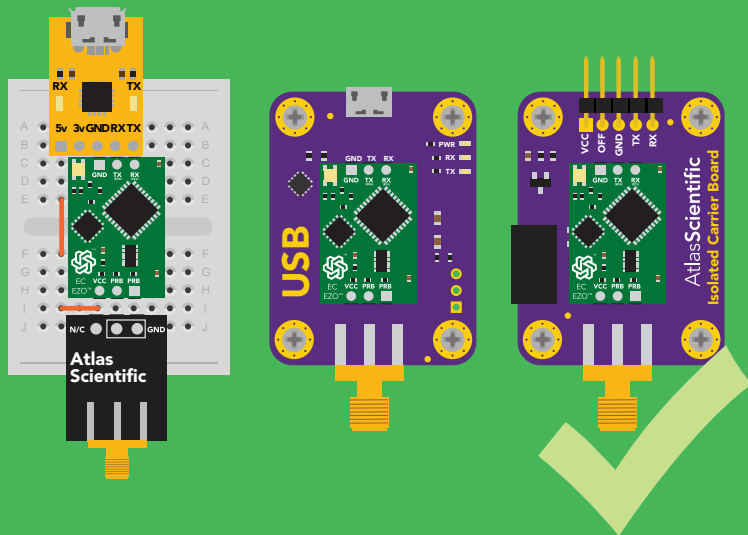
SOLDERING THIS DEVICE VOIDS YOUR WARRANTY.

Are there specific soldering instructions? Yes, see page 73.

Can you make a warranty claim after soldering? No.

If you have not used this product before; Observe how a properly working sensor behaves **BEFORE embedding it into your PCB.**

Get this device working using one of these methods first.



Do not embed before you have experience with this sensor.

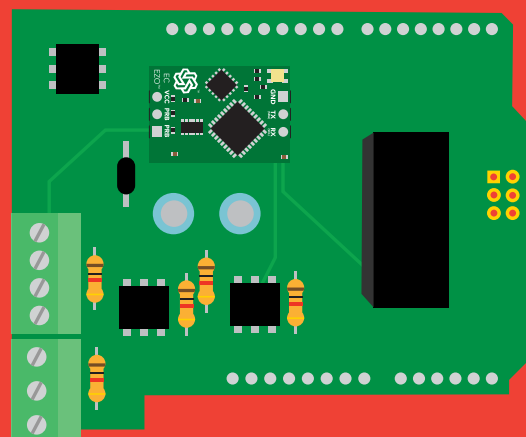


Table of contents

Available data protocols	5	Correct wiring	11
Circuit dimensions	8	Default state	12
Power consumption	8	Circuit footprint	75
Absolute max ratings	8	Datasheet change log	76
Electrical isolation	9	Warranty	81
Conductivity probe range	3		
Resolution	4		
Calibration theory	65		

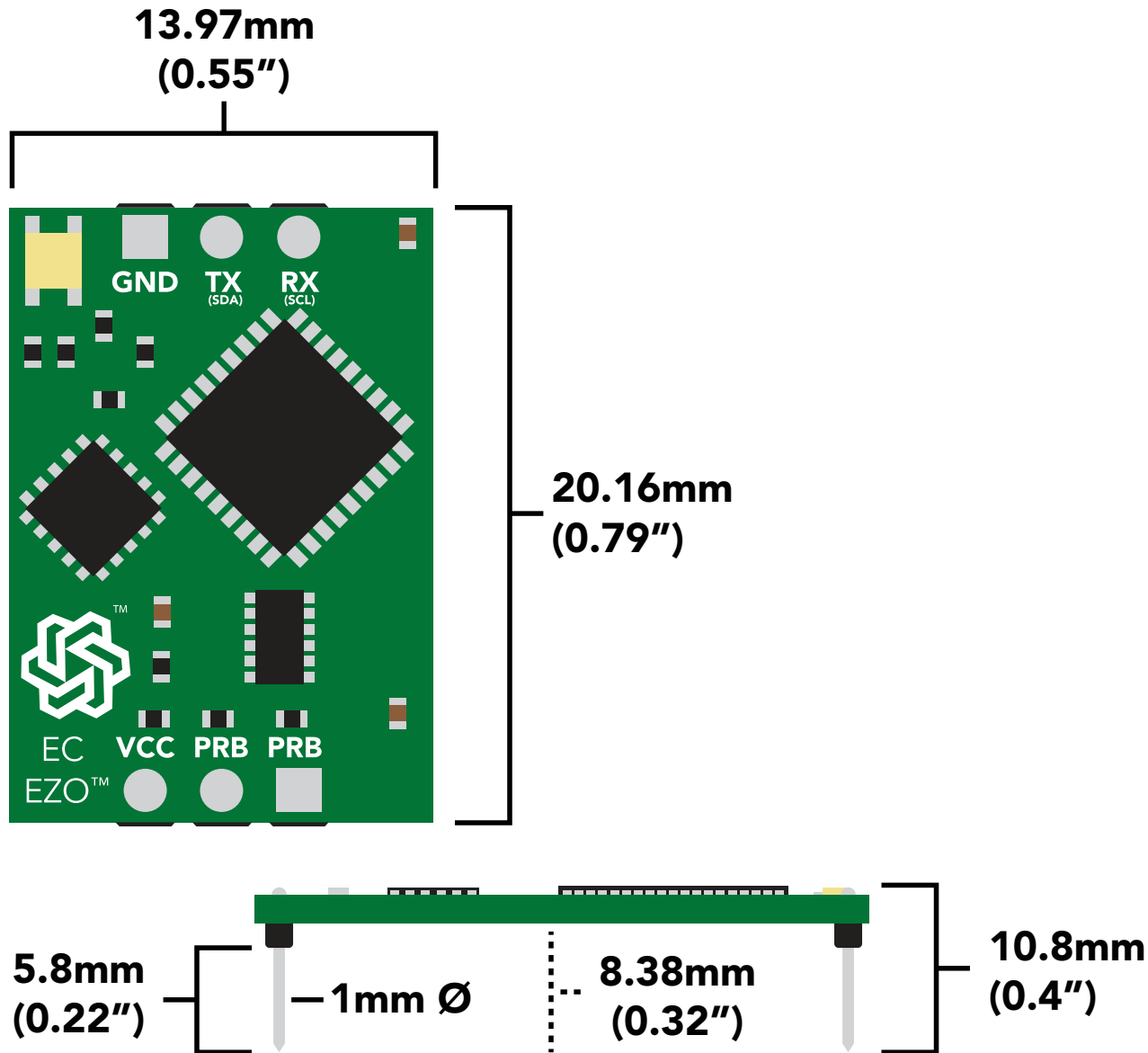
UART

UART mode	13
LED color definition	14
Receiving data from device	15
Sending commands to device	16
UART quick command page	17
LED control	18
Find	19
Continuous reading mode	20
Single reading mode	21
Calibration	22
Change TDS conversion factor	23
Export calibration	24
Import calibration	25
Setting the probe type	26
Temperature compensation	27
Enable/disable parameters	28
Naming device	29
Device information	30
Response codes	31
Reading device status	32
Sleep mode/low power	33
Change baud rate	34
Protocol lock	35
Factory reset	36
Change to I ² C mode	37
Manual switching to I ² C	38

I²C

I ² C mode	40
Sending commands	41
Requesting data	42
Response codes	43
LED color definition	44
I²C quick command page	45
LED control	46
Find	47
Taking reading	48
Calibration	49
Change TDS conversion factor	50
Export calibration	51
Import calibration	52
Setting the probe type	53
Temperature compensation	54
Enable/disable parameters	55
Naming device	56
Device information	57
Reading device status	58
Sleep mode/low power	59
Protocol lock	60
I ² C address change	61
Factory reset	62
Change to UART mode	63
Manual switching to UART	64

EZO™ circuit dimensions



Power consumption

	LED	MAX	STANDBY	SLEEP
5V	ON	50 mA	18.14 mA	0.7 mA
	OFF	45 mA	15.64 mA	
3.3V	ON	35 mA	16.85 mA	0.4 mA
	OFF	34 mA	15.85 mA	

Absolute max ratings

Parameter	MIN	TYP	MAX
Storage temperature (EZO™ Conductivity)	-60 °C		150 °C
Operational temperature (EZO™ Conductivity)	-40 °C	25 °C	125 °C
VCC	3.3V	5V	5.5V

Electrical isolation

Conductivity readings will introduce significant electrical interference into your water. This electrical interference will affect other sensors, such as pH, ORP, and dissolved oxygen. Electrical isolation is 100% effective in preventing this electrical interference.

Unlike other probes, a conductivity probe provides a low-resistance pathway from your water to your electronics. If an accidental electrical surge passes through your water, it will travel up your conductivity probe and into your electronics. Electrical isolation is 100% effective at stopping an accidental electrical surge from destroying your computer system.



Advice:

When reading conductivity along with other sensors, electrical isolation is strongly recommended. **Never build a commercial product without electrical isolation.**

Atlas Scientific offers several different electrical isolation products that can be used in your design. Select the electrical isolation product that works best for your design.



Basic EZO™
Inline Voltage Isolator



Vertical Isolator



Electrically Isolated
EZO™ Carrier Board



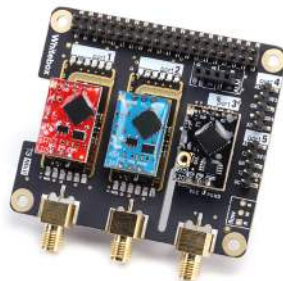
Gen 2 Electrically Isolated
USB EZO™ Carrier Board



Whitebox T1



Whitebox T3



Whitebox T3



Electrically Isolated EZO™
Carrier Board (old style)

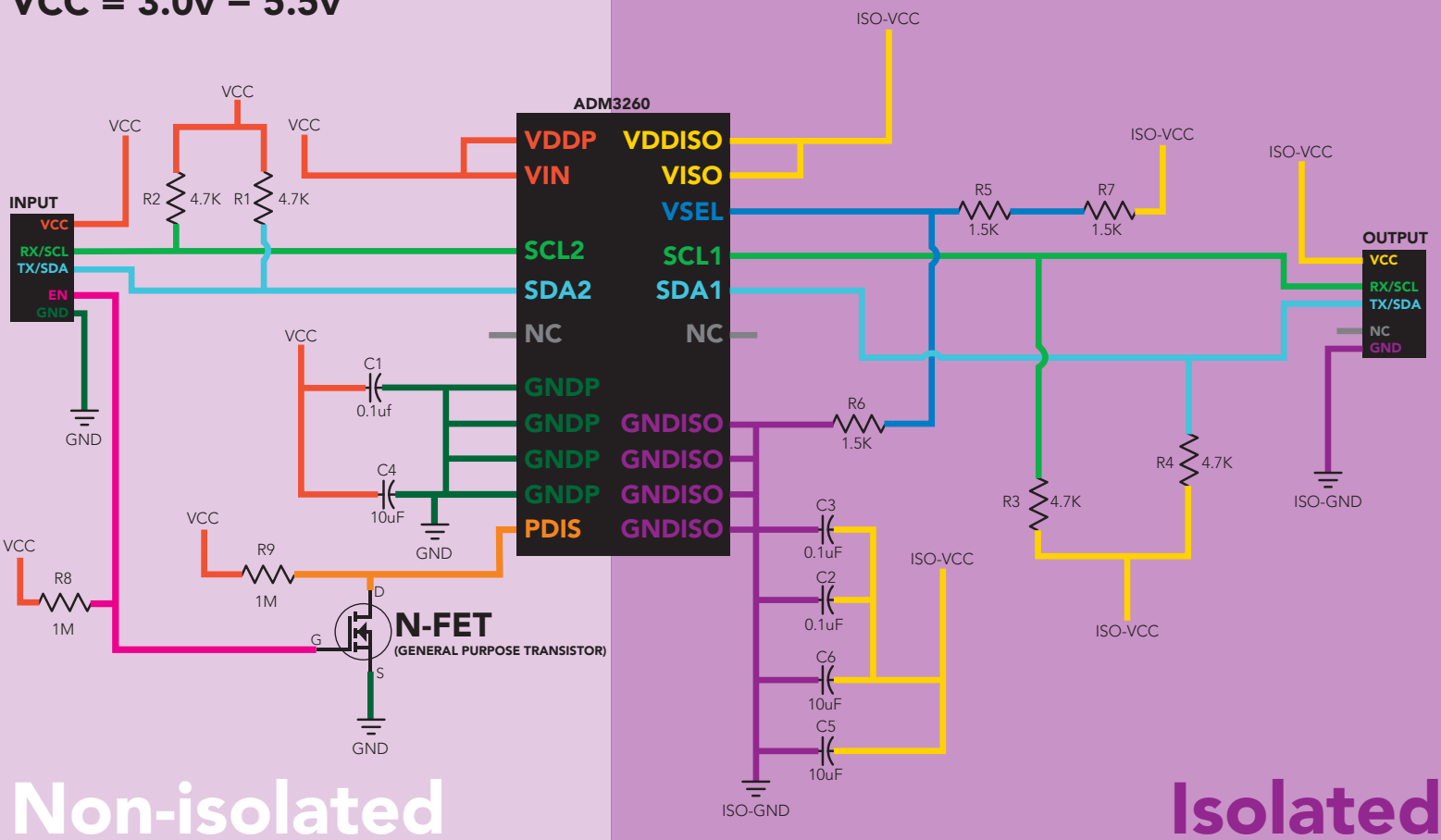
For various reasons, you may need to build your own electrical isolator. Because electrical isolation is so important, we have published our isolation schematic for anyone to use.

This isolation schematic is based on the ADM3260, which can output up to 150 mW of isolated power. PCB layout requires special attention for EMI/EMC and RF Control. Having good ground planes and keeping the capacitors as close to the chip as possible are crucial for proper performance.

The two data channels have a 4.7kΩ pull-up resistor on both the isolated and non-isolated lines (R1, R2, R3, and R4). The output voltage is set using a voltage divider (R5, R6, and R7). This produces a voltage of 3.9V regardless of your input voltage.

Isolated ground is different from non-isolated ground, these two lines should not be connected together.

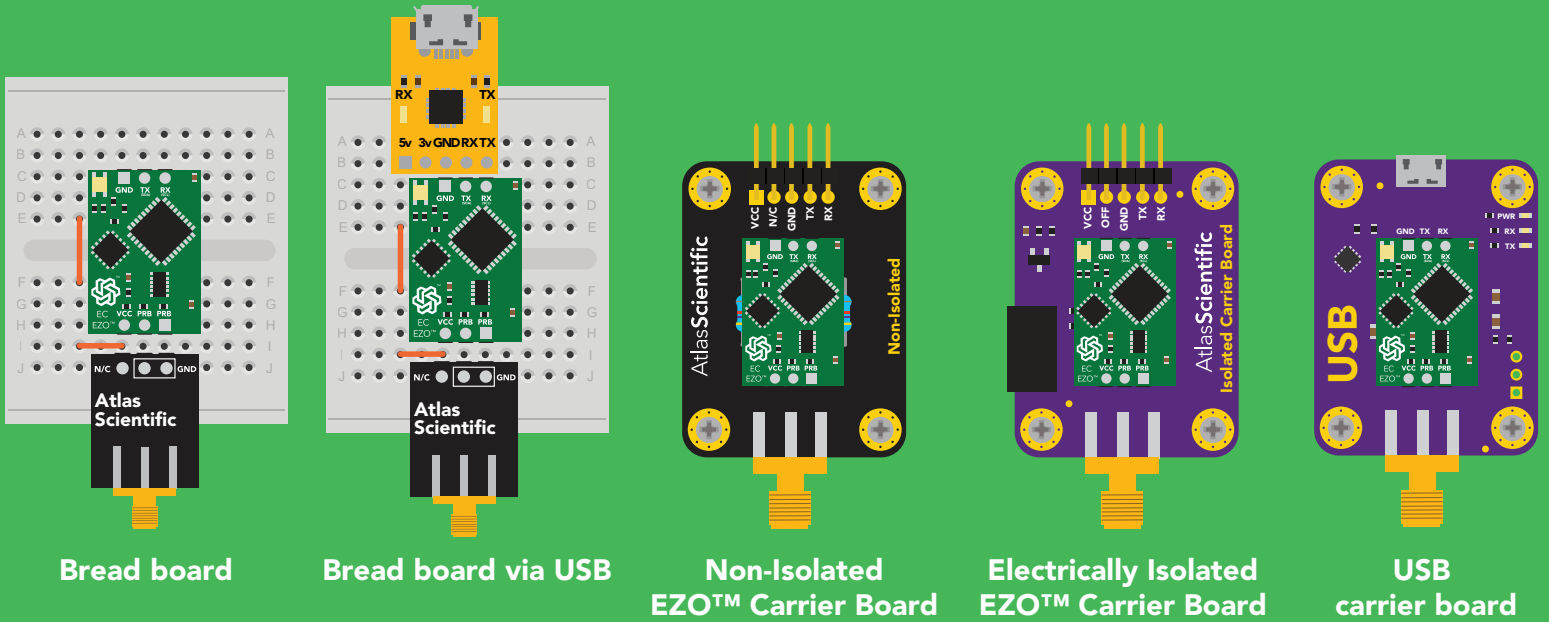
VCC = 3.0v – 5.5v



Non-isolated

Isolated

✓ Correct wiring



Bread board

Bread board via USB

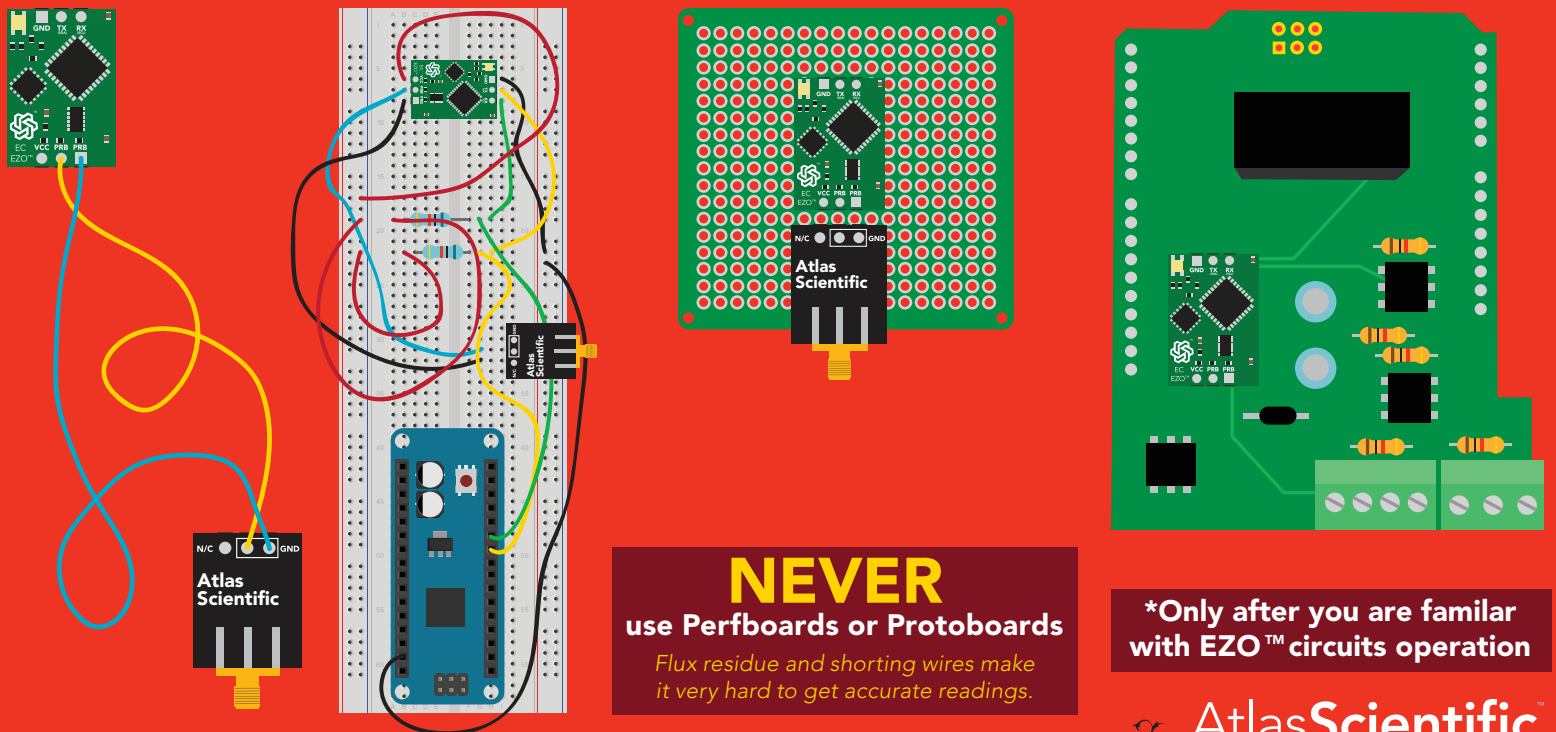
Non-Isolated
EZO™ Carrier Board

Electrically Isolated
EZO™ Carrier Board

USB
carrier board

X Incorrect wiring

Extended leads Sloppy setup Perfboards or Protoboards *Embedded into your device



Default state

UART mode

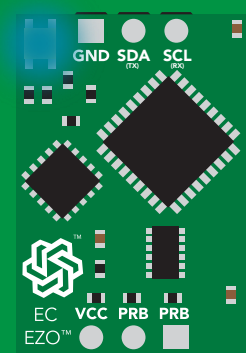
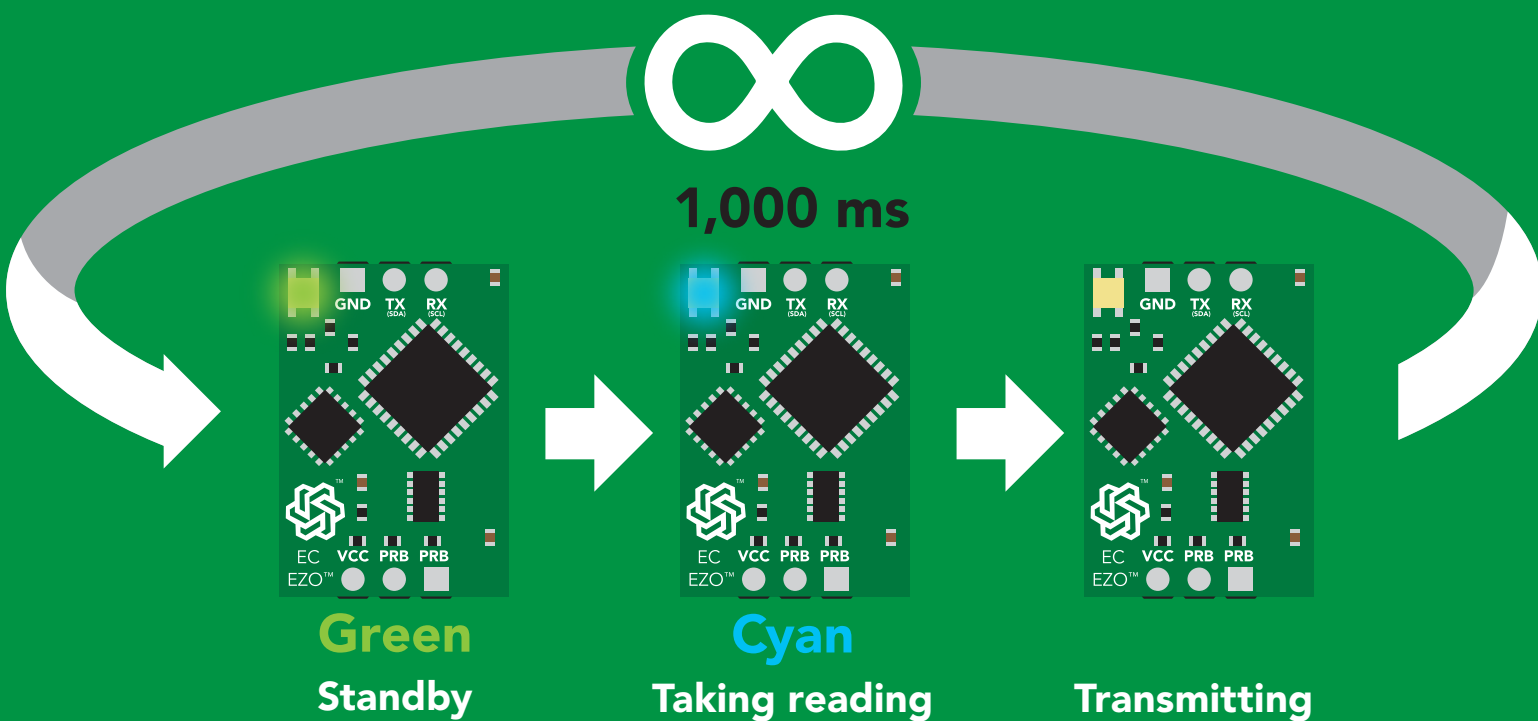
Baud 9,600

Readings continuous

Units $\mu\text{S}/\text{cm}$

Speed 1 reading per second

LED on



Solid Blue LED
in I²C mode
Not UART ready

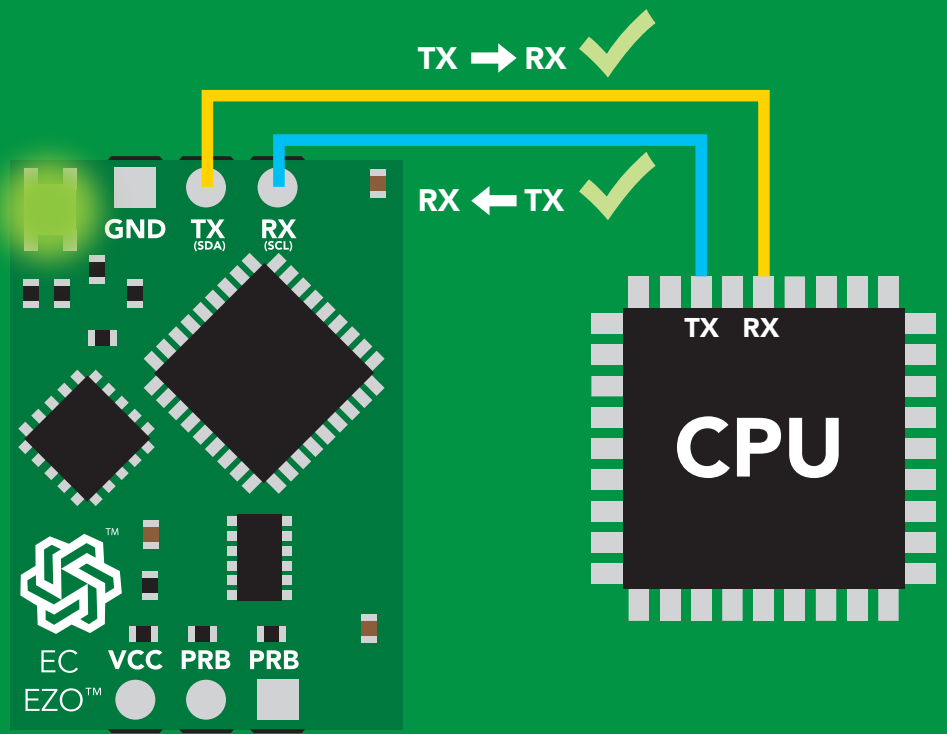
UART mode

8 data bits no parity
1 stop bit no flow control

Baud 300
1,200
2,400
9,600 default
19,200
38,400
57,600
115,200



Vcc 3.3V – 5.5V



Data format

Reading

Conductivity = **Default**

Total dissolved solids
Salinity
Specific gravity } = Must be enabled

Order EC, TDS, SAL, SG

Encoding ASCII

Format string

Terminator

Data type

Decimal places 3

Smallest string 3 characters

Largest string 40 characters

carriage return

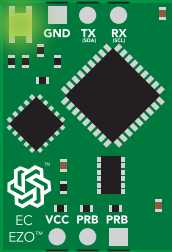
floating point

3

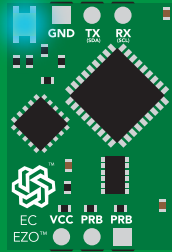
3 characters

40 characters

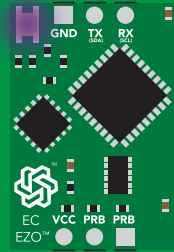
LED color definition



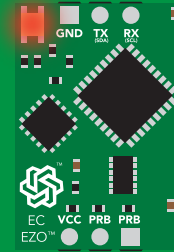
Green
UART standby



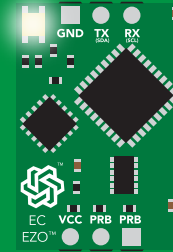
Cyan
Taking reading



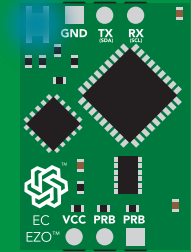
Purple
Changing baud rate



Red
Command not understood



White
Find



Blue
I2C standby

5V

LED ON
+2.5 mA

3.3V

+1 mA

Settings that are retained if power is cut

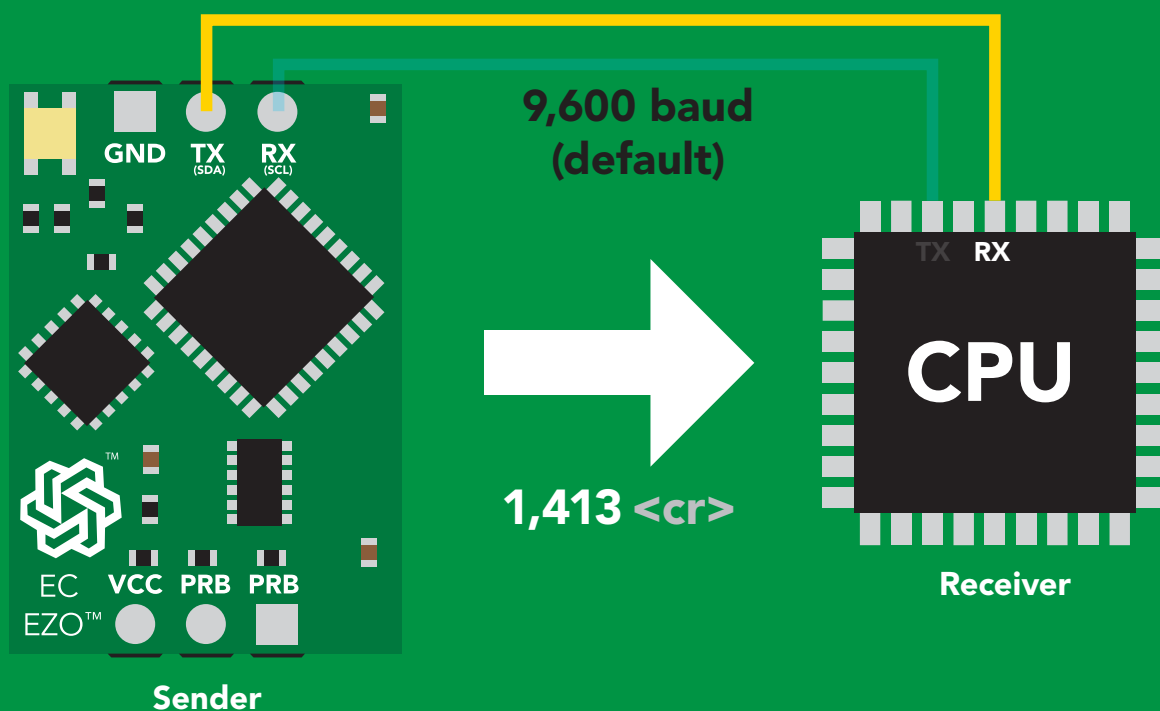
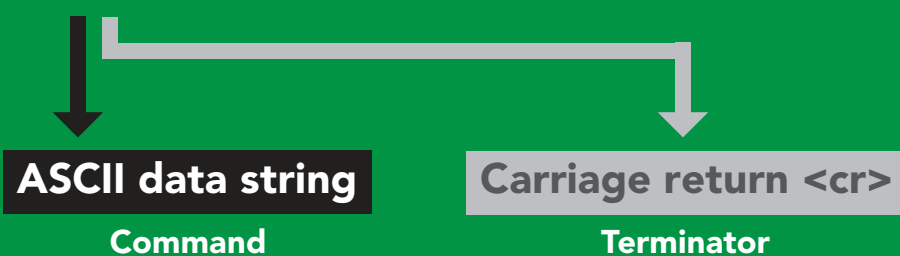
- Baud rate
- Calibration
- Continuous mode
- Device name
- Enable/disable parameters
- Enable/disable response codes
- Hardware switch to I²C mode
- LED control
- Protocol lock
- Software switch to I²C mode

Settings that are **NOT** retained if power is cut

- Find
- Sleep mode
- Temperature compensation

Receiving data from device

2 parts



Advanced

ASCII: 1 , 4 1 3 <cr>

Hex: 31 2C 34 31 33 0D

Dec: 49 44 52 49 51 13

Sending commands to device

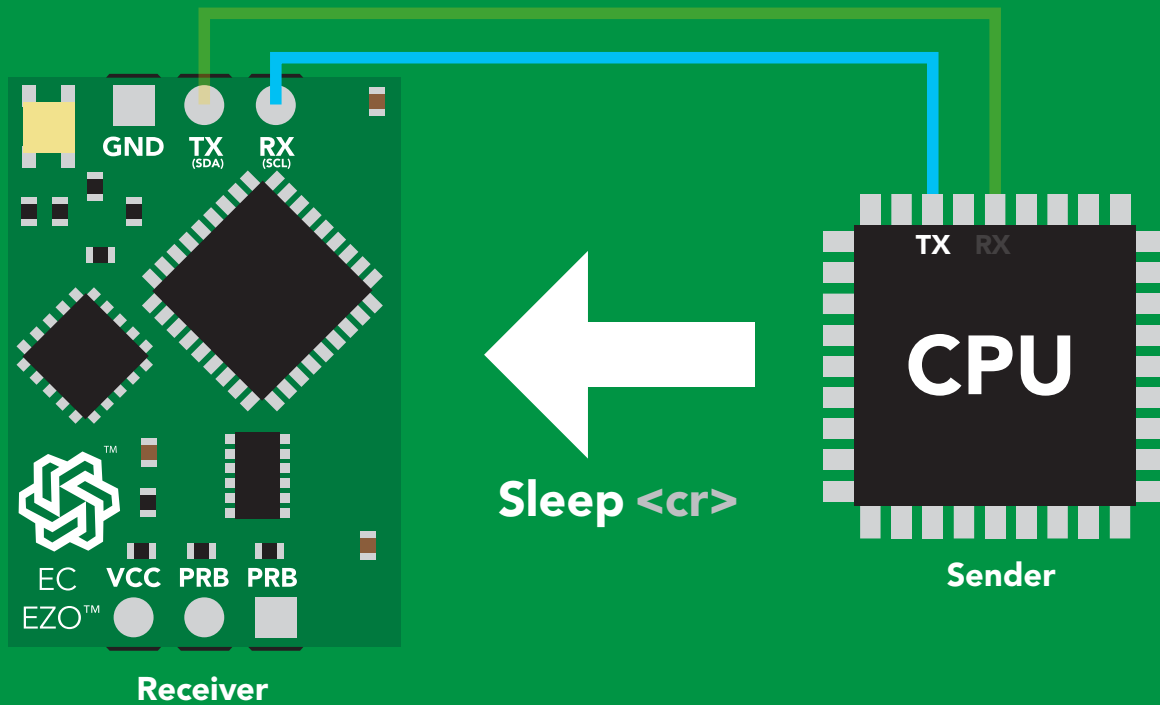
2 parts

Command (not case sensitive)

ASCII data string

Carriage return <cr>

Terminator



Advanced

ASCII: **S** **I** **e** **e** **p** **<cr>**

Hex: **53** **6C** **65** **65** **70** **0D**

Dec: **83** **108** **101** **101** **112** **13**

UART mode

command quick reference

All commands are ASCII strings or single ASCII characters.

Command	Function		Default state
Baud	change baud rate	pg. 34	9,600
C	enable/disable continuous reading	pg. 20	enabled
Cal	performs calibration	pg. 22	n/a
Export	export calibration	pg. 24	n/a
Factory	enable factory reset	pg. 36	n/a
Find	finds device with blinking white LED	pg. 19	n/a
i	device information	pg. 30	n/a
I2C	change to I ² C mode	pg. 37	not set
Import	import calibration	pg. 25	n/a
K	Set probe type	pg. 26	K 1.0
L	enable/disable LED	pg. 18	enabled
Name	set/show name of device	pg. 29	not set
O	enable/disable parameters	pg. 28	all enabled
Plock	enable/disable protocol lock	pg. 35	disabled
R	returns a single reading	pg. 21	n/a
Sleep	enter sleep mode/low power	pg. 33	n/a
Status	retrieve status information	pg. 32	enable
T	temperature compensation	pg. 27	25°C
TDS	change the TDS conversion factor	pg. 23	0.54
*OK	enable/disable response codes	pg. 31	enable

LED control

Command syntax

L,1 <cr> LED on **default**

L,0 <cr> LED off

L,? <cr> LED state on/off?

Example

Response

L,1 <cr>

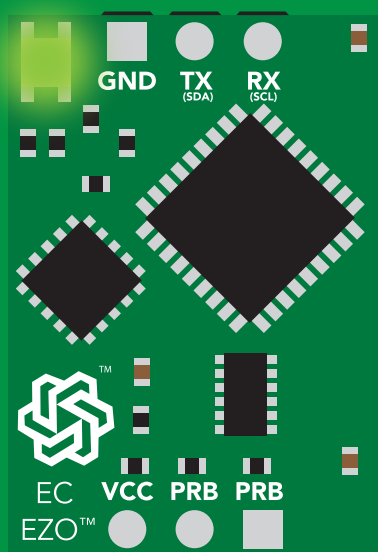
*OK <cr>

L,0 <cr>

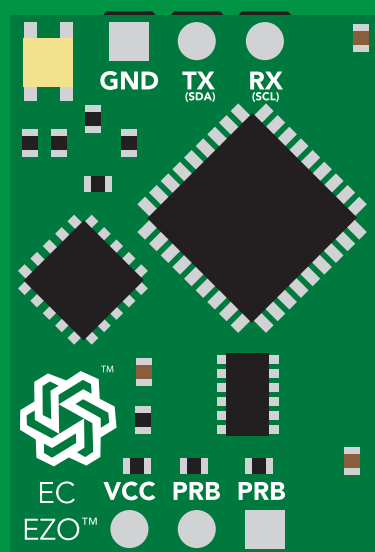
*OK <cr>

L,? <cr>

?L,1 <cr> or ?L,0 <cr>
*OK <cr>



L,1



L,0

Find

Command syntax

This command will disable continuous mode
Send any character or command to terminate find.

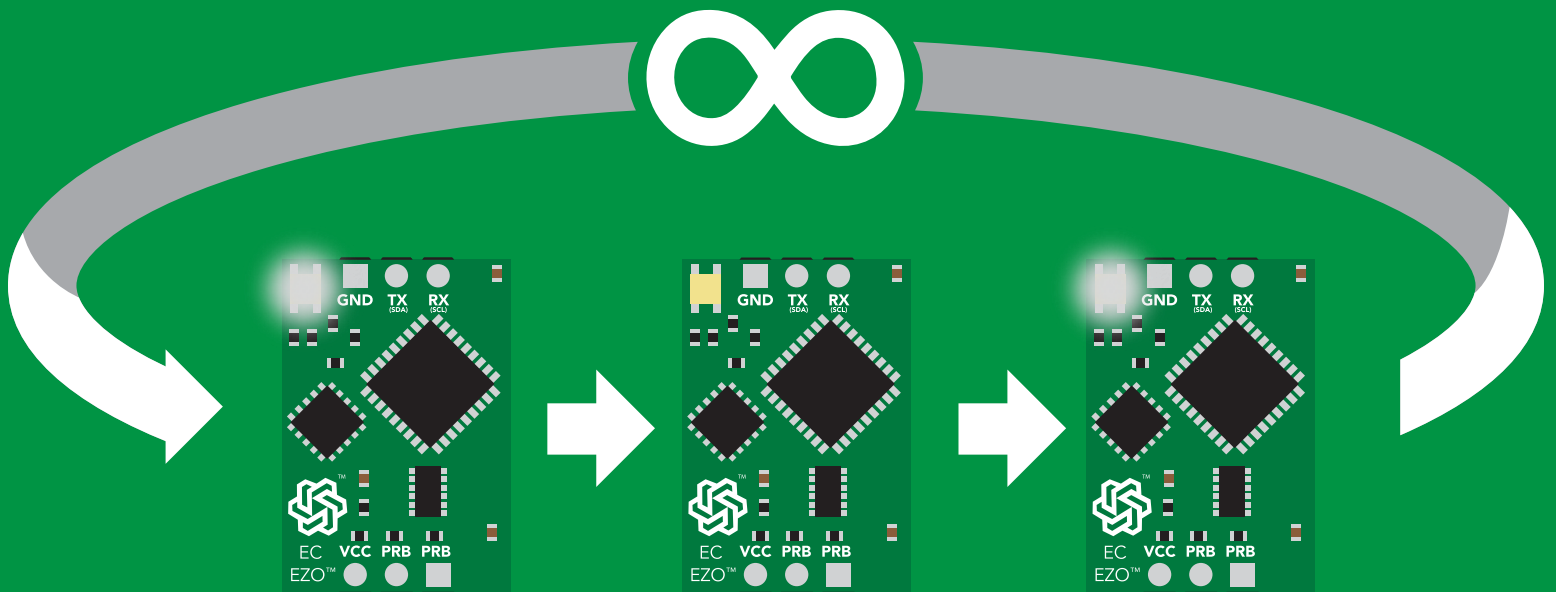
Find <cr> LED rapidly blinks white, used to help find device

Example

Response

Find <cr>

*OK <cr>



Continuous reading mode

Command syntax

- C,1 <cr>** enable continuous readings once per second **default**
- C,n <cr>** continuous readings every n seconds (n = 2 to 99 sec)
- C,0 <cr>** disable continuous readings
- C,? <cr>** continuous reading mode on/off?

Example

Response

C,1 <cr>

***OK <cr>**
EC,TDS,SAL,SG (1 sec) <cr>
EC,TDS,SAL,SG (2 sec) <cr>
EC,TDS,SAL,SG (3 sec) <cr>

C,30 <cr>

***OK <cr>**
EC,TDS,SAL,SG (30 sec) <cr>
EC,TDS,SAL,SG (60 sec) <cr>
EC,TDS,SAL,SG (90 sec) <cr>

C,0 <cr>

***OK <cr>**

C,? <cr>

?C,1 <cr> or ?C,0 <cr> or ?C,30 <cr>
***OK <cr>**

Single reading mode

Command syntax

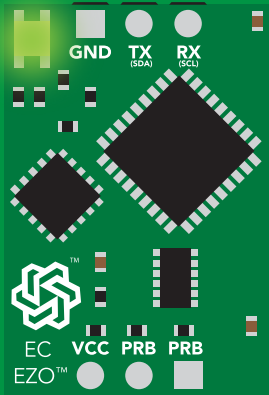
R <cr> takes single reading

Example

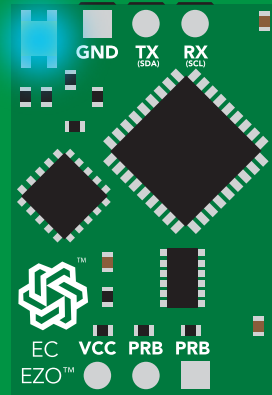
R <cr>

Response

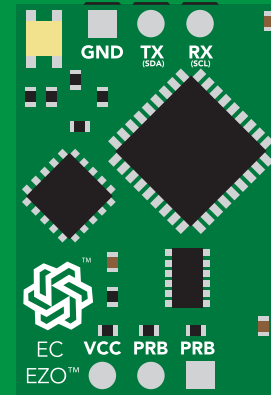
1,413 <cr>
*OK <cr>



Green
Standby



Cyan
Taking reading



Transmitting



600 ms

Calibration

Command syntax

Dry calibration must always be done first!

Cal,dry	<cr>	dry calibration
Cal,n	<cr>	single point calibration, where n = any value
Cal,low,n	<cr>	low end calibration, where n = any value
Cal,high,n	<cr>	high end calibration, where n = any value
Cal,clear	<cr>	delete calibration data
Cal,?	<cr>	device calibrated?

Example

Response

Cal,dry <cr>

*OK <cr>

Cal,84 <cr>

*OK <cr>

Cal,low,12880 <cr>

*OK <cr>

Cal,high,80000 <cr>

*OK <cr>

Cal,clear <cr>

*OK <cr>

Cal,? <cr>

?CAL,0 <cr> or ?CAL,1 <cr> or ?CAL,2
one point two point
*OK <cr>

One point calibration:

Step 1. "cal,dry"

Step 2. "cal,n"

Calibration complete!

Two point calibration:

Step 1 "cal,dry"

Step 2 "cal,low,n"

Step 3 "cal,high.n"

Calibration complete!

Changing the TDS (ppm) conversion factor

Command syntax

There are several different conversion factors used to read TDS(ppm). For some applications, it may be necessary to use a conversion factor other than the default value of 0.54

TDS,n <cr> set custom conversion factor, n = any value between 0.01 – 1.00
TDS,? <cr> conversion factor being used

Example

Response

TDS,? <cr>

?TDS,0.54 <cr>
*OK <cr>

R <cr>

EC **TDS**
↓ ↓
100,54 <cr>
*OK <cr>

TDS,0.46 <cr>

*OK <cr>

R <cr>

EC **TDS**
↓ ↓
100,46 <cr>
*OK <cr>

Common conversion factors

NaCl 0.47 – 0.50
KCL 0.50 - 0.57
"442" 0.65 – 0.85

Formula

EC x conversion factor = TDS

Export calibration

Command syntax

Export: Use this command to download calibration settings

Export,? <cr> calibration string info

Export <cr> export calibration string from calibrated device

Example

Response

Export,? <cr>

10,120 <cr>

Response breakdown

10, 120

of strings to export

of bytes to export

Export strings can be up to 12 characters long, and is always followed by <cr>

Export <cr>

59 6F 75 20 61 72 <cr> (1 of 10)

Export <cr>

65 20 61 20 63 6F <cr> (2 of 10)

(7 more)

⋮

Export <cr>

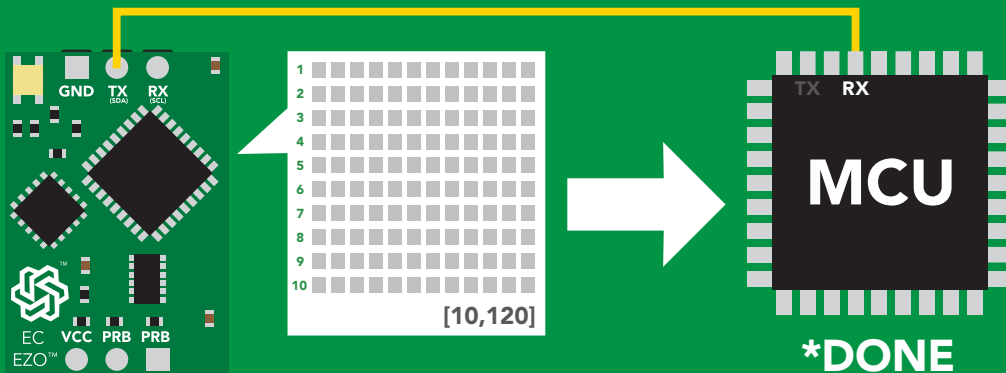
6F 6C 20 67 75 79 <cr> (10 of 10)

Export <cr>

***DONE**

Disabling *OK simplifies this process

Export <cr>



Import calibration

Command syntax

Import: Use this command to upload calibration settings to one or more devices.

Import,n <cr> import calibration string to new device

Example

Import, 59 6F 75 20 61 72 <cr> (1 of 10)

Import, 65 20 61 20 63 6F <cr> (2 of 10)

⋮

Import, 6F 6C 20 67 75 79 <cr> (10 of 10)

Response

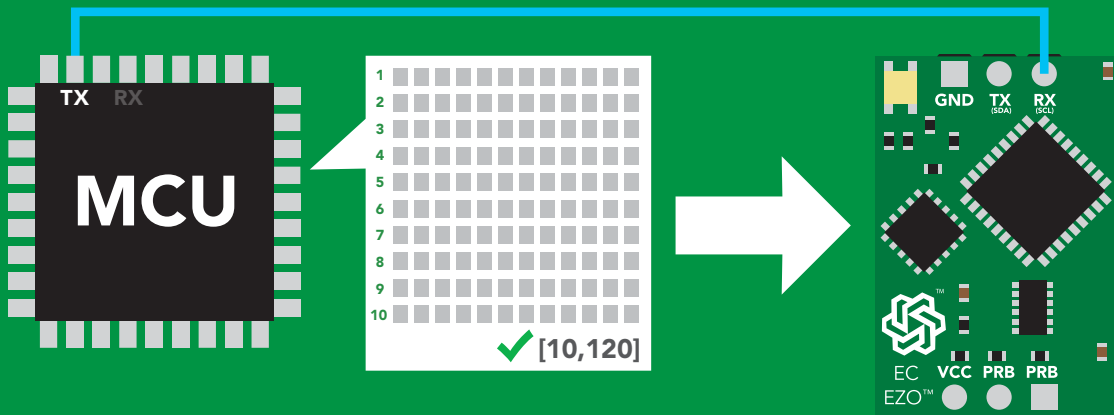
*OK <cr>

*OK <cr>

⋮

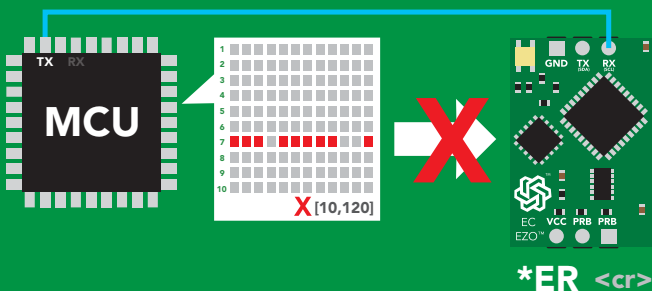
*OK <cr>

Import,n <cr>



*OK <cr>

system will reboot



*ER <cr>

* If one of the imported strings is not correctly entered, the device will not accept the import, respond with *ER and reboot.

Setting the probe type

Command syntax

K 1.0 is the default value

K,n <cr> n = any value; floating point in ASCII

K,? <cr> probe K value?

Example

Response

K,10 <cr>

***OK** <cr>

K,? <cr>

?K,10 <cr>

***OK** <cr>



K 0.1



K 1.0



K 10

Temperature compensation

Command syntax

Default temperature = 25°C
Temperature is always in Celsius
Temperature is not retained if power is cut

T,n <cr> n = any value; floating point or int

T,? <cr> compensated temperature value?

RT,n <cr> set temperature compensation and take a reading

Example

Response

T,19.5 <cr>

***OK** <cr>

RT,19.5 <cr>

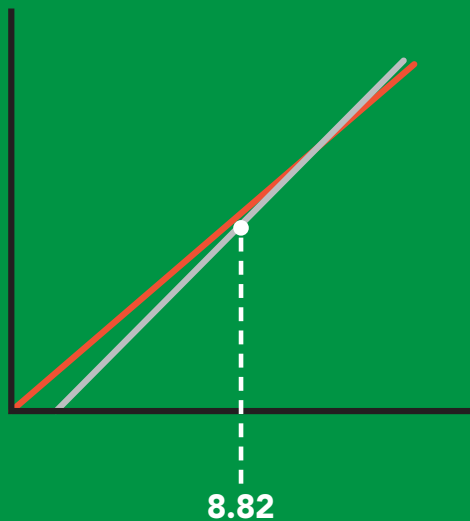
***OK** <cr>

8.91 <cr>

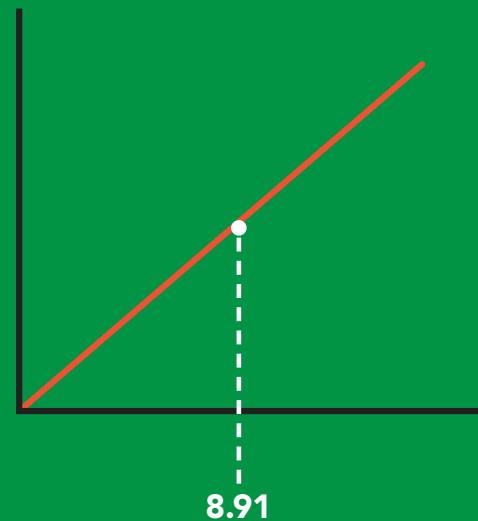
T,? <cr>

?T,19.5 <cr>

***OK** <cr>



→
T,19.5 <cr>



Enable/disable parameters from output string

Command syntax

O, [parameter],[1,0] <cr> enable or disable output parameter
O,? <cr> enabled parameter?

Example

O,EC,1 / O,EC,0 <cr>

Response

*OK <cr> enable / disable conductivity

O,TDS,1 / O,TDS,0 <cr>

*OK <cr> enable / disable total dissolved solids

O,S,1 / O,S,0 <cr>

*OK <cr> enable / disable salinity

O,SG,1 / O,SG,0 <cr>

*OK <cr> enable / disable specific gravity

O,? <cr>

?,O,EC,TDS,S,SG <cr> if all are enabled

Parameters

EC Conductivity = $\mu\text{S}/\text{cm}$
TDS Total dissolved solids = ppm
S Salinity = PSU (ppt) 0.00 – 42.00
SG Specific gravity (sea water only) = 1.00 – 1.300

* If you disable all possible data types your readings will display "no output".

Followed by 1 or 0

1 enabled
0 disabled

Naming device

Command syntax

Do not use spaces in the name

Name,n <cr> set name

Name, <cr> clears name

Name,? <cr> show name

n =

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Up to 16 ASCII characters

Example

Response

Name, <cr>

*OK <cr> name has been cleared

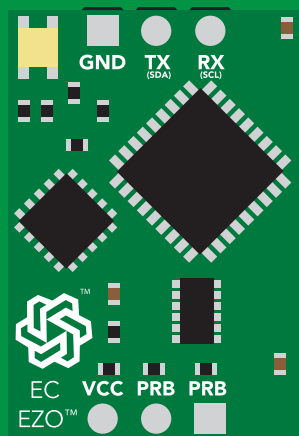
Name,zzt <cr>

*OK <cr>

Name,? <cr>

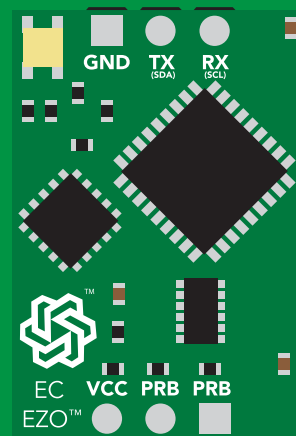
?Name,zzt <cr>
*OK <cr>

Name,zzt



*OK <cr>

Name,?



?Name,zzt <cr>
*OK <cr>

Device information

Command syntax

```
i <cr> device information
```

Example

```
i <cr>
```

Response

```
?i,EC,2.16 <cr>  
*OK <cr>
```

Response breakdown

?i,	EC,	2.16
	↑	↑
	Device	Firmware

Response codes

Command syntax

- *OK,1** <cr> enable response **default**
- *OK,0** <cr> disable response
- *OK,?** <cr> response on/off?

Example

Response

R <cr>

1,413 <cr>
***OK** <cr>

***OK,0** <cr>

no response, ***OK** disabled

R <cr>

1,413 <cr> ***OK** disabled

***OK,?** <cr>

?*OK,1 <cr> or **?*OK,0** <cr>

Other response codes

- *ER** unknown command
- *OV** over volt ($VCC \geq 5.5V$)
- *UV** under volt ($VCC \leq 3.1V$)
- *RS** reset
- *RE** boot up complete, ready
- *SL** entering sleep mode
- *WA** wake up

These response codes cannot be disabled

Reading device status

Command syntax

Status <cr> voltage at Vcc pin and reason for last restart

Example

```
Status <cr>
```

Response

```
?Status,P,5.038 <cr>  
*OK <cr>
```

Response breakdown

?Status,	P,	5.038
	↑	↑
	Reason for restart	Voltage at Vcc

Restart codes

P	powered off
S	software reset
B	brown out
W	watchdog
U	unknown

Sleep mode/low power

Command syntax

Send any character or command to awaken device.

Sleep <cr> enter sleep mode/low power

Example

Response

Sleep <cr>

*OK <cr>

*SL <cr>

Any command

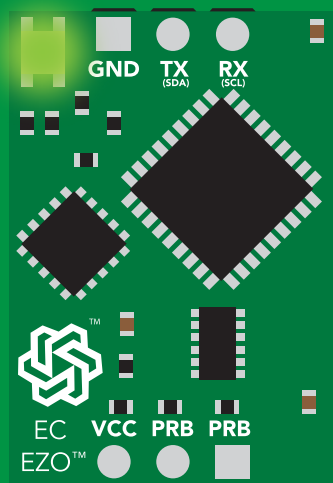
*WA <cr> wakes up device

5V

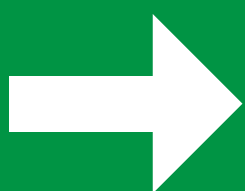
STANDBY	SLEEP
18.14 mA	0.7 mA

3.3V

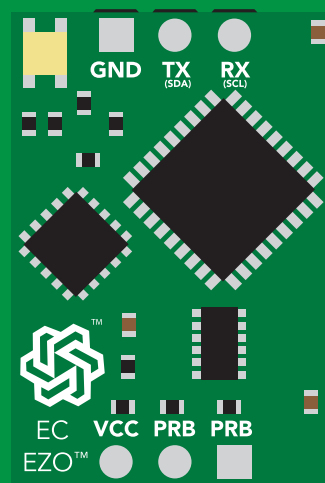
16.85 mA	0.4 mA
----------	--------



Standby
18.14 mA



Sleep <cr>



Sleep
0.7 mA

Change baud rate

Command syntax

Baud,n <cr> change baud rate

Example

Baud,38400 <cr>

Response

*OK <cr>

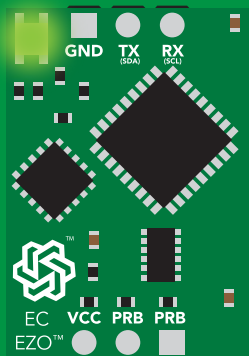
Baud,? <cr>

?Baud,38400 <cr>

*OK <cr>

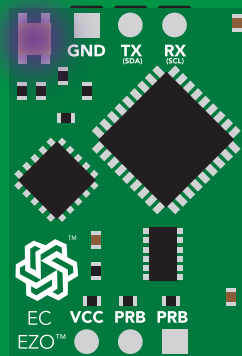
n =

- 300
- 1200
- 2400
- 9600 default**
- 19200
- 38400
- 57600
- 115200



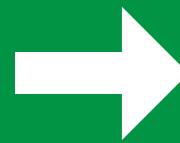
Standby

Baud,38400 <cr>

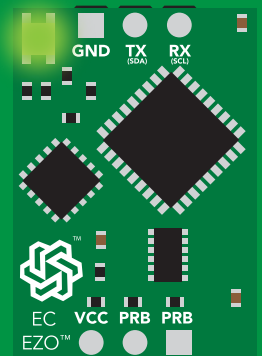


Changing
baud rate

*OK <cr>



(reboot)



Standby

Protocol lock

Command syntax

Locks device to UART mode.

Plock,1 <cr> enable Plock

Plock,0 <cr> disable Plock **default**

Plock,? <cr> Plock on/off?

Example

Response

Plock,1 <cr>

*OK <cr>

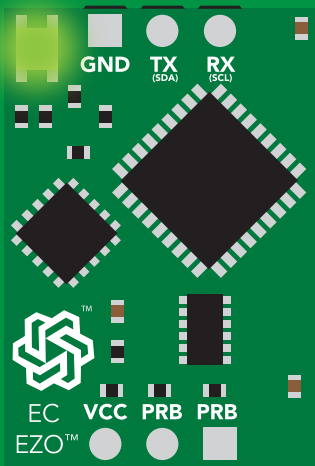
Plock,0 <cr>

*OK <cr>

Plock,? <cr>

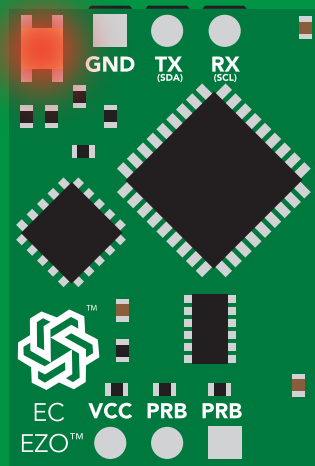
?Plock,1 <cr> or ?Plock,0 <cr>

Plock,1



*OK <cr>

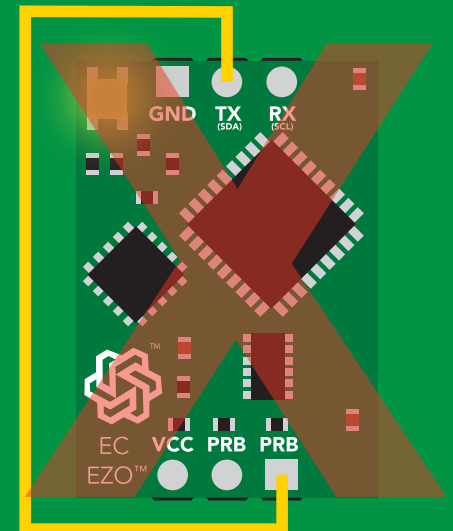
I2C,100



cannot change to I²C

*ER <cr>

Short



cannot change to I²C

Factory reset

Command syntax

Clears calibration
LED on
"*OK" enabled

Factory <cr> enable factory reset

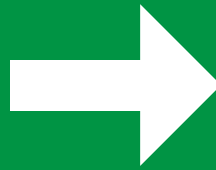
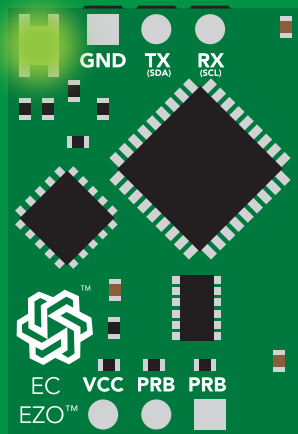
Example

Response

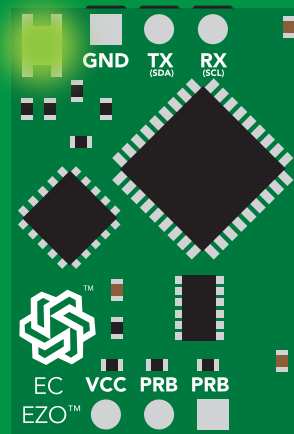
Factory <cr>

*OK <cr>

Factory <cr>



(reboot)



*OK <cr>

*RS <cr>

*RE <cr>

Baud rate will not change

Change to I²C mode

Command syntax

Default I²C address 100 (0x64)

I2C,n <cr> sets I²C address and reboots into I²C mode

n = any number 1 – 127

Example

Response

I2C,100 <cr>

*OK (reboot in I²C mode)

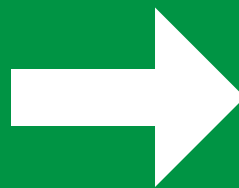
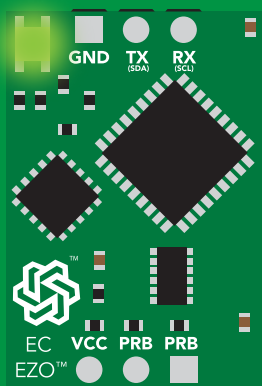
Wrong example

Response

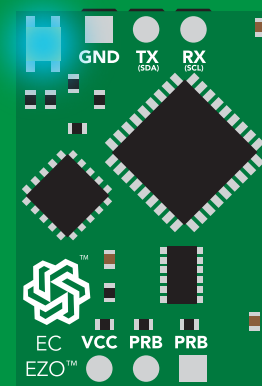
I2C,139 <cr> n ≠ 127

*ER <cr>

I2C,100



(reboot)



Green
*OK <cr>

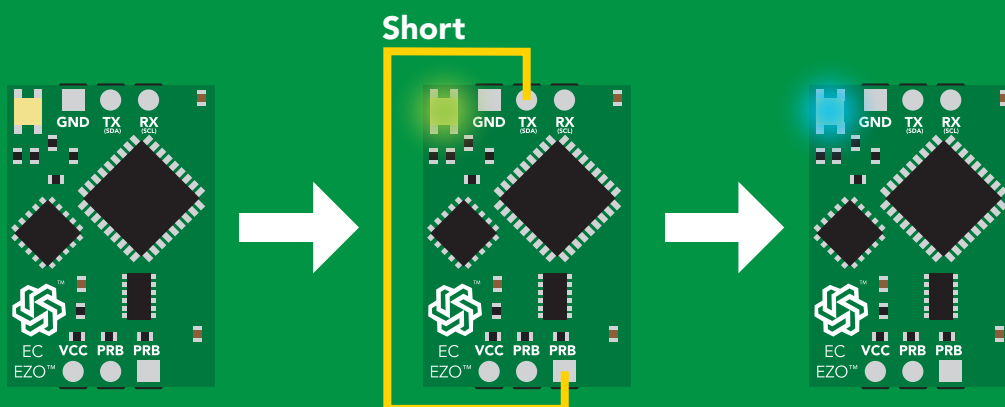
Blue
now in I²C mode

Manual switching to I²C

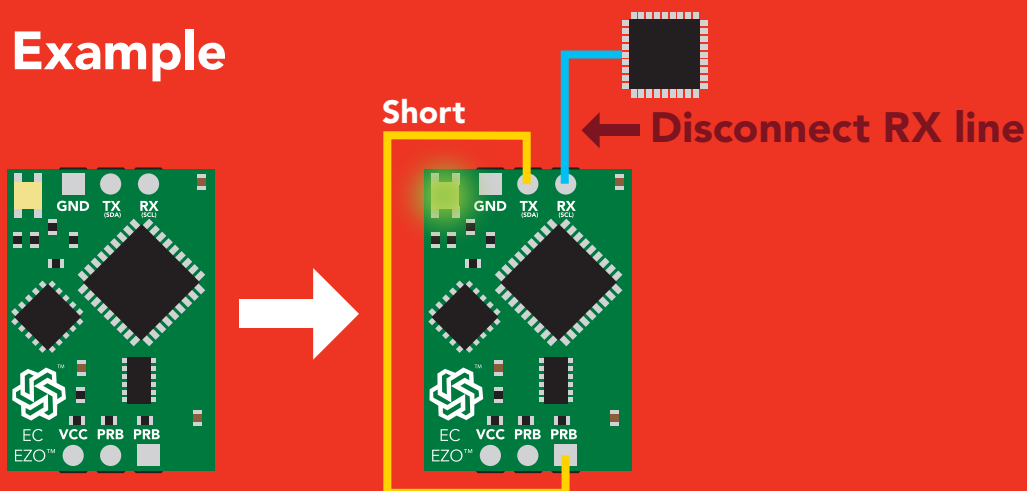
- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to the right PRB
- Confirm RX is disconnected
- Connect ground (power on)
- Wait for LED to change from **Green** to **Blue**
- Disconnect ground (power off)
- Reconnect all data and power

Manually switching to I²C will set the I²C address to 100 (0x64)

Example



Wrong Example



I²C mode

The I²C protocol is **considerably more complex** than the UART (RS-232) protocol. Atlas Scientific assumes the embedded systems engineer understands this protocol.

To set your EZO™ device into I²C mode click [here](#)

Settings that are retained if power is cut

- Calibration
- Change I²C address
- Enable/disable parameters
- Hardware switch to UART mode
- LED control
- Protocol lock
- Software switch to UART mode

Settings that are **NOT** retained if power is cut

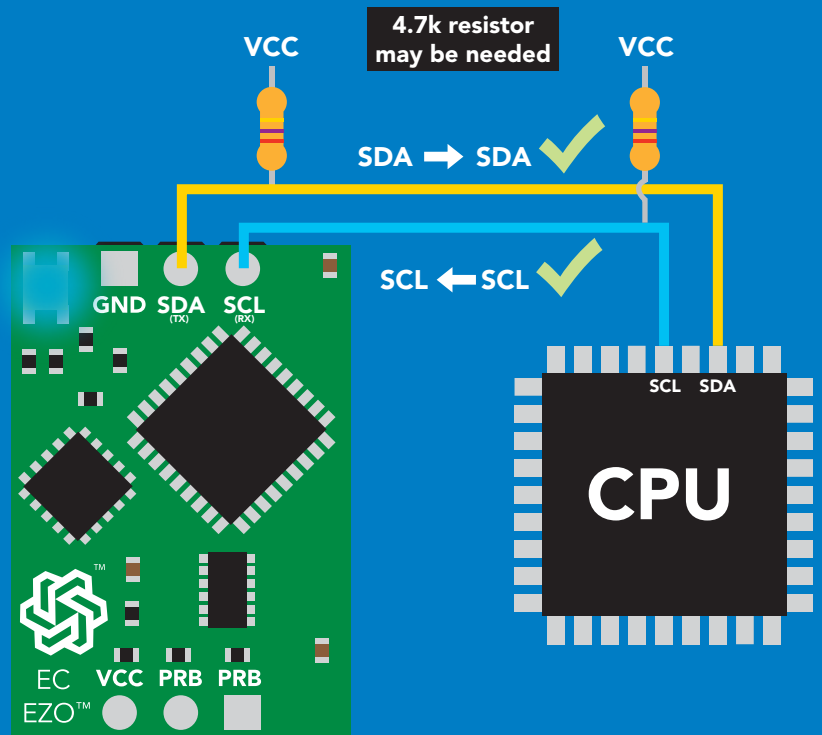
- Find
- Sleep mode
- Temperature compensation

I²C mode

I²C address (0x01 – 0x7F)
100 (0x64) default

V_{CC} 3.3V – 5.5V

Clock speed 100 – 400 kHz



Data format

Reading

Conductivity = **Default**

Total dissolved solids
Salinity
Specific gravity } = Must be enabled

Order **EC, TDS, SAL, SG**

Encoding **ASCII**

Format

Data type

Decimal places

Smallest string

Largest string

string

floating point

3

3 characters

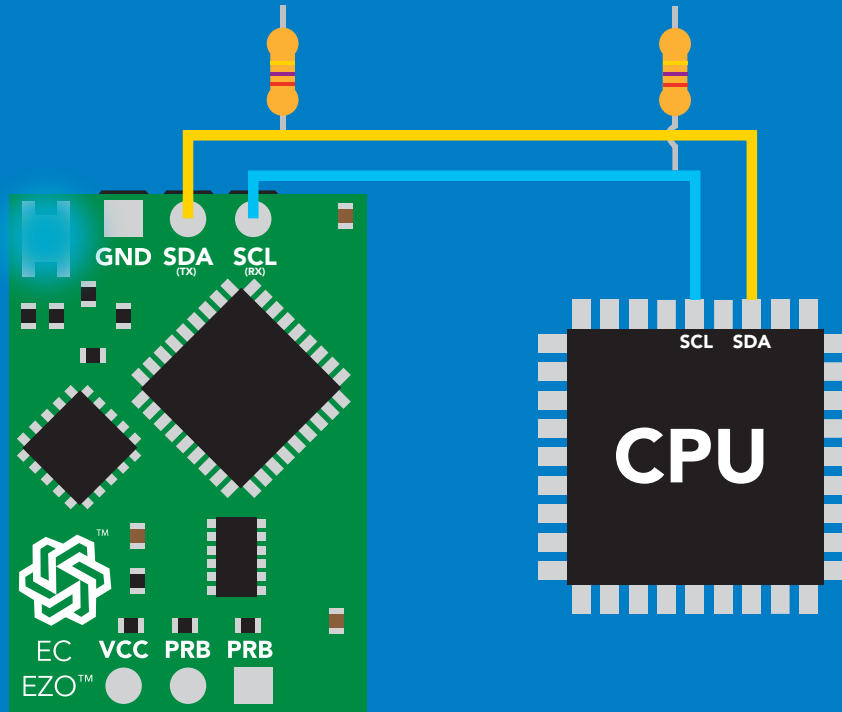
40 characters

Sending commands to device

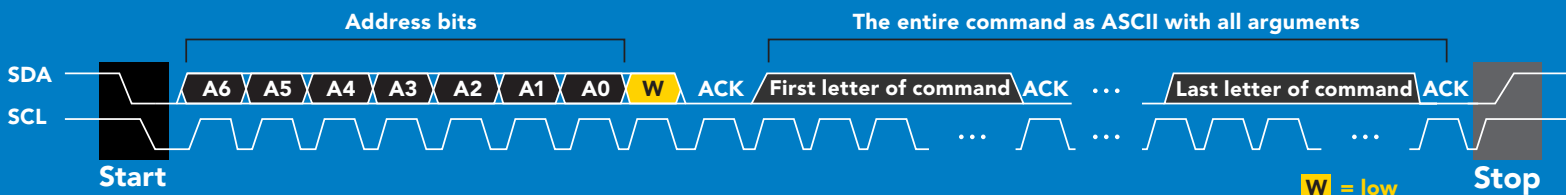
5 parts



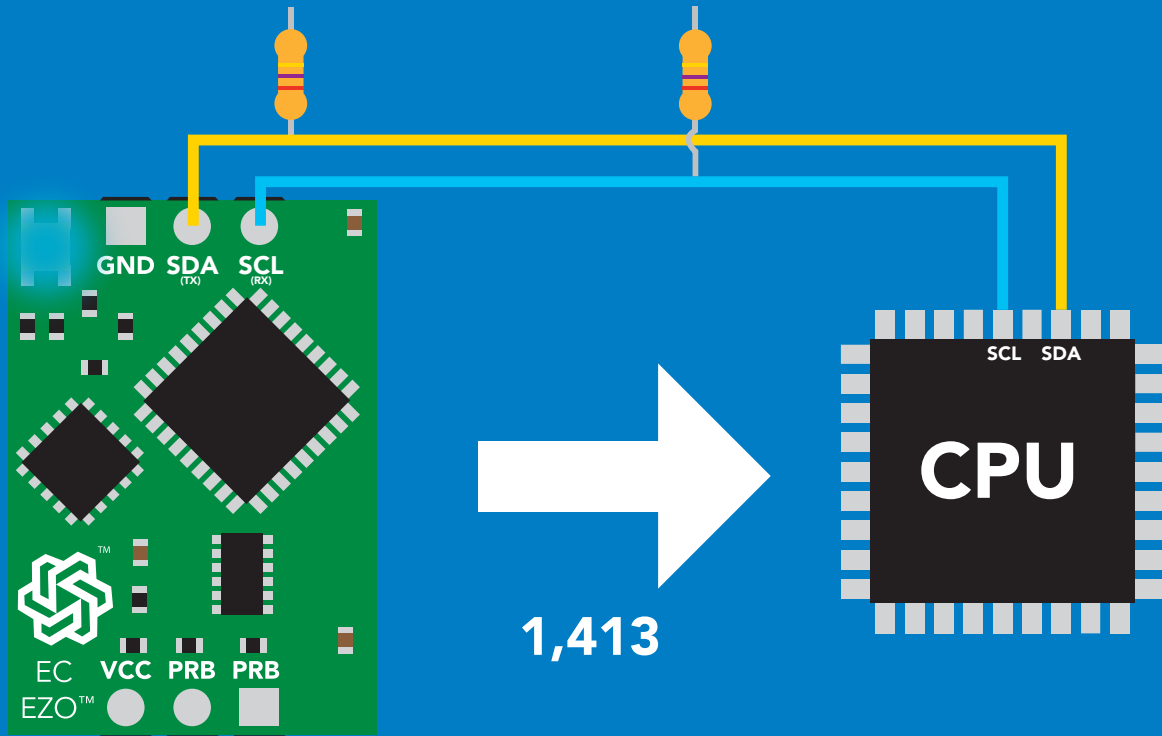
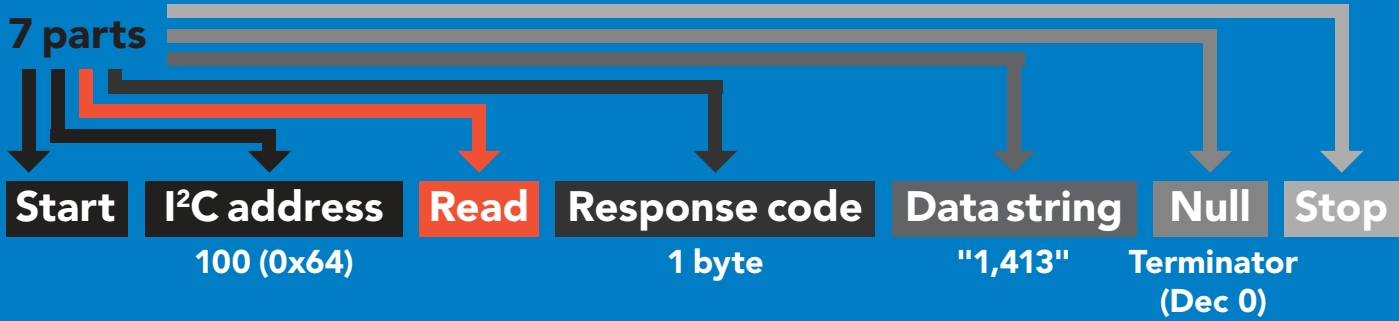
Example



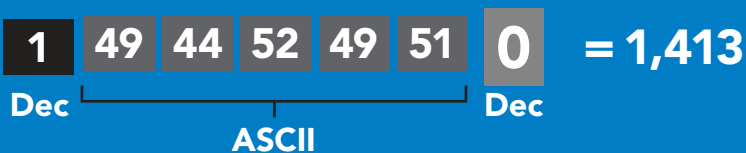
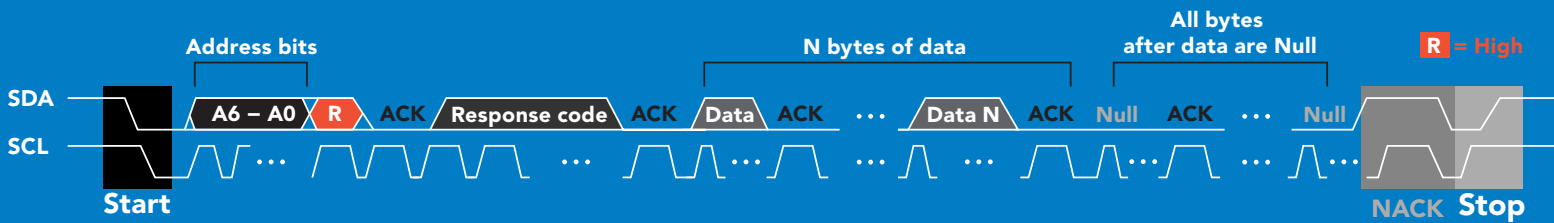
Advanced



Requesting data from device



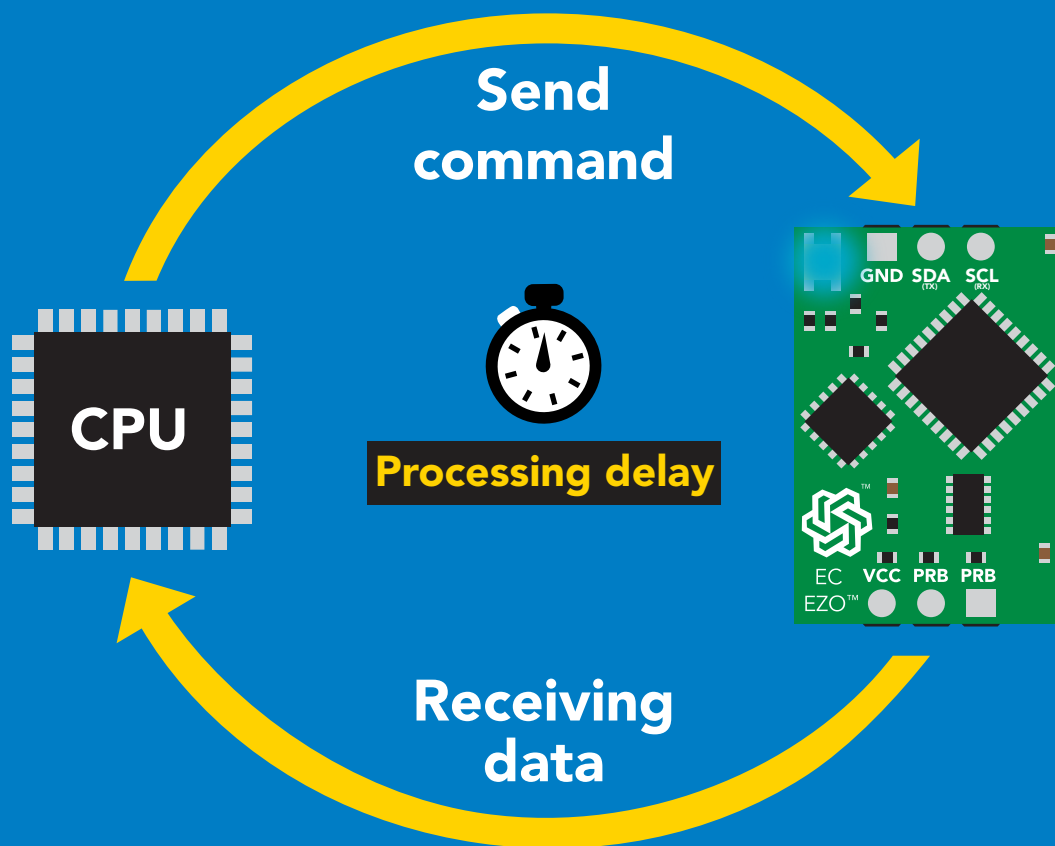
Advanced



Response codes

After a command has been issued, a 1 byte response code can be read in order to confirm that the command was processed successfully.

Reading back the response code is completely optional, and is not required for normal operation.



Example

```
I2C_start;  
I2C_address;  
I2C_write(EZO_command);  
I2C_stop;
```

```
delay(300);
```



```
Processing delay
```

```
I2C_start;  
I2C_address;  
Char[ ] = I2C_read;  
I2C_stop;
```

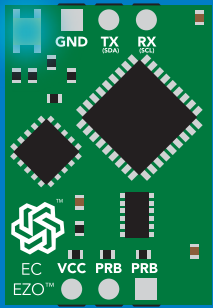
The response code will always be 254, if you do not wait for the processing delay.

Response codes

Single byte, not string

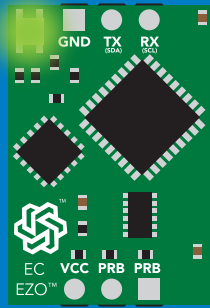
255	no data to send
254	still processing, not ready
2	syntax error
1	successful request

LED color definition



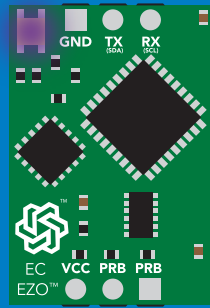
Blue

I²C standby



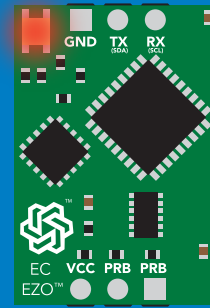
Green

Taking reading



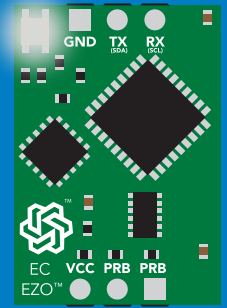
Purple

Changing
I²C address



Red

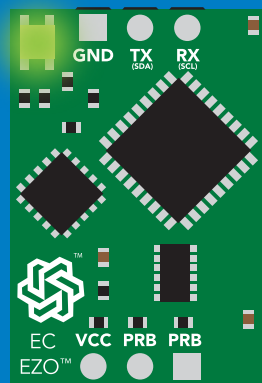
Command
not understood



White

Find

5V	LED ON +2.5 mA
3.3V	+1 mA



Solid Green LED

in UART mode
Not I²C ready

I²C mode

command quick reference

All commands are ASCII strings or single ASCII characters.

Command	Function	
Baud	switch back to UART mode	pg. 63
Cal	performs calibration	pg. 49
Export	export calibration	pg. 51
Factory	enable factory reset	pg. 62
Find	finds device with blinking white LED	pg. 47
i	device information	pg. 57
I2C	change I ² C address	pg. 61
Import	import calibration	pg. 52
K	set probe type	pg. 53
L	enable/disable LED	pg. 46
Name	set/show name of device	pg. 56
O	enable/disable parameters	pg. 55
Plock	enable/disable protocol lock	pg. 60
R	returns a single reading	pg. 48
Sleep	enter sleep mode/low power	pg. 59
Status	retrieve status information	pg. 58
T	temperature compensation	pg. 54
TDS	change the TDS conversion factor	pg. 50

LED control

Command syntax

300ms  processing delay

L,1 LED on **default**

L,0 LED off

L,? LED state on/off?

Example

Response

L,1

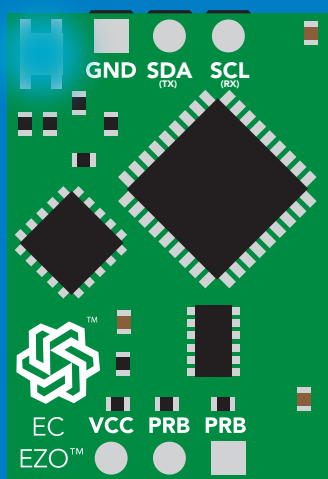

Wait 300ms **1** **0**
Dec Null

L,0

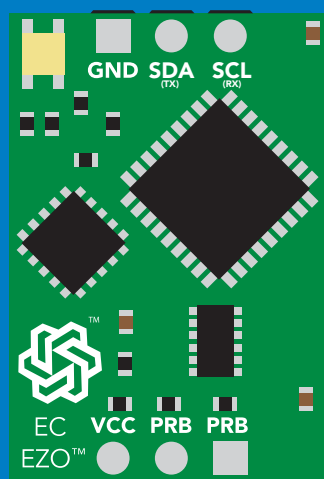

Wait 300ms **1** **0**
Dec Null

L,?


Wait 300ms **1** **?L,1** **0** or **1** **?L,0** **0**
Dec ASCII Null Dec ASCII Null



L,1



L,0

Find

300ms  processing delay

Command syntax

This command will disable continuous mode
Send any character or command to terminate find.

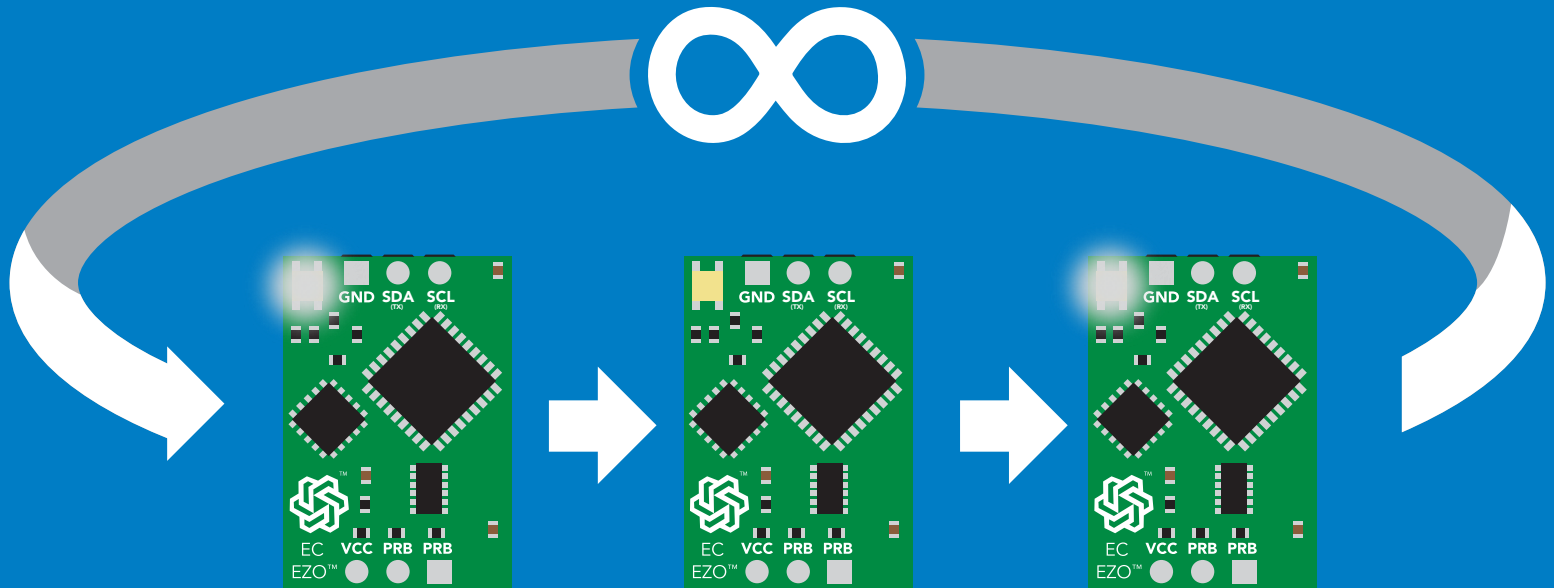
Find LED rapidly blinks white, used to help find device

Example

Response

Find

 Wait 300ms
1 Dec 0 Null



Taking reading

Command syntax

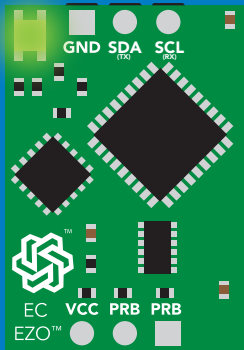
600ms  processing delay

R return 1 reading

Example

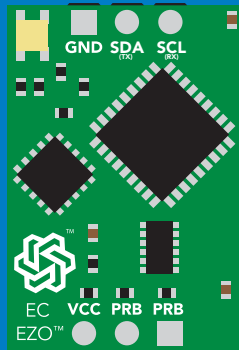
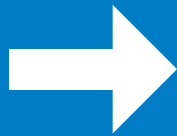
Response

R  **1** **1,413** **0**
Wait 600ms Dec ASCII Null

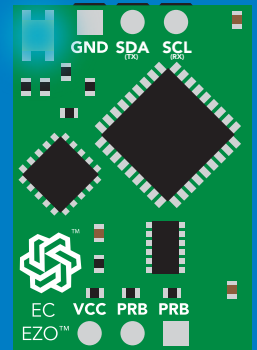
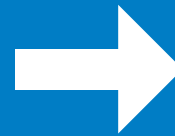


Green

Taking reading



Transmitting



Blue

Standby

Calibration

600ms  processing delay

Command syntax

Dry calibration must always be done first!

Cal,dry	dry calibration
Cal,n	single point calibration, where n = any value
Cal,low,n	low end calibration, where n = any value
Cal,high,n	high end calibration, where n = any value
Cal,clear	delete calibration data
Cal,?	device calibrated?

Example

Response

Cal,dry

 **Wait 600ms** **1** **0**
Dec Null

Cal,84

 **Wait 600ms** **1** **0**
Dec Null

Cal,low,12880

 **Wait 600ms** **1** **0**
Dec Null

Cal,high,80000

 **Wait 600ms** **1** **0**
Dec Null

Cal,clear

 **Wait 300ms** **1** **0**
Dec Null

Cal,?

 **Wait 300ms** **1** **?CAL,0** **0** or **1** **?CAL,1** **0** or **1** **?CAL,2** **0**
Dec ASCII Null Dec ASCII Null Dec ASCII Null
one point two point

One point calibration:

Step 1. "cal,dry"

Step 2. "cal,n"

Calibration complete!

Two point calibration:

Step 1 "cal,dry"

Step 2 "cal,low,n"

Step 3 "cal,high,n"

Calibration complete!

Changing the TDS (ppm) conversion factor

300ms  processing delay

There are several different conversion factors used to read TDS(ppm). For some applications, it may be necessary to use a conversion factor other than the default value of 0.54

Command syntax

TDS,n set custom conversion factor, n = any value between 0.01 – 1.00
TDS,? conversion factor being used


Example

Response

TDS,?

 Wait 300ms **1** **?TDS,0.54** **0**
 Dec ASCII Null


R

 Wait 300ms **1** **100,54** **0**
 Dec ASCII Null
 EC TDS
 ↓ ↓

TDS,0.46

 Wait 300ms **1** **0**
 Dec Null

R

 Wait 300ms **1** **100,46** **0**
 Dec ASCII Null
 EC TDS
 ↓ ↓

Common conversion factors

NaCl 0.47 – 0.50
 KCL 0.50 - 0.57
 "442" 0.65 – 0.85

Formula

EC x conversion factor = TDS

Export calibration

300ms  processing delay

Command syntax

Export: Use this command to download calibration settings

Export,? calibration string info

Export export calibration string from calibrated device

Example

Response

Export,?



Wait 300ms

1

Dec

10,120

ASCII

0

Null

Response breakdown

10, 120

of strings to export # of bytes to export

Export strings can be up to 12 characters long

Export



Wait 300ms

1

Dec

59 6F 75 20 61 72

ASCII

0

Null

(1 of 10)

Export



Wait 300ms

1

Dec

65 20 61 20 63 6F

ASCII

0

Null

(2 of 10)

(7 more)

⋮

Export



Wait 300ms

1

Dec

6F 6C 20 67 75 79

ASCII

0

Null

(10 of 10)

Export



Wait 300ms

1

Dec

*DONE

ASCII

0

Null

Import calibration

300ms  processing delay

Command syntax

Import: Use this command to upload calibration settings to one or more devices.

Import,n import calibration string to new device

Example

Import, 59 6F 75 20 61 72 (1 of 10)

Import, 65 20 61 20 63 6F (2 of 10)

⋮

Import, 6F 6C 20 67 75 79 (10 of 10)

Response

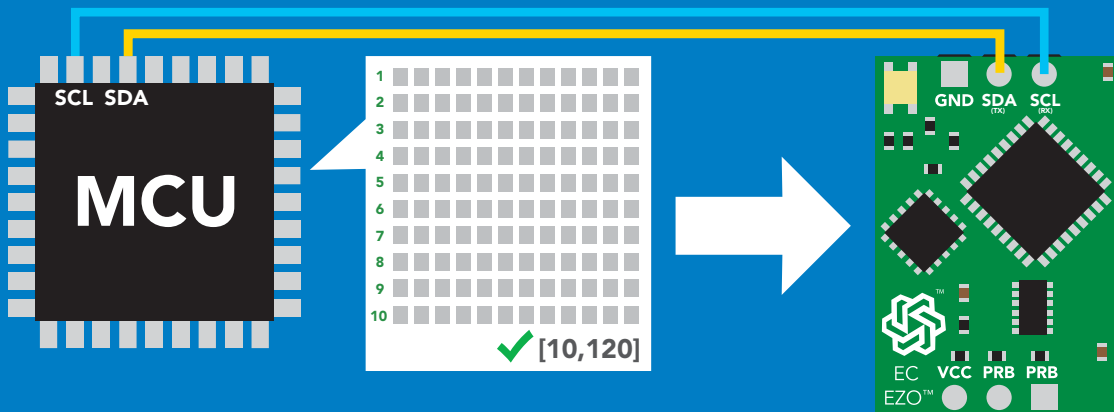
 **1** **0**
Wait 300ms Dec Null

 **1** **0**
Wait 300ms Dec Null

⋮

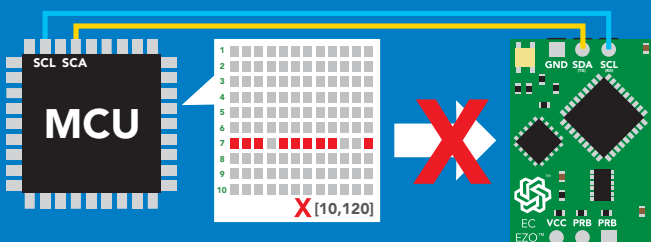
 **1** **0**
Wait 300ms Dec Null

Import,n



1 ***Pending** **0**
Dec ASCII Null

system will reboot



reboot

* If one of the imported strings is not correctly entered, the device will not accept the import and reboot.

Setting the probe type

Command syntax

300ms  processing delay

K,n n = any value; floating point in ASCII

K 1.0 is the default value

K,? probe K value?

Example

Response

K,10

 Wait 300ms
1 Dec 0 Null

K,?

 Wait 600ms
1 Dec K,10 ASCII 0 Null



K 0.1



K 1.0



K 10

Temperature compensation

Command syntax

Default temperature = 25°C
Temperature is always in Celsius
Temperature is not retained if power is cut

T,n n = any value; floating point or int 300ms  processing delay
T,? compensated temperature value?
RT,n set temperature compensation and take a reading

Example

Response

T,19.5

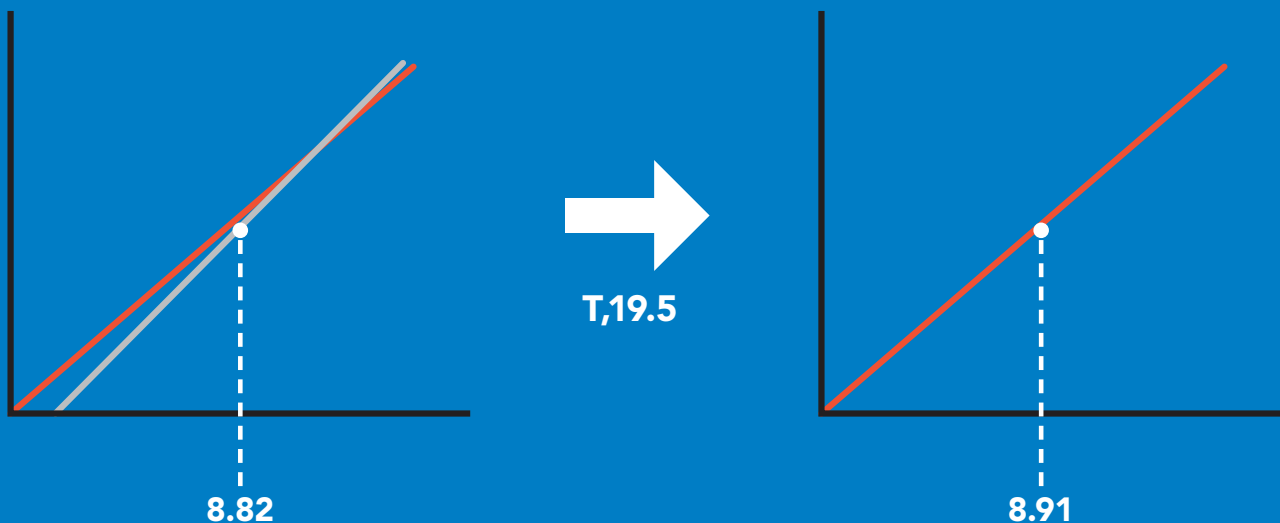
 **Wait 300ms** **1** **0**
Dec Null

RT,19.5

 **Wait 900ms** **1** **8.91** **0**
Dec ASCII Null

T,?

 **Wait 300ms** **1** **?T,19.5** **0**
Dec ASCII Null



Enable/disable parameters from output string

Command syntax

300ms  processing delay

O, [parameter],[1,0] enable or disable output parameter
O,? enabled parameter?

Example

Response

O,EC,1 / O,EC,0

 **1** **0** enable / disable conductivity
Wait 300ms Dec Null

O,TDS,1 / O,TDS,0

 **1** **0** enable / disable total dissolved solids
Wait 300ms Dec Null

O,S,1 / O,S,0

 **1** **0** enable / disable salinity
Wait 300ms Dec Null

O,SG,1 / O,SG,0

 **1** **0** enable / disable specific gravity
Wait 300ms Dec Null

O,?

 **1** **?O,EC,TDS,S,SG** **0** if all are enabled
Wait 300ms Dec ASCII Null

Parameters

EC Conductivity = $\mu\text{S/cm}$
TDS Total dissolved solids = ppm
S Salinity = PSU (ppt) 0.00 – 42.00
SG Specific gravity (sea water only) = 1.00 – 1.300

Followed by 1 or 0

1 enabled
0 disabled

* If you disable all possible data types your readings will display "no output".

Naming device

300ms  processing delay

Command syntax

Do not use spaces in the name

Name,n	set name	n =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Name,	clears name																	
Name,?	show name																	

Up to 16 ASCII characters

Example

Response

Name,



1
Dec

0
Null

name has been cleared

Name,zzt



1
Dec

0
Null

Name,?



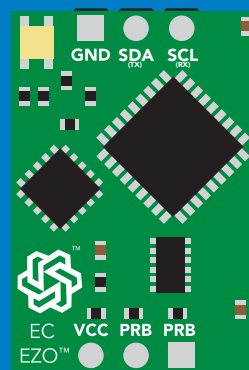
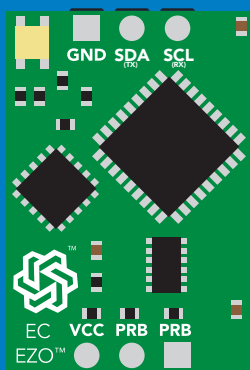
1
Dec

?Name,zzt
ASCII

0
Null

Name,zzt

Name,?



1 0

1 ?Name,zzt 0

Device information

Command syntax

300ms  processing delay

i device information

Example

Response

i



Wait 300ms

1

Dec

?i,EC, 2.16

ASCII

0

Null

Response breakdown

?i, EC, 2.16
↑ ↑
Device Firmware

Reading device status

Command syntax

300ms  processing delay

Status voltage at Vcc pin and reason for last restart

Example

Response

Status

 **1** **?Status,P,5.038** **0**
Wait 300ms Dec ASCII Null

Response breakdown

?Status, **P,** **5.038**
Reason for restart Voltage at Vcc

Restart codes

P powered off
S software reset
B brown out
W watchdog
U unknown

Sleep mode/low power

Command syntax

Sleep enter sleep mode/low power

Send any character or command to awaken device.

Example

Response

Sleep

no response

Do not read status byte after issuing sleep command.

Any command

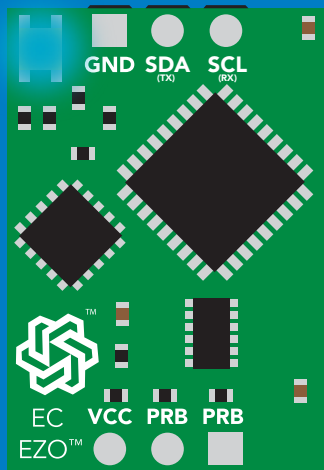
wakes up device

5V

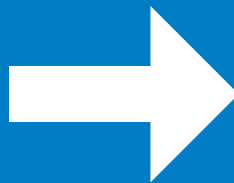
STANDBY	SLEEP
18.14 mA	0.7 mA

3.3V

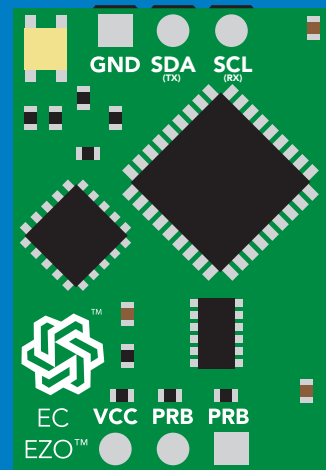
16.85 mA	0.4 mA
----------	--------



Standby



Sleep



Sleep

Protocol lock

Command syntax

300ms  processing delay

Plock,1 enable Plock

Plock,0 disable Plock

Plock,? Plock on/off?

Locks device to I²C mode.

default

Example

Response

Plock,1

 Wait 300ms
1 0
Dec Null

Plock,0

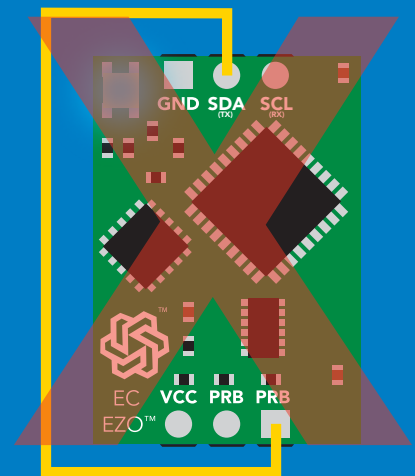
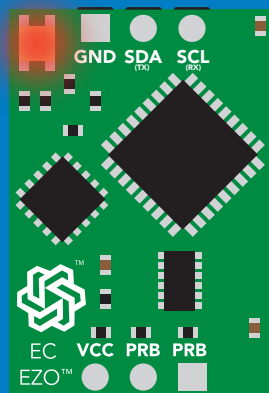
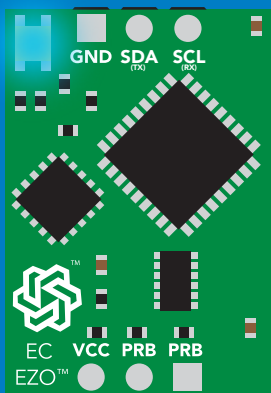
 Wait 300ms
1 0
Dec Null

Plock,?

 Wait 300ms
1 ?Plock,1 0
Dec ASCII Null

Plock,1

Baud, 9600



cannot change to UART

cannot change to UART

I²C address change

Command syntax

300ms  processing delay

I2C,n sets I²C address and reboots into I²C mode

Example

Response

I2C,101

device reboot
(no response given)

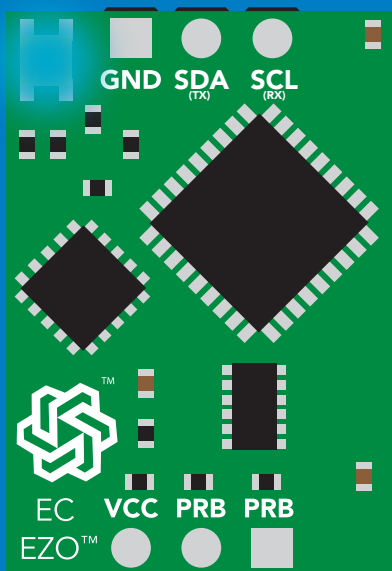
Warning!

Changing the I²C address will prevent communication between the circuit and the CPU until your CPU is updated with the new I²C address.

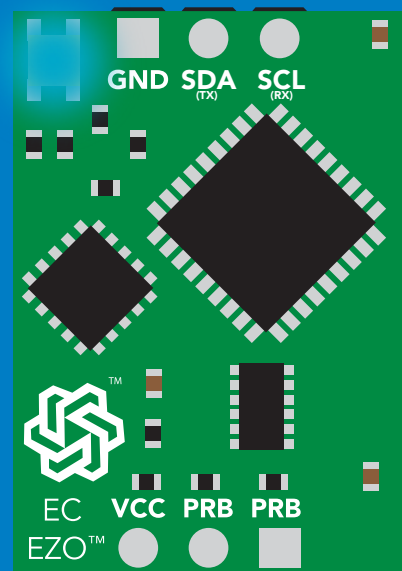
Default I²C address is 100 (0x64).

n = any number 1 – 127

I2C,101



(reboot)



Factory reset

Command syntax

Factory reset will not take the device out of I²C mode.

Factory enable factory reset

I²C address will not change

Example

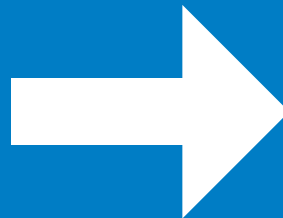
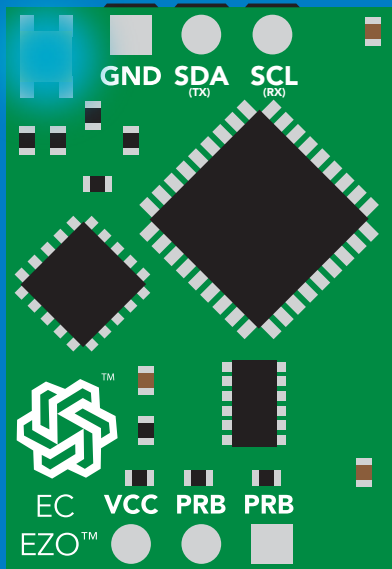
Response

Factory

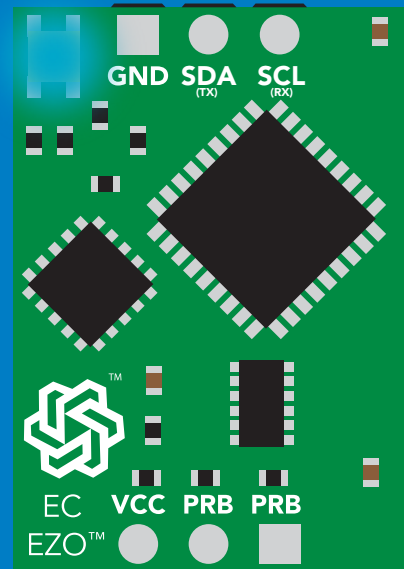
device reboot
(no response given)

Clears calibration
LED on
Response codes enabled

Factory



(reboot)



Change to UART mode

Command syntax

Baud,n switch from I²C to UART

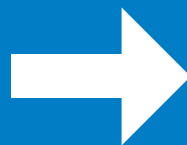
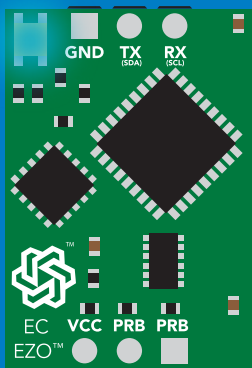
Example

Baud,9600

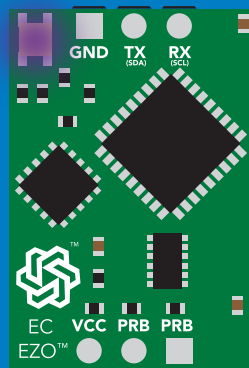
Response

reboot in UART mode
(no response given)

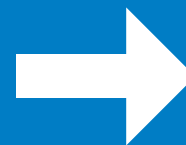
n = [300
1200
2400
9600
19200
38400
57600
115200



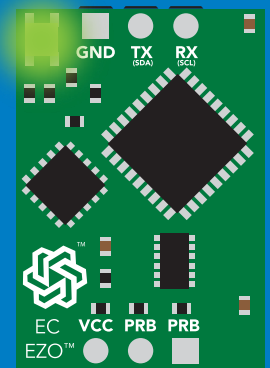
Baud,9600



Changing to
UART mode



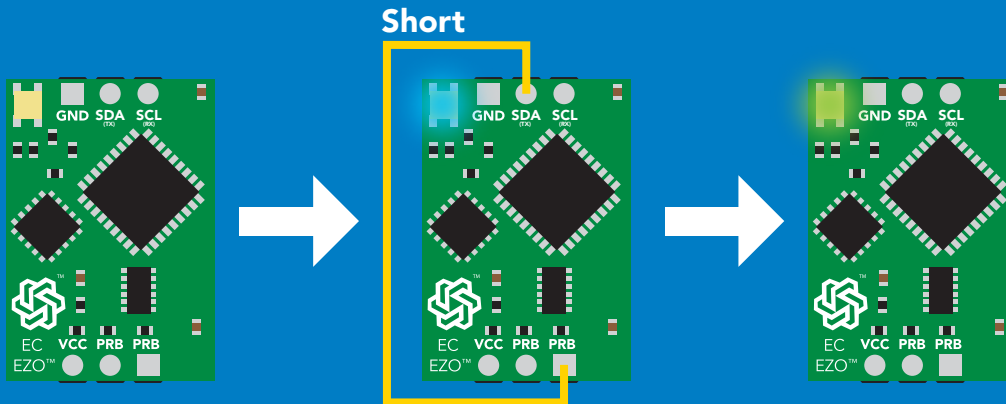
(reboot)



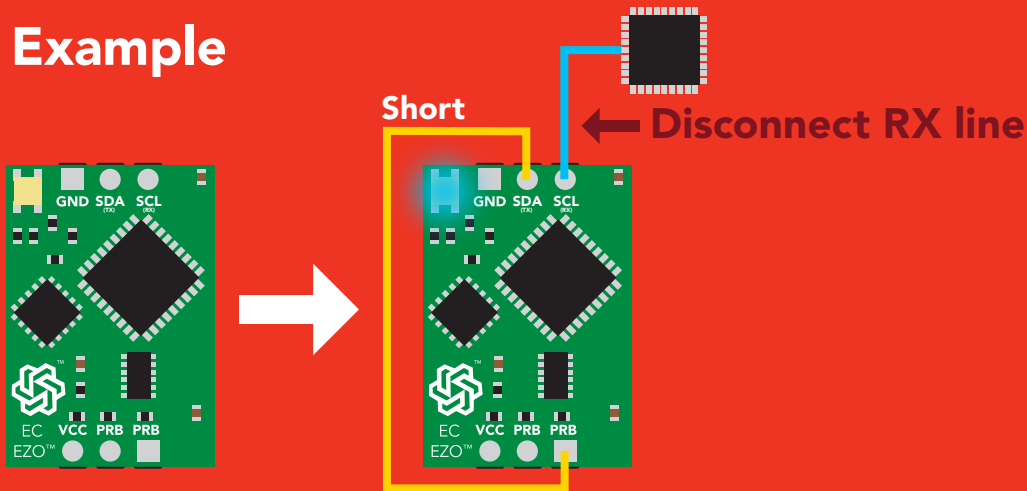
Manual switching to UART

- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to the right PRB
- Confirm RX is disconnected
- Connect ground (power on)
- Wait for LED to change from Blue to Green
- Disconnect ground (power off)
- Reconnect all data and power

Example



Wrong Example



Calibration theory

The accuracy of your readings is directly related to the quality of your calibration.
(Calibration is not difficult, and a little bit of care goes a long way)

A properly calibrated conductivity probe will never need recalibration. Once calibrated, you can use the probe continuously year after year without concern. This is because a conductivity probe does not contain any parts that wear out over time.

However, changing the cable length of the probe or moving the EZO-EC circuit from one machine to another may require recalibration. This is because such actions will change the electrical properties of the probe or EC circuit.



Two point or Three point calibration

No calibration



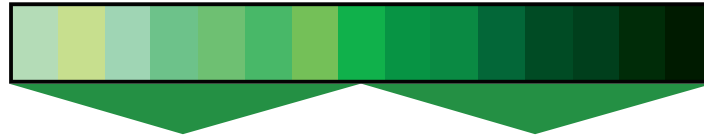
Approximation

Single point calibration



Narrow band accuracy

Two point calibration

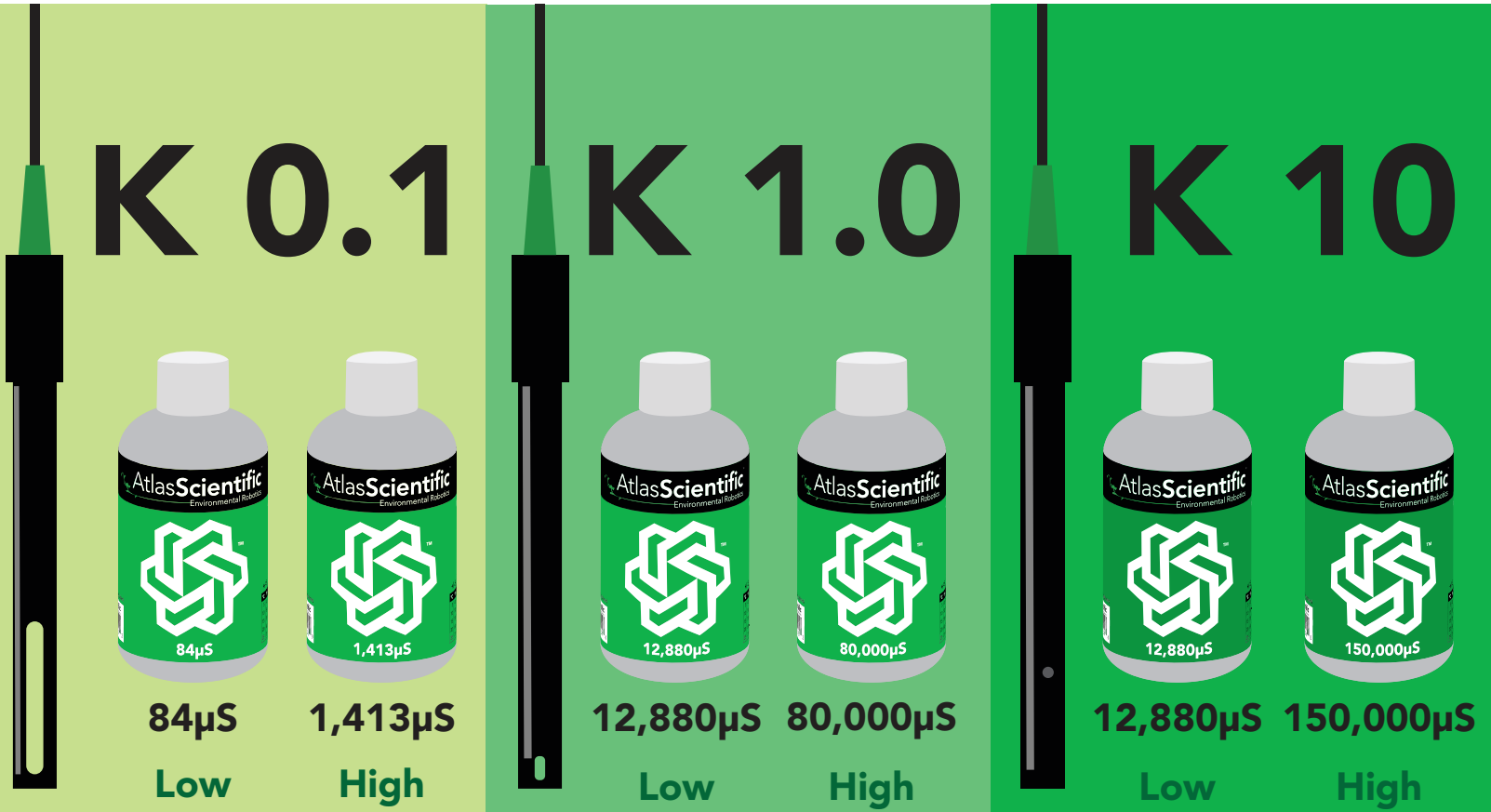


Low point

High point

Wide range accuracy

Recommended calibration points

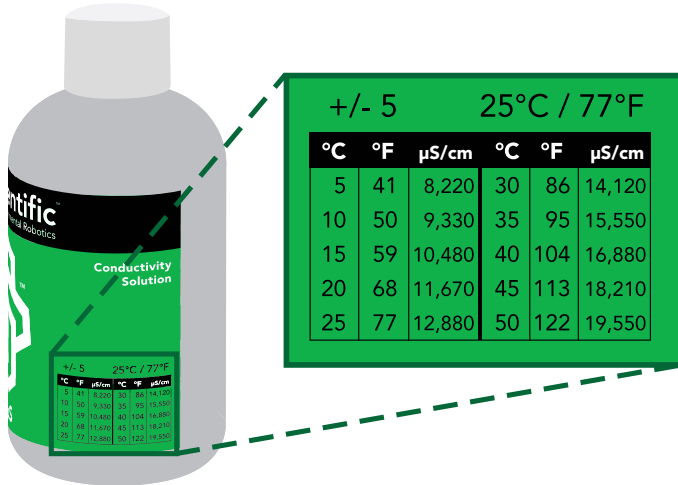


When calibrating, Atlas Scientific recommends using the above μS values. However, you can use any μS values you want.

Temperature compensation during calibration

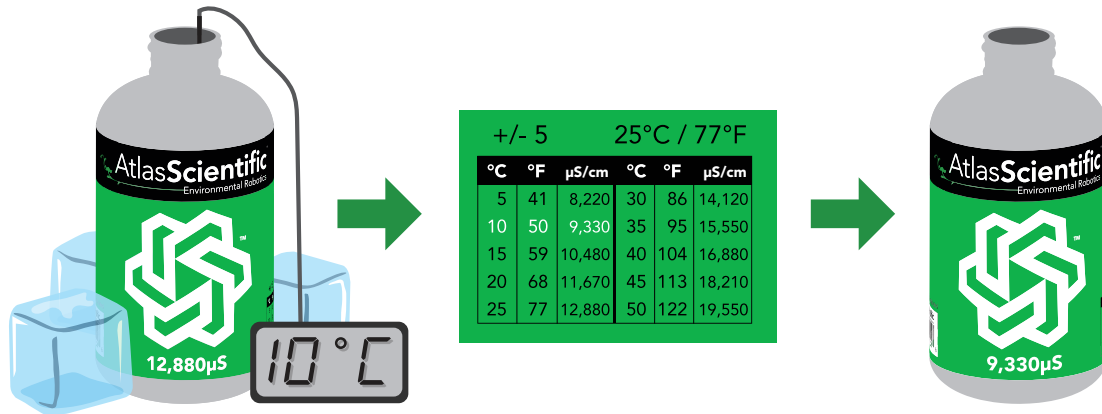
Temperature has a significant effect on conductivity readings. The EZO™ Conductivity circuit has its temperature compensation set to 25° C as the default. **At no point should you change the default temperature compensation during calibration.**

If the solution is +/- 5° C (or more), refer to the chart on the bottle, and calibrate to that value.



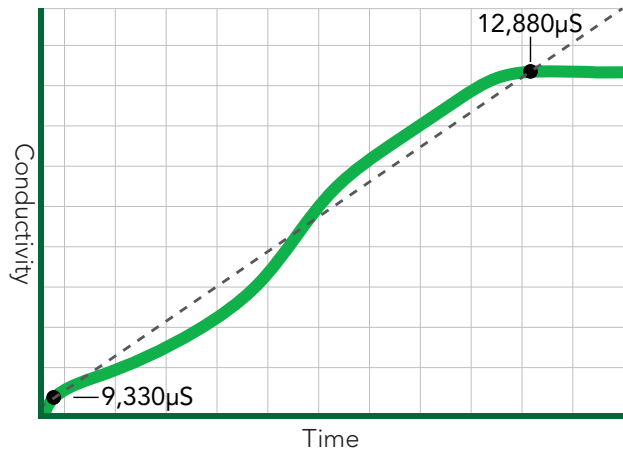
Temperature compensation example

For this example, we brought the temperature of the solution down to 10° C. Referring to chart on the bottle, you can see the value you should calibrate to is **9,330µS**.



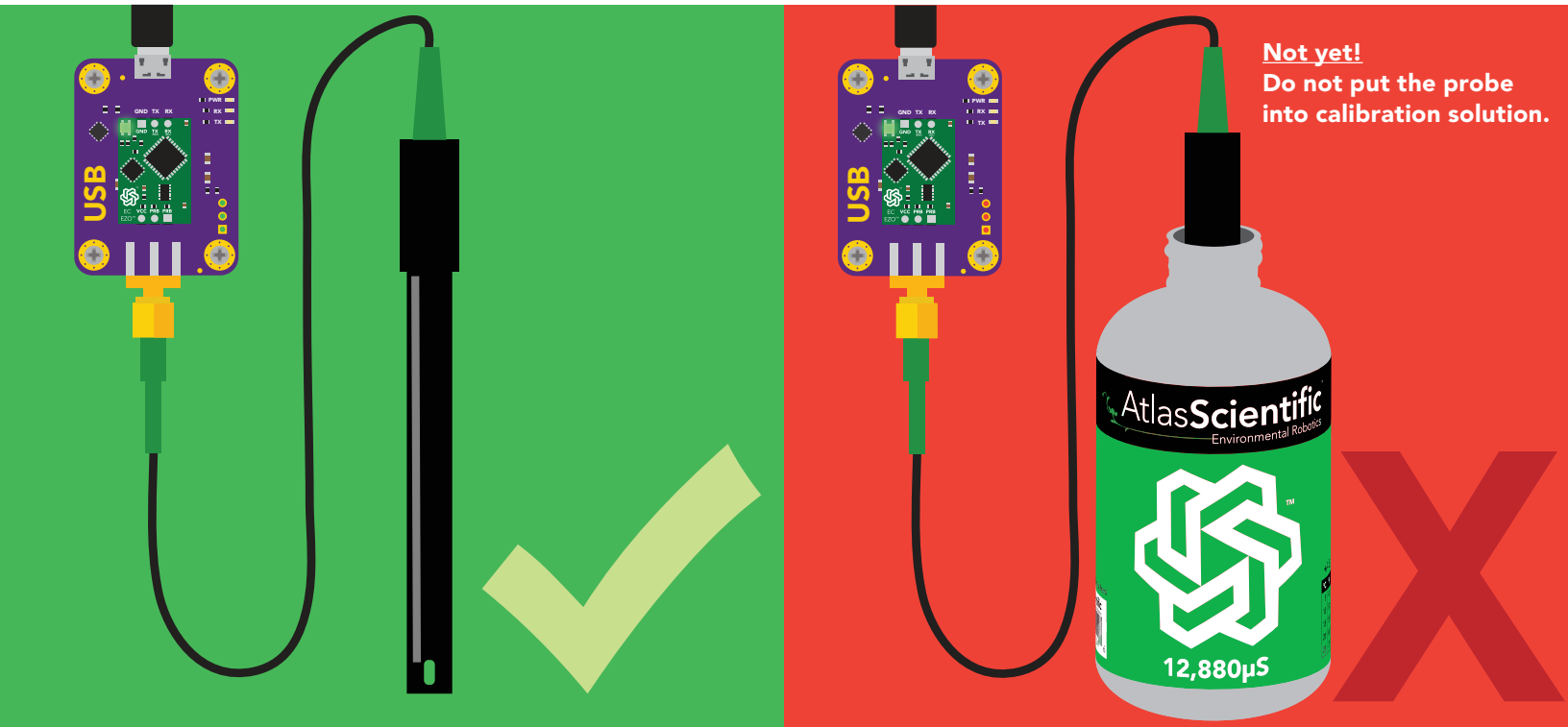
Over time, the readings will normalize as the solution warms to 25° C.

See pages [27](#) or [54](#) for more information.



1. Pre-calibration setup

Connect the dry conductivity probe and take continuous readings.



2. Set probe type

If your probe \neq K 1.0 (default), then set the probe type by using the "**K,n**" command. (where $n = K$ value of your probe) for more information, see page 26 or 53.

3. Dry calibration

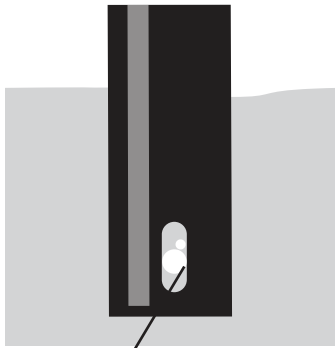
Perform a dry calibration using the command "**Cal,dry**". Even though you may see readings of 0.00 before issuing the "**Cal,dry**" command, it is still a necessary part of calibration.

00.00 → "**Cal,dry**" → 0.00 ✓ Correct

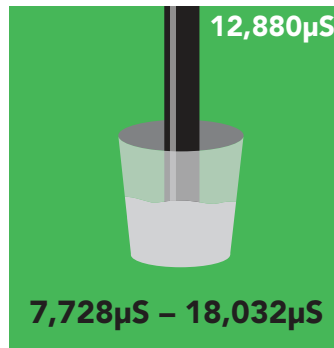
17.00 → "**Cal,dry**" → 0.00 ✓ Also correct

Two point calibration - low point

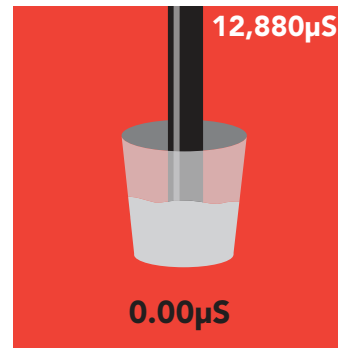
Pour a small amount of the low point calibration solution into a cup. Shake the probe to make sure you do not have trapped air bubbles in the sensing area. You should see readings that are off by **1 – 40%** from the stated value of the calibration solution. Wait for readings to stabilize (*small movement from one reading to the next is normal*).



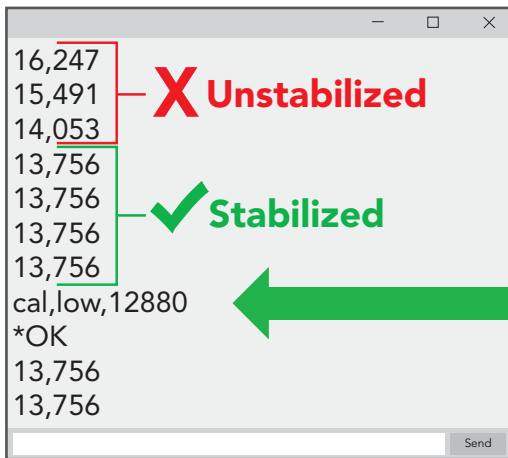
Trapped air in sensing area (shake to remove)



+/- 40%



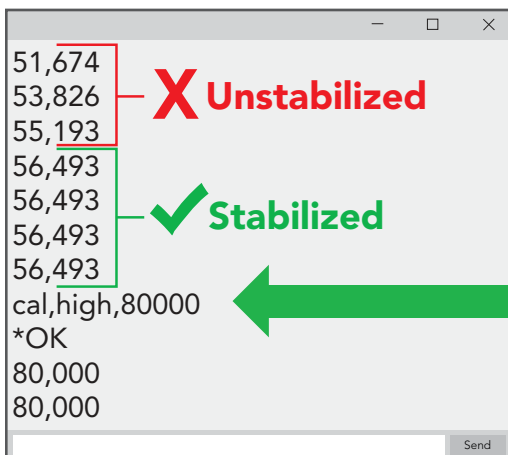
check probe connection, you cannot calibrate to 0.



Once the readings stabilize, issue the low point calibration command. "**cal,low,12880**" (Readings will **NOT** change)

Two point calibration - high point

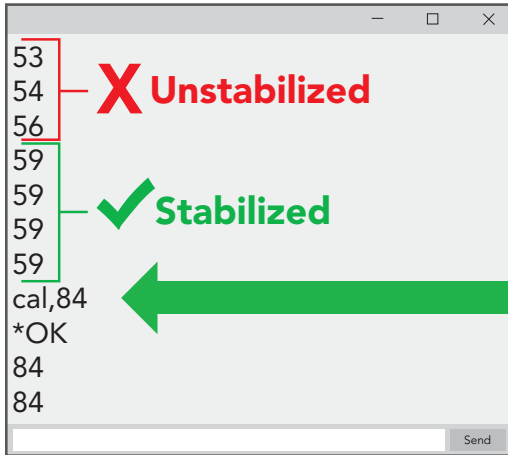
- Rinse off the probe before calibrating to the high point.
- Pour a small amount of the high point calibration solution into a cup.
- Shake the probe to remove trapped air.
- Readings may be off by +/- 40%
- Wait for readings to stabilize.



Once the readings stabilize, issue the high point calibration command. "**cal,high,80000**" (Readings **will** change, calibration complete).

Single point calibration

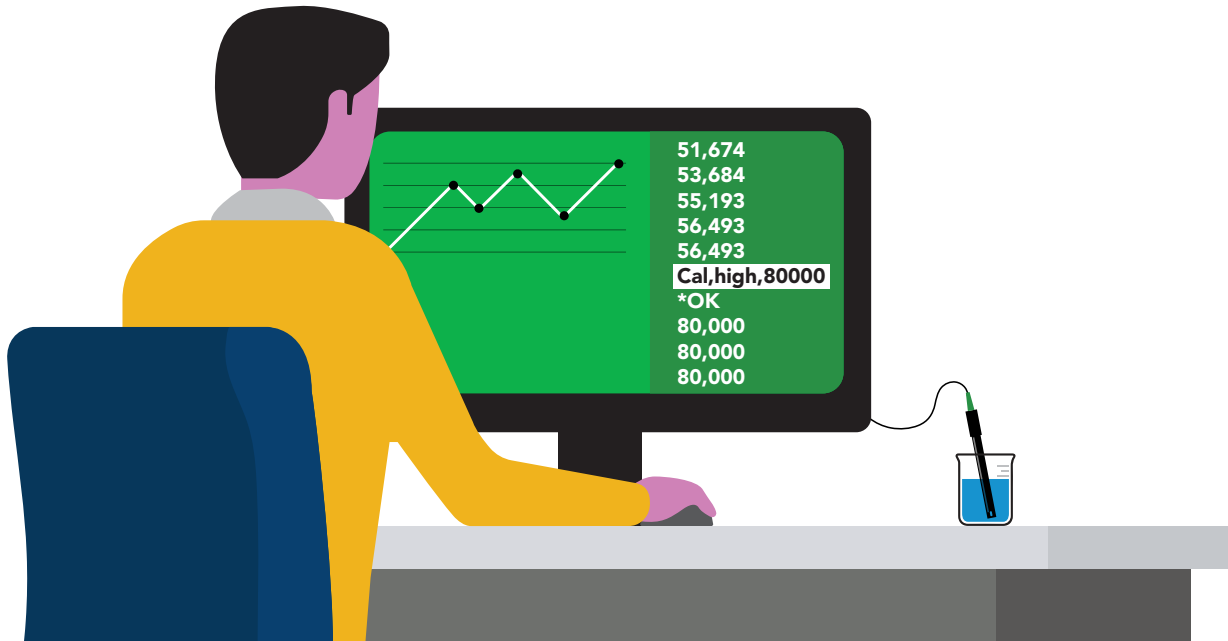
- Pour a small amount of calibration solution into a cup (μS value of your choice).
- Shake the probe to remove trapped air.
- Readings may be off by +/- 40%
- Wait for readings to stabilize.



Once the readings stabilize, issue the single point calibration command. "**cal,n**" where n = any value. (Readings **will** change, calibration complete).a

Best practices for calibration

Always watch the readings throughout the calibration process.
Issue calibration commands once the readings have stabilized.



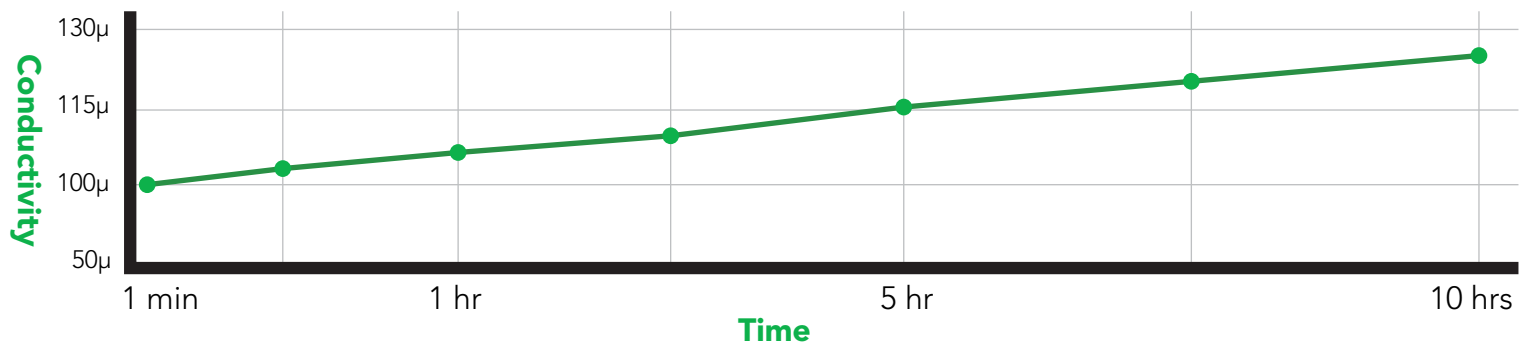
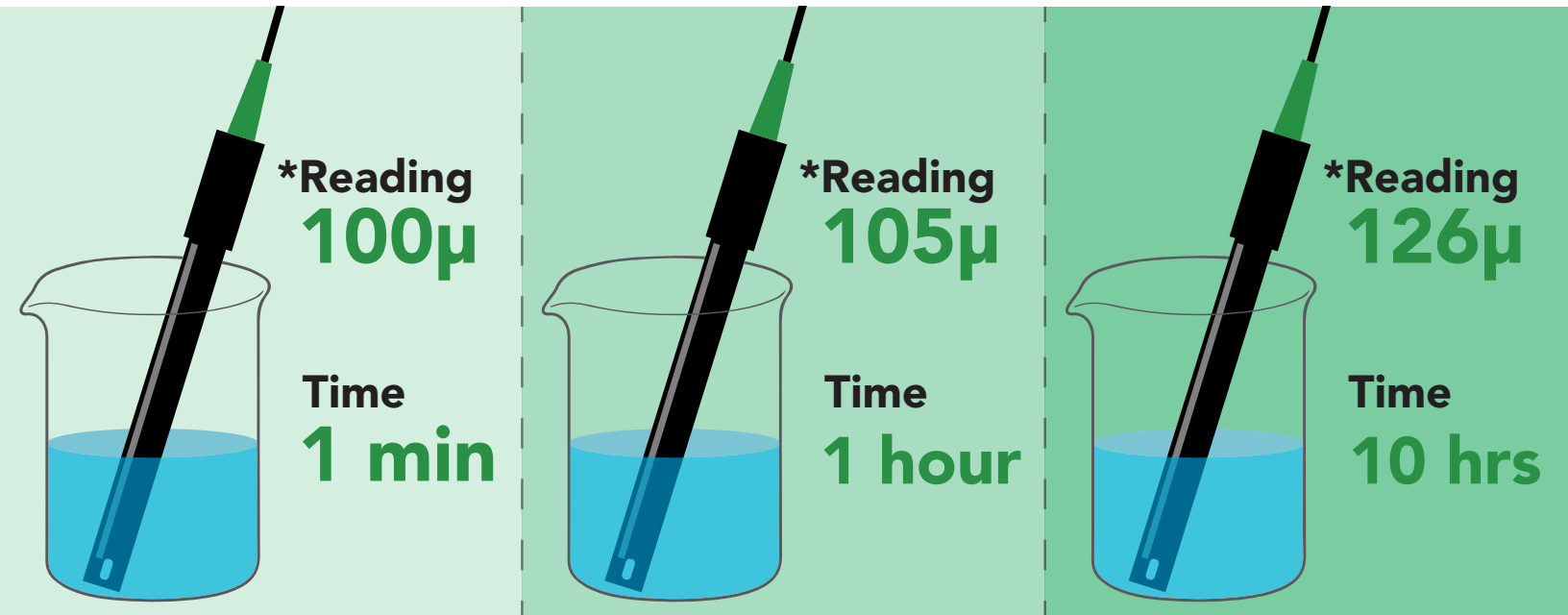
⚠ Never do a blind calibration! ⚠

Issuing a calibration command before the readings stabilize will result in drifting readings.



Long-term conductivity measurements in stagnant water

Taking continuous conductivity readings in stagnant water:



A small amount of energy must be put into the water to measure conductivity. This small amount of energy will start to affect the readings in stagnant water. Over time, the energy passing through the stagnant water will start to align the dissolved salts along a path of least resistance. Lowering the resistance of the water will increase the water's conductivity.

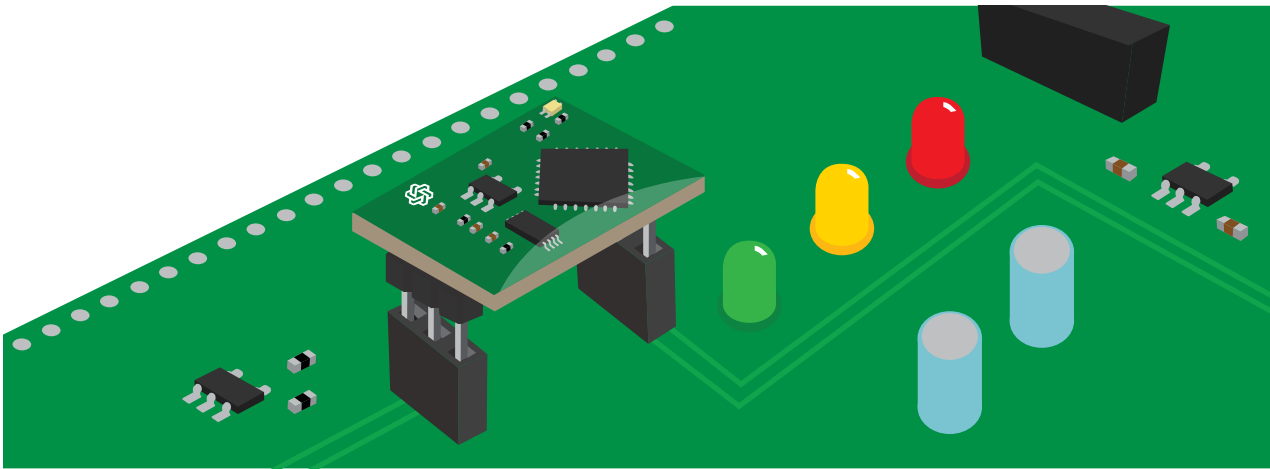
Moving the probe or the water will disrupt this alignment and cause the readings to suddenly return to normal.

****These are example readings; there is no way to predict how the readings will change over time.***

Soldering

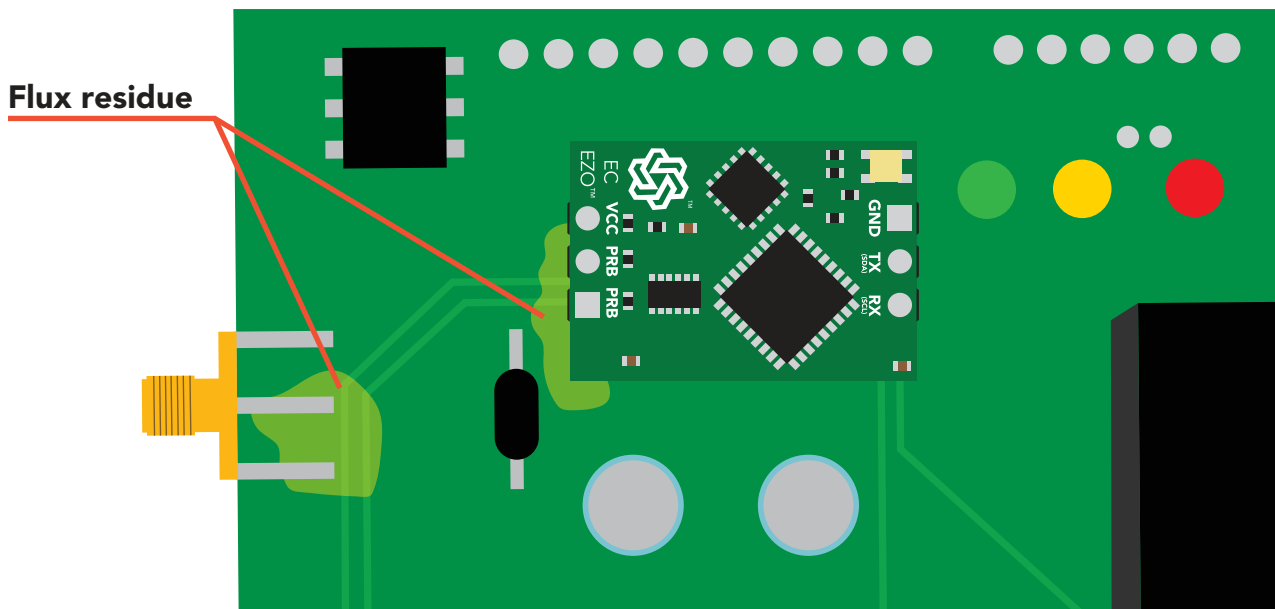
Do not directly solder an EZO circuit to your PCB. If something goes wrong during the soldering process it may become impossible to correct the problem. It is simply not worth the risk.

Instead, solder female header pins to your PCB and place the EZO device in the female headers.



Avoid using rosin core solder.
Use as little flux as possible.

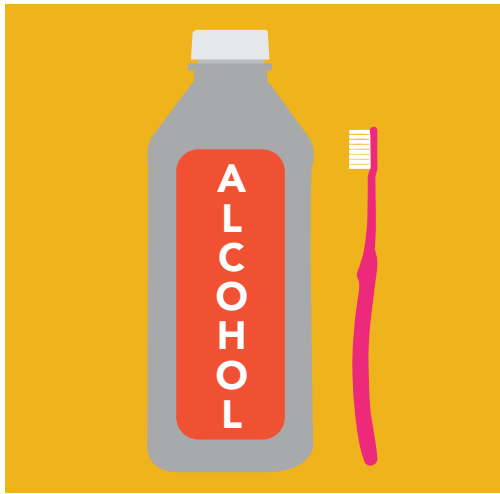
Flux residue will severely affect your readings. Any Flux residue that comes in contact with the PRB pins or your probes connector will cause a “flux short”.



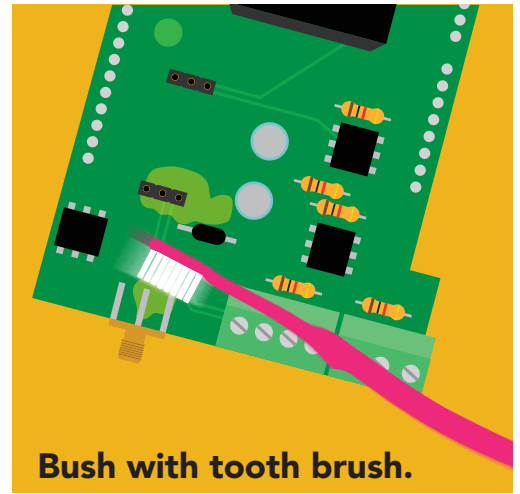
You **MUST** remove all the flux residue from your PCB after soldering.

Soldering

Removing flux residue can be done with commercially available products such as flux off or you can use alcohol and a tooth brush.



Remove EZO Circuit and soak in alcohol for 10 mins.



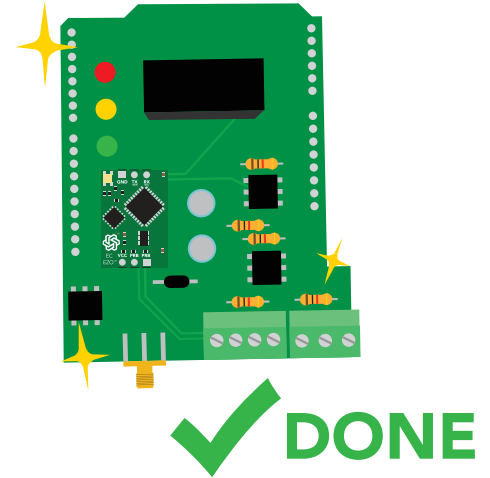
Bush with tooth brush.



Soak in alcohol for 5 mins.



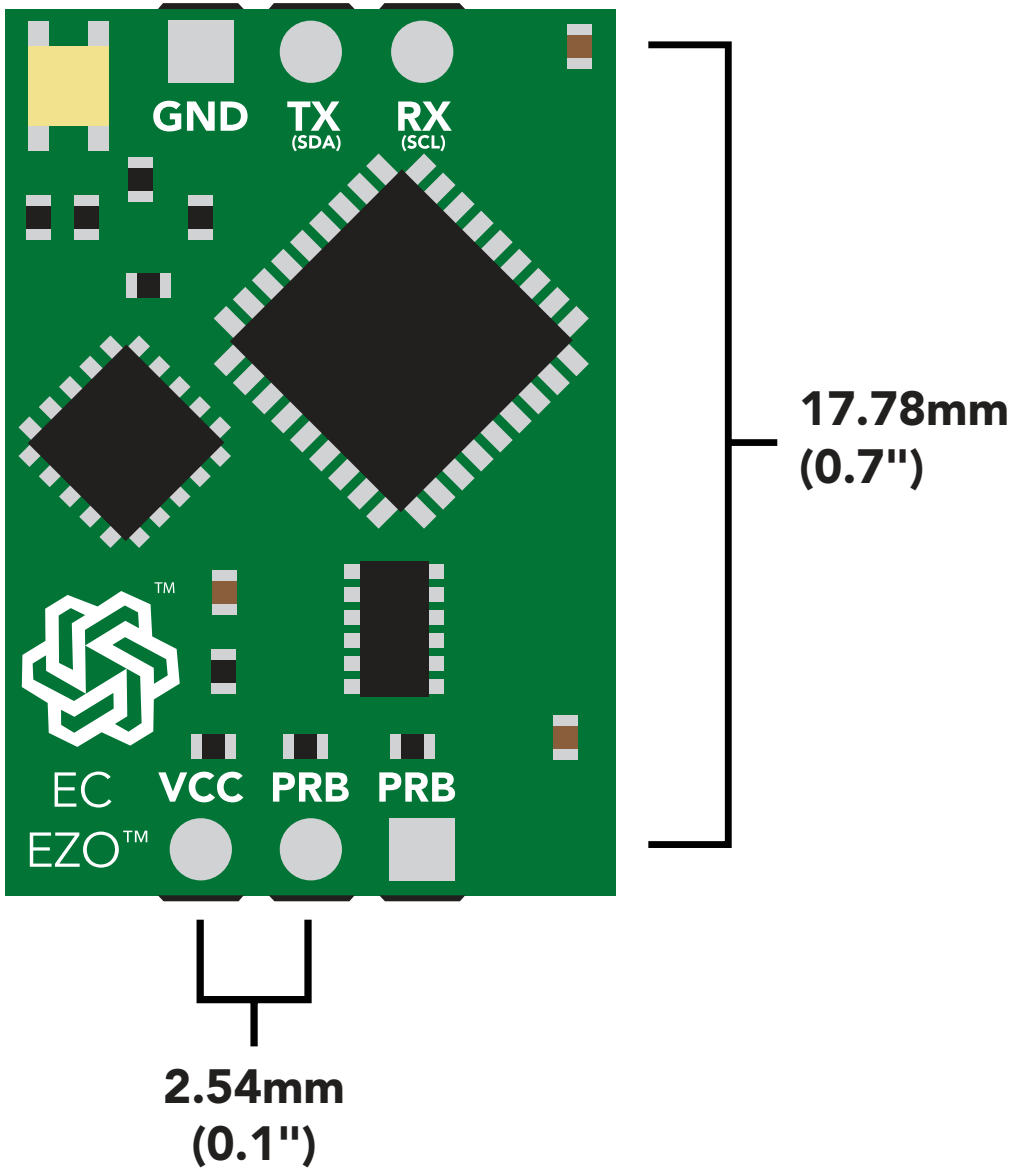
Let it dry in the air.



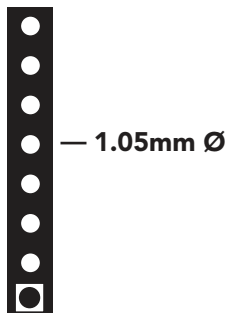
What does a flux short look like?

Readings move slowly and take several minutes to reach the correct value.

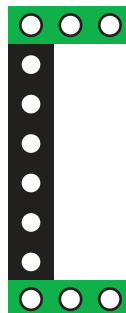
EZO™ circuit footprint



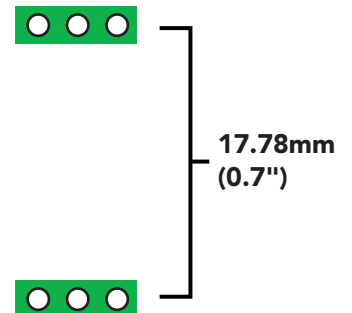
1 In your CAD software, place a 8 position header.



2 Place a 3 position header at both top and bottom of the 8 position.



3 Delete the 8 position header. The two 3 position headers are now 17.78mm (0.7") apart from each other.



Datasheet change log

Datasheet V 6.4

Revised entire document.

Datasheet V 6.3

Revised naming device info on pages 36 & 63.

Datasheet V 6.2

Added new command:

"TDS,n" Changing the TDS (ppm) conversion factor on pages 30 (UART) & 57 (I²C).

Datasheet V 6.1

Corrected typos within the datasheet.

Datasheet V 6.0

Changed the K value range from 0.1 to 0.01 on pg 5.

Datasheet V 5.9

Moved Default state to pg 17.

Datasheet V 5.8

Revised conductivity probe range information on pg 5.

Datasheet V 5.7

Revised response for the sleep command in UART mode on pg 39.

Datasheet V 5.6

Added more information on the Export calibration and Import calibration commands.

Datasheet V 5.5

Revised calibration theory pages, added information on temperature compensation on pg. 15, moved data isolation to pg 9, and correct wiring to pg 11.

Datasheet V 5.4

Revised isolation schematic on pg. 13

Datasheet V 5.3

Added new command:

"RT,n" for Temperature compensation located on pages 30 (UART) & 55 (I²C).
Added firmware information to Firmware update list.

Datasheet V 5.2

Revised calibration information on pages 27 & 52.

Datasheet V 5.1

Added more information about temperature compensation on pages 30 & 55.

Datasheet V 5.0

Changed "Max rate" to "Response time" on cover page.

Datasheet V 4.9

Removed note from certain commands about firmware version.
Added steps to calibration command pages 27 (UART) and 52 (I²C).

Datasheet V 4.8

Revised definition of response codes on pg 46.

Datasheet V 4.7

Revised cover page art.

Datasheet V 4.6

Updated calibration processing delay time on pg.52.

Datasheet V 4.5

Revised Enable/disable parameters information on pages 31 & 56.

Datasheet V 4.4

Updated High point calibration info on page 11.

Datasheet V 4.3

Updated calibration info on pages 27 (UART) and 52 (I²C).

Datasheet V 4.2

Revised Plock pages to show default value.

Datasheet V 4.1

Corrected I²C calibration delay on pg. 52.

Datasheet V 4.0

Revised entire datasheet.

Firmware updates

V1.0 – Initial release (April 17, 2014)

V1.1 – (June 2, 2014)

- Change specific gravity equation to return 1.0 when the uS reading is < 1000 (previously returned 0.0)
- Change accuracy of specific gravity from 2 decimal places to 3 decimal places
- Don't save temperature changes to EEPROM

V1.2 – (Aug 1, 2014)

- Baud rate change is now a long, purple blink

V1.5 – Baud rate change (Nov 6, 2014)

- Change default baud rate to 9600

V1.6 – I2C bug (Dec 1, 2014)

- Fixed I²C bug where the circuit may inappropriately respond when other I2C devices are connected

V1.8 – Factory (April 14, 2015)

- Changed "X" command to "Factory"

V1.95 – Plock (March 31, 2016)

- Added protocol lock feature "Plock"

V1.96 – EEPROM (April 26, 2016)

- Fixed bug where EEPROM would get erased if the circuit lost power 900ms into startup
This would cause the EZO circuit to revert back to UART mode if set to I2C

V2.10 – (April 12, 2017)

- Added "Find" command.
- Added "Export/import" command.
- Modified continuous mode to be able to send readings every "n" seconds.
- Default output changed from CSV string of 4 values to just conductivity; Other values must be enabled

V2.11 – (April 28, 2017)

- Fixed "Sleep" bug, where it would draw excessive current.

V2.12 – (May 9, 2017)

- Fixed bug in sleep mode, where circuit would wake up to a different I²C address.

V2.13 – (July 16, 2018)

- Added "RT" command to Temperature compensation

V2.14 – (Nov 26, 2019)

- The K value range has been extended to 0.01

V2.15 – (June 29, 2020)

- Fixed bug where output doesn't always round to 0

Firmware updates

V2.16 – (Dec 14, 2021)

- Internal update for new part compatibility.

Warranty

Atlas Scientific™ Warranties the EZO™ class Conductivity circuit to be free of defect during the debugging phase of device implementation, or 30 days after receiving the EZO™ class Conductivity circuit (which ever comes first).

The debugging phase

The debugging phase as defined by Atlas Scientific™ is the time period when the EZO™ class Conductivity circuit is inserted into a bread board, or shield. If the EZO™ class Conductivity circuit is being debugged in a bread board, the bread board must be devoid of other components. If the EZO™ class Conductivity circuit is being connected to a microcontroller, the microcontroller must be running code that has been designed to drive the EZO™ class Conductivity circuit exclusively and output the EZO™ class Conductivity circuit data as a serial string.

It is important for the embedded systems engineer to keep in mind that the following activities will void the EZO™ class Conductivity circuit warranty:

- **Soldering any part of the EZO™ class Conductivity circuit.**
- **Running any code, that does not exclusively drive the EZO™ class Conductivity circuit and output its data in a serial string.**
- **Embedding the EZO™ class Conductivity circuit into a custom made device.**
- **Removing any potting compound.**

Reasoning behind this warranty

Because Atlas Scientific™ does not sell consumer electronics; once the device has been embedded into a custom made system, Atlas Scientific™ cannot possibly warranty the EZO™ class Conductivity circuit, against the thousands of possible variables that may cause the EZO™ class Conductivity circuit to no longer function properly.

Please keep this in mind:

- 1. All Atlas Scientific™ devices have been designed to be embedded into a custom made system by you, the embedded systems engineer.**
- 2. All Atlas Scientific™ devices have been designed to run indefinitely without failure in the field.**
- 3. All Atlas Scientific™ devices can be soldered into place, however you do so at your own risk.**

Atlas Scientific™ is simply stating that once the device is being used in your application, Atlas Scientific™ can no longer take responsibility for the EZO™ class Conductivity circuits continued operation. This is because that would be equivalent to Atlas Scientific™ taking responsibility over the correct operation of your entire device.