EZO-EC™
Embedded Conductivity Circuit

Reads
- Conductivity = μS/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+ μS/cm

Accuracy
+/- 2%

Response time
1 reading per sec

Supported probes
K 0.1 – K 10 any brand

Calibration
1 or 2 point

Temp compensation
Yes

Data protocol
UART & I2C

Default I2C address
100 (0x64)

Operating voltage
3.3V – 5V

Data format
ASCII

This is an evolving document, check back for updates.
This is sensitive electronic equipment. Get this device working in a solderless breadboard first. Once this device has been soldered it is no longer covered by our warranty.

This device has been designed to be soldered and can be soldered at any time. Once that decision has been made, Atlas Scientific no longer assumes responsibility for the device’s continued operation. The embedded systems engineer is now the responsible party.

Get this device working in a solderless breadboard first!

Do not embed this device without testing it in a solderless breadboard!
Table of contents

Circuit dimensions 4
Power consumption 4
Absolute max ratings 4
Conductivity probe range 5
Resolution 6
Operating principle 7
Output units 8
Power and data isolation 9
Correct wiring 11
Calibration theory 12
Default state 17
Available data protocols 18

UART

UART mode 20
Receiving data from device 21
Sending commands to device 22
LED color definition 23
UART quick command page 24
LED control 25
Find 26
Continuous reading mode 27
Single reading mode 28
Calibration 29
Change TDS conversion factor 30
Export calibration 31
Import calibration 32
Setting the probe type 33
Temperature compensation 34
Enable/disable parameters 35
Naming device 36
Device information 37
Response codes 38
Reading device status 39
Sleep mode/low power 40
Change baud rate 41
Protocol lock 42
Factory reset 43
Change to I2C mode 44
Manual switching to I2C 45

I2C

I2C mode 47
Sending commands 48
Requesting data 49
Response codes 50
LED color definition 51
I2C quick command page 52
LED control 53
Find 54
Taking reading 55
Calibration 56
Change TDS conversion factor 57
Export calibration 58
Import calibration 59
Setting the probe type 60
Temperature compensation 61
Enable/disable parameters 62
Naming device 63
Device information 64
Reading device status 65
Sleep mode/low power 66
Protocol lock 67
I2C address change 68
Factory reset 69
Change to UART mode 70
Manual switching to UART 71

Circuit footprint 72
Datasheet change log 73
Warranty 77
Power consumption

<table>
<thead>
<tr>
<th></th>
<th>LED</th>
<th>MAX</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>ON</td>
<td>50 mA</td>
<td>18.14 mA</td>
<td>0.7 mA</td>
</tr>
<tr>
<td></td>
<td>OFF</td>
<td>45 mA</td>
<td>15.64 mA</td>
<td></td>
</tr>
<tr>
<td>3.3V</td>
<td>ON</td>
<td>35 mA</td>
<td>16.85 mA</td>
<td>0.4 mA</td>
</tr>
<tr>
<td></td>
<td>OFF</td>
<td>34 mA</td>
<td>15.85 mA</td>
<td></td>
</tr>
</tbody>
</table>

Absolute max ratings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage temperature (EZOTM Conductivity)</td>
<td>-60 °C</td>
<td></td>
<td>150 °C</td>
</tr>
<tr>
<td>Operational temperature (EZOTM Conductivity)</td>
<td>-40 °C</td>
<td>25 °C</td>
<td>125 °C</td>
</tr>
<tr>
<td>VCC</td>
<td>3.3V</td>
<td>5V</td>
<td>5.5V</td>
</tr>
</tbody>
</table>
Conductivity probe range

The EZO™ Conductivity circuit is capable of connecting to any two-conductor conductivity probe, ranging from:

K 0.01 → K 10

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

**K 0.1**
- **accurate reading range**: 0.07μS/cm – 50,000μS/cm
- TDS (ppm) 0 – 25,000
- Salinity (ppt) 0 – 33

**K 1.0**
- **accurate reading range**: 5μS/cm – 200,000+μS/cm
- TDS (ppm) 2 – 100,000
- Salinity (ppt) 0 – 42*
  *salinity scale cannot go any higher

**K 10**
- **accurate reading range**: 10μS/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.

<table>
<thead>
<tr>
<th>Conductivity Range</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.07 – 99.99</td>
<td>0.01μS/cm</td>
</tr>
<tr>
<td>100.1 – 999.9</td>
<td>0.1μS/cm</td>
</tr>
<tr>
<td>1,000 – 9,999</td>
<td>1.0μS/cm</td>
</tr>
<tr>
<td>10,000 – 99,990</td>
<td>10μS/cm</td>
</tr>
<tr>
<td>100,000 – 999,900</td>
<td>100μS/cm</td>
</tr>
</tbody>
</table>

![Diagram of EZO™ Conductivity circuit](image-url)
Operating principle

An E.C. (electrical conductivity) probe measures the electrical conductivity in a solution. It is commonly used in hydroponics, aquaculture and freshwater systems to monitor the amount of nutrients, salts or impurities in the water.

Inside the conductivity probe, two electrodes are positioned opposite from each other, an AC voltage is applied to the electrodes causing cations to move to the negatively charged electrode, while the anions move to the positively electrode. The more free electrolyte the liquid contains, the higher the electrical conductivity.
Output units

By default, EZO™ Conductivity circuits with firmware version 2.10 and above will only output EC. To enable these parameters see page 35 for UART, and 62 for I²C.

The EZO™ Conductivity circuit also has the capability to read:

- Conductivity = µS/cm
- Total dissolved solids = ppm
- Salinity = PSU (ppt) 0.00 – 42.00
- Specific gravity (sea water only) = 1.00 – 1.300

These parameters must be individually enabled within the device. See page 35 to enable each parameter in UART mode, and on page 62 for I²C mode.

Once these parameters have been enabled, output will be a CSV string.

Example
EC,TDS,SAL,SG

Default LED blink pattern

This is the LED pattern for Continuous Mode (default state)
This can only happen when the device is in UART mode.
Power and data isolation

The Atlas Scientific EZO™ Conductivity circuit is a very sensitive device. This sensitivity is what gives the Conductivity circuit its accuracy. This also means that the Conductivity circuit is capable of reading micro-voltages that are bleeding into the water from unnatural sources such as pumps, solenoid valves or other probes/sensors.

When electrical noise is interfering with the Conductivity readings it is common to see rapidly fluctuating readings or readings that are consistently off. To verify that electrical noise is causing inaccurate readings, place the Conductivity probe in a cup of water by itself. The readings should stabilize quickly, confirming that electrical noise was the issue.

When reading from two EZO™ Conductivity circuits, it is strongly recommended that they are electrically isolated from each other.

Correct

Incorrect

Without isolation, Conductivity readings will effect each other.
This schematic shows exactly how we isolate data and power using the ADM3260 and a few passive components. The ADM3260 can output isolated power up to 150 mW and incorporates two bidirectional data channels.

This technology works by using tiny transformers to induce the voltage across an air gap. PCB layout requires special attention for EMI/EMC and RF Control, having proper 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.
**Correct wiring**

- Bread board
- Bread board via USB
- Electrically Isolated EZO™ Carrier Board

**Incorrect wiring**

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

**Notes**

- Flux residue and shorting wires make it very hard to get accurate readings.
- *Only after you are familiar with EZO™ circuits operation*
The most important part of calibration is watching the readings during the calibration process.

It's easiest to calibrate the device in its default state (UART mode, with continuous readings enabled).

Switching the device to I^2C mode after calibration **will not** affect the stored calibration. If the device must be calibrated in I^2C mode be sure to **continuously request readings** so you can see the output from the probe.

1. **Pre-calibration setup**

Connect the dry conductivity probe and take continuous readings.

2. **Set probe type**

If your probe ≠ 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 33 or 60.
3. Dry calibration

Perform a dry calibration using the command "Cal,dry". Even though you may see reading 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

4. Single point or Two point calibration

No calibration

Single point calibration

Two point calibration

Narrow range of accuracy

Wide range of accuracy

Recommended calibration points

<table>
<thead>
<tr>
<th>K 0.1</th>
<th>K 1.0</th>
<th>K 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>84µS</td>
<td>12,880µS</td>
<td>12,880µS</td>
</tr>
<tr>
<td>1,413µS</td>
<td>80,000µS</td>
<td>150,000µS</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Temperature °F</th>
<th>Conductivity µS/cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>122</td>
<td>2,158</td>
</tr>
<tr>
<td>113</td>
<td>2,009</td>
</tr>
<tr>
<td>104</td>
<td>1,860</td>
</tr>
<tr>
<td>95</td>
<td>1,711</td>
</tr>
<tr>
<td>86</td>
<td>1,548</td>
</tr>
<tr>
<td>77</td>
<td>1,413</td>
</tr>
<tr>
<td>68</td>
<td>1,278</td>
</tr>
<tr>
<td>59</td>
<td>1,147</td>
</tr>
<tr>
<td>50</td>
<td>1,020</td>
</tr>
<tr>
<td>41</td>
<td>954</td>
</tr>
<tr>
<td>30</td>
<td>86</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Temperature °F</th>
<th>Conductivity µS/cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>122</td>
<td>2,158</td>
</tr>
<tr>
<td>113</td>
<td>2,009</td>
</tr>
<tr>
<td>104</td>
<td>1,860</td>
</tr>
<tr>
<td>95</td>
<td>1,711</td>
</tr>
<tr>
<td>86</td>
<td>1,548</td>
</tr>
<tr>
<td>77</td>
<td>1,413</td>
</tr>
<tr>
<td>68</td>
<td>1,278</td>
</tr>
<tr>
<td>59</td>
<td>1,147</td>
</tr>
<tr>
<td>50</td>
<td>1,020</td>
</tr>
<tr>
<td>41</td>
<td>954</td>
</tr>
<tr>
<td>30</td>
<td>86</td>
</tr>
</tbody>
</table>

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).

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 \(9,330\mu S\).

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

See pages 34 or 61 for more information.
Default state

UART mode

Baud
9,600

Readings
continuous

Units
μS/cm

Speed
1 reading per second

LED
on

1,000 ms

Green
Standby

Cyan
Taking reading

Transmitting
Available data protocols

UART

I²C

Unavailable data protocols

SPI
Analog
RS-485
Mod Bus
4–20mA
UART mode

**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 I\(^2\)C mode
- LED control
- Protocol lock
- Software switch to I\(^2\)C mode

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

- Find
- Sleep mode
- Temperature compensation
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

RX
Data in

TX
Data out

Vcc
3.3V – 5.5V

CPU

Data format

Reading
Conductivity = μS/cm
Total dissolved solids = ppm
Salinity = PSU (ppt) 0.00 – 42.00
Specific gravity (sea water only) = 1.00 – 1.300

Units
EC,TDS,SAL,SG

Terminator
carriage return

Data type
floating point

Decimal places
3

Smallest string
3 characters

Specific gravity

Largest string
40 characters

Encoding
ASCII

Format
string

Atlas Scientific
Environmental Robotics

Copyright © Atlas Scientific LLC
Receiving data from device

2 parts

**ASCII data string**

Command

**Carriage return <cr>**

Terminator

---

**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: Sleep <cr>
Hex: 53 6C 65 65 70 0D
Dec: 83 108 101 101 112 13
LED color definition

- **Green**: UART standby
- **Cyan**: Taking reading
- **Purple**: Changing baud rate
- **Red**: Command not understood
- **White**: Find

**LED ON**

- 5V +2.5 mA
- 3.3V +1 mA
# UART mode

## command quick reference

All commands are ASCII strings or single ASCII characters.

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Default state</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud</td>
<td>change baud rate</td>
<td>pg. 41</td>
<td>9,600</td>
</tr>
<tr>
<td>C</td>
<td>enable/disable continuous reading</td>
<td>pg. 27</td>
<td>enabled</td>
</tr>
<tr>
<td>Cal</td>
<td>performs calibration</td>
<td>pg. 29</td>
<td>n/a</td>
</tr>
<tr>
<td>Export</td>
<td>export calibration</td>
<td>pg. 31</td>
<td>n/a</td>
</tr>
<tr>
<td>Factory</td>
<td>enable factory reset</td>
<td>pg. 43</td>
<td>n/a</td>
</tr>
<tr>
<td>Find</td>
<td>finds device with blinking white LED</td>
<td>pg. 26</td>
<td>n/a</td>
</tr>
<tr>
<td>i</td>
<td>device information</td>
<td>pg. 37</td>
<td>n/a</td>
</tr>
<tr>
<td>I2C</td>
<td>change to I²C mode</td>
<td>pg. 44</td>
<td>not set</td>
</tr>
<tr>
<td>Import</td>
<td>import calibration</td>
<td>pg. 32</td>
<td>n/a</td>
</tr>
<tr>
<td>K</td>
<td>Set probe type</td>
<td>pg. 33</td>
<td>K 1.0</td>
</tr>
<tr>
<td>L</td>
<td>enable/disable LED</td>
<td>pg. 25</td>
<td>enabled</td>
</tr>
<tr>
<td>Name</td>
<td>set/show name of device</td>
<td>pg. 36</td>
<td>not set</td>
</tr>
<tr>
<td>O</td>
<td>enable/disable parameters</td>
<td>pg. 35</td>
<td>all enabled</td>
</tr>
<tr>
<td>Plock</td>
<td>enable/disable protocol lock</td>
<td>pg. 42</td>
<td>disabled</td>
</tr>
<tr>
<td>R</td>
<td>returns a single reading</td>
<td>pg. 28</td>
<td>n/a</td>
</tr>
<tr>
<td>Sleep</td>
<td>enter sleep mode/low power</td>
<td>pg. 40</td>
<td>n/a</td>
</tr>
<tr>
<td>Status</td>
<td>retrieve status information</td>
<td>pg. 39</td>
<td>enable</td>
</tr>
<tr>
<td>T</td>
<td>temperature compensation</td>
<td>pg. 34</td>
<td>25°C</td>
</tr>
<tr>
<td>TDS</td>
<td>change the TDS conversion factor</td>
<td>pg. 30</td>
<td>n/a</td>
</tr>
<tr>
<td>*OK</td>
<td>enable/disable response codes</td>
<td>pg. 38</td>
<td>enable</td>
</tr>
</tbody>
</table>
# LED control

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L,1</td>
<td>LED on (default)</td>
</tr>
<tr>
<td>L,0</td>
<td>LED off</td>
</tr>
<tr>
<td>L,?</td>
<td>LED state on/off?</td>
</tr>
</tbody>
</table>

## Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>L,1</td>
<td>*OK</td>
</tr>
<tr>
<td>L,0</td>
<td>*OK</td>
</tr>
<tr>
<td>L,?</td>
<td>?L,1 &lt;cr&gt; or ?L,0 &lt;cr&gt; *OK</td>
</tr>
</tbody>
</table>

## Diagrams

- **L,1**: LED on
- **L,0**: LED off

---

*Copyright © Atlas Scientific LLC*
Find

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

Example

Response

Find <cr> *OK <cr>

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

### Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C,1</td>
<td>enable continuous readings once per second</td>
</tr>
<tr>
<td>C,n</td>
<td>continuous readings every n seconds (n = 2 to 99 sec)</td>
</tr>
<tr>
<td>C,0</td>
<td>disable continuous readings</td>
</tr>
<tr>
<td>C,?</td>
<td>continuous reading mode on/off?</td>
</tr>
</tbody>
</table>

### Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
</table>
| C,1     | *OK  
EC,TDS,SAL,SG (1 sec)  
EC,TDS,SAL,SG (2 sec)  
EC,TDS,SAL,SG (3 sec)  |
| C,30    | *OK  
EC,TDS,SAL,SG (30 sec)  
EC,TDS,SAL,SG (60 sec)  
EC,TDS,SAL,SG (90 sec)  |
| C,0     | *OK  |
| C,?     | ?C,1 or ?C,0 or ?C,30  
*OK  |
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!**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal,dry</td>
<td>dry calibration</td>
</tr>
<tr>
<td>Cal,n</td>
<td>single point calibration, where n = any value</td>
</tr>
<tr>
<td>Cal,low,n</td>
<td>low end calibration, where n = any value</td>
</tr>
<tr>
<td>Cal,high,n</td>
<td>high end calibration, where n = any value</td>
</tr>
<tr>
<td>Cal,clear</td>
<td>delete calibration data</td>
</tr>
<tr>
<td>Cal,?</td>
<td>device calibrated?</td>
</tr>
</tbody>
</table>

**Example**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal,dry</td>
<td>*OK</td>
</tr>
<tr>
<td>Cal,84</td>
<td>*OK</td>
</tr>
<tr>
<td>Cal,low,12880</td>
<td>*OK</td>
</tr>
<tr>
<td>Cal,high,80000</td>
<td>*OK</td>
</tr>
<tr>
<td>Cal,clear</td>
<td>*OK</td>
</tr>
<tr>
<td>Cal,?</td>
<td>?CAL,0 or ?CAL,1 or ?CAL,2 *OK</td>
</tr>
</tbody>
</table>

**Example: One point calibration:**

Step 1. "cal,dry"
Step 2. "cal,n"
**Calibration complete!**

**Example: 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

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

Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDS,?</td>
<td>?TDS,0.54 &lt;cr&gt;</td>
</tr>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>R</td>
<td>EC TDS</td>
</tr>
<tr>
<td></td>
<td>100,54 &lt;cr&gt;</td>
</tr>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>TDS,0.46</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>R</td>
<td>EC TDS</td>
</tr>
<tr>
<td></td>
<td>100,460 &lt;cr&gt;</td>
</tr>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

Common conversion factors

<table>
<thead>
<tr>
<th>Substance</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>NaCl</td>
<td>0.47 – 0.50</td>
</tr>
<tr>
<td>KCL</td>
<td>0.50 - 0.57</td>
</tr>
<tr>
<td>&quot;442&quot;</td>
<td>0.65 – 0.85</td>
</tr>
</tbody>
</table>

Formula

EC x conversion factor = TDS
# Export calibration

## Command syntax

- **Export,?**  <cr>  calibration string info  
- **Export**  <cr>  export calibration string from calibrated device

## Example

<table>
<thead>
<tr>
<th>Export,?  &lt;cr&gt;</th>
<th>10,120  &lt;cr&gt;</th>
</tr>
</thead>
</table>

**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>.

<table>
<thead>
<tr>
<th>Export  &lt;cr&gt;</th>
<th>59 6F 75 20 61 72  &lt;cr&gt; (1 of 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export  &lt;cr&gt;</td>
<td>65 20 61 20 63 6F  &lt;cr&gt; (2 of 10)</td>
</tr>
<tr>
<td>(7 more)</td>
<td>:</td>
</tr>
<tr>
<td>Export  &lt;cr&gt;</td>
<td>6F 6C 20 67 75 79  &lt;cr&gt; (10 of 10)</td>
</tr>
</tbody>
</table>

*DONE  

Disabling *OK simplifies this process.
# Import calibration

## Command syntax

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

## Example

<table>
<thead>
<tr>
<th>Import</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import, 59 6F 75 20 61 72 &lt;cr&gt;</td>
<td>(1 of 10) *OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Import, 65 20 61 20 63 6F &lt;cr&gt;</td>
<td>(2 of 10) *OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Import, 6F 6C 20 67 75 79 &lt;cr&gt;</td>
<td>(10 of 10) *OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

* 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, n** <br>
  n = any value; floating point in ASCII

- **K, ?** <br>
  probe K value?

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>K, 10</td>
<td>*OK</td>
</tr>
<tr>
<td>K, ?</td>
<td>?K, 10</td>
</tr>
<tr>
<td></td>
<td>*OK</td>
</tr>
</tbody>
</table>

### Example response

- **K 0.1**
- **K 1.0**
- **K 10**

K 1.0 is the default value
Temperature compensation

Command syntax

- **T,n**  \(<\text{cr}>\)  \(n = \text{any value; floating point or int}\)
- **T,?**  \(<\text{cr}>\)  \(\text{compensated temperature value?}\)
- **RT,n**  \(<\text{cr}>\)  \(\text{set temperature compensation and take a reading}\)\(^*\)

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

* This is a new command for firmware V2.13

### Example

<table>
<thead>
<tr>
<th>T,19.5  (&lt;\text{cr}&gt;)</th>
<th>*OK  (&lt;\text{cr}&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT,19.5  (&lt;\text{cr}&gt;)</td>
<td><em>OK(^</em>) 8.91 (&lt;\text{cr}&gt;)</td>
</tr>
</tbody>
</table>
| T,?  \(<\text{cr}>\) | ?T,19.5  \(<\text{cr}>\)  
\*OK  \(<\text{cr}>\) |

\(^*\) This is a new command for firmware V2.13

---

Copyright © Atlas Scientific LLC
# Enable/disable parameters from output string

## Command syntax

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>O, [parameter],[1,0] &lt;CR&gt;</code></td>
<td>enable or disable output parameter</td>
</tr>
<tr>
<td><code>O,? &lt;CR&gt;</code></td>
<td>enabled parameter?</td>
</tr>
</tbody>
</table>

## Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>O,EC,1 / O,EC,0 &lt;CR&gt;</code></td>
<td>*OK &lt;CR&gt; enable / disable conductivity</td>
</tr>
<tr>
<td><code>O,TDS,1 / O,TDS,0 &lt;CR&gt;</code></td>
<td>*OK &lt;CR&gt; enable / disable total dissolved solids</td>
</tr>
<tr>
<td><code>O,S,1 / O,S,0 &lt;CR&gt;</code></td>
<td>*OK &lt;CR&gt; enable / disable salinity</td>
</tr>
<tr>
<td><code>O,SG,1 / O,SG,0 &lt;CR&gt;</code></td>
<td>*OK &lt;CR&gt; enable / disable specific gravity</td>
</tr>
<tr>
<td><code>O,? &lt;CR&gt;</code></td>
<td>?.O,EC,TDS,S,SG &lt;CR&gt; if all are enabled</td>
</tr>
</tbody>
</table>

## Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC</td>
<td>conductivity</td>
</tr>
<tr>
<td>TDS</td>
<td>total dissolved solids</td>
</tr>
<tr>
<td>S</td>
<td>salinity</td>
</tr>
<tr>
<td>SG</td>
<td>specific gravity</td>
</tr>
</tbody>
</table>

Followed by 1 or 0

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enabled</td>
</tr>
<tr>
<td>0</td>
<td>disabled</td>
</tr>
</tbody>
</table>

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

---

Copyright © Atlas Scientific LLC
Naming device

Command syntax

<table>
<thead>
<tr>
<th>Name, n</th>
<th>set name</th>
<th>n =</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name, ?</td>
<td>show name</td>
<td></td>
</tr>
</tbody>
</table>

Do not use spaces in the name

Up to 16 ASCII characters

Example

<table>
<thead>
<tr>
<th>Name, zzt</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name, zzt</td>
<td>*OK</td>
</tr>
<tr>
<td>Name, ?</td>
<td>?Name, zzt *OK</td>
</tr>
</tbody>
</table>

Example diagrams:

- Name, zzt
  - Initial: *OK
  - Final: Name, zzt *OK

- Name, ?
  - Initial: *OK
  - Final: Name, zzt *OK
## Device information

### Command syntax

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

### Example

<table>
<thead>
<tr>
<th>i &lt;cr&gt;</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>i &lt;cr&gt;</td>
<td>?i,EC,2.10 &lt;cr&gt;</td>
</tr>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

### Response breakdown

```
?i, EC, 2.10
```

<table>
<thead>
<tr>
<th>?i, EC, 2.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
</tr>
<tr>
<td>2.10</td>
</tr>
<tr>
<td>Firmware</td>
</tr>
</tbody>
</table>
# Response codes

## Command syntax

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*OK,1</td>
<td>enable response</td>
</tr>
<tr>
<td>*OK,0</td>
<td>disable response</td>
</tr>
<tr>
<td>*OK,?</td>
<td>response on/off?</td>
</tr>
</tbody>
</table>

**default**

## Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>1,413</td>
</tr>
<tr>
<td></td>
<td>*OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>*OK,0</th>
<th>no response, *OK disabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>1,413 *OK disabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>*OK,?</th>
<th>?*OK,1 or ?*OK,0</th>
</tr>
</thead>
</table>

## Other response codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*ER</td>
<td>unknown command</td>
</tr>
<tr>
<td>*OV</td>
<td>over volt (VCC&gt;=5.5V)</td>
</tr>
<tr>
<td>*UV</td>
<td>under volt (VCC&lt;=3.1V)</td>
</tr>
<tr>
<td>*RS</td>
<td>reset</td>
</tr>
<tr>
<td>*RE</td>
<td>boot up complete, ready</td>
</tr>
<tr>
<td>*SL</td>
<td>entering sleep mode</td>
</tr>
<tr>
<td>*WA</td>
<td>wake up</td>
</tr>
</tbody>
</table>

These response codes cannot be disabled.
Reading device status

Command syntax

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

Example

<table>
<thead>
<tr>
<th>Status &lt;cr&gt;</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>?Status,P,5.038 &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

Response breakdown

?Status, P, 5.038

Reason for restart Voltage at Vcc

Restart codes

<table>
<thead>
<tr>
<th>P</th>
<th>powered off</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>software reset</td>
</tr>
<tr>
<td>B</td>
<td>brown out</td>
</tr>
<tr>
<td>W</td>
<td>watchdog</td>
</tr>
<tr>
<td>U</td>
<td>unknown</td>
</tr>
</tbody>
</table>
# 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

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Standby Current</th>
<th>Sleep Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>18.14 mA</td>
<td>0.7 mA</td>
</tr>
<tr>
<td>3.3V</td>
<td>16.85 mA</td>
<td>0.4 mA</td>
</tr>
</tbody>
</table>

Sleep mode:

- **Standby**: 18.14 mA
- **Sleep**: 0.7 mA

[Image of circuit board showing transition from Standby to Sleep mode]
Change baud rate

Command syntax

Baud, n  <cr>  change baud rate

Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud, 38400</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Baud, ?</td>
<td>?Baud, 38400</td>
</tr>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

Example:

- **Baud, 38400**
  - Response: *OK

Example:

- **Baud, ?**
  - Response: *OK

Default baud rates:

- 300
- 1200
- 2400
- 9600 (default)
- 19200
- 38400
- 57600
- 115200

*(reboot)*

Example:

- **Baud, 38400**
  - Response: *OK

* Ok

**Command syntax**

Baud, n  <cr>  change baud rate

Example:

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud, 38400</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Baud, ?</td>
<td>?Baud, 38400</td>
</tr>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

Example:

- **Baud, 38400**
  - Response: *OK

Example:

- **Baud, ?**
  - Response: *OK

Default baud rates:

- 300
- 1200
- 2400
- 9600 (default)
- 19200
- 38400
- 57600
- 115200

*(reboot)*
Protocol lock

Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plock,1</td>
<td>enable Plock</td>
</tr>
<tr>
<td>Plock,0</td>
<td>disable Plock (default)</td>
</tr>
<tr>
<td>Plock,?</td>
<td>Plock on/off?</td>
</tr>
</tbody>
</table>

Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plock,1</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Plock,0</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Plock,?</td>
<td>?Plock,1 &lt;cr&gt; or ?Plock,0 &lt;cr&gt;</td>
</tr>
</tbody>
</table>

Example

- **Plock,1**
  - Locks device to UART mode.
  - *OK <cr>

- **I2C,100**
  - cannot change to I²C
  - *ER <cr>

- **Short**
  - cannot change to I²C
Factory reset

Command syntax

Factory <cr> enable factory reset

Example  Response

Factory <cr>  *OK <cr>

(reboot)

*OK <cr>  

*RS <cr>  

*RE <cr>  

Baud rate will not change
Change to I²C mode

Command syntax

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

Disconnected RX line
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**

**Vcc**  
3.3V – 5.5V

**Clock speed**  
100 – 400 kHz

**SDA**  

**SCL**  

**Data format**

**Reading**  
Conductivity = μS/cm  
Total dissolved solids = ppm  
Salinity = PSU (ppt) 0.00 – 42.00  
Specific gravity (sea water only) = 1.00 – 1.300

**Units**  
EC, TDS, SAL, SG

**Encoding**  
ASCII

**Format**  
string

**Data type**  
floating point

**Decimal places**  
3

**Smallest string**  
3 characters

**Largest string**  
40 characters
Sending commands to device

5 parts

Start  I²C address  Write  Command (not case sensitive)  Stop

ASCII command string

Example

Start  100 (0x64)  Write  Sleep  Stop

I²C address  Command

Advanced

Address bits  The entire command as ASCII with all arguments

Start  A6  A5  A4  A3  A2  A1  A0  W  ACK  First letter of command  ACK  ...  Last letter of command  ACK  Stop

W = low
Requesting data from device

7 parts

Start | I²C address | Read | Response code | Data string | Null | Stop

100 (0x64) | 1 byte | "1,413" | Terminator (Dec 0)

Advanced

Address bits | N bytes of data | All bytes after data are Null

Start | ACK | Response code | ACK | Data | ACK | Data | ACK | Data | ACK | Null | ACK | Null | … | NACK | Stop

1 49 44 52 49 51 0 = 1,413

Dec  Dec

ASCII
## 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

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

delay(300);
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.

<table>
<thead>
<tr>
<th>Response codes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>no data to send</td>
</tr>
<tr>
<td>254</td>
<td>still processing, not ready</td>
</tr>
<tr>
<td>2</td>
<td>syntax error</td>
</tr>
<tr>
<td>1</td>
<td>successful request</td>
</tr>
</tbody>
</table>

---

**CPU**

- **SCL**: Clock line
- **SDA**: Data line

**I2C**: Inter-Integrated Circuit

**Processing delay**

- **Send command**: CPU sends command to I2C device.
- **Receiving data**: CPU receives response code from I2C device.

---

**Copyright © Atlas Scientific LLC**
LED color definition

Blue
I²C standby

Green
Taking reading

Purple
Changing I²C address

Red
Command not understood

White
Find

<table>
<thead>
<tr>
<th>Voltage</th>
<th>LED ON</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>+2.5 mA</td>
<td></td>
</tr>
<tr>
<td>3.3V</td>
<td>+1 mA</td>
<td></td>
</tr>
</tbody>
</table>
## I²C mode command quick reference

All commands are ASCII strings or single ASCII characters.

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud</td>
<td>switch back to UART mode</td>
<td>70</td>
</tr>
<tr>
<td>Cal</td>
<td>performs calibration</td>
<td>56</td>
</tr>
<tr>
<td>Export</td>
<td>export calibration</td>
<td>58</td>
</tr>
<tr>
<td>Factory</td>
<td>enable factory reset</td>
<td>69</td>
</tr>
<tr>
<td>Find</td>
<td>finds device with blinking white LED</td>
<td>54</td>
</tr>
<tr>
<td>i</td>
<td>device information</td>
<td>64</td>
</tr>
<tr>
<td>I²C</td>
<td>change I²C address</td>
<td>68</td>
</tr>
<tr>
<td>Import</td>
<td>import calibration</td>
<td>59</td>
</tr>
<tr>
<td>K</td>
<td>set probe type</td>
<td>60</td>
</tr>
<tr>
<td>L</td>
<td>enable/disable LED</td>
<td>53</td>
</tr>
<tr>
<td>Name</td>
<td>set/show name of device</td>
<td>63</td>
</tr>
<tr>
<td>O</td>
<td>enable/disable parameters</td>
<td>62</td>
</tr>
<tr>
<td>Plock</td>
<td>enable/disable protocol lock</td>
<td>67</td>
</tr>
<tr>
<td>R</td>
<td>returns a single reading</td>
<td>55</td>
</tr>
<tr>
<td>Sleep</td>
<td>enter sleep mode/low power</td>
<td>66</td>
</tr>
<tr>
<td>Status</td>
<td>retrieve status information</td>
<td>65</td>
</tr>
<tr>
<td>T</td>
<td>temperature compensation</td>
<td>61</td>
</tr>
<tr>
<td>TDS</td>
<td>change the TDS conversion factor</td>
<td>57</td>
</tr>
</tbody>
</table>
# LED control

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L,1</td>
<td>LED on (default)</td>
</tr>
<tr>
<td>L,0</td>
<td>LED off</td>
</tr>
<tr>
<td>L,?</td>
<td>LED state on/off?</td>
</tr>
</tbody>
</table>

### 300ms processing delay

## Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>L,1</td>
<td>1 0 Dec Null</td>
</tr>
<tr>
<td>L,0</td>
<td>1 0 Dec Null</td>
</tr>
<tr>
<td>L,?</td>
<td>1 ?L,1 0 Dec ASCII Null or 1 ?L,0 0 Dec ASCII Null</td>
</tr>
</tbody>
</table>

### Example diagrams

- **L,1**
- **L,0**
## Find

### Command syntax

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

### Example

**Find**

LED rapidly blinks white, used to help find device

### Response

<table>
<thead>
<tr>
<th>Find</th>
<th>1</th>
<th>0</th>
<th>Null</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Wait 300ms</strong></td>
<td><strong>Dec</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Send any character or command to terminate find.

300ms processing delay

---

Copyright © Atlas Scientific LLC
**Command syntax**

| R | return 1 reading |

**Example**

<table>
<thead>
<tr>
<th>R</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>![Timer](Wait 600ms) 1 1,413 0 Dec ASCII Null</td>
</tr>
</tbody>
</table>

**Response**

- **R**
- **Wait 600ms**
- **Green**
  - Taking reading
- **Transmitting**
- **Blue**
  - Standby

---

**Processing delay**

- **600ms**
## Calibration

### Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal,dry</td>
<td>dry calibration</td>
</tr>
<tr>
<td>Cal,n</td>
<td>single point calibration, where n = any value</td>
</tr>
<tr>
<td>Cal,low,n</td>
<td>low end calibration, where n = any value</td>
</tr>
<tr>
<td>Cal,high,n</td>
<td>high end calibration, where n = any value</td>
</tr>
<tr>
<td>Cal,clear</td>
<td>delete calibration data</td>
</tr>
<tr>
<td>Cal,?</td>
<td>device calibrated?</td>
</tr>
</tbody>
</table>

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal,dry</td>
<td>[\text{Wait 600ms}] 1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,84</td>
<td>[\text{Wait 600ms}] 1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,low,12880</td>
<td>[\text{Wait 600ms}] 1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,high,80000</td>
<td>[\text{Wait 600ms}] 1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,clear</td>
<td>[\text{Wait 300ms}] 1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,?</td>
<td>[\text{Wait 300ms}] 1 ASCII 0 or 1 ASCII 1 or 1 ASCII 2 0 Null</td>
</tr>
</tbody>
</table>

### 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!**

---

Copyright © Atlas Scientific LLC
### Changing the TDS (ppm) conversion factor

#### Command syntax

- **TDS,n**: set custom conversion factor, \( n = \) any value between 0.01 – 1.00
- **TDS,?**: conversion factor being used

#### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDS,?</td>
<td><img src="EC" alt="Wait 300ms" /> 1 ?TDS,0.54 0 Dec ASCII Null</td>
</tr>
<tr>
<td>R</td>
<td><img src="EC" alt="Wait 300ms" /> 1 100,54 0 Dec ASCII Null</td>
</tr>
<tr>
<td>TDS,0.46</td>
<td><img src="EC" alt="Wait 300ms" /> 1 0 Dec Null</td>
</tr>
<tr>
<td>R</td>
<td><img src="EC" alt="Wait 300ms" /> 1 100,460 0 Dec ASCII Null</td>
</tr>
</tbody>
</table>

#### Common conversion factors

- **NaCl**: 0.47 – 0.50
- **KCL**: 0.50 - 0.57
- **“442”**: 0.65 – 0.85

#### Formula

\[
\text{EC} \times \text{conversion factor} = \text{TDS}
\]
# Export calibration

## Command syntax

**Export,**?  
calibration string info

**Export**  
export calibration string from calibrated device

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
</table>
| Export,? | Wait 300ms  
1  
10,120  
0  
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  
59 6F 75 20 61 72 6F 6C 20 67 75 79  |
| Export | Wait 300ms  
1  
65 20 61 20 63 6F 6C 20 67 75 79  |
| Export | Wait 300ms  
1  
6F 6C 20 67 75 79  |
| Export | Wait 300ms  
1  
*DONE*  
0  |
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

Wait 300ms  1 0
Dec  Null

System will 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

- **K,n**  
  - n = any value; floating point in ASCII
- **K,?**  
  - probe K value?

### 300ms processing delay

- K 1.0 is the default value

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>K,10</td>
<td>1 0 Dec Null</td>
</tr>
<tr>
<td>K,?</td>
<td>1 K,10 0 ASCII Null</td>
</tr>
</tbody>
</table>
# Temperature Compensation

## Command Syntax

- **T,n**: n = any value; floating point or int
- **T,?**: compensated temperature value?
- **RT,n**: set temperature compensation and take a reading*

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>T,19.5</td>
<td><img src="image" alt="Wait 300ms" /></td>
</tr>
<tr>
<td>RT,19.5</td>
<td><img src="image" alt="Wait 900ms" /></td>
</tr>
<tr>
<td>T,?</td>
<td><img src="image" alt="Wait 300ms" /></td>
</tr>
</tbody>
</table>

* This is a new command for firmware V2.13

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

<sup>© Atlas Scientific LLC 2023</sup>
Enable/disable parameters from output string

**Command syntax**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>O, [parameter],[1,0]</code></td>
<td>enable or disable output parameter</td>
</tr>
<tr>
<td><code>O,?</code></td>
<td>enabled parameter?</td>
</tr>
</tbody>
</table>

**Example**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>O,EC,1</code> / <code>O,EC,0</code></td>
<td>enable / disable conductivity</td>
</tr>
<tr>
<td><code>O,TDS,1</code> / <code>O,TDS,0</code></td>
<td>enable / disable total dissolved solids</td>
</tr>
<tr>
<td><code>O,S,1</code> / <code>O,S,0</code></td>
<td>enable / disable salinity</td>
</tr>
<tr>
<td><code>O,SG,1</code> / <code>O,SG,0</code></td>
<td>enable / disable specific gravity</td>
</tr>
<tr>
<td><code>O,?</code></td>
<td>if all are enabled</td>
</tr>
</tbody>
</table>

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC</td>
<td>conductivity</td>
</tr>
<tr>
<td>TDS</td>
<td>total dissolved solids</td>
</tr>
<tr>
<td>S</td>
<td>salinity</td>
</tr>
<tr>
<td>SG</td>
<td>specific gravity</td>
</tr>
</tbody>
</table>

Followed by 1 or 0

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enabled</td>
</tr>
<tr>
<td>0</td>
<td>disabled</td>
</tr>
</tbody>
</table>

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

---

© Atlas Scientific LLC

62 Copyright © Atlas Scientific LLC
### Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name, n</td>
<td>Set name</td>
</tr>
<tr>
<td>Name, ?</td>
<td>Show name</td>
</tr>
</tbody>
</table>

**n =**

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

**Up to 16 ASCII characters**

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name, zzt</td>
<td><img src="image1.png" alt="Image of device" /></td>
</tr>
<tr>
<td>Name, ?</td>
<td><img src="image2.png" alt="Image of device" /></td>
</tr>
</tbody>
</table>

**Response**

1 Dec 0

Wait 300ms

1 Dec ASCII 0

Wait 300ms
# Device information

## Command syntax

<table>
<thead>
<tr>
<th>i</th>
<th>device information</th>
</tr>
</thead>
</table>

### Example

<table>
<thead>
<tr>
<th>i</th>
<th>Wait 300ms</th>
</tr>
</thead>
</table>

### Response

| 1 | EC, ASCII, 2.10 | 0 | Null |

## Response breakdown

<table>
<thead>
<tr>
<th>?i, EC, 2.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
</tr>
<tr>
<td>Firmware</td>
</tr>
</tbody>
</table>
# Reading device status

## Command syntax

Wait 300ms

**Status**  
Voltage at Vcc pin and reason for last restart

### Example

<table>
<thead>
<tr>
<th>Status</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>?Status, P, 5.038</td>
<td>0</td>
</tr>
</tbody>
</table>

### 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
Any command | wakes up device

<table>
<thead>
<tr>
<th>5V</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18.14 mA</td>
<td>0.7 mA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3.3V</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16.85 mA</td>
<td>0.4 mA</td>
</tr>
</tbody>
</table>

Copyright © Atlas Scientific LLC
# Protocol lock

## Command syntax

- **Plock,1**: Enable Plock
- **Plock,0**: Disable Plock (default)
- **Plock,?**: Plock on/off?

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plock,1</td>
<td><img src="image" alt="300ms processing delay" /></td>
</tr>
<tr>
<td></td>
<td>Dec</td>
</tr>
<tr>
<td>Plock,0</td>
<td><img src="image" alt="300ms processing delay" /></td>
</tr>
<tr>
<td></td>
<td>Dec</td>
</tr>
<tr>
<td>Plock,?</td>
<td><img src="image" alt="300ms processing delay" /></td>
</tr>
<tr>
<td></td>
<td>Dec</td>
</tr>
</tbody>
</table>

**Example**

- **Plock,1**: Enables Plock
- **Plock,0**: Disables Plock
- **Plock,?**: Checks Plock status

**Response**

- **Wait 300ms**
- **1**: Dec
- **0**: Null
- **1**: Dec
- **0**: ASCII
- **?Plock,1**: ASCII
- **0**: Null

**Cannot change to UART**
I²C address change

Command syntax

I²C,n  sets I²C address and reboots into I²C mode

Example  

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>I²C,101</td>
<td>device reboot</td>
</tr>
</tbody>
</table>

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
## Factory reset

### Command syntax

Factory enable factory reset

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

### Example

<table>
<thead>
<tr>
<th>Factory</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory</td>
<td>device reboot</td>
</tr>
</tbody>
</table>

Clears calibration
LED on
Response codes enabled

---

### Factory

(reboot)
# Change to UART mode

## Command syntax

**Baud,n** switch from I²C to UART

| n = | 300 | 1200 | 2400 | 9600 | 19200 | 38400 | 57600 | 115200 |

## Example

| Baud,9600 | reboot in UART mode |

## Response

- **Baud,9600**
- **(reboot)**

Changing to UART mode

---

Copyright © Atlas Scientific LLC
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

Disconnect RX line
**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 6.2

Added new command:
"TDS,n" Changing the TDS (ppm) conversion factor on pages 30 (UART) & 57 (I^2C).

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 (I^2C).
Added firmware information to Firmware update list.

Datasheet V 5.2

Revised calibration information on pages 27 & 52.
Datasheet change log

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 I\(^2\)C 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 I\(^2\)C 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
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.