EZO™
Dissolved Oxygen

Circuit

Reads

Dissolved Oxygen
0.01 – 100+ mg/L
0.1 – 400+ % saturation

+/− 0.05 mg/L

Accuracy

Max rate

1 reading per sec

Supported probes

Any galvanic probe

Calibration

1 or 2 point

Yes

Temperature, salinity and pressure compensation

Data protocol

UART & I²C

97 (0x61)

Default I²C address

3.3V – 5V

Operating voltage

ASCII

Data format
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!
**EZO™ circuit dimensions**

Power consumption

<table>
<thead>
<tr>
<th>5V</th>
<th>LED</th>
<th>MAX</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>13.5 mA</td>
<td>13.1 mA</td>
<td>0.66 mA</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>12.7 mA</td>
<td>12.7 mA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3.3V</th>
<th>LED</th>
<th>MAX</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>12.1 mA</td>
<td>12 mA</td>
<td>0.3 mA</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>11.9 mA</td>
<td>11.9 mA</td>
<td></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 (EZO™ D.O.)</td>
<td>-65 °C</td>
<td></td>
<td>125 °C</td>
</tr>
<tr>
<td>Operational temperature (EZO™ D.O.)</td>
<td>-40 °C</td>
<td>25 °C</td>
<td>85 °C</td>
</tr>
<tr>
<td>VCC</td>
<td>3.3V</td>
<td>5V</td>
<td>5.5V</td>
</tr>
</tbody>
</table>
EZO™ Dissolved Oxygen circuit

Legacy Dissolved Oxygen circuit

Viewing correct datasheet

Viewing incorrect datasheet

Click here to view legacy datasheet
Operating principle

A galvanic dissolved oxygen probe consists of a Polytetrafluoroethylene membrane, an anode bathed in an electrolyte and a cathode. Oxygen molecules diffuse through the probes membrane at a constant rate (without the membrane the reaction happens quickly). Once the oxygen molecules have crossed the membrane they are reduced at the cathode and a small voltage is produced. If no oxygen molecules are present, the probe will output 0 mV. As the oxygen increases so does the mV output from the probe. Each probe will output a different voltage in the presence of oxygen. The only thing that is constant is that \(0 \text{mV} = 0 \text{Oxygen}\). (A galvanic dissolved oxygen probe can also be used to detect the Oxygen content in gases).

The Atlas Scientific™ E2O™ Dissolved Oxygen circuit works with:

- **Optical probe**
  Slow response, requires external power, expensive.

- **Polar Graphic probe**
  Requires external power, output in \(\mu\text{A}\).

- **Galvanic probe**
  Requires no external power, output in mV.

Flow Dependence

One of the drawbacks from using a galvanic probe is that it consumes a VERY small amount of the oxygen it reads. Therefore, a small amount of water movement is necessary to take accurate readings. Approximately 60 ml/min.
Calibration theory

**Calibrate first, compensate later.**
Temperature, salinity and pressure compensation values have no effect on calibration.

The Atlas Scientific EZO™ Dissolved Oxygen circuit, has a flexible calibration protocol, allowing for *single point* or *dual point* calibration.

**Single point calibration**

1. Pull off and discard cap from the Dissolved Oxygen probe. (*only used to protect probe during shipping*)
2. Let the Dissolved Oxygen probe sit, exposed to air until readings stabilize (5–30 sec).
3. Calibrate using the command "Cal".
4. After calibration is complete, you should see readings ~9.09 – 9.1Xmg/L.
   (*only if temperature, salinity and pressure compensation are at default values*)

**Dual point calibration (optional)**

*Only perform this calibration if you require accurate readings below 1.0 mg/L*

After you have calibrated using the command "Cal"

1. Stir probe in Zero D.O. calibration solution to remove trapped air, (*which could cause readings to go high*).
2. Let the probe sit in Zero D.O. calibration solution until readings stabilize (0:10 – 1:30).
3. Calibrate using the command "Cal,0".
How to preserve the Zero D.O. calibration solution

Oxygen is everywhere. The Zero D.O. calibration solution has been designed to chemically absorb oxygen. Once the bottle has been opened the test solution has been exposed to oxygen and will slowly stop working.

Inside each bottle of the calibration solution is a small amount of nitrogen gas that helps displace oxygen out of the bottle during the filling process. When the Dissolved Oxygen probe is removed from the bottle, oxygen will enter the bottle and begin to dissolve into the solution.

In order slow down this process, fill the void space of the bottle with any gas (other than oxygen) to preserve the calibration solution. Gas from a lighter works great if other gases are currently unobtainable.
Power and data isolation

The Atlas Scientific EZO™ Dissolved Oxygen circuit is a very sensitive device. This sensitivity is what gives the Dissolved Oxygen circuit its accuracy. This also means that the Dissolved Oxygen 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 Dissolved Oxygen 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 Dissolved Oxygen probe in a cup of water by itself. The readings should stabilize quickly, confirming that electrical noise was the issue.

When reading Dissolved Oxygen and Conductivity together, it is strongly recommended that the EZO™ Dissolved Oxygen circuit is electrically isolated from the EZO™ Conductivity circuit.

Correct

Incorrect

Basic EZO™ Inline Voltage Isolator

Without isolation, Conductivity readings will effect Dissolved Oxygen accuracy.
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 4.7k\( \Omega \) 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 3.7V 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
- Extended leads
- Sloppy setup
- Perfboards or Protoboards
- *Embedded into your device

Bread board via USB

Carrier board

USB carrier board

Incorrect wiring

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

NEVER use Perfboards or Protoboards

*Only after you are familiar with EZO™ circuits operation
Available data protocols

UART

Available data protocols

\( I^2C \)

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 I2C mode
- LED control
- Protocol lock
- Software switch to I2C mode

**Settings that are **NOT** retained if power is cut**
- Find
- Pressure compensation
- Salinity compensation
- Sleep mode
- Temperature compensation
### UART mode

- **Data format**
  - **Reading**: D.O.
  - **Units**: mg/L & (% sat)
    - When enabled
  - **Encoding**: ASCII
  - **Format**: string
    - (CSV string when % sat is enabled)
  - **Terminator**: carriage return
  - **Data type**: floating point
  - **mg/L = 2**
  - **% sat = 1**
  - **Decimal places**: 4 characters
  - **Smallest string**: 4 characters
  - **Largest string**: 16 characters

- **Vcc**: 3.3V – 5.5V
- **RX**: Data in
- **TX**: Data out
- **Baud**: 300, 1,200, 2,400, 9,600 default, 19,200, 38,400, 57,600, 115,200

---

**Note:**
- UART mode supports 8 data bits, no parity, 1 stop bit, and no flow control.
- Baud rates include 300, 1,200, 2,400, 9,600 default, 19,200, 38,400, 57,600, 115,200.
- Power supply voltage range is 3.3V – 5.5V.
- Data transmission is bidirectional with TX and RX connections.
- ASCII encoding ensures compatibility with standard text processing.
- String format allows for flexible data representation, especially with CSV for % saturation data.
- Carriage return is used as the terminator for line formatting in data output.
Default state

Mode: UART
Baud: 9,600
Readings: continuous
Speed: 1 reading per second
Temperature compensation: 20 °C
Salinity compensation: 0 (Fresh water)
Pressure compensation: 101.3 kPa (Sea level)
LED: on

1,000 ms

Green
Standby

Cyan
Taking reading

Transmitting
Receiving data from device

2 parts

ASCII data string
Command

Carriage return <cr>
Terminator

Advanced

ASCII: 7 . 8 2 <cr>
Hex: 37 2E 38 32 0D
Dec: 55 46 56 50 13
Sending commands to device

2 parts

Command (not case sensitive)
ASCII data string

Carriage return <cr>
Terminator

CPU
TX
RX
 Sender
Receiver

Advanced

ASCII: Slleep <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

5V  LED ON
    +0.4 mA

3.3V +0.2 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>
</tr>
</thead>
<tbody>
<tr>
<td>Baud</td>
<td>change baud rate</td>
<td>pg. 35 9,600</td>
</tr>
<tr>
<td>C</td>
<td>enable/disable continuous reading</td>
<td>pg. 22 enabled</td>
</tr>
<tr>
<td>Cal</td>
<td>performs calibration</td>
<td>pg. 24 n/a</td>
</tr>
<tr>
<td>Export/import</td>
<td>export/import calibration</td>
<td>pg. 25 n/a</td>
</tr>
<tr>
<td>Factory</td>
<td>enable factory reset</td>
<td>pg. 37 n/a</td>
</tr>
<tr>
<td>Find</td>
<td>finds device with blinking white LED</td>
<td>pg. 21 n/a</td>
</tr>
<tr>
<td>i</td>
<td>device information</td>
<td>pg. 31 n/a</td>
</tr>
<tr>
<td>I2C</td>
<td>change to I²C mode</td>
<td>pg. 38 not set</td>
</tr>
<tr>
<td>L</td>
<td>enable/disable LED</td>
<td>pg. 20 enabled</td>
</tr>
<tr>
<td>Name</td>
<td>set/show name of device</td>
<td>pg. 30 not set</td>
</tr>
<tr>
<td>O</td>
<td>enable/disable parameters</td>
<td>pg. 29 mg/L</td>
</tr>
<tr>
<td>P</td>
<td>pressure compensation</td>
<td>pg. 28 101.3 kPa</td>
</tr>
<tr>
<td>Plock</td>
<td>enable/disable protocol lock</td>
<td>pg. 36 disabled</td>
</tr>
<tr>
<td>R</td>
<td>returns a single reading</td>
<td>pg. 23 n/a</td>
</tr>
<tr>
<td>S</td>
<td>salinity compensation</td>
<td>pg. 27 n/a</td>
</tr>
<tr>
<td>Sleep</td>
<td>enter sleep mode/low power</td>
<td>pg. 34 n/a</td>
</tr>
<tr>
<td>Status</td>
<td>retrieve status information</td>
<td>pg. 33 n/a</td>
</tr>
<tr>
<td>T</td>
<td>temperature compensation</td>
<td>pg. 26 20°C</td>
</tr>
<tr>
<td>*OK</td>
<td>enable/disable response codes</td>
<td>pg. 32 enable</td>
</tr>
</tbody>
</table>
LED control

Command syntax

L,1  <cr>  LED on  default
L,0  <cr>  LED off
L,?  <cr>  LED state on/off?

Example

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

*This command is only available for firmware version 2.10 and above.

Example  |  Response
---|---
Find  <cr>  |  *OK  <cr>

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*

*This command is only available for firmware version 2.10 and above.
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?

```
*OK
```

Example Response

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
</table>
| **C,1 <cr>** | *OK <cr>  
DO (1 sec) <cr>  
DO (2 sec) <cr>  
DO (3 sec) <cr> |
| **C,30 <cr>** | *OK <cr>  
DO (30 sec) <cr>  
DO (60 sec) <cr>  
DO (90 sec) <cr> |
| **C,0 <cr>** | *OK <cr> |
| **C,? <cr>** | ?C,1 <cr> or ?C,0 <cr> or ?C,30 <cr>  
*OK <cr> |

*This command is only available for firmware version 2.10 and above.
Single reading mode

Command syntax

R <cr> takes single reading

Example | Response
---|---
R <cr> | 7.82 <cr> *OK <cr>

Green
Standby

Cyan
Taking reading

Transmitting

600 ms
# Calibration

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal</td>
<td>Calibrate to atmospheric oxygen levels</td>
</tr>
<tr>
<td>Cal,0</td>
<td>Calibrate device to 0 dissolved oxygen</td>
</tr>
<tr>
<td>Cal,clear</td>
<td>Delete calibration data</td>
</tr>
<tr>
<td>Cal,?</td>
<td>Device calibrated?</td>
</tr>
</tbody>
</table>

The EZO™ Dissolved Oxygen circuit uses single and/or two point calibration.

## Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal</td>
<td>*OK</td>
</tr>
<tr>
<td>Cal,0</td>
<td>*OK</td>
</tr>
<tr>
<td>Cal,clear</td>
<td>*OK</td>
</tr>
</tbody>
</table>
| Cal,?     | ?Cal,0   or ?Cal,1 or ?Cal,2  

*OK

Example Response

9.53 mg/L

8.82 mg/L
# Export/import calibration

## Command syntax

- **Export** `<cr>` export calibration string from calibrated device*
- **Import** `<cr>` import calibration string to new device*
- **Export,?** `<cr>` calibration string info*

*This command is only available for firmware version 2.10 and above.

## Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export,? <code>&lt;cr&gt;</code></td>
<td>10,120 <code>&lt;cr&gt;</code></td>
</tr>
</tbody>
</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 <code>&lt;cr&gt;</code></th>
<th>59 6F 75 20 61 72 <code>&lt;cr&gt;</code> (1 of 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export <code>&lt;cr&gt;</code></td>
<td>65 20 61 20 63 6F <code>&lt;cr&gt;</code> (2 of 10)</td>
</tr>
<tr>
<td>Export <code>&lt;cr&gt;</code></td>
<td>6F 6C 20 67 75 79 <code>&lt;cr&gt;</code> (3 of 10)</td>
</tr>
</tbody>
</table>

Disabling *OK simplifies this process.

<table>
<thead>
<tr>
<th>Import, n (FIFO)</th>
<th>Import, 59 6F 75 20 61 72 <code>&lt;cr&gt;</code> (1 of 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
Temperature compensation

Command syntax

\[ T_n <cr> \quad n = \text{any value; floating point or int} \]

\[ T_? <cr> \quad \text{compensated temperature value?} \]

Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ T,19.5 &lt;cr&gt; ]</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>[ T,? &lt;cr&gt; ]</td>
<td>?T,19.5 &lt;cr&gt;</td>
</tr>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

Default temperature = 20°C
Temperature is always in Celsius

Example Response

\[ ?T,19.5 <cr> \]
\[ 8.82 \text{ mg/L} \]

\[ T,19.5 <cr> \]
\[ 8.91 \text{ mg/L} \]
### Salinity compensation

**Command syntax**

- `S,n <cr>`  n = any value in microsiemens
- `S,n,ppt <cr>`  n = any value in ppt
- `S,? <cr>`  compensated salinity value?

**Example**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>S,50000 &lt;cr&gt;</code></td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td><code>S,37.5,ppt &lt;cr&gt;</code></td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td><code>S,? &lt;cr&gt;</code></td>
<td><code>?S,50000,μS cr</code> or <code>?S,37.5,ppt cr</code> *OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

**Example Response**

```
8.91 mg/L
```

**Diagram**

- `S,50000 <cr>`
- `8.91 mg/L`

- `8.01 mg/L`
## Pressure compensation

### Command syntax

- **P,**\text{n} <cr> \text{ n = any value in kPa}
- **P,**\text{?} <cr> \text{compensated pressure value?}

### Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>P,90.25 &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>P,? &lt;cr&gt;</td>
<td>?P,90.25 &lt;cr&gt; *OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

### Default value

Default value = 101.3 kPa
This parameter can be omitted if the water is less than 10 meters deep

### Example Response

8.01 mg/L

6.94 mg/L

Copyright © Atlas Scientific LLC
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,mg,1 / O,mg,0</code></td>
<td>*OK &lt;cr&gt; enable / disable mg/L</td>
</tr>
<tr>
<td><code>O,%,1 / O,%,0</code></td>
<td>*OK &lt;cr&gt; enable / disable percent saturation</td>
</tr>
<tr>
<td><code>O,?</code></td>
<td>?;O,%,mg &lt;cr&gt; if both are enabled</td>
</tr>
</tbody>
</table>

**Parameters**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mg</td>
<td>mg/L</td>
</tr>
<tr>
<td>%</td>
<td>percent saturation</td>
</tr>
</tbody>
</table>

Followed by 1 or 0

<table>
<thead>
<tr>
<th>Value</th>
<th>Status</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”.*
# Naming device

## Command syntax

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name, n &lt;cr&gt;</td>
<td>set name</td>
</tr>
<tr>
<td>Name, ? &lt;cr&gt;</td>
<td>show name</td>
</tr>
</tbody>
</table>

### Example

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

### Example Response

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

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

---

Copyright © Atlas Scientific LLC
Device information

Command syntax

i <cr> device information

Example | Response
--- | ---
i <cr> | ?i,D.O.,1.98 <cr>
 | *OK <cr>

Response breakdown

?i, D.O., 1.98

↑ Device    ↑ Firmware
# 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>

## Example

### Response

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>R &lt;cr&gt;</td>
<td>7.82 &lt;cr&gt; *OK &lt;cr&gt;</td>
</tr>
<tr>
<td>*OK,0</td>
<td>no response, *OK disabled</td>
</tr>
<tr>
<td>R &lt;cr&gt;</td>
<td>7.82 &lt;cr&gt; *OK disabled</td>
</tr>
<tr>
<td>*OK,?</td>
<td>?*OK,1 &lt;cr&gt; or ?*OK,0 &lt;cr&gt;</td>
</tr>
</tbody>
</table>

## Other response codes

- **ER**: unknown command
- **OV**: over volt (VCC>=5.5V)
- **UV**: under volt (VCC<=3.1V)
- **RS**: reset
- **RE**: boot up complete, ready
- **SL**: entering sleep mode
- **WA**: wake up

These response codes cannot be disabled.

---

Copyright © Atlas Scientific LLC
# Reading device status

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status &lt;cr&gt;</td>
<td>Voltage at Vcc pin and reason for last restart</td>
</tr>
</tbody>
</table>

## Example

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

## Response breakdown

```
?Status, P, 5.038
```

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Reason for restart</td>
<td>P,</td>
</tr>
<tr>
<td>Voltage at Vcc</td>
<td>5.038</td>
<td></td>
</tr>
</tbody>
</table>

## 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>  *SL

Any command  *WA  <cr>  wakes up device

<table>
<thead>
<tr>
<th>Voltage</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>13.1 mA</td>
<td>0.66 mA</td>
</tr>
<tr>
<td>3.3V</td>
<td>12 mA</td>
<td>0.3 mA</td>
</tr>
</tbody>
</table>

Sleep <cr>  enter sleep mode/low power

Standby  13.1 mA

Sleep  0.66 mA
**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 &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Baud,? &lt;cr&gt;</td>
<td>?Baud,38400 &lt;cr&gt;</td>
</tr>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

**n =**

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

---

*Note: The image contains a diagram illustrating the process of changing the baud rate.*
Protocol lock

Command syntax

Plock,1  <cr>  enable Plock
Plock,0  <cr>  disable Plock  **default**
Plock,?  <cr>  Plock on/off?

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>

Plock,1

*OK <cr>

cannot change to I²C

*ER <cr>

I2C,100

Short

cannot change to I²C
Factory reset

Command syntax

Factory <cr> enable factory reset

Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

(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

\[ n = \text{any number } 1 - 127 \]

## Example

| I²C,100 <cr> | *OK (reboot in I²C mode) |

## Wrong example

| I²C,139 <cr> | n ≠ 127 | *ER <cr> |

---

I²C,100

Green

*OK <cr>

(reboot)

Blue

now in I²C mode
Manual switching to I²C

- Make sure Plock is set to 0
- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to PGND
- 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 97 (0x61)**

**Example**

![Example Diagram](image1)

**Wrong Example**

![Wrong Example Diagram](image2)
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
- Pressure compensation
- Salinity compensation
- Sleep mode
- Temperature compensation
**I²C mode**

**I²C address**  (0x01 – 0x7F)  
97 (0x61) default

**Vcc**  3.3V – 5.5V

**Clock speed**  100 – 400 kHz

**SDA**  
0V 0V  
VCC  

**SCL**  
0V 0V  
VCC  

---

**Data format**

**Reading**  D.O.

**Units**  mg/L & (% sat)  
when enabled

**Encoding**  ASCII

**Format**  string  
(CSV string when % sat is enabled)

**Terminator**  carriage return

**Data type**  floating point

**Decimal places**  
mg/L = 2

**Smallest string**  
% sat = 1

**Largest string**  
4 characters

**Smallest string**  
16 characters
Sending commands to device

5 parts

Start | I²C address | Write | Command (not case sensitive) | Stop
--- | --- | --- | --- | ---
97 (0x61) | | | ASCII command string |

Example

Start | 97 (0x61) | Write | Sleep | Stop
--- | --- | --- | --- | ---
I²C address | | Command |

Advanced

Address bits

The entire command as ASCII with all arguments

4.7k resistor may be needed
Requesting data from device

7 parts

Start | \(1^\text{st}\) C address | Read | Response code | Data string | Null | Stop

97 (0x61) | 1 byte | "7.82" | Terminator (Dec 0)

Advanced

© 2023 Atlas Scientific LLC
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.

**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 ID#

Red
Command not understood

White
Find

LED ON

5V  +0.4 mA

3.3V  +0.2 mA

LED color definition

LED ON

5V  +0.4 mA

3.3V  +0.2 mA
# 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>change back to UART mode</td>
<td>62</td>
</tr>
<tr>
<td>Cal</td>
<td>performs calibration</td>
<td>50</td>
</tr>
<tr>
<td>Export/import</td>
<td>export/import calibration</td>
<td>51</td>
</tr>
<tr>
<td>Factory</td>
<td>enable factory reset</td>
<td>61</td>
</tr>
<tr>
<td>Find</td>
<td>finds device with blinking white LED</td>
<td>48</td>
</tr>
<tr>
<td>i</td>
<td>device information</td>
<td>56</td>
</tr>
<tr>
<td>I²C</td>
<td>change I²C address</td>
<td>62</td>
</tr>
<tr>
<td>L</td>
<td>enable/disble LED</td>
<td>47</td>
</tr>
<tr>
<td>O</td>
<td>removing parameters</td>
<td>55</td>
</tr>
<tr>
<td>P</td>
<td>pressure compensation</td>
<td>54</td>
</tr>
<tr>
<td>Plock</td>
<td>enable/disble protocol lock</td>
<td>59</td>
</tr>
<tr>
<td>R</td>
<td>returns a single reading</td>
<td>49</td>
</tr>
<tr>
<td>S</td>
<td>salinity compensation</td>
<td>53</td>
</tr>
<tr>
<td>Sleep</td>
<td>enter sleep mode/low power</td>
<td>58</td>
</tr>
<tr>
<td>Status</td>
<td>retrieve status information</td>
<td>57</td>
</tr>
<tr>
<td>T</td>
<td>temperature compensation</td>
<td>52</td>
</tr>
</tbody>
</table>
# LED control

## Command syntax

- **L,1**  LED on  **default**
- **L,0**  LED off
- **L,?**  LED state on/off?

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L,1</strong></td>
<td><img src="image" alt="Wait 300ms" /> 1 Dec 0 Null</td>
</tr>
<tr>
<td><strong>L,0</strong></td>
<td><img src="image" alt="Wait 300ms" /> 1 Dec 0 Null</td>
</tr>
<tr>
<td><strong>L,?</strong></td>
<td><img src="image" alt="Wait 300ms" /> 1 Dec ?L,1 0 ASCII Null  or  1 Dec ?L,0 0 ASCII Null</td>
</tr>
</tbody>
</table>

---

**Copyright © Atlas Scientific LLC**
Find

Command syntax

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

Example

Response

Find <cr>

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

*This command is only available for firmware version 2.10 and above.

300ms processing delay

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

Example Response

r 0.1
This command will disable continuous mode
Send any character or command to terminate find.

300ms processing delay

1 Dec 0 Null

Wait 300ms

Dec Null

Copyright © Atlas Scientific LLC

AtlasScientific
Environmental Robotics
Taking reading

Command syntax

R  return 1 reading

Example  Response

<table>
<thead>
<tr>
<th>R</th>
<th>1</th>
<th>7.82</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait 600ms</td>
<td>Dec</td>
<td>ASCII</td>
<td>Null</td>
</tr>
</tbody>
</table>

Example sequence:
- **Green** Taking reading
- **Blue** Transmitting
- **Standby**
Calibration

Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal</td>
<td>calibrate to atmospheric oxygen levels</td>
</tr>
<tr>
<td>Cal,0</td>
<td>calibrate device to 0 dissolved oxygen</td>
</tr>
<tr>
<td>Cal,clear</td>
<td>delete calibration data</td>
</tr>
<tr>
<td>Cal,?</td>
<td>device calibrated?</td>
</tr>
</tbody>
</table>

1300ms processing delay

Example Response

The EZO™ Dissolved Oxygen circuit uses single and/or two point calibration

Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal</td>
<td>Calibrate to atmospheric oxygen levels</td>
</tr>
<tr>
<td>Cal,0</td>
<td>Calibrate device to 0 dissolved oxygen</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</td>
<td>1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,0</td>
<td>1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,clear</td>
<td>1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,?</td>
<td>1 Dec ASCII 0 or 1 Dec ASCII 0 or 1 Dec ASCII 0</td>
</tr>
</tbody>
</table>

Copyright © Atlas Scientific LLC
## Export/import calibration

### Command syntax

**Export**
- export calibration string from calibrated device*

**Import**
- import calibration string to new device*

**Export,?**
- calibration string info*

*This command is only available for firmware version 2.10 and above.

### 300ms processing delay

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export,?</td>
<td>Wait 300ms</td>
</tr>
<tr>
<td>Export</td>
<td>Wait 300ms</td>
</tr>
<tr>
<td>Export</td>
<td>Wait 300ms</td>
</tr>
<tr>
<td>Import, n (FIFO)</td>
<td>Import, 59 6F 75 20 61 72</td>
</tr>
</tbody>
</table>

### Example Response

- **Export**: Use this command to save calibration settings to one or more devices.
- **Import**: Use this command to load calibration settings to one or more devices.

**Response breakdown**

<table>
<thead>
<tr>
<th># of strings to export</th>
<th># of bytes to export</th>
</tr>
</thead>
<tbody>
<tr>
<td>10, 120</td>
<td>10, 120</td>
</tr>
</tbody>
</table>

Export strings can be up to 12 characters long.
# Temperature compensation

## Command syntax

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>T,n</code></td>
<td><code>n</code> = any value; floating point or int</td>
</tr>
<tr>
<td><code>T,?</code></td>
<td>compensated temperature value?</td>
</tr>
</tbody>
</table>

**300ms processing delay**

Temperature is always in Celsius

## Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>T,19.5</code></td>
<td><img src="image1" alt="Response" /></td>
</tr>
<tr>
<td><code>T,?</code></td>
<td><img src="image2" alt="Response" /></td>
</tr>
</tbody>
</table>

**Example Response**

Temperature is always in Celsius

8.82 mg/L

8.91 mg/L

Copyright © Atlas Scientific LLC
# Salinity compensation

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>S,n</code></td>
<td><code>n = any value in microsiemens</code></td>
</tr>
<tr>
<td><code>S,n,ppt</code></td>
<td><code>n = any value in ppt</code></td>
</tr>
<tr>
<td><code>S,?</code></td>
<td>compensated salinity value?</td>
</tr>
</tbody>
</table>

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>S,50000</code></td>
<td><img src="wait.png" alt="Wait 300ms" /> 1 Dec Null</td>
</tr>
<tr>
<td><code>S,37.5,ppt</code></td>
<td><img src="wait.png" alt="Wait 300ms" /> 1 Dec Null</td>
</tr>
<tr>
<td><code>S,?</code></td>
<td><img src="wait.png" alt="Wait 300ms" /> 1 Dec ASCII Null or <img src="wait.png" alt="Wait 300ms" /> 1 Dec ASCII Null</td>
</tr>
</tbody>
</table>

*If the conductivity of your water is less than 2,500μS this command is irrelevant*
Pressure compensation

Command syntax

\[ P, n \quad n = \text{any value in kPa} \]

\[ P, ? \quad \text{compensated pressure value?} \]

300ms processing delay

Example Response

This parameter can be omitted if the water is less than 10 meters deep

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P,90.25 )</td>
<td><img src="1" alt="Wait 300ms" /> Dec 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P,? )</td>
<td><img src="1" alt="Wait 300ms" /> Dec ASCII</td>
</tr>
</tbody>
</table>

8.01 mg/L

6.94 mg/L

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

## Command syntax

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

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>O,mg,1 / O,mg,0</td>
<td>Wait 300ms</td>
</tr>
<tr>
<td>O,%,1 / O,%,0</td>
<td>Wait 300ms</td>
</tr>
<tr>
<td>O,?</td>
<td>Wait 300ms</td>
</tr>
</tbody>
</table>

### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mg</td>
<td>mg/L</td>
</tr>
<tr>
<td>%</td>
<td>percent saturation</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”.*
# Device information

## Command syntax

### 300ms processing delay

### i device information

### Example 

<table>
<thead>
<tr>
<th>i</th>
<th>?i,D.O.,1.98</th>
<th>0</th>
</tr>
</thead>
</table>

### Response breakdown

| ?i, D.O., 1.98 | 1 |
| Device | Dec | ASCII | null |

Copyright © Atlas Scientific LLC
# Reading device status

## Command syntax

### 300ms processing delay

<table>
<thead>
<tr>
<th>Status voltage at Vcc pin and reason for last restart</th>
</tr>
</thead>
</table>

## Example

### Response

<table>
<thead>
<tr>
<th>Status</th>
<th>Dec</th>
<th>ASCII</th>
<th>Null</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait 300ms</td>
<td>1</td>
<td>?Status,P,5.038</td>
<td>0</td>
</tr>
</tbody>
</table>

## Response breakdown

<table>
<thead>
<tr>
<th>?Status,</th>
<th>P,</th>
<th>5.038</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reason for restart</td>
<td>Voltage at Vcc</td>
<td></td>
</tr>
</tbody>
</table>

## 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.

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep</td>
<td>no response</td>
</tr>
<tr>
<td>Any command</td>
<td>wakes up device</td>
</tr>
</tbody>
</table>

**Example Response**

- **Sleep**
  - no response

- **Any command**
  - wakes up device

**5V**

- **STANDBY**
  - 13.1 mA

- **SLEEP**
  - 0.66 mA

**3.3V**

- **12 mA**

- **0.3 mA**

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>
</table>
| Plock,1 | ![Wait 300ms](image1)  
|         | ![1](image2)  Dec  ![0](image2)  Null |
| Plock,0 | ![Wait 300ms](image1)  
|         | ![1](image2)  Dec  ![0](image2)  Null |
| Plock,? | ![Wait 300ms](image1)  
|         | ![1](image2)  Dec  ![?Plock,1](image2)  ![0](image2)  ASCII  Null |

**Example of Command Execution:**

- **Plock,1**  Serial, 9600  
- **Serial, 9600**  cannot change to UART  
- **cannot change to UART**
# I²C address change

## Command syntax

I²C, n sets I²C address and reboots into I²C mode

### Example

<table>
<thead>
<tr>
<th>I²C, n</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>I²C, 100</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 97 (0x61).

n = any number 1 – 127
# Factory reset

## Command syntax

<table>
<thead>
<tr>
<th>Factory enabled factory reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2C address will not change</td>
</tr>
</tbody>
</table>

## Example

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

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

Clears calibration
LED on
Response codes enabled

Factory

(reboot)
Change to UART mode

Command syntax

Baud,n  switch from I²C to UART

Example

<table>
<thead>
<tr>
<th>Baud</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>9600</td>
<td>reboot in UART mode</td>
</tr>
</tbody>
</table>

n =

300
1200
2400
9600
19200
38400
57600
115200

Serial,9600

Changing to UART mode

(reboot)
Manual switching to UART

- Make sure Plock is set to 0
- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to PGND
- 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
In your CAD software place an 8 position header.

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

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 4.2
Revised definition of response codes on pg 44.

Datasheet V 4.1
Updated firmware changes on pg. 66.

Datasheet V 4.0
Revised Enable/disable parameters information on pages 29 & 55.

Datasheet V 3.9
Revised information on cover page.

Datasheet V 3.8
Update firmware changes on pg. 66.

Datasheet V 3.7
Revised Plock pages to show default value.

Datasheet V 3.6
Added new commands:
"Find" pages 21 & 48.
"Export/Import calibration" pages 25 & 51.
Added new feature to continous mode "C,n" pg 22.

Datasheet V 3.5
Added accuracy range on cover page, and revised isolation info on pg. 10.

Datasheet V 3.4
Added manual switching to UART information on pg. 59.

Datasheet V 3.3
Updated firmware changes to reflect V1.99 update.

Datasheet V 3.2
Revised entire datasheet.
## Firmware updates

**V1.1 – Initial release (Oct 30, 2014)**
- Change output to mg/L, then percentage (was previously percentage, then mg/L).

**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.

**V1.7 – 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 glitch where EEPROM would get erased if the circuit lost power 900ms into startup.

**V1.97 – EEPROM (Oct 10, 2016)**
- Fixed glitch in the cal clear command, improves how it calculates the DO, adds calibration saving and loading.

**V1.98 – EEPROM (Nov 14, 2016)**
- Updated firmware for new circuit design.

**V1.99 – (Feb 2, 2017)**
- Revised "O" command to accept mg.

**V2.10 – (April 12, 2017)**
- Added "Find" command.
- Added "Export/import" command.
- Modified continuous mode to be able to send readings every "n" seconds.

**V2.11 – (Sept 28, 2017)**
- Fixed glitch where the temperature would default to 0 on startup.

**V2.12 – (Dec 19, 2017)**
- Improved accuracy of dissolved oxygen equations.
Warranty

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

The debugging phase

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

- Soldering any part of the EZO™ class Dissolved Oxygen circuit.
- Running any code, that does not exclusively drive the EZO™ class Dissolved Oxygen circuit and output its data in a serial string.
- Embedding the EZO™ class Dissolved Oxygen circuit into a custom made device.
- Removing any potting compound.
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 Dissolved Oxygen circuit, against the thousands of possible variables that may cause the EZO™ class Dissolved Oxygen 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 Dissolved Oxygen circuits continued operation. This is because that would be equivalent to Atlas Scientific™ taking responsibility over the correct operation of your entire device.