EZO-pH™
Embedded pH Circuit

Reads
pH

Range
.001 – 14.000

Resolution
.001

Accuracy
+/- 0.002

Response time
1 reading per sec

Supported probes
Any type & brand

Calibration
1, 2, 3 point

Temp compensation
Yes

Data protocol
UART & I²C

Default I²C address
99 (0x63)

Operating voltage
3.3V – 5V

Data format
ASCII

Written by Jordan Press
Designed by Noah Press

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

<table>
<thead>
<tr>
<th>UART</th>
<th>I²C</th>
</tr>
</thead>
<tbody>
<tr>
<td>UART mode</td>
<td>I²C mode</td>
</tr>
<tr>
<td>Default state</td>
<td>Sending commands</td>
</tr>
<tr>
<td>Receiving data from device</td>
<td>Requesting data</td>
</tr>
<tr>
<td>Sending commands to device</td>
<td>Response codes</td>
</tr>
<tr>
<td>LED color definition</td>
<td>LED color definition</td>
</tr>
<tr>
<td>UART quick command page</td>
<td>I²C quick command page</td>
</tr>
<tr>
<td>LED control</td>
<td></td>
</tr>
<tr>
<td>Find</td>
<td></td>
</tr>
<tr>
<td>Continuous reading mode</td>
<td>Taking reading</td>
</tr>
<tr>
<td>Single reading mode</td>
<td>Calibration</td>
</tr>
<tr>
<td>Calibration</td>
<td>Export calibration</td>
</tr>
<tr>
<td>Export calibration</td>
<td>Import calibration</td>
</tr>
<tr>
<td>Import calibration</td>
<td>Slope</td>
</tr>
<tr>
<td>Slope</td>
<td>Temperature compensation</td>
</tr>
<tr>
<td>Temperature compensation</td>
<td></td>
</tr>
<tr>
<td>Naming device</td>
<td>Device information</td>
</tr>
<tr>
<td>Device information</td>
<td>Reading device status</td>
</tr>
<tr>
<td>Response codes</td>
<td>Sleep mode/low power</td>
</tr>
<tr>
<td>Reading device status</td>
<td>Protocol lock</td>
</tr>
<tr>
<td>Sleep mode/low power</td>
<td>I²C address change</td>
</tr>
<tr>
<td>Change baud rate</td>
<td>Factory reset</td>
</tr>
<tr>
<td>Protocol lock</td>
<td>Change to UART mode</td>
</tr>
<tr>
<td>Factory reset</td>
<td>Manual switching to UART</td>
</tr>
<tr>
<td>Change to I²C mode</td>
<td></td>
</tr>
<tr>
<td>Manual switching to I²C</td>
<td></td>
</tr>
</tbody>
</table>

| Circuit dimensions        | Power and data isolation|
| Power consumption         | Correct wiring          |
| Absolute max ratings      | Calibration theory      |
| Operating principle       | Available data protocols|

- Circuit footprint
- Datasheet change log
- Warranty
**EZO™ circuit dimensions**

- **13.97mm** (0.55”)
- **20.16mm** (0.79”)
- **5.8mm** (0.22”)
- **1mm Ø**
- **8.38mm** (0.32”)
- **10.8mm** (0.4”)

**Power consumption**

<table>
<thead>
<tr>
<th>Voltage</th>
<th>LED</th>
<th>MAX</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>ON</td>
<td>18.3 mA</td>
<td>16 mA</td>
<td>1.16 mA</td>
</tr>
<tr>
<td></td>
<td>OFF</td>
<td>13.8 mA</td>
<td>13.8 mA</td>
<td></td>
</tr>
<tr>
<td>3.3V</td>
<td>ON</td>
<td>14.5 mA</td>
<td>13.9 mA</td>
<td>0.995 mA</td>
</tr>
<tr>
<td></td>
<td>OFF</td>
<td>13.3 mA</td>
<td>13.3 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 (EZO™ pH)</td>
<td>-65 °C</td>
<td>125 °C</td>
<td></td>
</tr>
<tr>
<td>Operational temperature (EZO™ pH)</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>
Operating principle

A pH (potential of Hydrogen) probe measures the hydrogen ion activity in a liquid. At the tip of a pH probe is a glass membrane. This glass membrane permits hydrogen ions from the liquid being measured to defuse into the outer layer of the glass, while larger ions remain in the solution. The difference in the concentration of hydrogen ions (outside the probe vs. inside the probe) creates a VERY small current. This current is proportional to the concentration of hydrogen ions in the liquid being measured.

- Acid: pH < 7
- Neutral: pH = 7
- Base: pH > 7
Power and data isolation

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

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

Without isolation, Conductivity and Dissolved Oxygen readings will effect pH 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 4.7kΩ pull up resistor on both the isolated and non-isolated lines (R1, R2, R3, and R4) The output voltage is set using a voltage divider (R5, R6, and R7) this produces a voltage of 3.9V regardless of your input voltage.

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

VCC = 3.0v − 5.5v
Correct wiring

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
NEVER EXTEND THE CABLE WITH CHEAP JUMPER WIRES!

DO NOT CUT THE PROBE CABLE WITHOUT REFERING TO THIS DOCUMENT!
DO NOT MAKE YOUR OWN UNSHIELDED CABLES!

ONLY USE SHIELDED CABLES. REFER TO THIS DOCUMENT!
Calibration theory

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 I²C mode after calibration will not affect the stored calibration. If the device must be calibrated in I²C mode be sure to continuously request readings so you can see the output from the probe.

Calibration order

If this is your first time calibrating the EZO™ pH circuit, we recommend that you follow this calibration order.

1. Mid point
2. Low point
3. High point
Mid point calibration

Remove the soaker bottle and rinse off the pH probe. Pour a small amount of the pH 7.00 calibration solution into a cup. Let the pH probe sit in the calibration solution until the readings stabilize (small movement from one reading to the next is normal).

Once the readings have stabilized, issue the Mid point calibration command. "cal,mid,7"

Two point calibration will provide high accuracy between 7.00 and the second point calibrated against, such as a 4.00.

Three point calibration will provide high accuracy over the full pH range. Three point calibration at 4.00, 7.00 and 10.00 should be considered the standard.
Low point calibration

- Rinse off the probe before calibrating to the low point.
- Pour a small amount of the pH 4.00 calibration solution into a cup.
- Wait for readings to stabilize (1 – 2 minutes).

Once the readings have stabilized, issue the Low point calibration command. "cal,low,4"

High point calibration

- Rinse off the probe before calibrating to the high point.
- Pour a small amount of the pH 10.00 calibration solution into a cup.
- Wait for readings to stabilize (1 – 2 minutes).

Once the readings have stabilized, issue the High point calibration command. "cal,high,10"

Issuing the cal,mid command after the EZO™ pH circuit has been calibrated, will clear the other calibration points. Full calibration will have to be redone.

The EZO™ pH circuits default temperature compensation is set to 25° C. If the temperature of the calibration solution is +/- 2° C from 25° C, consider setting the temperature compensation first. Temperature changes of < 2° C are insignificant.
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 response codes
- Hardware switch to I²C mode
- LED control
- Protocol lock
- Software switch to I²C mode

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

<table>
<thead>
<tr>
<th>Baud</th>
<th>300</th>
<th>1,200</th>
<th>2,400</th>
<th>9,600 (default)</th>
<th>19,200</th>
<th>38,400</th>
<th>57,600</th>
<th>115,200</th>
</tr>
</thead>
</table>

**Vcc** 3.3V – 5.5V

**Data format**

- **Reading**: pH
- **Units**: pH
- **Encoding**: ASCII
- **Format**: string
- **Terminator**: carriage return
- **Data type**: floating point
- **Decimal places**: 3
- **Smallest string**: 4 characters
- **Largest string**: 40 characters
Default state

- **Mode**: UART
- **Baud**: 9,600
- **Readings**: continuous
- **Speed**: 1 reading per second
- **LED**: on

LED Indicators:
- **Green**: Standby
- **Cyan**: Taking reading
- **Transmitting**
Receiving data from device

2 parts

ASCII data string
Command

Carriage return <cr>
Terminator

CPU
TX
RX
Blue
I2C standby
SDA
(TX) (RX)
SCL
Green
Taking reading
SDA
(TX) (RX)
SCL
Purple
Changing
I2C address
SDA
(TX) (RX)
SCL
Red
Command
not understood
SDA
(TX) (RX)
SCL
White
Find

Advanced

ASCII: 9 . 5 6 0 <cr>
Hex: 39 2E 35 36 30 0D
Dec: 57 46 53 54 48 13

9.560 <cr>
9,600 baud (default)

Receiver
Sending commands to device

2 parts

Command (not case sensitive)
ASCII data string

Carriage return \textless cr\textgreater
Terminator

Advanced

ASCII: \texttt{Sleep} \textless cr\textgreater

Hex: \texttt{53 6C 65 65 70 0D}
Dec: \texttt{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

<table>
<thead>
<tr>
<th></th>
<th>LED ON</th>
<th>5V</th>
<th>+2.2 mA</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3V</td>
<td>+0.6 mA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © Atlas Scientific LLC
# 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>9,600</td>
<td>36</td>
</tr>
<tr>
<td>C</td>
<td>enable/disable continuous reading</td>
<td>enabled</td>
<td>24</td>
</tr>
<tr>
<td>Cal</td>
<td>performs calibration</td>
<td>n/a</td>
<td>26</td>
</tr>
<tr>
<td>Export</td>
<td>export calibration</td>
<td>n/a</td>
<td>27</td>
</tr>
<tr>
<td>Factory</td>
<td>enable factory reset</td>
<td>n/a</td>
<td>38</td>
</tr>
<tr>
<td>Find</td>
<td>finds device with blinking white LED</td>
<td>n/a</td>
<td>23</td>
</tr>
<tr>
<td>i</td>
<td>device information</td>
<td>n/a</td>
<td>32</td>
</tr>
<tr>
<td>I2C</td>
<td>change to I²C mode</td>
<td>not set</td>
<td>39</td>
</tr>
<tr>
<td>Import</td>
<td>import calibration</td>
<td>n/a</td>
<td>28</td>
</tr>
<tr>
<td>L</td>
<td>enable/disable LED</td>
<td>enabled</td>
<td>22</td>
</tr>
<tr>
<td>Name</td>
<td>set/show name of device</td>
<td>not set</td>
<td>31</td>
</tr>
<tr>
<td>Plock</td>
<td>enable/disable protocol lock</td>
<td>disabled</td>
<td>37</td>
</tr>
<tr>
<td>R</td>
<td>returns a single reading</td>
<td>n/a</td>
<td>25</td>
</tr>
<tr>
<td>Sleep</td>
<td>enter sleep mode/low power</td>
<td>n/a</td>
<td>35</td>
</tr>
<tr>
<td>Slope</td>
<td>returns the slope of the pH probe</td>
<td>n/a</td>
<td>29</td>
</tr>
<tr>
<td>Status</td>
<td>retrieve status information</td>
<td>enable</td>
<td>34</td>
</tr>
<tr>
<td>T</td>
<td>temperature compensation</td>
<td>25°C</td>
<td>30</td>
</tr>
<tr>
<td>*OK</td>
<td>enable/disable response codes</td>
<td>enable</td>
<td>33</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>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>L,1</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>L,0</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>L,?</td>
<td>?L,1 &lt;cr&gt; or ?L,0 &lt;cr&gt; &lt;br&gt; *OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

Example Response

```
r 0.1
```

Copyright © Atlas Scientific LLC
# Find

## Command syntax

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

### Example

<table>
<thead>
<tr>
<th>Find &lt;cr&gt;</th>
<th>*OK &lt;cr&gt;</th>
</tr>
</thead>
</table>

**Response**

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

### Command syntax

**C,1**  
<cr>  enable continuous readings once per second  
<default>

**C,n**  
<cr>  continuous readings every n seconds (n = 2 to 99 sec)

**C,0**  
<cr>  disable continuous readings

**C,?**  
<cr>  continuous reading mode on/off?

### Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
</table>
| **C,1**  
<cr> | *OK  
<cr>  
pH (1 sec)  
<cr>  
pH (2 sec)  
<cr>  
pH (n sec)  
<cr> |
| **C,30**  
<cr> | *OK  
<cr>  
pH (30 sec)  
<cr>  
pH (60 sec)  
<cr>  
pH (90 sec)  
<cr> |
| **C,0**  
<cr> | *OK  
<cr> |
| **C,?**  
<cr> | ?C,1  
<cr>  or  
?C,0  
<cr>  or  
?C,30  
<cr>  
*OK  
<cr> |
Single reading mode

Command syntax

R <cr> takes single reading

Example

| R <cr> | 9.560 <cr> *OK <cr> |

Response

Example Response

Green
Standby

Cyan
Taking reading

Transmitting

800 ms
## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Cal,mid,n</code></td>
<td>single point calibration at midpoint</td>
</tr>
<tr>
<td><code>Cal,low,n</code></td>
<td>two point calibration at lowpoint</td>
</tr>
<tr>
<td><code>Cal,high,n</code></td>
<td>three point calibration at highpoint</td>
</tr>
<tr>
<td><code>Cal,clear</code></td>
<td>delete calibration data</td>
</tr>
<tr>
<td><code>Cal,?</code></td>
<td>device calibrated?</td>
</tr>
</tbody>
</table>

## Example Response

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Cal,mid,7.00</code></td>
<td><code>*OK</code></td>
</tr>
<tr>
<td><code>Cal,low,4.00</code></td>
<td><code>*OK</code></td>
</tr>
<tr>
<td><code>Cal,high,10.00</code></td>
<td><code>*OK</code></td>
</tr>
<tr>
<td><code>Cal,clear</code></td>
<td><code>*OK</code></td>
</tr>
<tr>
<td><code>Cal,?</code></td>
<td><code>?Cal,0</code> or <code>?Cal,1</code> or <code>?Cal,2</code> or <code>?Cal,3</code></td>
</tr>
</tbody>
</table>

Issuing the `cal,mid` command after the EZO™ pH circuit has been calibrated, will clear the other calibration points. Full calibration will have to be redone.
# Export calibration

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export,? &lt;cr&gt;</td>
<td>Calibration string info</td>
</tr>
<tr>
<td>Export &lt;cr&gt;</td>
<td>Export calibration string from calibrated device</td>
</tr>
</tbody>
</table>

## Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export,? &lt;cr&gt;</td>
<td>10,120 &lt;cr&gt;</td>
</tr>
<tr>
<td>Export &lt;cr&gt;</td>
<td>59 6F 75 20 61 72 &lt;cr&gt; (1 of 10)</td>
</tr>
<tr>
<td>Export &lt;cr&gt;</td>
<td>65 20 61 20 63 6F &lt;cr&gt; (2 of 10)</td>
</tr>
<tr>
<td>Export &lt;cr&gt;</td>
<td>(7 more)</td>
</tr>
<tr>
<td>Export &lt;cr&gt;</td>
<td>6F 6C 20 67 75 79 &lt;cr&gt; (10 of 10)</td>
</tr>
<tr>
<td>Export &lt;cr&gt;</td>
<td>*DONE</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>`.

## Example Response

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export &lt;cr&gt;</td>
<td>10,120 &lt;cr&gt;</td>
</tr>
<tr>
<td>Export &lt;cr&gt;</td>
<td>59 6F 75 20 61 72 &lt;cr&gt; (1 of 10)</td>
</tr>
<tr>
<td>Export &lt;cr&gt;</td>
<td>65 20 61 20 63 6F &lt;cr&gt; (2 of 10)</td>
</tr>
<tr>
<td>Export &lt;cr&gt;</td>
<td>(7 more)</td>
</tr>
<tr>
<td>Export &lt;cr&gt;</td>
<td>6F 6C 20 67 75 79 &lt;cr&gt; (10 of 10)</td>
</tr>
<tr>
<td>Export &lt;cr&gt;</td>
<td>*DONE</td>
</tr>
</tbody>
</table>

Disabling *OK simplifies this process.

---

*Copyright © Atlas Scientific LLC*
## Import calibration

### Command syntax

**Import, n**  
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.

---

**MCU**

The import command is used to upload calibration settings to one or more devices. The command syntax is `Import, n <cr>`. The example shows how to use this command to upload calibration strings to new devices.

The response indicates whether the import was successful with a *OK* or if there was an error with an *ER*.

---

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

Command syntax

Slope,? <cr> returns the slope of the pH probe

Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slope,? &lt;cr&gt;</td>
<td>?Slope,99.7,100.3 &lt;cr&gt; *OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

Response breakdown

?Slope, 99.7, 100.3

99.7% is how closely the slope of the acid calibration line matched the “ideal” pH probe.

100.3% is how closely the slope of the base calibration matches the “ideal” pH probe.

After calibrating a pH probe issuing the slope command will show how closely (in percentage) the calibrated pH probe is working compared to the “ideal” pH probe.

Copyright © Atlas Scientific LLC
## Temperature compensation

### Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T,n</td>
<td>( n ) = any value; floating point or int</td>
</tr>
<tr>
<td>T,?</td>
<td>Compensated temperature value?</td>
</tr>
<tr>
<td>RT,n</td>
<td>Set temperature compensation and take a reading*</td>
</tr>
</tbody>
</table>

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>T,19.5</td>
<td>*OK</td>
</tr>
<tr>
<td>RT,19.5</td>
<td>*OK 8.91</td>
</tr>
<tr>
<td>T,?</td>
<td>?T,19.5     *OK</td>
</tr>
</tbody>
</table>

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

*OK

This is a new command for firmware V2.12
Naming device

Command syntax

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

Example Response

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

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

Up to 16 ASCII characters
# Device information

## Command syntax

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

## Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>i &lt;cr&gt;</td>
<td>?i,pH,1.98 &lt;cr&gt;</td>
</tr>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

## Response breakdown

<table>
<thead>
<tr>
<th>?i,</th>
<th>pH,</th>
<th>1.98</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>Firmware</td>
<td></td>
</tr>
</tbody>
</table>
# Response codes

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*OK,1 &lt;cr&gt;</td>
<td>enable response</td>
</tr>
<tr>
<td>*OK,0 &lt;cr&gt;</td>
<td>disable response</td>
</tr>
<tr>
<td>*OK,? &lt;cr&gt;</td>
<td>response on/off?</td>
</tr>
</tbody>
</table>

## Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
</table>
| R <cr>  | 9.560 <cr>  
          | *OK <cr>   |
| *OK,0 <cr> | no response, *OK disabled |
| R <cr>  | 9.560 <cr>  
          | *OK disabled |
| *OK,? <cr> | ?*OK,1 <cr> or ?*OK,0 <cr> |

## 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

<table>
<thead>
<tr>
<th>Status &lt;cr&gt;</th>
<th>voltage at Vcc pin and reason for last restart</th>
</tr>
</thead>
</table>

## 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

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

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

Any command

*WA <cr> wakes up device

<table>
<thead>
<tr>
<th>5V</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16 mA</td>
<td>1.16 mA</td>
</tr>
</tbody>
</table>

| 3.3V        | 13.9 mA  | 0.995 mA  |

Example Response

wakes up device

Sleep <cr>

Standby 16 mA

Sleep 1.16 mA
Change baud rate

Command syntax

Baud, n  <cr>  change baud rate

Example  |  Response
---|---
Baud,38400  <cr>  |  *OK  <cr>
Baud,?  <cr>  |  ?Baud,38400  <cr>  
                         |  *OK  <cr>

n =

300
1200
2400
9600  default
19200
38400
57600
115200

Baud,38400  <cr>  

(reboot)

Standby  

Changing baud rate  

*OK  <cr>  

Standby
# Protocol lock

## Command syntax

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

## Example

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

## Example Responses

- Locks device to UART mode.
- *OK <cr>
- *ER <cr>
- cannot change to I²C
- Short
- cannot change to I²C
**Factory reset**

**Command syntax**

Factory <cr> enable factory reset

**Example**

- Factory <cr> enable factory reset

**Response**

- *OK <cr>
- (reboot)
- *OK <cr>
- *RS <cr> *RE <cr>

* Baud rate will not change

Clears calibration
LED on
"*OK" enabled

---

Copyright © Atlas Scientific LLC
# Change to I²C mode

## Command syntax

I²C,\textit{n}  \texttt{<cr>}  sets I²C address and reboots into I²C mode  

\textit{n} = any number 1 – 127

## Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>I²C,100 \texttt{&lt;cr&gt;}</td>
<td>*OK (reboot in I²C mode)</td>
</tr>
</tbody>
</table>

## Wrong example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>I²C,139 \texttt{&lt;cr&gt;}</td>
<td>*ER \texttt{&lt;cr&gt;}</td>
</tr>
</tbody>
</table>

## Default I²C address 99 (0x63)

I²C,\textit{n} \texttt{<cr>} sets I²C address and reboots into I²C mode

\textit{n} = any number 1 – 127

---

I²C,100

Green  
*OK \texttt{<cr>}

(reboot)

Blue  
now in I²C mode

---

Copyright © Atlas Scientific LLC
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 99 (0x63)

Example

![Correct Example]

Wrong Example

![Incorrect Example]
I²C mode

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

To set your EZO™ device into I²C mode click here

**Settings that are retained if power is cut**
- Calibration
- Change I²C address
- 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

AtlasScientific
Environmental Robotics
I²C mode

I²C address (0x01 – 0x7F)  
99 (0x63) default

Vcc  
3.3V – 5.5V

Clock speed  
100 – 400 kHz

SDA

SCL

0V

VCC

4.7k resistor may be needed

CPU

Data format

Reading  pH
Units  pH
Encoding  ASCII
Format  string

Data type  floating point
Decimal places  3
Smallest string  4 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  99 (0x63)  Write  Sleep  Stop

I²C address  Command

Advanced

Address bits

SDA  SCL

Start

The entire command as ASCII with all arguments

SDA  SCL

A6 A5 A4 A3 A2 A1 A0 W  ACK

First letter of command \ACK  \ACK

Last letter of command \ACK

W = low

Stop

Copyright © Atlas Scientific LLC
Requesting data from device

1. Start
2. I2C address (99 0x63)
3. Read
4. Response code (1 byte)
5. Data string ("9.65")
6. Null
7. Stop

Advanced

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

Start

9.560

ASCII

44 Copyright © 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;
```

If there is no processing delay or the processing delay is too short, the response code will always be 254.

Response codes

- **255**: no data to send
- **254**: still processing, not ready
- **2**: syntax error
- **1**: successful request
LED color definition

Blue
I²C standby

Green
Taking reading

Purple
Changing I²C address

Red
Command not understood

White
Find

<table>
<thead>
<tr>
<th>Voltage</th>
<th>LED ON</th>
<th>mA</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>5V</td>
<td>+2.2 mA</td>
</tr>
<tr>
<td>3.3V</td>
<td>3.3V</td>
<td>+0.6 mA</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>62</td>
</tr>
<tr>
<td>Cal</td>
<td>performs calibration</td>
<td>51</td>
</tr>
<tr>
<td>Export</td>
<td>export calibration</td>
<td>52</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>49</td>
</tr>
<tr>
<td>i</td>
<td>device information</td>
<td>56</td>
</tr>
<tr>
<td>I2C</td>
<td>change I²C address</td>
<td>60</td>
</tr>
<tr>
<td>Import</td>
<td>import calibration</td>
<td>53</td>
</tr>
<tr>
<td>L</td>
<td>enable/disable LED</td>
<td>48</td>
</tr>
<tr>
<td>Plock</td>
<td>enable/disable protocol lock</td>
<td>59</td>
</tr>
<tr>
<td>R</td>
<td>returns a single reading</td>
<td>50</td>
</tr>
<tr>
<td>Sleep</td>
<td>enter sleep mode/low power</td>
<td>58</td>
</tr>
<tr>
<td>Slope</td>
<td>returns the slope of the pH probe</td>
<td>54</td>
</tr>
<tr>
<td>Status</td>
<td>retrieve status information</td>
<td>57</td>
</tr>
<tr>
<td>T</td>
<td>temperature compensation</td>
<td>55</td>
</tr>
</tbody>
</table>
# LED control

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>L,1</td>
<td>LED on default</td>
<td>L,1</td>
<td>1 Dec 0 Null</td>
</tr>
<tr>
<td>L,0</td>
<td>LED off</td>
<td>L,0</td>
<td>1 Dec 0 Null</td>
</tr>
<tr>
<td>L,?</td>
<td>LED state on/off?</td>
<td>L,?</td>
<td>1 Dec 0 ASCII Null or 1 Dec 0 ASCII Null</td>
</tr>
</tbody>
</table>

### Example

- **L,1 LED on**
  - Example: L,1
  - Response: 1 Dec 0 Null

- **L,0 LED off**
  - Example: L,0
  - Response: 1 Dec 0 Null

- **L,? LED state on/off?**
  - Example: L,?
  - Response: 1 Dec 0 ASCII Null or 1 Dec 0 ASCII Null
Find

Command syntax

Find LED rapidly blinks white, used to help find device

Example

Response

Find

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find</td>
<td></td>
</tr>
</tbody>
</table>

Response:

- **Wait 300ms**
- **0** Dec Null

Example Response:

```
r 0.1
```

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

300ms processing delay
# Taking reading

## Command syntax

<table>
<thead>
<tr>
<th>R</th>
<th>return 1 reading</th>
</tr>
</thead>
</table>

### Example

<table>
<thead>
<tr>
<th>R</th>
<th>Wait 900ms</th>
<th>1</th>
<th>9.560</th>
<th>0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Dec</th>
<th>ASCII</th>
<th>Null</th>
</tr>
</thead>
</table>

### Response

- **Green**
  - Taking reading
  - Wait 900ms

- **Transmitting**

- **Blue**
  - Standby
Calibration

**Command syntax**

- **Cal,mid,n** single point calibration at midpoint
- **Cal,low,n** two point calibration at lowpoint
- **Cal,high,n** three point calibration at highpoint
- **Cal,clear** delete calibration data
- **Cal,** device calibrated?

**Example**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal,mid,7.00</td>
<td><img src="image" alt="Wait 900ms" /> 1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,low,4.00</td>
<td><img src="image" alt="Wait 900ms" /> 1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,high,10.00</td>
<td><img src="image" alt="Wait 900ms" /> 1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,clear</td>
<td><img src="image" alt="Wait 300ms" /> 1 Dec 0 Null</td>
</tr>
<tr>
<td>Cal,?</td>
<td><img src="image" alt="Wait 300ms" /> 1 Dec ?Cal,0 ASCII 0 Null or 1 Dec ?Cal,1 ASCII one point 0 Null or 1 Dec ?Cal,2 ASCII two point 0 Null or 1 Dec ?Cal,3 ASCII three point 0 Null</td>
</tr>
</tbody>
</table>
# Export calibration

## Command syntax

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

## Example Response

### Export,?

- **Wait 300ms**
- 1: 10,120 0
- **Response breakdown**
  - # of strings to export: 10
  - # of bytes to export: 120
  - Export strings can be up to 12 characters long

### Export

- **Wait 300ms**
  - 1: 59 6F 75 20 61 20 63 6F
    - Dec: 1
    - ASCII: 1
    - Null: 0
  - (1 of 10)

- **Wait 300ms**
  - 1: 65 20 61 20 63 6F 61 20 63 6F
    - Dec: 1
    - ASCII: 1
    - Null: 0
  - (2 of 10)

(7 more)

### Export

- **Wait 300ms**
  - 1: 6F 6C 20 67 75 79
    - Dec: 1
    - ASCII: 1
    - Null: 0
  - (10 of 10)

### Export

- **Wait 300ms**
  - 1: *DONE 0
    - Dec: 1
    - ASCII: 1
    - Null: 0

---

Export: Use this command to download calibration settings

300ms processing delay
**Import calibration**

**Command syntax**

Import, n  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</td>
<td>(1 of 10)</td>
</tr>
<tr>
<td>Import, 65 20 61 20 63 6F</td>
<td>(2 of 10)</td>
</tr>
<tr>
<td>Import, 6F 6C 20 67 75 79</td>
<td>(10 of 10)</td>
</tr>
</tbody>
</table>

* If one of the imported strings is not correctly entered, the device will not accept the import and reboot.
Slope,? returns the slope of the pH probe

Example | Response
--- | ---
Slope,? | ?Slope,99.7,100.3

Response breakdown

?Slope, 99.7, 100.3

99.7% is how closely the slope of the acid calibration line matched the “ideal” pH probe.

100.3% is how closely the slope of the base calibration matches the “ideal” pH probe.
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="https://via.placeholder.com/150" alt="Wait 300ms" /></td>
</tr>
<tr>
<td>RT,19.5</td>
<td><img src="https://via.placeholder.com/150" alt="Wait 900ms" /></td>
</tr>
<tr>
<td>T,?</td>
<td><img src="https://via.placeholder.com/150" alt="Wait 300ms" /></td>
</tr>
</tbody>
</table>

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

This is a new command for firmware V2.12
## Device information

### Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>i</td>
<td>?i,pH,1.98,0</td>
</tr>
</tbody>
</table>

| Device | pH, 1.98 |

**Response breakdown**

- ?i, pH, 1.98
- Device: pH
- Firmware: 1.98
- ASCII: "?i,pH,1.98"
- Null: 0

**300ms processing delay**

Wait 300ms
# Reading device status

## Command syntax

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

### Example

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

### Response breakdown

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

### Restart codes

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

300ms processing delay

---

Copyright © Atlas Scientific LLC
**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

Do not read status byte after issuing sleep command.

<table>
<thead>
<tr>
<th>Voltage</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>16 mA</td>
<td>1.16 mA</td>
</tr>
<tr>
<td>3.3V</td>
<td>13.9 mA</td>
<td>0.995 mA</td>
</tr>
</tbody>
</table>

Standby to Sleep Transition

Sleep

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="Wait 300ms" /> 1 Dec 0 Null</td>
</tr>
<tr>
<td>Plock,0</td>
<td><img src="image" alt="Wait 300ms" /> 1 Dec 0 Null</td>
</tr>
<tr>
<td>Plock,?</td>
<td><img src="image" alt="Wait 300ms" /> 1 ?Plock,1 0 ASCII 0 Null</td>
</tr>
</tbody>
</table>

### Example Response

- **Plock,1 Baud, 9600**
  - cannot change to UART

- **Baud, 9600**
  - cannot change to UART
## I²C address change

### Command syntax

I²Cₙ  sets I²C address and reboots into I²C mode

### Example

<table>
<thead>
<tr>
<th>Example</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 the CPU is updated with the new I²C address.

Default I²C address is 99 (0x63).

### n = any number 1 – 127

300ms processing delay

---

Copyright © Atlas Scientific LLC

---
# Factory reset

## Command syntax

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

| Factory enable factory reset | I²C address will not change |

## Example

<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

Example  |  Response
----------|----------
Baud, 9600  |  reboot in UART mode

n =

300
1200
2400
9600
19200
38400
57600
115200

Baud, 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

Disconnect RX line
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 5.3
Revised response for the sleep command in UART mode on pg 35.

Datasheet V 5.2
Revised calibration theory on page 11, and added more information on the Export calibration and Import calibration commands.

Datasheet V 5.1
Revised isolation schematic on pg 10.

Datasheet V 5.0
Added more information about temperature compensation on pages 29 & 53.

Datasheet V 4.9
Changed "Max rate" to "Response time" on cover page.

Datasheet V 4.8
Added new command:
"RT,n" for Temperature compensation located on pages 29 (UART) & 53 (I2C).
Added firmware information to Firmware update list.

Datasheet V 4.7
Removed note from certain commands about firmware version.

Datasheet V 4.6
Added information to calibration theory on pg 7.

Datasheet V 4.5
Revised definition of response codes on pg 44.
Datasheet V 4.4
Added resolution range to cover page.

Datasheet V 4.3
Revised isolation information on pg 9.

Datasheet V 4.2
Revised Plock pages to show default value.

Datasheet V 4.1
Added new commands:
"Find" pages 23 (UART) & 46 (I²C).
"Export/Import calibration" pages 27 (UART) & 49 (I²C).
Added new feature to continuous mode "C,n" pg 24.

Datasheet V 4.0
Added accuracy range on cover page, and revised isolation info on pg. 10.

Datasheet V 3.9
Revised calibration theory on pg. 7.

Datasheet V 3.8
Revised entire datasheet.
## Firmware updates

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.5</td>
<td>Nov 6, 2014</td>
<td>Baud rate change: Default baud rate changed to 9600</td>
</tr>
<tr>
<td>V1.6</td>
<td>Dec 1, 2014</td>
<td>Fixed I2C bug where the circuit may inappropriately respond when other I2C devices are connected.</td>
</tr>
<tr>
<td>V1.7</td>
<td>Apr 14, 2015</td>
<td>Changed “X” command to “Factory”</td>
</tr>
<tr>
<td>V1.95</td>
<td>Mar 31, 2016</td>
<td>Added protocol lock feature “Plock”</td>
</tr>
<tr>
<td>V1.96</td>
<td>Apr 26, 2016</td>
<td>Fixed bug where EEPROM would get erased if the circuit lost power 900ms into startup</td>
</tr>
<tr>
<td>V1.97</td>
<td>Oct 10, 2016</td>
<td>Added option to save and load calibration.</td>
</tr>
<tr>
<td>V1.98</td>
<td>Nov 14, 2016</td>
<td>Fixed bug during calibration process.</td>
</tr>
<tr>
<td>V2.10</td>
<td>May 9, 2017</td>
<td>Added “Find” command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added “Export/import” command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Modified continuous mode to be able to send readings every “n” seconds.</td>
</tr>
<tr>
<td>V2.11</td>
<td>Jun 12, 2017</td>
<td>Fixed “I” command to return “pH” instead of “PH”.</td>
</tr>
<tr>
<td>V2.12</td>
<td>Apr 16, 2018</td>
<td>Fixed “cal,clear” was not clearing stored calibration in EEPROM.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added “RT” command to Temperature compensation.</td>
</tr>
</tbody>
</table>
Warranty

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

The debugging phase

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

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