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

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>Circuit dimensions</td>
<td>4</td>
</tr>
<tr>
<td>Power consumption</td>
<td>4</td>
</tr>
<tr>
<td>Absolute max ratings</td>
<td>4</td>
</tr>
<tr>
<td>Operating principle</td>
<td>5</td>
</tr>
<tr>
<td>Available data protocols</td>
<td>15</td>
</tr>
</tbody>
</table>

**UART**

- UART mode 17
- Receiving data from device 18
- Sending commands to device 19
- LED color definition 20
- **UART quick command page** 21
- LED control 22
- Find 23
- Continuous reading mode 24
- Single reading mode 25
- Calibration 26
- Export calibration 27
- Import calibration 28
- Slope 29
- Temperature compensation 30
- Naming device 31
- Device information 32
- Response codes 33
- Reading device status 34
- Sleep mode/low power 35
- Change baud rate 36
- Protocol lock 37
- Factory reset 38
- Change to I²C mode 39
- Manual switching to I²C 40

**I²C**

- I²C mode 42
- Sending commands 43
- Requesting data 44
- Response codes 45
- LED color definition 46
- **I²C quick command page** 47
- LED control 48
- Find 49
- Taking reading 50
- Calibration 51
- Export calibration 52
- Import calibration 53
- Slope 54
- Temperature compensation 55
- Device information 56
- Reading device status 57
- Sleep mode/low power 58
- Protocol lock 59
- I²C address change 60
- Factory reset 61
- Change to UART mode 62
- Manual switching to UART 63

| Circuit footprint | 64 |
| Datasheet change log | 65 |
| Warranty | 68 |
**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>18.3 mA</td>
<td>16 mA</td>
<td>1.16 mA</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>13.8 mA</td>
<td>13.8 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>14.5 mA</td>
<td>13.9 mA</td>
<td>0.995 mA</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>13.3 mA</td>
<td>13.3 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™ pH)</td>
<td>-65 °C</td>
<td></td>
<td>125 °C</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.

Neutral: pH = 7

Acid: pH < 7

Base: pH > 7

Silver wire
Junction
Silver chloride
KCL reference solution
Reference wire
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-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 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

Copyright © Atlas Scientific LLC
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!
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.
**Single, Two point, or Three point calibration**

**No calibration**

**Two point calibration**
- Low point
- Mid point

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

**Three point calibration**
- Low point
- Mid point
- High point

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.

**The first calibration point must be the Mid point (pH 7.00)**

**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"
**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).

![Unstabilized](image1)

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).

![Unstabilized](image2)

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

**Issuing the cal,mid command**

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.*
Default state

UART mode

Baud 9,600

Readings continuous

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**

<table>
<thead>
<tr>
<th>Settings that are retained if power is cut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud rate</td>
</tr>
<tr>
<td>Calibration</td>
</tr>
<tr>
<td>Continuous mode</td>
</tr>
<tr>
<td>Device name</td>
</tr>
<tr>
<td>Enable/disable response codes</td>
</tr>
<tr>
<td>Hardware switch to I2C mode</td>
</tr>
<tr>
<td>LED control</td>
</tr>
<tr>
<td>Protocol lock</td>
</tr>
<tr>
<td>Software switch to I2C mode</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Settings that are <strong>NOT</strong> retained if power is cut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find</td>
</tr>
<tr>
<td>Sleep mode</td>
</tr>
<tr>
<td>Temperature compensation</td>
</tr>
</tbody>
</table>
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

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
Receiving data from device

2 parts

- ASCII data string
- Carriage return <cr>

Command
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 OD
Dec: 57 46 53 54 48 13

9.560 <cr>
9,600 baud (default)
Sending commands to device

2 parts

Command (not case sensitive)

ASCII data string

Carriage return <cr>

Terminator

Advanced

ASCII: S l e e p <cr>

Hex: 53 6C 65 65 70 0D

Dec: 83 108 101 101 112 13

Copyright © Atlas Scientific LLC
LED color definition

Green
UART standby

Cyan
Taking reading

Purple
Changing baud rate

Red
Command not understood

White
Find

5V
+2.2 mA

3.3V
+0.6 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>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; *OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

### Example Response

```
*OK <cr>
```

![LED control diagram](image_url)
Find

Command syntax

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

Example | Response
---------|---------
Find <cr> | *OK <cr>
## 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 (default)</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>
<tbody>
<tr>
<td>C,1</td>
<td>*OK &lt;cr&gt; pH (1 sec) &lt;cr&gt; pH (2 sec) &lt;cr&gt; pH (n sec) &lt;cr&gt;</td>
</tr>
<tr>
<td>C,30</td>
<td>*OK &lt;cr&gt; pH (30 sec) &lt;cr&gt; pH (60 sec) &lt;cr&gt; pH (90 sec) &lt;cr&gt;</td>
</tr>
<tr>
<td>C,0</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>C,?</td>
<td>?C,1 &lt;cr&gt; or ?C,0 &lt;cr&gt; or ?C,30 &lt;cr&gt; *OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>
Single reading mode

Command syntax

R <cr> takes single reading

Example  |  Response
---|---
R <cr>  |  9.560 <cr> *OK <cr>

Green  |  Cyan  |  Transmitting
Standby  |  Taking reading  |  

800 ms
## Calibration

### Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal,mid,n</td>
<td>single point calibration at midpoint</td>
</tr>
<tr>
<td>Cal,low,n</td>
<td>two point calibration at lowpoint</td>
</tr>
<tr>
<td>Cal,high,n</td>
<td>three point calibration at highpoint</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 Response**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal,mid,7.00</td>
<td>*OK</td>
</tr>
<tr>
<td>Cal,low,4.00</td>
<td>*OK</td>
</tr>
<tr>
<td>Cal,high,10.00</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 or ?Cal,3</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

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

Example

Export,? <cr> 10,120 <cr>

Response breakdown
10, 120
# of strings to export  # of bytes to export

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

Export <cr> 59 6F 75 20 61 72 <cr> (1 of 10)
Export <cr> 65 20 61 20 63 6F <cr> (2 of 10)
(7 more)
Export <cr> 6F 6C 20 67 75 79 <cr> (10 of 10)
Export <cr> *DONE

Disabling *OK simplifies this process

Export <cr>

Disabling *OK simplifies this process
Import calibration

Command syntax

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

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

Example

<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.
Slope

Command syntax

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

Example  |  Response
---------|---------
Slope,? <cr> | ?Slope,99.7,100.3 <cr> *OK <cr>

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.
**Temperature compensation**

**Command syntax**

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

**Default temperature = 25°C**

**Temperature is always in Celsius**

**Temperature is not retained if power is cut**

**Example Response**

<table>
<thead>
<tr>
<th>Example</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>( RT,19.5 ) (&lt;cr&gt;)</td>
<td>*OK  (&lt;cr&gt;) 8.91  (&lt;cr&gt;)</td>
</tr>
<tr>
<td>( T,? ) (&lt;cr&gt;)</td>
<td>?T,19.5  (&lt;cr&gt;) *OK  (&lt;cr&gt;)</td>
</tr>
</tbody>
</table>

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

## Command syntax

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

### Example Response

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

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

### Example Diagram

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

- **Name,?**<br>**Name,zzt** <cr>*OK <cr>*
## Device information

### Command syntax

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

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

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</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

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>9.560</td>
</tr>
<tr>
<td></td>
<td>*OK</td>
</tr>
<tr>
<td>*OK,0</td>
<td>no response, *OK disabled</td>
</tr>
<tr>
<td>R</td>
<td>9.560</td>
</tr>
<tr>
<td>*OK,?</td>
<td>?*OK,1</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.
# 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>?Status,P,5.038 &lt;cr&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></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</td>
</tr>
<tr>
<td>Voltage at Vcc</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

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td></td>
<td>*SL &lt;cr&gt;</td>
</tr>
<tr>
<td>Any command</td>
<td>*WA &lt;cr&gt;  wakes up device</td>
</tr>
</tbody>
</table>

<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>

Example Response

Sleep <cr> wakes up device

Standby
16 mA

Sleep
1.16 mA
# Change baud rate

## Command syntax

```
Baud, n <cr> change baud rate
```

## Example

<table>
<thead>
<tr>
<th>Baud, n</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 &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

---

 стандартный

36 Copyright © Atlas Scientific LLC
## 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>

### Example Response

- `Plock,1`  
  - *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

Factory <cr> | Response
---|---
Factory <cr> | *OK <cr>

Baud rate will not change

(reboot)

*RS <cr>  
*RE <cr>
## Change to I²C mode

### Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2C,n</td>
<td>sets I²C address and reboots into I²C mode</td>
</tr>
</tbody>
</table>

n = any number 1 – 127

### Example

| I2C,100 | *OK (reboot in I²C mode) |

### Wrong example

| I2C,139 | n ≠ 127 |

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>*ER</td>
</tr>
</tbody>
</table>

Default I²C address 99 (0x63)

**I2C,100**

- **Green**
  - *OK <cr>
- **Blue**
  - now in I²C mode

(reboot)
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

Wrong Example

Disconnect 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
- 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)

99 (0x63) default

Vcc 3.3V – 5.5V

Clock speed 100 – 400 kHz

SDA

SCL

4.7k resistor may be needed

0V

VCC

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

I²C address 99 (0x63)

Example

Start 99 (0x63) Write Sleep Stop

I²C address Command

Advanced

Address bits

The entire command as ASCII with all arguments

W = low

Start

Stop

Copyright © Atlas Scientific LLC
Requesting data from device

7 parts

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

99 (0x63) | 1 byte | "9.65" | Terminator (Dec 0)

Advanced

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

Start | ACK | Response code | ACK | Data | ACK | ... | Data N | ACK | Null | ACK | ... | Null

SDA | SCL

1 57 46 53 54 48 0 = 9.560

Dec | 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**

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

Blue
I²C standby

Green
Taking reading

Purple
Changing I²C address

Red
Command not understood

White
Find

5V
- LED ON
- +2.2 mA

3.3V
- +0.6 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>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>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>L,1</td>
<td>LED on</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><img src="image1" alt="" /></td>
</tr>
<tr>
<td>L,0</td>
<td><img src="image2" alt="" /></td>
</tr>
<tr>
<td>L,?</td>
<td><img src="image3" alt="" /></td>
</tr>
</tbody>
</table>

### Example Response

```
300ms processing delay
```

- **Wait 300ms**

## Diagram

![Diagram](image5)
**Find**

**Command syntax**

300ms processing delay

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

<table>
<thead>
<tr>
<th>Find</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find</td>
<td></td>
</tr>
</tbody>
</table>

Wait 300ms

1 Dec

0 Null

300ms processing delay
Taking reading

Command syntax

R  return 1 reading

Example

<table>
<thead>
<tr>
<th>R</th>
<th>Wait 900ms</th>
<th>1</th>
<th>9.560</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dec</td>
<td>ASCII</td>
<td>Null</td>
<td></td>
</tr>
</tbody>
</table>

Response

Example Response

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

- **Transmitting**

- **Blue**
  - Standby

900ms processing delay

Copyright © Atlas Scientific LLC
## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal,mid,n</td>
<td>single point calibration at midpoint</td>
</tr>
<tr>
<td>Cal,low,n</td>
<td>two point calibration at lowpoint</td>
</tr>
<tr>
<td>Cal,high,n</td>
<td>three point calibration at highpoint</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

#### Response

<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 ASCII Null or <img src="image" alt="Wait 300ms" /> 1 Dec ASCII Null</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

**Export**: Use this command to download calibration settings

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

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

## Example Response

<table>
<thead>
<tr>
<th>Export,?</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Export,?</strong></td>
<td></td>
</tr>
</tbody>
</table>
  - 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 |

<table>
<thead>
<tr>
<th>Export</th>
<th>Response</th>
</tr>
</thead>
</table>
| **Export**  
  - Wait 300ms  
  - 1  
  - 59 6F 75 20 61 72 0  
  - (1 of 10) |

<table>
<thead>
<tr>
<th>Export</th>
<th>Response</th>
</tr>
</thead>
</table>
| **Export**  
  - Wait 300ms  
  - 1  
  - 65 20 61 20 63 6F 0  
  - (2 of 10) |

### (7 more)  

<table>
<thead>
<tr>
<th>Export</th>
<th>Response</th>
</tr>
</thead>
</table>
| **Export**  
  - Wait 300ms  
  - 1  
  - 6F 6C 20 67 75 79 0  
  - (10 of 10) |

<table>
<thead>
<tr>
<th>Export</th>
<th>Response</th>
</tr>
</thead>
</table>
| **Export**  
  - Wait 300ms  
  - 1  
  - *DONE*  
  - 0 |
## Import calibration

### Command syntax

**Import, n** import calibration string to new device

### Example

<table>
<thead>
<tr>
<th>Import</th>
<th>Response</th>
<th>(of 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import, 59 6F 75 20 61 72</td>
<td>[Wait 300ms] 1 Dec 0 Null</td>
<td>(1)</td>
</tr>
<tr>
<td>Import, 65 20 61 20 63 6F</td>
<td>[Wait 300ms] 1 Dec 0 Null</td>
<td>(2)</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Import, 6F 6C 20 67 75 79</td>
<td>[Wait 300ms] 1 Dec 0 Null</td>
<td>(10)</td>
</tr>
</tbody>
</table>

### Diagram

- **Import, n**
- **MCU**
- **SDA (TX) (RX)**
- **SCL**

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

**System will reboot**
### Command syntax

**Slope,?** returns the slope of the pH probe

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Slope,?</code></td>
<td><code>?Slope,99.7,100.3</code></td>
</tr>
</tbody>
</table>

### Response breakdown

- **?Slope,**: This command returns the slope of the pH probe.

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

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

### Diagram

The diagram illustrates the calibration process of a pH probe compared to an “ideal” pH probe. The zero point is where both probes intersect. The calibrated probe has a slope that deviates by 0.3% from the ideal probe. The processing delay is 300ms.

---

Copyright © Atlas Scientific LLC
Temperature compensation

**Command syntax**

\[ T, n \quad n = \text{any value; floating point or int} \]

\[ T, ? \quad \text{compensated temperature value?} \]

\[ RT, n \quad \text{set temperature compensation and take a reading*} \]

**Default temperature = 25°C**

**Temperature is always in Celsius**

**Temperature is not retained if power is cut**

---

**Example**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T,19.5 )</td>
<td><img src="8.82" alt="Wait 300ms" /> Dec Null</td>
</tr>
<tr>
<td>( RT,19.5 )</td>
<td><img src="8.91" alt="Wait 900ms" /> Dec ASCII Null</td>
</tr>
<tr>
<td>( T,? )</td>
<td><img src="?T,19.5" alt="Wait 300ms" /> Dec ASCII Null</td>
</tr>
</tbody>
</table>

---

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

## Command syntax

| i | device information |

### Example

<table>
<thead>
<tr>
<th>i</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td><img src="image" alt="Wait 300ms" /></td>
</tr>
</tbody>
</table>

### Response breakdown

<table>
<thead>
<tr>
<th>?i, pH, 1.98</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
</tr>
</tbody>
</table>
Reading device status

Command syntax

300ms processing delay

Status voltage at Vcc pin and reason for last restart

Example Response

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

<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>

### Power Consumption

<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>

---

Copyright © Atlas Scientific LLC
# 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</td>
</tr>
<tr>
<td>Plock,?</td>
<td>Plock on/off?</td>
</tr>
</tbody>
</table>

### 300ms processing delay

Locks device to I²C mode.

## Example

### Response

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plock,1</td>
<td>Wait 300ms 1 Dec 0 Null</td>
</tr>
<tr>
<td>Plock,0</td>
<td>Wait 300ms 1 Dec 0 Null</td>
</tr>
<tr>
<td>Plock,?</td>
<td>Wait 300ms 1 ?Plock,1 0 ASCII 0 Null</td>
</tr>
</tbody>
</table>

### Example Response

- **Plock,1**
  - Wait 300ms
  - Dec
  - Null

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

- **Plock,?**
  - Wait 300ms
  - Dec
  - ASCII
  - Null

### Plock,1

- Baud, 9600

- **cannot change to UART**

### Baud, 9600

- **cannot change to UART**
I²C address change

Command syntax

I2C,n  sets I²C address and reboots into I²C mode

Example  |  Response
---------|---------
I2C,100  |  device reboot

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

I2C,100
# Factory reset

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory reset</td>
<td>Factory reset will not take the device out of I²C mode.</td>
</tr>
<tr>
<td>Factory enable factory reset</td>
<td>I²C address will not change</td>
</tr>
</tbody>
</table>

## Example

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

## Notes

- Clears calibration
- LED on
- Response codes enabled

## Diagram

The diagram shows the factory reset process with an arrow indicating a device reboot.
# Change to UART mode

## Command syntax

Baud,\( n \) switch from I\(^2\)C to UART

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

**\( n = \)**

<table>
<thead>
<tr>
<th>Val</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
</tr>
<tr>
<td>1200</td>
</tr>
<tr>
<td>2400</td>
</tr>
<tr>
<td>9600</td>
</tr>
<tr>
<td>19200</td>
</tr>
<tr>
<td>38400</td>
</tr>
<tr>
<td>57600</td>
</tr>
<tr>
<td>115200</td>
</tr>
</tbody>
</table>

---

**Example Response**

**CPU**

**I2C standby**

**Blue**

Taking reading

**Green**

Changing I2C address

**Purple**

**I2C address**

**Red**

Command not understood

**White**

Find CPU

---

**Atlas Scientific**

Environmental Robotics

62 Copyright © Atlas Scientific LLC
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 5.4
Moved the Default state to pg 14.

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>Change default baud rate 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 the 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 &quot;Find&quot; command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added &quot;Export/import&quot; command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Modified continuous mode to be able to send readings every &quot;n&quot; seconds.</td>
</tr>
<tr>
<td>V2.11</td>
<td>Jun 12, 2017</td>
<td>Fixed &quot;I&quot; command to return &quot;pH&quot; instead of &quot;PH&quot;.</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.