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

EZO-pH™

This is an evolving document, check back for updates.

Written by Jordan Press
Designed by Noah Press

PATENT PROTECTED
SOLDERING THIS DEVICE VOIDS YOUR WARRANTY.

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!
**Power consumption**

<table>
<thead>
<tr>
<th></th>
<th>LED</th>
<th>MAX</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>5V</strong></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><strong>3.3V</strong></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 (EZOTM pH)</td>
<td>-65 °C</td>
<td></td>
<td>125 °C</td>
</tr>
<tr>
<td>Operational temperature (EZOTM 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>
EZO™ circuit identification

EZO™ pH circuit

Legacy pH circuit

Viewing correct datasheet

Viewing incorrect datasheet

Click here to view legacy datasheet
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.
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, continuous readings). 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 request readings continuously so you can see the output from the probe.

The Atlas Scientific EZO™ class pH circuit has a flexible calibration protocol, allowing for **single point**, **two point**, or **three point** calibration.

The first calibration point must be the Midpoint (pH 7)

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.**

### Single point calibration

1. Remove soaker bottle and rinse off pH probe.
2. Pour a small amount of the calibration solution into a cup.
3. Let the probe sit in calibration solution until readings stabilize (1 – 2 minutes).
4. Calibrate the midpoint value using the command **"Cal,mid,n"**.
   *Where "n" is any floating point value that represents the calibration midpoint.*
5. Do not pour the calibration solution back into the bottle.
### Two point calibration

1. Rinse off pH probe.
2. Pour a small amount of the calibration solution into a cup.
3. Let the probe sit in calibration solution until readings stabilize (1 – 2 minutes).
4. Calibrate the lowpoint value using the command "Cal, low, n".
   *Where "n" is any floating point value that represents the calibration lowpoint.*
5. Do not pour the calibration solution back into the bottle.

### Three point calibration

1. Rinse off pH probe.
2. Pour a small amount of the calibration solution into a cup.
3. Let the probe sit in calibration solution until readings stabilize (1 – 2 minutes).
4. Calibrate the highpoint value using the command "Cal, high, n".
   *Where "n" is any floating point value that represents the calibration highpoint.*
5. Do not pour the calibration solution back into the bottle.

---

**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.**
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 $4.7\,\Omega$ 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 3.9V 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

Bread board

Incorrect wiring

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

Bread board via USB

Carrier board

USB carrier board

Part # COM-104

Part # ISCCB

Part # USB-ISO

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!
## 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>

- **RX** (Data in)
  - Input data stream

- **TX** (Data out)
  - Output data stream

- **Vcc**
  - 3.3V – 5.5V

## Data format

<table>
<thead>
<tr>
<th>Reading</th>
<th>pH</th>
<th>Data type</th>
<th>floating point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Units</td>
<td>pH</td>
<td>Decimal places</td>
<td>3</td>
</tr>
<tr>
<td>Encoding</td>
<td>ASCII</td>
<td>Smallest string</td>
<td>4 characters</td>
</tr>
<tr>
<td>Format</td>
<td>string</td>
<td>Largest string</td>
<td>40 characters</td>
</tr>
<tr>
<td>Terminator</td>
<td>carriage return</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © Atlas Scientific LLC
Default state

Mode
UART

Baud
9,600

Readings
continuous

Speed
1 reading per second

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

9.560 <cr>

9,600 baud (default)

CPU

Receiver

Advanced

ASCII: 9 . 5 6 0 <cr>

Hex: 39 2E 35 36 30 0D

Dec: 57 46 53 54 48 13

Copyright © Atlas Scientific LLC
Sending commands to device

2 parts

Command (not case sensitive)  Carriage return <cr>

ASCII data string  Terminator

Advanced

ASCII: S l e e p <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

<table>
<thead>
<tr>
<th>Voltage</th>
<th>LED ON</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td></td>
<td>+2.2 mA</td>
</tr>
<tr>
<td>3.3V</td>
<td></td>
<td>+0.6 mA</td>
</tr>
</tbody>
</table>
# 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</td>
</tr>
<tr>
<td>C</td>
<td>enable/disable continuous reading</td>
<td>pg. 24</td>
</tr>
<tr>
<td>Cal</td>
<td>performs calibration</td>
<td>pg. 26</td>
</tr>
<tr>
<td>Export/import</td>
<td>export/import calibration</td>
<td>pg. 27</td>
</tr>
<tr>
<td>Factory</td>
<td>enable factory reset</td>
<td>pg. 37</td>
</tr>
<tr>
<td>Find</td>
<td>finds device with blinking white LED</td>
<td>pg. 23</td>
</tr>
<tr>
<td>i</td>
<td>device information</td>
<td>pg. 31</td>
</tr>
<tr>
<td>I2C</td>
<td>change to I2C mode</td>
<td>pg. 38</td>
</tr>
<tr>
<td>L</td>
<td>enable/disable LED</td>
<td>pg. 22</td>
</tr>
<tr>
<td>Name</td>
<td>set/show name of device</td>
<td>pg. 30</td>
</tr>
<tr>
<td>Plock</td>
<td>enable/disable protocol lock</td>
<td>pg. 36</td>
</tr>
<tr>
<td>R</td>
<td>returns a single reading</td>
<td>pg. 25</td>
</tr>
<tr>
<td>Sleep</td>
<td>enter sleep mode/low power</td>
<td>pg. 34</td>
</tr>
<tr>
<td>Slope</td>
<td>returns the slope of the pH probe</td>
<td>pg. 28</td>
</tr>
<tr>
<td>Status</td>
<td>retrieve status information</td>
<td>pg. 33</td>
</tr>
<tr>
<td>T</td>
<td>temperature compensation</td>
<td>pg. 29</td>
</tr>
<tr>
<td>*OK</td>
<td>enable/disable response codes</td>
<td>pg. 32</td>
</tr>
</tbody>
</table>
# LED control

## Command syntax

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

## Example Response

**Example**

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

**Image**

- L,1<br>
- L,0
Find

Command syntax

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

Example

Find <cr>

Response

*OK <cr>

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

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C,1</td>
<td>enable continuous readings once per second (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>

### Command syntax examples:

**Example** | **Response** |
---|---|
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

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>R &lt;cr&gt;</td>
<td>9.560 &lt;cr&gt; *OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

Green
Standby

Cyan
Taking reading

Transmitting

800 ms
**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**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal,mid,7.00</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Cal,low,4.00</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Cal,high,10.00</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Cal,clear</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Cal,?</td>
<td>?Cal,0 &lt;cr&gt; or ?Cal,1 &lt;cr&gt; or ?Cal,2 &lt;cr&gt; or ?Cal,3 &lt;cr&gt; *OK &lt;cr&gt;</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/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

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Export,</strong>  <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>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export  <code>&lt;cr&gt;</code></td>
<td>59 6F 75 20 61 72  <code>&lt;cr&gt;</code></td>
</tr>
<tr>
<td>Export  <code>&lt;cr&gt;</code></td>
<td>65 20 61 20 63 6F  <code>&lt;cr&gt;</code></td>
</tr>
<tr>
<td>(8 more)</td>
<td></td>
</tr>
<tr>
<td>Export  <code>&lt;cr&gt;</code></td>
<td>6F 6C 20 67 75 79  <code>&lt;cr&gt;</code></td>
</tr>
<tr>
<td>Export  <code>&lt;cr&gt;</code></td>
<td>*DONE</td>
</tr>
</tbody>
</table>

Disabling *OK simplifies this process

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import,  <code>&lt;cr&gt;</code></td>
<td>Import, 59 6F 75 20 61 72  <code>&lt;cr&gt;</code></td>
</tr>
</tbody>
</table>

(1 of 10)
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.

**Command syntax**

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

**Example**

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

---

![Diagram showing the calibration of a pH probe](image-url)

Delta 0.3% from ideal

Zero point

Calibrated probe

Ideal 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**  

This is a new command for firmware V2.12

### Example

<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>
</tbody>
</table>
| RT,19.5 <cr> | *OK <cr>  
  8.91 <cr> |
| T,? <cr> | ?T,19.5 <cr>  
  *OK <cr> |
Naming device

Command syntax

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

Example Response

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

Example:

1. `Name,zzt<cr>`
   - Response: `*OK<cr>`

2. `Name,?<cr>`
   - Response: `?Name,zzt<cr>`
   - `*OK<cr>`

Copyright © Atlas Scientific LLC
# Device information

## Command syntax

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

## Example | Response
---|---
```plaintext
i <cr>
```
```plaintext
?i, pH, 1.98 <cr>
*OK <cr>
```

## 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>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

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>R &lt;cr&gt;</td>
<td>9.560 &lt;cr&gt; *OK &lt;cr&gt;</td>
</tr>
<tr>
<td>*OK,0 &lt;cr&gt;</td>
<td>no response, *OK disabled</td>
</tr>
<tr>
<td>R &lt;cr&gt;</td>
<td>9.560 &lt;cr&gt; *OK disabled</td>
</tr>
<tr>
<td>*OK,? &lt;cr&gt;</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*
# Reading device status

## Command syntax

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

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

Send any character or command to awaken device.

**Example**

<table>
<thead>
<tr>
<th>Sleep  &lt;cr&gt;</th>
<th>enter sleep mode/low power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep  &lt;cr&gt;</td>
<td>*SL</td>
</tr>
<tr>
<td>Any command</td>
<td>*WA &lt;cr&gt; wakes up device</td>
</tr>
</tbody>
</table>

**5V**

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

**3.3V**

| 13.9 mA | 0.995 mA |

**Copyright © Atlas Scientific LLC**
**Change baud rate**

**Command syntax**

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

<table>
<thead>
<tr>
<th>Example</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

**Example Response**

```
*OK
```

**Diagram**

1. **Standby**
   - Baud,38400 <cr>
2. **Changing baud rate**
   - *OK <cr>
3. (reboot)
4. **Standby**
## 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>

*OK* `<cr>`

### Example

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

### Example Response

- **Plock,1**
  - *OK `<cr>`

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

### Diagram

- **Plock,1**: *OK `<cr>`
- **I2C,100**: cannot change to I²C
- **Short**: cannot change to I²C

---

*Locks device to UART mode.*

---

Copyright © Atlas Scientific LLC
Factory reset

Command syntax

Factory <cr> enable factory reset

Example

Factory <cr>

Response

*OK <cr>

Factory <cr>

(reboot)

*OK <cr>

*RS <cr>

*RE <cr>

Baud rate will not change
# Change to I²C mode

**Command syntax**

$\text{I2C},n \ <cr>$ sets I²C address and reboots into I²C mode

$n = \text{any number } 1 - 127$

**Example**

- I2C,100 $<cr>$
  - Response: *OK (reboot in I²C mode)

**Wrong example**

- I2C,139 $<cr>$
  - Response: *ER $<cr>$

### I2C,100

- **Green**: *OK $<cr>$
  - (reboot)
  - **Blue**: now in I²C mode

---

Default I²C address 99 (0x63)

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

Wrong Example

Disconnected RX line
I²C mode

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

To set your EZO™ device into I²C mode click here

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

Data format

Reading  pH
Units  pH
Encoding  ASCII
Format  string

Data type  floating point
Decimal places  3
Smallest string  4 characters
Largest string  399 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

The entire command as ASCII with all arguments

W = low

Start

Stop
Requesting data from device

**Terminator:**
- Dec 0
- 7 parts
- Start
- I²C address
- Read
- Response code (1 byte)
- Data string ("9.65")
- Null
- Stop

**Advanced:**
- ASCII code: 57 46 53 54 48
- Dec 1
- SDA
- SCL
- Start
- Address bits
- A6 - A0
- R
- ACK
- Response code
- ACK
- Data
- ACK
- Data N
- ACK
- Null
- ACK
- Null
- NACK
- Stop

- All bytes after data are Null
- RI - High

- 9.560
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 ID#
- **Red**: Command not understood
- **White**: Find

**5V**
- LED ON: +2.2 mA

**3.3V**
- LED ON: +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>60</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>59</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>54</td>
</tr>
<tr>
<td>I2C</td>
<td>change I²C address</td>
<td>58</td>
</tr>
<tr>
<td>L</td>
<td>enable/disable LED</td>
<td>47</td>
</tr>
<tr>
<td>Plock</td>
<td>enable/disable protocol lock</td>
<td>57</td>
</tr>
<tr>
<td>R</td>
<td>returns a single reading</td>
<td>49</td>
</tr>
<tr>
<td>Sleep</td>
<td>enter sleep mode/low power</td>
<td>56</td>
</tr>
<tr>
<td>Slope</td>
<td>returns the slope of the pH probe</td>
<td>52</td>
</tr>
<tr>
<td>Status</td>
<td>retrieve status information</td>
<td>55</td>
</tr>
<tr>
<td>T</td>
<td>temperature compensation</td>
<td>53</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 Response

#### L,1

- **Wait 300ms**
- Response: 1 Dec 0 Null

#### L,0

- **Wait 300ms**
- Response: 1 Dec 0 Null

#### L,?

- **Wait 300ms**
- Response: 1 Dec ?L,1 0 ASCII Null or 1 Dec ?L,0 0 ASCII Null
Find

Command syntax

Find          LED rapidly blinks white, used to help find device

Example          Response

Find

Find

Wait 300ms

Dec 0

Null

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

r 0.1

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

300ms processing delay

Copyright © Atlas Scientific LLC
Taking reading

**Command syntax**

R  return 1 reading

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>![Image of the device in green, indicating 'Taking reading']</td>
</tr>
<tr>
<td>![Image of the device in blue, indicating 'Transmitting']</td>
<td>![Image of the device in green, indicating 'Taking reading']</td>
</tr>
<tr>
<td>![Image of the device in green, indicating 'Standby']</td>
<td>![Image of the device in green, indicating 'Transmitting']</td>
</tr>
<tr>
<td>![Image of the device in green, indicating 'Taking reading']</td>
<td>![Image of the device in blue, indicating 'Standby']</td>
</tr>
<tr>
<td>![Image of the device in blue, indicating 'Transmitting']</td>
<td>![Image of the device in green, indicating 'Taking reading']</td>
</tr>
</tbody>
</table>

900ms processing delay

Wait 900ms

R return 1 reading

Example Response

900ms processing delay

r 0.1
### 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal,mid,7.00</td>
<td><img src="image" alt="Response" /></td>
</tr>
<tr>
<td>Cal,low,4.00</td>
<td><img src="image" alt="Response" /></td>
</tr>
<tr>
<td>Cal,high,10.00</td>
<td><img src="image" alt="Response" /></td>
</tr>
<tr>
<td>Cal,clear</td>
<td><img src="image" alt="Response" /></td>
</tr>
<tr>
<td>Cal,?</td>
<td><img src="image" alt="Response" /></td>
</tr>
</tbody>
</table>

**Wait 300ms processing delay**

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/import calibration

## Command syntax

### Export
- Export calibration string from calibrated device

### Import
- Import calibration string to new device

### Export,?
- Calibration string info

---

## Export,?

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export,?</td>
<td><img src="image" alt="Wait 300ms" /></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**

## Export

(8 more)

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

(1 of 10)

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

(10 of 10)

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

## Import, n

(FIFO)

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import, n</td>
<td><img src="image" alt="Import, 59 6F 75 20 61 72" /></td>
</tr>
</tbody>
</table>
Slope

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>Slope,?</td>
<td>?Slope,99.7,100.3</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.

300ms processing delay

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.

Δ0.3% from ideal
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*

300ms processing delay

**Example**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
</table>
| \( T,19.5 \) | ![Wait 300ms](images/wait_300ms.png)  
1  
Dec  
Null |
| \( RT,19.5 \) | ![Wait 900ms](images/wait_900ms.png)  
1  
Dec  
ASCII  
Null |
| \( T,? \) | ![Wait 300ms](images/wait_300ms.png)  
1  
Dec  
ASCII  
Null |

Example Response

- \( T,19.5 \) 8.91
- \( T,? \) 8.91

This is a new command for firmware V2.12

*Default temperature = 25°C*

Temperature is always in Celsius

Temperature is not retained if power is cut

---

53  Copyright © Atlas Scientific LLC
# Device Information

## Command Syntax

- **i**  
- **Wait 300ms**

### Example

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

### Response

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>?i,pH,1.98</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait 300ms</td>
<td>Dec</td>
<td>ASCII</td>
<td>Null</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>

---

300ms processing delay
# Reading device status

## Command syntax

**status** voltage at Vcc pin and reason for last restart

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>Voltage at Vcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>5.038</td>
</tr>
</tbody>
</table>

## Example Response

<table>
<thead>
<tr>
<th>Status</th>
<th>Voltage at Vcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>5.038</td>
</tr>
</tbody>
</table>

## Response breakdown

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

- **?Status,**
- **P,**
- **5.038**

## 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

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
<th>Send any character or command to awaken device.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep</td>
<td>no response</td>
<td>Do not read status byte after issuing sleep command.</td>
</tr>
<tr>
<td>Any command</td>
<td>wakes up device</td>
<td></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 images showing Standby and Sleep modes.
## Protocol lock

### Command syntax

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

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plock,1</td>
<td>1 0</td>
</tr>
<tr>
<td>Plock,0</td>
<td>1 0</td>
</tr>
<tr>
<td>Plock,?</td>
<td>1 ? 0</td>
</tr>
</tbody>
</table>

### Response

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

### Example Response

- 300ms processing delay
- Wait 300ms

### Locks device to I2C mode.

### Diagram

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

---

Copyright © Atlas Scientific LLC
## I²C address change

### Command syntax

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

### Example

<table>
<thead>
<tr>
<th>Command</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).

- Example: `I²C,100`
- (reboot)

- **n = any number 1 – 127**

---

Copyright © Atlas Scientific LLC
Factory reset

Command syntax

Factory enable factory reset

Example

Response

Factory

device reboot

Clears calibration
LED on
Response codes enabled

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

I2C address will not change

(reboot)
Change to UART mode

Command syntax

Baud, n  switch from I²C to UART

Example

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

n =

300
1200
2400
9600
19200
38400
57600
115200

![Image of I²C and UART pins with text](image-url)
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**

![Image of manual switching process]

**Wrong Example**

![Image of wrong manual switching process]
In your CAD software place a 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 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

**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)**
- Added the option to save and load calibration.

**V1.98 – EEPROM (Nov 14, 2016)**
- Fixed glitch during calibration process.

**V2.10 – (May 9, 2017)**
- Added "Find" command.
- Added "Export/import" command.
- Modified continuous mode to be able to send readings every "n" seconds.

**V2.11 – (June 12, 2017)**
- Fixed "I" command to return "pH" instead of "PH".

**V2.12 – (April 16, 2018)**
- Fixed “cal,clear” was not clearing stored calibration in EEPROM.
- Added “RT” command to Temperature compensation.
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.