pH EZO™ Circuit

- Reads: pH
- Range: .001 – 14.000
- Resolution: .001
- Accuracy: +/- 0.002
- Max rate: 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.

Written by Jordan Press
Designed by Noah Press
STOP

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!
Table of contents

Circuit dimensions 4
Power consumption 4
Absolute max ratings 4
EZO™ circuit identification 5
Operating principle 6
Calibration theory 7
Power and data isolation 9
Correct wiring 11
Available data protocols 14

UART

UART mode 16
Default state 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/import calibration 27
Slope 28
Temperature compensation 29
Naming device 30
Device information 31
Response codes 32
Reading device status 33
Sleep mode/low power 34
Change baud rate 35
Protocol lock 36
Factory reset 37
Change to I²C mode 38
Manual switching to I²C 39

I²C

I²C mode 41
Sending commands 42
Requesting data 43
Response codes 44
LED color definition 45
I²C quick command page 46
LED control 47
Find 48
Taking reading 49
Calibration 50
Export/import calibration 51
Slope 52
Temperature compensation 53
Device information 54
Reading device status 55
Sleep mode/low power 56
Protocol lock 57
I²C address change 58
Factory reset 59
Change to UART mode 60
Manual switching to UART 61

Circuit footprint 62
Datasheet change log 63
Warranty 65
**Power consumption**

<table>
<thead>
<tr>
<th></th>
<th>5V</th>
<th>3.3V</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED</td>
<td>MAX</td>
<td>STANDBY</td>
</tr>
<tr>
<td>ON</td>
<td>18.3 mA</td>
<td>16 mA</td>
</tr>
<tr>
<td>OFF</td>
<td>13.8 mA</td>
<td>13.8 mA</td>
</tr>
<tr>
<td></td>
<td>ON</td>
<td>14.5 mA</td>
</tr>
<tr>
<td></td>
<td>OFF</td>
<td>13.3 mA</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>
EZO™ pH circuit identification

**Viewing correct datasheet**

**EZO™ pH circuit**

Click here to view legacy datasheet

**Viewing incorrect datasheet**

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

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.

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-volages 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.

Correct

Incorrect

Basic EZO™ Inline Voltage Isolator

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 R,7) this produces a voltage of 3.7V regardless of your input voltage.

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

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

Incorrect wiring

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

*Only after you are familiar with EZO™ circuits operation

Part # COM-104
Part # ISCCB
Part # USB-ISO

ATLAS
Scientific

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!
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
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
Default state

Mode: UART

Baud: 9,600

Readings: continuous

Speed: 1 reading per second

LED:
- Green: Standby
- Cyan: Taking reading
- Transmitting: on

1,000 ms
Receiving data from device

2 parts

ASCII data string

Command

Carriage return <cr>

Terminator

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)
Sending commands to device

2 parts

Command (not case sensitive)
ASCII data string

Carriage return <cr>
Terminator

sender

CPU

Advanced

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

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

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>+2.2 mA</td>
</tr>
<tr>
<td>3.3V</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 9,600</td>
</tr>
<tr>
<td>C</td>
<td>enable/disable continuous reading</td>
<td>pg. 24 enabled</td>
</tr>
<tr>
<td>Cal</td>
<td>performs calibration</td>
<td>pg. 26 n/a</td>
</tr>
<tr>
<td>Export/import</td>
<td>export/import calibration</td>
<td>pg. 27 n/a</td>
</tr>
<tr>
<td>Factory</td>
<td>enable factory reset</td>
<td>pg. 37 n/a</td>
</tr>
<tr>
<td>Find</td>
<td>finds device with blinking white LED</td>
<td>pg. 23 n/a</td>
</tr>
<tr>
<td>i</td>
<td>device information</td>
<td>pg. 31 n/a</td>
</tr>
<tr>
<td>I2C</td>
<td>change to I²C mode</td>
<td>pg. 38 not set</td>
</tr>
<tr>
<td>L</td>
<td>enable/disable LED</td>
<td>pg. 22 enabled</td>
</tr>
<tr>
<td>Name</td>
<td>set/show name of device</td>
<td>pg. 30 not set</td>
</tr>
<tr>
<td>Plock</td>
<td>enable/disable protocol lock</td>
<td>pg. 36 disabled</td>
</tr>
<tr>
<td>R</td>
<td>returns a single reading</td>
<td>pg. 25 n/a</td>
</tr>
<tr>
<td>Sleep</td>
<td>enter sleep mode/low power</td>
<td>pg. 34 n/a</td>
</tr>
<tr>
<td>Slope</td>
<td>returns the slope of the pH probe</td>
<td>pg. 28 n/a</td>
</tr>
<tr>
<td>Status</td>
<td>retrieve status information</td>
<td>pg. 33 enable</td>
</tr>
<tr>
<td>T</td>
<td>temperature compensation</td>
<td>pg. 29 25˚C</td>
</tr>
<tr>
<td>*OK</td>
<td>enable/disable response codes</td>
<td>pg. 32 enable</td>
</tr>
</tbody>
</table>

All commands are ASCII strings or single ASCII characters.
**LED control**

**Command syntax**

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

**Example**

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

---

**Diagrams**

L,1

L,0
Find

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

Find <cr>  LED rapidly blinks white, used to help find device*
*This command is only available for firmware version 2.10 and above.

Example

Find <cr>

Response

*OK <cr>

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

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

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

## Example

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

\[ R \text{ <cr> takes single reading} \]

## Example

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

## Response

- **Green**: Standby
- **Cyan**: Taking reading
- **Transmitting**

![Diagram of device states](image)

- 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

<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;</td>
</tr>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>
## Export/import calibration

### Command syntax

**Export**  <cr>  export calibration string from calibrated device*

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

**Export,** ?  <cr>  calibration string info*

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

### Example

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

### Response breakdown

- **# of strings to export**: 10
- **# of bytes to export**: 120

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

Disabling *OK simplifies this process.

---

Export: Use this command to save calibration settings

Import: Use this command to load calibration settings to one or more devices.
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;</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.

\(\Delta 0.3\% \text{ from ideal}\)

Zero point

1 2 3 4 5 6 7 8 9

Calibrated probe

Ideal 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 &lt;cr&gt;</td>
<td>n = any value; floating point or int</td>
</tr>
<tr>
<td>T,? &lt;cr&gt;</td>
<td>compensated temperature value?</td>
</tr>
</tbody>
</table>

**Temperature is always in Celsius**

## Example

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

## Example Response

```
8.82
```

**Diagram:**

![Diagram showing temperature compensation](https://example.com/diagram.png)
## Naming device

### Command syntax

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

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

Up to 16 ASCII characters

### Example

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

### Example

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

## Command syntax

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

## Example

<table>
<thead>
<tr>
<th>i &lt;cr&gt;</th>
<th>?i,pH,1.98 &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>?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

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

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

## 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 &lt;cr&gt;</th>
<th>Reason for restart</th>
<th>Voltage at Vcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>?Status, P, 5.038 &lt;cr&gt;</td>
<td>Reason for restart</td>
<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.

<table>
<thead>
<tr>
<th>Sleep &lt;cr&gt; enter sleep mode/low power</th>
</tr>
</thead>
</table>

**Example**

<table>
<thead>
<tr>
<th>Sleep &lt;cr&gt;</th>
<th>*SL</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Any command</th>
<th>*WA &lt;cr&gt; wakes up device</th>
</tr>
</thead>
</table>

### Currents

<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
# Change baud rate

## Command syntax

[Baud,n <cr>] change baud rate

## Example

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

n =

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

---

*Example Response*

```
*OKBaud,38400
```

---

*Example Response*

```
*OK
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
Baud,38400 <cr>
```

---

*Example Response*

```
Baud,38400 <cr>
*OK <cr>
```

---

---

*Example Response*

```
Baud,38400 <cr>
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```

---

---

*Example Response*

```
*OK <cr>
```
**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><code>Plock,1 &lt;cr&gt;</code></td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td><code>Plock,0 &lt;cr&gt;</code></td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td><code>Plock,? &lt;cr&gt;</code></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>
- `I2C,100` ⇒ cannot change to I²C

Locks device to UART mode.
Factory reset

**Command syntax**

Factory <cr> enable factory reset

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

Factory <cr> enable factory reset

(reboot)

*OK <cr>

*RS <cr>

*RE <cr>

Baud rate will not change
Change to I\textsuperscript{2}C mode

**Command syntax**

I\textsuperscript{2}C,\(n\) \texttt{<cr>} sets I\textsuperscript{2}C address and reboots into I\textsuperscript{2}C mode

\(n = \text{any number } 1 - 127\)

**Example**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>I\textsuperscript{2}C,100 \texttt{&lt;cr&gt;}</td>
<td>*OK (reboot in I\textsuperscript{2}C mode)</td>
</tr>
</tbody>
</table>

**Wrong example**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>I\textsuperscript{2}C,139 \texttt{&lt;cr&gt;}</td>
<td>*ER \texttt{&lt;cr&gt;}</td>
</tr>
</tbody>
</table>

I\textsuperscript{2}C,100

Green

*OK \texttt{<cr>}

(reboot)

Blue

now in I\textsuperscript{2}C mode

Default I\textsuperscript{2}C 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

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
I²C mode

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

Vcc  3.3V – 5.5V

Clock speed  100 – 400 kHz

SDA  

SCL  

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

99 (0x63) | ASCII command string

Example

Start | 99 (0x63) | Write | Sleep | Stop

I²C address | Command

Advanced

Address bits

The entire command as ASCII with all arguments

First letter of command | \ACK | \ACK | \ACK

... | ... | ... | ...

Last letter of command | \ACK

W = low

Stop
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

9.560

1 57 46 53 54 48 0 = 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.

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

Copyright © Atlas Scientific LLC
LED color definition

Blue
I²C standby

Green
Taking reading

Purple
Changing I²C ID#

Red
Command not understood

White
Find

<table>
<thead>
<tr>
<th>Voltage</th>
<th>LED ON</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>+2.2 mA</td>
<td></td>
</tr>
<tr>
<td>3.3V</td>
<td>+0.6 mA</td>
<td></td>
</tr>
</tbody>
</table>
I²C mode
command quick reference

All commands are ASCII strings or single ASCII characters.

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud</td>
<td>switch back to UART mode</td>
<td>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

**Example**

- **L,1**
  - Wait 300ms
  - 1 Dec
  - 0 Null

- **L,0**
  - Wait 300ms
  - 1 Dec
  - 0 Null

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

### Diagrams

![Diagram of L,1](image1.png)

![Diagram of L,0](image2.png)
Find

**Command syntax**

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

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

**Example**

Find <cr>

**Response**

Send any character or command to terminate find.

Send any character or command to terminate find.

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

---

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

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

---

300ms processing delay

This command will disable continuous mode

Send any character or command to terminate find.
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 circuit board with green LED indicating &quot;Taking reading&quot; and a clock icon indicating &quot;Wait 900ms&quot;].</td>
</tr>
</tbody>
</table>

Example Response:

- **R**
  - **Wait 900ms**
  - **1**
  - **Dec**
  - **9.560**
  - **ASCII**
  - **0**
  - **Null**
## Calibration

### Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cal,mid,n</strong></td>
<td>single point calibration at midpoint</td>
</tr>
<tr>
<td><strong>Cal,low,n</strong></td>
<td>two point calibration at lowpoint</td>
</tr>
<tr>
<td><strong>Cal,high,n</strong></td>
<td>three point calibration at highpoint</td>
</tr>
<tr>
<td><strong>Cal,clear</strong></td>
<td>delete calibration data</td>
</tr>
<tr>
<td><strong>Cal,?</strong></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="Wait 900ms" /> 1 0 Dec Null</td>
</tr>
<tr>
<td>Cal,low,4.00</td>
<td><img src="image" alt="Wait 900ms" /> 1 0 Dec Null</td>
</tr>
<tr>
<td>Cal,high,10.00</td>
<td><img src="image" alt="Wait 900ms" /> 1 0 Dec Null</td>
</tr>
<tr>
<td>Cal,clear</td>
<td><img src="image" alt="Wait 300ms" /> 1 0 Dec Null</td>
</tr>
<tr>
<td>Cal,?</td>
<td>1 ?Cal,0 0 Dec ASCII Null or 1 ?Cal,1 0 Dec ASCII one point or 1 ?Cal,2 0 Dec ASCII two point or 1 ?Cal,3 0 Dec ASCII three point</td>
</tr>
</tbody>
</table>
### Export/import calibration

**Command syntax**

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

**Export**

- Export calibration string from calibrated device*

**Import**

- Import calibration string to new device*

**Export,**

- Export calibration string info*

#### 300ms processing delay

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

### Example

#### Response breakdown

- **# of strings to export**: 1
- **# of bytes to export**: 120

Export strings can be up to 12 characters long

#### Example Response

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

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
<th>Response breakdown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export,?</td>
<td>( \text{Wait 300ms} )</td>
<td>( \begin{array}{ccc} 1 &amp; 10,120 &amp; 0 \ \text{Dec} &amp; \text{ASCII} &amp; \text{Null} \end{array} )</td>
</tr>
<tr>
<td>Export</td>
<td>( \text{Wait 300ms} )</td>
<td>( \begin{array}{ccc} 1 &amp; 59 \ 6F \ 75 \ 20 \ 61 \ 72 &amp; 0 \ \text{Dec} &amp; \text{ASCII} &amp; \text{Null} \end{array} )</td>
</tr>
<tr>
<td>Export</td>
<td>( \text{Wait 300ms} )</td>
<td>( \begin{array}{ccc} 1 &amp; 65 \ 20 \ 61 \ 20 \ 63 \ 6F &amp; 0 \ \text{Dec} &amp; \text{ASCII} &amp; \text{Null} \end{array} )</td>
</tr>
<tr>
<td>Export</td>
<td>( \text{Wait 300ms} )</td>
<td>( \begin{array}{ccc} 1 &amp; *\text{DONE} &amp; 0 \ \text{Dec} &amp; \text{ASCII} &amp; \text{Null} \end{array} )</td>
</tr>
<tr>
<td>Import, n</td>
<td>Import, 59 \ 6F \ 75 \ 20 \ 61 \ 72</td>
<td>(1 of 10)</td>
</tr>
</tbody>
</table>

(Row 1 of 8 more)

(Row 2 of 10 of 10)

(Row 3 of 10 of 10)

(Row 4 of 10 of 10)
Slope

Command syntax

Slope,? returns the slope of the pH probe

Example

<table>
<thead>
<tr>
<th>Slop,?</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</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.

Δ0.3% from ideal

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.

300ms processing delay

Wait 300ms

Slope,?

99.7, 100.3

Dec

ASCII

Null

AtlasScientific

Environmental Robotics
Temperature compensation

Command syntax

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

$T,? \quad \text{compensated temperature value?}$

Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T,19.5$</td>
<td><img src="Image1" alt="Diagram" /></td>
</tr>
<tr>
<td>$T,?$</td>
<td><img src="Image2" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Example Response

$T,19.5$

$8.82$

$\text{Wait 300ms}$

$1 \quad \text{Dec}$

$0 \quad \text{Null}$

$T,19.5$

$8.91$

$\text{Wait 300ms}$

$1 \quad \text{Dec}$

$?T,19.5 \quad 0$

$\text{ASCII}$

$Null$

300ms processing delay

Temperature is always in Celsius

Copyright © Atlas Scientific LLC
# Device information

**Command syntax**

- **Wait 300ms**

- `i` device information

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

<table>
<thead>
<tr>
<th>ASCII</th>
<th>Dec</th>
<th>Firmware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null</td>
<td>1</td>
<td>1.98</td>
</tr>
</tbody>
</table>

**Response breakdown**

- `?i, pH, 1.98`

  - Device: `1`
  - Firmware: `1.98`
# Reading device status

## Command syntax

<table>
<thead>
<tr>
<th>Status 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>

#### Restart codes

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

### Response breakdown

- **?Status, P, 5.038**: Reason for restart
- **5.038**: Voltage at Vcc

300ms processing delay

Copyright © Atlas Scientific LLC
# Sleep mode/low power

**Command syntax**

<table>
<thead>
<tr>
<th>Sleep</th>
<th>enter sleep mode/low power</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Send any character or command to awaken device.</strong></td>
<td></td>
</tr>
</tbody>
</table>

### Example

<table>
<thead>
<tr>
<th>Sleep</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>

### Response

<table>
<thead>
<tr>
<th>Voltage (V)</th>
<th>Standby (mA)</th>
<th>Sleep (mA)</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 (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><img src="Wait300ms.png" alt="Wait 300ms" /></td>
</tr>
<tr>
<td></td>
<td><img src="1.png" alt="1" /> <img src="0.png" alt="0" /></td>
</tr>
<tr>
<td></td>
<td>Dec  Null</td>
</tr>
<tr>
<td>Plock,0</td>
<td><img src="Wait300ms.png" alt="Wait 300ms" /></td>
</tr>
<tr>
<td></td>
<td><img src="1.png" alt="1" /> <img src="0.png" alt="0" /></td>
</tr>
<tr>
<td></td>
<td>Dec  Null</td>
</tr>
<tr>
<td>Plock,?</td>
<td><img src="Wait300ms.png" alt="Wait 300ms" /></td>
</tr>
<tr>
<td></td>
<td><img src="1.png" alt="1" /> <img src="?" alt="?" /> <img src="0.png" alt="0" /></td>
</tr>
<tr>
<td></td>
<td>Dec  ASCII  Null</td>
</tr>
</tbody>
</table>

Locks device to I²C mode.

---

Copyright © Atlas Scientific LLC

---

Plock,1 Serial, 9600 cannot change to UART

Serial, 9600 cannot change to UART
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 | Response
--- | ---
I²C,100 | device reboot

I²C,n sets I²C address and reboots into I²C mode

n = any number 1 – 127
## 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>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory</td>
<td>device reboot</td>
</tr>
</tbody>
</table>

**Clears calibration**

**LED on**

**Response codes enabled**

---

**Factory**

![Image of device before and after factory reset]

(reboot)

---

Copyright © Atlas Scientific LLC
# Change to UART mode

## Command syntax

Baud,\(n\)  switch from \(\text{i}^2\text{C}\) to UART

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

\(n = \begin{cases} 
300 \\
1200 \\
2400 \\
9600 \\
19200 \\
38400 \\
57600 \\
115200 
\end{cases}\)

![Diagram of I2C standby](image1)

Serial,9600  \rightarrow  Changing to UART mode  \rightarrow  (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 an 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 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 & 46.
"Export/Import calibration" pages 27 & 49.
   Added new feature to continous 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.
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.1</td>
<td>Oct 30, 2014</td>
<td>Change output to mg/L, then percentage (was previously percentage, then mg/L)</td>
</tr>
<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>Fix 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 glitch 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 glitch 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>
</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.