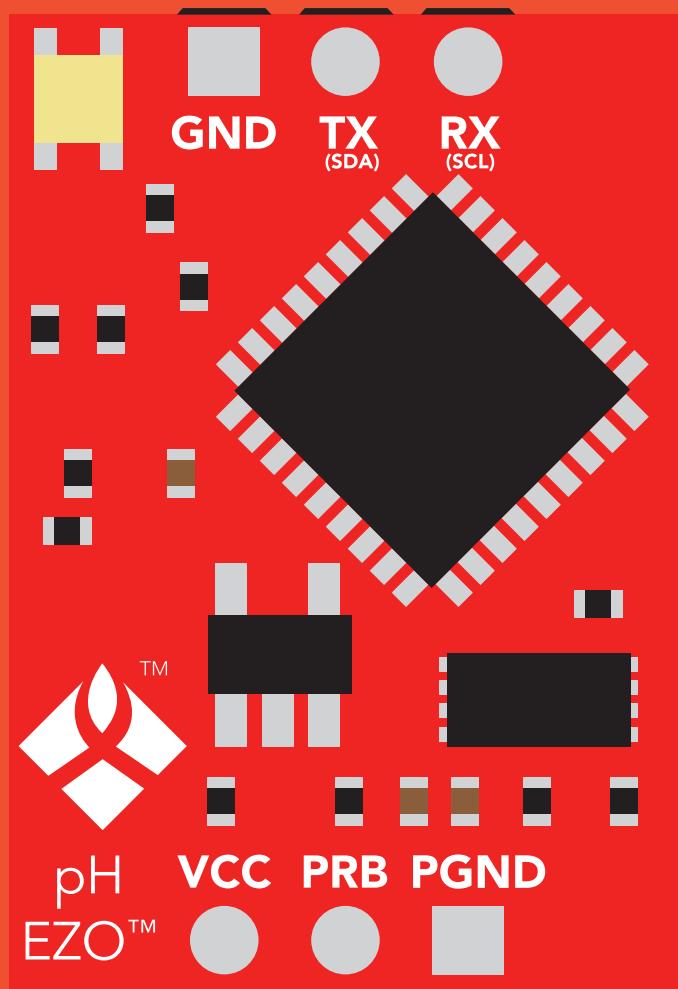


# pH EZO™ Circuit

Reads	<b>pH</b>
Range	<b>.001 – 14.000</b>
Resolution	<b>.001</b>
Accuracy	<b>+/- 0.002</b>
Max rate	<b>1 reading per sec</b>
Supported probes	<b>Any type &amp; brand</b>
Calibration	<b>1, 2, 3 point</b>
Temp compensation	<b>Yes</b>
Data protocol	<b>UART &amp; I<sup>2</sup>C</b>
Default I <sup>2</sup> C address	<b>99 (0x63)</b>
Operating voltage	<b>3.3V – 5V</b>
Data format	<b>ASCII</b>



**PATENT PROTECTED**



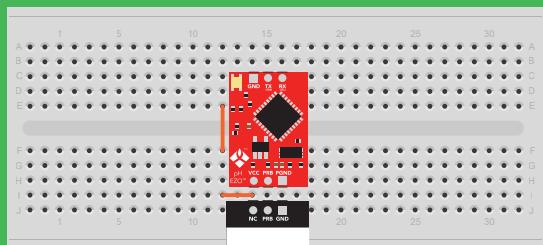
# STOP

**SOLDERING THIS DEVICE VOIDS YOUR WARRANTY.**

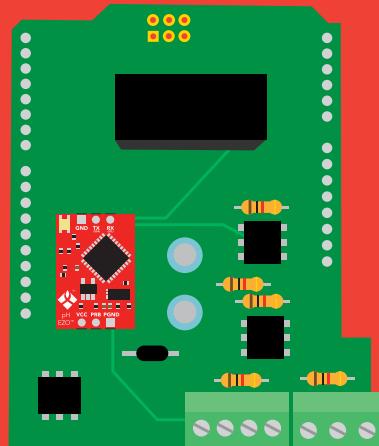
**This is sensitive electronic equipment. Get this device working in a solderless breadboard first. Once this device has been soldered it is no longer covered by our warranty.**

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

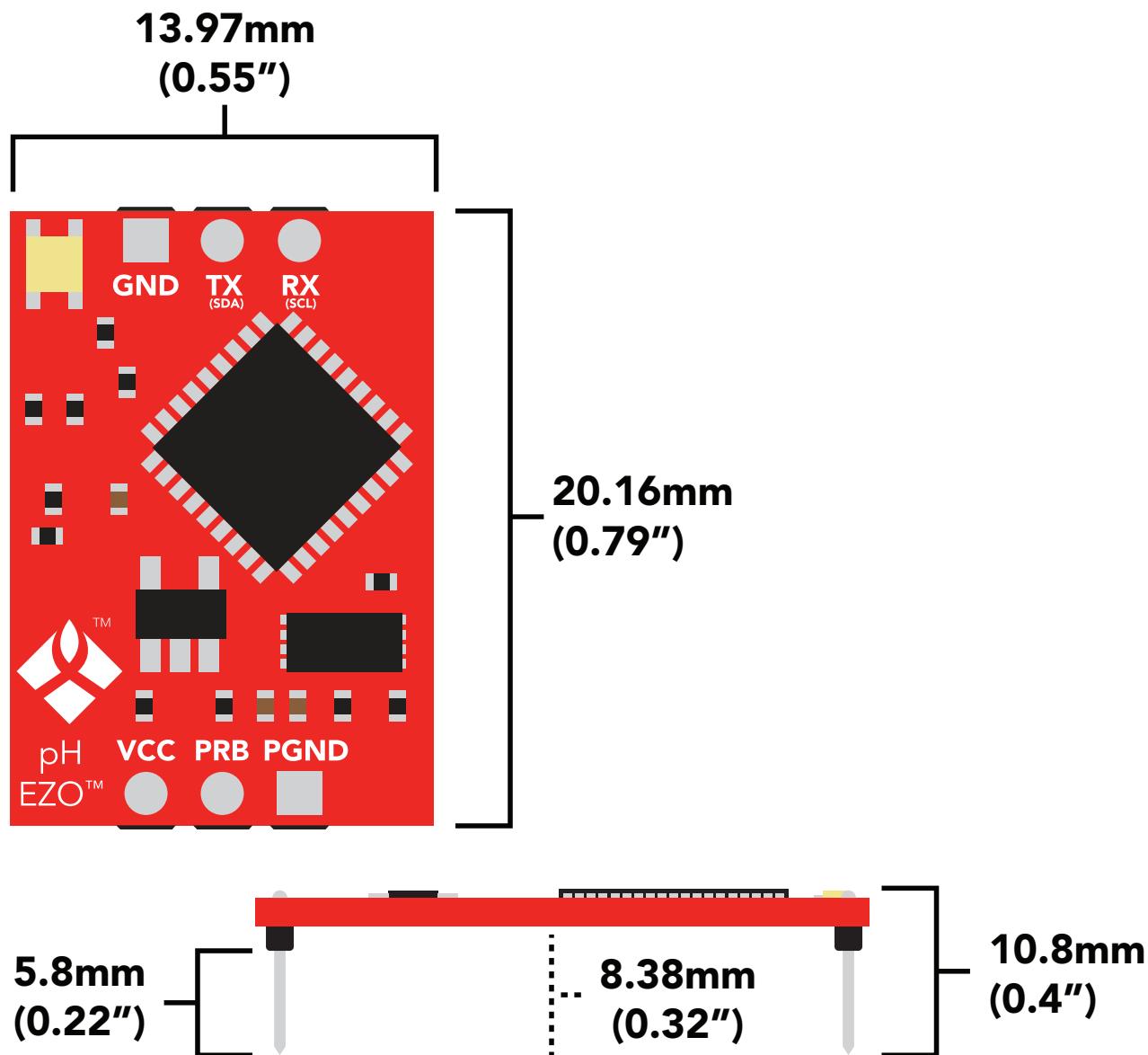
## UART

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 I <sup>2</sup> C mode	38
Manual switching to I <sup>2</sup> C	39

## I<sup>2</sup>C

I <sup>2</sup> C mode	41
Sending commands	42
Requesting data	43
Response codes	44
LED color definition	45
I <sup>2</sup> C 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
I <sup>2</sup> C 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

# EZO™ circuit dimensions



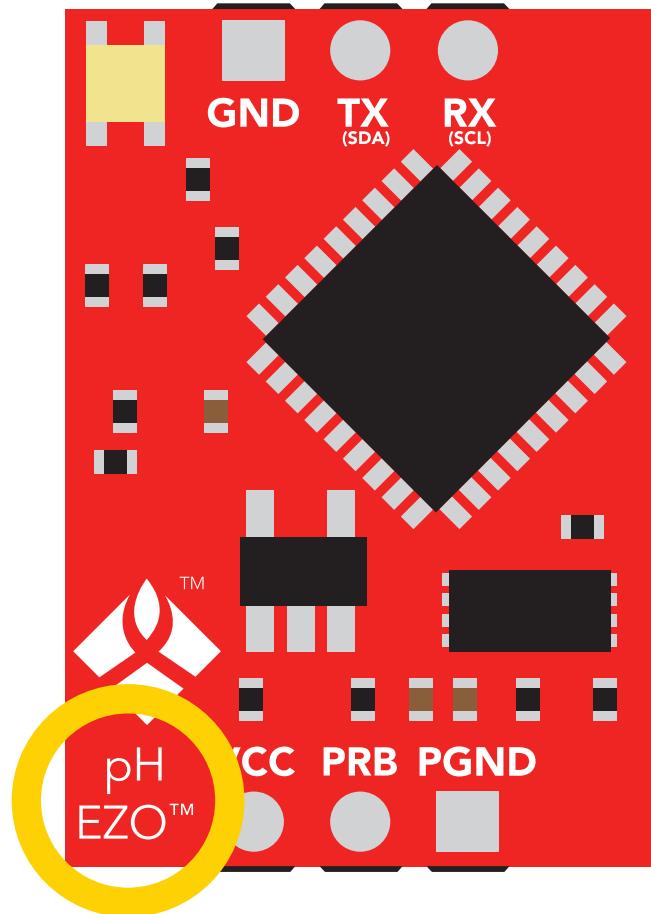
## Power consumption

	LED	MAX	STANDBY	SLEEP
5V	ON	18.3 mA	16 mA	1.16 mA
	OFF	13.8 mA	13.8 mA	
3.3V	ON	14.5 mA	13.9 mA	0.995 mA
	OFF	13.3 mA	13.3 mA	

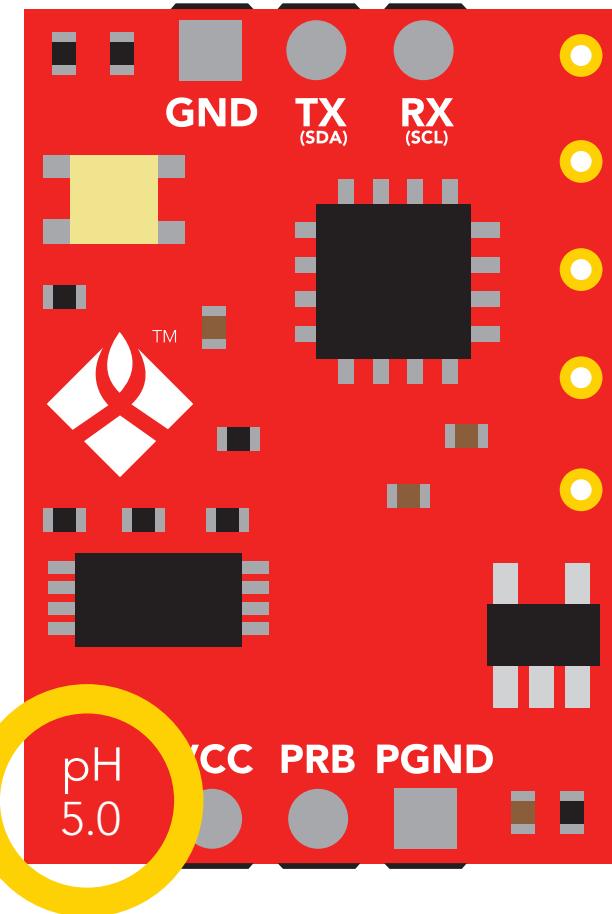
## Absolute max ratings

	Parameter	MIN	TYP	MAX
	Storage temperature (EZO™ pH)	-65 °C		125 °C
	Operational temperature (EZO™ pH)	-40 °C	25 °C	85 °C
	VCC	3.3V	5V	5.5V

# EZO™ circuit identification



EZO™ pH circuit



Legacy pH circuit



Viewing correct datasheet

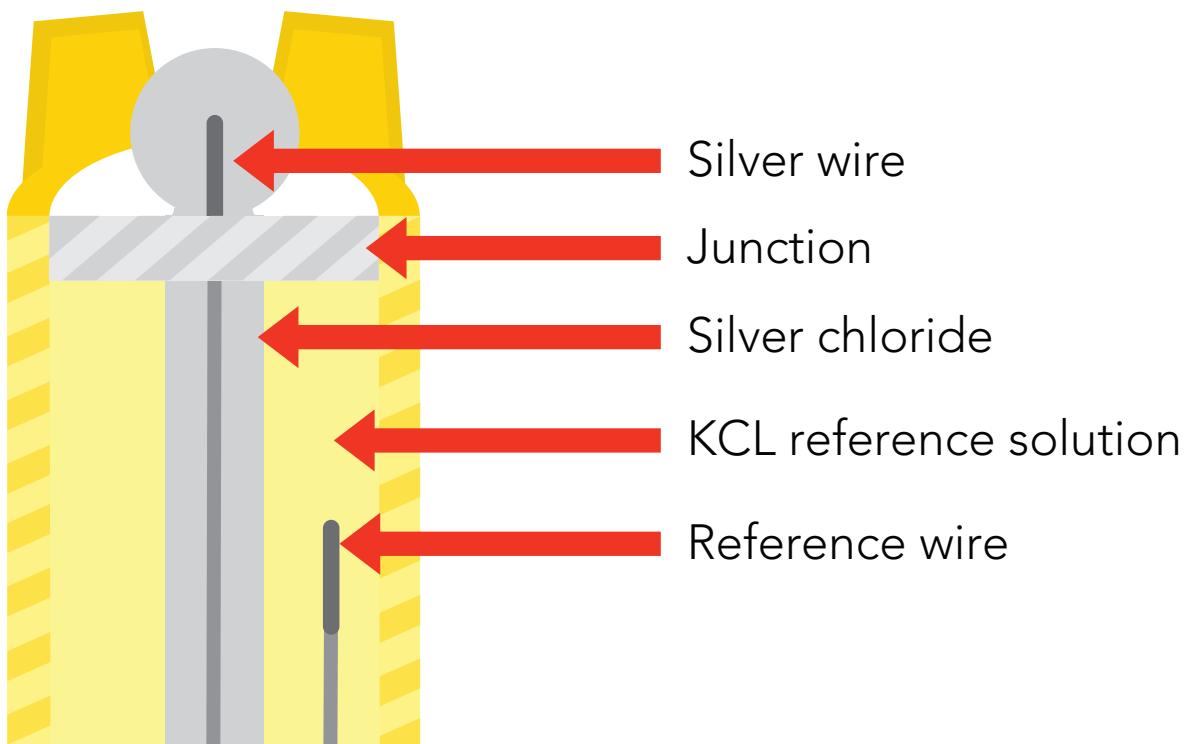
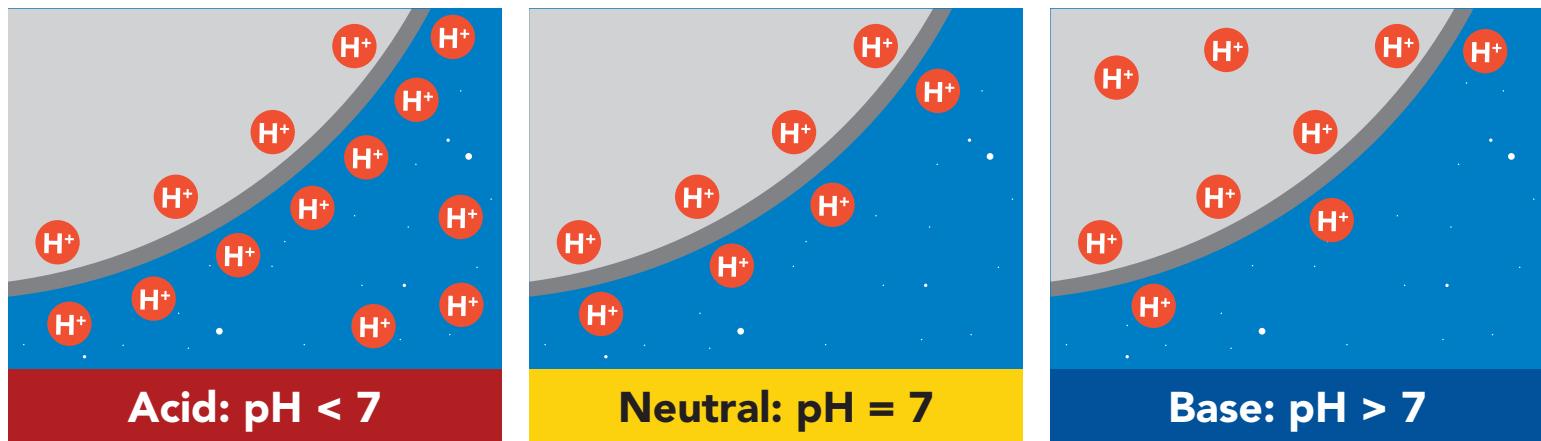


Viewing incorrect datasheet

[Click here to view legacy datasheet](#)

# Operating principle

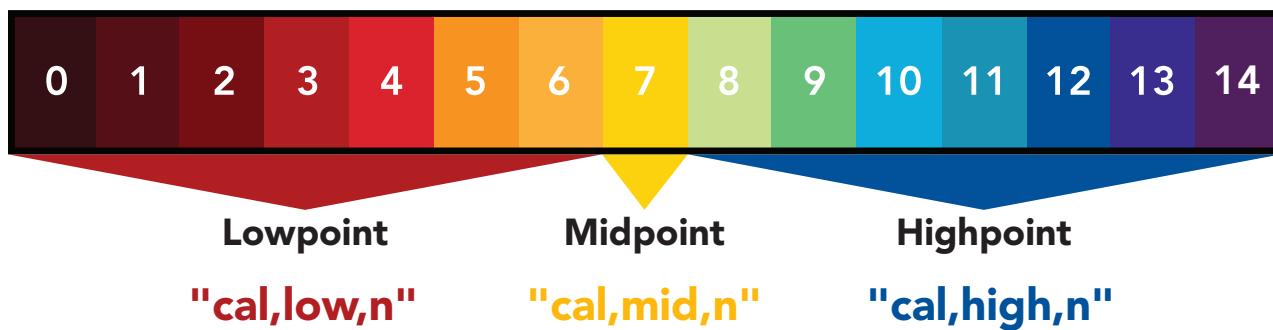
A pH (**potential of Hydrogen**) probe measures the hydrogen ion activity in a liquid. At the tip of a pH probe is a glass membrane. This glass membrane permits hydrogen ions from the liquid being measured to diffuse into the outer layer of the glass, while larger ions remain in the solution. The difference 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 I<sup>2</sup>C mode after calibration **will not** affect the stored calibration. If the device must be calibrated in I<sup>2</sup>C mode be sure to request readings continuously so you can see the output from the probe.

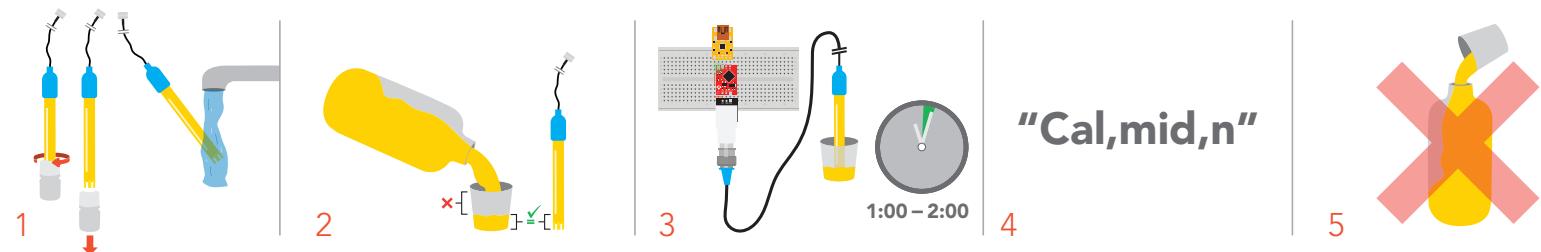
The Atlas Scientific EZO™ class pH circuit has a flexible calibration protocol, allowing for **single point**, **two point**, or **three point** calibration.



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

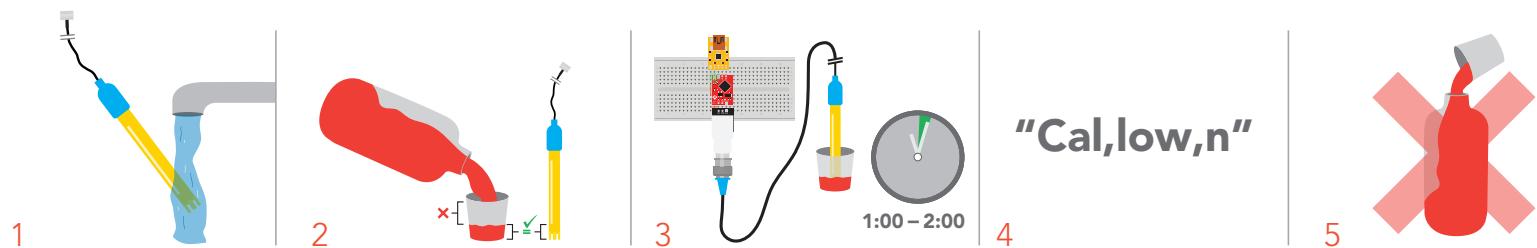
The EZO™ pH circuits default temperature compensation is set to 25° C. If the temperature of the calibration solution is +/- 2° C from 25° C, consider setting the temperature compensation first. **Temperature changes of < 2° C are insignificant.**

## Single point calibration



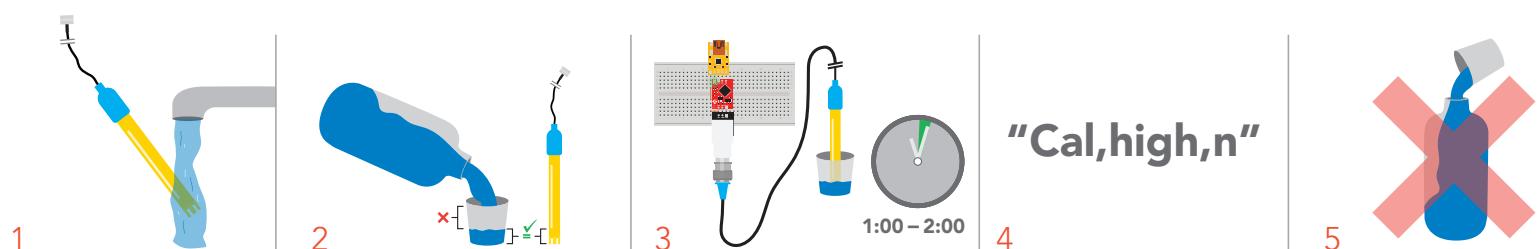
1. Remove soaker bottle and rinse off pH probe.
2. Pour a small amount of the calibration solution into a cup.
3. Let the probe sit in calibration solution until readings stabilize (1 – 2 minutes).
4. Calibrate the midpoint value using the command "**Cal,mid,n**".  
Where "n" is any floating point value 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

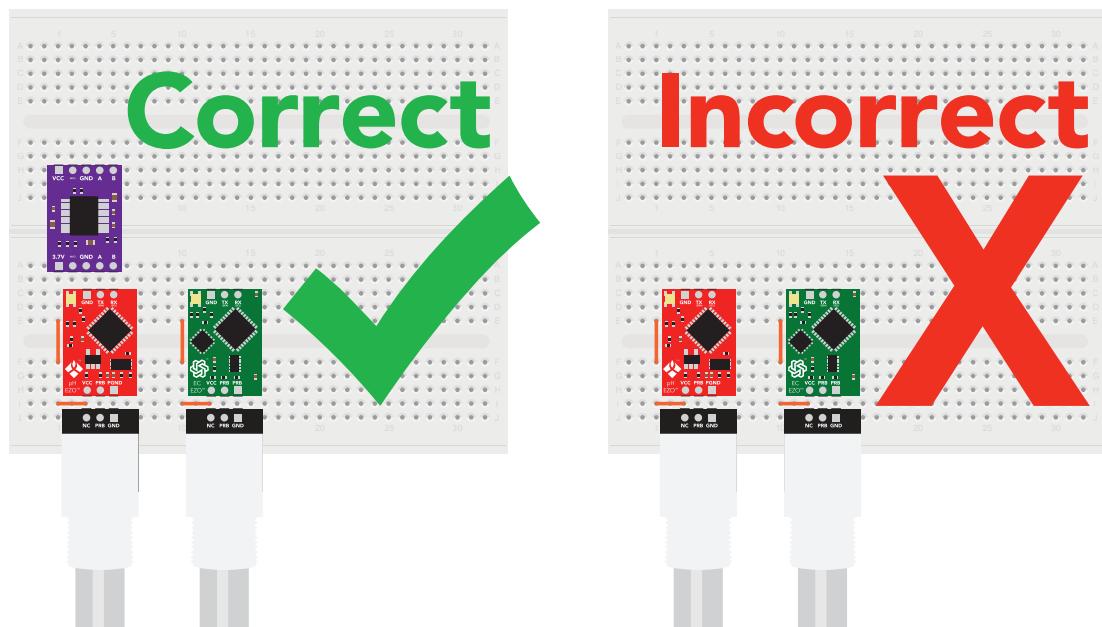
The Atlas Scientific EZO™ pH circuit is a very sensitive device. This sensitivity is what gives the pH circuit its accuracy. This also means that the pH circuit is capable of reading micro-voltages that are bleeding into the water from unnatural sources such as pumps, solenoid valves or other probes/sensors.

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.

Basic EZO™  
Inline Voltage Isolator

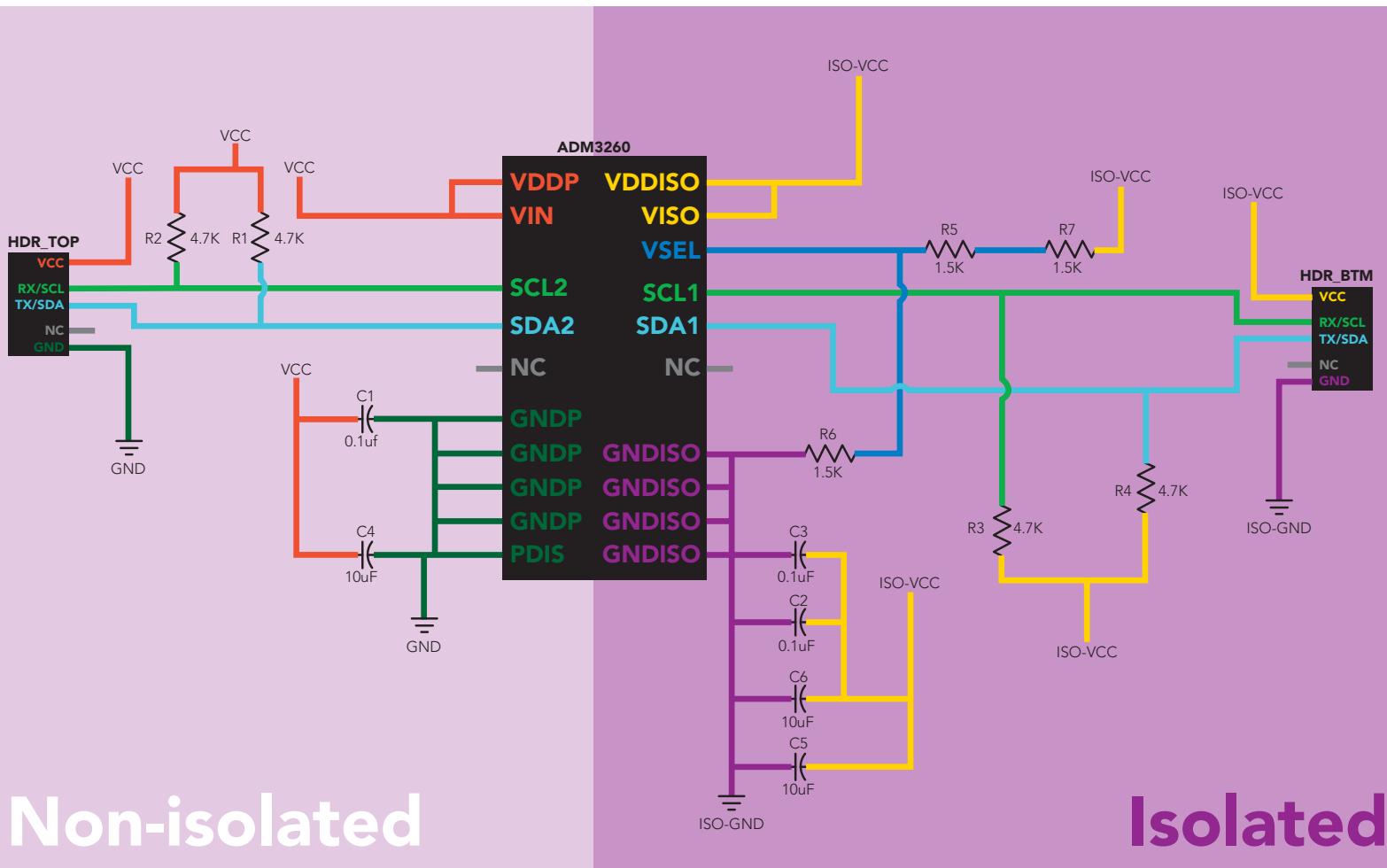


**Without isolation, Conductivity and Dissolved Oxygen  
readings will effect pH accuracy.**

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.

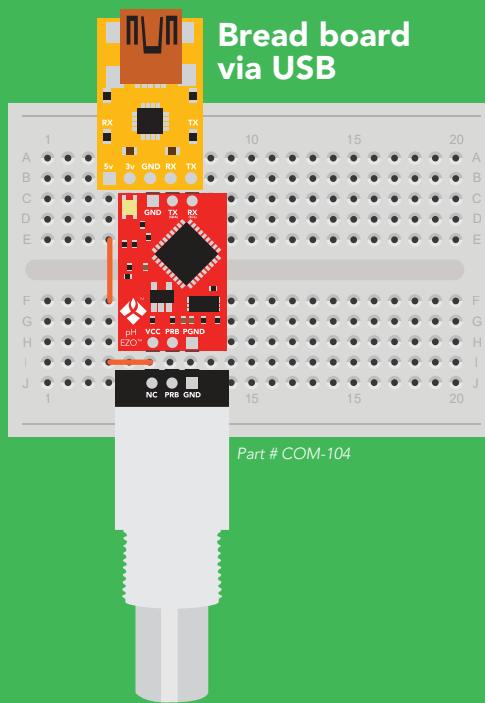
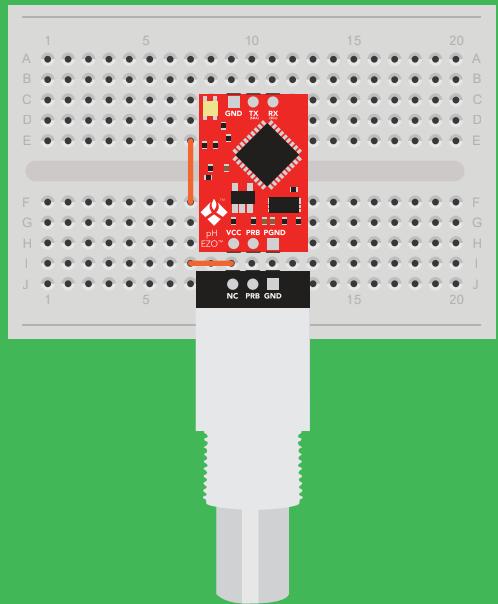
This technology works by using tiny transformers to induce the voltage across an air gap. PCB layout requires special attention for EMI/EMC and RF Control, having proper ground planes and keeping the capacitors as close to the chip as possible are crucial for proper performance. The two data channels have a  $4.7\text{k}\Omega$  pull up resistor on both the isolated and non-isolated lines (R1, R2, R3, and R4) The output voltage is set using a voltage divider (R5, R6, and R7) this produces a voltage of 3.7V regardless of your input voltage.

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

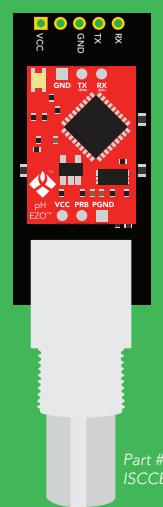


# ✓ Correct wiring

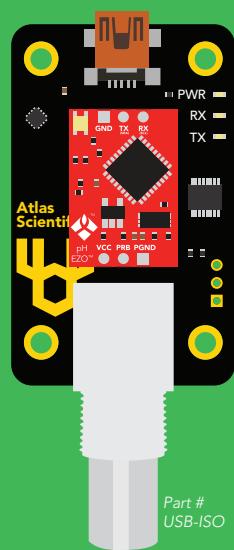
Bread board



Carrier board



USB carrier board

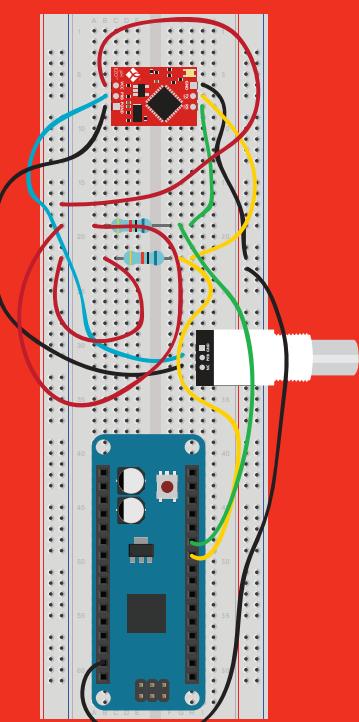


# ✗ Incorrect wiring

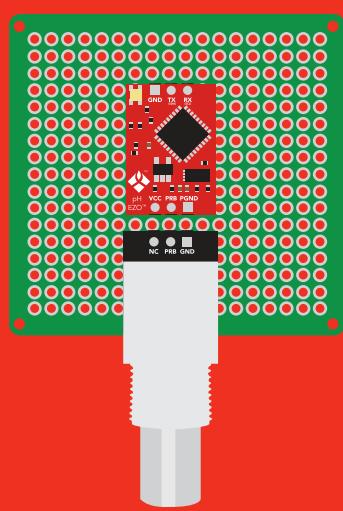
Extended leads



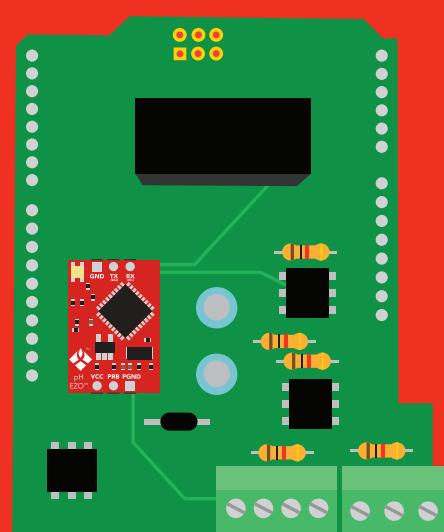
Sloppy setup



Perfboards or Protoboards



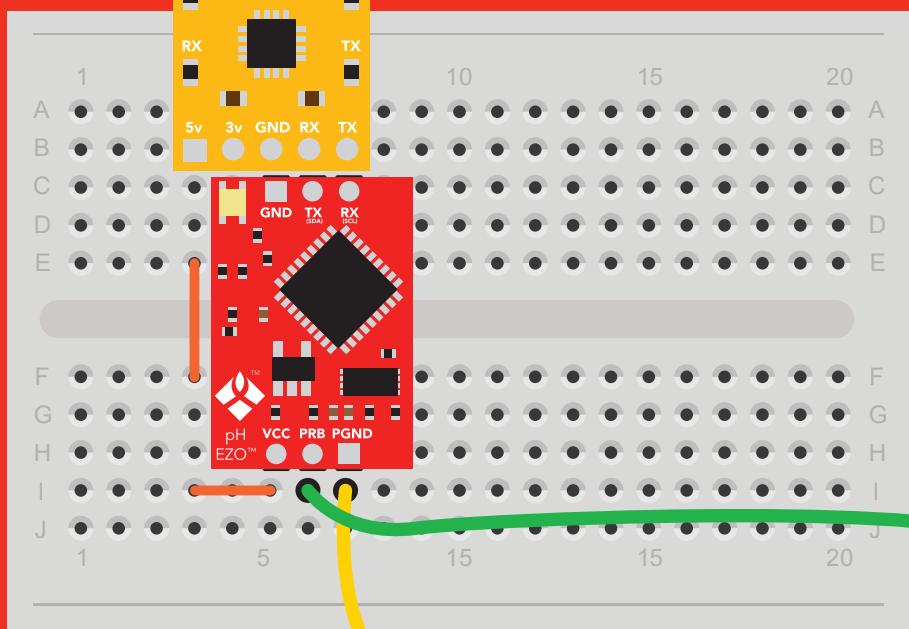
\*Embedded into your device



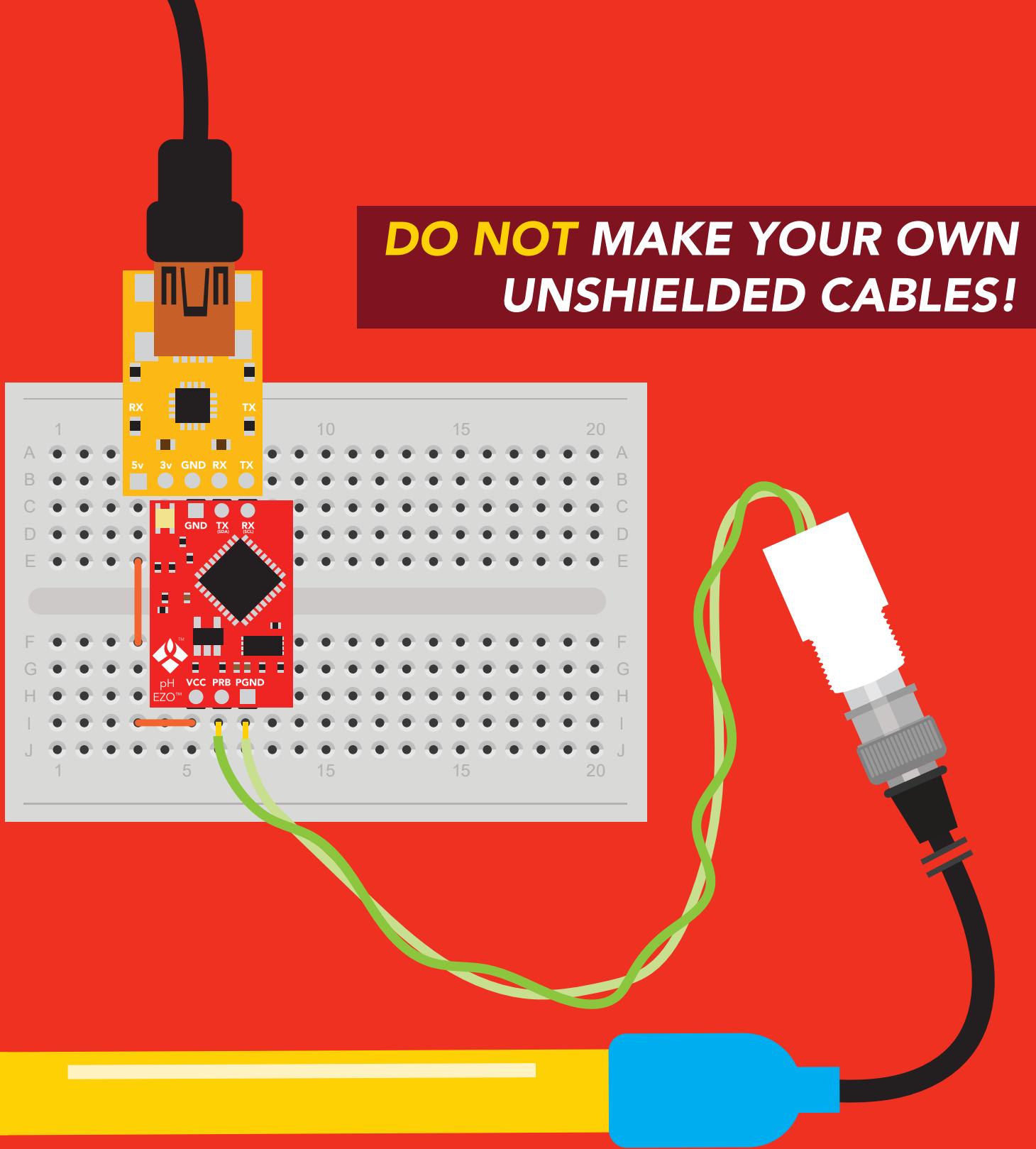
**NEVER**  
use Perfboards  
or Protoboards

\*Only after you are familiar  
with EZO™ circuits operation

**NEVER EXTEND THE CABLE  
WITH CHEAP JUMPER WIRES!**



**DO NOT CUT THE PROBE CABLE  
WITHOUT REFERING TO **THIS DOCUMENT!****



**ONLY USE SHIELDED CABLES.  
REFER TO **THIS DOCUMENT!****

 Available data protocols

**UART**

**Default**

**I<sup>2</sup>C**

 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 I<sup>2</sup>C mode  
LED control  
Protocol lock  
Software switch to I<sup>2</sup>C mode

## Settings that are **NOT** retained if power is cut

Find  
Sleep mode

# UART mode

8 data bits  
1 stop bit

no parity  
no flow control

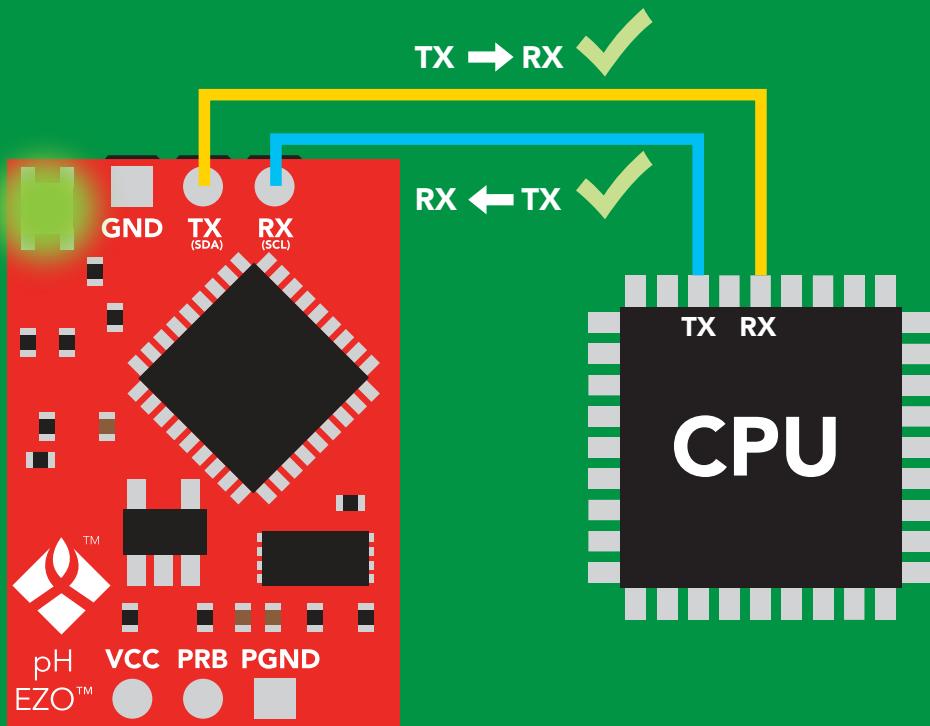
Baud 300  
1,200  
2,400  
**9,600 default**  
19,200  
38,400  
57,600  
115,200

**RX** Data in

**TX** Data out

**Vcc** 3.3V – 5.5V

0V  0V



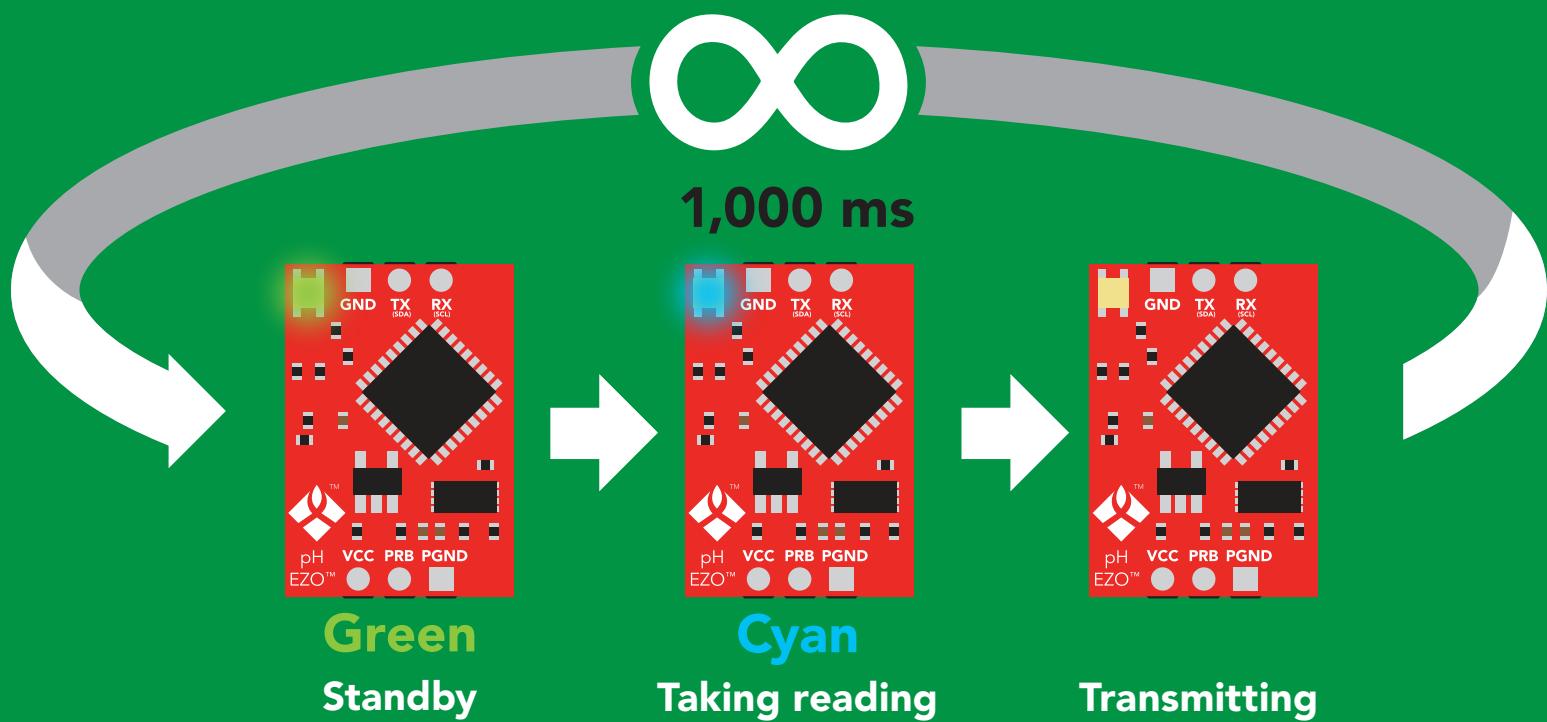
## Data format

Reading	pH
Units	pH
Encoding	ASCII
Format	string
Terminator	carriage return

Data type	<b>floating point</b>
Decimal places	<b>3</b>
Smallest string	<b>4 characters</b>
Largest string	<b>40 characters</b>

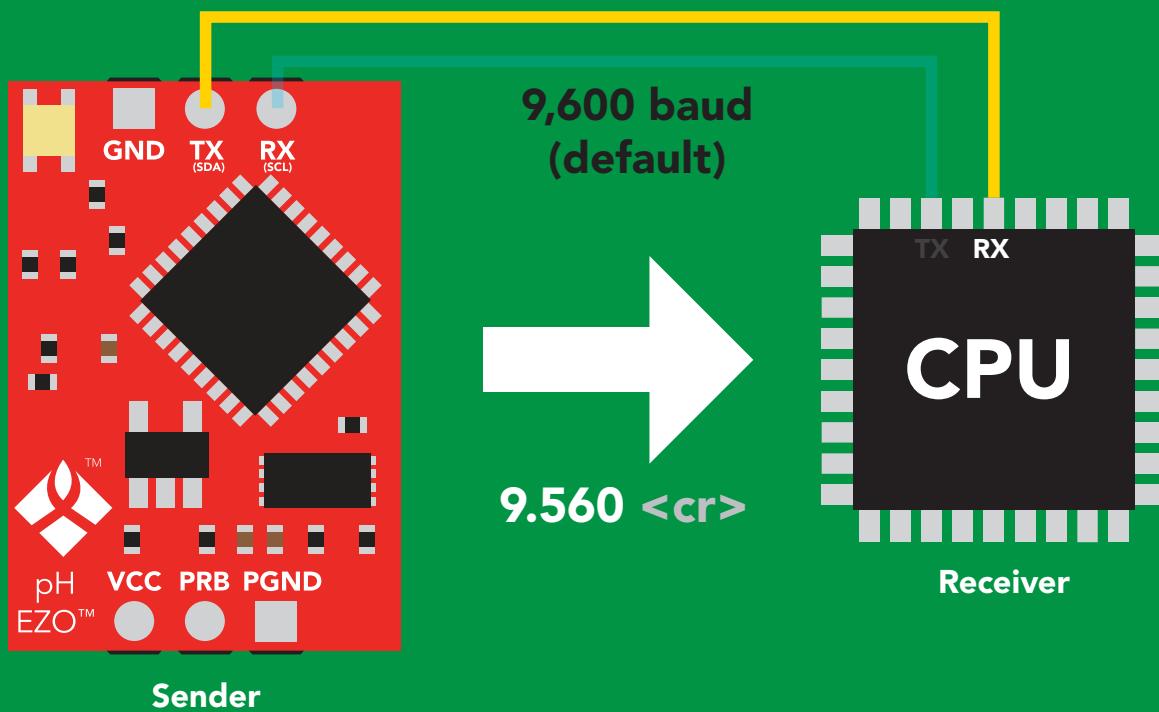
# Default state

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



# Receiving data from device

2 parts



## Advanced

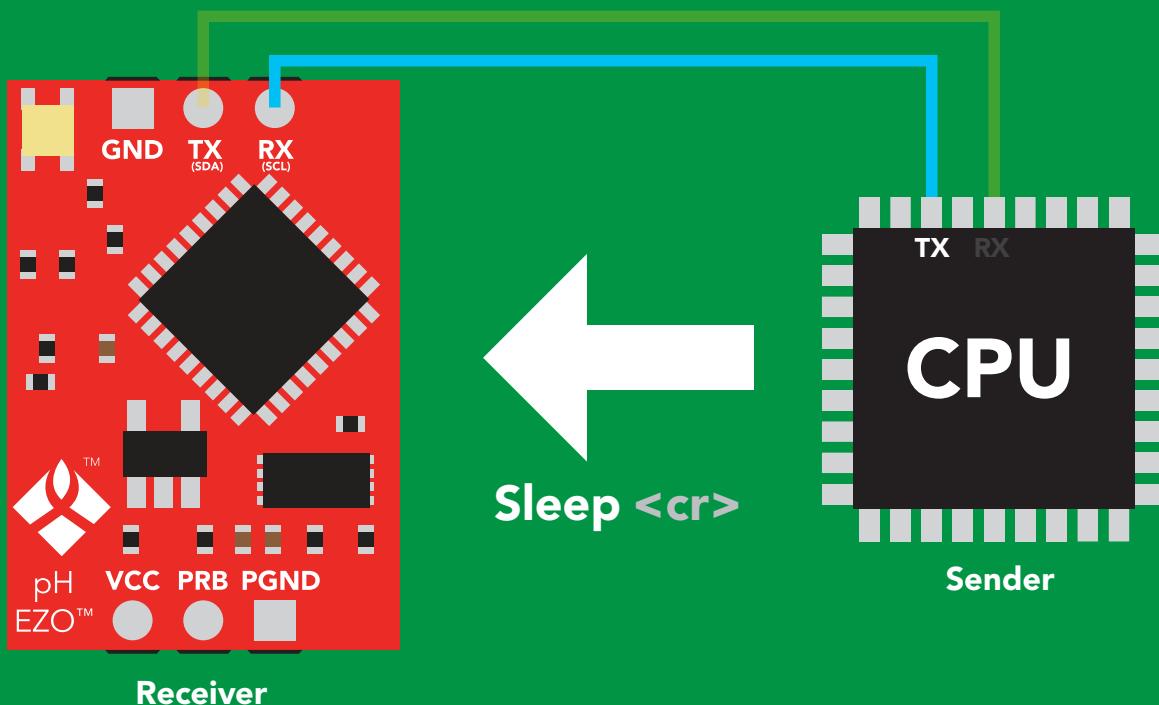
ASCII: 9 . 5 6 0 <cr>

Hex: 39 2E 35 36 30 0D

Dec: 57 46 53 54 48 13

# Sending commands to device

2 parts



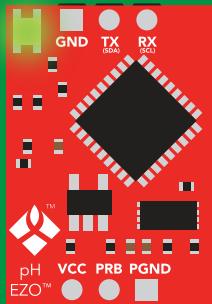
## Advanced

ASCII: S I e e p <cr>

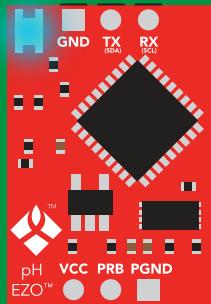
Hex: 53 6C 65 65 70 0D

Dec: 83 108 101 101 112 13

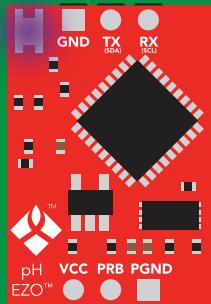
# LED color definition



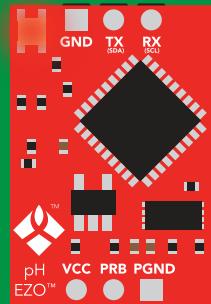
**Green**  
UART standby



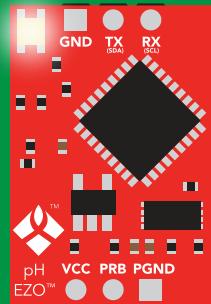
**Cyan**  
Taking reading



**Purple**  
Changing baud rate



**Red**  
Command  
not understood



**White**  
Find

<b>5V</b>	LED ON <b>+2.2 mA</b>
<b>3.3V</b>	<b>+0.6 mA</b>

# UART mode

## command quick reference

All commands are ASCII strings or single ASCII characters.

Command	Function		Default state
Baud	change baud rate	pg. 35	9,600
C	enable/disable continuous reading	pg. 24	enabled
Cal	performs calibration	pg. 26	n/a
Export/import	export/import calibration	pg. 27	n/a
Factory	enable factory reset	pg. 37	n/a
Find	finds device with blinking white LED	pg. 23	n/a
i	device information	pg. 31	n/a
I2C	change to I <sup>2</sup> C mode	pg. 38	not set
L	enable/disable LED	pg. 22	enabled
Name	set/show name of device	pg. 30	not set
Plock	enable/disable protocol lock	pg. 36	disabled
R	returns a single reading	pg. 25	n/a
Sleep	enter sleep mode/low power	pg. 34	n/a
Slope	returns the slope of the pH probe	pg. 28	n/a
Status	retrieve status information	pg. 33	enable
T	temperature compensation	pg. 29	25°C
*OK	enable/disable response codes	pg. 32	enable

# LED control

## Command syntax

L,1 <cr> LED on **default**

L,0 <cr> LED off

L,? <cr> LED state on/off?

## Example      Response

L,1 <cr>

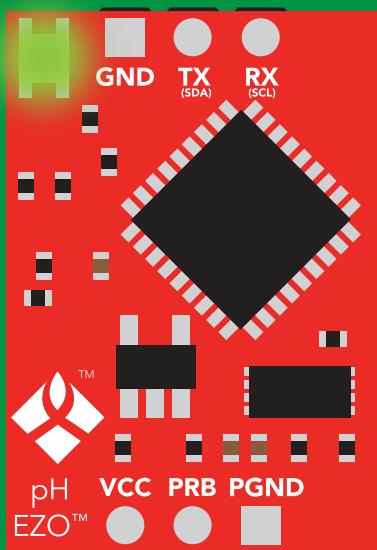
\*OK <cr>

L,0 <cr>

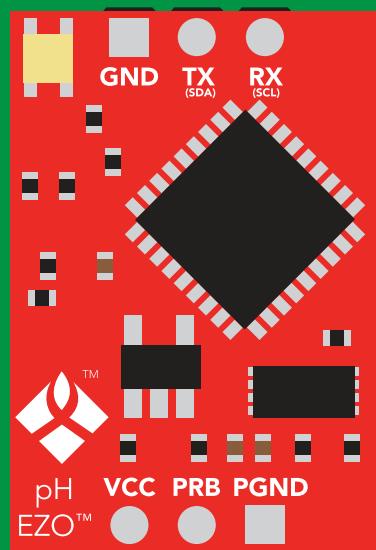
\*OK <cr>

L,? <cr>

?L,1 <cr> or ?L,0 <cr>  
\*OK <cr>



L,1



L,0

# Find

## Command syntax

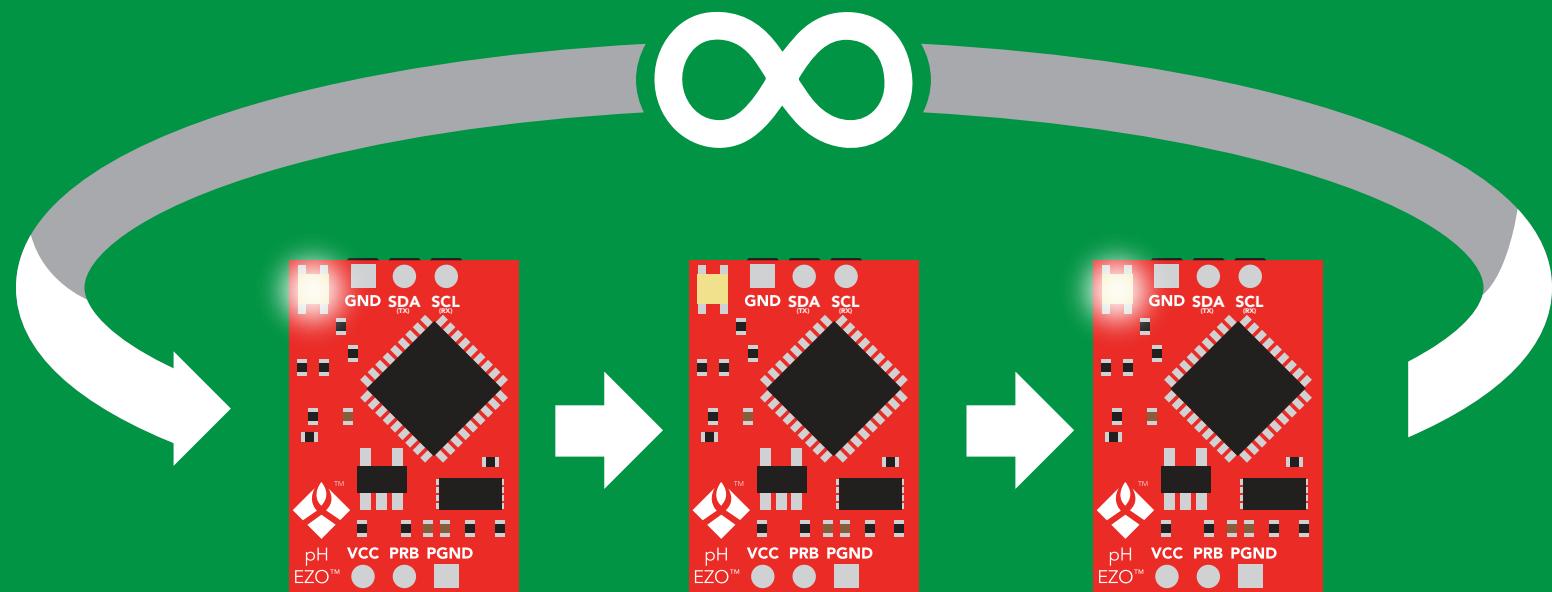
This command will disable continuous mode  
Send any character or command to terminate find.

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

## Example Response

Find <cr>

\*OK <cr>



# Continuous reading mode

## Command syntax

- C,1 <cr> enable continuous readings once per second **default**
- C,n <cr> continuous readings every n seconds (n = 2 to 99 sec)
- C,0 <cr> disable continuous readings
- C,? <cr> continuous reading mode on/off?

## Example      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

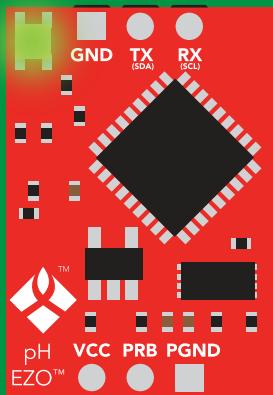
R <cr> takes single reading

### Example      Response

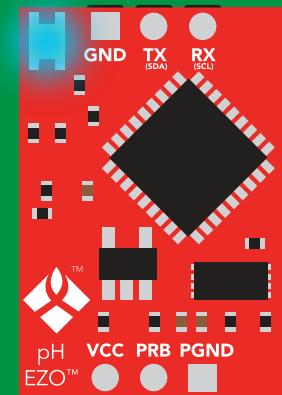
R <cr>

9.560 <cr>

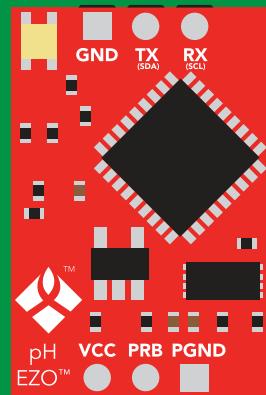
\*OK <cr>



Green  
Standby



Cyan  
Taking reading



Transmitting



# Calibration

## Command syntax

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.

<b>Cal,mid,n &lt;cr&gt;</b>	<b>single point calibration at midpoint</b>
<b>Cal,low,n &lt;cr&gt;</b>	<b>two point calibration at lowpoint</b>
<b>Cal,high,n &lt;cr&gt;</b>	<b>three point calibration at highpoint</b>
<b>Cal,clear &lt;cr&gt;</b>	<b>delete calibration data</b>
<b>Cal,? &lt;cr&gt;</b>	<b>device calibrated?</b>

## Example Response

<b>Cal,mid,7.00 &lt;cr&gt;</b>	<b>*OK &lt;cr&gt;</b>
<b>Cal,low,4.00 &lt;cr&gt;</b>	<b>*OK &lt;cr&gt;</b>
<b>Cal,high,10.00 &lt;cr&gt;</b>	<b>*OK &lt;cr&gt;</b>
<b>Cal,clear &lt;cr&gt;</b>	<b>*OK &lt;cr&gt;</b>
<b>Cal,? &lt;cr&gt;</b>	<b>?Cal,0 &lt;cr&gt; or ?Cal,1 &lt;cr&gt; or one point ?Cal,2 &lt;cr&gt; or ?Cal,3 &lt;cr&gt; two point three point *OK &lt;cr&gt;</b>

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

## Example

Export,? <cr>

## Response

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

Import, n  
(FIFO)

Import, 59 6F 75 20 61 72 <cr> (1 of 10)

# Slope

## Command syntax

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.

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

### Example Response

**Slope,? <cr>**

**?Slope,99.7,100.3 <cr>  
\*OK <cr>**

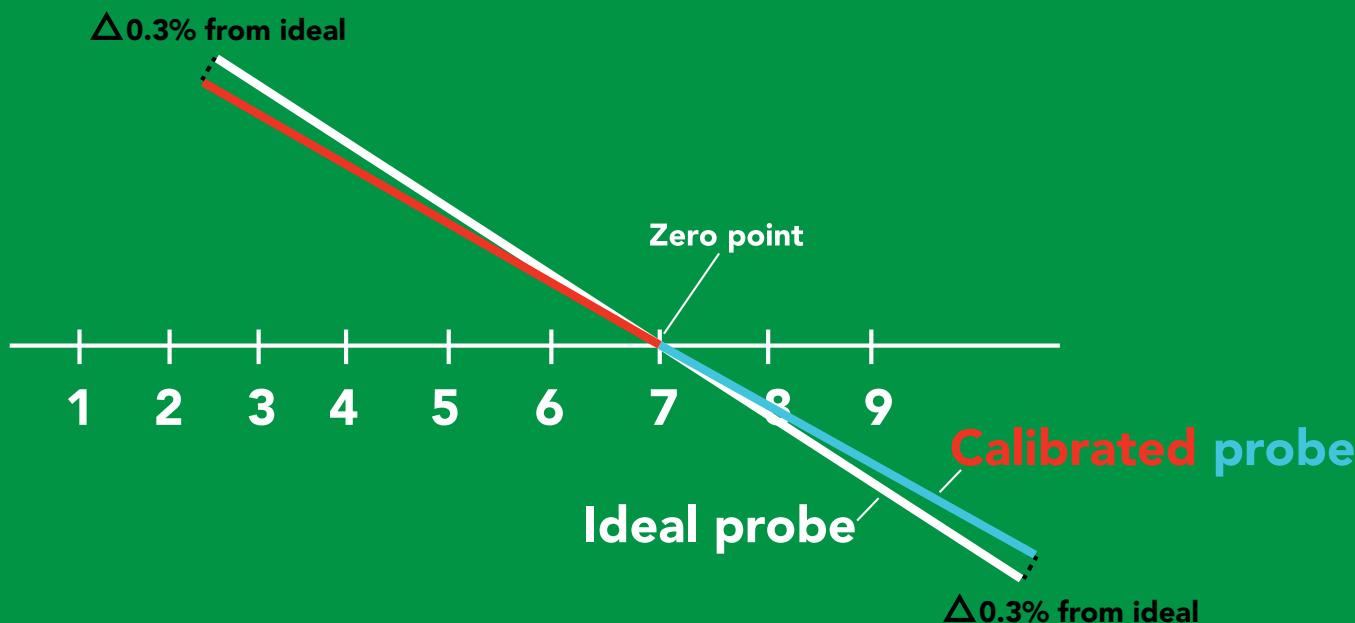
### Response breakdown

**?Slope,** **99.7,**

↑  
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 line matched the "ideal" pH probe.



# Temperature compensation

## Command syntax

Temperature is always in Celsius

T,n <cr> n = any value; floating point or int

T,? <cr> compensated temperature value?

## Example

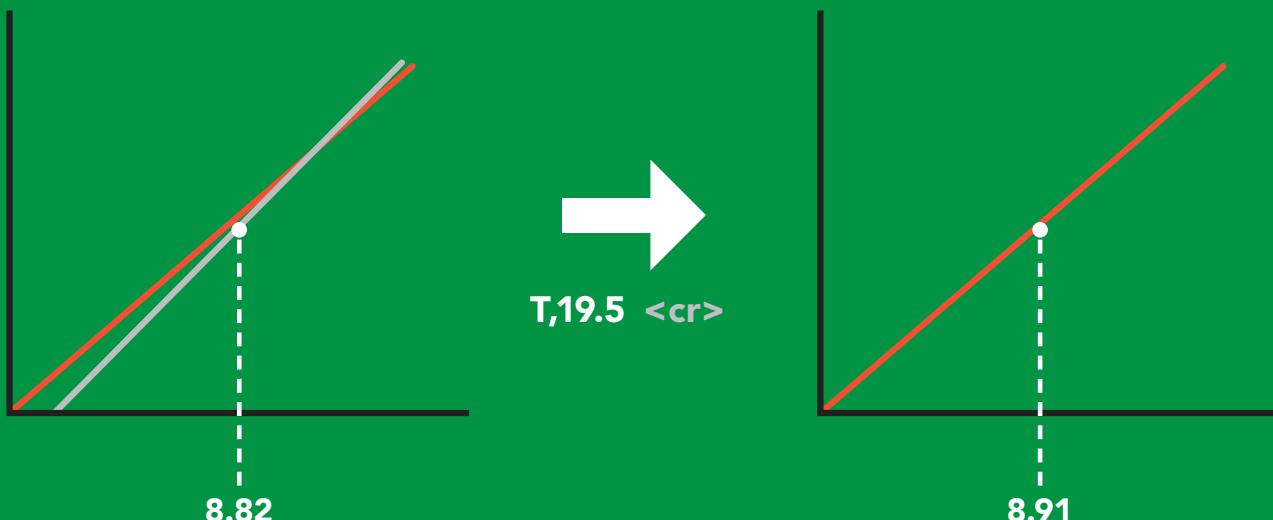
T,19.5 <cr>

## Response

\*OK <cr>

T,? <cr>

?T,19.5 <cr>  
\*OK <cr>



# Naming device

## Command syntax

Name,n <cr> set name

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

Name,? <cr> show name

Up to 16 ASCII characters

## Example

Name,zzt <cr>

\*OK <cr>

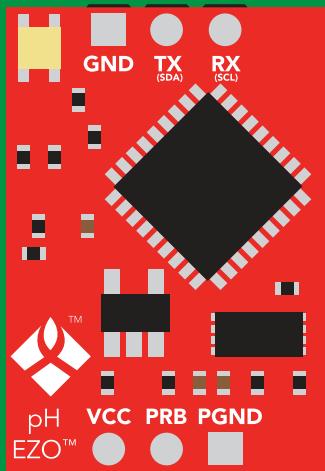
Name,? <cr>

?Name,zzt <cr>

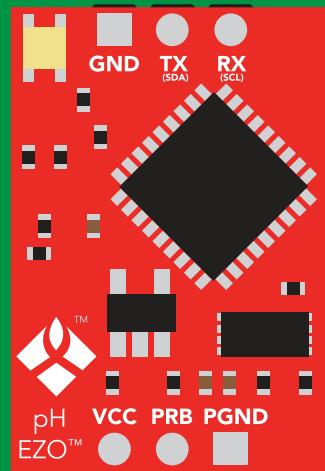
\*OK <cr>

## Response

Name,zzt



Name,?



\*OK <cr>

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

# Device information

## Command syntax

i <cr> device information

### Example      Response

i <cr>

?i,pH,1.98 <cr>

\*OK <cr>

### Response breakdown

?i, pH, 1.98  
↑      ↑  
Device    Firmware

# Response codes

## Command syntax

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

Example	Response
R <cr>	9.560 <cr> *OK <cr>
*OK,0 <cr>	<b>no response, *OK disabled</b>
R <cr>	9.560 <cr> *OK disabled
*OK,? <cr>	?*OK,1 <cr> or ?*OK,0 <cr>

## 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      Response

Status <cr>

?Status,P,5.038 <cr>

\*OK <cr>

### Response breakdown

?Status, P,  
↑  
Reason for restart      5.038  
                         ↑  
                         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.

**Sleep <cr> enter sleep mode/low power**

### Example

Sleep <cr>

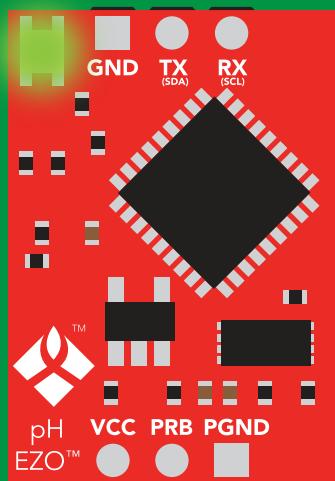
### Response

\*SL

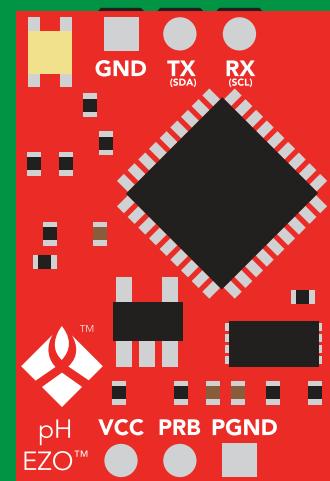
### Any command

\*WA <cr> wakes up device

	STANDBY	SLEEP
5V	16 mA	1.16 mA
3.3V	13.9 mA	0.995 mA



**Sleep <cr>**



# Change baud rate

## Command syntax

Baud,n <cr> change baud rate

### Example

Baud,38400 <cr>

### Response

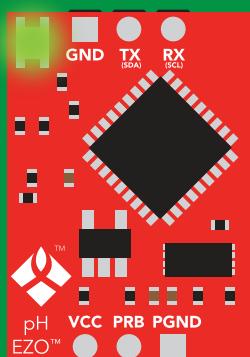
\*OK <cr>

Baud,? <cr>

?Baud,38400 <cr>

\*OK <cr>

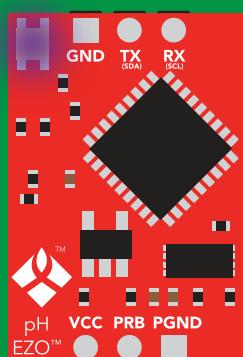
n = [ 300  
1200  
2400  
**9600 default**  
19200  
38400  
57600  
115200 ]



Standby



Baud,38400 <cr>

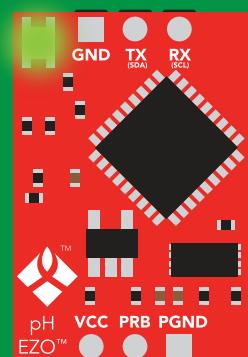


Changing  
baud rate

\*OK <cr>



(reboot)



Standby

# Protocol lock

## Command syntax

Locks device to UART mode.

Plock,1 <cr> enable Plock

Plock,0 <cr> disable Plock **default**

Plock,? <cr> Plock on/off?

## Example

Plock,1 <cr>

\*OK <cr>

Plock,0 <cr>

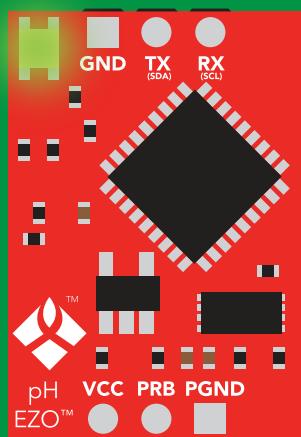
\*OK <cr>

Plock,? <cr>

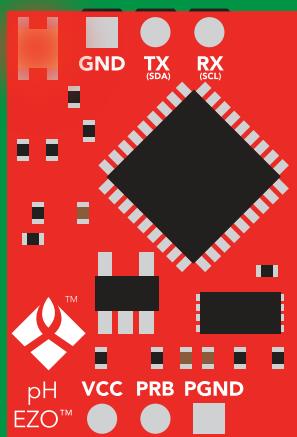
?Plock,1 <cr> or ?Plock,0 <cr>

## Response

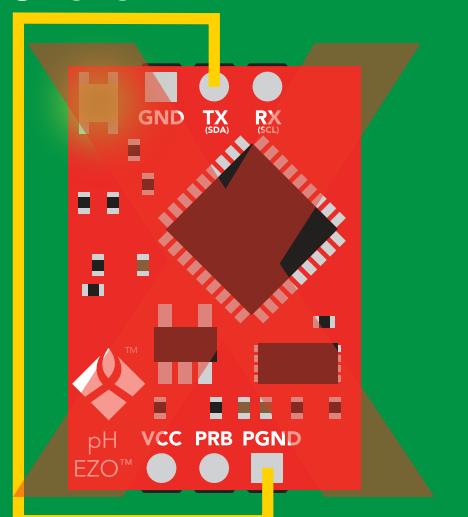
Plock,1



I2C,100



Short



\*OK <cr>

cannot change to I<sup>2</sup>C

\*ER <cr>

cannot change to I<sup>2</sup>C

# Factory reset

## Command syntax

Clears calibration  
LED on  
"\*OK" enabled

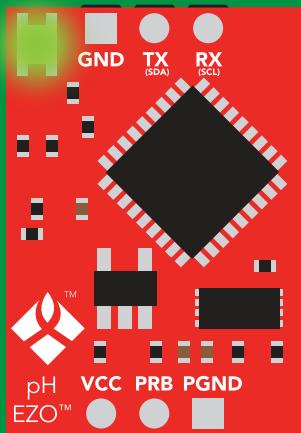
Factory <cr> enable factory reset

## Example Response

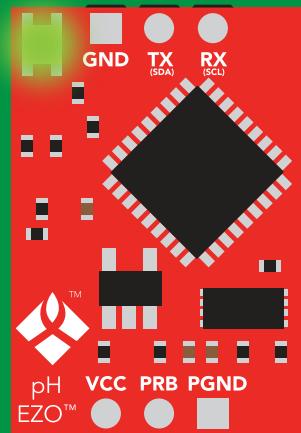
Factory <cr>

\*OK <cr>

Factory <cr>



(reboot)



\*OK <cr>

\*RS <cr>

\*RE <cr>

Baud rate will not change

# Change to I<sup>2</sup>C mode

## Command syntax

Default I<sup>2</sup>C address 99 (0x63)

I<sup>2</sup>C,n <cr> sets I<sup>2</sup>C address and reboots into I<sup>2</sup>C mode

n = any number 1 – 127

## Example Response

I<sup>2</sup>C,100 <cr>

\*OK (reboot in I<sup>2</sup>C mode)

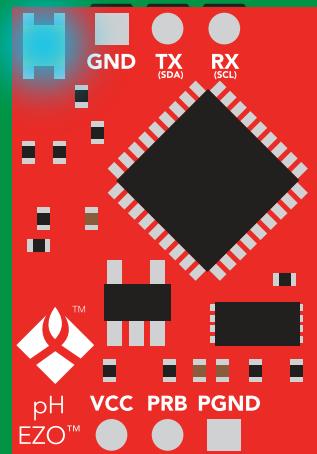
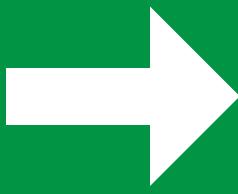
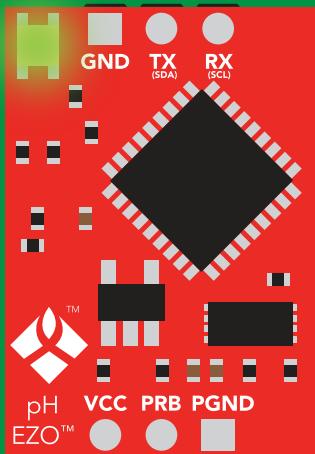
## Wrong example

I<sup>2</sup>C,139 <cr> n ≠ 127

## Response

\*ER <cr>

I<sup>2</sup>C,100



Green  
\*OK <cr>

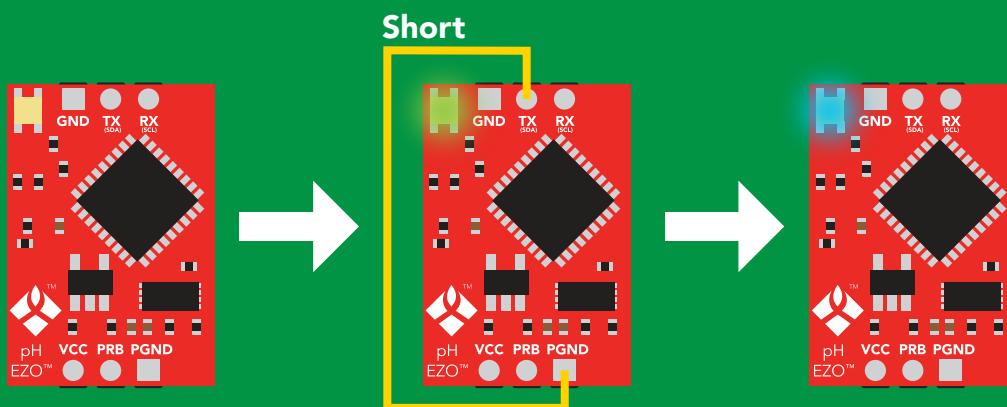
Blue  
now in I<sup>2</sup>C mode

# Manual switching to I<sup>2</sup>C

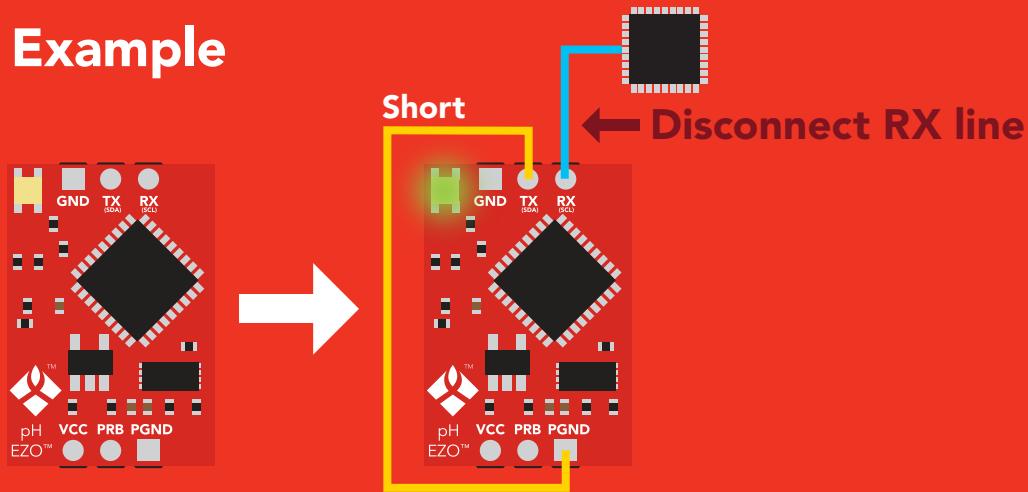
- Make sure Plock is set to 0
- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to PGND
- Confirm RX is disconnected
- Connect ground (power on)
- Wait for LED to change from **Green** to **Blue**
- Disconnect ground (power off)
- Reconnect all data and power

Manually switching to I<sup>2</sup>C will set the I<sup>2</sup>C address to 99 (0x63)

## Example



## Wrong Example



# I<sup>2</sup>C mode

The I<sup>2</sup>C 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 I<sup>2</sup>C mode [click here](#)

## Settings that are retained if power is cut

Calibration  
Change I<sup>2</sup>C address  
Hardware switch to UART mode  
LED control  
Protocol lock  
Software switch to UART mode

## Settings that are **NOT** retained if power is cut

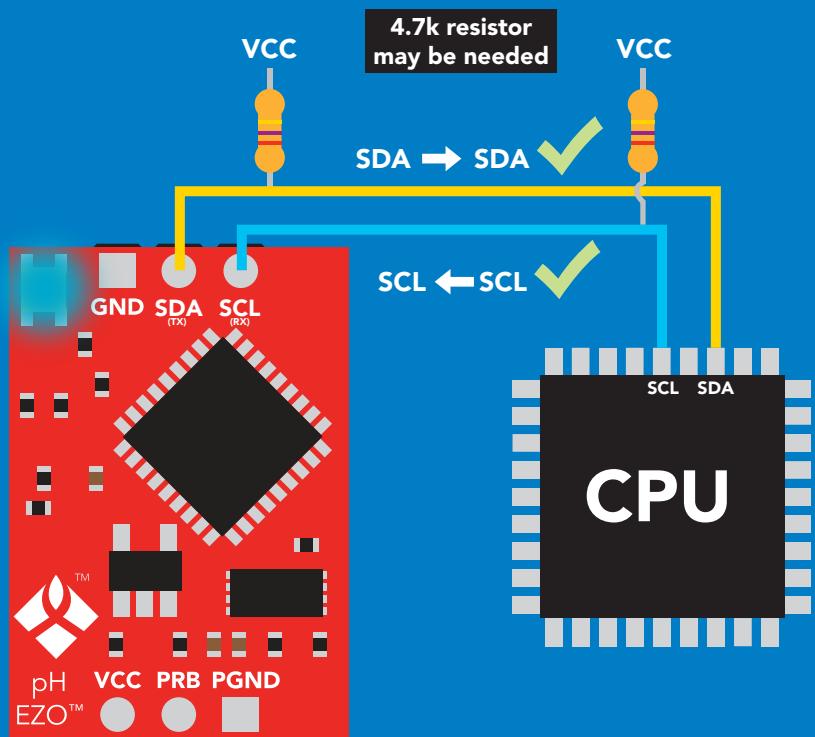
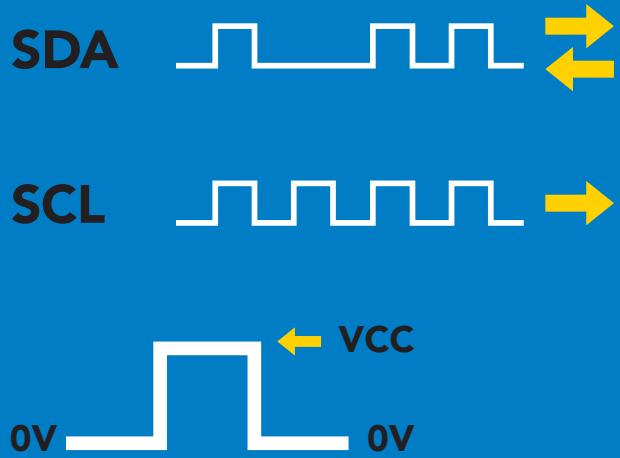
Find  
Sleep mode

# I<sup>2</sup>C mode

I<sup>2</sup>C address (0x01 – 0x7F)  
**99 (0x63) default**

V<sub>cc</sub> 3.3V – 5.5V

Clock speed 100 – 400 kHz



## Data format

Reading pH

Units pH

Encoding ASCII

Format string

Data type floating point

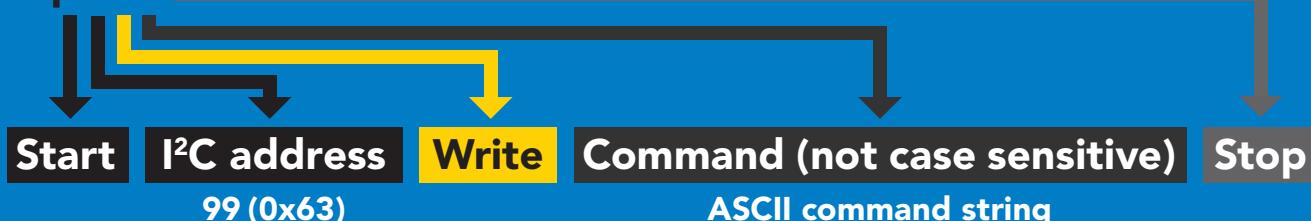
Decimal places 3

Smallest string 4 characters

Largest string 399 characters

# Sending commands to device

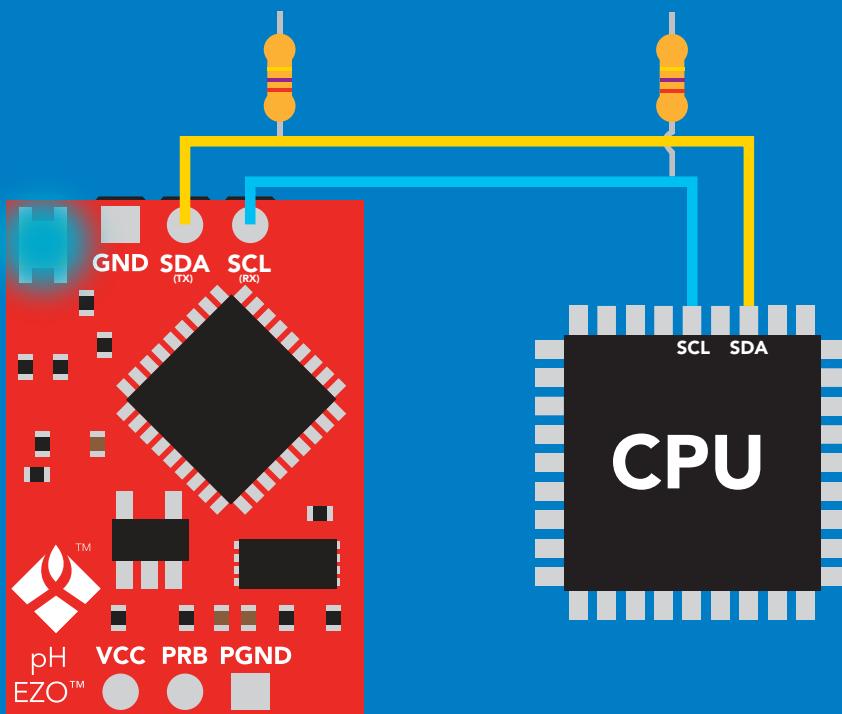
5 parts



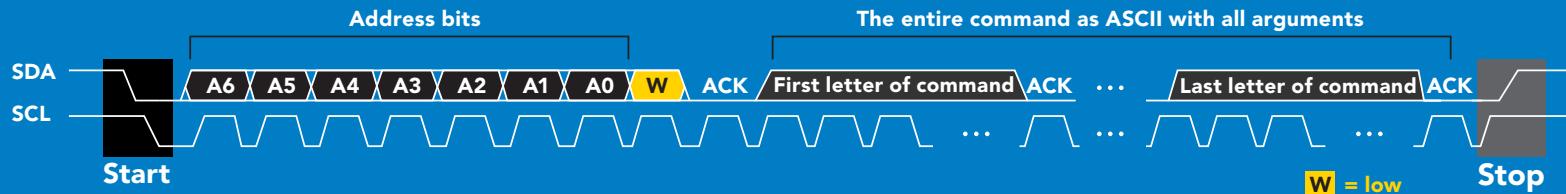
## Example

Start    99 (0x63)    Write    Sleep    Stop

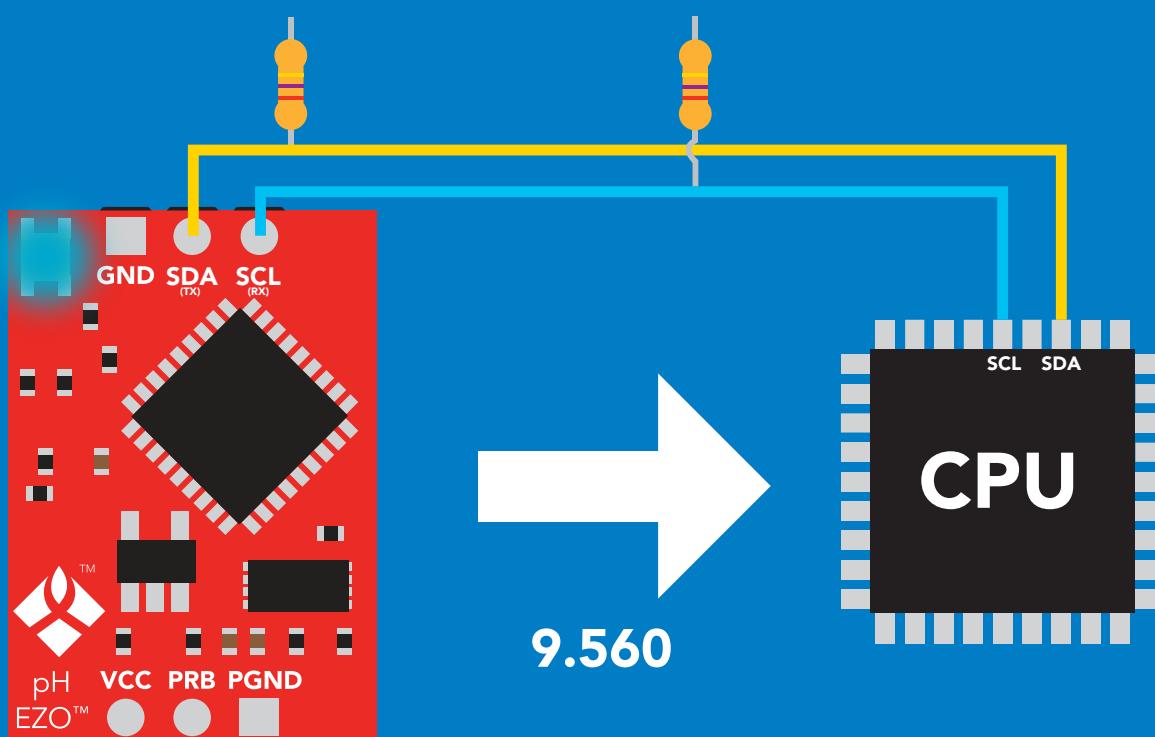
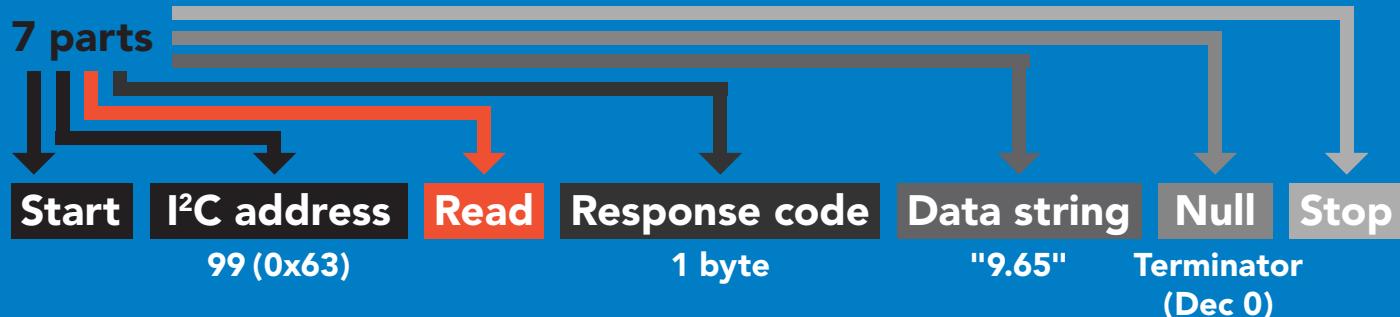
I<sup>2</sup>C address              Command



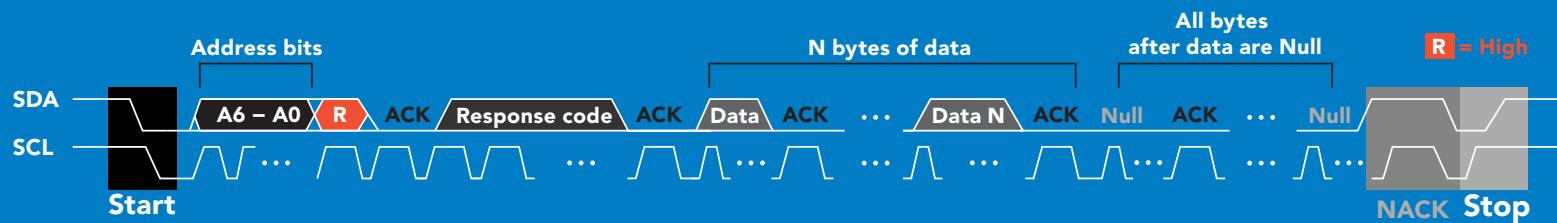
## Advanced



# Requesting data from device



# Advanced

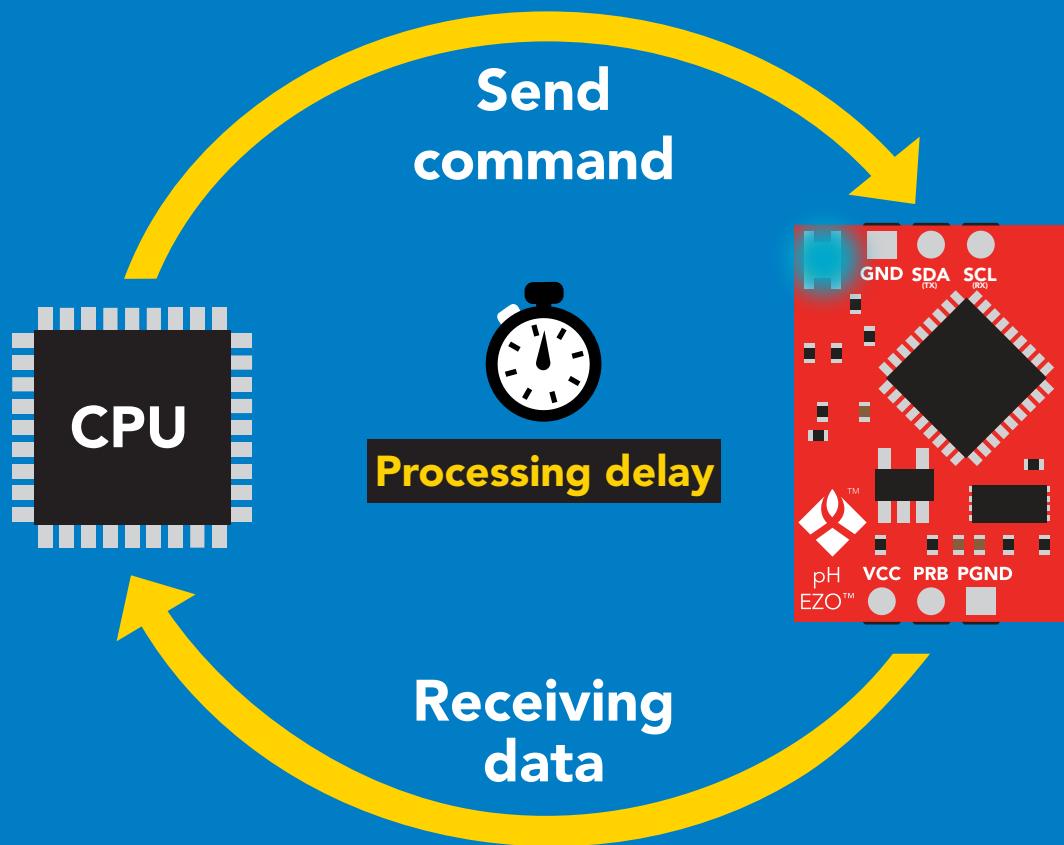


**1 57 46 53 54 48 0 = 9.560**

# 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

```
I2C_start;  
I2C_address;  
I2C_write(EZO_command);  
I2C_stop;
```

```
delay(300); →  Processing delay
```

```
I2C_start;  
I2C_address;  
Char[ ] = I2C_read;  
I2C_stop;
```

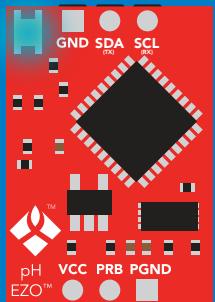
If there is no processing delay or the processing delay is too short, the response code will always be 254.

### Response codes

Single byte, not string

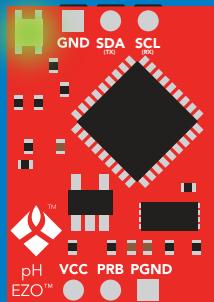
- |     |                             |
|-----|-----------------------------|
| 255 | no data to send             |
| 254 | still processing, not ready |
| 2   | syntax error                |
| 1   | successful request          |

# LED color definition



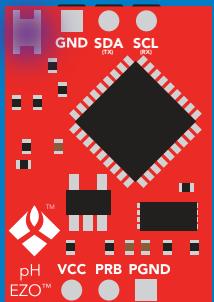
**Blue**

I<sup>2</sup>C standby



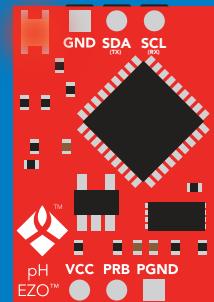
**Green**

Taking reading



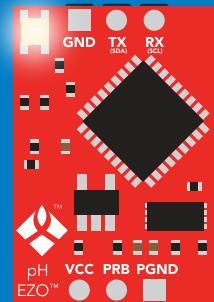
**Purple**

Changing I<sup>2</sup>C ID#



**Red**

Command not understood



**White**

Find

5V	LED ON <b>+2.2 mA</b>
3.3V	<b>+0.6 mA</b>

# I<sup>2</sup>C mode

## command quick reference

All commands are ASCII strings or single ASCII characters.

Command	Function	
Baud	switch back to UART mode	pg. 60
Cal	performs calibration	pg. 50
Export/import	export/import calibration	pg. 51
Factory	enable factory reset	pg. 59
Find	finds device with blinking white LED	pg. 48
i	device information	pg. 54
I2C	change I <sup>2</sup> C address	pg. 58
L	enable/disable LED	pg. 47
Plock	enable/disable protocol lock	pg. 57
R	returns a single reading	pg. 49
Sleep	enter sleep mode/low power	pg. 56
Slope	returns the slope of the pH probe	pg. 52
Status	retrieve status information	pg. 55
T	temperature compensation	pg. 53

# LED control

## Command syntax

300ms  processing delay

L,1 LED on **default**

L,0 LED off

L,? LED state on/off?

## Example

L,1

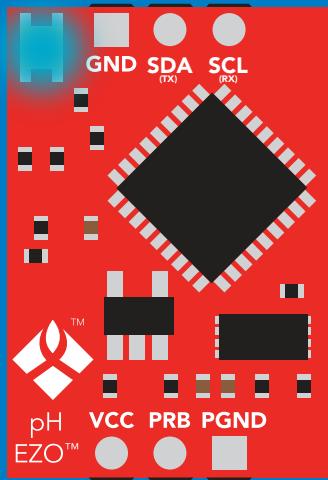
 Wait 300ms  
1 Dec 0 Null

L,0

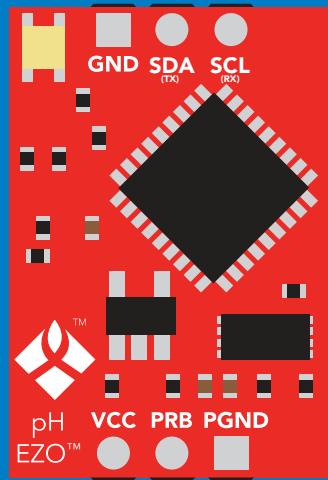
 Wait 300ms  
1 Dec 0 Null

L,?

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



L,1



L,0

# Find

300ms  processing delay

## Command syntax

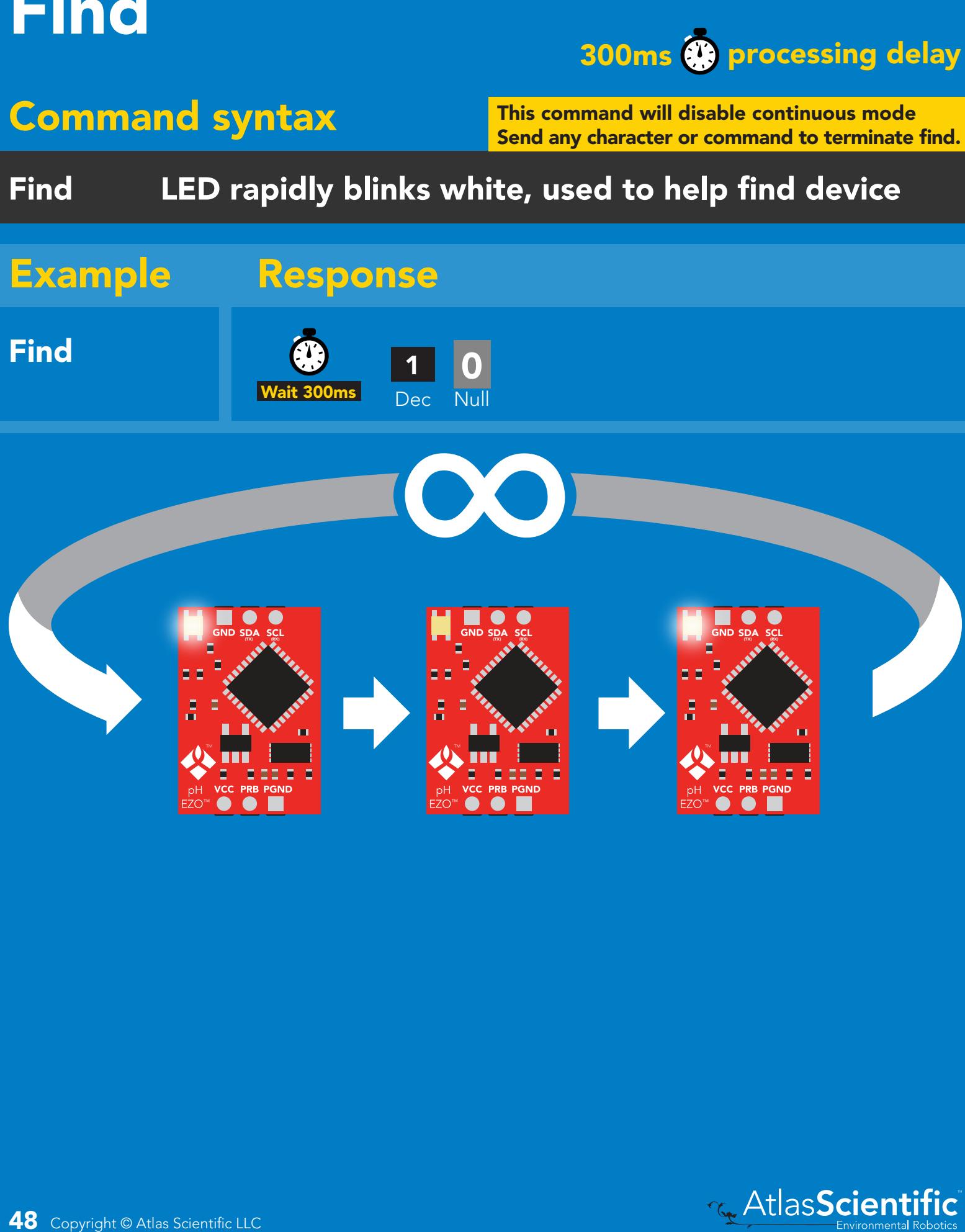
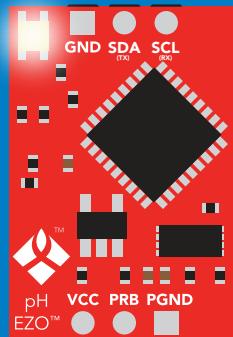
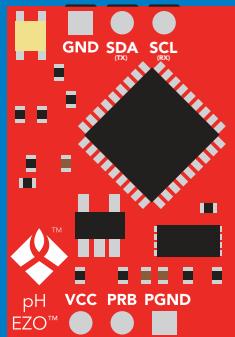
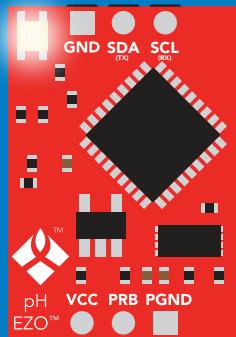
This command will disable continuous mode  
Send any character or command to terminate find.

Find      LED rapidly blinks white, used to help find device

## Example Response

Find

 Wait 300ms  
**1** Dec **0** Null



# Taking reading

Command syntax

900ms  processing delay

R return 1 reading

Example

Response

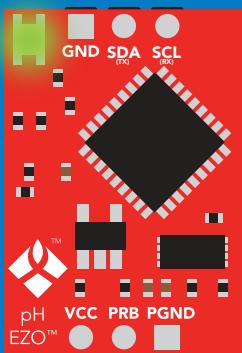
R



1  
Dec

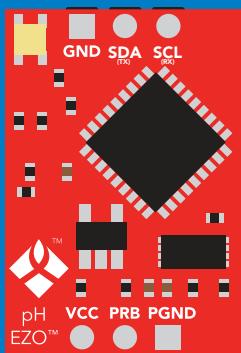
9.560  
ASCII

0  
Null

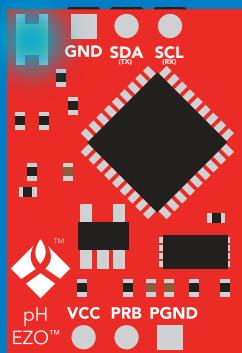


Green

Taking reading



Transmitting



Blue

Standby

# Calibration

## Command syntax

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.

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

## Example

## Response

Cal,mid,7.00

 Wait 900ms  
1 Dec 0 Null

Cal,low,4.00

 Wait 900ms  
1 Dec 0 Null

Cal,high,10.00

 Wait 900ms  
1 Dec 0 Null

Cal,clear

 Wait 300ms  
1 Dec 0 Null

Cal,?

 Wait 300ms  
1 Dec ?Cal,0 0 Null or 1 Dec ?Cal,1 0 Null  
or 1 Dec ?Cal,2 0 Null or 1 Dec ?Cal,3 0 Null  
ASCII one point  
ASCII two point  
ASCII three point

# 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,?

calibration string info

300ms  processing delay

## Example

Export,?

## Response



1 Dec 10,120 ASCII 0 Null

### Response breakdown

10, 120  
↑ ↑  
# of strings to export # of bytes to export

Export strings can be up to 12 characters long

Export

(8 more)

Export

Export

Import, n  
(FIFO)



1 Dec 59 6F 75 20 61 72 0 Null

(1 of 10)

⋮



1 Dec 65 20 61 20 63 6F 0 Null

(10 of 10)



1 Dec \*DONE 0 Null

# Slope

## Command syntax

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.

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

## Example Response

**Slope,?**



Wait 300ms

**1**

?Slope,99.7,100.3

Dec

ASCII

**0**

Null

## Response breakdown

?Slope,

**99.7,**

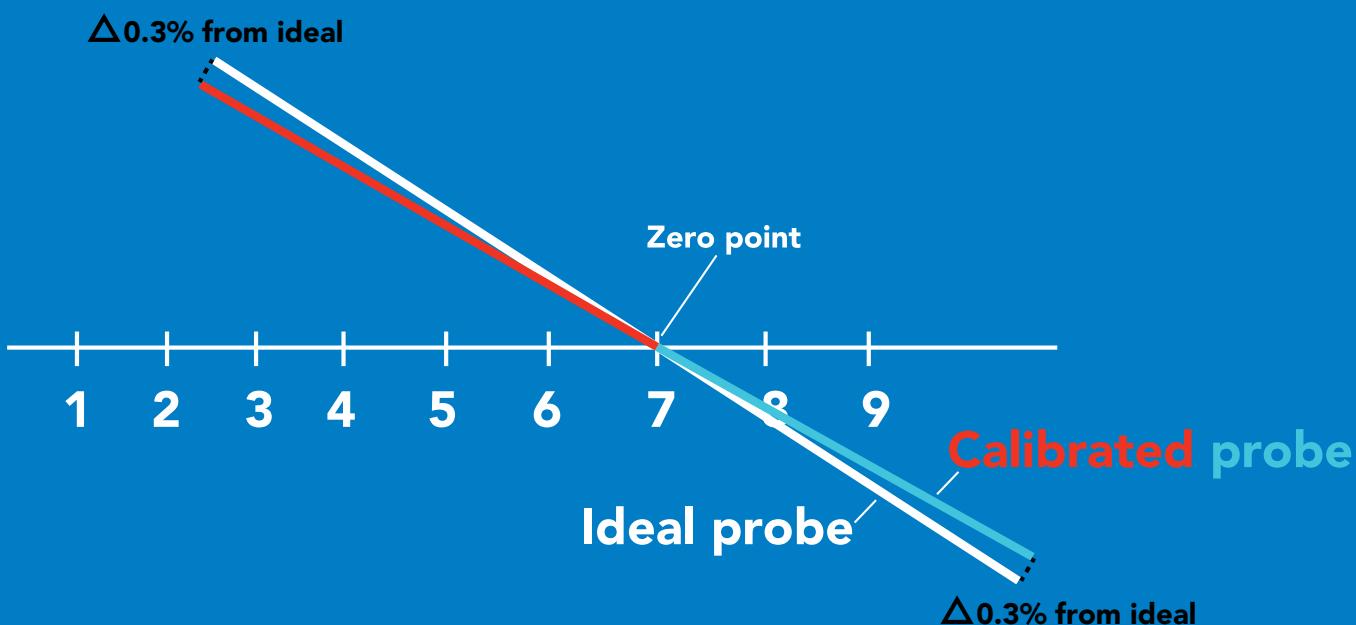


99.7% is how closely the slope of the **acid** calibration line matched the "ideal" pH probe.

**100.3**



100.3% is how closely the slope of the **base** calibration line matches the "ideal" pH probe.



# Temperature compensation

300ms  processing delay

## Command syntax

Temperature is always in Celsius

T,n    n = any value; floating point or int

T,?    compensated temperature value?

## Example

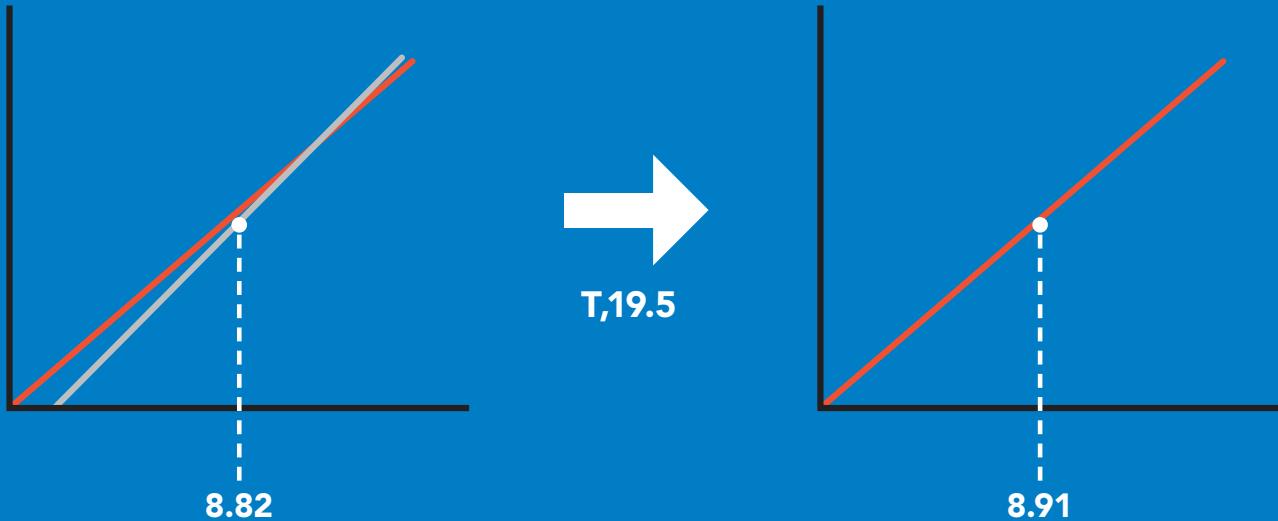
T,19.5

## Response

 Wait 300ms  
1 Dec 0 Null

T,?

 Wait 300ms  
1 Dec ?T,19.5 0 ASCII Null



# Device information

Command syntax

300ms  processing delay

i device information

Example Response

i



Wait 300ms

1  
Dec

?i,pH, 19.8  
ASCII

0  
Null

Response breakdown

?i, pH, 1.98  
Device Firmware

# Reading device status

## Command syntax

300ms  processing delay

Status voltage at Vcc pin and reason for last restart

## Example Response

Status



Wait 300ms

1

?Status,P,5.038

Dec

ASCII

0

Null

## 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

**Sleep**

**no response**

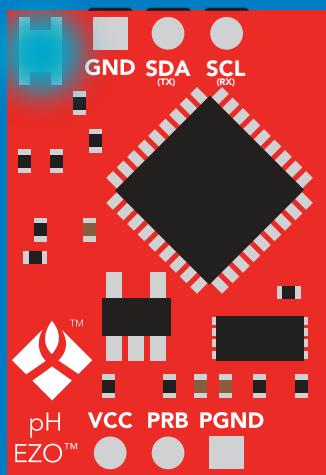
Do not read status byte after issuing sleep command.

**Any command**

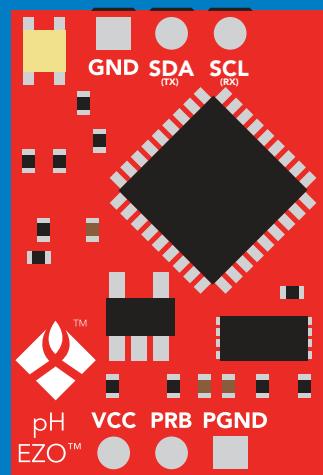
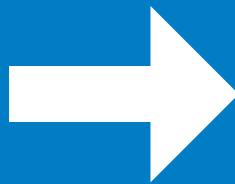
**wakes up device**

	STANDBY	SLEEP
<b>5V</b>	<b>16 mA</b>	<b>1.16 mA</b>

<b>3.3V</b>	<b>13.9 mA</b>	<b>0.995 mA</b>
-------------	----------------	-----------------



Standby



Sleep

# Protocol lock

## Command syntax

300ms  processing delay

Plock,1 enable Plock

Locks device to I<sup>2</sup>C mode.

Plock,0 disable Plock **default**

Plock,? Plock on/off?

## Example

Plock,1

 Wait 300ms  
1 Dec 0 Null

Plock,0

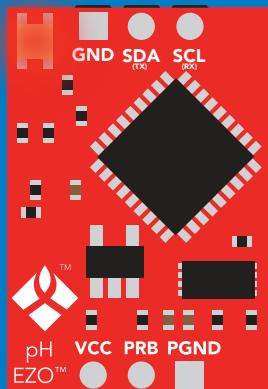
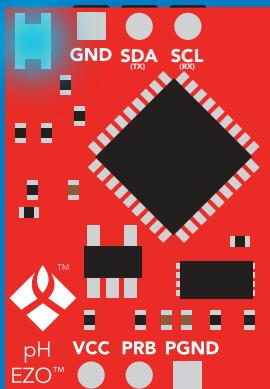
 Wait 300ms  
1 Dec 0 Null

Plock,?

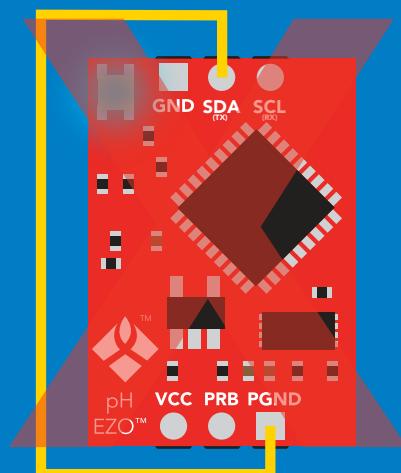
 Wait 300ms  
1 Dec ?Plock,1 ASCII 0 Null

Plock,1

Baud, 9600



cannot change to UART



cannot change to UART

# I<sup>2</sup>C address change

## Command syntax

300ms  processing delay

I<sup>2</sup>C,n sets I<sup>2</sup>C address and reboots into I<sup>2</sup>C mode

## Example Response

I<sup>2</sup>C,100

device reboot

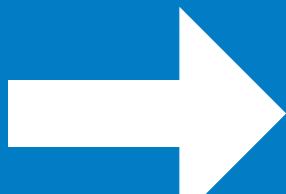
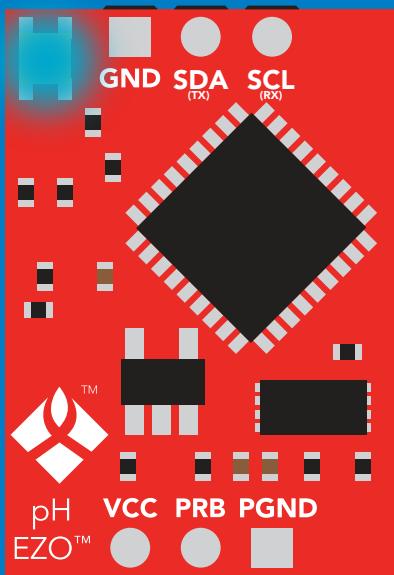
### Warning!

Changing the I<sup>2</sup>C address will prevent communication between the circuit and the CPU, until the CPU is updated with the new I<sup>2</sup>C address.

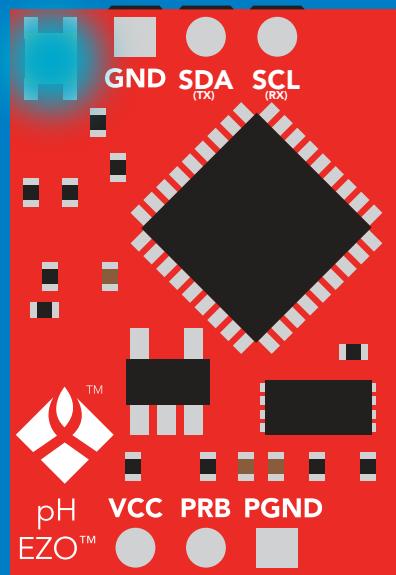
Default I<sup>2</sup>C address is 99 (0x63).

n = any number 1 – 127

I<sup>2</sup>C,100



(reboot)



# Factory reset

## Command syntax

Factory reset will not take the device out of I<sup>2</sup>C mode.

Factory enable factory reset

I<sup>2</sup>C address will not change

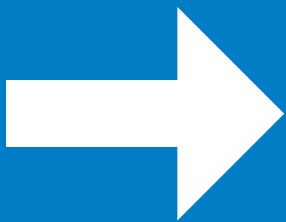
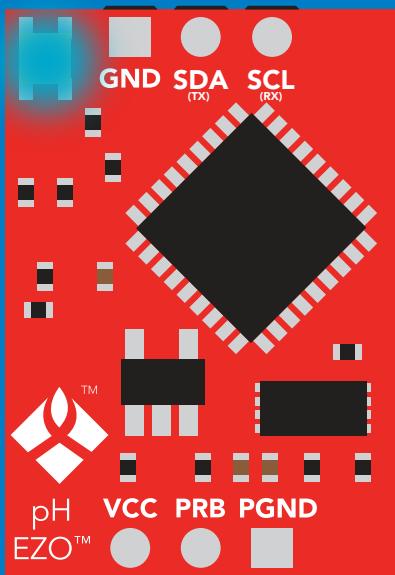
## Example Response

Factory

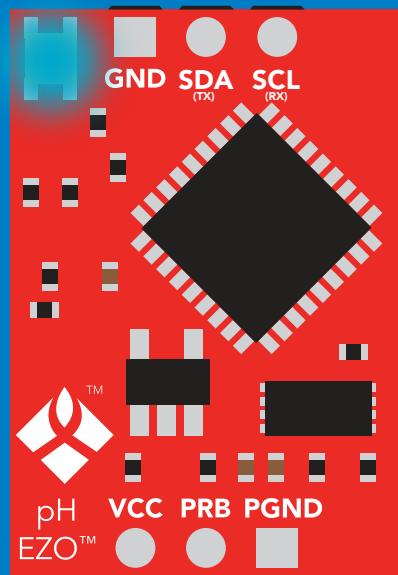
device reboot

Clears calibration  
LED on  
Response codes enabled

Factory



(reboot)



# Change to UART mode

## Command syntax

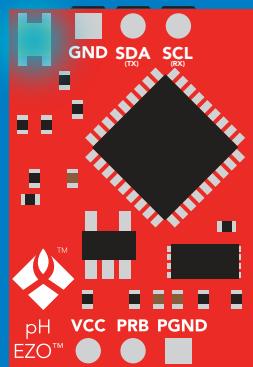
Baud,n switch from I<sup>2</sup>C to UART

## Example Response

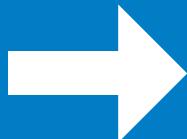
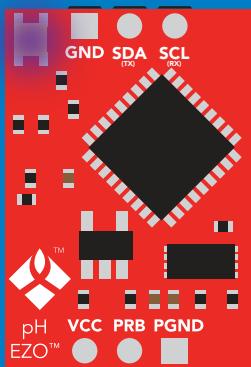
Baud,9600

reboot in UART mode

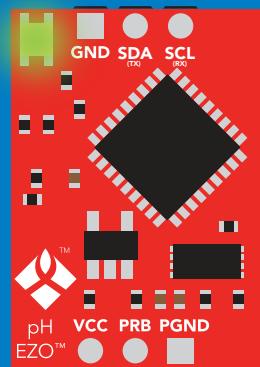
n = [ 300  
1200  
2400  
9600  
19200  
38400  
57600  
115200 ]



Baud,9600



(reboot)

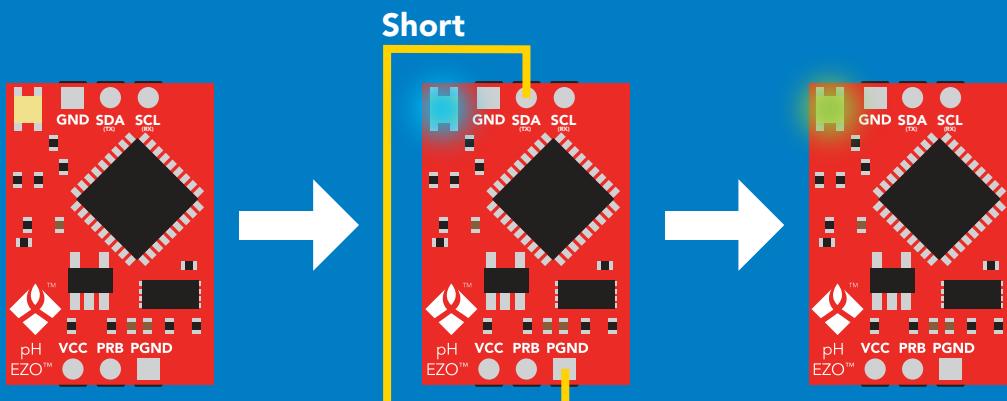


Changing to UART  
mode

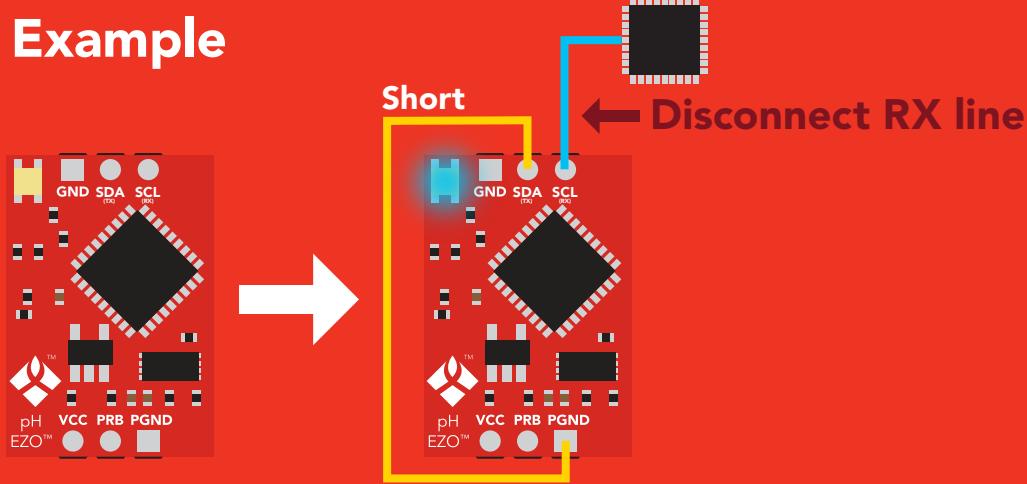
# Manual switching to UART

- 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

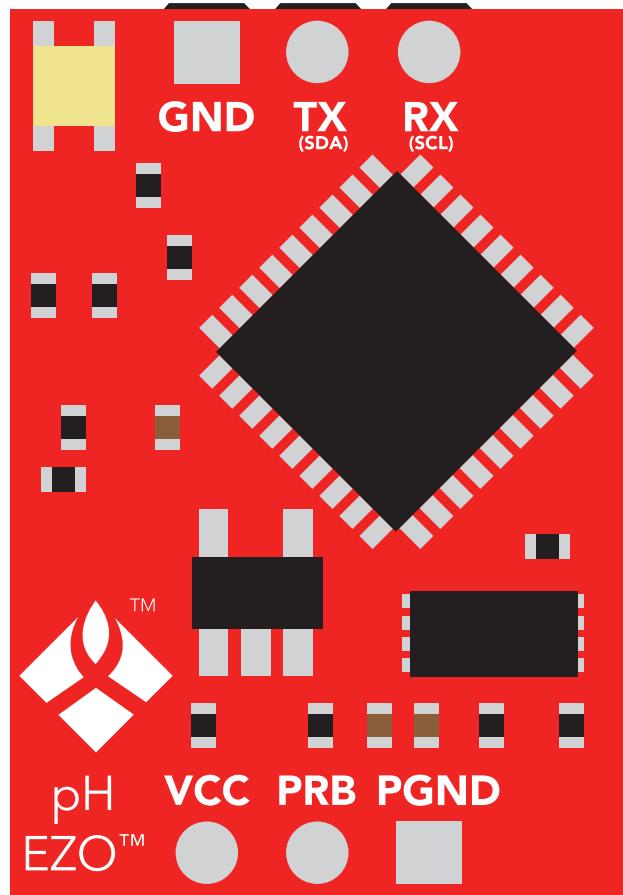
## Example



## Wrong Example



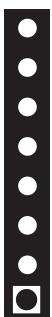
# EZO™ circuit footprint



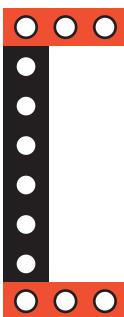
17.78mm  
(0.7")

2.54mm  
(0.1")

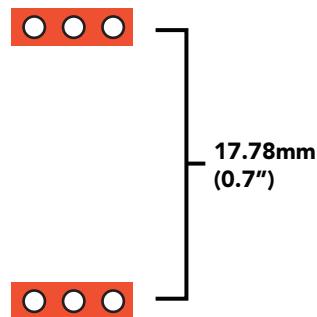
**1** In your CAD software place an 8 position header.



**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 4.7

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 & 46.

"Export/Import calibration" pages 27 & 49.

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

V1.1 – Initial release (Oct 30, 2014)

- Change output to mg/L, then percentage (was previously percentage, then mg/L)

V1.5 – Baud rate change (Nov 6, 2014)

- Change default baud rate to 9600

V1.6 – I<sup>2</sup>C bug (Dec 1, 2014)

- Fix I<sup>2</sup>C bug where the circuit may inappropriately respond when other I<sup>2</sup>C 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".

# 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.