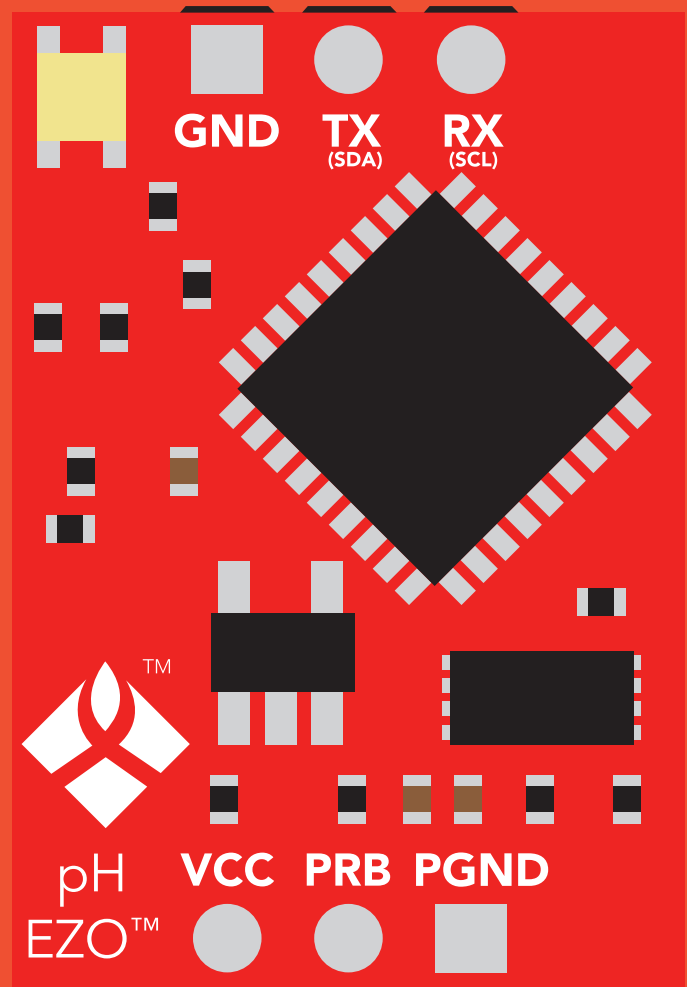


EZO-pHTM

Embedded pH Circuit

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



PATENT PROTECTED



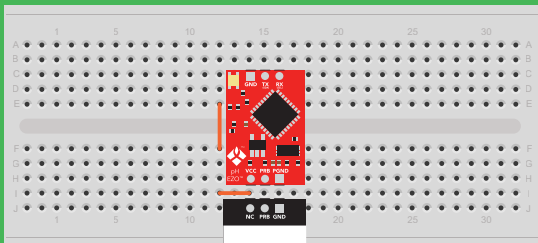
STOP

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!

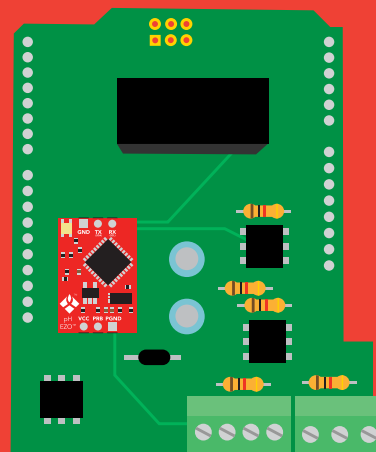


Table of contents

Circuit dimensions	4	Power and data isolation	6
Power consumption	4	Correct wiring	8
Absolute max ratings	4	Calibration theory	11
Operating principle	5	Default state	14
		Available data protocols	15

UART

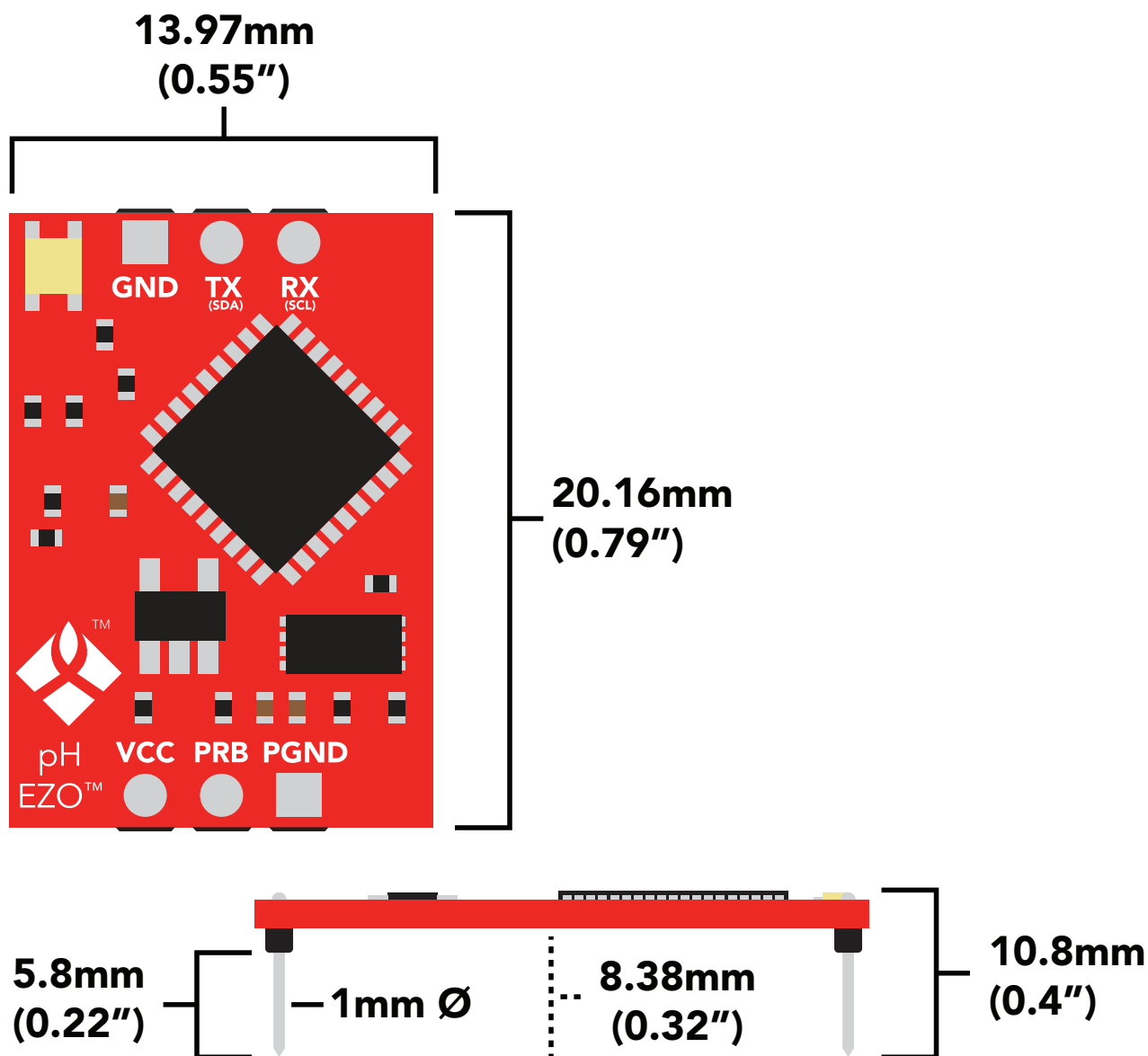
UART mode	17
Receiving data from device	18
Sending commands to device	19
LED color definition	20
UART quick command page	21
LED control	22
Find	23
Continuous reading mode	24
Single reading mode	25
Calibration	26
Export calibration	27
Import calibration	28
Slope	29
Temperature compensation	30
Naming device	31
Device information	32
Response codes	33
Reading device status	34
Sleep mode/low power	35
Change baud rate	36
Protocol lock	37
Factory reset	38
Change to I ² C mode	39
Manual switching to I ² C	40

I²C

I ² C mode	42
Sending commands	43
Requesting data	44
Response codes	45
LED color definition	46
I²C quick command page	47
LED control	48
Find	49
Taking reading	50
Calibration	51
Export calibration	52
Import calibration	53
Slope	54
Temperature compensation	55
Device information	56
Reading device status	57
Sleep mode/low power	58
Protocol lock	59
I ² C address change	60
Factory reset	61
Change to UART mode	62
Manual switching to UART	63

Circuit footprint	64
Datasheet change log	65
Warranty	68

EZO™ circuit dimensions



Power consumption

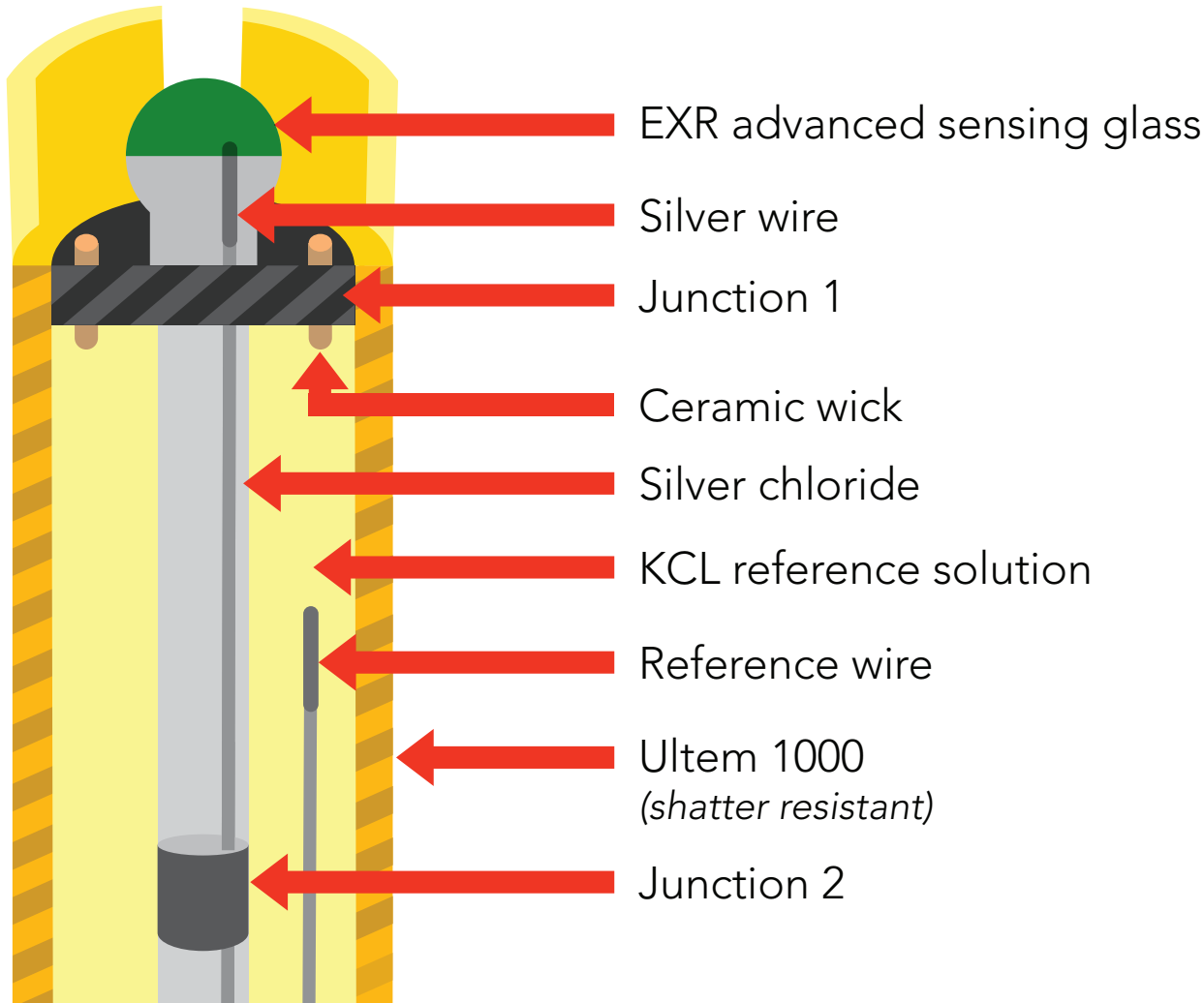
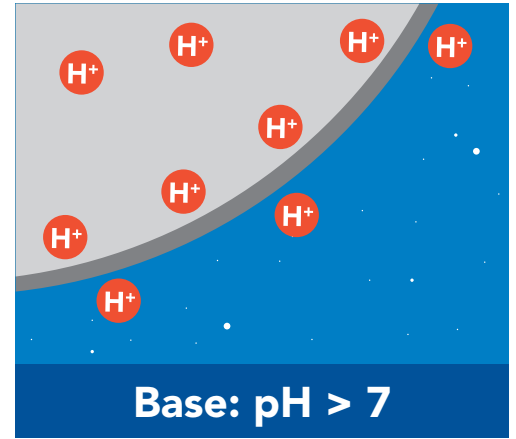
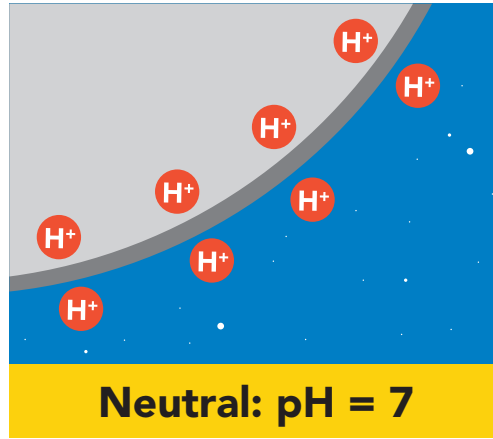
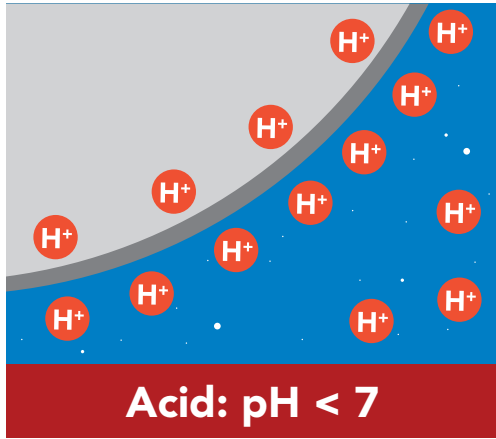
	LED	MAX	STANDBY	SLEEP
5V	ON	18.3 mA	16 mA	1.16 mA
	OFF	13.8 mA	13.8 mA	
3.3V	ON	14.5 mA	13.9 mA	0.995 mA
	OFF	13.3 mA	13.3 mA	

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

Operating principle

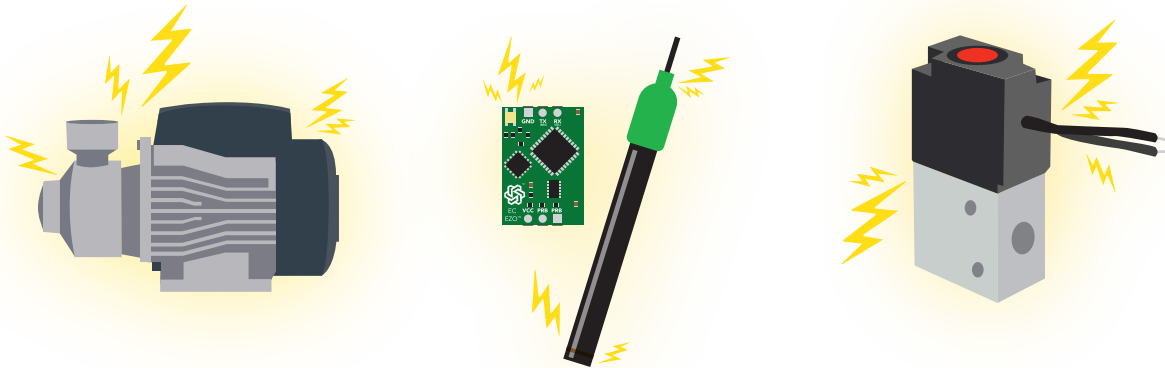
A pH (**potential of Hydrogen**) probe measures the hydrogen ion activity in a liquid. At the tip of a pH probe is a glass membrane. This glass membrane permits hydrogen ions from the liquid being measured to diffuse into the outer layer of the glass, while larger ions remain in the solution. The difference 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.



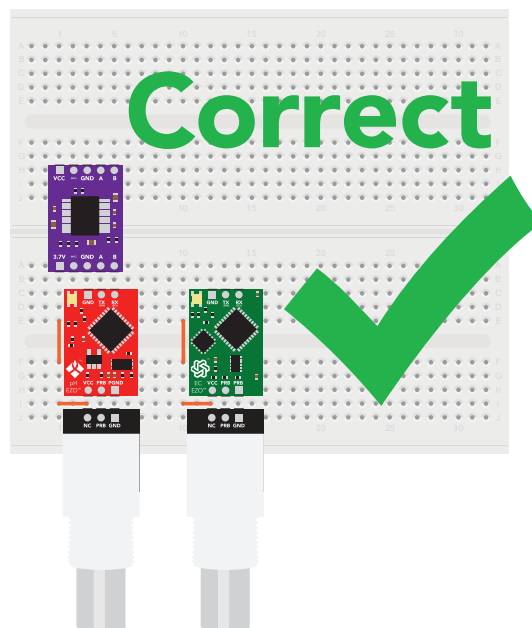
Power and data isolation

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.

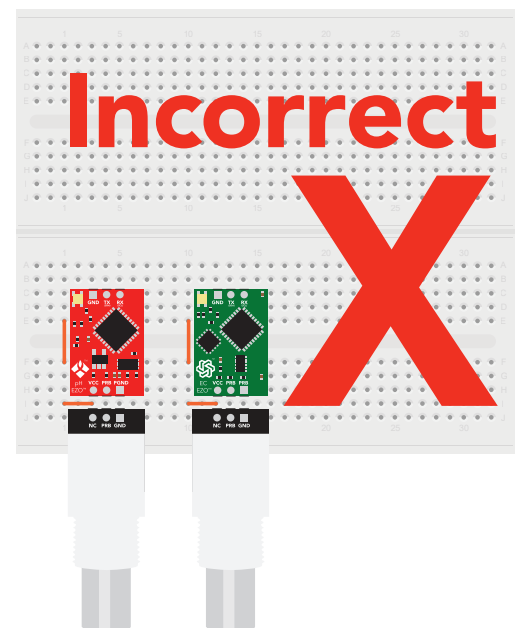
When electrical noise is interfering with the pH readings it is common to see rapidly fluctuating readings or readings that are consistently off. To verify that electrical noise is causing inaccurate readings, place the pH probe in a cup of water by itself. The readings should stabilize quickly, confirming 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



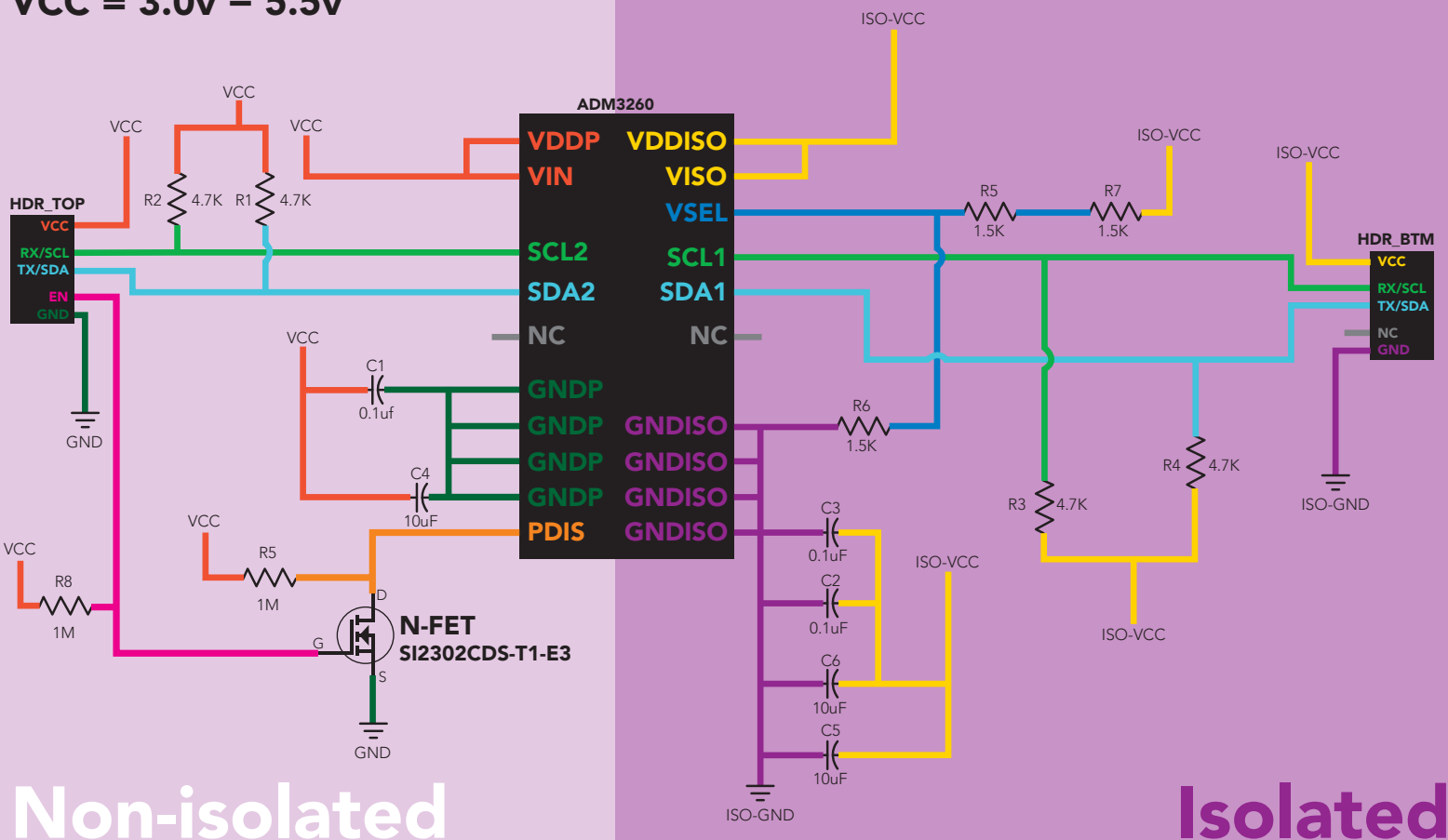
Without isolation, Conductivity and Dissolved Oxygen readings will effect pH accuracy.

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 4.7k Ω pull up resistor on both the isolated and non-isolated lines (R1, R2, R3, and R4) The output voltage is set using a voltage divider (R5, R6, and R7) this produces a voltage of 3.9V regardless of your input voltage.

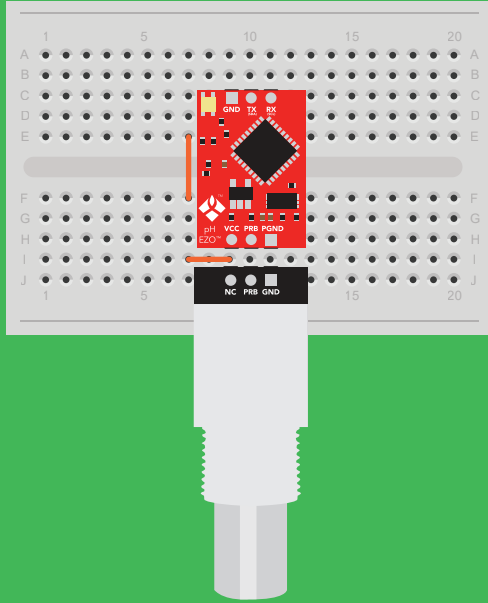
Isolated ground is different from non-isolated ground, these two lines should not be connected together.

VCC = 3.0v – 5.5v

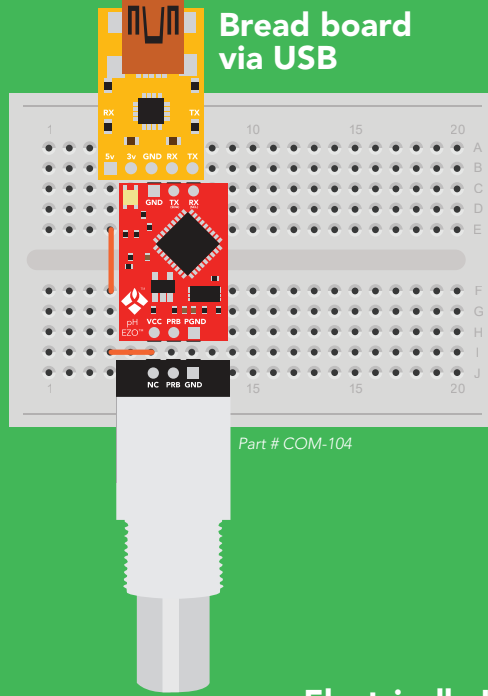


✓ Correct wiring

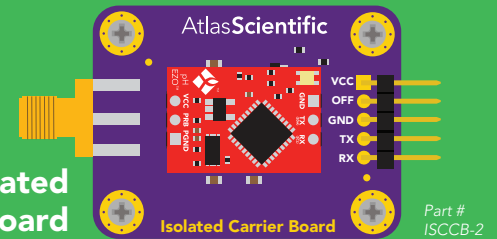
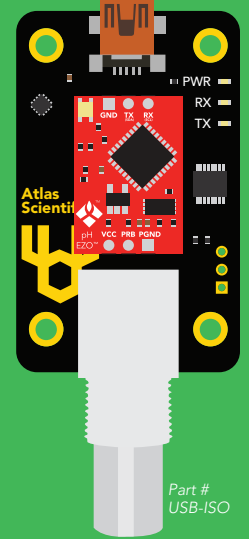
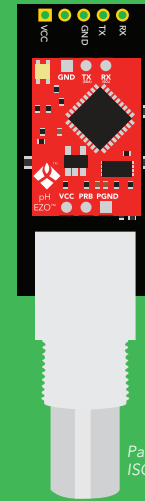
Bread board



Bread board
via USB



Carrier board USB
carrier board



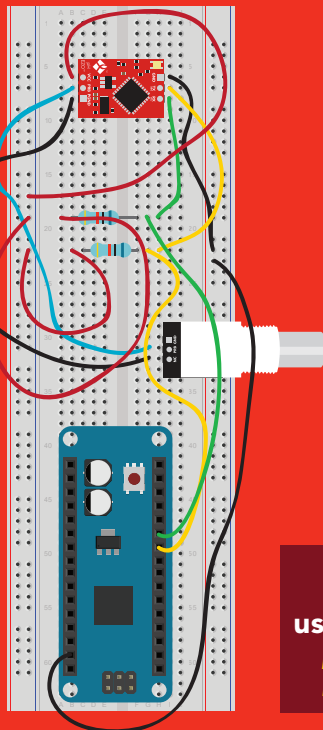
Electrically Isolated
EZO™ Carrier Board

X Incorrect wiring

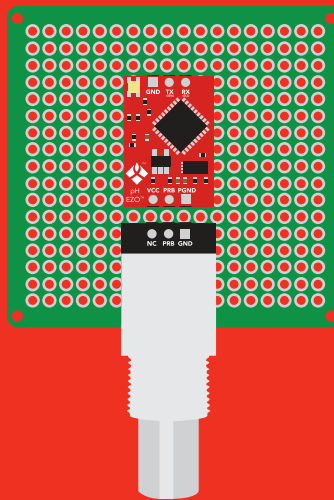
Extended leads



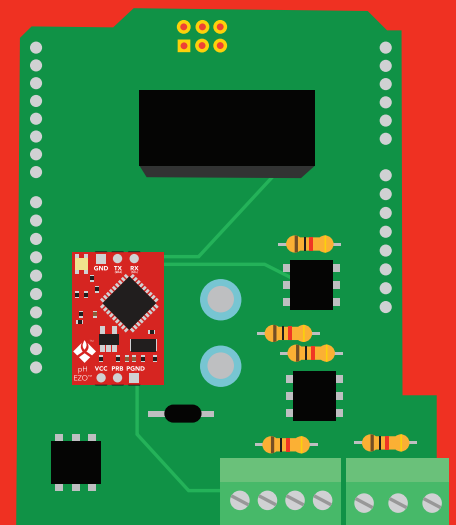
Sloppy setup



Perfboards or Protoboards



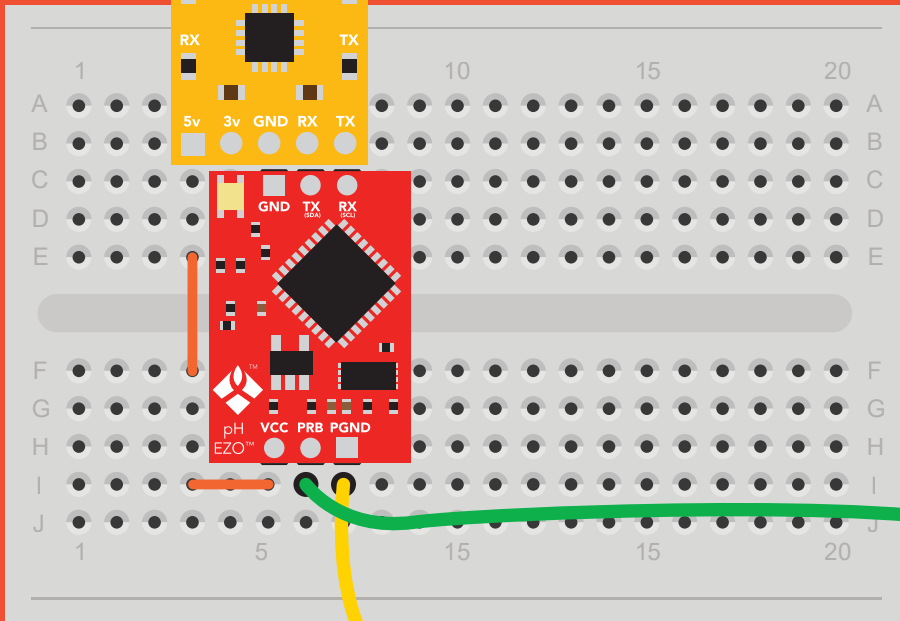
*Embedded into your device



NEVER
use Perfboards or Protoboards
Flux residue and shorting wires make it very hard to get accurate readings.

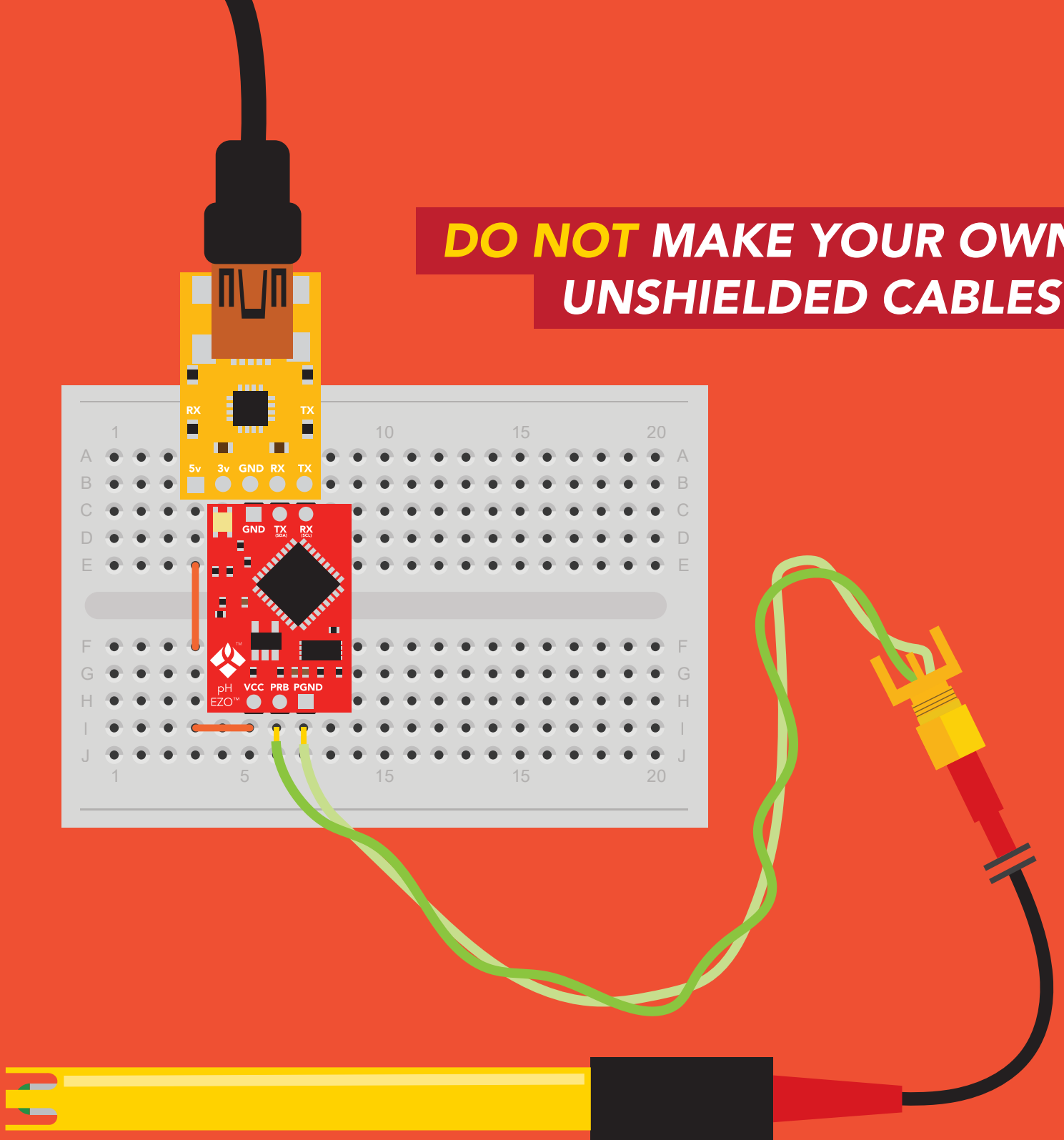
***Only after you are familiar
with EZO™ circuits operation**

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



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

**DO NOT MAKE YOUR OWN
UNSHIELDED CABLES!**



ONLY USE SHIELDED CABLES.

Calibration theory

Simple calibration

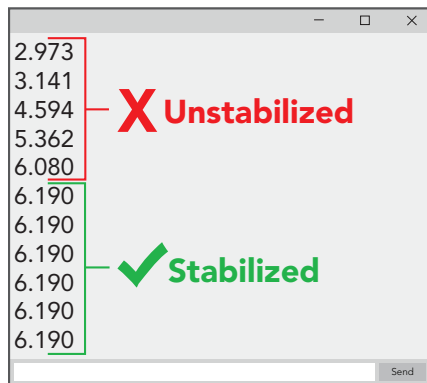
UART mode

Continuous readings

Advanced calibration

I²C mode

Continuously request readings



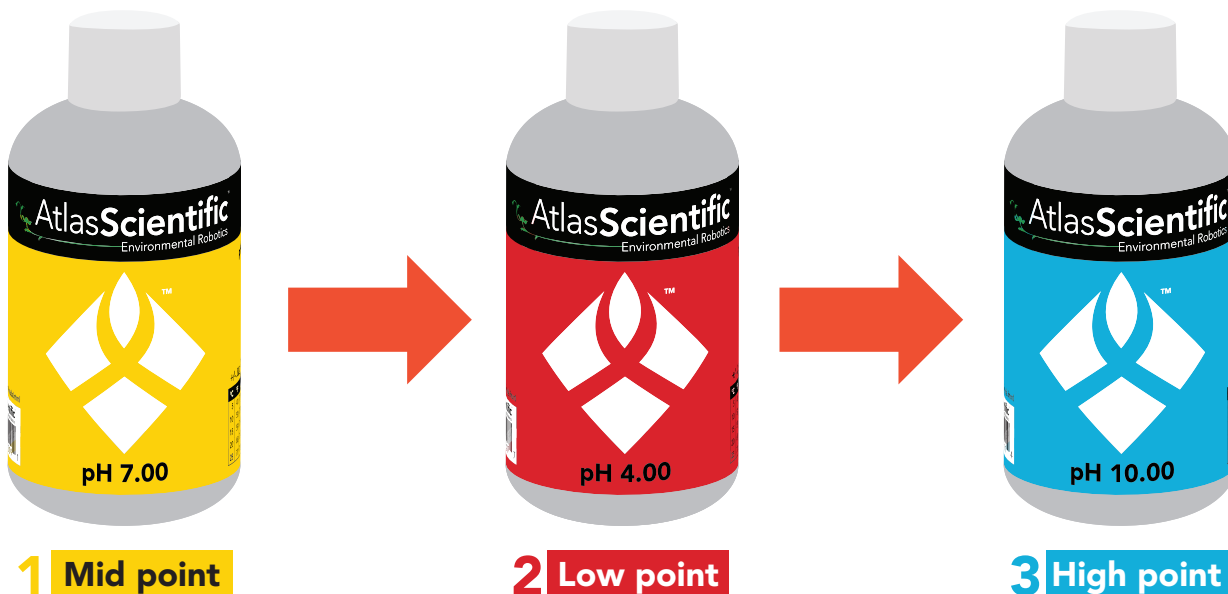
The most important part of calibration is watching the readings during the calibration process.

It's easiest to calibrate the device in its default state (UART mode, with continuous readings enabled).

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

Calibration order

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

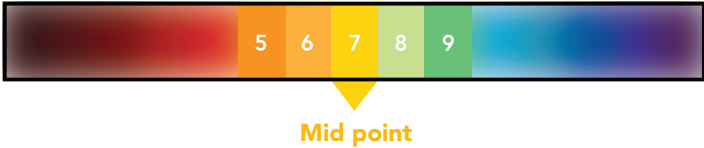


Single, Two point, or Three point calibration

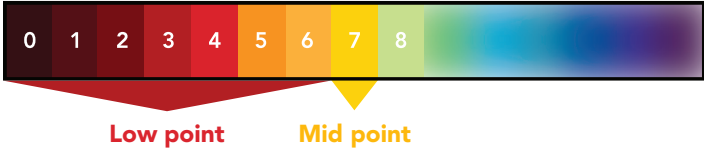
No calibration



Single point calibration



Two point calibration



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

Three point calibration

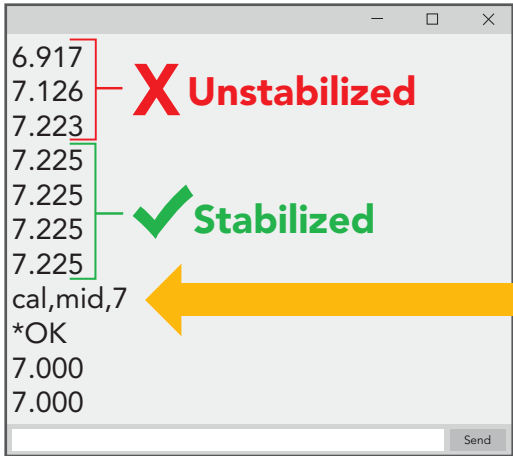
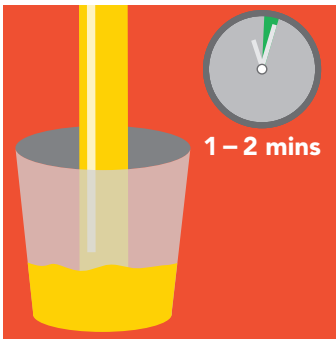
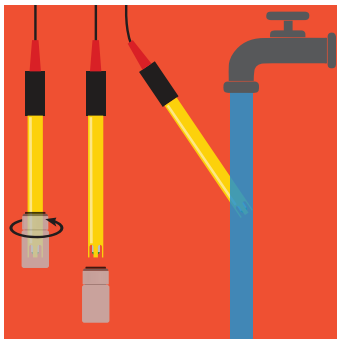


Three point calibration will provide high accuracy over the full pH range. Three point calibration at **4.00**, **7.00** and **10.00** should be considered the standard.

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

Mid point calibration

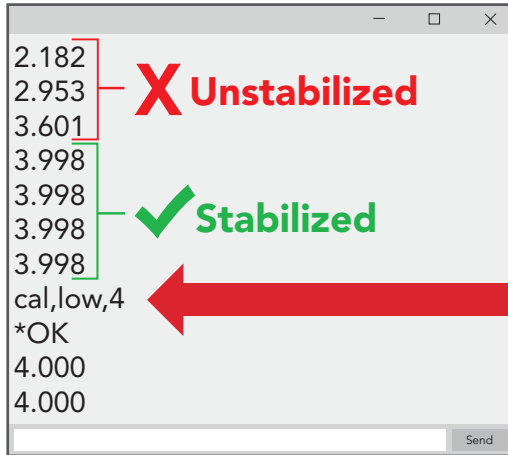
Remove the soaker bottle and rinse off the pH probe. Pour a small amount of the pH **7.00** calibration solution into a cup. Let the pH probe sit in the calibration solution until the readings stabilize (*small movement from one reading to the next is normal*).



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

Low point calibration

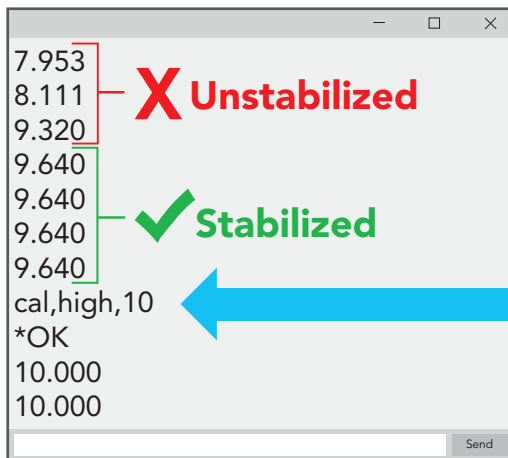
- Rinse off the probe before calibrating to the low point.
- Pour a small amount of the pH **4.00** calibration solution into a cup.
- Wait for readings to stabilize (1 – 2 minutes).



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

High point calibration

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



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



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



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

Default state

UART mode

Baud

9,600

Readings

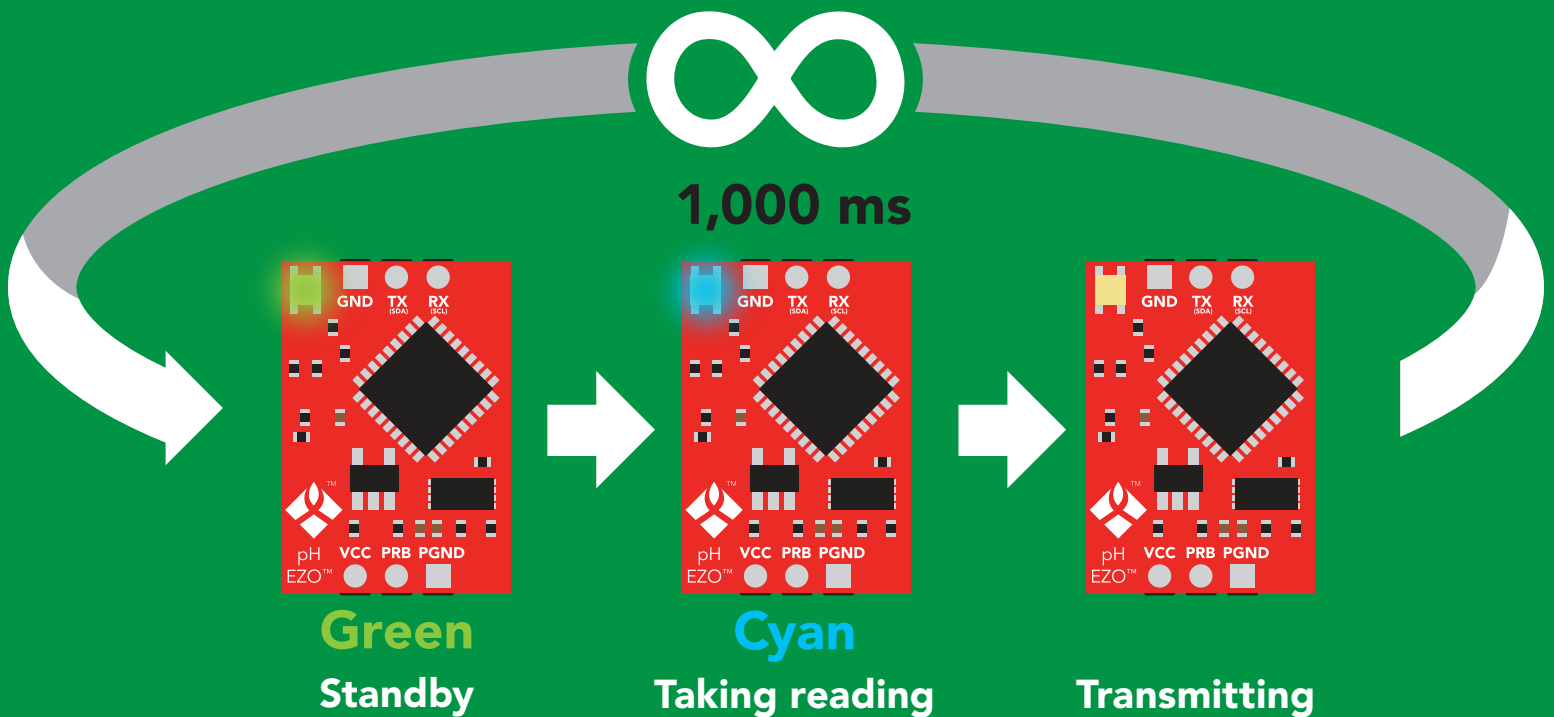
continuous

Speed

1 reading per second

LED

on



✓ Available data protocols

UART

Default

I²C

✗ Unavailable data protocols

SPI

Analog

RS-485

Mod Bus

4–20mA

UART mode

Settings that are retained if power is cut

- Baud rate
- Calibration
- Continuous mode
- Device name
- Enable/disable response codes
- Hardware switch to I²C mode
- LED control
- Protocol lock
- Software switch to I²C mode

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

- Find
- Sleep mode
- Temperature compensation

UART mode

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



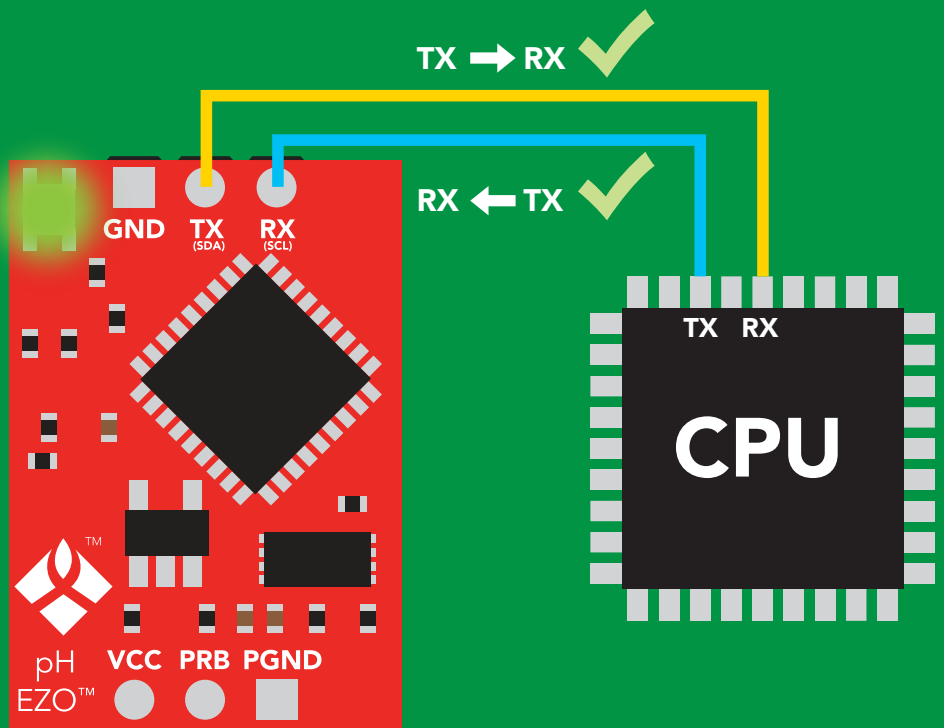
Vcc 3.3V – 5.5V

0V



Vcc

0V



Data format

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

Receiving data from device

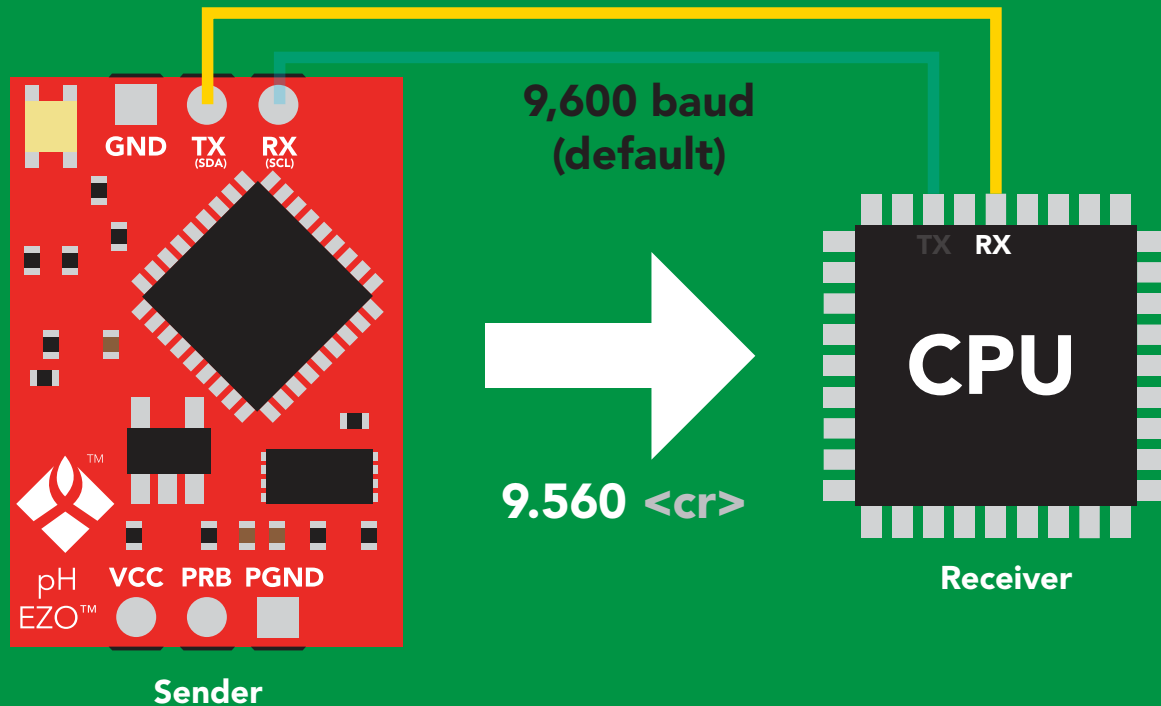
2 parts

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

Sending commands to device

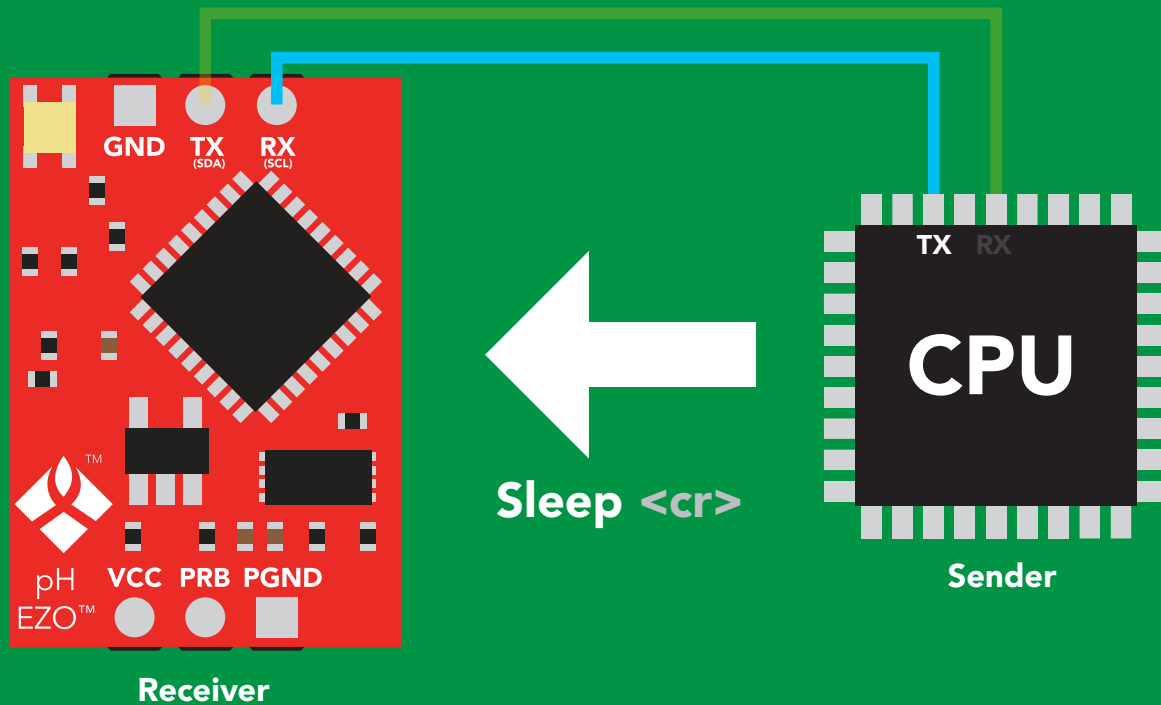
2 parts

Command (not case sensitive)

ASCII data string

Carriage return <cr>

Terminator



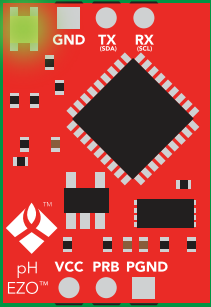
Advanced

ASCII: **S** **I** **e** **e** **p** **<cr>**

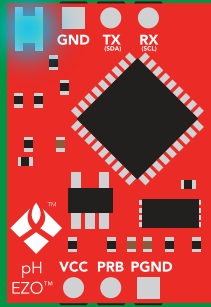
Hex: **53** **6C** **65** **65** **70** **0D**

Dec: **83** **108** **101** **101** **112** **13**

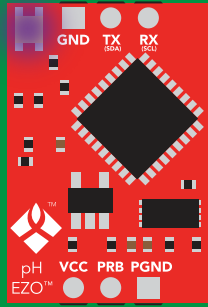
LED color definition



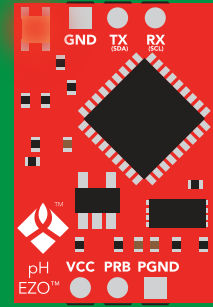
Green
UART standby



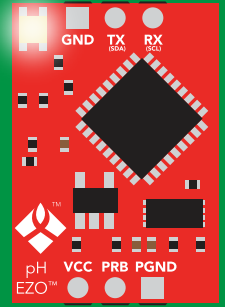
Cyan
Taking reading



Purple
Changing
baud rate



Red
Command
not understood



White
Find

5V

LED ON
+2.2 mA

3.3V

+0.6 mA

UART mode

command quick reference

All commands are ASCII strings or single ASCII characters.

Command	Function		Default state
Baud	change baud rate	pg. 36	9,600
C	enable/disable continuous reading	pg. 24	enabled
Cal	performs calibration	pg. 26	n/a
Export	export calibration	pg. 27	n/a
Factory	enable factory reset	pg. 38	n/a
Find	finds device with blinking white LED	pg. 23	n/a
i	device information	pg. 32	n/a
I2C	change to I ² C mode	pg. 39	not set
Import	import calibration	pg. 28	n/a
L	enable/disable LED	pg. 22	enabled
Name	set/show name of device	pg. 31	not set
Plock	enable/disable protocol lock	pg. 37	disabled
R	returns a single reading	pg. 25	n/a
Sleep	enter sleep mode/low power	pg. 35	n/a
Slope	returns the slope of the pH probe	pg. 29	n/a
Status	retrieve status information	pg. 34	enable
T	temperature compensation	pg. 30	25°C
*OK	enable/disable response codes	pg. 33	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>

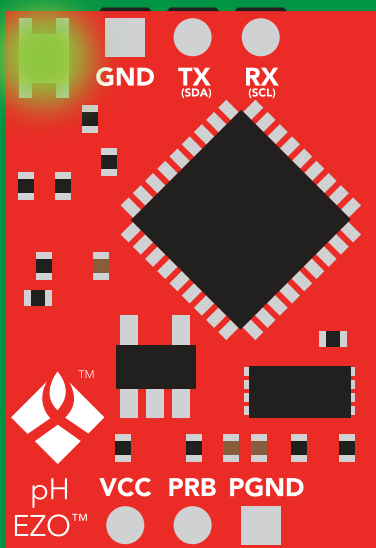
*OK <cr>

L,0 <cr>

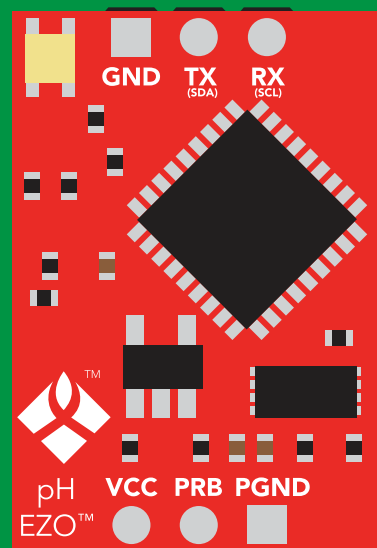
*OK <cr>

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.

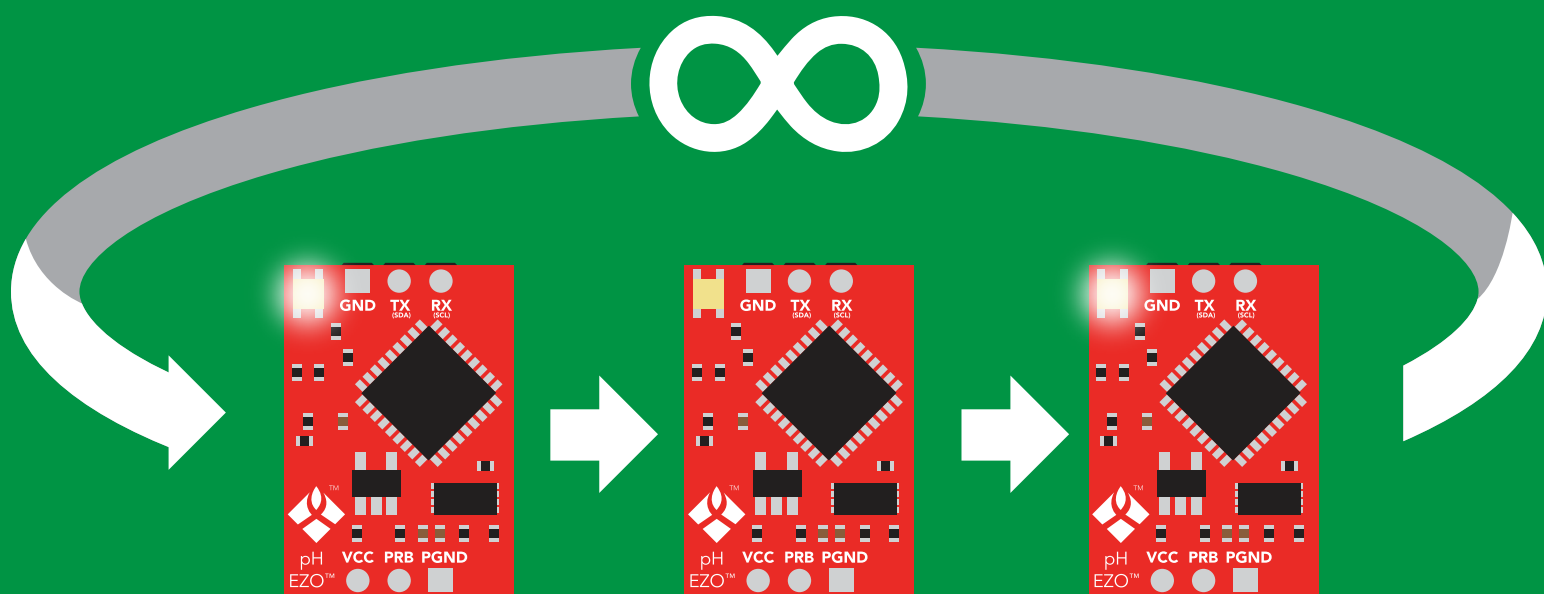
Find <cr> LED rapidly blinks white, used to help find device

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

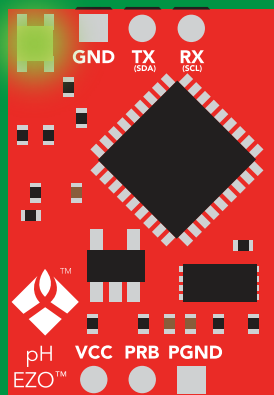
R <cr> takes single reading

Example

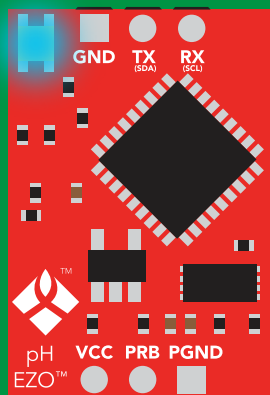
R <cr>

Response

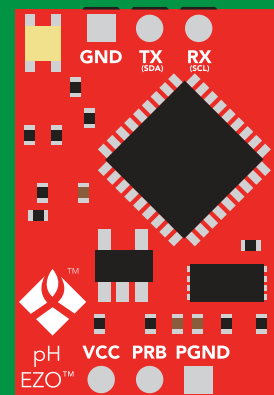
9.560 <cr>
***OK** <cr>



Green
Standby



Cyan
Taking reading



Transmitting



800 ms

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.

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

Example

Response

Cal,mid,7.00 <cr>

***OK <cr>**

Cal,low,4.00 <cr>

***OK <cr>**

Cal,high,10.00 <cr>

***OK <cr>**

Cal,clear <cr>

***OK <cr>**

Cal,? <cr>

?Cal,0 <cr> or ?Cal,1 <cr> or
one point
?Cal,2 <cr> or ?Cal,3 <cr>
two point three point
***OK <cr>**

Export calibration

Command syntax

Export: Use this command to download calibration settings

Export,? <cr> calibration string info

Export <cr> export calibration string from calibrated device

Example

Response

Export,? <cr>

10,120 <cr>

Response breakdown

10, 120

of strings to export

of bytes to export

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

Export <cr>

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

Export <cr>

65 20 61 20 63 6F <cr> **(2 of 10)**

(7 more)

⋮

Export <cr>

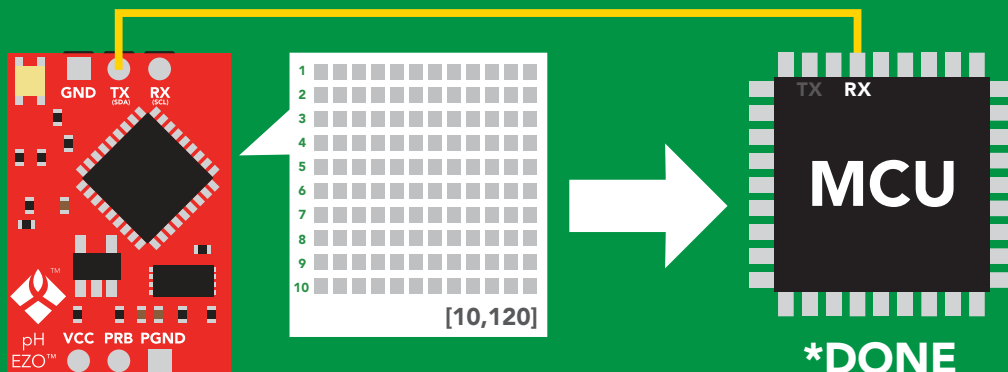
6F 6C 20 67 75 79 <cr> **(10 of 10)**

Export <cr>

***DONE**

Disabling *OK simplifies this process

Export <cr>



Import calibration

Command syntax

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

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

Example

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

Import, 65 20 61 20 63 6F <cr> (2 of 10)

⋮

Import, 6F 6C 20 67 75 79 <cr> (10 of 10)

Response

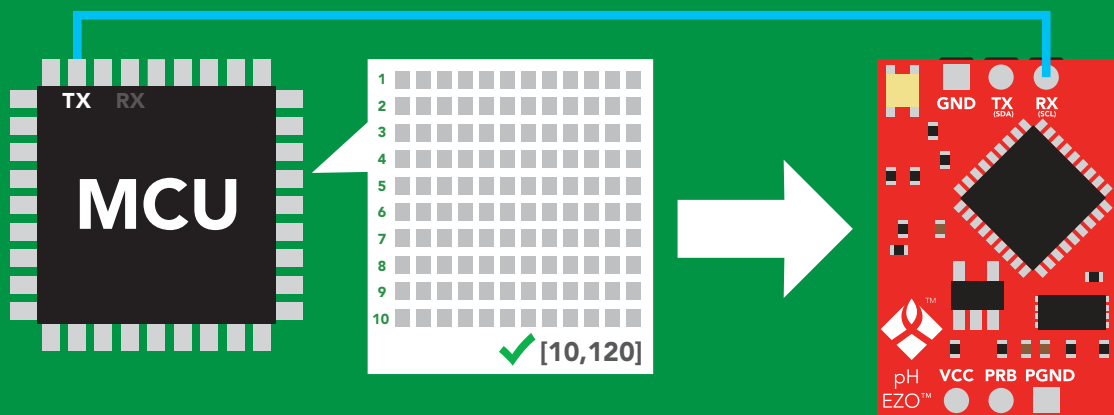
***OK** <cr>

***OK** <cr>

⋮

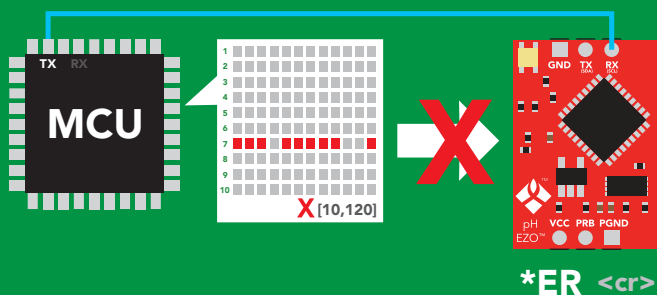
***OK** <cr>

Import,n <cr>



***OK** <cr>

system will reboot



***ER** <cr>

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

Slope

Command syntax

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

Slope,? <cr>

Response

?Slope,99.7,100.3, -0.89 <cr>
***OK <cr>**

Response breakdown

?Slope,

99.7

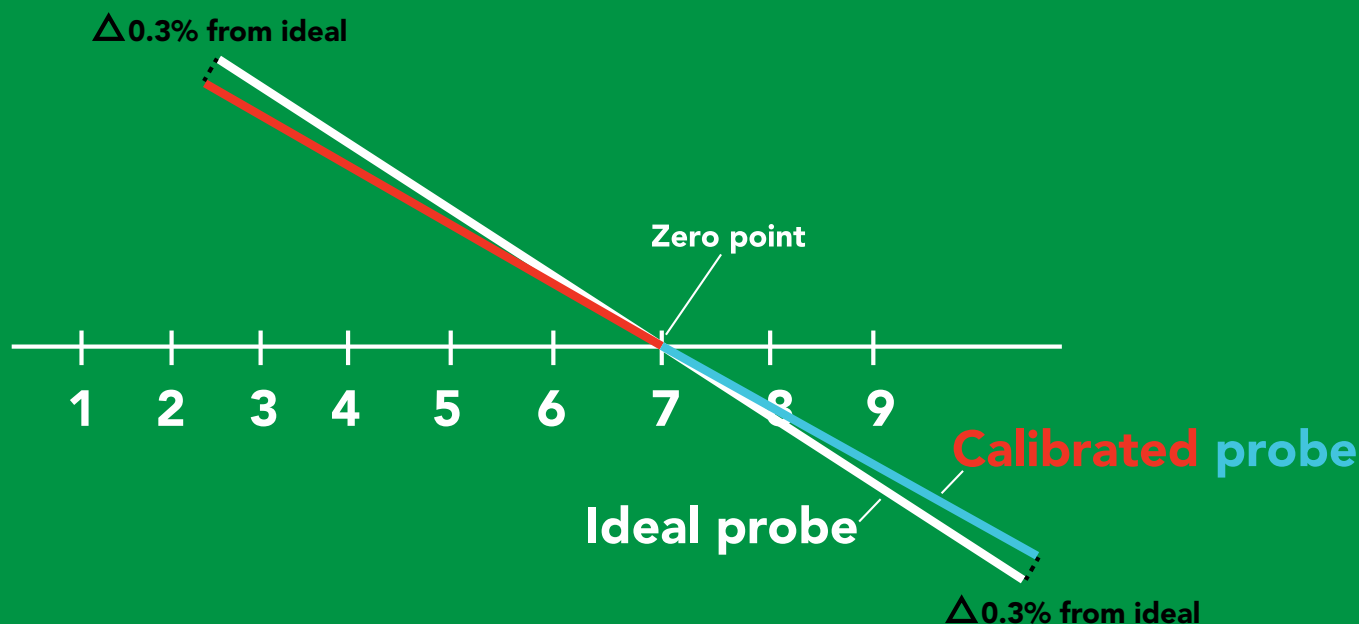
99.7% is how closely the slope of the **acid** calibration line matched the "ideal" pH probe.

100.3

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

-0.89

This is how many millivolts the zero point is off from true 0.



Temperature compensation

Command syntax

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

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

T,? <cr> compensated temperature value?

RT,n <cr> set temperature compensation and take a reading*

This is a new command
for firmware V2.12

Example

Response

T,19.5 <cr>

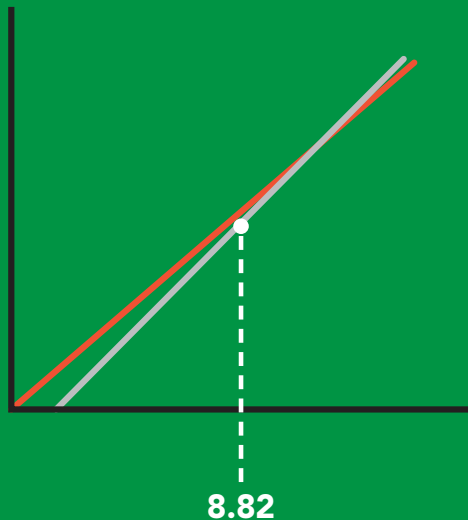
*OK <cr>

RT,19.5 <cr>

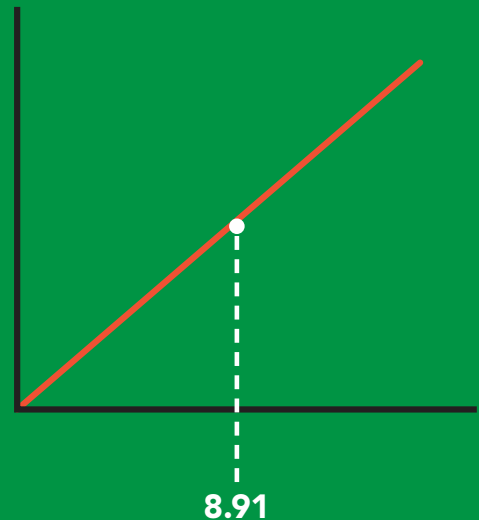
*OK <cr>
8.91 <cr>

T,? <cr>

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



→
T,19.5 <cr>



Naming device

Command syntax

Name,n <cr> set name

Name,? <cr> show name

n =

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

Up to 16 ASCII characters

Example

Name,zzt <cr>

Response

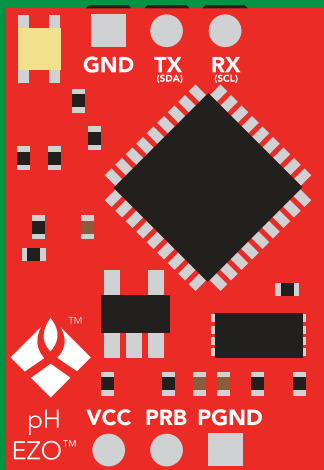
*OK <cr>

Name,? <cr>

?Name,zzt <cr>

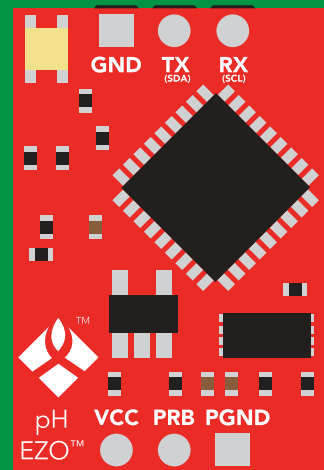
*OK <cr>

Name,zzt



*OK <cr>

Name,?



Name,zzt <cr>

*OK <cr>

Device information

Command syntax

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

Example

```
i <cr>
```

Response

```
?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>

no response, ***OK** disabled

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 ($VCC \geq 5.5V$)
***UV** under volt ($VCC \leq 3.1V$)
***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

Status <cr>

Response

?Status,P,5.038 <cr>
*OK <cr>

Response breakdown

?Status,	P,	5.038
	↑	↑
	Reason for restart	Voltage at Vcc

Restart codes

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

Sleep mode/low power

Command syntax

Send any character or command to awaken device.

Sleep <cr> enter sleep mode/low power

Example

Response

Sleep <cr>

***OK** <cr>

***SL** <cr>

Any command

***WA** <cr> wakes up device

5V

STANDBY

16 mA

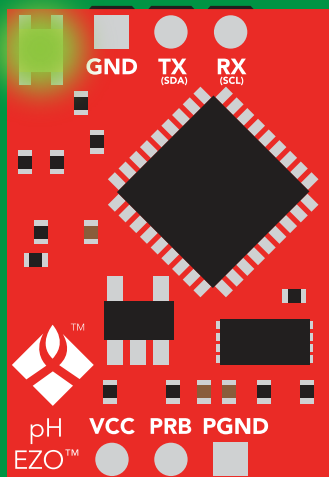
SLEEP

1.16 mA

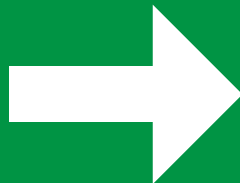
3.3V

13.9 mA

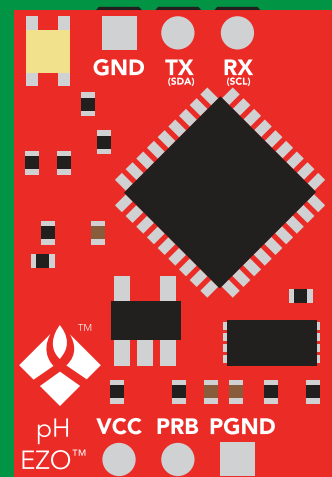
0.995 mA



Standby
16 mA



Sleep <cr>



Sleep
1.16 mA

Change baud rate

Command syntax

Baud,n <cr> change baud rate

Example

Baud,38400 <cr>

Response

*OK <cr>

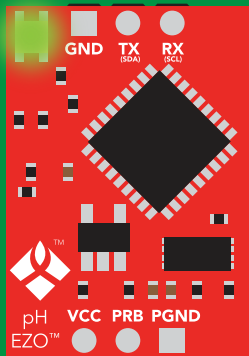
Baud,? <cr>

?Baud,38400 <cr>

*OK <cr>

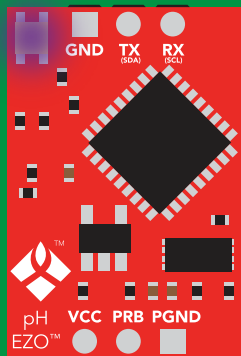
n =

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



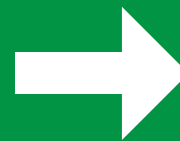
Standby

Baud,38400 <cr>

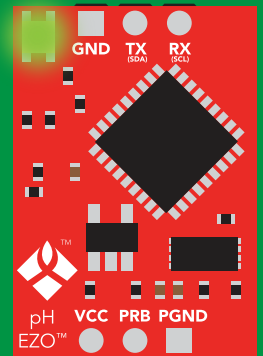


Changing
baud rate

*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

Response

Plock,1 <cr>

*OK <cr>

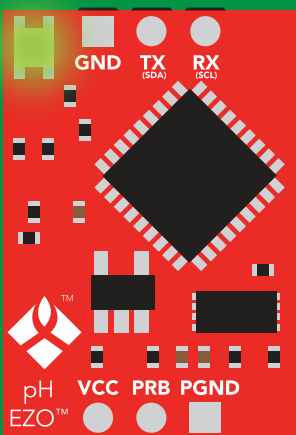
Plock,0 <cr>

*OK <cr>

Plock,? <cr>

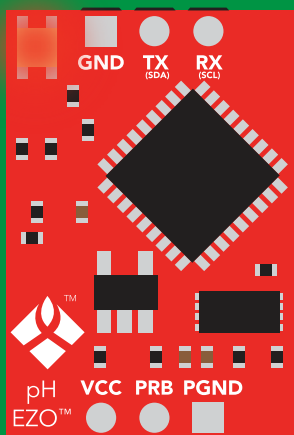
?Plock,1 <cr> or ?Plock,0 <cr>

Plock,1



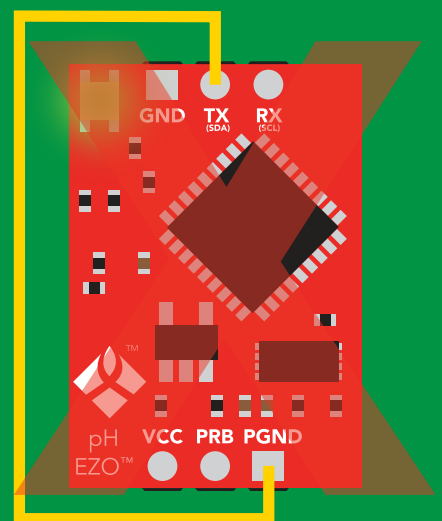
*OK <cr>

I2C,100



cannot change to I²C
*ER <cr>

Short



cannot change to I²C

Factory reset

Command syntax

Clears calibration
LED on
"*OK" enabled

Factory <cr> enable factory reset

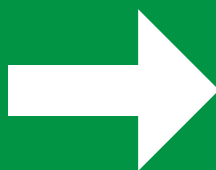
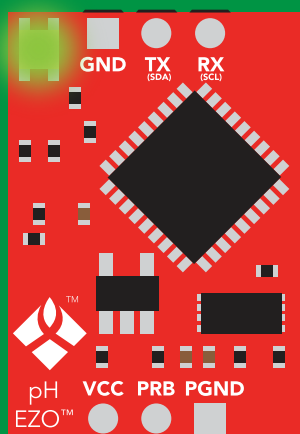
Example

Response

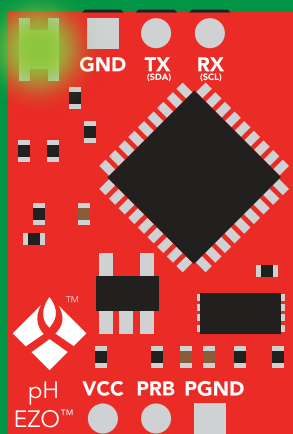
Factory <cr>

*OK <cr>

Factory <cr>



(reboot)



*OK <cr>

*RS <cr>

*RE <cr>

Baud rate will not change

Change to I²C mode

Command syntax

Default I²C address 99 (0x63)

I2C,n <cr> sets I²C address and reboots into I²C mode

n = any number 1 – 127

Example

I2C,100 <cr>

Response

*OK (reboot in I²C mode)

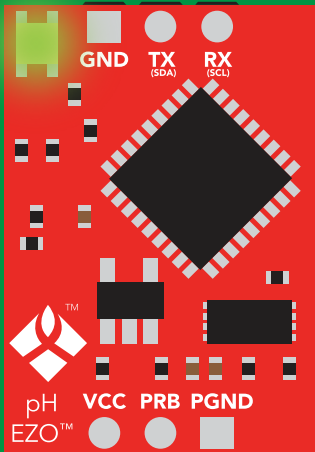
Wrong example

I2C,139 <cr> n ≠ 127

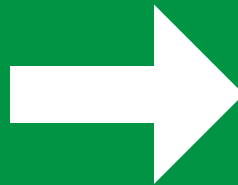
Response

*ER <cr>

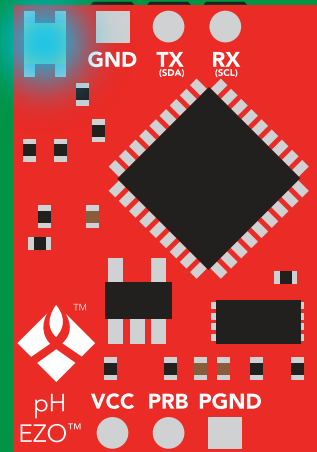
I2C,100



Green
*OK <cr>



(reboot)



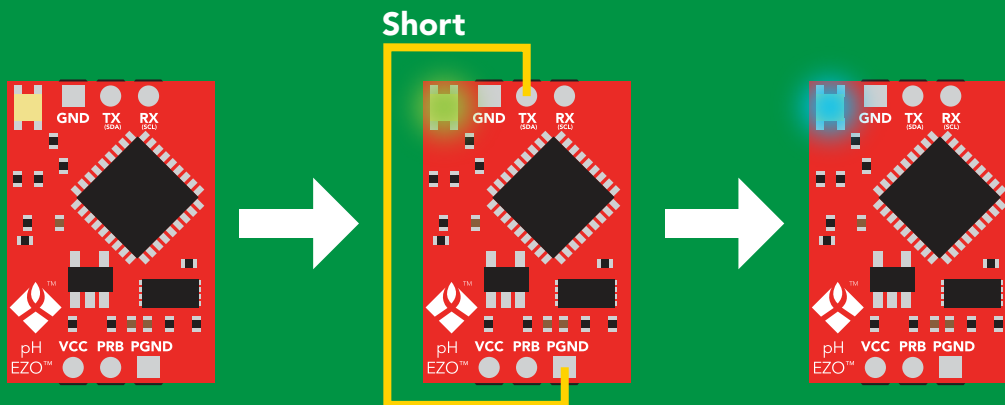
Blue
now in I²C mode

Manual switching to I²C

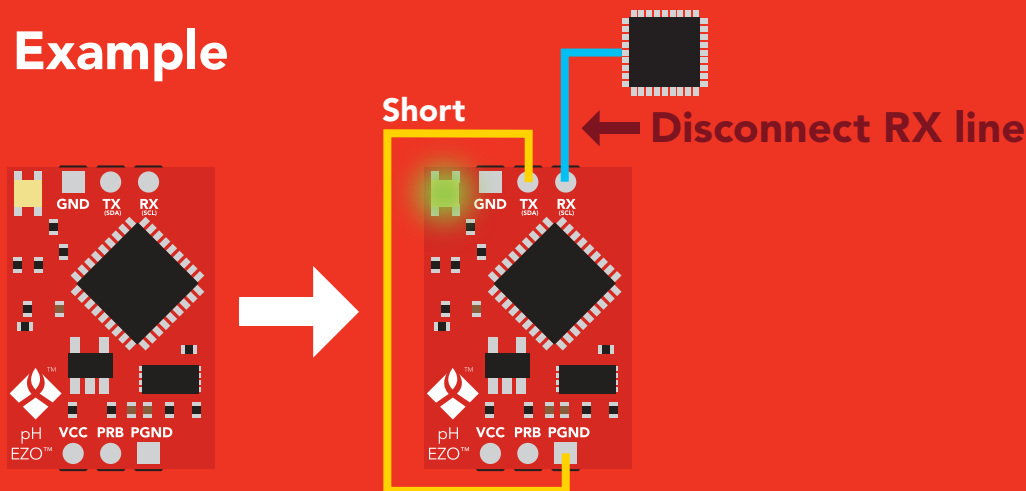
- Make sure Plock is set to 0
- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to PGND
- Confirm RX is disconnected
- Connect ground (power on)
- Wait for LED to change from **Green** to **Blue**
- Disconnect ground (power off)
- Reconnect all data and power

Manually switching to I²C will set the I²C address to 99 (0x63)

Example



Wrong Example



I²C mode

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

To set your EZO™ device into I²C mode [click here](#)

Settings that are retained if power is cut

- Calibration
- Change I²C address
- Hardware switch to UART mode
- LED control
- Protocol lock
- Software switch to UART mode

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

- Find
- Sleep mode
- Temperature compensation

I²C mode

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

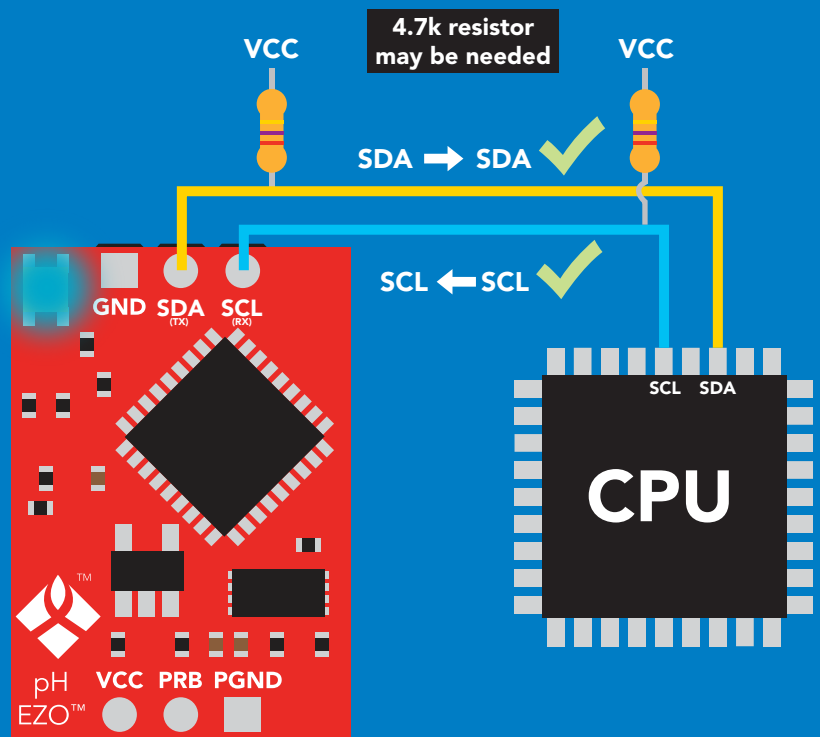
Vcc 3.3V – 5.5V

Clock speed 100 – 400 kHz

SDA 

SCL 





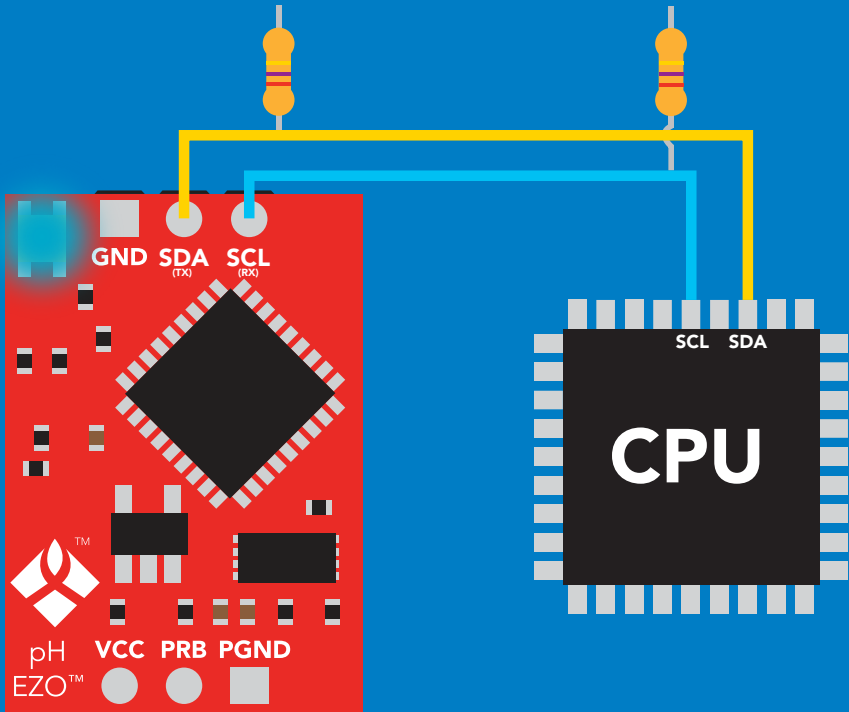
Data format

Reading	pH	Data type	floating point
Units	pH	Decimal places	3
Encoding	ASCII	Smallest string	4 characters
Format	string	Largest string	40 characters

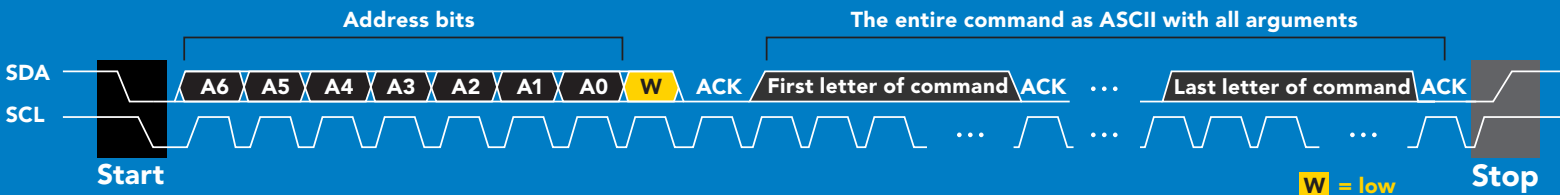
Sending commands to device



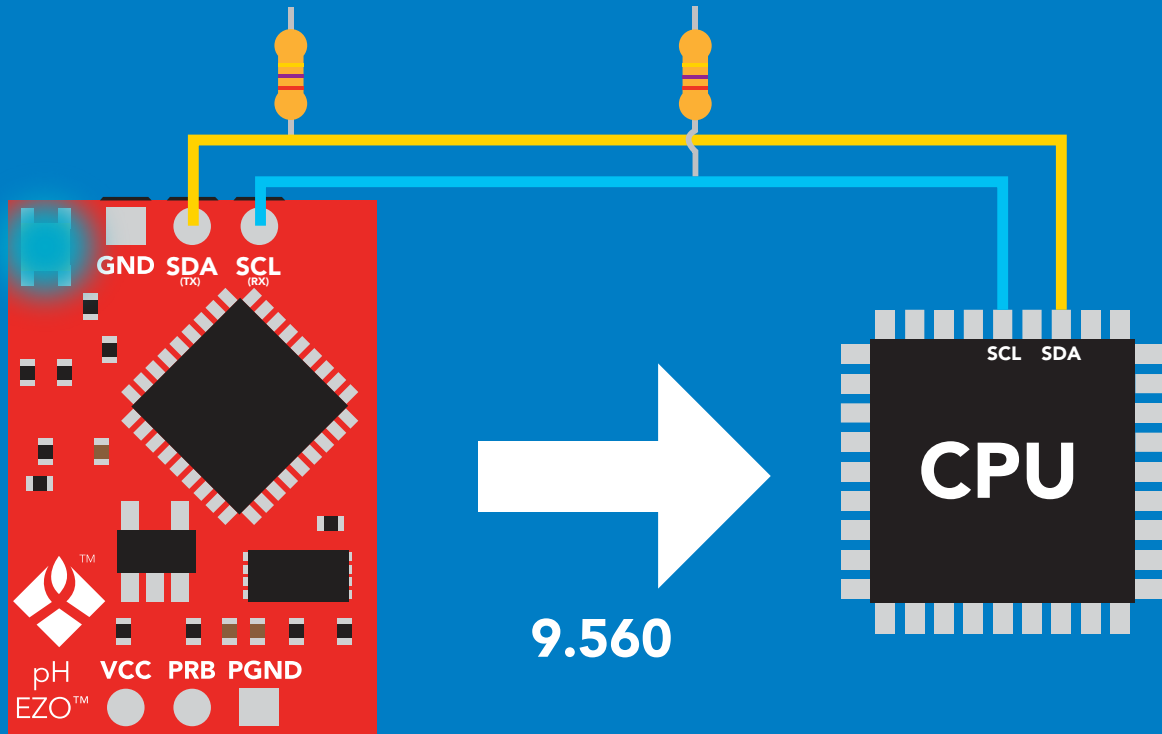
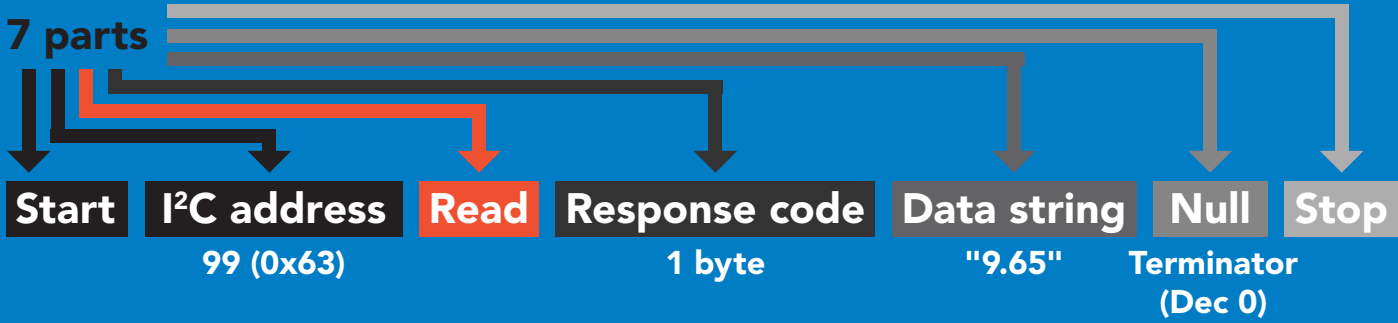
Example



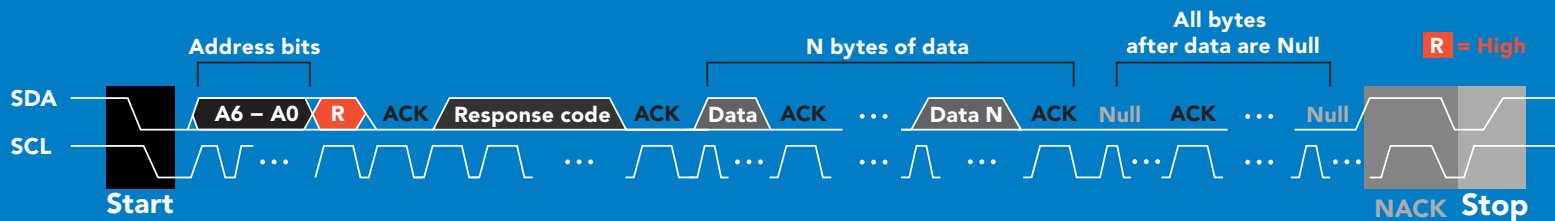
Advanced



Requesting data from device



Advanced



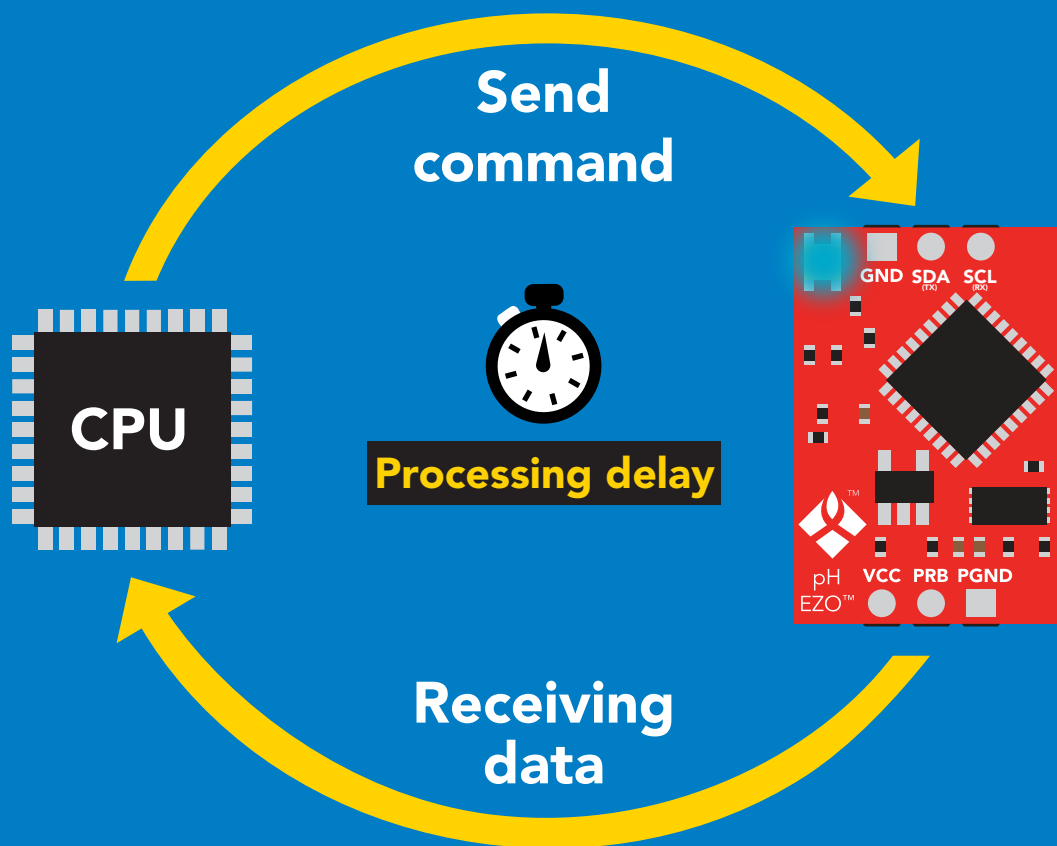
1 **57** **46** **53** **54** **48** **0** = 9.560

Dec ASCII Dec

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;
```

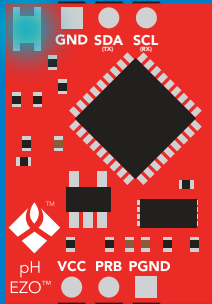
If there is no processing delay or the processing delay is too short, the response code will always be 254.

Response codes

Single byte, not string

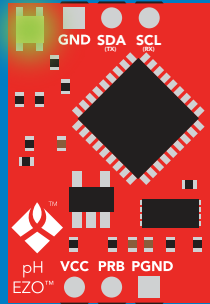
255	no data to send
254	still processing, not ready
2	syntax error
1	successful request

LED color definition



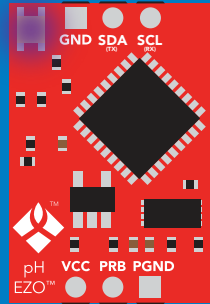
Blue

I²C standby



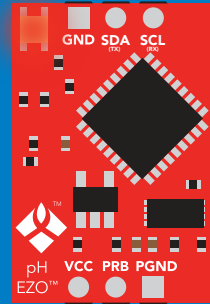
Green

Taking reading



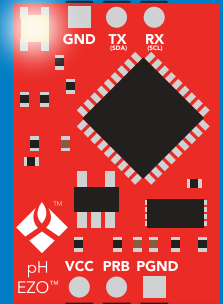
Purple

**Changing
I²C address**



Red

**Command
not understood**



White

Find

5V

**LED ON
+2.2 mA**

3.3V

+0.6 mA

I²C mode

command quick reference

All commands are ASCII strings or single ASCII characters.

Command	Function	
Baud	switch back to UART mode	pg. 62
Cal	performs calibration	pg. 51
Export	export calibration	pg. 52
Factory	enable factory reset	pg. 61
Find	finds device with blinking white LED	pg. 49
i	device information	pg. 56
I2C	change I ² C address	pg. 60
Import	import calibration	pg. 53
L	enable/disable LED	pg. 48
Plock	enable/disable protocol lock	pg. 59
R	returns a single reading	pg. 50
Sleep	enter sleep mode/low power	pg. 58
Slope	returns the slope of the pH probe	pg. 54
Status	retrieve status information	pg. 57
T	temperature compensation	pg. 55

LED control

Command syntax

300ms  processing delay

L,1 LED on **default**

L,0 LED off

L,? LED state on/off?

Example

Response

L,1


Wait 300ms

1	0
Dec	Null

L,0


Wait 300ms

1	0
Dec	Null

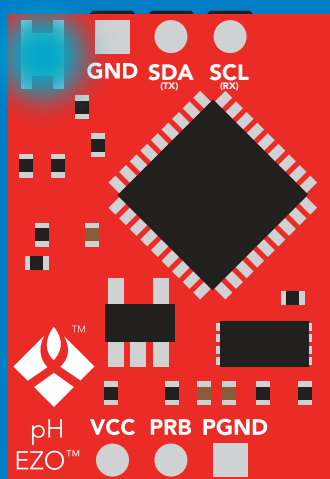
L,?


Wait 300ms

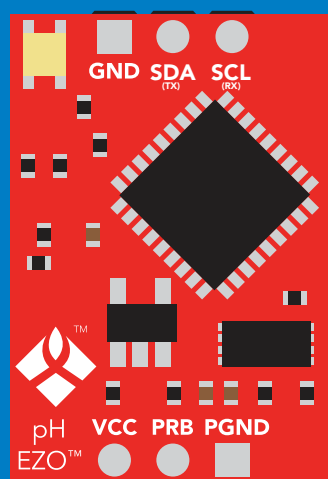
1	?L,1	0
Dec	ASCII	Null

or

1	?L,0	0
Dec	ASCII	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.

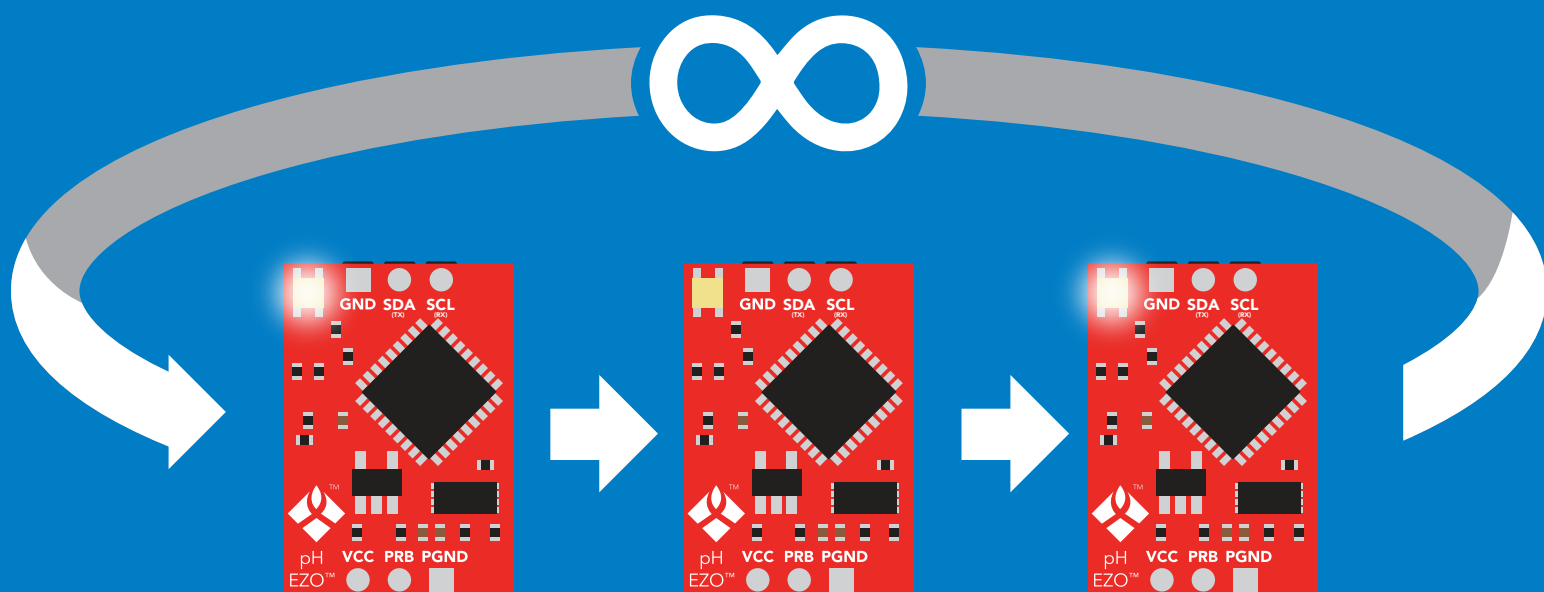
Find LED rapidly blinks white, used to help find device

Example

Response

Find

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



Taking reading

Command syntax

900ms  processing delay

R return 1 reading

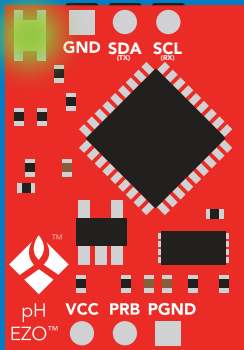
Example

Response

R

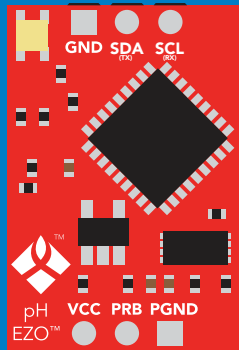
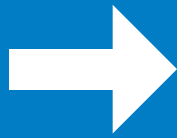

Wait 900ms

1	9.560	0
Dec	ASCII	Null

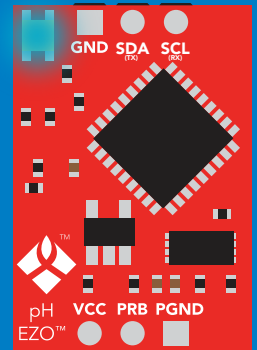
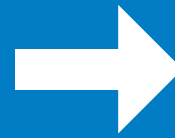


Green

Taking reading



Transmitting



Blue

Standby

Calibration

900ms  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.

Command syntax

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

Example

Response

Cal,mid,7.00

 **1** **0**
Wait 900ms Dec Null

Cal,low,4.00

 **1** **0**
Wait 900ms Dec Null


Cal,high,10.00

 **1** **0**
Wait 900ms Dec Null

Cal,clear

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

Cal,?

 **1** **?Cal,0** **0** or **1** **?Cal,1** **0**
Wait 300ms Dec ASCII Null one point

1 **?Cal,2** **0** or **1** **?Cal,3** **0**
Dec ASCII Null two point three point

Export calibration

300ms  processing delay

Command syntax

Export: Use this command to download calibration settings

Export,? calibration string info

Export export calibration string from calibrated device

Example

Response

Export,?



Wait 300ms

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



Wait 300ms

1

Dec

59 6F 75 20 61 72

ASCII

0

Null

(1 of 10)

Export



Wait 300ms

1

Dec

65 20 61 20 63 6F

ASCII

0

Null

(2 of 10)

(7 more)

⋮

Export



Wait 300ms

1

Dec

6F 6C 20 67 75 79

ASCII

0

Null

(10 of 10)

Export



Wait 300ms

1

Dec

*DONE

ASCII

0

Null

Import calibration

300ms  processing delay

Command syntax

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

Import,n import calibration string to new device

Example

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

Import, 65 20 61 20 63 6F (2 of 10)

⋮

Import, 6F 6C 20 67 75 79 (10 of 10)

Response

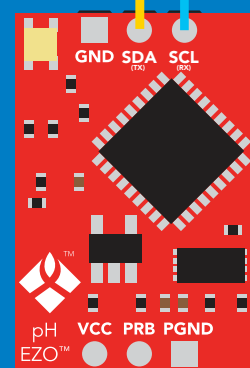
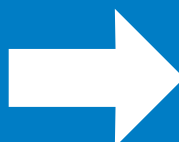
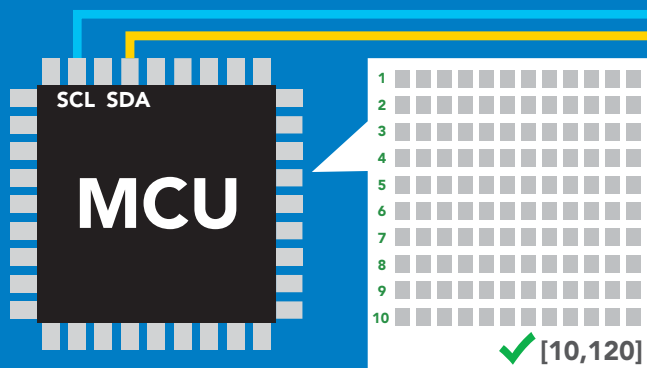
 **1** **0**
Wait 300ms Dec Null

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

⋮

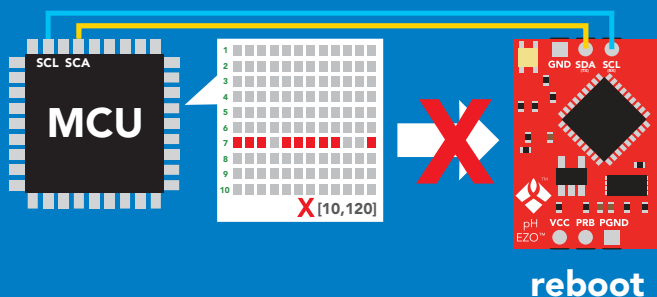
 **1** **0**
Wait 300ms Dec Null

Import,n



1 ***Pending** **0**
Dec ASCII Null

system will reboot



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

Slope

300ms  processing delay

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,? returns the slope of the pH probe

Example

Slope,?

Response



1
Dec

?Slope,99.7,100.3, -0.89
ASCII

0
Null

Response breakdown

?Slope,

99.7

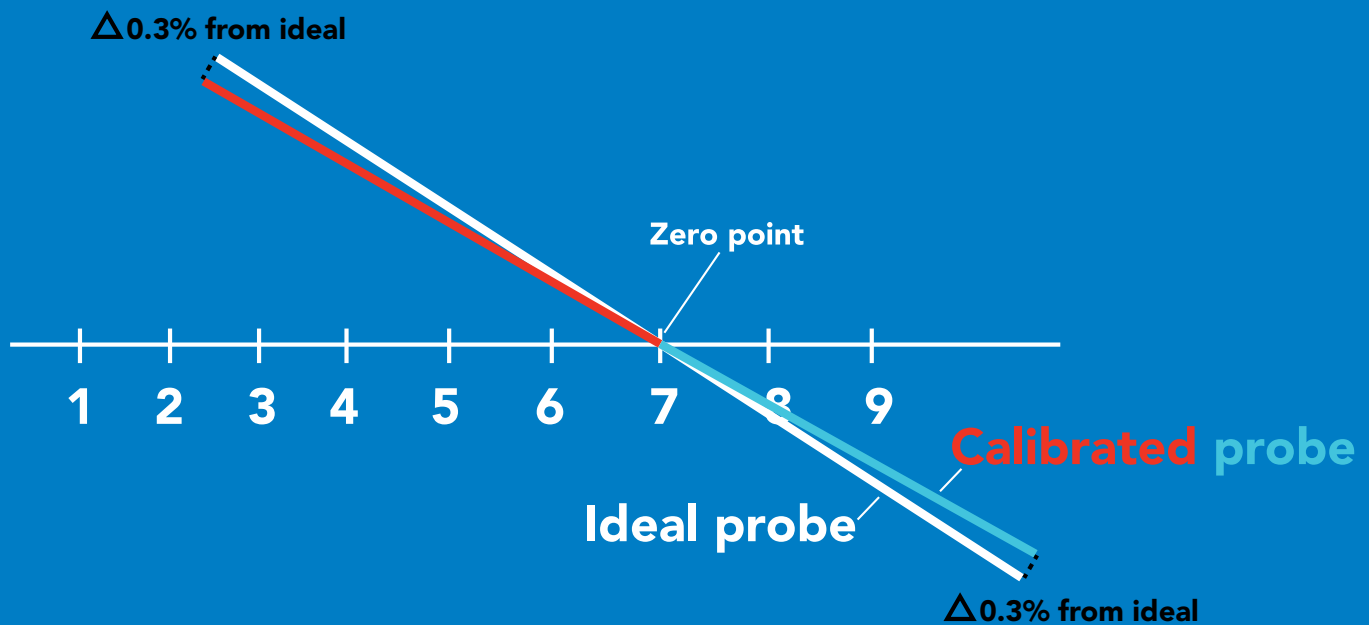
↑
99.7% is how closely the slope of the **acid** calibration line matched the "ideal" pH probe.

100.3

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

-0.89

↑
This is how many millivolts the zero point is off from true 0.



Temperature compensation

Command syntax

Default temperature = 25°C

Temperature is always in Celsius

Temperature is not retained if power is cut

T,n n = any value; floating point or int 300ms  processing delay

T,? compensated temperature value?

RT,n set temperature compensation and take a reading*

This is a new command
for firmware V2.12

Example

Response

T,19.5

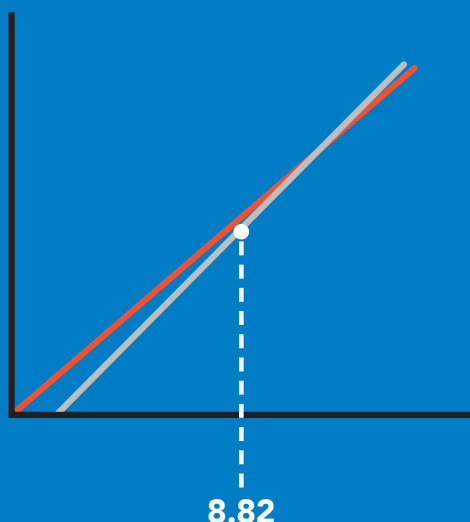
 Wait 300ms 1 Dec 0 Null

RT,19.5

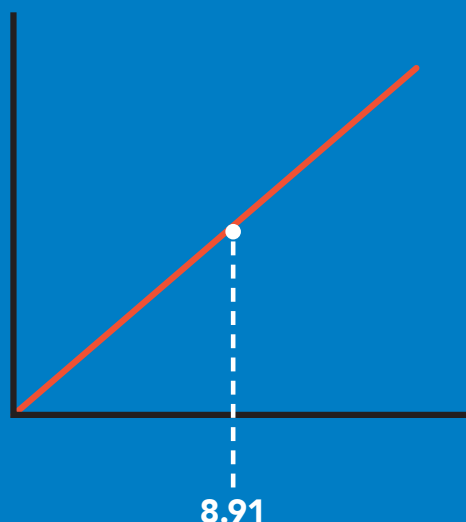
 Wait 900ms 1 Dec 8.91 ASCII 0 Null

T,?

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



→
T,19.5



Device information

Command syntax

300ms  processing delay

i device information

Example

i

Response



Wait 300ms

1

Dec

?i,pH,1.98

ASCII

0

Null

Response breakdown

?i,

pH,

↑
Device

1.98

↑
Firmware

Reading device status

Command syntax

300ms  processing delay

Status voltage at Vcc pin and reason for last restart

Example

Response

Status



1
Dec

?Status,P,5.038
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.

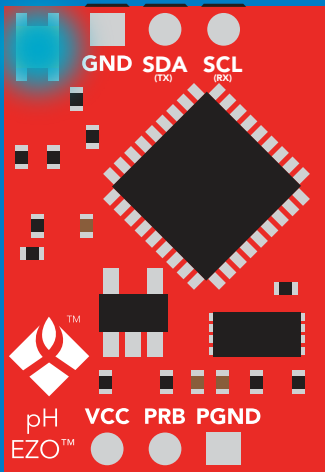
Example	Response
---------	----------

Sleep	no response
-------	-------------

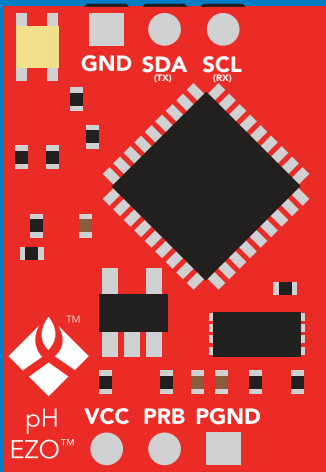
Do not read status byte after issuing sleep command.

Any command	wakes up device
-------------	-----------------

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



Standby



Sleep

Protocol lock

Command syntax

300ms  processing delay

Plock,1 enable Plock

Plock,0 disable Plock

Plock,? Plock on/off?

Locks device to I²C mode.

default

Example

Response


Plock,1

 Wait 300ms
1 0
Dec Null

Plock,0

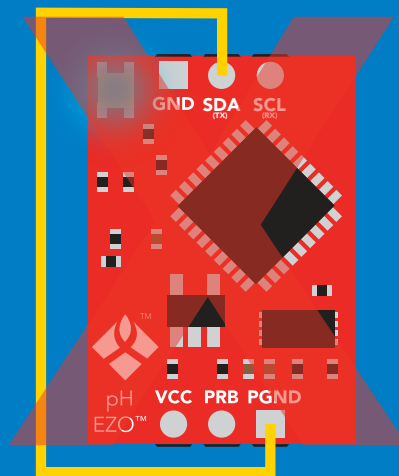
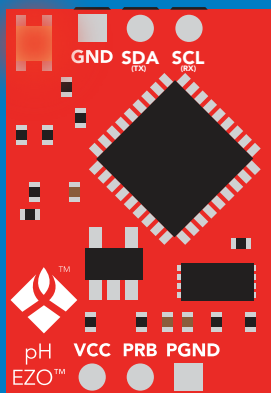
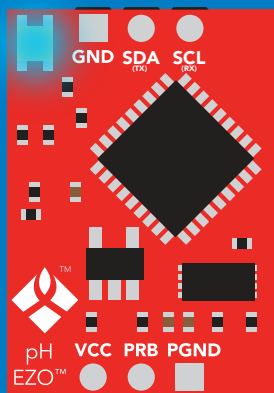
 Wait 300ms
1 0
Dec Null

Plock,?

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

Plock,1

Baud, 9600



cannot change to UART

cannot change to UART

I²C address change

Command syntax

300ms  processing delay

I2C,n sets I²C address and reboots into I²C mode

Example

I2C,100

Response

device reboot

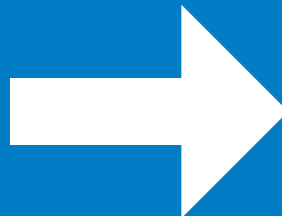
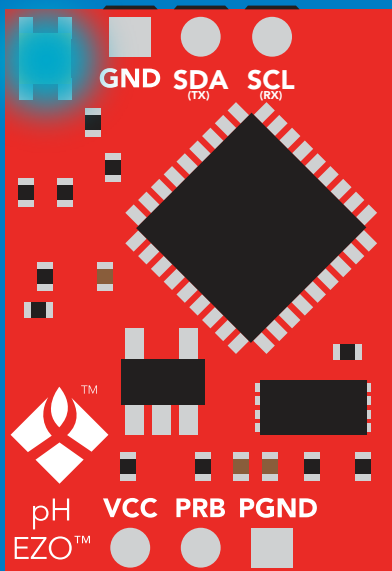
Warning!

Changing the I²C address will prevent communication between the circuit and the CPU until the CPU is updated with the new I²C address.

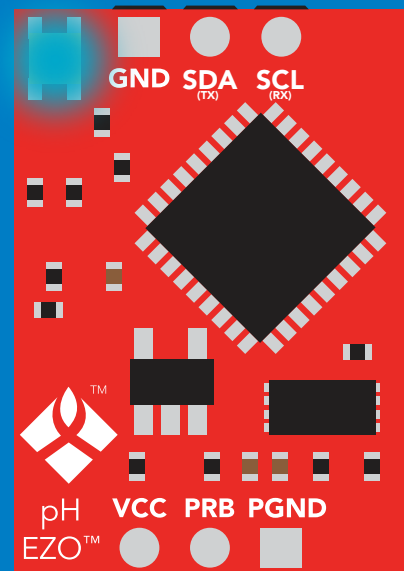
Default I²C address is 99 (0x63).

n = any number 1 – 127

I2C,100



(reboot)



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

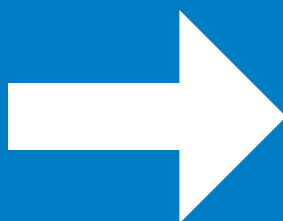
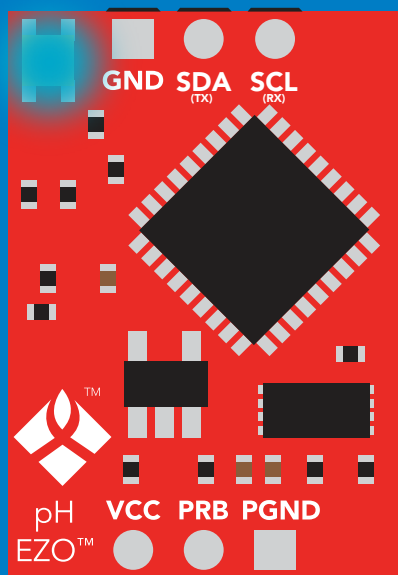
Response

Factory

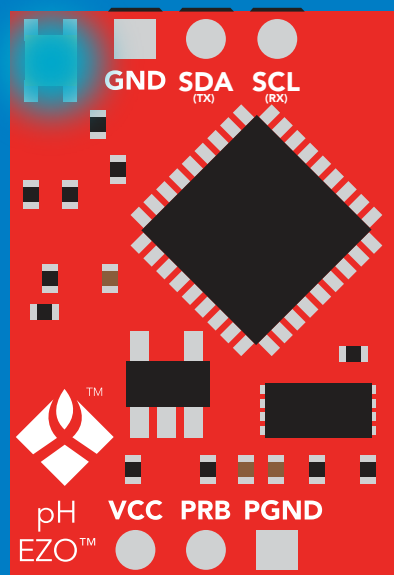
device reboot

Clears calibration
LED on
Response codes enabled

Factory



(reboot)



Change to UART mode

Command syntax

Baud,n switch from I²C to UART

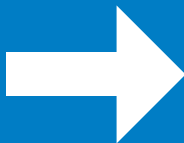
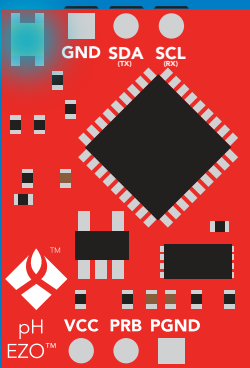
Example

Baud,9600

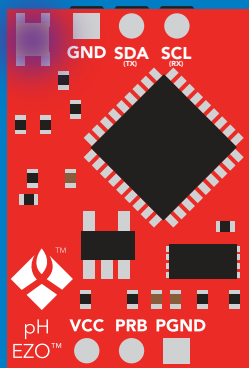
Response

reboot in UART mode

n = $\left\{ \begin{array}{l} 300 \\ 1200 \\ 2400 \\ 9600 \\ 19200 \\ 38400 \\ 57600 \\ 115200 \end{array} \right.$



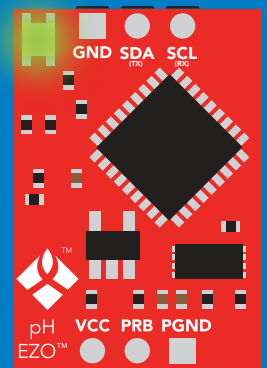
Baud,9600



Changing to UART mode



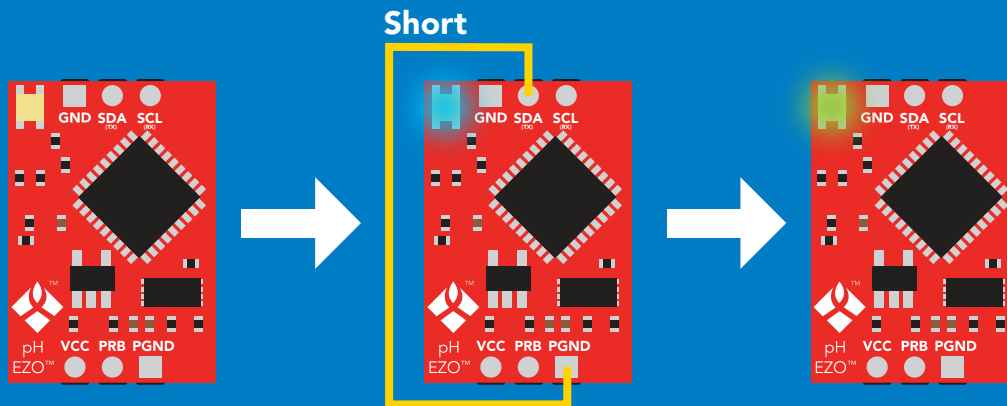
(reboot)



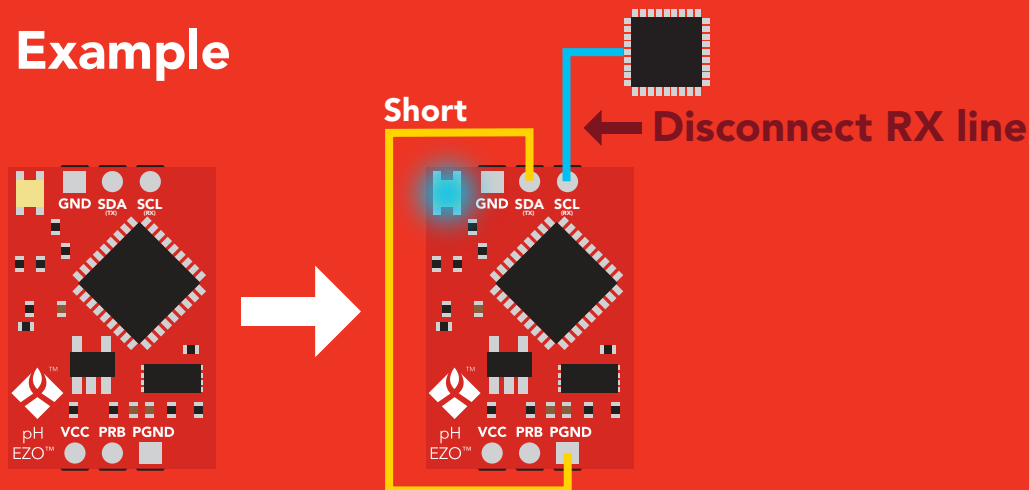
Manual switching to UART

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

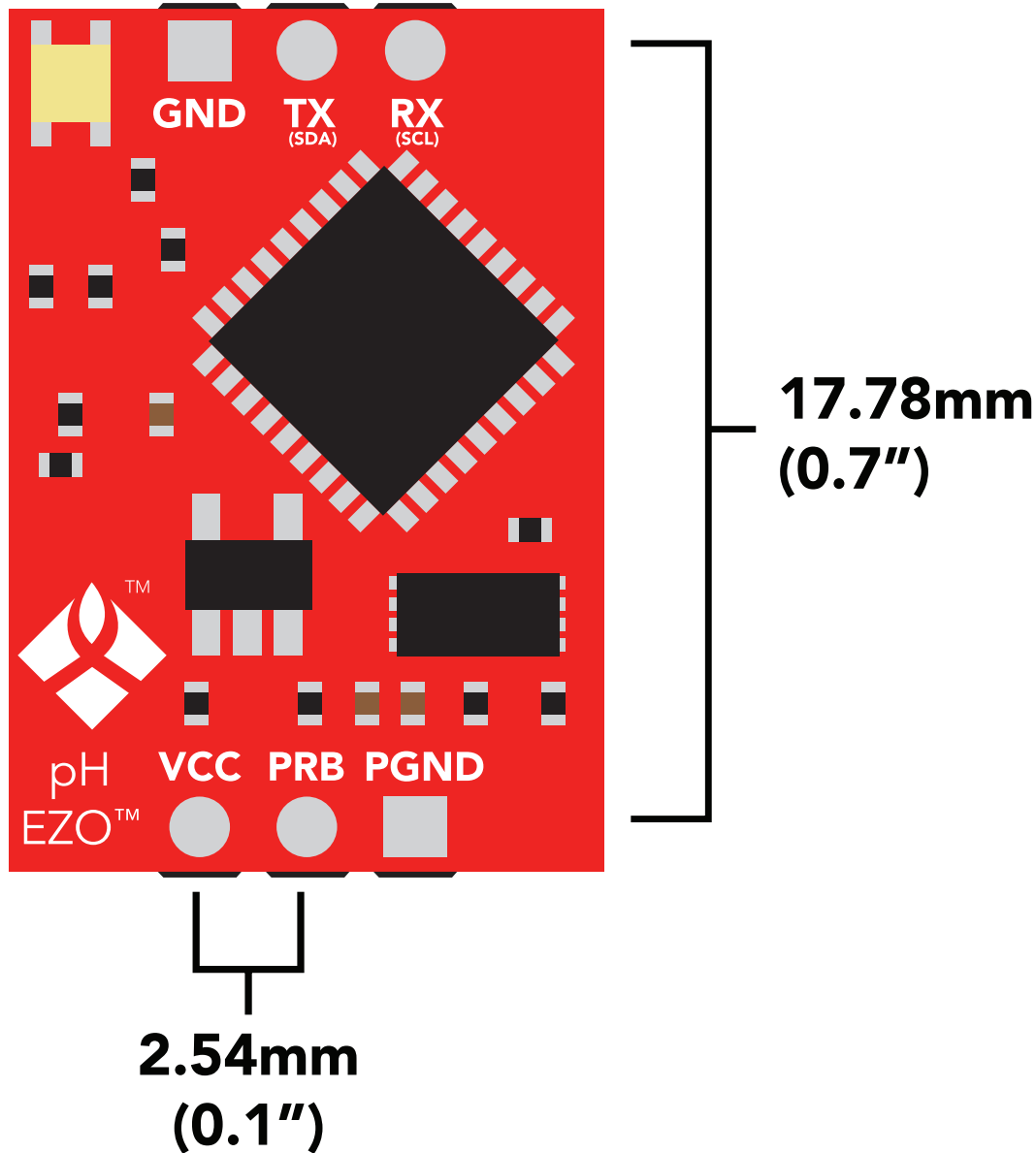
Example



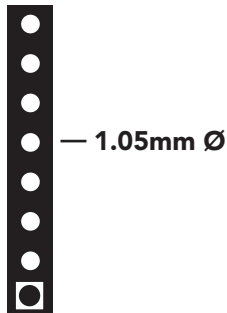
Wrong Example



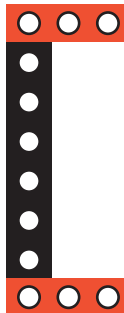
EZO™ circuit footprint



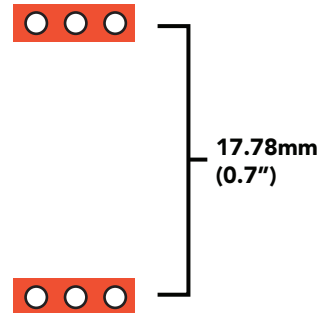
1 In your CAD software place a 8 position header.



2 Place a 3 position header at both top and bottom of the 8 position.



3 Delete the 8 position header. The two 3 position headers are now 17.78mm (0.7") apart from each other.



Datasheet change log

Datasheet V 5.6

Revised information on the slope command found on pages 29 & 54.

Datasheet V 5.5

Revised artwork within datasheet.

Datasheet V 5.4

Moved the Default state to pg 14.

Datasheet V 5.3

Revised response for the sleep command in UART mode on pg 35.

Datasheet V 5.2

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

Datasheet V 5.1

Revised isolation schematic on pg 10.

Datasheet V 5.0

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

Datasheet V 4.9

Changed "Max rate" to "Response time" on cover page.

Datasheet V 4.8

Added new command:

"RT,n" for Temperature compensation located on pages 29 (UART) & 53 (I²C).

Added firmware information to Firmware update list.

Datasheet V 4.7

Removed note from certain commands about firmware version.

Datasheet V 4.6

Added information to calibration theory on pg 7.

Datasheet V 4.5

Revised definition of response codes on pg 44.

Datasheet V 4.4

Added resolution range to cover page.

Datasheet V 4.3

Revised isolation information on pg 9.

Datasheet V 4.2

Revised Plock pages to show default value.

Datasheet V 4.1

Added new commands:

"Find" pages 23 (UART) & 46 (I²C).

"Export/Import calibration" pages 27 (UART) & 49 (I²C).

Added new feature to continuous mode "C,n" pg 24.

Datasheet V 4.0

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

Datasheet V 3.9

Revised calibration theory on pg. 7.

Datasheet V 3.8

Revised entire datasheet.

Firmware updates

V1.5 – Baud rate change (Nov 6, 2014)

- Change default baud rate to 9600

V1.6 – I²C bug (Dec 1, 2014)

- Fixed I²C bug where the circuit may inappropriately respond when other I²C devices are connected.

V1.7 – Factory (April 14, 2015)

- Changed "X" command to "Factory"

V1.95 – Plock (March 31, 2016)

- Added protocol lock feature "Plock"

V1.96 – EEPROM (April 26, 2016)

- Fixed bug 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 bug during calibration process.

V2.10 – (May 9, 2017)

- Added "Find" command.
- Added "Export/import" command.
- Modified continuous mode to be able to send readings every "n" seconds.

V2.11 – (June 12, 2017)

- Fixed "I" command to return "pH" instead of "PH".

V2.12 – (April 16, 2018)

- Fixed "cal,clear" was not clearing stored calibration in EEPROM.
- Added "RT" command to Temperature compensation.

V2.13 – (June 25, 2019)

- Added calibration offset to slope.
- Added calibration with temperature compensation.

Warranty

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

The debugging phase

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

It is important for the embedded systems engineer to keep in mind that the following activities will void the EZO™ class pH circuit warranty:

- **Soldering any part of the EZO™ class pH circuit.**
- **Running any code, that does not exclusively drive the EZO™ class pH circuit and output its data in a serial string.**
- **Embedding the EZO™ class pH circuit into a custom made device.**
- **Removing any potting compound.**

Reasoning behind this warranty

Because Atlas Scientific™ does not sell consumer electronics; once the device has been embedded into a custom made system, Atlas Scientific™ cannot possibly warranty the EZO™ class pH circuit, against the thousands of possible variables that may cause the EZO™ class pH circuit to no longer function properly.

Please keep this in mind:

- 1. All Atlas Scientific™ devices have been designed to be embedded into a custom made system by you, the embedded systems engineer.**
- 2. All Atlas Scientific™ devices have been designed to run indefinitely without failure in the field.**
- 3. All Atlas Scientific™ devices can be soldered into place, however you do so at your own risk.**

Atlas Scientific™ is simply stating that once the device is being used in your application, Atlas Scientific™ can no longer take responsibility for the EZO™ class pH circuits continued operation. This is because that would be equivalent to Atlas Scientific™ taking responsibility over the correct operation of your entire device.