

# EZO-PRS™

## Embedded Pressure Sensor

Reads	<b>Pressure (Gauge)</b> psi (0 - 74.000) <b>Default</b> atm (0 - 5.03) bar (0 - 5.102) kPa (0 - 510.212) inches of water (0 - 2,050.36") cm of water (0 - 5,202.71 cm)
Response time	<b>1 reading per second</b>
Resolution	<b>0.001</b>
Accuracy	<b>+/- 2% (1 psi)</b>
Threaded connection	<b>1/4" NPT</b>
Cable	<b>1 meter / 5 lead</b>
Data protocol	<b>UART &amp; I<sup>2</sup>C</b>
Default I <sup>2</sup> C address	<b>106 (0x6A)</b>
Data format	<b>ASCII</b>
Operating voltage	<b>3.3V – 5V</b>
Water resistant/dust proof	<b>IP67</b>



# Table of contents

EZO-PRS™ dimensions	3	Typical applications	6
Operating principle	4	Calibration theory	8
Power consumption	4	Pin out	8
Absolute max ratings	4	Default state	9
Gauge pressure vs Absolute pressure	5	Available data protocols	10

## UART

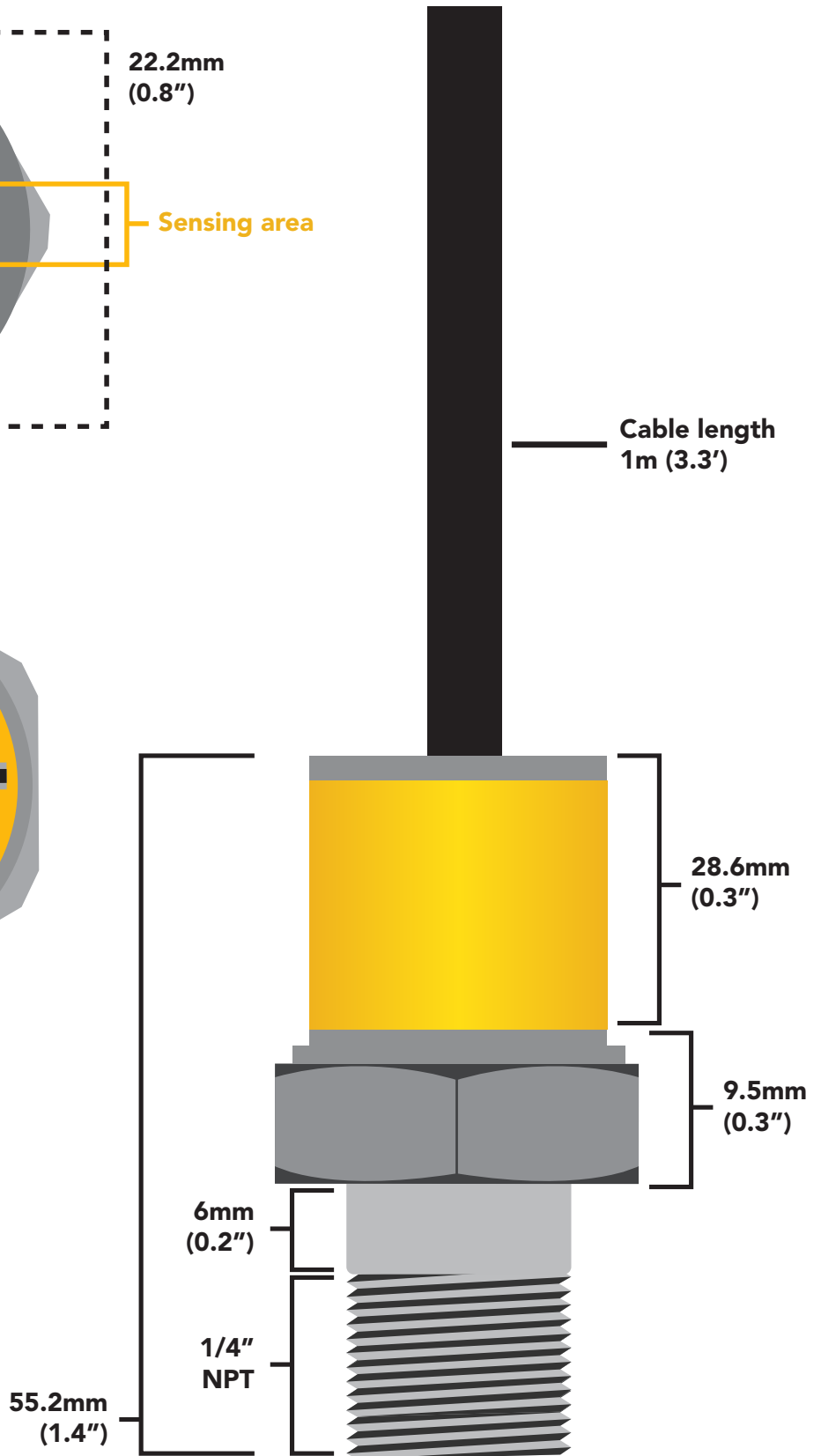
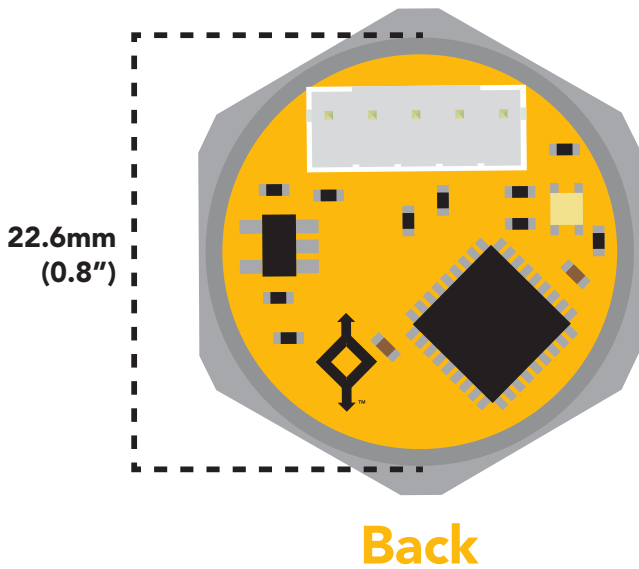
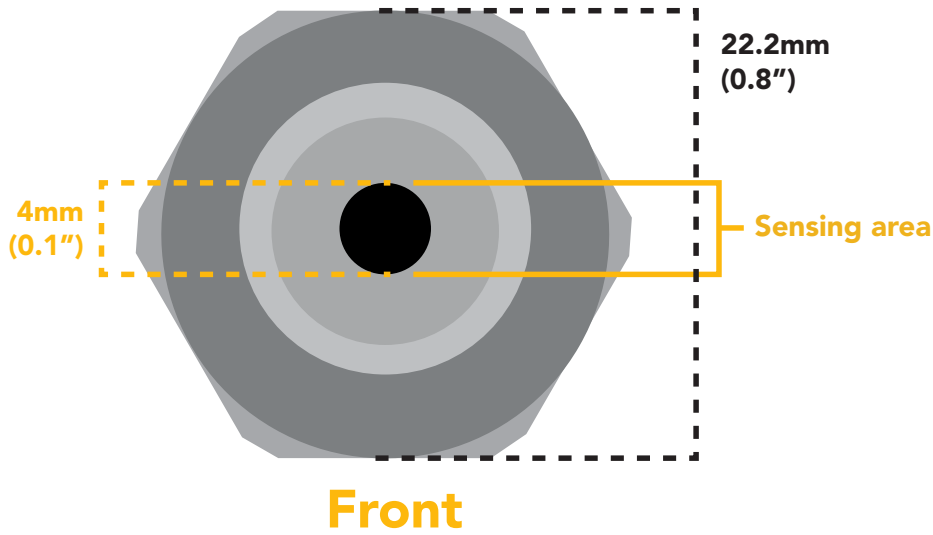
UART mode	12
Receiving data from device	13
Sending commands to device	14
LED color definition	15
<b>UART quick command page</b>	<b>16</b>
LED control	17
Find	18
Continuous mode	19
Single reading mode	20
Alarm	21
Custom calibration	22
Add/remove decimal places	23
Pressure units	24
Naming device	25
Device information	26
Response codes	27
Reading device status	28
Sleep mode/low power	29
Change baud rate	30
Protocol lock	31
Factory reset	32
Change to I <sup>2</sup> C mode	33
Manual switching to I <sup>2</sup> C	34

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

I <sup>2</sup> C mode	36
Sending commands	37
Requesting data	38
Response codes	39
LED color definition	40
<b>I<sup>2</sup>C quick command page</b>	<b>41</b>
LED control	42
Find	43
Taking reading	44
Alarm	45
Custom calibration	46
Add/remove decimal places	47
Pressure units	48
Naming device	49
Device information	50
Reading device status	51
Sleep mode/low power	52
Protocol lock	53
I <sup>2</sup> C address change	54
Factory reset	55
Change to UART mode	56
Manual switching to UART	57

Datasheet change log	58
Firmware updates	58
Warranty	59

# EZO-PRS™ dimensions



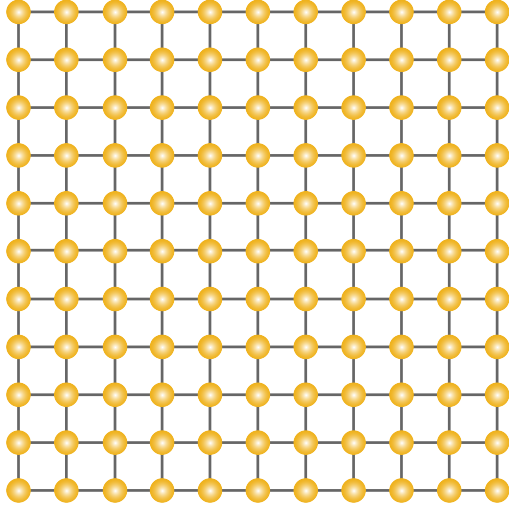
**Weight** 108g

**Body** 304L Stainless Steel

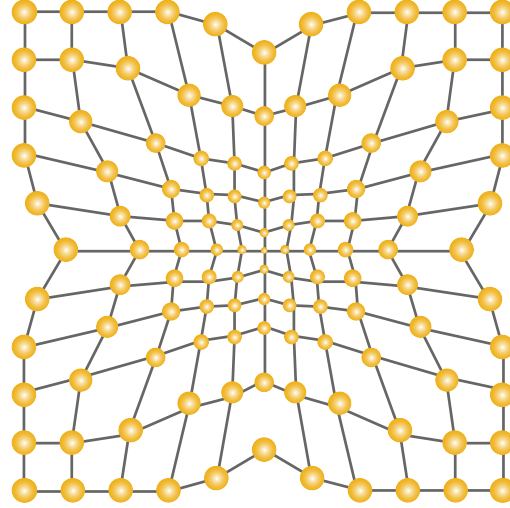
**Durability** IP67

# Operating principle

Internally the pressure sensor uses a piezoresistive semiconducting element. The semiconducting element (*a silicon wafer*) changes its resistance in proportion to pressure. As the pressure increases the atomic spacing of the silicon atoms decreases, this in turn lowers the resistance of the silicon wafer.



**Atmospheric pressure**



**50 PSI**

## Chemical compatibility

Any gas, liquid or oil compatible with 304L Stainless Steel.

## Power consumption

	LED	MAX	STANDBY	SLEEP
<b>5V</b>	ON	14.25 mA	14.25 mA	1.66 mA
	OFF	14.00 mA	14.00 mA	
<b>3.3V</b>	ON	13.21 mA	13.21 mA	0.85 mA
	OFF	12.95 mA	12.95 mA	

## Absolute max ratings

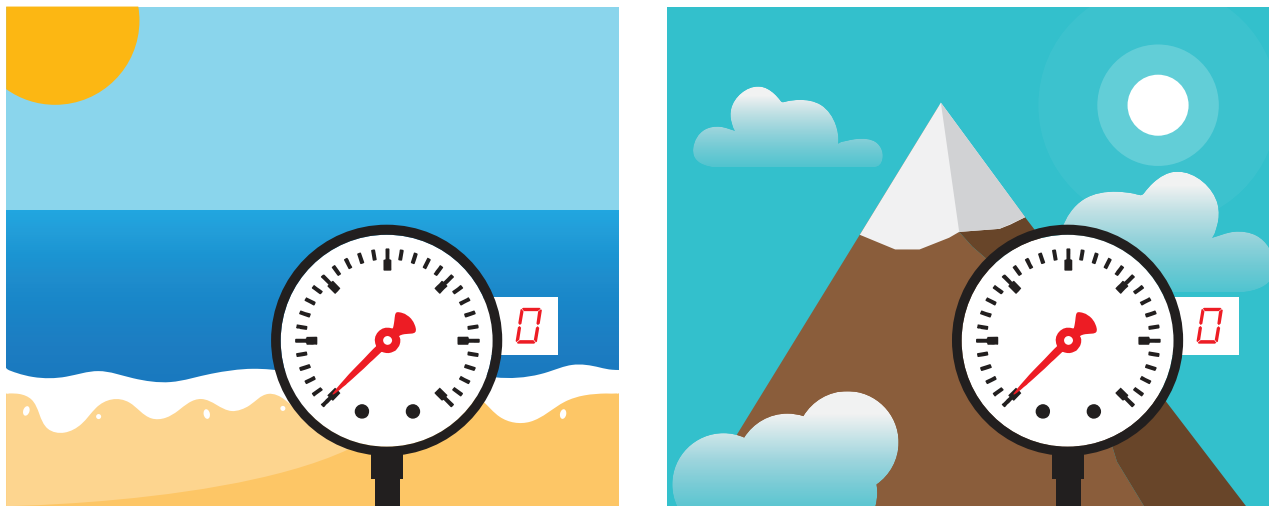
Parameter	MIN	TYP	MAX
Storage temperature (EZO-Pressure)	-65 °C		125 °C
Operational temperature (EZO-Pressure)	-40 °C	25 °C	105 °C
VCC	3.3V	5V	5.5V
Pressure limit (sensor damage)			~150 psi
Burst Pressure			7,500 psi

# Gauge pressure vs Absolute pressure

The EZO-PRS™ reads ***gauge pressure*** only.

## Gauge pressure

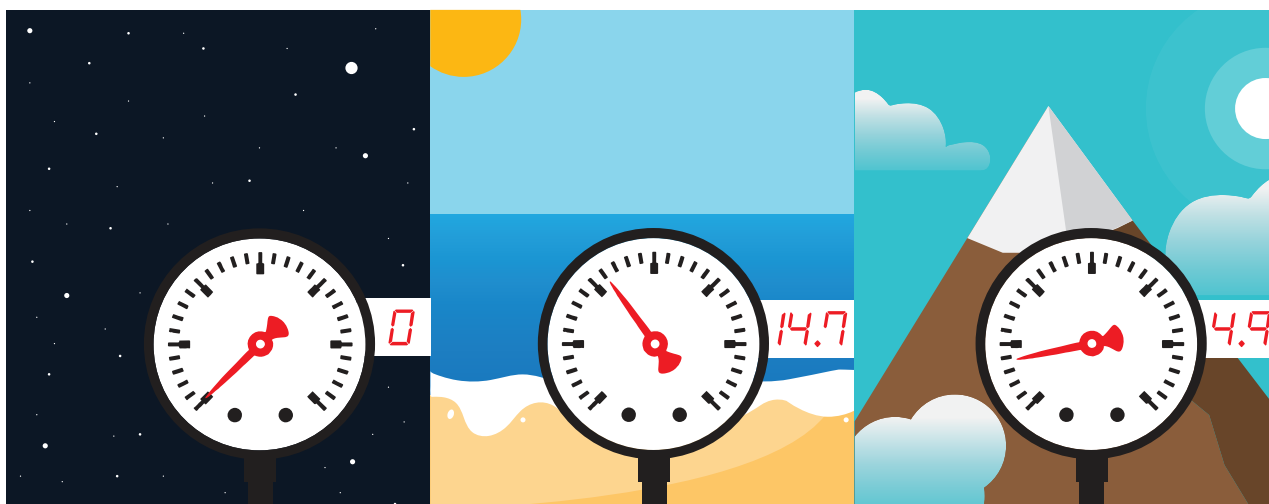
A gauge pressure sensor reads pressure relative to atmospheric pressure.



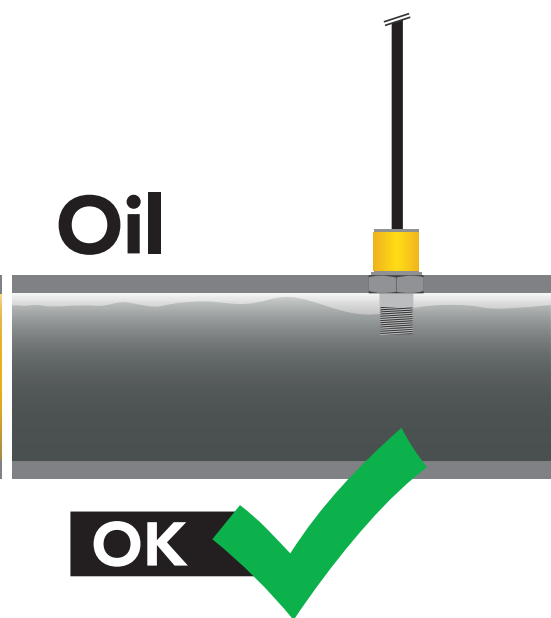
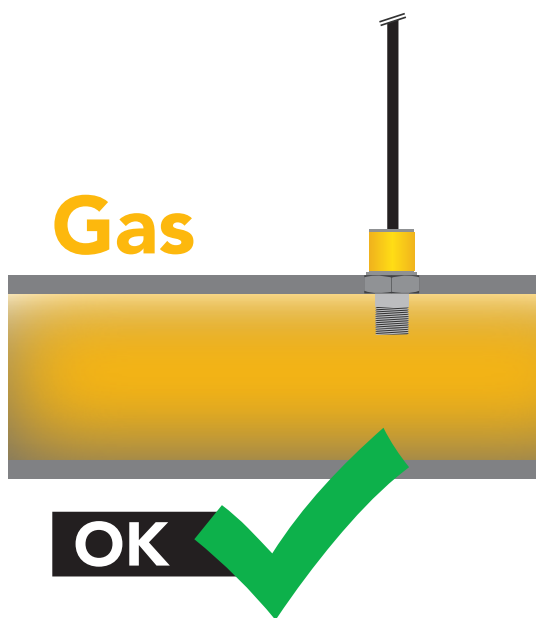
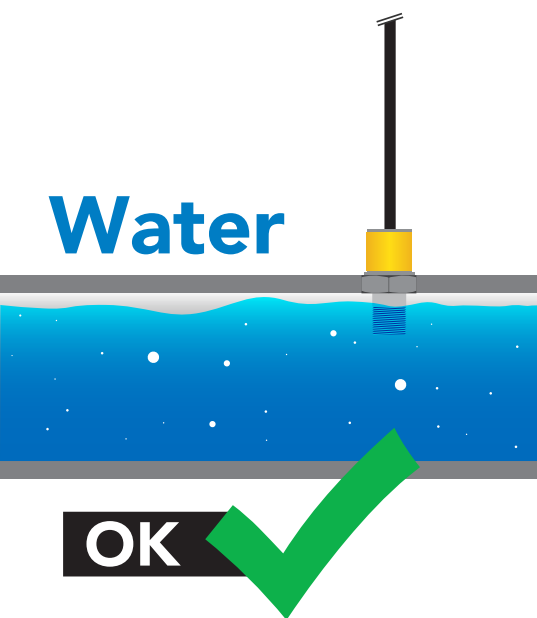
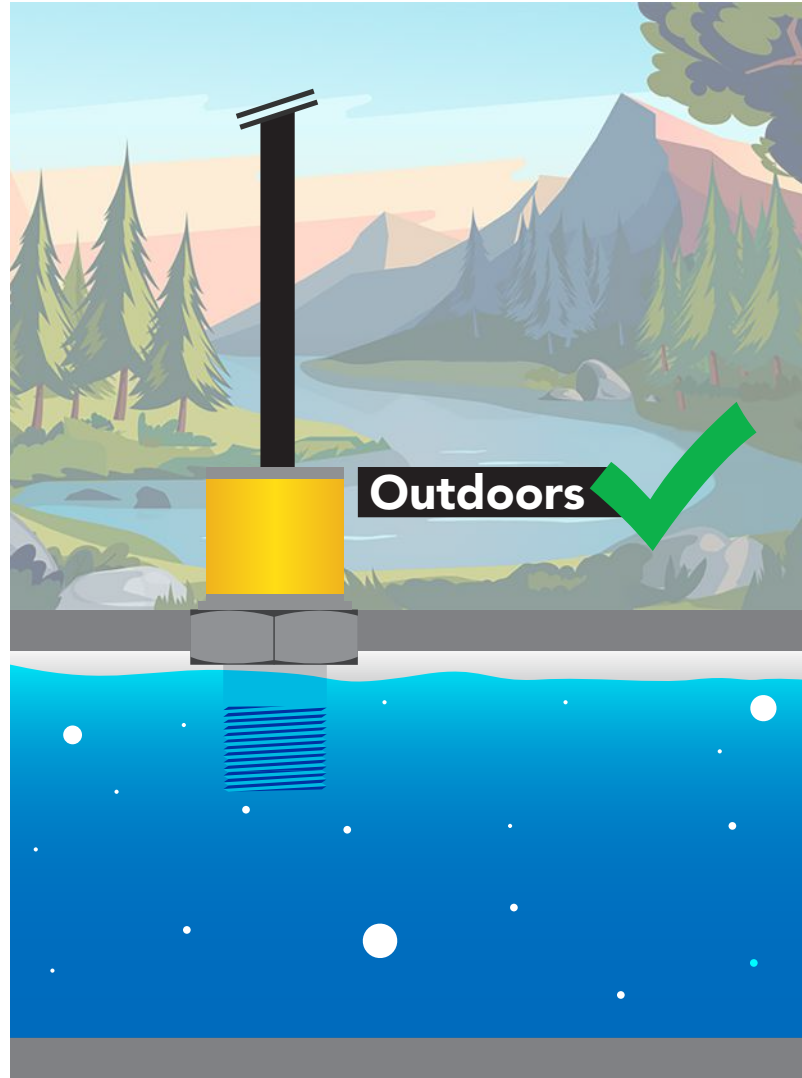
A gauge pressure sensor will always read atmospheric pressure as 0.

## Absolute pressure

An absolute pressure sensor reads pressure relative to the vacuum of space.



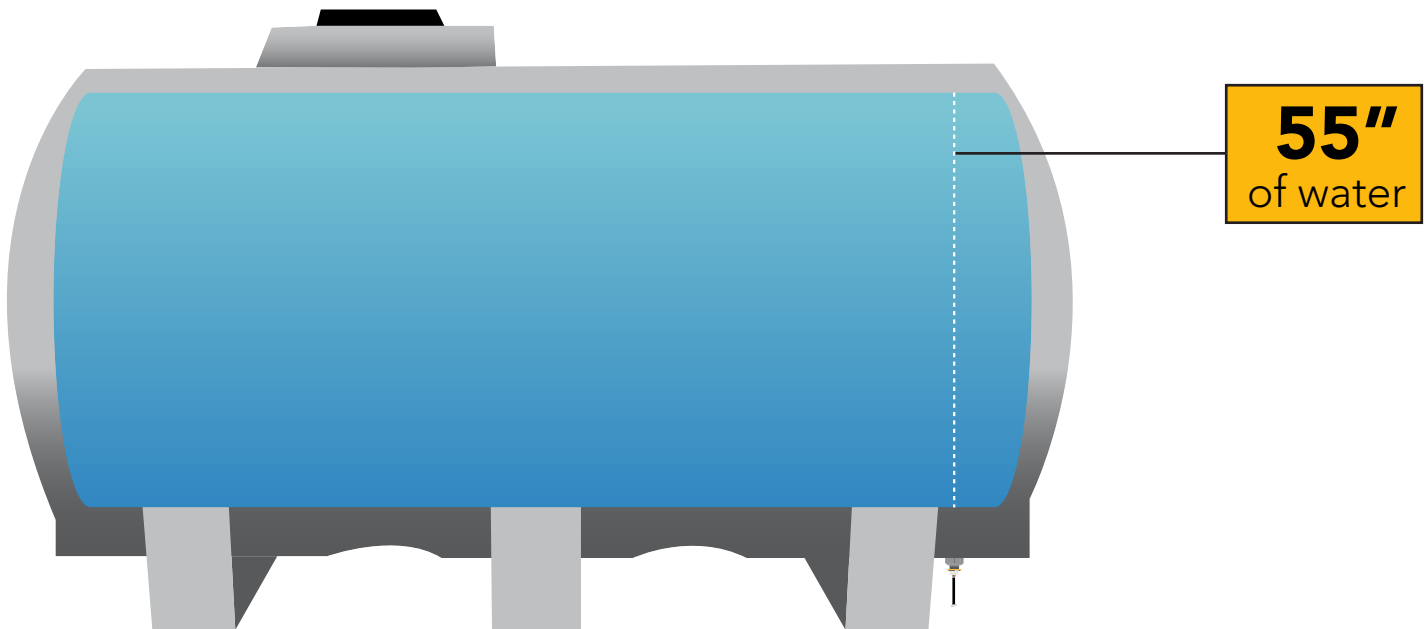
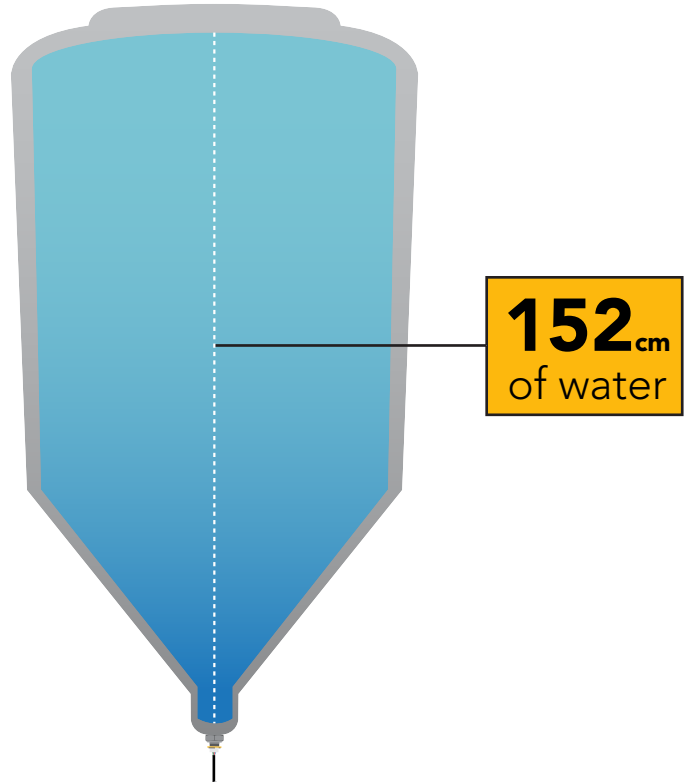
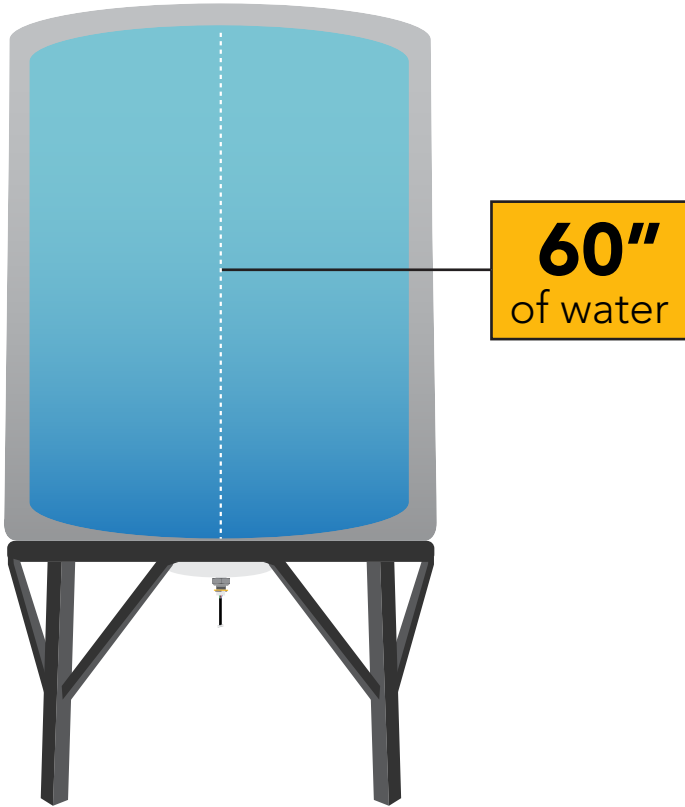
# Typical applications



# Typical applications

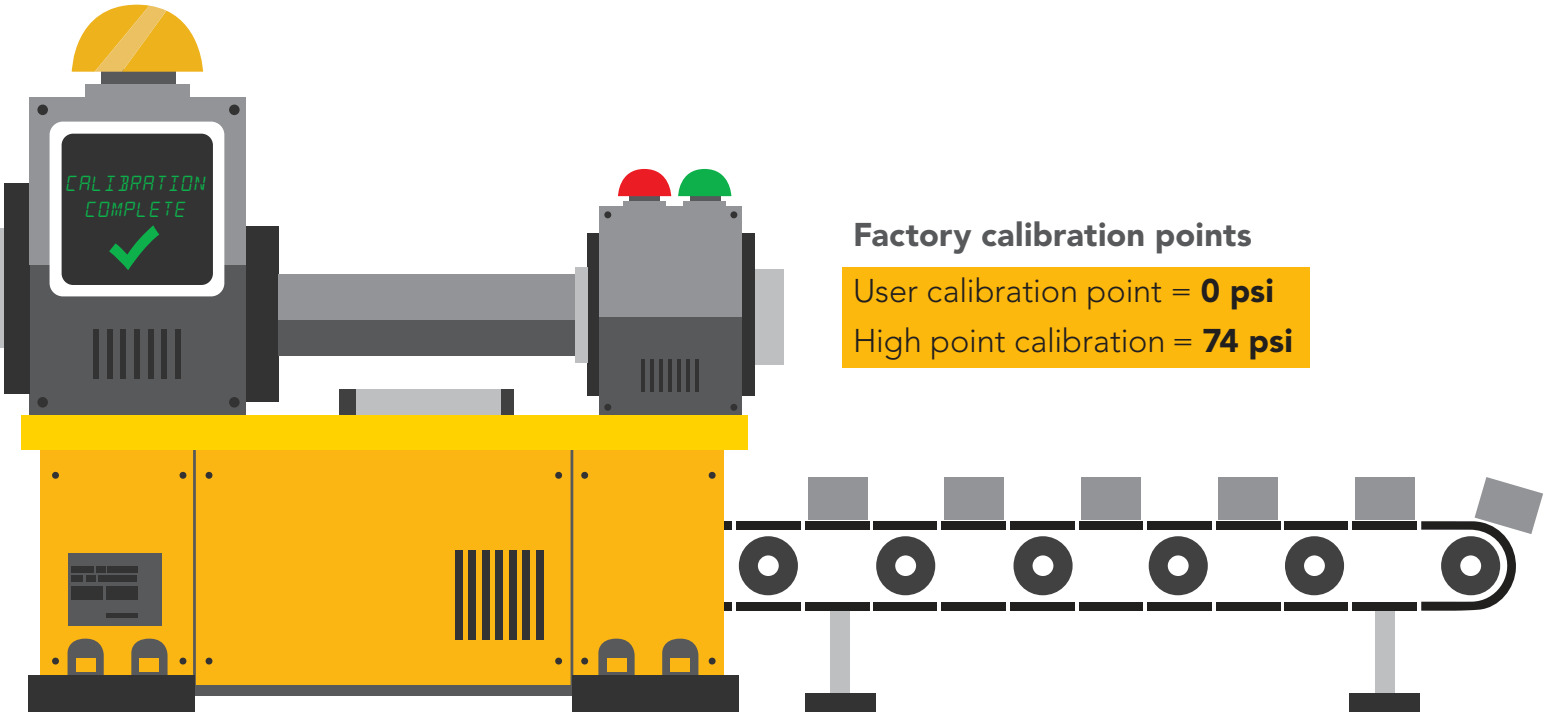
## Measuring the water level in a tank

Setting the EZO-PRS™ to measure the height of water is a great way to measure the volume of a tank. See page [24](#) or [48](#) for more info.



# Calibration theory

The Atlas Scientific EZO-PRS™ Embedded Pressure Sensor comes half-calibrated. When using the sensor for the first time, it is common to see a small pressure reading even though it is not connected to a pressure source. Issue the "Cal,0" command to complete the calibration process.



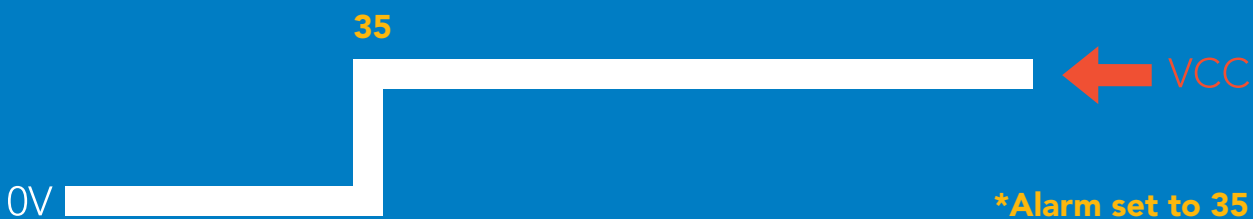
## Pin out

### Data and power cable pinout

White	-	<b>RX/SCL</b>
Green	-	<b>TX/SDA</b>
Black	-	<b>GND</b>
Red	-	<b>VCC</b>
Blue	-	<b>ALM</b>



The interrupt pin will go high when a set pressure level has been crossed.



If unused leave **ALM** floating. Do not connect **ALM** to **VCC** or **GND**.

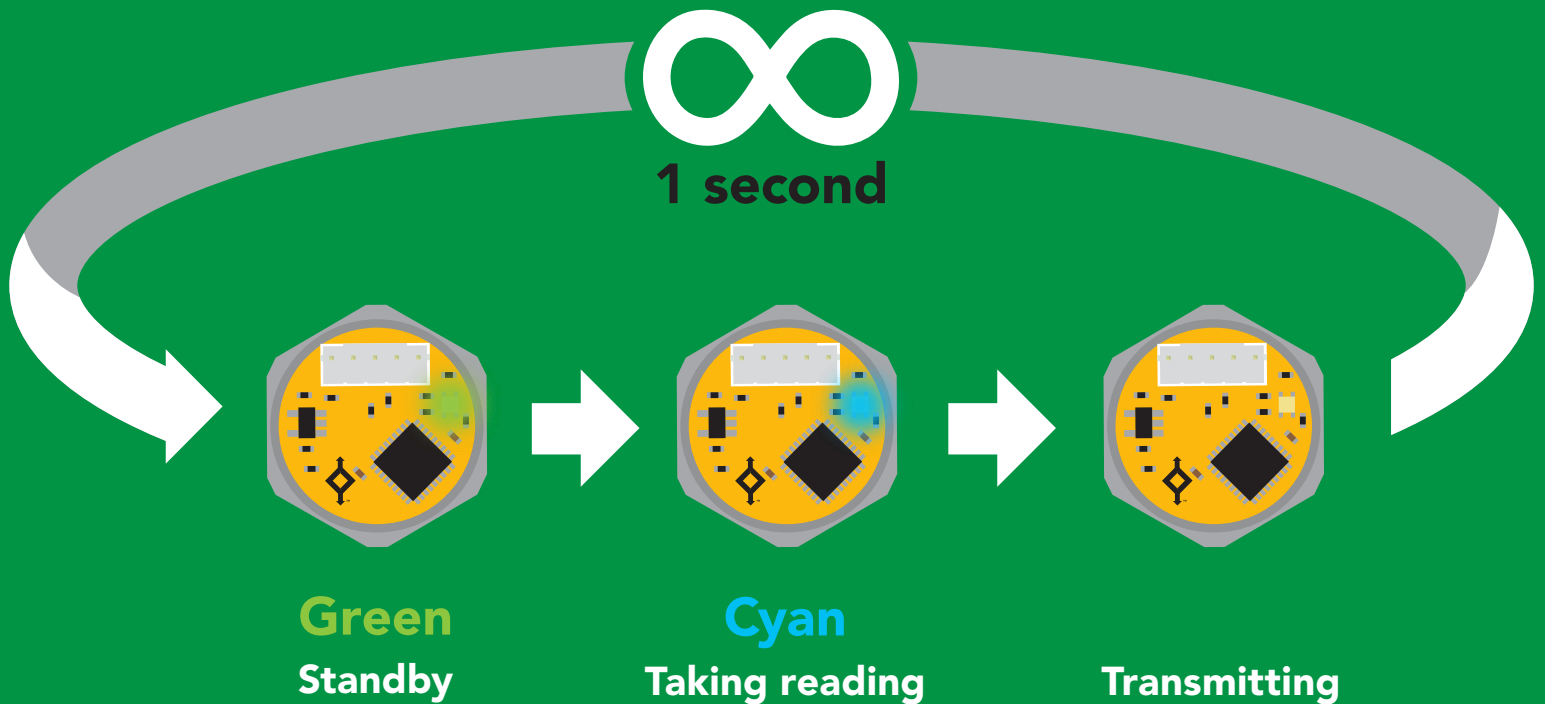
See page **21** or **45** to enable pressure level alarm.



# Default state

# UART mode

<b>Baud</b>	9,600
<b>Readings</b>	continuous
<b>Units</b>	PSI
<b>Speed</b>	1 reading per second
<b>LED</b>	on



# ✓ Available data protocols

# UART

Default

# I<sup>2</sup>C

# X 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
- Custom calibration
- 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

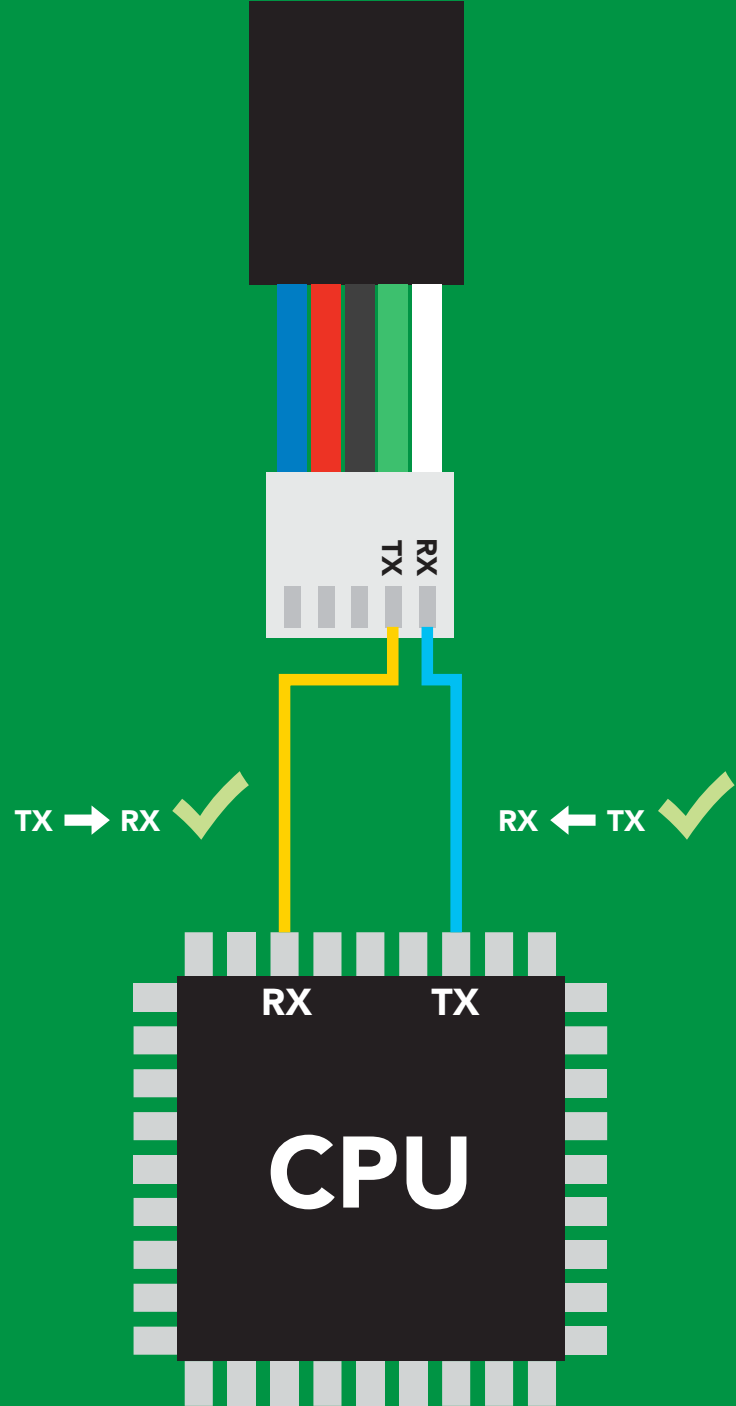
# UART mode

8 data bits      no parity  
1 stop bit        no flow control

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



**Vcc** 3.3V – 5.5V



## Data format

**Output**      pressure  
**Units**        **PSI default**  
                  (ATM, kPa, bar,  
                  inch's of water  
                  cm of water)  
**Encoding**    **ASCII**  
**Format**        **string**

**Terminator**    carriage return  
**Data type**     floating point  
**Decimal places** 3  
**Smallest string** 3 characters  
**Largest string** 39 characters

# Receiving data from device

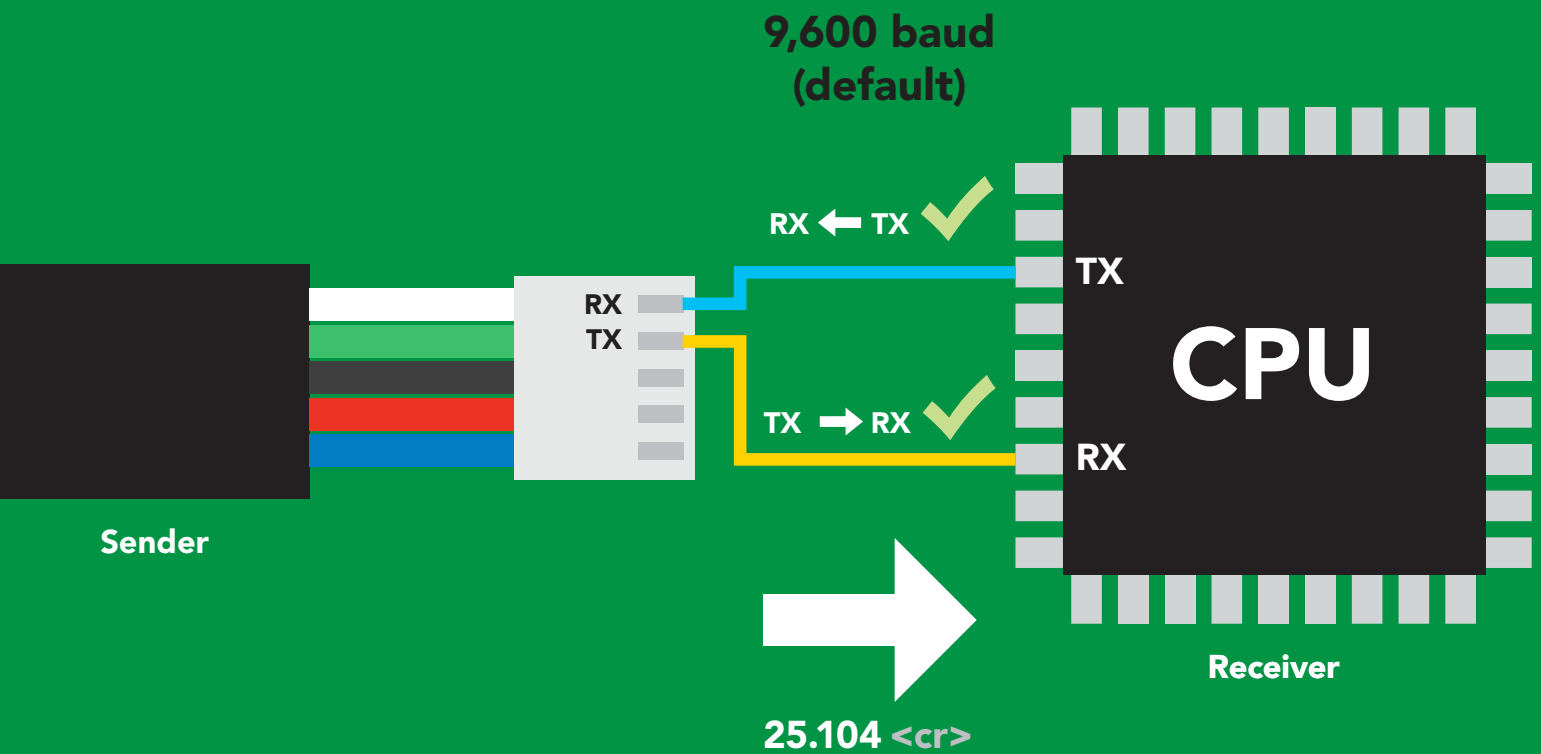
2 parts

ASCII data string

Command

Carriage return <cr>

Terminator



## Advanced

ASCII: 2 5 . 1 0 4 <cr>

Hex: 32 35 2E 31 30 34 0D

Dec: 50 53 46 49 48 52 13

# Sending commands to device

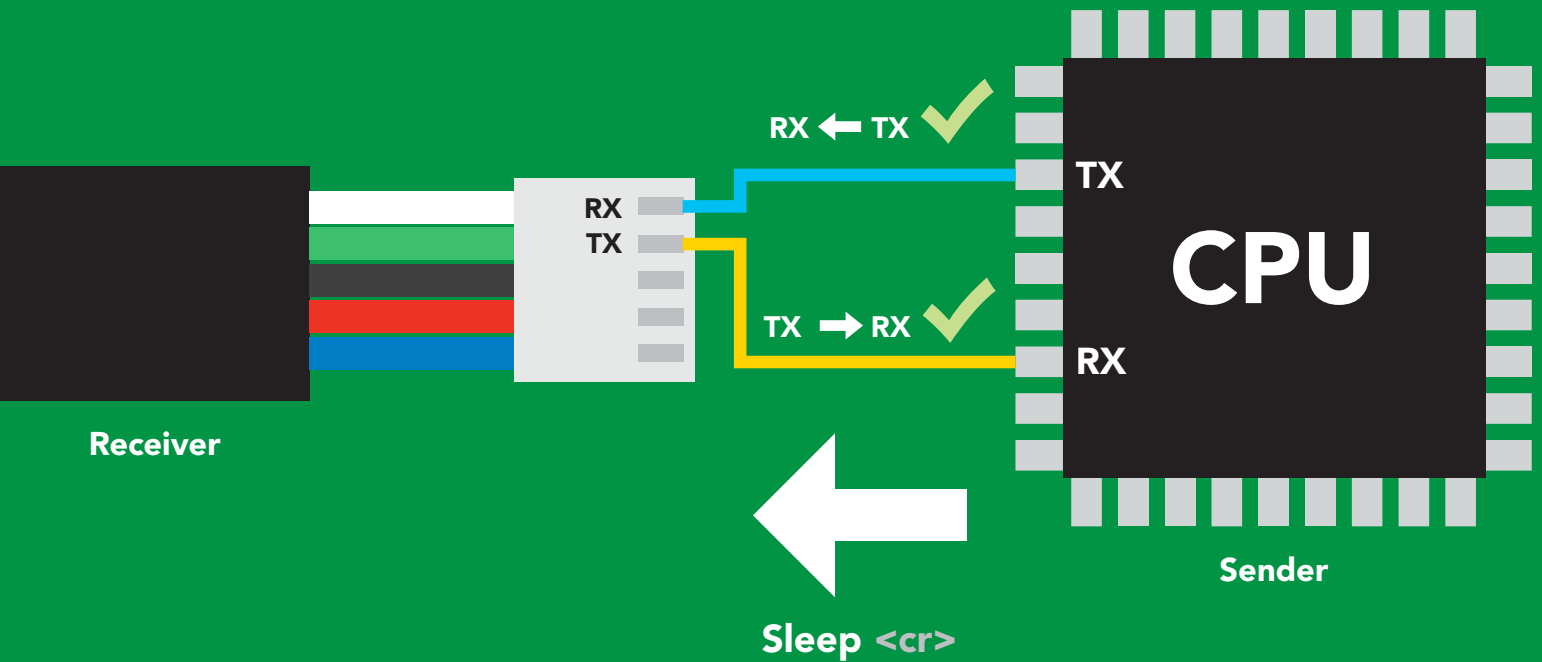
2 parts

**Command (not case sensitive)**

ASCII data string

**Carriage return <cr>**

Terminator



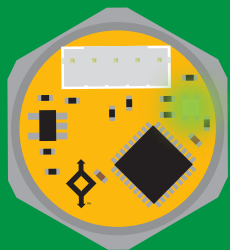
## Advanced

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

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

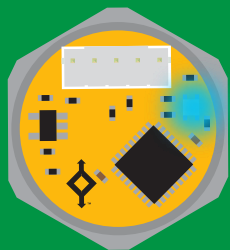
Dec: **83** **108** **101** **101** **112** **13**

# LED color definition



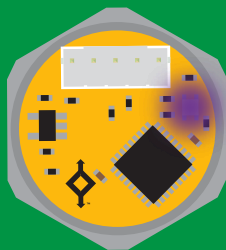
**Green**

UART standby



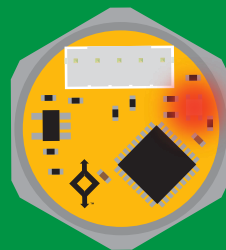
**Cyan**

Taking reading



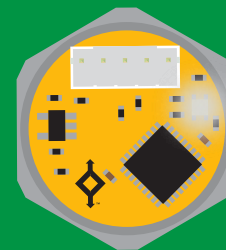
**Purple**

Changing  
baud rate



**Red**

Command  
not understood



**White**

Find

**5V**

LED ON  
**+0.25 mA**

**3.3V**

**+0.26 mA**

# UART mode

## command quick reference

All commands are ASCII strings or single ASCII characters.

Command	Function		Default state
Alarm	enable/disable alarm	pg. 21	n/a
Baud	change baud rate	pg. 30	9,600
C	enable/disable continuous mode	pg. 19	enabled
Cal	performs custom calibration	pg. 22	n/a
Dec	add/remove decimal places	pg. 23	n/a
Factory	enable factory reset	pg. 32	n/a
Find	finds device with blinking white LED	pg. 18	n/a
i	device information	pg. 26	n/a
I2C	change to I <sup>2</sup> C mode	pg. 33	not set
L	enable/disable LED	pg. 17	enabled
Name	set/show name of device	pg. 25	not set
Plock	enable/disable protocol lock	pg. 31	disabled
R	returns a single reading	pg. 20	n/a
Sleep	enter sleep mode/low power	pg. 29	n/a
Status	retrieve status information	pg. 28	enable
U	pressure units	pg. 24	psi
*OK	enable/disable response codes	pg. 27	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>

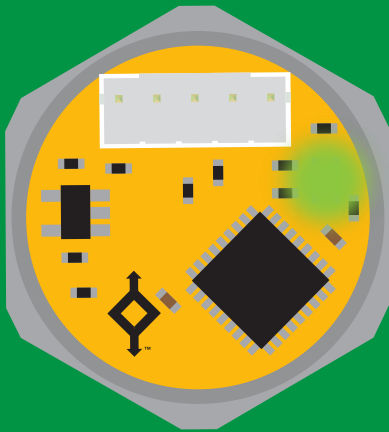
L,0 <cr>

\*OK <cr>

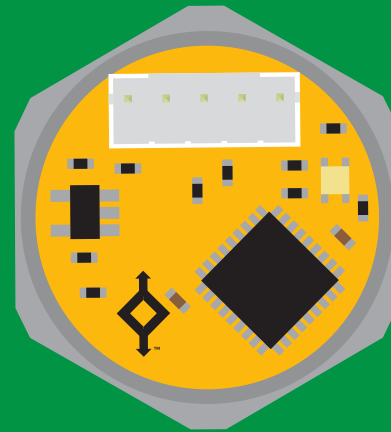
L,? <cr>

?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.

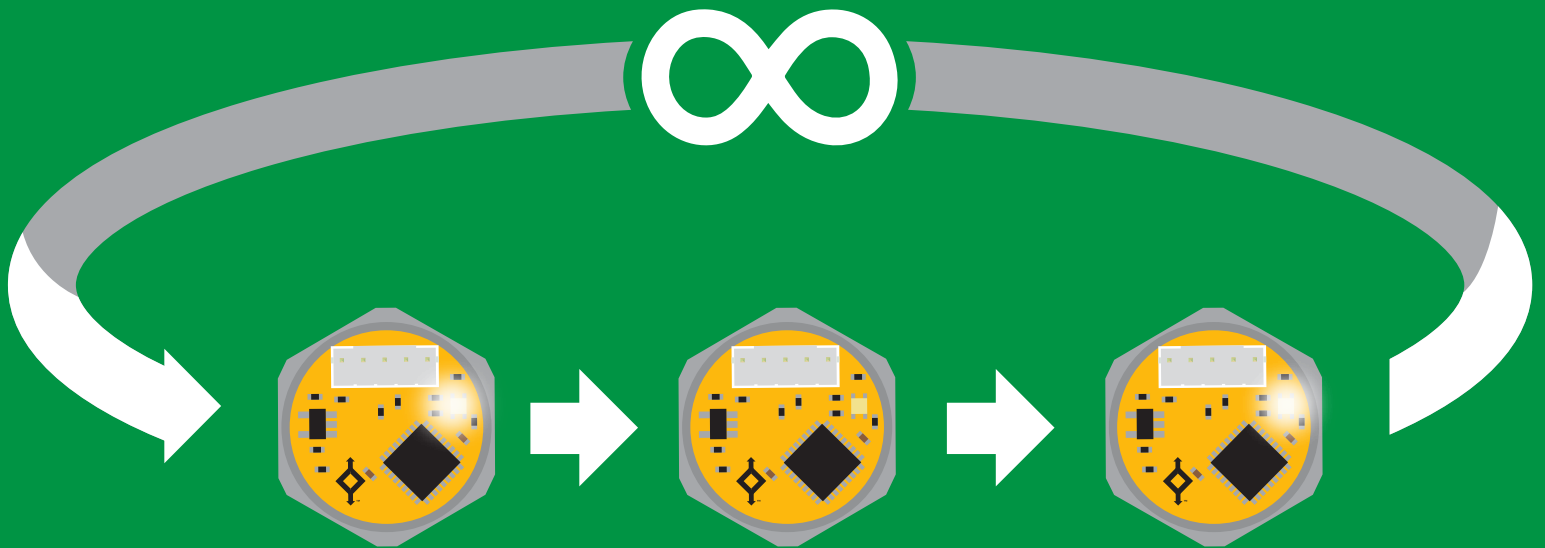
Find <cr> LED rapidly blinks white, used to help find device

## Example

## Response

Find <cr>

\*OK <cr>



# Continuous 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>**  
**pressure (1 sec) <cr>**  
**pressure (2 sec) <cr>**  
**pressure (n sec) <cr>**

**C,30 <cr>**

**\*OK <cr>**  
**pressure (30 sec) <cr>**  
**pressure (60 sec) <cr>**  
**pressure (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

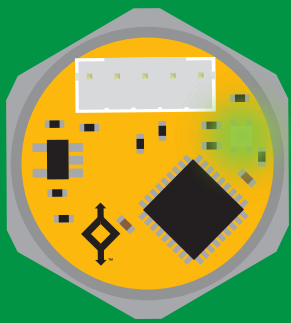
### Example

**R** <cr>

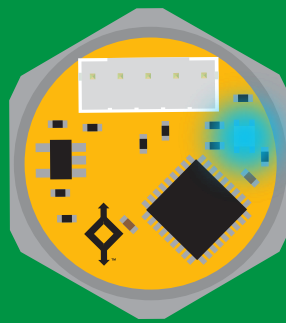
### Response

**38.462** <cr>

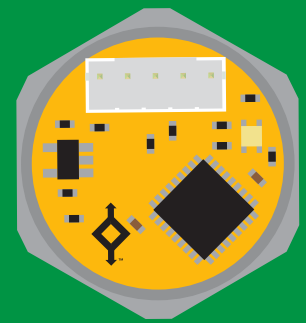
**\*OK** <cr>



**Green**  
Standby



**Cyan**  
Taking reading



**Transmitting**



**1 second**

# Alarm

## Command syntax

The alarm pin will = 1 when pressure levels are > alarm set point. Alarm tolerance sets how far below the set point pressure levels need to drop before the pin will = 0 again.

**Alarm,en,[1,0]** <cr> **enable / disable alarm**

**Alarm,n** <cr> **sets alarm**

**Alarm,tol,n** <cr> **sets alarm tolerance**

**Alarm,?** <cr> **alarm set?**

## Example

## Response

**Alarm,en,1** <cr>

**\*OK** <cr> **Enable alarm**

**Alarm,35** <cr>

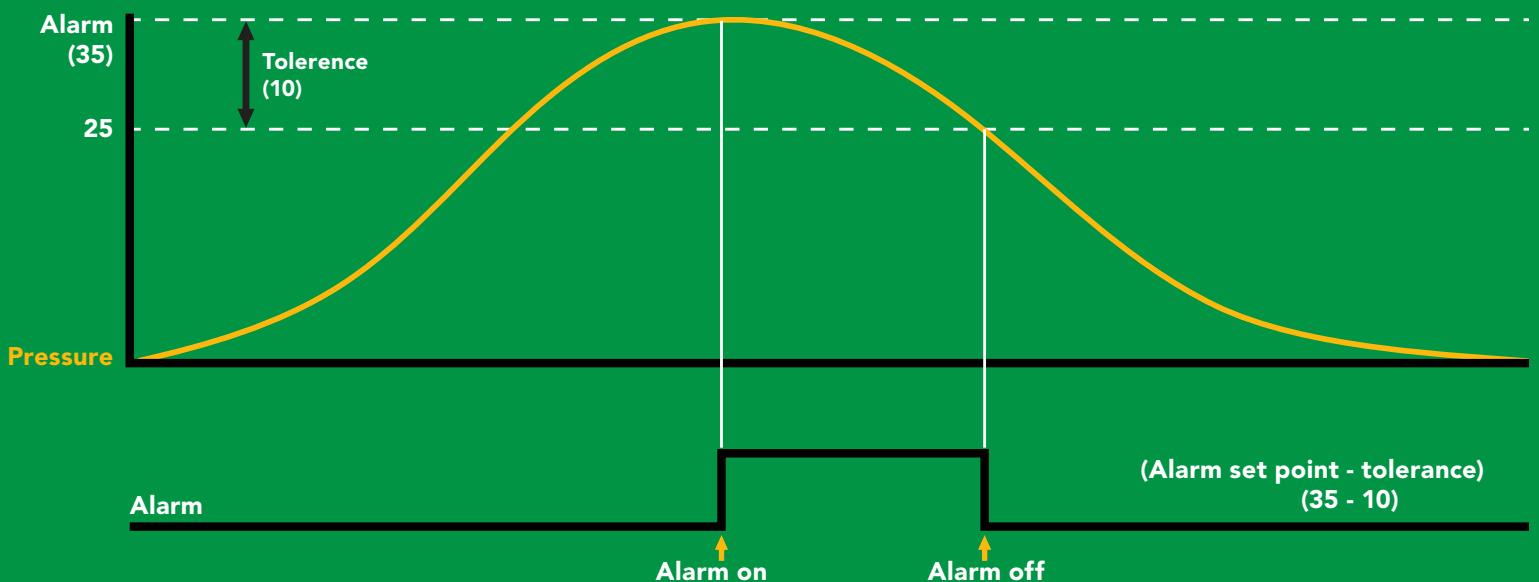
**\*OK** <cr>

**Alarm,tol,10** <cr>

**\*OK** <cr> **Pressure level must fall 10 units below set point for alarm to reset.**

**Alarm,?** <cr>

**?,alarm,35,10,1** <cr> **if all are enabled**



# Custom calibration

## Command syntax

Although calibration is not required, it may be necessary to adjust your 0 point or perform a custom calibration.

Cal,n	<cr>	calibrates the high point
Cal,0	<cr>	calibrates the zero point
Cal,clear	<cr>	restores calibration to factory settings
Cal,?	<cr>	device calibrated?

## Example

## Response

Cal,50 <cr>

\*OK <cr> high point calibration in psi

Cal,0 <cr>

\*OK <cr> low point calibration in psi

Cal,clear <cr>

\*OK <cr>

Cal,? <cr>

?Cal,0 <cr> or ?Cal,1 <cr> or ?Cal,2 <cr> or  
no calibration                      only zero point calibration                      only high point calibration

?Cal,3 <cr> \*OK <cr>  
zero and high point calibration

Calibration should be done using the pressure scale you have set the sensor to.

### Example

Readings are set to bar.  
High point calibration = 3.44  
(3.44 bar = 50 psi)

# Add/remove decimal places

## Command syntax

Change how many decimal points the reading outputs.

**Dec,n <cr>** n = number of decimal points between 0 and 3

**Dec,? <cr>** number of decimal points the output is set to

## Example

## Response

**R <cr>**

**38.462 <cr>**

**Dec,1 <cr>**

**\*OK <cr>**

**R <cr>**

**38.4 <cr>**

**Dec,? <cr>**

**?Dec,1 <cr>**

# Pressure units

(psi, atm, bar, kPa, inch's of water, cm of water)

## Command syntax

U,[1/0]	<cr>	1 will add a unit identifier to the output	
U,psi	<cr>	output will be in psi	<b>default</b>
U,atm	<cr>	output will be in atm	
U,bar	<cr>	output will be in bar	
U,kPa	<cr>	output will be in kPa	
U,inh2o	<cr>	output will be in inches of water	(Resolution: 0.027")
U,cmh2o	<cr>	output will be in cm of water	(Resolution: 0.7mm)
U,?	<cr>	pressure units?	

## Example

## Response

U,bar <cr>

\*OK <cr>

U,1 <cr>

\*OK <cr>

1.228,bar <cr>

U,? <cr>

?U,bar <cr>



# Naming device

## Command syntax

Do not use spaces in the name

Name,n <cr> set name

Name, <cr> clears 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

## Response

Name, <cr>

\*OK <cr> name has been cleared

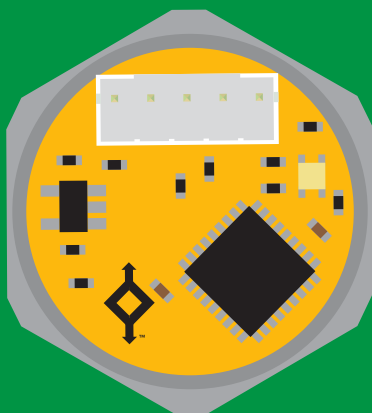
Name,zzt <cr>

\*OK <cr>

Name,? <cr>

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

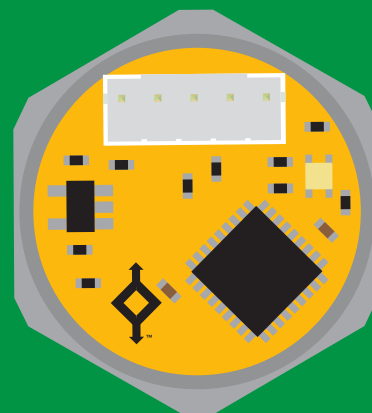
Name,zzt



\*OK <cr>



Name,?



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

# Device information

## Command syntax

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

### Example

```
i <cr>
```

### Response

```
?i,PRS,1.0 <cr>  
*OK <cr>
```

## Response breakdown

?i,	PRS	1.0
	↑	↑
	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>

**38.462** <cr>  
**\*OK** <cr>

**\*OK,0** <cr>

no response, **\*OK** disabled

**R** <cr>

**38.462** <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

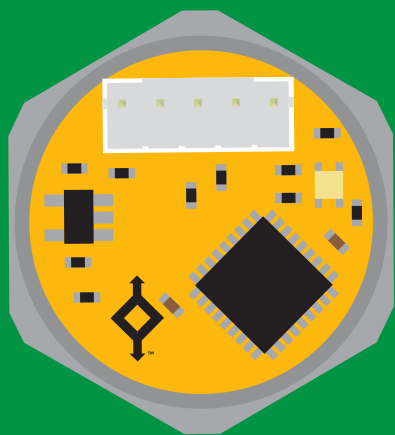
\*WA <cr> wakes up device

5V

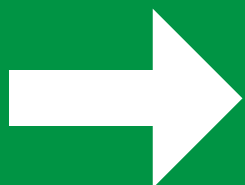
STANDBY	SLEEP
13.4 mA	0.415 mA

3.3V

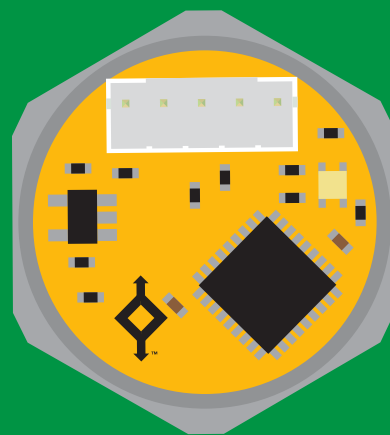
12.4 mA	0.13 mA
---------	---------



Standby  
13.4 mA



Sleep <cr>



Sleep  
0.415 mA

# Change baud rate

## Command syntax

Baud,n <cr> change baud rate

### Example

Baud,38400 <cr>

### Response

\*OK <cr>

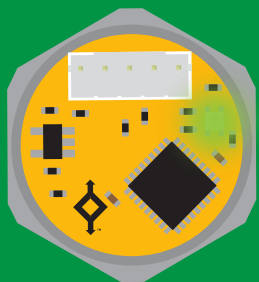
Baud,? <cr>

?Baud,38400 <cr>

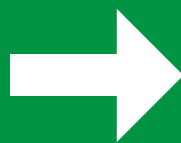
\*OK <cr>

n =

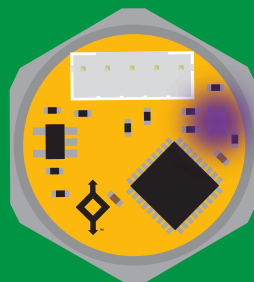
- 300
- 1200
- 2400
- 9600 default**
- 19200
- 38400
- 57600
- 115200



Standby

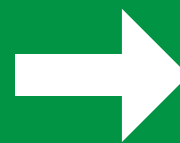


Baud,38400 <cr>

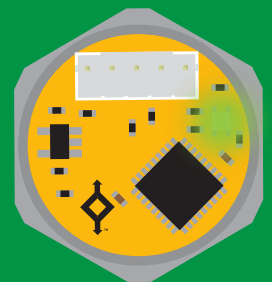


Changing  
baud rate

\*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>

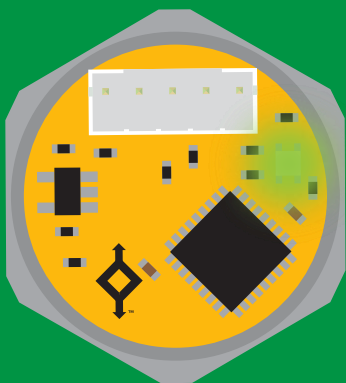
**Plock,0** <cr>

**\*OK** <cr>

**Plock,?** <cr>

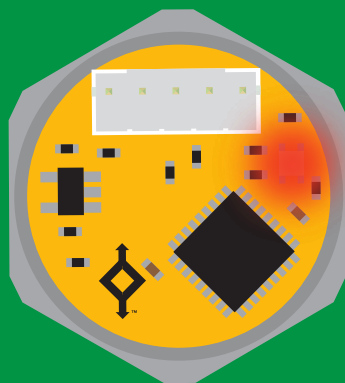
**?Plock,1** <cr> **or** **?Plock,0** <cr>

**Plock,1**

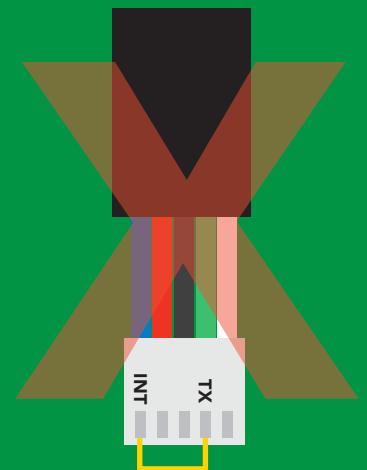


**\*OK** <cr>

**I2C,100**



cannot change to 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

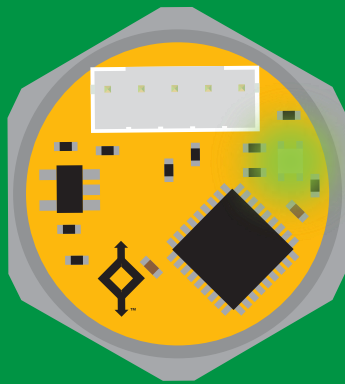
### Example

### Response

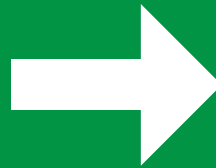
Factory <cr>

\*OK <cr>

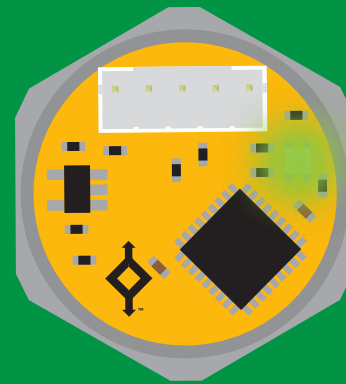
Factory <cr>



\*OK <cr>



(reboot)



\*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 106 (0x6A)

I2C,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

I2C,100 <cr>

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

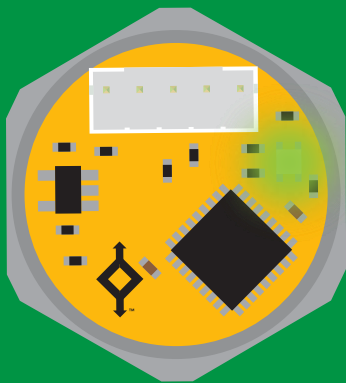
### Wrong example

### Response

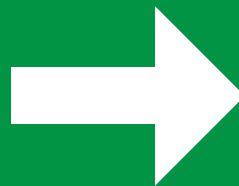
I2C,139 <cr> n ≠ 127

\*ER <cr>

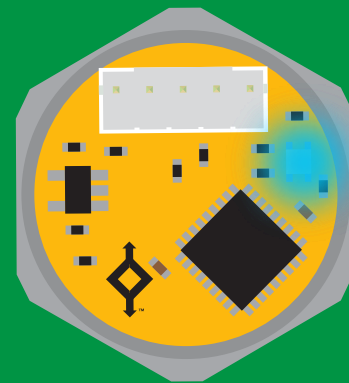
I2C,100



Green  
\*OK <cr>



(reboot)



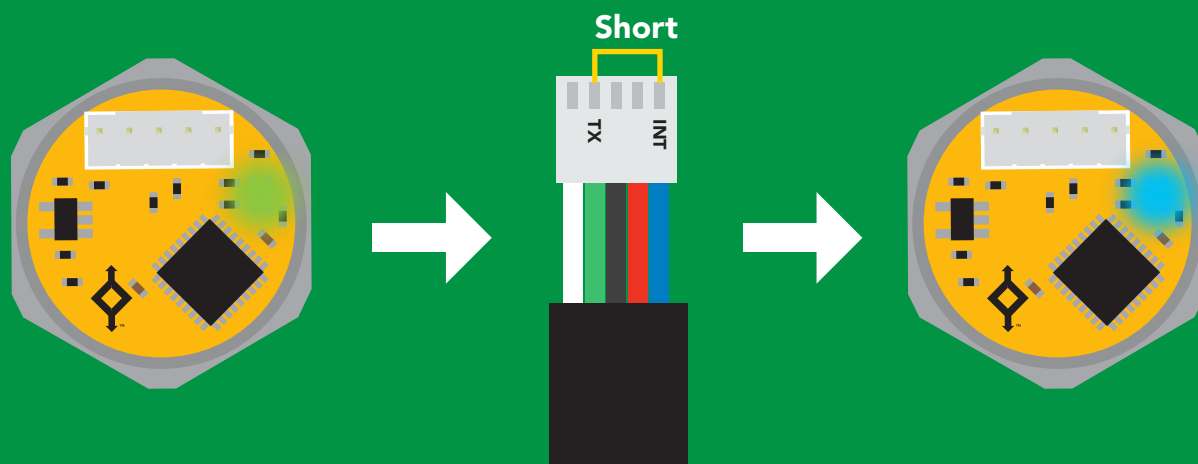
Blue  
now in I<sup>2</sup>C mode

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

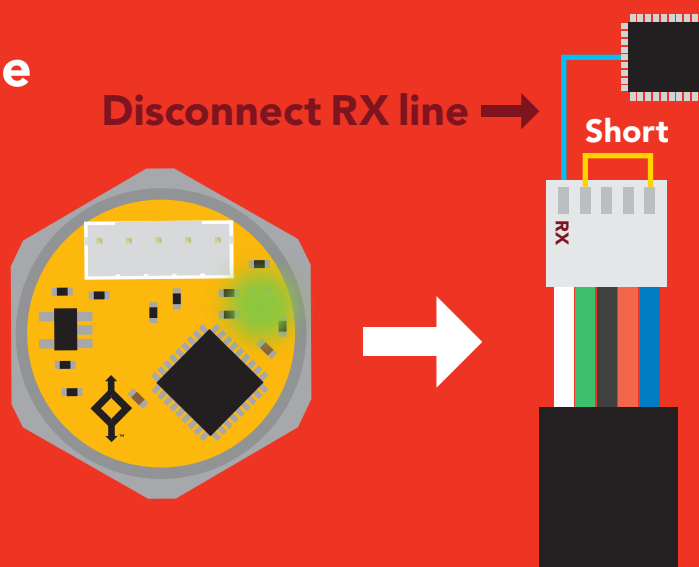
- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to INT
- Confirm RX is disconnected
- Connect ground (power on)
- Wait for LED to change from **Green to Blue**
- Disconnect ground (power off)
- Reconnect all data and power

Manually switching to I<sup>2</sup>C will set the I<sup>2</sup>C address to 106 (0x6A)

## 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-PRS™ 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
- Custom calibration
- 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)  
**106 (0x6A) default**

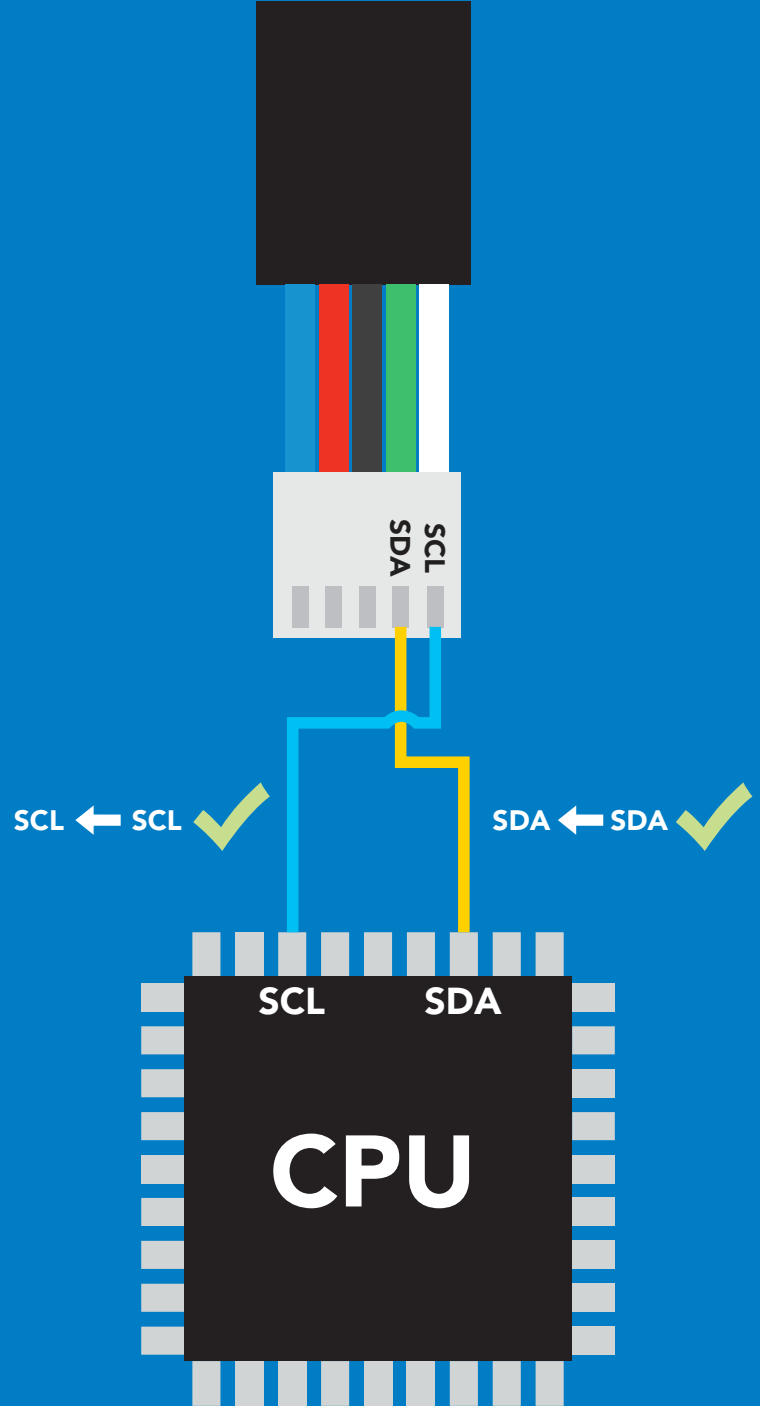
Vcc 3.3V – 5.5V

Clock speed 100 – 400 kHz

SDA 

SCL 





## Data format

Output **pressure**

Units **PSI default**  
(ATM, kPa, bar,  
inch's of water  
cm of water)

Encoding **ASCII**  
Format **string**

Data type **floating point**

Decimal places **3**

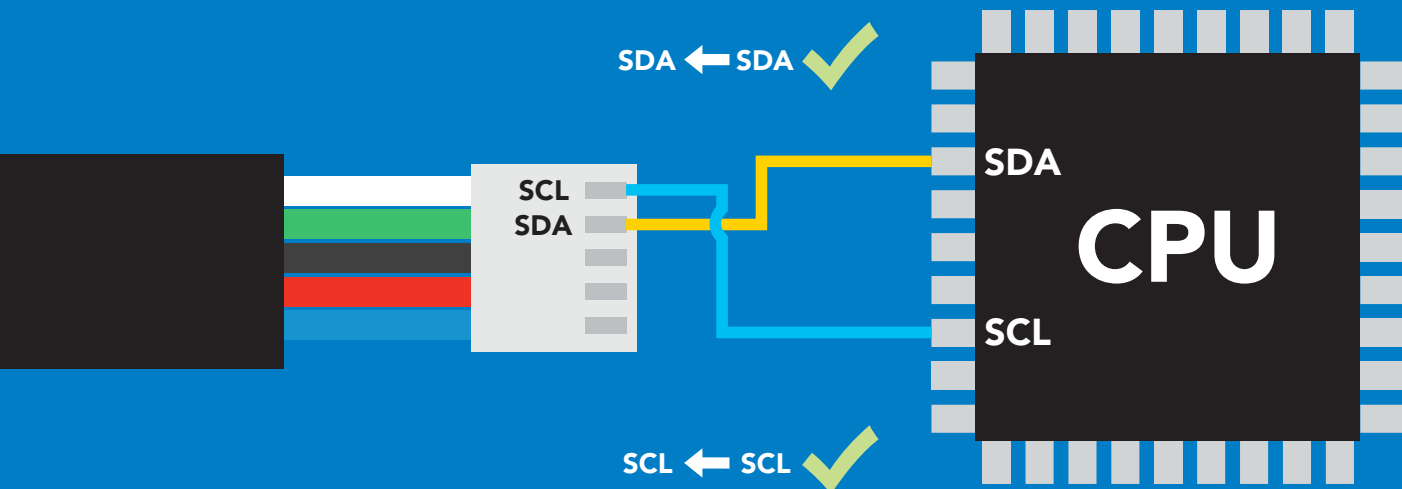
Smallest string **3 characters**

Largest string **39 characters**

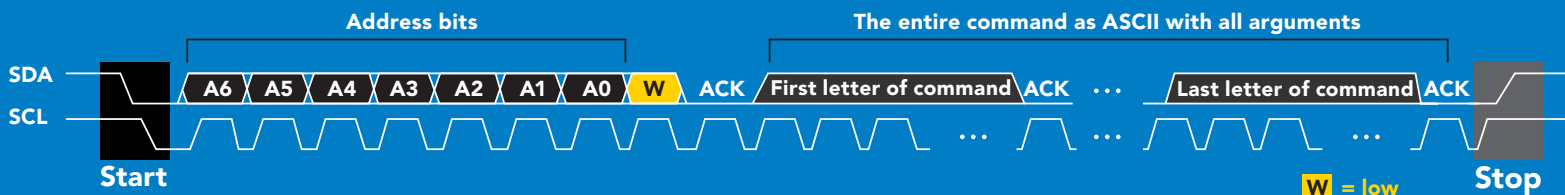
# Sending commands to device



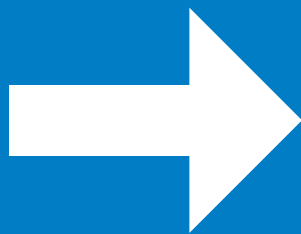
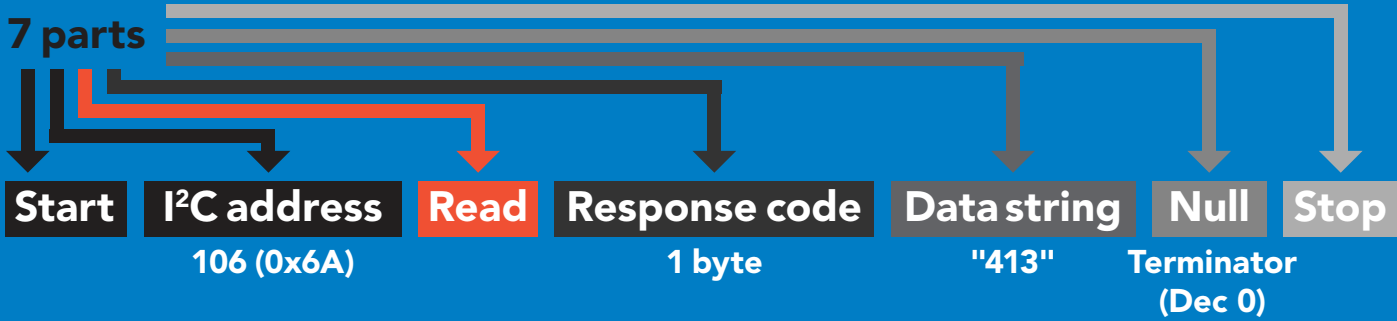
## Example



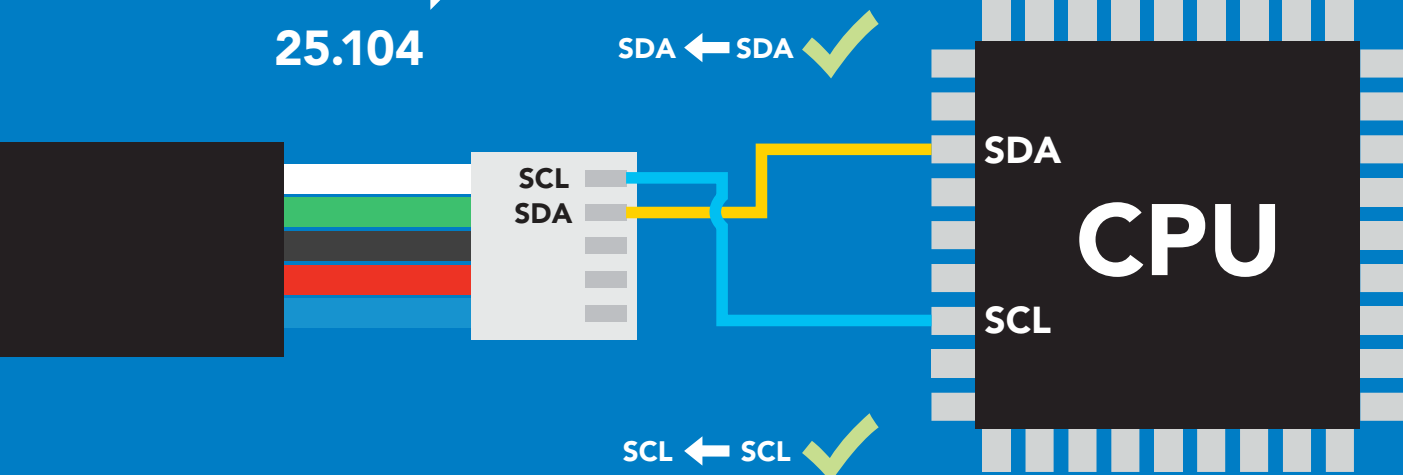
## Advanced



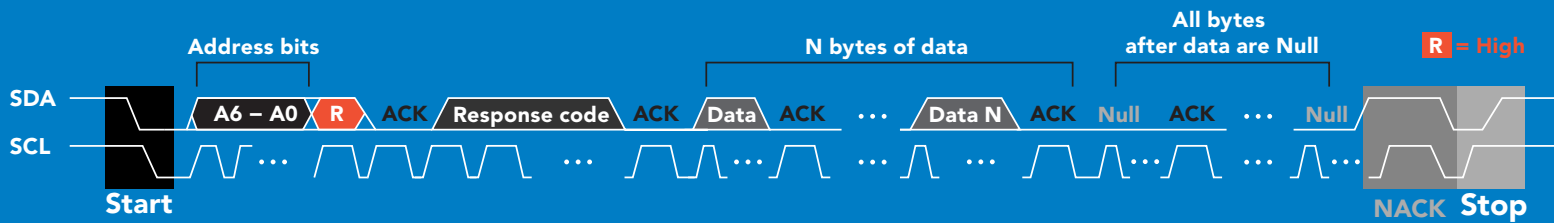
# Requesting data from device



25.104



## Advanced



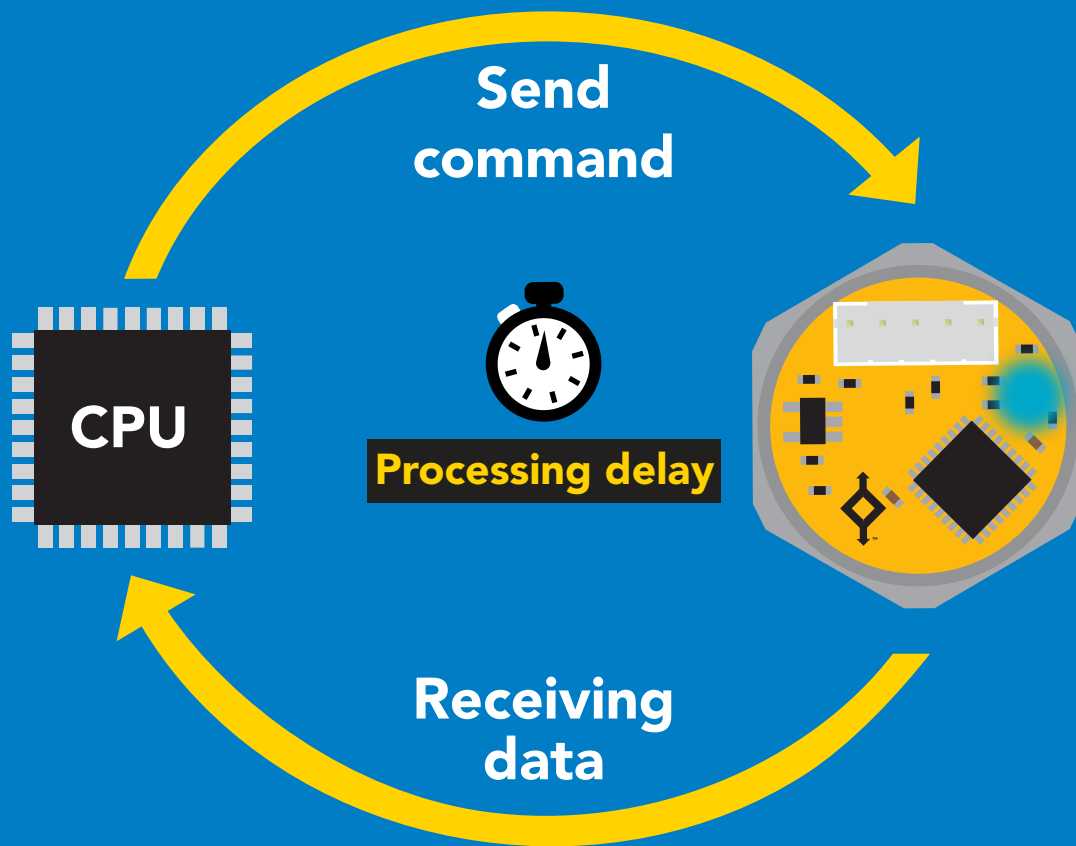
1 50 53 46 49 48 52 0 = 25.104

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;
```

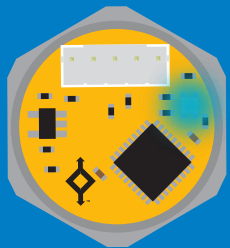
If there is no processing delay or the processing delay is too short, the response code will always be 254.

### Response codes

Single byte, not string

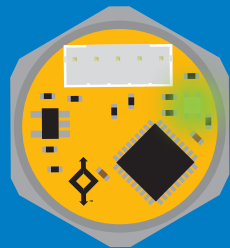
255	no data to send
254	still processing, not ready
2	syntax error
1	successful request

# LED color definition



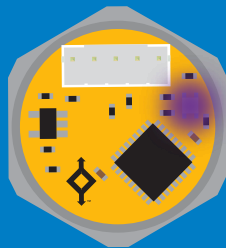
**Blue**

I<sup>2</sup>C standby



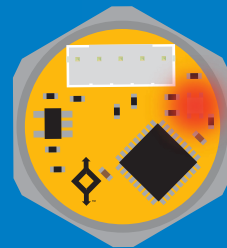
**Green**

Taking reading



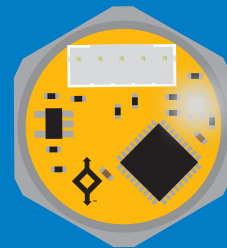
**Purple**

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



**Red**

Command  
not understood



**White**

Find

**5V**

LED ON  
**+0.25 mA**

**3.3V**

**+0.26 mA**



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

## command quick reference

All commands are ASCII strings or single ASCII characters.

<b>Command</b>	<b>Function</b>	
<b>Alarm</b>	enable/disable alarm	<b>pg. 45</b>
<b>Baud</b>	switch back to UART mode	<b>pg. 56</b>
<b>Cal</b>	performs custom calibration	<b>pg. 46</b>
<b>Dec</b>	add/remove decimal places	<b>pg. 47</b>
<b>Factory</b>	enable factory reset	<b>pg. 55</b>
<b>Find</b>	finds device with blinking white LED	<b>pg. 43</b>
<b>i</b>	device information	<b>pg. 50</b>
<b>I2C</b>	change I <sup>2</sup> C address	<b>pg. 54</b>
<b>L</b>	enable/disable LED	<b>pg. 42</b>
<b>Name</b>	set/show name of device	<b>pg. 49</b>
<b>Plock</b>	enable/disable protocol lock	<b>pg. 53</b>
<b>R</b>	returns a single reading	<b>pg. 44</b>
<b>Sleep</b>	enter sleep mode/low power	<b>pg. 52</b>
<b>Status</b>	retrieve status information	<b>pg. 51</b>
<b>U</b>	pressure units	<b>pg. 48</b>

# 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

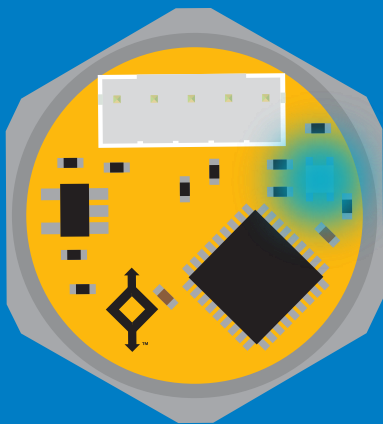
 **Wait 300ms** **1** **0**  
Dec Null

L,0

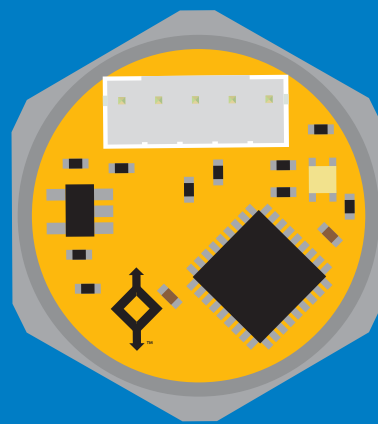
 **Wait 300ms** **1** **0**  
Dec Null

L,?

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

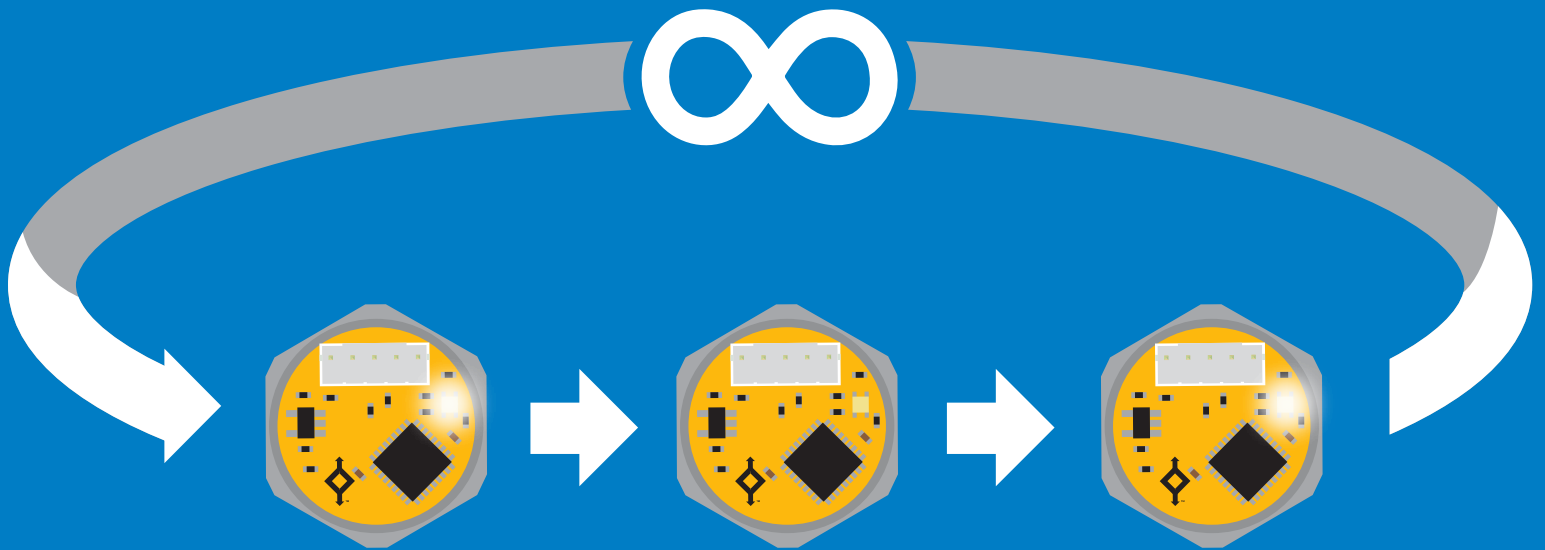
Find LED rapidly blinks white, used to help find device

## Example

## Response

Find

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



# Taking reading

## Command syntax


900ms  processing delay

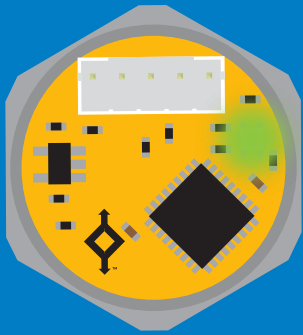
R return 1 reading

## Example

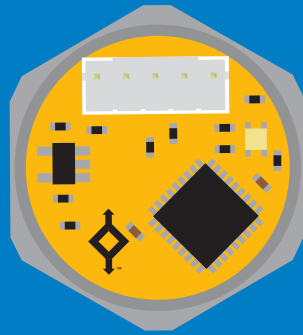
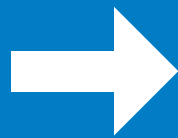
## Response

R

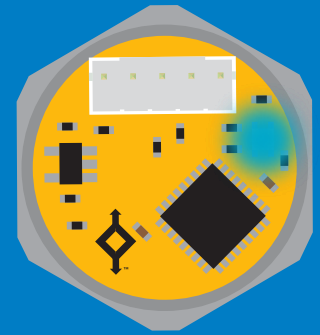
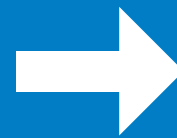
 Wait 900ms **1** **38.462** **0**  
Dec ASCII Null



**Green**  
Taking reading



Transmitting



**Cyan**  
Standby

# Alarm

300ms  processing delay

## Command syntax

The alarm pin will = 1 when pressure levels are > alarm set point. Alarm tolerance sets how far below the set point pressure levels need to drop before the pin will = 0 again.

- Alarm,en,[1,0] enable / disable alarm
- Alarm,n sets alarm
- Alarm,tol,n sets alarm tolerance
- Alarm,? alarm set?

## Example

## Response


Alarm,en,1

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

Alarm,35

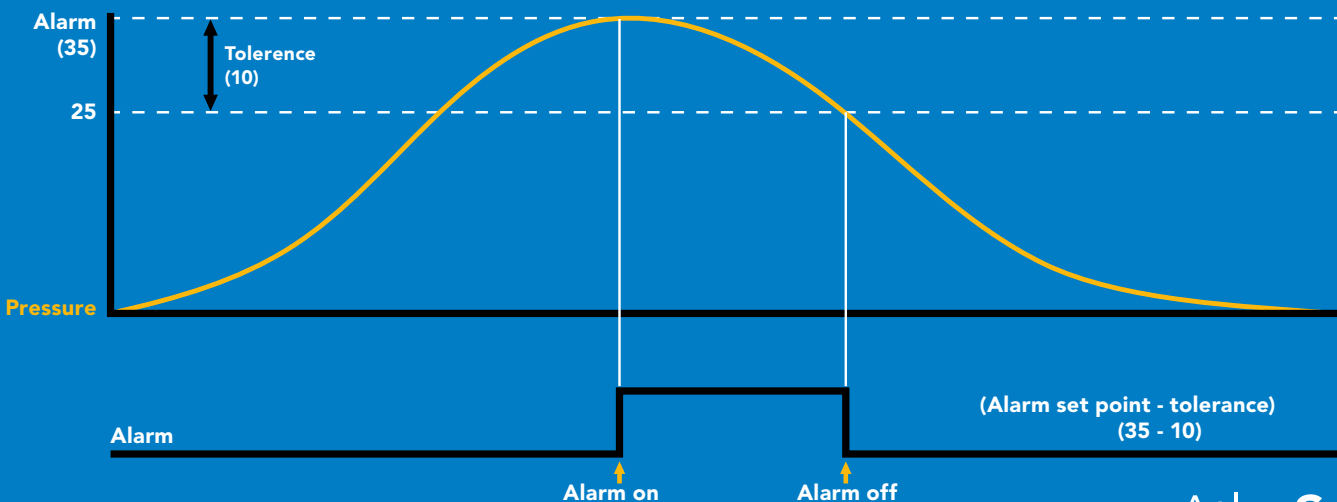
 Wait 300ms **1** **0**  
Dec Null

Alarm,tol,10

 Wait 300ms **1** **0** Pressure level must fall 10 units below set point for alarm to reset.  
Dec Null

Alarm,?

 Wait 300ms **1** **?,alarm,35,10,1** **0** if all are enabled  
Dec ASCII Null



# Custom calibration

900ms  processing delay

## Command syntax

Although calibration is not required, it may be necessary to adjust your 0 point or perform a custom calibration.

Cal,n	calibrates the high point
Cal,0	calibrates the zero point
Cal,clear	restores calibration to factory settings
Cal,?	device calibrated?

## Example

## Response

Cal,50



**1** **0**  
Dec Null

high point calibration in psi

Cal,0



**1** **0**  
Dec Null

low point calibration in psi

Cal,clear



**1** **0**  
Dec Null

Cal,?



**1** **?Cal,0** **0**  
Dec ASCII Null  
no calibration

or

**1** **?Cal,1** **0**  
Dec ASCII Null  
only zero point calibration

or

**1** **?Cal,2** **0**  
Dec ASCII Null  
only high point calibration

or

**1** **?Cal,3** **0**  
Dec ASCII Null  
zero and high point calibration

Calibration should be done using the pressure scale you have set the sensor to.

### Example

Readings are set to bar.  
High point calibration = 3.44  
(3.44 bar = 50 psi)

# Add/remove decimal places

900ms  processing delay

## Command syntax

Change how many decimal points the reading outputs.

Dec,n    n = number of decimal points between 0 and 3

Dec,?    number of decimal points the output is set to

## Example

## Response

R  Wait 900ms **1** **38.462** **0**  
Dec ASCII Null

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

R  Wait 900ms **1** **38.4** **0**  
Dec ASCII Null

Dec,?  Wait 900ms **1** **?Dec,1** **0**  
Dec ASCII Null

# Pressure units

(psi, atm, bar, kPa, inch's of water, cm of water)

## Command syntax

U,[1/0]	1 will add a unit identifier to the output	
U,psi	output will be in psi	
U,atm	output will be in atm	
U,bar	output will be in bar	
U,kPa	output will be in kPa	
U,inh2o	output will be in inches of water	(Resolution: 0.027")
U,cmh2o	output will be in cm of water	(Resolution: 0.7mm)
U,?	pressure units?	

## Example

## Response

U,bar

  
Wait 300ms

1	0
Dec	Null

U,1


  
Wait 300ms

1	0
Dec	Null

  
Wait 300ms

1	1.228,bar	0
Dec	ASCII	Null

U,?

  
Wait 300ms

1	?U,bar	0
Dec	ASCII	Null



# Naming device

300ms  processing delay

## Command syntax

Do not use spaces in the name

Name,n	set name	n =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Name,	clears name		Up to 16 ASCII characters															
Name,?	show name																	

## Example

## Response

Name,

 Wait 300ms

<b>1</b>	<b>0</b>	name has been cleared
Dec	Null	

Name,zzt

 Wait 300ms

<b>1</b>	<b>0</b>
Dec	Null

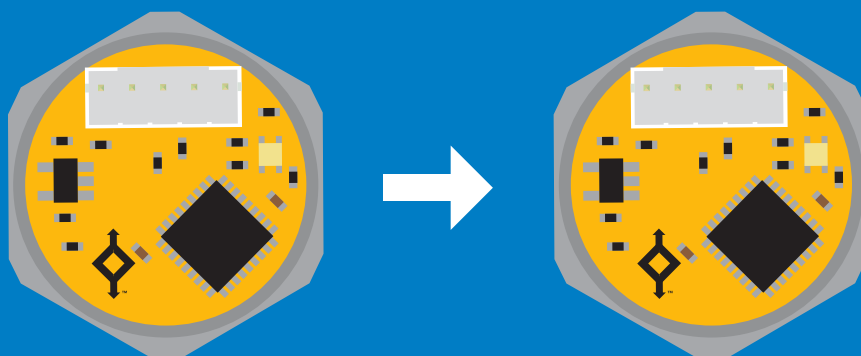
Name,?

 Wait 300ms

<b>1</b>	<b>?Name,zzt</b>	<b>0</b>
Dec	ASCII	Null

Name,zzt

Name,?



**1** **0**

**1** **?Name,zzt** **0**

# Device information

## Command syntax

300ms  processing delay

i device information

## Example

i

## Response



Wait 300ms

1

Dec

?i,PRS, 1.0

ASCII

0

Null

## Response breakdown

?i, PRS, 1.0  
↑     ↑  
Device Firmware

# Reading device status

Command syntax

300ms  processing delay

Status voltage at Vcc pin and reason for last restart

Example

Response

Status

 **1** **?Status,P,5.038** **0**  
Wait 300ms Dec ASCII 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**

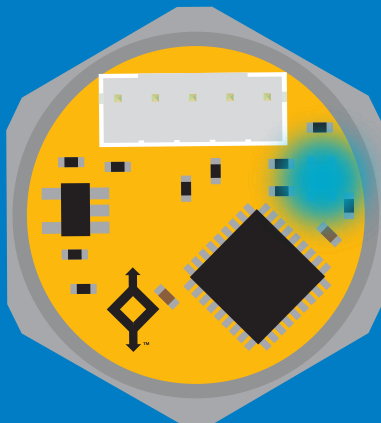
**wakes up device**

**5V**

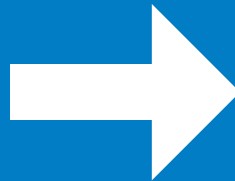
STANDBY	SLEEP
<b>13.4 mA</b>	<b>0.415 mA</b>

**3.3V**

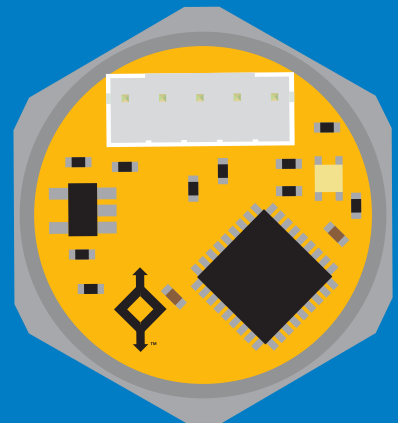
<b>12.4 mA</b>	<b>0.13 mA</b>
----------------	----------------



**Standby**



**Sleep**



**Sleep**

# Protocol lock

## Command syntax

300ms  processing delay

Plock,1 enable Plock

Plock,0 disable Plock

Plock,? Plock on/off?

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

**default**

## Example

## Response

Plock,1

  
Wait 300ms


<b>1</b>	<b>0</b>
Dec	Null

Plock,0

  
Wait 300ms

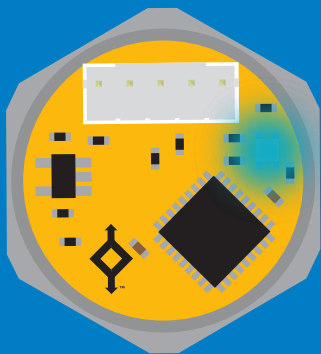
<b>1</b>	<b>0</b>
Dec	Null

Plock,?

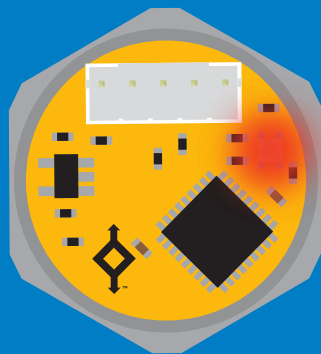
  
Wait 300ms

<b>1</b>	<b>?Plock,1</b>	<b>0</b>
Dec	ASCII	Null

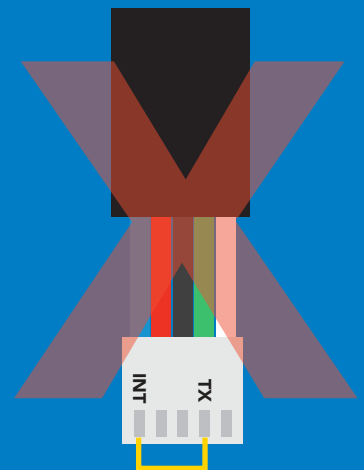
Plock,1



Baud, 9600



cannot change to UART



cannot change to UART

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

**Command syntax**

**300ms**  **processing delay**

**I2C,n** sets I<sup>2</sup>C address and reboots into I<sup>2</sup>C mode

**Example**

**Response**

**I2C,101**

**device reboot**  
(no response given)

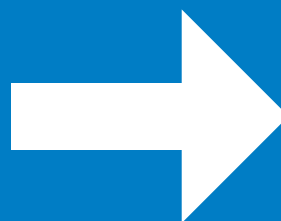
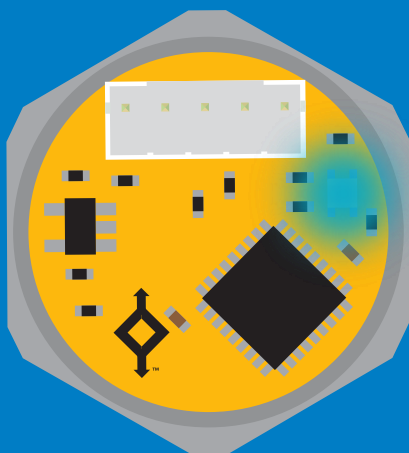
## Warning!

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

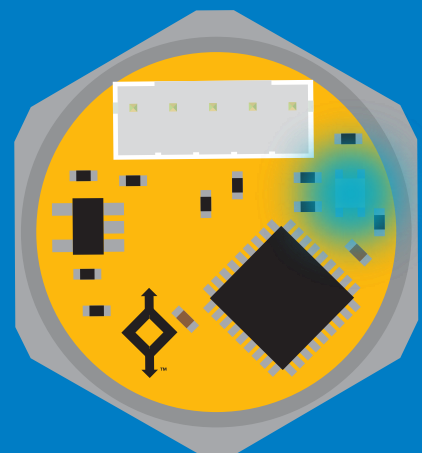
Default I<sup>2</sup>C address is 106 (0x6A).

**n = any number 1 – 127**

**I2C,101**



**(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

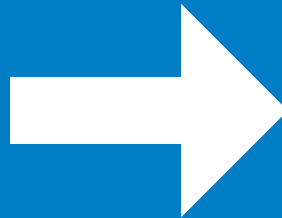
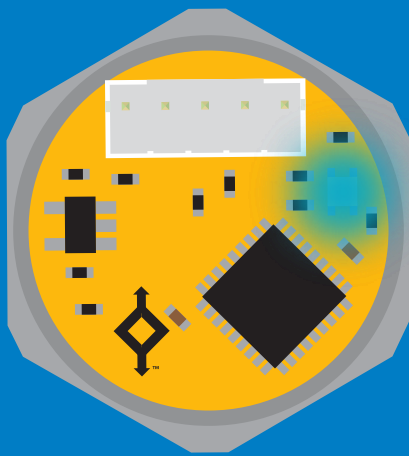
## Response

Factory

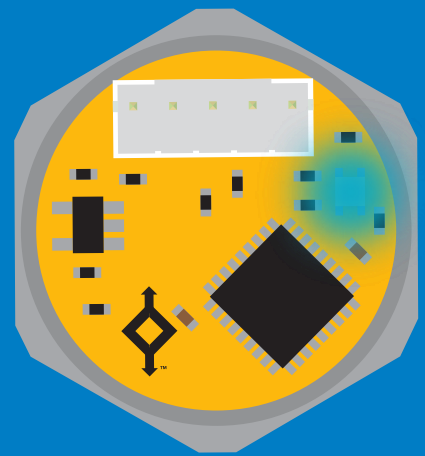
device reboot  
(no response given)

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

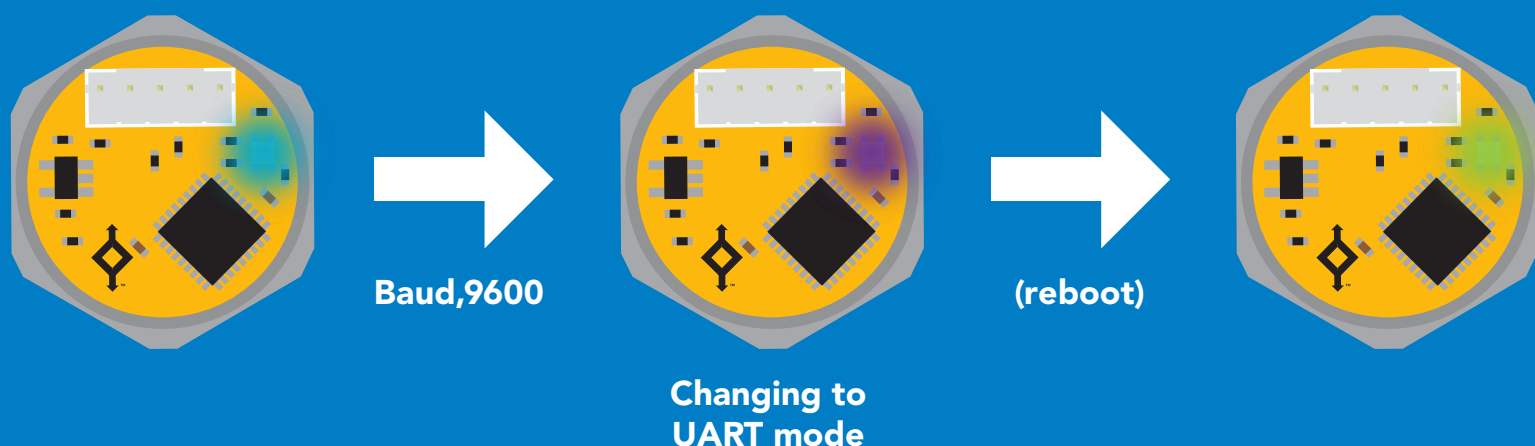
### Example

Baud,9600

### Response

reboot in UART mode  
(no response given)

n = [ 300  
1200  
2400  
9600  
19200  
38400  
57600  
115200

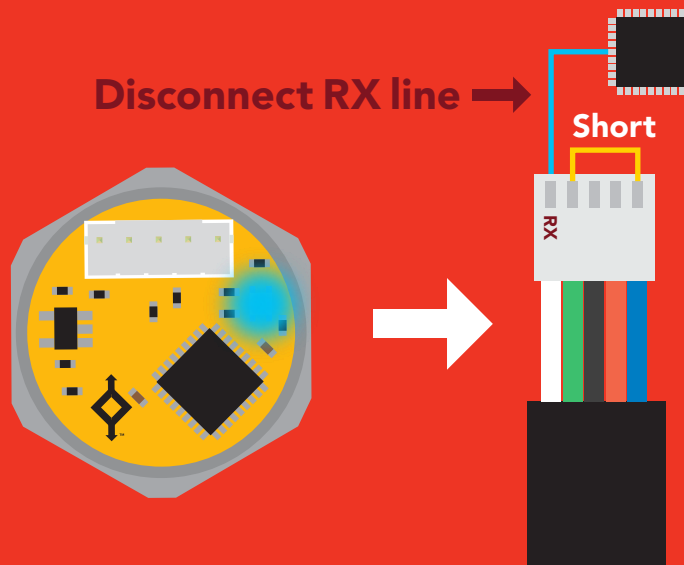
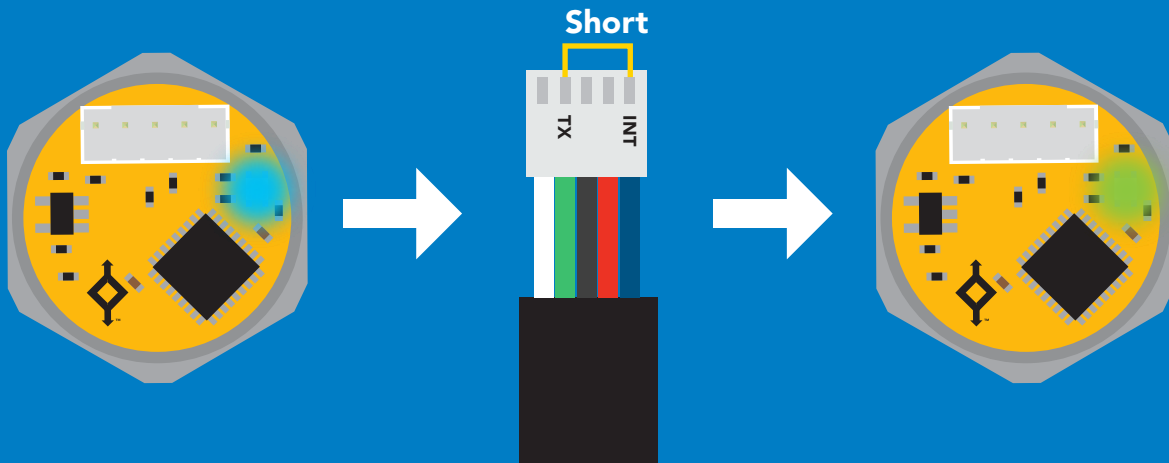




# Manual switching to UART

- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to INT
- 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



# Datasheet change log

## Datasheet V 2.0

Revised design of EZO-PRS throughout document.

## Datasheet V 1.6

Revised naming device info on pages 25 & 49.

## Datasheet V 1.5

Added the custom calibration pages on pages 22 & 46.

## Datasheet V 1.4

Revised accuracy value on cover page.

## Datasheet V 1.3

Revised pressure output in PSI to 50.000 on cover pg.  
added inches of water & cm of water resolution info on pages 23 and 46.

## Datasheet V 1.2

Updated firmware info on pg 55.

## Datasheet V 1.1

Moved Default state to pg 9.

## Datasheet V 1.0

Initial release - New datasheet

# Firmware updates

V1.0 – Initial release (Aug, 7 2019)

V1.01 – (Nov, 5 2019)

- Fixed glitch where the alarm was not initially set correctly.

V1.02 – (April, 9 2021)

- Added custom calibration

# Warranty

Atlas Scientific™ Warranties the EZO-PRS™ Embedded Pressure Sensor to be free of defect during the debugging phase of device implementation, or 30 days after receiving the EZO-PRS™ Embedded Pressure Sensor (which ever comes first).

## The debugging phase

The debugging phase as defined by Atlas Scientific™ is the time period when the EZO-PRS™ Embedded Pressure Sensor is inserted into a bread board, or shield. If the EZO-PRS™ is being debugged in a bread board, the bread board must be devoid of other components. If the EZO-PRS™ Embedded Pressure Sensor is being connected to a microcontroller, the microcontroller must be running code that has been designed to drive the EZO-PRS™ Embedded Pressure Sensor exclusively and output the EZO-PRS™ 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-PRS™ Embedded Pressure Sensor warranty:**

- **Soldering any part of the EZO-PRS™ Embedded Pressure Sensor.**
- **Running any code, that does not exclusively drive the EZO-PRS™ Embedded Dosing Pump and output its data in a serial string.**
- **Embedding the EZO-PRS™ Embedded Pressure Sensor 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-PRS™ Embedded Pressure Sensor, against the thousands of possible variables that may cause the EZO-PRS™ Embedded Pressure Sensor 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-PRS™ Embedded Pressure Sensors continued operation. This is because that would be equivalent to Atlas Scientific™ taking responsibility over the correct operation of your entire device.