

EZO-CO₂TM

Embedded NDIR CO₂ Sensor

Reads	Gaseous CO₂
Range	0 – 10,000 ppm
Calibration	Factory calibrated
Response time	1 reading per second
Resolution	1 ppm
Accuracy	+/- 3% +/- 30 ppm
Warmup time	10 seconds
Cable length	1 meter
Data protocol	UART & I²C
Default I ² C address	105 (0x69)
Data format	ASCII
Operating voltage	3.3V – 5V



Table of contents

Operating principle	4	Sensor warm-up	7
Physical properties	5	Calibration theory	8
Sensor properties	5	Custom calibration	8
Pin out	6	Available data protocol	9

UART

Default state	10
UART mode	11
Receiving data from device	12
Sending commands to device	13
LED color definition	14
UART quick command page	15
LED control	16
Find	17
Continuous mode	18
Single reading mode	19
Alarm	20
Custom calibration	21
Export/import calibration	22
Enable/disable internal temp	23
Naming device	24
Device information	25
Response codes	26
Reading device status	27
Sleep mode/low power	28
Change baud rate	29
Protocol lock	30
Factory reset	31
Change to I2C mode	32
Manual switching to I2C	33

I²C

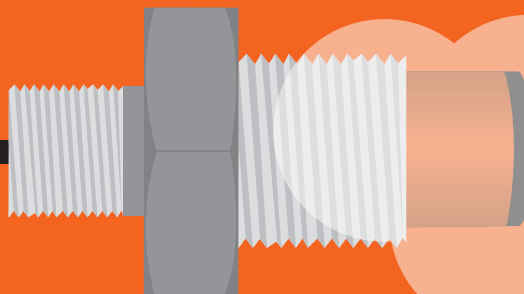
I ² C mode	35
Sending commands	36
Requesting data	37
Response codes	38
Processing delay	38
LED color definition	39
I²C quick command page	40
LED control	41
Find	42
Taking reading	43
Alarm	44
Custom calibration	45
Export/import calibration	46
Enable/disable internal temp	47
Device information	48
Reading device status	49
Sleep mode/low power	50
Protocol lock	51
I ² C address change	52
Factory reset	53
Change to UART mode	54
Manual switching to UART	55

Datasheet change log	56
Firmware updates	56
Warranty	57

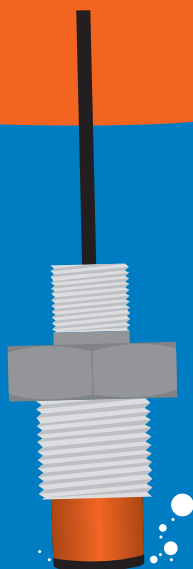
Attention

The EZO-CO2™ is 100% operational out of the box.
CALIBRATION IS UNNECESSARY

This sensor detects
GASEOUS CO2



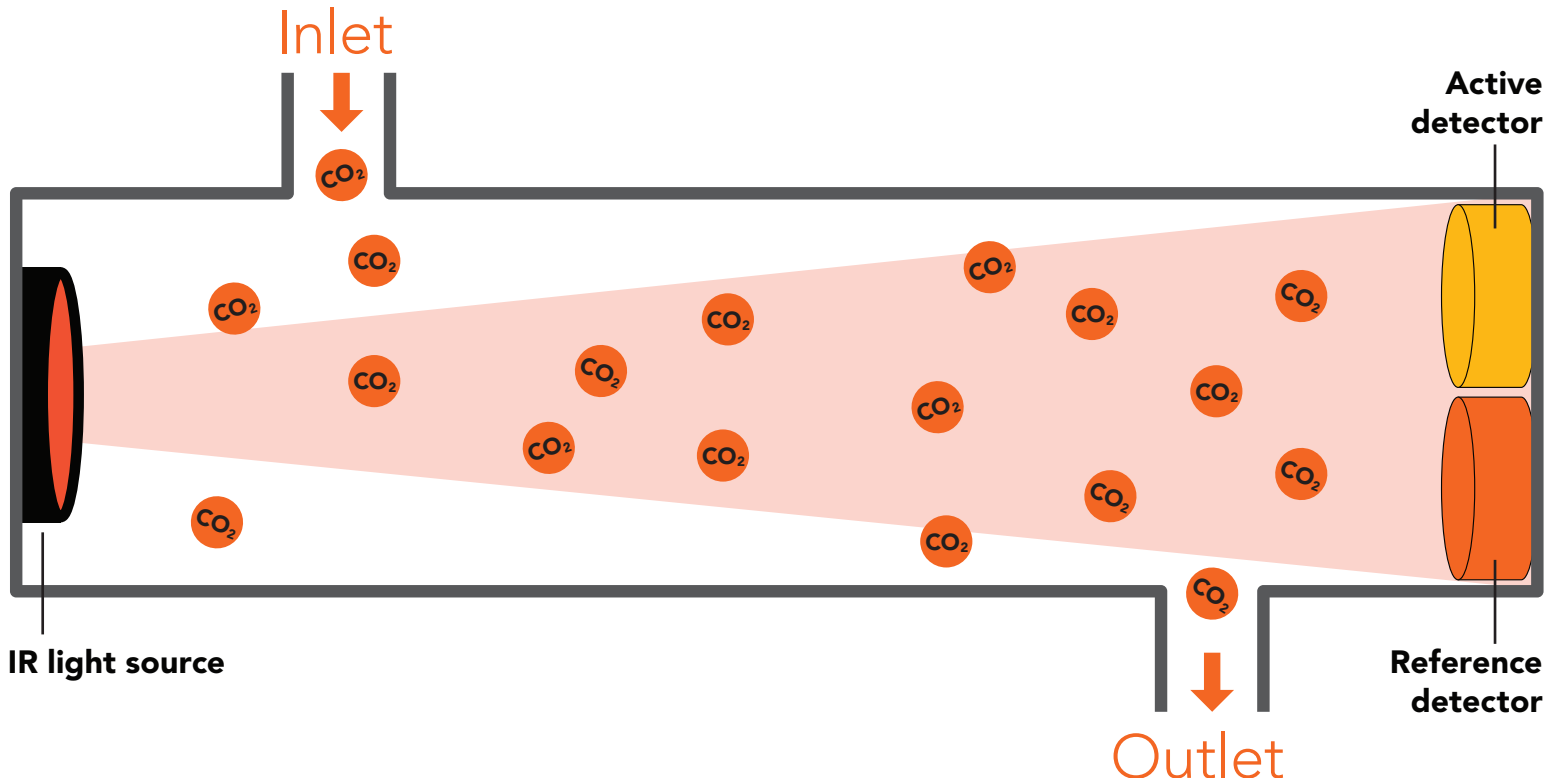
X



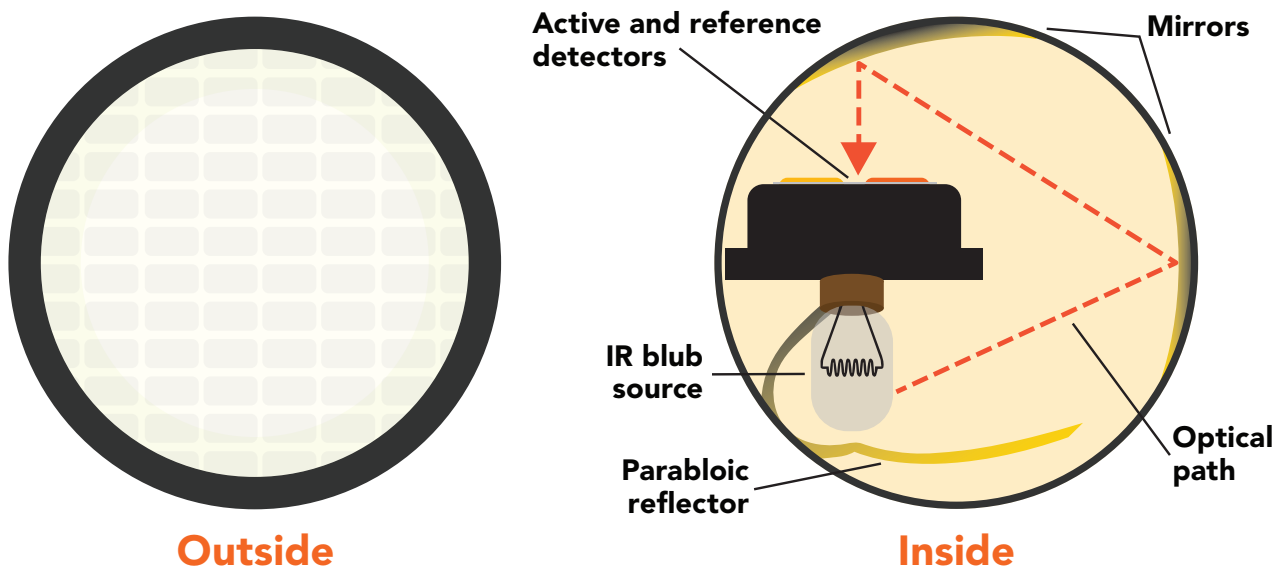
This sensor does not
read dissolved CO2.
DO NOT SUBMERGE!

Operating principle

The Atlas Scientific EZO-CO2™ Embedded CO2 Sensor uses a non-dispersive infra-red (NDIR) gas detection cell to derive CO2 content in a gaseous matrix. The NDIR detection cell is a single wavelength spectrophotometer that has been specifically designed to detect 4.2µm infrared radiation.

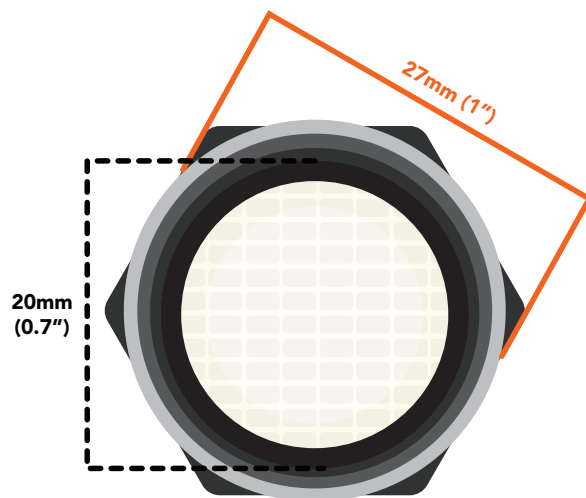
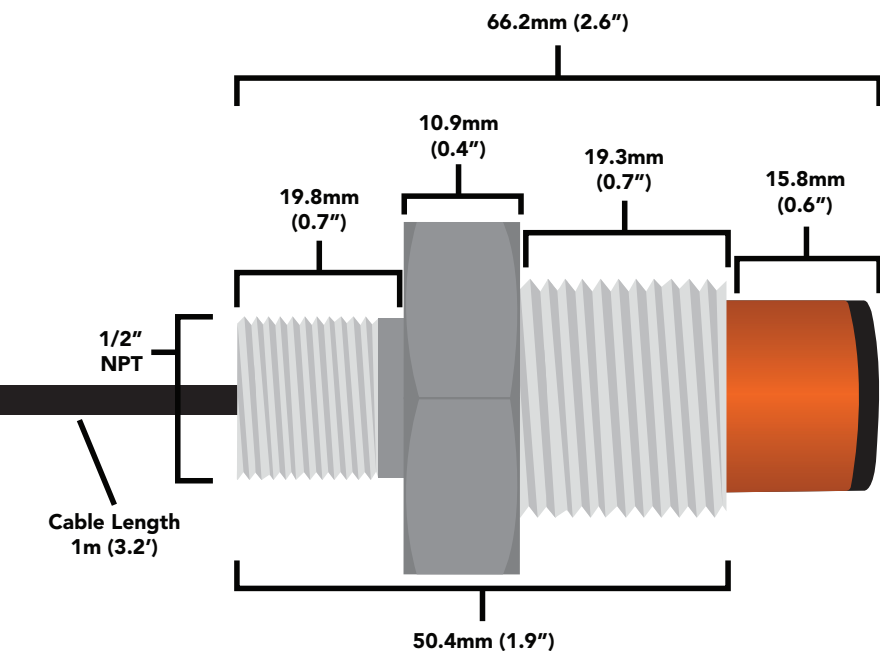


Gaseous CO_2 has a prominent absorption band centered at 4.2µm. CO_2 content is derived by quantifying how much light energy has been lost when it travels through a gaseous matrix over a fixed distance.



Physical properties

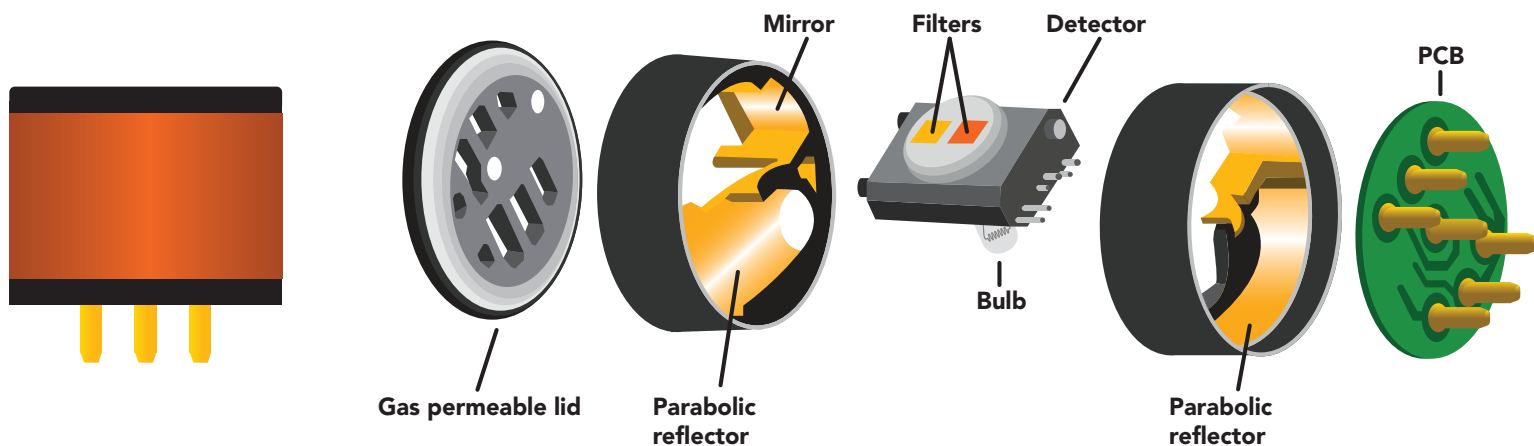
The EZO-CO₂™ sensor only detects gaseous CO₂ concentrations. This device cannot read dissolved CO₂ concentrations. **DO NOT SUBMERGE IN LIQUID.**



Weight 133g



Sensor properties



Pin out

Data and power cable pinout

White – RX/SCL
Green – TX/SDA
Black – GND
Red – VCC
Blue – ALM



The interrupt pin will go high when a set CO2 level has been crossed.



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

See page 20 to enable CO2 level alarm.

Power consumption

	LED	MAX	SLEEP
5V	ON	45 mA	3.4 mA
	OFF	44 mA	
3.3V	ON	42 mA	3.0 mA
	OFF	41 mA	

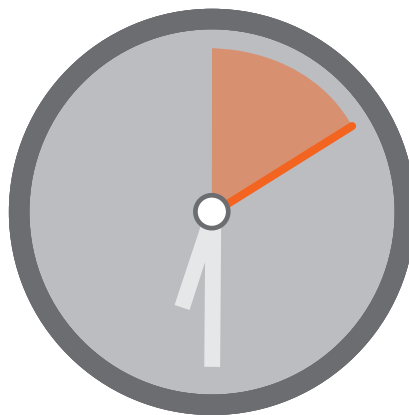
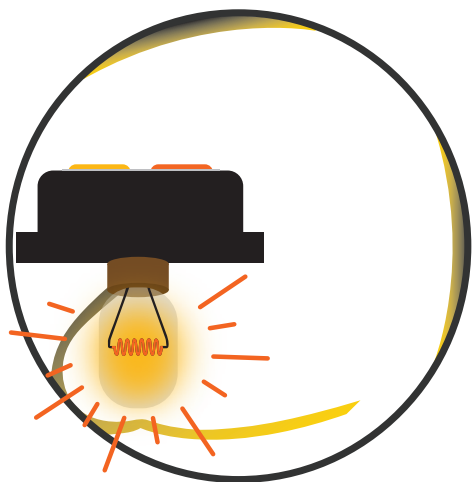
Absolute max ratings

Parameter	MIN	TYP	MAX
Storage temperature	-65 °C		75 °C
Operational temperature	-20 °C	25 °C	50 °C
VCC	3.3V	3.3V	5.5V

Humidity Range 0 to 95% rh non-condensing

Sensor warm-up

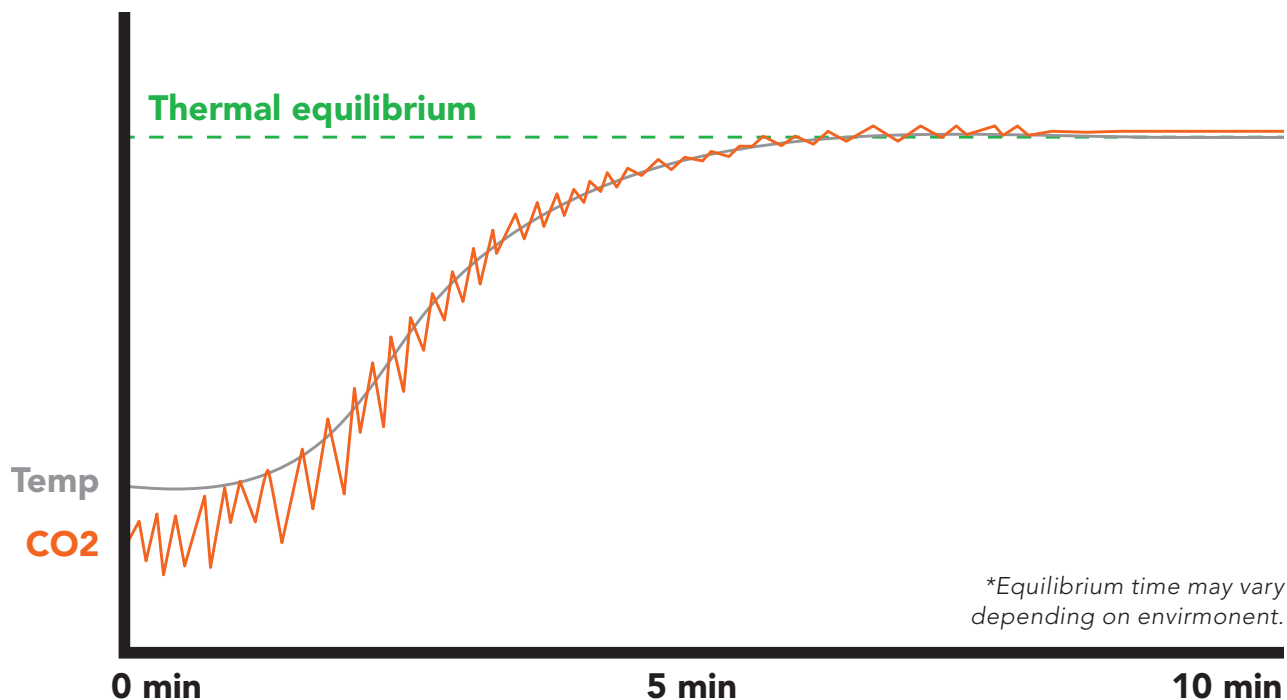
When the Atlas Scientific EZO-CO2™ Embedded CO2 Sensor is first powered on (or wakes up from sleep mode) the sensor must warm-up before it can output readings. The warm-up process takes 10 seconds to complete.



10 sec

During the first 10 seconds of operation the output will be: ***warm**

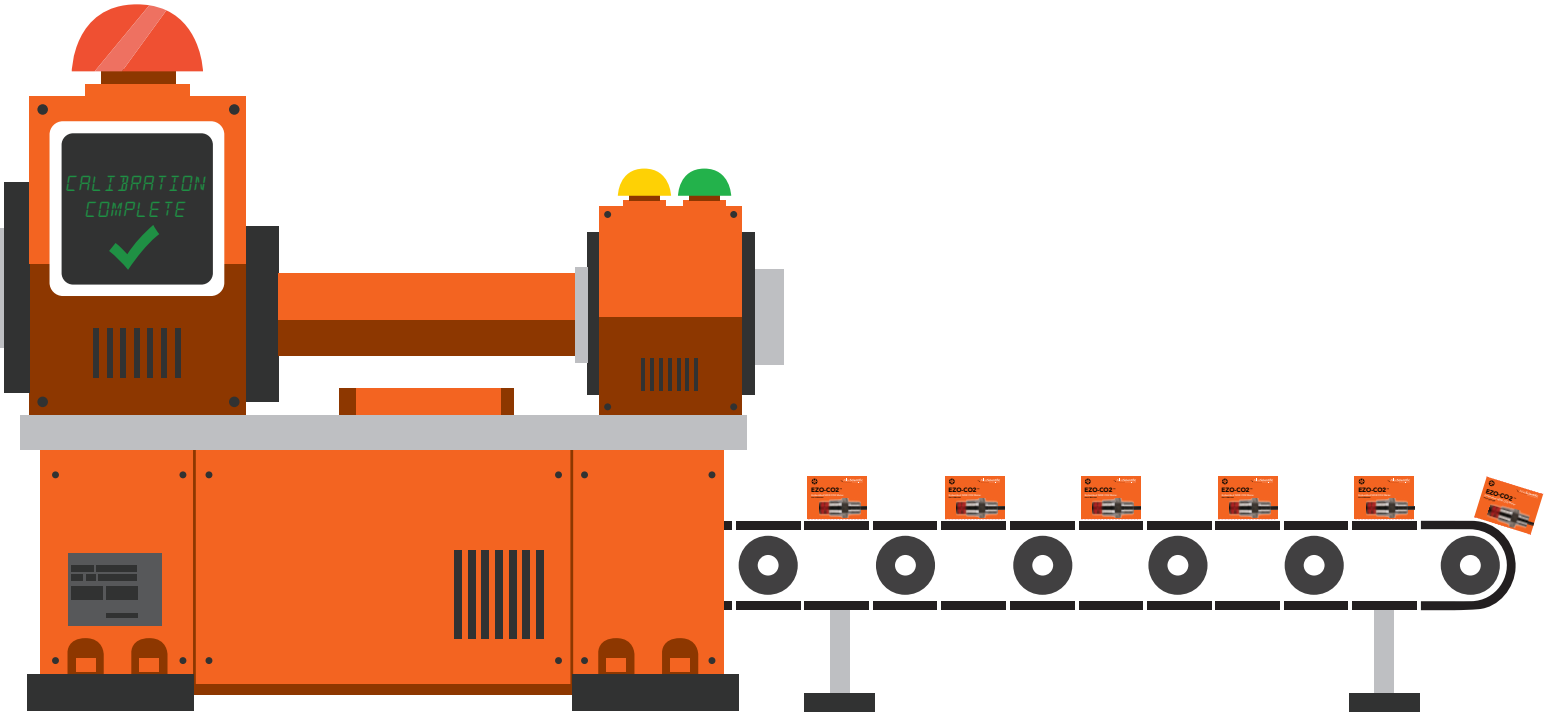
Once warming is finished, CO2 readings will be output. The device will continue to warm-up over several minutes. As the internal temperature stabilizes, so will the CO2 readings.



To see the internal temperature of the sensor and watch as it stabilizes, use the 'O' command found on page 23.

Calibration theory

The Atlas Scientific EZO-CO2™ Embedded CO2 Sensor comes pre-calibrated, and does not need to be recalibrated. Atlas Scientific performs a two-point factory calibration as part of the manufacturing process.



Low point calibration = 0 ppm
High point calibration = 4,000 ppm

The factory calibration data is permanently stored in the sensor and cannot be erased.

Custom calibration

One or two-point calibration can be done at any time. When custom calibration is used, factory calibration will be ignored. To revert back to the factory calibration simply clear the custom calibration.

See page 21 for custom calibration commands.

✓ Available data protocols

UART

default

I²C

✗ Unavailable data protocols

SPI

Analog

RS-485

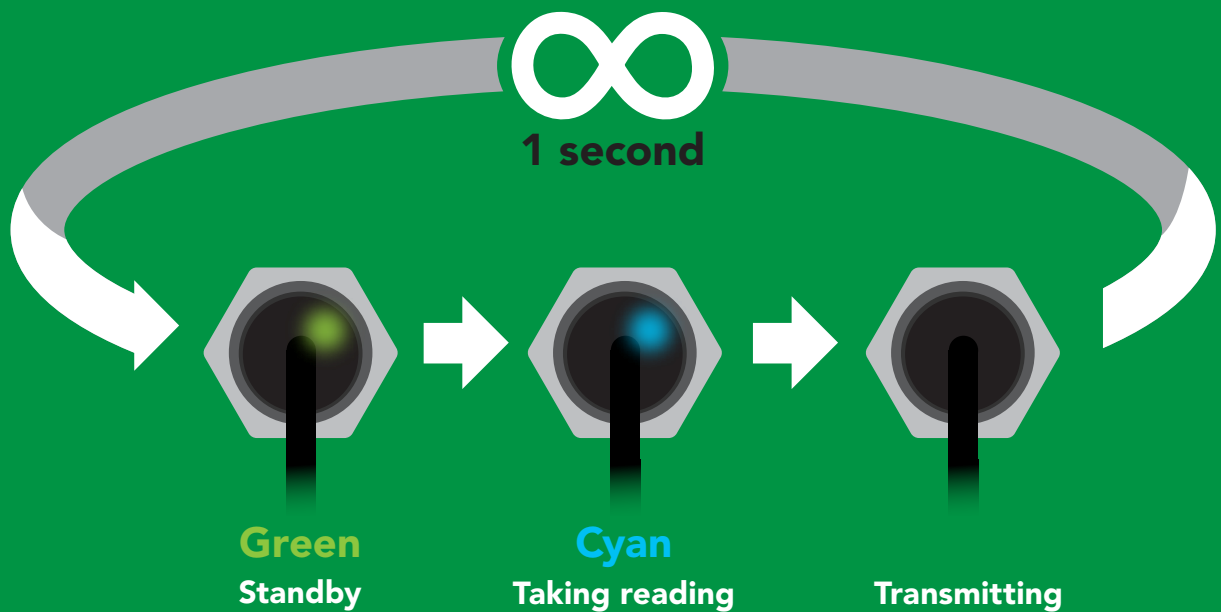
Mod Bus

4–20mA

Default state

UART mode

Baud 9,600
Readings continuous
Speed 1 second
LED on



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

- Sleep mode

UART mode

8 data bits
1 stop bit

no parity
no flow control

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

RX
Data in



TX
Data out



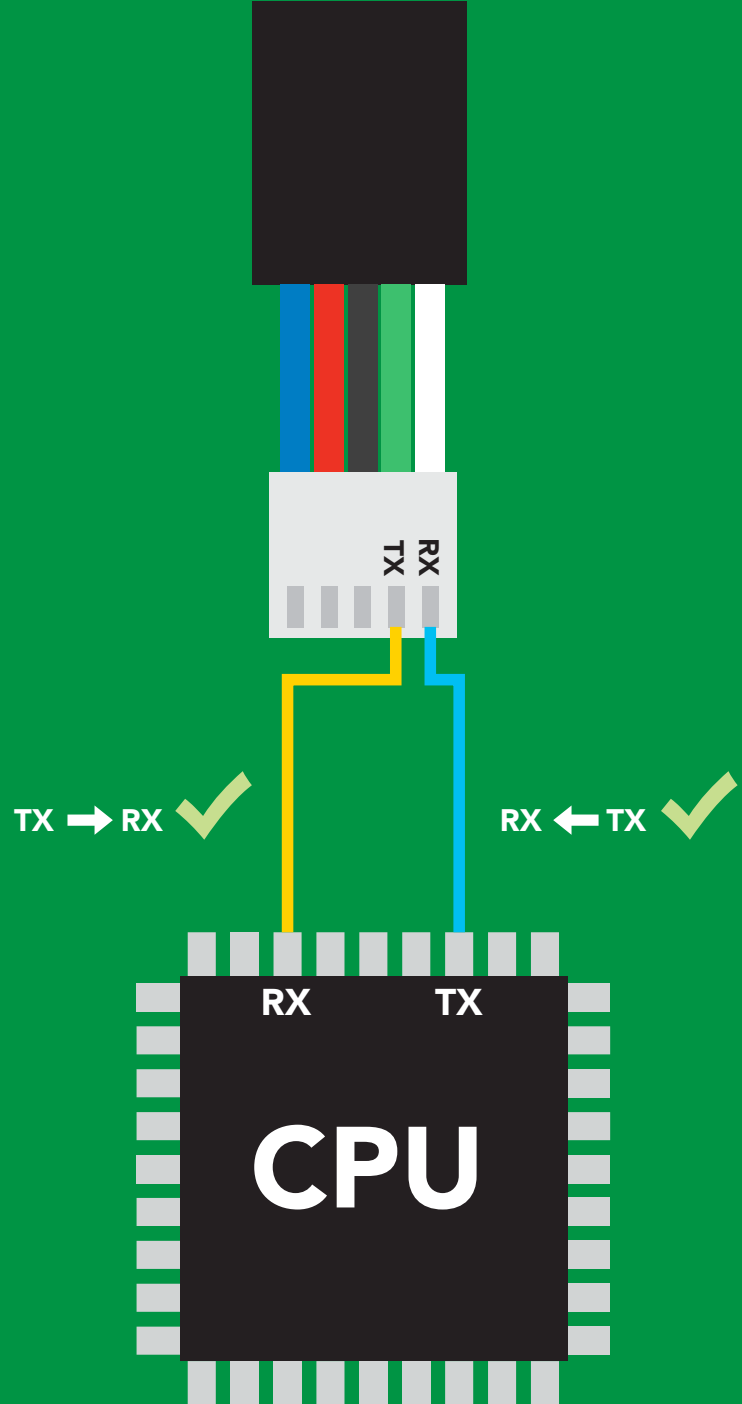
Vcc 3.3V – 5V

0V



VCC

0V



Data format

Units PPM
Encoding ASCII
Format string
Terminator carriage return

Data type unsigned int
Decimal places 0
Smallest string 2 characters
Largest string 12 characters

Receiving data from device

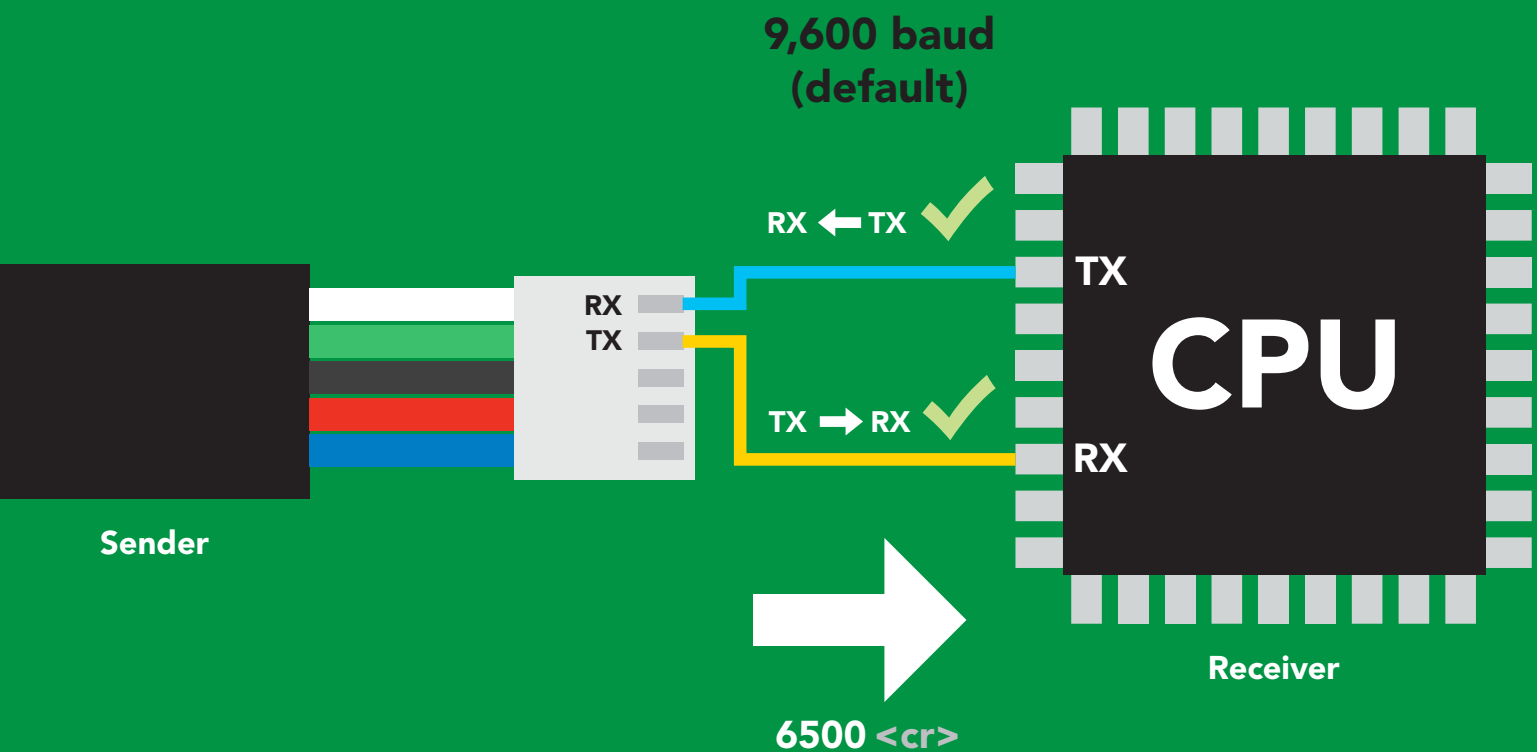
2 parts

ASCII data string

Command

Carriage return <cr>

Terminator



Advanced

ASCII: 6 5 0 0 <cr>

Hex: 36 35 30 30 0D

Dec: 54 53 48 48 13

Sending commands to device

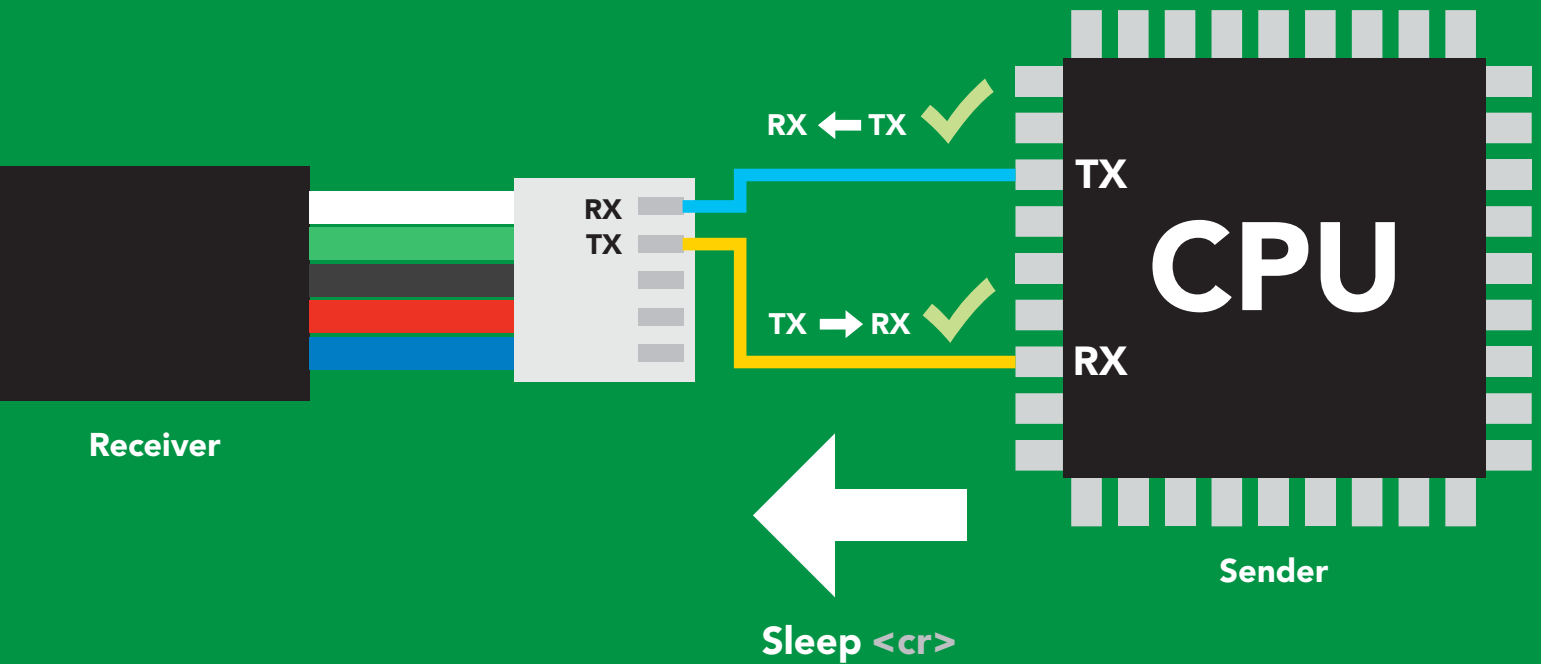
2 parts

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



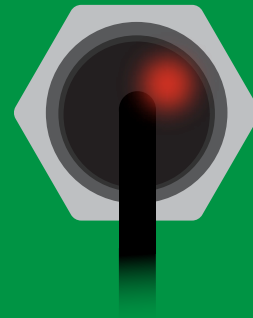
Cyan

Taking reading



Purple

Changing
baud rate



Red

Command
not understood



White

Find

5V

LED ON
+2.5 mA

3.3V

+1 mA

UART mode

command quick reference

All commands are ASCII strings or single ASCII characters.

Command	Function		Default state
Alarm	enable/disable alarm	pg. 20	n/a
Baud	change baud rate	pg. 29	9,600
C	enable/disable continuous mode	pg. 18	enabled
Cal	performs custom calibration	pg. 21	n/a
Export/import	export/import calibration	pg. 22	n/a
Factory	enable factory reset	pg. 31	n/a
Find	finds device with blinking white LED	pg. 17	n/a
i	device information	pg. 25	n/a
I2C	change to I ² C mode	pg. 32	not set
L	enable/disable LED	pg. 16	enabled
Name	set/show name of device	pg. 24	not set
O	enable/disable internal temperature	pg. 23	disabled
Plock	enable/disable protocol lock	pg. 30	n/a
R	returns a single reading	pg. 19	n/a
Sleep	enter sleep mode/low power	pg. 28	n/a
Status	retrieve Status Information	pg. 27	n/a
*OK	enable/disable response codes	pg. 26	n/a

LED control

Command syntax

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

L,0 <cr> LED off

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

Example

L,1 <cr>

Response

*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.

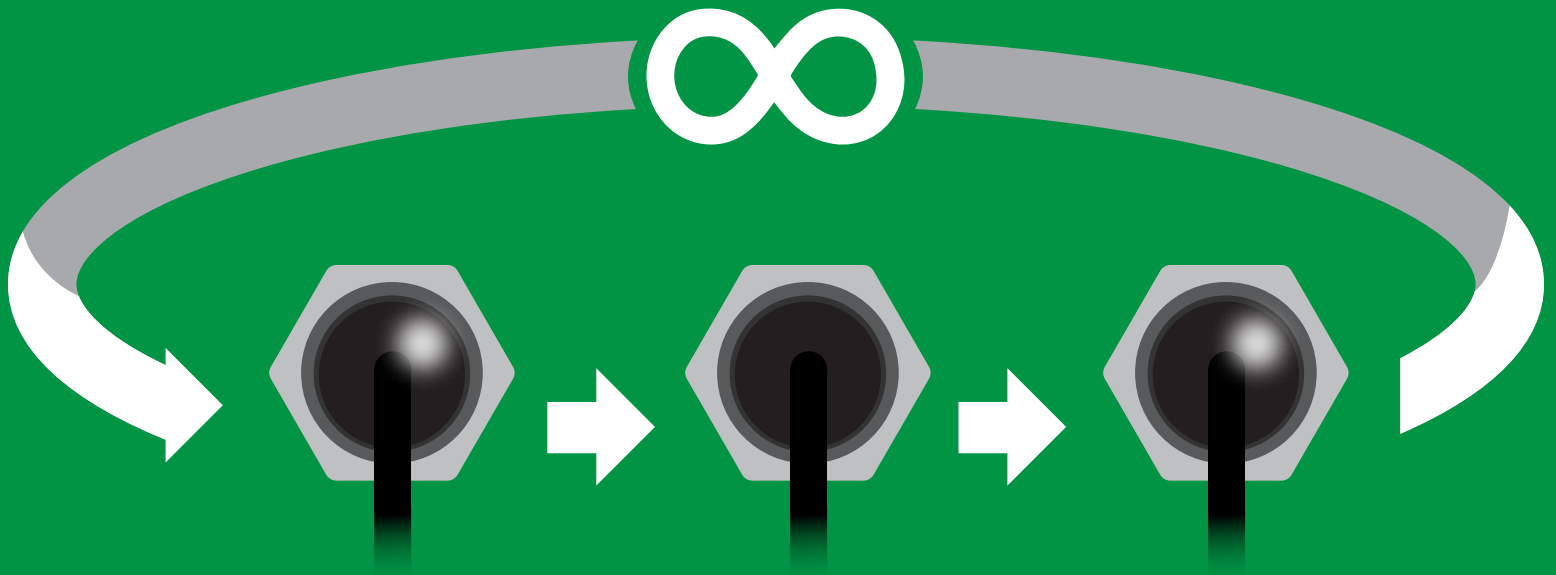
Find <cr> LED rapidly blinks white, used to help find device

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>**
CO2 (1 sec) <cr>
CO2 (2 sec) <cr>
CO2 (n sec) <cr>

C,30 <cr>

***OK <cr>**
CO2 (30 sec) <cr>
CO2 (60 sec) <cr>
CO2 (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

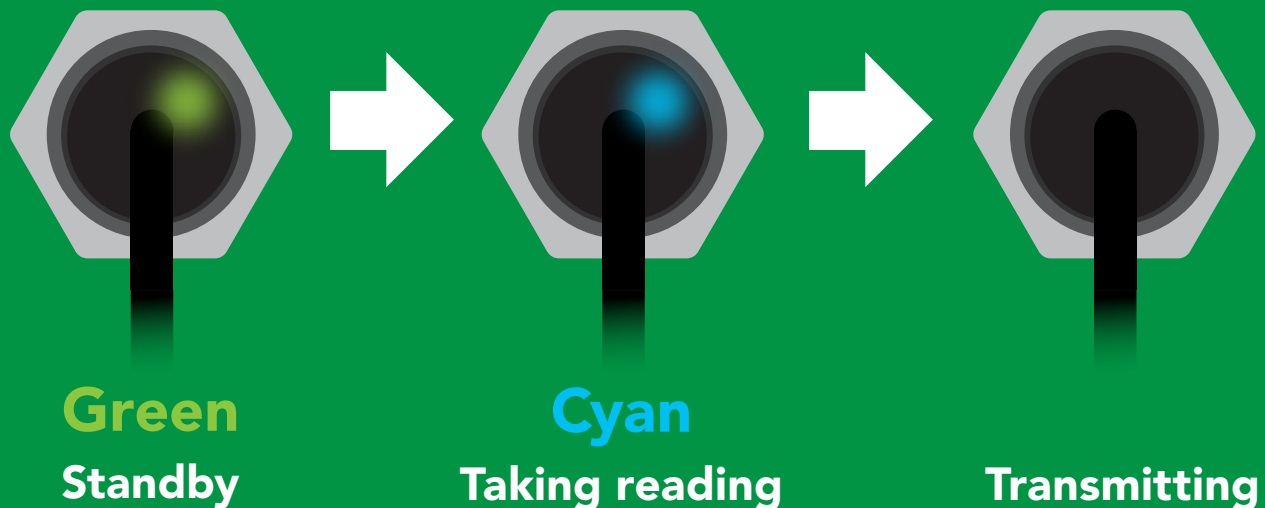
Example

R <cr>

Response

6500 <cr>

***OK** <cr>



1 second

Alarm

Command syntax

The alarm pin will = 1 when CO2 levels are > alarm set point. Alarm tolerance sets how far below the set point CO2 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 (0 - 500 ppm)
Alarm,?	<cr>	alarm set?

Example

Response

Alarm,en,1 <cr>

*OK <cr> Enable alarm

Alarm,1200 <cr>

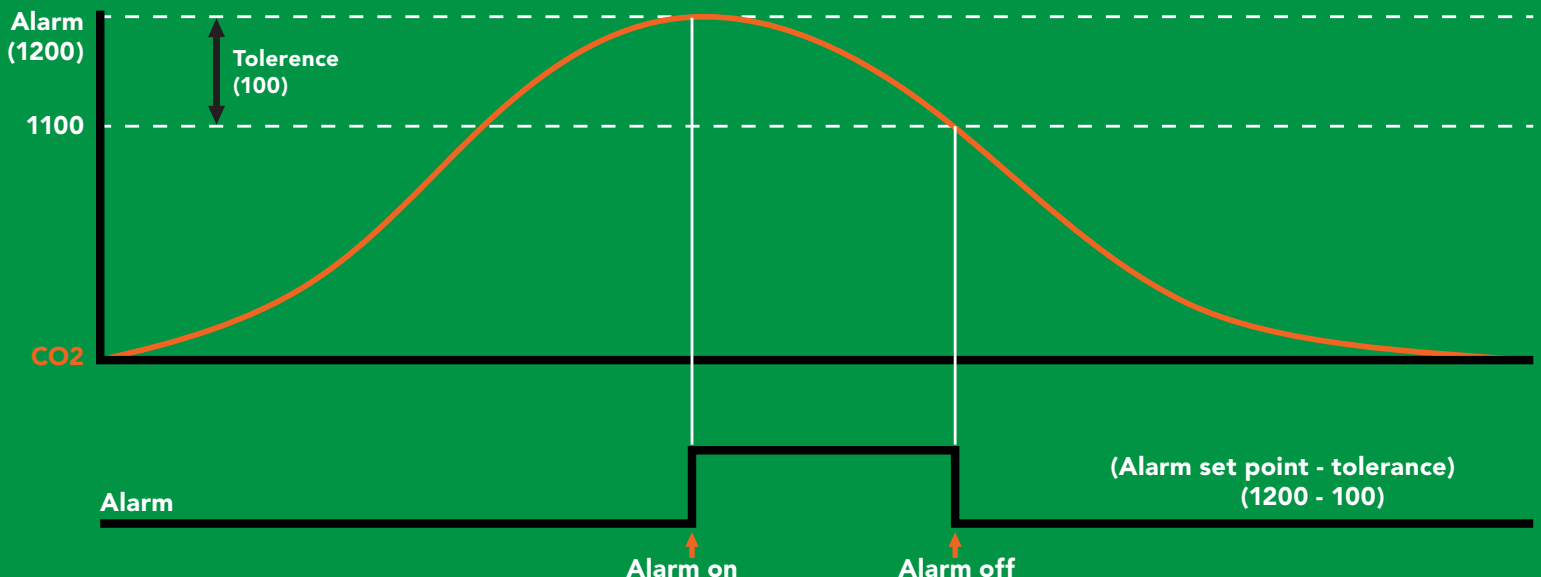
*OK <cr>

Alarm,tol,100 <cr>

*OK <cr> CO2 level must fall 100 ppm below set point for alarm to reset.

Alarm,? <cr>

?,alarm,1200,100 <cr> if all are enabled



Custom calibration

Command syntax

High point calibration can be from 3,000 ppm to 5,000 ppm. Calibration outside of that range may lead to accuracy issues.

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,3900 <cr>

*OK <cr>

Cal,0 <cr>

*OK <cr>

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

This device comes pre-calibrated.

Custom calibration should not be performed without scientific grade calibration gasses.

Export/import calibration

Command syntax

Export: Use this command to save calibration settings
Import: Use this command to load calibration settings

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

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)**

(8 more)

⋮

Export <cr>

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

Export <cr>

***DONE**

Disabling *OK simplifies this process

Import, n
(FIFO)

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

Enable/disable internal temperature from output string

Command syntax

O,t,[1,0] <cr> enable or disable internal temperature

Example	Response
O,t,1 <cr>	*OK <cr> enable temperature
O,t,0 <cr>	*OK <cr> disable temperature
O,? <cr>	?O,ppm,t <cr> if internal temp is enabled

Enabling the internal temperature should only be used to confirm that the device is at thermal equilibrium. Refer to page 6

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

Response

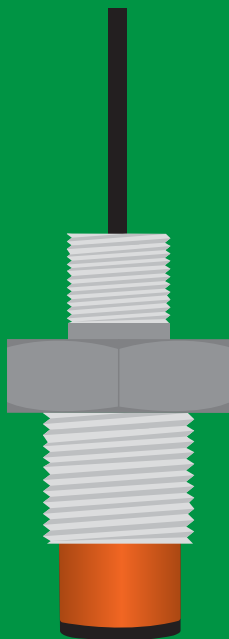
Name,zzt <cr>

*OK <cr>

Name,? <cr>

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

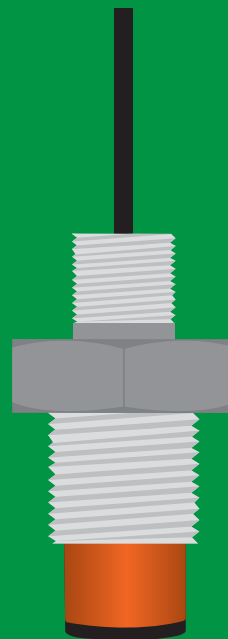
Name,zzt <cr>



*OK <cr>



Name,? <cr>



Name,zzt <cr>
*OK <cr>

Device information

Command syntax

i <cr> device information

Example

i <cr>

Response

?i,CO2,1.0 <cr>
*OK <cr>

Response breakdown

?i,	CO2,	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>

6,500 <cr>
***OK** <cr>

***OK,0** <cr>

no response, ***OK** disabled

R <cr>

6,500 <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>

***SL**

Any command

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

5V

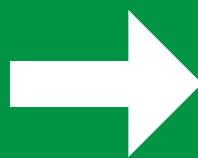
MAX
45 mA

SLEEP
3.4 mA

3.3V

42 mA

3.0 mA



Sleep <cr>



Change baud rate

Command syntax

Baud,n <cr> change baud rate

Example

Baud,38400 <cr>


Response

*OK <cr>

Baud,? <cr>

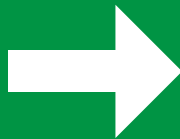
?Baud,38400 <cr>

*OK <cr>

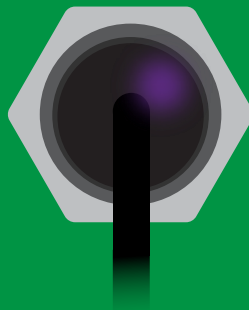
n =  300
1200
2400
9600 default
19200
38400
57600
115200



Standby



Baud,38400 <cr>



Changing
baud rate

*OK <cr>



(reboot)



Standby

Protocol lock

Command syntax

Locks device to UART mode.

Plock,1 <cr> enable Plock

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

Plock,? <cr> Plock on/off?

Example

Response

Plock,1 <cr>

*OK <cr>

Plock,0 <cr>

*OK <cr>

Plock,? <cr>

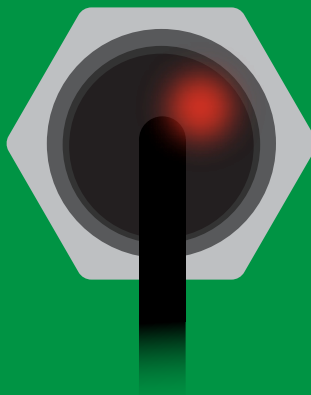
?Plock,1 <cr> or ?Plock,0 <cr>

Plock,1

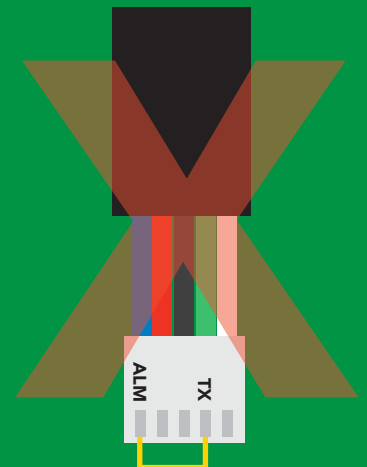


*OK <cr>

I2C,100



cannot change to I²C
*ER <cr>



cannot change to I²C

Factory reset

Command syntax

Clears custom calibration
"*OK" enabled

Factory <cr> enable factory reset

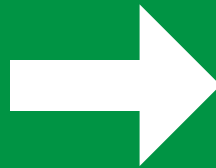
Example

Factory <cr>

Response

***OK** <cr>

Factory <cr>



(reboot)



***OK** <cr>

***RS** <cr>

***RE** <cr>

Baud rate will not change

Change to I²C mode

Command syntax

Default I²C address 105 (0x69)

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

n = any number 1 – 127

Example

Response

I2C,100 <cr>

*OK (reboot in I²C mode)

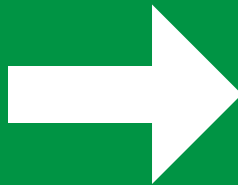
Wrong example

Response

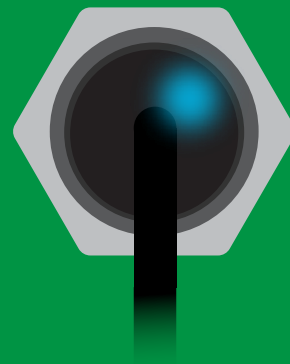
I2C,139 <cr> n ≠ 127

*ER <cr>

I2C,100



(reboot)



Green
*OK <cr>

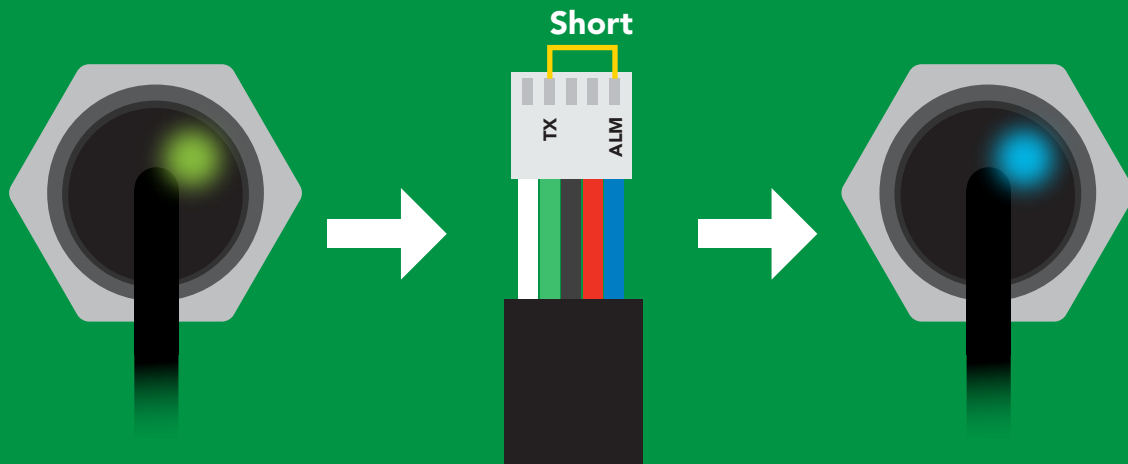
Blue
now in I²C mode

Manual switching to I²C

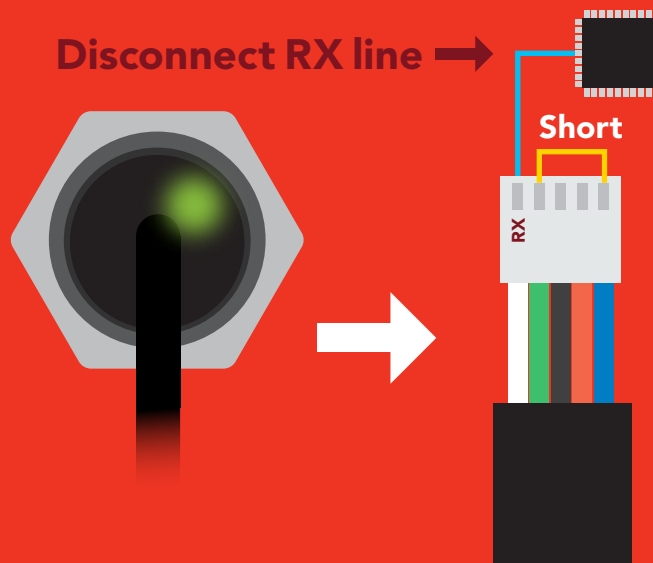
- Make sure Plock is set to 0
- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to ALM
- Confirm RX is disconnected
- Connect ground (power on)
- Wait for LED to change from **Green** to **Blue**
- Disconnect ground (power off)
- Reconnect all data and power

Manually switching to I²C will set the I²C address to 105 (0x69)

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

- Sleep mode

I²C mode

I²C address (0x01 – 0x7F)
105 (0x69) default

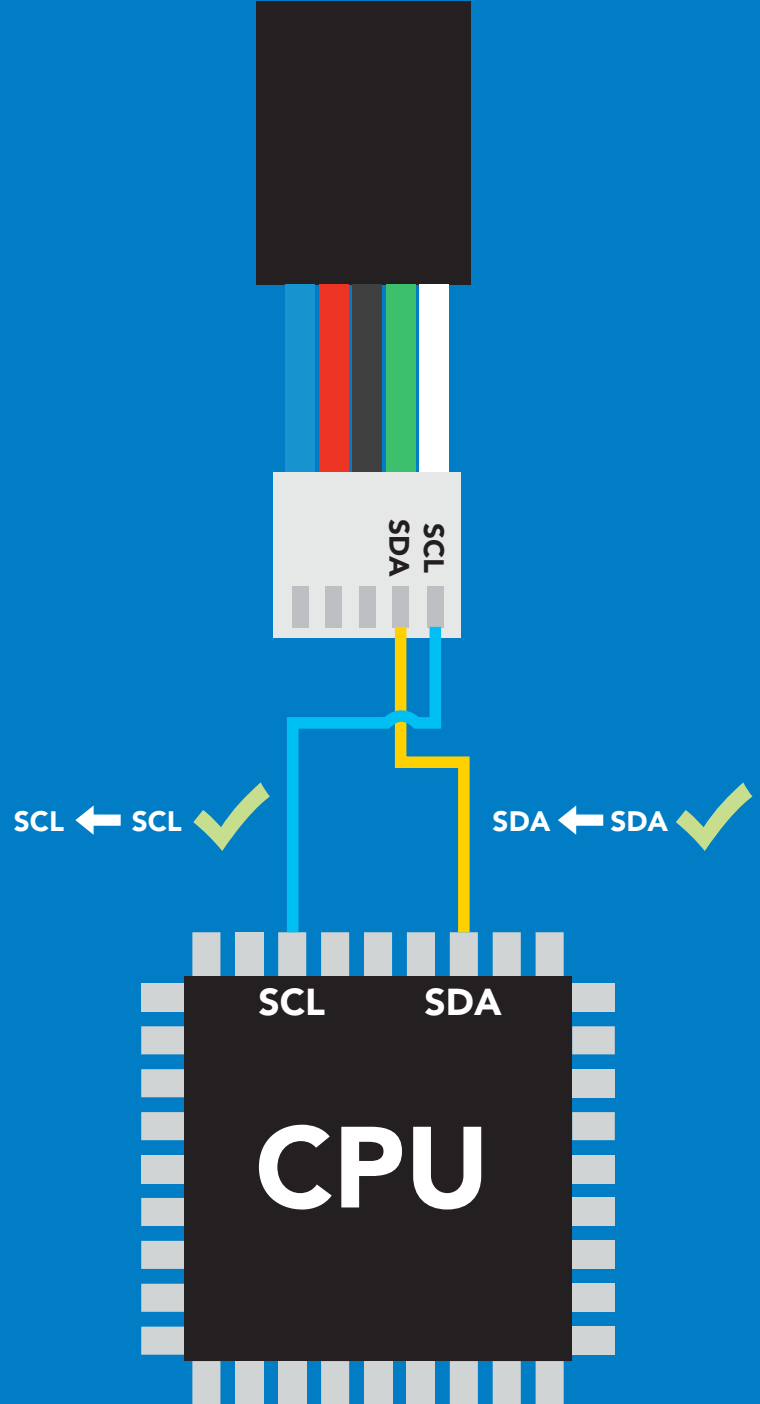
Vcc 3.3V – 5.5V

Clock speed 100 – 400 kHz

SDA 

SCL 

 VCC
0V 0V



Data format

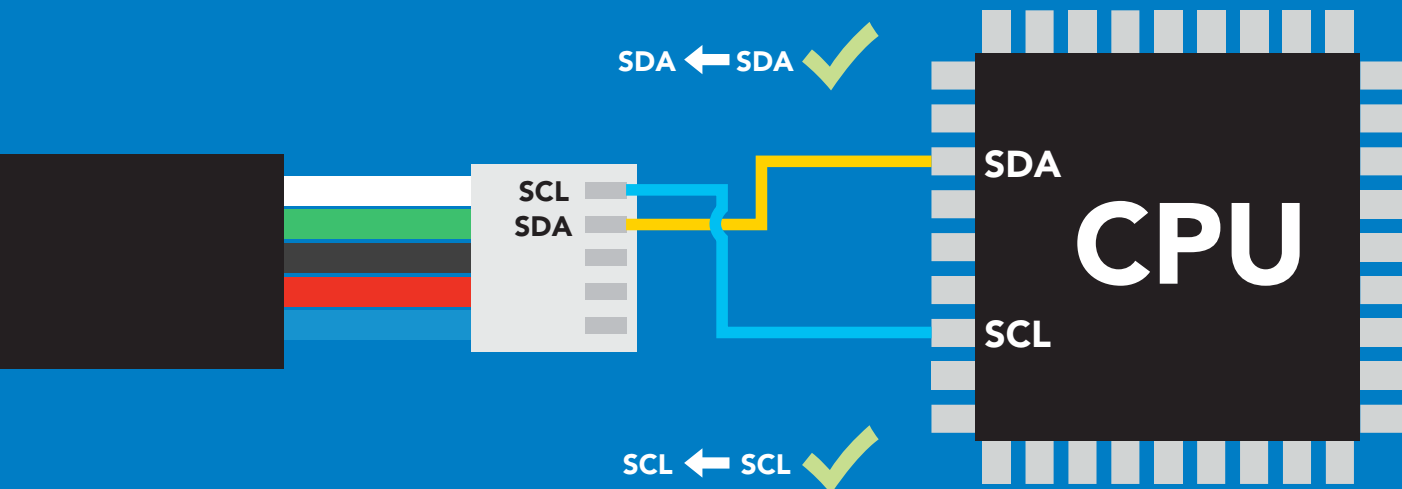
Reading	CO2	Data type	unsigned int
Units	PPM	Decimal places	0
Encoding	ASCII	Smallest string	2 characters
Format	string	Largest string	12 characters

Sending commands to device

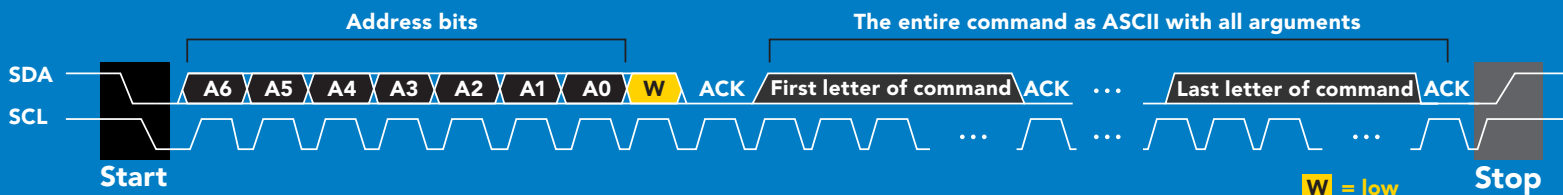
5 parts



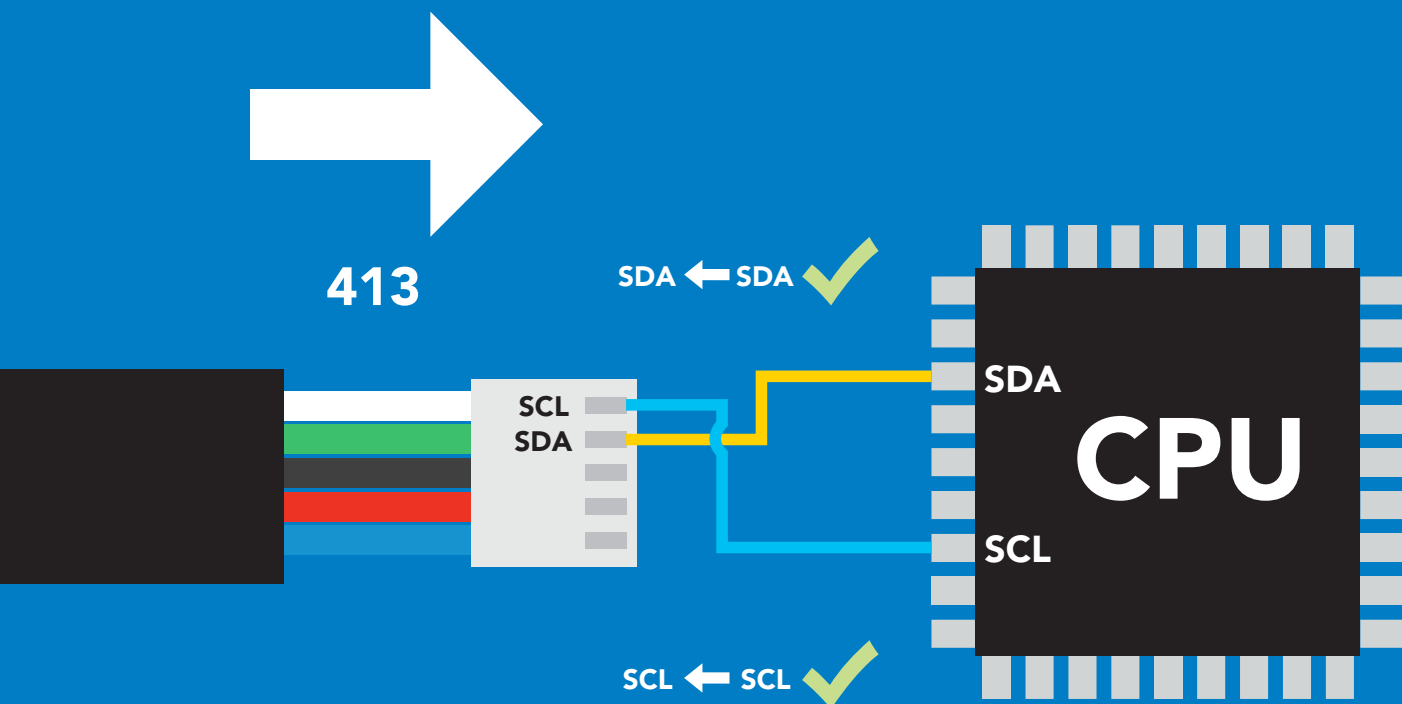
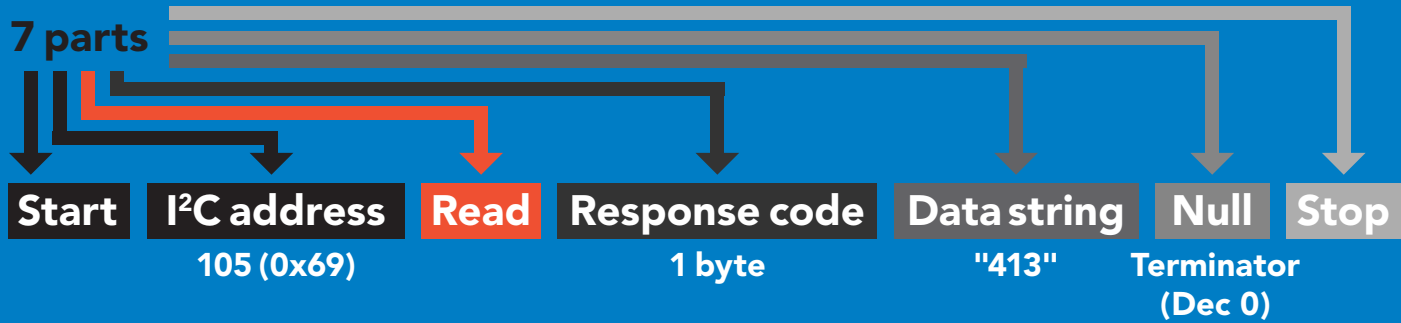
Example



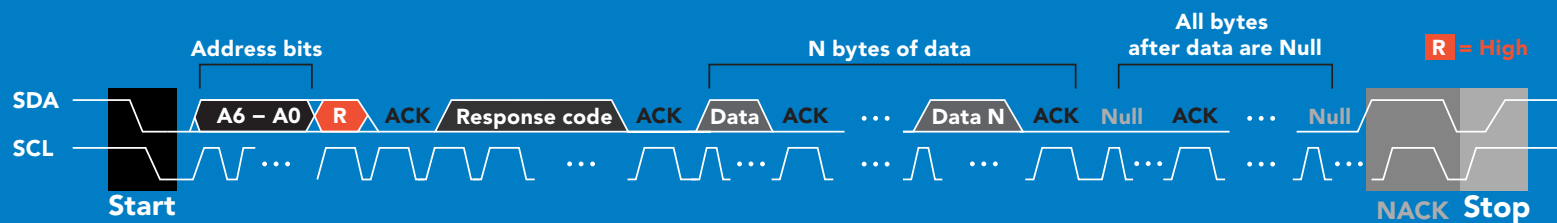
Advanced



Requesting data from device



Advanced



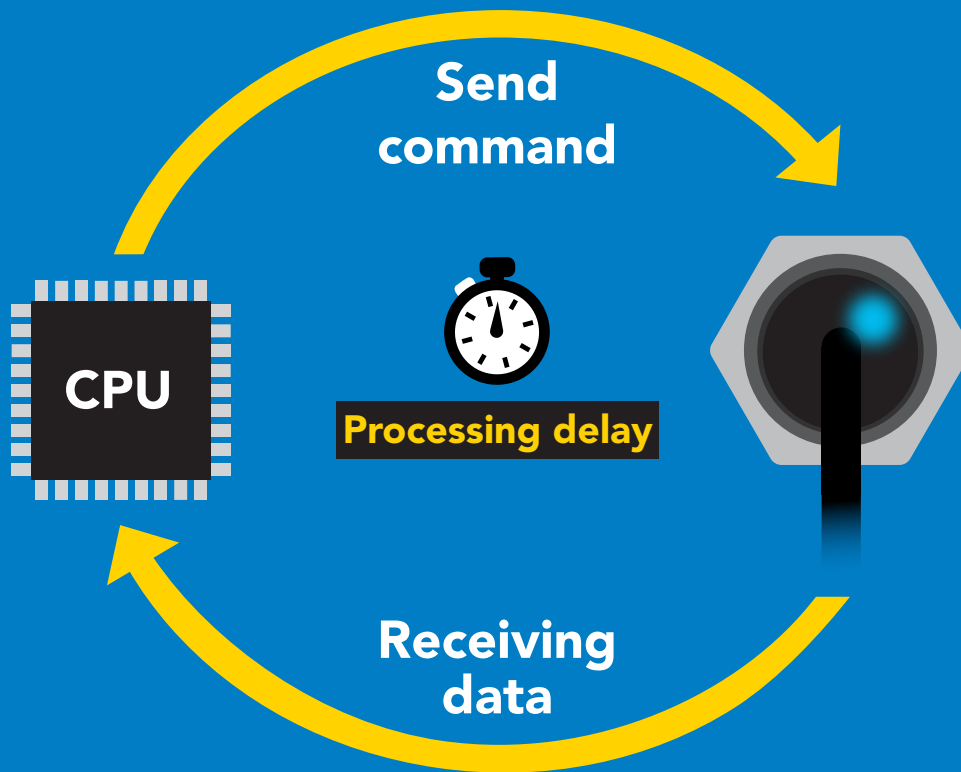
1 52 49 51 0 = 413

Dec ASCII Dec

Response codes & processing delay

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;
```

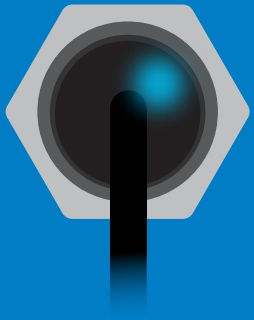
If there is no processing delay or the processing delay is too short, the response code will always be 254.

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



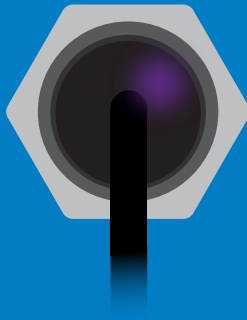
Blue

I²C standby



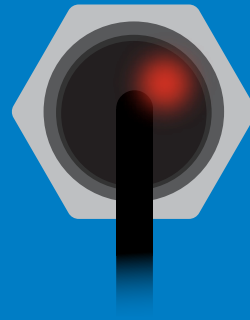
Green

Taking reading



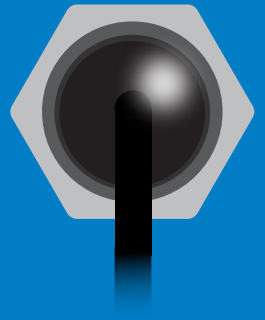
Purple

Changing
I²C address



Red

Command
not understood



White

Find

5V

LED ON
+2.5 mA

3.3V

+1 mA

I²C mode

command quick reference

All commands are ASCII strings or single ASCII characters.

Command	Function	
Alarm	enable/disable alarm	pg. 44
Baud	switch back to UART mode	pg. 54
Cal	performs custom calibration	pg. 45
Export/import	export/import calibration	pg. 46
Factory	enable factory reset	pg. 53
Find	finds device with blinking white LED	pg. 42
i	device information	pg. 48
I2C	change I ² C address	pg. 52
L	enable/disable LED	pg. 41
O	enable/disable internal temp	pg. 47
Plock	enable/disable protocol lock	pg. 51
R	returns a single reading	pg. 43
Sleep	enter sleep mode/low power	pg. 50
Status	retrieve status information	pg. 49

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

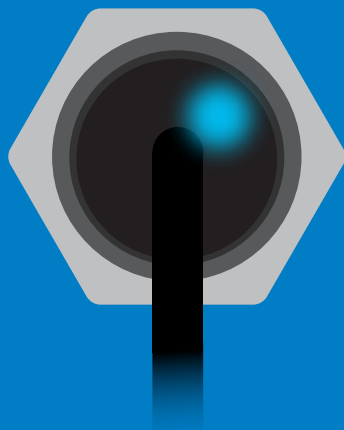
 **Wait 300ms** **1** **0**
Dec Null

L,0

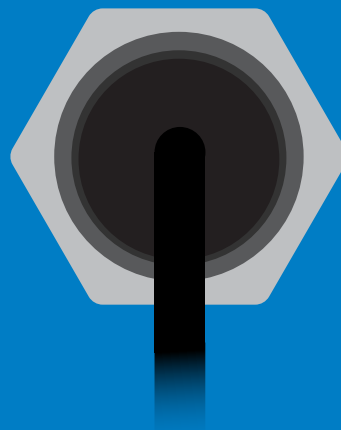
 **Wait 300ms** **1** **0**
Dec Null

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

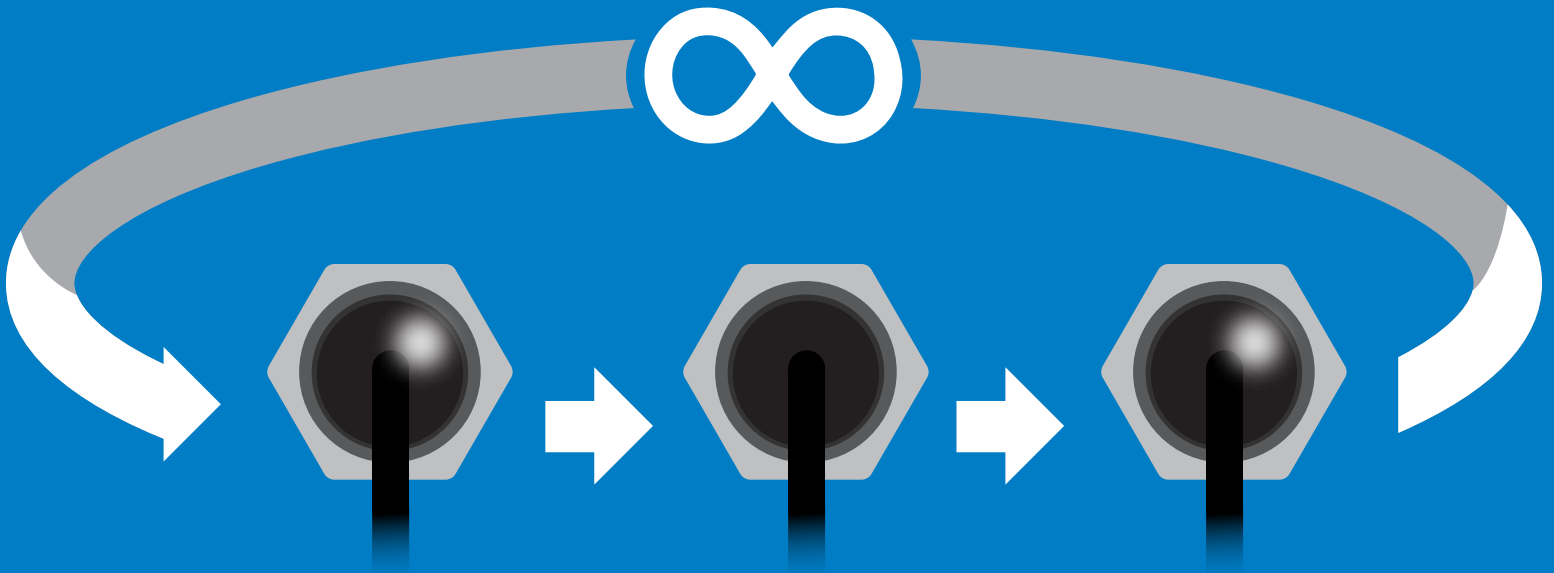
Find LED rapidly blinks white, used to help find device

Example

Response

Find

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



Taking reading

Command syntax

900ms  processing delay

R return 1 reading

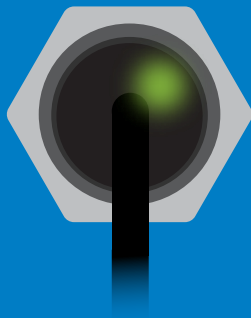
Example

Response

R

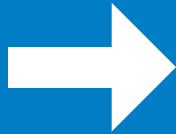
 Wait 900ms

1	800	0
Dec	ASCII	Null

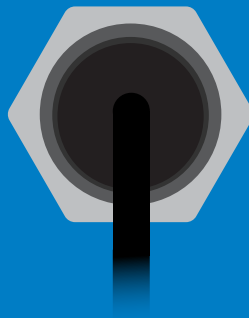


Green

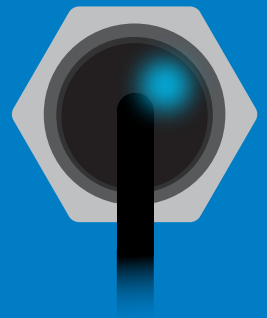
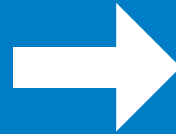
Taking reading



Wait 900ms



Transmitting



Cyan

Standby

Alarm

300ms  processing delay

Command syntax

The alarm pin will = 1 when CO2 levels are > alarm set point. Alarm tolerance sets how far below the set point CO2 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 (0 - 500 ppm)
Alarm,?	alarm set?

Example

Response

Alarm,en,1

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

Alarm,1200

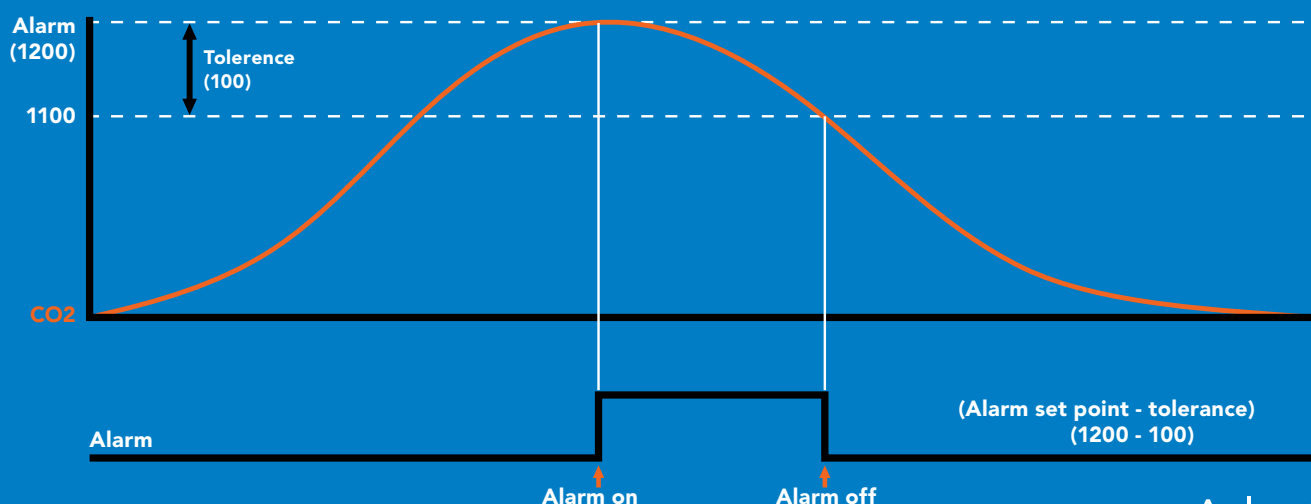
 **1** **0**
Wait 300ms Dec Null

Alarm,tol,100

 **1** **0** **CO2 level must fall 100 ppm below set point for alarm to reset.**
Wait 300ms Dec Null

Alarm,?

 **1** **?,alarm,1200,100** **0** **if all are enabled**
Wait 300ms Dec ASCII Null



Custom calibration

900ms  processing delay

Command syntax

High point calibration can be from 3,000 ppm to 5,000 ppm. Calibration outside of that range may lead to accuracy issues.

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,3900


Wait 900ms 1 0
Dec Null


Cal,0


Wait 900ms 1 0
Dec Null

Cal,clear


Wait 300ms 1 0
Dec Null

Cal,?


Wait 300ms 1 ?Cal,0 0 or 1 ?Cal,1 0
Dec ASCII Null Dec ASCII Null
no calibration only zero point calibration

or 1 ?Cal,2 0 or 1 ?Cal,3 0
Dec ASCII Null Dec ASCII Null
only high point calibration zero and high point calibration

This device comes pre-calibrated.

Custom calibration should not be performed without scientific grade calibration gasses.

Export/import calibration

Command syntax

Export: Use this command to save calibration settings
Import: Use this command to load calibration settings

Export	export calibration string from calibrated device
Import	import calibration string to new device
Export,?	calibration string info

300ms  processing delay

Example

Response

Export,?



1	10,120	0
Dec	ASCII	Null

Response breakdown

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

Export strings can be up to 12 characters long

Export

(8 more)



1	59 6F 75 20 61 72	0
Dec	ASCII	Null

(1 of 10)

⋮

Export



1	65 20 61 20 63 6F	0
Dec	ASCII	Null

(10 of 10)

Export



1	*DONE	0
Dec	ASCII	Null

Import, n
(FIFO)

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

Enable/disable internal temperature from output string

Command syntax

300ms  processing delay

O,t,[1,0] enable or disable internal temperature

Example

Response

O,t,1

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

enable temperature

O,t,0

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

disable temperature

O,?

 **Wait 300ms** **1** **?O,ppm,t** **0**
Dec ASCII Null

if internal temp
is enabled

Enabling the internal temperature should only be used to confirm that the device is at thermal equilibrium. Refer to page 6

Device information

Command syntax

300ms  processing delay

i device information

Example

i

Response



Wait 300ms

1

Dec

?i,CO2,1.00

ASCII

0

Null

Response breakdown

?i, CO2, 1.00
 ↑ ↑
 Device 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,

↑
Reason for restart

5.038

↑
Voltage at Vcc

Restart codes

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

Sleep mode/low power

Command syntax

Sleep enter sleep mode/low power

Send any character or command to awaken device.

Example

Response

Sleep

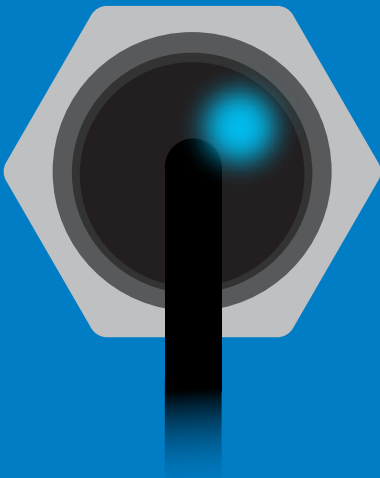
no response

Do not read status byte after issuing sleep command.

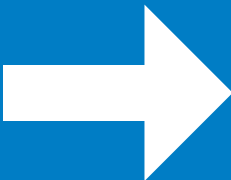
Any command

wakes up device

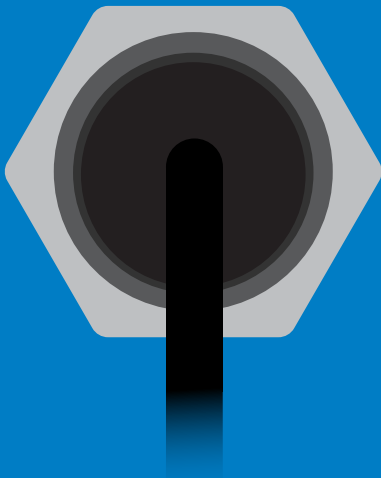
5V	STANDBY	SLEEP
	45 mA	3.4 mA
3.3V	42 mA	3.0 mA



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 I²C mode.

default

Example

Response

Plock,1

 Wait 300ms
1 0
Dec Null

Plock,0

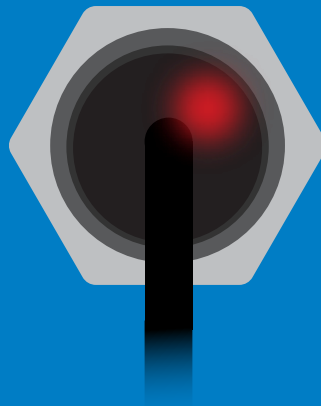
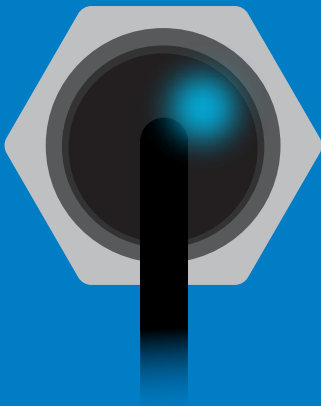
 Wait 300ms
1 0
Dec Null

Plock,?

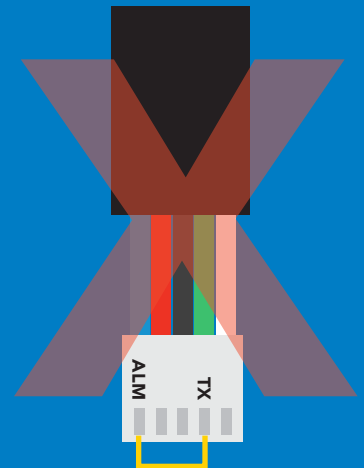
 Wait 300ms
1 ?Plock,1 0
Dec ASCII Null

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

Response

I2C,101

device reboot

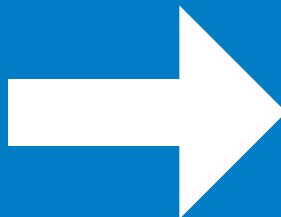
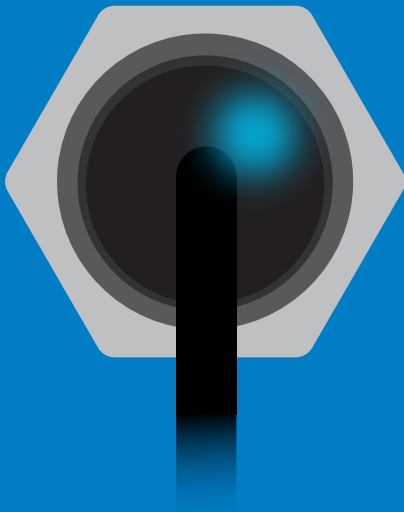
Warning!

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

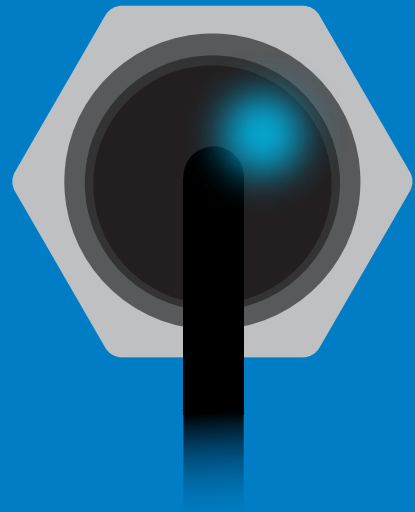
Default I²C address is 105 (0x69).

n = any number 1 – 127

I2C,101



(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

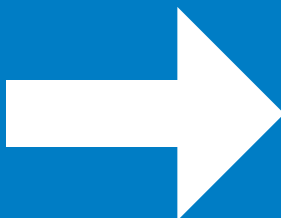
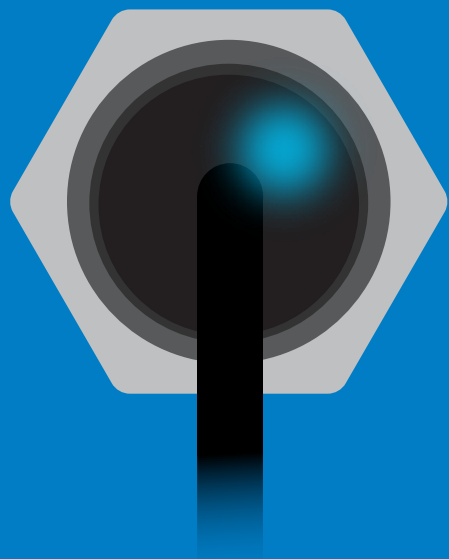
Response

Factory

device reboot

Clears custom calibration
LED on
Response codes enabled

Factory



(reboot)



Change to UART mode

Command syntax

Baud,n switch from I²C to UART

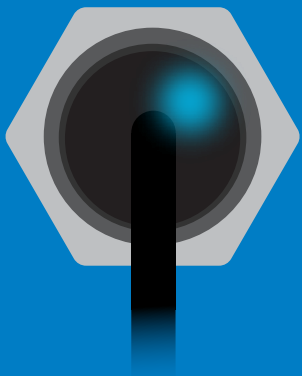
Example

Baud,9600

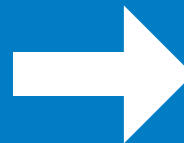
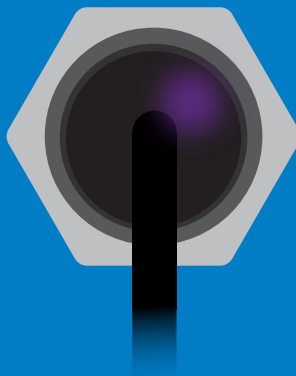
Response

reboot in UART mode

n = [300
1200
2400
9600
19200
38400
57600
115200



Baud,9600



(reboot)

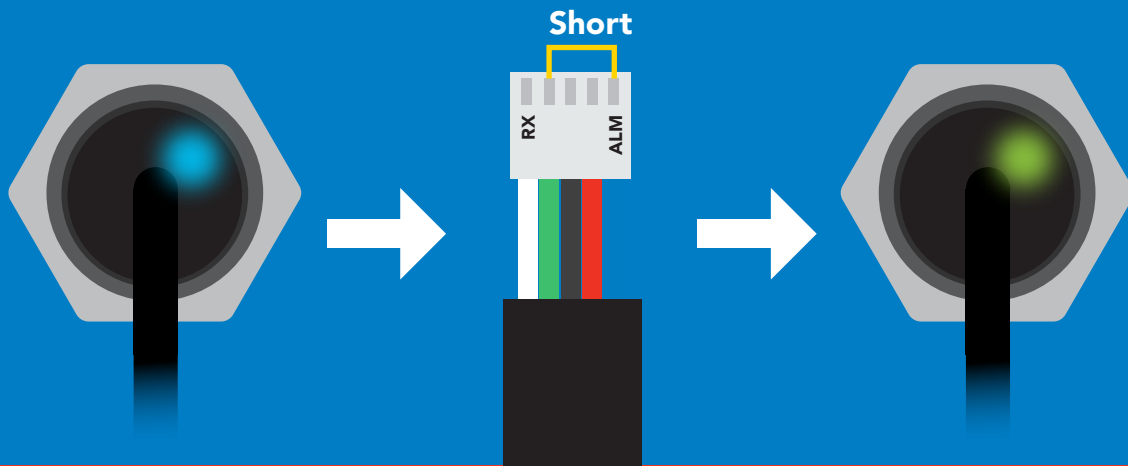


Changing to
UART mode

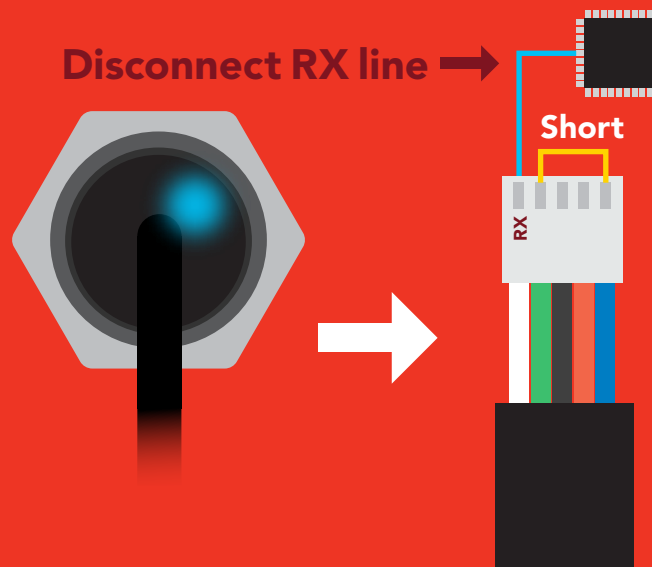
Manual switching to UART

- Make sure Plock is set to 0
- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to ALM
- 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



Datasheet change log

Datasheet V 1.0

New datasheet

Firmware updates

V1.0 – Initial release (Sept 12, 2018)

Warranty

Atlas Scientific™ Warranties the EZO-CO2™ Embedded NDIR CO2 Sensor to be free of defect during the debugging phase of device implementation, or 30 days after receiving the EZO-CO2™ Embedded NDIR CO2 Sensor (which ever comes first).

The debugging phase

The debugging phase as defined by Atlas Scientific™ is the time period when the EZO-CO2™ Embedded NDIR CO2 Sensor is connected into a bread board, or shield. If the EZO-CO2™ Embedded NDIR CO2 Sensor is being debugged in a bread board, the bread board must be devoid of other components. If the EZO-CO2™ Embedded NDIR CO2 Sensor is being connected to a microcontroller, the microcontroller must be running code that has been designed to drive the EZO-CO2™ Embedded NDIR CO2 Sensor exclusively and output the EZO-CO2™ Embedded NDIR CO2 Sensor 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-CO2™ Embedded NDIR CO2 Sensor warranty:

- **Soldering any part to the EZO-CO2™ Embedded NDIR CO2 Sensor.**
- **Running any code, that does not exclusively drive the EZO-CO2™ Embedded NDIR CO2 Sensor and output its data in a serial string.**
- **Embedding the EZO-CO2™ Embedded NDIR CO2 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-CO2™ Embedded NDIR CO2 Sensor, against the thousands of possible variables that may cause the EZO-CO2™ Embedded NDIR CO2 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-CO2™ Embedded NDIR CO2 Sensor continued operation. This is because that would be equivalent to Atlas Scientific™ taking responsibility over the correct operation of your entire device.