

# EZO-PMP™

## Embedded Dosing Pump

Flow rate **0.5ml to 105ml/min**

Accuracy **+/- 1%**

Viscosity **0.1 – 2,000 cP**

Modes of operation  
**Continuous dispensing**  
**Volume dispensing**  
**Dose over time**  
**Constant flow rate**  
**Dispense at startup**

Connector **5 lead data cable**

Calibration **Single point**

Tubing size **Any 5mm O.D. tubing**

Data protocol **UART & I<sup>2</sup>C**

Default I<sup>2</sup>C address **103 (0x67)**

Operating voltage  
**3.3V – 5V (logic)**  
**12V – 24V (motor)**

Pump head **8.1 meters (26.5')**

Data format **ASCII**

Food Safe **Yes**



# Table of contents

EZO-PMP™ dimensions	4	Operating modes	6
EZO-PMP™ tubing	5	Available data protocols	8
Operating principle	6	Default state	9

## UART

UART mode	11
Receiving data from device	12
Sending commands to device	13
LED color definition	14
<b>UART quick command page</b>	<b>15</b>
LED control	16
Find	17
Continuous mode	18
Single reading mode	19
Continuous dispensing	20
Volume dispensing	21
Dose over time	22
Constant flow rate	23
Dispense at startup	24
Pause dispensing	27
Stop dispensing	28
Invert dispensing direction	29
Total volume dispensed	30
Calibration	31
Enable/disable parameters	32
Pump voltage	33
Naming device	34
Device information	35
Response codes	36
Reading device status	37
Sleep mode/low power	38
Change baud rate	39
Protocol lock	40
Factory reset	41
Change to I <sup>2</sup> C mode	42
Manual switching to I <sup>2</sup> C	43

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

I <sup>2</sup> C mode	45
Sending commands	46
Requesting data	47
Response codes	48
LED color definition	49
<b>I<sup>2</sup>C quick command page</b>	<b>50</b>
LED control	51
Find	52
Single report mode	53
Continuous dispensing	54
Volume dispensing	55
Dose over time	56
Constant flow rate	57
Dispense at startup	58
Pause dispensing	61
Stop dispensing	62
Invert dispensing direction	63
Total volume dispensed	64
Calibration	65
Enable/disable parameters	66
Pump voltage	67
Naming device	68
Device information	69
Reading device status	70
Sleep mode/low power	71
Protocol lock	72
I <sup>2</sup> C address change	73
Factory reset	74
Change to UART mode	75
Manual switching to UART	76

Calibration theory	77
Accuracy	82
Viscosity	83

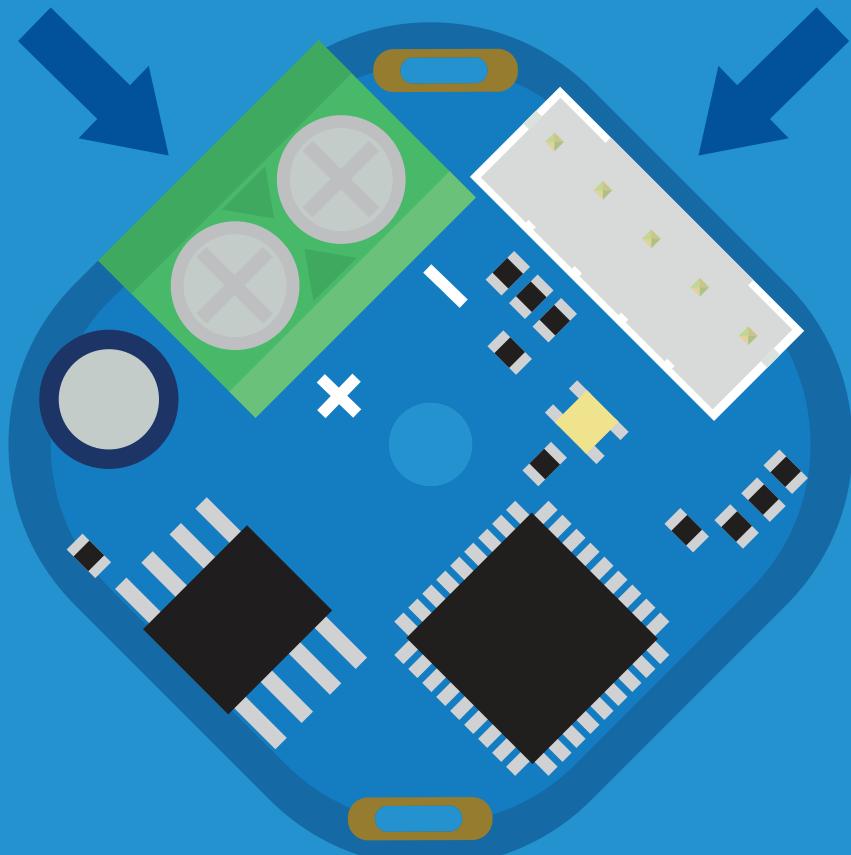
Mounting the EZO-PMP™	84
Datasheet change log	85
Warranty	87

# Attention

The EZO-PMP™ Embedded Dosing Pump requires two power supplies to operate.

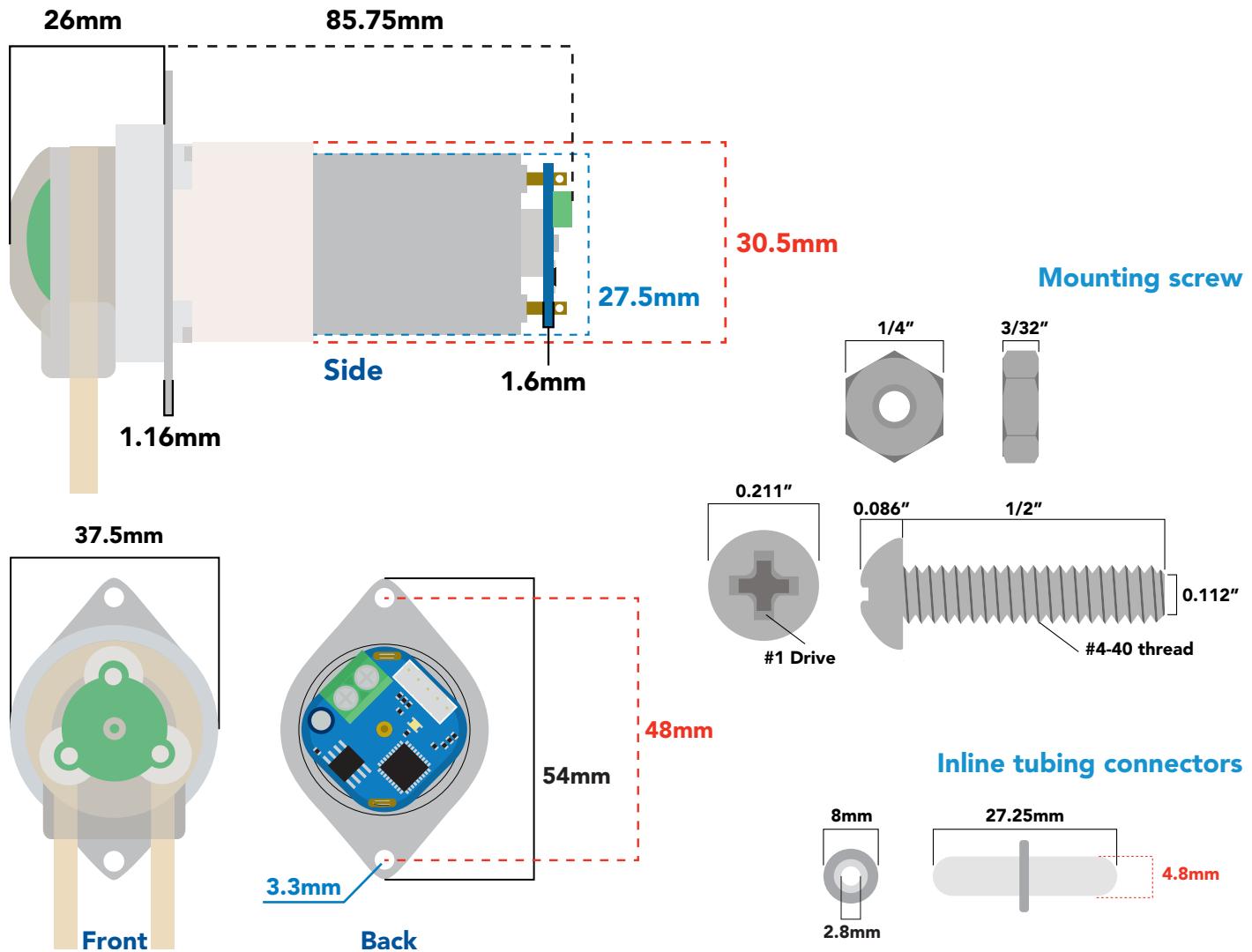
**12V – 24V**  
to drive the motor

**3.3V – 5.5V**  
for the control system



**Control system**  
(Back side of dosing pump)

# EZO-PMP™ dimensions



## Power consumption

	LED	MAX	STANDBY	SLEEP
<b>5V</b>	ON	13.7 mA	13.4 mA	0.415 mA
	OFF	13.1 mA	12.8 mA	
<b>3.3V</b>	ON	12.5 mA	12.4 mA	0.13 mA
	OFF	12.3 mA	12.2 mA	
<b>Motor</b>	12V = ~400mA	24V = ~200mA		

Tubing life span	+1,000 hrs.
Cassette life span	1,500 hrs.
Motor life span	5,000 hrs.

## Absolute max ratings

Parameter	MIN	TYP	MAX
Storage temperature (EZO-PMP™)	-65 °C		125 °C
Operational temperature (EZO-PMP™)	-40 °C	25 °C	85 °C
VCC	3.3V	5V	5.5V
Motor	10.8V	12V	24V
Max input / output pressure			80 kPa

# EZO-PMP™ tubing

## NSF/ANSI 51 Compliant

### Tan tubing

Saint-Gobain™ PharMed™ BPT tubing

Length: 15.24cm

Outer diameter: 5mm

Inner diameter: 3mm

This tubing is highly chemically resistant and has 30X more resistance to mechanical wear than silicone tubing.



### Inline tubing connectors

HDPE

Length: 2.54cm

Outer diameter: 8mm

Inner diameter: 2.8mm



### Blue tubing

Silicone

Length: 2x 30.48cm

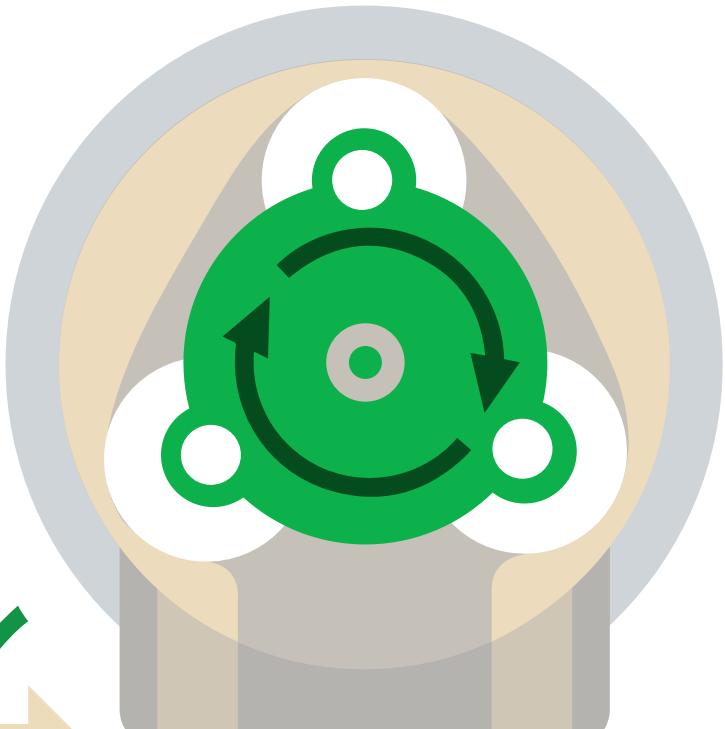
Outer diameter: 5mm

Inner diameter: 3mm

Bend radius: 15mm

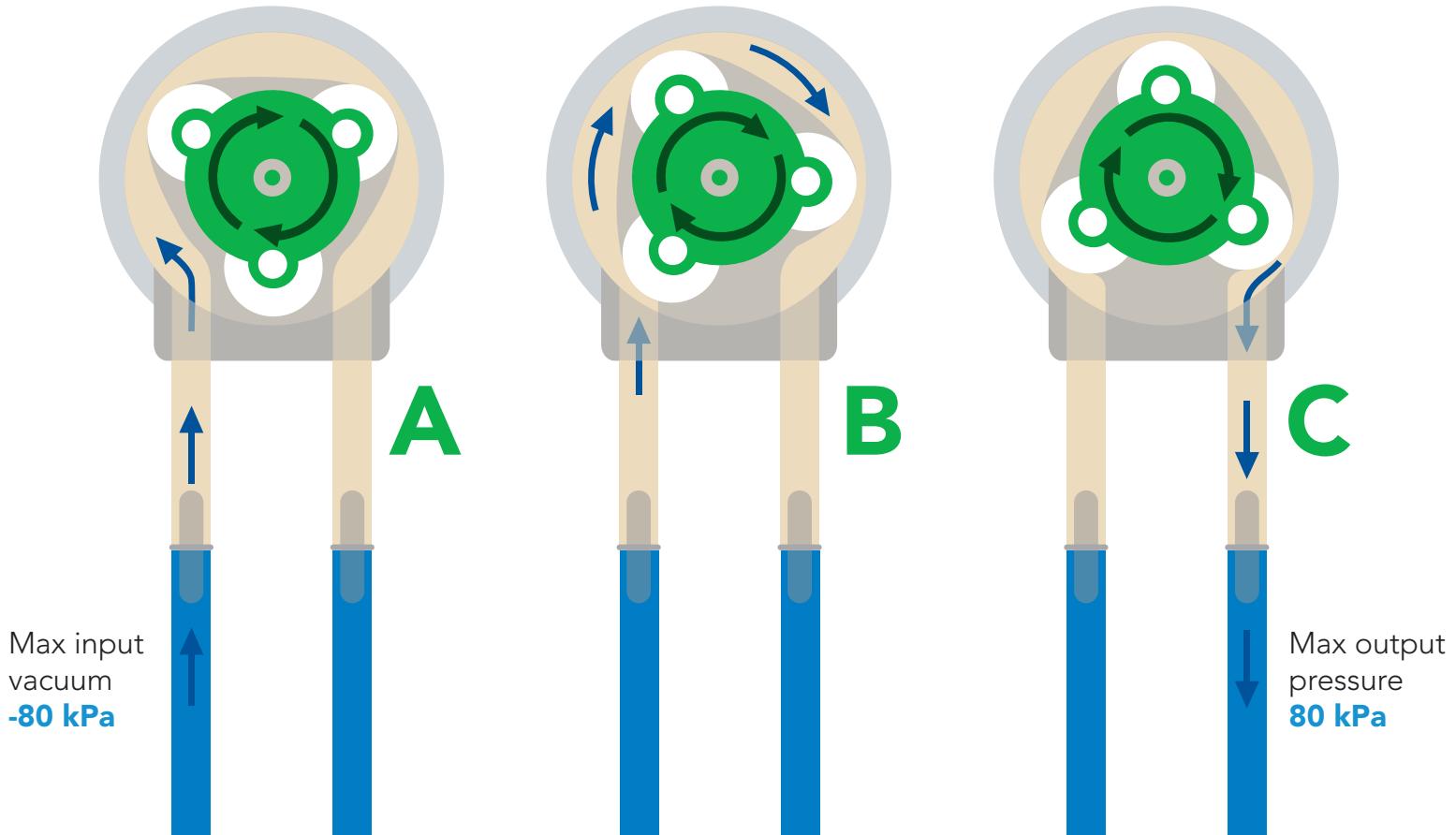
Temperature -67°C to 200°C

Max pressure: 69 kPa (10 PSI)



# Operating principle

- ✓ Self-priming
- ✓ Run dry



## Operating modes

The EZO-PMP™ can operate in four different modes.

### Continuous dispensing

Run the pump continuously  
105 ml/min  $\infty$  (with supplied tubing)

### Volume dispensing

Pump a specific volume  
(Smallest possible volume is 0.5 ml)

**Volume is always in ml.**

### Dose over time

Pump a specific volume over a set time

### Constant flow rate

Pump a specific volume per minute

### Dispense at startup

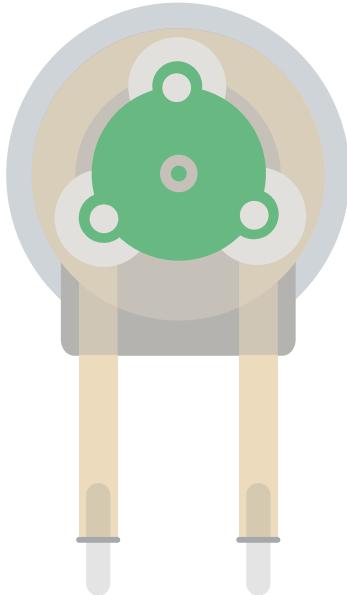
- Dispense a specific volume at startup
- Continuous dispensing at startup
- Dose over time at startup

## This device requires two power supplies

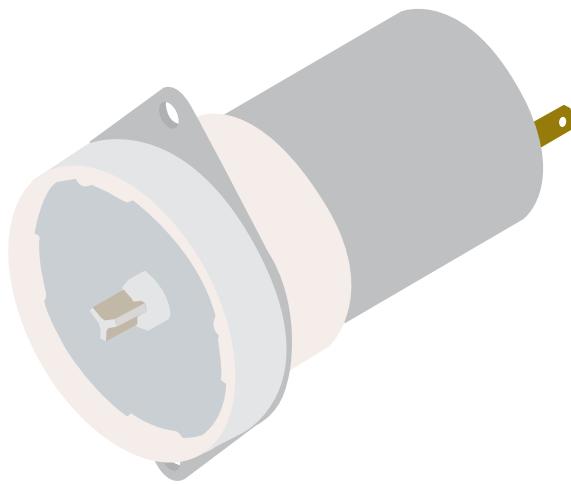
3.3V–5.5V for the control system

12V–24V to drive the motor

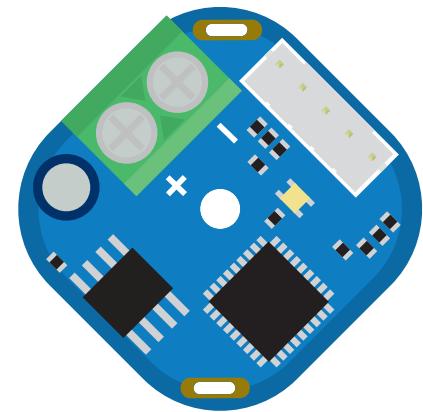
The Atlas Scientific EZO-PMP™ consists of three main components.



**Cassette**



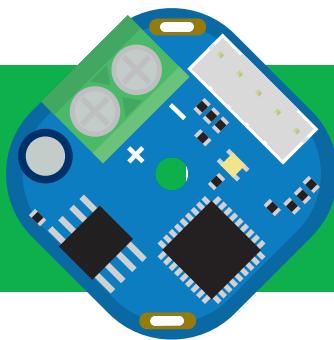
**12 volt motor**



**Control system**

The actual peristaltic pumping is done within the cassette. It has been designed to be easily detached from the motor and disassembled.

The 12 volt motor and control system have been soldered together. Both components are designed to operate as one single unit.



### The control system has three main components

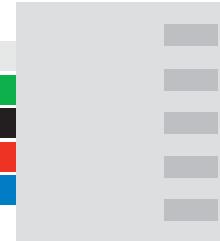
Keyed data and power connector

12–24 volt power input

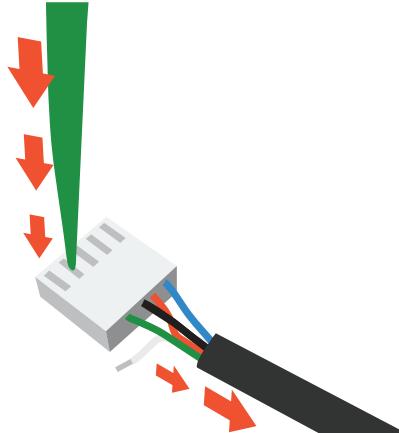
Status indicator LED

### Data and power cable pinout

White	- RX/SCL
Green	- TX/SDA
Black	- GND
Red	- VCC
Blue	- INT



Should you need to remove this connector from the data cable, follow the provided illustration.



 Available data protocols

**UART**

**Default**

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

 Unavailable data protocols

**SPI**

**Analog**

**RS-485**

**Mod Bus**

**4–20mA**

# Default state

# UART mode

Baud

9,600

Readings

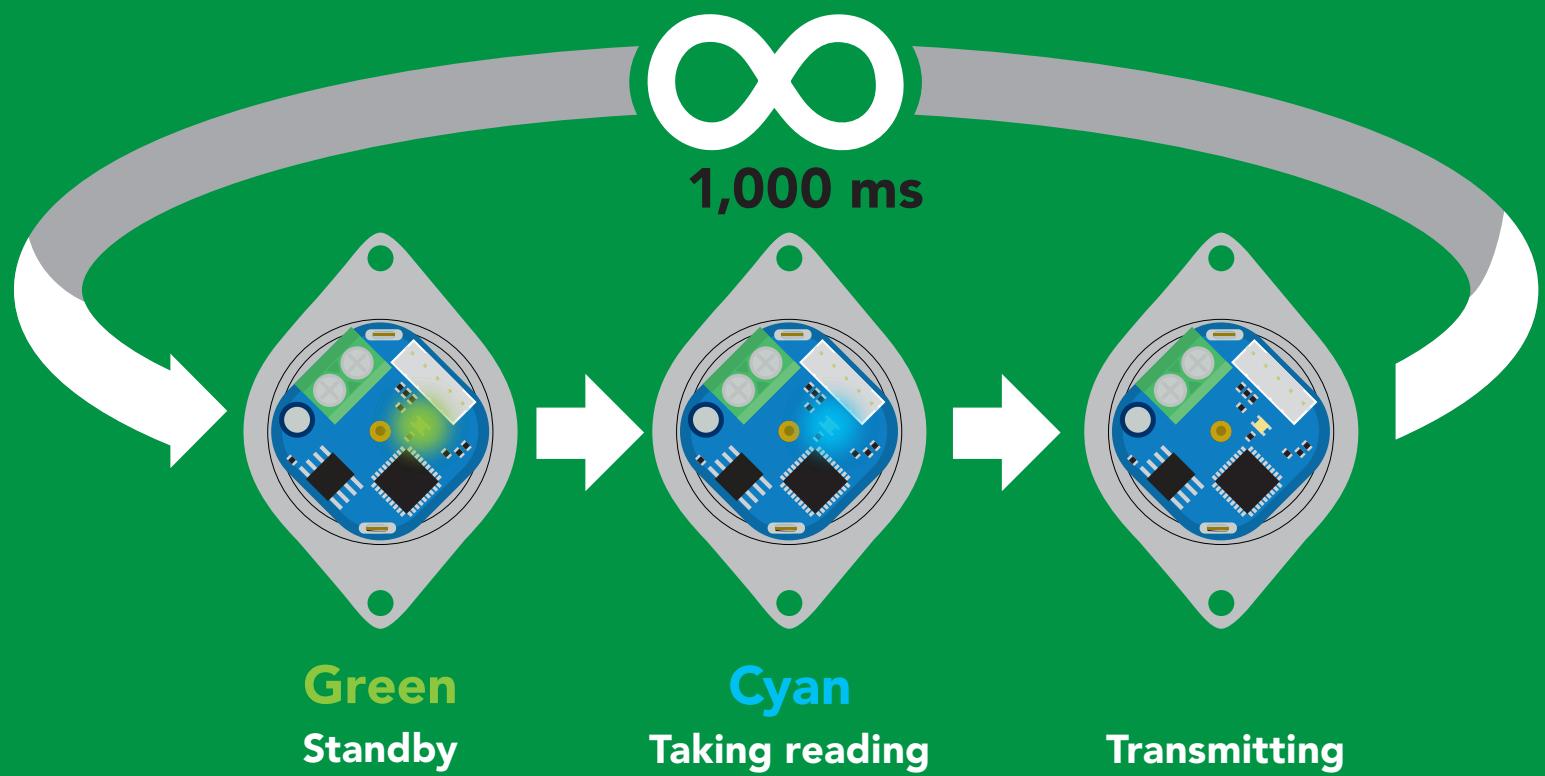
continuous

Speed

1 reading per second

LED

on



# UART mode

## Settings that are retained if power is cut

Baud rate  
Calibration  
Continuous mode  
Device name  
Enable/disable parameters  
Enable/disable response codes  
Hardware switch to I<sup>2</sup>C mode  
Invert  
LED control  
Protocol lock  
Software switch to I<sup>2</sup>C mode

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

Absolute total volume  
Find  
Sleep mode  
Total volume

# UART mode

8 data bits      no parity  
1 stop bit      no flow control

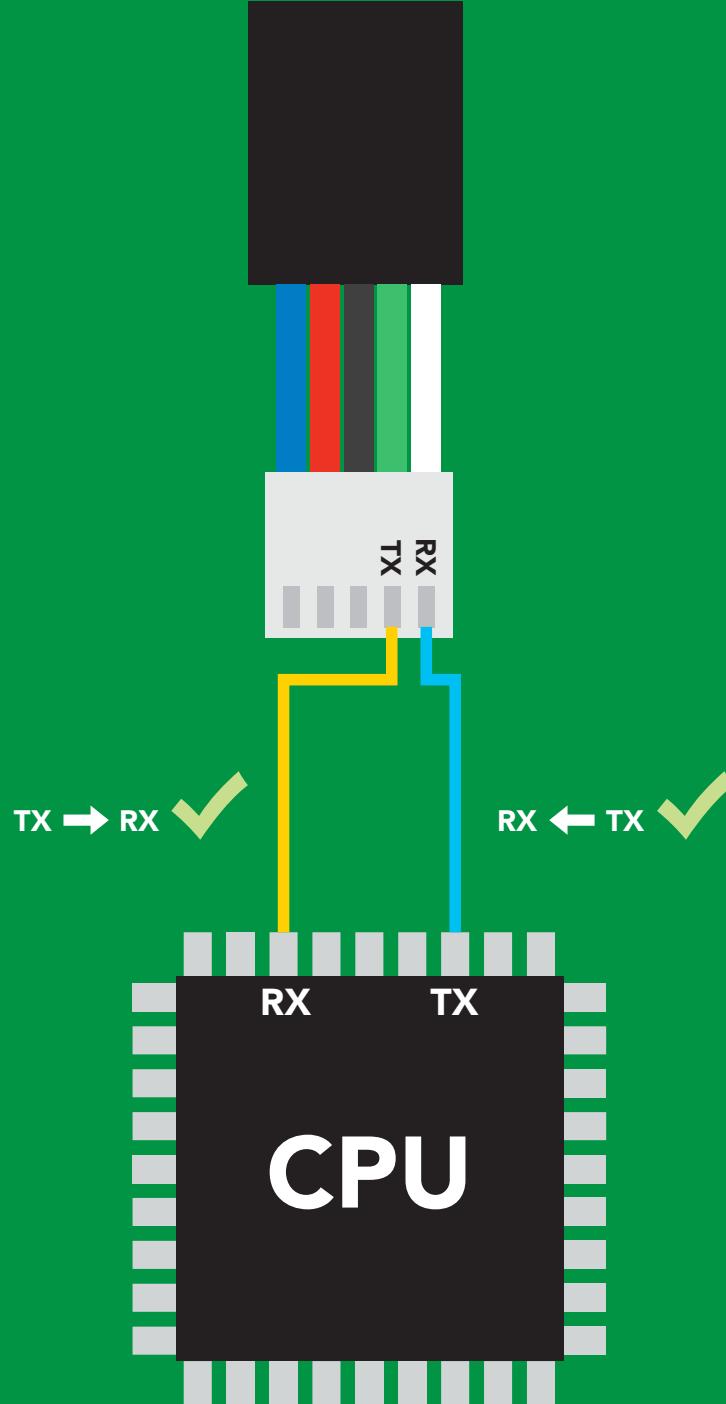
Baud    300  
1,200  
2,400  
**9,600 default**  
19,200  
38,400  
57,600  
115,200

**RX**      Data in  


**TX**      Data out  


**Vcc**      3.3V – 5.5V

 VCC  
0V                  0V



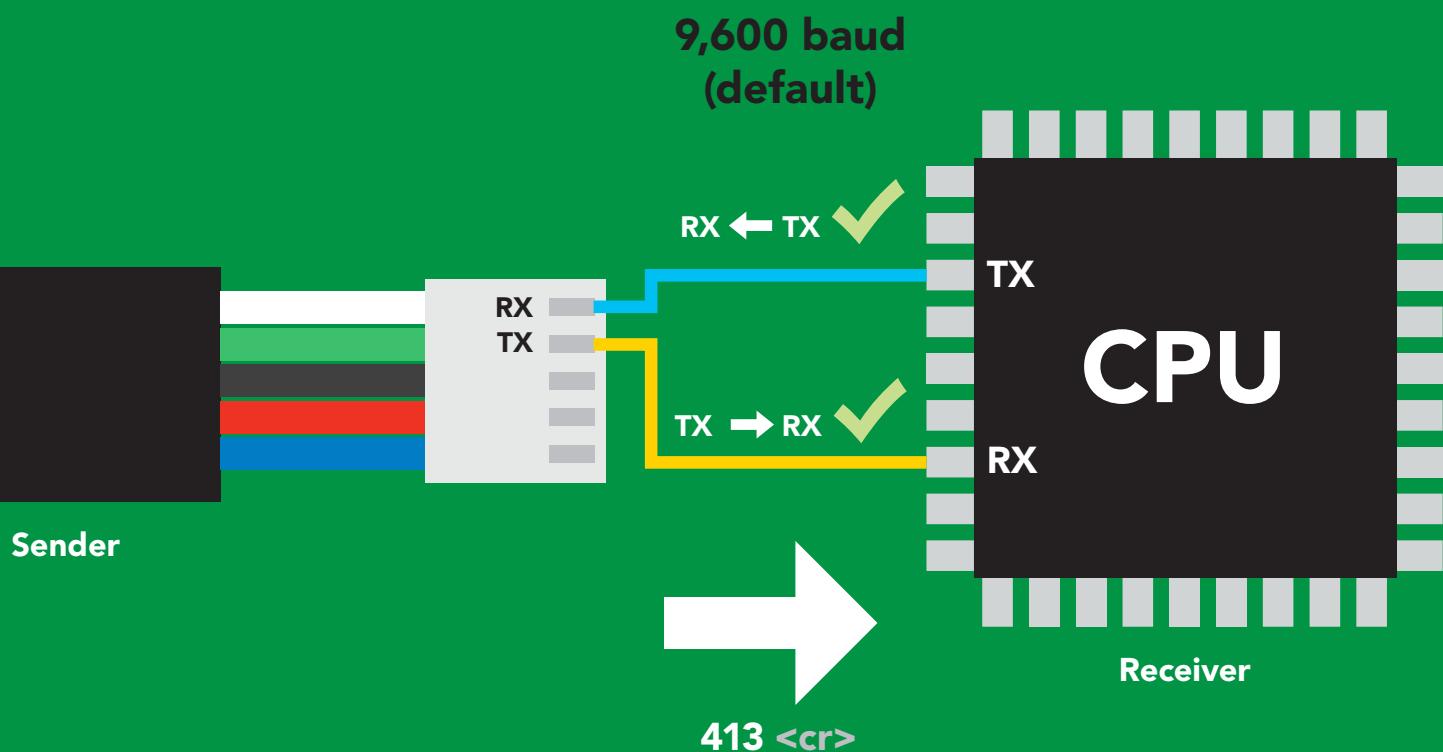
## Data format

**Output**      volume  
**Units**          ml  
**Encoding**       ASCII  
**Format**          string

**Terminator**      carriage return  
**Data type**       floating point  
**Decimal places**   2  
**Smallest string**   3 characters  
**Largest string**   39 characters

# Receiving data from device

2 parts

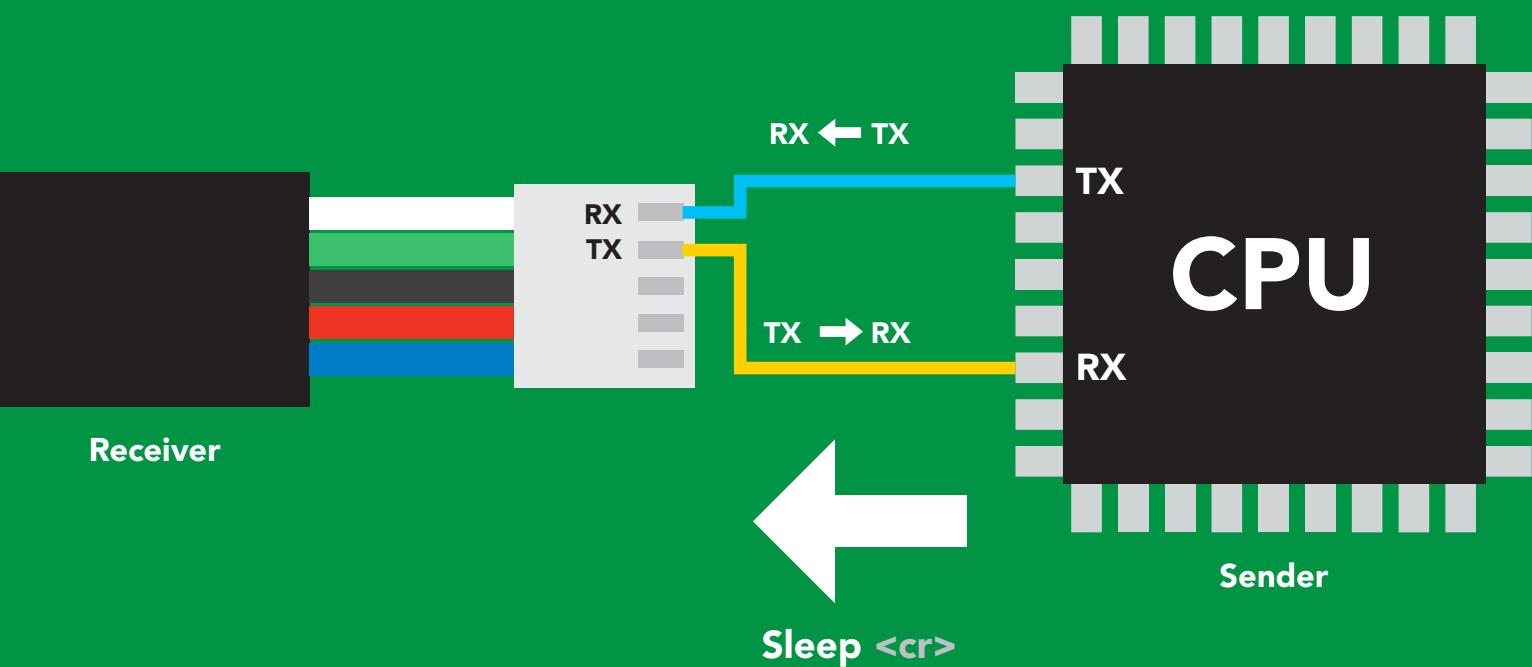


## Advanced

ASCII:	4	1	3	<cr>
Hex:	34	31	33	0D
Dec:	52	49	51	13

# Sending commands to device

2 parts



## Advanced

ASCII: S I e e p <cr>

Hex: 53 6C 65 65 70 0D

Dec: 83 108 101 101 112 13

# LED color definition



**Green**  
UART standby



**Cyan**  
Taking reading



**Purple**  
Changing baud rate



**Red**  
Command not understood



**White**  
Find

<b>5V</b>	LED ON <b>+2.5 mA</b>
<b>3.3V</b>	<b>+1 mA</b>

# UART mode

## command quick reference

All commands are ASCII strings or single ASCII characters.

Command	Function	Default state
Baud	change baud rate	pg. 39 9,600
C	enable/disable continuous mode	pg. 18 enabled
Cal	performs calibration	pg. 31 n/a
D	dispense modes	pg. 20 – 26 n/a
Factory	enable factory reset	pg. 41 n/a
Find	finds device with blinking white LED	pg. 17 n/a
i	device information	pg. 35 n/a
Invert	invert dispensing direction	pg. 29 n/a
I2C	change to I <sup>2</sup> C mode	pg. 42 not set
L	enable/disable LED	pg. 16 enabled
Name	set/show name of device	pg. 34 not set
O	enable/disable parameters	pg. 32 all enabled
P	pause dispensing	pg. 27 n/a
Plock	enable/disable protocol lock	pg. 40 disabled
Pv	check pump voltage	pg. 33 n/a
R	returns a single reading	pg. 19 n/a
Sleep	enter sleep mode/low power	pg. 38 n/a
Status	retrieve status information	pg. 37 enable
Tv	total volume dispensed	pg. 30 n/a
X	stop dispensing	pg. 28 n/a
*OK	enable/disable response codes	pg. 36 enable

# 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>

L,0 <cr>

L,? <cr>

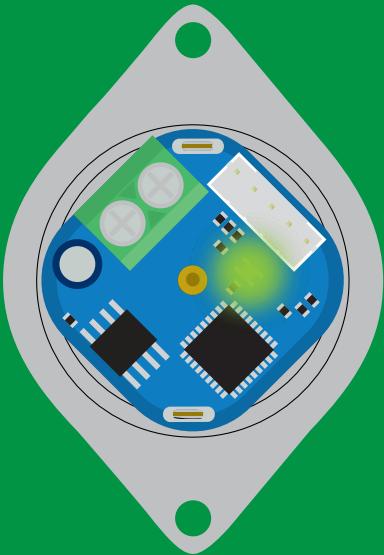
### Response

\*OK <cr>

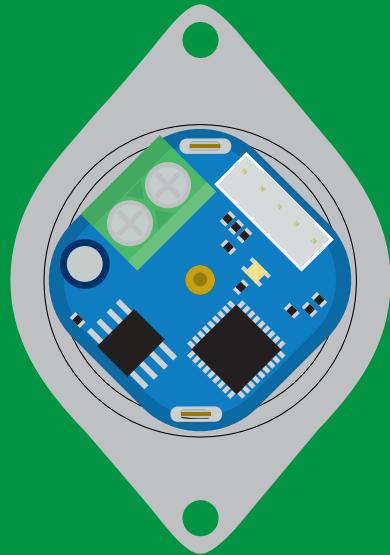
\*OK <cr>

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

\*OK <cr>



L,1



L,0

# Find

## Command syntax

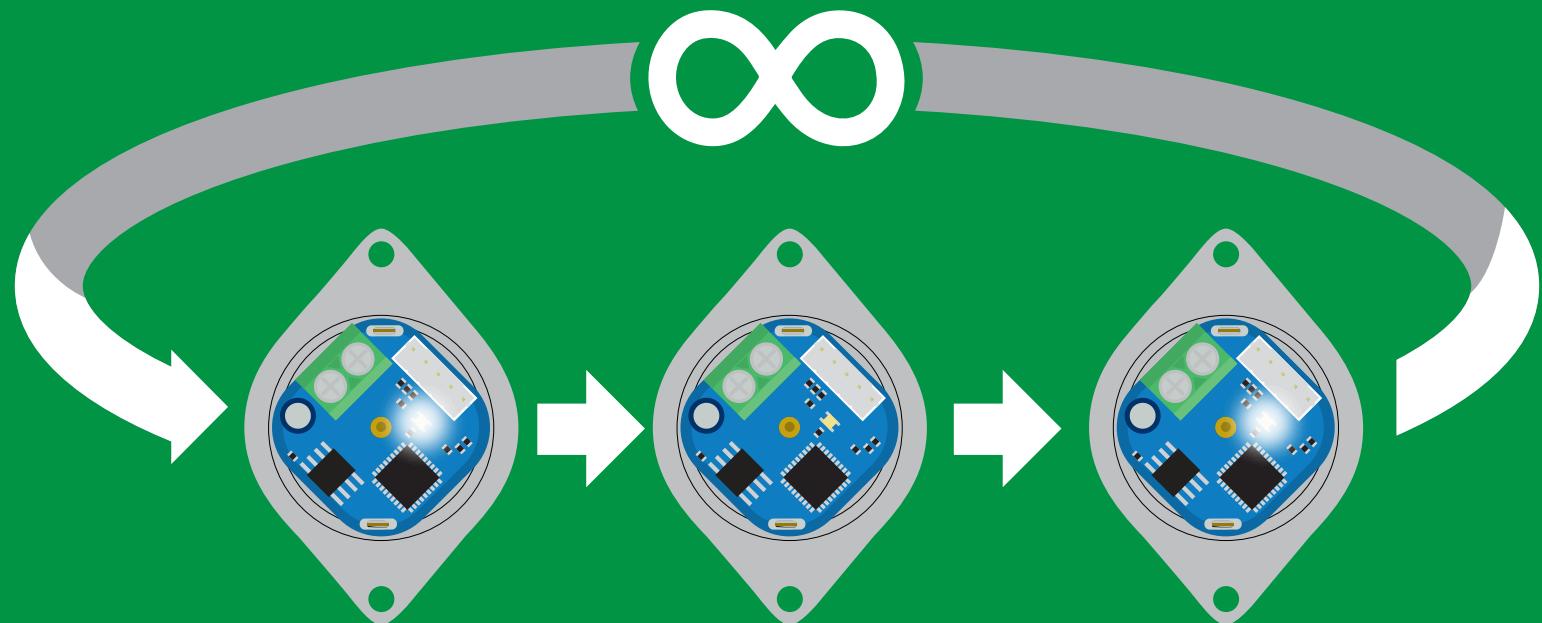
This command will disable continuous mode  
Send any character or command to terminate find.

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

## Example Response

Find <cr>

\*OK <cr>



# Continuous mode

## Command syntax

- C,\* <cr> continuously reports volume once per second **default**  
C,1 <cr> continuously reports volume only when pumping  
C,0 <cr> disable continuous reporting  
C,? <cr> continuous reporting mode on/off?

## Example      Response

dispense 3ml

C,\* <cr>

1.2 <cr>  
3.0 <cr>  
**\*Done,3.00 <cr>**  
3.0 <cr>  
3.0 <cr>

C,1 <cr>

1.2 <cr>  
3.0 <cr>  
**\*Done,3.00 <cr>**

C,0 <cr>

**\*Done,3.00 <cr>**

C,? <cr>

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

# Single reading mode

## Command syntax

R <cr> returns a single value showing dispensed volume

### Example      Response

R <cr>

2.50 <cr> (If issued half way through dispensing 5ml)  
\*OK <cr>

5.00 <cr> (If issued once dispensing has stopped)  
\*OK <cr>

# Continuous dispensing

Pump on/pump off

## Command syntax

After running in continuous mode for 20 days the EZO-PMP™ will reset.

D,\* <cr> dispense until the stop command is given

D,-\* <cr> dispense in reverse until the stop command is given

D,? <cr> dispense status

## Example Response

D,* <cr>	*OK <cr> pump will continuously run at ~105ml/min (with supplied tubing)
D,-* <cr>	*OK <cr> pump will continuously run in reverse at ~105ml/min (with supplied tubing)
D,? <cr>	?D,*,1 <cr> *OK <cr>

## Response breakdown

?D,\*,1

↑      ↑  
last volume    pump on  
requested

# Volume dispensing

Pump a specific volume

## Command syntax

where [ml] is any volume in millimeters  $\geq 0.5$

D,[ml] <cr> dispense [this specific volume]

D,[-ml] <cr> dispense [in reverse this specific volume]

D,? <cr> dispense status

## Example

## Response

D,15 <cr>

\*OK <cr> 15 ml will be dispensed

D,-40.5 <cr>

\*OK <cr> 40.5 ml will be dispensed *in reverse*

D,? <cr>

?D,-40.50,0 <cr>

\*OK <cr>

## Response breakdown

?D,-40.50,0

↑  
last volume  
dispensed      ↑  
pump off

# Dose over time

Pump a fixed volume over a fixed time

## Command syntax

**D,[ml],[min] <cr> Dispense [this volume], [over this many minutes]**

### Example

**D,85,10 <cr>**

### Response

**\*OK <cr> Dispense 85ml over 10 minutes**



# Constant flow rate

Maintain a constant flow rate

## Command syntax

After running in continuous mode for 20 days the EZO-PMP™ will reset.

**DC,[ml/min],[min or \*] <cr>** [maintain this rate],[for this much time]

**DC,? <cr>** reports maximum possible flow rate

[ml/min] = a single number (int or float) representing the desired flow rate

[min or \*] = the number of minutes to run or (\*) indefinitely

A negative value for ml/min = reverse

## Example Response

**DC,25,40 <cr>**

**\*OK <cr> Dispense 25ml per minute for 40 minutes**

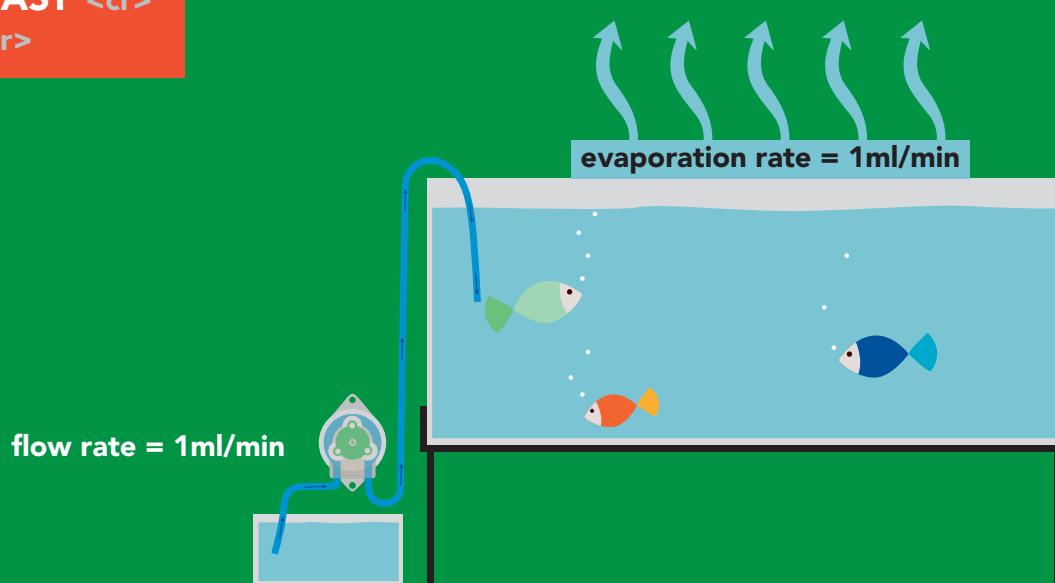
**DC,? <cr>**

**?MAXRATE,58.5 <cr>  
\*OK <cr>**

The maximum flow rate is determined after calibration.  
If the flowrate entered is too fast the EZO-PMP™ will send an error.

**\*TOOFAST <cr>**

**\*ER <cr>**



# Dispense at startup

Pump a specific volume at startup and then stop

Use this command to make a simple fixed-volume pump

## Command syntax

**Dstart,[ml]** <cr> dispense [this specific volume] at startup

**Dstart,off** <cr> disables dispense at startup mode

**Dstart,?** <cr> startup dispense status

## Example

Dstart,10 <cr>

Dstart,off <cr>

Dstart,? <cr>

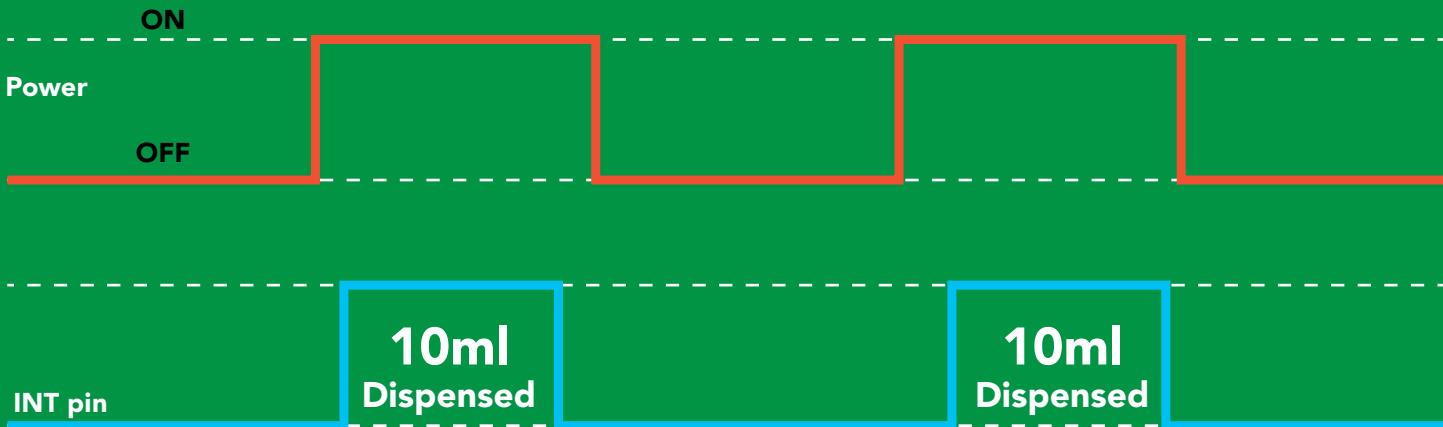
## Response

\*OK <cr>

\*OK <cr>

?Dstart,10 <cr> or ?Dstart,0 <cr>

\*OK <cr>



# Continuous dispensing at startup

Pump on & continuously dispense

## Command syntax

After running in continuous mode for 20 days the EZO-PMP™ will reset.

**Dstart,\* <cr>** dispense at startup until the stop command is given

**Dstart,-\* <cr>** dispense in reverse at startup until the stop command is given

**Dstart,? <cr>** startup dispense status

## Example

## Response

**Dstart,\* <cr>**

**\*OK <cr>**

Pump will startup and continuously run at ~105ml/min (with supplied tubing)

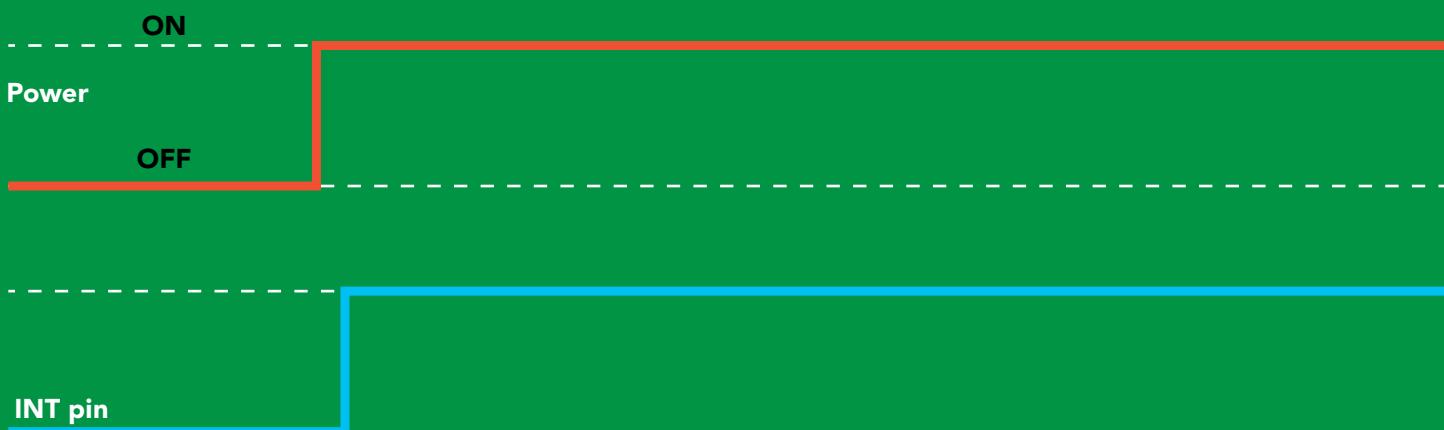
**Dstart,-\* <cr>**

**\*OK <cr>**

Pump will startup and continuously run in reverse at ~105ml/min (with supplied tubing)

**Dstart,? <cr>**

**?Dstart,\* <cr>**



# Dose Over time at startup

Pump a fixed volume over a fixed time at startup

## Command syntax

**Dstart[ml],[min] <cr>** Dispense [volume], [over this many minutes] at startup

### Example

**Dstart,85,10 <cr>**

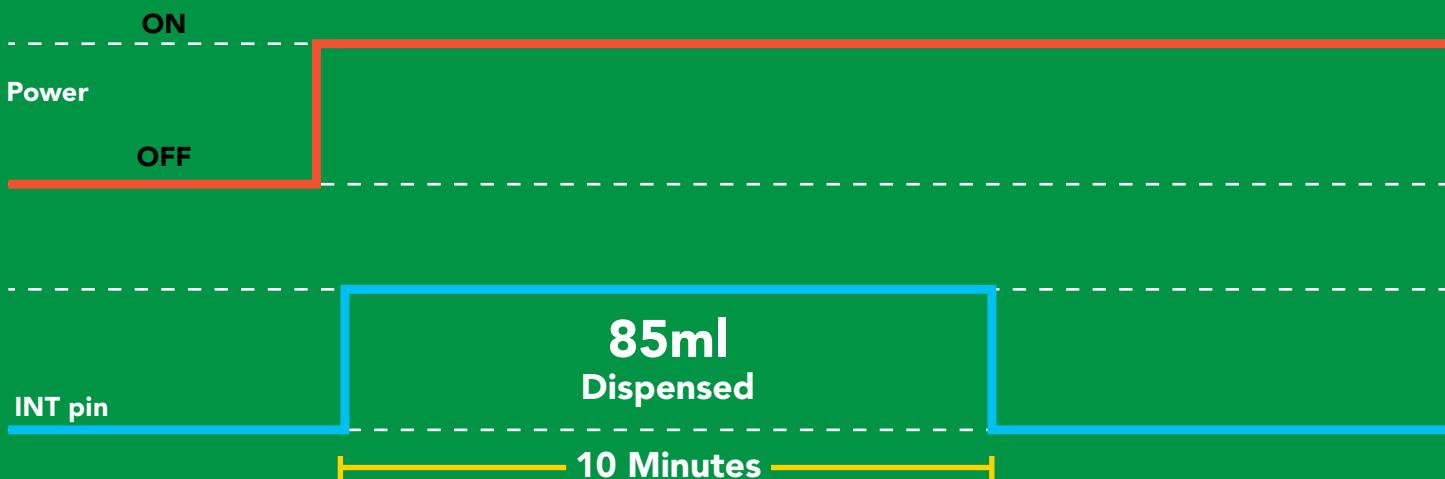
### Response

**\*OK <cr>**

Pump will startup and dispense  
85ml over 10 minutes

**Dstart,? <cr>**

**?Dstart,85.00,10.00 <cr>**



# Pause dispensing

## Command syntax

Issue the command again to resume dispensing

P <cr> pauses the pump during dispensing

P? <cr> pause status

## Example

P <cr>

\*OK <cr>

P? <cr>

?P,1 <cr> or ?P,0 <cr>  
paused  
unpaused

\*OK <cr>

P

dispensing

P

paused

dispensing

# Stop dispensing

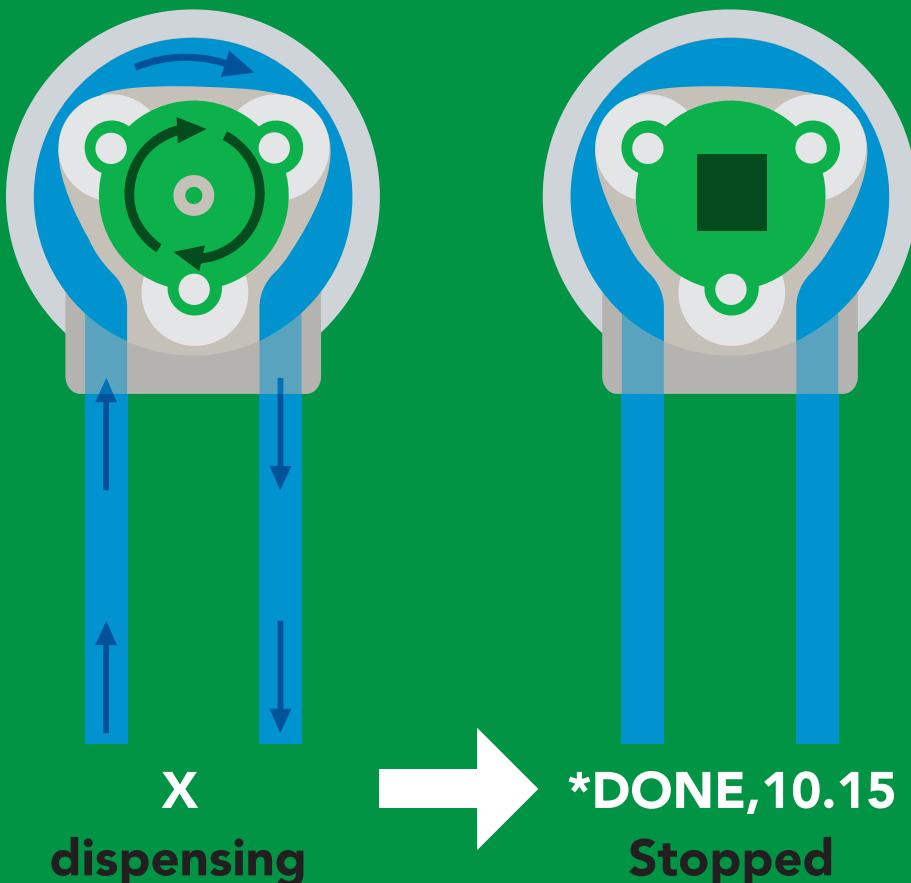
## Command syntax

X <cr> stop dispensing

### Example      Response

X <cr>

\*DONE,v <cr>    v = volume dispensed



# Invert dispensing direction

## Command syntax

Invert direction will be retained if power is cut

**Invert <cr>** changes dispensing direction of pump

## Example Response

**Invert <cr>**

**\*OK <cr>**

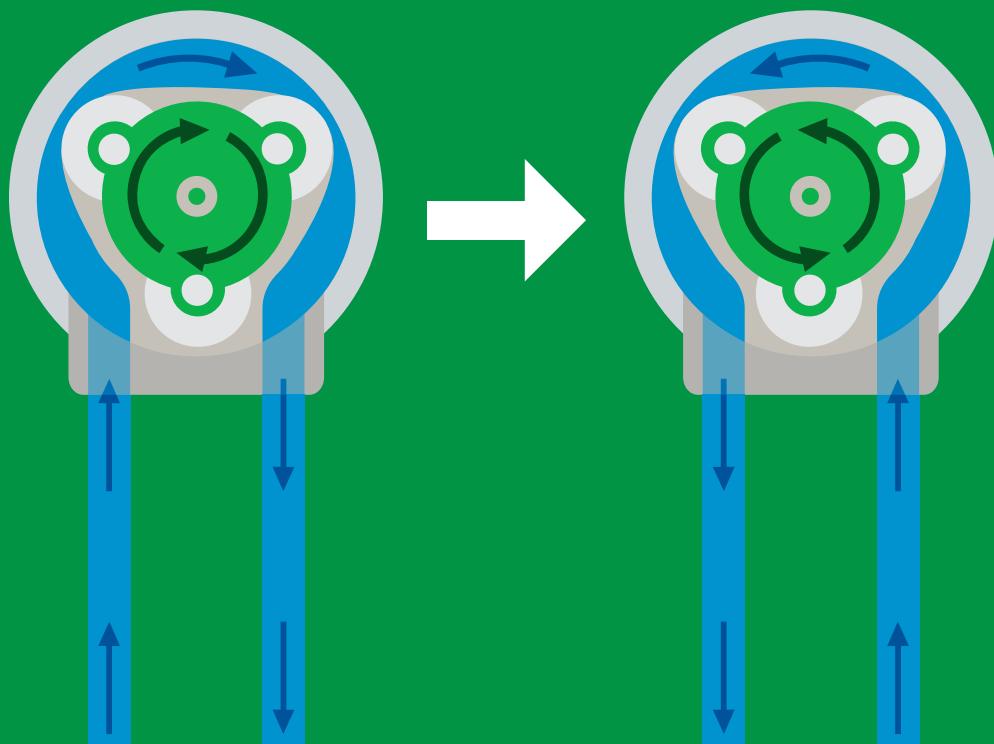
**Invert,? <cr>**

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

inverted

uninverted

**\*OK <cr>**



# Total volume dispensed

## Command syntax

**TV,? <cr>** shows total volume dispensed

**ATV,? <cr>** absolute value of the total volume dispensed

**Clear <cr>** clears the total dispensed volume

## Example

TV,? <cr>

## Response

?TV,434.50 <cr>

ATV,? <cr>

?ATV,623.00 <cr>

Clear <cr>

\*OK <cr> total now 0.00

This data will be lost if the power is cut.

# Calibration

## Command syntax

Calibrate to the actual volume dispensed.

**Cal,v** <cr> v = corrected volume

**Cal,clear** <cr> delete all calibration data

**Cal,?** <cr> device calibrated?

This command is used for both, single dose and dose over time calibrations.

### Example

**Cal,24.01** <cr>

\*OK <cr>

**Cal,clear** <cr>

\*OK <cr>

**Cal,?** <cr>

?Cal,1 <cr> or ?Cal,2 <cr> or  
fixed volume  
?Cal,3 <cr> or ?Cal,0 <cr>  
volume/time  
both  
uncalibrated

\*OK <cr>

### Response

# Enable/disable parameters from output string

## Command syntax

O, [parameter],[1,0] <cr> enable or disable output parameter

O,? <cr> enabled parameter?

### Example

O,V,1 <cr>

### Response

\*OK <cr> enable volume being pumped

O,TV,0 <cr>

\*OK <cr> disable total volume pumped

O,ATV,1 <cr>

\*OK <cr> enable absolute volume pumped

O,? <cr>

? ,O,V,TV,ATV <cr> if all three are enabled

# Pump voltage

## Command syntax

**PV,? <cr> check pump voltage**

### Example      Response

**PV,? <cr>**

**?PV,13.86 <cr>  
\*OK <cr>**

### Response breakdown

**?PV, 13.86**

Pump input voltage

# Naming device

## Command syntax

Do not use spaces in the name

Name,n <cr> set name

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

Name, <cr> clears name

Up to 16 ASCII characters

Name,? <cr> show name

## Example

## Response

Name, <cr>

\*OK <cr> name has been cleared

Name,zzt <cr>

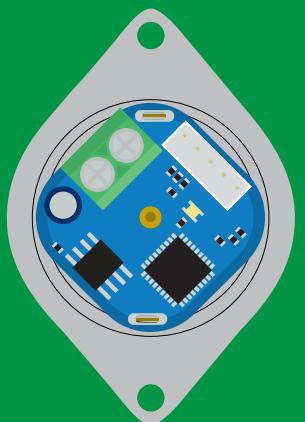
\*OK <cr>

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      Response

i <cr>

?i,PMP,1.1 <cr>  
\*OK <cr>

### Response breakdown

?i, PMP, 1.1  
↑            ↑  
Device    Firmware

# Response codes

## Command syntax

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

## Example

R <cr>

413 <cr>

\*OK <cr>

\*OK,0 <cr>

**no response, \*OK disabled**

R <cr>

413 <cr> \*OK disabled

\*OK,? <cr>

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

## Response

## Other response codes

*ER	unknown command
*OV	over volt (VCC>=5.5V)
*UV	under volt (VCC<=3.1V)
*RS	reset
*RE	boot up complete, ready
*SL	entering sleep mode
*WA	wake up
*DONE	dispensing complete
*MINVOL	dispense amount too low
*TOOFAST	ml/min set to fast

These response codes  
cannot be disabled

# Reading device status

## Command syntax

Status <cr> voltage at Vcc pin and reason for last restart

### Example      Response

Status <cr>

?Status,P,5.038 <cr>

\*OK <cr>

### Response breakdown

?Status, P, 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

**Sleep <cr>**

### Response

\*OK <cr>  
\*SL <cr>

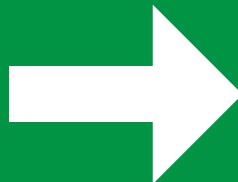
### Any command

\*WA <cr> wakes up device

	STANDBY	SLEEP
<b>5V</b>	<b>13.4 mA</b>	<b>0.415 mA</b>
<b>3.3V</b>	<b>12.4 mA</b>	<b>0.13 mA</b>



**Standby**  
**13.4 mA**



**Sleep <cr>**



**Sleep**  
**0.415 mA**

# Change baud rate

## Command syntax

Baud,n <cr> change baud rate

### Example      Response

Baud,38400 <cr>

\*OK <cr>

Baud,? <cr>

?Baud,38400 <cr>

\*OK <cr>

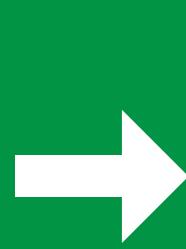
n = [ 300  
1200  
2400  
**9600 default**  
19200  
38400  
57600  
115200 ]



Baud,38400 <cr>



Changing  
baud rate  
\*OK <cr>



(reboot)



Standby

# Protocol lock

## Command syntax

Locks device to UART mode.

**Plock,1 <cr>** enable Plock

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

**Plock,? <cr>** Plock on/off?

## Example

**Plock,1 <cr>**

**\*OK <cr>**

**Plock,0 <cr>**

**\*OK <cr>**

**Plock,? <cr>**

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

**Plock,1**

**I2C,100**

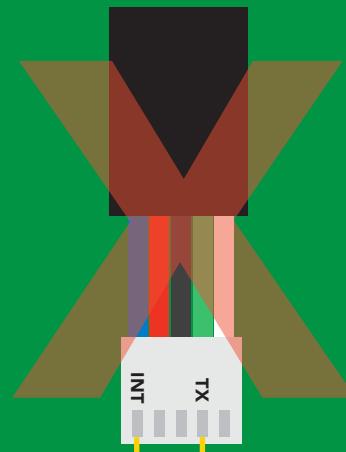


**\*OK <cr>**



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

**\*ER <cr>**



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

# Factory reset

## Command syntax

Clears calibration  
LED on  
"\*OK" enabled

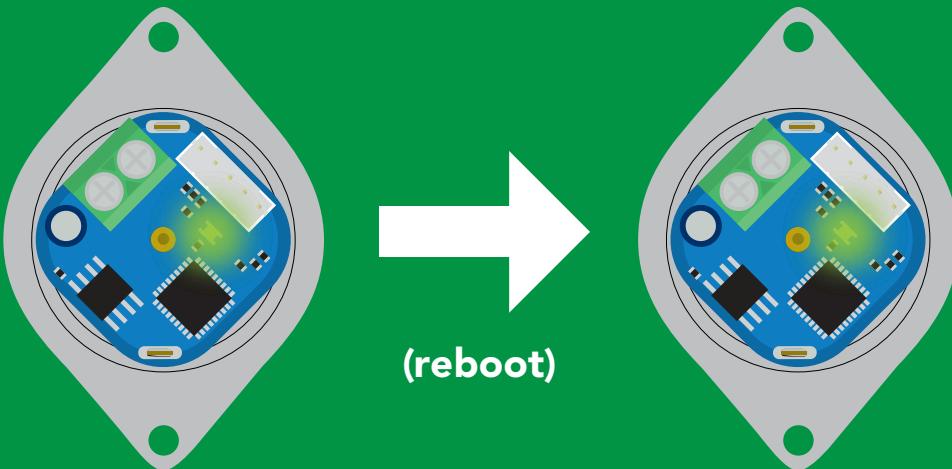
Factory <cr> enable factory reset

## Example Response

Factory <cr>

\*OK <cr>

Factory <cr>



\*OK <cr>

\*RS <cr>

\*RE <cr>

Baud rate will not change

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

## Command syntax

Default I<sup>2</sup>C address 103 (0x67)

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

n = any number 1 – 127

## Example      Response

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

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

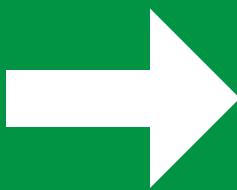
## Wrong example

I<sup>2</sup>C,139 <cr> n > 127

## Response

\*ER <cr>

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



(reboot)



Green

\*OK <cr>

Blue

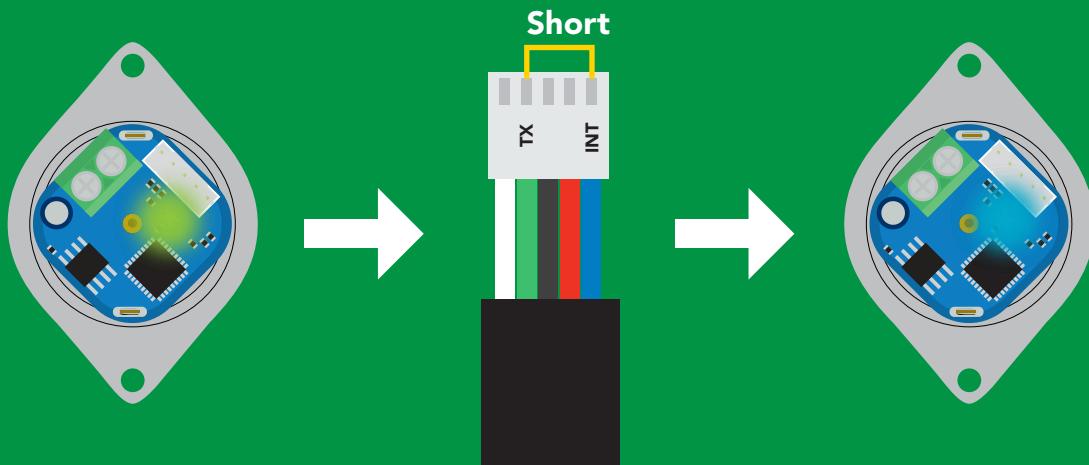
now in I<sup>2</sup>C mode

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

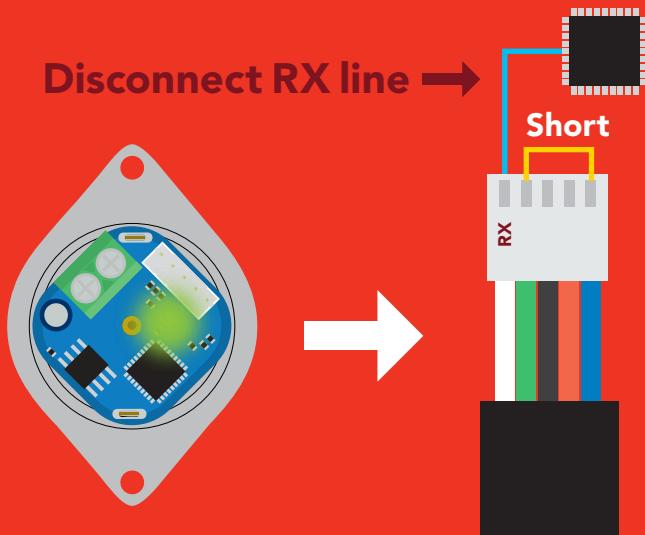
- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to INT
- Confirm RX is disconnected
- Connect ground (power on)
- Wait for LED to change from Green to Blue
- Disconnect ground (power off)
- Reconnect all data and power

Manually switching to I<sup>2</sup>C will set the I<sup>2</sup>C address to 103 (0x67)

## 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-PMP™ 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
- Enable/disable parameters
- Hardware switch to UART mode
- Invert
- LED control
- Protocol lock
- Software switch to UART mode

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

- Absolute total volume
- Find
- Sleep mode
- Total volume

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

I<sup>2</sup>C address (0x01 – 0x7F)  
103 (0x67) default

V<sub>cc</sub> 3.3V – 5.5V

Clock speed 100 – 400 kHz

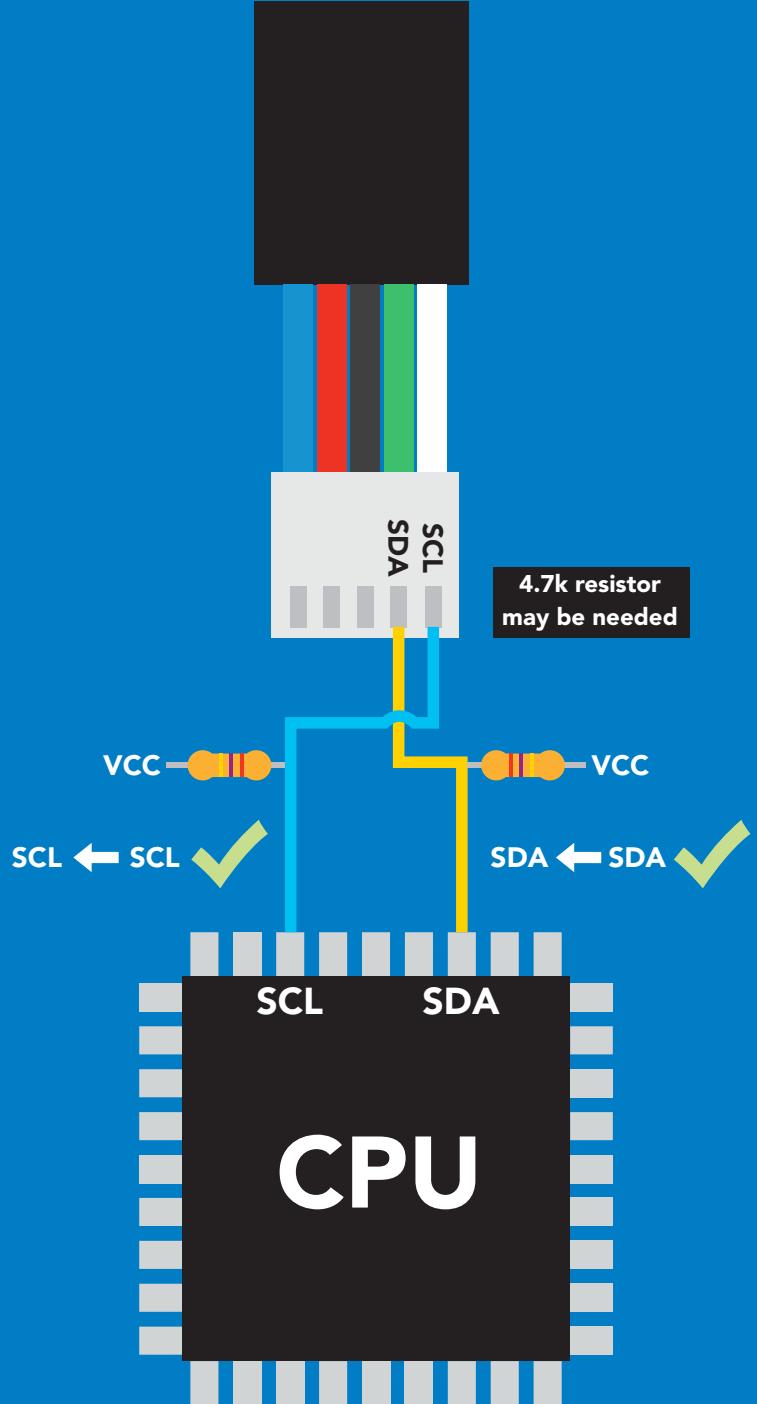
SDA



SCL



0V



## Data format

Reading volume  
Units ml  
Encoding ASCII  
Format string

Data type floating point  
Decimal places 2  
Smallest string 3 characters  
Largest string 39 characters

# Sending commands to device

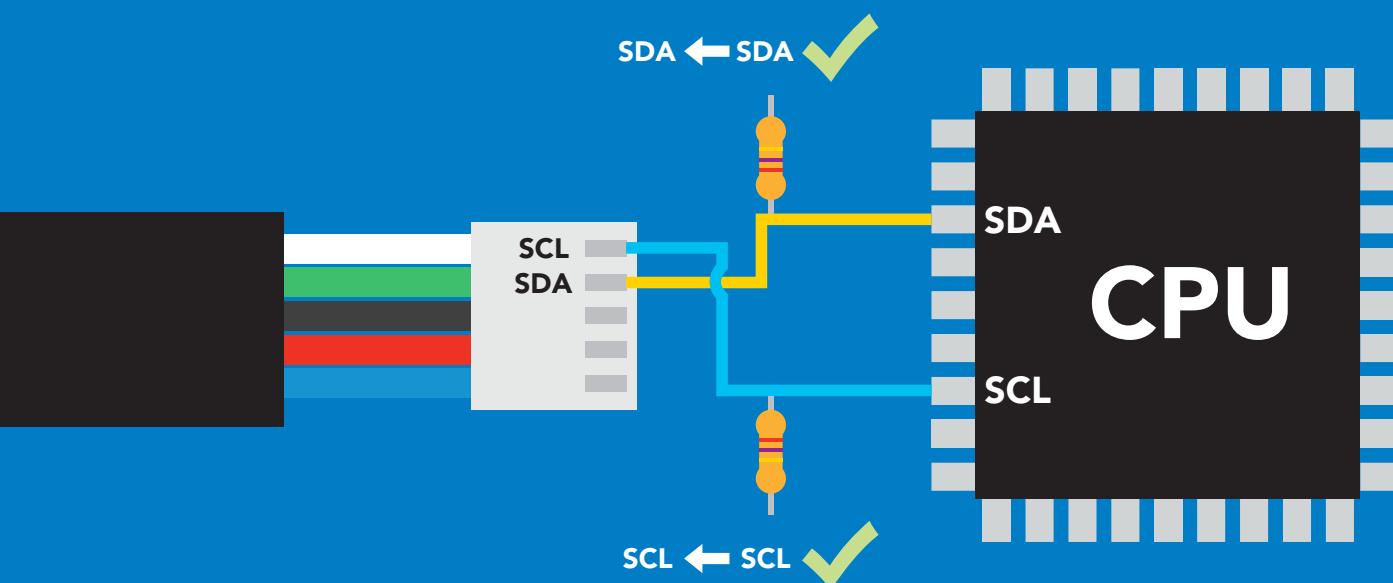
5 parts



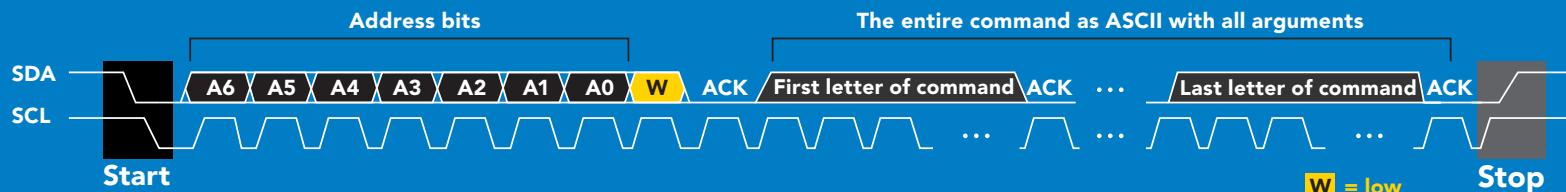
## Example

**Start**    **103 (0x67)**    **Write**    **Sleep**    **Stop**

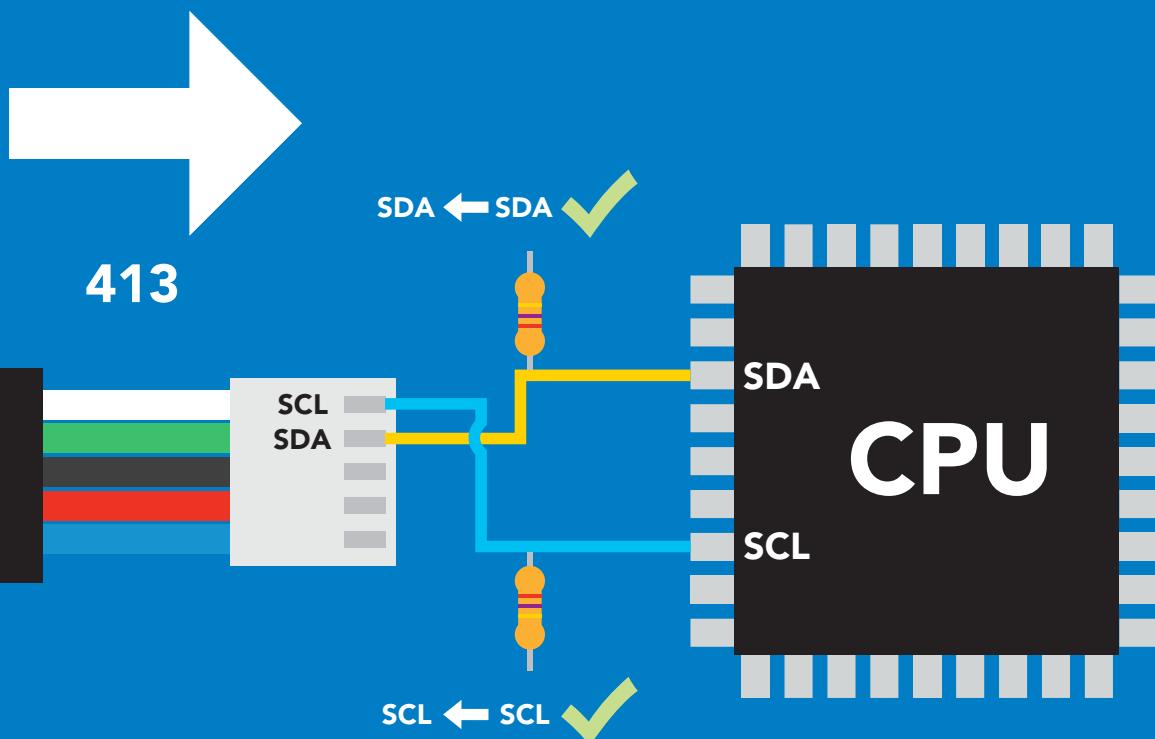
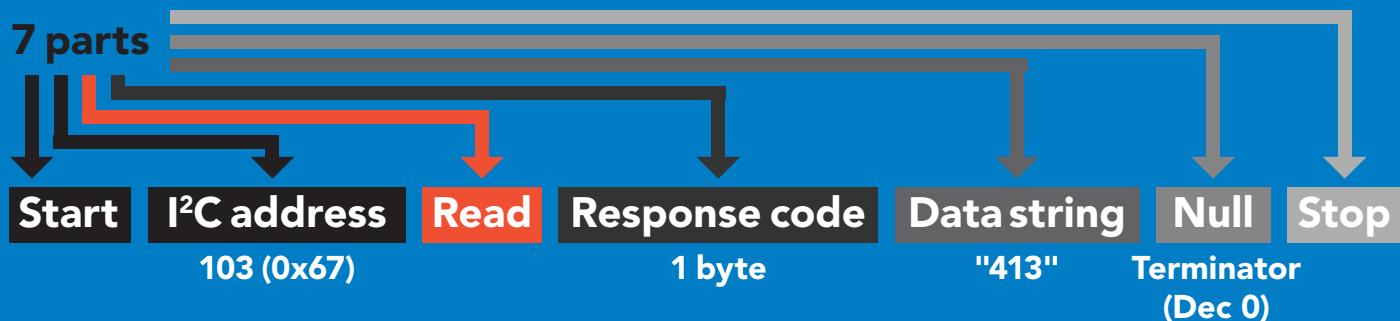
I<sup>2</sup>C address                      Command



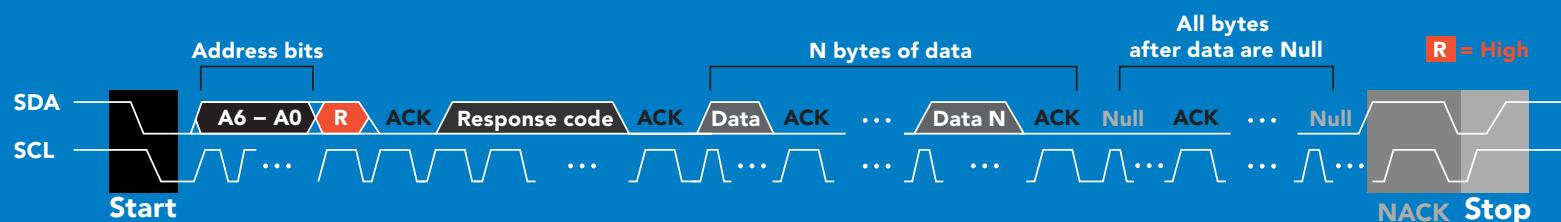
## Advanced



# Requesting data from device



## Advanced

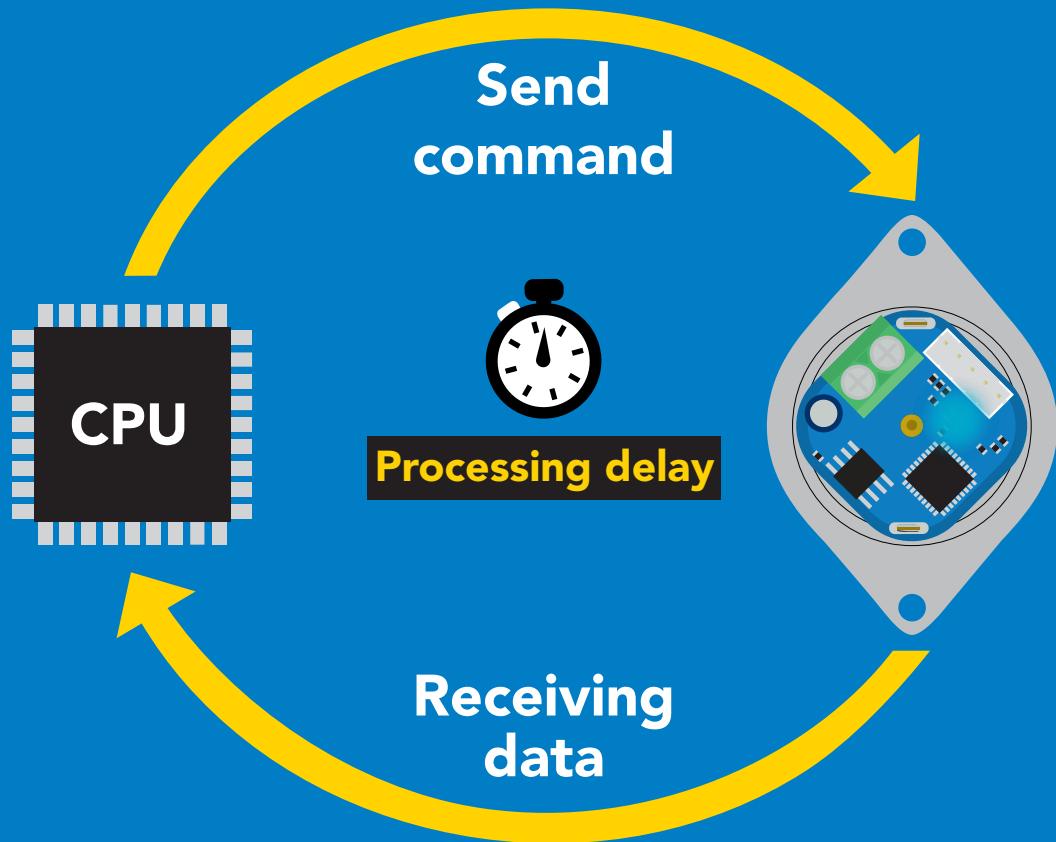


1 52 49 51 0 = 413  
Dec Dec  
ASCII

# 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

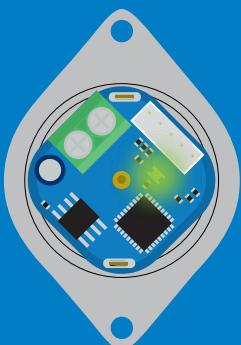
Single byte, not string

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

<b>5V</b>	LED ON <b>+2.5 mA</b>
<b>3.3V</b>	<b>+1 mA</b>

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

## command quick reference

All commands are ASCII strings or single ASCII characters.

Command	Function	
Baud	switch back to UART mode	pg. 75
Cal	performs calibration	pg. 65
D	dispense modes	pg. 54 – 60
Factory	enable factory reset	pg. 74
Find	finds device with blinking white LED	pg. 52
i	device information	pg. 69
Invert	invert dispensing direction	pg. 63
I2C	change I <sup>2</sup> C address	pg. 73
L	enable/disable LED	pg. 51
Name	set/show name of device	pg. 68
O	enable/disable parameters	pg. 66
P	pauses the pump during dispensing	pg. 61
Plock	enable/disable protocol lock	pg. 72
Pv	check pump voltage	pg. 67
R	returns a single reading	pg. 53
Sleep	enter sleep mode/low power	pg. 71
Status	retrieve status information	pg. 70
Tv	total volume dispensed	pg. 64
X	stop dispensing	pg. 62

# LED control

## Command syntax

300ms  processing delay

L,1 LED on **default**

L,0 LED off

L,? LED state on/off?

## Example

L,1

  
Wait 300ms

**1**  
Dec      **0**  
Null

L,0

  
Wait 300ms

**1**  
Dec      **0**  
Null

L,?

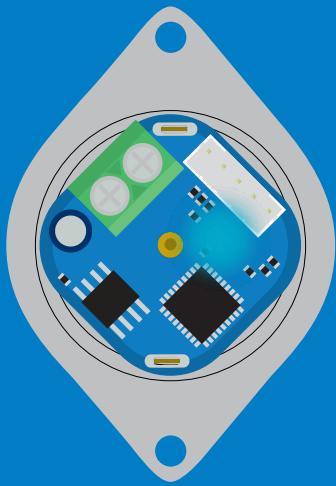
  
Wait 300ms

**1**    ?L,1    **0**  
Dec    ASCII    Null

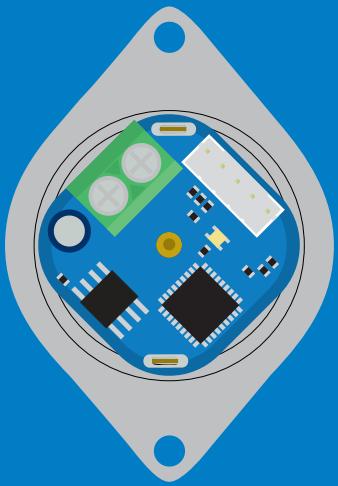
or

  
Wait 300ms

**1**    ?L,0    **0**  
Dec    ASCII    Null



L,1



L,0

# Find

300ms  processing delay

## Command syntax

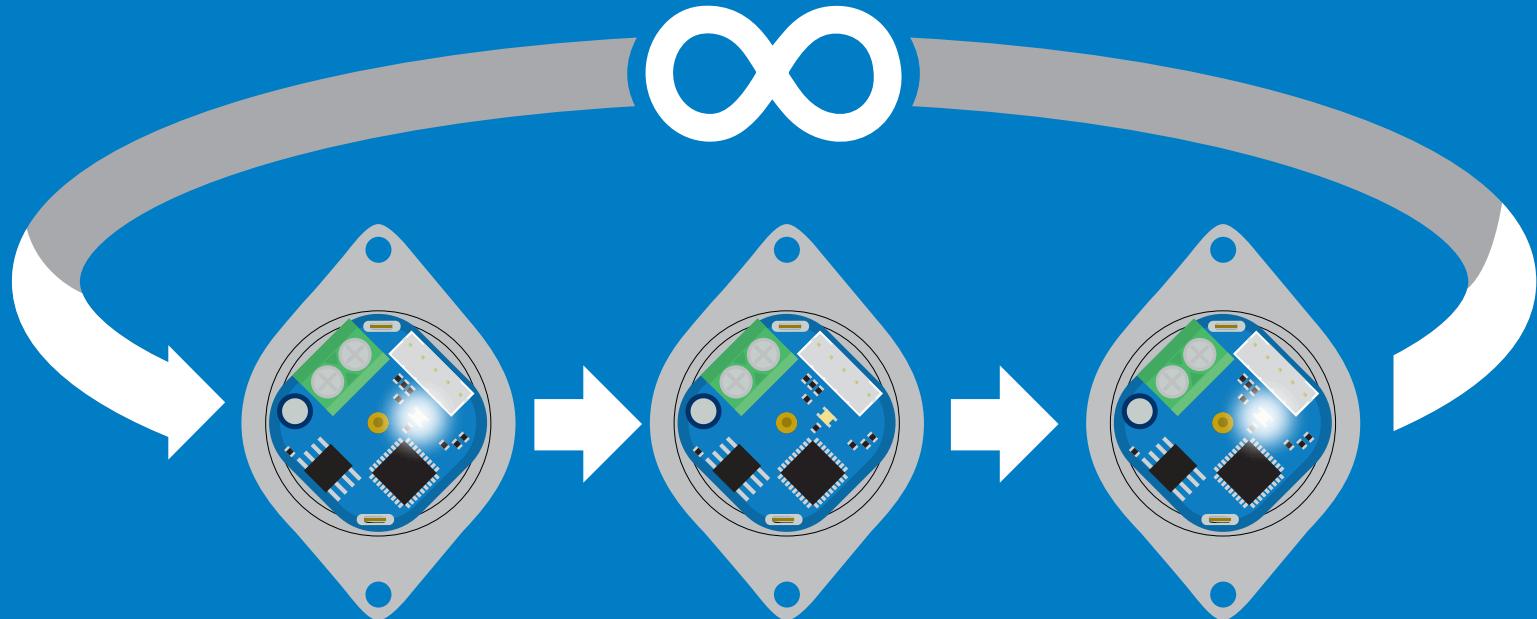
This command will disable continuous mode  
Send any character or command to terminate find.

Find      LED rapidly blinks white, used to help find device

## Example      Response

Find

 Wait 300ms  
1 Dec 0 Null



# Single report mode

## Command syntax

300ms  processing delay

R returns a single value showing dispensed volume

### Example

R



Wait 300ms

1

2.50

0

Dec

ASCII

Null

(If issued half way through dispensing 5ml)



Wait 300ms

1

5.00

0

Dec

ASCII

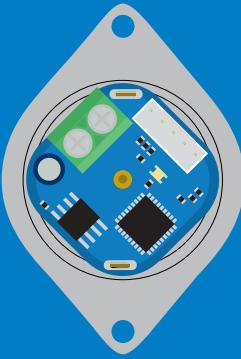
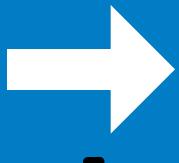
Null

(If issued once dispensing has stopped)

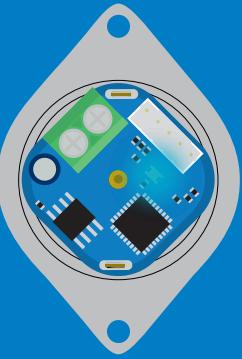
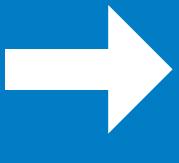


Green

Taking reading



Transmitting



Blue

Standby

# Continuous dispensing

Pump on/pump off

300ms  processing delay

## Command syntax

After running in continuous mode for 20 days the EZO-PMP™ will reset.

D,\* dispense until the stop command is given

D,-\* dispense in reverse until the stop command is given

D,? dispense status

## Example

## Response

D,\*



1  
Dec  
0  
Null

pump will continuously run at  
~105ml/min (with supplied tubing)

D,-\*



1  
Dec  
0  
Null

pump will continuously run in reverse  
at ~105ml/min (with supplied tubing)

D,?



1  
Dec  
?D,10.00,1  
ASCII  
0  
Null

## Response breakdown

?D,\*,1

↑  
last volume  
↑  
pump on  
requested

# Volume dispensing

Pump a specific volume

300ms  processing delay

## Command syntax

where [ml] is any volume in millimeters  $\geq 0.5$

D,[ml] dispense [this specific volume]

D,[-ml] dispense [in reverse this specific volume]

D,? dispense status

## Example

## Response

D,15



1 Dec 0 Null

15 ml will be dispensed

D,-40.5



1 Dec 0 Null

40.5 ml will be dispensed  
in reverse

D,?



1 Dec ?D,-40.50,0 0 Null

## Response breakdown

?D,-40.50,0

↑      ↑  
last volume    pump off  
dispensed

# Dose over time

Pump a fixed volume over a fixed time

## Command syntax

300ms  processing delay

**D,[ml],[min]** Dispense [this volume], [over this many minutes]

## Example

**D,85,10**

  
Wait 300ms

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

Dispense 85ml over 10 mins



# Constant flow rate

Maintain a constant flow rate

300ms  processing delay

## Command syntax

**DC,[ml/min], [min or \*]**

[maintain this rate], [for this much time]

**DC,?**

reports maximum possible flow rate

[ml/min] = a single number (int or float) representing the desired flow rate

[min or \*] = the number of minutes to run or (\*) indefinitely

A negative value for ml/min = reverse

## Example

**DC,25,40**



1  
Dec  
Null

Dispense 25ml per minute  
for 40 minutes

**DC,?**

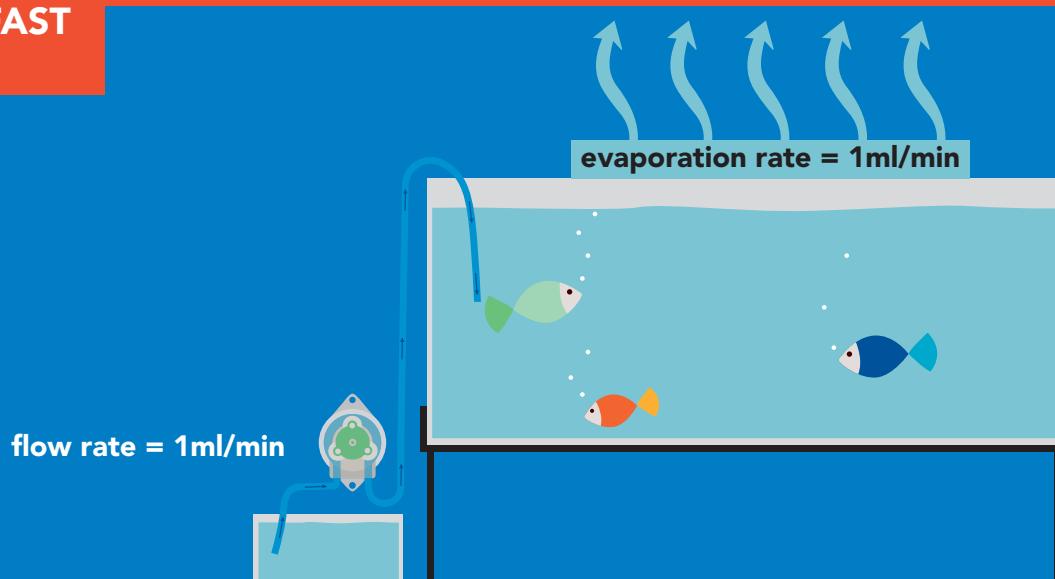


1  
Dec  
ASCII  
0  
Null

The maximum flow rate is determined after calibration.  
If the flowrate entered is too fast the EZO-PMP™ will send an error.

\*TOOFAST

\*ER



# Dispense at startup

Pump a specific volume at startup and then stop

Use this command to make a simple fixed-volume pump

## Command syntax

300ms  processing delay

**Dstart,[ml]** dispense [this specific volume] at startup

**Dstart,off** disables dispense at startup mode

**Dstart,?** startup dispense status

## Example

## Response

**Dstart,10**

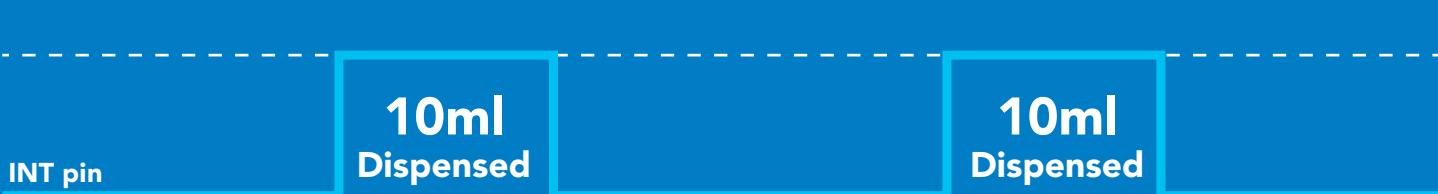
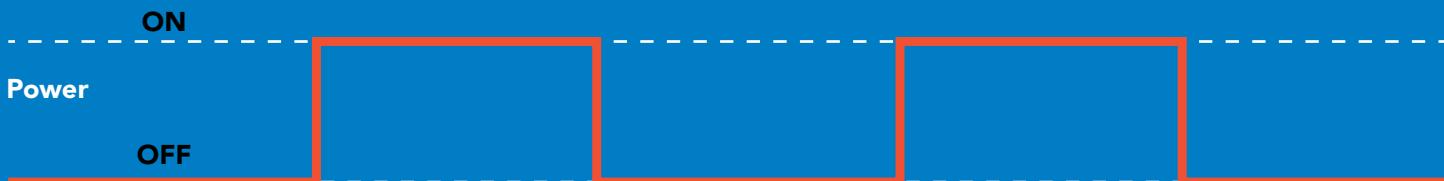
 Wait 300ms  
1 Dec 0 Null

**Dstart,off**

 Wait 300ms  
1 Dec 0 Null

**D,?**

 Wait 300ms  
1 Dec ?Dstart,10 0 Null or  Wait 300ms  
1 Dec ?Dstart,0 0 Null



# Continuous dispensing at startup

Pump on & continuously dispense

300ms  processing delay

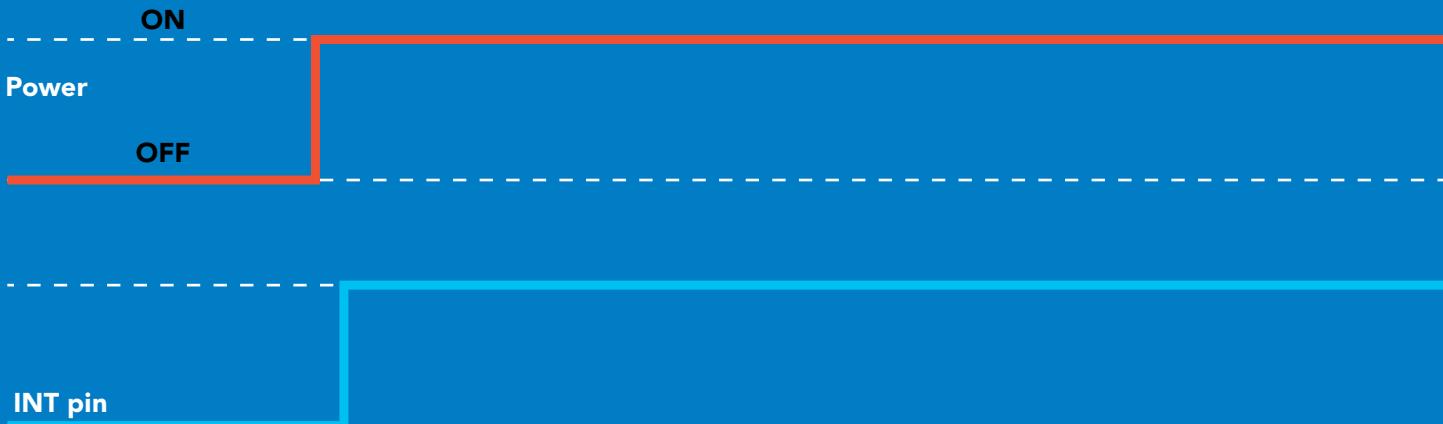
## Command syntax

After running in continuous mode for 20 days the EZO-PMP™ will reset.

- Dstart,\*** dispense at startup until the stop command is given
- Dstart,-\*** dispense in reverse at startup until the stop command is given
- Dstart,?** startup dispense status

## Example Response

Example	Response
<b>Dstart,*</b>	 Wait 300ms 1 Dec 0 Null Pump will startup and continuously run at ~105ml/min (with supplied tubing)
<b>Dstart,-*</b>	 Wait 300ms 1 Dec 0 Null Pump will startup and continuously run in reverse at ~105ml/min (with supplied tubing)
<b>Dstart,?</b>	 Wait 300ms 1 Dec ?Dstart,* 0 Null ?Dstart,*



# Dose Over time at startup

Pump a fixed volume over a fixed time at startup

## Command syntax

**Dstart[ml],[min]**

Dispense [volume], [over this many minutes] at startup

300ms  processing delay

## Example

**Dstart,85,10**



1  
Dec  
0  
Null

Pump will startup and dispense 85ml over 10 minutes

## Response

**Dstart,?**



1  
Dec  
?Dstart,85.00,10.00  
ASCII  
0  
Null



# Pause dispensing

## Command syntax

300ms  processing delay

Issue the command again to resume dispensing

P pauses the pump during dispensing

P? pause status

## Example

P

 Wait 300ms  
1 Dec 0 Null

P?

 Wait 300ms  
1 Dec ?P,1 0 ASCII paused or  
1 Dec ?P,0 0 ASCII unpause



dispensing

P



paused

P



dispensing

# Stop dispensing

Command syntax

300ms  processing delay

X stop dispensing

Example

Response

X

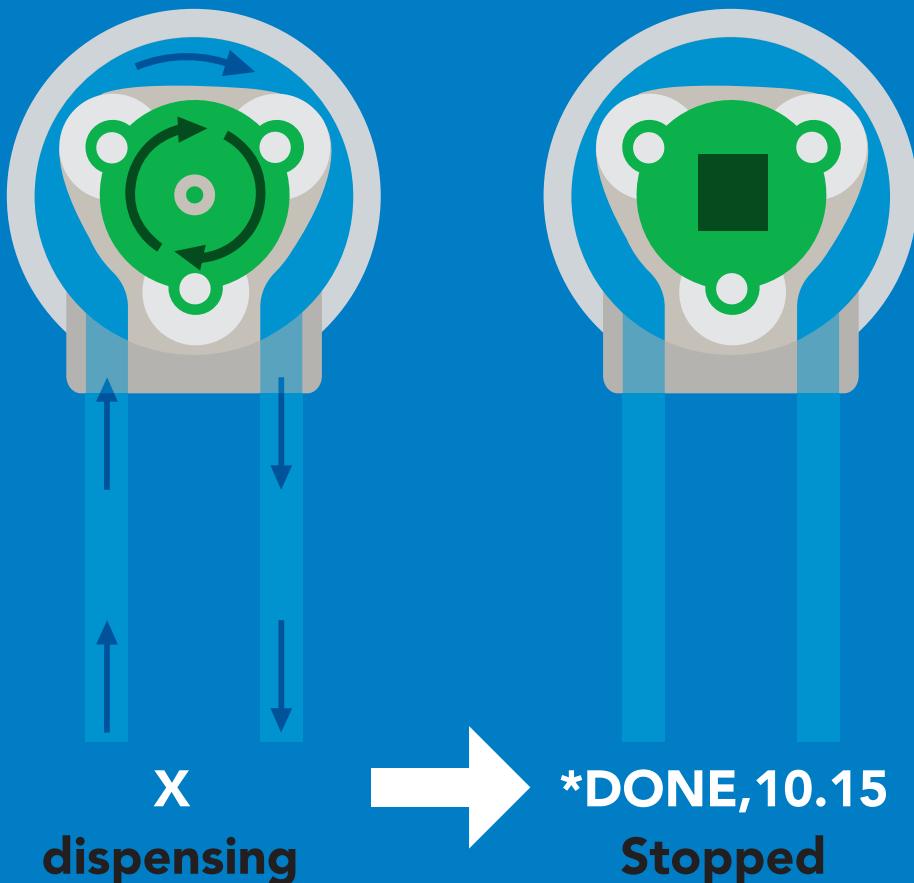
  
Wait 300ms

1  
Dec

\*DONE,v  
ASCII

0  
Null

v = volume dispensed



# Invert dispensing direction

300ms  processing delay

## Command syntax

Invert direction will be retained if power is cut

Invert changes dispensing direction of pump

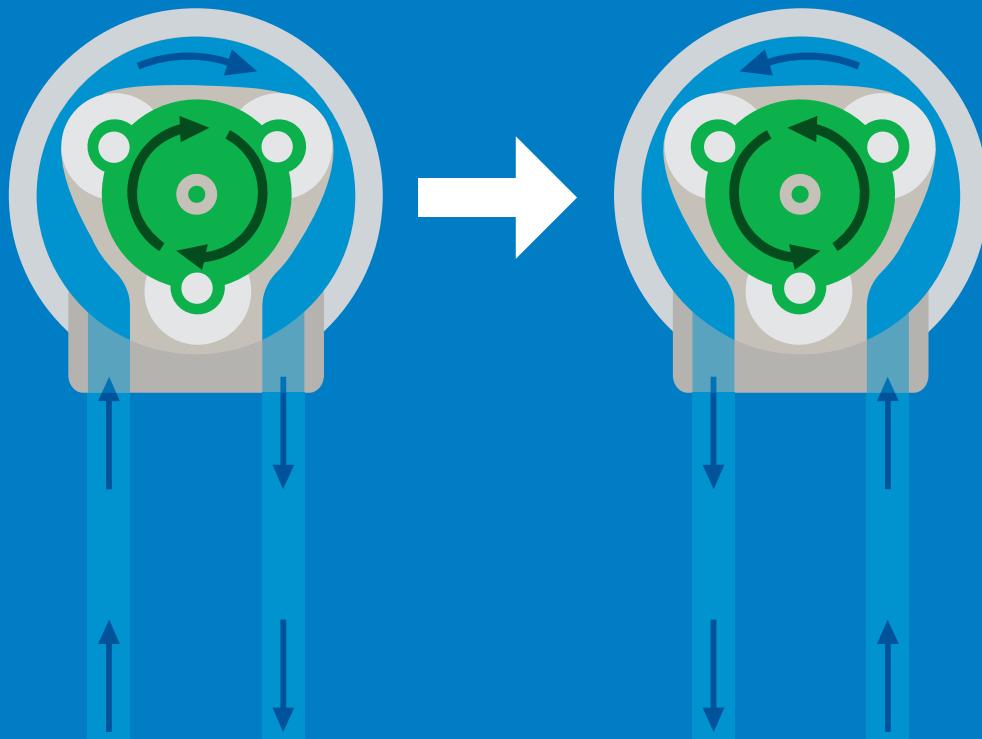
## Example Response

Invert

 Wait 300ms  
1 Dec 0 Null

Invert,?

 Wait 300ms  
1 Dec ?Invert,1 0 Null or  
?Invert,0 0 Null  
Dec ASCII inverted  
Dec ASCII uninverted



# Total volume dispensed

## Command syntax

300ms  processing delay

**TV,?** shows total volume dispensed

**ATV,?** absolute value of the total volume dispensed

**Clear** clears the total dispensed volume

## Example

**TV,?**

## Response



Wait 300ms

1

Dec

?TV,623.00

0

ASCII

Null

**ATV,?**



Wait 300ms

1

Dec

?ATV,434.50

0

ASCII

Null

**clear**



Wait 300ms

1

Dec

0

total now 0.00

This data will be lost if the power is cut.



# Enable/disable parameters from output string

## Command syntax

300ms  processing delay

O, [parameter],[1,0] enable or disable output parameter

O,? enabled parameter?

## Example

O,V,1

## Response



Wait 300ms

1  
Dec

0  
Null

enable volume being pumped

O,TV,0



Wait 300ms

1  
Dec

0  
Null

disable total volume pumped

O,ATV,1



Wait 300ms

1  
Dec

0  
Null

enable absolute  
volume pumped

O,?



Wait 300ms

1  
Dec

? , O, V, TV, ATV

ASCII

0  
Null

if all three  
are enabled

# Pump voltage

## Command syntax

300ms  processing delay

**PV,?** check pump voltage

## Example

## Response

**PV,?**



Wait 300ms

**1**

**?PV,13.86**

Dec

ASCII

**0**

Null

## Response breakdown

**?PV, 13.86**

Pump input voltage

# 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

1 Dec 0 Null

name has been cleared

Name,zzt

 Wait 300ms

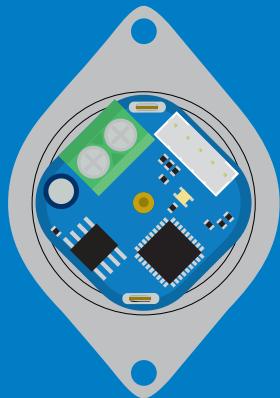
1 Dec 0 Null

Name,?

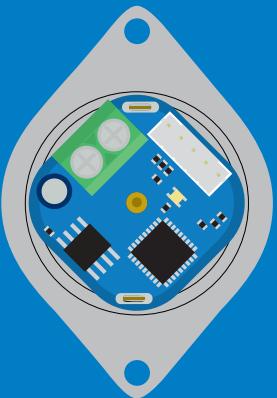
 Wait 300ms

1 Dec ?Name,zzt 0 Null

Name,zzt



Name,?



1 0

1 ?Name,zzt 0

# Device information

## Command syntax

300ms  processing delay

i device information

## Example

## Response

i



Wait 300ms

1

Dec

?i,PMP, 1.1

ASCII

0

Null

## Response breakdown

?i, PMP, 1.1  
↑      ↑  
Device Firmware

# Reading device status

## Command syntax

300ms  processing delay

Status voltage at Vcc pin and reason for last restart

## Example Response

Status



Wait 300ms

1

?Status,P,5.038

Dec

ASCII

0

Null

## Response breakdown

?Status, P,  
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**

	STANDBY	SLEEP
<b>5V</b>	<b>13.4 mA</b>	<b>0.415 mA</b>

<b>3.3V</b>	<b>12.4 mA</b>	<b>0.13 mA</b>
-------------	----------------	----------------



# Protocol lock

## Command syntax

300ms  processing delay

Plock,1 enable Plock

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

Plock,0 disable Plock **default**

Plock,? Plock on/off?

## Example

Plock,1

  
Wait 300ms

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

Plock,0

  
Wait 300ms

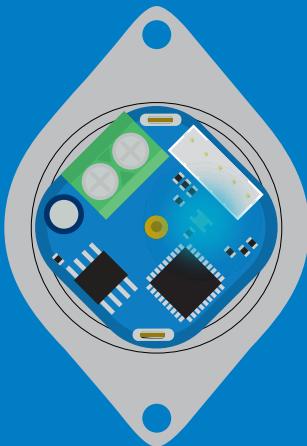
**1** Dec **0** Null

Plock,?

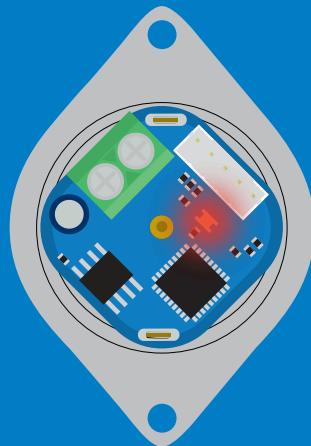
  
Wait 300ms

**1** Dec **?Plock,1** ASCII **0** Null

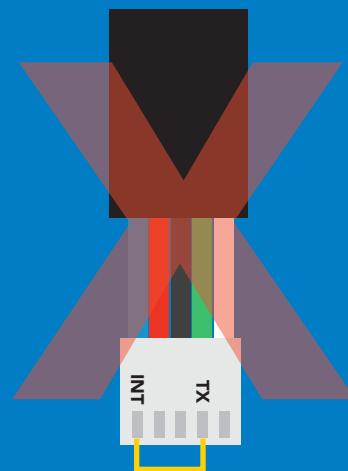
Plock,1



Baud, 9600



cannot change to UART



cannot change to UART

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

## Command syntax

300ms  processing delay

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)

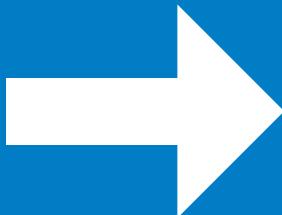
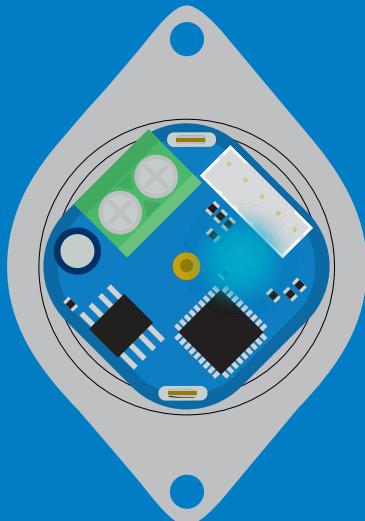
## Warning!

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

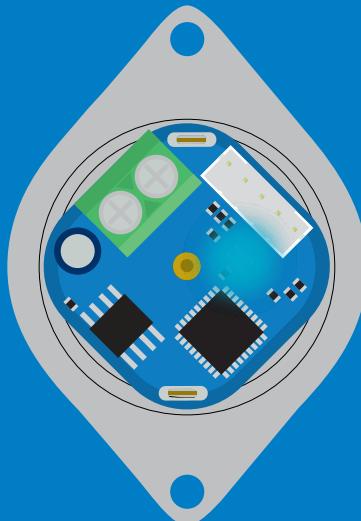
Default I<sup>2</sup>C address is 103 (0x67).

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

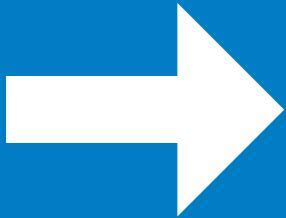
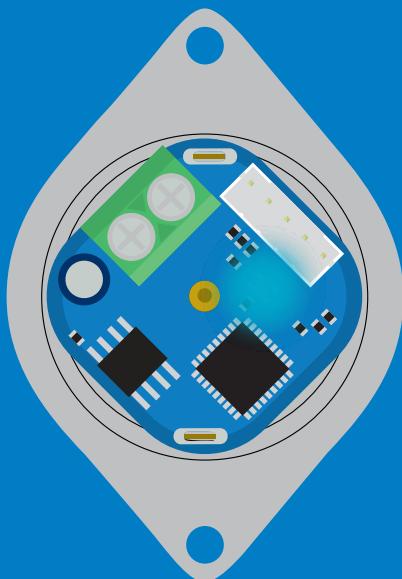
Factory

device reboot

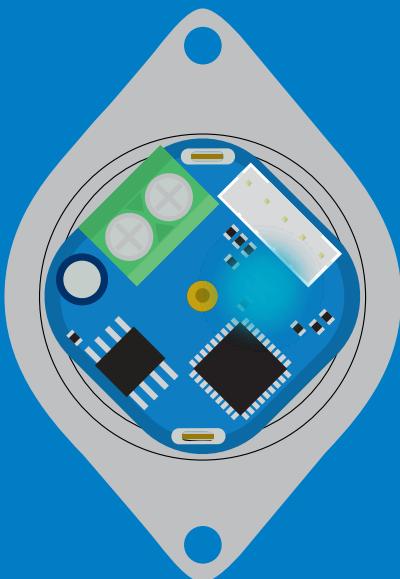
(no response given)

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

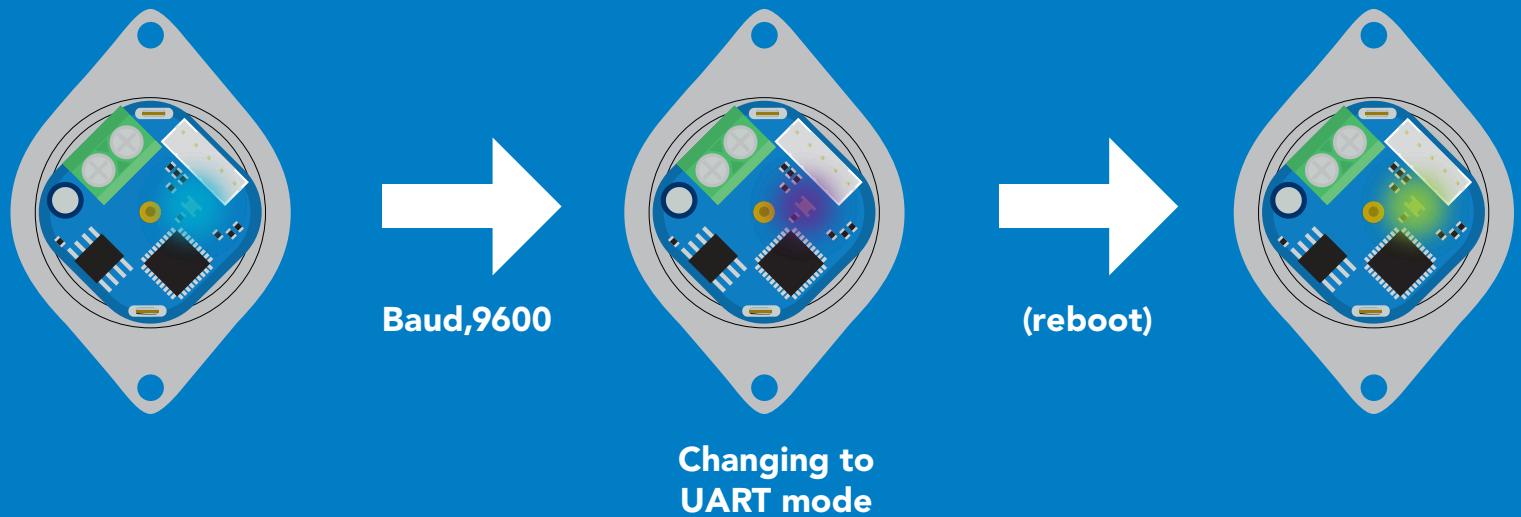
### Example      Response

Baud,9600

reboot in UART mode

(no response given)

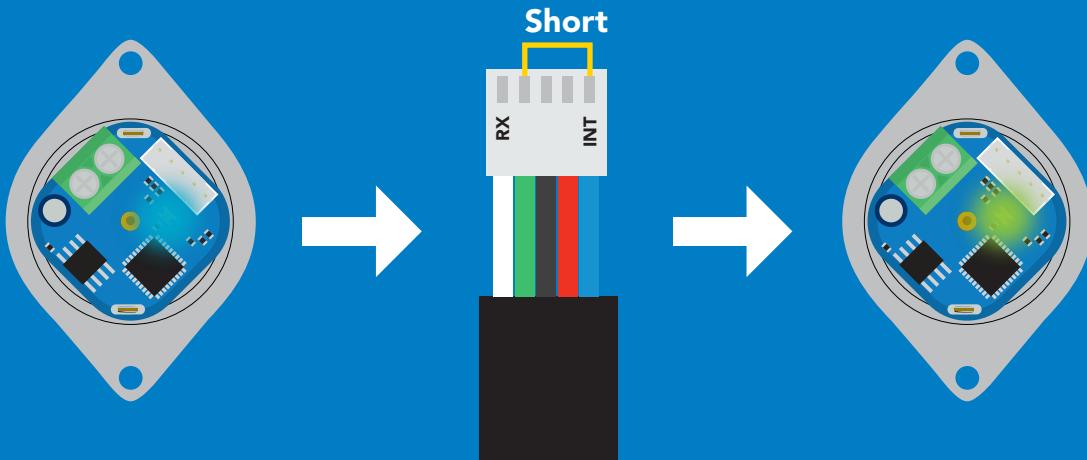
$$n = \begin{bmatrix} 300 \\ 1200 \\ 2400 \\ 9600 \\ 19200 \\ 38400 \\ 57600 \\ 115200 \end{bmatrix}$$



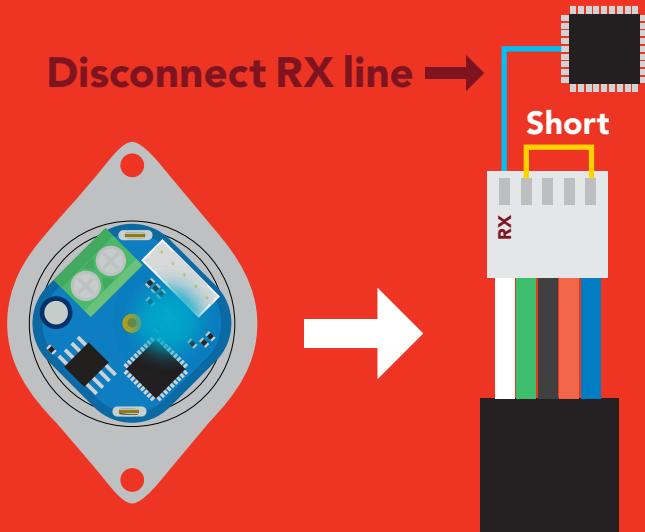
# Manual switching to UART

- 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



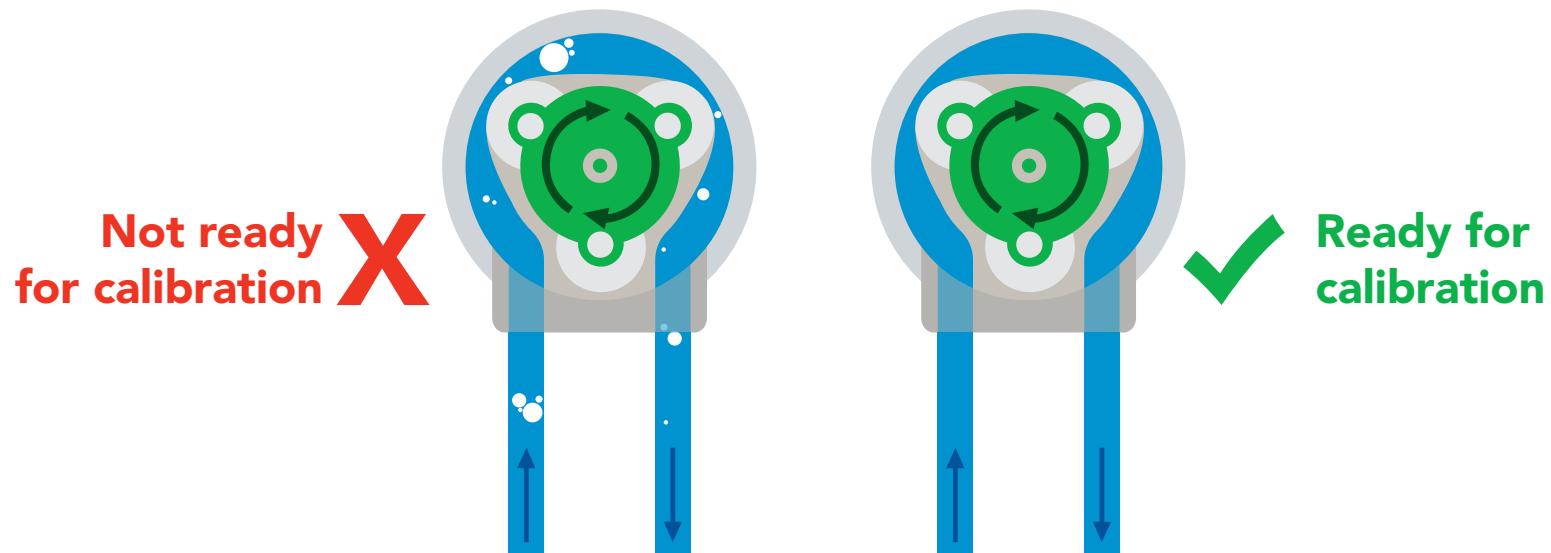
## Wrong Example



# Calibration theory

Uncalibrated accuracy +/- 5%  
Calibrated accuracy +/- 1%

**Before calibration is attempted** all the air bubbles should be removed from the tubing. This is done by running the pump while tapping the tubing. If air bubbles are not removed from the tubing they will slowly group together into larger air bubbles. Over time this will lead to accuracy issues.



## Calibration types

Volume calibration

Volume over time calibration

Calibration is optional. Both types of calibration are independent of each other and can be done at any time. Calibration can be done at any volume however; Atlas Scientific recommends using volumes above 5ml.

## Equipment needed for calibration



An accurate graduated cylinder of at least 10ml.

Or



1 gram of water = 1ml  
23.56 grams of water = 23.56ml

An accurate scale with a resolution of at least 0.1 grams

# Calibration procedure

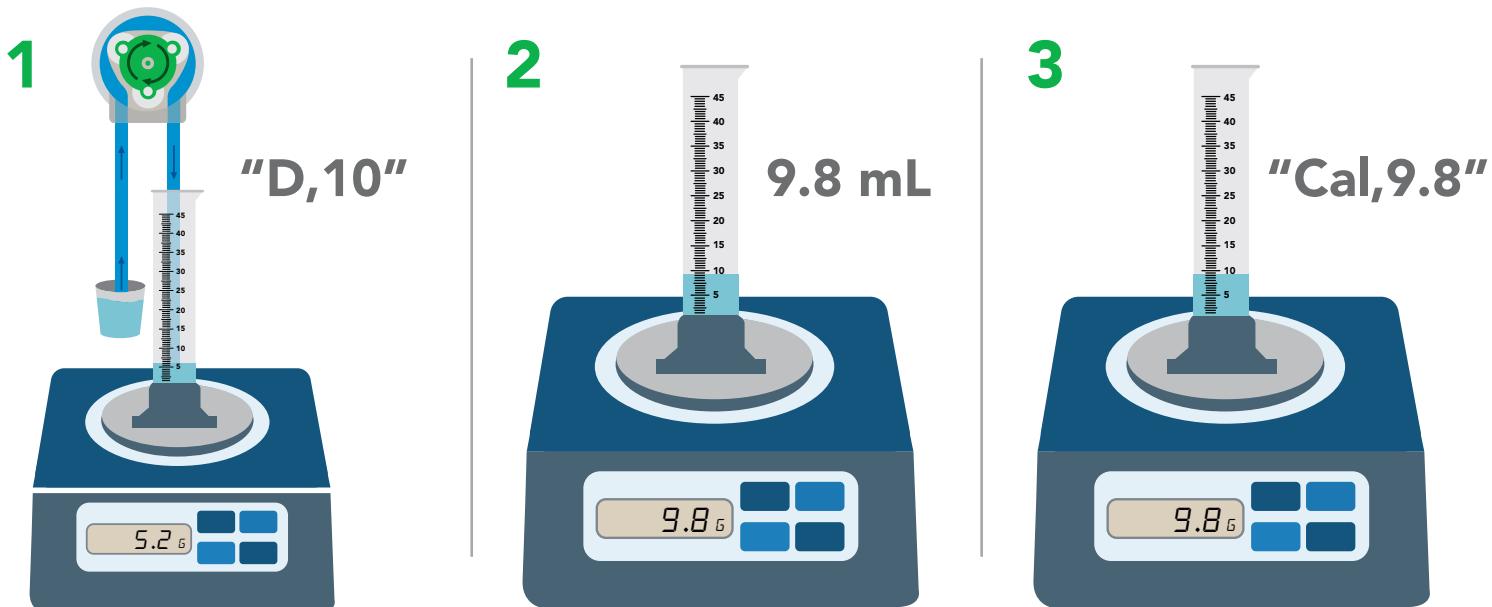
Calibration should be done with water and not a chemical

**Make sure the tubing is full of water and has no bubbles before calibrating.**

1. Instruct the pump to dispense a volume of water.
2. Measure the dispensed amount to determine how much water was actually dispensed.
3. Calibrate the pump by sending it the volume of liquid you have measured.

## Example

Calibrate the pump by dispensing 10ml

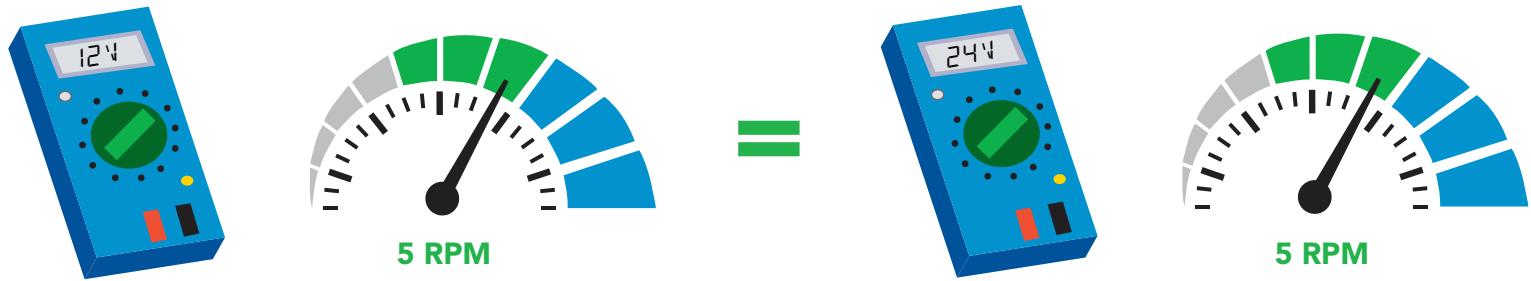


1. Instruct the pump to dispense 10ml into a graduated cylinder or beaker on a scale.
2. Measure the amount of liquid that was actually dispensed.
3. Inform the pump how much liquid was actually dispensed.
4. Calibration is now complete.

Once the pump has been calibrated, it will accurately dispense any volume of liquid. Use the same procedure to perform a volume over time calibration.

# Pump speed vs. voltage

There is no change in pump speed at different voltages.

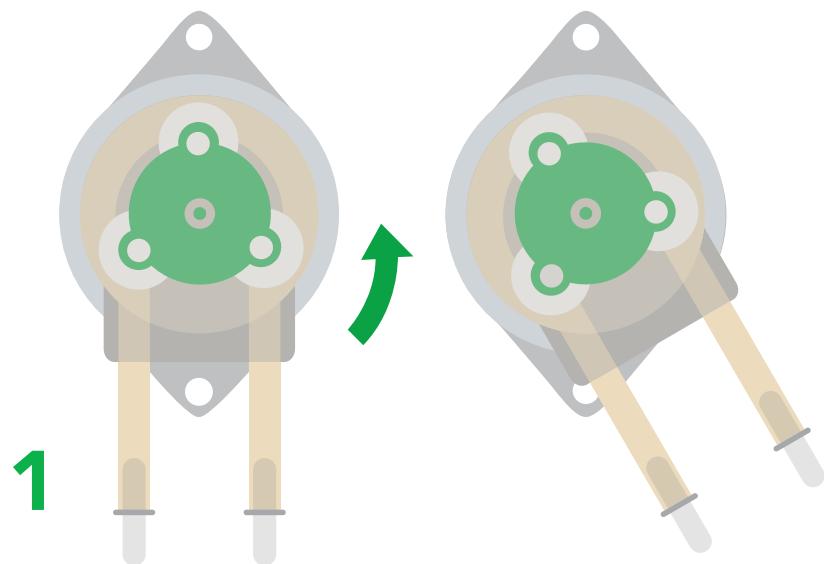


## Interrupt pin

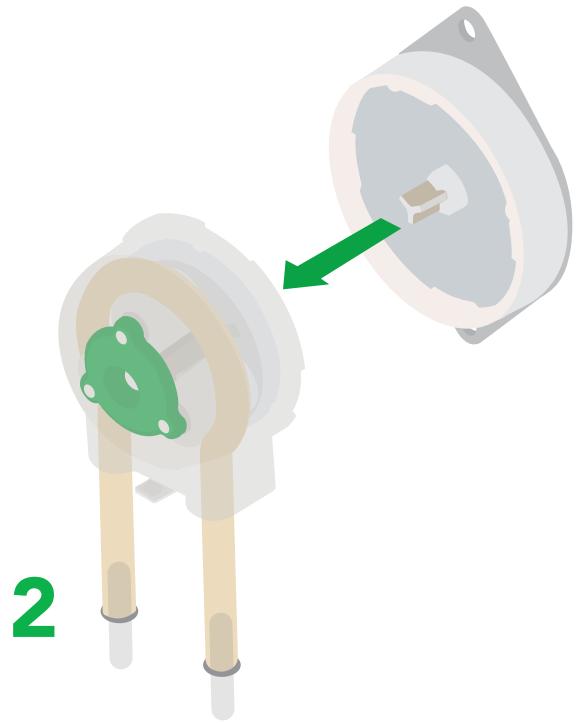
When the pump is dispensing the interrupt pin goes high.



## Removing cassette



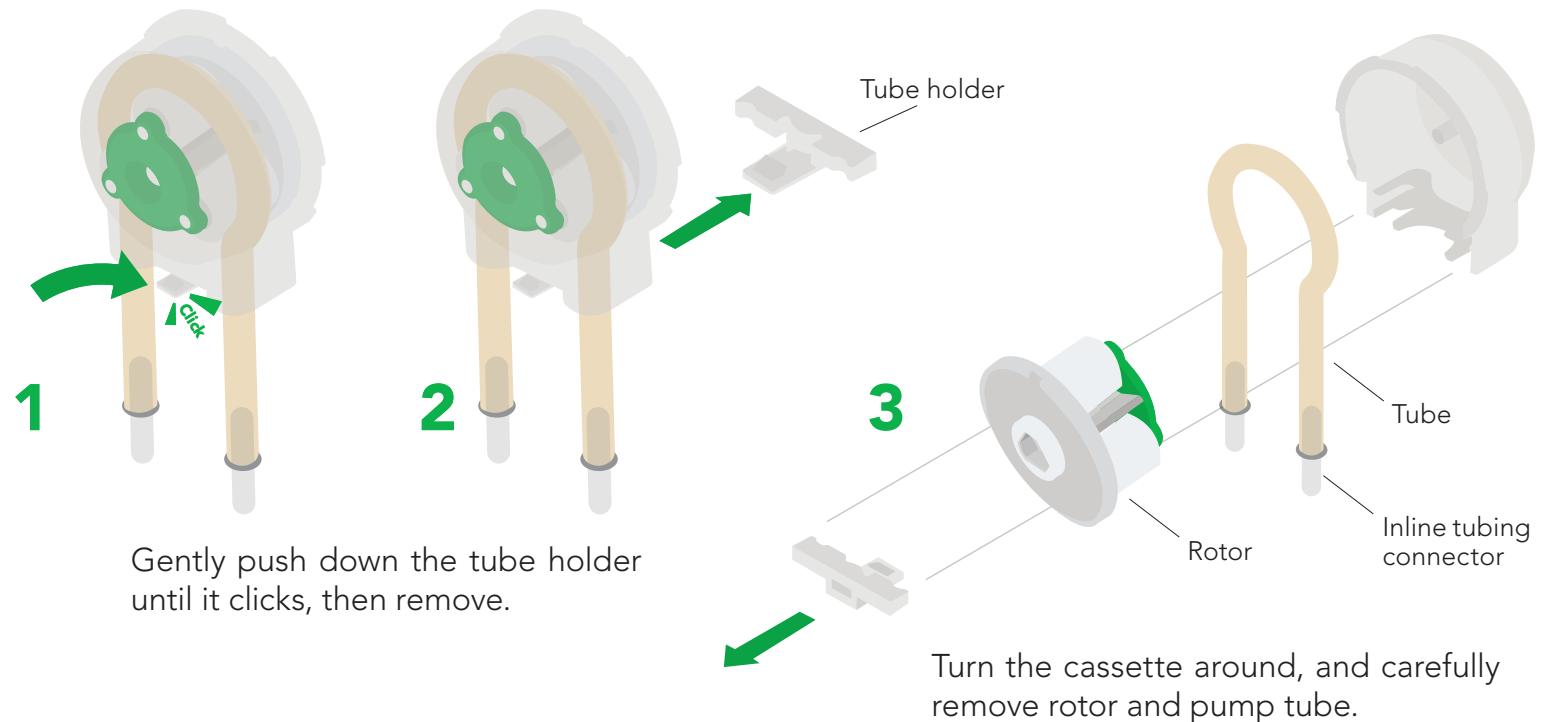
Turn cassette counterclockwise until it stops.



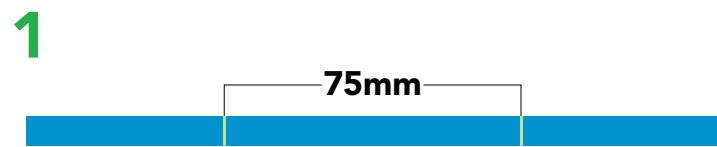
Pull cassette off the motor.

# Removing tube assembly

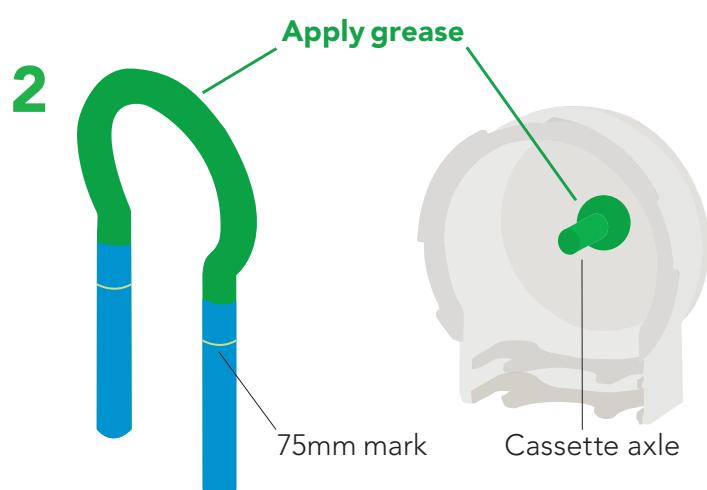
The inner workings of the cassette are fragile and must be dismantled by hand.  
Using tools can damage or break the cassette.



# Installing new tube assembly



Measure 75mm of pump tubing, and mark both ends with a soft-tip pen or marker.

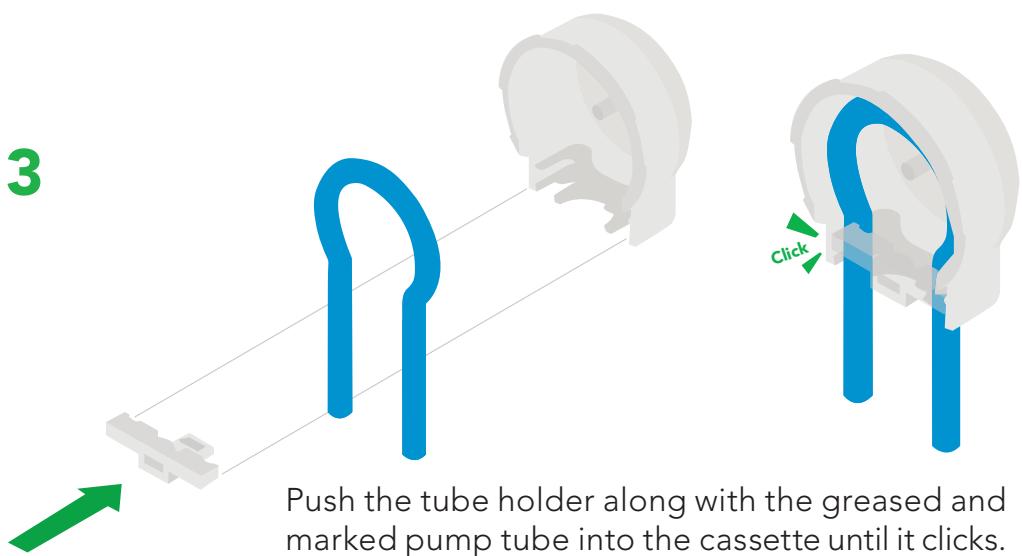


Apply silicone lubricating grease to the marked areas on both the tubing and cassette axle.

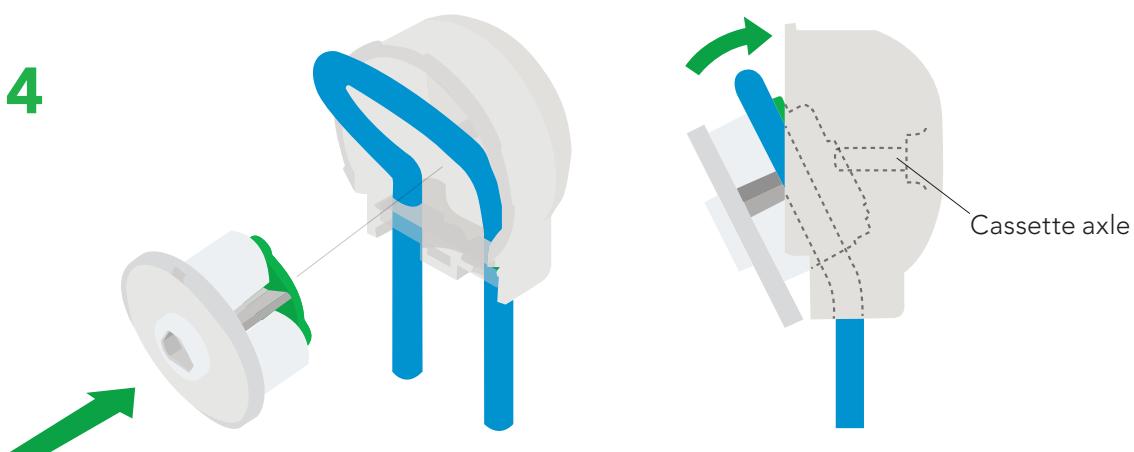
**Do not operate this device without lubrication!**

Atlas Scientific recommends using **Super Lube** silicone lubricating grease.



**3**

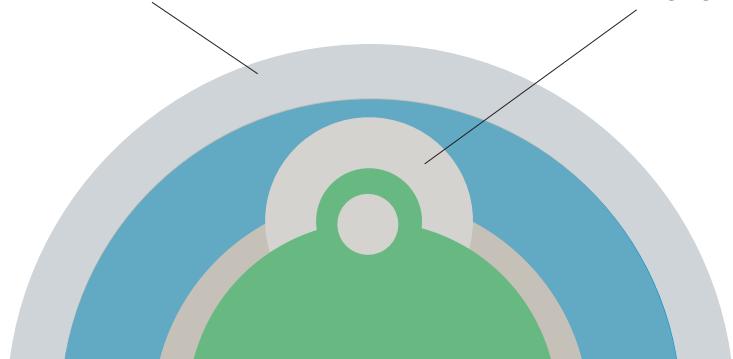
Push the tube holder along with the greased and marked pump tube into the cassette until it clicks.

**4**

Gently pull out the pump tube, and insert the rotor into the pump tube. Align pump tube and rotor with the cassette axle.

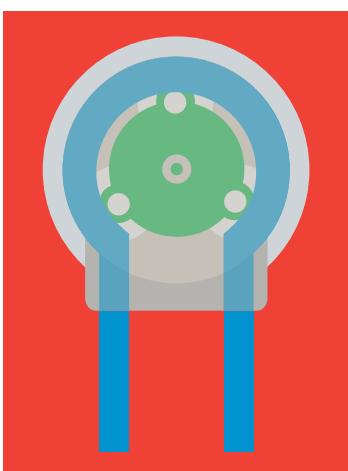
Cassette

Roller



✓ Correct

✗ Incorrect



**Make sure the pump tube is held between the roller and cassette.**

Once the tubing has been replaced, run the pump for 3–5 minutes to break in the new tubing. **Remember, this pump can be run dry and does not need to pump liquid for the 3–5 minute break in period.**

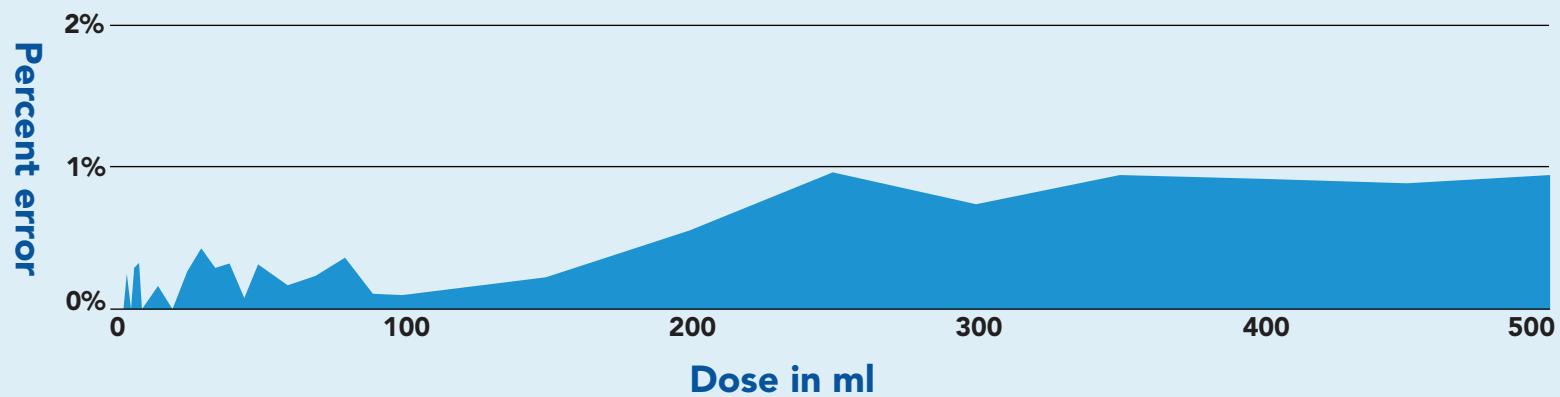
# Accuracy

Uncalibrated accuracy +/- 5%

Calibrated accuracy +/- 1%

## Volume dispensing mode

calibrated at 10ml



## Dose over time mode

calibrated at 10ml over 90 seconds

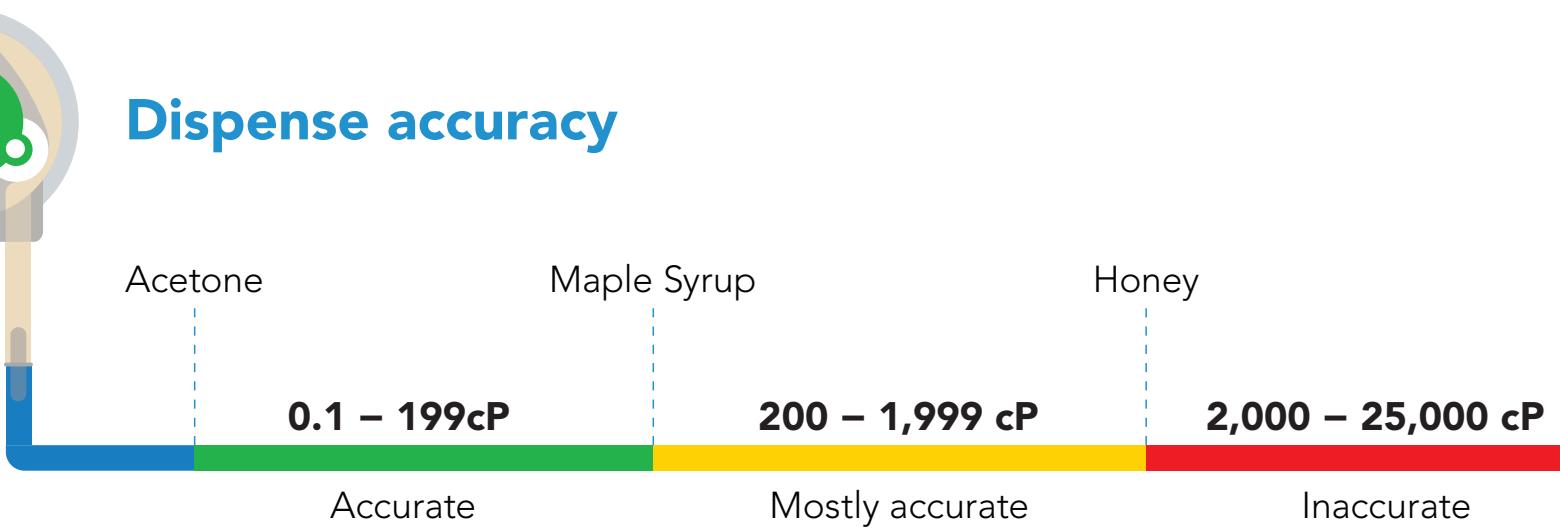


# Viscosity

The EZO-PMP™ is capable of pumping liquids within a viscosity range of **0.1 – 2,000 cP**.

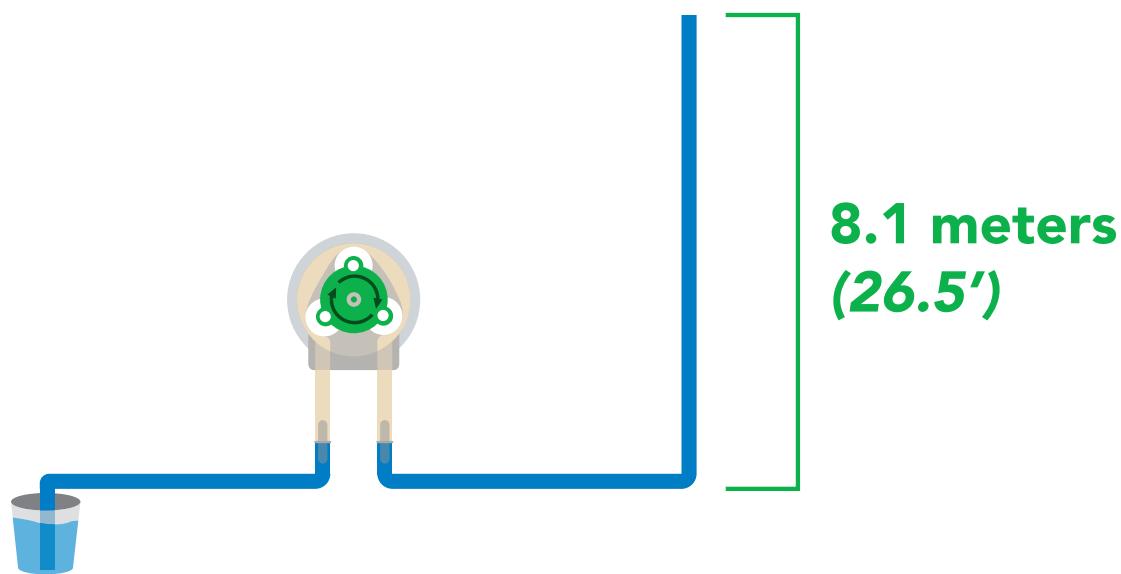
0.6	= Acetone
1	= Water
10	= Kerosene
100	= Corn Syrup
200	= Maple Syrup
2,000	= Honey
10,000	= Hershey Chocolate Syrup

## Dispense accuracy



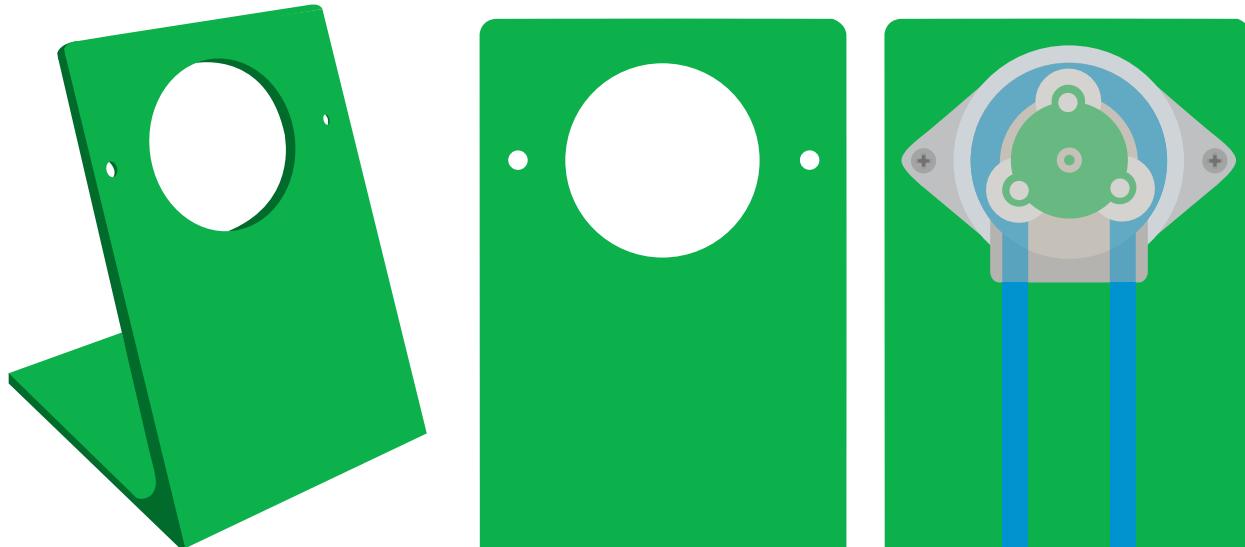
## Pump head

Pump head refers to the maximum vertical height a pump can dispense. The EZO-PMP™ has a pump head of 8.1 meters (26.5').

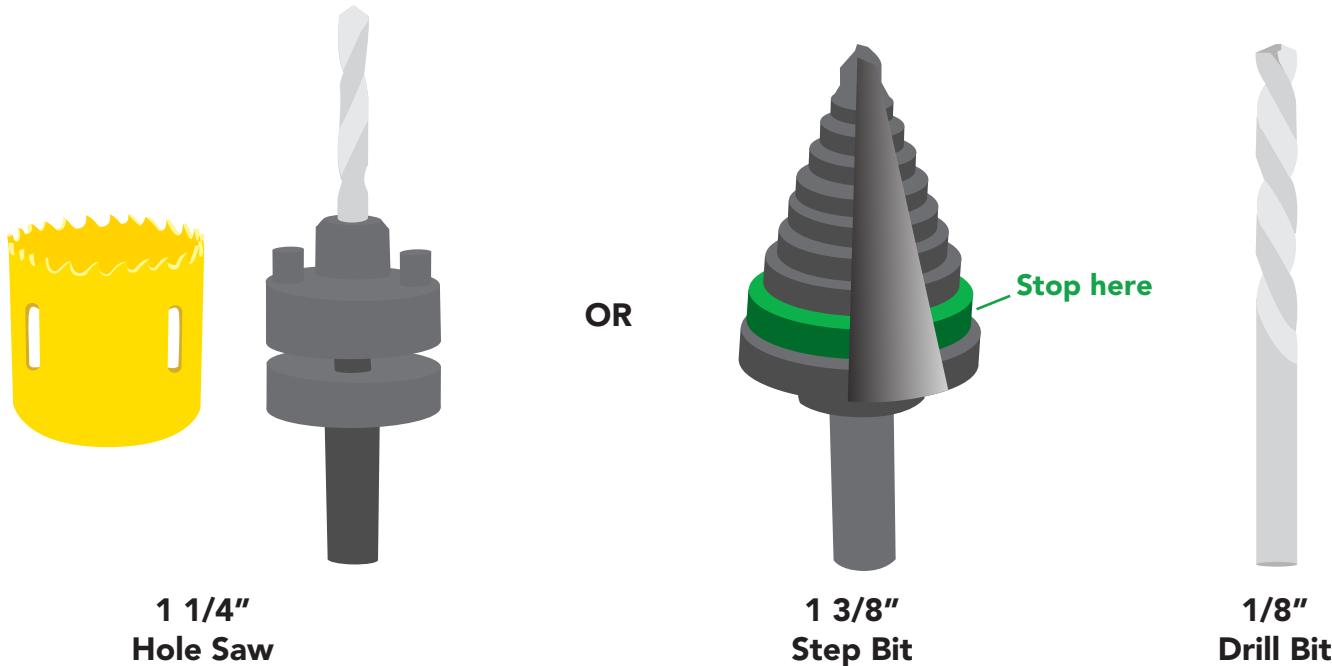


# Mounting the EZO-PMP™

There are many different ways to mount the EZO-PMP™ Embedded Dosing Pump. If you have a 3D printer you can use the dosing pump stand we created, by clicking [here](#). The dosing pump stand has been measured to perfectly fit the EZO-PMP™ and even has screw holes in place for you to help mount the dosing pump to the stand. Feel free to modify this stand design as needed.



However, if you would like to mount the EZO-PMP™ Embedded Dosing Pump into other materials, you will need the following tools:



**1 1/4"**  
**Hole Saw**

**OR**

**1 3/8"**  
**Step Bit**

**1/8"**  
**Drill Bit**

Either are fine to make the larger hole.

Perfect for screw holes.

# Datasheet change log

## Datasheet V 2.9

Revised table of contents and added invert dispensing direction command on pages 29 & 63.

## Datasheet V 2.8

Revised naming device info on pages 38 & 69.

## Datasheet V 2.7

Revised pump head information on pg 14.

## Datasheet V 2.6

Revised settings that remain when power is cut on pages 17 & 48.

## Datasheet V 2.5

Revised Total Volume Dispensed commands on pages 34 & 65.

## Datasheet V 2.4

### **Added new dispensing mode:**

"Dispense at startup" see pages 31 (UART) & 62 (I<sup>2</sup>C).

## Datasheet V 2.3

Added motor life span on pg 4.

## Datasheet V 2.2

Added page explaining the power supply needs of the EZO-PMP on pg 3.

## Datasheet V 2.1

Moved Default state to pg 14.

## Datasheet V 2.0

Revised response for the sleep command in UART mode on pg 40.

## Datasheet V 1.9

Added section on viscosity on page 13.

## Datasheet V 1.8

Added Find command on pages 22 & 53.

## Datasheet V 1.7

Added information on pump tubing on pg 4.

# Datasheet change log

## Datasheet V 1.6

Added life span of tubing and cassette on pg 3.

## Datasheet V 1.5

Added max input / output pressure info to pg 3 and pg 4.

## Datasheet V 1.4

Revised definition of response codes on pg 47.

## Datasheet V 1.3

Revised art and added pump head information on pg 11.

## Datasheet V 1.2

Revised Plock pages to show default value.

## Datasheet V 1.1

Added mounting information on pg 70.

# Firmware updates

V1.0 – Initial release (April 28, 2017)

V1.01 – (May 9, 2017)

- Fixed bug where the circuit wakes up on I2C commands sent to other addresses

V1.02 – (July 28, 2017)

- Fixed undervolt output typo

V1.03 – (June 26, 2020)

- Added command dstart, which lets the pump automatically dispense a dose on startup

V1.04 – (March 2, 2021)

- Added commands for ease of manufacturing

V1.05 (April 5, 2022)

- Expands dstart command with \* and dispense over time

# Warranty

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

## The debugging phase

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

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