

# EZO-PMP<sup>TM</sup>

## Embedded Dosing Pump

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

Accuracy **+/- 1%**

Modes of operation **Continuous dispensing  
Volume dispensing  
Constant flow rate  
Dose over time mode**

Calibration **Single point**

Supplied tubing **61 cm**

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

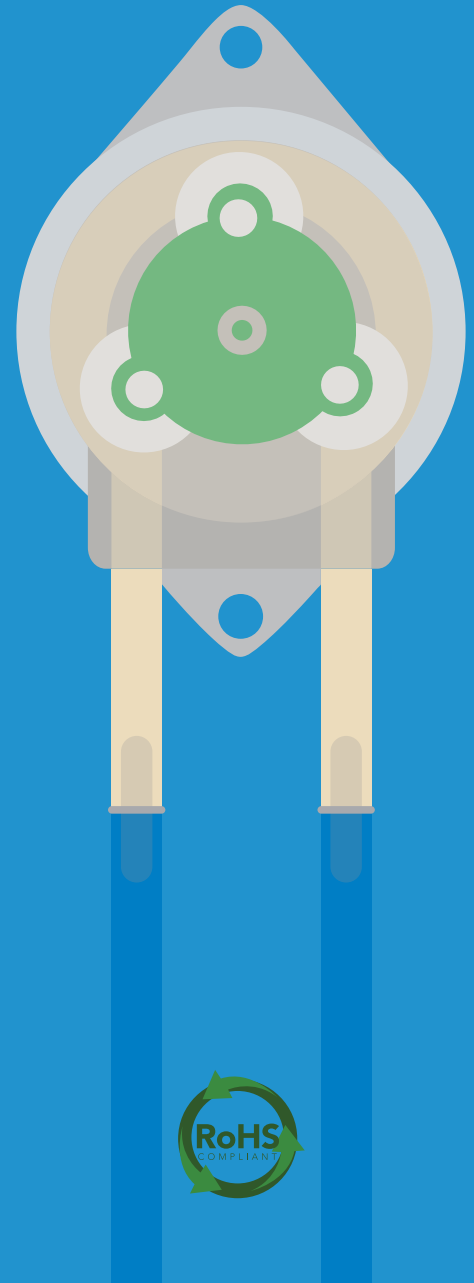
Data protocol **UART & I<sup>2</sup>C**

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

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

Pump head **2 meters**

Data format **ASCII**



# Table of contents

EZO-PMP™ dimensions	3	Operating modes	4
Power consumption	3	Calibration theory	9
Absolute max ratings	3	Accuracy	11
Operating principle	4	Available data protocols	12

## UART

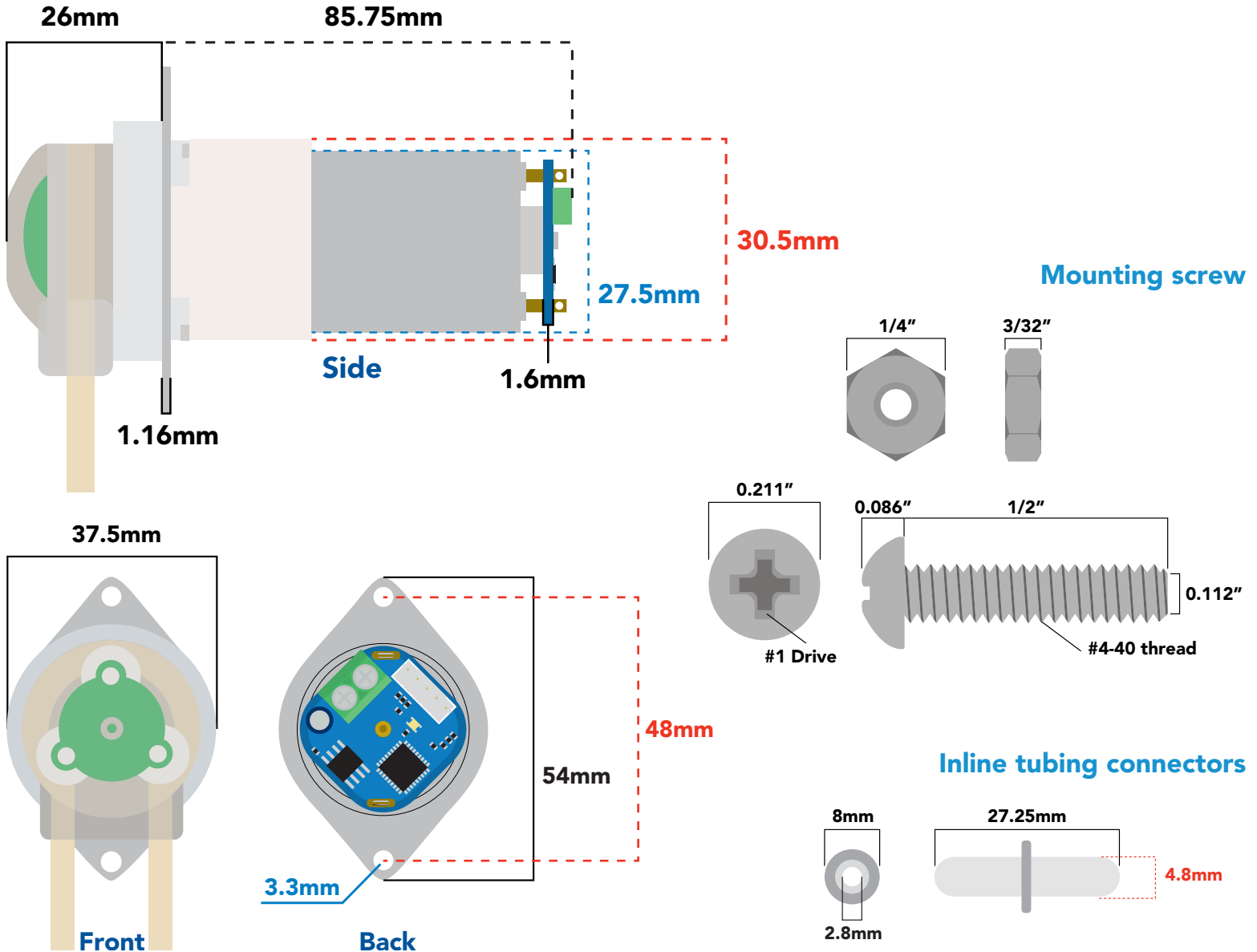
UART mode	14
Default state	15
Receiving data from device	16
Sending commands to device	17
LED color definition	18
<b>UART quick command page</b>	<b>19</b>
LED control	20
Continuous mode	21
Single reading mode	22
Continuous dispensing	23
Volume dispensing	24
Dose over time	25
Constant flow rate	26
Pause dispensing	27
Stop dispensing	28
Total volume dispensed	29
Calibration	30
Enable/disable parameters	31
Pump voltage	32
Naming device	33
Device information	34
Response codes	35
Reading device status	36
Sleep mode/low power	37
Change baud rate	38
Protocol lock	39
Factory reset	40
Change to I <sup>2</sup> C mode	41
Manual switching to I <sup>2</sup> C	42

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

I <sup>2</sup> C mode	44
Sending commands	45
Requesting data	46
Response codes	47
LED color definition	48
<b>I<sup>2</sup>C quick command page</b>	<b>49</b>
LED control	50
Single report mode	51
Continuous dispensing	52
Volume dispensing	53
Dose over time	54
Constant flow rate	55
Pause dispensing	56
Stop dispensing	57
Total volume dispensed	58
Calibration	59
Enable/disable parameters	60
Pump voltage	61
Device information	62
Reading device status	63
Sleep mode/low power	64
Protocol lock	65
I <sup>2</sup> C address change	66
Factory reset	67
Change to UART mode	68
Manual switching to UART	69

Mounting the EZO-PMP™	70
Datasheet change log	71
Warranty	72

# EZO-PMP™ dimensions



## Power consumption

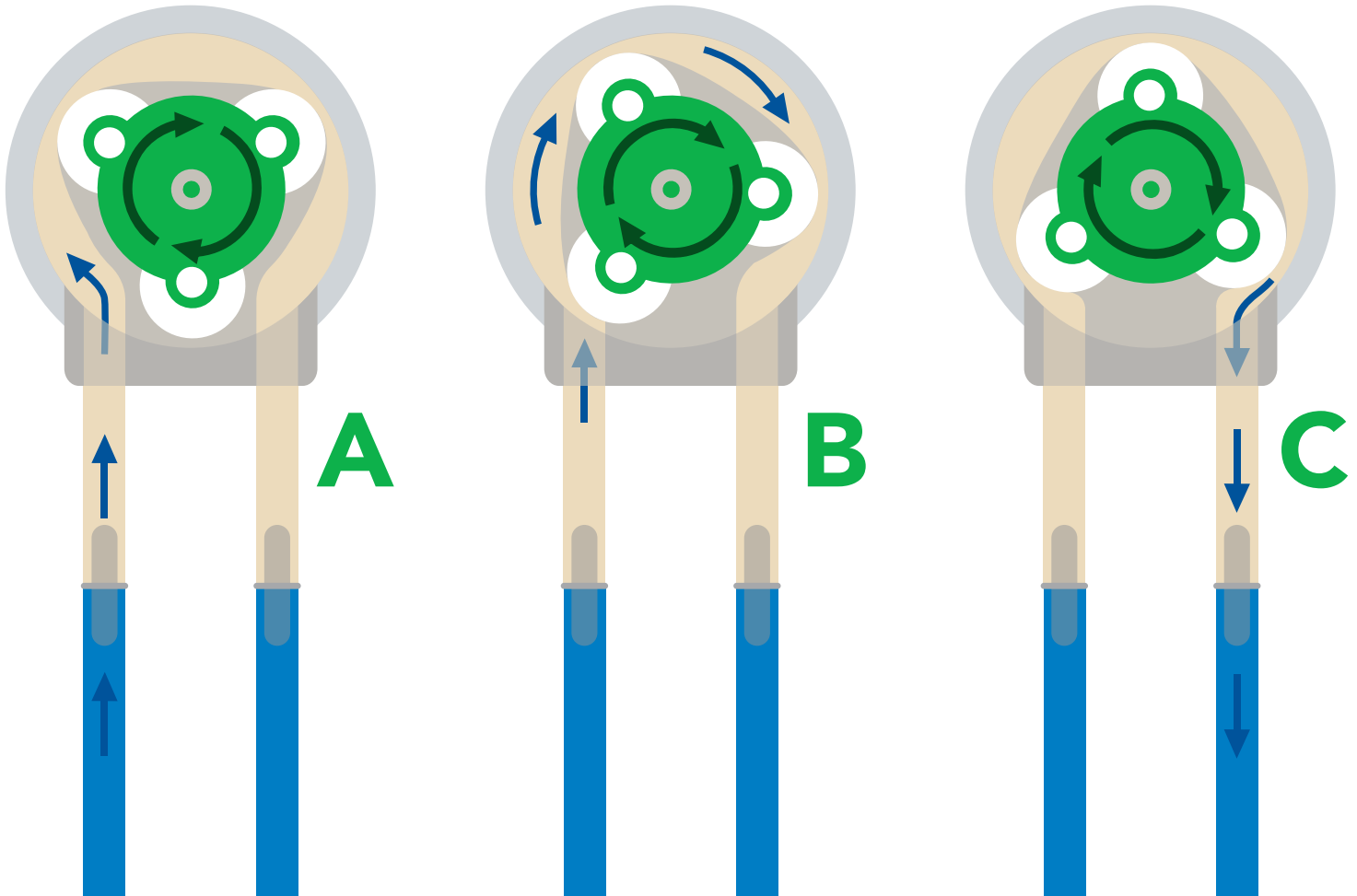
	LED	MAX	STANDBY	SLEEP
<b>5V</b>	ON	13.7 mA	13.4 mA	0.415 mA
	OFF	13.1 mA	12.8 mA	
<b>3.3V</b>	ON	12.5 mA	12.4 mA	0.13 mA
	OFF	12.3 mA	12.2 mA	
<b>Motor</b>	12V = ~400mA		24V = ~200mA	

## 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

# 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)

## Constant flow rate

Pump a specific volume per minute

## Dose over time mode

Pump a specific volume over a set time

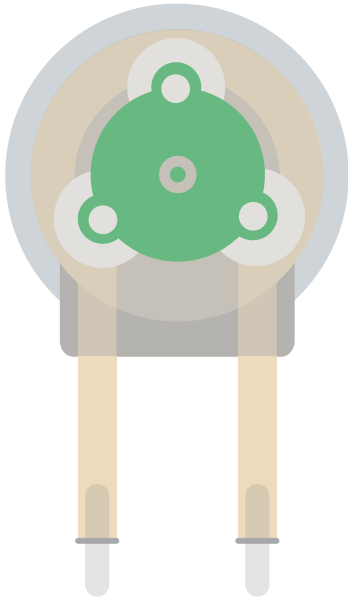
**Volume is always in ml.**

**This device requires two power supplies**

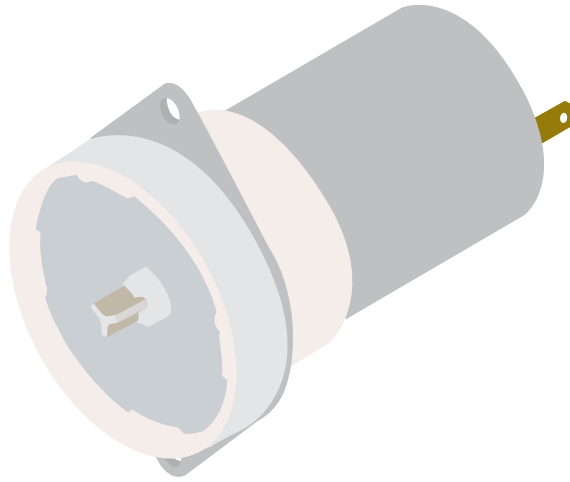
3.3V–5.5V for the control system

12V–24V to drive the motor

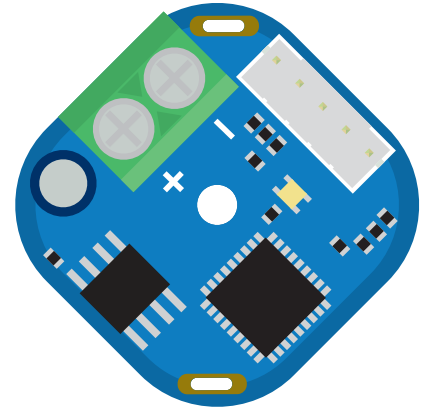
**The Atlas Scientific EZO-PMP™ consists of three main components.**



**Cassette**



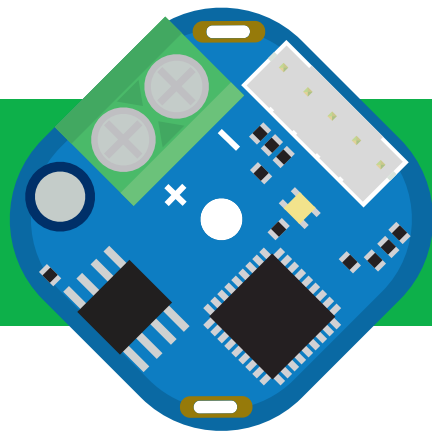
**12 volt motor**



**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

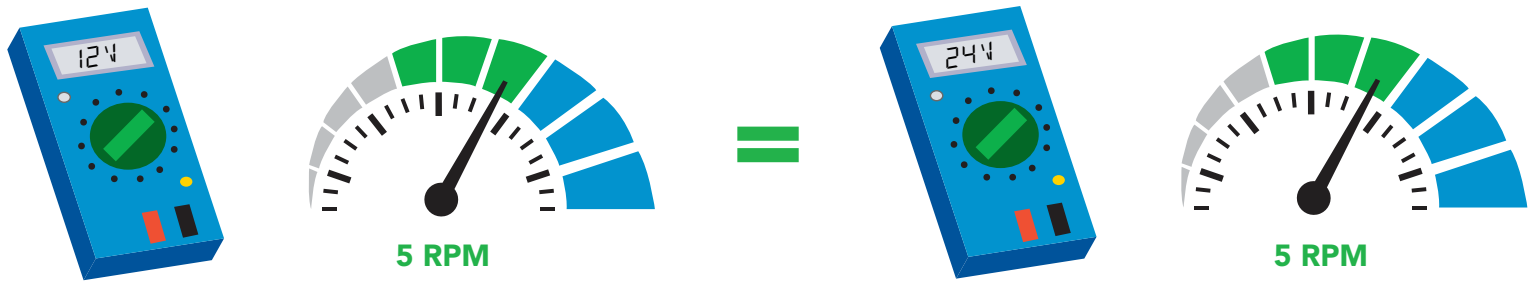
**Data and power cable pinout**

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



# Pump speed vs. voltage

There is no change in pump speed at different voltages.

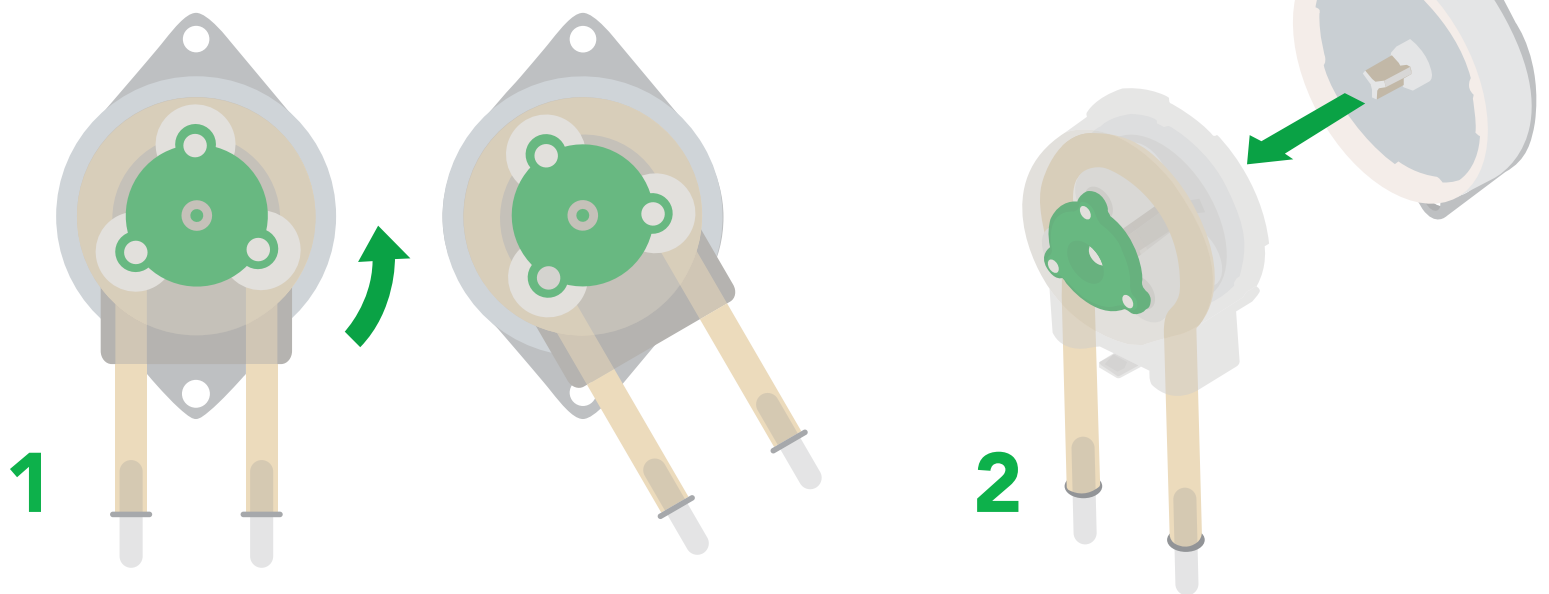


# Interrupt pin

When the pump is dispensing the interrupt pin goes high.



# Removing cassette

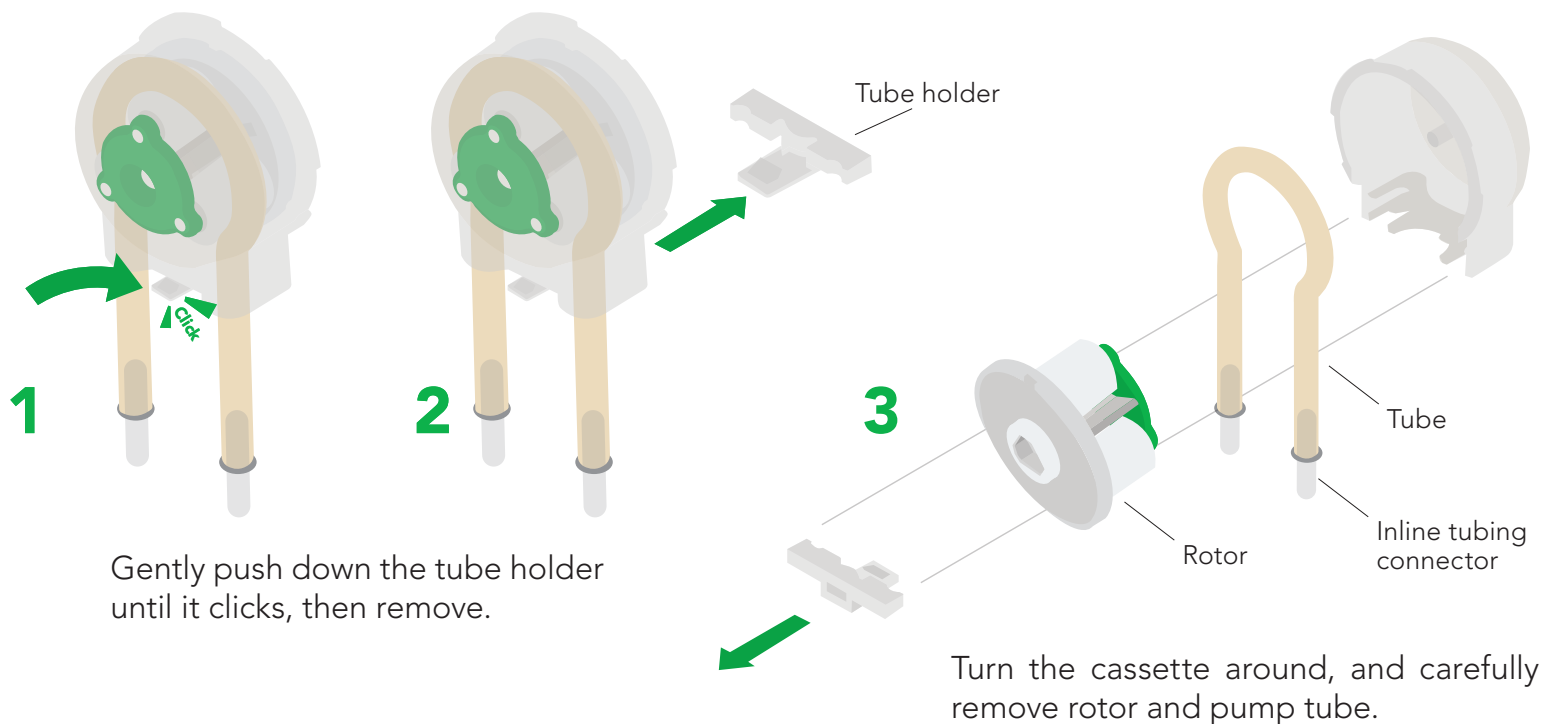


1 Turn cassette counterclockwise until it stops.

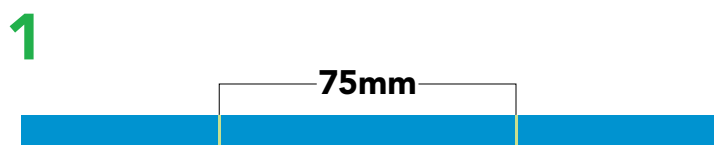
2 Pull cassette off the motor.

# Removing tube assembly

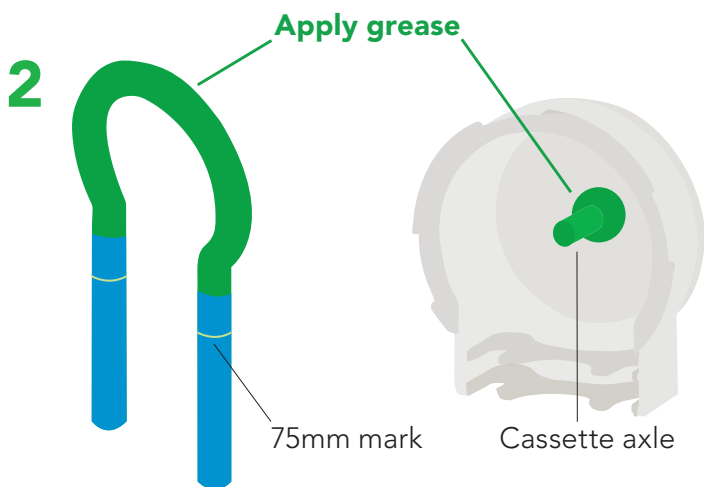
The inner workings of the cassette are fragile and must be dismantled by hand. Using tools can damage or break the cassette.



# Installing new tube assembly



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



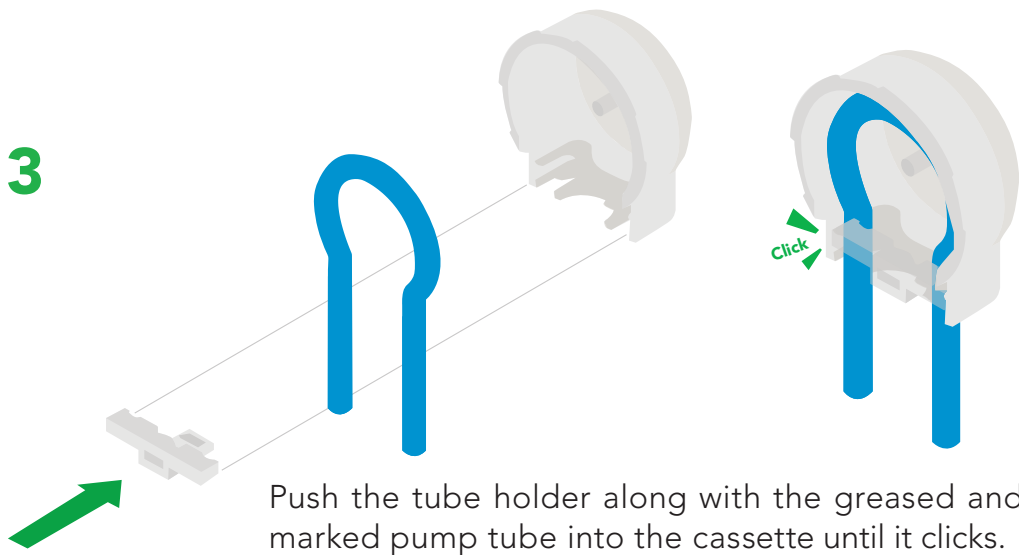
Apply silicone lubricating grease to the marked areas on both the tubing and cassette axle.

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

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

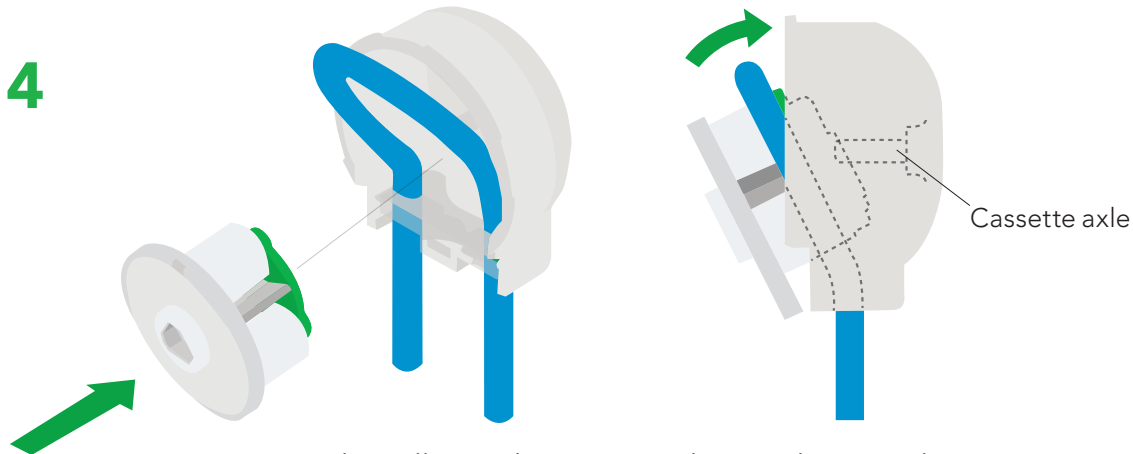


3

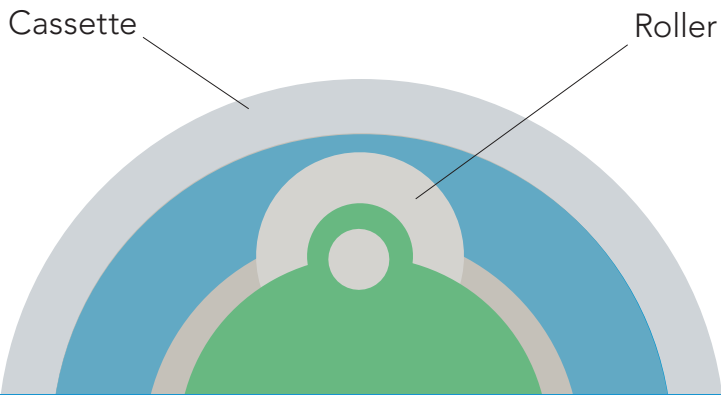


Push the tube holder along with the greased and marked pump tube into the cassette until it clicks.

4



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

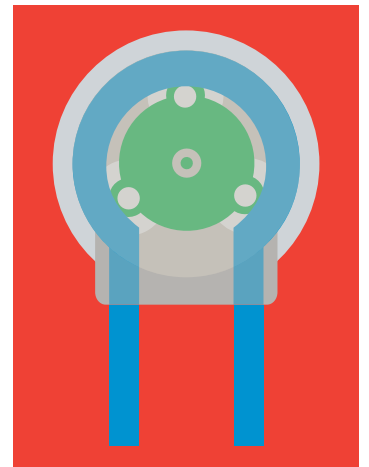


**Make sure the pump tube is held between the roller and cassette.**

✓ **Correct**



**X Incorrect**



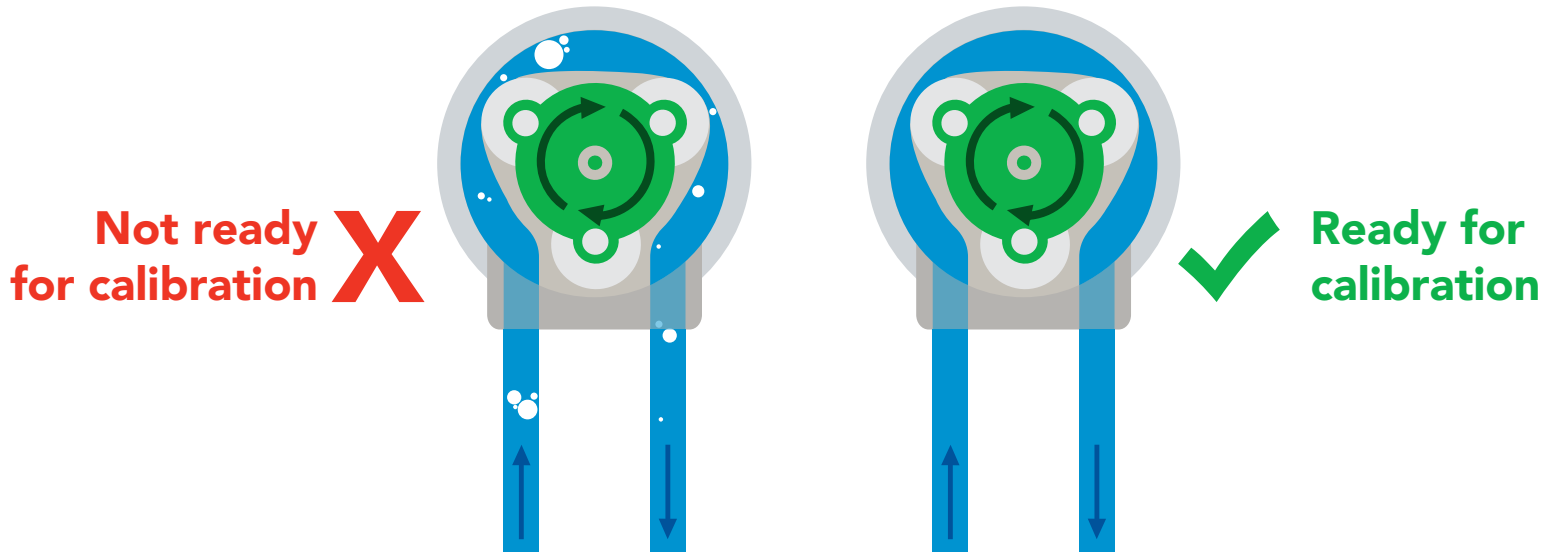
Once the tubing has been replaced, run the pump for 3–5 minutes to break in the new tubing. **Remember, this pump can be run dry and does not need to pump liquid for the 3–5 minute break in period.**



# 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.

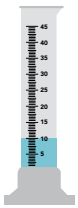


## Calibration types

Volume calibration  
Volume over time calibration

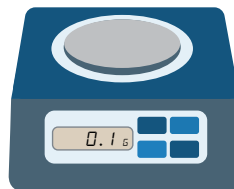
Calibration is optional. Both types of calibration are independent of each other and can be done at any time. Calibration can be done at any volume however; Atlas Scientific recommends using volumes above 5ml.

## Equipment needed for calibration



An accurate graduated cylinder of at least 10ml.

Or



An accurate scale with a resolution of at least 0.1 grams

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

# Calibration procedure

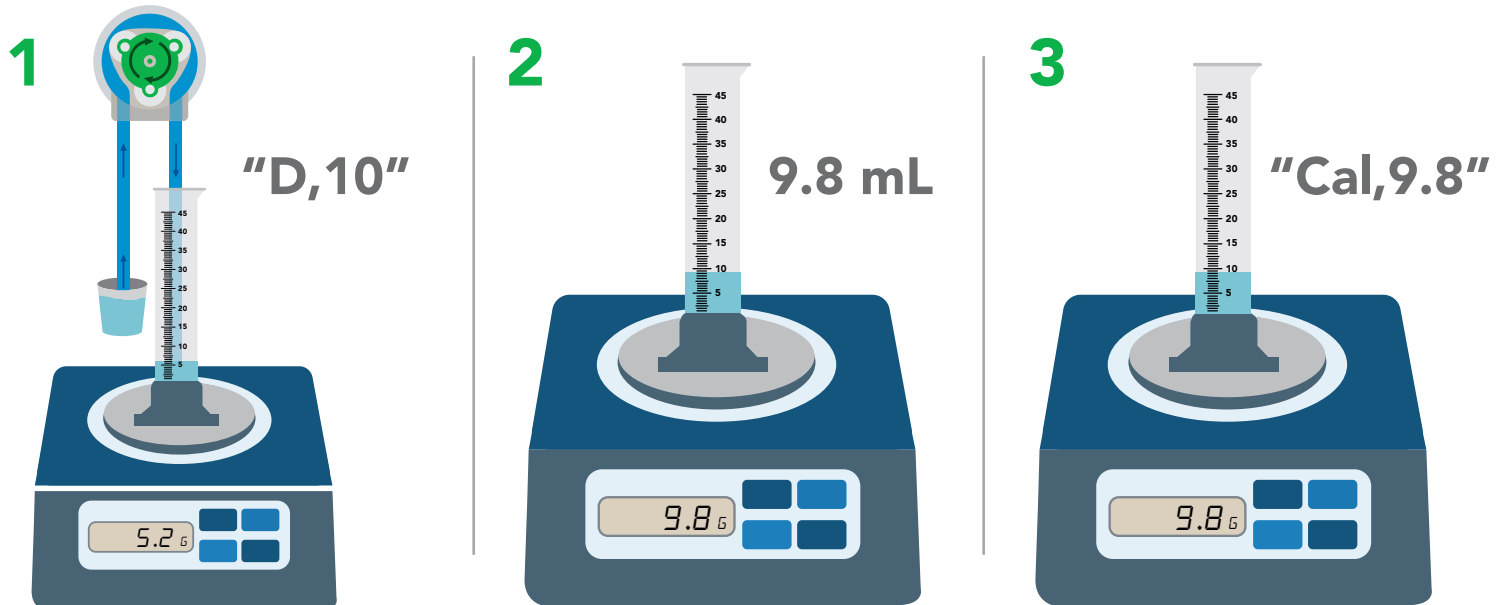
Calibration should be done with water and not a chemical

**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. It has not been calibrated specifically to the volume used during the calibration procedure (10 ml). It has now been calibrated to all volumes.

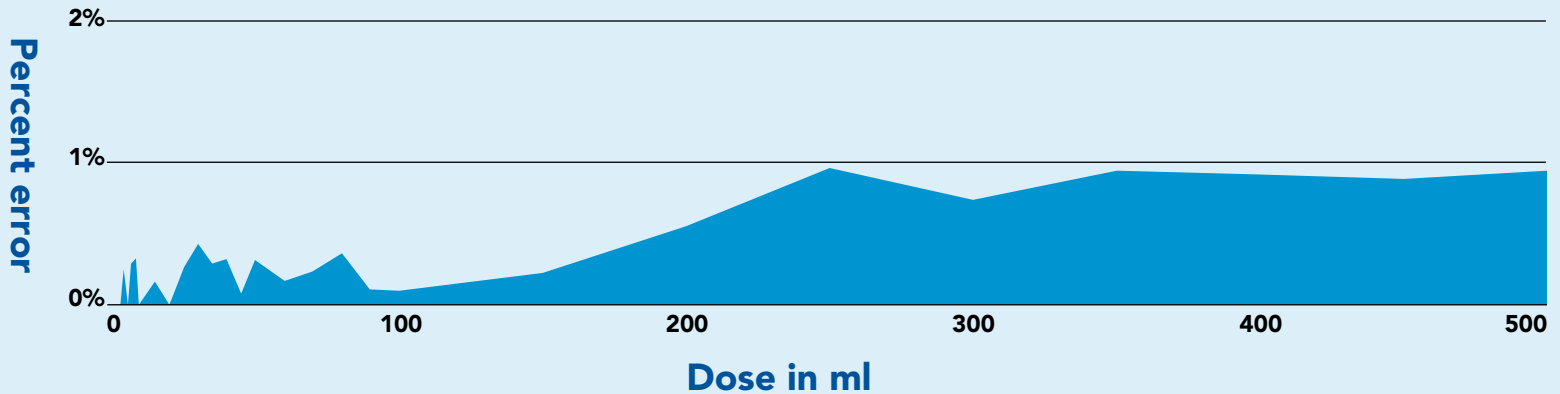
Use the same procedure to perform a volume over time calibration.

# Accuracy

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

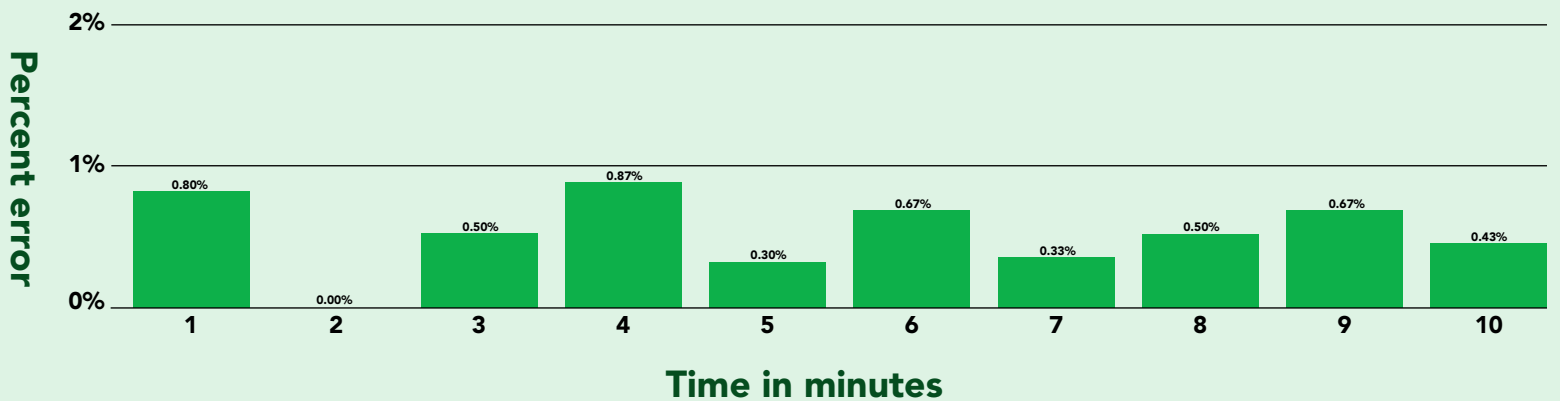
## Volume dispensing mode

calibrated at 10ml



## Dose over time mode

calibrated at 10ml over 90 seconds



# Pump head

Pump head refers to the maximum vertical height a pump can dispense. The EZO-PMP™ has a pump head of 2 meters (6.5').

# ✓ Available data protocols

# UART

Default

# I<sup>2</sup>C

# ✗ Unavailable data protocols

# SPI

# Analog

# RS-485

# Mod Bus

# 4–20mA

# UART mode

## Settings that are retained if power is cut

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

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

- Find
- Sleep mode

# UART mode

8 data bits      no parity  
1 stop bit        no flow control

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

**RX**  
Data in

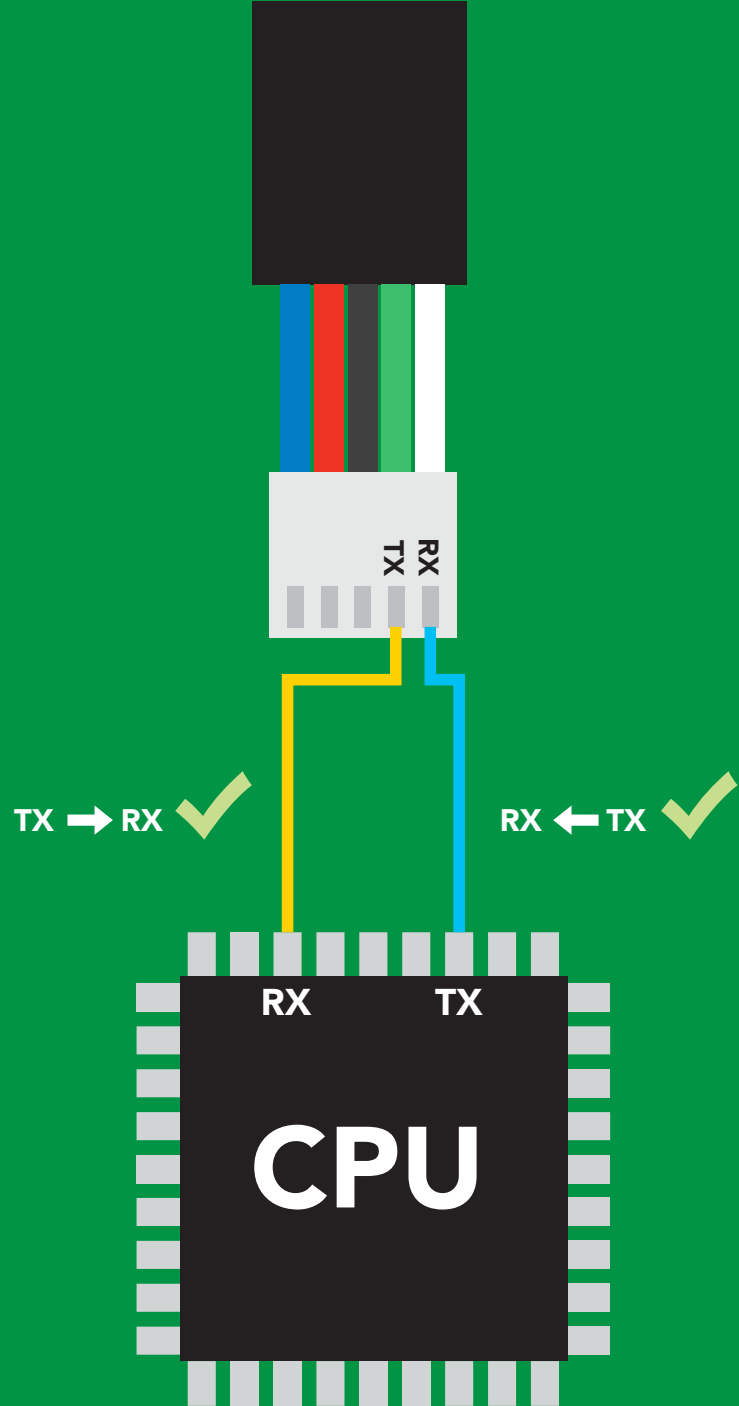


**TX**  
Data out



**Vcc** 3.3V – 5.5V

0V  0V



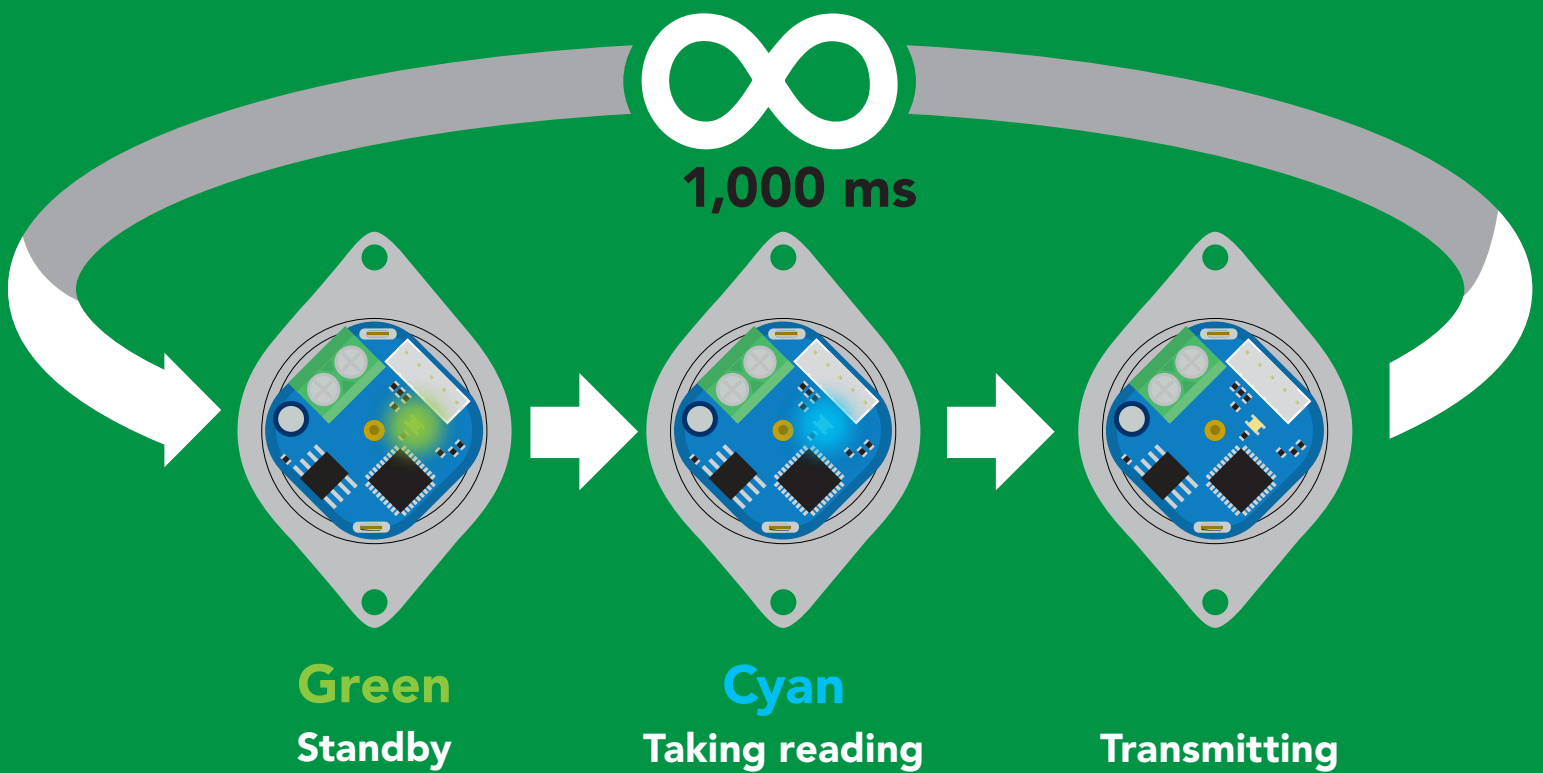
## Data format

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

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

# Default state

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



# Receiving data from device

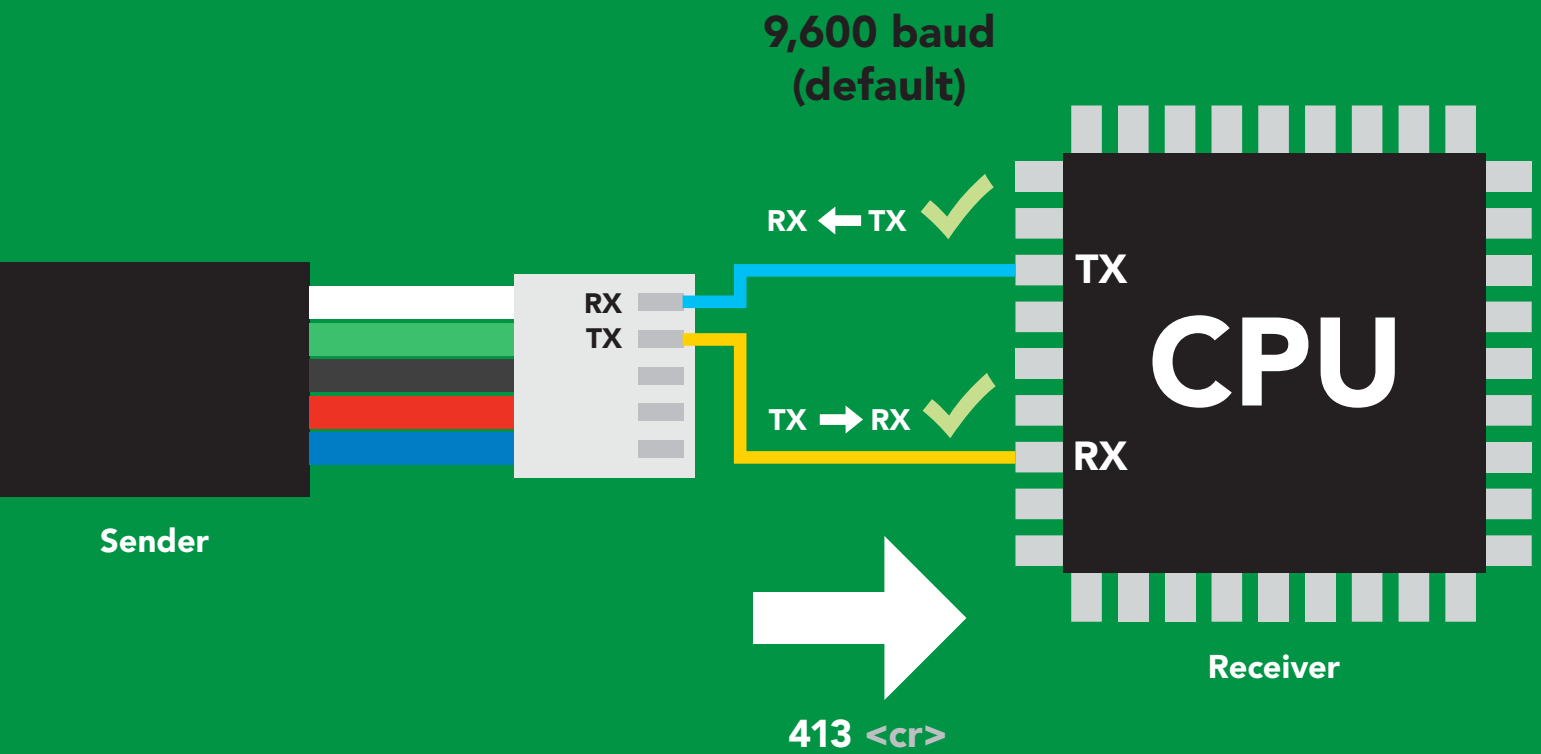
2 parts

ASCII data string

Command

Carriage return <cr>

Terminator



## Advanced

ASCII: 4 1 3 <cr>

Hex: 34 31 33 0D

Dec: 52 49 51 13



# Sending commands to device

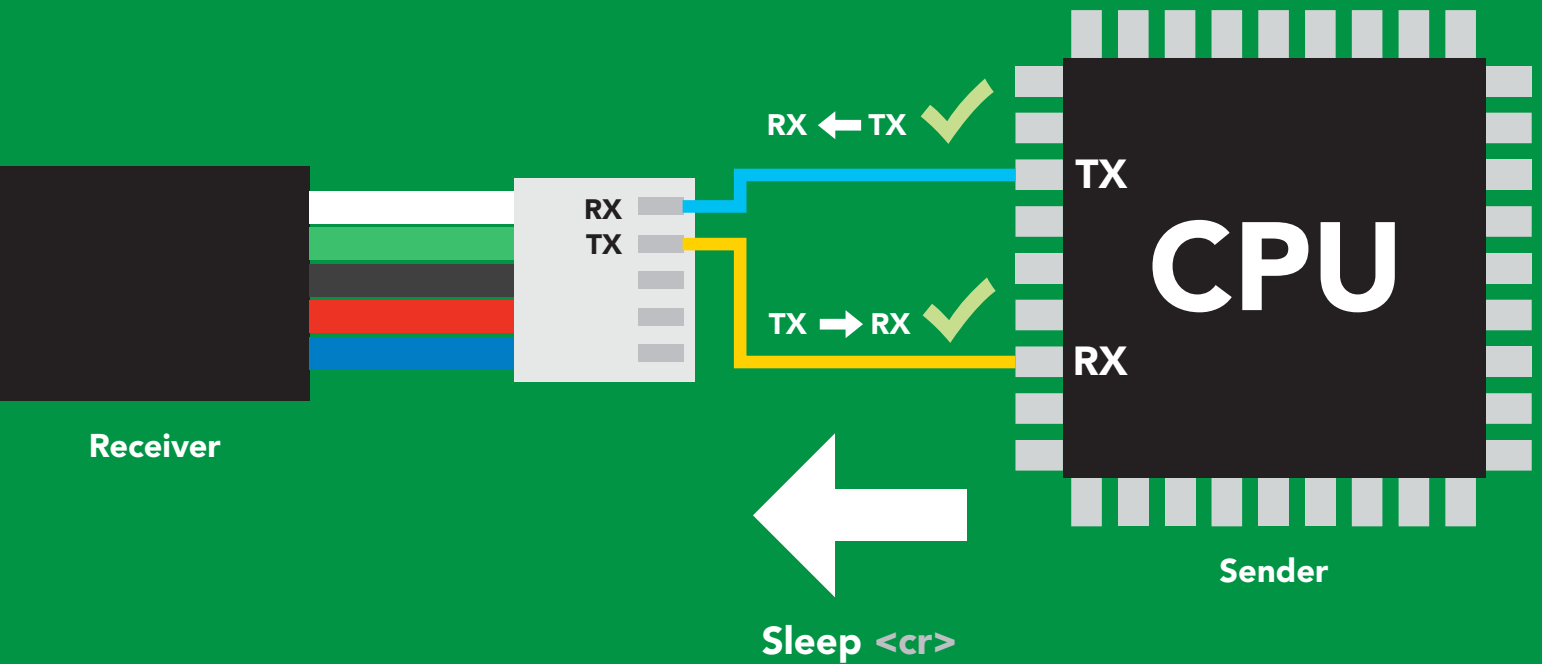
2 parts

**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



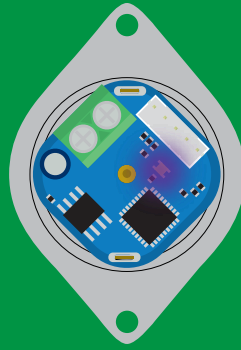
**Green**

UART standby



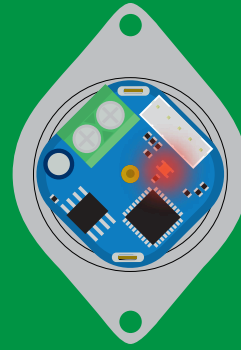
**Cyan**

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
Baud	change baud rate	pg. 38	9,600
C	enable/disable continuous mode	pg. 21	enabled
Cal	performs calibration	pg. 30	n/a
D	dispense modes	pg. 23 – 26	n/a
Factory	enable factory reset	pg. 40	n/a
i	device information	pg. 34	n/a
I2C	change to I <sup>2</sup> C mode	pg. 41	not set
L	enable/disable LED	pg. 20	enabled
Name	set/show name of device	pg. 33	not set
O	enable/disable parameters	pg. 31	all enabled
P	pause dispensing	pg. 27	n/a
Plock	enable/disable protocol lock	pg. 39	disabled
Pv	check pump voltage	pg. 32	n/a
R	returns a single reading	pg. 22	n/a
Sleep	enter sleep mode/low power	pg. 37	n/a
Status	retrieve status information	pg. 36	enable
Tv	total volume dispensed	pg. 29	n/a
X	stop dispensing	pg. 28	n/a
*OK	enable/disable response codes	pg. 35	enable

# LED control

## Command syntax

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

L,0 <cr> LED off

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

## Example

## Response

L,1 <cr>

\*OK <cr>

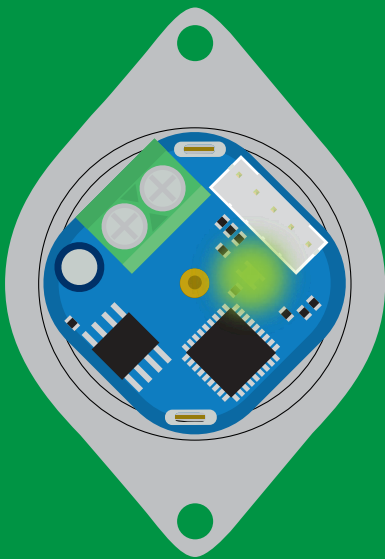
L,0 <cr>

\*OK <cr>

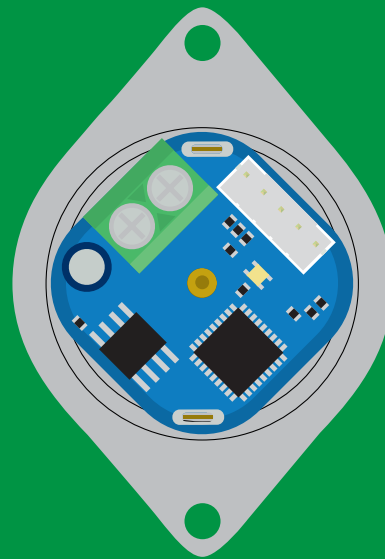
L,? <cr>

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

\*OK <cr>



L,1



L,0

# 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

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,-405 <cr>

\*OK <cr> 405 ml will be dispensed *in reverse*

D,? <cr>

?D,22.50,0 <cr>

\*OK <cr>

## Response breakdown

?D,22.50,0

↑ 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

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

## Response

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

**DC,? <cr>**

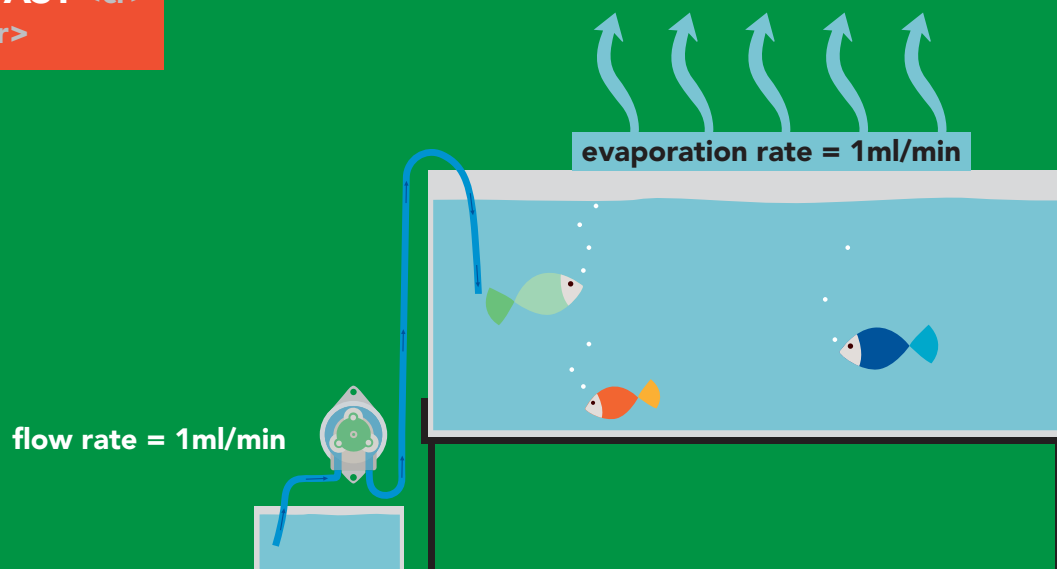
**?MAXRATE,58.5 <cr>**

**\*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>**



# Pause dispensing

## Command syntax

Issue the command again to resume dispensing

**P** <cr> pauses the pump during dispensing

**P,?** <cr> pause status

## Example

## Response

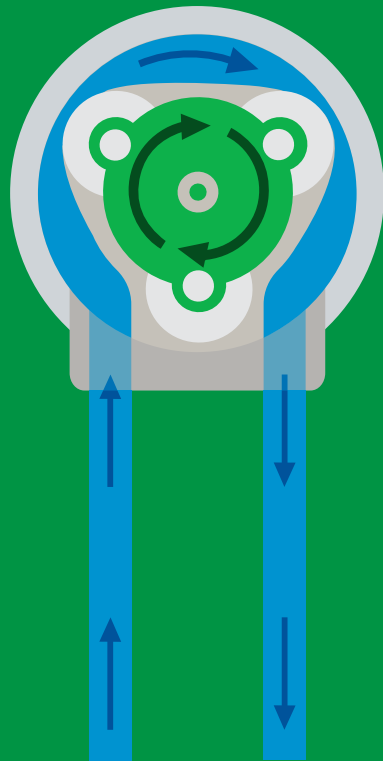
**P** <cr>

**\*OK** <cr>

**P,?** <cr>

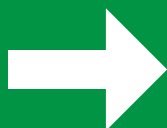
**?P,1** <cr> **or** **?P,0** <cr>  
paused unpaused

**\*OK** <cr>



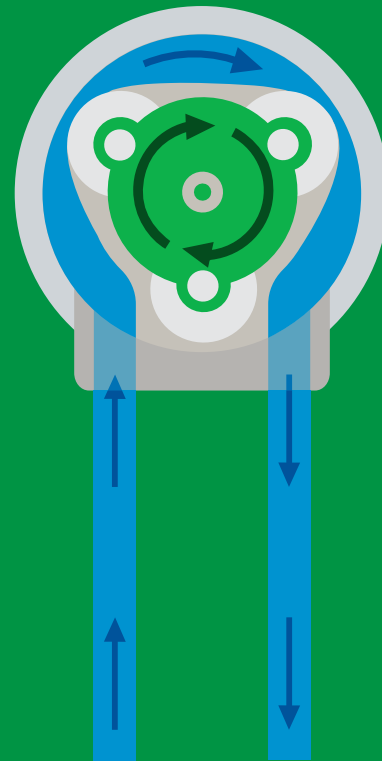
dispensing

**P**



paused

**P**



dispensing

# Stop dispensing

## Command syntax

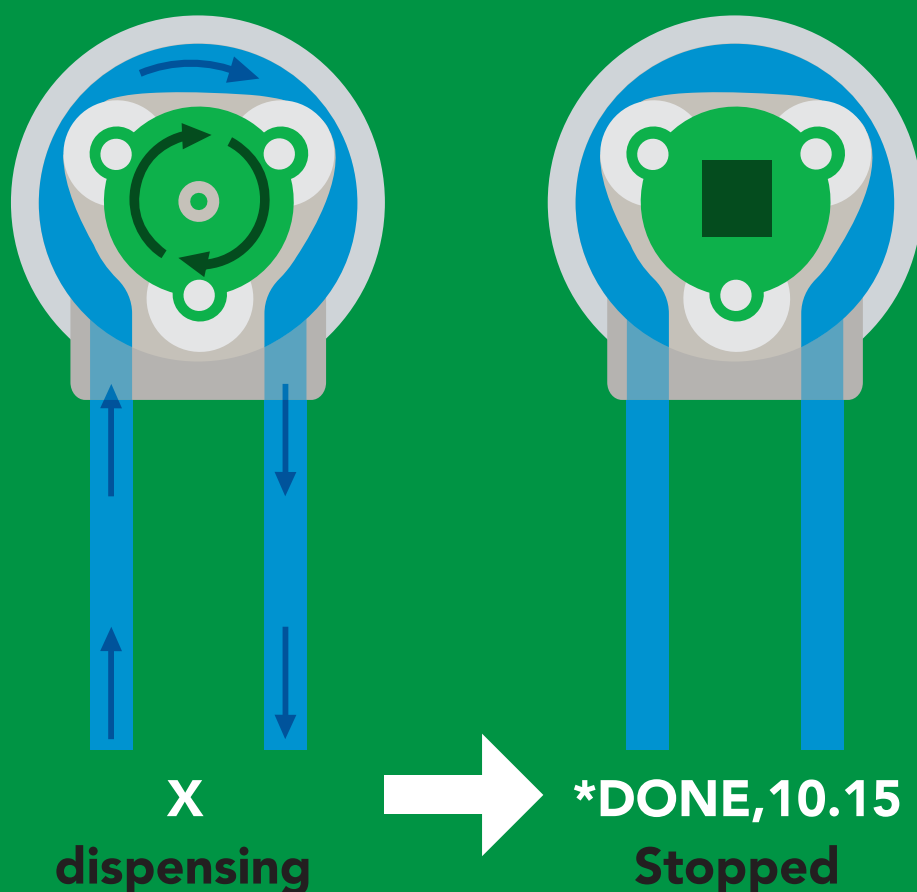
**X** <cr> stop dispensing

### Example

**X** <cr>

### Response

**\*DONE,v** <cr> v = volume dispensed



# 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

### Response

**TV,? <cr>**

**?total,434.50 <cr>**

**ATV,? <cr>**

**?total,623.00 <cr>**

**Clear <cr>**

**\*OK <cr> total now 0.00**

# 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

### Response

**Cal,24.01** <cr>

**\*OK** <cr>

**Cal,clear** <cr>

**\*OK** <cr>

**Cal,?** <cr>

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

# Enable/disable parameters from output string

## Command syntax

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

### Example

### Response

O,V,1 <cr>

\*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

**PV,?** <cr>

### Response

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

## Response breakdown

**?PV, 13.86**



Pump input voltage



# Naming device

## Command syntax

Name,n <cr> set name

Name,? <cr> show name

n =

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

Up to 16 ASCII characters

## Example

Name,zzt <cr>

## Response

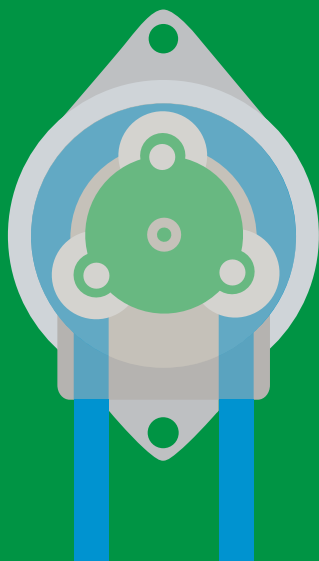
\*OK <cr>

Name,? <cr>

?Name,zzt <cr>

\*OK <cr>

Name,zzt



\*OK <cr>



Name,?



Name,zzt <cr>

\*OK <cr>

# Device information

## Command syntax

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

### Example

```
i <cr>
```

### Response

```
?i,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

## Response

**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>

## 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
- \*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

```
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

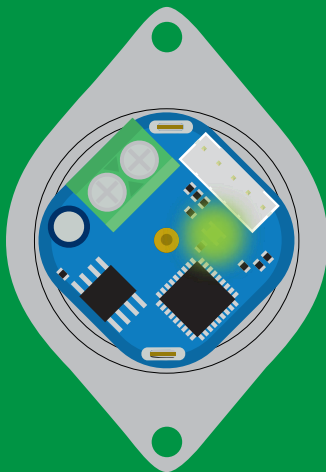
\*WA <cr> wakes up device

5V

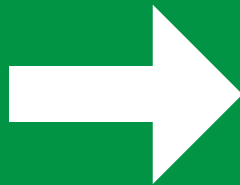
STANDBY	SLEEP
13.4 mA	0.415 mA

3.3V

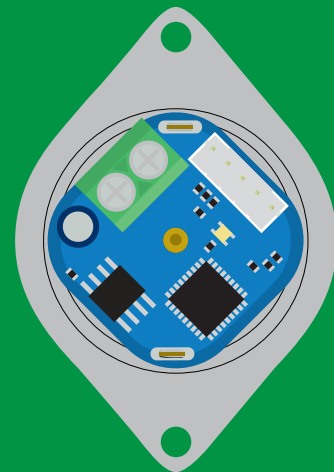
12.4 mA	0.13 mA
---------	---------



Standby  
13.4 mA



Sleep <cr>



Sleep  
0.415 mA

# Change baud rate

## Command syntax

Baud,n <cr> change baud rate

### Example

Baud,38400 <cr>

### Response

\*OK <cr>

Baud,? <cr>

?Baud,38400 <cr>

\*OK <cr>

n =

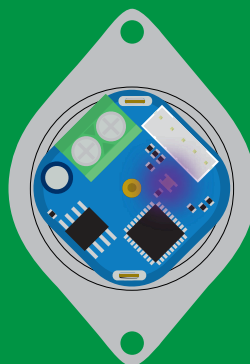
- 300
- 1200
- 2400
- 9600 default**
- 19200
- 38400
- 57600
- 115200



Standby



Baud,38400 <cr>



Changing  
baud rate

\*OK <cr>



(reboot)



Standby

# Protocol lock

## Command syntax

Locks device to UART mode.

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

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

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

## Example

## Response

**Plock,1** <cr>

**\*OK** <cr>

**Plock,0** <cr>

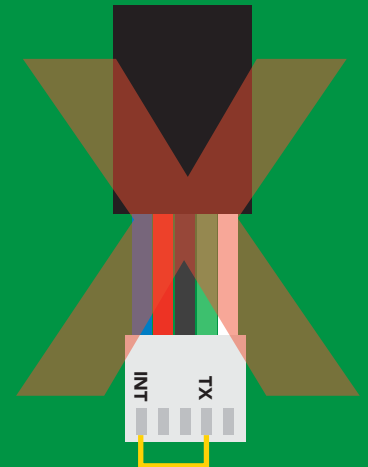
**\*OK** <cr>

**Plock,?** <cr>

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

**Plock,1**

**I2C,100**



**\*OK** <cr>

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

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

**\*ER** <cr>

# Factory reset

## Command syntax

Clears calibration  
LED on  
"\*OK" enabled

Factory <cr> enable factory reset

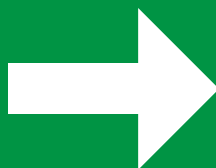
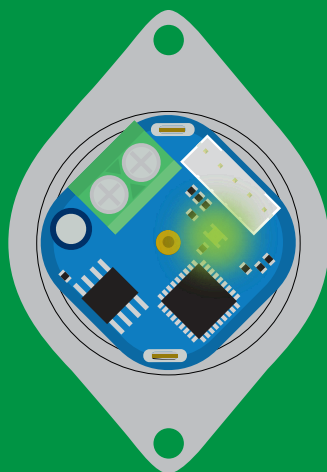
## Example

## Response

Factory <cr>

\*OK <cr>

Factory <cr>



(reboot)



\*OK <cr>

\*RS <cr>

\*RE <cr>

Baud rate will not change



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

## Command syntax

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

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

n = any number 1 – 127

## Example

## Response

I2C,100 <cr>

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

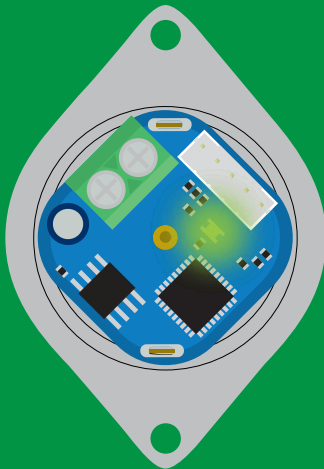
## Wrong example

## Response

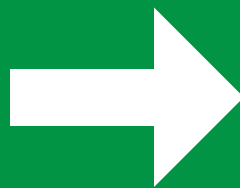
I2C,139 <cr> n ≠ 127

\*ER <cr>

I2C,100



Green  
\*OK <cr>



(reboot)



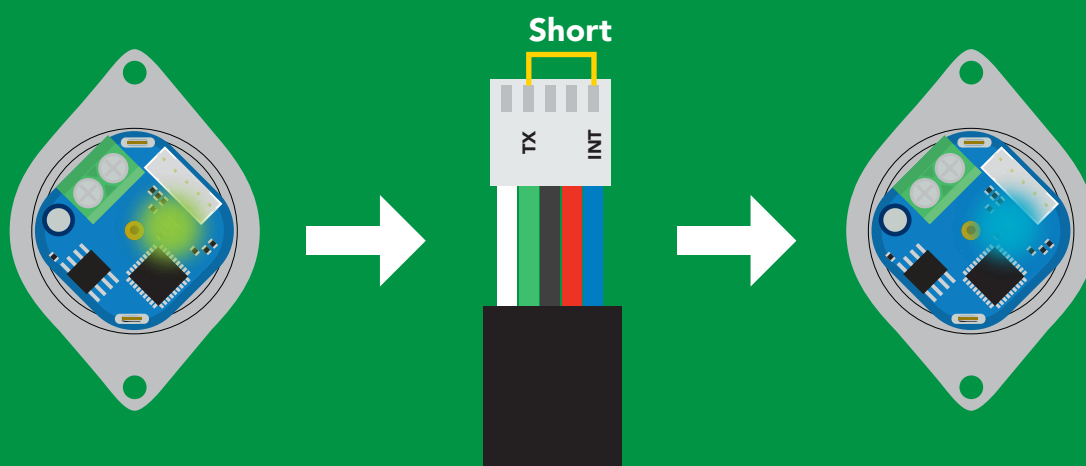
Blue  
now in I<sup>2</sup>C mode

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

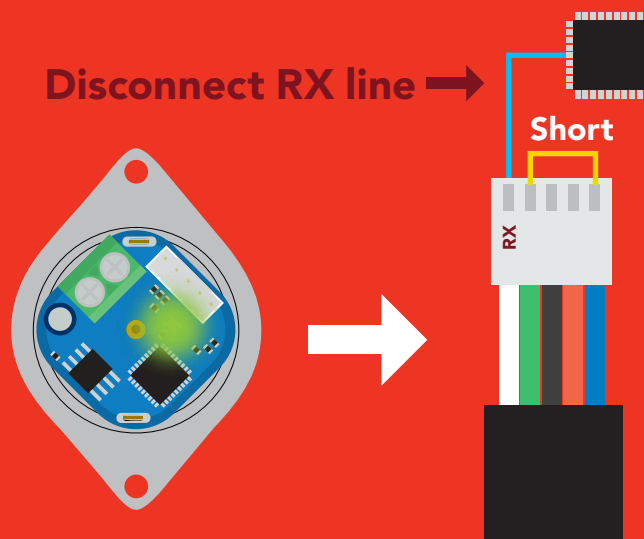
- Make sure Plock is set to 0
- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to INT
- Confirm RX is disconnected
- Connect ground (power on)
- Wait for LED to change from **Green** to **Blue**
- Disconnect ground (power off)
- Reconnect all data and power

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
- LED control
- Protocol lock
- Software switch to UART mode

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

- Find
- Sleep mode

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

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

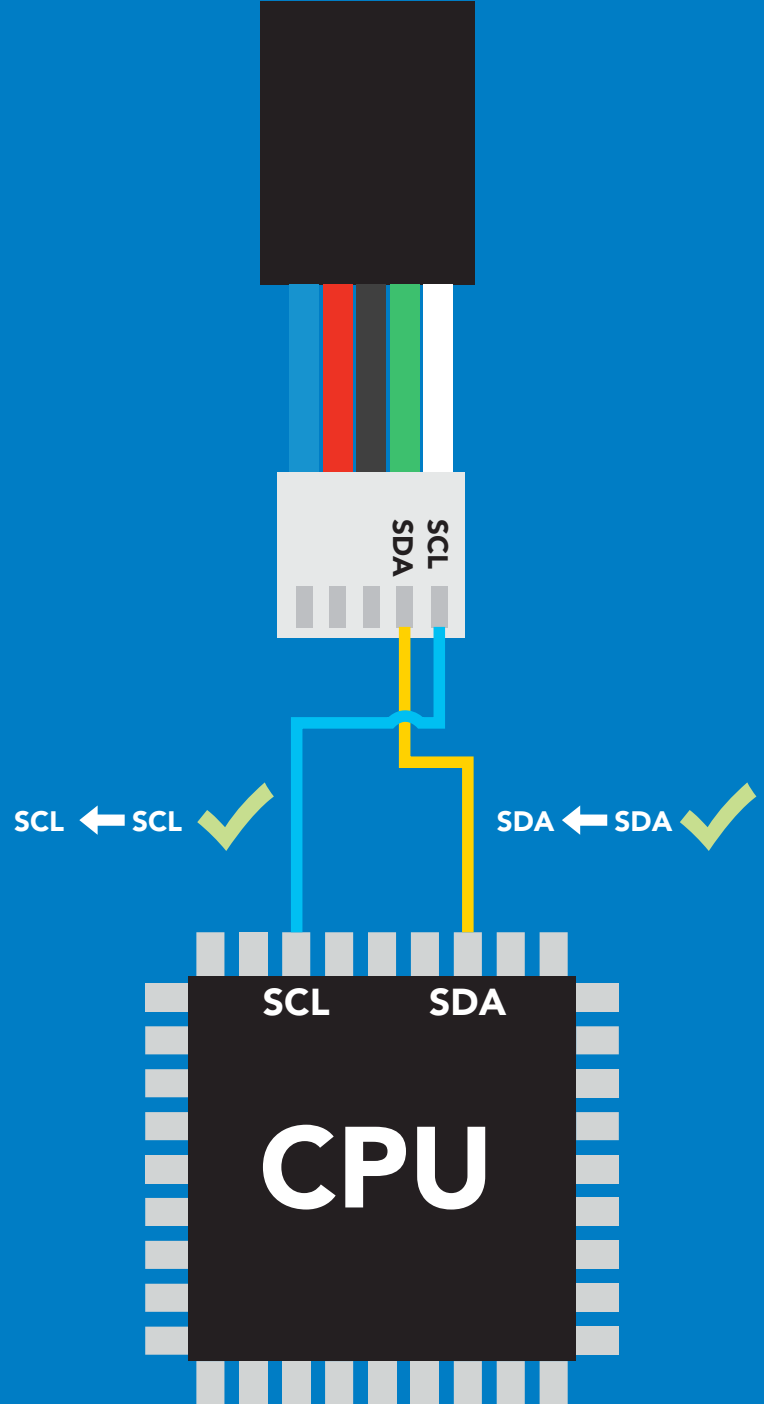
Vcc 3.3V – 5.5V

Clock speed 100 – 400 kHz

SDA 

SCL 

  
0V VCC 0V



## Data format

Reading volume

Units ml

Encoding ASCII

Format string

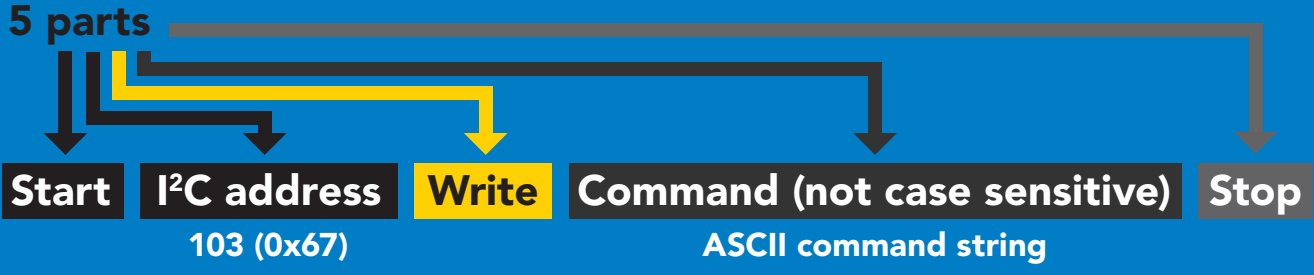
Data type floating point

Decimal places 3

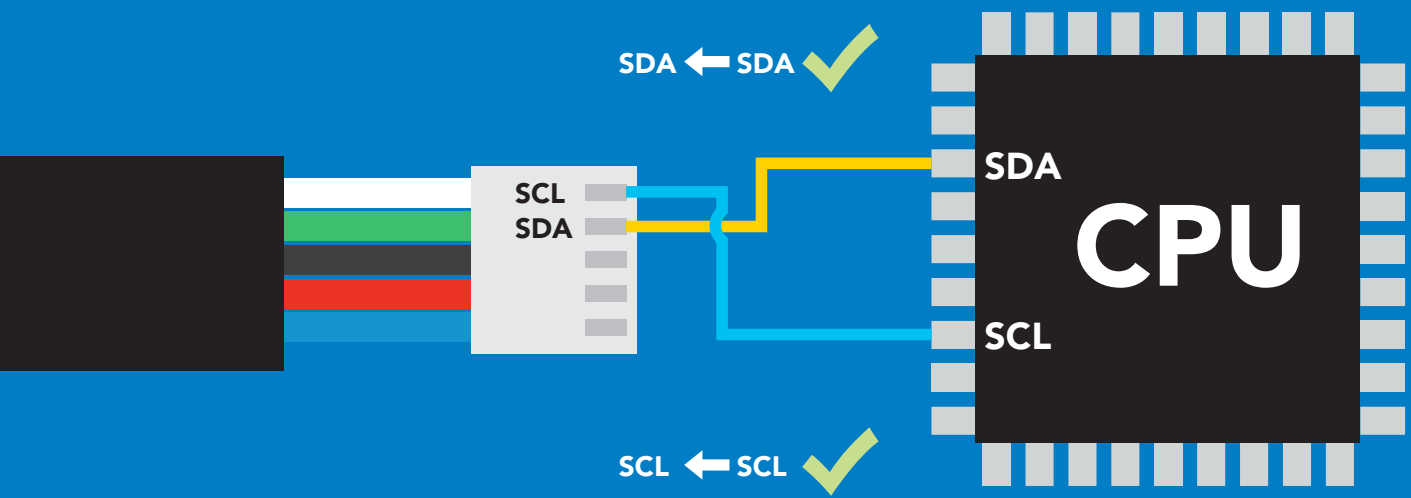
Smallest string 3 characters

Largest string 39 characters

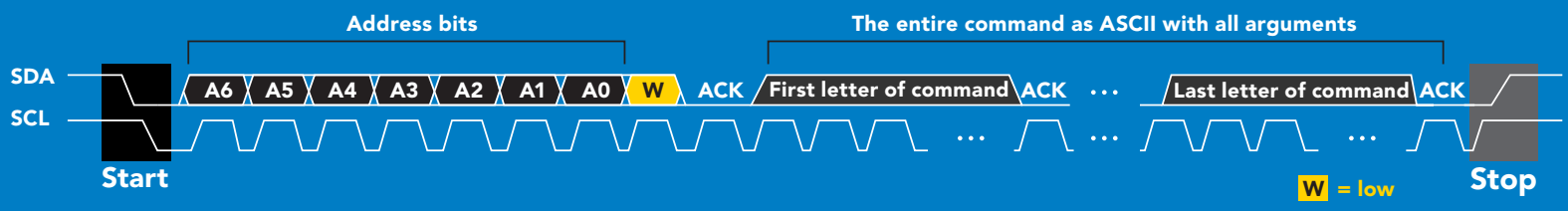
# Sending commands to device



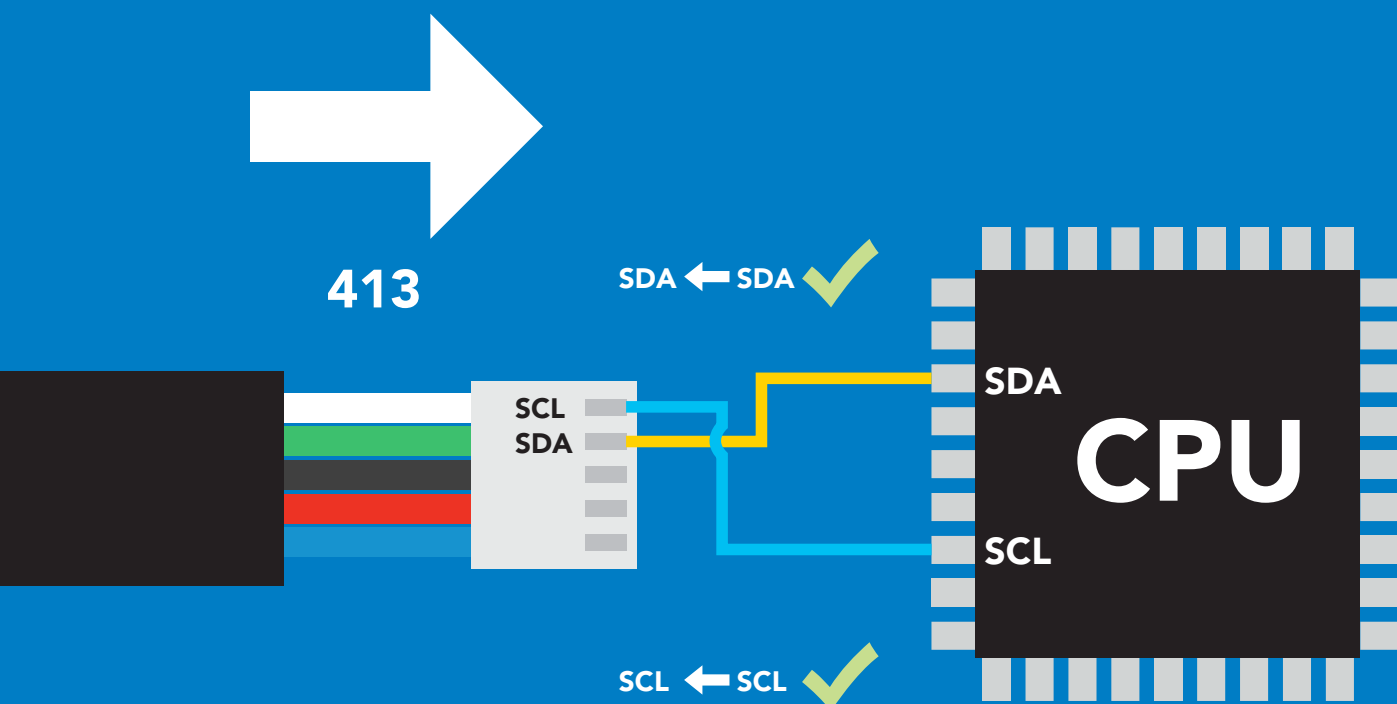
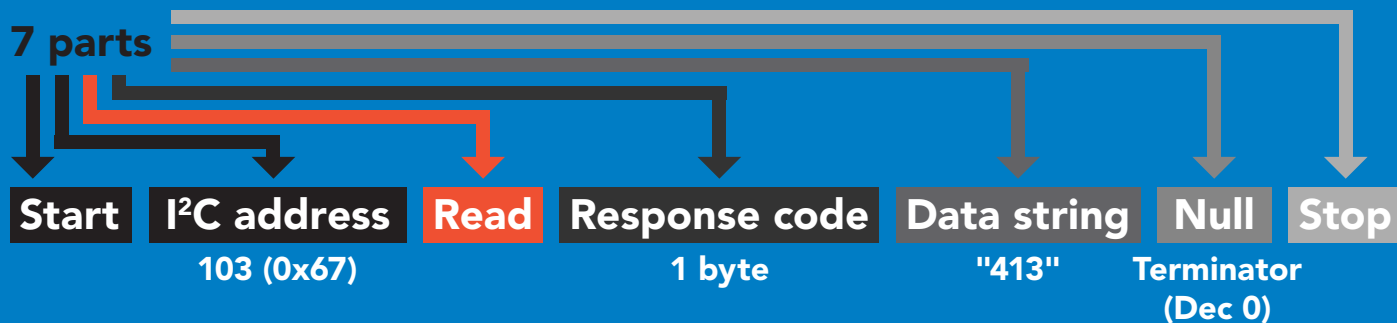
## Example



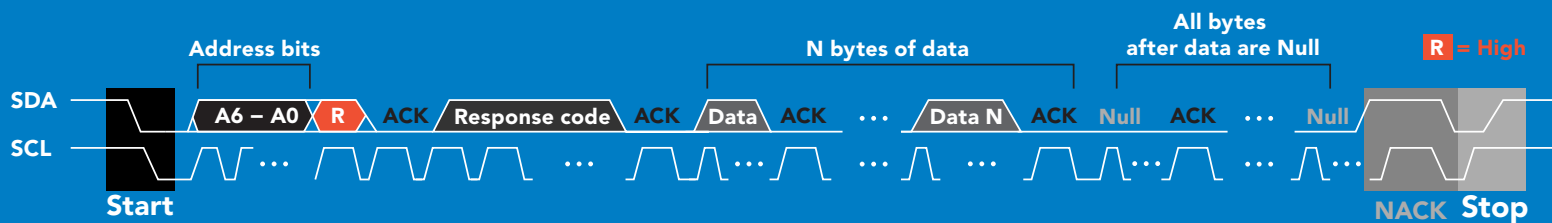
## Advanced



# Requesting data from device



## Advanced

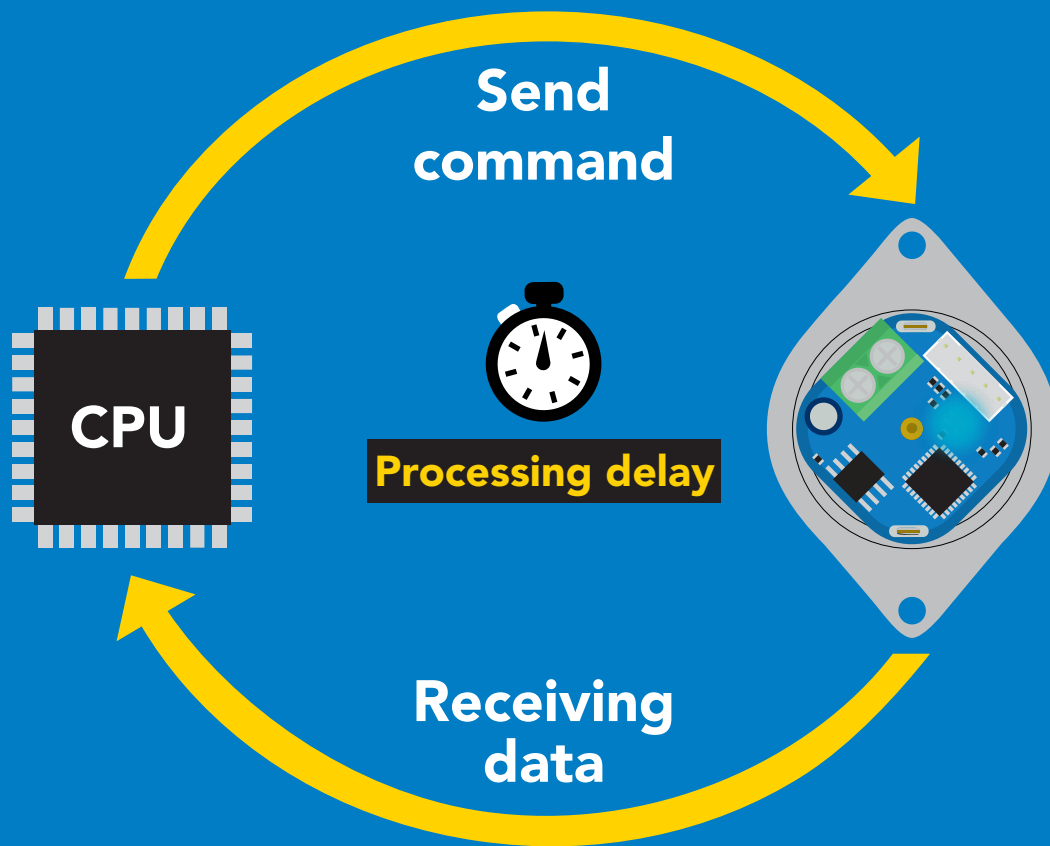


$$\begin{matrix} \mathbf{1} & \mathbf{52} & \mathbf{49} & \mathbf{51} & \mathbf{0} & = & \mathbf{413} \\ \text{Dec} & \text{ASCII} & & & \text{Dec} & & \end{matrix}$$

# 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

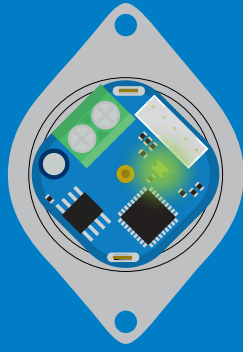
255	no data to send
254	still processing, not ready
2	error
1	successful request

# LED color definition



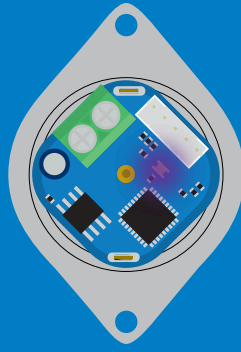
**Blue**

I<sup>2</sup>C standby



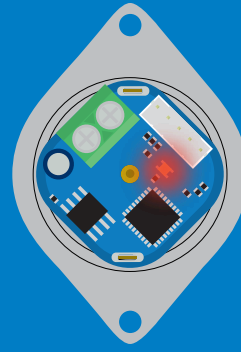
**Green**

Taking reading



**Purple**

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



**Red**

Command  
not understood



**White**

Find

**5V**

LED ON  
**+2.5 mA**

**3.3V**

**+1 mA**



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

## command quick reference

All commands are ASCII strings or single ASCII characters.

<b>Command</b>	<b>Function</b>	
<b>Baud</b>	switch back to UART mode	<b>pg. 68</b>
<b>Cal</b>	performs calibration	<b>pg. 59</b>
<b>D</b>	dispense modes	<b>pg. 52 – 55</b>
<b>Factory</b>	enable factory reset	<b>pg. 67</b>
<b>i</b>	device information	<b>pg. 62</b>
<b>I2C</b>	change I <sup>2</sup> C address	<b>pg. 66</b>
<b>L</b>	enable/disable LED	<b>pg. 50</b>
<b>O</b>	enable/disable parameters	<b>pg. 60</b>
<b>P</b>	enable/disable protocol lock	<b>pg. 56</b>
<b>Plock</b>	enable/disable protocol lock	<b>pg. 65</b>
<b>Pv</b>	check pump voltage	<b>pg. 61</b>
<b>R</b>	returns a single reading	<b>pg. 51</b>
<b>Sleep</b>	enter sleep mode/low power	<b>pg. 64</b>
<b>Status</b>	retrieve status information	<b>pg. 63</b>
<b>Tv</b>	total volume dispensed	<b>pg. 58</b>
<b>X</b>	stop dispensing	<b>pg. 57</b>

# LED control

## Command syntax

300ms  processing delay

- L,1 LED on **default**
- L,0 LED off
- L,? LED state on/off?

## Example

## Response

L,1

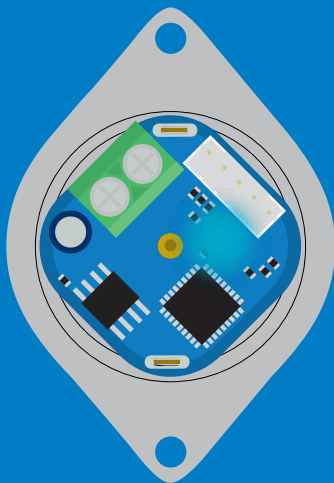
 **Wait 300ms**    **1**    **0**  
Dec    Null

L,0

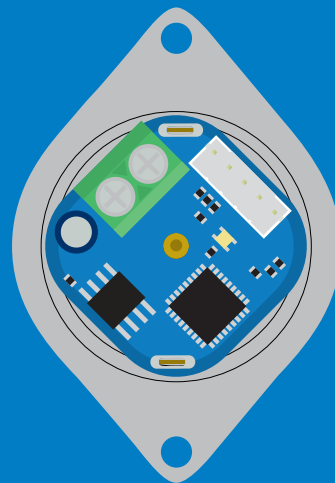
 **Wait 300ms**    **1**    **0**  
Dec    Null

L,?

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



L,1



L,0

# Single report mode

## Command syntax

300ms  processing delay

**R** returns a single value showing dispensed volume

## Example

## Response

**R**



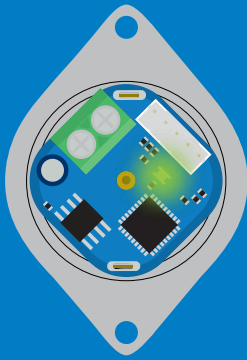
<b>1</b>	<b>2.50</b>	<b>0</b>
Dec	ASCII	Null

(If issued half way through dispensing 5ml)



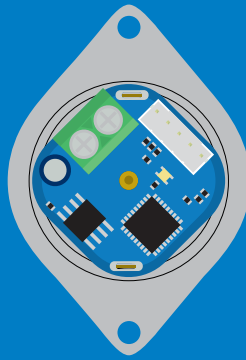
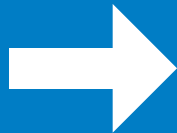
<b>1</b>	<b>5.00</b>	<b>0</b>
Dec	ASCII	Null

(If issued once dispensing has stopped)

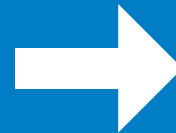


**Green**

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** **0**  
Dec Null

15 ml will be dispensed

D,-405



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

405 ml will be dispensed  
*in reverse*

D,?



**1** **?D,22.50,0** **0**  
Dec ASCII Null

## Response breakdown

?D,22.50,0

↑ last volume dispensed  
↑ pump off

# 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**

**Response**

  
**Wait 300ms**

**1**  
Dec

**0**  
Null

**Dispense 85ml over 10 mins**



# Constant flow rate

Maintain a constant flow rate

300ms  processing delay

## Command syntax

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

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 0  
Dec Null

Dispense 25ml per minute  
for 40 minutes

DC,?

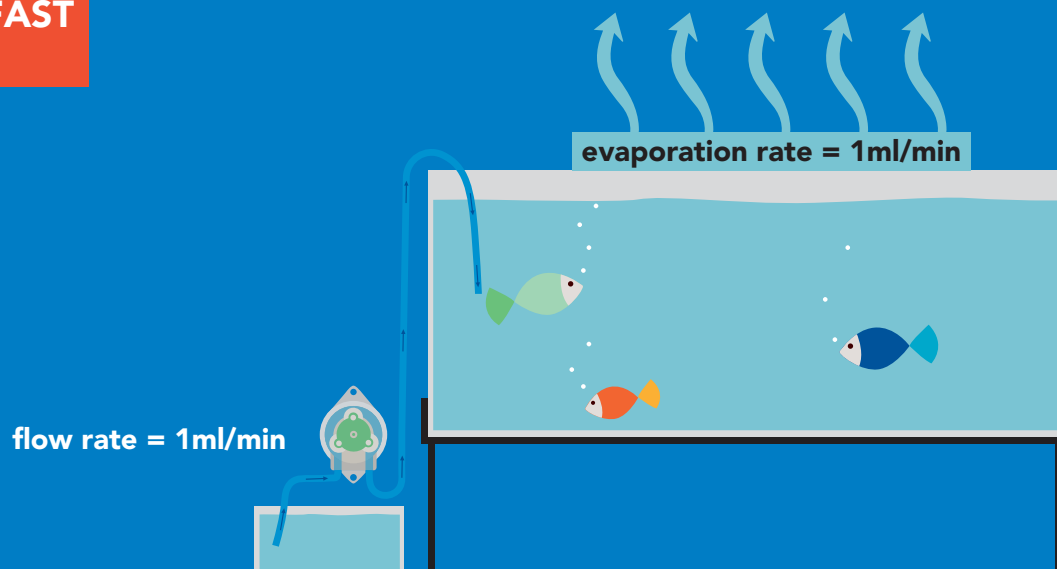


1 ?maxrate,58.5 0  
Dec ASCII Null

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

\*TOOFAST

\*ER



# Pause dispensing

300ms  processing delay

## Command syntax

Issue the command again to resume dispensing

**P** pauses the pump during dispensing

**P,?** pause status

## Example

## Response

**P**

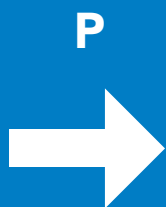
 **Wait 300ms** **1** **0**  
Dec Null

**P,?**

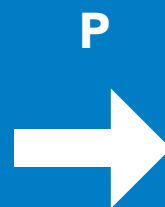
 **Wait 300ms** **1** **?P,1** **0** or  **Wait 300ms** **1** **?P,0** **0**  
Dec ASCII Null  
paused unpaused



dispensing



paused



dispensing



# Stop dispensing

## Command syntax

300ms  processing delay

X stop dispensing

## Example

X

## Response

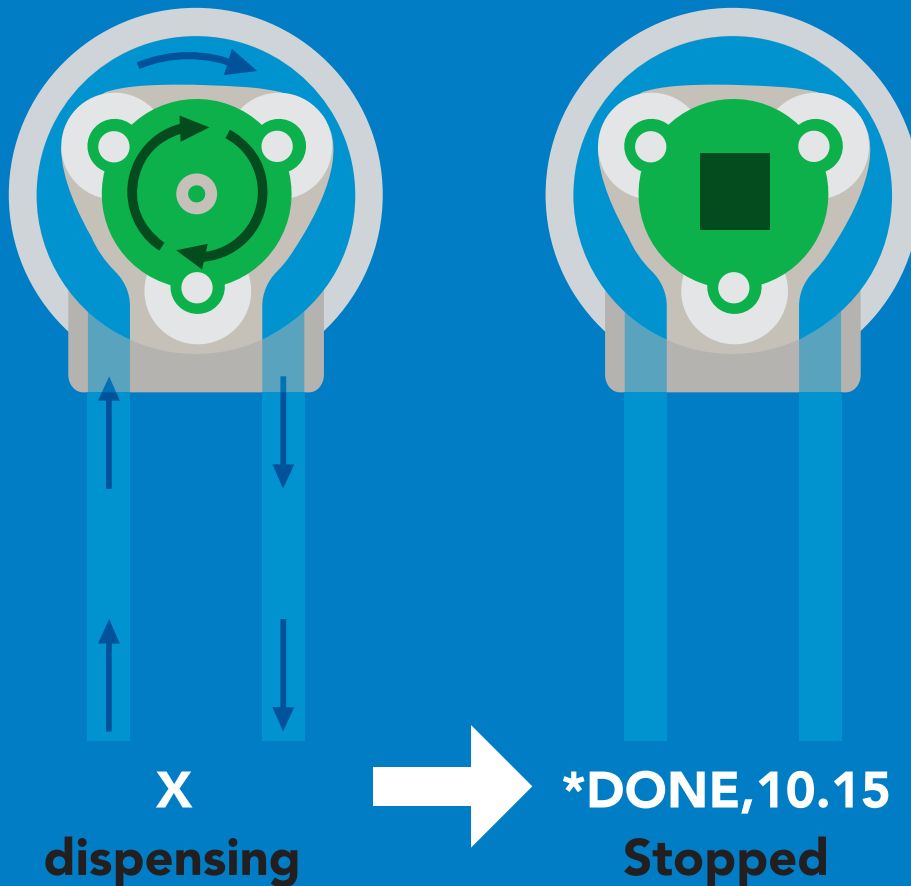
  
Wait 300ms

**1**  
Dec

**\*DONE,v**  
ASCII

**0**  
Null

v = volume dispensed



# 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

## Response

**TV,?**



Wait 300ms

1

Dec

?total,623.00

ASCII

0

Null

**ATV,?**



Wait 300ms

1

Dec

?total,434.50

ASCII

0

Null

**clear**



Wait 300ms

1

Dec

0

Null

**total now 0.00**

# Calibration

300ms  processing delay

## Command syntax

Calibrate to the actual volume dispensed.

Cal,v      v = corrected volume  
Cal,clear   delete calibration data  
Cal,?      device calibrated?

## Example

## Response



Cal,24.01



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

Cal,clear

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

Cal,?

 **Wait 300ms**      **1**      **?Cal,1**      **0**      or       **Wait 300ms**      **1**      **?Cal,2**      **0**  
Dec      ASCII      Null      Dec      ASCII      Null  
fixed volume      volume/time

 **Wait 300ms**      **1**      **?Cal,3**      **0**      or       **Wait 300ms**      **1**      **?Cal,0**      **0**  
Dec      ASCII      Null      Dec      ASCII      Null  
both      uncalibrated

# Enable/disable parameters from output string

## Command syntax

300ms  processing delay

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

## Example

## Response

O,V,1

 **Wait 300ms** **1** **0** enable volume being pumped  
Dec Null

O,TV,0

 **Wait 300ms** **1** **0** disable total volume pumped  
Dec Null

O,ATV,1

 **Wait 300ms** **1** **0** enable absolute volume pumped  
Dec Null

O,?

 **Wait 300ms** **1** **?O,V,TV,ATV** **0** if all three are enabled  
Dec ASCII Null

# Pump voltage

## Command syntax

300ms  processing delay

PV,? check pump voltage

## Example

## Response

PV,?



Wait 300ms

1

Dec

?PV,13.86

ASCII

0

Null

## Response breakdown

?PV, 13.86



Pump input voltage

# 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

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

## Response breakdown

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

### Restart codes

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

# Sleep mode/low power

## Command syntax

**Sleep** enter sleep mode/low power

Send any character or command to awaken device.

### Example

### Response

**Sleep**

**no response**

Do not read status byte after issuing sleep command.

**Any command**

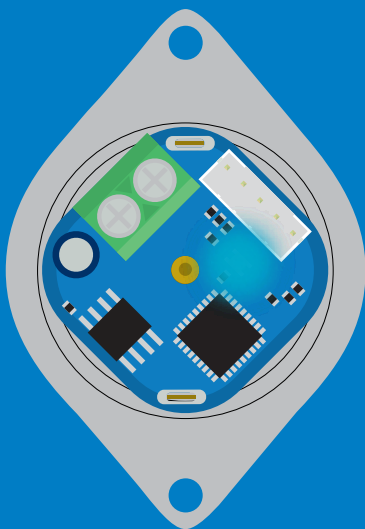
**wakes up device**

**5V**

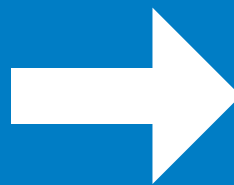
STANDBY	SLEEP
<b>13.4 mA</b>	<b>0.415 mA</b>

**3.3V**

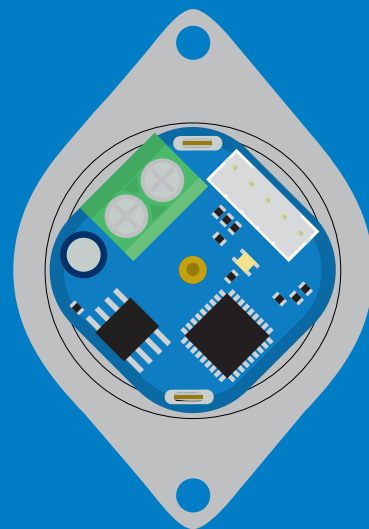
<b>12.4 mA</b>	<b>0.13 mA</b>
----------------	----------------



**Standby**



**Sleep**



**Sleep**



# Protocol lock

## Command syntax

300ms  processing delay

Plock,1 enable Plock

Plock,0 disable Plock

Plock,? Plock on/off?

**default**

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

## Example

## Response

Plock,1

  
Wait 300ms

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

Plock,0

  
Wait 300ms

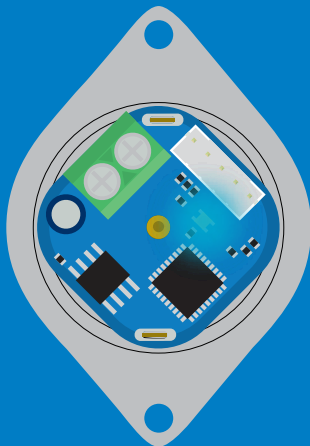
<b>1</b>	<b>0</b>
Dec	Null

Plock,?

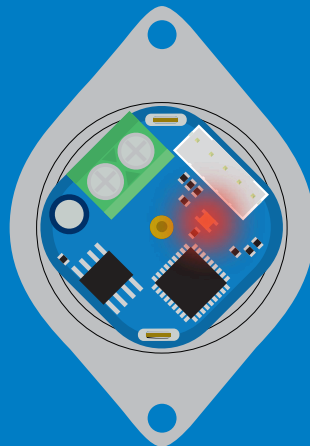
  
Wait 300ms

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

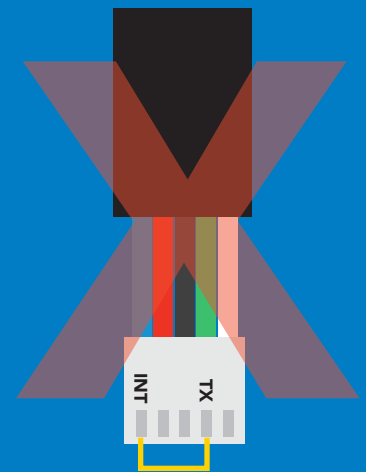
Plock,1



Serial, 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

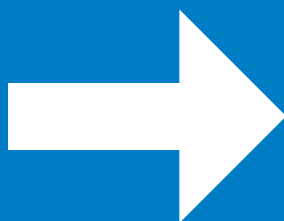
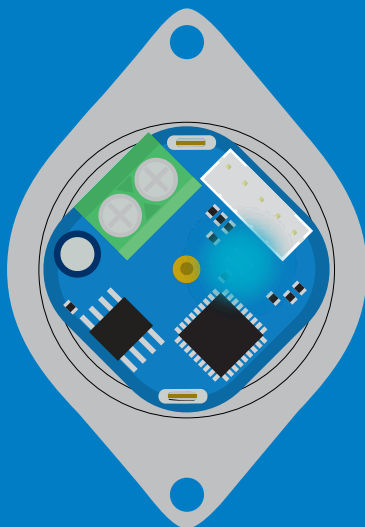
## Warning!

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.

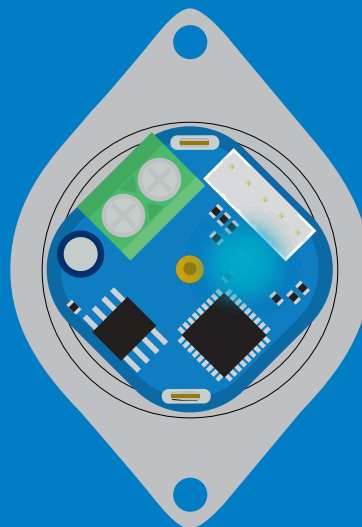
Default I<sup>2</sup>C address is 103 (0x67).

**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

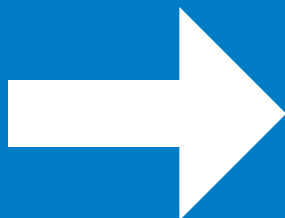
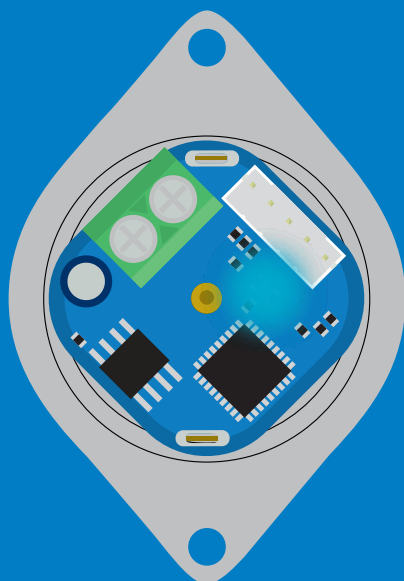
## Response

Factory

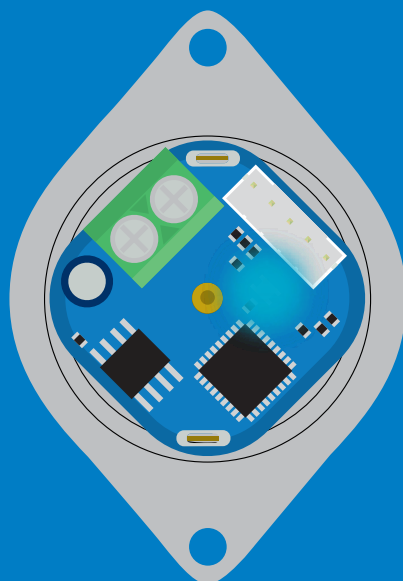
device reboot

Clears calibration  
LED on  
Response codes enabled

Factory



(reboot)



# Change to UART mode

## Command syntax

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

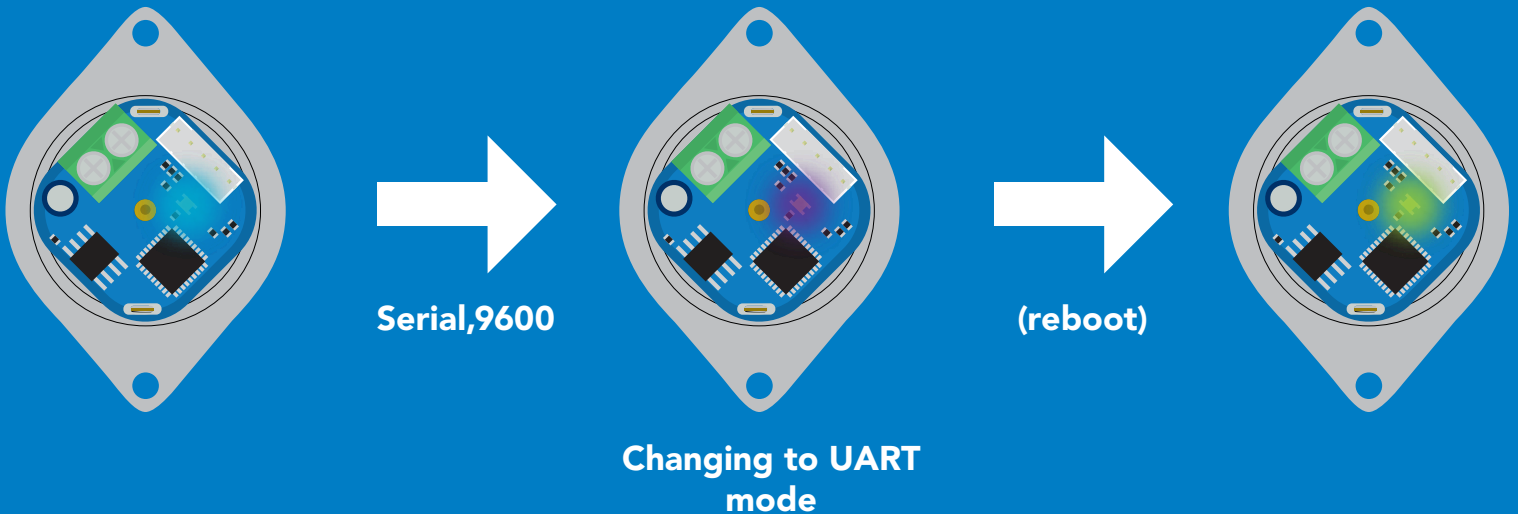
### Example

Baud,9600

### Response

reboot in UART mode

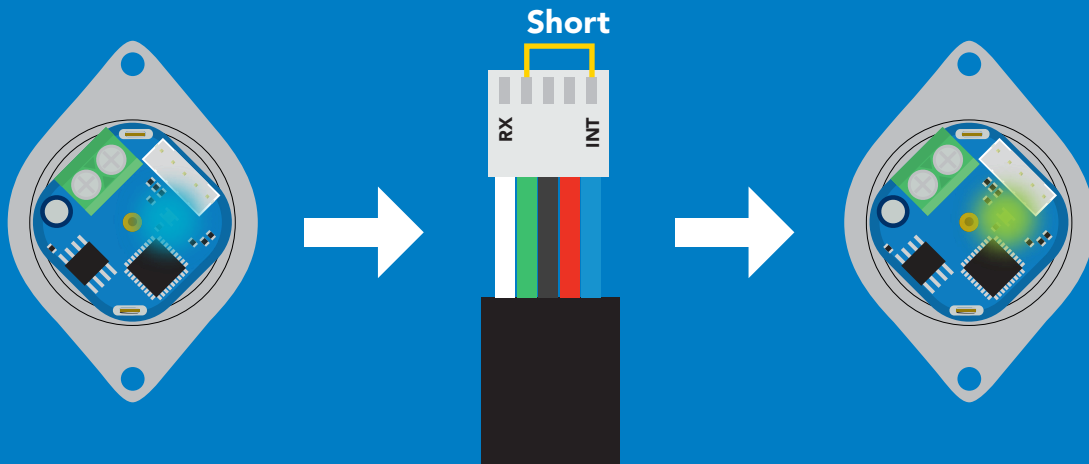
n = [ 300  
1200  
2400  
9600  
19200  
38400  
57600  
115200



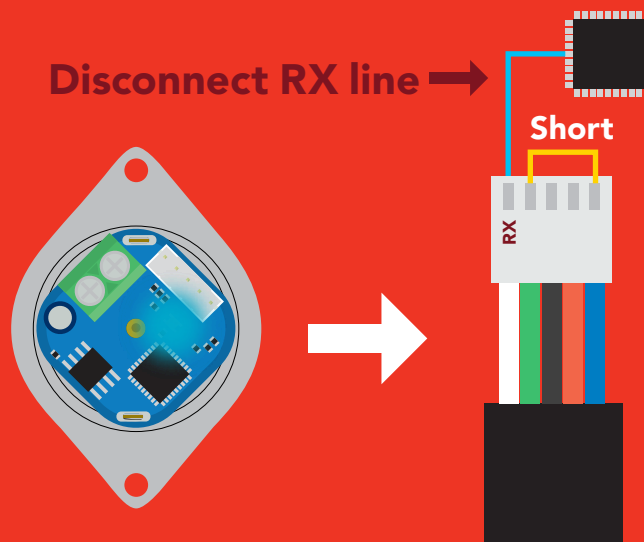
# Manual switching to UART

- Make sure Plock is set to 0
- 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

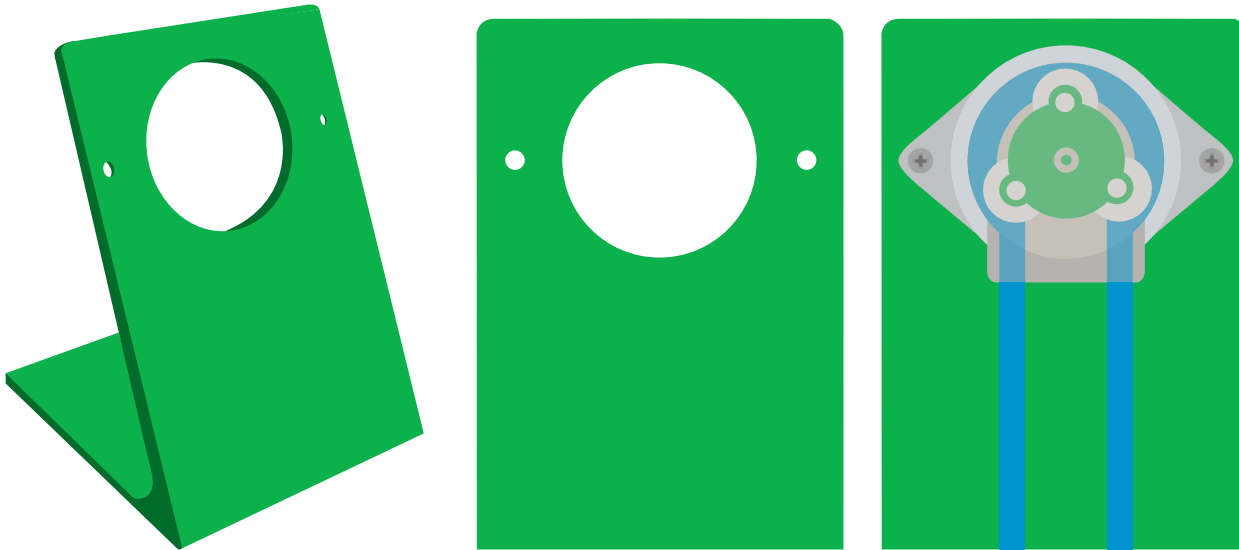


## Wrong Example

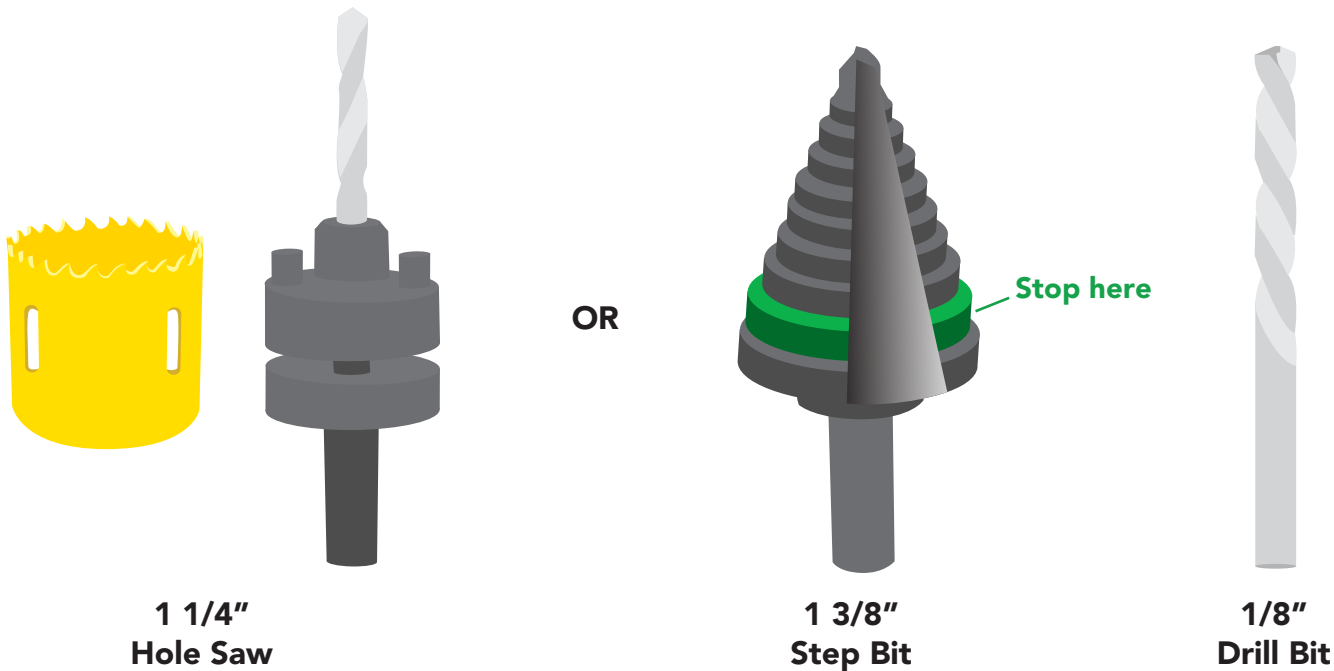


# Mounting the EZO-PMP™

There are a 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:



Either are fine to make the larger hole.

Perfect for screw holes.

# Datasheet change log

## 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.

## EZO-PMP™ firmware changes

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

# 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 (whichever 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.