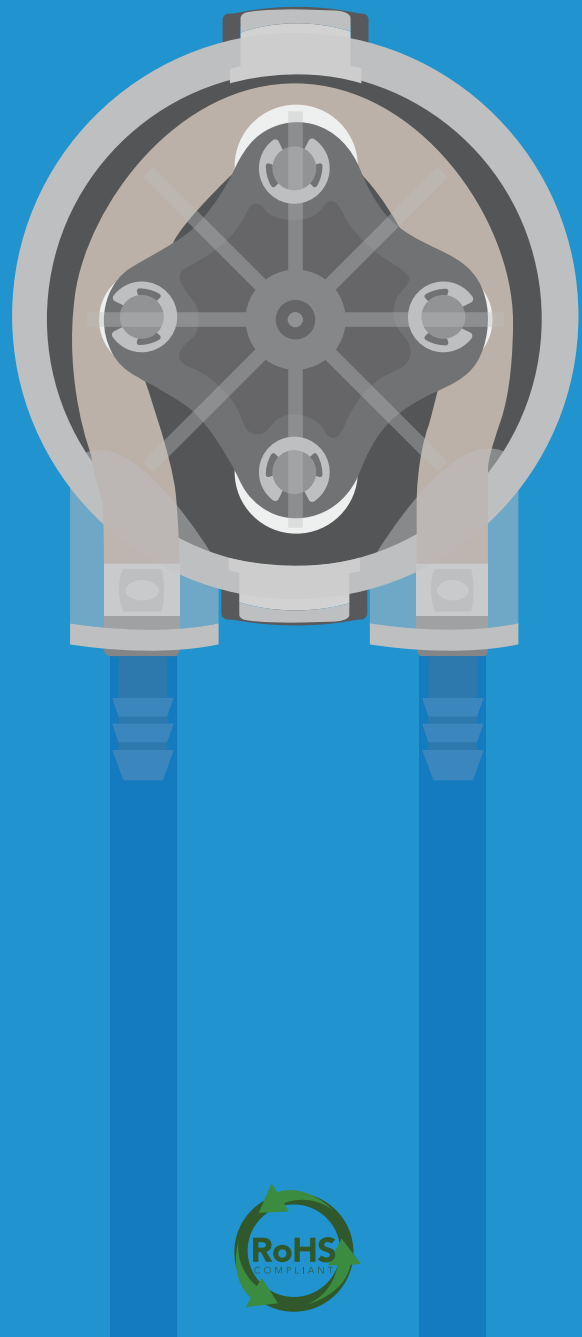


# EZO-PMP-L<sup>TM</sup>

## Large Embedded Dosing Pump

Flow rate	<b>10ml to 750ml/min</b>
Accuracy	<b>+/- 2%, +/-2ml</b>
Modes of operation	<b>Continuous dispensing Volume dispensing Dose over time Constant flow rate Dispense at startup</b>
Connector	<b>5 lead data cable</b>
Calibration	<b>Single point</b>
Tubing size	<b>Any 8mm O.D. tubing</b>
Data protocol	<b>UART &amp; I<sup>2</sup>C</b>
Default I <sup>2</sup> C address	<b>109 (0x6D)</b>
Operating voltage	<b>3.3V–5V (logic) 24V (motor)</b>
Pump head	<b>10.3 meters (34')</b>
Data format	<b>ASCII</b>
Food safe	<b>Yes</b>



# Table of contents

EZO-PMP-L™ dimensions	4	Power supplies	8
EZO-PMP-L™ tubing	5	Dispensing volumes	9
Operating principle	6	Available data protocols	10
Operating modes	6	Default state	11

## UART

UART mode	13
Receiving data from device	14
Sending commands to device	15
LED color definition	16
<b>UART quick command page</b>	<b>17</b>
LED control	18
Find	19
Continuous mode	20
Single reading mode	21
Continuous dispensing	22
Volume dispensing	23
Dose over time	24
Constant flow rate	25
Dispense at startup	26
Pause dispensing	29
Stop dispensing	30
Invert dispensing direction	31
Total volume dispensed	32
Calibration	33
Enable/disable parameters	34
Pump voltage	35
Naming device	36
Device information	37
Response codes	38
Reading device status	39
Sleep mode/low power	40
Change baud rate	41
Protocol lock	42
Factory reset	43
Change to I <sup>2</sup> C mode	44
Manual switching to I <sup>2</sup> C	45

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

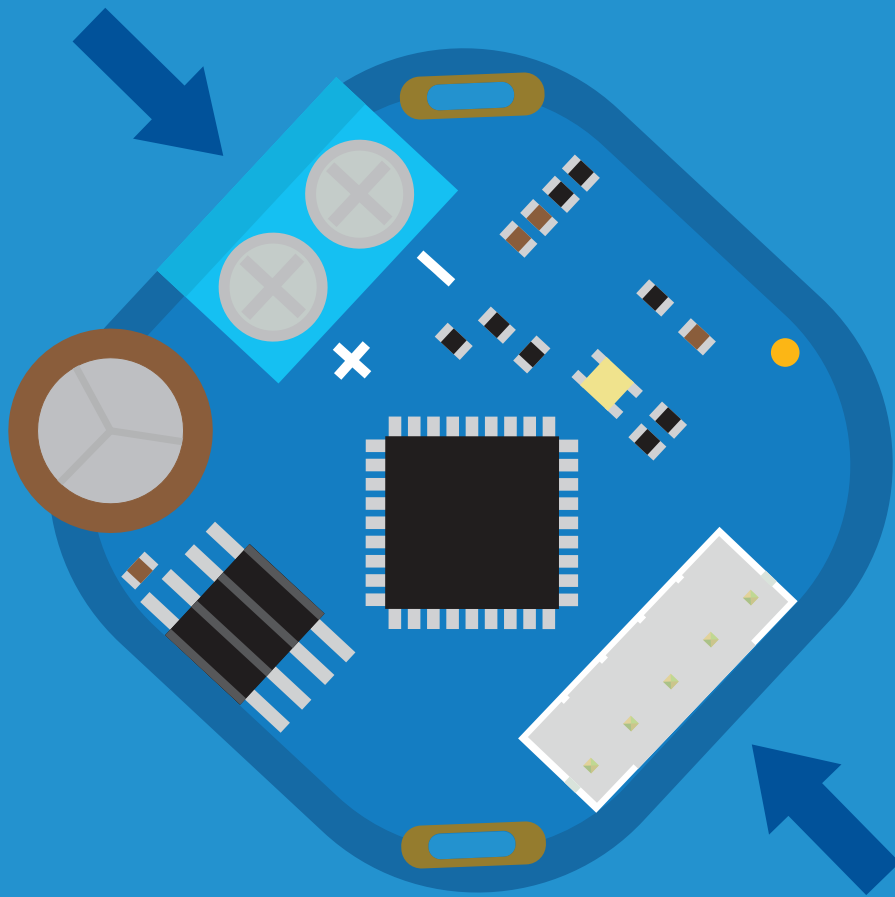
I <sup>2</sup> C mode	47
Sending commands	48
Requesting data	49
Response codes	50
LED color definition	51
<b>I<sup>2</sup>C quick command page</b>	<b>52</b>
LED control	53
Find	54
Single report mode	55
Continuous dispensing	56
Volume dispensing	57
Dose over time	58
Constant flow rate	59
Dispense at startup	60
Pause dispensing	63
Stop dispensing	64
Invert dispensing direction	65
Total volume dispensed	66
Calibration	67
Enable/disable parameters	68
Pump voltage	69
Naming device	70
Device information	71
Reading device status	72
Sleep mode/low power	73
Protocol lock	74
I <sup>2</sup> C address change	75
Factory reset	76
Change to UART mode	77
Manual switching to UART	78

Calibration theory	79
Mounting plate	82
Datasheet change log	83
Warranty	84

# Attention

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

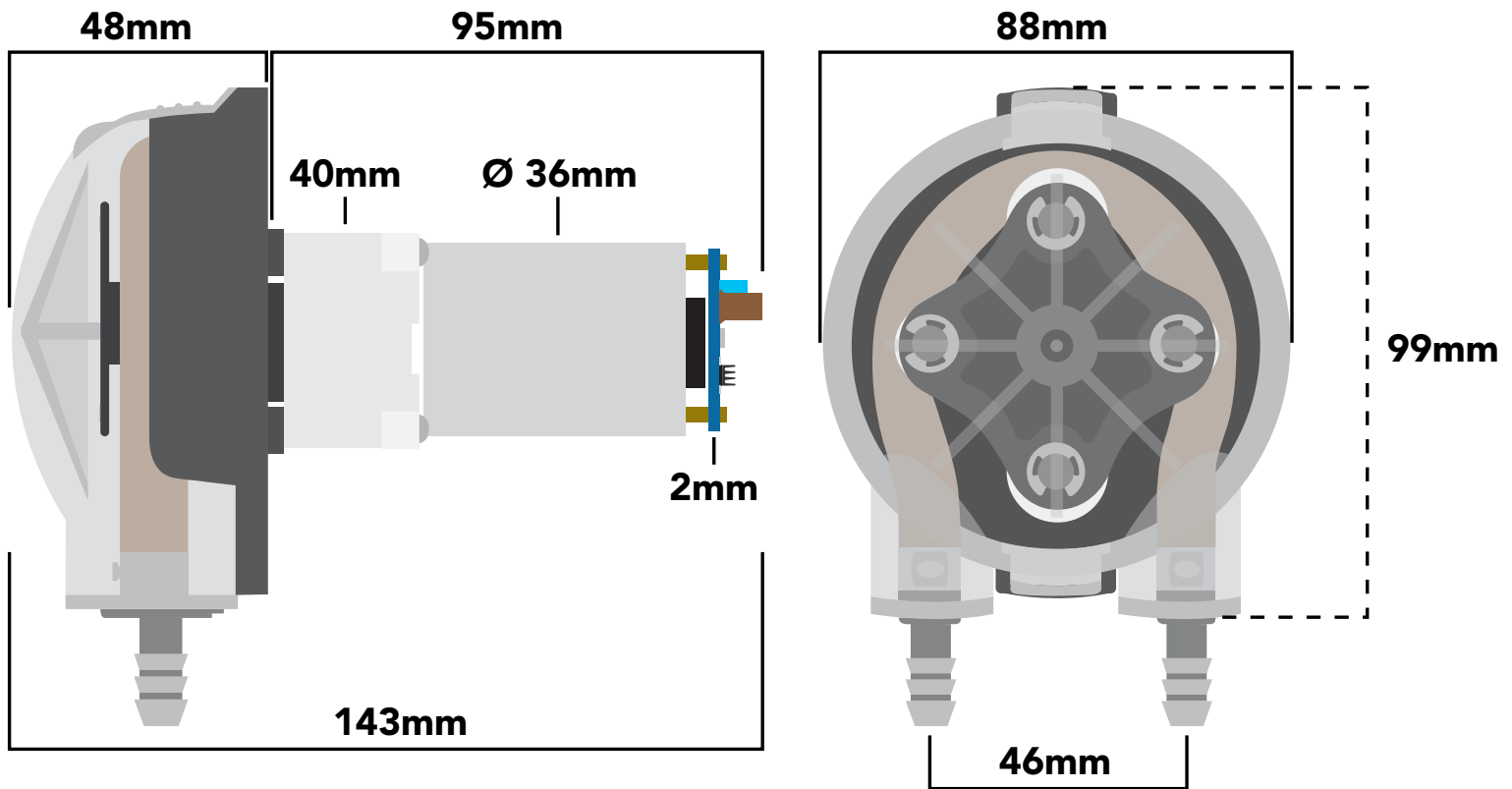
**24V**  
to drive the motor



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

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

# EZO-PMP-L™ dimensions



**Weight 405g**

## 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>	24V = ~700mA			

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-L™)	-65 °C		125 °C
Operational temperature (EZO-PMP-L™)	-40 °C	25 °C	85 °C
VCC	3.3V	5V	5.5V
Motor	17V	24V	26V
Max input / output pressure	101.3 kPa		

# EZO-PMP-L™ tubing

## NSF/ANSI 51 Compliant

### Tan tubing

Saint-Gobain™ PharMed™ BPT tubing

Length: 15.24cm

Outer diameter: 10mm

Inner diameter: 8mm

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

Food safe ✓

### Hose barb fitting

ETFE

Length: 2.1cm

Outer diameter: 10mm

Inner diameter: 4.4mm

Food safe ✓

### Blue tubing

Silicone

Length: 2x 30.48cm

Outer diameter: 8mm

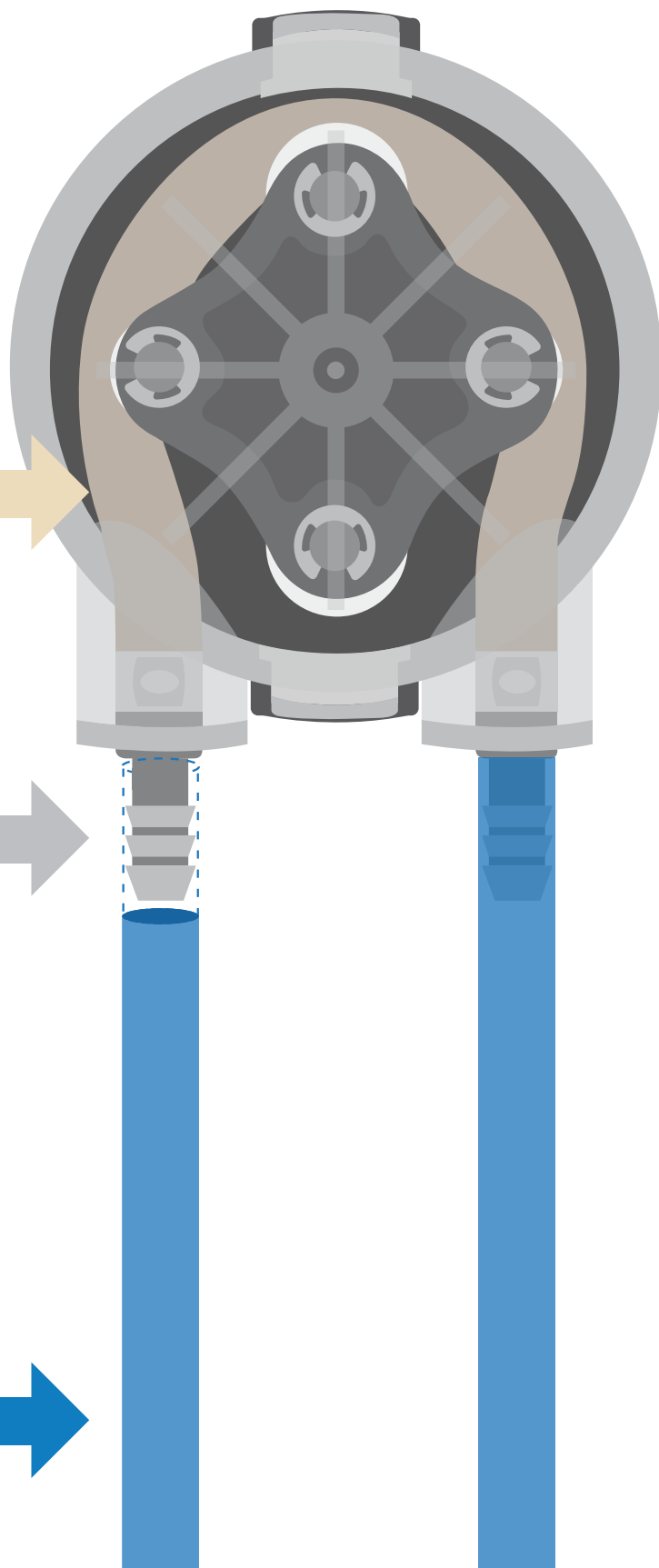
Inner diameter: 6mm

Bend radius: 24mm

Temperature -67°C to 200°C

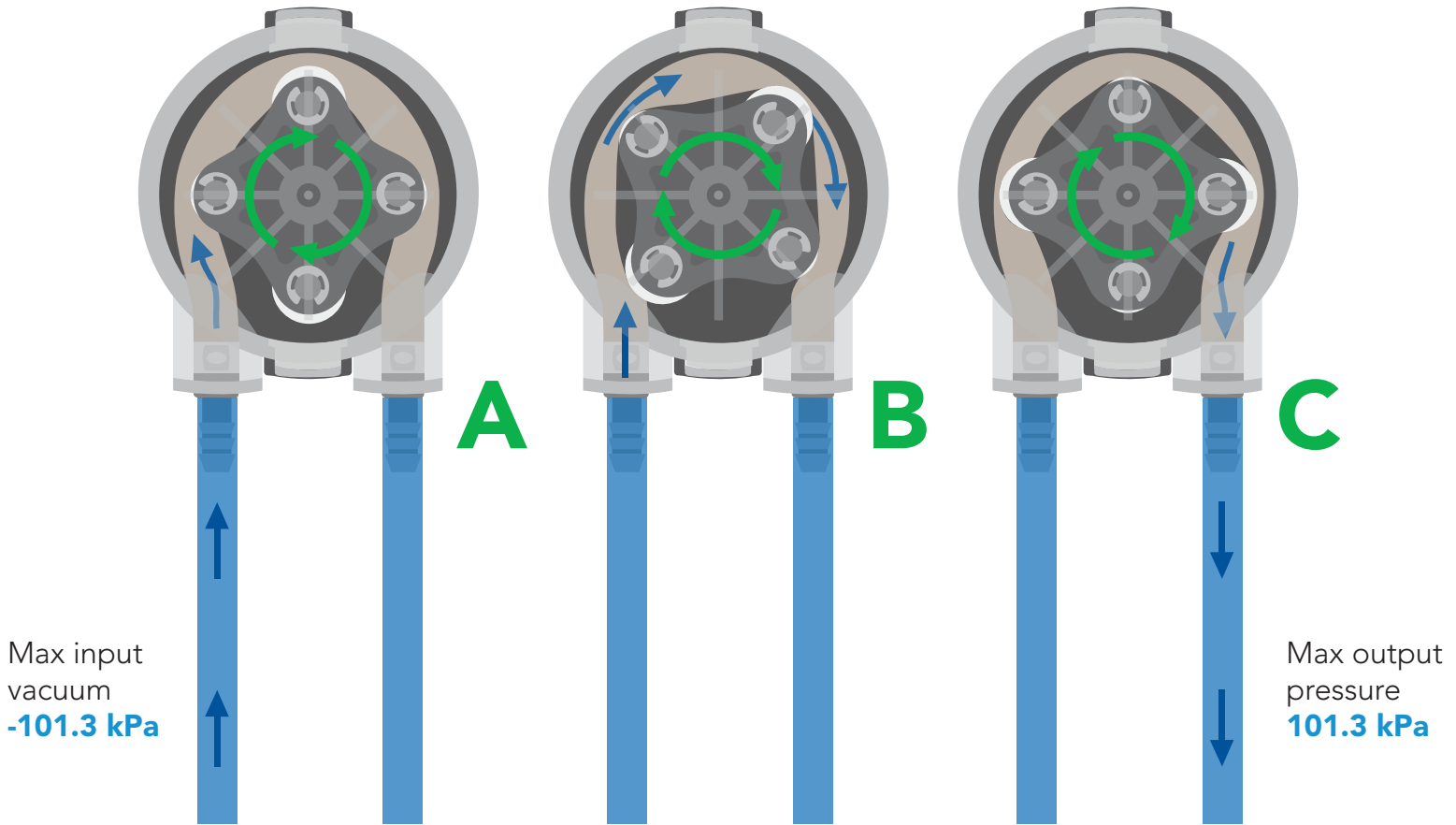
Max pressure: 34 kPa (5 PSI)

Food safe ✓



# Operating principle

- ✓ Self-priming
- ✓ Run dry



# Operating modes

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

## Continuous dispensing

Run the pump continuously  
750ml/min ∞ (with supplied tubing)

## Volume dispensing

Pump a specific volume  
(Smallest possible volume is 10 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

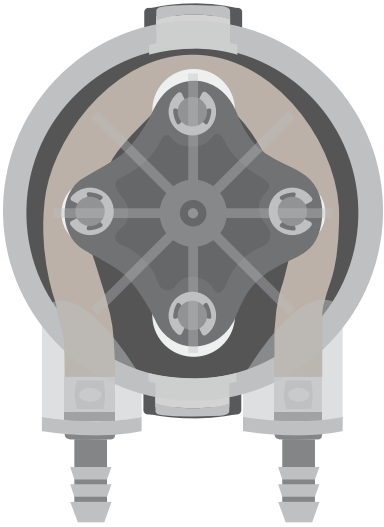
- Dispense a specific volume at startup
- Continuous dispensing at startup
- Dose over time at startup

**This device requires two power supplies**

3.3V–5.5V for the control system

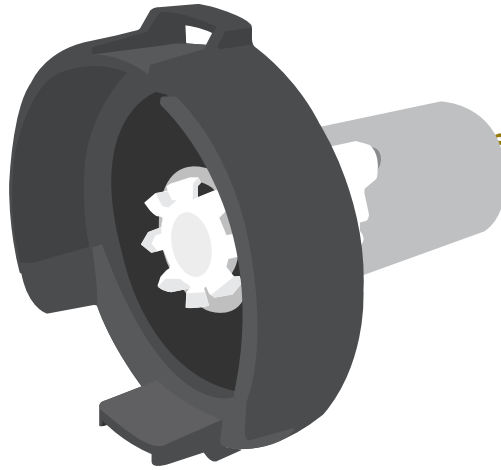
24V to drive the motor

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

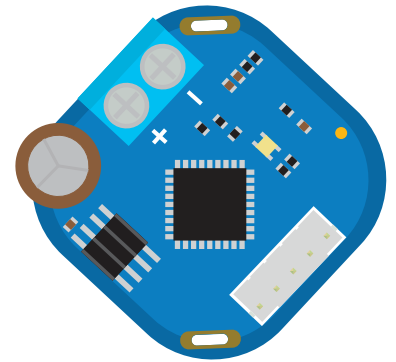


## Cassette

*Cassettes can be autoclave sterilized*



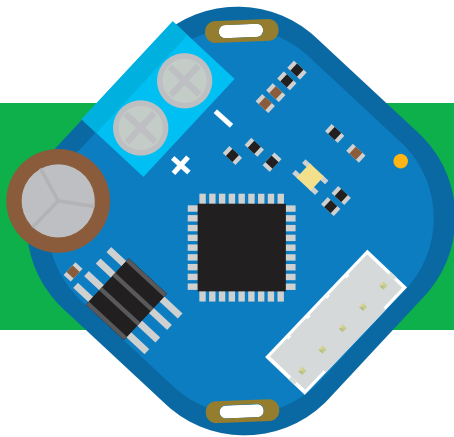
## 24 volt motor



## 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 24 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  
24 volt power input  
Status indicator LED

**Data and power cable pinout**

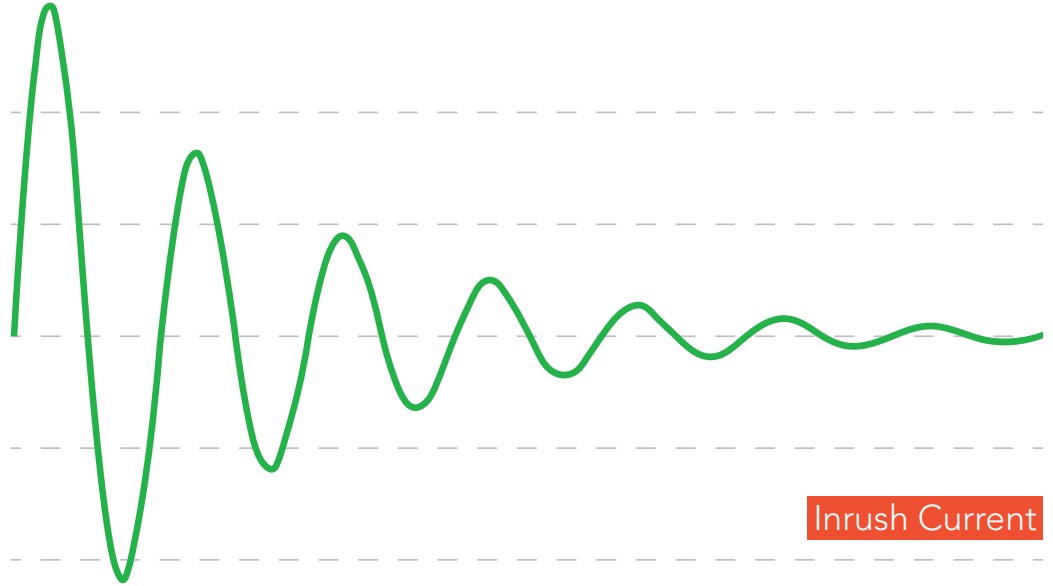
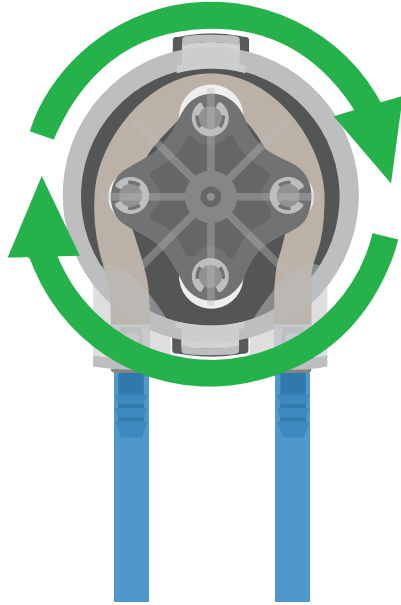
White	–	RX/SCL
Green	–	TX/SDA
Black	–	GND
Red	–	VCC
Blue	–	INT



# Power supplies



The nature of this inductive motor can cause a high inrush current upon its first spin. Therefore, not all power supplies will work with the EZO-PMP-L™.



Most power supplies have a built in protection against inrush currents and will prevent the motor from spinning. It has become very difficult to determine which power supplies will and won't work for the EZO-PMP-L™, as the manufacturers do not list whether it can handle inrush currents in the specs.

Atlas Scientific has tested both of these two power supplies, and can verify that they work with the EZO-PMP-L™.

## Mean Well #GE30I24-P1J



Included with the EZO-PMP-L™ Kit

## Mean Well #EDR-120-24

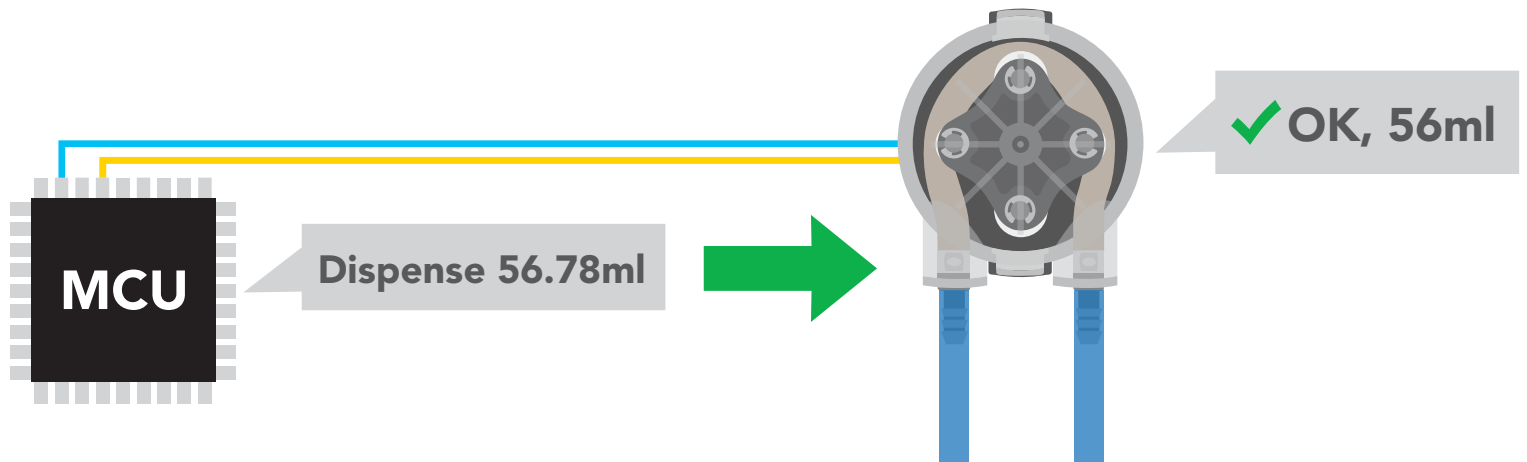
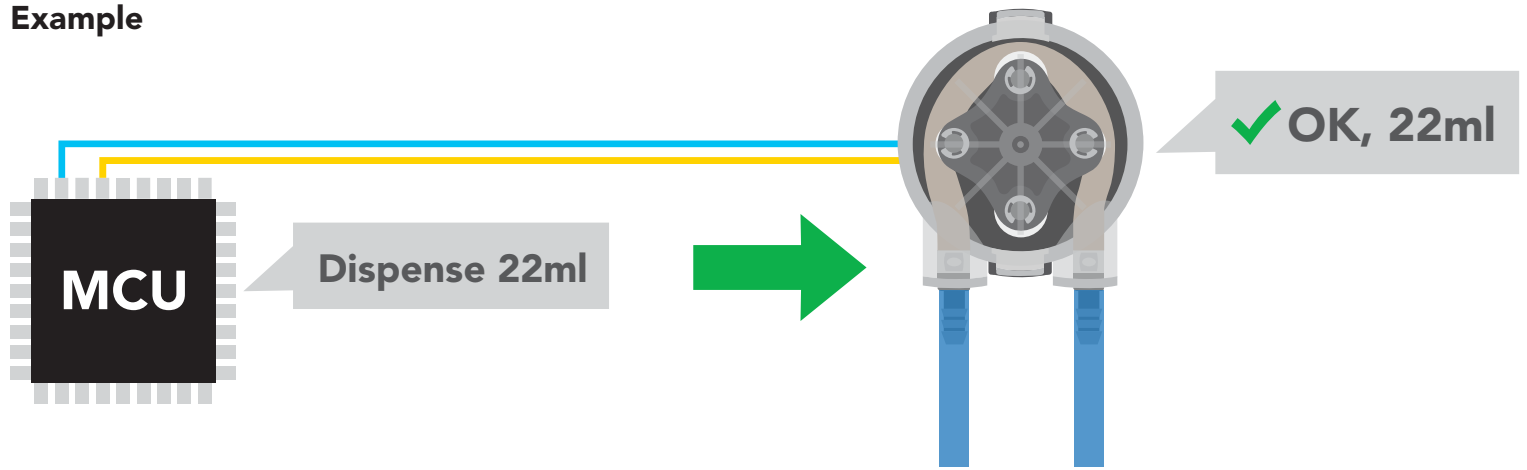




# Dispensing volumes

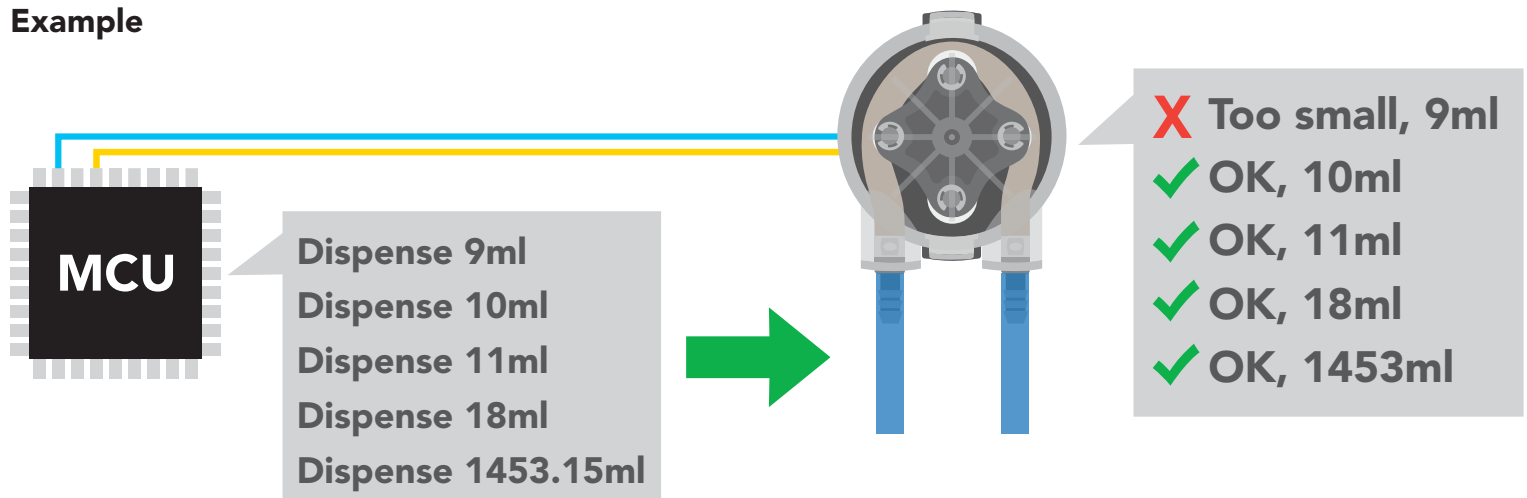
The EZO-PMP-L™ can only dispense volumes in whole numbers. The pump will not reject a request to dispense a volume with a decimal place, it will just ignore the decimal.

## Example



The minimum volume the EZO-PMP-L™ can dispense is 10 ml.

## Example



✓ Available data protocols

# UART

Default

# I<sup>2</sup>C

✗ Unavailable data protocols

SPI

Analog

RS-485

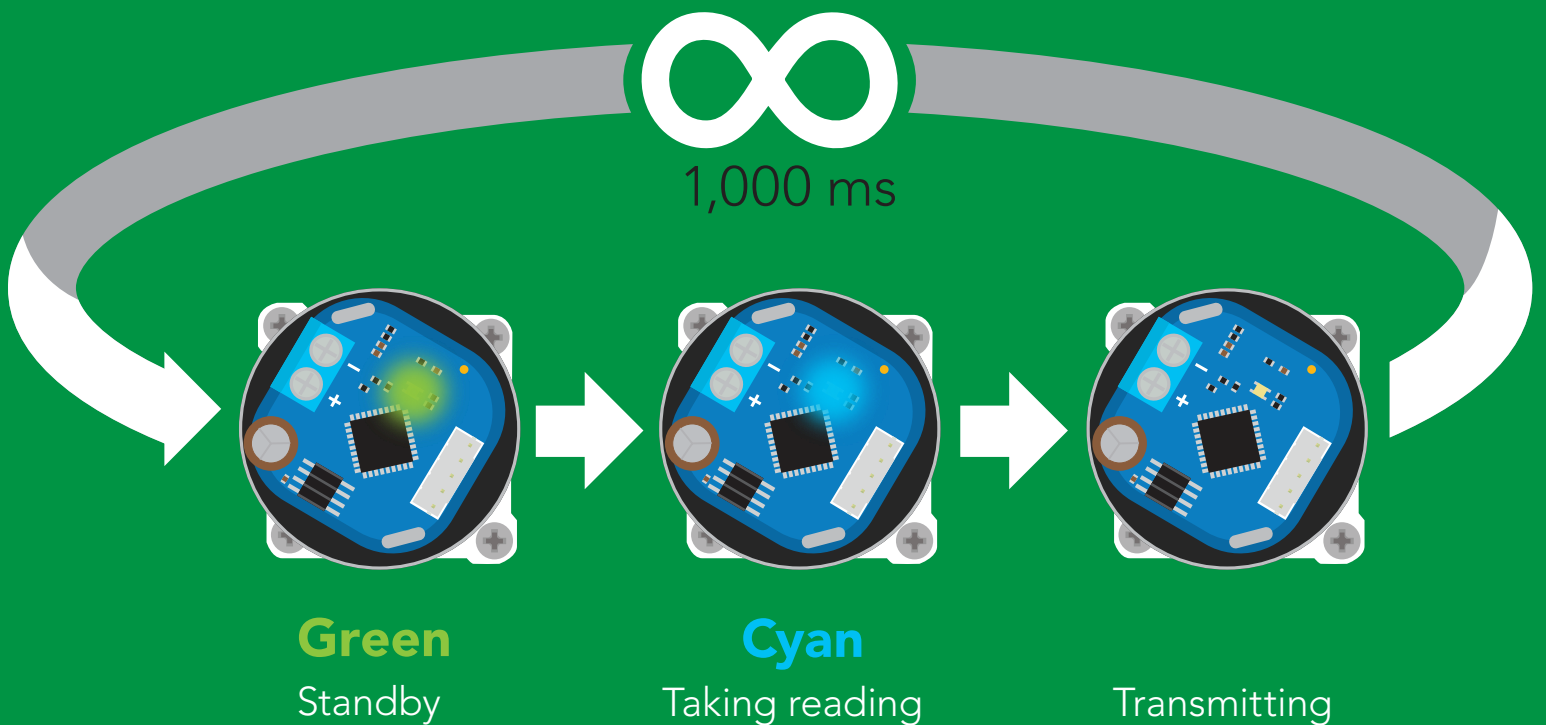
Mod Bus

4–20mA

# Default state

# UART mode

<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>



# 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

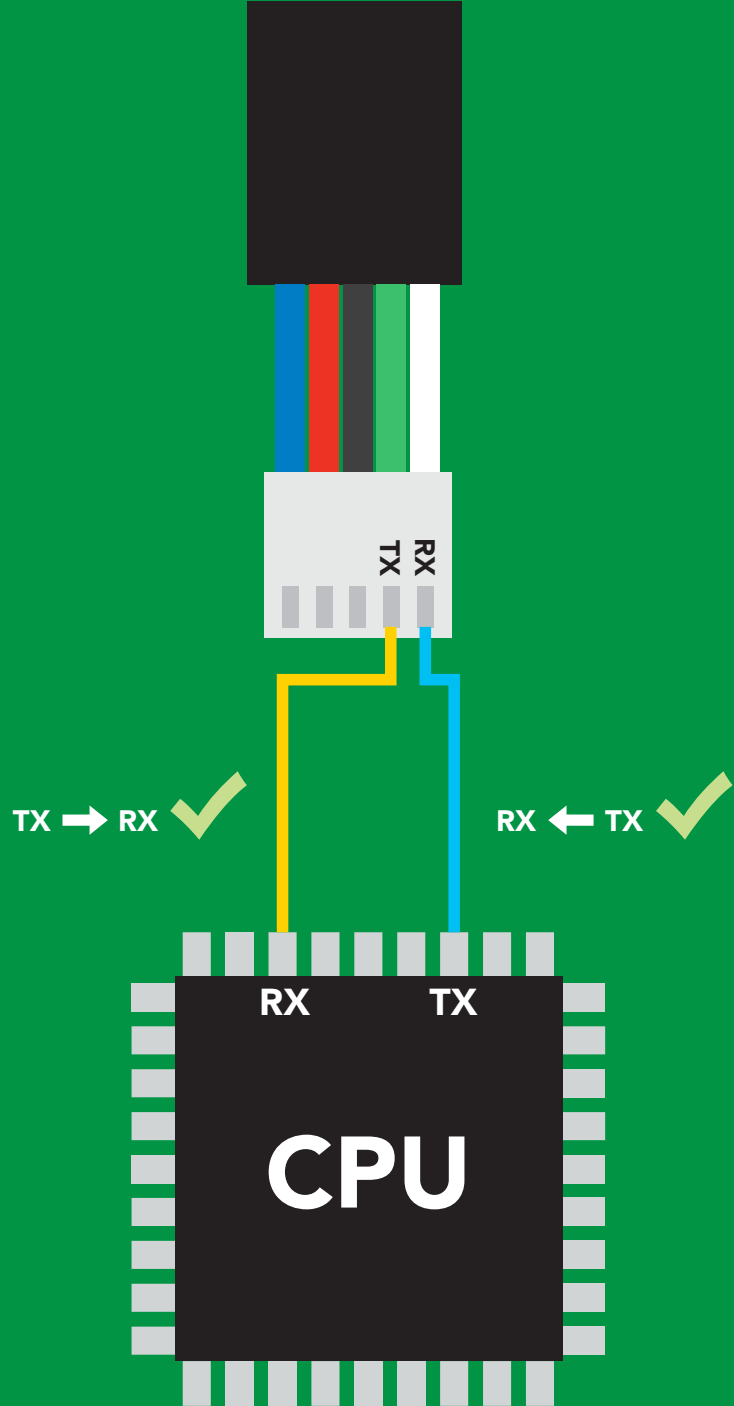
8 data bits  
1 stop bit

no parity  
no flow control

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



**Vcc** 3.3V – 5.5V



## Data format

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

**Terminator** carriage return  
**Data type** long int  
**Decimal places** none  
**Smallest string** 3 characters  
**Largest string** 39 characters

# Receiving data from device

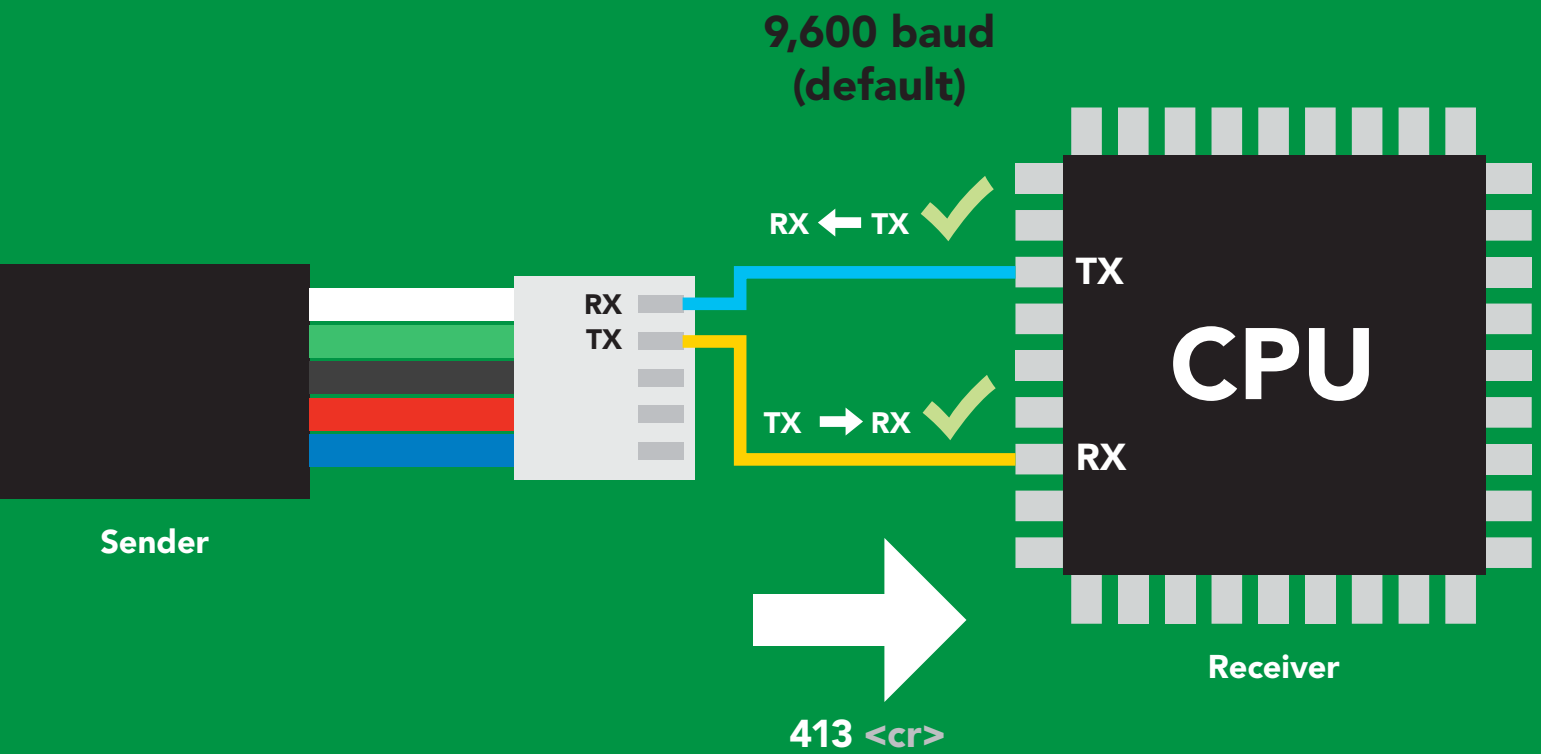
2 parts

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

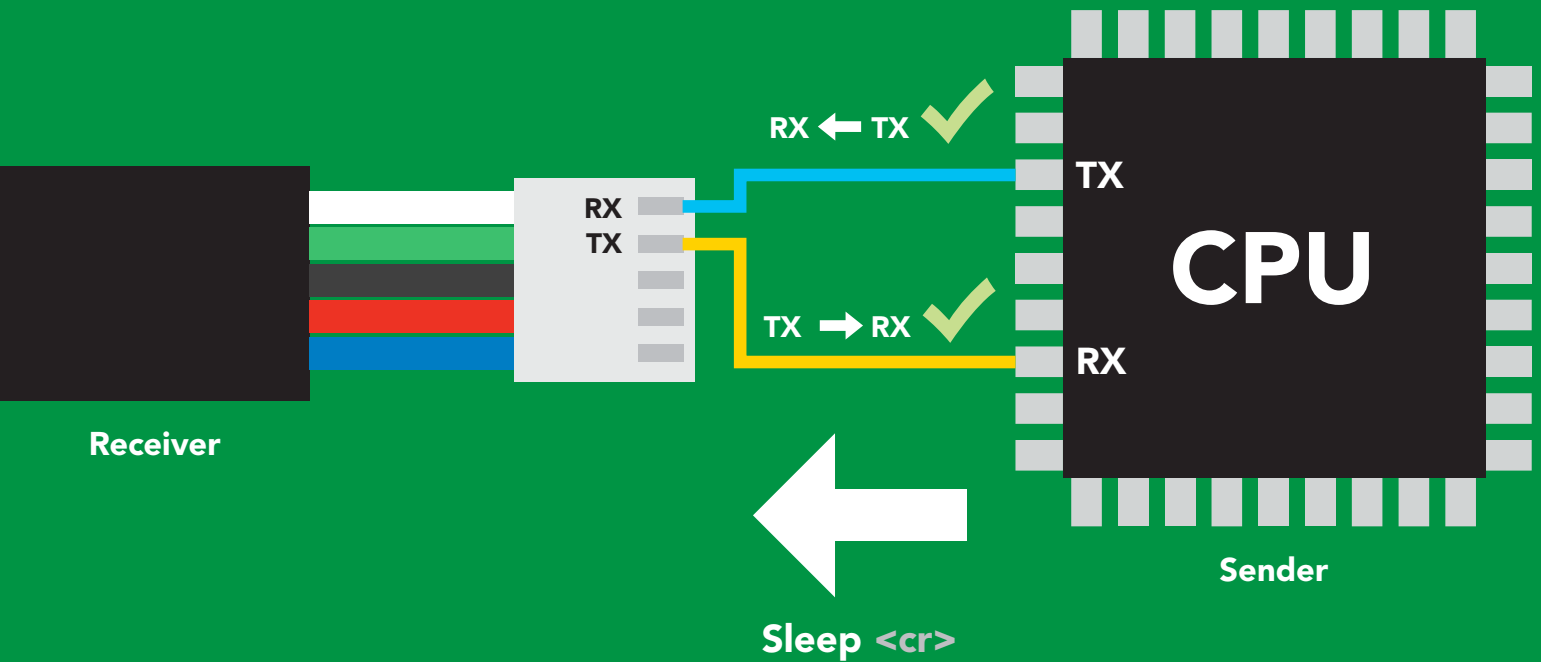
2 parts

**Command (not case sensitive)**

ASCII data string

**Carriage return <cr>**

Terminator



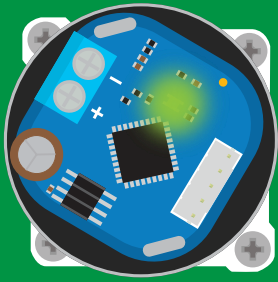
## Advanced

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

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

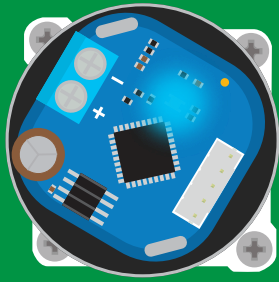
Dec: **83** **108** **101** **101** **112** **13**

# LED color definition



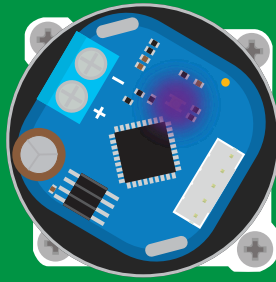
**Green**

UART standby



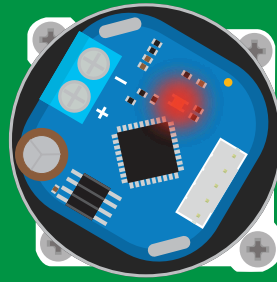
**Cyan**

Taking reading



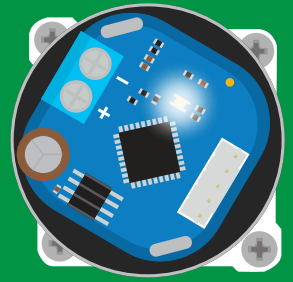
**Purple**

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. 41	9,600
C	enable/disable continuous mode	pg. 20	enabled
Cal	performs calibration	pg. 33	n/a
D	dispense modes	pg. 22 – 28	n/a
Factory	enable factory reset	pg. 43	n/a
Find	finds device with blinking white LED	pg. 19	n/a
i	device information	pg. 37	n/a
Invert	invert dispensing direction	pg. 31	n/a
I2C	change to I <sup>2</sup> C mode	pg. 44	not set
L	enable/disable LED	pg. 18	enabled
Name	set/show name of device	pg. 36	not set
O	enable/disable parameters	pg. 34	all enabled
P	pause dispensing	pg. 29	n/a
Plock	enable/disable protocol lock	pg. 42	disabled
Pv	check pump voltage	pg. 35	n/a
R	returns a single reading	pg. 21	n/a
Sleep	enter sleep mode/low power	pg. 40	n/a
Status	retrieve status information	pg. 39	enable
Tv	total volume dispensed	pg. 32	n/a
X	stop dispensing	pg. 30	n/a
*OK	enable/disable response codes	pg. 38	enable

# LED control

## Command syntax

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

L,0 <cr> LED off

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

## Example

## Response

L,1 <cr>

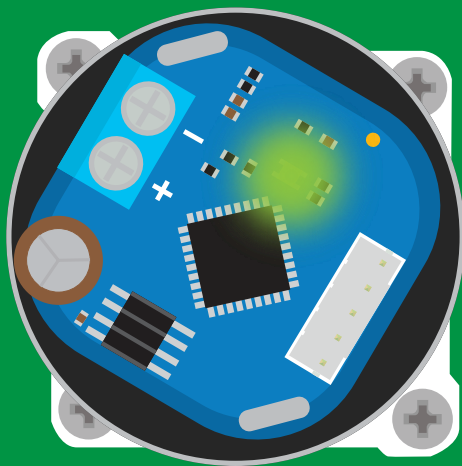
\*OK <cr>

L,0 <cr>

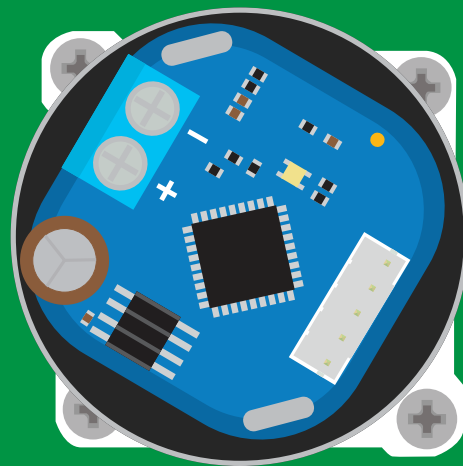
\*OK <cr>

L,? <cr>

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



L,1



L,0

# Find

## Command syntax

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

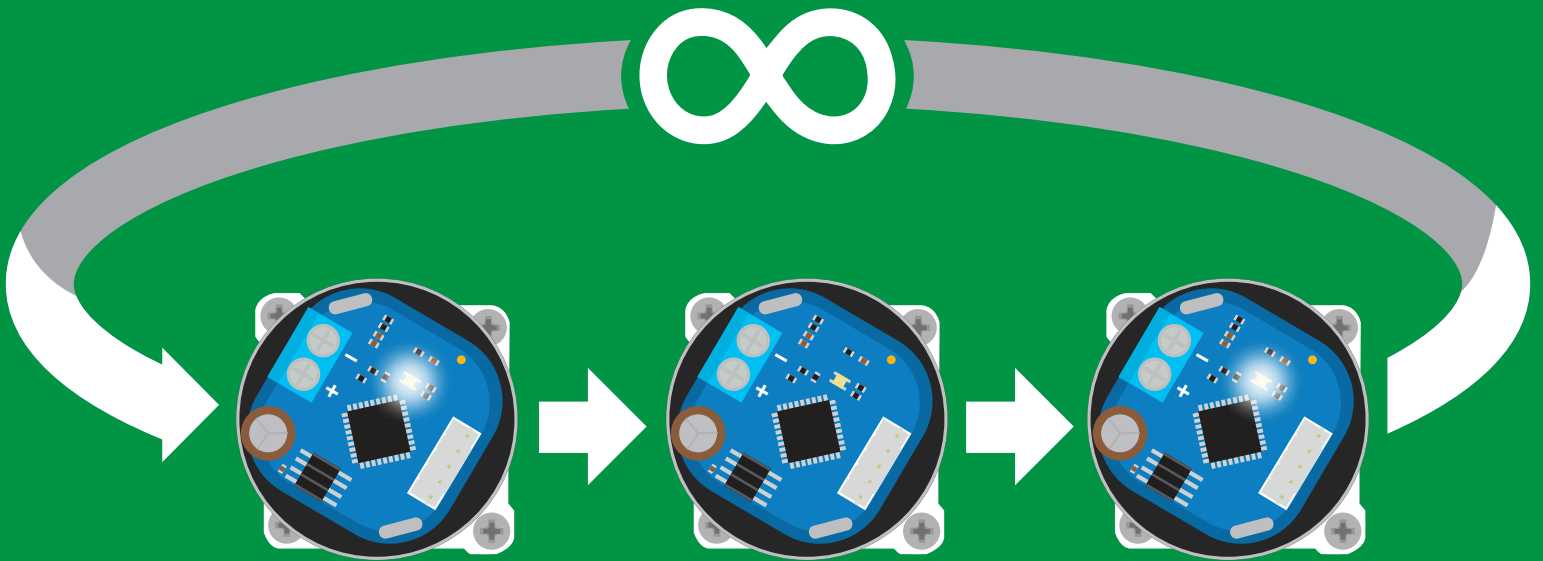
Find <cr> LED rapidly blinks white, used to help find device

## Example

## Response

Find <cr>

\*OK <cr>



# Continuous mode

## Command syntax

- C,\* <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 23ml

**C,\* <cr>**

**18 <cr>**  
**23 <cr>**  
**\*Done,23 <cr>**  
**23 <cr>**  
**23 <cr>**

**C,1 <cr>**

**18 <cr>**  
**23 <cr>**  
**\*Done,23 <cr>**

**C,0 <cr>**

**\*Done,23 <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>**

**25 <cr>** (If issued half way through dispensing 50ml)  
**\*OK <cr>**

**50 <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-L™ 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 ~750ml/min (with supplied tubing)

**D,-\*** <cr>

**\*OK** <cr> pump will continuously run in reverse at ~750ml/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 <cr>

\*OK <cr> 40 ml will be dispensed *in reverse*

D,? <cr>

?D,-40,0 <cr>

\*OK <cr>

## Response breakdown

?D,-40,0

↑ last volume dispensed  
↑ pump off

# Dose over time

Pump a fixed volume over a fixed time

## Command syntax

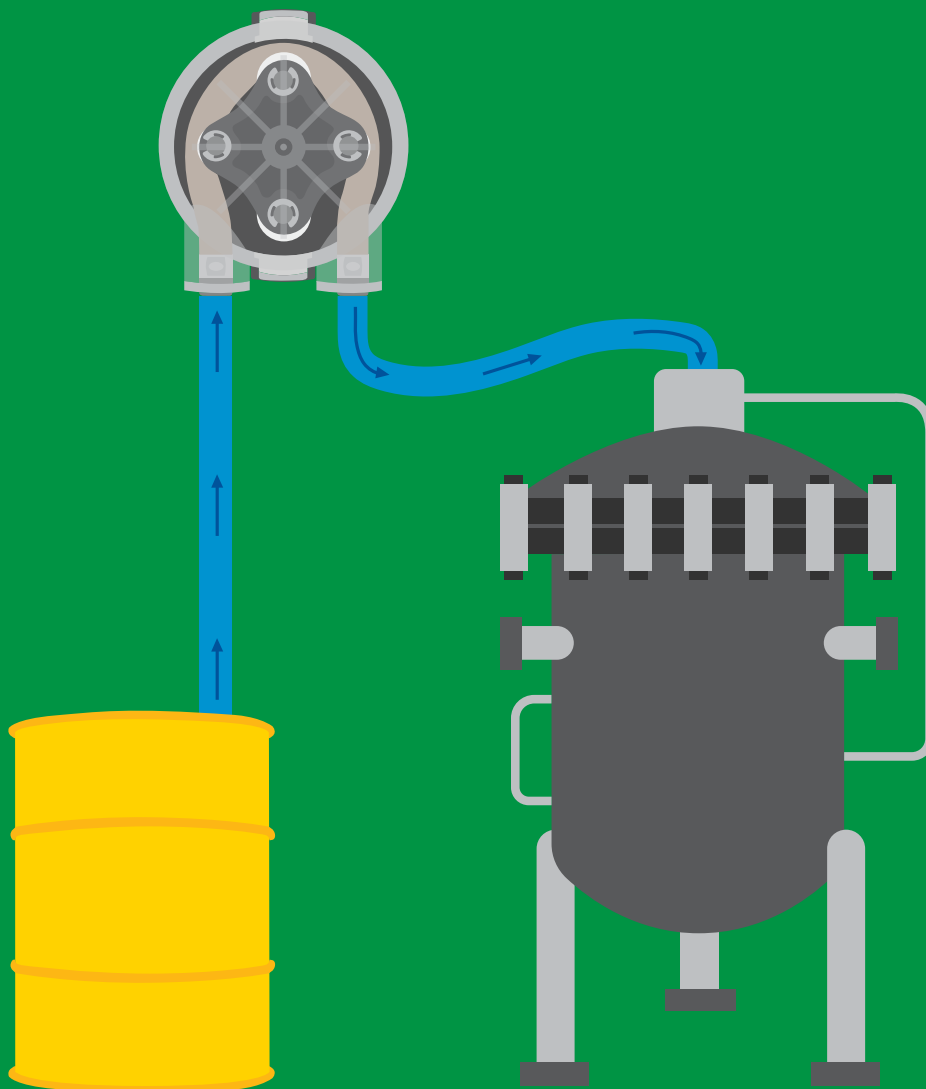
**D,[ml],[min] <cr> Dispense [this volume], [over this many minutes]**

### Example

**D,7000,20 <cr>**

### Response

**\*OK <cr> Dispense 7000ml over 20 minutes**





# Constant flow rate

Maintain a constant flow rate

## Command syntax

After running in continuous mode for 20 days the EZO-PMP-L™ 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,50,40 <cr>**

**\*OK <cr> Dispense 50ml per minute for 40 minutes**

**DC,? <cr>**

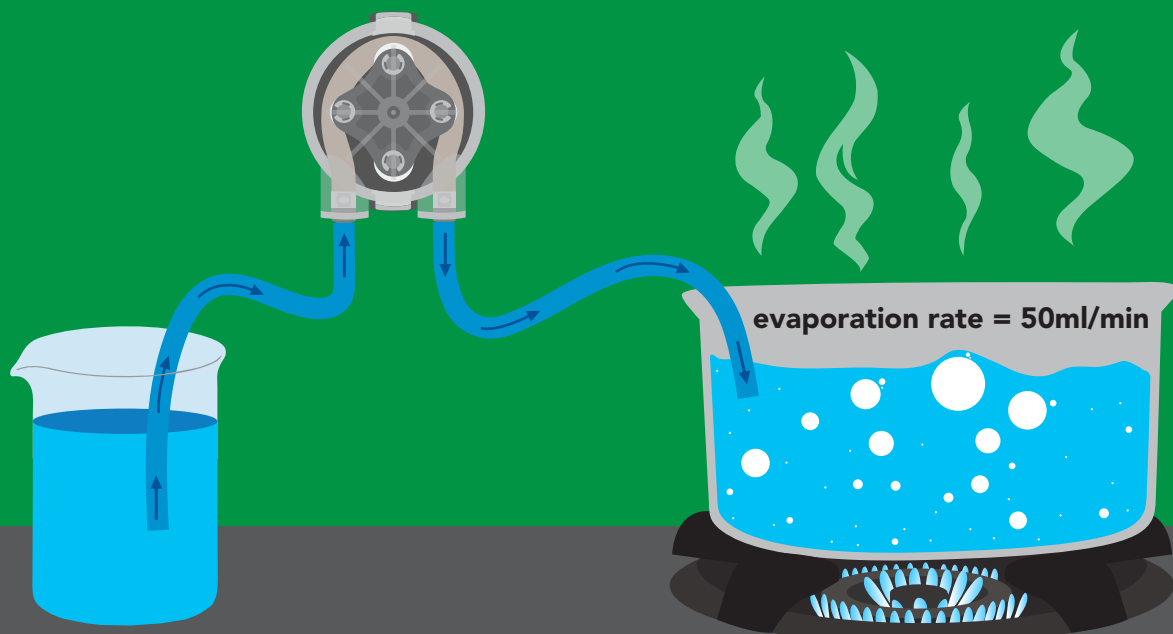
**?MAXRATE,385 <cr>**

**\*OK <cr>**

The maximum flow rate is determined after calibration.  
If the flowrate entered is too fast the EZO-PMP-L™ 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

## Response

**Dstart,100** <cr>

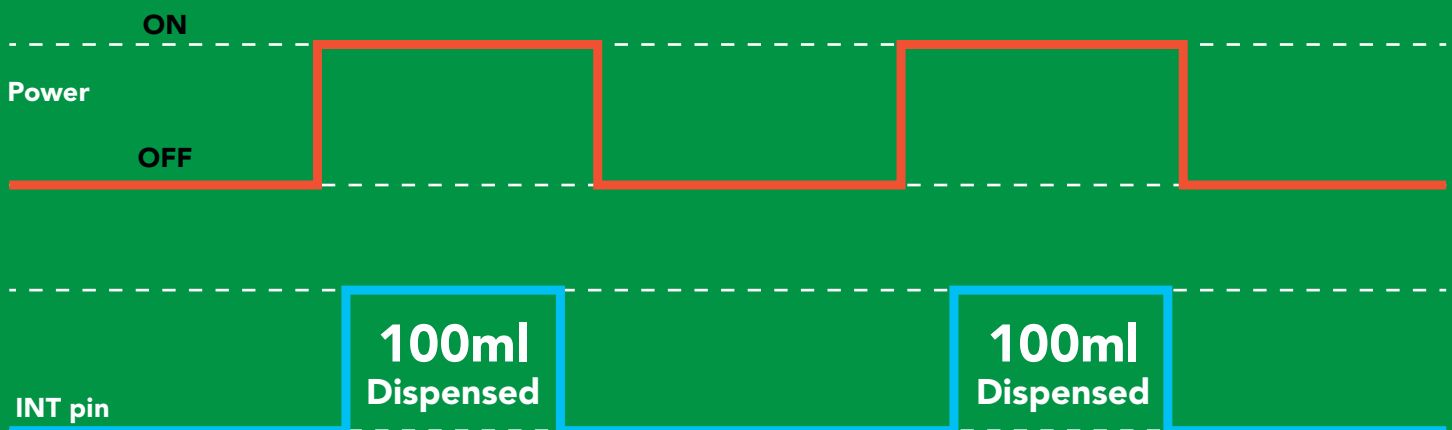
**\*OK** <cr>

**Dstart,off** <cr>

**\*OK** <cr>

**Dstart,?** <cr>

**?Dstart,100** <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-L™ 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 ~750ml/min (with supplied tubing)

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

**\*OK** <cr>

Pump will startup and continuously run in reverse at ~750ml/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

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

### Example

**Dstart,7000,20 <cr>**

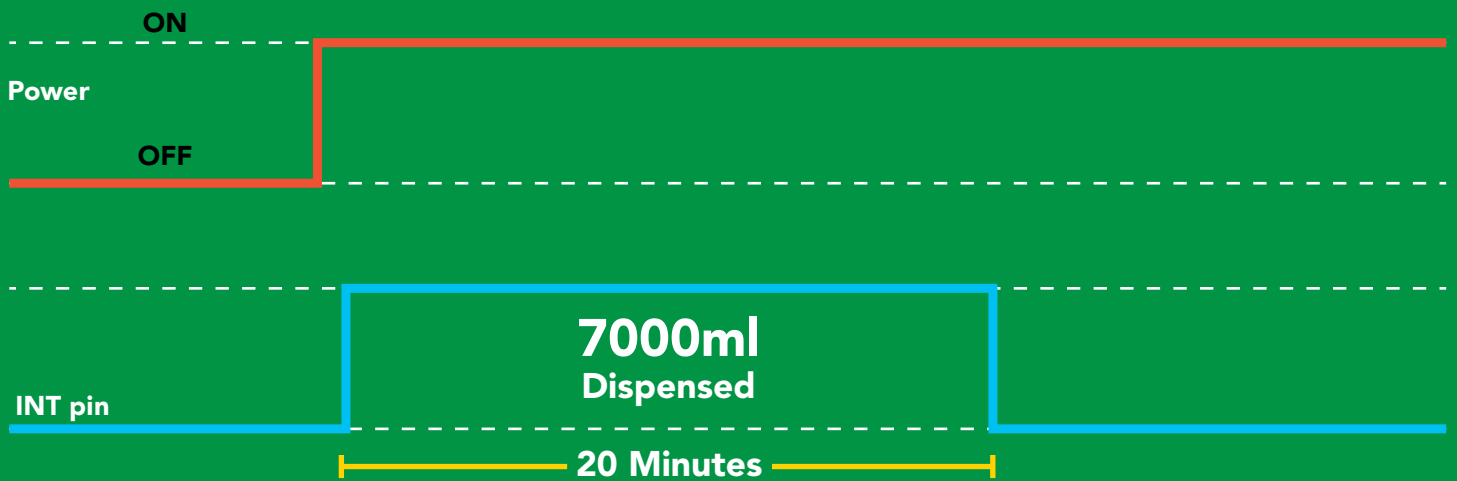
### Response

**\*OK <cr>**

Pump will startup and dispense 7000ml over 20 minutes

**Dstart,? <cr>**

**?Dstart,7000,20.00 <cr>**



# Pause dispensing

## Command syntax

Issue the command again to resume dispensing

**P** <cr> pauses the pump during dispensing  
**P,?** <cr> pause status

## Example

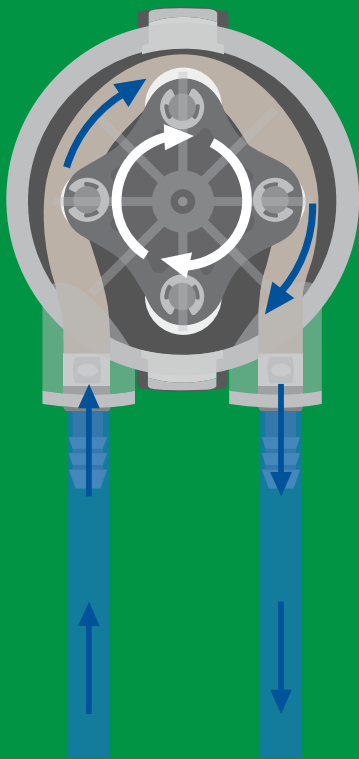
**P** <cr>

## Response

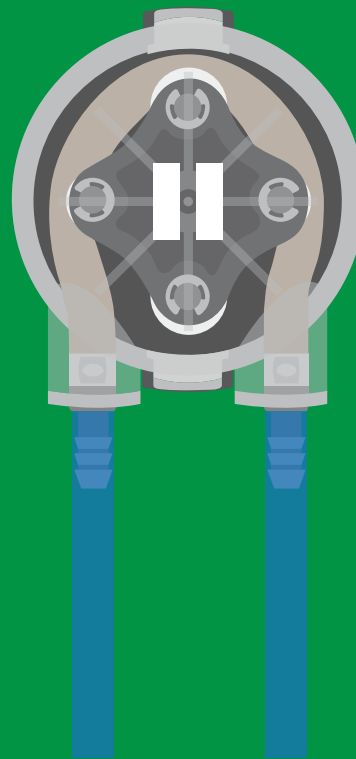
**\*OK** <cr>

**P,?** <cr>

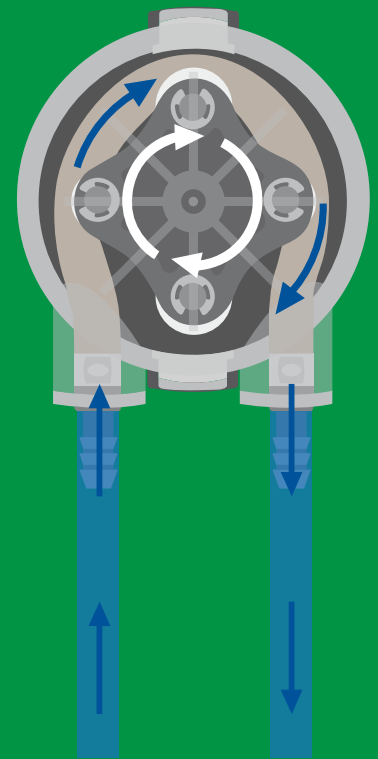
**?P,1** <cr> **or** **?P,0** <cr>  
paused unpaused  
**\*OK** <cr>



**P**



**P**



# Stop dispensing

## Command syntax

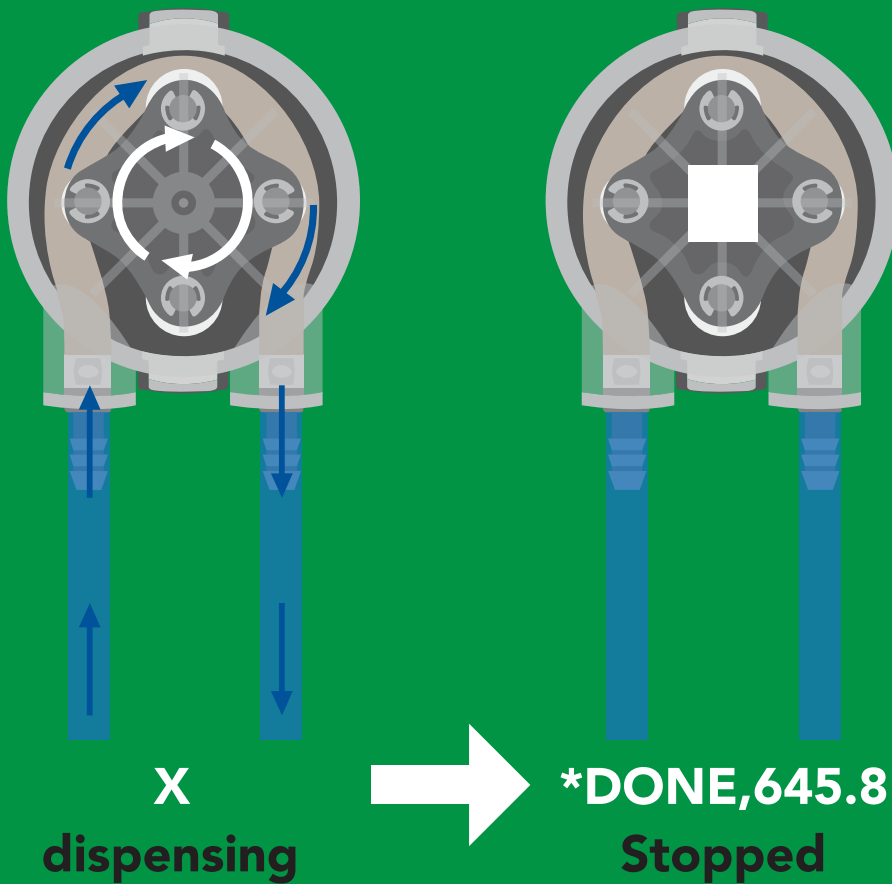
**X** <cr> stop dispensing

### Example

**X** <cr>

### Response

**\*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

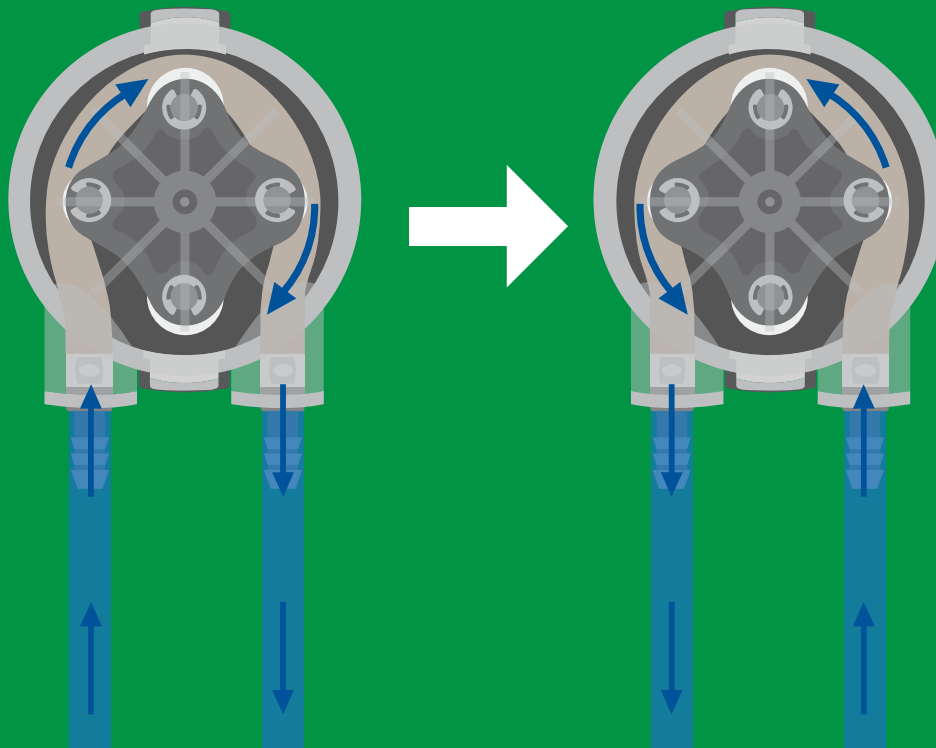
**Invert** <cr>

**\*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

### Response

**TV,? <cr>**

**?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

### Response

**Cal,146.2** <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,24.67** <cr>  
**\*OK** <cr>

## Response breakdown

**?PV, 24.67**



Pump input voltage

# Naming device

## Command syntax

Do not use spaces in the name

Name,n <cr> set name

Name, <cr> clears name

Name,? <cr> show name

n =

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

Up to 16 ASCII characters

## Example

## Response

Name, <cr>

\*OK <cr> name has been cleared

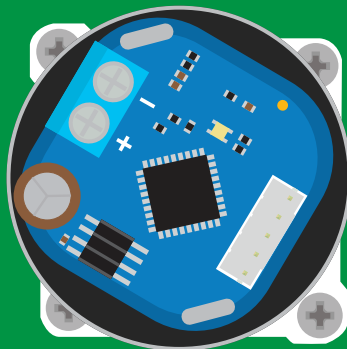
Name,zzt <cr>

\*OK <cr>

Name,? <cr>

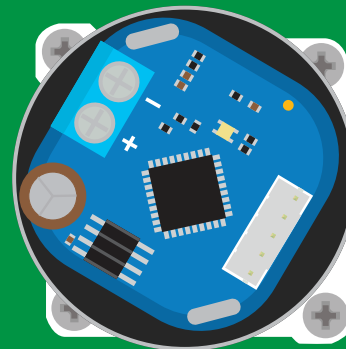
?Name,zzt <cr>  
\*OK <cr>

Name,zzt



\*OK <cr>

Name,?



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

# Device information

## Command syntax

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

### Example

```
i <cr>
```

### Response

```
?i,PMPL,1.1 <cr>  
*OK <cr>
```

## Response breakdown

```
?i, PMPL, 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>

\*OK <cr>

\*SL <cr>

Any command

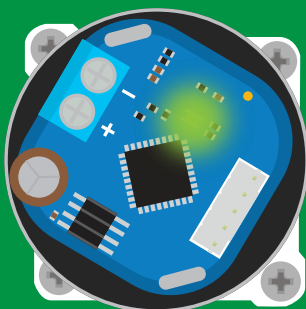
\*WA <cr> wakes up device

5V

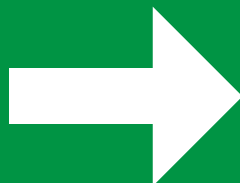
STANDBY	SLEEP
13.4 mA	0.415 mA

3.3V

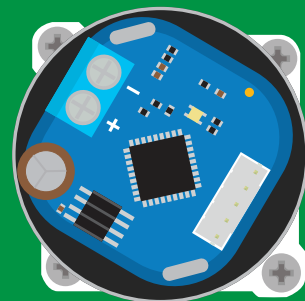
12.4 mA	0.13 mA
---------	---------



Standby  
13.4 mA



Sleep <cr>



Sleep  
0.415 mA



# Change baud rate

## Command syntax

Baud,n <cr> change baud rate

### Example

Baud,38400 <cr>

### Response

\*OK <cr>

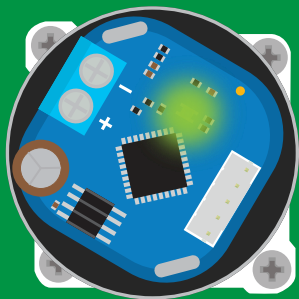
Baud,? <cr>

?Baud,38400 <cr>

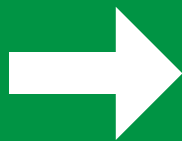
\*OK <cr>

n =

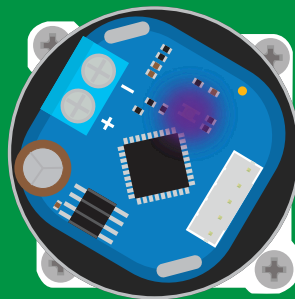
- 300
- 1200
- 2400
- 9600 default**
- 19200
- 38400
- 57600
- 115200



Standby

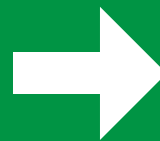


Baud,38400 <cr>

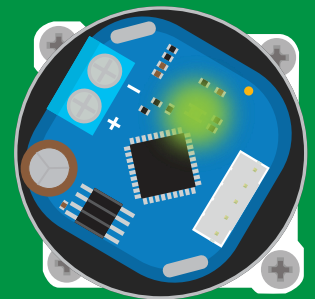


Changing baud rate

\*OK <cr>



(reboot)



Standby

# Protocol lock

## Command syntax

Locks device to UART mode.

Plock,1 <cr> enable Plock

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

Plock,? <cr> Plock on/off?

## Example

## Response

Plock,1 <cr>

\*OK <cr>

Plock,0 <cr>

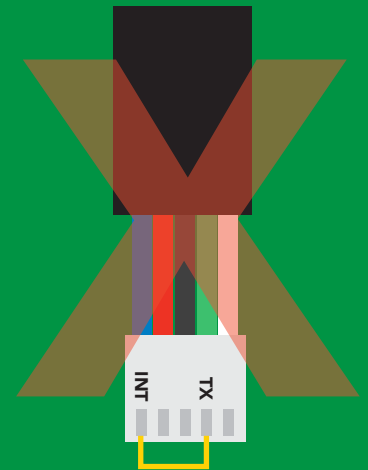
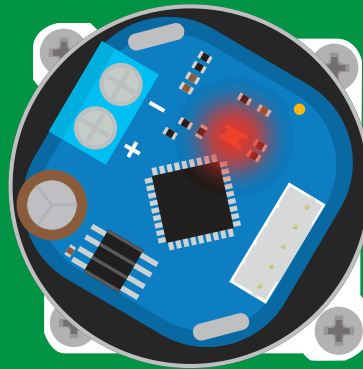
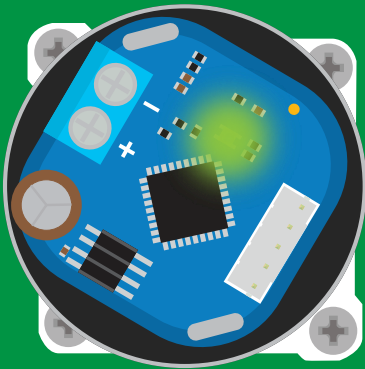
\*OK <cr>

Plock,? <cr>

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

Plock,1

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

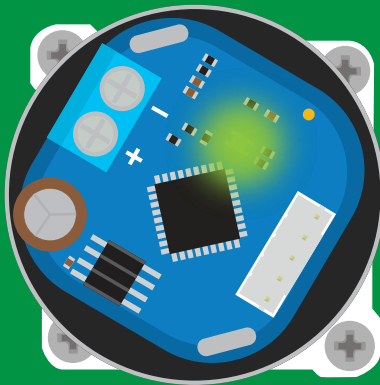
## Example

## Response

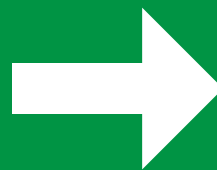
Factory <cr>

\*OK <cr>

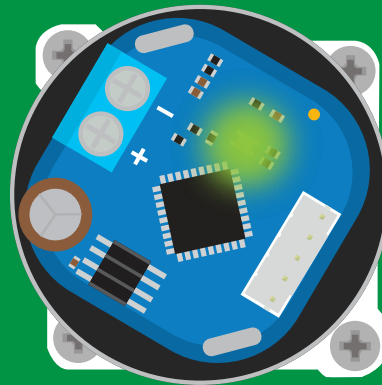
Factory <cr>



\*OK <cr>



(reboot)



\*RS <cr>

\*RE <cr>

Baud rate will not change

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

## Command syntax

Default I<sup>2</sup>C address 109 (0x6D)

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

n = any number 1 – 127

### Example

### Response

I2C,100 <cr>

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

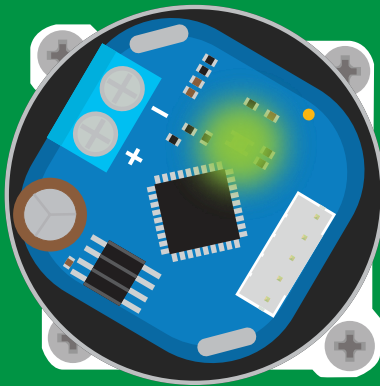
### Wrong example

### Response

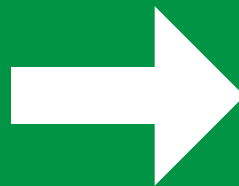
I2C,139 <cr> n ≠ 127

\*ER <cr>

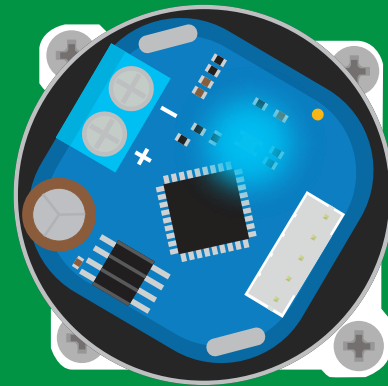
I2C,100



Green  
\*OK <cr>



(reboot)



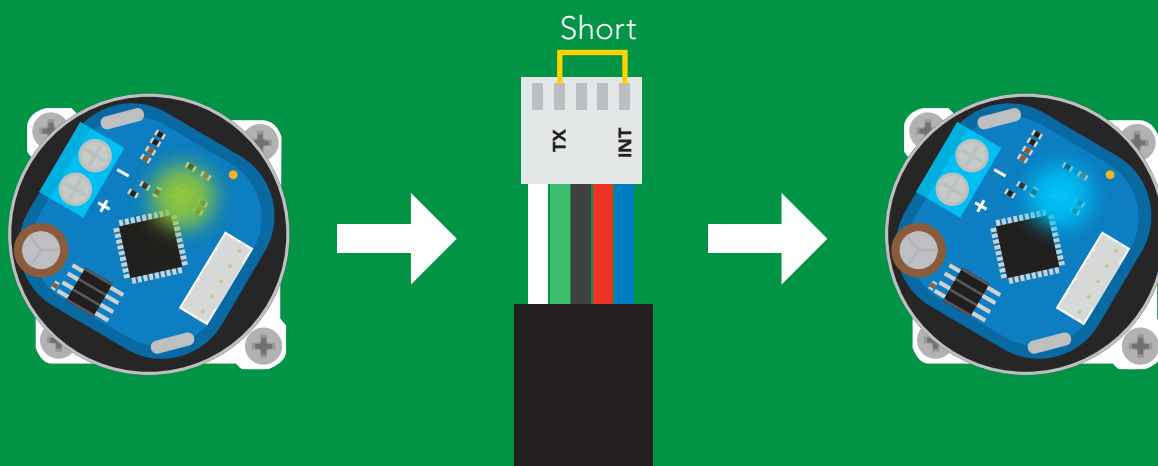
Blue  
now in I<sup>2</sup>C mode

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

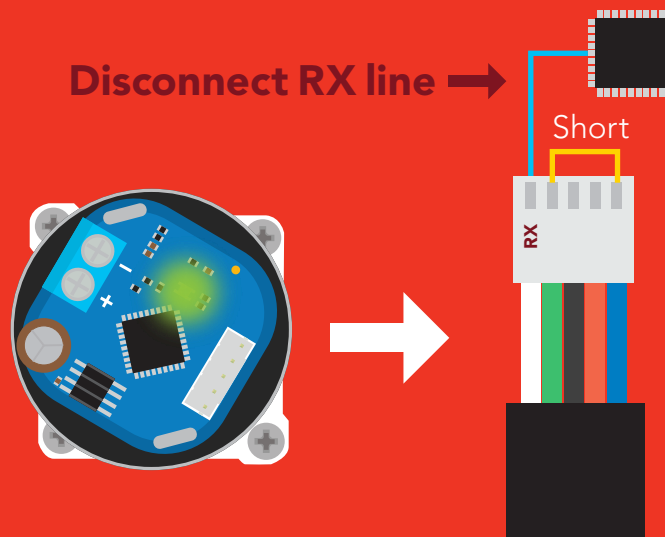
- Disconnect ground (power off)
- Disconnect TX and RX
- Connect TX to INT
- Confirm RX is disconnected
- Connect ground (power on)
- Wait for LED to change from **Green** to **Blue**
- Disconnect ground (power off)
- Reconnect all data and power

Manually switching to I<sup>2</sup>C will set the I<sup>2</sup>C address to 109 (0x6D)

## 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-L™ 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)  
**109 (0x6D) default**

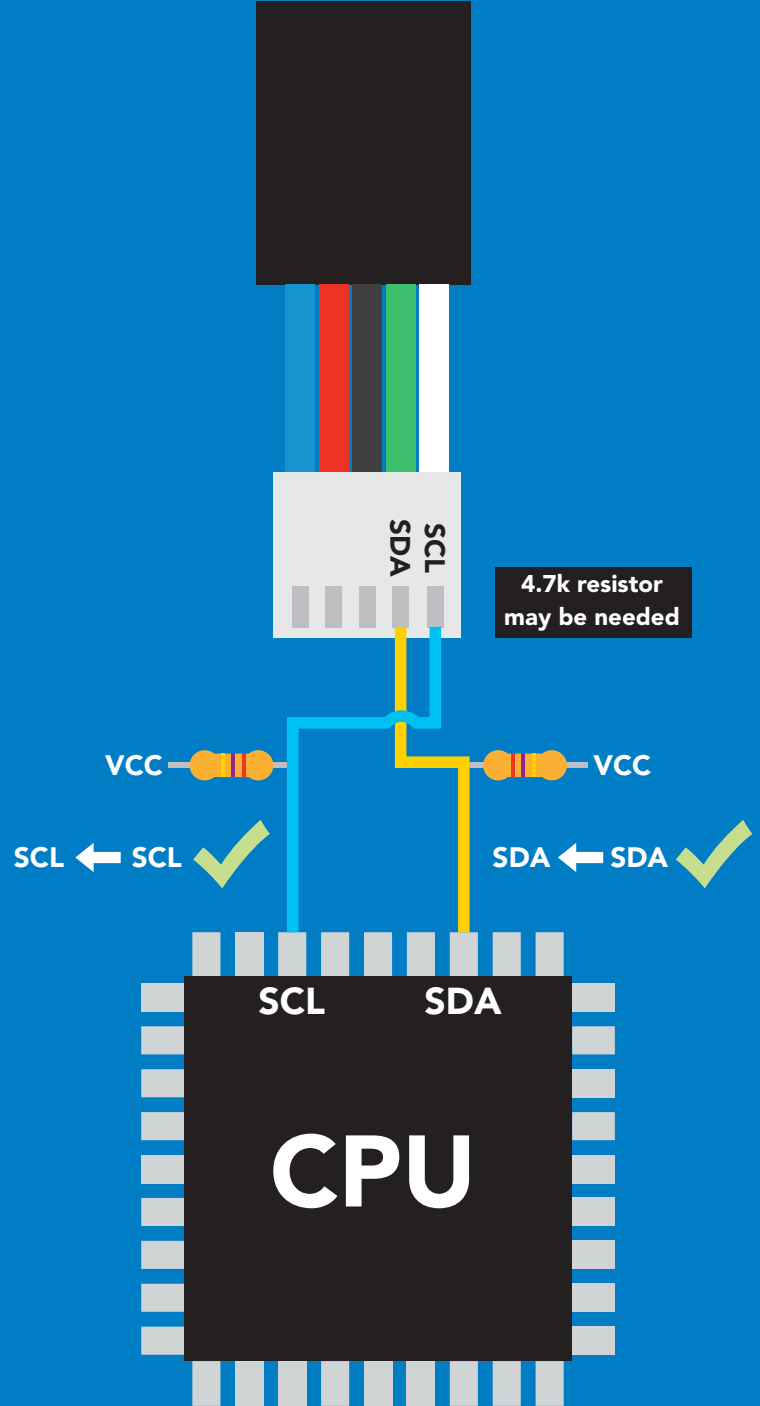
Vcc 3.3V – 5.5V

Clock speed 100 – 400 kHz

SDA 

SCL 





## Data format

Reading **volume**

Units **ml**

Encoding **ASCII**

Format **string**

Data type **long int**

Decimal places **none**

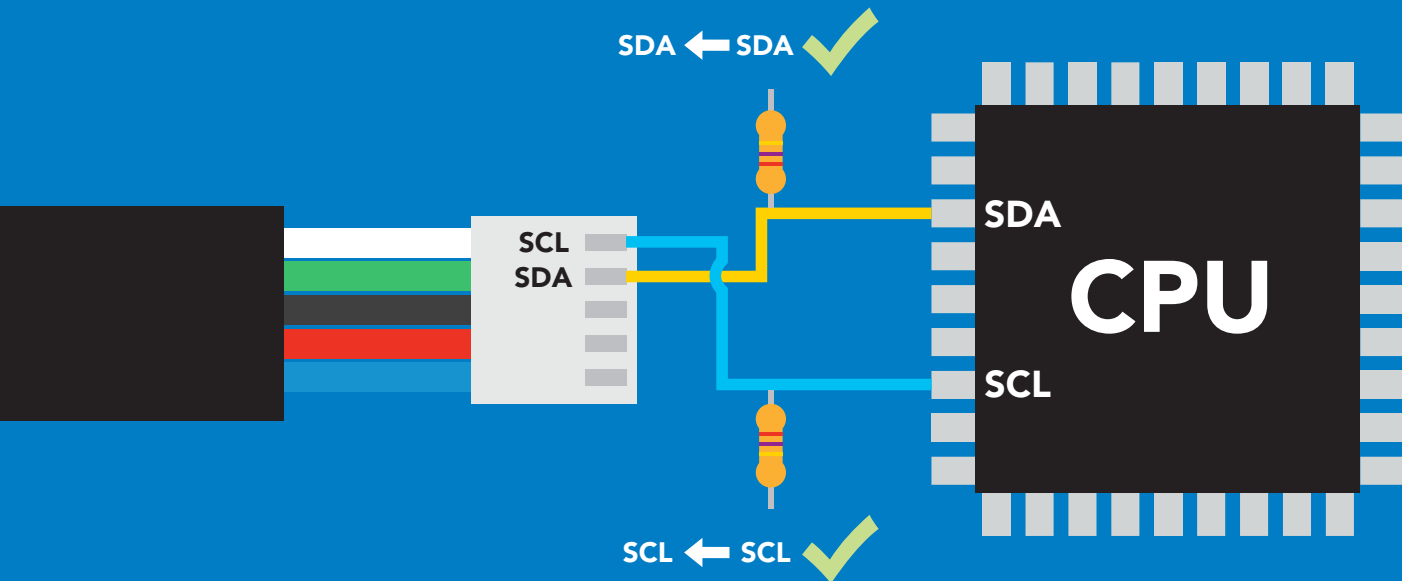
Smallest string **3 characters**

Largest string **39 characters**

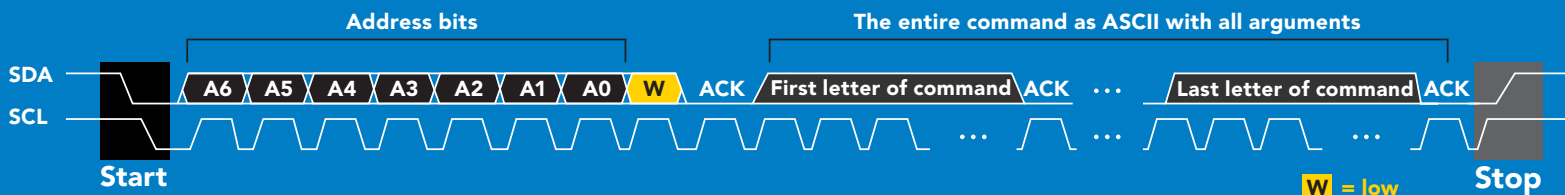
# Sending commands to device



## Example

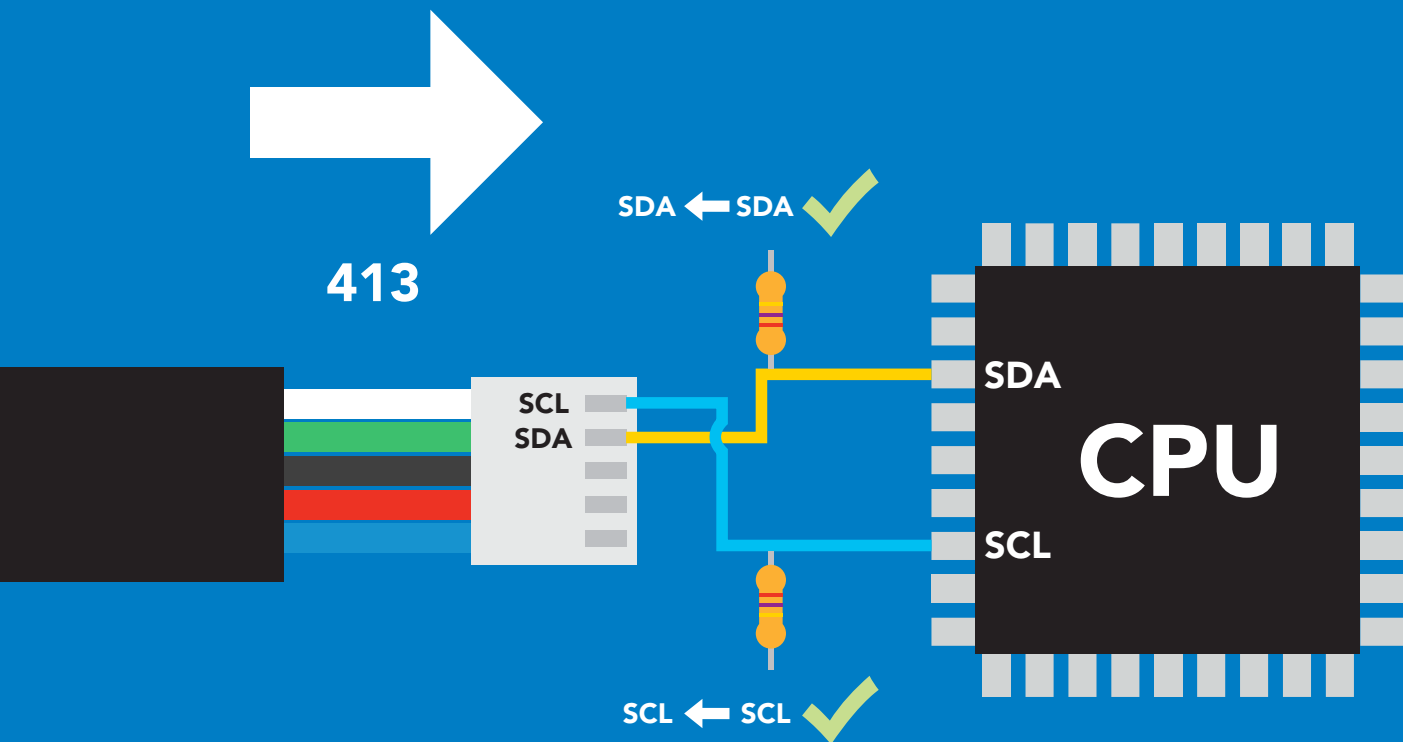
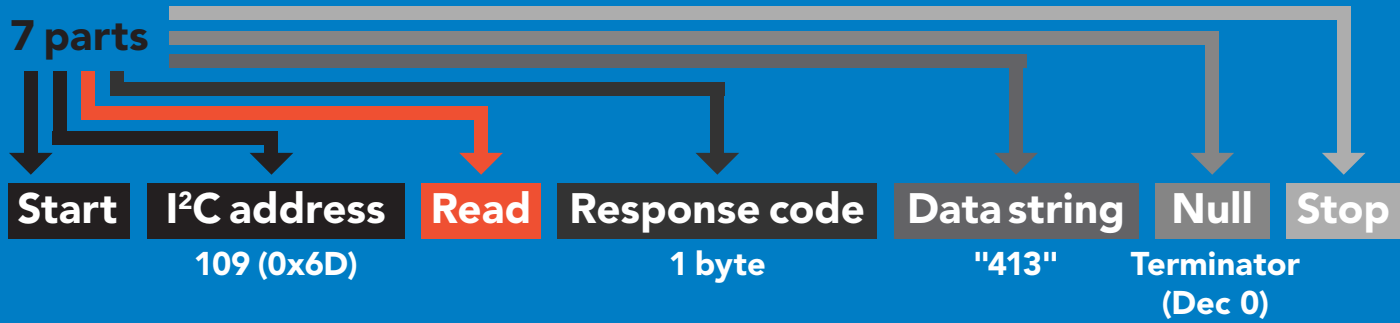


## Advanced

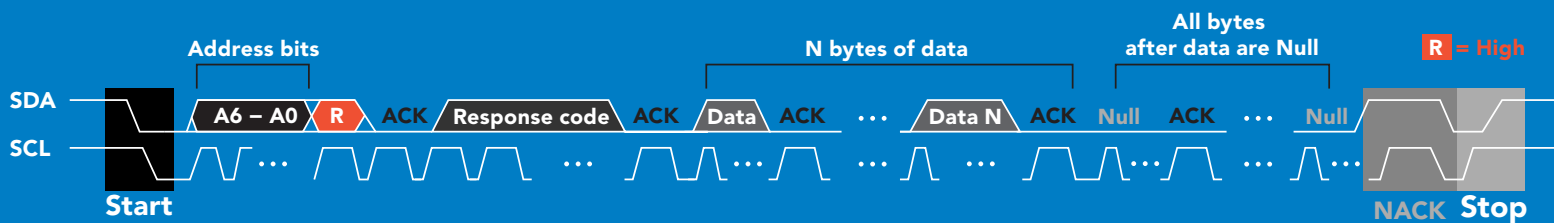




# Requesting data from device



## Advanced



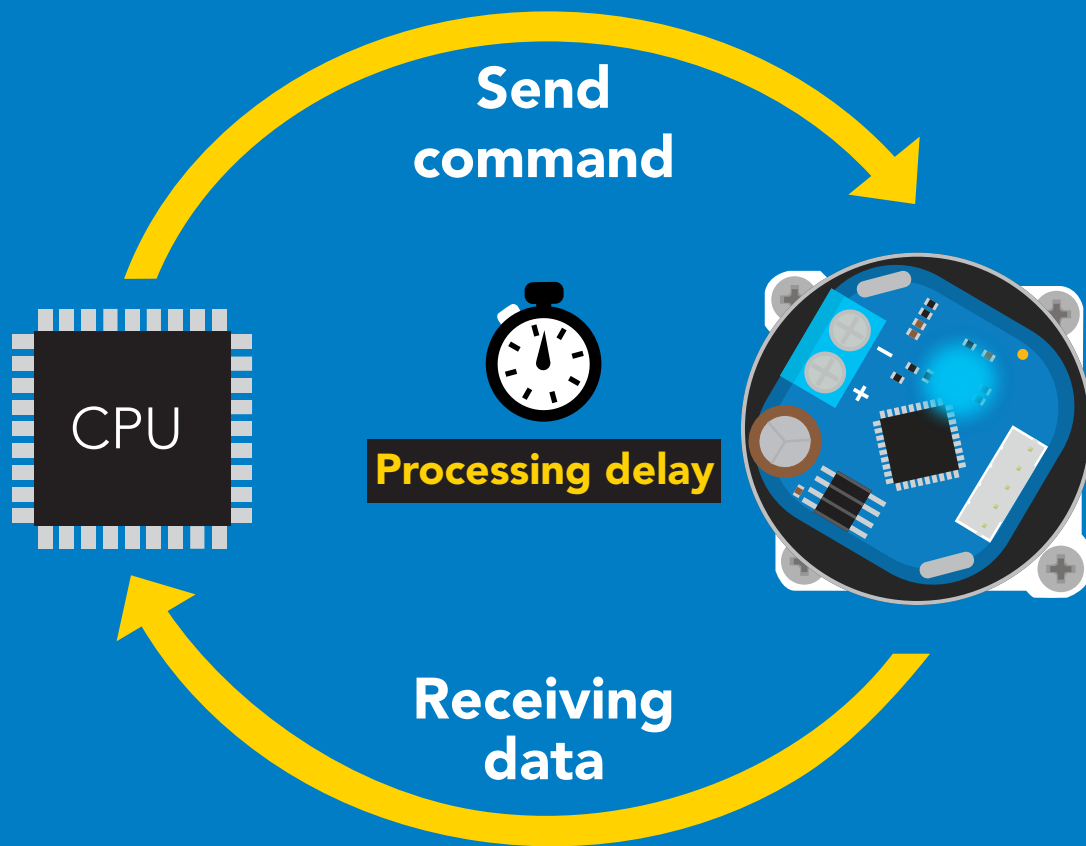
1    52    49    51    0    = 413

Dec    ASCII    Dec

# Response codes

After a command has been issued, a 1 byte response code can be read in order to confirm that the command was processed successfully.

*Reading back the response code is completely optional, and is not required for normal operation.*



## Example

```
I2C_start;  
I2C_address;  
I2C_write(EZO_command);  
I2C_stop;
```

```
delay(300);
```



```
Processing delay
```

```
I2C_start;  
I2C_address;  
Char[ ] = I2C_read;  
I2C_stop;
```

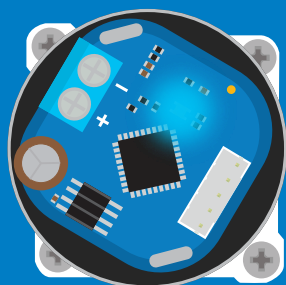
If there is no processing delay or the processing delay is too short, the response code will always be 254.

### Response codes

Single byte, not string

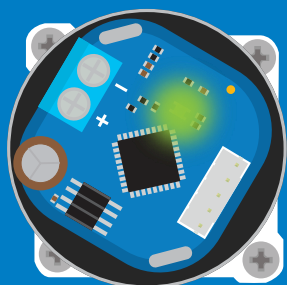
255	no data to send
254	still processing, not ready
2	syntax error
1	successful request

# LED color definition



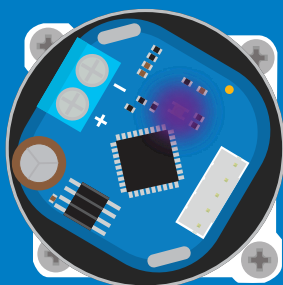
**Blue**

I<sup>2</sup>C standby



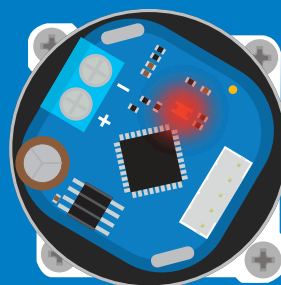
**Green**

Taking reading



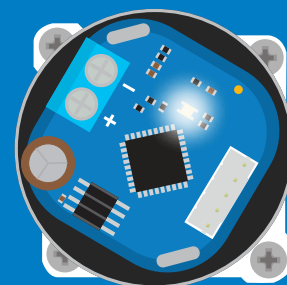
**Purple**

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



**Red**

Command  
not understood



**White**

Find

**5V**

LED ON  
**+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. 77</b>
<b>Cal</b>	performs calibration	<b>pg. 67</b>
<b>D</b>	dispense modes	<b>pg. 56 – 62</b>
<b>Factory</b>	enable factory reset	<b>pg. 76</b>
<b>Find</b>	finds device with blinking white LED	<b>pg. 54</b>
<b>i</b>	device information	<b>pg. 71</b>
<b>I2C</b>	change I <sup>2</sup> C address	<b>pg. 75</b>
<b>Invert</b>	invert dispensing direction	<b>pg. 65</b>
<b>L</b>	enable/disable LED	<b>pg. 53</b>
<b>Name</b>	set/show name of device	<b>pg. 70</b>
<b>O</b>	enable/disable parameters	<b>pg. 68</b>
<b>P</b>	pauses the pump during dispensing	<b>pg. 63</b>
<b>Plock</b>	enable/disable protocol lock	<b>pg. 74</b>
<b>Pv</b>	check pump voltage	<b>pg. 69</b>
<b>R</b>	returns a single reading	<b>pg. 55</b>
<b>Sleep</b>	enter sleep mode/low power	<b>pg. 73</b>
<b>Status</b>	retrieve status information	<b>pg. 72</b>
<b>Tv</b>	total volume dispensed	<b>pg. 66</b>
<b>X</b>	stop dispensing	<b>pg. 64</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

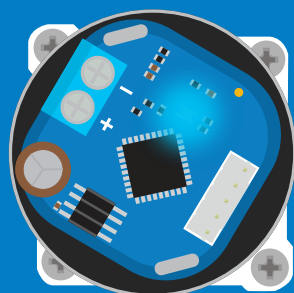
  
Wait 300ms    **1**    **0**  
Dec    Null

L,0

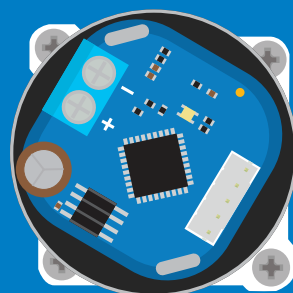
  
Wait 300ms    **1**    **0**  
Dec    Null

L,?

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

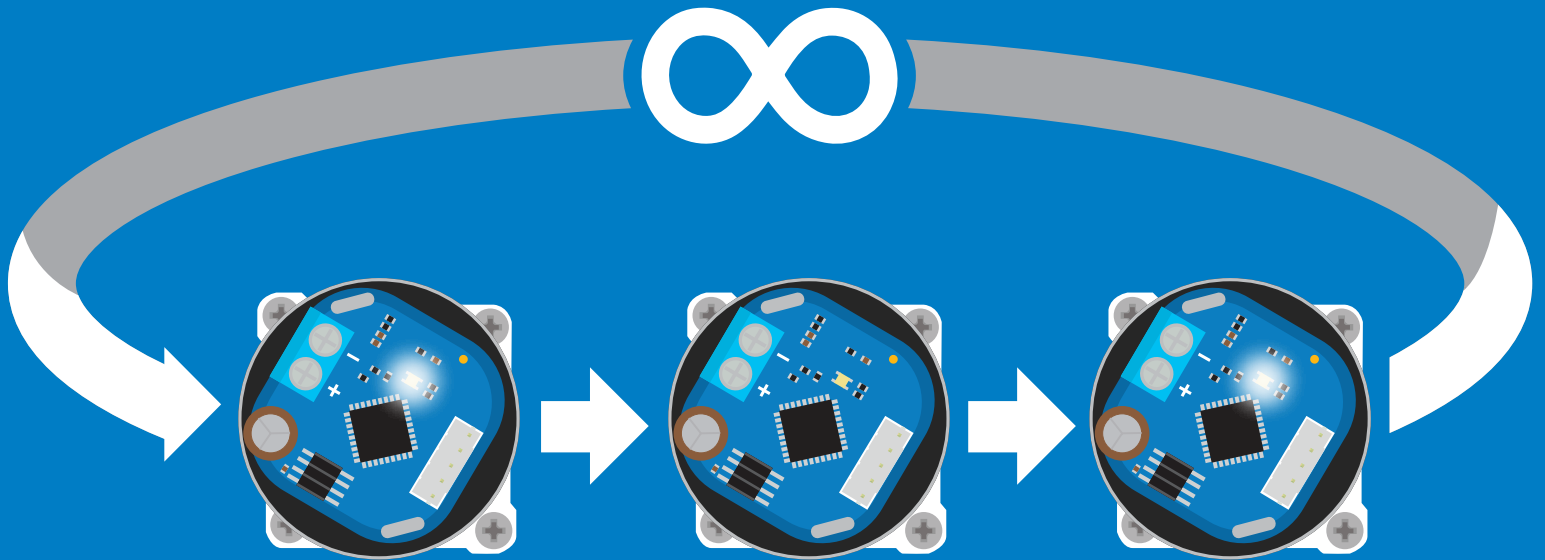
## Response

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**

## Response

  
**Wait 300ms**

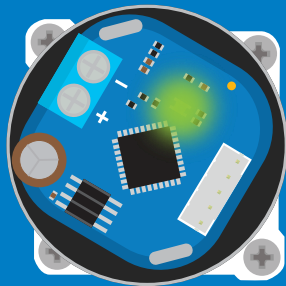
<b>1</b>	<b>25</b>	<b>0</b>
Dec	ASCII	Null

(If issued half way through dispensing 50ml)

  
**Wait 300ms**

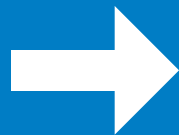
<b>1</b>	<b>50</b>	<b>0</b>
Dec	ASCII	Null

(If issued once dispensing has stopped)

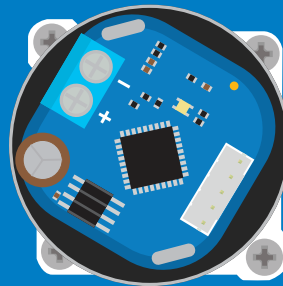


**Green**

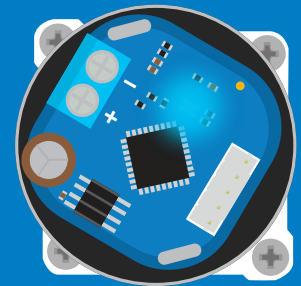
Taking reading



**Wait 600ms**



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-L™ 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 ~750ml/min (with supplied tubing)

D,-\*



1  
Dec

0  
Null

pump will continuously run in reverse at ~750ml/min (with supplied tubing)

D,?



1  
Dec

?D,\*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,-40



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

40 ml will be dispensed  
*in reverse*

D,?



**1** **?D,-40,0** **0**  
Dec ASCII Null

## Response breakdown

?D,-40,0

↑ ↑  
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,7000,20

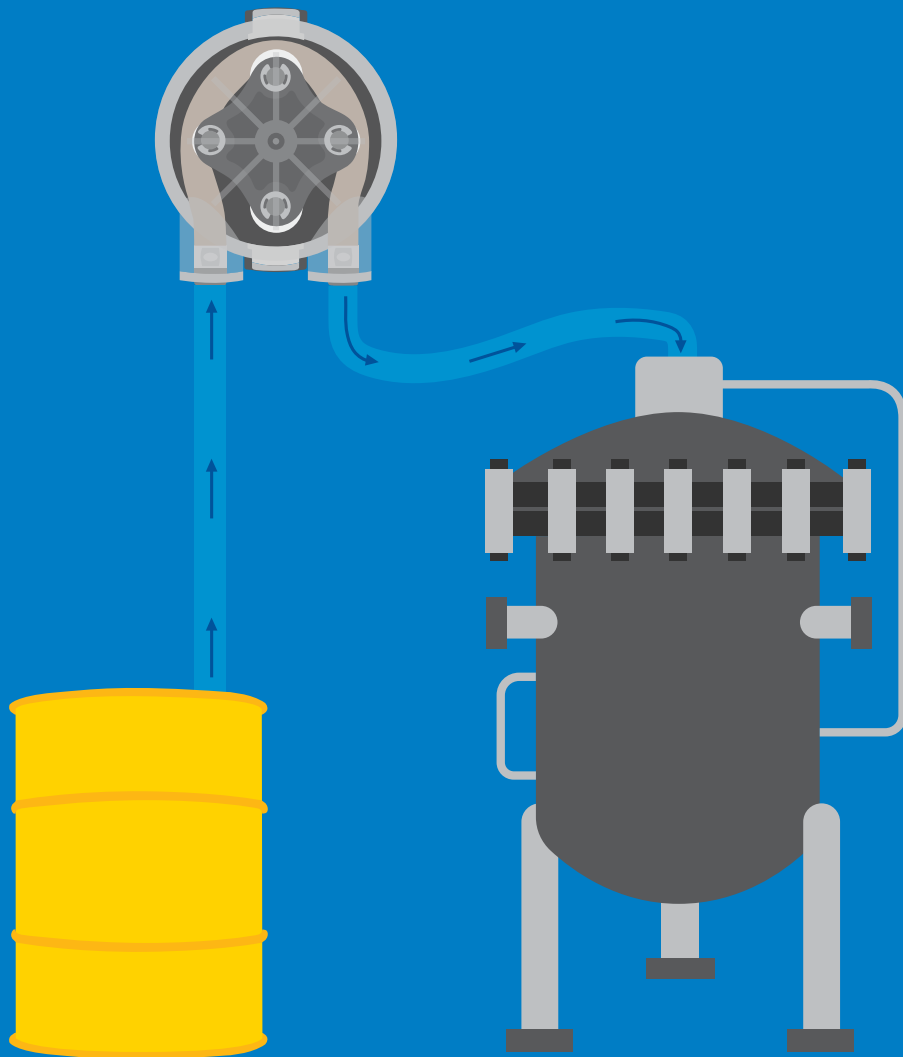
## Response

  
Wait 300ms

1  
Dec

0  
Null

Dispense 7000ml over 20 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-L™ 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

## Response

DC,50,40

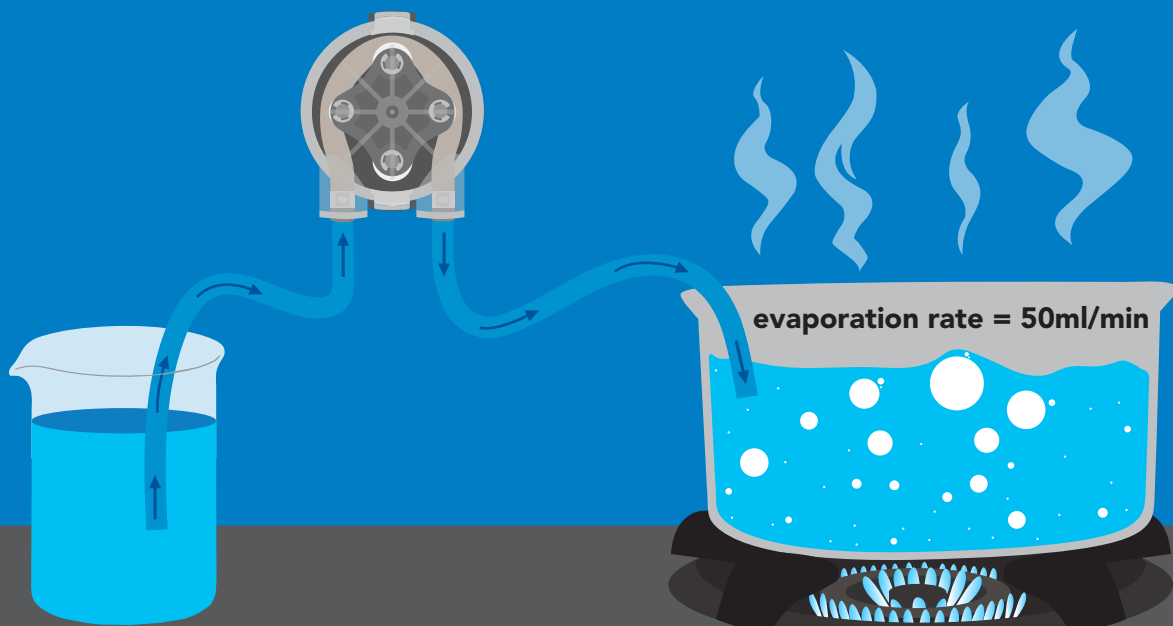
 Wait 300ms    **1**    **0**    Dispense 50ml per minute for 40 minutes  
Dec    Null

DC,?

 Wait 300ms    **1**    **?maxrate,385**    **0**  
Dec    ASCII    Null

The maximum flow rate is determined after calibration.  
If the flowrate entered is too fast the EZO-PMP-L™ 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,100**

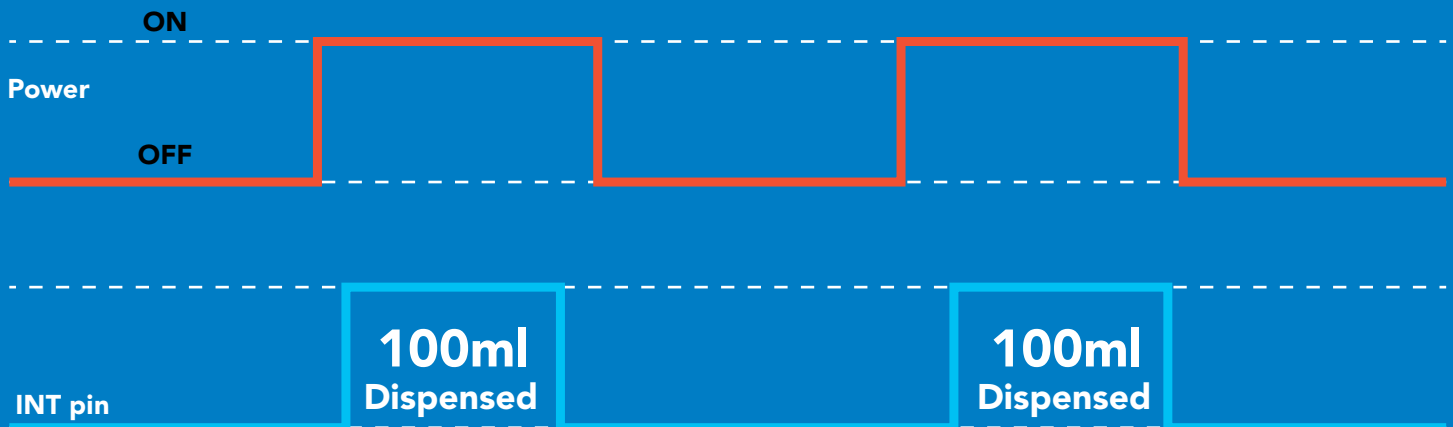
 Wait 300ms **1** **0**  
Dec Null

**Dstart,off**

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

**D,?**

 Wait 300ms **1** **?Dstart,100** **0** or  Wait 300ms **1** **?Dstart,0** **0**  
Dec ASCII Null Dec ASCII 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-L™ 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

**Dstart,\***



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

Pump will startup and continuously run at ~750ml/min (with supplied tubing)

**Dstart,-\***



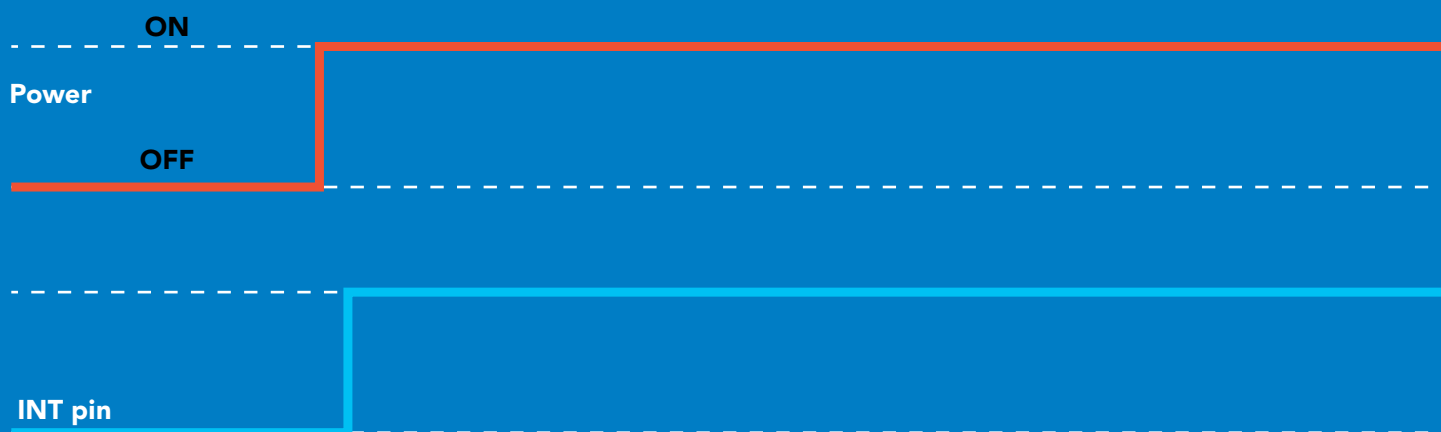
**1** **0**  
Dec Null

Pump will startup and continuously run in reverse at ~750ml/min (with supplied tubing)

**Dstart,?**



**1** **?Dstart,\*** **0**  
Dec ASCII Null



# Dose Over time at startup

Pump a fixed volume over a fixed time at startup

## Command syntax

300ms  processing delay

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

## Example

## Response

Dstart,7000,20



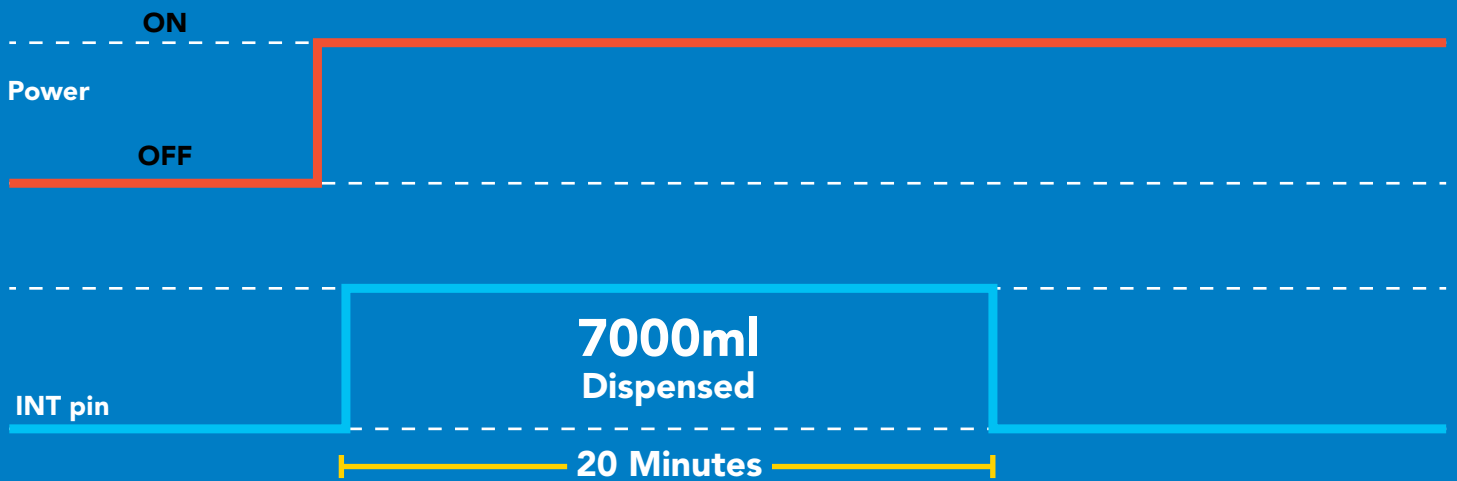
**1**   **0**  
Dec   Null

Pump will startup and dispense 7000ml over 20 minutes

Dstart,?



**1**   **?Dstart,7000,10.00**   **0**  
Dec   ASCII   Null



# 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**

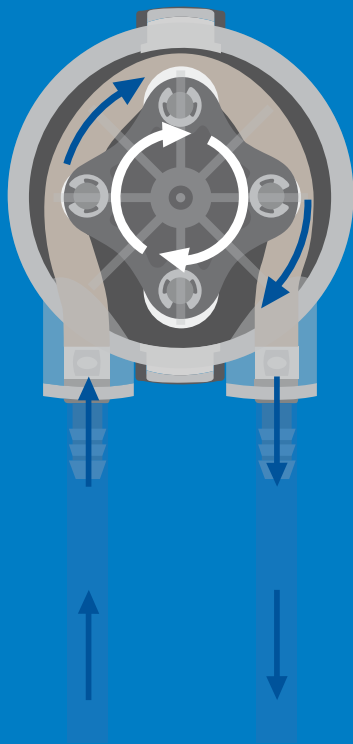
 Wait 300ms  
**1** **0**  
Dec Null

**P,?**

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

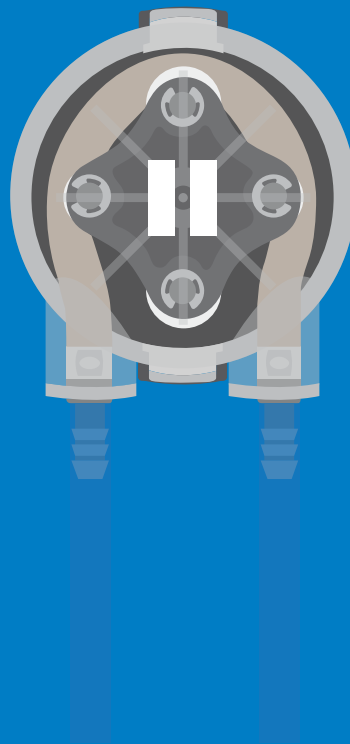
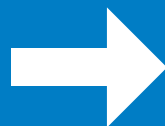
or

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



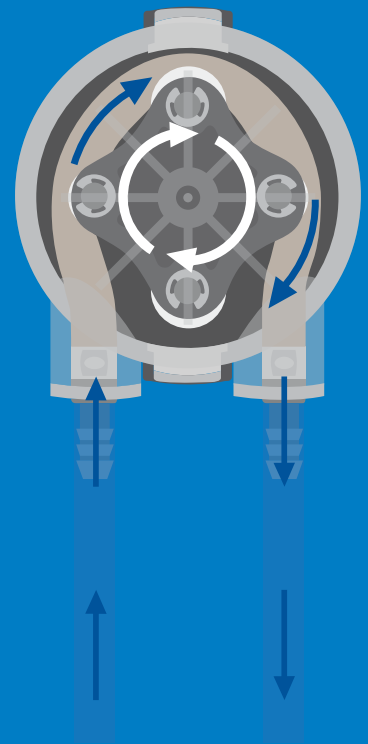
dispensing

**P**



paused

**P**



dispensing

# Stop dispensing

## Command syntax

300ms  processing delay

**X** stop dispensing

## Example

## Response

**X**

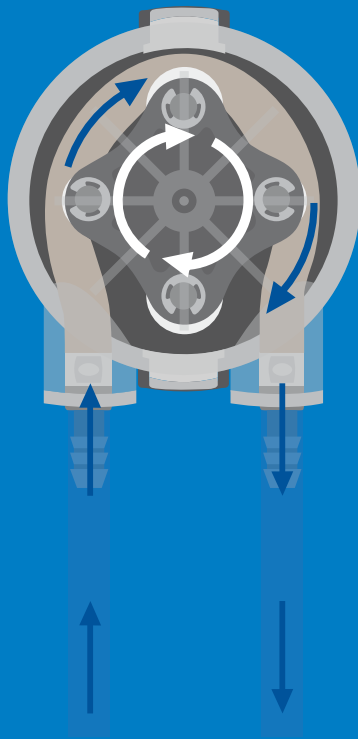
  
Wait 300ms

**1**  
Dec

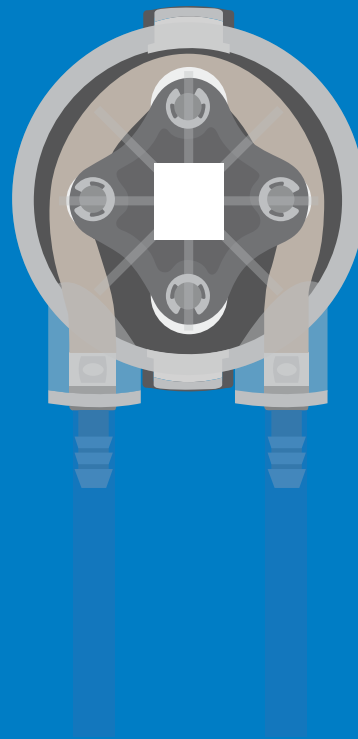
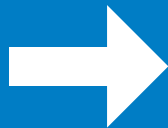
**\*DONE,v**  
ASCII

**0**  
Null

**v = volume dispensed**



**X**  
dispensing



**\*DONE,645.8**  
Stopped



# Invert dispensing direction

300ms  processing delay

## Command syntax

Invert direction will be retained if power is cut

**Invert** changes dispensing direction of pump

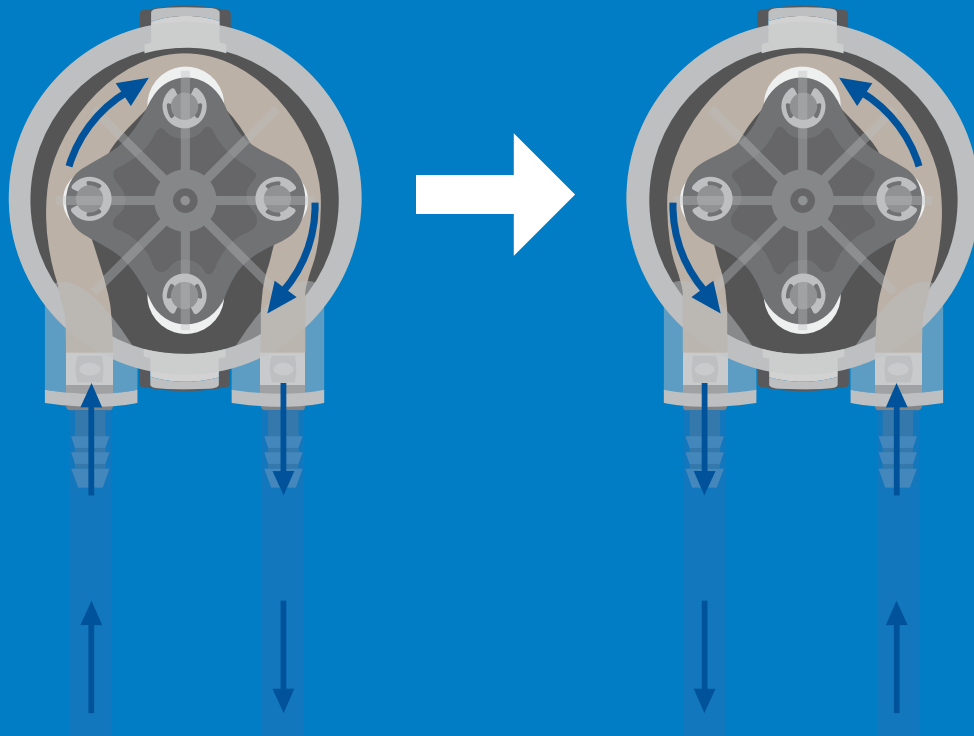
## Example

**Invert**

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

**Invert,?**

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



# 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,?**

 **1** **?TV,623.00** **0**  
Wait 300ms Dec ASCII Null

**ATV,?**

 **1** **?ATV,434.50** **0**  
Wait 300ms Dec ASCII Null

**clear**

 **1** **0** **total now 0.00**  
Wait 300ms Dec Null

This data will be lost if the power is cut.

# 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,146.2**



 **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**  
Dec Null

enable volume being pumped

O,TV,0

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

disable total volume pumped

O,ATV,1

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

enable absolute volume pumped

O,?

 **Wait 300ms** **1** **?O,V,TV,ATV** **0**  
Dec ASCII Null

if all three are enabled

# Pump voltage

## Command syntax

300ms  processing delay

PV,? check pump voltage

## Example

## Response

PV,?



Wait 300ms

1

Dec

?PV,24.67

ASCII

0

Null

## Response breakdown

?PV, 24.67



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																	
Name,?	show name																	

Up to 16 ASCII characters

## Example

## Response

Name,



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

name has been cleared

Name,zzt



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

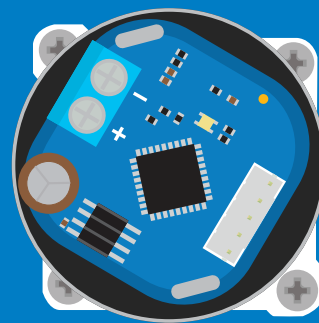
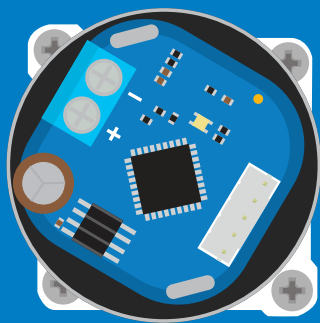
Name,?



**1** **?Name,zzt** **0**  
Dec ASCII 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,PMPL, 1.1

ASCII

0

Null

## Response breakdown

?i, PMPL, 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**

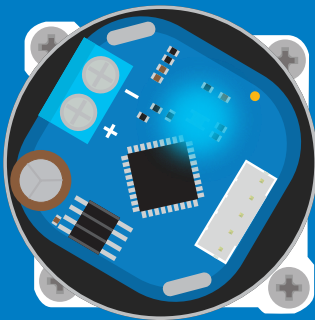
**wakes up device**

**5V**

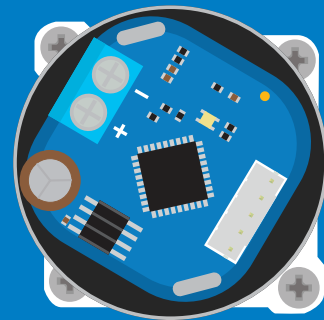
STANDBY	SLEEP
<b>13.4 mA</b>	<b>0.415 mA</b>

**3.3V**

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



**Standby**



**Sleep**

# Protocol lock

## Command syntax

300ms  processing delay

Plock,1 enable Plock

Plock,0 disable Plock

Plock,? Plock on/off?

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

default

## Example

## Response

Plock,1

  
Wait 300ms


1	0
Dec	Null

Plock,0

  
Wait 300ms

1	0
Dec	Null

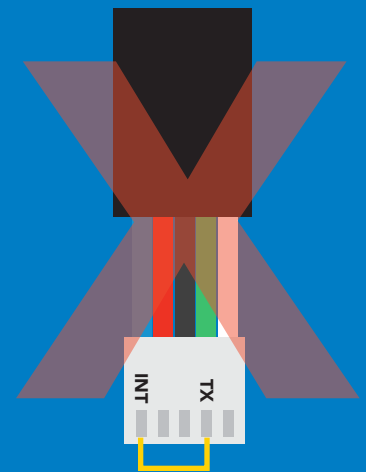
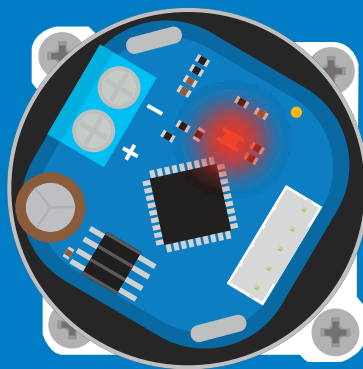
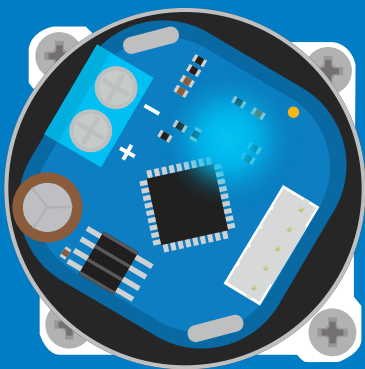
Plock,?

  
Wait 300ms

1	?Plock,1	0
Dec	ASCII	Null

Plock,1

Baud, 9600



cannot change to UART

cannot change to UART

# 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)

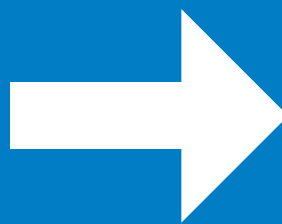
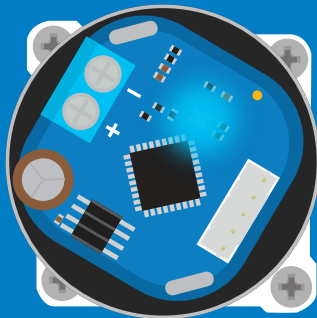
## Warning!

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.

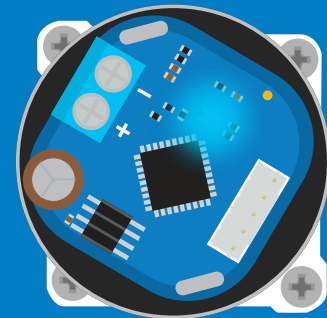
Default I<sup>2</sup>C address is 109 (0x6D).

n = any number 1 – 127

I2C,101



(reboot)



# Factory reset

## Command syntax

Factory reset will not take the device out of I<sup>2</sup>C mode.

Factory enable factory reset

I<sup>2</sup>C address will not change

## Example

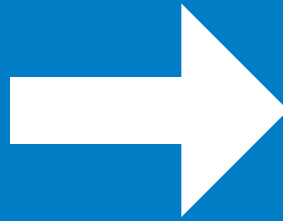
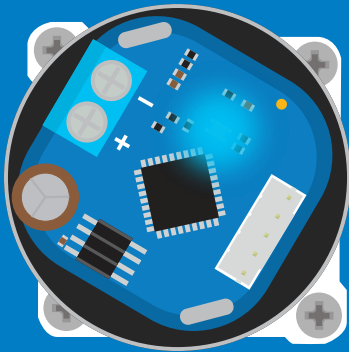
## Response

Factory

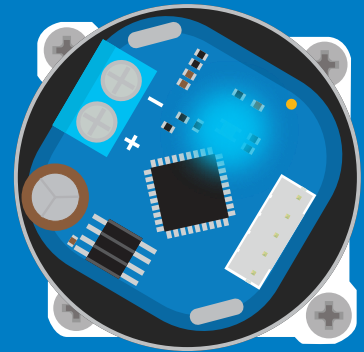
device reboot  
(no response given)

Clears calibration  
LED on  
Response codes enabled

Factory



(reboot)



# Change to UART mode

## Command syntax

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

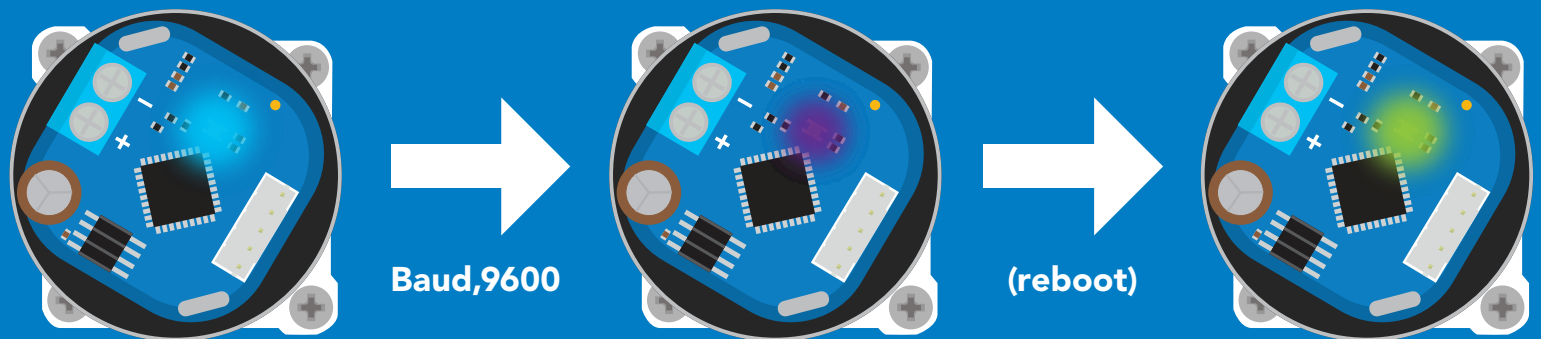
### Example

Baud,9600

### Response

reboot in UART mode  
(no response given)

n = [ 300  
1200  
2400  
9600  
19200  
38400  
57600  
115200

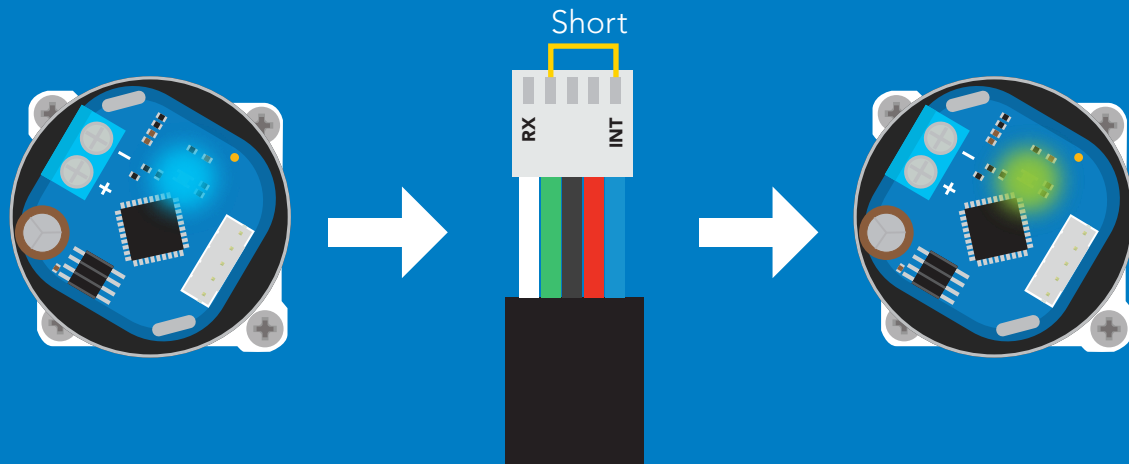


Changing to UART mode

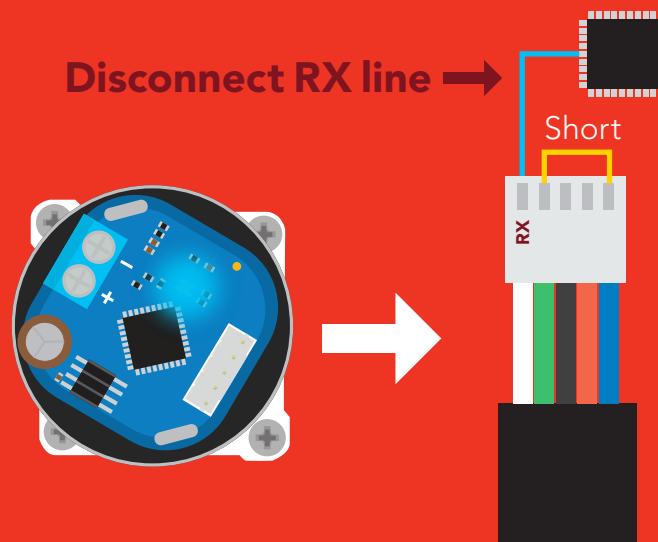
# Manual switching to UART

- 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

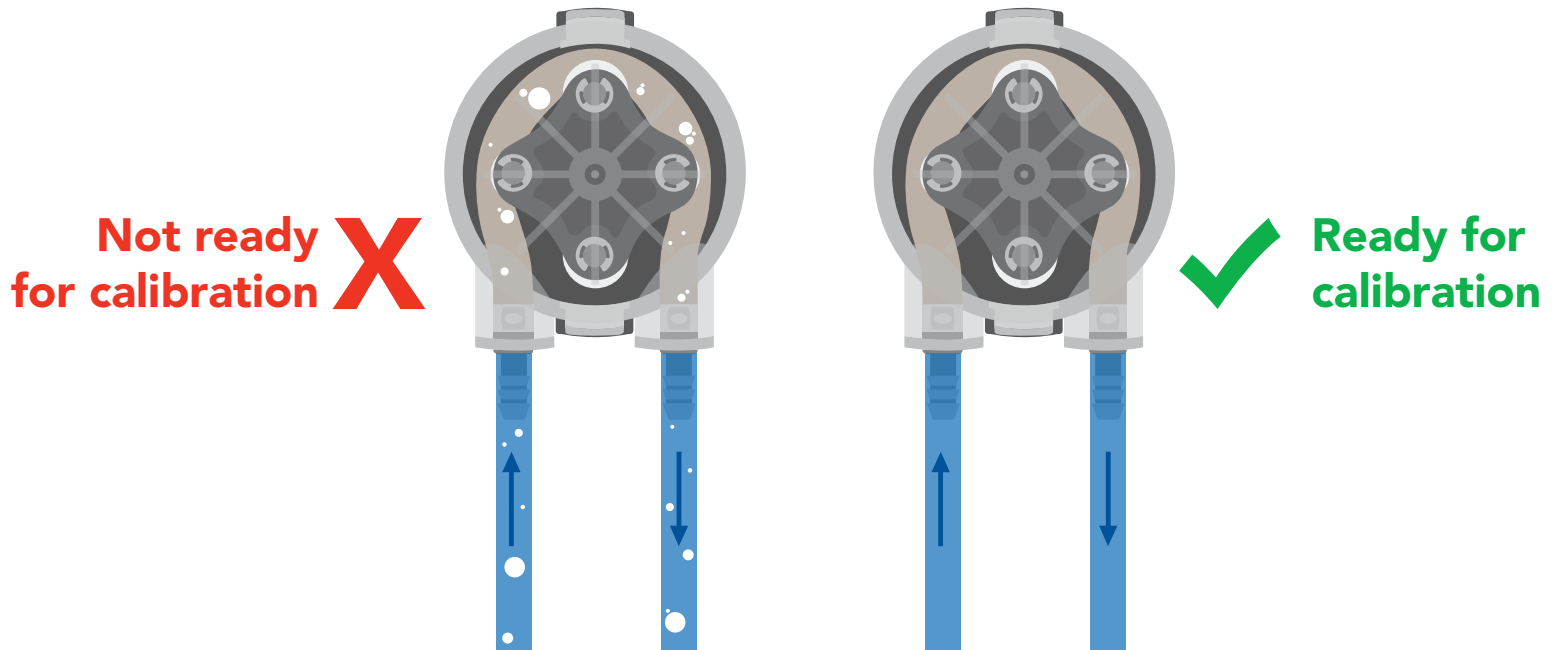


## Wrong Example



# Calibration theory

**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.

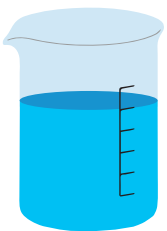


## Calibration types

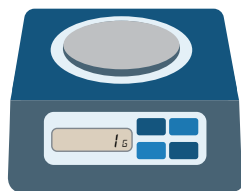
**Volume calibration**  
**Volume over time calibration**

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 150ml.

## Equipment needed for calibration



Or



An accurate beaker of at least 250ml.

An accurate scale with a resolution of at least 1 gram.

**1 gram of water = 1ml**  
**250 grams of water = 250ml**

# Calibration procedure

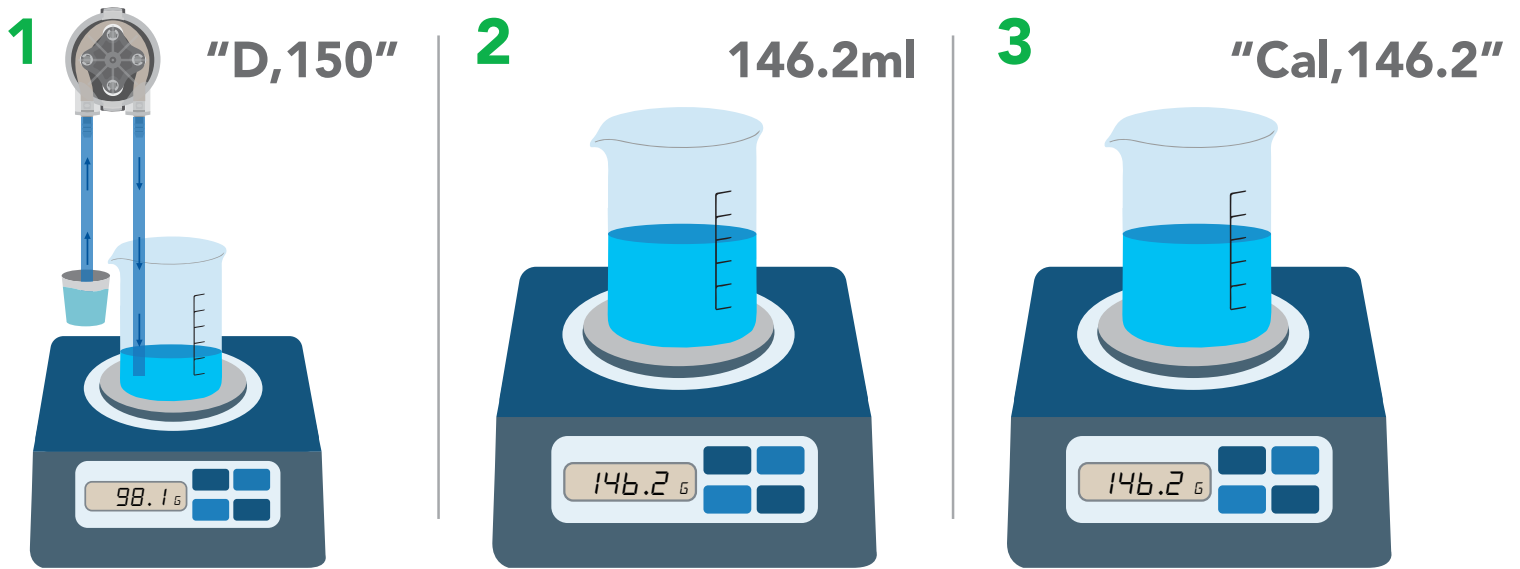
Calibration should be done with water and not a chemical

**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 150ml.



1. Instruct the pump to dispense 150ml into a 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 (150ml). It has now been calibrated to all volumes.

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.

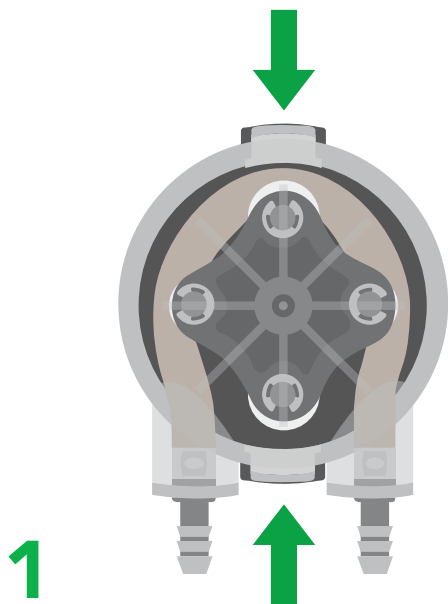


# Interrupt pin

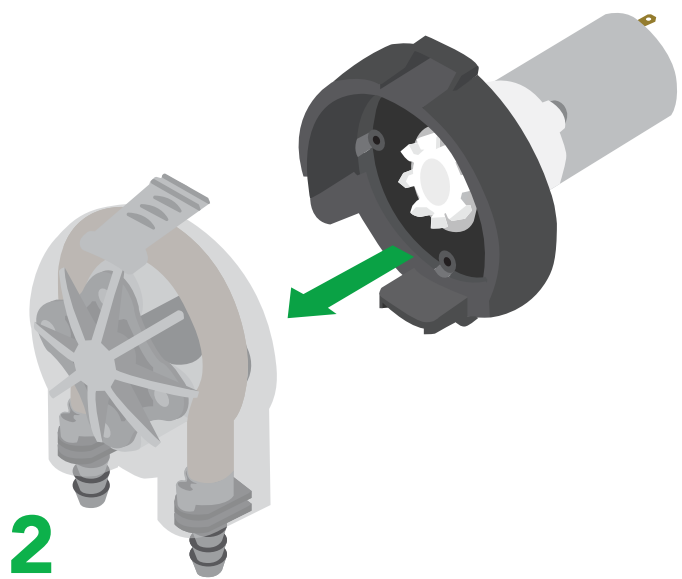
When the pump is dispensing the interrupt pin goes high.



# Removing cassette



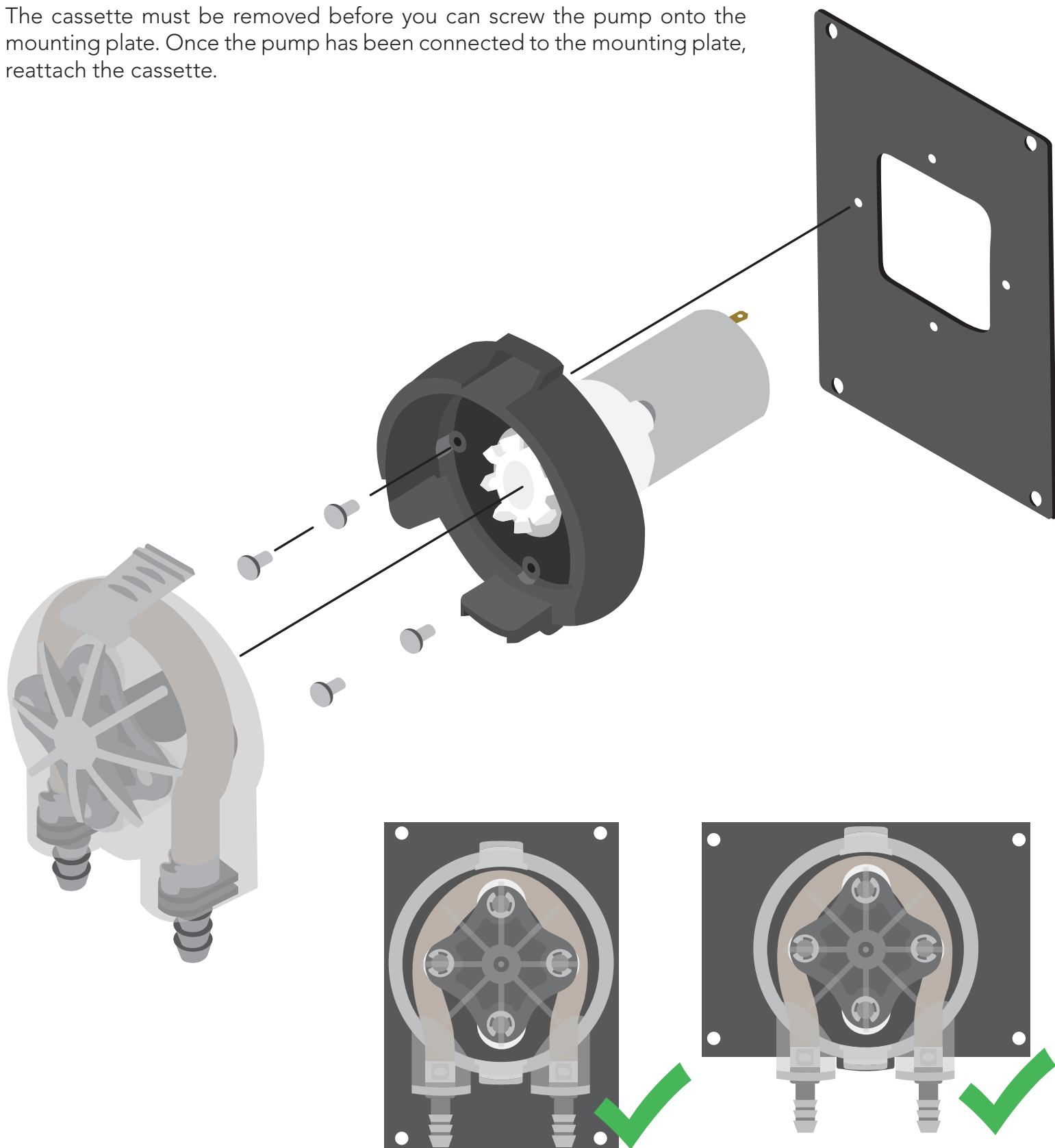
Press both release tabs on the cassette.



Pull the cassette off of the motor.

# Mounting plate (optional)

The cassette must be removed before you can screw the pump onto the mounting plate. Once the pump has been connected to the mounting plate, reattach the cassette.



# Datasheet change log

## Datasheet V 1.4

Added a new page about power supplies on page 8.

## Datasheet V 1.3

Revised table of contents and added invert dispensing direction command on pages 30 & 64.

## Datasheet V 1.2

Revised naming device info on pages 36 & 67.

## Datasheet V 1.1

Added Mounting plate info on pg. 9

## Datasheet V 1.0

New Datasheet.

# Firmware updates

V1.0 – Initial release (March, 2021)

V1.01 - (May 26, 2021)

- Changed minimum voltage for activation

V1.02 - (Feb 15, 2022)

- Internal update for new part compatibility.

V1.03 - (April 20, 2022)

- Expands dstart command with \* and dispense over time

# Warranty

Atlas Scientific™ Warranties the EZO-PMP-L™ Large Embedded Dosing Pump to be free of defect during the debugging phase of device implementation, or 30 days after receiving the EZO-PMP-L™ Large 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-L™ Large Embedded Dosing Pump is inserted into a bread board, or shield. If the EZO-PMP-L™ Large Embedded Dosing Pump is being debugged in a bread board, the bread board must be devoid of other components. If the EZO-PMP-L™ Large Embedded Dosing Pump is being connected to a microcontroller, the microcontroller must be running code that has been designed to drive the EZO-PMP-L™ Large Embedded Dosing Pump exclusively and output the EZO-PMP-L™ Large 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-L™ Embedded Dosing Pump warranty:**

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