RTD Temperature

EZO™

Circuit

Reads

Temperature

−126.000 °C – 1254 °C

Resolution

0.001

Accuracy

+/− (0.10°C + 0.0017 × °C)

Max rate

1 reading per sec

Supported probes

Any type & brand
PT-100 or PT-1000 RTD

Calibration

Single point

Temperature output

°C, °K, or °F

Data protocol

UART & I²C

Default I²C address

102 (0x66)

Operating voltage

3.3V – 5.5V

Data format

ASCII

Onboard Data Logger

50 Readings

Electrical Isolation not needed
This is sensitive electronic equipment. Get this device working in a solderless breadboard first. Once this device has been soldered it is no longer covered by our warranty.

This device has been designed to be soldered and can be soldered at any time. Once that decision has been made, Atlas Scientific no longer assumes responsibility for the device’s continued operation. The embedded systems engineer is now the responsible party.

**SOLDERING THIS DEVICE VOIDS YOUR WARRANTY.**

Get this device working in a solderless breadboard first! Do not embed this device without testing it in a solderless breadboard!
Table of contents

Circuit dimensions ................................................. 4
Power consumption .............................................. 4
Absolute max ratings ........................................... 4
Temperature circuit range ..................................... 5
Temperature circuit accuracy ................................. 5
Atlas Scientific PT-1000 probe ................................. 6
Using other brand PT-100/PT-1000 ......................... 7
Operating principle .............................................. 8
Calibration theory ................................................ 9
On board data logger ............................................ 10
Correct wiring ...................................................... 12
Available data protocols ...................................... 13

UART

UART mode .......................................................... 15
Default state ....................................................... 16
Receiving data from device .................................... 17
Sending commands to device ................................. 18
LED color definition ........................................... 19
UART quick command page ................................... 20
LED control ......................................................... 21
Find ................................................................. 22
Continuous reading mode ..................................... 23
Single reading mode ............................................ 24
Calibration .......................................................... 25
Export/import calibration ..................................... 26
Temperature scale ................................................ 27
Enable/disable data logger .................................... 28
Memory recall ...................................................... 29
Memory clear ...................................................... 30
Naming device ..................................................... 31
Device information ............................................. 32
Response codes .................................................. 33
Reading device status ......................................... 34
Sleep mode/low power ........................................ 35
Change baud rate ............................................... 36
Protocol lock ...................................................... 37
Factory reset ....................................................... 38
Change to I²C mode ............................................. 39
Manual switching to I²C ........................................ 40

I²C

I²C mode ............................................................ 42
Sending commands ............................................. 43
Requesting data ................................................... 44
Response codes ................................................... 45
LED color definition ........................................... 46
I²C quick command page ....................................... 47
LED control ......................................................... 48
Find ................................................................. 49
Taking reading ...................................................... 50
Calibration .......................................................... 51
Export/import calibration ..................................... 52
Temperature scale ................................................ 53
Enable/disable data logger .................................... 54
Memory recall ...................................................... 55
Memory clear ...................................................... 56
Device information ............................................. 57
Reading device status ......................................... 58
Sleep mode/low power ........................................ 59
Protocol lock ...................................................... 60
I²C address change .............................................. 61
Factory reset ....................................................... 62
Change to UART mode ......................................... 63
Manual switching to UART ................................. 64

Circuit footprint ................................................... 65
Datasheet change log .......................................... 66
Warranty ........................................................... 68

Atlas Scientific
Environmental Robotics
**EZOTM circuit dimensions**

**Power consumption**

<table>
<thead>
<tr>
<th>5V</th>
<th>LED</th>
<th>MAX</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>16 mA</td>
<td>15.4 mA</td>
<td>0.4 mA</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>15.3 mA</td>
<td>15 mA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3.3V</th>
<th>LED</th>
<th>MAX</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>14.3 mA</td>
<td>13.8 mA</td>
<td>0.09 mA</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>14 mA</td>
<td>13.6 mA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Absolute max ratings**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage temperature (EZOTM RTD)</td>
<td>-65 °C</td>
<td></td>
<td>125 °C</td>
</tr>
<tr>
<td>Operational temperature (EZOTM RTD)</td>
<td>-40 °C</td>
<td>25 °C</td>
<td>85 °C</td>
</tr>
<tr>
<td>VCC</td>
<td>3.3V</td>
<td>5V</td>
<td>5.5V</td>
</tr>
</tbody>
</table>
EZO™ RTD temperature circuit range

EZO™ RTD temperature circuit accuracy
To read temperatures above, or below the max cable temperature, an additional probe housing (thermowell) is needed to protect the cable.

**Atlas Scientific PT-1000 probe**

- Accuracy +/- (0.15 + (0.002*t))
- Probe type: class A platinum, RTD
- Cable length: 81cm (32”)
- Cable material: silicone rubber
- 30mm sensing area (304 SS)
- 6mm diameter
- BNC connector
- Reaction time: 90% value in 13 seconds
- Probe output: analog
- Full sensing range -200 °C to 850 °C
- Cable max temp 125 °C
- Cable min temp -55 °C

The Atlas Scientific EZO™ RTD Temperature circuit only works with PT-100 and PT-1000 probes.

---

100mm Temperature Thermowell
50mm Temperature Thermowell
30mm Temperature Thermowell
Using other brand PT-100/PT-1000

The EZO™ RTD Temperature circuit will auto-detect if the connected probe is PT-100 or PT-1000.

<table>
<thead>
<tr>
<th>Probe class</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>±(0.10°C + 0.0017 x T)</td>
</tr>
<tr>
<td>A</td>
<td>±(0.15°C + 0.002 x T)</td>
</tr>
<tr>
<td>B</td>
<td>±(0.3°C + 0.005 x T)</td>
</tr>
<tr>
<td>C</td>
<td>±(0.6°C + 0.01 x T)</td>
</tr>
</tbody>
</table>

It makes no difference which lead of the temperature probe is connected to the two probe pins.

BOTH ARE CORRECT
Operating principle

The Atlas Scientific EZO™ RTD Temperature circuit is a small footprint computer system that is specifically designed to be used in robotic applications where the embedded systems engineer requires accurate and precise measurements of temperature through a generic PT-100/PT-1000 temperature probe.

RTD = Resistance Temperature Detector
PT = Platinum
PT-100 = 100 Ω at 0°C
PT-1000 = 1k Ω at 0°C

Unlike any other material, platinums correlation between resistance and temperature seems to be woven into the fabric of the universe. It is for this reason, that the platinum RTD temperature sensor is the industrial standard for temperature measurement.
Calibration theory

Calibration can be done at any value, a simple method is to calibrate the probe in boiling water.

100 °C

Atlas Scientific recommends calibration be done every three years.

Elevation and Boiling Point table

<table>
<thead>
<tr>
<th>Elevation in meters</th>
<th>Boiling point</th>
</tr>
</thead>
<tbody>
<tr>
<td>305</td>
<td>98.9 °C</td>
</tr>
<tr>
<td>229</td>
<td>99.2 °C</td>
</tr>
<tr>
<td>152</td>
<td>99.5 °C</td>
</tr>
<tr>
<td>76</td>
<td>99.7 °C</td>
</tr>
<tr>
<td>0</td>
<td>100 °C</td>
</tr>
<tr>
<td>-76</td>
<td>100.3 °C</td>
</tr>
<tr>
<td>-152</td>
<td>100.5 °C</td>
</tr>
</tbody>
</table>

Use purified/distilled water

For accurate calibration using different temperature values, you must use a tool called a "dry block calibrator."
On board data logger

- 50 readings
- Programmable storage interval

Minimum – 10 seconds
Maximum – 320,000 seconds

Temperature readings that are stored to the data logger will be retained even if the power is cut.

When memory becomes full, the data logger will overwrite memory location 1.
Power and data isolation

ELECTRICAL ISOLATION IS NOT NEEDED.
Correct wiring

Bread board

Extended leads

Sloppy setup

Perfboards or Protoboards

*Embedded into your device

Incorrect wiring

Extended leads

Sloppy setup

Perfboards or Protoboards

*Embedded into your device

*Only after you are familiar with EZO™ circuits operation
Available data protocols

UART

I²C

Unavailable data protocols

SPI
Analog
RS-485
Mod Bus
4–20mA
**UART mode**

<table>
<thead>
<tr>
<th>Settings that are retained if power is cut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud rate</td>
</tr>
<tr>
<td>Calibration</td>
</tr>
<tr>
<td>Continuous mode</td>
</tr>
<tr>
<td>Device name</td>
</tr>
<tr>
<td>Enable/disable response codes</td>
</tr>
<tr>
<td>Hardware switch to I²C mode</td>
</tr>
<tr>
<td>LED control</td>
</tr>
<tr>
<td>Protocol lock</td>
</tr>
<tr>
<td>Software switch to I²C mode</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Settings that are <strong>NOT</strong> retained if power is cut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find</td>
</tr>
<tr>
<td>Sleep mode</td>
</tr>
</tbody>
</table>
**UART mode**

<table>
<thead>
<tr>
<th>Baud</th>
<th>300</th>
<th>1,200</th>
<th>2,400</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>9,600 default</strong></td>
<td>19,200</td>
<td>38,400</td>
</tr>
<tr>
<td></td>
<td>57,600</td>
<td>115,200</td>
<td></td>
</tr>
</tbody>
</table>

**RX**
- Data in

**TX**
- Data out

**Vcc**
- 3.3V – 5.5V

**Data format**

<table>
<thead>
<tr>
<th>Reading</th>
<th>temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Units</td>
<td>°C, °K, or °F</td>
</tr>
<tr>
<td>Encoding</td>
<td>ASCII</td>
</tr>
<tr>
<td>Format</td>
<td>string</td>
</tr>
<tr>
<td>Terminator</td>
<td>carriage return</td>
</tr>
</tbody>
</table>

**Data type**: floating point

**Decimal places**: 3

**Smallest string**: 4 characters

**Largest string**: 399 characters
**Default state**

**Mode**  UART

**Baud**  9,600

**Temperature**  °C

**Readings**  continuous

**Speed**  1 reading per second

**With probe**  ttt.ttt

**Without probe**  -1023.000

**LED**  on
Receiving data from device

2 parts

ASCII data string

Command

Carriage return <cr>

Terminator

CPU

9,600 baud (default)

9,600 baud (default)

25.104 <cr>

Advanced

ASCII: 2 5 . 1 0 4 <cr>

Hex: 32 35 2E 31 30 34 OD

Dec: 50 53 46 49 48 52 13
Sending commands to device

2 parts

Command (not case sensitive)
ASCII data string

Carriage return <cr>
Terminator

Advanced
ASCII: Slleep<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

<table>
<thead>
<tr>
<th>Voltage</th>
<th>LED ON</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td></td>
<td>+0.4 mA</td>
</tr>
<tr>
<td>3.3V</td>
<td></td>
<td>+0.2 mA</td>
</tr>
</tbody>
</table>
# UART mode command quick reference

All commands are ASCII strings or single ASCII characters.

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Default state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud</td>
<td>change baud rate</td>
<td>pg. 36</td>
</tr>
<tr>
<td>C</td>
<td>enable/disable continuous reading</td>
<td>pg. 23</td>
</tr>
<tr>
<td>Cal</td>
<td>performs calibration</td>
<td>pg. 25</td>
</tr>
<tr>
<td>D</td>
<td>enable/disable data logger</td>
<td>pg. 28</td>
</tr>
<tr>
<td>Export/import</td>
<td>export/import calibration</td>
<td>pg. 26</td>
</tr>
<tr>
<td>Factory</td>
<td>enable factory reset</td>
<td>pg. 38</td>
</tr>
<tr>
<td>Find</td>
<td>finds device with blinking white LED</td>
<td>pg. 22</td>
</tr>
<tr>
<td>i</td>
<td>device information</td>
<td>pg. 32</td>
</tr>
<tr>
<td>I2C</td>
<td>change to I²C mode</td>
<td>pg. 39</td>
</tr>
<tr>
<td>L</td>
<td>enable/disable LED</td>
<td>pg. 21</td>
</tr>
<tr>
<td>M</td>
<td>memory recall/clear</td>
<td>pg. 29</td>
</tr>
<tr>
<td>Name</td>
<td>set/show name of device</td>
<td>pg. 31</td>
</tr>
<tr>
<td>Plock</td>
<td>enable/disable protocol lock</td>
<td>pg. 37</td>
</tr>
<tr>
<td>R</td>
<td>returns a single reading</td>
<td>pg. 24</td>
</tr>
<tr>
<td>S</td>
<td>temperature scale (°C, °K, °F)</td>
<td>pg. 27</td>
</tr>
<tr>
<td>Sleep</td>
<td>enter sleep mode/low power</td>
<td>pg. 35</td>
</tr>
<tr>
<td>Status</td>
<td>retrieve status information</td>
<td>pg. 34</td>
</tr>
<tr>
<td>*OK</td>
<td>enable/disable response codes</td>
<td>pg. 33</td>
</tr>
</tbody>
</table>

All commands are ASCII strings or single ASCII characters.
# LED control

## Command syntax

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>L,1 &lt;cr&gt;</td>
<td>LED on <strong>default</strong></td>
</tr>
<tr>
<td>L,0 &lt;cr&gt;</td>
<td>LED off</td>
</tr>
<tr>
<td>L,? &lt;cr&gt;</td>
<td>LED state on/off?</td>
</tr>
</tbody>
</table>

## Example | Response

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>L,1 &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>L,0 &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>
| L,? <cr> | ?L,1 <cr> or ?L,0 <cr>  
*OK <cr> |
Find

Command syntax

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

Example Response

Find <cr>  *OK <cr>

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

*This command is only available for firmware version 2.10 and above.

Copyright © Atlas Scientific LLC
# Continuous reading mode

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C,1</td>
<td>enable continuous readings once per second</td>
</tr>
<tr>
<td>C,n</td>
<td>continuous readings every n seconds (n = 2 to 99 sec)*</td>
</tr>
<tr>
<td>C,0</td>
<td>disable continuous readings</td>
</tr>
<tr>
<td>C,?</td>
<td>continuous reading mode on/off?</td>
</tr>
</tbody>
</table>

### Command syntax

- **C,1** <cr> enable continuous readings once per second (default)
- **C,n** <cr> continuous readings every n seconds (n = 2 to 99 sec)*
- **C,0** <cr> disable continuous readings
- **C,?** <cr> continuous reading mode on/off?

### Example Response

**Example** | **Response**
---|---
C,1 <cr> | *OK <cr> °C (1 sec) <cr> °C (2 sec) <cr> °C (n sec) <cr>
C,30 <cr> | *OK <cr> °C (30 sec) <cr> °C (60 sec) <cr> °C (90 sec) <cr>
C,0 <cr> | *OK <cr>
C,? <cr> | ?C,1 <cr> or ?C,0 <cr> or ?C,30 <cr> *OK <cr>

*This command is only available for firmware version 2.10 and above.*
# Single reading mode

## Command syntax

\[ R \text{ <cr>} \] takes single reading

## Example

<table>
<thead>
<tr>
<th>R &lt;cr&gt;</th>
<th>25.104 &lt;cr&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>*OK &lt;cr&gt;</td>
<td></td>
</tr>
</tbody>
</table>

## Example Response

Green
- Standby

Cyan
- Taking reading

Transmitting

![Image showing states: Green (Standby), Cyan (Taking reading), Transmitting, with a 600 ms timer]
The EZO™ RTD circuit uses single point calibration.

**Command syntax**

- **Cal,t <cr>**  \( t = \) any temperature
- **Cal,clear <cr>** delete calibration data
- **Cal,? <cr>** device calibrated?

**Example**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal,100.00 &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Cal,clear &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Cal,? &lt;cr&gt;</td>
<td>?Cal,1 &lt;cr&gt; or ?Cal,0 &lt;cr&gt;</td>
</tr>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

**Example Response**

- ?Cal,1 <cr> or ?Cal,0 <cr>
- *OK <cr>

**Graph**

- **Cal,100.00 <cr>**
- **100.35 °C**
- **100.00 °C**
# Export/import calibration

## Command syntax

**Export**: Use this command to save calibration settings

**Import**: Use this command to load calibration settings to one or more devices.

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

*This command is only available for firmware version 2.10 and above.*

## Example

### Example

- **Export,? `<cr>`**
  - **Response**: `10,120 `<cr>``
    - **Response breakdown**
      - **10, 120**
        - # of strings to export
        - # of bytes to export
    - Export strings can be up to 12 characters long, and is always followed by `<cr>`

- **Export `<cr>`**
  - **Response**: `59 6F 75 20 61 72 `<cr>` (1 of 10)
  - **Export `<cr>`**
    - **Response**: `65 20 61 20 63 6F `<cr>` (2 of 10)
    - (7 more)
  - **Export `<cr>`**
    - **Response**: `6F 6C 20 67 75 79 `<cr>` (10 of 10)
  - **Export `<cr>`**
    - **Response**: *DONE*

- **Import, n**
  - **Response**: `Import, 59 6F 75 20 61 72 `<cr>` (1 of 10)

*Disabling *OK simplifies this process*
## Temperature scale (°C, °K, °F)

### Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S,c</td>
<td>celsius (default)</td>
</tr>
<tr>
<td>S,k</td>
<td>kelvin</td>
</tr>
<tr>
<td>S,f</td>
<td>fahrenheit</td>
</tr>
<tr>
<td>S,?</td>
<td>temperature scale?</td>
</tr>
</tbody>
</table>

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>S,c</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>S,k</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>S,f</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>S,?</td>
<td>?S,c  &lt;cr&gt; or ?S,k  &lt;cr&gt; or ?S,f  &lt;cr&gt; *OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

### Diagrams

- **(celsius)**
  - -126 °C to 1,254 °C
- **(kelvin)**
  - 147.15 °K to 1,527.15 °K
- **(fahrenheit)**
  - -194.8 °F to 2,289.2 °F
Enable/disable data logger

Command syntax

The time period \(n\) is in 10 second intervals and can be any value from 1 to 32,000.

\[
D,n \ <\text{cr} > \ n = (n \times 10 \text{ seconds})
\]

\[
D,0 \ <\text{cr} > \ \text{disable default}
\]

\[
D,? \ <\text{cr} > \ \text{data logger storage interval?}
\]

Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>D,6 &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>D,0 &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>D,? &lt;cr&gt;</td>
<td>?D,6 &lt;cr&gt; *OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

* indicates reading has been logged
# Memory recall

## Command syntax

Disable data logger to recall memory.

- **M** <cr> recall 1 sequential stored reading
- **M,all** <cr> recall all readings in a CSV string
- **M,?** <cr> display memory location of last stored reading

## Example

<table>
<thead>
<tr>
<th><strong>Example</strong></th>
<th><strong>Response</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M</strong> &lt;cr&gt;</td>
<td>1,100.00 &lt;cr&gt; 2,104.00 &lt;cr&gt; *OK &lt;cr&gt;</td>
</tr>
<tr>
<td><strong>M,all</strong> &lt;cr&gt;</td>
<td>100.00,104.00,108.00,112.00 &lt;cr&gt; Oldest Newest</td>
</tr>
<tr>
<td><strong>M,?</strong> &lt;cr&gt;</td>
<td>?M,4&lt;cr&gt; *OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>
Memory clear

Command syntax

M,clear <cr> clear all stored memory

Example Response

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>M,clear &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>
# Naming device

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set name</td>
<td>Name,n &lt;cr&gt;</td>
<td>n = 1-16 ASCII characters</td>
</tr>
<tr>
<td>Show name</td>
<td>Name,? &lt;cr&gt;</td>
<td></td>
</tr>
</tbody>
</table>

## Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name,zzt &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Name,? &lt;cr&gt;</td>
<td>?Name,zzt &lt;cr&gt;</td>
</tr>
<tr>
<td>Name,zzt</td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

## Example Response

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name,zzt</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td>Name,zzt</td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>
# Device information

## Command syntax

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

## Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
</table>
| i <cr>  | ?i,RTD,2.01 <cr>
|         | *OK <cr> |

## Response breakdown

`?i, RTD, 2.01`  
1. **Device**: RTD  
2. **Firmware**: 2.01
Response codes

Command syntax

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

Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
</table>
| R <cr>  | 25.104 <cr>  
|        | *OK <cr> |
| *OK,0 <cr> | no response, *OK disabled |
| R <cr> | 25.104 <cr>  
|        | *OK disabled |
| *OK,? <cr> | ?*OK,1 <cr> or ?*OK,0 <cr> |

Other response codes

*ER unknown command
*OV over volt (VCC>=5.5V)
*UV under volt (VCC<=3.1V)
*RS reset
*RE boot up complete, ready
*SL entering sleep mode
*WA wake up

These response codes cannot be disabled
Reading device status

Command syntax

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

Example

<table>
<thead>
<tr>
<th>Status &lt;cr&gt;</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>?Status,P,5.038 &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

Response breakdown

?Status, P, 5.038

↑ Reason for restart

↑ Voltage at Vcc

Restart codes

<table>
<thead>
<tr>
<th>P</th>
<th>powered off</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>software reset</td>
</tr>
<tr>
<td>B</td>
<td>brown out</td>
</tr>
<tr>
<td>W</td>
<td>watchdog</td>
</tr>
<tr>
<td>U</td>
<td>unknown</td>
</tr>
</tbody>
</table>
Sleep mode/low power

Command syntax

Sleep <cr> enter sleep mode/low power

Example | Response
--- | ---
Sleep <cr> | *SL
Any command | *WA <cr> wakes up device

<table>
<thead>
<tr>
<th>Voltage</th>
<th>STANDBY</th>
<th>SLEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>15.40 mA</td>
<td>0.4 mA</td>
</tr>
<tr>
<td>3.3V</td>
<td>13.80 mA</td>
<td>0.09 mA</td>
</tr>
</tbody>
</table>

Sleep <cr>

Standby
15.40 mA

Sleep
3.00 mA
Change baud rate

Command syntax

Baud, n <cr> change baud rate

Example

<table>
<thead>
<tr>
<th>Baud, 38400 &lt;cr&gt;</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Baud, ? &lt;cr&gt;</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>?Baud, 38400 &lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
</tbody>
</table>

n =

300
1200
2400
9600 default
19200
38400
57600
115200

---

CPU
TX RX
RX TX
TX RX

Green
UART standby
Purple
Changing
baud rate
Red
Command
not understood
White
Find
Cyan
Taking reading

*OK <cr>
(reboot)

Standby

Baud, 38400 <cr>

Changing baud rate

*OK <cr>

Standby

---
Protocol lock

Command syntax

- **Plock,1** <cr> enable Plock
- **Plock,0** <cr> disable Plock \(\text{default}\)
- **Plock,?** <cr> Plock on/off?

Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Plock,1</strong>&lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td><strong>Plock,0</strong>&lt;cr&gt;</td>
<td>*OK &lt;cr&gt;</td>
</tr>
<tr>
<td><strong>Plock,?</strong>&lt;cr&gt;</td>
<td>?Plock,1 &lt;cr&gt; or ?Plock,0 &lt;cr&gt;</td>
</tr>
</tbody>
</table>

Example Response

- **Plock,1** <cr> *OK <cr>
- **I2C,100** <cr> cannot change to I²C
  *ER <cr>
- **Short** cannot change to I²C
## Factory reset

### Command syntax

**Factory cr**  enable factory reset

### Example

<table>
<thead>
<tr>
<th>Factory</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory</td>
<td>*OK cr</td>
</tr>
</tbody>
</table>

### Response

- Factory <cr>
- *(reboot) *
- *OK cr*
- *RS cr*
- *RE cr*

- **Baud rate will not change**
# Change to I²C mode

## Command syntax

$I2C,n$  
sets I²C address and reboots into I²C mode  
$n = \text{any number } 1 \text{ – } 127$

## Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I2C,100$</td>
<td>*OK (reboot in I²C mode)</td>
</tr>
</tbody>
</table>

## Wrong example

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I2C,139$</td>
<td>127</td>
</tr>
<tr>
<td>$n \neq 127$</td>
<td>*ER &lt;cr&gt;</td>
</tr>
</tbody>
</table>

## I²C,100

![I²C mode example](image)

**Green**  
*OK <cr>*

(reboot)

**Blue**  
now in I²C mode
Manual switching to I²C

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

Manually switching to I²C will set the I²C address to 102 (0x66)

Example

Wrong Example: Disconnect RX line

Connecting TX to PRB only works for the EZO™ RTD Temperature circuit.
I²C mode

The I²C protocol is *considerably more complex* than the UART (RS–232) protocol. Atlas Scientific assumes the embedded systems engineer understands this protocol.

To set your EZO™ device into I²C mode click here

<table>
<thead>
<tr>
<th>Settings that are retained if power is cut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration</td>
</tr>
<tr>
<td>Change I²C address</td>
</tr>
<tr>
<td>Hardware switch to UART mode</td>
</tr>
<tr>
<td>LED control</td>
</tr>
<tr>
<td>Protocol lock</td>
</tr>
<tr>
<td>Software switch to UART mode</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Settings that are <strong>NOT</strong> retained if power is cut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find</td>
</tr>
<tr>
<td>Sleep mode</td>
</tr>
</tbody>
</table>
I²C mode

I²C address  
(0x01 – 0x7F)  
102 (0x66) default

Vcc  
3.3V – 5.5V

Clock speed  
100 – 400 kHz

SDA  
SCL

Data format

Reading  temperature
Units  °C, °K, or °F
Encoding  ASCII
Format  string

Data type  floating point
Decimal places  3
Smallest string  4 characters
Largest string  14 characters

4.7k resistor may be needed
Sending commands to device

5 parts

Start | I²C address | Write | Command (not case sensitive) | Stop

120 (0x66) | ASCII command string

Example

Start | 102 (0x66) | Write | Sleep | Stop

I²C address | Command

Advanced

Address bits | The entire command as ASCII with all arguments

Start | A0 | A1 | A2 | A3 | A4 | A5 | A6 | W | ACK | First letter of command | ACK | … | … | … | … | Last letter of command | ACK | Stop

W = low

4.7k resistor may be needed

0V 0V

VCC

CPU

(SDAD (TX) (RX)

SCL

GND

SDA

SCL

SCL

VCC

SDA

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)

SCL

CPU

SCL

GND

SDA

SCL

VCC

SDA

SCL

VCC

SDA

(SDAD (TX) (RX)
Requesting data from device

7 parts

Start | I2C address | Read | Response code | Data string | Null | Stop

102 (0x66) | 1 byte | "25.104" | Terminator (Dec 0)

Advanced

CPU

25.104

Address bits

N bytes of data

All bytes after data are Null

Start

ACK

Response code

Data

ACK

Null

ACK

NACK

Stop

RTD

VCC

SDA

R

SCL

A6 \rightarrow A0

R

Response code

Data

ACK

Null

ACK

NACK

Stop

1 50 53 46 49 48 52 0 = 25.104

Dec

ASCII

Dec

50 53 46 49 48 52 0 = 25.104

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

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

delay(300);

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.

<table>
<thead>
<tr>
<th>Response code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>no data to send</td>
</tr>
<tr>
<td>254</td>
<td>still processing, not ready</td>
</tr>
<tr>
<td>2</td>
<td>syntax error</td>
</tr>
<tr>
<td>1</td>
<td>successful request</td>
</tr>
</tbody>
</table>
LED color definition

Blue
I²C standby

Green
Taking reading

Purple
Changing I²C ID#

Red
Command not understood

White
Find

<table>
<thead>
<tr>
<th>Voltage</th>
<th>LED ON</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>+0.4 mA</td>
<td></td>
</tr>
<tr>
<td>3.3V</td>
<td>+0.2 mA</td>
<td></td>
</tr>
</tbody>
</table>

Copyright © Atlas Scientific LLC
# I²C mode

command quick reference

All commands are ASCII strings or single ASCII characters.

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud</td>
<td>switch back to UART mode</td>
<td>63</td>
</tr>
<tr>
<td>Cal</td>
<td>performs calibration</td>
<td>51</td>
</tr>
<tr>
<td>D</td>
<td>enable/disable data logger</td>
<td>54</td>
</tr>
<tr>
<td>Export/import</td>
<td>export/import calibration</td>
<td>52</td>
</tr>
<tr>
<td>Factory</td>
<td>enable factory reset</td>
<td>62</td>
</tr>
<tr>
<td>Find</td>
<td>finds devices with white blinking LED</td>
<td>49</td>
</tr>
<tr>
<td>i</td>
<td>device information</td>
<td>57</td>
</tr>
<tr>
<td>I2C</td>
<td>change I²C address</td>
<td>61</td>
</tr>
<tr>
<td>L</td>
<td>enable/disable LED</td>
<td>48</td>
</tr>
<tr>
<td>M</td>
<td>memory recall/clear</td>
<td>55</td>
</tr>
<tr>
<td>Plock</td>
<td>enable/disable protocol lock</td>
<td>60</td>
</tr>
<tr>
<td>R</td>
<td>returns a single reading</td>
<td>50</td>
</tr>
<tr>
<td>S</td>
<td>temperature scale (°C, °K, °F)</td>
<td>53</td>
</tr>
<tr>
<td>Sleep</td>
<td>enter sleep mode/low power</td>
<td>59</td>
</tr>
<tr>
<td>Status</td>
<td>retrieve status information</td>
<td>58</td>
</tr>
</tbody>
</table>
# LED control

## Command syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L,1</td>
<td>LED on (default)</td>
</tr>
<tr>
<td>L,0</td>
<td>LED off</td>
</tr>
<tr>
<td>L,?</td>
<td>LED state on/off?</td>
</tr>
</tbody>
</table>

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>L,1</td>
<td><img src="image1" alt="Wait 300ms" /> Dec 1 ASCII 0 Null</td>
</tr>
<tr>
<td>L,0</td>
<td><img src="image2" alt="Wait 300ms" /> Dec 1 ASCII 0 Null</td>
</tr>
<tr>
<td>L,?</td>
<td><img src="image3" alt="Wait 300ms" /> Dec 1 ASCII 0 or Dec 1 ASCII 0 Null</td>
</tr>
</tbody>
</table>

300ms processing delay
Find

LED rapidly blinks white, used to help find device*

Command syntax

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

Example Response

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

*This command is only available for firmware version 2.10 and above.
Taking reading

Command syntax

R  return 1 reading

Example  Response

R

Wait 600ms

Green
Taking reading

Transmitting

Blue
Standby

600ms processing delay

R 0.1

Copyright © Atlas Scientific LLC
Calibration

Command syntax

- **Cal,t**  \( t = \text{any temperature} \)
- **Cal,clear**  delete calibration data
- **Cal,?**  device calibrated?

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
</table>
| **Cal,t** | ![Wait 600ms](600ms)
| | ![Dec](Dec) 0 ![Null](Null) |
| **Cal,clear** | ![Wait 300ms](Wait 300ms)
| | ![Dec](Dec) 0 ![Null](Null) |
| **Cal,?** | ![Wait 300ms](Wait 300ms)
| | ![Dec](Dec) ![ASCI](ASCI) ![Null](Null) or ![Dec](Dec) ![ASCI](ASCI) ![Null](Null) |

**Example Response**

- **600ms** processing delay
- **EZO™** RTD circuit uses single point calibration.
## Export/import calibration

### Command syntax

**Export**
- export calibration string from calibrated device*

**Import**
- import calibration string to new device*

**Export,**
- calibration string info*

---

### 300ms processing delay

---

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export,?</td>
<td></td>
</tr>
</tbody>
</table>

- **Wait 300ms**

  - 1 10,120 0

  - Dec  ASCII  Null

  **Response breakdown**
  - 10, 120
  - # of strings to export  # of bytes to export

  *Export strings can be up to 12 characters long*

---

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export</td>
<td></td>
</tr>
</tbody>
</table>

- **Wait 300ms**

  - 1 59 6F 75 20 61 72 0

  - Dec  ASCII  Null

  (1 of 10)

  (8 more)

---

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export</td>
<td></td>
</tr>
</tbody>
</table>

- **Wait 300ms**

  - 1 65 20 61 20 63 6F 0

  - Dec  ASCII  Null

  (10 of 10)

---

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export</td>
<td></td>
</tr>
</tbody>
</table>

- **Wait 300ms**

  - 1 *DONE* 0

  - Dec  ASCII  Null

---

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import, n</td>
<td></td>
</tr>
</tbody>
</table>

- **FIFO**

  - Import, 59 6F 75 20 61 72

  - ASCII

  (1 of 10)
## Temperature scale (°C, °K, °F)

### Command syntax

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Celsius</th>
<th>Kelvin</th>
<th>Fahrenheit</th>
<th>Temperature Scale?</th>
</tr>
</thead>
<tbody>
<tr>
<td>S,c</td>
<td>default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S,k</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S,f</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S,?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Example

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>S,c</td>
<td><img src="clock" alt="Wait 300ms" /> 0m 0s Dec 0 Null</td>
</tr>
<tr>
<td>S,k</td>
<td><img src="clock" alt="Wait 300ms" /> 0m 0s Dec 0 Null</td>
</tr>
<tr>
<td>S,f</td>
<td><img src="clock" alt="Wait 300ms" /> 0m 0s Dec 0 Null</td>
</tr>
<tr>
<td>S,?</td>
<td><img src="clock" alt="Wait 300ms" /> 0m 0s Dec ASCII Null</td>
</tr>
</tbody>
</table>

### Example Response

- 300ms processing delay
- Wait 300ms
- 1 Dec 0 Null
- 1 Dec 0 Null
- 1 Dec 0 Null
- 1 Dec ASCII Null
- 1 Dec ASCII Null
- 1 Dec ASCII Null

### Temperature Range

- -126 °C to 1,254 °C (Celsius)
- 147.15 °K to 1,527.15 °K (Kelvin)
- -194.8 °F to 2,289.2 °F (Fahrenheit)
Enable/disable data logger

**Command syntax**

- **D,n**  
  \( n = (n \times 10 \text{ seconds}) \)
- **D,0**  
  disable
- **D,?**  
  data logger storage interval?

**Example**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
</table>
| **D,6** | ![Wait 300ms](image)  
  1 Dec 0 Null |
| **D,0** | ![Wait 300ms](image)  
  1 Dec 0 Null |
| **D,?** | ![Wait 300ms](image)  
  1 ?D,6 0 Dec ASCII Null |

**Example Response**

- **Wait 300ms**
- **Dec**  
  0 Null
- **Dec**  
  0 Null
- **Dec**  
  ASCII Null

**D,6 (after 60 seconds)**

300ms processing delay

The time period \( n \) is in 10 second intervals and can be any value from 1 to 32,000.
Memory recall

Command syntax

- **M** recall 1 sequential stored reading
- **M,?** display memory location of last stored reading

Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>![Clock] 1 1,100.00 0</td>
</tr>
<tr>
<td>M,?</td>
<td>![Clock] 1 4,112.00 0</td>
</tr>
</tbody>
</table>

300ms processing delay

55 Copyright © Atlas Scientific LLC
## Memory clear

### Command syntax

**M,clear**  clear all stored memory

### Example

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>M,clear</td>
<td><img src="image" alt="Timer" /> 1 Dec 0 Null</td>
</tr>
</tbody>
</table>

300ms processing delay

---

**SDA** (TX) (RX) **SCL**

---

**SDA** (TX) (RX) **SCL**

---

**SDA** (TX) (RX) **SCL**

---

**SDA** (TX) (RX) **SCL**

---

**SDA** (TX) (RX) **SCL**

---

**SDA** (TX) (RX) **SCL**

---

**SDA** (TX) (RX) **SCL**

---

**SDA** (TX) (RX) **SCL**

---

**SDA** (TX) (RX) **SCL**

---

**SDA** (TX) (RX) **SCL**

---

**SDA** (TX) (RX) **SCL**

---

**SDA** (TX) (RX) **SCL**
Device information

Command syntax

Example

Response breakdown

300ms processing delay

$i$, device information

Example

Response

$i$

Wait 300ms

$i$, RTD, 2.01

Dec ASCII Null

Response breakdown

$i$, RTD, 2.01

Device Firmware
## Reading device status

**Command syntax**

```
300ms processing delay
```

**Status**  Voltage at Vcc pin and reason for last restart

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Status</strong></td>
<td><img src="image" alt="Wait 300ms" /></td>
</tr>
</tbody>
</table>

**Response breakdown**

<table>
<thead>
<tr>
<th>?Status,</th>
<th>P,</th>
<th>5.038</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reason for restart</td>
<td>Voltage at Vcc</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Restart codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>P powered off</td>
</tr>
<tr>
<td>S software reset</td>
</tr>
<tr>
<td>B brown out</td>
</tr>
<tr>
<td>W watchdog</td>
</tr>
<tr>
<td>U unknown</td>
</tr>
</tbody>
</table>

---

Copyright © Atlas Scientific LLC
# Sleep mode/low power

## Command syntax

**Sleep**  enter sleep mode/low power

- **Send any character or command to awaken device.**

## Example & Response

<table>
<thead>
<tr>
<th>Example</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep</td>
<td>no response</td>
</tr>
<tr>
<td>Any command</td>
<td>wakes up device</td>
</tr>
</tbody>
</table>

## Power Consumption

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Standby Current</th>
<th>Sleep Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>15.40 mA</td>
<td>0.4 mA</td>
</tr>
<tr>
<td>3.3V</td>
<td>13.80 mA</td>
<td>0.09 mA</td>
</tr>
</tbody>
</table>

---

[AtlasScientific](https://www.atlasscientific.com) Environmental Robotics

Copyright © Atlas Scientific LLC
# Protocol lock

## Command syntax

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plock, 1</td>
<td>enable Plock</td>
</tr>
<tr>
<td>Plock, 0</td>
<td>disable Plock</td>
</tr>
<tr>
<td>Plock, ?</td>
<td>Plock on/off?</td>
</tr>
</tbody>
</table>

**Plock,1**  
Enable Plock  
Locks device to I²C mode.  
Serial, 9600

**Plock,0**  
Disable Plock  
cannot change to UART

**Plock,?**  
Plock on/off?  
Wait 300ms

**300ms processing delay**

## Example

<table>
<thead>
<tr>
<th>Command Syntax</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plock, 1</td>
<td><img src="image" alt="Plock,1" /></td>
</tr>
<tr>
<td>Plock, 0</td>
<td><img src="image" alt="Plock,0" /></td>
</tr>
<tr>
<td>Plock, ?</td>
<td><img src="image" alt="Plock,?" /></td>
</tr>
</tbody>
</table>

**Example Response**

<table>
<thead>
<tr>
<th>Plock, 1</th>
<th>1</th>
<th>Dec</th>
<th>0</th>
<th>ASCII</th>
<th>Null</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plock, 0</td>
<td>1</td>
<td>Dec</td>
<td>0</td>
<td>ASCII</td>
<td>Null</td>
</tr>
<tr>
<td>Plock, ?</td>
<td>1</td>
<td>Dec</td>
<td>?</td>
<td>ASCII</td>
<td>0</td>
</tr>
</tbody>
</table>

cannot change to UART  
Serial, 9600  
AtlasScientific  
Copyright © Atlas Scientific LLC
**I²C address change**

**Command syntax**

I²C, n  sets I²C address and reboots into I²C mode

**Example** | **Response**
--- | ---
I²C, 100 | device reboot

**Warning!**

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

Default I²C address is 102 (0x66).

**n = any number 1 – 127**

I²C, 100

(reboot)
Factory reset

Command syntax

**Factory enable factory reset**

I²C address will not change

**Factory device reboot**

**Example**

**Response**

| Factory | device reboot |

Clears calibration
LED on
Response codes enabled
Clears data logger

(reboot)
# Change to UART mode

## Command syntax

Baud,\(n\) switch from I\(^2\)C to UART

## Example

<table>
<thead>
<tr>
<th>Baud</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>9600</td>
<td>reboot in UART mode</td>
</tr>
</tbody>
</table>

### Baud Options

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

**Diagram:**

- Initial state: Serial, 9600
- Transition: Changing to UART mode
- Final state: (reboot)
Manual switching to UART

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

Connecting TX to PRB only works for the EZO™ RTD Temperature circuit.

Wrong Example
1. In your CAD software, place an 8 position header.

2. Place a 3 position header at both top and bottom of the 8 position.

3. Delete the 8 position header. The two 3 position headers are now 17.78mm (0.7”) apart from each other.
Datasheet change log

Datasheet V 2.7
Revised definition of response codes on pg 45.

Datasheet V 2.6
Updated calibration processing delay time on pg.51.

Datasheet V 2.5
Revised Plock pages to show default value.

Datasheet V 2.4
**Added new commands:**
*Find* pages 22 & 49.
*Export/Import calibration* pages 26 & 52.
Added new feature to continuous mode "C,n" pg 23.

Datasheet V 2.3
Added manual switching to UART information on pg. 59.

Datasheet V 2.2
Revised Baud command information on pg. 33.

Datasheet V 2.1
Revised entire datasheet.
<table>
<thead>
<tr>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.02 –  Plock (March 31, 2016)</td>
<td>• Added protocol lock feature “Plock”</td>
</tr>
<tr>
<td>V1.03 –  EEPROM (April 26, 2016)</td>
<td>• Fixed glitch where EEPROM would get erased if the circuit lost power 900ms into startup</td>
</tr>
<tr>
<td>V1.11 –  Glitch Fix (June 9, 2016)</td>
<td>• Fixed glitch where a blank name would result in garbage output</td>
</tr>
<tr>
<td>V2.01 –  Update (January 1, 2017)</td>
<td>• Replaced command “response” with “*OK”</td>
</tr>
<tr>
<td></td>
<td>• Replaced command “Serial” with “Baud”</td>
</tr>
<tr>
<td>V2.02 –  Glitch Fix (February 16, 2017)</td>
<td>• Fixed glitch where calibration would not accept floating point numbers.</td>
</tr>
<tr>
<td>V2.10 –  (May 9, 2017)</td>
<td>• Added &quot;Find&quot; command.</td>
</tr>
<tr>
<td></td>
<td>• Added &quot;Export/import&quot; command.</td>
</tr>
<tr>
<td></td>
<td>• Modified continuous mode to be able to send readings every &quot;n&quot; seconds.</td>
</tr>
<tr>
<td></td>
<td>• Sleep current is lowered.</td>
</tr>
</tbody>
</table>
Warranty

Atlas Scientific™ Warrants the EZO™ class RTD circuit to be free of defect during the debugging phase of device implementation, or 30 days after receiving the EZO™ class RTD circuit (which ever comes first).

The debugging phase

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

- Soldering any part of the EZO™ class RTD circuit.
- Running any code, that does not exclusively drive the EZO™ class RTD circuit and output its data in a serial string.
- Embedding the EZO™ class RTD circuit 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 em-
bedded into a custom made system, Atlas Scientific™ cannot possibly warranty the EZO™
class RTD circuit, against the thousands of possible variables that may cause the EZO™ class
RTD circuit 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™ class RTD circuits
continued operation. This is because that would be equivalent to Atlas Scientific™ taking
responsibility over the correct operation of your entire device.