



GSS Sensor User Guide

This documentation is provided on an as-is basis and no warranty as to its suitability or accuracy for any particular purpose is either made or implied. Gas Sensing Solutions Ltd will not accept any claim for damages howsoever arising as a result of use or failure of this information. Your statutory rights are not affected. This information is not intended for use in any medical appliance, device or system in which the failure of the might product reasonably be expected to result in personal injury. As GSS is committed to continuous improvement, this document provides information that may be subject to change without notice.

Contents

1	Serial Format and Connection	4
1.1	Connection	4
1.2	Serial Connection	5
1.3	Reading Format	6
1.3.1	CO ₂ Measurement	6
1.3.2	Temperature Measurement (Factory Fit option on COZIR-A and ExplorIR-W sensors only)	7
1.3.3	Humidity Measurement (Option on CozIR-A and ExplorIR-W sensors only.)	7
1.4	Example of T,H and CO ₂	8
2	Command Summary	9
3	Operating Modes	10
3.1	Mode 0 Command Mode	111
3.2	Mode 1 Streaming Mode	11
3.3	Mode 2 Polling Mode	11
4	Output Fields	11
5	Zero Point Calibration	12
5.1	Zero in a known gas concentration (recommended)	13
5.2	Zero in Nitrogen	13
5.3	Zero in Fresh Air (assumed to be 400ppm)	13
5.4	Fine Tune the Zero Point	13
5.5	Zero Point Adjustment	14
6	AutoCalibration	14
6.1	Principle of Operation	14
6.2	Requirements for Auto-calibration	15
6.3	Setting the Auto Calibration Parameters	15
6.4	Auto-Calibration Intervals	15
6.5	Background Concentration	15
6.6	Initial Auto-calibration Interval	15
6.7	Autocalibration Intervals	15
6.8	Read the Autocalibration Settings	16
6.9	Disable Autocalibration	16
6.10	Background Concentration	16
6.11	Customising Autocalibration	17
6.12	Threshold	17
6.13	Proportional	17
6.14	Spread Implementation	17
6.15	Register Values for Customising Autocalibration.	18
7	Altitude Compensation	19
8	User Settings	21
8.1	Digital Filter	21
8.1.1	Customising the Sensor Response	21
8.1.2	Setting the Digital Filter	22
8.1.3	Reading the Digital Filter Setting	22
8.2	User Options – EEPROM Settings	22
8.2.1	Setting EEPROM	22
8.2.2	Reading EEPROM	23

8.2.3	EEPROM Settings.....	23
8.2.4	Autocalibration Settings (locations 3-7).....	24
8.2.5	User EEPROM.....	24
9	Command Reference.....	24
9.1	Customisation.....	25
9.2	Information	26
9.3	Switching between Modes	26
9.4	Zeroing and Calibration	26
9.5	Polling Commands	28

Conventions

In this guide:

`\r\n`

Is used to indicate carriage return <CR>, line feed <LF> characters, (0x0d,0x0a) which are required at the end of each string sent to the sensor, and are appended to all transmissions from the sensor.

`Z 12345\r\n`

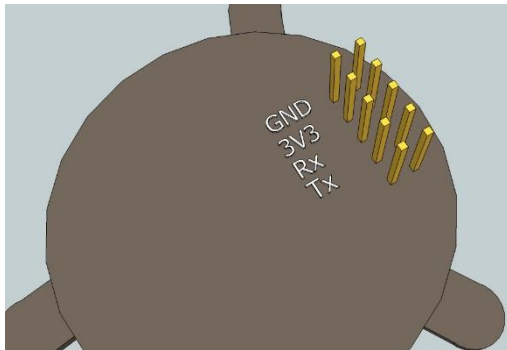
Courier fixed pitch font is used to show commands sent to the sensor, and received from the sensor

This documentation is provided on an as-is basis and no warranty as to its suitability or accuracy for any particular purpose is either made or implied. Gas Sensing Solutions Ltd will not accept any claim for damages howsoever arising as a result of use or failure of this information. Your statutory rights are not affected. This information is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury. As GSS is committed to continuous improvement, this document provides information that may be subject to change without notice.

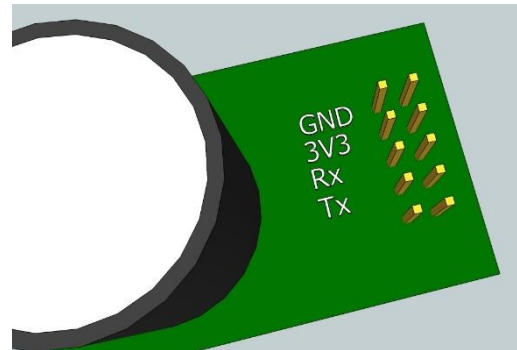
1 Serial Format and Connection

1.1 Connection

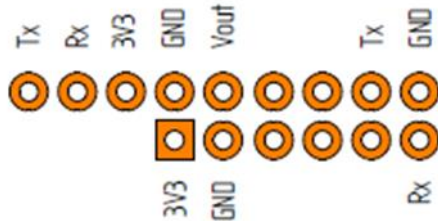
Communication to and from the GSS CO₂ sensor is via a serial connection. Pins are shown looking at the connector of the sensor and full details of pinouts are given in the datasheet for each sensor.



CozIR®-A

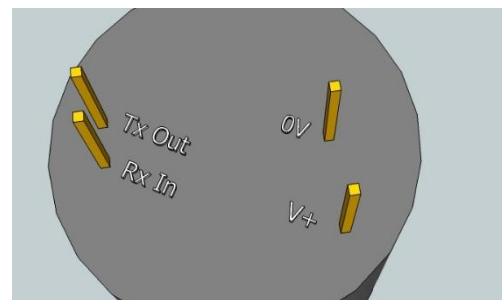


ExplorIR®-W

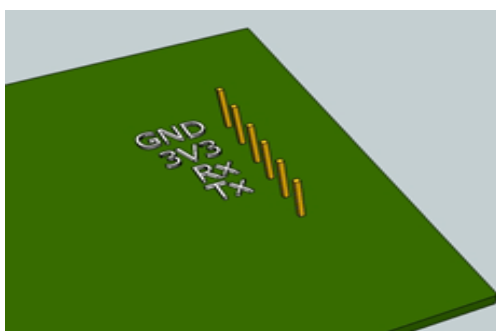


MISIR

Only one GND connection is required.

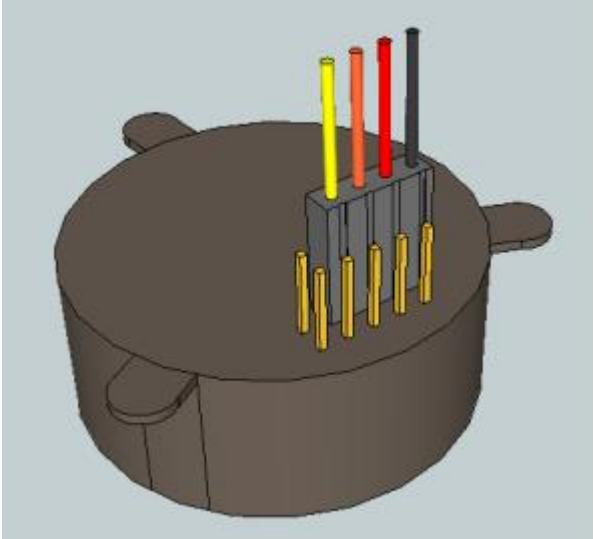


ExplorIR®-M



CozIR®-LP

If using the GSS USB serial cable and an interface board to connect to the sensor then note that the black cable is the GND connection on the connector as illustrated below for connecting to a CozIR®-A sensor.

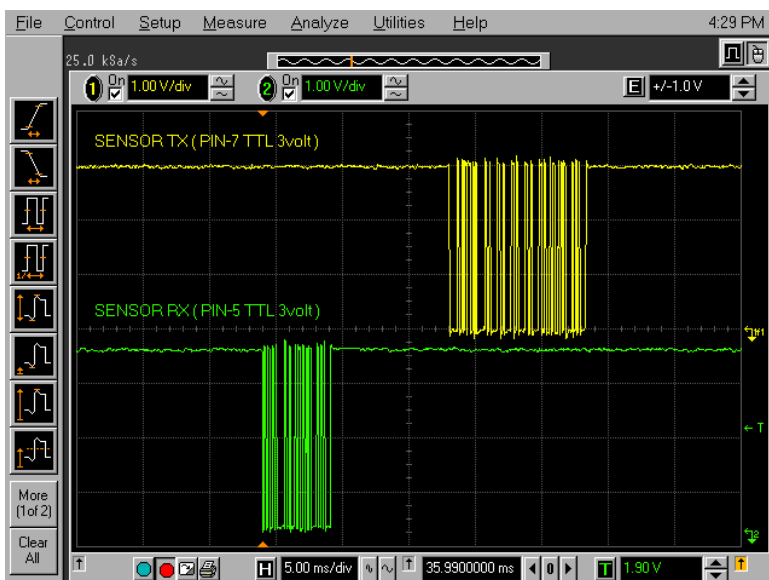


1.2 Serial Connection

The Rx and Tx pins are normally high, suitable for direct connection to a UART. If the sensor is to be read by a true RS232 device (e.g. a PC) it is necessary to pass through a level converter to step up/down the voltage and invert the signal.

Sensors have different connector sizes. For all sensors, only GND, 3V3, Rx and Tx are required for bi-directional serial connection. All other pins should be left unconnected.

An evaluation kit is available to allow simple interfacing between the sensor and a PC. Contact GSS for details: www.gassensing.co.uk | info@gassensing.co.uk



Scope trace showing command sent to the sensor (green) and sensor response (yellow)

Parameter	Value
Baud Rate	9600
Data Bits	8
Parity	None
Stop Bits	1
Format	UART (normally high)
Hardware Flow Control	None
Voltage Voh	3V (MISIR $V_{oh} = V_{supply}$)
Voltage Vih	3V-5V

NB If you connect to the sensor using HyperTerminal®, you must select the box “Send line ends with line feeds” under ASCII setup.

When initially powered, the sensor will immediately start to transmit readings (see Mode 1 in “**Operating Modes**”)

1.3 Reading Format

1.3.1 CO₂ Measurement

The CO₂ measurement is reported as:

```
Z ##### z #####\r\n
```

where

Z ##### shows the CO₂ concentration after digitally filtering

and

z ##### shows the instantaneous CO₂ concentration without any digital filtering.

The concentration is reported in the following units

Type	Range	Units	Example
CozIR®-A CozIR®-LP MISIR	Up to 2%	ppm	Z 00631 = 631ppm
ExplorIR®-W SprintIR®-W ExplorIR®-M	Up to 65%	ppm/10	Z 01200 = 12000ppm = 1.2%
ExplorIR®-W-100 SprintIR®-W-100 ExplorIR®-M-100	Up to 100%	ppm/100	Z 01500 = 150000ppm = 15%

Note that the same units must be used when sending concentration information to the sensor (for example, the X command and the F command). If in doubt, the ‘.’ Command (see below) will indicate what multiplier should be applied to the Z output to convert to ppm.

Z 00842 z 00765	<p>Sample output from a sensor with factory settings.</p> <p>This is a COZIR-A, so the reported CO₂ reading is 842ppm.</p> <p>The second figure shows the instantaneous (unfiltered) CO₂ reading. See "Digital Filter" for more details.</p>
Z 00842 z 00738	
Z 00842 z 00875	
Z 00842 z 00858	
Z 00842 z 00817	
Z 00842 z 00839	
Z 00842 z 00817	
Z 00842 z 00828	
Z 00842 z 00850	
Z 00842 z 00875	
Z 00842 z 00804	

Note that all output data from the sensor has a leading space.

1.3.2 Temperature Measurement (Factory Fit option on COZIR®-A and ExplorIR®-W sensors only)

The temperature measurement is reported as:

T #####\r\n

where ##### is a five digit number.

To convert to °C, subtract 1000 and divide by 10.

For example:

T 01235\r\n

Represents 23.5°C

NB the temperature and humidity sensor is a factory fit option. If it is not fitted, the sensor will return "T 01000".

1.3.3 Humidity Measurement (Option on CozIR-A and ExplorIR-W sensors only.)

The humidity measurement is reported as:

H #####\r\n

Where ##### is a five digit number.

To convert to relative humidity (%), divide by 10.

For example:

H 00551\r\n

Represents 55.1% RH

NB the temperature and humidity sensor is a factory fit option. If it is not fitted, the sensor will return "H 00000".

1.4 Example of T,H and CO₂

When shipped, the sensor default output is CO₂ only. To output temperature, humidity and CO₂, send "M 4164\r\n" (see "**Output Fields**"). The output format will have the form:

```
H 00345 T 01195 Z 00651\r\n
```

This example indicates 34.5% RH, 19.5°C and 651ppm CO₂

2 Command Summary

For complete details of the commands and their correct usage, please refer to the “**Command Reference**”.

NB

- All commands **must** be terminated with a carriage return and line feed <CR><LF>. In this document, this is shown as ‘\r\n’.
-
- Commands which take a parameter **always** have a space between the letter and the parameter.
-
- The sensor will respond with a ‘?’ if a command is not recognized. The two most common causes are missing spaces or missing <CR><LF> terminators.

Command	Use	Example	Response	Comments
A ###\r\n	Set the digital Filter	A 16\r\n	A 00016\r\n	See “ User Settings ”
a\r\n	Return the digital filter setting	a\r\n	a 00016\r\n	See “ User Settings ”
F #####\r\n	Fine Tune the zero point	F 410 400\r\n	F 33000\r\n	See “ Zero Point Calibration ”
G\r\n	Zero point calibration using fresh air.	G\r\n	G 33000\r\n	See “ Zero Point Calibration ”
H\r\n	Return most recent humidity measurement	H\r\n	H 00552\r\n	=55.2% in the example. Humidity sensing is a factory fit option.
K #\r\n	Selects the operating mode	K 1\r\n	K 00001\r\n	See “ Operating Modes ”
M #####\r\n	Sets the output fields	M 6\r\n	M 00006\r\n	See “ Output Fields ”
P ### ##\r\n	Sets a user configurable field in EEPROM	P 1 10\r\n	P 00001 00010\r\n	See “ User Settings ”
p ###\r\n	Reads a user-configurable field from EEPROM	p 10\r\n	p 00010 00001\r\n	See “ User Settings ”
Q\r\n	Return most recent fields	Q\r\n		See “ Command Reference ”
S #####\r\n	Sets the altitude compensation value	S 8192\r\n	S 08192\r\n	See “ Altitude Compensation ”
s\r\n	Return the altitude compensation value	s\r\n	s 08192\r\n	See “ Altitude Compensation ”
T\r\n	Return the most recent temperature measurement	T\r\n	T 01225\r\n	22.5°C in the example. Temperature sensing is a factory fit option.

Command	Use	Example	Response	Comments
U\r\n	Zero point calibration using nitrogen.	U\r\n	U 33000\r\n	See " Zero Point Calibration "
u #####\r\n	Manual setting of the zero point.	u 32997\r\n	u 32997\r\n	See " Zero Point Calibration "
X #####\r\n	Zero point setting using a known gas calibration	X 2000\r\n	X 32997\r\n	See " Zero Point Calibration "
Y\r\n	Return firmware version and sensor serial number	Y\r\n	Returns <u>two</u> lines	See " Command Reference " for details
Z\r\n	Return the most recent CO2 measurement.	Z\r\n	Z 01521\r\n	1521ppm in the example
@ #.# #.#\r\n	Autocalibration configuration	@ 1.0 8.0\r\n	@ 1.0 8.0\r\n	See " Autocalibration " for details
.\r\n	Return the multiplier required to convert the Z output to ppm	.\r\n	. 00100\r\n	Multiply by 100 in the example
*\r\n	Return configuration information.	*\r\n		See the " Command Reference " for details

All communications are in ASCII and are terminated by carriage return, line feed (ASCII characters 13 and 10). This document uses the protocol "\r\n" to indicate the carriage return line feed.

All responses from the sensor, including measurements, have a leading space (ASCII character 32).

The character '#' represents an ASCII representation of a numeric character (0-9).

Note that there is a space between the first letter and any parameter. For example, the X command reads "X space 2000 carriage return line feed".

Note that all settings are stored in non-volatile memory, so the sensor only has to be configured once. It should not be configured every time it is powered up.

3 Operating Modes

The CozIR®-LP, CozIR®-A, SprintIR®-W, ExplorIR®-M and MISIR™ sensors can be operated in three different modes. Users can switch between the modes using the "K" command.

3.1 Mode 0 Command Mode

This is primarily intended for use when extracting larger chunks of information from the sensor (for example using the Y and * commands).

In this mode, the sensor is stopped waiting for commands. No measurements are made, and the sensor will run through a warm-up cycle after exiting this command. There is no latency in command responses.

The power consumption is less than 1mW as no measurement activity takes place. Commands which report measurements or alter the zero point setting are disabled in mode 0.

Mode 0 is NOT retained after power cycling. The sensor will always power up in streaming or polling mode, whichever was the most recently used.

3.2 Mode 1 Streaming Mode

This is the factory default. Measurements are reported twice per second. Commands are processed when received, except during measurement activity, so there may be a time delay of up to 100mS in responding to commands. The power consumption is 3.5mW (assuming one field of information is transmitted, and there is no temperature and humidity sensor).

3.3 Mode 2 Polling Mode

In polling mode, the sensor only reports readings when requested. The measurement cycle continues in the background, but the output stream is suppressed. The power consumption depends on the frequency of polling but is approximately the same as the streaming mode power consumption.

Note that the sensor will power up in the mode last used. If it was last used in K0 mode, it will power up in either K1 or K2 mode, depending on which was most recently used.

In Polling Mode, measurements can be accessed using the polling commands H, L, Q, T and Z (see "**Command Reference**").

4 Output Fields

GSS sensors can be configured to output up to five fields of information. Typically, the only fields of interest are the CO₂ concentration, Temperature (if fitted) and Humidity (if fitted).

This allows users to customise the output string transmitted by the sensor. Up to five values can be transmitted in the string. The format is always the same: each field is identified by a single character, followed by a space, followed by the five digit number indicating the value of the parameter.

The output fields can be set by sending a command of the format *M 12345\r\n* where 12345 represents a mask value which defines the output fields.

The mask value is created by adding the mask values for the parameters required (see table below).

The sensor will output a maximum of five fields. If the mask setting represents more than five fields, only the first five (those with the highest mask values) will be output.

NB SprintIR® sensors have a limited time to transmit information, so no more than two fields should be selected for output.

Parameter	Field Identifier	Mask Value	Comments
Reserved		32768	Reserved
Reserved		16384	Reserved
Reserved		8192	Reserved
Humidity	H	4096	Reports the humidity output of the Temperature and Humidity Sensor (if Fitted).
D digitally filtered	d	2048	Reports a value related to the normalized LED signal strength (smoothed)
D unfiltered	D	1024	Reports a value related to the normalized LED signal strength
Reserved		512	Reserved
Zero Set Point	h	256	Reports a value related to the normalized LED signal strength
Sensor Temperature (unfiltered)	V	128	Reports a value which varies inversely with the sensor temperature.
Temperature	T	64	Reports the temperature output of the Temperature and Humidity Sensor (if Fitted).
LED Signal (digitally filtered)	o	32	Reports a value which gives an indication of the LED signal strength (smoothed)
LED Signal (unfiltered)	O	16	Reports a value which gives an indication of the LED signal strength.
Sensor Temperature (filtered)	v	8	Reports a value which varies inversely with the sensor temperature. (smoothed)
CO2 Output (Digitally Filtered)	Z	4	Digitally filtered CO2 reading
CO2 Output (not filtered)	z	2	Instantaneous CO2 reading
Reserved		1	Reserved

Note that most fields are for advanced use only and require specific guidance from GSS engineering for their correct interpretation and use. Measurement fields are indicated in **bold**.

For example, to output the temperature, humidity and CO2 measurements, send:

```
M 4164\r\n
```

The output string will then be:

```
H 12345 T 12345 Z 00010\r\n
```

5 Zero Point Calibration

There are a several methods to calibrate the zero point of the sensor. The recommended method is zero point calibration in a known gas (see X command) which will give the most accurate zero setting.

In all cases, the best zero is obtained when the gas concentration is stable and the sensor is at a stabilized temperature.

Note that zero point calibrations are not cumulative – only the latest zero point calibration is effective. For example, there is no benefit in zero point calibrating in nitrogen, and then calibrating in a calibration gas. The sensor will store only the latest zero point.

5.1 Zero in a known gas concentration (recommended)

Place the sensor in a known gas concentration and allow time for the sensor temperature to stabilize, and for the gas to be fully diffused into the sensor.

Send the command `X ###\r\n`

The concentration must be in the same units as the sensor output (see "**Reading Format**").

The sensor will respond with an echo of the command and the new zero point.

For example, to set the zero point in a CozIR[®]-A when the sensor is in a known gas concentration of 2000ppm

```
send:      X 2000\r\n
response:  X 32950\r\n
```

5.2 Zero in Nitrogen

Place the sensor in a gas containing no CO₂ (typically nitrogen). and allow time for the sensor temperature to stabilize, and for the gas to be fully diffused into the sensor.

Send the command `U\r\n`

The sensor will respond with an echo of the command and the new zero point.

For example,

```
send:      U\r\n
response:  U 32950\r\n
```

5.3 Zero in Fresh Air (assumed to be 400ppm)

If there is no calibration gas and no nitrogen available, the sensor zero point can be set in fresh air. The sensor is programmed to assume that fresh air is 400ppm (this value is user configurable – see "*User Settings*").

Place the sensor in a fresh air environment and allow time for the sensor temperature to stabilize, and for the fresh air to be fully diffused into the sensor.

Send the command `G\r\n`

The sensor will respond with an echo of the command and the new zero point.

For example,

```
send:      G\r\n
response:  G 32950\r\n
```

5.4 Fine Tune the Zero Point

If the CO₂ concentration and the sensor reported concentration are known, the zero point can be adjusted using the known concentration to fine tune the zero point. This is similar in operation to the "X" command (see above) but can operate on historic data. For example, if the sensor has been in an environment in which it is known to have been exposed to outside air, and the sensor reading is known at that time, the zero point can be fine tuned to correct the reading. This is typically used to implement automated calibration routines.

The command takes two parameters, separated by a space. The first parameter is the reading reported by the sensor. The second is the corrected reading. Both parameters must be in the same units as the sensor output (see "**Reading Format**")

The sensor will respond with an echo of the command and the new zero point.

For example,

```
send:      F 400 380\r\n
response:  F 32950\r\n
```

In this example, the sensor zero point would be corrected so that a reading of 400ppm, would now be reported as 380ppm.

5.5 Zero Point Adjustment

The precise zero point can be fine-tuned by sending a zero point to the sensor. This is not recommended for general use.

Send the command `u #####\r\n` where ##### is the new zero point.

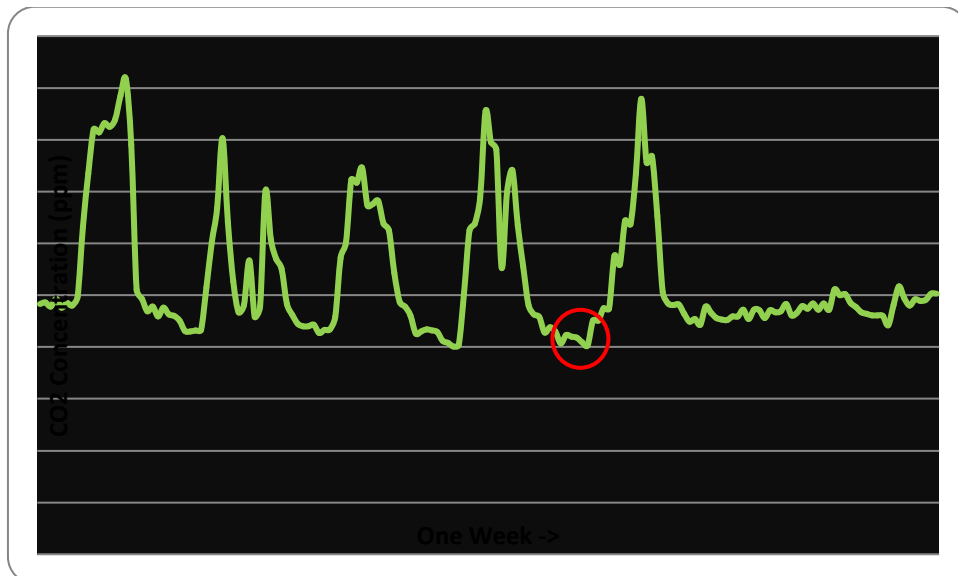
6 AutoCalibration

6.1 Principle of Operation

GSS sensors are fully calibrated prior to shipping from the factory. Over time, the zero point of the sensor needs to be calibrated to maintain the long term stability of the sensor.

In many applications, this can happen automatically using the built in auto-calibration function.

This technique can be used in situations in which sensors will be exposed to typical background levels (400-450ppm) at least once during the auto-calibration period. For example, many buildings will drop quickly to background CO₂ levels when unoccupied overnight or at weekends. The auto-calibration function uses the information gathered at these periods to recalibrate.



This recording from a sensor shows a typical one week recording in an office environment. The auto-calibration function uses the low point (circled) and uses it to recalibrate the zero point.

Users can modify the control parameters for autocalibration (see '**Setting the Autocalibration Parameters**'), and modify the autocalibration behavior (see '**Customising Autocalibration**').

6.2 Requirements for Auto-calibration

Exposure to Fresh Air

The sensor must 'see' fresh air at least once during the auto-calibration period. You do not need to know when the fresh air will be sensed, just that it will be sensed at some point during the period.

Continuously Powered

The auto-calibration information is deleted when the sensor is switched off. This ensures that each installation is unaffected by any previous history of the sensor. For auto-calibration to function, it must be power on for the whole of the auto-calibration period.

6.3 Setting the Auto Calibration Parameters

Three parameters are required to enable the auto-calibration routine: autocalibration interval, background CO2 level and start-up interval.

6.4 Auto-Calibration Intervals

This determines how often the auto-calibration takes place.

6.5 Background Concentration

Typically 400-450ppm. This is the level the sensor will use as background.

6.6 Initial Auto-calibration Interval

It is possible for the first auto-calibration to take place more quickly than the regular auto-calibration event. This can be useful to compensate for any drift soon after or as a result of installation.

Note that the autocalibration timers are reset when the power to the sensor is interrupted. On power on, the sensor will always time an initial autocalibration interval first, then settle into the regular autocalibration cycle.

The regular autocalibration timer is reset automatically if the user calibrates the sensor using the U,G,X,F or u commands.

6.7 Autocalibration Intervals

The autocalibration intervals are set using the '@' command. This command allows the autocalibration periods to be set, interrogated or disabled.

To set the autocalibration intervals, the command structure is

@ initialinterval regularinterval\r\n

Where both the initial interval and regular interval are given in days. Both must be entered with a decimal point and one figure after the decimal point.

For example

```
send:      @ 1.0 8.0\r\n
response:  @ 1.0 8.0\r\n
```

Will set the autocalibration interval to 8 days, and the initial interval to 1 day.

Note that there is a space between the @ and the first number, and a space between the two numbers. In hex, the example above reads

```
40 20 31 2E 30 20 38 2E 30 0D 0A
```

6.8 Read the Autocalibration Settings

To determine the current autocalibration settings:

```
send:      @ \r\n
response:  @ 1.0 8.0\r\n
```

If the autocalibration is enabled, the sensor will respond with the format above showing the initial and regular autocalibration intervals. If the autocalibration is disabled, the sensor will respond with

```
send:      @ \r\n
response:  @ 0\r\n
```

6.9 Disable Autocalibration

To disable the autocalibration:

```
send:      @ 0\r\n
response:  @ 0\r\n
```

i.e., @ followed by a space followed by a zero terminated with 0x0d 0x0a

6.10 Background Concentration

The background concentration depends somewhat on the area the sensor is installed. Typically, a figure between 400ppm and 450ppm is used. The factory default is 400ppm.

To set this, send

```
P 8 x\r\n
P 9 y\r\n
```


where x and y depend on the concentration you want to set.

Concentration	x	Y
380	1	124
400	1	144
425	1	169
450	1	194

This is stored as a two byte value, the high byte being in location 8 and the low byte in location 9. The value represents the concentration.

To calculate other values,

$$x = \text{int}(\text{concentration}/256)$$

$$y = \text{the remainder after dividing } \text{concentration}/256$$

Note that the value must be in the same units as the sensor reads our CO2 in. This is checked by the dot command (.)

6.11 Customising Autocalibration

By default, the autocalibration will implement a correction to the sensor at the frequency specified by the user. This is the most common application and is recommended by GSS. If required, the behaviour can be modified in a number of ways:

6.12 Threshold

Corrections above a particular threshold level will not be implemented. ie if the correction is too large, the autocalibration will not take place. This does not permanently disable the autocalibration function, just prevents it from implementing the correction on that specific occasion.

6.13 Proportional

This allows only a proportion of the correction to be applied. For example, if the calculated correction is 100ppm, users can specify that only (eg) half of the correction should be applied. This has the benefit of limiting individual changes, but the drawback that the full correction may never be applied. If the threshold level is set, only corrections larger than the threshold level will be restricted in this way. Corrections lower than the threshold level will be applied in full.

6.14 Spread Implementation

In this case, the correction is applied in full, but is applied gradually over the autocalibration period. For example, if the correction to be applied is 50ppm, the autocalibration period is 8 days, and the initial autocalibration period is 1 day, then one eighth of the correction (=50ppm/8) will be applied each day. If the initial autocalibration period is 12 hours, one sixteenth of the correction will be applied every 12 hours.

6.15 Register Values for Customising Autocalibration.

These modes are implemented by setting the values in registers P7, P16, P17 and P18

Register	Meaning	Example	
P7	Controls Autocalibration Mode (see table below)	P 7 0/r/n disables autocalibration	
P16	Sets divider for proportional autocalibration	P 16 2/r/n sets the divider to 2	
P17	Autocalibration Threshold Value – High Byte	P 17 0/r/n	sets threshold to 100
P18	Autocalibration Threshold Value – Low Byte	P 18 100/r/n	

The main control is the register P7.

P7 Setting	Autocalibration Mode	Effect of Threshold
0	Autocalibration Disabled	None
1	Standard Autocalibration Enabled	None
3	Proportional Autocalibration	Corrections larger than threshold value will be limited by dividing by register P16
5	Spread Autocalibration	Corrections larger than threshold value will be spread over the autocalibration period.
7	Standard Autocalibration, with Threshold	Corrections larger than the threshold value will not be implemented.

The effect of different modes and threshold values is illustrated below. In all cases, it is assumed that the initial autocal period is 1 day, and the autocalibration period is eight days. For the example, a threshold of 50 is assumed.

Two cases are illustrated: a small correction of 10 ppm, and a large correction of 100 ppm

P7 Setting	10ppm Correction	100ppm Correction
0 – autocalibration disabled	No correction	No Correction
1 – standard autocalibration	10 ppm correction	100ppm correction
3 – proportional autocalibration (assumed register P16 is set to 2)	10ppm correction (below threshold value, full correction is applied)	50ppm correction (above threshold value, correction is divided by P16 value)
5 – spread autocalibration	10ppm correction (below threshold value, full	6ppm correction applied per day.

	correction is applied)	
7 – standard autocalibration, threshold enabled	10ppm correction	No correction (above threshold value)

7 Altitude Compensation

Important

This feature was introduced in sensors manufactured after July 2013 using firmware version AL17 or higher. The firmware version can be identified by sending the Y or * command.

Altitude compensation applies a permanent correction to the sensor response, so should only be used when it is known that the sensor will be operating at altitude permanently.

NDIR gas sensors, such as the COZIR, SprintIR, ExplorIR and MISIR family of sensors, detect the concentration of gas by measuring the degree of light absorption by the gas analyte. The degree of light absorption is then converted into a concentration reported by the sensor.

The absorption process is pressure dependant, so that a change in pressure will cause a change in the reported gas concentration. As the pressure increases, the reported gas concentration also increases. As the pressure decreases, the reported concentration decreases.

This effect takes place at a molecular level and is common to all NDIR gas sensors.

In common use close to 1013mbar and at concentrations typical in buildings etc, the reading will vary by approximately 0.12% of reading for each mbar change in barometric pressure (the sensors are calibrated at 1013mbar).

If the sensor is installed at an elevated altitude, the mean barometric pressure will be lower than 1013mbar. It is possible to configure the sensor to correct for this effect, by setting the altitude compensation factor at installation. This will apply a permanent correction to the output of the sensor, depending on the altitude setting selected.

To apply this correction,

- 1) Select the appropriate code from the table below (intermediate values can be interpolated)
- 2) Send the "`S ####\r\n`" to the sensor, where #### is the code from the table below.

For example, to correct the sensor for permanent installation at 305m elevation,

```
send:      S 8592\r\n
response:  S 08592\r\n
```

Altitude (ft)	Altitude (m)	Barometric Pressure	Code
---------------	--------------	---------------------	------

		(mbar)	
-1000	-305	1050	7762
0	0	1013	8192
1000	305	976	8592
2000	610	942	8978
3000	915	908	9364
4000	1219	875	9763
5000	1524	843	10183
6000	1823	812	10632
7000	2133	781	11116
8000	2438	752	11657

The current setting can be determined by sending a lower case s:

```
send:      s\r\n
response:  s 08592\r\n
```

At high gas levels > 10000ppm the compensation factor increases to 0.0015% per mbar. Full pressure compensation is a complex issue for gas sensors and if higher accuracy over wide pressure and gas concentration ranges is required GSS can supply improved compensation algorithms.

8 User Settings

8.1 Digital Filter

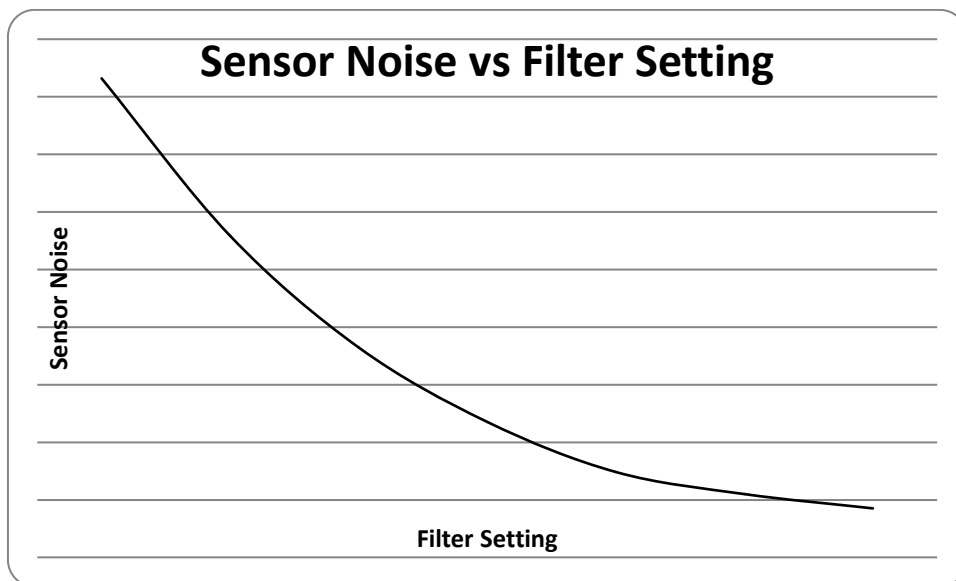
The CO₂ measurement is passed through a digital filter to condition the signal. The characteristics of the filter can be altered by the user to tune the sensor performance to specific applications.

The filter operates as a low pass filter - increasing the filter parameter reduces measurement noise, but slows the response. There is a tradeoff between noise(resolution) and speed of response.

8.1.1 Customising the Sensor Response

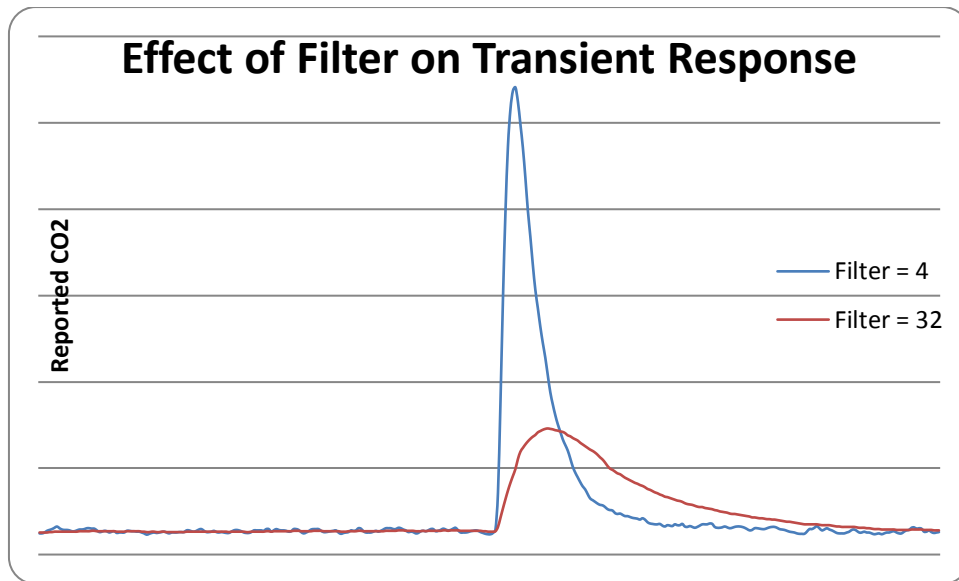
The filter can be set to a value between 1 and 65535. Settings larger than 64 are not recommended for normal use, though much higher settings can be used in SprintIR sensors if response time is not important. A low value will result in the fastest response to changes in gas concentration, a high value will result in a slower response. Note that the response is also determined by the diffusion rate into the sensor. The default setting is 32.

This chart shows the effect of changing the filter setting:



Increasing the filter setting has a beneficial impact on noise, so improves the sensor resolution. It also slows the sensor response to transients. This can be used to improve the detection of average CO₂ conditions. In building control, for example, a fast response to breathing near the sensor is undesirable. If the transient response is important either for speed of response or because the shape of the transient is required, a low filter setting should be used.

The following chart shows the same transient event capture using a filter setting of 4, and using a filter setting of 32.



8.1.2 Setting the Digital Filter

To change the setting, type `A ###\r\n` where `###` is the required filter setting.

For most applications, a filter setting of 32 is recommended.

```
send:      A 32\r\n
response:  A 0032\r\n
```

If the filter is set to zero, a smart filter mode will be used in which the filter response is altered to suit the prevailing conditions. This is useful if there is a combination of steady state conditions, with some periods of rapidly changing concentrations.

8.1.3 Reading the Digital Filter Setting

The current setting for the digital filter can be determined by sending `a\r\n`

```
send:      a\r\n
response:  a 0032\r\n
```

8.2 User Options – EEPROM Settings

Some user settings can be altered in the internal EEPROM.

These settings can be set by using the parameter setting command "P", and read using a lower case "p".

There are also 32 bytes of user EEPROM storage available.

8.2.1 Setting EEPROM

To set an EEPROM location, send

```
P ### ##\r\n
```

where the first parameter is the address, and the second is the value.

Note that two byte values must be set one byte at a time.

For example, to change the default value of the ambient gas concentration used for ambient calibration (ie the assumed CO₂ concentration in fresh air) to 380ppm, send

```
send:      P 10 1\r\n
response:  P 00010 00001\r\n
send:      P 11 124\r\n
response:  P 00011 00124\r\n
```

8.2.2 Reading EEPROM

To read a parameter value from an EEPROM location, send "p #####\r\n" where ##### is the address of the parameter.

Note that two byte values must be read one byte at a time.

For example, to read the value of the ambient gas concentration used for ambient calibration (i.e. the assumed CO₂ concentration in fresh air)

```
send:      p 10\r\n
response:  p 00010 00001\r\n
send:      p 11\r\n
response:  p 00011 00124\r\n
```

8.3 EEPROM Settings

Most of the EEPROM settings are two byte values, indicated by HI and LO in the variable name in the following table. We recommend contacting GSS before altering the default values.

Location	Name	Purpose	Default Value
0	AHHI	Reserved	0
1	ANLO_	Reserved	0
2	ANSOURCE	Reserved	0
3	ACINITHI	Autocalibration Preload. This preloads the autocalibration timer so that the first autocalibration occurs after a shorter time.	87
4	ACINITLO	Low byte of above.	192
5	ACHI	Autocalibration Interval. Sets the time interval between autocalibrations.	94
6	ACLO	Low byte of above.	128
7	ACONOFF	Switches Autocalibration ON/OFF	0
8	ACPPMHI	Autocalibration Background	1

		Concentration. This determines what background CO ₂ level is assumed for autocalibration.	
9	ACPPMLO	Low byte of above.	194
10	AMBHI	Ambient Concentration (for G command). This determines what background CO ₂ level is assumed for ambient calibration using the "G" command.	1
11	AMBLO	Low byte of above.	194
12	BCHI	Buffer clear time. This will clear any incomplete commands from the serial buffer after a fixed period of inactivity. The time is in half second increments.	0
13	BCLO	Low byte of above.	8
14		Reserved	0
15		Reserved	0
16	ACALDIV	Divider for proportional autocalibration	1
17	ACALTHHI	Threshold for autocalibration	0
18	ACALTHLO	Low Byte of Above	0
200-231		User EEPROM	255

8.3.1 Autocalibration Settings (locations 3-7)

These are included now to maintain compatibility with previous firmware versions. GSS recommends using the @ command to set autocalibration timings.

8.3.2 User EEPROM

Locations 200 to 231 can be used to store user values. Each location is a single byte. These locations are not used by the sensor.

E.g. to store the number 42 in the first user EEPROM location:

```
send:      P 200 42\r\n
response:  P 00200 00042\r\n
```

and to read it

```
send:      p 200\r\n
response:  p 00200 00042\r\n
```

Note that the EEPROM is only guaranteed for 100,000 write cycles.

9 Command Reference

This gives the complete command set for the CozIR[®]-A, CozIR[®]-LP, SprintIR[®]-W, SprintIR[®]-6S, ExplorIR[®]-W, ExplorIR[®]-M and MISIR sensors and illustrates use of some of the more commonly used options.

Key points to note are:

- In all cases, commands are terminated with a carriage return, line feed (“\r\n”).
- Commands are case sensitive.
- The commands use all use ASCII characters. Each command lists the ASCII letter and includes the hex code for avoidance of doubt.
- Always check for a correct response before sending another command.
- If a command is unrecognized, the sensor will respond with a “?”

WARNING

This document is provided to give a complete reference of the command set and outputs from the GSS CO₂ sensors. It is intended for advanced users only. If in doubt, please contact GSS engineering prior to use.

9.1 Customisation

A COMMAND (0x41)

USER CONFIGURATION

Example: `A 128\r\n`

Description: Set the value for the digital filter.

Syntax: ASCII character 'A', SPACE, decimal, terminated by 0x0d 0x0a (CR & LF)

Response: `A 00032\r\n`

a COMMAND (0x61)

INFORMATION

Example: `a\r\n`

Description: Return the value for the digital filter.

Syntax: ASCII Character 'a' terminated by 0x0d 0x0a (CR & LF)

Response: `a 00032\r\n`

M COMMAND (0x4D)

USER CONFIGURATION

Example: `M 212\r\n`

Description: Determines which values are going to be returned by the unit.

Syntax: "M", SPACE, followed by an up-to 5 digit number, each bit of which dictates which item will be returned by the sensor, terminated by 0x0d 0x0a (CR & LF).

Response: `M 212\r\n` (see "Output Fields" for details)

P COMMAND (0x50)

USER CONFIGURATION

Example: `P 10 1\r\n`

Description: Sets a user configurable parameter.

Syntax: "P", SPACE, followed by an up to 2 digit number, SPACE followed by an up to 3 digit number, terminated by 0x0d 0x0a (CR & LF).

Response: `P 00001 00010\r\n` (see "User Settings" for details)

p COMMAND (0x70)

USER CONFIGURATION

Example: `p 10\r\n`

Description: Returns a user configurable parameter.

Syntax: "P", SPACE, followed by an up-to 2 digit number, terminated by 0x0d 0x0a (CR & LF).
 Response: *P 10 1\r\n* (see "User Settings" for details)

9.2 Information

Y COMMAND (0x59)

INFORMATION

Example: *Y\r\n*

Description: the present version string for the firmware
 Syntax: ASCII character 'Y', terminated by 0x0d 0x0a (CR & LF)
 Response: *Y,Jan 30 2013,10:45:03,AL17\r\n
 B 00233 00000\r\n*

NB This command returns two lines split by a carriage return line feed, and terminated by a carriage return line feed. This command requires that the sensor has been stopped (see 'K' command).

*** COMMAND (0x59)**

INFORMATION

Example: **\r\n*

Description: Returns a number of fields of information giving information about the sensor configuration and behavior.
 Syntax: ASCII character '*', terminated by 0x0d 0x0a (CR & LF)
 Response: Contact GSS for details.

. COMMAND (0x2E)

INFORMATION

Example: *.\r\n*

Description: Returns a number indicating what multiplier must be applied to the Z or z output to convert it into ppm.
 Syntax: ASCII character '.', terminated by 0x0d 0x0a (CR & LF)
 Response: *. 00001\r\n* (this number is variable).

9.3 Switching between Modes

For discussion of different modes of operation, see the section "Operating Modes".

K COMMAND (0x4B)

USER CONFIGURATION

Example: *K 1*
 Description: Switches the sensor between the operating modes.
 Syntax: ASCII character "K", SPACE, followed by the mode number, terminated by 0x0d 0x0a (CR & LF).
 Response: *K 1\r\n* (the number mirrors the input value).

9.4 Zeroing and Calibration

See examples of each of the zero and calibration commands in the following section.

U COMMAND (0x55)**CALIBRATION – USE WITH CARE**

Example: `U\r\n`

Description: Calibrates the zero point assuming the sensor is in 0ppm CO₂.

Syntax: ASCII Character 'U' terminated by 0x0d 0x0a (CR & LF)

Response: `U 32767\r\n` (the number is variable)

G COMMAND (0x47)**CALIBRATION – USE WITH CARE**

Example: `G\r\n`

Description : Calibrates the zero point assuming the sensor is in 400ppm CO₂.

Syntax: ASCII character 'G'

Response: `G 33000\r\n` (the number is variable).

F COMMAND (0x46)**CALIBRATION – USE WITH CARE**

Example: `F 410 390\r\n`

Description : Calibrates the zero point using a known reading and known CO₂ concentration.

Syntax: ASCII character 'F' then a space, then the reported gas concentration then a space then the actual gas concentration.

Response: `F 33000\r\n` (the numbers are variable).

X COMMAND (0x58)**CALIBRATION – USE WITH CARE**

Example: `X 1000\r\n`

Description : Calibrates the zero point with the sensor in a known concentration of CO₂.

Syntax: ASCII character 'X' then a space, then the gas concentration.

Response: `X 33000\r\n` (the number is variable).

S COMMAND (0x53)**CALIBRATION – USE WITH CARE**

Example: `S 8192\r\n`

Description: Set the 'Altitude Compensation' value in EEPROM

Syntax: ASCII character 'S', SPACE, decimal, terminated by 0x0d 0x0a (CR & LF)

Response: `S 8192\r\n` (the number mirrors the input value).

s COMMAND (0x73)**INFORMATION**

Example: `s\r\n`

Description: Reports the Altitude Compensation value in EEPROM. See “**Altitude Compensation**”.

Syntax: ASCII Character 's', terminated by 0x0d 0x0a (CR & LF)

Response: `s 8193\r\n`

u COMMAND (0x75)**USE ONLY WITH GSS GUIDANCE**

Example: `u 32767\r\n`

Description: Send a zero set point.
 Syntax: ASCII character 'u', SPACE, decimal, terminated by 0x0d 0x0a (CR & LF)
 Response: *u 32767\r\n*

NB for advanced use only. Contact GSS before using this command.

There are three variants of the autocalibration configuration command:

@ COMMAND (0x40) **INFORMATION**

Example: *@\r\n*

Description: Return the autocalibration settings
 Syntax: ASCII character '@' terminated by 0x0d 0x0a (CR & LF)
 Response: *@ 1.0 8.0\r\n* (if autocalibration is enabled).
@ 0\r\n (if autocalibration is disabled).

@ COMMAND (0x40) **CALIBRATION – USE WITH CARE**

Example: *@ 0\r\n*

Description: Switch off the autocalibration function
 Syntax: ASCII character '@' followed by a SPACE followed by a zero terminated by 0x0d 0x0a (CR & LF)
 Response: *" @ 0\r\n"*

@ COMMAND (0x40) **CALIBRATION – USE WITH CARE**

Example: *@ 1.0 8.0\r\n*

Description: Set the Autocalibration timing
 Syntax: See **"Autocalibration"** section
 Response: *@ 1.0 8.0\r\n* (the number mirrors the input value).

9.5 Polling Commands

H COMMAND (0x48) **INFORMATION**

Example: *H\r\n*

Description: Reports the humidity measurement from the temperature and humidity sensor (if fitted). Divide by 10 to get the %RH
 Syntax: ASCII Character 'H', terminated by 0x0d 0x0a (CR & LF)
 Response: *H 00551\r\n*

T COMMAND (0x54) **INFORMATION**

Example: *T\r\n*

Description: Reports the humidity measurement from the temperature and humidity sensor (if fitted). Subtract 1000 and divide by 10 to get the temperature in °C.
 Syntax: ASCII Character 'T', terminated by 0x0d 0x0a (CR & LF)
 Response: *T 01224\r\n*

Z COMMAND (0x5A)**INFORMATION**

Example: `Z\r\n`

Description: Reports the latest CO2 measurement in ppm.

Syntax: ASCII Character 'Z', terminated by 0x0d 0x0a (CR & LF)

Response: `Z 00512\r\n`

Q COMMAND (0x51)**INFORMATION**

Example: `Q\r\n`

Description: Reports the latest measurement fields as defined by the most recent 'M' command.

Syntax: ASCII Character 'Q', terminated by 0x0d 0x0a (CR & LF)

Response: `H 12345 T 12345 Z 00010\r\n`