

Communications Interface

The CO₂ sensors communicate with an external host by means of an asynchronous, RS-232, UART serial communications port. All communications over this UART serial interface must be wrapped in the proprietary Telaire Tsunami-Lite Communications Protocol.

UART Communications Logic and Timing

Timing information for the CO₂ sensor may be considered at three levels:

- Timing related to byte/command transfers
- General system timing
- Communications at Power-Up

Timing Related to Byte/command Transfers

The UART communications interface expects a frame size of 8 bits, no parity, one stop bit, and a baud rate of 19200.

No communication reply can be initiated during data acquisition or processing. When a sensor fails to reply to a command request, simply send the request again. This phenomenon is more apparent in models with a fast cycle period.

General System Timing

The internal cycle of data acquisition and processing (“dsp cycle”) is dependent upon the sensor model. The duration of the cycle may be from one to several seconds. For a given sensor model, it is advised to keep the interval for gas concentration requests no shorter than the internal data acquisition and processing cycle. The host could interrogate the sensor more frequently for concentration readings, but it is not recommended and not productive.

The time interval between other commands is less restricted. In general, with the exception of a Status command following a Calibration command, a subsequent command can be issued as soon as the reply from the previous command has been received. For a Status command following a Calibration command, the host should wait at least one dsp cycle length before issuing the first Status command. This allows time for the sensor to begin the calibration process.

Communications at Power-Up

A communications delay of several seconds occurs when the sensor is powered up or power cycled. This communications delay time is necessary for the sensor to initialize and achieve full functionality.

After initialization, the sensor stays in a Warm-up mode. The duration of the Warm-up period is dependent upon the sensor model. The difference between Warm-Up mode and normal operating mode is that, in the Warm-Up, the sensor may not yet report accurate readings, and hence cannot execute any calibration commands. All other commands can be executed during Warm-up.

The Status of the sensor can be checked by using the Status command (see the commands description below). This command returns the status byte with a number of flags, including the Warm-Up status flag.

The gas ppm concentration can be read while the sensor is in Warm-Up mode; however, the data may not be accurate.

Note: *If for any reason the sensor does not respond to a request, simply re-send the command.*

UART Serial Communications Interface

The CO₂ sensor communicates over an asynchronous, UART interface at 19200 baud, no parity, 8 data bits, and 1 stop bit. When a host computer or PC communicates with the sensor, the host computer sends a request to the sensor, and the sensor returns a response. The host computer acts as a master, initiating all communications, and the sensor acts as a slave, responding with a reply.

All sensor commands and replies are wrapped in the proprietary Telaire Tsunami-Lite Communications Protocol to insure the integrity and reliability of the data exchange. The Communications Protocol for the serial interface and the command set for the CO₂ sensor are described in detail in the sections that follow.

UART Tsunami-Lite Communications Protocol

Each command to a sensor consists of a length byte, a command byte, and any additional data required by the command. Each response from the sensor consists of a length byte and the response data if any. Both the commands to the sensor and the responses from the sensor are wrapped in the Tsunami-Lite communications protocol layer.

Command: <length><command><additional_data>
Response: <length><response_data>

The communications protocol consists of a flag bytes (0xFF) and an address byte as a header. The protocol has no trailer.

Header Message Body
<flag><address> <Command/Response>

UART Commands from PC to Sensor

Commands sent from a host computer or PC to the sensor have the following format:

<flag> <address> <length> <command> <additional_data>

where:

- <flag> the hex value 0xFF
- <address> one byte hex value. The byte 0xFE is an address to which all sensors respond.
- <length> total length in bytes of the command and additional data
- <command> one byte hex command, values explained below
- <additional_data> may or may not be applicable, depending upon the command

For example, to request sensor identification, the following command is used:

```
0x FF  0xFE  0x02  0x02  0x01
<flag> <address> |  |  |
                <length> | <additional data> = SERIAL_NUMBER
                    <command> = CMD_READ
```

The length of the command is 0x02, since the command CMD_READ, SERIAL_NUMBER consists of the two bytes “0x02 0x01”.

UART Response from Sensor to PC

Responses returned from the sensor to the host computer or PC have the following format:

<flag> <address> <length> <response_data>

where:

- <flag> the hex value 0xFF.
- <address> one byte hex value. The byte 0xFA signifies “to master” in a master/slave communication.
- <length> total length in bytes of the response data
- <response_data> may or may not be applicable, depending upon the command

In response to the above identification command CMD_READ SERIAL_NUMBER, one sensor replied with the following byte stream:

```
0xFF  0xFA  0x0F  0x4E  0x4F  0x42  0x30  0x30  0x31  0x32  0x34  0x00  0x00  0x00  0x00
<flag> <address> <length> <response_data> -----
```

The length of the response_data is fifteen bytes (0x0F). The first eight bytes of the response_data, “4E 4F 42 30 30 31 32 34”, is the ASCII string “NOB00124”, the serial number for the sensor. The remaining bytes of the fifteen byte response are filled with nulls.

UART Acknowledgement or <ACK> Reply

Some commands require that a sensor only confirm that the command was received and the appropriate action was taken. In this case, when a sensor does not need to return data in response to a command, it will instead reply with an Acknowledgement response, called an <ACK>. This is a response packet formatted as shown above, but with the <length> equal to 0x00, and no response data present:

```
0xFF  0xFA  0x00  
<flag> <address> <length>
```

Examples of commands that expect an Acknowledgement response are Update Commands, Calibrate Commands, and the Skip Warmup Command. Detailed descriptions of these commands are given below.

Command Reference for the CO₂ Sensor

Every common exchange of data between a host processor (or PC) and the sensor starts with a request data-packet sent to the sensor, followed by a response data-packet returned from the sensor. The request data-packet contains a command byte telling what data or sensor action is required. The command byte also determines what additional data is included in the request packet.

Note: *Each request and response must be wrapped in the Tsunami-Lite communications protocol, as described above. The following Command Reference gives only the command syntax and response, and omits the protocol wrapping.*

In the following Commands tables, hex bytes are represented as “0x12” for clarity. However, when sending the byte string in a message, the “0x” notation must be omitted. The commands listed in the following sections are common to all members of the CO₂ sensors unless noted otherwise.

CMD_READ Commands

The CMD_READ Command reads a data value or parameter from the sensor. For almost all <data ID> values, this value is read from RAM. Only COMPILE_SUBVOL and COMPILE_DATE are read directly from Flash.

CMD_READ = 0x02		Read a data value or parameter from the sensor
Req:	0x02 <data ID>	Request is the command byte, 0x02, followed by a byte number that identifies which data value or parameter to read
Resp:	<data> [... <data>]	Response is one or more bytes of data. Details of useful values that can be read from the CO ₂ sensor follow.
CMD_READ GAS_PPM		Read the gas PPM as measured by the sensor.
Req:	0x02 0x03	
Resp:	<ppm_msb> <ppm_lsb>	Response is a 2-byte binary value giving the PPM. For some models, the PPM value is an unsigned integer between 0 and 65,535. For other models, it is a signed value between -32768 and 32767. Order of the returned bytes is <msb,lsb>. For some sensor models, this value must be multiplied by 16 to obtain the actual PPM.
CMD_READ SERIAL_NUMBER		Read the serial number from the sensor.
Req:	0x02 0x01	
Resp:	[ASCII string, 15 bytes, null filled]	Response is 15 bytes, the first of which are an ASCII string of printable characters, for example, "074177". The remaining bytes of the response are the null character, 0x00.
CMD_READ COMPILE_SUBVOL		Read the compilation subvolume for the sensor control software. COMPILE_DATE and COMPILE_SUBVOL together identify the software version.
Req:	0x02 0x0D	
Resp:	[3-byte ASCII string]	Response is a 3-byte ASCII string (e.g., "A10").
CMD_READ COMPILE_DATE		Read the compilation date for the sensor control software. COMPILE_DATE and COMPILE_SUBVOL together identify the software version.
Req:	0x02 0x0C	
Resp:	[6-byte ASCII string]	Response is an ASCII string representing a date, for example "060708" for July 8, 2006.
CMD_READ ELEVATION		Read the elevation in feet above sea level, a required operating parameter for the sensor. The sensor's elevation setting is used to estimate air pressure and is factored into the calculation of the PPM.
Req:	0x02 0x0F	
Resp:	<elevation_msb> <elevation_lsb>	Response is a 2-byte binary value giving the elevation value between 0 and 65,535 feet. The practical limit is 5000 feet. Order of the returned bytes is <msb,lsb>.

CMD_UPDATE Commands

The CMD_UPDATE Command writes a data value to both RAM and FLASH memories.

CMD_UPDATE ELEVATION	Set/Write the elevation in feet above sea level, a required operating parameter for the sensor. Elevation is expressed as a 2-byte binary value. Order of the bytes is <msb,lsb>. Elevation is normally expressed in increments of 500 feet from 0 to 5000 feet.
Req: 0x03 0x0F <elevation_msb> <elevation_lsb>	
Resp <ACK>	Response is an “acknowledgement” or <ACK>, a response data-packet with the length byte set to zero and no data bytes. The CMD_UPDATE command should be followed by the corresponding CMD_READ command to verify that the expected value was written.

WARMUP Command

CMD_WARM	Reset the sensor, which puts it in a known state, similar to power up. The sensor initializes itself, waits a period of time in warm-up mode, and then starts to measure gas PPM. The sensor attempts to send an <ACK>, but transmission may be aborted by the reset.
Req: 0x84	
Resp <ACK> or <no response>	The sensor experiences the same communications delay as at power-up.

CALIBRATION Command

CMD_SGPT_CALIBRATE	This command tells the sensor to start a single point calibration. Before sending this command, a reference gas should be flowing to the sensor.
Req: 0x9B	
Resp: <ACK>	The <ACK> response indicates that the calibration request has been received. To verify that calibration has started, wait at least one dsp cycle length and then send command CMD_STATUS to see if the calibration bit is set. When calibration is finished, the calibration bit in the status byte is cleared. A single point calibration will not start if a sensor is in warm-up mode or in error condition. See Communication Examples, below.
CMD_VFY_SGPT_PPM	This command asks the sensor to reply with the current single point set point. This is to verify before a single point calibration that the sensor is going to calibrate to the correct CO ₂ concentration.
Req: 0x02 0x11	
Resp: <data_bytes>	
CMD_SET_SGPT_PPM	Set/Write a single point ppm to the sensor for calibration. Set single point is expressed as a 2-byte binary value. Order of the bytes is <msb,lsb>.
Req: 0x03 0x11 <ppm_msb> <ppm_lsb>	
Resp <ACK>	Response is an “acknowledgement” or <ACK>, a response data-packet with the length byte set to zero and no data bytes. The CMD_SET_SGPT_PPM command should be followed by the corresponding CMD_VFY_SGPT_PPM command to verify that the expected value was written.

STATUS and OPERATING Commands

CMD_STATUS	Read a status byte from the sensor. The status byte indicates whether the sensor is functioning and is measuring PPM concentration.
Req:	0xB6
Resp:	<status> The response is a single byte, <status>, of bit flags. (Note: bit 0 is the least significant bit.) Bit 0: Error Bit 1: Warmup Mode Bit 2: Calibration Bit 3: Idle Mode Bits 4 - 6: (internal) Bit 7: Self Test Mode If a given status bit is “1”, the sensor is in that state or mode. If a status bit is “0”, the sensor is not in that mode.
CMD_IDLE_ON	This command tells the sensor to go into Idle Mode. In Idle Mode, the Lamp is turned off and no data collection takes place.
Req:	0xB9 0x01
Resp:	<ACK> Send the CMD_STATUS command to verify that the sensor has entered Idle Mode (status bit 3 = 1).
CMD_IDLE_OFF	This command tells the sensor to exit Idle Mode and resume data collection.
Req:	0xB9 0x02
Resp:	<ACK> The sensor resumes data collection as soon as the command is received. However, several data cycles (similar to Warm-Up) are required before PPM readings are accurate. Send the CMD_STATUS command to verify that the sensor has come out of Idle Mode (status bit 3 = 0)
CMD_ABC_LOGIC	This command queries the sensor for its ABC_LOGIC state.
Req:	0xB7 0x00
Resp:	<abc_state> If ABC_LOGIC is ON, <abc_state> = 0x01. If ABC_LOGIC is OFF, <abc_state> = 0x02.
CMD_ABC_LOGIC_ON	This command turns the ABC_LOGIC ON. The reply <0x01> indicates that the ABC_LOGIC has been turned on.
Req:	0xB7 0x01
Resp:	<0x01>
CMD_ABC_LOGIC_RESET	This command turns the ABC_LOGIC ON and resets the ABC_LOGIC to its startup state. The reply <0x01> indicates that the ABC_LOGIC has been turned on.
Req:	0xB7 0x03
Resp:	<0x01>
CMD_ABC_LOGIC_OFF	This command turns the ABC_LOGIC OFF. The reply <0x02> indicates that the ABC_LOGIC has been turned off.
Req:	0xB7 0x02
Resp:	<0x02>

TEST Commands

CMD_HALT	This command is used strictly for testing. It tells the sensor to put itself into error mode and act as though a fatal error has occurred. The sensor should automatically reset itself and go into Warmup Mode.
Req: 0x95	
Resp: <ACK>	See Communication Examples, below.
CMD_LOOPBACK	This command is used strictly for testing. The data_bytes (up to 16 bytes) following the 0x00 command are echoed back in the response packet.
Req: 0x00 <data_bytes>	
Resp: <data_bytes>	
CMD_SELF_TEST START	This command is used strictly for testing to start the internal Self Tests. While the Self Tests are in progress, CMD_STATUS will return a byte with bit 7 ON. The Self Tests last for 16 times the DSP sample rate.
Req: 0xC0 0x00	
Resp: <ACK>	
CMD_SELF_TEST RESULTS	This command is used strictly for testing to return the results of the most recent Self Tests. Results are valid only after CMD_STATUS has returned to 00 after a previous reply of 80.
Req: 0xC0 0x01	
Resp: <test flag><PGA status><# good dsp><total dsp>	<p><test flag> should be 0F for a completed Self Test <PGA status> is 01 for PASS, 00 for FAIL <# good dsp> is the number of dsp cycles that have good data (should be 0C) <total dsp> is the total number of dsp cycles that were tested (should be 0C)</p>

STREAM DATA Command

Upon power-up, some sensor models start streaming gas concentration data out the UART. For other sensor models, data streaming occurs only when the command CMD_STREAM_DATA is given. In either case, the data stream is either two or three bytes, depending upon the sensor model. If any UART command is sent to a sensor while streaming data, the data streaming is stopped. In order to resume data streaming, command CMD_STREAM_DATA should be given. If command CMD_STREAM_DATA is given while a sensor is streaming data, it will stop and then restart streaming data.

CMD_STREAM_DATA	
Req: 0xBD	Start streaming gas concentration data after each data collection cycle.
Resp: <data bytes>	The response data stream is either 2 or 3 bytes, depending upon sensor model. If two bytes are returned, the format is <msb, lsb>. For models returning three bytes, the data is the actual ppm in the format <msb, mid, lsb >. In all formats, the data is non-negative and bounded.

Miscellaneous Commands

Different models in the CO₂ sensor may support sensor specific commands. Consult the manufacturer for details.

UART Communication Examples

The following examples illustrate request and response packets with the UART Tsunami-Lite Communication Protocol. Requests and responses are expressed in hexadecimal bytes. The <command> portion of a request and the <response_data> are in bold type.

UART Read Gas PPM

Req> FF FE 02 02 03	Req: In the request “02 03” is CMD_READ GAS_PPM (see Command Reference, above.)
Resp> FF FA 02 02 50	Resp: The response is “02 50” (since 592 PPM = 0x0250) For models that require the result be multiplied by 16, the actual PPM would be 592 * 16 = 9472.

UART CMD_STATUS to Verify Normal Operation

Req> FF FE 01 B6	Req: In the request, “B6” is CMD_STATUS (see Command Reference, above.)
Resp> FF FA 01 00	Resp: In the response, “00” is the status byte. The zero value indicates that the sensor is in normal mode where it is measuring gas PPM. It is not in warm-up mode, it is not in calibration mode, and it is not in an error condition. Further examples of CMD_STATUS are given in the examples below.

UART Read and Update Elevation

In this set of interchanges you first read the sensor’s elevation parameter and find it is set at 1000 ft. Then you change the elevation setting to 2500 ft. Then you read back the new elevation setting and verify that it is set to 2500 ft.

Req 1> FF FE 02 02 0F	Req 1: In request 1, “02 0F” is CMD_READ, ELEVATION (see Command Reference, above.)
Resp1> FF FA 02 03 E8	Resp 1: In the first response, “03 E8” is the elevation, 1000 ft (1000 = 0x03E8).
Req 2> FF FE 04 03 0F 09 C4	Req 2: In request 2, “03 0F” is CMD_UPDATE, ELEVATION, and “09 C4” is the elevation, 2500 ft (2500 = 0x09C4).
Resp2> FF FA 00	Resp 2: The second response is an <ACK>, since the length is 0x00.
Req 3> FF FE 02 02 0F	Req 3: The third request and response are formatted just like the first, reading back the new elevation setting, 2500 ft.
Resp3> FF FA 02 09 C4	

UART Error Simulation with Recovery

In this set of interchanges you first verify that the sensor is operating normally. Then you send a command that forces the sensor into an error state. The sensor automatically recovers by resetting itself, and going into Warmup Mode. You then send the command to skip warm-up, thus putting the sensor back into the normal state.

Req 1> FF FE 01 B6	Req 1: CMD_STATUS.
Resp1> FF FA 01 00	Resp1: status byte is 0x00. Sensor is in normal mode, measuring gas PPM.
Req 2> FF FE 01 95	Req 2: CMD_HALT. Puts sensor in error mode.
Resp2> FF FA 00	Resp 2: <ACK>
Req 3> FF FE 01 B6	Req 3: CMD_STATUS.
Resp3> FF FA 01 02	Resp3: status byte is 0x02. Bit 1 high indicates sensor is in warm-up mode. (If CMD_STATUS is sent quickly enough, the sensor may respond with 0x01, indicating the brief error state prior to reset.) Wait several seconds.
Req 4> FF FE 01 B6	Req 4: CMD_STATUS
Resp4> FF FA 01 00	Resp 4: status byte is 0x00. sensor is in normal mode, measuring gas PPM.

UART Single Point Calibration

In this set of interchanges you run a single point calibration on the sensor. Before sending any commands, you start by flowing a premixed stable concentration of CO₂ gas to the sensor. Then you verify that the sensor is in normal operating mode, since calibration will not work if the sensor is not in normal operating mode. Then you send the Set Single Point PPM command with the concentration of the calibration gas, and then verify with the Verify Single Point PPM command. Then the Single Point Calibration command is sent. You check the sensor's status and see that it is in calibration mode. Later you check the status again and see that the sensor has finished calibration and returned to normal operating mode.

Req 1> FF FE 01 B6	Req 1: CMD_STATUS.
Resp1> FF FA 01 00	Resp1: status byte is 0x00. Sensor is in normal mode, measuring gas PPM.
Req 2> FF FE 04 03 11 02 58	Req 2: CMD_SET_SGPT_PPM. Set the single point calibration to 600 ppm (0x0258 = 600)
Resp2> FF FA 00	Resp 2: <ACK>
Req3> FF FE 02 02 11	Req 3: CMD_VFY_SGPT_PPM. Request sensor to output current single point ppm set point.
Resp3> FF FA 02 02 58	Resp 3: sensor responds with current single point ppm, currently 600 ppm.
Req 4> FF FE 01 9B	Req 4: CMD_SGPT_CALIBRATE. Starts the calibration process.
Resp4> FF FA 00	Resp4: <ACK> Wait 2 – 4 seconds.
Req 5> FF FE 01 B6	Req 5: CMD_STATUS.
Resp5> FF FA 01 04	Resp 5: status byte is 0x04. Sensor is in calibration mode. Wait 15 seconds and repeat Req 4 at intervals of 15 seconds until sensor is out of calibration (return byte is 0x00).
Req 6> FF FE 01 B6	Req 6: CMD_STATUS.
Resp6> FF FA 01 00	Resp 6: status byte is 0x00. Sensor is in normal mode, measuring gas PPM.

Appendix A. Summary of Commands

CMD_READ Commands

Command	Request	Response
CMD_READ	0x02 <data ID>	<data>, [... <data>]
CMD_READ CO₂_PPM	0x02 0x03	<ppm_msb> <ppm_lsb>
CMD_READ SERIAL_NUMBER	0x02 0x01	[ASCII, 15 bytes, null padded]
CMD_READ COMPILE_SUBVOL	0x02 0x0D	[3-byte ASCII string]
CMD_READ COMPILE_DATE	0x02 0x0C	[6-byte ASCII string]
CMD_READ ELEVATION	0x02 0x0F	<elev_lsb> <elev_msb>

CMD_UPDATE Commands

Command	Request	Response
CMD_UPDATE ELEVATION	0x03 0x0F <elev_msb> <elev_lsb>	<ACK>

WARMUP Command

Command	Request	Response
CMD_WARM	0x84	<ACK> or <no response>

CALIBRATION Command

Command	Request	Response
CMD_SGPT_CALIBRATE	0x9B	<ACK>
CMD_SET_SGPT_PPM	0x03 0x11 <ppm_msb> <ppm_lsb>	<ACK>
CMD_VFY_SGPT_PPM	0x02 0x11	<ppm_msb> <ppm_lsb>

STATUS and OPERATING Commands

Command	Request	Response
CMD_STATUS	0xB6	<status>
CMD_IDLE_ON	0xB9 0x01	<ACK>
CMD_IDLE_OFF	0xB9 0x02	<ACK>
CMD_ABC_LOGIC	0xB7 0x00	<abc_state>
CMD_ABC_LOGIC_ON	0xB7 0x01	<0x01>
CMD_ABC_LOGIC_RESET	0xB7 0x03	<0x01>
CMD_ABC_LOGIC_OFF	0xB7 0x02	<0x02>

TEST Commands

Command	Request	Response
CMD_HALT	0x95	<ACK>
CMD_LOOPBACK	0x00 <data_bytes>	<data_bytes>
CMD_SELF_TEST START	0xC0 0x00	<ACK>
CMD_SELF_TEST RESULTS	0xC0 0x01	<test flag><PGA status> <# good dsp><total dsp>

CMD_STREAM_DATA

Command	Request	Response
CMD_STREAM_DATA	0xBD	<data_bytes>

Warranty Repairs

Amphenol Thermometrics, Inc. will repair Telaire product that fails to meet the terms provided for in the Return and Warranty Policy Statement (*See, <http://www.amphenol-sensors.com>*). Warranty period shall start from date of manufacture and be based on product category and type of equipment as specified in Table 1: Product Warranty Periods. For all warranty repairs, Amphenol Thermometrics, Inc. will bear all product repair parts, labor, and standard ground shipping charges.

This product is covered by one or more of the following patents:

5,650,624/5,721,430/5,444,249/5,747,808/5,834,777/5,163,332/
5,340,986/5,502,308/6,344,798/6,023,069/5,370,114/5,601,079/
5,691,704/5,767,776/5,966,077/6,107,925/5,798,700/5,945,924/
5,592,147/6,255,653/6,250,133/6,285,290

Customer Support Centers

U.S.A.

Sales and Services

(Repair/Calibration):

Amphenol Thermometrics, Inc.
St Marys Center
967 Windfall Road
St Marys, Pennsylvania 15857
U.S.A.
T: +1 814-834-9140
F: +1 814-781-7969

U.K.

Sales and Service:

Amphenol Thermometrics (U.K.) Limited
Crown Industrial Estate Priorswood Road
Taunton, TA2 8QY, UK
T: +44 1823-335-200

Brazil

Sales and Service

Amphenol TFC DO Brazil LTDA
Rodovia Governador Adhemar
Pereira de Barros KM 121,5 S/N
13098-396 Campinas
Sao Paulo, Brazil

U.S.A.

Technical Support:

Amphenol Thermometrics, Inc.
St Marys Center
967 Windfall Road
St Marys, Pennsylvania 15857
U.S.A.
T: +1 814-834-9140
F: +1 814-781-7969

China:

Amphenol (Changzhou)
Connector Systems
305 Room, 5D
Jintong Industrial Park
Wujin, Changzhou, Jiangsu, China
T:+86 519 8831 8080 ext. 50087
F:+86 519 8831 2601

Amphenol

Advanced Sensors

www.amphenol-sensors.com

www.telaire.com