



## METER

# ATMOS 14 INTEGRATOR GUIDE

## SENSOR DESCRIPTION

The ATMOS 14 sensor is an accurate tool for monitoring vapor pressure, relative humidity (RH), temperature, and barometric pressure. For a more detailed description of how this sensor makes measurements, refer to the [ATMOS 14 Gen 2 User Manual](#).

## APPLICATIONS

- Greenhouse monitoring
- Canopy monitoring
- Reference evapotranspiration calculations
- Routine weather monitoring
- Building humidity monitoring
- Mold remediation
- Modeling processes that are affected by vapor pressure or humidity

## ADVANTAGES

- Fully calibrated relative humidity and temperature
- Three-wire sensor interface: power, ground, and data
- Digital sensor communicates multiple measurements over a serial interface
- Robust thermistor for accurate temperature measurements
- Low-input voltage requirements
- Low-power design supports battery-operated data loggers
- SDI-12 or DDI serial communications protocols supported
- Modern design optimized for low-cost sensing

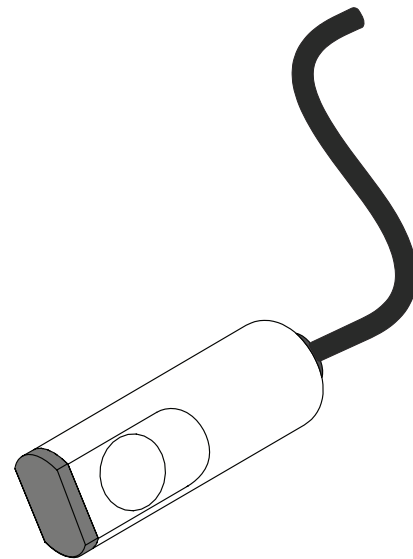


Figure 1 ATMOS 14 sensor

## PURPOSE OF THIS GUIDE

METER Group provides the information in this integrator guide to help ATMOS 14 customers establish communication between these sensors and their data acquisition equipment or field data loggers. Customers using data loggers that support SDI-12 sensor communications should consult the data logger user manual. This sensor is fully integrated into the METER system of plug-and-play sensors, cellular-enabled data loggers, and data analysis software. For more information about an integrated systems, please contact [Customer Support](#).

## COMPATIBLE FIRMWARE VERSIONS

This guide is compatible with firmware versions 2.00 or newer for the ATMOS 14 Gen 2.

# SPECIFICATIONS

## MEASUREMENT SPECIFICATIONS

Relative Humidity (RH)	
Range	0–100% RH (0.00–1.00)
Resolution	0.10% RH
Accuracy	Sensor measurement accuracy is variable across a range of RH. Refer to <a href="#">Figure 2</a> .

HUMIDITY (%RH)	TEMPERATURE (°C)									
	0	10	20	30	40	50	60	70	80	
100	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%
90	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%
80	±2.0%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%	±2.0%	±2.0%
70	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%	±2.0%
60	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%	±2.0%
50	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%	±2.0%
40	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%
30	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%
20	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%
10	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%
0	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%

Figure 2 RH accuracy

Equilibration Time ( $\tau$ , 60%)	<25 s (response time in 1 m/s air stream)
Hysteresis	±0.80% RH, typical
Long-Term Drift	±0.25% RH/year, typical

Temperature	
Range	–40 to 80 °C
Resolution	0.1 °C
Accuracy	±0.2 °C
Equilibration Time ( $\tau$ , 63%)	<165 s (response time in 1 m/s air stream)
Long-Term Drift	<0.03 °C/year, typical

Vapor Pressure	
Range	0–47 kPa
Resolution	0.01 kPa
Accuracy	Sensor measurement accuracy is variable across a range of temperatures and RH. Refer to <a href="#">Figure 3</a> .

HUMIDITY (%RH)	TEMPERATURE (°C)									
	0	10	20	30	40	50	60	70	80	
100	±0.03	±0.05	±0.09	±0.16	±0.27	±0.44	±0.69	±1.33	±2.38	
90	±0.03	±0.05	±0.09	±0.15	±0.26	±0.42	±0.66	±1.26	±2.24	
80	±0.03	±0.04	±0.07	±0.12	±0.21	±0.34	±0.63	±1.20	±2.10	
70	±0.02	±0.04	±0.07	±0.12	±0.20	±0.32	±0.50	±1.13	±1.96	
60	±0.02	±0.03	±0.06	±0.11	±0.18	±0.30	±0.47	±1.06	±1.82	
50	±0.02	±0.03	±0.06	±0.10	±0.17	±0.28	±0.45	±0.99	±1.68	
40	±0.02	±0.03	±0.05	±0.09	±0.16	±0.26	±0.42	±0.76	±1.54	
30	±0.01	±0.03	±0.05	±0.09	±0.15	±0.24	±0.39	±0.69	±1.40	
20	±0.01	±0.02	±0.04	±0.08	±0.14	±0.23	±0.36	±0.62	±1.26	
10	±0.01	±0.02	±0.04	±0.07	±0.12	±0.21	±0.33	±0.55	±1.13	
0	±0.01	±0.02	±0.04	±0.06	±0.11	±0.19	±0.30	±0.48	±0.99	

Figure 3 Vapor pressure accuracy

Barometric Pressure	
Range	1–120 kPa
Resolution	0.01 kPa
Accuracy	±0.05 kPa at 25 °C
Equilibration Time ( $\tau$ , 63%)	<10 ms
Long-Term Drift	<0.1 kPa/year, typical

## COMMUNICATION SPECIFICATIONS

Output	
	DDI Serial or SDI-12 communications protocol

Data Logger Compatibility	
	Any data acquisition system capable of 4.0- to 15.0-VDC power and serial or SDI-12 communications

## PHYSICAL SPECIFICATIONS

Dimensions	
Diameter	2.0 cm (0.8 in)
Height	5.4 cm (2.1 in)

Operating Temperature Range	
Minimum	–40 °C
Typical	NA
Maximum	+80 °C

NOTE: Sensors may be used at higher temperatures under certain conditions; contact [Customer Support](#) for assistance.

Cable Length	
	5 m (standard) 75 m (maximum custom cable length)

NOTE: Contact [Customer Support](#) if a nonstandard cable length is needed.

Cable Diameter	
	0.165 ±0.004 in (4.20 ±0.10 mm), with minimum jacket of 0.030 in (0.76 mm)

Connector Types	
	Stereo plug connector or stripped and tinned wires

Stereo Plug Connector Diameter	
	3.5-mm

Conductor Gauge	
	22-AWG / 24-AWG drain wire

## ELECTRICAL AND TIMING CHARACTERISTICS

Supply Voltage (VCC to GND)	
Minimum	4.0 VDC
Typical	NA
Maximum	15.0 VDC
Digital Input Voltage (logic high)	
Minimum	2.8 V
Typical	3.6 V
Maximum	5.0 V
Digital Input Voltage (logic low)	
Minimum	-0.3 V
Typical	0.0 V
Maximum	0.8 V
Digital Output Voltage (logic high)	
Minimum	NA
Typical	3.6 V
Maximum	NA
Power Line Slew Rate	
Minimum	1.0 V/ms
Typical	NA
Maximum	NA
Current Drain (during measurement)	
Minimum	1.2 mA
Typical	3.5 mA
Maximum	12.0 mA

Current Drain (while asleep)	
Minimum	NA
Typical	0.03 mA
Maximum	NA
Power-Up Time (DDI Serial)	
Minimum	NA
Typical	NA
Maximum	110 ms
Power-Up Time (SDI-12)	
Minimum	140 ms
Typical	260 ms
Maximum	400 ms
Power Up Time (SDI-12, DDI disabled)	
Minimum	NA
Typical	170 ms
Maximum	200 ms
Measurement Duration	
Minimum	NA
Typical	50 ms
Maximum	NA

COMPLIANCE	
Manufactured under ISO 9001:2015	
2004/108/EC and 2011/65/EU	
EN61326-1:2013	
EN50581:2012	

## EQUIVALENT CIRCUIT AND CONNECTION TYPES

Refer to [Figure 4](#) and [Figure 5](#) to connect the ATMOS 14 to a data logger. [Figure 4](#) provides a low-impedance variant of the recommended SDI-12 specification.

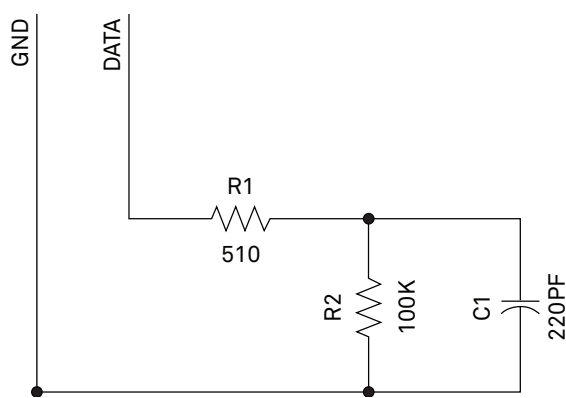
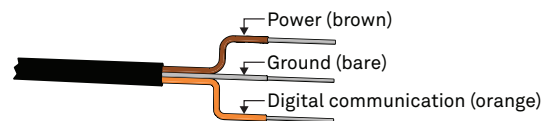


Figure 4 Equivalent circuit diagram

### PIGTAIL CABLE



### STEREO CABLE

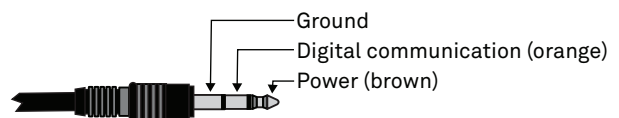


Figure 5 Connection types

## **⚠ PRECAUTIONS**

METER sensors are built to the highest standards, but misuse, improper protection, or improper installation may damage the sensor and possibly void the warranty. Before integrating sensors into a sensor network, follow the recommended installation instructions and implement safeguards to protect the sensor from damaging interference.

## **SURGE CONDITIONS**

Sensors have built-in circuitry that protects them against common surge conditions. Installations in lightning-prone areas, however, require special precautions, especially when sensors are connected to a well-grounded third-party logger.

Visit [metergroup.com](http://metergroup.com) for articles containing more information.

## **POWER AND GROUNDING**

Ensure there is sufficient power to simultaneously support the maximum sensor current drain for all the sensors on the bus. The sensor protection circuitry may be insufficient if the data logger is improperly powered or grounded. Refer to the data logger installation instructions. Improper grounding may affect the sensor output as well as sensor performance.

Visit [metergroup.com](http://metergroup.com) for articles containing more information.

## **CABLES**

Improperly protected cables can lead to severed cables or disconnected sensors. Cabling issues can be caused by many factors, including rodent damage, driving over sensor cables, tripping over the cable, not leaving enough cable slack during installation, or poor sensor wiring connections. To relieve strain on the connections and prevent loose cabling from being inadvertently snagged, gather and secure the cable travelling between the ATMOS 14 and the data acquisition device to the mounting mast in one or more places. Install cables in conduit or plastic cladding when near the ground to avoid rodent damage. Tie excess cable to the data logger mast to ensure cable weight does not cause sensor to unplug.

## **SENSOR COMMUNICATIONS**

METER digital sensors feature a serial interface with shared receive and transmit signals for communicating sensor measurements on the data wire (Figure 5). The sensor supports two different protocols: SDI-12 and DDI serial. Each protocol has implementation advantages and challenges. Please contact [Customer Support](#) if the protocol choice for the desired application is not obvious.

### **SDI-12 INTRODUCTION**

SDI-12 is a standards-based protocol for interfacing sensors to data loggers and data acquisition equipment. Multiple sensors with unique addresses can share a common 3-wire bus (power, ground, and data). Two-way communication between the sensor and logger is possible by sharing the data line for transmit and receive as defined by the standard. Sensor measurements are triggered by protocol command. The SDI-12 protocol requires a unique alphanumeric sensor address for each sensor on the bus so that a data logger can send commands to and receive readings from specific sensors.

Download the [SDI-12 Specification v1.3](#) to learn more about the SDI-12 protocol.

### **DDI SERIAL INTRODUCTION**

The DDI serial protocol is the method used by the METER data loggers for collecting data from the sensor. This protocol uses the data line configured to transmit data from the sensor to the receiver only (simplex). Typically, the receive side is a microprocessor UART or a general-purpose I/O pin using a bitbang method to receive data. Sensor measurements are triggered by applying power to the sensor.

### **INTERFACING THE SENSOR TO A COMPUTER**

The serial signals and protocols supported by the sensor require some type of interface hardware to be compatible with the serial port found on most computers (or USB-to-serial adapters). There are several SDI-12 interface adapters available in the marketplace; however, METER has not tested any of these interfaces and cannot make a recommendation as to which adapters work with METER sensors. METER data loggers and the ZSC and PROCHECK handheld devices can operate as a computer-to-sensor interface for making on-demand sensor measurements. For more information, please contact [Customer Support](#).

## METER SDI-12 IMPLEMENTATION

METER sensors use a low-impedance variant of the SDI-12 standard sensor circuit (Figure 4). During the power-up time, sensors output some sensor diagnostic information and should not be communicated with until the power-up time has passed. After the power up time, the sensors are fully compatible with all commands listed in the [SDI-12 Specification v1.3](#) except for the continuous measurement commands (aR0 and aRC0). These commands are implemented for convenience to customers integrating with systems not requiring strict SDI-12 compliance. If using the SDI-12 verifier, please contact [Customer Support](#) for instructions on suppressing the aR0! and aRC0! commands. M, C, and R command implementations are found on pages 7–8. The aXR3 and aXR4 extended commands are used by METER systems and as a result use a space delimiter, instead of a sign delimiter as required by SDI-12.

Out of the factory, all METER sensors start with SDI-12 address 0 and print out the DDI serial startup string during the power-up time. This can be interpreted by non-METER SDI-12 sensors as a pseudo-break condition followed by a random series of bits.

The ATMOS 14 will omit the DDI serial startup string when the SDI-12 address is nonzero or if `<suppress ionState>` is set to 1. Changing the address to a nonzero address is recommended for this reason.

## SENSOR BUS CONSIDERATIONS

SDI-12 sensor buses require regular checking, sensor upkeep, and sensor troubleshooting. If one sensor goes down, that may take down the whole bus even if the remaining sensors are functioning normally. Power cycling the SDI-12 bus when a sensor is failing is acceptable, but METER does not recommend scheduling power cycling events on an SDI-12 bus more than once or twice per day. Many factors influence the effectiveness of the bus configuration. Visit [metergroup.com](http://metergroup.com) for articles and virtual seminars containing more information.

## SENSOR ERROR CODES

The ATMOS 14 Gen 2 has three error codes:

- -9999 is output in place of the measured value if the sensor detects that the measurement function has been compromised and the subsequent measurement values have no meaning
- -9992 is output in place of the measured value if the sensor detects corrupt or lost calibrations
- -9991 is output in place of the measured value if the sensor detects insufficient voltage to perform the measurement

## SDI-12 CONFIGURATION

Table 1 lists the SDI-12 communications configuration.

Baud Rate	1,200
Start Bits	1
Data Bits	7 (LSB first)
Parity Bits	1 (even)
Stop Bits	1
Logic	Inverted (active low)

## SDI-12 TIMING

All SDI-12 commands and responses must adhere to the format in Figure 6 on the data line. Both the command and response are preceded by an address and terminated by a carriage return and line feed combination (`<CR><LF>`) and follow the timing shown in Figure 7.

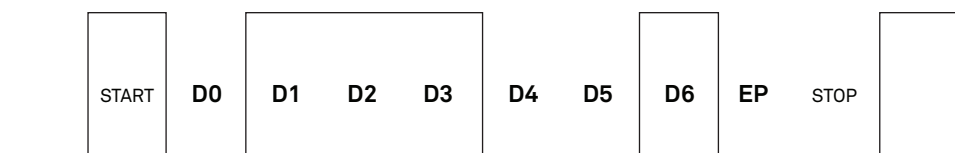


Figure 6 Example SDI-12 transmission of the character 1 (0x31)

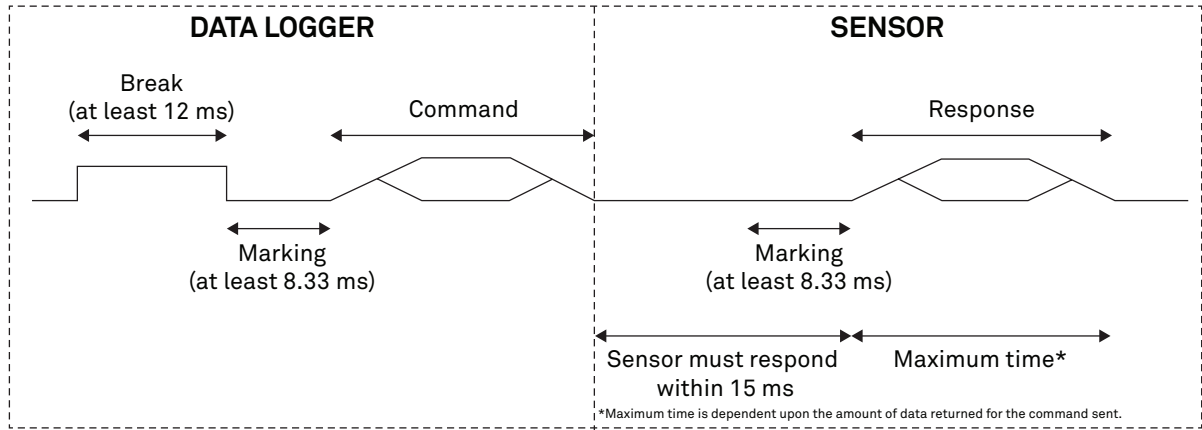


Figure 7 Example data logger and sensor communication

### COMMON SDI-12 COMMANDS

This section includes tables of common SDI-12 commands that are often used in an SDI-12 system and the corresponding responses from METER sensors.

#### IDENTIFICATION COMMAND (aI!)

The Identification command can be used to obtain a variety of detailed information about the connected sensor. An example of the command and response is shown in Example 1, where the command is in **bold** and the response follows the command.

**Example 1** **1I!113METER\_ \_ \_ ATM14\_ \_ 100631800001**

Parameter	Fixed Character Length	Description
<b>1I!</b>	3	Data logger command. Request to the sensor for information from sensor address <b>1</b> .
<b>1</b>	1	Sensor address. Prepended on all responses, this indicates which sensor on the bus is returning the following information.
<b>13</b>	2	Indicates that the target sensor supports <a href="#">SDI-12 Specification v1.3</a> .
<b>METER_ _ _ _</b>	8	Vendor identification string. (METER and three spaces <b>_ _ _</b> )
<b>ATM14_ _</b>	6	Sensor model string. This string is specific to the sensor type. For the ATMOS 14, the string is <b>ATM14_ _</b> .
<b>100</b>	3	Sensor version. This number divided by 100 is the METER sensor version (e.g., 100 is version 1.00).
<b>631800001</b>	≤13, variable	Sensor serial number. This is a variable length field. It may be omitted for older sensors.

#### CHANGE ADDRESS COMMAND (aAB!)

The Change Address command is used to change the sensor address to a new address. All other commands support the wildcard character as the target sensor address except for this command. All METER sensors have a default address of 0 (zero) out of the factory. Supported addresses are alphanumeric (i.e., a–z, A–Z, and 0–9). An example output from a METER sensor is shown in Example 2, where the command is in **bold** and the response follows the command.

### Example 2 1A0!0

Parameter	Fixed Character Length	Description
1A0!	4	Data logger command. Request to the sensor to change its address from 1 to a new address of 0 (zero).
0	1	New sensor address. For all subsequent commands, this new address will be used by the target sensor.

### ADDRESS QUERY COMMAND (?!)

While disconnected from a bus, the Address Query command can be used to determine which sensor is currently being communicated with. Sending this command over a bus will cause a bus contention where all the sensors will respond simultaneously and corrupt the data line. This command is helpful when trying to isolate a failed sensor. Example 3 shows an example of the command and response, where the command is in **bold** and the response follows the command. The question mark (?) is a wildcard character that can be used in place of the address with any command except the Change Address command.

### Example 3 ?!0

Parameter	Fixed Character Length	Description
?!	2	Data logger command. Request for a response from any sensor listening on the data line.
0	1	Sensor address. Returns the sensor address to the currently connected sensor.

## COMMAND IMPLEMENTATION

The following tables list the relevant Measurement (M), Concurrent (C), Verification (V), Extended (X), and Continuous (R) commands and subsequent Data (D) commands, when necessary.

### MEASUREMENT COMMAND IMPLEMENTATION

Measurement (M) commands are sent to a single sensor on the SDI-12 bus and require that subsequent Data (D) commands are sent to that sensor to retrieve the sensor output data before initiating communication with another sensor on the bus.

Please refer to Table 2 and for an explanation of the command sequence and see Table 9 for an explanation of response parameters.

Table 2 aM! measurement command sequence

Command	Response
This command reports instantaneous values.	
aM!	atttn
aD0!	a+<vaporPressure>±<temperature>+<relativeHumidity>±<atmosphericPressure>

NOTE: The measurement and corresponding data commands are intended to be used back to back. After a measurement command is processed by the sensor, a service request a <CR><LF> is sent from the sensor signaling the measurement is ready. Either wait until ttt seconds have passed or wait until the service request is received before sending the data commands. See the SDI-12 Specifications v1.3 document for more information.

### CONCURRENT MEASUREMENT COMMANDS IMPLEMENTATION

Concurrent (C) measurement commands are typically used with sensors connected to a bus. Concurrent (C) commands for this sensor deviate from the standard C command implementation. First, send the C command, wait the specified amount of time detailed in the C command response, and then use D commands to read its response prior to communicating with another sensor.

Please refer to Table 3 for an explanation of the command sequence and see Table 9 for an explanation of response parameters.

**Table 3 aC! measurement command sequence**

Command	Response
This command reports instantaneous values.	
aC!	attnn
aD0!	a+<vaporPressure>±<temperature>+<relativeHumidity>±<atmosphericPressure>

NOTE: This command does not adhere to the SDI-12 concurrent command reset requirements.

### VERIFICATION MEASUREMENT COMMANDS IMPLEMENTATION

The Verification (V) command is intended to give users a means to determine information about the current state of the sensor. First, the V command is sent followed by D commands to read the response.

**Table 4 aV! measurement command sequence**

Command	Response
This command reports instantaneous values.	
aV!	attnn
aD0!	a+<meta>

### EXTENDED COMMANDS IMPLEMENTATION

Extended (X) commands provide sensors with a means of performing manufacturer-specific functions. Additionally, X commands are utilized by METER systems and use a different response format than standard SDI-12 commands. X commands are required to be prefixed with the address and terminated with an exclamation point. Responses are required to be prefixed with the address and terminated with <CR><LF>.

METER implements the following X commands: aXRx! to trigger a sensor measurement and return the data automatically after the readings are completed without needing to send additional commands and aX0! (capital 0 as in Oscar) to suppress the DDI string.

Please refer to Table 5 through Table 7 for an explanation of the command sequence and Table 9 for an explanation of response parameters.

**Table 5 aX0! measurement command sequence**

Command	Response
aX0!	a<suppressionState>
aX0!<suppressionState>	a0K

NOTE: Command uses capital 0 as in Oscar (not a zero).

**Table 6 aXR3! measurement command sequence**

Command	Response
aXR3!	a<TAB><vaporPressure> <temperature> <atmosphericPressure><CR><sensorType><Checksum><CRC>

**Table 7 aXR4! measurement command sequence**

Command	Response
aXR4!	a<TAB><vaporPressure> <temperature> <atmosphericPressure><CR><sensorType><Checksum><CRC>

### CONTINUOUS MEASUREMENT COMMANDS IMPLEMENTATION

Continuous (R) commands trigger a sensor measurement and return the data automatically after the readings are completed without needing to send a D command.

Please refer to Table 8 for an explanation of the command sequence and see Table 9 for an explanation of response parameters.



**Table 8 aR0! measurement command sequence**

Command	Response
	This command reports instantaneous values.
aR0!	a+<vaporPressure>±<temperature>+<relativeHumidity>±<atmosphericPressure>

NOTE: This command does not adhere to the SDI-12 response timing. See [METER SDI-12 Implementation](#) for more information.

**PARAMETERS**

Table 9 lists the parameters, unit measurement, and a description of the parameters returned in command responses for ATMOS 14.

**Table 9 Parameter Descriptions**

Parameter	Unit	Description
<vaporPressure>	kPa	Vapor pressure
<temperature>	°C	Temperature
<relativeHumidity>	—	Relative humidity as a dimensionless fraction between 0.00 and 1.00
<atmosphericPressure>	kPa	Atmospheric pressure
<meta>	—	Auxiliary sensor information. See <a href="#">Table 10</a> .
<suppressionState>	—	0: DDI unsuppressed 1: DDI suppressed
<CR>	—	Carriage return character
<LF>	—	Line feed character
<sensorType>	—	123 or { for ATMOS 14
<TAB>	—	Tab character
<Checksum>	—	METER serial checksum
<CRC>	—	METER 6-bit CRC
±	—	Positive or negative sign denoting sign of the next value
a	—	SDI-12 address
n	—	Number of measurements (fixed width of 1)
nn	—	Number of measurements with leading zero if necessary (fixed width of 2)
ttt	s	Maximum time measurement will take (fixed width of 3)

**SENSOR METADATA VALUE**

The sensor metadata value contains information to help alert users to sensor-identified conditions that may compromise optimal sensor operation. The output of the aV! aD0! sequence will output a <meta> integer value. This integer represents a binary bitfield, with each individual bit representing an error flag.

Table 10 lists the possible error flags that can be set by the ATMOS 14. If multiple error flags are set, the sensor metadata integer value will be the sum of the individual values. To decode an integer value not explicitly in Table 10, find the largest error flag value that will fit in the integer value and accept that error as being present. Then, subtract that error flag value from the integer value and repeat the process on the remainder until the result is zero. For example, a sensor metadata integer value of 384 is the sum of the individual error flag values 256 + 128, so this sensor has corrupt firmware and a corrupt or lost sensor calibration.

**Table 10 Error flag values and issue resolution**

Error Flag Value	Issue Present	Resolution
128	Sensor firmware is corrupt	Contact <a href="#">Customer Support</a> for instructions on reloading firmware
256	Sensor calibrations lost or corrupted	Contact <a href="#">Customer Support</a> for instructions on reloading sensor calibrations

## DDI SERIAL COMMUNICATION

The DDI serial communications protocol is ideal for systems that have dedicated serial signaling lines for each sensor or use a multiplexer to handle multiple sensors. The serial communications are compatible with many TTL serial implementations that support active-high logic levels using 0–3.6 V signal levels. When the sensor is first powered, it automatically makes measurements of the integrated transducers then outputs a response over the data line. Systems using this protocol control the sensor excitation to initiate data transfers from the sensor. This protocol is subject to change as METER improves and expands the line of digital sensors and data loggers. ATMOS 14 will omit the DDI serial startup string when the SDI-12 address is nonzero.

**NOTE:** Out of the factory, all METER sensors start with SDI-12 address zero and print out the startup string when power cycled.

## DDI SERIAL TIMING

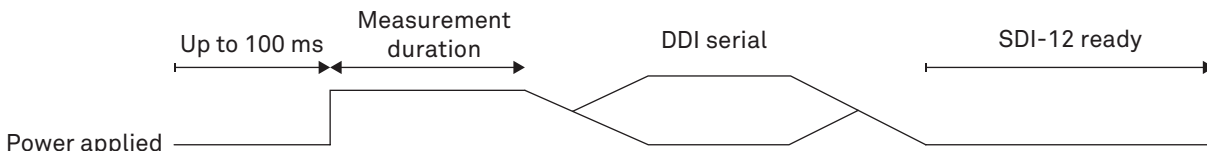
[Table 11](#) lists the DDI serial communications configuration.

**Table 11 DDI serial communications configuration**

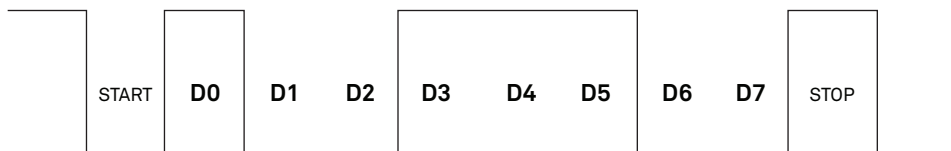
Baud Rate	1,200
Start Bits	1
Data Bits	8 (LSB first)
Parity Bits	0 (none)
Stop Bits	1
Logic	Standard (active high)

At power up, the sensor will pull the data line high within 100 ms to indicate that the sensor is taking a reading ([Figure 8](#)). When the reading is complete, the sensor begins sending the serial signal out the data line adhering to the format shown in [Figure 9](#). Once the data is transmitted, the sensor goes into SDI-12 communication mode. To get another serial signal, the sensor must be power cycled.

**NOTE:** Sometimes the signaling from the sensor can confuse typical microprocessor UARTs. The sensor holds the data line low while taking measurements. The sensor raises the line high to signal the logger that it will send a measurement. Then the sensor may take some additional measurements before starting to clock out the first data byte starting with a typical start bit (low). Once the first start bit is sent, typical serial timing is valid; however, the signal transitions before this point are not serial signaling and may be misinterpreted by the UART.



**Figure 8 Data line DDI serial timing**



**Figure 9 Example DDI serial transmission of the character 9 (0x39)**

## DDI SERIAL RESPONSE

Table 12 details the DDI serial response.

Table 12 DDI serial response

Command	Response
NA	<TAB><vaporPressure> <temperature> <atmosphericPressure><CR><sensorType><Checksum><CRC>

NOTE: There is no actual command. The response is returned automatically upon power up.

The values in this command are space delimited. As such a + sign is not assigned between values and a - sign is only present if the value is negative.

## DDI SERIAL CHECKSUM

These checksums are used in the continuous commands R3 and R4 as well as the DDI serial response. The legacy checksum is computed from the start of the transmission to the sensor identification character, excluding the sensor address.

Legacy checksum example input is <TAB>1.222 23.4 92.81<CR>{/6 and the resulting checksum output is /.

```
uint8_t LegacyChecksum(const char * response)
{
    uint16_t length;
    uint16_t i;
    uint16_t sum = 0;

    // Finding the length of the response string
    length = strlen(response);

    // Adding characters in the response together
    for(i = 0; i < length; i++)
    {
        sum += response[i];
        if(response[i] == '\r')
        {
            // Found the beginning of the metadata section of the response
            break;
        }
    }

    // Include the sensor type into the checksum
    sum += response[++i];

    // Convert checksum to a printable character
    sum = sum % 64 + 32;

    return sum;
}
```

The more robust CRC6, if available, utilizes the CRC-6-CDMA2000-A polynomial with the value 48 added to the results to make this a printable character and is computed from the start of the transmission to the legacy checksum character.

CRC6 checksum example input is `<TAB>1.222 23.4 92.81<CR>{/6` and the resulting checksum output is `6`.

```
uint8_t CRC6_Offset(const char *buffer)
{
    uint16_t byte;
    uint16_t i;
    uint16_t bytes;
    uint8_t bit;
    uint8_t crc = 0xfc; // Set upper 6 bits to 1's

    // Calculate total message length—updated once the metadata section is found
    bytes = strlen(buffer);

    // Loop through all the bytes in the buffer
    for(byte = 0; byte < bytes; byte++)
    {
        // Get the next byte in the buffer and XOR it with the crc
        crc ^= buffer[byte];

        // Loop through all the bits in the current byte
        for(bit = 8; bit > 0; bit--)
        {
            // If the uppermost bit is a 1...
            if(crc & 0x80)
            {
                // Shift to the next bit and XOR it with a polynomial
                crc = (crc << 1) ^ 0x9c;
            }
            else
            {
                // Shift to the next bit
                crc = crc << 1;
            }
        }
        if(buffer[byte] == '\r')
        {
            // Found the beginning of the metadata section of the response
            // both sensor type and legacy checksum are part of the crc6
            // this requires only two more iterations of the loop so reset
            // "bytes"

            // bytes is incremented at the beginning of the loop, so 3 is added
            bytes = byte + 3;
        }
    }

    // Shift upper 6 bits down for crc
    crc = (crc >> 2);

    // Add 48 to shift crc to printable character avoiding \r \n and !
    return (crc + 48);
}
```

## CUSTOMER SUPPORT

### NORTH AMERICA

Customer service representatives are available for questions, problems, or feedback Monday through Friday, 7:00 am to 5:00 pm Pacific time.

**Email:** [support.environment@metergroup.com](mailto:support.environment@metergroup.com)  
[sales.environment@metergroup.com](mailto:sales.environment@metergroup.com)

**Phone:** +1.509.332.5600

**Fax:** +1.509.332.5158

**Website:** [metergroup.com](http://metergroup.com)

### EUROPE

Customer service representatives are available for questions, problems, or feedback Monday through Friday, 8:00 to 17:00 Central European time.

**Email:** [support.europe@metergroup.com](mailto:support.europe@metergroup.com)  
[sales.europe@metergroup.com](mailto:sales.europe@metergroup.com)

**Phone:** +49 89 12 66 52 0

**Fax:** +49 89 12 66 52 20

**Website:** [metergroup.de](http://metergroup.de)

If contacting METER by email, please include the following information:

Name	Email address
Address	Instrument serial number
Phone number	Description of problem

**NOTE:** For products purchased through a distributor, please contact the distributor directly for assistance.

## REVISION HISTORY

The following table lists document revisions.

Revision	Date	Compatible Firmware	Description
01	01.2022	2.00	Stereo connector spec updates, Legacy Checksum corrections, Tables updates & basic template updates.
00	3.19.2021	2.00	Original release.