18036-01
10.2024

METER

ATMOS 41 GEN 2 INTEGRATOR GUIDE

SENSOR DESCRIPTION

The ATMOS 41 Gen 2 All-in-One Weather Station is designed for continuous monitoring of environmental variables, including all standard weather measurements (see Measurement Specifications). All sensors are integrated into a single unit, requiring minimal installation effort. Ultra-low power consumption and a robust design that prevents errors because of wear or fouling make the ATMOS 41 Gen 2 ideal for long-term, remote installations.

APPLICATIONS

- Weather monitoring
- Microenvironment monitoring
- Spatially distributed environmental monitoring
- Crop weather monitoring
- Fire danger monitoring/mapping
- Weather networks

ADVANTAGES

- Robust, minimal maintenance design
- Small form factor
- Integrated design for easy installation
- Low-input voltage requirements
- Low-power design supports battery-operated data loggers
- Supports the SDI-12 three-wire interface and Modbus RTU
- Tilt sensor informs user of out-of-level conditions
- No configuration necessary
- Measures all standard weather variables (plus several others)

PURPOSE OF THIS GUIDE

METER Group provides the information in this integrator's guide to help ATMOS 41 Gen 2 All-in-One Weather Station 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. METER sensors are fully integrated into the METER system of plug-and-play sensors, cellular-enabled data loggers, and data analysis software.

COMPATIBLE FIRMWARE VERSIONS

This guide is compatible with firmware versions 6.08 or newer.

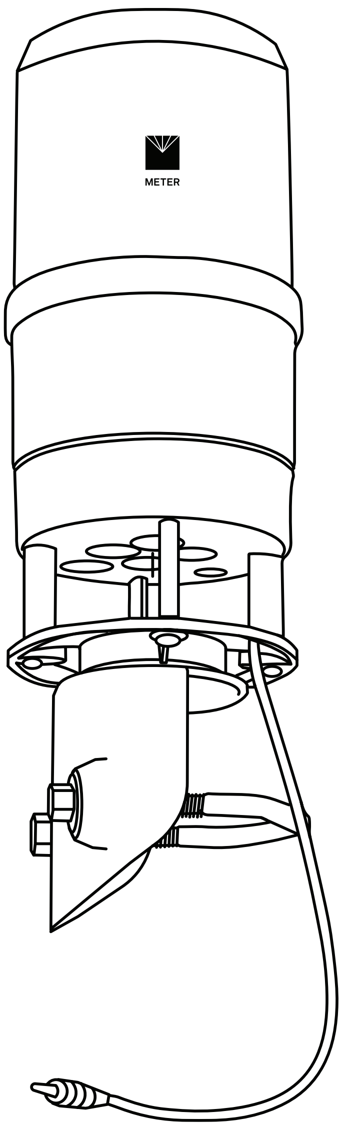

Figure 1 ATMOS 41 Gen 2 All-in-One Weather Station

METER Group, Inc.
2365 NE Hopkins Court, Pullman, WA 99163
T +1.509.332.2756 **F** +1.509.332.5158
E info@metergroup.com **W** metergroup.com

SPECIFICATIONS

MEASUREMENT SPECIFICATIONS

Solar Radiation

Range	0–1750 W/m²
Resolution	1 W/m²
Accuracy	±5% of measurement typical

Precipitation

Range	0–2000 mm/h
Resolution	0.017 mm
Accuracy	±5% of measurement from 0 to 1000 mm/h

Electrical Conductivity

Range	0–3 mS/cm
Resolution	0.001 mS/cm
Accuracy	The greater of 0.005 mS/cm or 15% of measurement

Vapor Pressure

Range	0–47 kPa
Resolution	0.01 kPa
Accuracy	Varies with temperature and humidity, ±0.2 kPa typical below 40 °C

HUMIDITY (%RH)	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

TEMPERATURE (°C)

Relative Humidity

Range	0–100% RH (0.00–1.00)
Resolution	0.1% RH
Accuracy	Varies with temperature and humidity, ±3% RH typical

HUMIDITY (%RH)	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%
90	±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%
70	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%
60	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%
50	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%	±2.0%
40	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%
30	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%
20	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%
10	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%
0	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±1.5%	±2.0%

TEMPERATURE (°C)

Hysteresis	±0.80% RH, typical
Long-Term Drift	±0.25% RH/year, typical

Air Temperature

Range	−65 to 60 °C
Resolution	0.1 °C
Accuracy	±.2 °C at 25 °C ±.6 °C at -20 to 50 °C

Humidity Sensor Temperature

Range	−40 to 80 °C
Resolution	0.1 °C
Accuracy	±.2 °C

Barometric Pressure

Range	1-120 kPa
Resolution	0.01 kPa
Accuracy	±0.05 kPa at 25 ºC
Equilibration Time (τ, 63%)	<10 ms
Long-Term Drift	<0.1 kPa/year, typical

Horizontal Wind Speed

Range	0–60 m/s
Resolution	0.01 m/s
Accuracy	The greater of 0.3 m/s or 3% of measurement

Wind Gust

Range	0–60 m/s
Resolution	0.01 m/s
Accuracy	The greater of 0.3 m/s or 3% of measurement

Wind Direction

Range	0°–359.9°
Resolution	0.1°
Accuracy	±1°

Tilt

Range	−90° to 90° or 0° to 180° (single orientation)
Resolution	0.1°
Accuracy	±1°

Lightning Strike Count

Range	0–65,535 strikes
Resolution	1 strike
Accuracy	Variable with distance, >25% detection at <10 km typical

Lightning Average Distance	
Range	0–40 km
Resolution	3 km
Accuracy	Variable

COMMUNICATION SPECIFICATIONS

Output

SDI-12 communication or
Modbus RTU (on enabled untits)

Data Logger Compatibility

METER ZL6 and EM60 data loggers or any data aquisition systems capable of switched 3.6- to 15.0-VDC excitation and SDI-12 communication or Modbus RTU (on enabled units)

PHYSICAL SPECIFICATIONS

Dimensions	
Diameter	10 cm (3.9 in)
Height	28 cm (11.0 in), includes rain gauge filter

Operating Temperature Range	
Minimum	–50 °C
Typical	NA
Maximum	60 °C

NOTE: Barometric pressure and relative humidity sensors operate accurately at a minimum of –40 °C.

Cable Length

5 m (stereo plug)

1.5 m (5-pin)

75 m (maximum custom cable length for additional cost)

NOTE: Contact Customer Support if 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 3 stripped and tinned wires

5-pin M12 connector or 4 stripped and tinned wires

Connector Diameter

3.5 mm (diameter stereo plug)

14.4 mm (diameter M12)

Conductor Gauge

22-AWG / 24-AWG drain wire

ELECTRICAL AND TIMING CHARACTERISTICS

Supply Voltage (VCC to GND)	
Minimum	3.6 VDC continuous
Typical	NA
Maximum	25.0 VDC continuous

NOTE: ATMOS 41 must be continuously powered to work properly.

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

NOTE: For the ATMOS 41 to meet digital logic levels specified by SDI-12, it must be excited to 3.9 VDC or greater.

Power Line Slew Rate	
Minimum	1.0 V/ms
Typical	NA
Maximum	NA

Current Drain (during measurement)	
Minimum	0.2 mA
Typical	8.0 mA
Maximum	33.0 mA

Current Drain (while asleep, SDI-12 only)	
Minimum	0.2 mA
Typical	0.3 mA
Maximum	0.4 mA

Current Drain (while asleep, Modbus enabled)	
Minimum	2 mA
Typical	2.5 mA
Maximum	3.5 mA

Power Up Time (SDI ready)—aRx! Commands	
Minimum	NA
Typical	5 s
Maximum	NA
Power Up Time (SDI ready)—Other Commands	
Minimum	NA
Typical	310 ms
Maximum	NA
Power Up Time (SDI-12, DDI disabled)	
Minimum	NA
Typical	240 ms maximum
Maximum	1000 ms
Measurement Duration	
Minimum	NA
Typical	110 ms
Maximum	3,000 ms

COMPLIANCE

EM ISO/IEC 17050:2010 (CE Mark)

EQUIVALENT CIRCUIT AND CONNECTION TYPES

The following sections explain the ATMOS 41 Gen 2 connection types available. ATMOS 41 Gen 2 units are either SDI-12 only or Modbus RTU enabled. Units that are SDI-12 only are shipped with a stereo plug connector or a three-wire pigtail cable. Units that are Modbus RTU enabled are capable of SDI-12 communication in addition to Modbus RTU, and are shipped with a 5-pin M12 connector or four-wire pigtail cable. The hardware version of the ATMOS 41 Gen 2 can also be determined from the serial number of the unit: units with a serial number format of A41G2S0000001 are SDI12 only, and units with serial number format A41G2M0000001 are Modbus RTU enabled.

SDI- 12 ONLY VERSION

Refer to Figure 2 and Figure 3 to connect the ATMOS 41 Gen 2 to a logger. Figure 2 provides a low-impedance variant of the recommended SDI-12 specification.

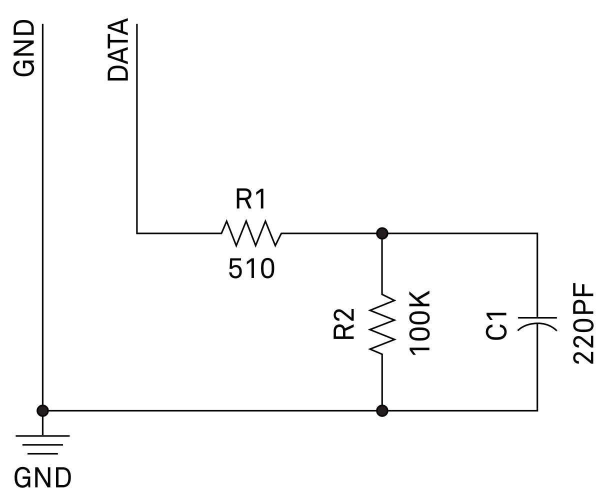

Figure 2 Equivalent circuit diagram

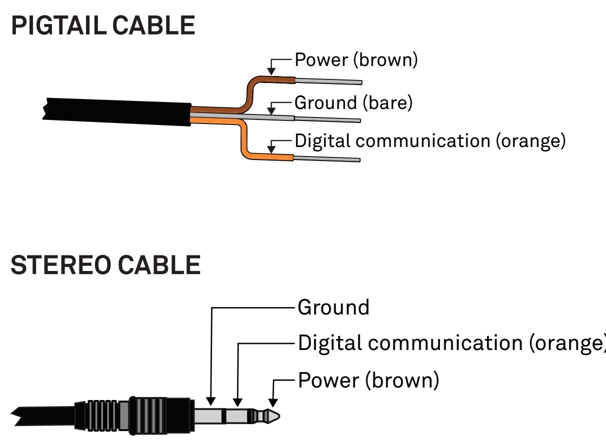

Figure 3 Connection types

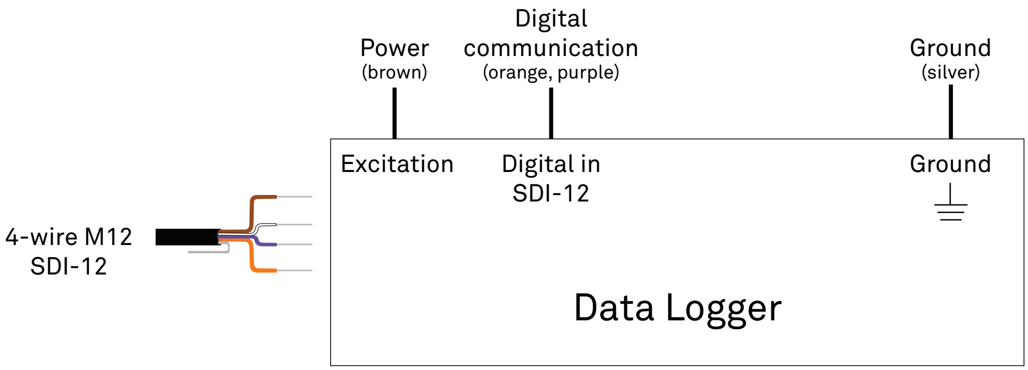

Figure 4 **Four wire connector SDI-12 wiring diagram**

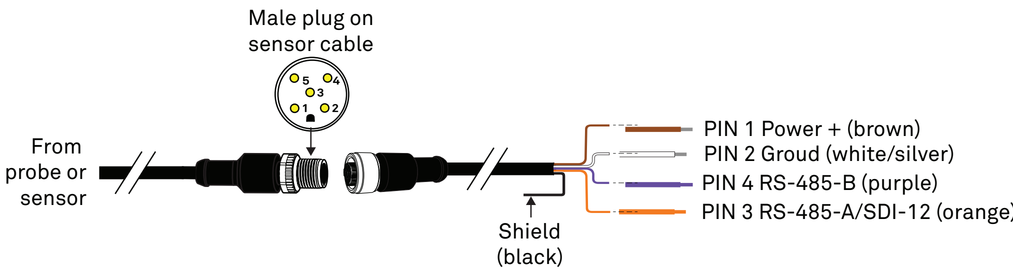

Figure 5 **Four-wire M12 connector and pigtail adapter for use with screw terminals**

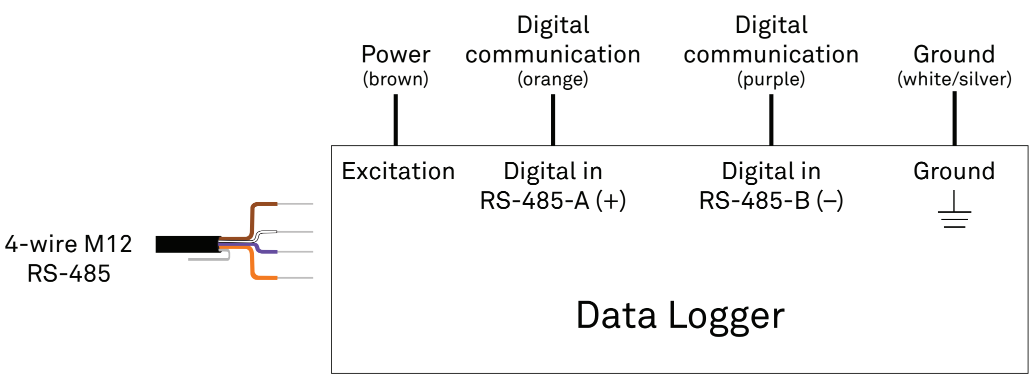

Figure 6 **Four wire M12 connector RS-485 wiring diagram**

⚠ 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 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 41 Gen 2 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 3). The sensor supports two different protocols: SDI-12, and Modbus RTU (on enabled units). A third protocol, the METER DDI Serial Protocol, is used in a limited form (see below). 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 and learn more about the SDI-12 protocol.

MODBUS RTU RS-485 INTRODUCTION (ON ENABLED UNITS)

Modbus RTU is a common serial communications protocol used by Programmable Logic Controllers (PLCs) or data loggers to communicate with all kinds of digital devices. The communication works over the physical RS-485 connection. The combination of RS-485 for the physical connection and Modbus as serial communications protocol allows fast and reliable data transfer for a high number of sensors connected to one serial bus wire. Use the following links for more Modbus information: Wikipedia and modbus.org.

DDI SERIAL INTRODUCTION

The DDI serial protocol is the method used by the METER family of 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 IO pin using a bitbang method to receive data. Sensor measurements are triggered by applying power to the sensor. When the ATMOS 41 Gen 2 is set to address `0`, a DDI serial string is sent on power up, identifying the sensor. the ATMOS 41 Gen 2 does not report sensor measurements using the DDI serial protocol. The ATMOS 41 Gen 2 DDI serial message only contains information to identify the sensor to METER data loggers.

INTERFACING THE SENSOR TO A PC

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 personal 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 device can operate as a computer-to-sensor interface for making on-demand sensor measurements. PROCHECK's SDI-12 passthrough mode may be used to communicate with the sensor, but spot measurements and data logging are not supported in PROCHECK. 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 2). 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 compatible with all commands listed in the SDI-12 Specification v1.3 except for the continuous measurement commands (`aR0`–`aR9` and `aRC0`–`aRC9`) and the concurrent measurement commands. `M`, `R`, and `C` command implementations are found on pages 9–11. 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 41 Gen 2 will omit the DDI serial startup string (sensor identification) when the SDI-12 address is nonzero or if or if `<suppressionState>` is set to 1 (See Common SDI-12 Commands).

ATMOS 41 GEN 2 INTERNAL MEASUREMENT SEQUENCE

Upon power up, the ATMOS 41 Gen 2 initializes an internal timer to 57. This internal timer is incremented by 1 every second and resets to `0` after incrementing to `59`. In addition, issuing an averaging command (`aM!`, `aR0!`, `aR3!`, `aR7!`, , `C3!`, `C4!` and `aC!`) resets this timer to `58`.

While powered up, the ATMOS 41 Gen 2 continuously counts drops from the precipitation sensor and takes solar radiation, wind, and air temperature measurements every 3 s and logs these values internally. Vapor pressure, atmospheric pressure, and relative humidity are measured every 10 s at internal timer intervals of 0, 10, 20, 30, 40, 50 and are logged internally. Orientation, vapor pressure, atmospheric pressure, and relative humidity are measured every 60 s at the internal timer interval of 4 and logged internally. The `aR4!` command will output instantaneous measurements of these parameters.

The `aM!`, `aR0!`, `aR3!`, `aR7!`, `aC!`, `C3!`, and `C4!` commands (and subsequent `D` commands when necessary) will compute and output the averages, accumulations, or maximums of these measurements (and derived measurements) and reset internal averaging counters and accumulators. Therefore, it is not necessary to oversample the ATMOS 41 Gen 2 and compute averages, accumulations, and maximums in external data systems. Less frequent sampling has the additional benefit of decreasing data acquisition systems and ATMOS 41 power consumption. If the `aM!`, `aR0!`, `aR3!`, `aR7!`, `C3!`, `C4!` and `aC!` commands are issued more frequently than 2 times their measurement interval, the ATMOS 41 Gen 2 will not average the measurements and will output instantaneous values.

SENSOR ERROR CODES

The ATMOS 41 Gen 2 has four error codes available:

- `-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
- `-9990` is output in place of the measured value if the sensor detects a temporary issue with one of its measurement values (e.g. rain detected on the sonic anemometer transducers). The sensor will resume outputting the measured value as normal once it detects the issue is no longer occurring.

SDI-12 CONFIGURATION

Table 1 lists the SDI-12 communication configuration.

Table 1 SDI-12 communication 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 7 on the data line. Both the command and response are preceded by an address and terminated by a carriage return line feed combination and follow the timing shown in Figure 8.

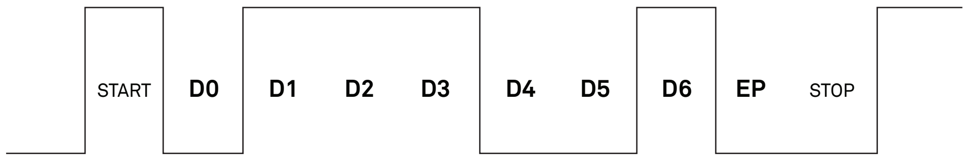

Figure 7 Example SDI-12 transmission of the character 1 (`0x31`)

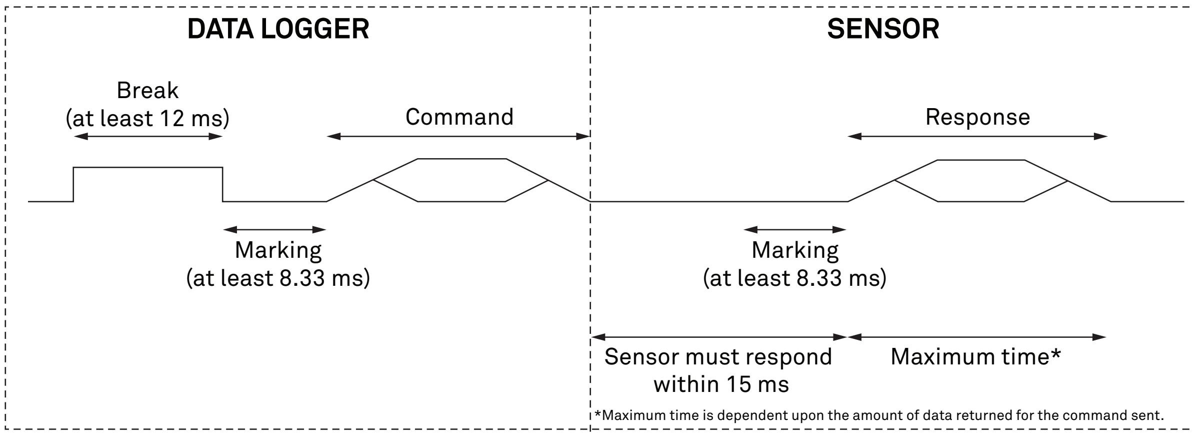

Figure 8 Example data logger and sensor communication

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␣␣␣AT41G2␣608A41G2S0001234`

Parameter	Fixed Character Length	Description
`1I!`	3	Data logger command Request to the sensor for information from sensor address `1`.
`1`	1	Sensor address Prepended on all responses, this indicates which sensor on the bus is returning the following information.
`13`	2	Indicates that the target sensor supports SDI-12 Specification v1.3
`METER␣␣␣`	8	Vendor identification string (`METER` and three spaces `␣␣␣` for all METER sensors)
`AT41G2`	6	Sensor model string This string is specific to the sensor type. For the ATMOS 41, the string is `AT41G2`.
`608`	3	Sensor version This number divided by 100 is the METER sensor version (e.g., 608 is version 6.08).
`A41G2S0001234`	≤13, variable	Sensor serial number This is a variable length field.

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.
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 sensors are 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), Continuous (R), Concurrent (C), and Verification (V) commands and subsequent Data (D) commands when necessary.

MEASUREMENT COMMANDS 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, Table 3, and Table 4 for an explanation of the command sequence and see Table 18 for an explanation of response parameters.

Table 2 aM! command sequence

Command	Response
This command reports average, accumulated, or maximum values. Please see ATMOS 41 Gen 2 Internal Measurement Sequence for more details.	
aM!	atttn
aD0!	a+<solar>+<precipitation>+<strikes>
aD1!	a+<windSpeed>+<windDirection>+<gustWindSpeed>
aD2!	a±<airTemperature>+<vaporPressure>+<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 METER SDI-12 Implementation document for more information.

Table 3 aM1! command sequence

Command	Response
This command reports instantaneous values.	
aM1!	atttn
aD0!	a±<xOrientation>±<yOrientation>+<nullValue>

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.

Table 4 aM3! command sequence

Command	Response
This command reports instantaneous values.	
aM3!	atttn
aD0!	a+<strikeDistance>+<relativeHumidity>>±<humiditySensorTemperature>
aD1!	a±<xOrientation>±<yOrientation>+<nullValue>
aD2!	a±<NorthWindSpeed>±<EastWindSpeed>

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.

CONTINUOUS MEASUREMENT COMMANDS IMPLEMENTATION

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

After an R command is issued, the ATMOS 41 Gen 2 will take new measurements of all its available subsensors within 2 seconds after the end of the R command response. If it is desired to "oversample" the ATMOS 41 Gen 2's measurement parameters, it is recommended to issue R commands at a rate of 2 seconds or greater.

If the aR8! command is issued after the aR0! or aR7! commands, the measurement data for both commands will be reported from the same measurement interval. This is intended to allow for the retrieval of measurements new to the ATMOS 41 Gen 2 via the aR8! command while adhering to the response time and response length called out in the SDI-12 Specification v1.3. Any averaging M, R, or C commands issued after aR8! will reset the ATMOS 41 Gen 2's internal accumulators as normal.

aR0!, aR3!, and aR4! return more characters in their responses than the 75-character limitation called out in the SDI-12 Specification v1.3. It is recommended to use a buffer that can store at least 116 characters.

Please refer to Table 5 through Table 9 for an explanation of the command sequence and see Table 18 for an explanation of response parameters.

Table 5 `aR0!` measurement command sequence

Command	Response
This command reports average, accumulated, or maximum values. Please see ATMOS 41 Gen 2 Internal Measurement Sequence for more details.	
`aR0!`	`a+<solar>+<precipitation>+<strikes>+<strikeDistance>+<windSpeed>+<windDirection>+<gustWindSpeed>±<airTemperature>+<vaporPressure>+<atmosphericPressure>+<relativeHumidity>±<humiditySensorTemperature>±<xOrientation>±<yOrientation>+<nullValue>±<NorthWindSpeed>±<EastWindSpeed>`
NOTE: This command does not adhere to the SDI-12 response format. See METER SDI-12 Implementation for more information.	

Table 6 `aR3!` measurement command sequence

Command	Response
This command reports average, accumulated, or maximum values. Please see ATMOS 41 Gen 2 Internal Measurement Sequence for more details.	
`aR3!`	`a<TAB><solar> <precipitation> <strikes> <strikeDistance> <NorthWindSpeed> <EastWindSpeed> <gustWindSpeed> <airTemperature> <vaporPressure> <atmosphericPressure> <xOrientation> <yOrientation> <nullValue> <humiditySensorTemperature><CR><sensortype><Checksum><CRC>`
NOTE: This command does not adhere to the SDI-12 response format. See METER SDI-12 Implementation for more information. 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.	

Table 7 `aR4!` measurement command sequence

Command	Response
This command reports instantaneous values.	
`aR4!`	`a<TAB><solar> <precipitation> <strikes> <strikeDistance> <NorthWindSpeed> <EastWindSpeed> <gustWindSpeed> <airTemperature> <vaporPressure> <atmosphericPressure> <xOrientation> <yOrientation> <nullValue> <humiditySensorTemperature><CR><sensortype><Checksum><CRC>`
NOTE: This command does not adhere to the SDI-12 response. See METER SDI-12 Implementation for more information. 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.	

Table 8 `aR7!` measurement command sequence

Command	Response
This command reports average, accumulated, or maximum values. Please see ATMOS 41 Gen 2 Internal Measurement Sequence for more details.	
`aR7!`	`a+<solar>+<precipitation>+<strikes>+<strikeDistance>+<windSpeed>+<windDirection>+<gustWindSpeed>±<airTemperature>+<vaporPressure>+<atmosphericPressure>+<relativeHumidity>±<humiditySensorTemperature>±<xOrientation>±<yOrientation>`
NOTE: See METER SDI-12 Implementation for more information.	

Table 9 aR8! measurement command sequence

Command	Response
This command reports average, accumulated, or maximum values. Please see ATMOS 41 Gen 2 Internal Measurement Sequence for more details.	
aR8!	a+<precipDropCount>+<precipTipCount>+<precipitationEC>+<singleOrientation>±<Tmin>±<Tmax>

NOTE: See METER SDI-12 Implementation for more information.

CONCURRENT MEASUREMENT COMMANDS IMPLEMENTATION

Concurrent (C) measurement commands are typically used with sensors connected to a bus. Measurements are initiated with a C command and subsequent D commands are sent to the sensor to retrieve the readings.

Please refer to Table 2,Table 10, and Table 11 for an explanation of the command sequence and see Table 18 for an explanation of response parameters.

Table 10 aC! measurement command sequence

Command	Response
This command reports average, accumulated, or maximum values. Please see ATMOS 41 Gen 2 Internal Measurement Sequence for more details.	
aC!	atttnn
aD0!	a+<solar>+<precipitation>+<strikes>+<strikeDistance>
aD1!	a+<windSpeed>+<windDirection>+<gustWindSpeed>
aD2!	a±<airTemperature>+<vaporPressure>+<atmosphericPressure>+<relativeHumidity>±<humiditySensorTemperature>
aD3!	a±<xOrientation>±<yOrientation>+<nullValue>
aD4!	a±<NorthWindSpeed>±<EastWindSpeed>+<gustWindSpeed>

NOTE: Please see the METER SDI document for more information.

Table 11 aC3! measurement command sequence

Command	Response
This command reports average, accumulated, or maximum values. Please see ATMOS 41 Gen 2 Internal Measurement Sequence for more details.	
aC3!	atttnn
aD0!	a+<solar>+<precipitation>+<precipDropCounts>+<precipTipCounts>+<precipitationEC>
aD1!	a+<strikes>+<strikeDistance>+<windSpeed>+<windDirection>+<gustWindSpeed>
aD2!	a±<airTemperature>+<vaporPressure>+<atmosphericPressure>+<relativeHumidity>±<humiditySensorTemperature>
aD3!	a±<xOrientation>±<yOrientation>±<Tmin>±<Tmax>

NOTE: Please see the METER SDI document for more information.

Table 12 **aC4!** measurement command sequence

Command	Response
This command reports average, accumulated, or maximum values. Please see ATMOS 41 Gen 2 Internal Measurement Sequence for more details.	
aC4!	atttnn
aD0!	a+<solar>+<precipitation>+<precipDropCounts>+<precipTipCounts>+<precipitationEC>
aD1!	a+<strikes>+<strikeDistance>+<windSpeed>+<windDirection>+<gustWindSpeed>
aD2!	a±<airTemperature>+<vaporPressure>+<atmosphericPressure>+<relativeHumidity>±<humiditySensorTemperature>
aD3!	a+<singleOrientation>±<Tmin>±<Tmax>±<NorthWindSpeed>±<EastWindSpeed>

NOTE: Please see the METER SDI document for more information.

VERIFICATION COMMAND IMPLEMENTATION

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

Please refer to Table 13 for an explanation of the command sequence and Table 18 for an explanation of those response parameters.

Table 13 **aV!** measurement command sequence

Command	Response
aV!	atttnn
aD0!	a+<meta>

NOTE: Please see the METER SDI document for more information.

EXTENDED COMMAND IMPLEMENTATION

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

• aXO! (with capital O as in Oscar) to suppress the DDI Serial string.

Please refer toTable 14-Table 16 for an explanation of the aXRx! command sequence and seeTable 18 for an explanation of response parameters. The aXRx! commands do not adhere to the 75-character SDI-12 response length limit. It is recommended to use a buffer that can store 140 characters for aXR0! and 125 characters for aXR3! and aXR4!.

Table 14 **aXR0!** measurement command sequence

Command	Response
aXR0!	a+<solar>+<precipitation>+<strikes>+<strikeDistance>+<windSpeed>+<windDirection>+<gustWindSpeed>±<airTemperature>+<vaporPressure>+<atmosphericPressure>+<relativeHumidity>±<humiditySensorTemperature>±<xOrientation>±<yOrientation>+<nullValue>±<NorthWindSpeed>±<EastWindSpeed>+<precipDropCount>+<precipTipCount>+<precipitationEC>+<singleOrientation>[+-]<Tmin>[+-]<Tmax><CR><LF>

NOTE: Please see the METER SDI document for more information.

Table 15 `aXR3!` measurement command sequence

Command	Response
aXR3!	a<TAB><solarRadiation> <precipitation> <precipDropCount> <precipTipCount> <precipitationEC> <strikeCount> <strikeDistance> <northWindSpeed> <eastWindSpeed> <gustWindSpeed> <airTemperature> <vaporPressure> <atmosphericPressure> <singleOrientation> <Tmin> <Tmax> <rhSensorTemperature><CR><sensorType><legacyChecksum><CRC6><CR><LF>

NOTE: Please see the METER SDI document for more information.

Table 16 `aXR4!` measurement command sequence

Command	Response
aXR4!	a<TAB><solarRadiation> <precipitation> <precipDropCount> <precipTipCount> <precipitationEC> <strikeCount> <strikeDistance> <northWindSpeed> <eastWindSpeed> <gustWindSpeed> <airTemperature> <vaporPressure> <atmosphericPressure> <singleOrientation> <Tmin> <Tmax> <rhSensorTemperature><CR><sensorType><legacyChecksum><CRC6><CR><LF>

NOTE: This command reports instantaneous values. The typical duration from the end of this command to the beginning of the command response is 275 ms. Please allow up to 350 ms for the sensor to respond to this command. This command does not adhere to the SDI-12 response format. Please see the METER SDI document for more information.

Table 17 `aX0!` measurement command sequence

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

PARAMETERS

Table 18 lists the parameters, unit measurement, and a description of the parameters returned in command responses for ATMOS 41 Gen 2.

Table 18 Parameter Descriptions

Parameter	Unit	Description
±	—	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)
<TAB>	—	Tab character
<CR>	—	Carriage return character
<LF>	—	Line feed character
<solar>	W/m²	Solar radiation (average since the last measurement or instantaneous value depending on SDI-12 command used)
<precipitation>	mm	Rainfall since the last measurement
<precipDropCount>		Number of rain drops detected since last measurement
<precipTipCount>		Number of tipping spoon tips detected since last measurement
<precipitationEC>	uS/cm	Bulk electrical conductivity of precipitation basin contents (average since the last measurement; value will read 0 if there were 0 tipping spoon tip events since last measurement)

Table 18 Parameter Descriptions (continued)

Parameter	Unit	Description
<strikes>	—	Number of lightning strikes detected since last measurement
<strikeDistance>	km	Average strike distance from sensor since last measurement
<NorthWindSpeed>	m/s	Wind speed from the northerly direction (negative values denote southerly direction) (average since the last measurement or instantaneous value depending on SDI-12 command used)
<EastWindSpeed>	m/s	Wind speed from the easterly direction (negative values denote westerly direction) (average since the last measurement or instantaneous value depending on SDI-12 command used)
<windSpeed>	m/s	Combined wind speed magnitude of the <NorthWindSpeed> and <EastWindSpeed> (average since the last measurement or instantaneous value depending on SDI-12 command used)
<gustWindSpeed>	m/s	Maximum measured <windSpeed> since the last measurement
<windDirection>	°	Wind heading clockwise from north reference (average since the last measurement or instantaneous value depending on SDI-12 command used)
<airTemperature>	°C	Air temperature (average since the last measurement or instantaneous value depending on SDI-12 command used)
<vaporPressure>	kPa	Vapor pressure (average since the last measurement or instantaneous value depending on SDI-12 command used)
<atmosphericPressure>	kPa	Atmospheric pressure (average since the last measurement or instantaneous value depending on SDI-12 command used)
<relativeHumidity>	RH	Relative humidity as a dimensionless fraction computed with either average or instantaneous values of <vaporPressure> and <airTemperature>, depending on SDI-12 command used
<humiditySensor Temperature>	°C	Internal temperature measured with the relative humidity sensor (average since the last measurement or instantaneous value depending on SDI-12 command used)
<xOrientation>	°	X orientation angle (0 is level) (average since the last measurement or instantaneous value depending on SDI-12 command used)
<yOrientation>	°	Y orientation angle (0 is level) (average since the last measurement or instantaneous value depending on SDI-12 command used)
<Tmax>	°C	Highest <airTemperature> seen since the last measurement.
<Tmin>	°C	Lowest <airTemperature> seen since the last measurement.
<nullValue>	—	This parameter is reported as 0. ATMOS 41 Gen 1 sensors reported a compass heading prior to firmware version 4.65. ATMOS 41 Gen 2 sensors continue to output this parameter to enable drop-in compatibility with data acquisition systems configured to read ATMOS 41 Gen 1 parameters.
<meta>	—	Auxiliary sensor information. See Table 19.
<sensortype>	—	ASCII character denoting the sensor type For ATMOS 41 Gen 2, the character is 'X' (ASCII decimal value 88)
<suppressionState>	—	0: DDI Serial string unsuppressed 1: DDI Serial string suppressed
<Checksum>	—	METER serial checksum
<CRC>	—	METER serial 6-bit CRC

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 19 lists the possible error flags that can be set by the ATMOS 41. 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 19, 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 208 is the sum of the individual error flag values 128 + 64 + 16, so this sensor sensor secondary temperature measurement error flag, sensor firmware corrupt error flag, and the sensor misorientation error flag.

Table 19 Error flag values and issue resolution

Error Flag Value	Issue Present	Resolution
0	No issue present	–
16	Sensor misorientation error will likely affect readings	Use the ZENTRA Utility app to reorient the X orientation or Y orientation of the sensor.
64	Sensor thermistor is broken and sensor is using a backup measurement	Contact Customer Support. Irreversible sensor damage is likely.
128	Sensor firmware is corrupt	Contact Customer Support for instructions on reloading firmware.
256	Sensor calibrations lost or corrupted	Contact Customer Support for instructions on reloading sensor calibrations.
512	Discrepancy detected between rain electrode and tipping spoon data	Contact Customer Support for instructions on reloading sensor calibrations.

DDI SERIAL CHECKSUM

These checksums are used in the continuous commands `R3`, `R4`, `XR3`, and `XR4` as well as DDI serial response. The legacy checksum is computed from the start of the transmission to the sensor identification character.

Legacy checksum example input is `<TAB>0 0.000 1 1 0.22 0.21 0.30 24.3 1.26 92.74 –1.5 –4.0 0 24.4<CR>]Ah` and the resulting checksum output is `A`.

```
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`, supported in firmware version 4.61 or newer, 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>0 0.000 1 1 0.22 0.21 0.30 24.3 1.26 92.74 –1.5 –4.0 0 24.4<CR>]Ah` and the resulting checksum is the character `h`.

```
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);
}
```

METER MODBUS RTU SERIAL IMPLEMENTATION

Modbus over Serial Line is specified in two versions—ASCII and RTU. Modbus enabled ATMOS 41 Gen 2 sensors communicate using RTU mode exclusively. The following explanation is always related to RTU.

Table 20 Modbus communication configuration

Baud Rate	9,600
Start Bits	1
Data Bits	8 (LSB first)

Table 20 Modbus communication configuration

Parity Bits	1 (even)
Stop Bits	1
Logic	Standard (active high)

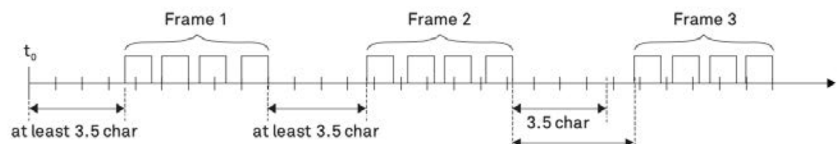

Figure 9 Modbus message

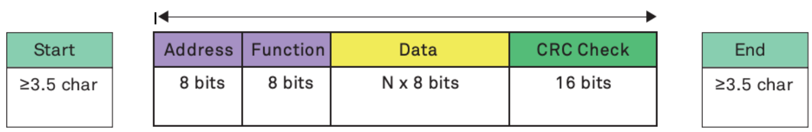

Figure 10 Modbus RTU message frame

A message in Modbus RTU format is shown in Figure 9. The length of the message is determined by the size of the data. By default, the format of each byte in the message has 11 bits, including the Start and Stop bit. Each byte is sent from left to right: Least Significant Bit (LSB) to Most Significant Bit (MSB). If no parity is implemented, an additional Stop bit is transmitted to fill out the character frame to a full 11-bit asynchronous character.

From the factory, ATMOS 41 Gen 2s use a Modbus server address of 1, even parity, and one stop bit. The device address, parity bit, and stop bits may be configured using Modbus RTU requests.

The Modbus application layer implements a set of standard function codes that are divided into three categories—public, user-defined, and reserved. This document covers ATMOS 41 Gen 2 sensor-supported public functions that are well-defined function codes documented in the Modbus Organization, Inc. (modbus.org) community.

SUPPORTED MODBUS FUNCTIONS

Table 21 Modbus Function Definitions

Function Code	Action	Description
01	Read coil/port status	Reads the on/off status of discrete output(s) in the Modbus server
02	Read input status	Reads the on/off status of discrete input(s) in the Modbus server
03	Read holding registers	Reads the binary contents of holding register(s) in the Modbus server
04	Read input registers	Reads the binary contents of input register(s) in the Modbus server
05	Write single coil/port	Sets a single coil/port in the Modbus server to either on or off
06	Write single register	Writes a value into a holding register in the Modbus server
15	Write multiple coils/ports	Sets multiple coils/ports in the Modbus server to either on or off
16	Write multiple registers	Writes values into a series of holding registers in the Modbus server

DATA REPRESENTATION AND REGISTERS

Data values (setpoint values, parameters, sensor specific measurement values, etc.) sent to and from the ATMOS 41 Gen 2 sensors using 16-bit holding (or input) registers. The registers use the standard 4- or 5-digit address notation specified in the Modbus Application Protocol v1.1b3 and do not use the Modicon or Modbus Enron register conventions.

16-bit register values are in big-endian order as called out in the Modbus standard. 32-bit values such as IEEE754 single-precision floating point or 32-bit unsigned values will span two 16-bit registers. Table 22 below shows the default byte ordering (big endian) for an example 32-bit value.

Table 22 32-bit Value Byte Order Example

Example Value: 123456.0 (0x471F2000)			
Register n		Register n+1	
Most Significant Byte 0x47	Least Significant Byte 0x1F	Most Significant Byte 0x20	Least Significant Byte 0x00

Register Mapping lists the Holding and Input register map for the ATMOS 41 Gen 2 sensor. The registers are listed with their 1-based register number; the register's actual address (i.e. the address that appears in a Modbus RTU request PDU) is the register number decremented by 1. The ATMOS 41 Gen 2 does not use coils for sensor data or actions and Modbus requests with public function codes 01, 05, or 15 will result in the Modbus exception code 01 (Illegal Function).

REGISTER MAPPING

Table 23 lists the Holding registers available on the ATMOS 41 Gen 2. These Holding registers may be read or written to retrieve or configure (respectively) the sensor's Modbus RTU settings. These settings are stored in sensor nonvolatile memory and will be maintained even after the sensor is power cycled or reset.

Any configuration updates take place immediately after the response to the initial Modbus client Holding register request. For example, after changing the server address, the sensor will first confirm the Holding register request using the old server address; all further Modbus requests to the sensor should then be sent using the sensor's new server address.

Table 23 Holding Registers

4401	**Modbus Server Address**
Detailed description	Read or update the sensor Modbus address
Data type	16-bit unsigned integer, Big-Endian
Allowed Range	1–247
Unit	—
Comments	—
4402	**Modbus Serial Baud Rate**
Detailed description	Read or update the sensor Modbus serial baud rate.
Data type	16-bit unsigned integer, Big-Endian
Allowed Range	0
Unit	—
Comments	0 is 9600 baud. 19200 baud is not currently supported.
4403	**Modbus Parity Bit**
Detailed description	Read or update the parity bit format used in each Modbus RTU byte.
Data type	16-bit unsigned integer, Big-Endian
Allowed Range	0-2
Unit	—
Comments	0 is no parity, 1 is odd parity, and 2 is even parity.

Table 23 (continued)

4404	Modbus Stop Bit Count
Detailed description	Read or update the amount of stop bits used in each Modbus RTU byte.
Data type	16-bit unsigned integer, Big-Endian
Allowed Range	1–2
Unit	—
Comments	—

Table 24 lists the Input registers containing ATMOS 41 Gen 2 sensor measurements. When these registers are accessed, the ATMOS 41 Gen 2 will compute and output the averages, accumulations, or maximums of its internal measurement accumulators (and derived measurements) and reset these internal averaging counters and accumulators.

The measurement values listed are all 32-bit values that span two 16-bit registers. For example, to access solar radiation starting at register number 3001, the Read Input Register request (function code `04`) should access register 3001 and register 3002. Both registers are required for the measurement value to be valid; attempting to access an odd number of Input registers in the range 3001 to 3044 will result in Modbus exception code `02` (Illegal Data Address).

NOTE: When the ATMOS 41 Gen 2 returns measurements in response to a Read input register request, the sensor's internal counters and accumulators for all measurements are reset, even if not all sensor measurement registers are accessed. For this reason, it is recommended to "burst read" all desired measurement values using one Read Input Register request.

Table 24 Input Registers

3001-3002	Solar Radiation
Detailed description	Average solar radiation since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0.0 to 1750.0
Unit	W/m^2
Comments	—
3003-3004	**Total Precipitation**
Detailed description	Total combined rainfall from dual rain gauges since the last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0.000 to 2000.000
Unit	mm
Comments	—
3005-3006	**Precipitation Drop Counter**
Detailed description	Total rain drops detected by drop counter since the last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0 to 65535
Unit	—
Comments	—
3007-3008	**Precipitation Tip Counter**
Detailed description	Total precipitation tipping spoon tips since the last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0 to 65535
Unit	—
Comments	—

Table 24 Input Registers (continued)

3009-3010	**Precipitation Electrical Conductivity**
Detailed description	Average precipitation basin bulk EC since the last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0 to 3000
Unit	uS/cm
Comments	If no tipping spoon tips were detected since the last measurement, EC will read as 0.
3011-3012	**Lightning Strike Count**
Detailed description	Total lightning strike count since the last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0 to 65535
Unit	—
Comments	—
3013-3014	**Average Lightning Strike Distance**
Detailed description	Average strike distance from sensor since the last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0 to 40
Unit	km
Comments	If no lightning strikes were detected since the last measurement, distance will read as 0.
3015-3016	**Wind Speed**
Detailed description	Average wind speed magnitude since the last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0.00 to 60.00
Unit	m/s
Comments	—
3017-3018	**Wind Direction**
Detailed description	Average wind heading clockwise from North (0°) since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0.0 to 359.9
Unit	°
Comments	—
3019-3020	**Gust Wind Speed**
Detailed description	Maximum wind speed magnitude seen since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0.00 to 60.00
Unit	m/s
Comments	—
3021-3022	**Air Temperature**
Detailed description	Average air temperature since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	-65.0 to 60.0
Unit	°C
Comments	—

Table 24 Input Registers (continued)

3023-3024	**Vapor Pressure**
Detailed description	Average vapor pressure since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	1.00 to 47.00
Unit	kPa
Comments	—
3025-3026	**Atmospheric Pressure**
Detailed description	Average atmospheric pressure since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	1.00 to 120.00
Unit	kPa
Comments	—
3027-3028	**Relative Humidity**
Detailed description	Average relative humidity since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0.000 to 1.000
Unit	% / 100
Comments	—
3029-3030	**Humidity Sensor Temperature**
Detailed description	Average RH sensor temperature since last measurement
Data type	32-bit floating Big-Endian
Allowed Range	-40.0 to 85.0
Unit	°C
Comments	—
3031-3032	**Single-Value Orientation**
Detailed description	Average sensor polar orientation since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	0.0 to 180.0
Unit	°
Comments	—
3033-3034	**Minimum Air Temperature**
Detailed description	Lowest air temperature seen since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	-65.0 to 60.0
Unit	°C
Comments	—
3035-3036	**Maximum Air Temperature**
Detailed description	Highest air temperature seen since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	-65.0 to 60.0
Unit	°C
Comments	—

Table 24 Input Registers (continued)

3037-3038	North Wind Speed
Detailed description	Average wind speed from northerly direction since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	-60.00 to 60.00
Unit	m/s
Comments	—
3039-3040	**East Wind Speed**
Detailed description	Average wind speed from easterly direction since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	-60.00 to 60.00
Unit	m/s
Comments	—
3041-3042	**X-Direction Orientation**
Detailed description	Average sensor orientation in the X direction since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	-90.0 to 90.0
Unit	°
Comments	—
3043-3044	**Y-Direction Orientation**
Detailed description	Average sensor orientation in the Y direction since last measurement
Data type	32-bit floating point, Big-Endian
Allowed Range	-90.0 to 90.0
Unit	°
Comments	—

Table 25 lists the ATMOS 41 Gen 2 sensor identity Input registers.

Table 25 Identity Input Registers

3401	Sensor Type
Detailed description	Sensor type number
Data type	16-bit unsigned integer, Big-Endian
Comments	For ATMOS 41 Gen 2, this value is the decimal number 88.
3402-3403	**Sensor Numeric Serial Number**
Detailed description	Numeric portion of sensor serial number
Data type	32-bit unsigned integer, Big-Endian
Comments	This is the **XXXXXXX** portion of the sensor serial number A41G2M**XXXXXXX**
3404	**Sensor Firmware Version**
Detailed description	METER digital sensor firmware major and minor version
Data type	16-bit unsigned integer, Big-Endian
Comments	This number divided by 100 is the METER sensor version (e.g. 608 is version 6.08).
3405	**Sensor Build Version**
Detailed description	METER digital sensor firmware build/patch version
Data type	16-bit unsigned integer, Big-Endian
Comments	This number appended to the end of the METER sensor version forms the full firmware version number. (e.g. 16 appended to version 6.08 is version 6.08.16).

Table 25 Idenity Registers (continued)

3406	**Sensor Hardware Revision**
Detailed description	Sensor hardware revision number
Data type	16-bit unsigned integer, Big-Endian
Comments	This number indicates the hardware version and hardware capabilities of the sensor.
3407-3418	**Sensor Model String**
Detailed description	Sensor model information
Data type	UTF-16 array, Big-Endian
Comments	For ATMOS 41 Gen 2, this value is **AT41G2** encoded as **UTF-16**.
3419-3425	**Sensor ASCII Serial Number**
Detailed description	Sensor serial number
Data type	13-character ASCII string plus one null terminator
Comments	This is the sensor's full serial number, i.e. **A41G2MXXXXXXX**

EXAMPLE USING A CR6 DATALOGGER AND MODBUS RTU

The Campbell Scientific, Inc. CR6 Measurement and Control Datalogger supports Modbus client and Modbus server communication for integration in Modbus SCADA networks. The Modbus communications protocol facilitates the exchange of information and data between a computer/HMI software, instruments (RTUs), and Modbus-compatible sensors. The CR6 datalogger communicates in RTU mode exclusively. In a Modbus network, each server on the bus has a unique address. Therefore, sensor devices must be properly configured before being connected to a Modbus Network. Addresses range from `1` to `247`. Address `0` is reserved for universal broadcasts.

PROGRAMMING A CR6 DATALOGGER

The programs running on the CR6 (and CR1000) dataloggers are written in CRBasic, a language developed by Campbell Scientific. It is a high-level language designed to provide an easy, yet extremely flexible and powerful method of instructing the data logger how and when to take measurements, process data, and communicate. Programs can be created using either the ShortCut Software or be edited using the CRBasic Editor, both available for downloading as a stand-alone application on the official Campbell Scientific website (www.campbellsci.com).

A typical CRBasic program for a Modbus application consists of the following:

- Variables and constants declarations (public or private)
- Units declarations
- Configuration parameters
- Data tables declarations
- Logger initializations
- Scan (Main Loop) with all the sensors to be quired
- Function call to the data tables

CR6 DATALOGGER RS-485 CONNECTION INTERFACE

The universal (`U`) terminal of the CR6 offers 12 channels that connect to nearly any sensor type. It gives the CR6 the ability to match more applications and eliminates the use of many external peripherals.

The Modbus CR6 connection shown in Figure 13 uses the RS-485 (A/B) interface mounted on terminals (`C1–C2`) and (`C3–C4`). These interfaces can operate in Half-Duplex and Full-Duplex; the ATMOS 41 Gen 2 sensors uses RS-485 Half-Duplex. The serial interface of the ATMOS 41 Gen 2 sensor used for this example is connected to (`C1–C2`) terminals.

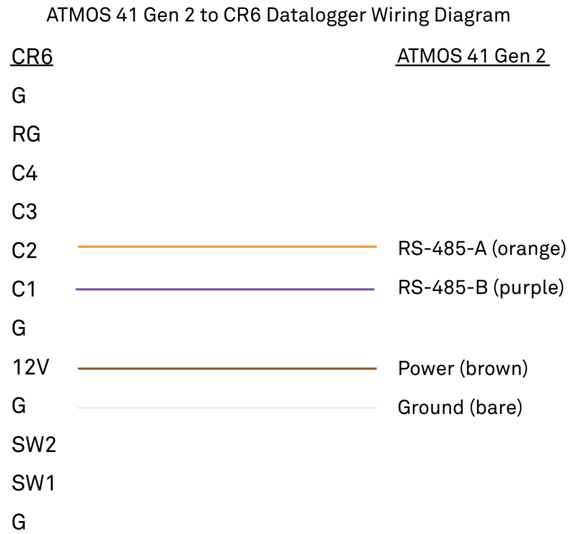

Figure 11 RS-485 interface

NOTE: The A/B pinout on the ATMOS 41 Gen 2 is different from the labels of the `C1` (A-) and `C2` (B+) ports of the `CR6`. Connect the ATMOS 41 Gen 2 RS-485-A line to `C2` and the RS-485-B line to `C1`.

After assigning the ATMOS 41 Gen 2 sensor a unique Modbus server address (and configuring its RTU parity and stop bit settings, if required) it can be wired according to Figure 11 to the CR6 logger. Make sure to connect the purple and the orange wires according to their signals respectively to the `C1` and `C2` ports. Connect the brown wire to `12 V` (`V+`) and the bare wire to `G` (`GND`). If the power supply must be controlled through the program, connect the brown wire directly to one of the `SW12` terminals (switched `12 V` outputs).

EXAMPLE PROGRAM

```
'CR6 Datalogger

'This is an example program for reading the ATMOS 41 Gen 2 All-in-One weather sensor
'using a CR6 series datalogger and the Modbus RTU protocol over half-duplex RS485. The
'measurement values polled

'This program runs a scan every 1 Min and stores the data in a 1 Min table.

'Declare Constants
Const MB_ADDR=1      'Default ATMOS 41 Gen 2 Modbus server address
Const MB_TIMEOUT= 15      '150ms request timeout (value * 0.01 sec)
Const MB_RETRIES= 1     'Number of times to retry Modbus client request

'Declare Public Variables
Public PTemp, batt_volt
Public mb_status 'variable used for monitoring the Modbus poll status
Dim A41G2_Data(22) 'Input register data; 22 32-bit values for 44 registers total

'Declare Private Variables

'Aliases used for the ATMOS 41 Gen 2
Alias A41G2_Data (1) = Solar: Units Solar=W/m^2
Alias A41G2_Data (2) = Precipitation: Units Precipitation=mm
Alias A41G2_Data (3) = PrecipDrops: Units PrecipDrops=counts
Alias A41G2_Data (4) = PrecipTips: Units PrecipTips=counts
Alias A41G2_Data (5) = PrecipitationEC: Units PrecipitationEC=uS/cm
Alias A41G2_Data(6) = Strikes: Units Strikes=counts
Alias A41G2_Data(7) = StrikeDistance: Units StrikeDistance=km
Alias A41G2_Data(8) = WindSpeed: Units WindSpeed=m/s
Alias A41G2_Data(9) = WindDirection: Units WindDirection=degrees
Alias A41G2_Data(10) = GustWindSpeed: Units GustWindSpeed=m/s
Alias A41G2_Data(11) = AirTemperature: Units AirTemperature=DegC
Alias A41G2_Data(12) = Vapor_Pressure: Units Vapor_Pressure=kPa
Alias A41G2_Data(13) = AtmPressure: Units AtmPressure=kPa
Alias A41G2_Data(14) = RelHumidity: Units RelHumidity=percent/100
Alias A41G2_Data(15) = RelHumidityTemp: Units RelHumidityTemp=DegC
Alias A41G2_Data(16) = SingleOrientation: Units SingleOrientation=degrees
Alias A41G2_Data(17) = Tmin : Units Tmin=DegC
Alias A41G2_Data(18) = Tmax: Units Tmax=DegC
Alias A41G2_Data(19) = NorthWindSpeed: Units NorthWindSpeed=m/s
Alias A41G2_Data(20) = EastWindSpeed: Units EastWindSpeed=m/s
Alias A41G2_Data(21) = XOrientation: Units XOrientation=degrees
Alias A41G2_Data(22) = YOrientation: Units YOrientation=degrees
```

```
Define Data Tables.
DataTable (A41G2_Table,1,-1)   'Set table size to # of records, or -1 to auto allocate.
  DataInterval (0,1,Min,10)  'Store new measurement every 1 Minute
  Minimum (1,batt_volt,FP2,False,False)
  Sample (1,PTemp,FP2)
  Sample (1,Solar,IEEE4)
  Sample (1,Precipitation,IEEE4)
  Sample (1,PrecipDrops,IEEE4)
  Sample (1,PrecipTips,IEEE4)
  Sample (1,PrecipitationEC,IEEE4)
  Sample (1,Strikes,IEEE4)
  Sample (1,StrikeDistance,IEEE4)
  Sample (1,WindSpeed,IEEE4)
  Sample (1,WindDirection,IEEE4)
  Sample (1,GustWindSpeed,IEEE4)
  Sample (1,AirTemperature,IEEE4)
  Sample (1,Vapor_Pressure,IEEE4)
  Sample (1,AtmPressure,IEEE4)
  Sample (1,RelHumidity,IEEE4)
  Sample (1,RelHumidityTemp,IEEE4)
  Sample (1,SingleOrientation,IEEE4)
  Sample (1,Tmin,IEEE4)
  Sample (1,Tmax,IEEE4)
  Sample (1,NorthWindSpeed,IEEE4)
  Sample (1,EastWindSpeed,IEEE4)
  Sample (1,XOrientation,IEEE4)
  Sample (1,YOrientation,IEEE4)
EndTable

'Main Program
BeginProg
'Port C1/C2, 9600 baud, even parity, 100us TX delay, 120 byte buffer, RS485 half-duplex
SerialOpen (ComC1,9600,2,100,120,4)
  Scan (1,Min,0,0) 'Read data from sensor and log every minute
    'Read 22 32-bit, big-endian values from A41G2 starting at input reg number 3001
    ModbusServer(mb_status,ComC1,9600,MB_ADDR,4,A41G2_Data,3001,22,MB_RETRIES,MB_TIMEOUT,2)
    CallTable A41G2_Table
  NextScan
EndProg
```

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
sales.environment@metergroup.com

Phone: +1.509.332.5600

Fax: +1.509.332.5158

Website: 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
sales.europe@metergroup.com

Phone: +49 89 12 66 52 0

Fax: +49 89 12 66 52 20

Website: metergroup.com

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
00	8.15.2024	6.08	Initial release.
01	10.1.2024	6.08	Changes to Table 24 and Table 25