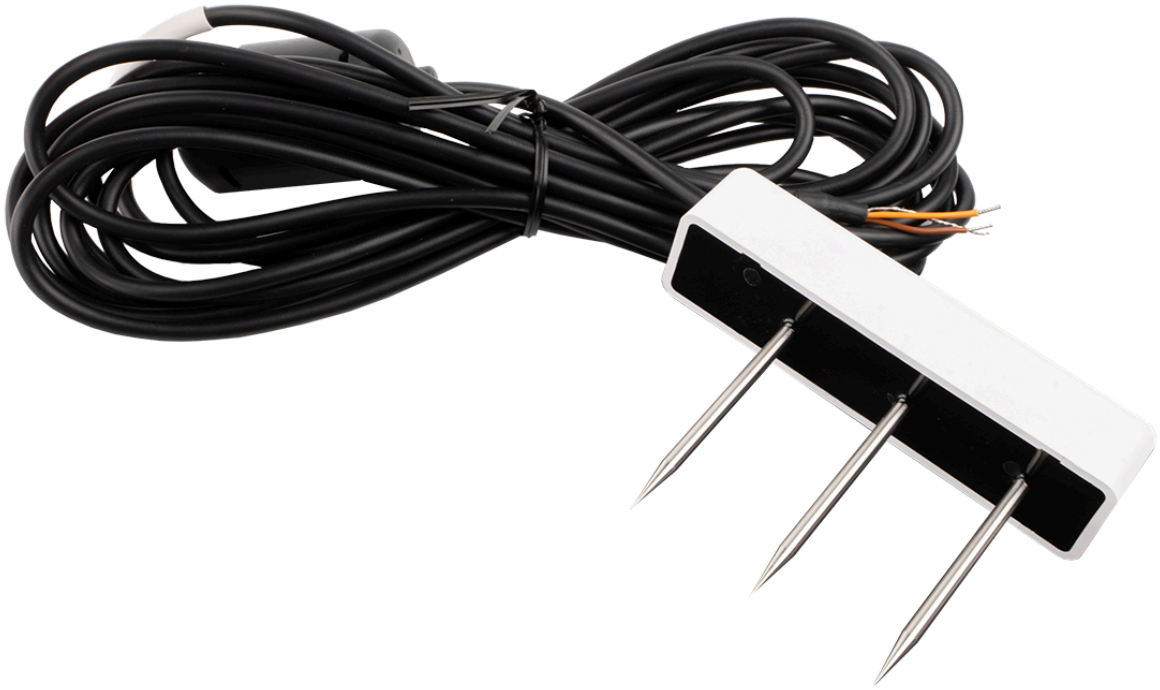
DATA SHEET

ELT Soil Moisture sensor Teros 12

Produal

Distributed by AKRIBIS



METER

TEROS 11/12 INTEGRATOR GUIDE

SENSOR DESCRIPTION

The TEROS 11 Soil Moisture and Temperature sensor and the TEROS 12 Soil Moisture, Temperature, and Electrical Conductivity (EC) sensor are accurate tools for monitoring volumetric water content (VWC), temperature in soil and soilless substrates, and electrical conductivity (TEROS 12 only). The TEROS 11/12 determines VWC using capacitance/frequency-domain technology. The sensor uses a 70 MHz frequency, which minimizes textural and salinity effects, making the TEROS 11/12 accurate in most mineral soils. The TEROS 11/12 uses a thermistor in the center needle to measure temperature and electrical conductivity (TEROS 12 only) using a stainless-steel electrode array.

For a more detailed description of how this sensor makes measurements, refer to the TEROS 11/12 User Manual.

APPLICATIONS

TEROS 11

- Volumetric water content (VWC)
- Soil–water storage measurement
- Irrigation management
- Soil/substrate temperature measurement
- Plant growth research

TEROS 12

- Volumetric water content (VWC)
- Soil/substrate water balance
- Irrigation management
- Soil/substrate temperature measurement
- Solute/fertilizer movement
- Soil EC measurement

ADVANTAGES

- Three-wire sensor interface: power, ground, and data
- Digital sensor communicates multiple measurements over serial interface
- Robust thermistor for accurate temperature measurements
- Low-input voltage requirements
- Low-power design supports battery-operated data loggers
- Robust epoxy encapsulation resists corrosive environments
- Supports SDI-12 or DDI Serial communications protocols
- Modern design optimized for low-cost sensing

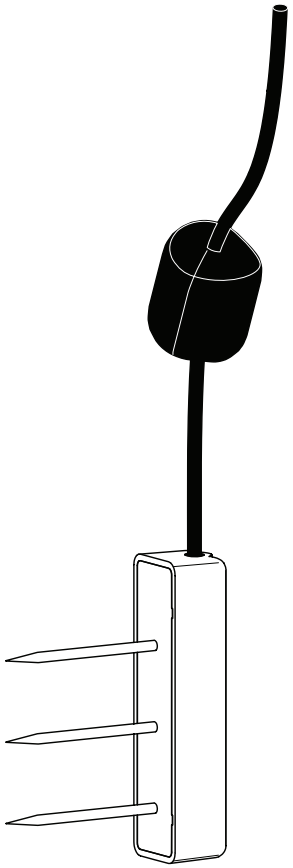


Figure 1 TEROS 11/12 sensor

PURPOSE OF THIS GUIDE

METER provides the information in this integrator guide to help 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 3.00 or newer.

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

AKRIBIS · www.akribis.info · contacto@akribis.info

SPECIFICATIONS

MEASUREMENT SPECIFICATIONS

Volumetric Water Content (VWC)	
Range	
Mineral soil calibration	0.00–0.70 m^3/m^3
Soilless media calibration	0.0–1.0 m^3/m^3
Apparent dielectric permittivity (ε_a)	1 (air) to 80 (water)

NOTE: The VWC range is dependent on the media the sensor is calibrated to. A custom calibration will accommodate the necessary ranges for most substrates.

Resolution	0.001 m^3/m^3
Accuracy	
Generic calibration	±0.03 m^3/m^3 typical in mineral soils that have solution electrical conductivity < 8,000 µS/cm
Medium specific calibration	±0.01–0.02 m^3/m^3 in any porous medium
Apparent dielectric permittivity (ε_a)	1–40 (soil range), ±1 ε_a (unitless) 40–80, 15% of measurement

Dielectric Measurement Frequency
70 MHz

Temperature	
TEROS 11	
Range	–40 to +60 °C
Resolution	0.1 °C
Accuracy	±1.0 °C from –40 to 0 °C ±0.5 °C from 0 to +60 °C

NOTE: Temperature measurement, for applicable sensors, may not be accurate if sensor is not fully immersed in the medium of interest, due to longer equilibration time.

TEROS 12	
Range	–40 to +60 °C
Resolution	0.1 °C
Accuracy	±0.5 °C from –40 to 0 °C ±0.3 °C from 0 to +60 °C

Bulk Electrical Conductivity (EC_b) (TEROS 12 Only)	
Range	0–20,000 µS/cm (bulk)
Resolution	1 µS/cm
Accuracy	±(5% + 10 µS/cm) from 0–10,000 µS/cm ±8% from 10,000–20,000 µS/cm

COMMUNICATION SPECIFICATIONS

Output
DDI Serial or SDI-12 communications protocol

Data Logger Compatibility
METER ZL6, EM60, and Em50 data loggers or any data acquisition system capable of 4.0- to 15-VDC power and serial or SDI-12 communications

PHYSICAL SPECIFICATIONS

Dimensions	
Length	9.4 cm (3.70 in)
Width	2.4 cm (0.95 in)
Height	7.5 cm (2.95 in)

Prong Length
5.5 cm (2.17 in)

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
3.50 mm

Conductor Gauge
22-AWG / 24-AWG drain wire

ELECTRICAL AND TIMING CHARACTERISTICS

Supply Voltage (power to ground)	
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

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

Current Drain (during 25-ms measurement)	
Minimum	3.0 mA
Typical	3.6 mA
Maximum	16.0 mA

Current Drain (while asleep)	
Minimum	NA
Typical	0.03 mA
Maximum	NA

Operating Temperature Range	
Minimum	–40 °C
Typical	NA
Maximum	+60 °C

NOTE: Sensors may be used at higher temperatures under certain conditions; contact Customer Support for assistance.

Power Up Time (DDI Serial)	
Minimum	80 ms
Typical	NA
Maximum	100 ms

Power Up Time (SDI-12)	
Minimum	NA
Typical	245 ms
Maximum	NA

Power Up Time (SDI-12, DDI Serial disabled)	
Minimum	145 ms
Typical	155 ms
Maximum	165 ms

Measurement Duration	
Minimum	25 ms
Typical	NA
Maximum	150 ms

COMPLIANCE

EM ISO/IEC 17050:2010 (CE Mark)

2014/30/EU and 2011/65/EU

EN61326-1:2013 and EN55022/CISPR 22

EN 55011:2016/A1:2017 (GROUP 1, CLASS A-RCM mark)

EQUIVALENT CIRCUIT AND CONNECTION TYPES

Refer to Figure 2 and Figure 3 to connect the TEROS 11/12 to the logger. Figure 2 provides a low-impedance variant of the recommended SDI-12 specification.

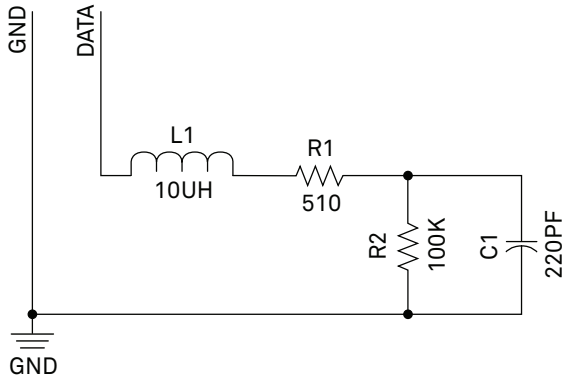


Figure 2 Equivalent circuit diagram

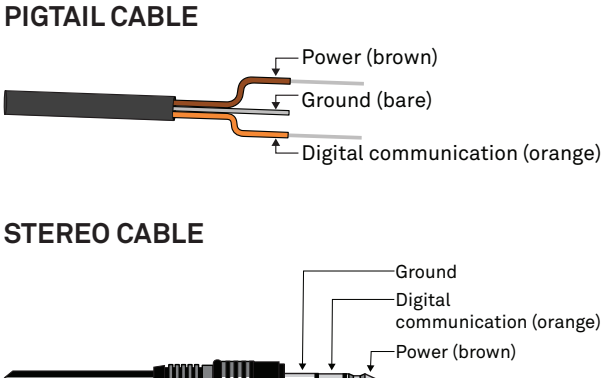


Figure 3 Connection types

⚠ 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.

Read the application note Lightning surge and grounding practices on the METER website for 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's installation instructions. Improper grounding may affect the sensor output as well as sensor performance.

Read the application note Lightning surge and grounding practices on the METER website for 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 TEROS 11/12 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 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 hand-held device 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 2). During the power-up time, sensors output a sensor reading formatted as a DDI Serial message 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 (`aC`–`aC9` and `aCC0`–`aCC9`). `M`, `R`, and `C` command implementations are found on pages 8–10. The `aR3` and `aR4` commands are used by METER systems and as a result use space delimination, instead of the sign delimination required by the SDI-12 standard.

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 TEROS 11/12 will omit the DDI Serial startup string (sensor identification) when the SDI-12 address is nonzero. 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. METER SDI-12 sensors can be power-cycled and read on the desired measurement interval or powered continuously and commands sent when a measurement is desired. Many factors influence the effectiveness of the bus configuration. Visit metergroup.com for articles and virtual seminars containing more information.

SENSOR ERROR CODES

The TEROS 11/12 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.

Table 1 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 4 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 5.

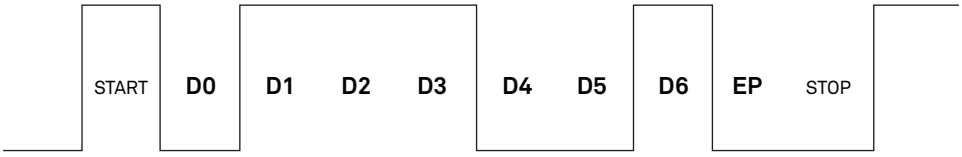


Figure 4 Example SDI-12 transmission of the character 1 (0x31)

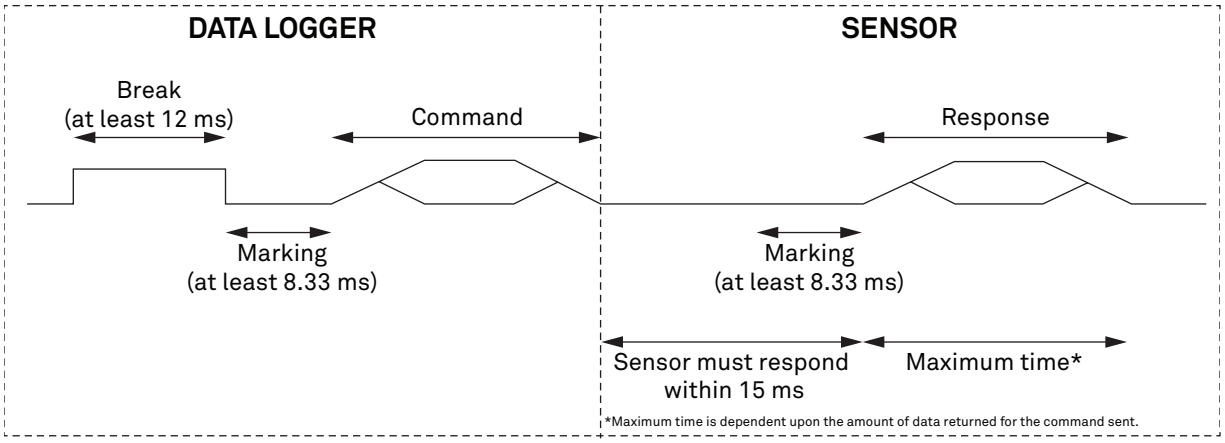


Figure 5 Example data logger and sensor communication

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 TEROS 11 command and response is shown in Example 1 and the TEROS 12 command and response is shown in Example 2, where the command is in **bold** and the response follows the command.

Example 1 **1I!**113METER␣␣␣TER11␣114631800001 **(TEROS 11)**

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)
TER11␣	6	Sensor model string. This string is specific to the sensor type. For the TEROS 11, the string is TER11␣.
114	3	Sensor version. This number divided by 100 is the METER sensor version (e.g., 114 is version 1.14).
631800001	≤13, variable	Sensor serial number. This is a variable length field. It may be omitted for older sensors.

Example 2 `1I!113METER␣␣␣TER12␣114631800001` **(TEROS 12)**

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)
`TER12␣`	6	Sensor model string. This string is specific to the sensor type. For the TEROS 12, the string is `TER12␣`.
`114`	3	Sensor version. This number divided by 100 is the METER sensor version (e.g., 114 is version 1.14).
`631800001`	≤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 3, where the command is in **bold** and the response follows the command.

Example 3 `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 4 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 4 `?!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.

NOTE: SDI-12 commands `MC`, `CC`, and `RC` that request the sensor response include cyclical redundancy check characters may also be used.

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 and for an explanation of the command sequence and see Table 11 for an explanation of response parameters.

Table 2 aM! command sequence

Command	Response
This command reports instantaneous values.	
aM!	atttn
aD0!	a+<calibratedCountsVWC>±<temperature>+<electricalConductivity>

NOTES: <electricalConductivity> is only output on the TEROS 12.

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 Specification 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 11 for an explanation of response parameters.

Table 3 aC! measurement command sequence

Command	Response
This command reports instantaneous values.	
aC!	atttnn
aD0!	a+<calibratedCountsVWC>±<temperature>+<electricalConductivity>

NOTES: <electricalConductivity> is output on the TEROS 12 only.

This command does not adhere to the SDI-12 concurrent command requirements. See METER SDI-12 Implementation for more information.

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 Specification v1.3 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.

Table 4 aV! measurement command sequence

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

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 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.
- `aXO!` (with capital O) to suppress the DDI Serial string.

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

Table 5 `aXO!` measurement command sequence

Command	Response
`aXO!`	`a<suppressionState>`
`aXO<suppressionState>!`	`aOK`

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

Table 6 `aXR3!` measurement command sequence

Command	Response
`aXR3!`	`a<TAB><calibratedCountsVWC> <temperature> <electricalConductivity><CR><sensorType><Checksum><CRC>`

NOTES: `<electricalConductivity>` is output on the TEROS 12 only.

This command does not adhere to the SDI-12 response format or timing. 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. See METER SDI-12 Implementation for more information.

Table 7 `aXR4!` measurement command sequence

Command	Response
`aXR4!`	`a<TAB> <calibratedCountsVWC> <temperature> <electricalConductivity><CR><sensorType><Checksum><CRC>`

NOTES: `<electricalConductivity>` is output on the TEROS 12 only.

This command does not adhere to the SDI-12 response format or timing. 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. See METER SDI-12 Implementation 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.

Please refer to Table 8 through Table 10 for an explanation of the command sequence and see Table 11 for an explanation of response parameters.

Table 8 `aR0!` measurement command sequence

Command	Response
This command reports instantaneous values.	
`aR0!`	`a+<calibratedCountsVWC>±<temperature>+<electricalConductivity>`

NOTE: `<electricalConductivity>` is output on the TEROS 12 only.

This command does not adhere to the SDI-12 response timing. See METER SDI-12 Implementation for more information.

Table 9 `aR3!` measurement command sequence

Command	Response
This command reports instantaneous values.	

NOTE: `<electricalConductivity>` is output on the TEROS 12 only.

This command does not adhere to the SDI-12 response format or timing. 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. See METER SDI-12 Implementation for more information.

Table 9 `aR3!` measurement command sequence

Command	Response
`aR3!`	`a<TAB><calibratedCountsVWC> <temperature> <electricalConductivity><CR><sensorType><Checksum><CRC>`

NOTE: `<electricalConductivity>` is output on the TEROS 12 only.

This command does not adhere to the SDI-12 response format or timing. 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. See METER SDI-12 Implementation for more information.

Table 10 `aR4!` measurement command sequence

Command	Response
This command reports instantaneous values.	
`aR4!`	`a<TAB><calibratedCountsVWC> <temperature> <electricalConductivity><CR><sensorType><Checksum><CRC>`

NOTES: `<electricalConductivity>` is output on the TEROS 12 only.

This command does not adhere to the SDI-12 response format or timing. 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. See METER SDI-12 Implementation for more information.

PARAMETERS

Table 11 lists the parameters, unit measurement, and a description of the parameters returned in command responses for TEROS 11/12.

Table 11 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
`<calibratedCountsVWC>`	—	Calibrated ADC counts for volumetric water content
`<temperature>`	°C	Air temperature
`<electricalConductivity>`	µS/cm	Bulk electrical conductivity normalized to 25 °C (TEROS 12 only)
`<meta>`	—	Auxiliary sensor information. See Table 12.
`<suppressionState>`	—	`0`: DDI Serial message present at power up `1`: DDI Serial message suppressed at power up
`<sensorType>`	—	ASCII character denoting the sensor type For TEROS 11, the character is `h` For TEROS 12, the character is `g`
`<Checksum>`	—	METER serial checksum
`<CRC>`	—	METER 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!` `aDO!` sequence will output a `<meta>` integer value. This integer represents a binary bitfield, with each individual bit representing an error flag.

Table 12 lists the possible error flags that can be set by the TEROS 11/12. 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 12, 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 12 Error flag values and issue resolution

Error Flag Value	Issue Present	Resolution
64	Sensor thermistor is broken and sensor is using a backup measurement	Contact Customer Support to replace sensor
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

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- to 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. TEROS 11/12 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 `0` and print out the DDI Serial string when power cycled.

DDI SERIAL TIMING

Table 13 lists the DDI Serial communications configuration.

Table 13 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 6). When the reading is complete, the sensor begins sending the serial signal out the data line adhering to the format shown in Figure 7. 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.

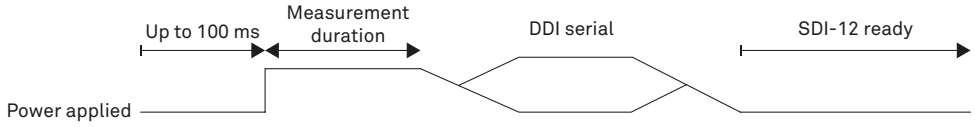


Figure 6 Data line DDI Serial timing

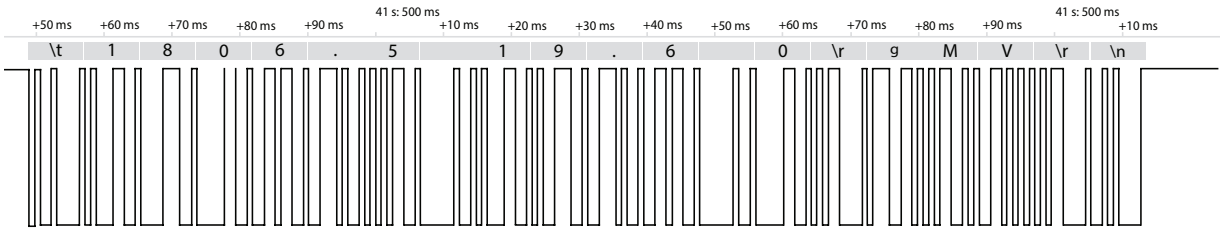


Figure 7 Example DDI Serial transmission

DDI SERIAL RESPONSE

This section contains tables detailing the DDI Serial response.

Table 14 DDI Serial response

Command	Response
NA	<TAB><calibratedCountsVWC> <temperature> <electricalConductivity><CR><sensorType><Checksum><CRC>

NOTES: `<electricalConductivity>` **is only output on the TEROS 12.**

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

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.

TEROS 11 LEGACY CHECKSUM EXAMPLE

Legacy checksum example input is `<TAB>1797.7 21.8<CR>h` and the resulting checksum output is `D`.

TEROS 12 LEGACY CHECKSUM EXAMPLE

Legacy checksum example input is `<TAB>2749.0 23.8 660<CR>g` and the resulting checksum output is `8`.

```
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;
}
```

TEROS 11 CRC6

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, excluding the sensor address.

`CRC6` checksum example input is `<TAB>1797.2 21.8<CR>hD` and the resulting checksum output is `2`.

TEROS 12 CRC6

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, excluding the sensor address.

`CRC6` checksum example input is `<TAB>2749.0 23.8 660<CR>g8` and the resulting checksum output is `0` (uppercase O).

```
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
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.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
10	12.2023	3.00	Updated RCM compliance in spec table
09	3.2023	3.00	Updated ISO
08	12.2022	3.00	Updated waveform example
07	10.2022	3.00	Corrected checksum tables.
06	2.2022	3.00	Add cable specifications and update specifications.
05	12.18.2020	1.14	Added Verification and Extended commands, error codes, and metadata values.
04	11.30.2019	1.07	Changed maximum sensor measurement time.
03	9.11.2019	1.07	Converted specification units to microsiemens.
02	2.15.2019	1.07	Add TEROS 11.
01	4.30.2018	1.00	Correct specifications and measurement value.
00	3.1.2018	1.00	Initial release.