18315-02
3.27.2020

METER

TEROS 32 INTEGRATOR GUIDE

SENSOR DESCRIPTION

The TEROS 32 Soil Water Potential and Temperature sensor is a precision tensiometer that measures water potential in the critical range (–85 kPa to +50 kPa) for water movement, critical range for plant response, and slope stability with its ability to measure positive pore pressures. The TEROS 32 is designed to be installed down an augered hole, plugged into a data logger, and left to log water potential data. Tensiometers will require periodic refilling when measurements go beyond the measuring range of the sensor.

For a more detailed description of how this sensor makes measurements, refer to the TEROS 32 User Manual.

APPLICATIONS

- Soil-water tension measurement
- Soil-water storage measurement
- Irrigation management
- Soil temperature measurement
- In-situ retention curves

ADVANTAGES

- Plug-and-play tensiometer
- Three-wire sensor interface: power, ground, and data
- Digital sensor communicates multiple measurements over a serial interface
- Low-input voltage requirements
- Low-power design supports battery-operated data loggers
- SDI-12 or DDI serial communication protocols
- Modern design optimized for low-cost sensing

PURPOSE OF THIS GUIDE

METER provides the information in this integrator guide to help TEROS 32 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. For more information about METER integrated systems, please contact Customer Support.

Figure 1 **TEROS 32 Soil Water Potential and Temperature sensor**

COMPATIBLE FIRMWARE VERSIONS

This guide is compatible with firmware versions 1.01 or newer.

METER Group, Inc. USA
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

Water Potential	
Range	−85 to +50 kPa
Resolution	0.0012 kPa
Accuracy	±0.15 kPa

Temperature	
Range	−30 to +60 °C
Resolution	±0.01 °C
Accuracy	±0.1 °C between −20 and +40 °C (±1 °C outside of this range)

COMMUNICATION SPECIFICATIONS

Output

DDI serial
SDI-12 communication protocol

Data Logger Compatibility

METER ZL6 and EM60 data loggers or any data acquisition system capable of 3.6- to 28.0-VDC power and SDI-12.

PHYSICAL SPECIFICATIONS

Dimensions	
Length	40.0 cm (15.75 in) 80.0 cm (31.50 in) 120.0 cm (47.24 in)
Diameter	2.5 cm (0.98 in)

Operating Temperature	
Minimum	-30 °C (0 °C for water-filled tensiometer)
Typical	NA
Maximum	50 °C

Materials	
Ceramic cup	Al_2O_3, bubble point 1,500 kPa
Shaft	PMMA
Sensor body	POM GF
Refilling tube	Stainless steel

Installation Angle

10° to 80° from horizontal (downward)
−10° to −80° from horizontal (upward)

Cable Length

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

NOTE: Contact Customer Support if a nonstandard cable length is needed.

Connector Types

3.5-mm stereo plug connector or stripped and tinned wires

ELECTRICAL AND TIMING CHARACTERISTICS

Supply Voltage (VCC to GND)	
Minimum	3.6 V
Typical	12.0 V
Maximum	28.0 V

Digital Input Voltage (logic high)	
Minimum	1.6 V
Typical	3.3 V
Maximum	5.0 V

Digital Input Voltage (logic low)	
Minimum	−0.3 V
Typical	0.0 V
Maximum	0.9 V

Digital Output Voltage (logic high)	
Minimum	NA
Typical	3.6 V
Maximum	NA

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

Current Drain (during measurement)	
Minimum	18 mA
Typical	25 mA
Maximum	30 mA

Current Drain (while asleep)	
Minimum	0.03 mA
Typical	0.05 mA
Maximum	0.90 mA
Power Up Time (DDI serial)	
Minimum	125 ms
Typical	130 ms
Maximum	150 ms
Power Up Time (SDI-12)	
Minimum	125 ms
Typical	130 ms
Maximum	150 ms

Measurement Duration	
Minimum	60 ms
Typical	65 ms
Maximum	70 ms

COMPLIANCE

Manufactured under ISO 9001:2015

EM ISO/IEC 17050:2010 (CE Mark)

2014/30/EU and 2011/65/EU

EN61326-1:2013 and EN55022/CISPR 22

EQUIVALENT CIRCUIT AND CONNECTION TYPES

Refer to Figure 2 and Figure 3 to connect the TEROS 32 to a data logger. Figure 2 provides a low-impedance variant of the recommended SDI-12 specification.

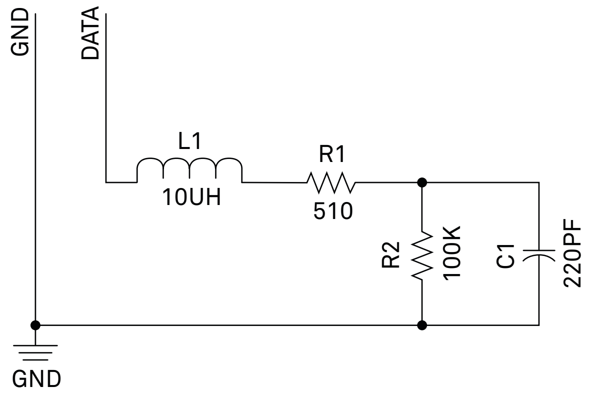

Figure 2 Equivalent circuit diagram

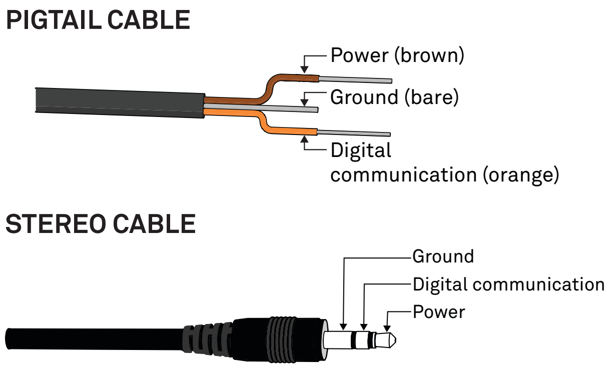

Figure 3 Connection types

⚠ 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 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 32 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 PROCHECK handheld 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 some sensor diagnostic information and should not be communicated with until the power-up time has passed. After the power-up time, the sensors are fully compatible with all commands listed in the SDI-12 Specification v1.3. However, the TEROS 32 will not return measurements in response to continuous measurement (`R`) commands. `M` and `C` command implementations are found on page 7.

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

SENSOR BUS CONSIDERATIONS

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

SENSOR ERROR CODE

The TEROS 32 has one error code: -9999. This error code 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.

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 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.

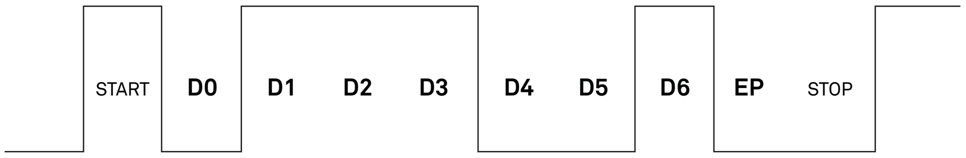

Figure 4 Example SDI-12 transmission of the character 1 (0x31)

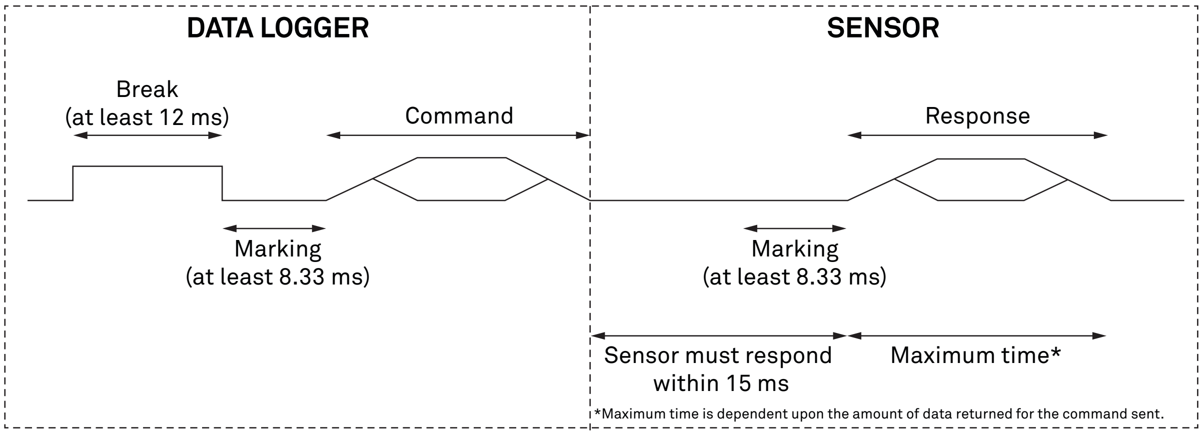

Figure 5 Example data logger and sensor communication

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 command and response is shown in Example 1, where the command is in **bold** and the response follows the command.

Example 1 **1I!**113METER␣␣␣TER32␣100631800001

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)
TER32␣	6	Sensor model string. This string is specific to the sensor type. For the TEROS 32, the string is TER32.
100	3	Sensor version. This number divided by 100 is the METER sensor version (e.g., 100 is version 1.00).
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 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`) and Concurrent (`C`) 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 and for an explanation of the command sequence and to Table 6 for an explanation of response parameters.

Table 2 `aM!` command sequence

Command	Response
This command reports instantaneous values.	
`aM!`	`atttn`
`aD0!`	`a+<matricPotential>±<temperature>+<meta>`
`aD1!`	`a±<pitch>±<roll>`
`aD2!`	`a+<uPressure>±<temperature>`

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 `<CR><LF>` is sent from the sensor signaling the measurement is ready. Either wait until `ttt` seconds have passed or wait until the service request is received before sending the data commands. See the SDI-12 Specifications v1.3 document for more information.

CONCURRENT MEASUREMENT COMMANDS IMPLEMENTATION

Concurrent Measurement (`C`) commands are typically used with sensors connected to a bus. `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 to Table 6 for an explanation of response parameters.

Table 3 `aC!` measurement command sequence

Command	Response
This command reports instantaneous values.	
`aC!`	`atttnn`
`aD0!`	`a+<matricPotential>±<temperature>+<meta>`
`aD1!`	`a±<pitch>±<roll>`
`aD2!`	`a+<uPressure>±<temperature>`

NOTE: Please see the SDI-12 Specifications 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. First the `V` command is sent followed by `D` commands to read the response.

Please refer to Table 4 for an explanation of the command sequence and see Table 6 for an explanation of response parameters.

Table 4 **aV! measurement command sequence**

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

NOTE: Please see the SDI-12 Specifications v1.3 document for more information.

EXTENDED COMMANDS IMPLEMENTATION

Extended (X) commands provided sensors with a means of performing manufacturer-specific functions. METER implements the following extended command to allow integrators an alternative way to turn off the DDI string. Sending the command without a parameter will return the current setting for <suppressionState>. Sending a value for <suppressionState> will set that value. Extended 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>.

Please refer to Table 5 for an explanation of the command sequence and see Table 6 for an explanation of response parameters.

Table 5 **aX0! measurement command sequence**

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

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

PARAMETERS

Table 6 lists the parameters, unit measurement, and a description of the parameters returned in command responses for TEROS 32.

Table 6 **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
<matricPotential>	kPa	Matric potential
<temperature>	°C	Sensor temperature
<meta>	—	Auxiliary sensor information 0: No sensor error 1: Sensor has experienced temperatures below freezing 16: Sensor refill orientation error 17: Both 1 and 16
<pitch>	°	Sensor pitch (0° is parallel to Earth's surface)
<roll>	°	Sensor roll (0° is top reference point straight up)
<uPressure>	kPa	Uncalibrated transducer pressure

Table 6 Parameter descriptions (continued)

Parameter	Unit	Description
`<suppressionState>`	—	0: DDI string unsuppressed 1: DDI string suppressed
`<sensorType>`	—	ASCII character denoting the sensor type For TEROS 32, the character is b
`<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!`, `aD0!` sequence will output a `<meta>` integer value. This integer represents a binary bitfield, with each individual bit representing an error flag. Below are the possible error flags that can be set by the TEROS 32. If multiple error flags are set, the sensor metadata integer value will be the sum of their individual values. To decode an integer value not explicitly called out in the table below, find the largest error flag value in the table 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 81 is the sum of individual error flag values 64 + 16 + 1, so this sensor has freezing error flag, sensor misorientation error flag, and sensor calibrations lost or corrupted error flag.

Table 7 Error flag values and issue resolution

Error Flag Value	Issue Present	Resolution
0	No issue present	NA
1	Sensor has experienced temperature below freezing	Contact Customer Support. Irreversible sensor damage is likely.
16	Sensor misorientation will prevent effective refilling	Use ZENTRA Utility app to reorient the pitch or roll of the 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 0–3.6 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.

The TEROS 32 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 startup string when power cycled.

DDI SERIAL TIMING

Table 8 lists the DDI serial communication configuration.

Table 8 DDI serial communication 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.

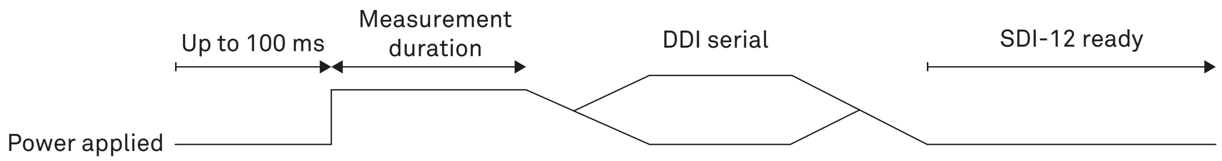

Figure 6 Data line DDI serial timing

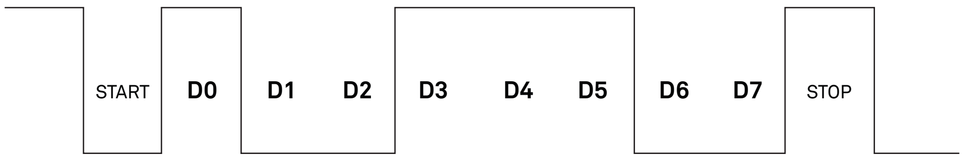

Figure 7 Example DDI serial transmission of the character 9 (0x39)

DDI SERIAL RESPONSE

Table 9 details the DDI serial response.

Table 9 DDI serial response

Command	Response
NA	`<TAB><matricPotential> <temperature> <meta><CR><sensorType><Checksum><CRC>`

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

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

DDI SERIAL CHECKSUM

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

Example input is `<TAB>92.953 23.6 0<CR>b` and the resulting checksum output is `K`.

```
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 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>92.953 23.6 0<CR>bK and the resulting checksum output is A.

```
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 meta data 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
02	3.27.2020	1.01	Updated DDI response table
01	1.10.2020	1.01	Added sections: Sensor Error Code, Verification Command Implementation, Extended Commands Implementation, and Sensor Metadata Value
00	9.13.2019	1.00	Initial release