



METER

HYDROS 21 GEN 2 INTEGRATOR GUIDE

SENSOR DESCRIPTION

The HYDROS 21 Gen 2 is a low-cost, accurate tool for monitoring water level, electrical conductivity (EC), and temperature in both groundwater and surface water. The sensor employs a precision pressure transducer to sense water levels between 0 and 10 m. The sensor cable is vented to remove the effects of barometric pressure changes. The integrated 4-probe electrical conductivity transducer accurately senses EC up to 120,000 $\mu\text{S}/\text{cm}$. The sensor also features a precision thermistor to measure temperature. The HYDROS 21 has a compact 3.4-cm-diameter body. The electronic circuitry is encapsulated in a marine-grade epoxy to protect the sensor in corrosive environments.

For a more detailed description of how this sensor operates, refer to the [HYDROS 21 Gen 2 User Manual](#).

APPLICATIONS

- Aquifer recharge and recovery
- Saltwater intrusion, desalination, and wastewater
- Wetland monitoring
- Groundwater contamination monitoring
- Surface water monitoring

ADVANTAGES

- Precision pressure transducer for water depth measurements
- Accurate 4-probe EC measurement
- Robust thermistor for accurate temperature measurements
- Differential pressure measurement referenced to atmospheric pressure so no external pressure sensor is needed
- Robust marine-grade epoxy encapsulation to resist corrosive environments
- 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
- Supports SDI-12 or DDI Serial communications protocols
- Modern design optimized for low-cost sensing

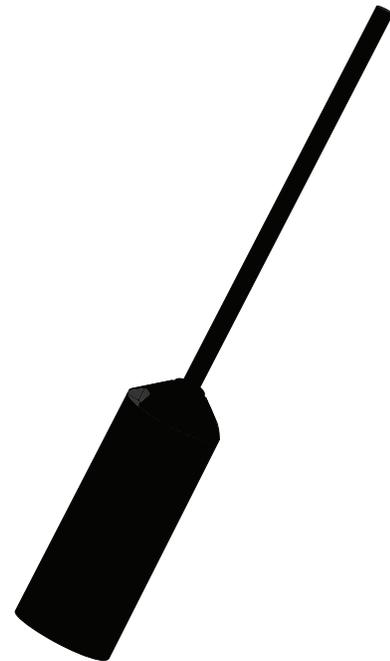


Figure 1 HYDROS 21 sensor

PURPOSE OF THIS GUIDE

METER provides the information in this integrator guide to help HYDROS 21 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 5.03 or newer for the HYDROS 21 Gen 2.

SPECIFICATIONS

MEASUREMENT SPECIFICATIONS

Water Depth	
Range	0–10,000 mm
Resolution	1 mm
Accuracy	±0.25% of full scale at 20 °C
NOTE: Depth measurement accuracy assumes no abrupt temperature variations.	
Temperature	
Range	–40 to +60 °C
Resolution	0.1 °C
Accuracy	±1 °C
NOTE: The pressure transducer may be damaged or destroyed if frozen in ice. Remove the sensor if the water temperature could drop below 0 °C (32 °F).	
Electrical Conductivity (EC)	
Range	0–120,000 µS/cm
Resolution	1 µS/cm
Accuracy	±10 µS/cm or ±10%, whichever is greater
NOTE: The EC measurement is corrected to a standard temperature of 25 °C.	

COMMUNICATION SPECIFICATIONS

Output	
DDI Serial or SDI-12 communications protocol	
Data Logger Compatibility	
METER ZL6 data loggers and any data acquisition system capable of 4.0- to 15-VDC power and serial or SDI-12 communications	

PHYSICAL SPECIFICATIONS

Dimensions	
Length	9.0 cm (3.5 in)
Width	3.4 cm (1.3 in)
Operating Temperature Range	
Minimum	0 °C
Maximum	60 °C
NOTE: The pressure transducer may be damaged or destroyed if frozen in ice. No damage will occur at subfreezing temperatures if the sensor is not in water. Contact Customer Support for assistance.	

Cable Length	
10 m (standard)	
20 m	
40 m (maximum)	
Cable Diameter	
0.240 ±0.010 in (6.00 ±0.25 mm) with minimum jacket of 0.031 in (0.80 mm)	
Connector Types	
Stereo plug connector or stripped and tinned wires	
Stereo Plug Connector Diameter	
3.50 mm	
Conductor Gauge	
20-AWG signal and power / 21-AWG ground wire	

ELECTRICAL AND TIMING CHARACTERISTICS

Supply Voltage (power to ground)	
Minimum	4.0 V
Typical	NA
Maximum	15.0 V
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 measurement)	
Minimum	2.5 mA
Typical	3.5 mA
Maximum	8.0 mA

Current Drain (while asleep)		Power Up Time (SDI-12, DDI Serial disabled)	
Minimum	NA	Minimum	NA
Typical	0.35 mA	Typical	175 ms
Maximum	NA	Maximum	200 ms
Power Up Time (DDI Serial)		Measurement Duration	
Minimum	52 ms	Minimum	NA
Typical	NA	Typical	450 ms
Maximum	100 ms	Maximum	600 ms
Power Up Time (SDI-12)		COMPLIANCE	
Minimum	NA	Manufactured under ISO 9001:2015	
Typical	700 ms	EM ISO/IEC 17050:2010 (CE Mark)	
Maximum	750 ms		

EQUIVALENT CIRCUIT AND CONNECTION TYPES

Refer to [Figure 2](#) and [Figure 3](#) to connect the HYDROS 21 to a data logger. [Figure 2](#) provides a low-impedance variant of the recommended SDI-12 specification.

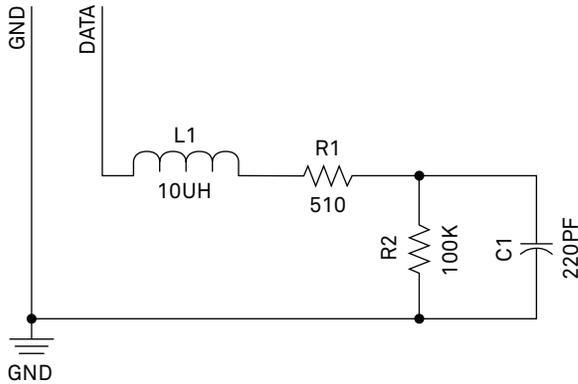


Figure 2 Equivalent circuit diagram

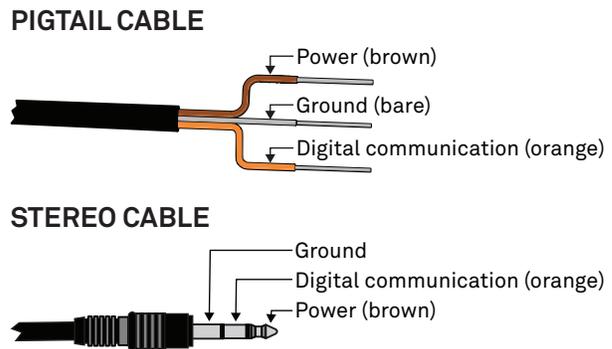


Figure 3 Connection types

⚠ 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 HYDROS 21 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.

The HYDROS 21 cable has an integrated vent tube to provide an atmospheric pressure reference for the depth sensor. The cable should be installed without sharp bends that would crimp this internal vent tube. Please see the [HYDROS 21 Gen 2 User Manual](#) for more information on sensor installation and protecting this vent.

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 PROCHECK handheld devices can operate as a computer-to-sensor interface for making on-demand sensor measurements. For more information, please contact [Customer Support](#).

METER SDI-12 IMPLEMENTATION

METER sensors use a low-impedance variant of the SDI-12 standard sensor circuit ([Figure 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 ([aR3](#) and [aRC3](#)). See [page 7](#) for [M](#), [R](#), and [C](#) command implementations.

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 HYDROS 21 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. 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 HYDROS 21 has four 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.
- -9990 is output in place of the measured value if the sensor measurement is out of range.

SDI-12 CONFIGURATION

Table 1 lists the SDI-12 communications configuration.

Table 1 SDI-12 communications configuration

Baud Rate	1,200 bps
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.

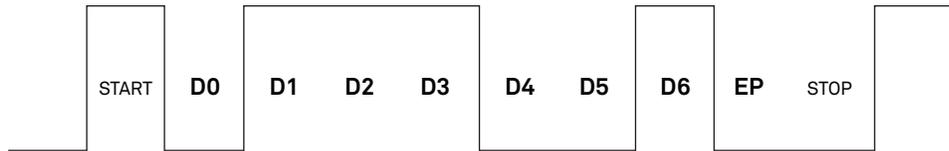


Figure 4 Example SDI-12 transmission of the character 1 (0x31)

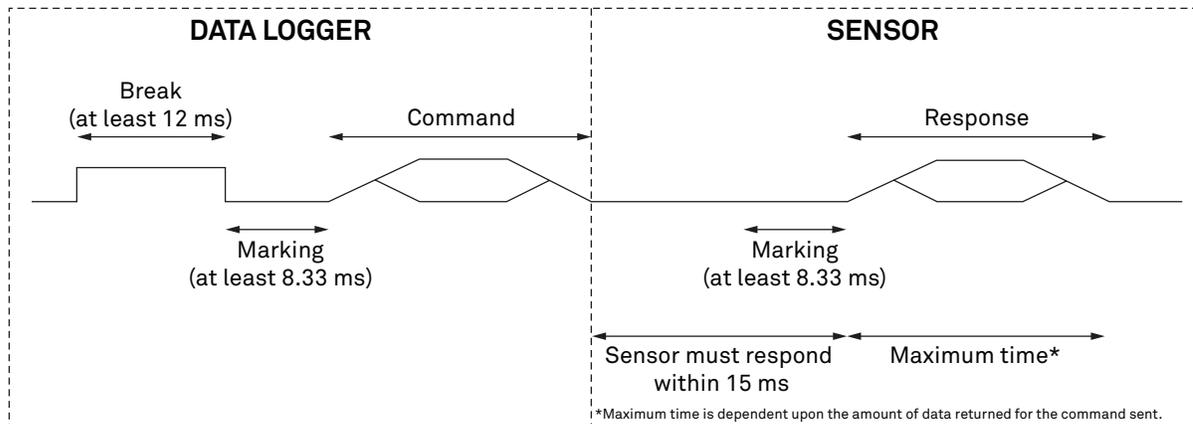


Figure 5 Example data logger and sensor communication

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_ _ _ _HYD21_ _502H21G200000075

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 _ _ _ _)
HYD21_ _	6	Sensor model string. This string is specific to the sensor type. For the HYDROS 21, the string is HYD21_ _ .
502	3	Sensor version. This number divided by 100 is the METER sensor version (e.g., 502 is version 5.02).
H21G200000075	≤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
<code>?!</code>	2	Data logger command. Request for a response from any sensor listening on the data line.
<code>0</code>	1	Sensor address. Returns the sensor address to the currently connected sensor.

COMMAND IMPLEMENTATION

The following tables list the relevant Measurement (M), Continuous (R), and Concurrent (C) 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 to [Table 9](#) for an explanation of response parameters.

Table 2 `aM!` command sequence

Command	Response
<code>aM!</code>	<code>atttn</code>
<code>aD0!</code>	<code>a±<depth>±<temperature>+<electricalConductivity></code>

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

CONCURRENT MEASUREMENT COMMANDS IMPLEMENTATION

Concurrent Measurement (C) commands are typically used with sensors connected to a bus. This sensor supports the Concurrent Measurement command; however, the implementation is not fully compliant with the SDI-12 specification for Concurrent Measurement commands. It is not possible for a recorder to interrupt a Concurrent Measurement command by issuing a second command to this sensor during the time the sensor is making a measurement (`ttt`). This sensor will only respond to further commands after the specified amount of time detailed in the C command response. The recorder is free to send commands to other sensors during the time the sensor is making measurements.

Please refer to [Table 3](#) for an explanation of the command sequence and to [Table 9](#) for an explanation of response parameters.

Table 3 `aC!` measurement command sequence

Command	Response
<code>aC!</code>	<code>atttnn</code>
<code>aD0!</code>	<code>a±<depth>±<temperature>+<electricalConductivity></code>

NOTE: Please see the [SDI-12 Specifications v1.3](#) document for more information.

VERIFICATION MEASUREMENT 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.

Table 4 aV! measurement command sequence

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

EXTENDED COMMANDS IMPLEMENTATIONS

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

METER implements the following X commands:

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

Please refer to Table 5 through Table 8 for an explanation of the command sequence and Table 9 for an explanation of response parameters.

Table 5 HYDROS 21 aX0! measurement command sequence

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

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

Table 6 HYDROS 21 aXR3! measurement command sequence

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

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. This sensor is not compliant with SDI-12 specification requiring the sensor to respond within 15 ms to a Continuous (R) command. This sensor will respond after the measurement duration (see [Electrical and Timing Characteristics](#)). Use of the Continuous (R) commands with this sensor is deprecated and may not be supported in future versions of the sensor. Please see the Extended Read (XR) command as the preferred single-command measurement.

Please refer to Table 7 through Table 8 for an explanation of the command sequence and see Table 9 for an explanation of response parameters.

Table 7 HYDROS 21 aR0! measurement command sequence

Command	Response
aR0!	a±<depth>±<temperature>+<electricalConductivity>

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

Table 8 HYDROS 21 aR3! measurement command sequence

Command	Response
aR3!	a<TAB><depth> <temperature> <electricalConductivity> <meta><CR><sensorType><Checksum><CRC>

NOTE: This command does not adhere to the SDI-12 response format or timing. 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.

PARAMETERS

Table 9 lists the parameters, unit measurement, and a description of the parameters returned in command responses for HYDROS 21.

Table 9 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
<electricalConductivity>	µS/cm	Electrical conductivity
<depth>	mm	Depth—Values will typically range from 0 to 4,000 mm
<temperature>	°C	Temperature
<sensorType>	—	ASCII character denoting the sensor type For HYDROS 21, the character is c
<meta>	—	Auxilliary sensor information See Table 10
<suppressionState>	—	0: DDI Serial message present at power up 1: DDI Serial message suppressed at power up
<CRC>	—	METER 6-bit CRC
<Checksum>	—	METER serial checksum

SENSOR METADATA VALUE

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

Table 10 lists the possible error flags that can be set by the HYDROS 21. If multiple error flags are set, the sensor metadata integer value will be the sum of the individual values. To decode an integer value not explicitly in Table 10, find the largest error flag value that will fit in the integer value and accept that error as being present. Then, subtract that error flag value from the integer value and repeat the process on the remainder until the result is zero. For example, a sensor metadata integer value of 129 is the sum of the individual error flag values 128 + 1, so this sensor has corrupt firmware and experienced temperatures below freezing..

Table 10 Error flag values and issue resolution

Error Flag Value	Issue Present	Resolution
1	Sensor has experienced temperatures below freezing	Contact Customer Support to replace sensors
128	Sensor firmware is corrupt	Contact Customer Support for instructions on reloading firmware

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 0.0- 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.

The HYDROS 21 will omit the DDI Serial message when the SDI-12 address is nonzero or suppressed with the `ax01!` command. METER recommends suppressing the DDI Serial message when this signaling causes negative issues for a sensor measurement device.

NOTE: Out of the factory, all METER sensors start with SDI-12 address 0 and print out the DDI Serial message when power cycled.

DDI SERIAL TIMING

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

Table 11 DDI Serial communications configuration

Baud Rate	1,200 bps
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.

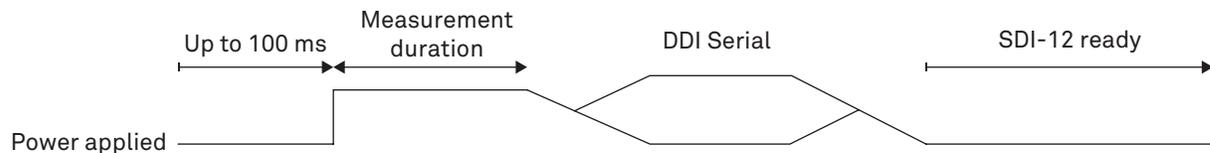


Figure 6 Data line DDI Serial timing

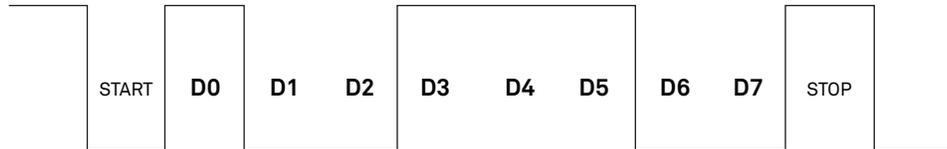


Figure 7 Example DDI Serial transmission of the character 9 (0x39)

DDI SERIAL RESPONSE

Table 12 details the DDI Serial response.

Table 12 HYDROS 21 DDI Serial response

COMMAND	RESPONSE
NA	<TAB><depth> <temperature> <electricalConductivity><CR><sensorType><Checksum><CRC>

NOTE: There is no actual command. The response is returned automatically upon power up. The values in this command are space delimited. As such, a + sign is not assigned between values and a - sign is only present if the value is negative.

DDI SERIAL CHECKSUM

These checksums are used in the continuous commands R3, and XR3, as well as the DDI Serial response. The legacy checksum is deprecated in favor of the CRC6 check character and may be ignored. The legacy checksum is computed using the characters after the sensor address (when used with the R3 or XR3 command) and includes the sensor identification character.

HYDROS 21 example input is <TAB>146 21.9 1034<CR>cFM and the resulting checksum output is F.

```
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 calculates a check character with the CRC-6-CDMA-A polynomial. The calculated CRC6 is added with the value 48 to make the results a printable character. The CRC6 is computed using the characters after the sensor address (if included) and includes the sensor identification character, and the legacy checksum.

HYDROS 21 CRC6 checksum example input is <TAB>146 21.9 1034<CR>cFM and the resulting checksum output is M.

```

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 = strien(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 meta data 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
00	4.2022	5.03	Initial HYDROS 21 Gen 2 release.