

TEROS-11 & 12 土壤水分・温度・ECセンサー インテグレーターズガイド

センサー概要

TEROS-11 土壤水分・温度センサーならびに TEROS-12 土壤水分、温度、EC センサーは、体積含水量（VWC）、土壤および培地の温度、および電気伝導率(TEROS-12のみ)をモニターする正確なツールです。TEROS-11/12 は、静電容量/周波数領域技術を使用して VWC を決定します。センサーは 70MHz の周波数を用いています。この領域ではテクスチャーや塩分の影響を最小限となるため、TEROS-11/12 は大半の土壤において正確な測定が可能です。TEROS-11/12 は、中央のニードル内部のサーミスタで温度を測定し、ステンレス鋼電極アレイを使用して電気伝導率を測定します。

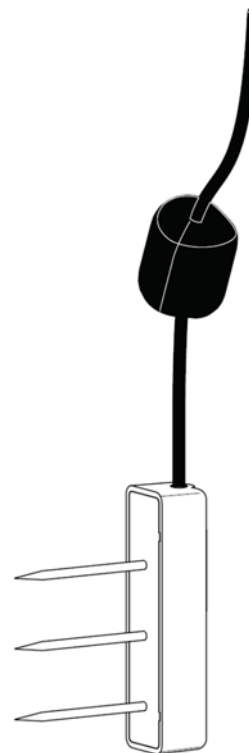
このセンサーがどのようにして測定を行うかについての詳細は、TEROS-11/12ユーザー・マニュアルをご参照ください。

アプリケーション

- 体積含水率の測定
- 土壤水分バランス
- 灌漑管理
- 土壤の電気伝導度の測定
- 土壤/培地の温度測定
- 施肥管理

利点

- 3つセンサーワイヤー：Power、Ground、Data
- デジタルセンサーは3つの測定値を通信します。
- 正確な測定のための丈夫なサーミスタ
- 低入力電圧
- 低電力設計は電池で動作するロガーに対応
- 腐食性の環境に耐性のエポキシ密封形成
- SDI-12、またはDDIシリアル通信プロトコルをサポート
- 低コストセンシングの為に最適化された最新設計



本ガイドの目的

METER 社は、このインテグレーター・ガイドの情報をセンサーとデータ収集機器またはフィールド・データロガーとの間の通信をお客様が確立するのに役立つように提供します。SDI-12 センサー通信をサポートするデータロガーを使用しているお客様は、データロガーのユーザー・マニュアルを参照してください。METER 社センサーは、プラグ・アンド・プレイ・センサー、携帯電話対応データロガー、およびデータ分析ソフトウェアの METER システムに完全に統合されています。

互換性ファームウェア・バージョン

このガイドは、ファームウェア・バージョン 1.07 以降と互換性があります。



測定仕様

体積含水率(VWC)	
測定範囲	砂質土壌 : 0~0.70m ³ /m ³ 土壌ではない媒体 : 0~1.0 m ³ /m ³ 誘電率 : 1(空気)~80(水) ※測定範囲はキャリブレーションされた媒体に依存します。カスタムキャリブレーションは、たいていの媒体の必要な範囲を可能にします。
分解能	0.001m ³ /m ³
精度	±0.03 m ³ /m ³ (8dS/m以下の砂質土壌) キャリブレーション時±1~2%VWC 誘電率 : 1~40±1(土壌レンジ) 誘電率 : 40~80±15%
測定周波数	70MHz
温度	
測定範囲	-40~60℃
分解能	0.1℃
精度	±0.5℃(0~60℃)、±1℃(-40~0℃)
適用可能なセンサーであっても、より長い平衡化時間のために、センサーが対象媒体に完全に浸されていない場合は、温度測定が正確にならない可能性があります。	
電気伝導度(EC) ※TEROS-12のみ	
測定範囲	0~20dS/m(bulk)
分解能	0.001dS/m
精度	±5%(0~10dS/m)、±8%(10~20dS/m)
出力	
SDI-12、またはDDI Serial	
データロガー互換	
METER社データロガー ZL6、Em50、EM60、または4~15VDCの電源を供給できるシリアル、またはSDI-12通信が可能なデータ収集システム	

物理特性

寸法	9.4(長さ)×2.4(幅)×7.5(高さ)cm
プローブ長さ	5.5cm
ケーブル長さ	5m(標準)、カスタムケーブル生産最大長さ75m 注意 : ケーブルを自身で延長された場合には通信状態を確認して下さい。
コネクタ	3.5mmステレオプラグ、または3芯先バラケーブル

電氣的、及びタイミング特性

供給電圧(VCC to GND)	
Minimum	4.0VDC
Typical	NA
Maximum	15.0VDC
デジタル入力電圧(logic High)	
Minimum	2.8V
Typical	3.6V
Maximum	3.9V
デジタル入力電圧(logic low)	
Minimum	-0.3V
Typical	0.00V
Maximum	0.8V
デジタル出力電圧(logic High)	
Minimum	NA
Typical	3.6V
Maximum	NA
電源スルー・レート	
Minimum	1.0V/ms
Typical	NA
Maximum	NA
電流ドレイン(25ms測定中)	
Minimum	3.0mA
Typical	3.6mA
Maximum	16.0mA

電流ドレイン(スリープ中)	
Minimum	NA
Typical	0.03mA
Maximum	NA
動作温度範囲	
Minimum	-40℃
Typical	NA
Maximum	+60℃
起動時間(DDIシリアル)	
Minimum	80ms
Typical	NA
Maximum	100ma
起動時間(SDI-12)	
Minimum	NA
Typical	245ms
Maximum	NA
測定時間	
Minimum	25ms
Typical	NA
Maximum	50ms
コンプライアンス	
ISO 9001-2015に基づき製造	
EM ISO/IEC 17050:2010(CE Mark)	
2014/30/EU and 2011/65/EU	
EN61326-1:2013 and EN55022/CISPR 22	

等価回路および接続タイプ

TEROS 11/12をデータロガーに接続するには、Figure2とFigure3を参照してください。Figure2は、推奨SDI-12仕様の低インピーダンス変種を示しています。

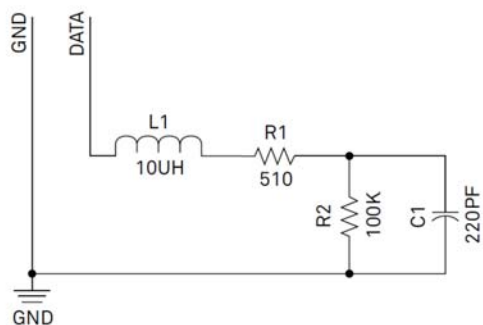


Figure 2 Equivalent circuit diagram

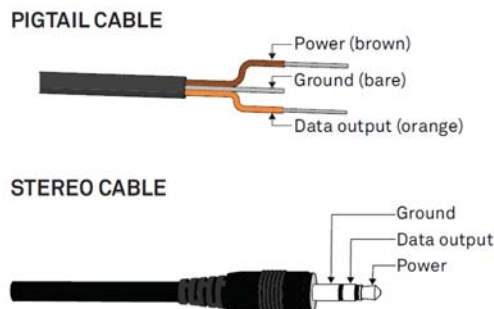


Figure 3 Connection types

安全性に関する注意事項

METER社センサーは最高水準で製作されていますが、誤用、不適切な保護、不適切な設置などはセンサーを損傷し、メーカー保証を無効にする可能性があります。センサーを各自のセンサー・ネットワークに統合する前に、推奨された設置方法に従って設置が行われており、センサーに損傷を与える乾燥から保護するための安全装置を実装してください。

サージ条件

センサーには、一般的なサージ条件から保護するための回路が組み込まれています。ただし、雷が発生しやすい場所に設置する場合は、特にセンサーがガード・パーティのロガーに接続されている場合は、特に注意が必要です。

より詳細な情報の項目については [Lightning surge and grounding practices](#) をご覧ください。

電源と接地

バス上のすべてのセンサーの最大センサー電流ドレインを同時にサポートするのに十分な電力があることを確認してください。データロガー装置への電力供給や接地が不適切なやり方で行われている場合、センサー保護回路が不十分になることがあります。データロガーのインストール手順を参照してください。不適切な接地は、センサーの性能だけでなくセンサーの出力にも影響を与える可能性があります。

より詳細な情報の項目については metergroup.com をご覧ください。

ケーブル

不適切に保護されているケーブルは、ケーブルの切断やセンサーの外れにつながる可能性があります。ケーブルの問題は、げっ歯類による損傷、センサーケーブル同士の乗っかり、ケーブルの引っかかり、取り付け時に十分なケーブルのたるみがないこと、センサー配線の接続不良など、さまざまな要因によって引き起こされる可能性があります。接続への緊張を軽減し、ゆるんでいるケーブルが誤って引っかからないようにするために、TEROS-11/12 とデータ収集装置の間を移動するケーブルを 1 か所以上の場所でマストに固定します。地面の近くの場合は、げっ歯類による損傷を避けるために、ケーブルを導管またはプラスチックの被覆材内に設置します。余分なケーブルをデータロガー・マストに接続して、ケーブルの重さによってセンサーがコネクタから抜けないようにします。

センサー通信

METER 社デジタル・センサーは、データ・ワイヤー上でセンサー測定値を通信するための送受信信号を共有するシリアル・インターフェイスを備えていることを特徴としています(Figure3)。センサーは 2 つの異なるプロトコルをサポートします：SDI-12 と DDI シリアル。各プロトコルには、実装上の利点と課題があります。目的のアプリケーションのプロトコル選択が明確でない場合は、カスタマーサポートまでご連絡ください。

SDI-12 の紹介

SDI-12 は、センサーをデータロガーおよびデータ収集機器に接続するための標準に基づくプロトコルです。固有のアドレスを持つ複数のセンサーは、共通の 3 線式バス（電源、グランド、およびデータ）を共有できます。センサーとロガー間の双方向通信は、規格で定義されているように送信と受信のためのデータラインを共有することによって可能です。センサー測定はプロトコル・コマンドによって起動されます。SDI-12 プロトコルでは、データロガーが特定のセンサーにコマンドを送信したり、特定のセンサーから読み取り値を受信したりできるように、バス上の各センサーに固有の英数字のセンサー・アドレスが必要です。

SDI-12 プロトコルの詳細を知るためには、[SDI-12 Specification v1.3](#) をダウンロードしてください。

DDI シリアルの紹介

DDI シリアル・プロトコルは、センサーからデータを収集するための METER 社データロガーが使用する方法です。このプロトコルは、センサーから受信機だけにデータを送信するように設定されたデータラインを使用します（シンプレックス）。通常、受信側は、デー

タを受信するためにビットバン方式を使用するマイクロプロセッサ UART または汎用 I/O ピンです。センサー測定は、センサーに電源を入れることによって起動されます。

センサーをコンピューターにインターフェイスする

センサーでサポートされているシリアル信号とプロトコルは、ほとんどのコンピューター（またはUSBからシリアルへのアダプター）にあるシリアル・ポートと互換のある種のインターフェイス・ハードウェアが必要です。市販されているSDI-12インターフェイス・アダプターはいくつかあります。ただし、METER社はこれらのインターフェイスのいずれもテストしていないため、どのアダプターがMETER社センサーと連携するかについて推奨することはできません

METER 社データロガーと ProCheck ハンドヘルドリーダーは、オンデマンド・センサー測定を実行するためのコンピューターとセンサー間のインターフェイスとして機能できます。

METER SDI-12 の実装

METER 社センサーは、SDI-12 標準センサー回路の低インピーダンス変種を使用しています (figure2)。起動時には、センサーはいくつかのセンサー診断情報を出力しますので、起動時間が経過するまで通信しないでください。起動時間後は、センサーは [SDI-12 Specification v1.3](#) に記載されているすべてのコマンドと互換性があります。ただし、連続測定コマンド (aR0~aR9 と aRC0~aRC9) および同時測定コマンド (aC~aC9 と aCC0~aCC9) を除きます。M、R、および C コマンドの実装は、8~9 ページにあります。aR3 および aR4 コマンドは METER 社システムで使用されています。その結果、SDI-12 規格で要求されている符号の除去の代わりに、スペースの除去を使用します。

工場出荷時には、すべての METER 社センサーは SDI-12 アドレス 0 から開始し、起動時には、DDI シリアル起動文字列を印字します。これは、METER SDI-12 以外のセンサーにより、疑似ブレイク条件とそれに続くランダムな一連のビットとして解釈できます。

SDI-12 アドレスがゼロ以外のとき、TEROS 11/12 は DDI シリアル起動文字列（センサー識別）を省略します。このため、アドレスをゼロ以外のアドレスに変更することをお勧めします。

センサー・バスの考慮事項

SDI-12 センサー・バスには、定期的な点検、センサーの維持、センサーのトラブルシューティングが必要です。1つのセンサーがダウンすると、残りのセンサーが正常に機能していても、バス全体が停止する可能性があります。センサーが故障しているときに SDI-12 バスの電源を入れ直すことは許容されますが、METER 社は SDI-12 バスに 1 日、1、2 回以上電源の入れ直しイベントをスケジュールすることを推奨しません。バス構成の有効性には多くの要因が影響します。より詳細な情報の項目とそれらを含むバーチャル・セミナーについては、metergroup.com をご覧ください。

SDI-12 の構成

表 1 に SDI-12 通信構成を示します。

表 1 SDI-12 通信構成

ボーレート	1200
スタート・ビット	1
データ・ビット	7 (LSB first)
パリティ・ビット	1 (even)
ストップ・ビット	1
論理	反転 (active low)

SDI-12のタイミング

すべてのSDI-12コマンドと応答は、データ行に関するFigure4の形式に準拠しなければコマンドと応答の両方にアドレスが先行し、キャリッジリターンとラインフィードの組み合わせ (<CR> <LF>) で終了し、Figure5に示すタイミングに従います。

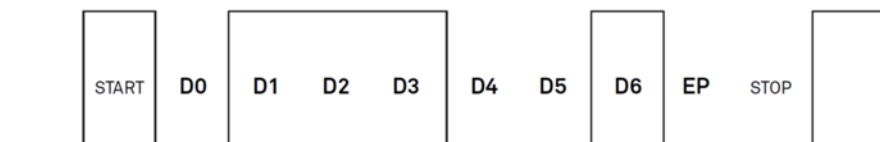


Figure 4 Example SDI-12 transmission of the character 1 (0x31)

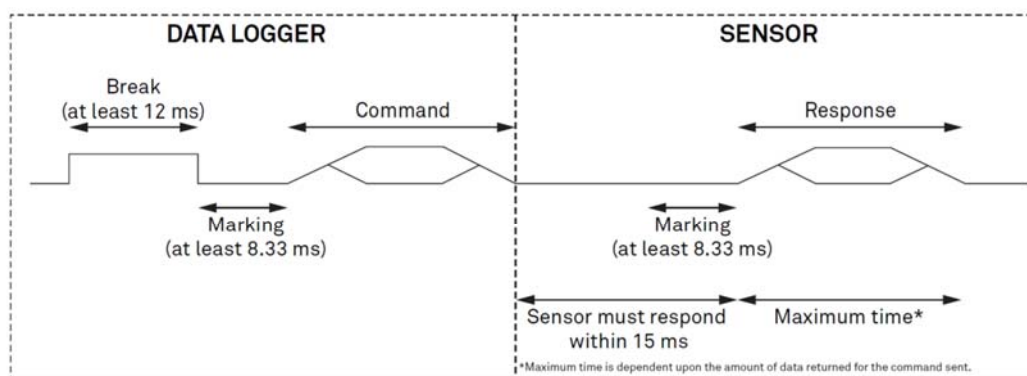


Figure 5 Example data logger and sensor communication

一般的な SDI 12 のコマンド

本節には、SDI-12システムでよく使用される一般的なSDI-12コマンドとそれに対応するMETER社センサーからの応答の表が含まれています。

識別コマンド (aI!)

識別コマンドは、接続されているセンサーに関するさまざまな詳細情報を取得するために使用できます。TEROS 11のコマンドと応答の例を例1に、TEROS 12のコマンドと応答の例を例2に示します。コマンドは太字で、応答はコマンドの後に続きます。

例 1 **I!**113METER_ _ _TER11_107631800001 (TEROS 11)

パラメーター	固定文字長さ	説明
I!	3	データロガー・コマンド センサー・アドレス1からの情報をセンサーに要求
1	1	センサー・アドレス： すべての応答の前に、これはバス上のどのセンサーが以下の情報を返しているかを示します。
13	2	ターゲット・センサーが SDI-12 Specification v1.3 をサポートしていることを示します。
METER_ _ _	8	ベンダー識別文字列 (METER社とすべてのMETER社センサー用の3つのスペース_ _ _)
TER11_	6	センサーモデル文字列

		この文字列はセンサータイプに固有なものです。TEROS11の場合、文字列はTER11です。
107	3	センサーバージョン この数を100で割ったものがMETER社センサーのバージョンです(例えば、107はバージョン1.07です)。
631800001	≤13, 変数	センサーシリアル番号 これは可変長フィールドです。古いセンサーの場合は省略されることがあります。

例2 1I!113METER_ _ _TER12_107631800001 (TEROS 12)

パラメーター	固定文字長さ	説明
1I!	3	データロガー・コマンド センサー・アドレス1からの情報をセンサーに要求
1	1	センサー・アドレス: すべての応答の前に、これはバス上のどのセンサーが以下の情報を返しているかを示します。
13	2	ターゲット・センサーが SDI-12 Specification v1.3 をサポートしていることを示します。
METER_ _ _	8	ベンダー識別文字列 (METER社とすべてのMETER社センサー用の3つのスペース_ _ _)
TER12_	6	センサーモデル文字列 この文字列はセンサータイプに固有なものです。TEROS12の場合、文字列はTER12です。
107	3	センサーバージョン この数を100で割ったものがMETER社センサーのバージョンです(例えば、107はバージョン1.07です)。
631800001	≤13, 変数	センサーシリアル番号 これは可変長フィールドです。古いセンサーの場合は省略されることがあります。

アドレス変更コマンド (aAB!)

アドレス変更コマンドは、センサー・アドレスを新しいアドレスに変更するために使用されます。このコマンド以外の他のすべてのコマンドは、ターゲット・センサー・アドレスとしてワイルドカード文字をサポートします。工場出荷時には、すべてのMETER社センサーのデフォルト・アドレスは0(ゼロ)です。サポートされているアドレスは英数字です(すなわち、a-z、A-Z、0-9です)。METER社センサーからの出力例を例3に示します。この例では、コマンドは太字で表示され、応答はコマンドの後に続きます。

例3 **1A0!0**

パラメーター	固定文字長さ	説明
1A0!	4	データロガー・コマンド アドレスを1から新しいアドレス0に変更するようにセンサーに要求します。
0	1	新しいセンサー・アドレス: その後のすべてのコマンドでは、この新しいアドレスがターゲットセンサーによって使用されます。

アドレス問い合わせコマンド (?!)

バスから切断されている間、アドレス問い合わせコマンドを使用して、現在どのセンサーと通信しているかを判断できます。このコマンドをバス経由で送信すると、すべてのセンサーが同時に応答してデータラインが破損するというバスの競合が発生します。このコマンドは、故障したセンサーを特定するときに役立ちます。例 4 はコマンドと応答の例を示します。コマンドは太字で、応答はコマンドの後に続きます。疑問符 (?) は、アドレス変更コマンド以外のコマンドでアドレスの代わりに使用できるワイルドカード文字です。

例4 ?!0

パラメーター	固定文字長さ	説明
?!	2	データロガー・コマンド データラインをリスニングしているセンサーからの応答を要求します。
0	1	センサー・アドレス： 現在接続されているセンサーにセンサーアドレスを返します。

コマンドの実施

以下の表は、関連する測定 (M)、連続 (R)、および同時 (C) コマンドと、必要に応じて後続のデータ (D) コマンドを示しています。

測定コマンドの実施

測定 (M) コマンドは SDI-12 バス上の単一のセンサーに送信され、そして続くデータ (D) コマンドがそのセンサーに送られ、バス上の別のセンサーとの通信を開始する前に、センサー出力データを取得するよう要求します。

コマンド・シーケンスの説明については表 2 を、応答パラメーターの説明については表 7 を参照してください。

表 2 aM! コマンド・シーケンス

コマンド	応答
このコマンドは瞬間値を報告します。	
aM!	atttn
aD0!	a+<calibratedCountsVWC>±<temperature>+<electricalConductivity>

注：<electricalConductivity>は TEROS 12 にのみ出力されます。

測定および対応するデータコマンドは、続けて用いることを目的としています。測定コマンドがセンサーによって処理された後、サービス要求<CR> <LF>がセンサーから送信され、測定の準備が整います。データコマンドを送信する前に、ttt 秒が経過するまで待つか、サービス要求を受け取るまで待ちます。詳細については [SDI-12 Specification v1.3](#) を参照してください。

同時測定コマンドの実施

同時 (C) 測定コマンドは通常、バスに接続されているセンサーで使用されます。このセンサーの同時 (C) コマンドは、標準の C コマンドの実施とは異なります。まず、C コマンドを送信し、C コマンドの応答で詳述されている指定された時間待機してから、他のセンサーと通信する前に D コマンドを使用してその応答を読み取ります。

コマンド・シーケンスの説明については表 3 を、応答パラメーターの説明については表 7 を参照してください。

**表 3 aC! 測定コマンド・シーケンス**

コマンド	応答
このコマンドは瞬間値を報告します。	
aC!	attnn
aD0!	a+<calibratedCountsVWC>±<temperature>+<electricalConductivity>

注：<electricalConductivity>は TEROS 12 にのみ出力されます。

測定および対応するデータコマンドは、続けて用いることを目的としています。測定コマンドがセンサーによって処理された後、サービス要求<CR> <LF>がセンサーから送信され、測定の準備が整います。データコマンドを送信する前に、ttt秒が経過するまで待つか、サービス要求を受け取るまで待ちます。詳細については[SDI-12 Specification v1.3](#)を参照してください。

連続測定コマンドの実施

連続 (R) 測定コマンドはセンサー測定を起動し、読み取りが完了した後に D コマンドを送信する必要なしに自動的にデータを返します。

コマンド・シーケンスの説明については表 4 から表 6 を、応答パラメーターの説明については表 7 を参照してください。

表 4 aR0! 測定コマンド・シーケンス

コマンド	応答
このコマンドは瞬間値を報告します。	
aR0!	a+<calibratedCountsVWC>±<temperature>+<electricalConductivity>

注：<electricalConductivity>は TEROS 12 にのみ出力されます。

このコマンドは、SDI-12 のレスポンス・タイミングに準拠していません。詳細については、METER SDI-12 の実装を参照してください。

表 5 aR3! 測定コマンド・シーケンス

コマンド	応答
このコマンドは瞬間値を報告します。	
aR3!	a<TAB><calibratedCountsVWC> <temperature> <electricalConductivity><CR><sensorType><Checksum><CRC>

注：<electricalConductivity>は TEROS 12 にのみ出力されます。

このコマンドは、SDI-12 のレスポンス・フォーマットあるいはタイミングに準拠していません。詳細については、METER SDI-12 の実装を参照してください。

表 6 aR4! 測定コマンド・シーケンス

コマンド	応答
このコマンドは瞬間値を報告します。	
aR4!	a<TAB><calibratedCountsVWC> <temperature> <electricalConductivity><CR><sensorType><Checksum><CRC>

注：<electricalConductivity>は TEROS 12 にのみ出力されます。

このコマンドは、SDI-12 のレスポンス・フォーマットあるいはタイミングに準拠していません。詳細については、METER SDI-12 の実装を参照してください。

パラメーター

表 7 は、パラメーター、単位測定、および TEROS 11/12 のコマンド応答で返されるパラメーターの説明をリストしています。

表 7 パラメーターの説明

パラメーター	単位	説明
±	—	次に続く値の符号を表す正または負の符号
a	—	SDI-12アドレス
n	—	測定数(固定幅1)
nn	—	必要に応じて先行ゼロ付き測定数(固定幅2)
ttt	s	最大時間測定にかかる(固定幅3)
<TAB>	—	タブ文字
<CR>	—	キャリッジリターン文字
<LF>	—	ラインフィード文字
<calibratedCountsVWC>	—	体積含水率に対して校正されたADC(ADコンバーター)のカウント数
<temperature>	℃	大気温度
<electricalConductivity>	dS/m	25℃で正規化されたバルク電気伝導度(TEROS-12のみ)
<sensorType>	—	センサータイプを指定するASCII文字 TEROS-11の場合の文字はh、TEROS-12の場合の文字はgです。
<Checksum>	—	METER社シリアル・チェックサム
<CRC>	—	METER社6ビットCRC

DDI シリアル通信

DDI シリアル通信プロトコルは、各センサーに専用のシリアル信号線を持っているか、または、マルチプレクサーを使用して複数のセンサーを処理するシステムに最適です。シリアル通信は、0~3.6 V の信号レベルを使用してアクティブ・ハイのロジック・レベルをサポートする多くの TTL シリアル実装と互換性があります。センサーに最初に電源が入ったとき、組み込まれたトランスデューサーが自動的に測定を行ない、次にデータラインを介して応答が出力されます。このプロトコルを使用するシステムは、センサーからのデータ転送を開始するためにセンサーの励起を制御します。このプロトコルは、METER 社がデジタル・センサーとデータロガーのラインを改良し拡張するにつれて変更されることもあります。SDI-12 アドレスがゼロ以外のとき、TEROS 11/12 は DDI シリアル起動文字列を省略します。

注：工場出荷時には、すべてのMETER社センサーはSDI-12アドレス0から始まり、電源を入れ直すと起動文字列を印字します。

DDI シリアル・タイミング

表 8 に DDI シリアル通信構成を示します。

表 8 DDI シリアル通信構成

ボーレート	1200
スタート・ビット	1
データ・ビット	8 (LSB first)
パリティ・ビット	0 (無し)
ストップ・ビット	1
論理	標準 (active High)

電源投入時に、センサーは 100 ms 以内にデータラインをハイに引き上げて、センサーが読み取りを行っていることを示します (Figure 6)。読み取りが完了すると、センサーはデータラインから Figure 7 に示すフォーマットに準拠してシリアル信号の送信を開始します。データが送信されると、センサーは SDI-12 通信モードに入ります。別のシリアル信号を取得するためには、センサーの電源を入れ直さなければなりません。

注: 時々センサーからのシグナリングは典型的なマイクロプロセッサ UART を混乱させることもあります。測定中、センサーはデータラインをローに保ちます。センサーはラインをハイにして、測定値を送信することをロガーに合図します。それから、センサーは最初のデータ・バイトのクロックアウトを開始する前に、典型的なスタートビット (ロー) から開始して追加の測定を行うことがあります。最初のスタート・ビットが送信されると、典型的なシリアル・タイミングが有効になります。ただし、この時点より前の信号遷移はシリアル信号ではなく、UART によって誤って解釈される可能性があります

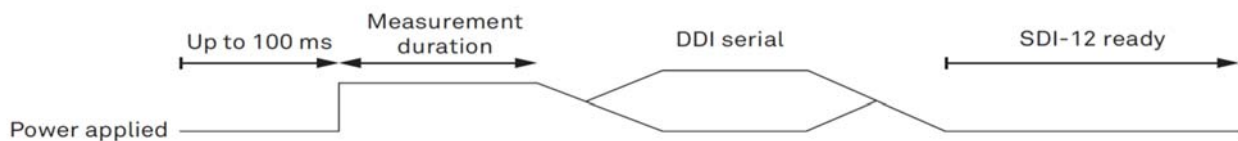


Figure 6 Data line DDI serial timing

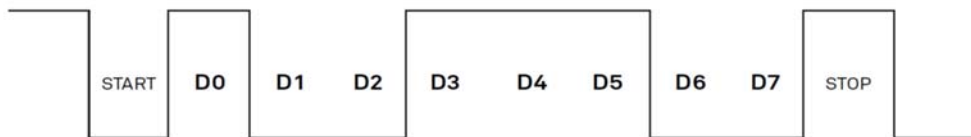


Figure 7 Example DDI serial transmission of the character 9 (0x39)

DDI シリアル応答

本節は、DDI シリアル応答を詳述する表を含みます。

表 9 DDI シリアル応答

コマンド	応答
NA	<TAB><calibratedCountsVWC> <temperature> <electricalConductivity><CR><sensorType><Checksum><CRC>

注: <electricalConductivity>は TEROS 12 にのみ出力されます。

実際のコマンドはありません。応答は電源投入時に自動的に返されます。

DDI シリアル・チェックサム

これらのチェックサムは、DDI シリアル応答と同様に連続コマンド R3 と R4 で使用されます。レガシー・チェックサムは、センサー・アドレスを除く、センサー識別文字への送信の開始から計算されます。

TEROS-11 のレガシー・チェックサムの例

レガシー・チェックサムの例の入力は<TAB> 1797.7 21.8 <CR> h で、結果としてのチェックサム出力は D です。

TEROS-12 のレガシー・チェックサムの例

レガシー・チェックサムの入力例は<TAB> 2749.0 23.8 660<CR>g で、結果としてのチェックサム出力は 8 です。

```
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 meta data 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

より堅牢な CRC6 は、利用可能であれば、CRC-6-CDMA2000-A 多項式を結果に値 48 を加えて使用して、これを印刷可能文字にし、送信の最初からレガシー・チェックサム文字までを、センサー・アドレスを除いて計算されます。

CRC チェックサムの入力例は<TAB>1797.2 21.8<CR>hD で、結果としてのチェックサム出力は 2 です。

TEROS 12 CRC6

より堅牢な CRC6 は、利用可能であれば、CRC-6-CDMA2000-A 多項式を結果に値 48 を加えて使用して、これを印刷可能文字にし、送信の最初からレガシー・チェックサム文字までを、センサー・アドレスを除いて計算されます。

CRC6 チェックサムの入力例は<TAB>2749.0 23.8 660<CR>g8 で、結果としてのチェックサム出力は 0 (大文字の 0) です。

```
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);
}
```