Modbus TCP/IP イーサネットドライバー

目次

Modbus TCP/IP イーサネットドライバー 1
目次 2
Modbus TCP/IP イーサネットドライバー 5
概要 6
サポートされるデバイスモデル 6
設定 7
チャネルのプロパティ - 一般 7
タグ数 8
チャネルのプロパティ - イーサネット通信 8
チャネルのプロパティ - 書き込み最適化 8
チャネルのプロパティ - 詳細 9
チャネルのプロパティ - 通信シリアル化 10
チャネルのプロパティ - イーサネット 11
デバイスのプロパティ - 一般 12
デバイスのプロパティ - スキャンモード 13
デバイスのプロパティ - タイミング 14
デバイスのプロパティ - 自動格下げ 15
デバイスのプロパティ - タグ生成 15
デバイスのプロパティ - 変数のインポート設定 17
デバイスのプロパティ - 非送信請求 17
Modbus クライアントと Modbus サーバーに関する考慮事項 19
デバイスのプロパティ - エラー処理 19
デバイスのプロパティ - イーサネット 20
デバイスのプロパティ - 設定 20
デバイスのプロパティ - ブロックサイズ 23
デバイスのプロパティ - 冗長 25
構成 API Modbus Ethernet の例 26
列挙 27
デバイスモデル列挙 28
自動タグデータベース生成 29
カスタムアプリケーションからのインポート 29
通信の最適化 29
データ型の説明 31
アドレスの説明 32
ドライバーのシステムタグのアドレス指定 32
ファンクションコードの説明 33
Applicom のサブモデルとアドレス指定 33
ジェネリック Modbus のアドレス指定 33
TSX Quantum 36
TSX Premium 40
CEG のアドレス指定 43

Fluenta のアドレス指定 43
Instromet のアドレス指定 43
メールボックスのアドレス指定 43
Modbus のアドレス指定 44
Roxar のアドレス指定 47
イベントログメッセージ 48
Winsock 通信を開始できませんでした。 48
非送信請求通信を開始できませんでした。 48
未定義のデバイスに対して非送信請求メールボックスアクセスが行われました。ソケットを閉じています。| IP アドレス = '<アドレス>'。 48
受信した要求は非送信請求メールボックスでサポートされていません。| IP アドレス = '<アドレス>'。 48
非送信請求メールボックスのメモリ割り当てエラー。| IP アドレス = '<アドレス>'。 49
ソケット接続を作成できません。 49
タグデータベースのインポート用のファイルを開くときにエラーが発生しました。| OS エラー = '<エラー>'。 49
不良配列。| 配列範囲 = <開始> ～ <終了>。 49
ブロックに不良アドレスがあります。| ブロック範囲 = <アドレス> から <アドレス>。 49
ホストの解決に失敗しました。| ホスト名 = '<名前>'。 50
指定された出力コイルブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (コイル)、最大ブロックサイズ = <数値> (コイル)。 50
指定された入力コイルブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (コイル)、最大ブロックサイズ = <数値> (コイル)。 50
指定された内部レジスタブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (レジスタ)、最大ブロックサイズ = <数値> (レジスタ)。 50
指定された保持レジスタブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (レジスタ)、最大ブロックサイズ = <数値> (レジスタ)。 50
ブロック要求で例外が返されました。| ブロック範囲 = <アドレス> から <アドレス>、例外 = <コード>。 50
ブロック要求で例外が返されました。| ブロック範囲 = <アドレス> から <アドレス>、関数コード = <コード>、例外 = <コード>。 51
受信したブロックの長さは不適切です。| ブロック範囲 = <開始> ～ <終了>。 51
メモリリソース量の低下によりタグインポートが失敗しました。 51
タグのインポート中にファイル例外が発生しました。 51
インポートファイルのレコードの解析でエラーが発生しました。| レコード番号 = <数値>、フィールド = <field>。 51
インポートファイルのレコードの説明が切り詰められました。| レコード番号 = <数値>。 52
インポートされたタグ名が無効のため変更されました。| タグ名 = '<タグ>'、変更後のタグ名 = '<タグ>'。 52
データ型がサポートされていないため、タグをインポートできませんでした。| タグ名 = '<タグ>'、サポートされていないデータ型 = '<タイプ>'。 52
アドレスに書き込めません。デバイスは例外を返しました。| アドレス = '<アドレス>'、例外 = <コード>。 52
イーサネットマネージャが開始されました。 53
イーサネットマネージャが停止しました。 53
タグデータベースをインポートしています。| ソースファイル = '<ファイル名>'。 53
クライアントアプリケーションはシステムタグ _CEGExtension を介して CEG 拡張を変更しました。| 拡張 = '<拡張>'。 53
非送信請求通信を開始しています。| プロトコル = '<名前>'、ポート = <数値>。 53
Modbus サーバーデバイス用のメモリが作成されました。| Modbus サーバーデバイス ID = <デバイス>。 53
すべてのチャネルが仮想ネットワークを利用しているか、すべてのデバイスがリモートアドレスを受信しているため、非送信請求通信を停止しています。 53

チャネルは仮想ネットワーク内にあるため、すべてのデバイスがデバイスにつき 1 つのソケットを使用する設定に戻りました。 53
接続されているクライアントでデバイス ID を Modbus クライアントモードからサーバーモードに変更できません。 53
接続されているクライアントでデバイス ID を Modbus サーバーモードからクライアントモードに変更できません。 54
チャネルが仮想ネットワーク内にある場合、Modbus サーバーモードは許可されません。デバイス ID にループバックアドレスまたはローカル IP アドレスが含まれていてはなりません。 54
チャネルが仮想ネットワーク内にある場合、メールボックスモデルは許可されません。 54
Modbus 例外コード 55
Modbus Ethernet チャネルのプロパティ 56
Modbus Ethernet デバイスのプロパティ 56
Modbus Ethernet タグのプロパティ 56
索引 **58**

Modbus TCP/IP イーサネットドライバー

ヘルプバージョン 1.151

目次

概要
Modbus TCP/IP イーサネットドライバー とは

設定
このドライバーを使用するためにチャネルとデバイスを構成する方法

API を使用した設定
構成 API を使用してチャネルとデバイスを設定する方法

自動タグデータベース生成
Modbus TCP/IP イーサネットドライバー 用にタグを設定する方法

通信の最適化
Modbus TCP/IP イーサネットドライバー から最高のパフォーマンスを得る方法

データ型の説明
Modbus TCP/IP イーサネットドライバーでサポートされるデータ型

アドレスの説明
Modbus イーサネットデバイスでデータ位置を参照する方法

イベントログメッセージ
Modbus TCP/IP イーサネットドライバー で生成されるメッセージ

概要

Modbus TCP/IP イーサネットドライバー は Modbus イーサネットデバイスが HMI、SCADA、Historian、MES、ERP や多数のカスタムアプリケーションを含むクライアントアプリケーションに接続するための信頼性の高い手段を提供します。このドライバーを使用するためには、ユーザーは TCP/IP を適切にインストールする必要があります。 *セットアップの詳細については、Windows のドキュメントを参照してください。*

注記: 処理中にエラーが発生した場合、このドライバーはメッセージを通知します。

サポートされるデバイスモデル

Applicom

このモデルはジェネリック Modbus、TSX Premium、および TSX Quantum デバイス用に Applicom アドレス指定構文をサポートしています。

イーサネットから Modbus Plus へのブリッジ

このドライバーはイーサネットから Modbus Plus へのブリッジを介して Modbus Plus デバイスと通信できます。使用するデバイス ID は、ブリッジの IP アドレスと Modbus Plus ブリッジインデックスを組み合わせたものになります。たとえば、ブリッジ IP が 205.167.7.12、ブリッジインデックスが 5 の場合、デバイス ID は 205.167.7.12.5 となります。MBE から MBP へのブリッジを取得して設定する方法については、Modicon/Schneider Automation の販売代理店までお問い合わせください。

CEG

このモデルは CEG デバイスの拡張ブロックサイズをサポートしています。

Fluenta

このモデルは Fluenta FGM 100/130 フローコンピュータの非標準 Modbus マッピングをサポートしています。

Instromet

このモデルは Instromet デバイスの非標準 Modbus マッピングをサポートしています。

メールボックス

このモデルは非送信請求要求の処理方法に影響を与えます。メールボックスデバイスを定義することによって、ドライバーはネットワーク上の PLC として機能しなくなります。代わりに、定義された各メールボックスデバイスのストレージ領域として機能するようになります。ドライバーは非送信請求コマンドを受信すると、メッセージの送信元の IP アドレスを検出し、そのデバイスに割り当てられているストレージ領域内にデータを配置します。IP アドレスがメールボックスデバイスとして定義されていないデバイスからメッセージが送信されている場合、そのメッセージは処理されません。このタイプのデバイスを読み書きするクライアントアプリケーションは、物理デバイス内ではなくドライバー内のストレージ領域を読み書きします。

Modbus TCP/IP イーサネットドライバーに非送信請求要求を送信する方法については、Modicon のドキュメントで MSTR 命令のトピックを参照してください。

注記: Modbus メールボックスでは ファンクションコード 22 (0x16) はサポートされていません。0x10 (複数の保持レジスタへの書き込み) と 0x6 (単一の保持レジスタへの書き込み) はサポートされています。デバイスのプロパティで「保持レジスタのビット書き込み」を無効化すると、個々のビットに書き込むことが可能です。これにより、ビットに直接書き込む代わりに、読み取り/修正/書き込みシーケンスが使用されます。これを機能させるためには、(メールボックスではなく) クライアント Modbus デバイスの設定だけを変更する必要があります。

メールボックスデバイスモデルに対するメールボックスクライアントの権限

Modbus クライアント

ほとんどのプロジェクトは、Modbus クライアントとして機能するように設定されています。このモードでは、ドライバーは物理デバイスにアクセスします (TSX Quantum や、Modbus オープンイーサネットと互換性があるその他のデバイスなど)。

Modbus 非送信請求またはサーバーモード

Modbus がモデルとして選択され、そのデバイス ID としてホストマシンの IP アドレスが設定されている場合、Modbus TCP/IP イーサネットドライバーはネットワーク上のデバイスとして機能します。ドライバーは受信したすべての非送信請求コマンドを受け入れて、別の PLC であるかのようにこれらの処理を試みます。ネットワーク上のすべての Modbus クライアントは、このシミュレーション対象デバイスの ID を使用して、このデバイスと通信することができます。

YYY.YYY.YYY.YYY.XXX として Modbus サーバーデバイスのデバイス ID が指定されます。*YYY* には、ループバックアドレスか、そのドライバーを実行している PC のローカル IP アドレスを指定できます。*XXX* には、Modbus サーバーのステーション ID を 0 から 255 までの範囲で指定します。

複数の Modbus サーバーデバイスで同じステーション ID を指定することができます。その場合、ステーション ID を共有するすべてのデバイスが 1 つの共通のシミュレーション対象デバイスを指します。存在しない Modbus サーバーデバイス (ステーション ID) のデータをリモート Modbus クライアントが要求した場合、その応答にはステーション 0 からのデータが含まれます。プロジェクト内に Modbus サーバーデバイスが作成されると、その Modbus サーバーが有効になり、サーバーがシャットダウンするまで有効なままとなります。ステーション ID を変更すると新しい Modbus サーバーデバイスが有効になり、サーバーがシャットダウンするまで有効なままとなります。

1 から 65536 のアドレスは出力コイル、入力コイル、内部レジスタ、保持レジスタ用に実装されています。非送信請求モードでは、ドライバーは外部デバイスからのこれらの値を読み書きする有効なすべての要求に応答します (ファンクションコード [10 進] 01、02、03、04、05、06、15、16)。さらに、このドライバーにはループバック (ファンクションコード 08、サブコード 00) が実装されています。これらの位置には、Modbus サーバーデバイスに割り当てられたタグとしてホスト PC からローカルにアクセスすることができます。
注記: 非送信請求デバイスでは書き込み専用アクセスは許可されません。

Roxar
このモデルは Roxar RFM 含水率メーターの非標準 Modbus マッピングをサポートしています。
関連項目: *デバイスモデル列挙*と*デバイスのプロパティ*を参照してください。

設定

チャネルとデバイスの制限値
このドライバーでサポートされているチャネルの最大数は 1024 です。このドライバーでサポートされているデバイスの最大数は、1 つのチャネルにつき 8192 です。

ヒント: チャネルレベルの設定はこのチャネルで設定されているすべてのデバイスに適用されます。

注記: Modbus TCP/IP イーサネットドライバー では Winsock V1.1 以上が必要です。

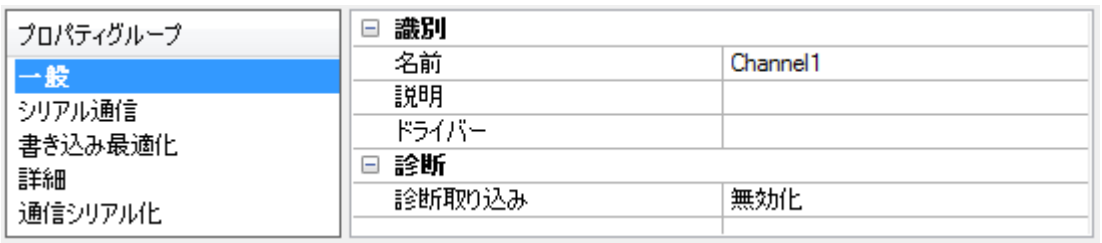
通信シリアル化
Modbus TCP/IP イーサネットドライバー では、データ転送を一度に 1 つのチャネルに制限するかどうかを指定する通信シリアル化がサポートされています。
詳細については、通信シリアル化を参照してください。

注記:

- チャネルのシリアル化を有効にすると、非送信請求通信と「デバイスあたりの最大ソケット数」プロパティが無効になります。チャネルシリアル化ではメールボックスモデルは使用できません。
- すべてのプロパティをすべてのモデルで使用できるわけではありません。

関連項目: 構成 API コマンドを使用したデバイスの設定、Modbus を使用した API の例

チャネルのプロパティ - 一般
このサーバーでは、複数の通信ドライバーを同時に使用することができます。サーバープロジェクトで使用される各プロトコルおよびドライバーをチャネルと呼びます。サーバープロジェクトは、同じ通信ドライバーまたは一意の通信ドライバーを使用する多数のチャネルから成ります。チャネルは、OPC リンクの基本的な構成要素として機能します。このグループは、識別属性や動作モードなどの一般的なチャネルプロパティを指定するときに使用します。



識別

「名前」: このチャネルのユーザー定義識別情報を指定します。各サーバープロジェクトで、それぞれのチャネル名が一意でなければなりません。名前は最大 256 文字ですが、一部のクライアントアプリケーションではOPC サーバーのタグ空間をブラウズする際の表示ウィンドウが制限されています。チャネル名はOPC ブラウザ情報の一部です。チャネルの作成にはこのプロパティが必要です。
予約済み文字の詳細については、サーバーのヘルプで「チャネル、デバイス、タグ、およびタググループに適切な名前を付ける方法」を参照してください。

「説明」: このチャネルに関するユーザー定義情報を指定します。
「説明」などのこれらのプロパティの多くには、システムタグが関連付けられています。

「ドライバー」: このチャネル用のプロトコル/ドライバーを指定します。このプロパティでは、チャネル作成時に選択されたデバイスドライバーが示されます。チャネルのプロパティではこの設定を変更することはできません。チャネルの作成にはこのプロパティが必要です。
注記: サーバーがオンラインで常時稼働している場合、これらのプロパティをいつでも変更できます。これには、クライアントがデータをサーバーに登録できないようにチャネル名を変更することも含まれます。チャネル名を変更する前にクライアントがサーバーからアイテムをすでに取得している場合、それらのアイテムは影響を受けません。チャネル名が変更された後で、クライアントアプリケーションがそのアイテムを解放し、古いチャネル名を使用して再び取得しようとしても、そのアイテムは取得されません。大規模なクライアントアプリケーションを開発した場合は、プロパティを変更しないようにしてください。オペレータがプロパティを変更したりサーバーの機能にアクセスしたりすることを防ぐため、適切なユーザー役割を使用し、権限を正しく管理する必要があります。

診断

「診断取り込み」: このオプションが有効な場合、チャネルの診断情報がOPC アプリケーションに取り込まれ、。サーバーの診断機能は最小限のオーバーヘッド処理を必要とするので、必要なときにだけ利用し、必要がないときには無効にしておくことをお勧めします。デフォルトでは無効になっています。
注記: ドライバーで診断機能がサポートされていない場合、このプロパティは使用できません。
詳細については、サーバーのヘルプの「通信診断」および「統計タグ」を参照してください。

タグ数

「静的タグ」: デバイスレベルまたはチャネルレベルで定義される静的タグの数を指定します。この情報は、トラブルシューティングと負荷分散を行う場合に役立ちます。

チャネルのプロパティ - イーサネット通信

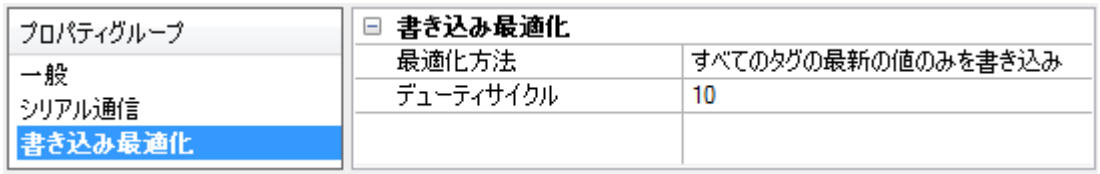
イーサネット通信を使用してデバイスと通信できます。

イーサネット設定

「ネットワークアダプタ」: バインドするネットワークアダプタを指定します。空白のままにするか、「デフォルト」を選択した場合、オペレーティングシステムはデフォルトのアダプタを選択します。

チャネルのプロパティ - 書き込み最適化

サーバーは、クライアントアプリケーションから書き込まれたデータをデバイスに遅延なく届ける必要があります。このため、サーバーに用意されている最適化プロパティを使用して、特定のニーズを満たしたり、アプリケーションの応答性を高めたりすることができます。



書き込み最適化

「最適化方法」: 基礎となる通信ドライバーに書き込みデータをどのように渡すかを制御します。以下のオプションがあります。

- **「すべてのタグのすべての値を書き込み」**: このオプションを選択した場合、サーバーはすべての値をコントローラに書き込もうとします。このモードでは、サーバーは書き込み要求を絶えず収集し、サーバーの内部書き込みキューにこれらの要求を追加します。サーバーは書き込みキューを処理し、デバイスにできるだけ早くデータを書き込むことによって、このキューを空にしようとします。このモードでは、クライアントアプリケーションから書き込まれたすべてのデータがターゲットデバイスに送信されます。ターゲットデバイスで書き込み操作の順序または書き込みアイテムのコンテンツが一意に表示される必要がある場合、このモードを選択します。
- **「非 Boolean タグの最新の値のみを書き込み」**: デバイスにデータを実際に送信するのに時間がかかっているために、同じ値への多数の連続書き込みが書き込みキューに累積することがあります。書き込みキューにすでに置かれている書き込み値をサーバーが更新した場合、同じ最終出力値に達するまでに必要な書き込み回数ははるかに少なくなります。このようにして、サーバーのキューに余分な書き込みが累積することがなくなります。ユーザーがスライドスイッチを動かすのをやめると、ほぼ同時にデバイス内の値が正確な値になります。モード名からもわかるように、Boolean 値でない値はサーバーの内部書き込みキュー内で更新され、次の機会にデバイスに送信されます。これによってアプリケーションのパフォーマンスが大幅に向上します。
  注記: このオプションを選択した場合、Boolean 値への書き込みは最適化されません。モーメンタリプッシュボタンなどの Boolean 操作で問題が発生することなく、HMI データの操作を最適化できます。
- **「すべてのタグの最新の値のみを書き込み」**: このオプションを選択した場合、2 つ目の最適化モードの理論がすべてのタグに適用されます。これはアプリケーションが最新の値だけをデバイスに送信する必要がある場合に特に役立ちます。このモードでは、現在書き込みキューに入っているタグを送信する前に更新することによって、すべての書き込みが最適化されます。これがデフォルトのモードです。

「デューティサイクル」: 読み取り操作に対する書き込み操作の比率を制御するときに使用します。この比率は必ず、読み取り 1 回につき書き込みが 1 から 10 回の間であることが基になっています。デューティサイクルはデフォルトで 10 に設定されており、1 回の読み取り操作につき 10 回の書き込みが行われます。アプリケーションが多数の連続書き込みを行っている場合でも、読み取りデータを処理する時間が確実に残っている必要があります。これを設定すると、書き込み操作が 1 回行われるたびに読み取り操作が 1 回行われるようになります。実行する書き込み操作がない場合、読み取りが連続処理されます。これにより、連続書き込みを行うアプリケーションが最適化され、データの送受信フローがよりバランスのとれたものとなります。
注記: 本番環境で使用する前に、強化された書き込み最適化機能との互換性が維持されるようにアプリケーションのプロパティを設定することをお勧めします。

チャネルのプロパティ - 詳細

このグループは、チャネルの詳細プロパティを指定するときに使用します。すべてのドライバーがすべてのプロトコルをサポートしているわけではないので、サポートしていないデバイスには詳細グループが表示されません。

「非正規化浮動小数点処理」: 非正規化値は無限、非数 (NaN)、または非正規化数として定義されます。デフォルトは「ゼロで置換」です。ネイティブの浮動小数点処理が指定されているドライバーはデフォルトで「未修正」になります。「非正規化浮動小数点処理」では、ドライバーによる非正規化 IEEE-754 浮動小数点データの処理方法を指定できます。オプションの説明は次のとおりです。

- **「ゼロで置換」**: このオプションを選択した場合、ドライバーが非正規化 IEEE-754 浮動小数点値をクライアントに転送する前にゼロで置き換えることができます。
- **「未修正」**: このオプションを選択した場合、ドライバーは IEEE-754 非正規化、正規化、非数、および無限の値を変換または変更せずにクライアントに転送できます。

注記: ドライバーが浮動小数点値をサポートしていない場合や、表示されているオプションだけをサポートする場合、このプロパティは無効になります。チャネルの浮動小数点正規化の設定に従って、リアルタイムのドライバータグ (値や配列など) が浮動小数点正規化の対象となります。たとえば、EFM データはこの設定の影響を受けません。

浮動小数点値の詳細については、サーバーのヘルプで「非正規化浮動小数点値を使用する方法」を参照してください。

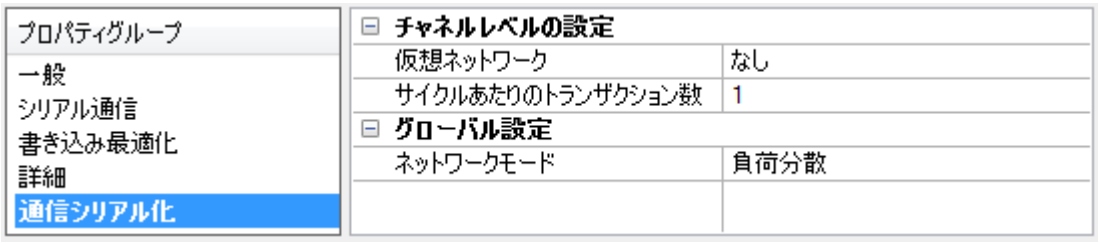
「デバイス間遅延」: 通信チャネルが同じチャネルの現在のデバイスからデータを受信した後、次のデバイスに新しい要求を送信するまで待機する時間を指定します。ゼロ (0) を指定すると遅延は無効になります。

注記: このプロパティは、一部のドライバー、モデル、および依存する設定では使用できません。

チャネルのプロパティ - 通信シリアル化

サーバーのマルチスレッドアーキテクチャにより、チャネルはデバイスとの並列通信が可能になります。これは効率的ですが、物理ネットワークに制約がある (無線イーサネットなど) 場合には通信をシリアル化できます。通信シリアル化によって、仮想ネットワーク内で同時に通信可能なチャネルは 1 つに制限されます。

"仮想ネットワーク" という用語は、通信に同じパイプラインを使用するチャネルと関連デバイスの集合を表します。たとえば、無線イーサネットのパイプラインはクライアント無線です。同じクライアント無線を使用しているチャネルは、すべて同じ仮想ネットワークに関連付けられています。チャネルは "ラウンドロビン" 方式で 1 つずつ順番に通信できます。デフォルトでは、チャネルが 1 つのトランザクションを処理した後で、通信を別のチャネルに渡します。トランザクションには 1 つ以上のタグが含まれることがあります。要求に応答しないデバイスが制御チャネルに含まれている場合、そのトランザクションがタイムアウトになるまでチャネルは制御を解放できません。これによって、仮想ネットワーク内のその他のチャネルでデータ更新の遅延が生じます。



チャネルレベルの設定

「仮想ネットワーク」: 通信シリアル化のチャネルのモードを指定します。オプションには「なし」、「ネットワーク 1」 - 「ネットワーク 500」があります。デフォルトは「なし」です。オプションの説明は次のとおりです。

- **「なし」**: このオプションを選択した場合、チャネルの通信シリアル化は無効になります。
- **「ネットワーク 1」 - 「ネットワーク 500」**: このオプションでは、チャネルを割り当てる仮想ネットワークを指定します。

「サイクルあたりのトランザクション数」: チャネルで実行可能な単一ブロック/非ブロック読み取り/書き込みトランザクションの数を指定します。あるチャネルが通信する機会を得ると、この数だけトランザクションが試みられます。有効な範囲は 1 から 99 です。デフォルトは 1 です。

グローバル設定

「ネットワークモード」: このプロパティでは、チャネル通信を委譲する方法を制御します。**「負荷分散」**モードでは、各チャネルが 1 つずつ順番に通信する機会を得ます。**「優先順位」**モードでは、チャネルは次の規則 (最も高い優先順位から最も低い優先順位の順) に従って通信する機会を得ます。

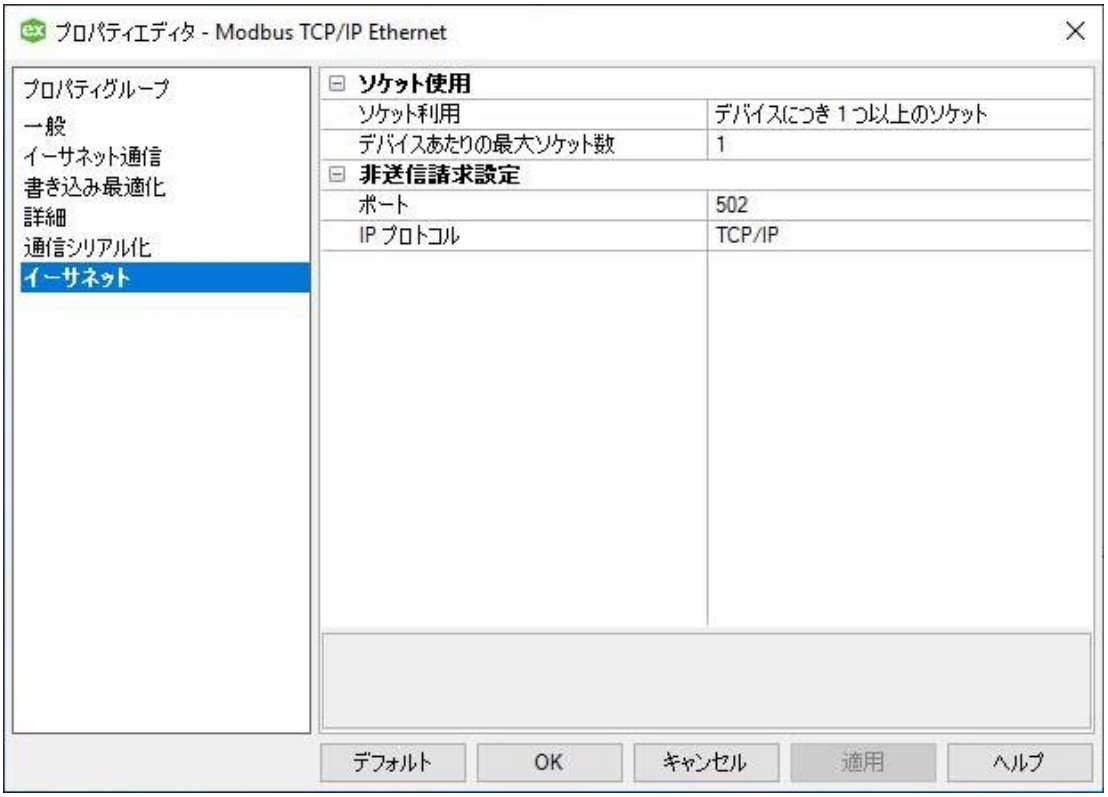
1. 書き込みが保留中になっているチャネルの優先順位が最も高くなります。
2. (内部のプラグインまたは外部のクライアントインタフェースによって) 明示的な読み取りが保留中になっているチャネルは、その読み取りの優先順位に基づいて優先順位が決まります。
3. スキャン読み取りおよびその他の定期的イベント (ドライバー固有)。

デフォルトは「負荷分散」であり、すべての仮想ネットワークとチャネルに影響します。
非送信請求応答に依存するデバイスを仮想ネットワーク内に配置してはなりません。通信をシリアル化する必要がある場合、「自動格下げ」を有効にすることをお勧めします。

データを読み書きする方法はドライバーによって異なるので (単一ブロック/非ブロックトランザクションなど)、アプリケーションの「サイクルあたりのトランザクション数」プロパティを調整する必要があります。その場合、次の要因について検討します。

- 各チャネルから読み取る必要があるタグの数
- 各チャネルにデータを書き込む頻度
- チャネルが使用しているのはシリアルドライバーかイーサネットドライバーか?
- ドライバーは複数の要求に分けてタグを読み取るか、複数のタグをまとめて読み取るか?
- デバイスのタイミングプロパティ(「要求のタイムアウト」や「連続した *x* 回のタイムアウト後の失敗」など) が仮想ネットワークの通信メディアに最適化されているか?

チャネルのプロパティ - イーサネット



「ソケット使用」

「ソケット利用」:ドライバーがこのチャネル上のすべてのデバイスと単一のソケットを共有するか、複数のソケットを使用してデバイスと通信するかを指定します。デバイスが使用可能な接続数に上限がある場合、ドライバーが接続を維持することが望ましくないことがあります。通常、ターゲットデバイスで接続に使用可能なポートの数は制限されています。ドライバーがポートを使用している場合、その他のシステムはターゲットデバイスにアクセスできません。これらの場合にこのパラメータは便利です。ドライバーを使用して Modbus-イーサネット/Modbus-RTU 間ブリッジ製品と通信している場合、ドライバーをシングルソケットモードにできることが重要です。これらの製品の多くは、複数の RS-485 シリアルベースデバイスを単一の Modbus-イーサネット/Modbus-RTU 間ブリッジに接続できます。

- **「デバイスにつき 1 つ以上のソケット」:** ドライバーがネットワーク上のデバイスごとに 1 つ以上のソケットを使用し、そのソケットをアクティブな接続として維持することを指定します。これがデフォルトの設定および動作です。特定のデバイスのデータの読み取りや書き込みが行われるたびにドライバーによって接続が再確立されることがないため、接続オーバーヘッドが低減します。そのため、**「チャネルにつき 1 ソケット」**を使用する場合に比べ、パフォーマンスが向上する可能性があります。ゲートウェイデバイスで多数のシリアルデバイスを処理する場合は、この設定を使用することをお勧めします。
  注記: ゲートウェイ (およびデバイス) は通常、通信の競合を回避するために同時接続の数を制限します。この制限を超えないようにしてください。この制限を超えた場合、接続に失敗したというメッセージがドライバーによって書き込まれます。
- **「チャネルにつき 1 ソケット (共有)」**:ドライバーは、同じ共有ソケットを使用してすべてのデバイスと通信します。単一のソケットを共有するこの代替構成では、チャネル内の各デバイスでソケットが再利用されるため、各接続を

開いたり閉じたりする必要があります。このオプションを選択した場合は、最適なパフォーマンスを実現するための追加の調整が必要になることがあります。

「デバイスあたりの最大ソケット数」: デバイスに使用可能なソケットの最大数を指定します。デフォルトは 1 です。

注記: 複数のソケットを設定すると、ドライバーの読み取りおよび書き込み操作のパフォーマンスが大幅に向上する可能性があります。これは次の動作が理由です。

- 複数のソケットを設定すると、ドライバーはターゲットデバイスに対して読み取りまたは書き込むデータを、ターゲットデバイスで使用されているすべての使用可能なソケットに分散します。その後、読み取りまたは書き込み操作が、すべてのソケットからデバイスに同時に発行されます。
- デバイス応答メッセージは、ドライバーで同時に受信できます。デバイスの応答はチャネルレベルで 1 つのスレッドごとに順次処理されます。ただし、チャネルレベルのデータ処理は非常に高速 (数十ミリ秒以内) で実行されるため、**「デバイスあたりの最大ソケット数」**で複数のソケットを使用するように設定すると、パフォーマンスを大幅に向上できる可能性があります。

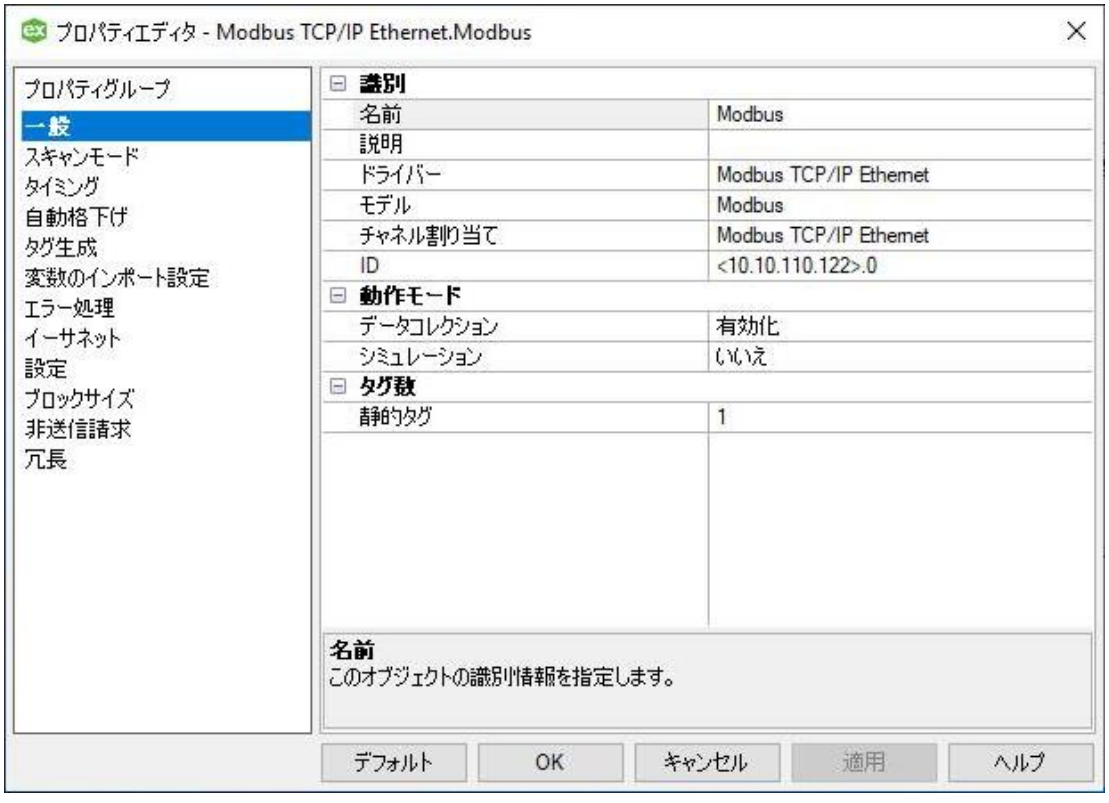
非送信請求設定

Modbus TCP/IP イーサネットドライバー がクライアントモードになっている場合、非送信請求の要求が許可されます。OPC サーバーによってドライバーがロードされると、ドライバーは非送信請求データの受信待機スレッドを起動します。このスレッドは OPC サーバーで設定されているすべてのチャネルに対してグローバルです。たとえば、OPC サーバープロジェクトに 3 つのチャネルが定義されていて、1 つのチャネルで何らかの設定が変更された場合、ほかの 2 つのチャネルに対して同じ変更が行われます。変更が適用されると、受信待機スレッドが再起動されます。イベントログは再起動のイベントを記録します。

「ポート」: ドライバーが非送信請求要求を受信待機しているときに使用するポート番号を指定します。有効な範囲は 0 から 65535 です。デフォルトは 502 です。

「IP プロトコル」: ドライバーが非送信請求要求を受信待機しているときに使用するプロトコルを指定します。オプションには「ユーザーデータグラムプロトコル (UDP)」と「伝送制御プロトコル (TCP/IP)」があります。デフォルトは TCP/IP です。

デバイスのプロパティ - 一般



識別

「名前」: このデバイスのユーザー定義の識別情報。

「説明」: このデバイスに関するユーザー定義の情報。

「チャネル割り当て」: このデバイスが現在属しているチャネルのユーザー定義の名前。

「ドライバー」: このデバイスに設定されているプロトコルドライバー。
● *特定のデバイスモデルの詳細については、**サポートされるデバイスモデル**を参照してください。*

「モデル」: このデバイスのバージョン。

「ID」: デバイスの IP アドレスおよびイーサネットネットワーク上の Modbus ブリッジインデックスを指定します。デバイス ID は <ホスト>.XXX として指定し、ここで <ホスト> は標準 UNC/DNS 名または IP アドレスです。*XXX* はデバイスの Modbus ブリッジインデックスを示し、0 から 255 の範囲になります。ブリッジが使用されていない場合、このインデックスを 0 に設定する必要があります。モデルとデバイス ID によっては、非送信請求デバイスまたは Modbus クライアントデバイスとして機能するようにデバイスを設定することができます。
● *非送信請求モードの詳細については、**「Modbus 非送信請求またはサーバーモード」**を参照してください。*

例

1. IP アドレスが 205.167.7.19 である Modicon TSX Quantum デバイスからデータを要求する場合、デバイス ID を 205.167.7.19.0 として入力する必要があります。
2. ブリッジインデックスが 5 である Modbus イーサネットブリッジに接続している、IP アドレスが 205.167.7.50 である Modbus Plus デバイスからデータを要求する場合、デバイス ID を 205.167.7.50.5 として入力する必要があります。

動作モード

「データコレクション」: このプロパティでは、デバイスのアクティブな状態を制御します。デバイスの通信はデフォルトで有効になっていますが、このプロパティを使用して物理デバイスを無効にできます。デバイスが無効になっている場合、通信は試みられません。クライアントから見た場合、そのデータは無効としてマークされ、書き込み操作は許可されません。このプロパティは、このプロパティまたはデバイスのシステムタグを使用していつでも変更できます。

「シミュレーション」: このオプションは、デバイスをシミュレーションモードにします。このモードでは、ドライバーは物理デバイスとの通信を試みませんが、サーバーは引き続き有効な OPC データを返します。シミュレーションモードではデバイスとの物理的な通信は停止しますが、OPC データは有効なデータとして OPC クライアントに返されます。シミュレーションモードでは、サーバーはすべてのデバイスデータを自己反映的データとして扱います。つまり、シミュレーションモードのデバイスに書き込まれたデータはすべて再び読み取られ、各 OPC アイテムは個別に処理されます。アイテムのメモリマップはグループ更新レートに基づきます。(サーバーが再初期化された場合などに) サーバーがアイテムを除去した場合、そのデータは保存されません。デフォルトは「いいえ」です。

● **注記:**

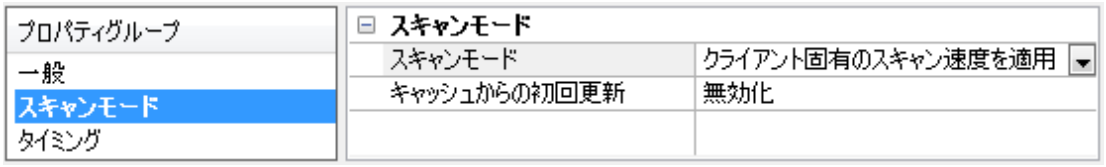
1. システムタグ (_Simulated) は読み取り専用であり、ランタイム保護のため、書き込みは禁止されています。このシステムタグを使用することで、このプロパティをクライアントからモニターできます。
2. シミュレーションモードでは、アイテムのメモリマップはクライアントの更新レート (OPC クライアントではグループ更新レート、ネイティブおよび DDE インタフェースではスキャン速度) に基づきます。つまり、異なる更新レートで同じアイテムを参照する 2 つのクライアントは異なるデータを返します。

●シミュレーションモードはテストとシミュレーションのみを目的としています。本番環境では決して使用しないでください。

●関連項目: **構成 API コマンドを使用したデバイスの設定**、**Modbus を使用した API の例**

デバイスのプロパティ - スキャンモード

「スキャンモード」では、デバイスとの通信を必要とする、サブスクリプション済みクライアントが要求したタグのスキャン速度を指定します。同期および非同期デバイスの読み取りと書き込みは可能なかぎりただちに処理され、「スキャンモード」のプロパティの影響を受けません。



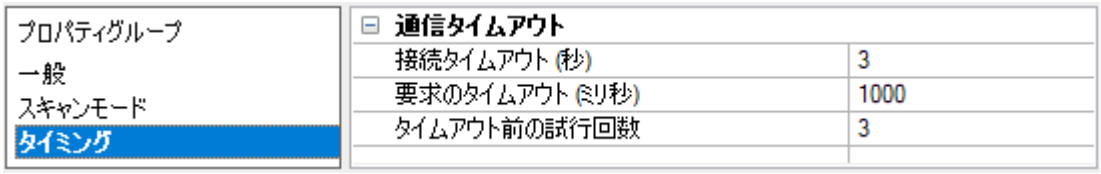
「スキャンモード」: 購読しているクライアントに送信される更新についてデバイス内のタグをどのようにスキャンするかを指定します。オプションの説明は次のとおりです。

- **「クライアント固有のスキャン速度を適用」**: このモードでは、クライアントによって要求されたスキャン速度を使用します。
- **「指定したスキャン速度以下でデータを要求」**: このモードでは、最大スキャン速度として設定されている値を指定します。有効な範囲は 10 から 99999990 ミリ秒です。デフォルトは 1000 ミリ秒です。
  注記: サーバーにアクティブなクライアントがあり、デバイスのアイテム数とスキャン速度の値が増加している場合、変更はただちに有効になります。スキャン速度の値が減少している場合、すべてのクライアントアプリケーションが切断されるまで変更は有効になりません。
- **「すべてのデータを指定したスキャン速度で要求」**: このモードでは、指定した速度で購読済みクライアント用にタグがスキャンされます。有効な範囲は 10 から 99999990 ミリ秒です。デフォルトは 1000 ミリ秒です。
- **「スキャンしない、要求ポールのみ」**: このモードでは、デバイスに属するタグは定期的にポーリングされず、アクティブになった後はアイテムの初期値の読み取りは実行されません。更新のポーリングは、_DemandPoll タグに書き込むか、個々のアイテムについて明示的なデバイス読み取りを実行することによって、OPC クライアントが行います。詳細については、サーバーのヘルプで「デバイス要求ポール」を参照してください。
- **「タグに指定のスキャン速度を適用」**: このモードでは、静的構成のタグプロパティで指定されている速度で静的タグがスキャンされます。動的タグはクライアントが指定したスキャン速度でスキャンされます。

「キャッシュからの初回更新」: このオプションを有効にした場合、サーバーは保存 (キャッシュ) されているデータから、新たにアクティブ化されたタグ参照の初回更新を行います。キャッシュからの更新は、新しいアイテム参照が同じアドレス、スキャン速度、データ型、クライアントアクセス、スケール設定のプロパティを共有している場合にのみ実行できます。1 つ目のクライアント参照についてのみ、初回更新にデバイス読み取りが使用されます。デフォルトでは無効になっており、クライアントがタグ参照をアクティブ化したときにはいつでも、サーバーがデバイスから初期値の読み取りを試みます。

デバイスのプロパティ - タイミング

デバイスのタイミングのプロパティでは、エラー状態に対するデバイスの応答をアプリケーションのニーズに合わせて調整できます。多くの場合、最適なパフォーマンスを得るためにはこれらのプロパティを変更する必要があります。電気的に発生するノイズ、モデムの遅延、物理的な接続不良などの要因が、通信ドライバーで発生するエラーやタイムアウトの数に影響します。タイミングのプロパティは、設定されているデバイスごとに異なります。



通信タイムアウト

「接続タイムアウト」: このプロパティ (イーサネットベースのドライバーで主に使用) は、リモートデバイスとのソケット接続を確立するために必要な時間を制御します。デバイスの接続時間は、同じデバイスへの通常の通信要求よりも長くかかることがよくあります。有効な範囲は 1 から 30 秒です。デフォルトは通常は 3 秒ですが、各ドライバーの特性によって異なる場合があります。この設定がドライバーでサポートされていない場合、無効になります。
注記: UDP 接続の特性により、UDP を介して通信する場合には接続タイムアウトの設定は適用されません。

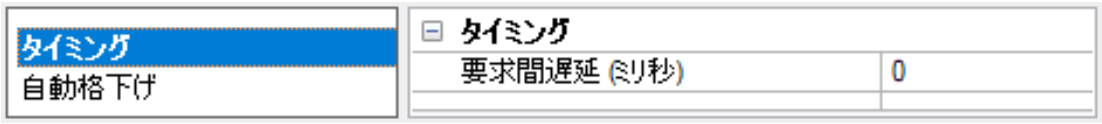
「要求のタイムアウト」: すべてのドライバーがターゲットデバイスからの応答の完了を待機する時間を決定するために使用する間隔を指定します。有効な範囲は 50 から 9999 ミリ秒 (167 分) です。デフォルトは通常は 1000 ミリ秒ですが、ドライバーによって異なる場合があります。ほとんどのシリアルドライバーのデフォルトのタイムアウトは 9600 ボー以上のボーレートに基づきます。低いボーレートでドライバーを使用している場合、データの取得に必要な時間が増えることを補うため、タイムアウト時間を増やします。

「タイムアウト前の試行回数」: ドライバーが通信要求を発行する回数を指定します。この回数を超えると、要求が失敗してデバイスがエラー状態にあると見なされます。有効な範囲は 1 から 10 です。デフォルトは通常は 3 ですが、各ドライ

バーの特性によって異なる場合があります。アプリケーションに設定される試行回数は、通信環境に大きく依存します。このプロパティは、接続の試行と要求の試行の両方に適用されます。

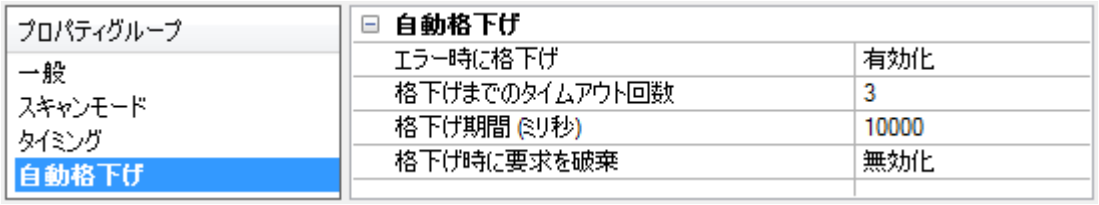
タイミング

「要求間遅延」: ドライバーがターゲットデバイスに次の要求を送信するまでの待ち時間を指定します。デバイスに関連付けられているタグおよび 1 回の読み取りと書き込みの標準のポーリング間隔がこれによってオーバーライドされます。この遅延は、応答時間が長いデバイスを扱う際や、ネットワークの負荷が問題である場合に役立ちます。デバイスの遅延を設定すると、そのチャネル上のその他すべてのデバイスとの通信に影響が生じます。可能な場合、要求間遅延を必要とするデバイスは別々のチャネルに分けて配置することをお勧めします。その他の通信プロパティ (通信シリアル化など) によってこの遅延が延長されることがあります。有効な範囲は 0 から 300,000 ミリ秒ですが、一部のドライバーでは独自の設計の目的を果たすために最大値が制限されている場合があります。デフォルトは 0 であり、ターゲットデバイスへの要求間に遅延はありません。
注記: すべてのドライバーで「要求間遅延」がサポートされているわけではありません。使用できない場合にはこの設定は表示されません。



デバイスのプロパティ - 自動格下げ

自動格下げのプロパティを使用することで、デバイスが応答していない場合にそのデバイスを一時的にスキャン停止にできます。応答していないデバイスを一定期間オフラインにすることで、ドライバーは同じチャネル上のほかのデバイスとの通信を引き続き最適化できます。停止期間が経過すると、ドライバーは応答していないデバイスとの通信を再試行します。デバイスが応答した場合はスキャンが開始され、応答しない場合はスキャン停止期間が再開します。



「エラー時に格下げ」: 有効にした場合、デバイスは再び応答するまで自動的にスキャン停止になります。
ヒント: システムタグ _AutoDemoted を使用して格下げ状態をモニターすることで、デバイスがいつスキャン停止になったかを把握できます。

「格下げまでのタイムアウト回数」: デバイスをスキャン停止にするまでに要求のタイムアウトと再試行のサイクルを何回繰り返すかを指定します。有効な範囲は 1 から 30 回の連続エラーです。デフォルトは 3 です。

「格下げ期間」: タイムアウト値に達したときにデバイスをスキャン停止にする期間を指定します。この期間中、そのデバイスには読み取り要求が送信されず、その読み取り要求に関連するすべてのデータの品質は不良に設定されます。この期間が経過すると、ドライバーはそのデバイスのスキャンを開始し、通信での再試行が可能になります。有効な範囲は 100 から 3600000 ミリ秒です。デフォルトは 10000 ミリ秒です。

「格下げ時に要求を破棄」: スキャン停止期間中に書き込み要求を試行するかどうかを選択します。格下げ期間中も書き込み要求を必ず送信するには、無効にします。書き込みを破棄するには有効にします。サーバーはクライアントから受信した書き込み要求をすべて自動的に破棄し、イベントログにメッセージを書き込みません。

デバイスのプロパティ - タグ生成

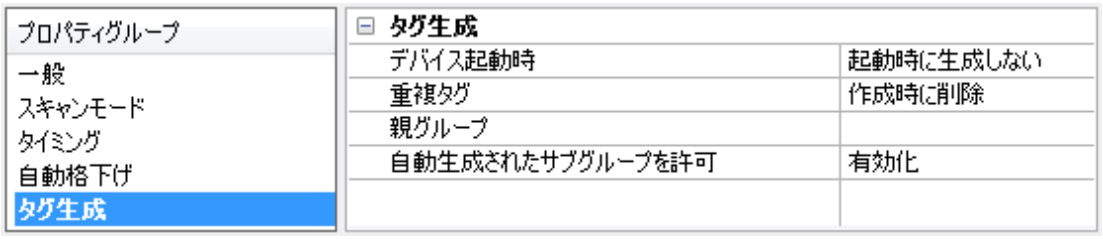
自動タグデータベース生成機能によって、アプリケーションの設定がプラグアンドプレイ操作になります。デバイス固有のデータに対応するタグのリストを自動的に構築するよう通信ドライバーを設定できます。これらの自動生成されたタグ (サポートしているドライバーの特性によって異なる) をクライアントからブラウズできます。

一部のデバイスやドライバーは自動タグデータベース生成のフル機能をサポートしていません。また、すべてのデバイスやドライバーが同じデータ型をサポートするわけではありません。詳細については、データ型の説明を参照するか、各ドライバーがサポートするデータ型のリストを参照してください。

ターゲットデバイスが独自のローカルタグデータベースをサポートしている場合、ドライバーはそのデバイスのタグ情報を読み取って、そのデータを使用してサーバー内にタグを生成します。デバイスが名前付きのタグをネイティブにサポートしていない場合、ドライバーはそのドライバー固有の情報に基づいてタグのリストを作成します。この 2 つの条件の例は次のとおりです。

1. データ取得システムが独自のローカルタグデータベースをサポートしている場合、通信ドライバーはデバイスで見つかったタグ名を使用してサーバーのタグを構築します。
2. イーサネット I/O システムが独自の使用可能な I/O モジュールタイプの検出をサポートしている場合、通信ドライバーはイーサネット I/O ラックにプラグイン接続している I/O モジュールのタイプに基づいてサーバー内にタグを自動的に生成します。

注記: 自動タグデータベース生成の動作モードを詳細に設定できます。詳細については、以下のプロパティの説明を参照してください。



「プロパティ変更時」: デバイスが、特定のプロパティが変更された際の自動タグ生成をサポートする場合、**「プロパティ変更時」**オプションが表示されます。これはデフォルトで**「はい」**に設定されていますが、**「いいえ」**に設定してタグ生成を実行する時期を制御できます。この場合、タグ生成を実行するには**「タグを作成」**操作を手動で呼び出す必要があります。

「デバイス起動時」: OPC タグを自動的に生成するタイミングを指定します。オプションの説明は次のとおりです。

- **「起動時に生成しない」**: このオプションを選択した場合、ドライバーは OPC タグをサーバーのタグ空間に追加しません。これはデフォルトの設定です。
- **「起動時に常に生成」**: このオプションを選択した場合、ドライバーはデバイスのタグ情報を評価します。さらに、サーバーが起動するたびに、サーバーのタグ空間にタグを追加します。
- **「最初の起動時に生成」**: このオプションを選択した場合、そのプロジェクトが初めて実行されたときに、ドライバーがデバイスのタグ情報を評価します。さらに、必要に応じて OPC タグをサーバーのタグ空間に追加します。

注記: OPC タグを自動生成するオプションを選択した場合、サーバーのタグスペースに追加されたタグをプロジェクトとともに保存する必要があります。ユーザーは**「ツール」**|**「オプション」**メニューから、自動保存するようプロジェクトを設定できます。

「重複タグ」: 自動タグデータベース生成が有効になっている場合、サーバーが以前に追加したタグや、通信ドライバーが最初に作成した後で追加または修正されたタグを、サーバーがどのように処理するかを設定する必要があります。この設定では、自動生成されてプロジェクト内に現在存在する OPC タグをサーバーがどのように処理するかを制御します。これによって、自動生成されたタグがサーバーに累積することもなくなります。

たとえば、**「起動時に常に生成」**に設定されているサーバーのラックで I/O モジュールを変更した場合、通信ドライバーが新しい I/O モジュールを検出するたびに新しいタグがサーバーに追加されます。古いタグが削除されなかった場合、多数の未使用タグがサーバーのタグ空間内に累積することがあります。以下のオプションがあります。

- **「作成時に削除」**: このオプションを選択した場合、新しいタグが追加される前に、以前にタグ空間に追加されたタグがすべて削除されます。これはデフォルトの設定です。
- **「必要に応じて上書き」**: このオプションを選択した場合、サーバーは通信ドライバーが新しいタグに置き換えているタグだけ除去します。上書きされていないタグはすべてサーバーのタグ空間に残ります。
- **「上書きしない」**: このオプションを選択した場合、サーバーは以前に生成されたタグやサーバーにすでに存在するタグを除去しません。通信ドライバーは完全に新しいタグだけを追加できます。
- **「上書きしない、エラーを記録」**: このオプションには上記のオプションと同じ効果がありますが、タグの上書きが発生した場合にはサーバーのイベントログにエラーメッセージも書き込まれます。

注記: OPC タグの除去は、通信ドライバーによって自動生成されたタグ、および生成されたタグと同じ名前を使用して追加されたタグに影響します。ドライバーによって自動生成されるタグと一致する可能性がある名前を使用してサーバーにタグを追加しないでください。

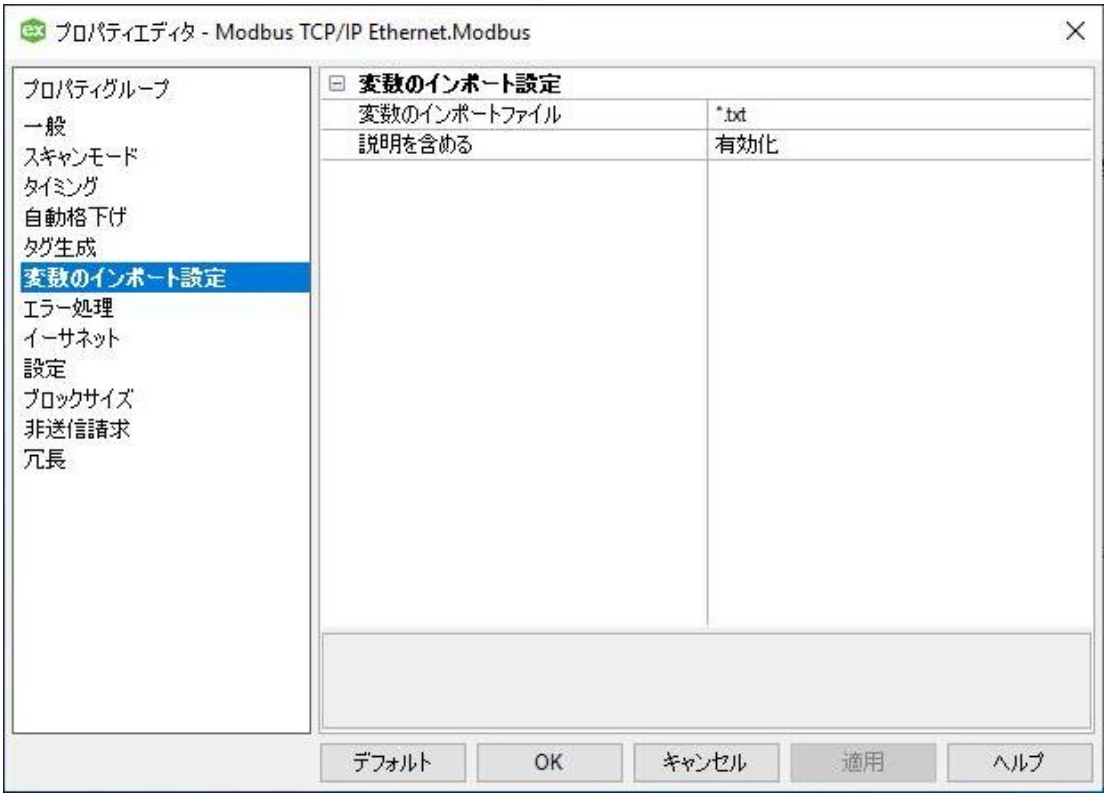
「親グループ」: このプロパティでは、自動生成されたタグに使用するグループを指定することで、自動生成されたタグと、手動で入力したタグを区別します。グループの名前は最大 256 文字です。この親グループは、自動生成されたすべてのタグが追加されるルートブランチとなります。

「自動生成されたサブグループを許可」: このプロパティでは、自動生成されたタグ用のサブグループをサーバーが自動的に作成するかどうかを制御します。これはデフォルトの設定です。無効になっている場合、サーバーはグループを作成しないで、デバイスのタグをフラットリスト内に生成します。サーバープロジェクトで、生成されたタグには名前としてアドレスの値が付きます。たとえば、生成プロセス中はタグ名は維持されません。
注記: サーバーがタグを生成しているときに、タグに既存のタグと同じ名前が割り当てられた場合、タグ名が重複しないようにするため、番号が自動的に 1 つ増分します。たとえば、生成プロセスによってすでに存在する "AI22" という名前のタグが作成された場合、代わりに "AI23" としてタグが作成されます。

「作成」: 自動生成 OPC タグの作成を開始します。**「タグを作成」**が有効な場合、デバイスの構成が修正されると、ドライバーはタグ変更の可能性についてデバイスを再評価します。システムタグからアクセスできるため、クライアントアプリケーションはタグデータベース作成を開始できます。
注記: 構成がプロジェクトをオフラインで編集する場合、**「タグを作成」**は無効になります。

デバイスのプロパティ - 変数のインポート設定

Modbus ドライバー向け CSV ファイルの詳細については、Modbus ドライバー向け CSV ファイルの作成を参照してください。

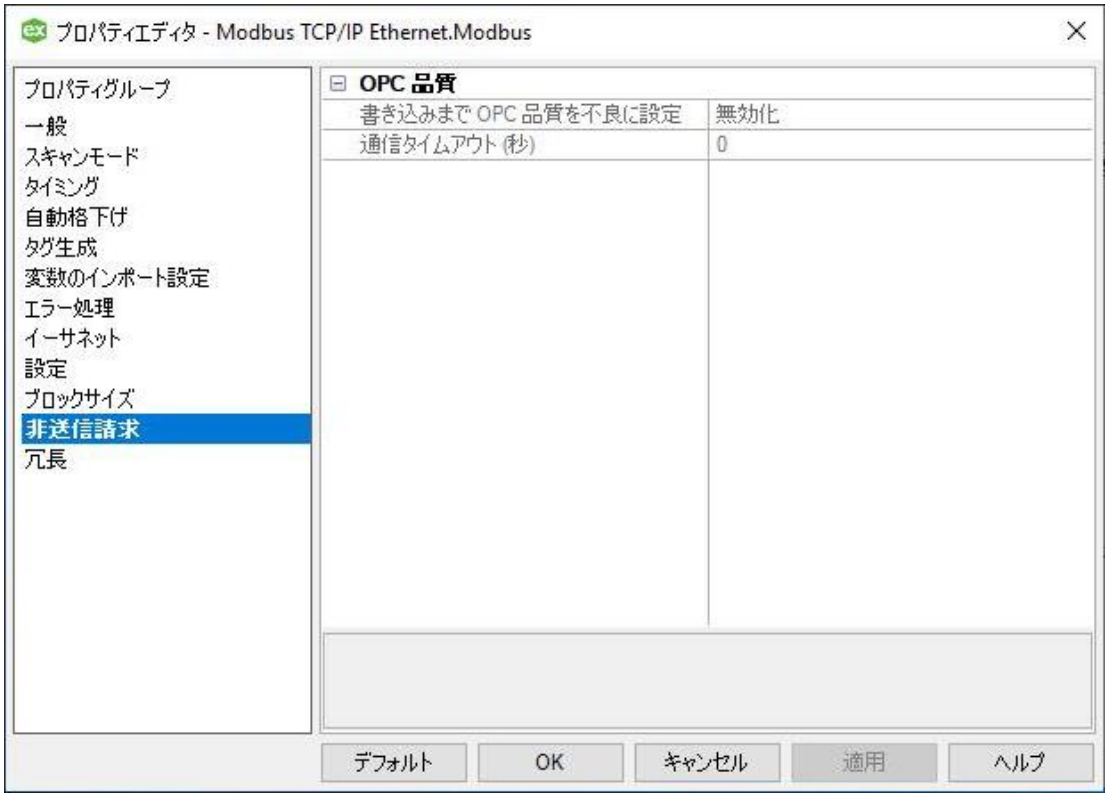


「変数のインポートファイル」: このパラメータでは、自動タグデータベース生成機能が有効になっている場合にこのドライバーが使用する変数インポートファイルの正確な場所を指定します。

「説明を含める」: 有効な場合、タグの説明がインポートされます (ファイル内に存在する場合)。

自動タグデータベース生成機能の設定 (および変数インポートファイルの作成方法) については、自動タグデータベース生成を参照してください。

デバイスのプロパティ - 非送信請求



OPC 品質

「書き込みまで OPC 品質を不良に設定」: このドライバーに関連付けられるタグの初期 OPC 品質を制御します。無効にした場合、すべてのタグの初期値は 0 となり、OPC 品質は良好に設定されます。これがデフォルトの状態です。有効にした場合、すべてのタグの初期値は 0 となり、OPC 品質は不良に設定されます。タグが参照するすべてのコイルまたはレジスタが Modbus クライアントまたはクライアントアプリケーションによって書き込まれるまで、タグの品質は不良のままとなります。たとえば、アドレスが 400001 でデータ型が DWord であるタグは 2 つの保持レジスタ 400001 および 400002 を参照します。両方の保持レジスタに書き込まれるまでタグの品質は良好になりません。
注記: デバイスが非送信請求モードでない場合、このオプションは暗色表示されます。

「通信タイムアウト」: ドライバーが受信する要求を待機する時間 (秒) を設定します。この時間が経過すると、ドライバーはそのデバイスのタグの品質を不良に設定します。タイムアウトの発生後、タイムアウトをリセットしてすべてのタグを通常どおりに処理するには、リモートクライアントとの通信を再確立するか、タムアウト値を 0 に設定して通信タイムアウトを無効にする必要があります。これ以外の方法はありません。有効にする場合、有効な範囲は 1 から 64,800 秒 (18 時間) です。

注記:

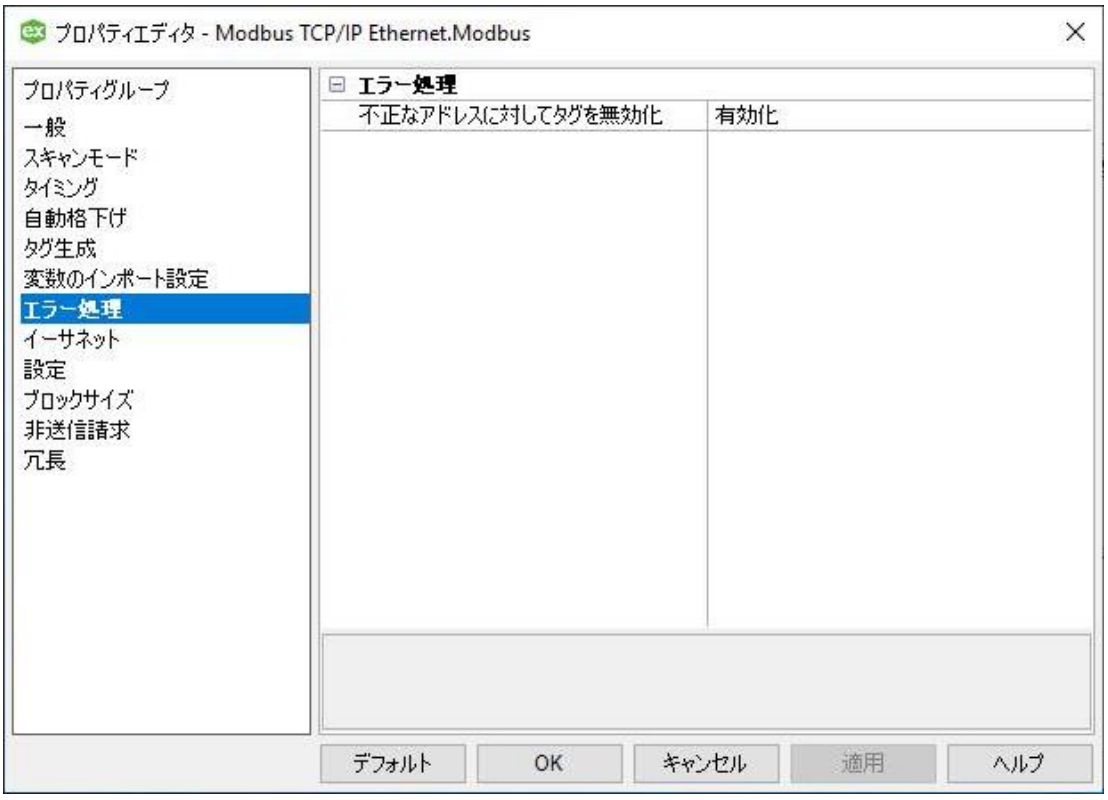
1. 存在しないサーバーデバイス (ステーション ID) に対する要求を受信した場合、その要求はステーション 0 に転送されます。その場合、タイムアウト期間中にリモート通信を明示的に受信しなくても、ステーション ID が 0 のサーバーデバイスでタイムアウトが発生することはありません。
2. 非送信請求デバイスのモデルは Modbus、デバイス ID は *IP_Address.yyy* (ここで *IP_Address* はこのドライバーを実行している PC のローカル IP アドレス) である必要があります。たとえば、127.xxx.xxx.xxx (*xxx* = 0-255)、*yyy* (ステーション ID) = 0-255 です。
3. サーバーデバイスに対する最初の非送信請求を受信すると、イベントログに「<日付>__<time>__<レベル>__<source>__<イベント>」という情報メッセージが表示されます。たとえば、「2/4/2011__4:53:10 PM__情報__Modbus TCP/IP イーサネット__サーバーデバイス <サーバー番号> 用のメモリが作成されました」などのメッセージが表示されます。
4. このドライバーでは、「サーバー」と「非送信請求」という用語は同じ意味になります。

Modbus クライアントと Modbus サーバーに関する考慮事項

以下に、Modbus クライアントデバイスと Modbus サーバーデバイスの両方に関する注意事項を示します。

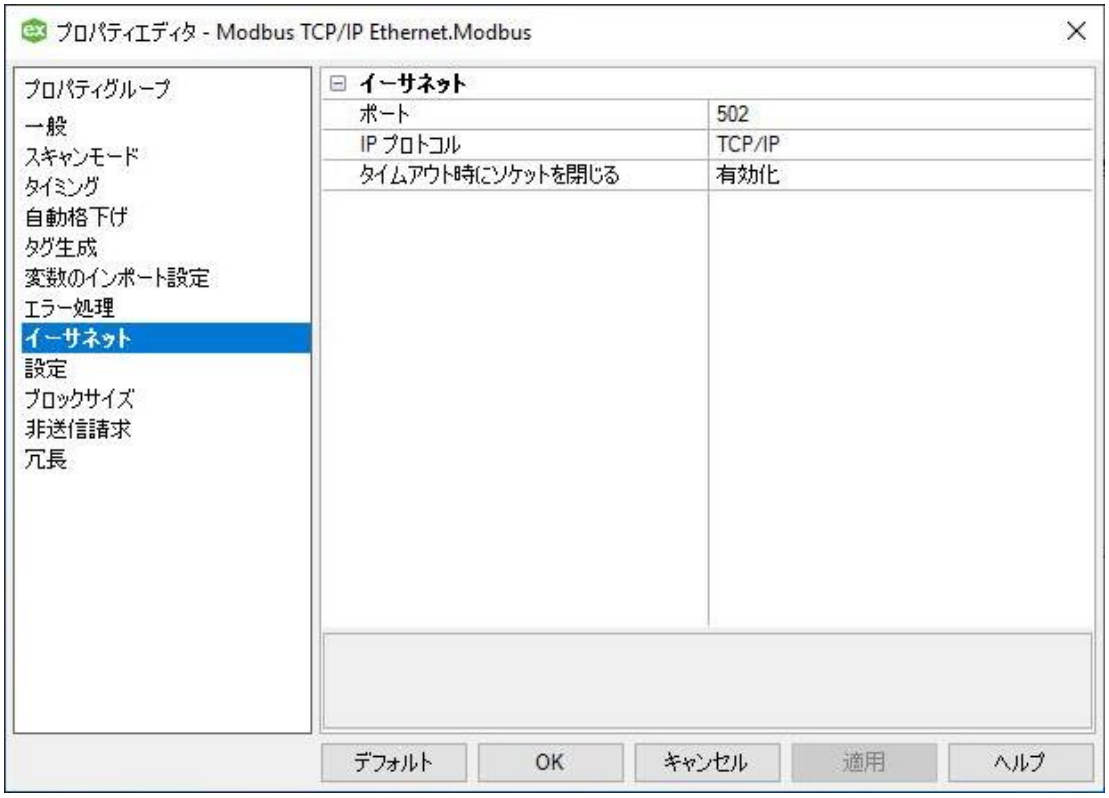
- メールボックスデバイスと Modbus デバイスを同じマシンに配置することはお勧めしません。クライアントは一度に一方のデバイスからのみデータを取得するため、どちらのデバイスからデータを取得するのかを予測することはできません。
- サーバーデバイスのタグ処理を最適な方法で実行するために、クライアントデバイスとサーバーデバイスをサーバープロジェクト内の別々のチャネルに配置することをお勧めします。
- OPC クライアントが接続されている場合、デバイスのモードを変更しない (クライアントからサーバーに変更したり、サーバーからクライアントに変更したりしない) かぎり、デバイス ID を変更することができます。ループバックまたはローカル IP アドレスを別の IP アドレスに変更したかその逆の変更によってモードが変更されました。ドライバーが稼働している PC のループバックアドレスとローカル IP アドレスはサーバーモード (非送信請求モード) を示し、その他の IP アドレスはデバイスのクライアントモードを示しています。OPC クライアントが接続されていない場合は、デバイスのモードを自由に変更することができます (クライアントからクライアント、クライアントからサーバー、サーバーからサーバー、サーバーからクライアントに変更など)。
  注記: 127.xxx.xxx.xxx というフォーマット (ここで *xxx* は 0-255 の範囲) のアドレスはループバックアドレスです。
- クライアントデバイスとサーバーデバイスで「データエンコーディング」グループの設定が一致している必要があります。たとえば、Modbus クライアントとして設定されているデバイスが、Modbus サーバーとして設定されているデバイスと通信する場合などに、この設定が一致している必要があります。
- サーバープロジェクト全体では、一意のサーバー ID ごとに 1 台ずつ、最大 255 台のサーバーデバイスを使用することができます。同じサーバー ID を複数のチャネルで使用することはできません。
- サーバーは、localhost IP アドレスであるループバックアドレス (127.x.x.x) をサーバー自身に対する参照として認識し、サーバー ID 固有の共有メモリ空間を作成します。同じレジスタメモリを使用する、複数のチャネルの同じ ID を持つサーバーが、同じサーバーデバイスとして認識されます。
- 同じサーバー ID をプロジェクト内で複数回使用する必要がある場合は、同じサーバーデバイス ID を持つほかのインスタンスと重複しないタグアドレス範囲を選択してください。タグアドレスの範囲が重複していると、同じサーバー ID エクスペリエンスで同じタグアドレス範囲を使用している複数のチャネルやデバイスが相互に通信するため、データが破損する可能性があります。
- このドライバーでは、「サーバー」と「非送信請求」という用語は同じ意味になります。

デバイスのプロパティ - エラー処理



「不正なアドレスでタグを無効化」: デバイスがデータブロックの読み取りに応答して Modbus 例外コード 2 (不正なアドレス) または 3 (ポイント数などの不正なデータ) を返した場合にドライバーがそのブロックのポーリングを停止するには「有効化」を選択します。エラーの場合もドライバーがそのデータブロックを引き続きポーリングするには「無効化」を選択します。デフォルトで有効になっています。

デバイスのプロパティ - イーサネット

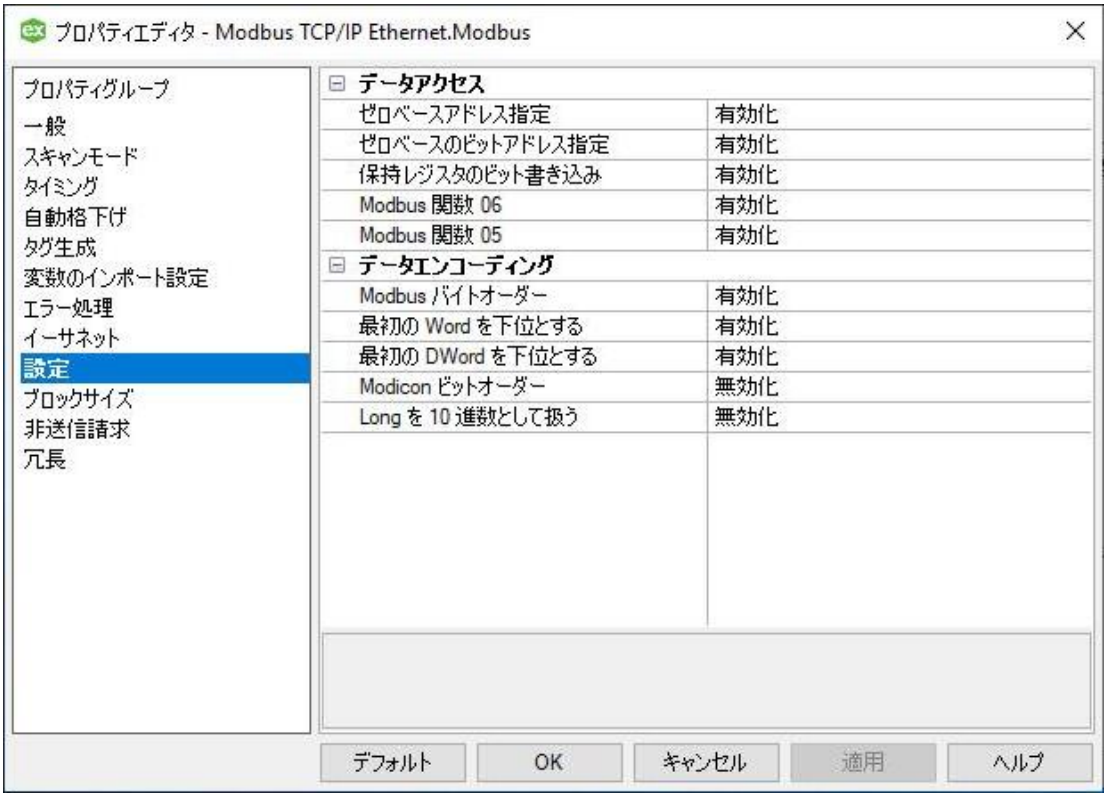


「ポート」: リモートデバイスで使用するよう設定されているポート番号を指定します。有効な範囲は 0 から 65535 です。デフォルトは 502 です。このポート番号はデバイスに対して送信請求要求を行う際に使用されます。
ポートのシステムタグが使用されている場合、ポート番号の設定が変更されます。詳細については、ドライバーのシステムタグのアドレスを参照してください。

「IP プロトコル」: ドライバーがリモートデバイスに接続する際にユーザーデータグラムプロトコル (UDP) を使用するか伝送制御プロトコル (TCP/IP) を使用するかを指定します。クライアントとサーバーの設定が一致している必要があります。たとえば、サーバーの IP プロトコル設定が TCP/IP になっている場合、そのサーバーに対するクライアントの IP プロトコル設定も TCP/IP になっている必要があります。
注記: このドライバーでは Winsock V1.1 以上が必要です。

「タイムアウト時にソケットを閉じる」: デバイスがタイムアウトの期限内に応答しなかった場合にドライバーが TCP ソケット接続を閉じるかどうかを指定します。有効にした場合 (デフォルト)、ドライバーはタイムアウトになるとソケット接続を閉じます。無効にした場合、エラーを受信するか、物理デバイスがソケットを閉じるか、ドライバーがシャットダウンするまで、ドライバーは同じ TCP ソケットを使用し続けます。
注記: Modbus TCP/IP イーサネットドライバー はソケットエラー時にソケット接続を閉じます。

デバイスのプロパティ - 設定



データアクセス

「ゼロベースアドレス指定」: デバイスのアドレス番号付けの規則で番号がゼロではなく 1 で開始する場合、デバイスのパラメータを定義する際にこの値を指定できます。デフォルトでは、Modbus デバイスと通信するためにフレームを構築する場合はユーザーが入力したアドレスから 1 が引かれます。デバイスがこの規則に従わない場合、「無効」を選択します。デフォルトの動作は Modicon PLC の規則に従います。

「ゼロベースのビットアドレス指定」: レジスタ内で、Word 内のビットをブールとして参照可能なメモリタイプ。アドレス指定の表記は *<address><bit>* であり、ここで *<bit>* は Word 内のビット番号を表します。このオプションによって、ある Word 内の 1 ビットを 2 つの方法 (ゼロベースまたは 1 ベース) によってアドレス指定できます。ゼロベースとは最初のビットが 0 で始まることを意味し (範囲 = 0-15)、1 ベースとは最初のビットが 1 で始まることを意味します (範囲 = 1-16)。

「保持レジスタのビット書き込み」: 保持レジスタ内のビット位置に書き込む際、ドライバーは対象のビットのみを修正する必要があります。一部のデバイスはレジスタ内の 1 ビットを操作する特別なコマンドをサポートしています (ファンクションコード 0x16 (16 進) または 22 (10 進))。デバイスがこの機能をサポートしていない場合、ドライバーは 1 ビットだけが変更されるように読み取り/修正/書き込み操作を実行する必要があります。有効になっている場合、この単一レジスタ書き込みの設定に関係なく、レジスタへの書き込みにファンクションコード 0x16 を使用します。無効になっている場合、ドライバーは単一レジスタ書き込みの「Modbus 関数 06」の選択に応じて、ファンクションコード 0x06 または 0x10 を使用します。デフォルトでは無効になっています。
注記: Modbus バイトオーダーが無効になっている場合、このコマンドで送信されるマスクのバイトオーダーは Intel バイトオーダーになります。

「Modbus 関数 06」: このドライバーは、保持レジスタのデータをターゲットデバイスに書き込む Modbus プロトコルファンクションをサポートしています。ほとんどの場合、このドライバーは書き込み対象のレジスタの数に基づいてファンクション 06 と 16 を切り替えます。単一の 16 ビットレジスタに書き込む場合、このドライバーは通常、Modbus ファンクション 06 を使用します。32 ビット値を 2 つのレジスタに書き込む場合、このドライバーは Modbus ファンクション 16 を使用します。標準の Modicon PLC では、このどちらのファンクションを使用しても問題ありません。ただし、Modbus プロトコルを使用する多くのサードパーティデバイスとこれらのデバイスの多くは、保持レジスタへの書き込みに Modbus ファンクション 16 のみをサポートしています。この選択はデフォルトで有効になっており、ドライバーは必要に応じて 06 と 16 を切り替えることができます。デバイスが Modbus ファンクション 16 のみを使用してすべての書き込みを行う必要がある場合、この選択を無効にします。
注記: Word 内のビットの書き込みでは、「保持レジスタのビットマスク」プロパティがこのオプションよりも優先されます。「保持レジスタのビットマスク」が有効になっている場合、このプロパティにかかわらず、ファンクションコード 0x16 が使用さ

れます。無効になっている場合、Word 内のビットの書き込みにはファンクションコード 0x06 または 0x10 が使用されます。

「Modbus 関数 05」: このドライバーは、出力コイルのデータをターゲットデバイスに書き込む Modbus プロトコルファンクションをサポートしています。ほとんどの場合、このドライバーは書き込み対象のコイルの数に基づいてこの 2 つのファンクションを切り替えます。単一のコイルに書き込む場合、このドライバーは Modbus ファンクション 05 を使用します。コイルの配列に書き込む場合、このドライバーは Modbus ファンクション 15 を使用します。標準の Modicon PLC では、このどちらのファンクションを使用しても問題ありません。ただし、Modbus プロトコルを使用する多くのサードパーティデバイスとこれらのデバイスの多くは、コイルの数に関係なく、出力コイルへの書き込みに Modbus ファンクション 15 の使用のみをサポートしています。この選択はデフォルトで有効になっており、ドライバーは必要に応じて 05 と 15 を切り替えることができます。デバイスが Modbus ファンクション 15 のみを使用してすべての書き込みを行う必要がある場合、この選択を無効にします。

「CEG 拡張」: この Modbus ドライバーは、拡張ブロックサイズをサポートする CEG デバイスか、CEG モデルで設定されている場合には Modbus デバイスと通信できます。このプロパティは CEG モデルでのみ使用できます。デフォルトは「有効」であり、そのデバイスは拡張ブロックサイズをサポートする CEG デバイスであることを示します。「無効」の場合、そのデバイスは拡張ブロックサイズをサポートしません。

注記: アクティブな OPC クライアント接続が存在する場合でもこのプロパティを修正できます。その場合、このオプションを無効にするとブロックサイズの範囲が変わります。ブロックサイズのプロパティのいずれかが最大値を超えている場合、それらは新しい最大値に自動的に調整されます。

「メールボックスクライアントの権限」: Modbus ドライバーは次のオプションでメールボックスクライアントと通信できます。

- **「メモリマップ読み取り専用」**: クライアントアプリケーションはメールボックスメモリマップから読み取りのみが可能です。
- **「メモリマップ読み書き」**: クライアントアプリケーションはメールボックスメモリマップとの間で読み書きが可能です。
- **「デバイス書き込み-メモリマップ読み取り」**: クライアントアプリケーションはデバイスへの書き込みのみが可能であり、読み取りはメモリマップから行います。

データエンコーディング

「Modbus バイトオーダー」: 各レジスタ/16 ビット値のデータエンコーディングを設定します。この選択を使用することで、バイトオーダーをデフォルトの Modbus バイトオーダーから Intel バイトオーダーに変更できます。デフォルトでは有効になっており、Modbus 対応デバイスでは標準の設定です。デバイスが Intel バイトオーダーを使用する場合、このプロパティを無効にすることで Intel フォーマットのデータを読み取ります。

「最初の Word を下位とする」: 32 ビット値と、64 ビット値の DWord のデータエンコーディングを設定します。Modbus デバイスでは 32 ビットデータ型に 2 つの連続するレジスタアドレスが使用されます。ドライバーはこのオプションに基づいて、最初の Word を 32 ビット値の下位 Word または上位 Word として読み取ることができます。デフォルトは「有効」で最初の Word が下位となり、Modicon Modsoft プログラミングソフトウェアの規則に従います。

「最初の DWord を下位とする」: 64 ビット値のデータエンコーディングを設定します。Modbus デバイスでは 64 ビットデータ型に 4 つの連続するレジスタアドレスが使用されます。ドライバーは最初の DWord を 64 ビット値の下位 DWord または上位 DWord として読み取ることができます。デフォルトは「有効」で最初の DWord が下位となり、32 ビットデータ型のデフォルトの規則に従います。

「Modicon ビットオーダー」: 有効な場合、ドライバーはレジスタに対する読み書きの際にビットオーダーを反転して Modicon Modsoft プログラミングソフトウェアの規則に従います。たとえば、このオプションが有効になっている場合、アドレス 40001.0/1 への書き込みはこのデバイスのビット 15/16 に影響します。このオプションはデフォルトで無効になっています。

次の例では、そのドライバーが使用しているレジスタ内のビットアドレス指定がゼロベースか 1 ベースかに応じて、1 から 16 番目のビットは 0-15 ビットまたは 1-16 ビットを表します。
MSB = 最上位ビット
LSB = 最下位ビット

「Modicon ビットオーダー」が有効

MSB															LSB
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

「Modicon ビットオーダー」が無効

MSB															LSB
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

「Long を 10 進数として扱う」: 有効な場合、ドライバーは倍精度符号なしの Long データ型と DWord データ型を 0 から 99999999 の範囲の値としてエンコード/デコードします。このフォーマットでは各 Word が 0 から 9999 の値を表します。この範囲を超える読み取られた値はクランプされませんが、動作は定義されていません。読み取られた値はすべて [読み取られた値] = 上位 Word * 10000 + 下位 Word、という式を使用してデコードされます。99999999 より大きい書き込まれた値は最大値にクランプされます。書き込まれた値はすべて、生データ = [書き込まれた値]/10000 + [書き込まれた値] % 10000 という式を使用してエンコードされます。

設定に関するヒント

データ型	Modbus バイトオーダー	最初の Word を下位とする	最初の DWord を下位とする
Word、Short、BCD	適用可能	なし	なし
Float、DWord、Long、LBCD	適用可能	適用可能	なし
Double	適用可能	適用可能	適用可能

必要な場合、以下の情報とデバイスのドキュメントを参照して、データエンコーディングオプションの正しい設定を調べてください。

ほとんどの Modbus デバイスではデフォルト設定で問題ありません。

データエンコーディングのオプション	データエンコーディング	
Modbus バイトオーダー	上位バイト (15..8)	下位バイト (7..0)
Modbus バイトオーダー	下位バイト (7..0)	上位バイト (15..8)
最初の Word を下位とする	上位 Word (31..16) 64 ビットデータ型での DWord の上位 Word (63..48)	下位 Word (15..0) 64 ビットデータ型での DWord の下位 Word (47..32)
最初の Word を下位とする	下位 Word (15..0) 64 ビットデータ型での DWord の下位 Word (47..32)	上位 Word (31..16) 64 ビットデータ型での DWord の上位 Word (63..48)
最初の DWord を下位とする	上位 DWord (63..32)	下位 DWord (31..0)
最初の DWord を下位とする	下位 DWord (31..0)	上位 DWord (63..32)

デバイスのプロパティ - ブロックサイズ

コイル

「出力コイル」: 出力ブロックのサイズをビット数で指定します。コイルは 8 から 2000 ポイント (ビット) の範囲で一度に読み取ることができます。デフォルトは 32 です。

「入力コイル」: 入力ブロックのサイズをビット数で指定します。コイルは 8 から 2000 ポイント (ビット) の範囲で一度に読み取ることができます。デフォルトは 32 です。

「レジスタ」

「内部レジスタ」: 内部レジスタのブロックサイズをビット数で指定します。1 から 120 の標準 16 ビット Modbus レジスタを一度で読み取ることができます。デフォルトは 32 です。

「保持レジスタ」: 保持レジスタのブロックサイズをビット数で指定します。1 から 120 の標準 16 ビット Modbus レジスタを一度で読み取ることができます。デフォルトは 32 です。

「ブロック」

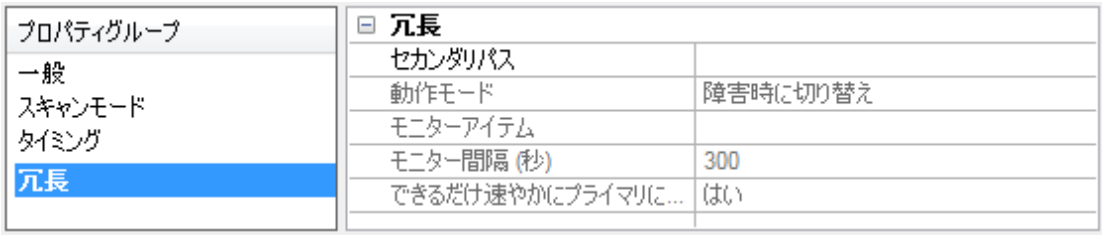
「ブロック読み取り文字列」: 通常は個別に読み取る文字列タグをグループ/ブロックで読み取ります。文字列タグは選択したブロックサイズに応じてグループ化されます。ブロック読み取りは Modbus モデルの文字列タグに対してのみ実行できます。

注記:

1. Instromet、Roxar、および Fluenta モデル (32 ビットおよび 64 ビットレジスタをサポート) では特別な注意が必要です。Modbus プロトコルではブロックサイズが 256 バイト以下に制限されます。これにより、これらのモデルでは最大ブロックサイズが 64 (32 ビットレジスタ) または 32 (64 ビットレジスタ) になります。
2. CEG モデルでサポートされるコイルのブロックサイズは 8 から 8000 の範囲の 8 の倍数であり、レジスタのブロックサイズは 1 から 500 の範囲です。このモデルは必ず CEG デバイスとともに使用する必要があります。

3. レジスタのブロックサイズとして 120 より大きい値が設定され、任意のタグに 32 ビットまたは 64 ビットデータ型が使用されている場合、「ブロックに不良アドレスがあります」というエラーが発生することがあります。これを防止するには、ブロックサイズの値を 120 に減らしてください。
4. 一部のデバイスではデフォルトサイズのブロック読み取り操作がサポートされていないことがあります。小さい Modicon PLC および Modicon 以外のデバイスでは、Modbus イーサネットネットワークでサポートされているデータ転送の最大長がサポートされないことがあります。
5. 一部のデバイスには連続しないアドレスが含まれていることがあります。この場合、未定義のメモリを含むデータブロックをドライバーが読み取ろうとすると、その要求は却下されることがあります。

デバイスのプロパティ - 冗長



冗長設定はメディアレベルの冗長プラグインで使用できます。
詳細については、Web サイトまたはユーザーマニュアルを参照するか、営業担当者までお問い合わせください。

構成 API Modbus Ethernet の例

チャネルとデバイスの定義および列挙のリストについては、REST クライアントを使用して次のエンドポイントにアクセスしてください。

チャネル定義

エンドポイント (GET):

```
https://<ホスト名または IP>:<ポート
>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/channels
```

デバイス定義

エンドポイント (GET):

```
https://<ホスト名または IP>:<ポート
>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/devices
```

Modbus チャネルの作成

エンドポイント (POST):

```
https://<ホスト名または IP>:<ポート>/config/v1/project/channels
```

ボディ:

```
{ "common.ALLTYPES_NAME": "MyChannel", "servermain.MULTIPLE_TYPES_DEVICE_DRIVER":
"Modbus TCP/IP Ethernet" }
```

関連項目: チャネルのプロパティのリストについては、付録 A を参照してください。

Modbus デバイスの作成

エンドポイント (POST):

```
https://<ホスト名または IP>:<ポート>/config/v1/project/channels/MyChannel/devices
```

ボディ:

```
{ "common.ALLTYPES_NAME": "MyDevice", "servermain.DEVICE_ID_STRING":
"<192.160.0.1>.0", "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Modbus TCP/IP
Ethernet" }
```

関連項目: デバイスのプロパティのリストについては、付録 B を参照してください。

デバイス ID の更新

REST クライアントから "PUT" コマンドを使用して、デバイス ID を更新します。
以下のエンドポイントの例は、"ModbusTCPIP" チャネル名と "ModbusDevice" デバイス名を持つ "demo-project.json" プロジェクト構成を参照しています。

デバイス ID の例

エンドポイント (PUT):

```
https://<ホスト名または IP>:<ポート
>/config/v1/project/channels/ModbusTCPIP/devices/ModbusDevice
```

ボディ:

```
{ "project_id": <GET で取得したプロジェクト ID>, "servermain.DEVICE_ID_STRING": "<IP アドレス
>" }
```

Modbus タグの作成

エンドポイント (POST):

```
https://<ホスト名または IP>:<ポート
>/config/v1/project/channels/MyChannel/devices/MyDevice/tags
```

ボディ:

```
[ { "common.ALLTYPES_NAME": "MyTag1", "servermain.TAG_ADDRESS": "40001" } {
"common.ALLTYPES_NAME": "MyTag2", "servermain.TAG_ADDRESS": "40002" } ]
```

- 関連項目: タグのプロパティのリストについては、付録 C を参照してください。

- *構成 API を使用したプロジェクト設定の詳細については、サーバーのヘルプを参照してください。*

列挙

デバイスモデルなどの一部のプロパティには、列挙にマッピングされる値が含まれています。列挙とその値の有効なリストを確認するには、'content=property_definitions' を使用してデバイスのエンドポイントをクエリーするか、ドキュメント定義のエンドポイントをクエリーします。

たとえば、"MyChannel" というチャネルの下にある "MyDevice" というデバイスのプロパティ定義を表示するには、GET リクエストを以下の URL に送信します。

```
https://<ホスト名または IP>:<ポート
>/config/v1/project/channels/MyChannel/devices/MyDevice/?content=property definitions
```

プロパティ定義は、チャネル、タグなど、その他のオブジェクトでも使用することができます。

または、ドライバーのチャネルとデバイスのプロパティ定義を構成 API の設定内で有効にすると、以下の URL でそれらのプロパティ定義を表示することができます。

```
https://<ホスト名または IP>:<ポート>/config/v1/doc/drivers/<ドライバー名>/Channels
```

```
https://<ホスト名または IP>:<ポート>/config/v1/doc/drivers/<ドライバー名>/Devices
```

データ型列挙の例

ドキュメントエンドポイントでタグのデータ型のクエリーを実行すると、以下の列挙が表示されます。

```
{
   "Default": -1,
   "String": 0,
   "Boolean": 1,
   "Char": 2,
   "Byte": 3,
   "Short": 4,
   "Word": 5,
   "Long": 6,
   "DWord": 7,
   "Float": 8,
   "Double": 9,
   "BCD": 10,
   "LBCD": 11,
   "Date": 12,
   "LLong": 13,
   "QWord": 14,
   "String Array": 20,
   "Boolean Array": 21,
   "Char Array": 22,
   "Byte Array": 23,
```

```
   "Short Array": 24,
   "Word Array": 25,
   "Long Array": 26,
   "DWord Array": 27,
   "Float Array": 28,
   "Double Array": 29,
   "BCD Array": 30,
   "LBCD Array": 31,
   "Date Array": 32,
   "LLong Array": 33,
   " QWord Array": 34
}
```

注記: サポートされるデータ型は、プロトコルとドライバーによって異なります。

デバイスモデル列挙

デバイスモデルのプロパティには、以下に示す列挙にマッピングされる値が含まれています。以下の表は、参照用として記載されています。デバイスエンドポイントの詳しい情報と最新の情報については、以下の URL を参照してください。

```
https://<ホスト名または IP>:<ポート
>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/Channels
```

```
https://<ホスト名または IP>:<ポート
>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/Devices
```

列挙	デバイスモデル
0	Modbus
1	メールボックス
2	Instromet
3	Roxar RFM
4	Fluenta FGM
5	Applicom
6	CEG

自動タグデータベース生成

このドライバーは自動タグデータベース生成をサポートしているため、ドライバーはデバイスのラダープログラムによって使用されるデータポイントにアクセスするタグを自動的に作成できます。構成に応じて、タグ生成はサーバープロジェクトが開始したときに自動的に開始するか、後から手動で開始できます。「イベントログ」には、タグ生成の開始時刻、変数インポートファイルの処理中に発生したエラー、このプロセスの完了時刻が示されます。

詳細については、サーバーのヘルプドキュメントを参照してください。

タグデータベースの構築に必要な情報をデバイスに対して照会可能な場合もありますが、このドライバーは代わりに変数インポートファイルを使用する必要があります。変数インポートファイルは Concept や ProWORX などのデバイスプログラミングアプリケーションを使用して生成できます。このインポートファイルは、Concept デバイスプログラミングアプリケーションのデフォルトのエクスポートファイルフォーマットであるセミコロン区切りの .txt フォーマットでなければなりません。

関連項目: カスタムアプリケーションからのインポート

変数インポートファイルの作成方法については、技術情報「Modbus ドライバー向け CSV ファイルの作成」を参照してください。

カスタムアプリケーションからのインポート

次の CSV ファイルフォーマットを使用してカスタムタグをインポートできます。

[レコードタイプ] ; [変数名] ; [データ型] ; [アドレス] ; [設定値] ; [コメント]

- **レコードタイプ:** これはタグをインポートするもう 1 つの手段である Concept ソフトウェアで使用されているフラグです。N または E を指定できます。どちらのフラグも同様に処理されます。
- **変数名:** これはサーバー内の静的タグの名前です。長さは最大 256 文字です。
- **データ型:** これはタグのデータ型です。サポートされるデータ型は次のとおりです。
  - BOOL
  - DINT
  - INT
  - REAL (32 ビット Float)
  - UDINT
  - UINT
  - WORD
  - BYTE
  - TIME (DWord として処理)
  - STRING
- **アドレス**: これはタグの Modbus アドレスです。長さは最大 16 文字です。
- **設定値**: これは無視され、空白のままにする必要があります。
- **コメント**: これはサーバー内のタグの説明です。長さは最大 255 文字です。

例

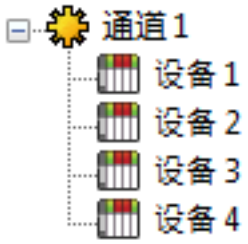
- N;Amps;WORD;40001;;Current in
- N;Volts;WORD;40003;;Volts in
- N;Temperature;REAL;40068;;Tank temp

通信の最適化

Modbus TCP/IP イーサネットドライバーは、システム全体のパフォーマンスへの影響を最小限に抑えながら最大のパフォーマンスが得られるように設計されています。このドライバーは高速ですが、このアプリケーションを制御および最適化して最大のパフォーマンスを得るために参考となるいくつかのガイドラインがあります。

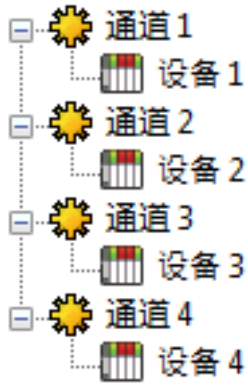
このサーバーでは、Modbus イーサネットなどの通信プロトコルのことをチャネルと呼びます。アプリケーションで定義されている各チャネルは、サーバーでの個々の実行パスを表します。チャネルが定義された後、そのチャネルの下に一連のデバイスを定義する必要があります。これらのデバイスそれぞれが、データの収集元となる単一の Modbus コントローラを表します。このアプローチに従ってアプリケーションを定義することで高いパフォーマンスが得られますが、ドライバーやネットワー

クがフルに利用されるわけではありません。単一のチャネルを使用して構成されているアプリケーションの表示例を次に示します。



各デバイスは、単一の Modbus イーサネットチャネルの下で定義されます。この構成では、ドライバーは効果的な速度で情報を収集するために、できるだけ速やかにあるデバイスから次のデバイスに移動する必要があります。さらにデバイスが追加されたり、1 つのデバイスからより多くの情報が要求されたりするにしたがい、全体的な更新レートが低下していきます。

Modbus TCP/IP イーサネットドライバーで定義できるチャネルの数が 1 つだけの場合、上に示した例が唯一の方法になりますが、このドライバーでは、最大 1024 チャネルを定義することができます。複数のチャネルを使用して複数の要求をネットワークに同時に発行することで、データ収集のワークロードが分散されます。パフォーマンスを改善するために同じアプリケーションを複数のチャネルを使用して構成した場合の例を次に示します。



各デバイスを、そのデバイス専用のチャネルの下で定義することができます。この構成では、各デバイスからのデータ収集タスクごとに 1 つの実行パスが割り当てられます。アプリケーションのデバイス数が 1024 台以下の場合は、この方法で最適化することができます。

アプリケーションのデバイスの数が多い場合でもパフォーマンスは改善されます。デバイスの数は少ないことが理想的ですが、そうでない場合でもアプリケーションは追加のチャネルから恩恵を受けます。デバイスの負荷をチャネルすべてに分散してもサーバーはデバイスを切り替えますが、単一のチャネルで処理するデバイスの数ははるかに少なくなります。

ブロックサイズ

ブロックサイズは、Modbus TCP/IP イーサネットドライバーのパフォーマンスに影響を与える可能性があります。ブロックサイズパラメータはデバイスごとに用意されており、デバイスプロパティのブロックサイズの設定で定義されています。ブロックサイズには、デバイスから一度に要求可能なレジスタまたはビットの数を指定します。ブロックサイズを 1 から 120 レジスタまたは 8 から 2000 ビットに設定することでドライバーのパフォーマンスを微調整できます。

ヒント:

- **「タイムアウト時にソケットを閉じる」**プロパティを有効にすることで、さらなるパフォーマンスゲインを実現できます。
- タイムアウトとタイミングのプロパティを調整することによっても、さらなるパフォーマンスゲインを実現できます。

詳細については、イーサネットのプロパティ、通信タイムアウト、タイミングを参照してください。

データ型の説明

データ型	説明
Boolean	1 ビット
Word	符号なし 16 ビット値 ビット 0 が下位ビット ビット 15 が上位ビット
Short	符号付き 16 ビット値 ビット 0 が下位ビット ビット 14 が上位ビット ビット 15 が符号ビット
DWord	符号なし 32 ビット値 ビット 0 が下位ビット ビット 31 が上位ビット
Long	符号付き 32 ビット値 ビット 0 が下位ビット ビット 30 が上位ビット ビット 31 が符号ビット
BCD	2 バイトパックされた BCD 値の範囲は 0-9999 です。この範囲外の値には動作が定義されていません。
LBCD	4 バイトパックされた BCD 値の範囲は 0-99999999 です。この範囲外の値には動作が定義されていません。
String	Null 終端 ASCII 文字列 Modbus モデルでサポートされ、バイトオーダーを HiLo/LoHi から選択できます。
Double*	64 ビット浮動小数点値 ドライバーは最後の 2 つのレジスタを上位 DWord、最初の 2 つのレジスタを下位 DWord とすることで、連続する 4 つのレジスタを倍精度値として解釈します。
Double の例	レジスタ 40001 が Double として指定されている場合、レジスタ 40001 のビット 0 は 64 ビットデータ型のビット 0 になり、レジスタ 40004 のビット 15 は 64 ビットデータ型のビット 63 になります。
Float*	32 ビット浮動小数点値 ドライバーは最後のレジスタを上位 Word、最初のレジスタを下位 Word とすることで、連続する 2 つのレジスタを単精度値として解釈します。
Float の例	レジスタ 40001 が Float として指定されている場合、レジスタ 40001 のビット 0 は 32 ビットデータ型のビット 0 になり、レジスタ 40002 のビット 15 は 32 ビットデータ型のビット 31 になります。

*この説明は、64 ビットデータ型では最初の DWord を下位とし、32 ビットデータ型では最初の Word を下位とするデフォルト設定のデータ処理を前提としています。

アドレスの説明

アドレスの仕様は使用されているモデルによって異なります。対象のモデルのアドレス情報を取得するには、次のリストからリンクを選択してください。

Applicom のアドレス指定
CEG のアドレス指定
Fluenta のアドレス指定
Instromet のアドレス指定
メールボックスのアドレス指定
Modbus のアドレス指定
Roxar のアドレス指定

ドライバーのシステムタグのアドレス指定

内部タグ

タグ	説明	データ型	アクセス
ポート	ポートのシステムタグにより、クライアントアプリケーションはポート番号の設定を読み書きできます。このタグへの書き込みが行われると、ドライバーはデバイスから切断し、指定されたポートへの再接続を試みます。	Word、Short、DWord、Long	読み取り/書き込み

注記:

- デバイスのポート設定は、ドライバーによるサーバー通信には使用されません。
- このドライバーでは、「サーバー」と「非送信請求」という用語は同じ意味になります。
- このタグに対する変更によってプロジェクトが修正され、サーバーはシャットダウン時にプロジェクトの保存を確認するプロンプトを表示します。

システムタグ

タグ	説明	データ型	アクセス
_CEGExtension	このタグはCEGモデルのデバイスのみに使用されます。このタグを使用することで、**CEG 拡張**のデバイスのプロパティをクライアントアプリケーションから変更できます。	Boolean	読み取り/書き込み
_InputCoilBlockSize	このタグを使用することで、「入力コイル」ブロックサイズのプロパティをクライアントアプリケーションから変更できます。	DWord	読み取り/書き込み
_OutputCoilBlockSize	このタグを使用することで、「出力コイル」ブロックサイズのプロパティをクライアントアプリケーションから変更できます。	DWord	読み取り/書き込み
_InternalRegisterBlockSize	このタグを使用することで、「内部レジスタ」ブロックサイズのプロパティをクライアントアプリケーションから変更できます。	DWord	読み取り/書き込み
_HoldingRegisterBlockSize	このタグを使用することで、「保持レジスタ」ブロックサイズのプロパティをクライアントアプリケーションから変更できます。	DWord	読み取り/書き込み

注記: これらのタグに対する変更によってプロジェクトが修正され、サーバーはシャットダウン時にプロジェクトの保存を確認するプロンプトを表示します。

関連項目: *イーサネット*

ファンクションコードの説明

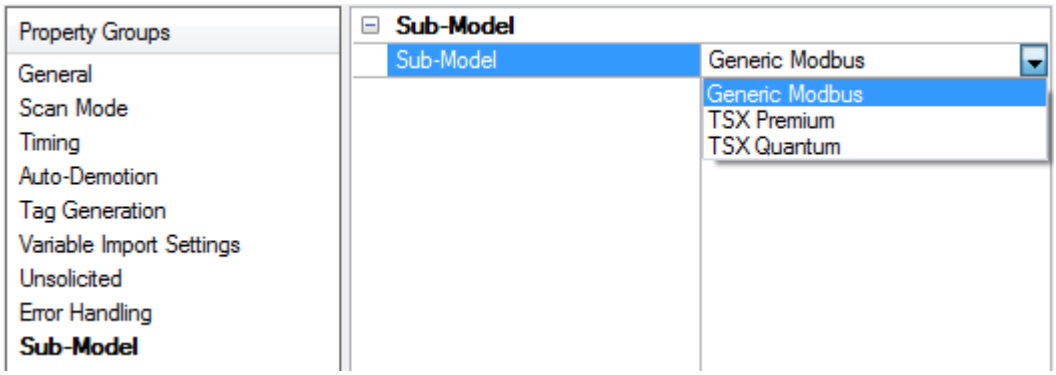
次の表に示すファンクションコードは Modbus モデルと Applicom モデルの デバイスによってサポートされています。

10 進	16 進	説明
01	0x01	コイルのステータスを読み取り
02	0x02	入力ステータスを読み取り
03	0x03	保持レジスタを読み取り
04	0x04	内部レジスタを読み取り
05	0x05	単一コイルを適用
06	0x06	単一レジスタをプリセット
15	0x0F	複数コイルを適用
16	0x10	複数レジスタをプリセット
22	0x16	レジスタへのマスク書き込み

Applicom のサブモデルとアドレス指定

Applicom デバイスは 3 つの Applicom サブモデルをサポートしています。接続するデバイスに適したサブモデルを選択します。アドレス情報については、以下のサブモデルのリンクをクリックしてください。

ジェネリック Modbus
TSX Premium
TSX Quantum



ジェネリック Modbus のアドレス指定

すべてのファンクションコードが 10 進数で表示されます。詳細については、ファンクションコードの説明を参照してください。

出力コイル

アドレス	範囲	データ型	アクセス	ファンクションコード
Bxxxxx	0-65535	Boolean	読み取り/書き込み	01, 05, 15

配列のサポート
出力コイルのアドレスでは配列がサポートされています。配列を宣言する構文を次に示します。
Bxxxxx_列数 (行数は 1 であるものと見なされます)。
Bxxxxx_行数_列数。

ベースアドレス + (行数 * 列数) が 65535 を超えてはなりません。要求されるコイルの総数が、このデバイスに指定された出力コイルのブロックサイズを超えてはなりません。

入力コイル

アドレス	範囲	データ型	アクセス	ファンクションコード
Blxxxxx	0-65535	Boolean	読み取り専用	02

配列のサポート
入力コイルのアドレスでは配列がサポートされています。配列を宣言する構文を次に示します。
Blxxxxx_列数 (行数は 1 であるものと見なされます)。
Blxxxxx_行数_列数。

ベースアドレス + (行数 * 列数) が 65535 を超えてはなりません。要求されるコイルの総数が、このデバイスに指定された入力コイルのブロックサイズを超えてはなりません。

内部レジスタ
デフォルトのデータ型を**太字**で示しています。

内部レジスタの位置では Boolean と String 以外のすべてのデータ型で配列がサポートされています。

注記: サーバーデバイスの場合、読み取り専用の位置は読み取り/書き込みが可能です。

アドレス	範囲	データ型	アクセス	ファンクションコード
Wlxxxxx	0-65535 0-65534 0-65532	**Word**、Short、BCD Float、DWord、Long、 LBCD Double	読み取り専用	04
Wlxxxxx.bb	xxxxx=0-65535 bb=0/1-15/16*	**Boolean**	読み取り専用	04
Wlxxxxx:Xbb	xxxxx=0-65535 bb=0/1-15/16*	**Boolean**	読み取り専用	04
Dlxxxxx	0-65534	**DWord**	読み取り専用	04
Flxxxxx	0-65534	**Float**	読み取り専用	04
Wlxxxxx_S	0-65535	**Short**	読み取り専用	04
Wlxxxxx_B	0-65535	**BCD**	読み取り専用	04
Wlxxxxx_A**	0-65535	**String**	読み取り専用	04
Wlxxxxx_X<1, 2, 3>***	0-65535 0-65534	**Word**、Short、BCD Float、DWord、Long、 LBCD	読み取り専用	04
Dlxxxxx_S	0-65534	**Long**	読み取り専用	04
Dlxxxxx_B	0-65534	**LBCD**	読み取り専用	04
Dlxxxxx_X<1, 2, 3>***	0-65534	**DWord**	読み取り専用	04
Flxxxxx_X<1, 2, 3>***	0-65534	**Float**	読み取り専用	04
M_Wlxxxxx_n(H) (String)、HiLo バイトオーダー (H はオプション)	xxxxx=0-65535 n は文字列長 範囲は 1 から 120 Word	**String**	読み取り専用	04
M_Wlxxxxx_nL (String)、LoHi バイトオーダー	xxxxx=0-65535 n は文字列長 範囲は 1 から 120	**String**	読み取り専用	04

アドレス	範囲	データ型	アクセス	ファンクションコード
	Word			

*詳細については、設定の「ゼロベースのビットアドレス指定」を参照してください。
**文字列の長さは2バイトです。
***詳細については、バイト交換のサフィックスを参照してください。

配列のサポート
内部レジスタアドレスでは配列がサポートされています。配列を宣言する構文を次に示します。
Wlxxxxx_列数 (行数は1であるものと見なされます)。
Wlxxxxx_行数_列数。

Word、Short、BCD 配列の場合、ベースアドレス + (行数 * 列数) が 65535 を超えてはなりません。
Float、DWord、Long、および Long BCD 配列の場合、ベースアドレス + (行数 * 列数 * 2) が 65534 を超えてはなりません。
すべての配列で、要求されるレジスタの総数が、このデバイスに指定された内部レジスタのブロックサイズを超えてはなりません。

保持レジスタ
デフォルトのデータ型を**太字**で示しています。

保持レジスタの位置では Boolean と String 以外のすべてのデータ型で配列がサポートされています。

注記: サーバーデバイスの場合、読み取り専用の位置は読み取り/書き込みが可能です。

アドレス	範囲	データ型	アクセス	ファンクションコード
Wxxxxx	0-65535 0-65534 0-65532	**Word**、Short、BCD Float、DWord、 Long、LBCD Double	読み取り/書き込み	03, 06, 16
Wxxxxx.bb	xxxxx=0-65535 bb=0/1-15/16*	**Boolean**	読み取り/書き込み	03, 06, 16, 22
Wxxxxx:Xbb	xxxxx=0-65535 bb=0/1-15/16*	**Boolean**	読み取り/書き込み	03, 06, 16, 22
Dxxxxx	0-65534	**DWord**	読み取り/書き込み	03, 06, 16
Fxxxxx	0-65534	**Float**	読み取り/書き込み	03, 06, 16
Wxxxxx_S	0-65535	**Short**	読み取り/書き込み	03, 06, 16
Wxxxxx_B	0-65535	**BCD**	読み取り/書き込み	03, 06, 16
Wxxxxx_A**	0-65535	**String**	読み取り専用	03, 16
Wxxxxx_X<1, 2, 3>***	0-65535 0-65534	**Word**、Short、BCD Float、DWord、 Long、LBCD	読み取り/書き込み	03, 06, 16
Dxxxxx_S	0-65534	**Long**	読み取り/書き込み	03, 06, 16
Dxxxxx_B	0-65534	**LBCD**	読み取り/書き込み	03, 06, 16
Dxxxxx_X<1, 2, 3>***	0-65534	**DWord**	読み取り/書き込み	03, 06, 16
Fxxxxx_X<1, 2, 3>***	0-65534	**Float**	読み取り/書	03, 06, 16

アドレス	範囲	データ型	アクセス	ファンクションコード
			き込み	
M_Wxxxxx_n(H) (String)、HiLo バイトオーダー (H はオプション)	xxxxx=0-65535 n は文字列長 範囲は 1 から 120 Word	**String**	読み取り/書き込み	03, 16
M_Wxxxxx_nL (String)、LoHi バイトオーダー	xxxxx=0-65535 n は文字列長 範囲は 1 から 120 Word	**String**	読み取り/書き込み	03, 16

*詳細については、設定の「ゼロベースのビットアドレス指定」を参照してください。
**文字列の長さは 2 バイトです。
***詳細については、バイト交換のサフィックスを参照してください。

配列のサポート
保持レジスタアドレスでは配列がサポートされています。10 進アドレス指定を使用して配列を宣言する構文を次に示します。
Wxxxxx_列数 (行数は 1 であるものと見なされます)。
Wxxxxx_行数_列数。

Word、Short、BCD 配列の場合、ベースアドレス + (行数 * 列数) が 65535 を超えてはなりません。
Float、DWord、Long、および Long BCD 配列の場合、ベースアドレス + (行数 * 列数 * 2) が 65534 を超えてはなりません。
すべての配列で、要求されるレジスタの総数が、このデバイスに指定された保持レジスタのブロックサイズを超えてはなりません。

文字列のサポート
Applicom モデルでは保持レジスタメモリを ASCII 文字列として読み書きできます。文字列データに保持レジスタを使用している場合、各レジスタに 2 バイトの ASCII データが格納されます。文字列の長さは 1 から 120 Word の範囲です。文字列タグに対してブロック読み取りを実行する方法については、ブロックサイズを参照してください。

注記: デバイスで許可される書き込み要求の最大サイズによって文字列の長さが制限されることがあります。サーバーのイベントウィンドウで「デバイス <デバイス> のアドレス <アドレス> に書き込めません: デバイスは例外コード 3 を返しました」というエラーメッセージを受信した場合、そのデバイスはこの文字列長をサポートしていません。これを解決するには、サポートされている長さまで文字列を短くしてください。

バイト交換のサフィックス
これらのサフィックスは 16 ビット Word、32 ビット DWord、または 32 ビット Float のデータを構成するバイトの交換に使用されます。バイト交換は、デバイスレベルの設定である「Modbus バイトオーダー」と「最初の Word を下位とする」が適用された後で適用されます。詳細については、設定を参照してください。

バイト交換のサフィックスは内部レジスタおよび保持レジスタでのみ使用できます。アイテムのサフィックスとデータ型に依存する各種交換については、次の表を参照してください。

サフィックス	16 ビットデータ型 (Word、Short、BCD)	32 ビットデータ型 (DWord、Long、LBCD、Float)
_X1	O1 O2 -> O2 O1 (バイト交換)	O1 O2 O3 O4 -> O4 O3 O2 O1 (バイト交換)
_X2	O1 O2 -> O2 O1 (バイト交換)	O1 O2 O3 O4 -> O3 O4 O1 O2 (Word 交換)
_X3	O1 O2 -> O2 O1 (バイト交換)	O1 O2 O3 O4 -> O2 O1 O4 O3 (Word 内のバイト交換)

TSX Quantum
すべてのファンクションコードが 10 進数で表示されます。詳細については、*ファンクションコードの説明*を参照してください。

出力コイル

アドレス	範囲	データ型	アクセス	ファンクションコード
0xxxxx	1-65536	Boolean	読み取り/書き込み	01, 05, 15

配列のサポート
出力コイルのアドレスでは配列がサポートされています。配列を宣言する構文を次に示します。
0xxxxx_列数 (行数は 1 であるものと見なされます)。
0xxxxx_行数_列数。

ベースアドレス + (行数 * 列数) が 65536 を超えてはなりません。要求されるコイルの総数が、このデバイスに指定された出力コイルのブロックサイズを超えてはなりません。

入力コイル

アドレス	範囲	データ型	アクセス	ファンクションコード
1xxxxx	1-65536	Boolean	読み取り専用	02

配列のサポート
入力コイルのアドレスでは配列がサポートされています。配列を宣言する構文を次に示します。
1xxxxx_列数 (行数は 1 であるものと見なされます)。
1xxxxx_行数_列数。

ベースアドレス + (行数 * 列数) が 65536 を超えてはなりません。要求されるコイルの総数が、このデバイスに指定された入力コイルのブロックサイズを超えてはなりません。

内部レジスタ
デフォルトのデータ型を**太字**で示しています。

内部レジスタの位置では Boolean と String 以外のすべてのデータ型で配列がサポートされています。

注記: サーバーデバイスの場合、読み取り専用の位置は読み取り/書き込みが可能です。

アドレス	範囲	データ型	アクセス	ファンクションコード
3xxxxx	1-65536 1-65535 1-65533	**Word**、Short、BCD Float、DWord、Long、 LBCD Double	読み取り専用	04
3xxxxx.bb	xxxxx=1-65536 bb=0/1-15/16*	**Boolean**	読み取り専用	04
3xxxxx:Xbb	xxxxx=0-65535 bb=0/1-15/16*	**Boolean**	読み取り専用	04
D3xxxxx	1-65535	**DWord**	読み取り専用	04
F3xxxxx	1-65535	**Float**	読み取り専用	04
3xxxxx_S	1-65536	**Short**	読み取り専用	04
3xxxxx_B	1-65536	**BCD**	読み取り専用	04
3xxxxx_A**	1-65536	**String**	読み取り専用	04
3xxxxx_X<1, 2, 3>***	1-65536 1-65535	**Word**、Short、BCD Float、DWord、Long、 LBCD	読み取り専用	04
D3xxxxx_S	1-65535	**Long**	読み取り専用	04
D3xxxxx_B	1-65535	**LBCD**	読み取り	04

アドレス	範囲	データ型	アクセス	ファンクションコード
			専用	
D3xxxxx_X<1, 2, 3>***	1-65535	**DWord**	読み取り専用	04
F3xxxxx_X<1, 2, 3>***	1-65535	**Float**	読み取り専用	04
M_3xxxxx_n(H) (String)、HiLo バイトオーダー (H はオプション)	xxxxx=1-65536 n は文字列長 範囲は 1 から 120 Word	**String**	読み取り専用	04
M_3xxxxx_nL (String)、LoHi バイトオーダー	xxxxx=1-65536 n は文字列長 範囲は 1 から 120 Word	**String**	読み取り専用	04

*詳細については、**設定**の「ゼロベースのビットアドレス指定」を参照してください。
**文字列の長さは 2 バイトです。
***詳細については、**バイト交換のサフィックス**を参照してください。

配列のサポート
内部レジスタアドレスでは配列がサポートされています。配列を宣言する構文を次に示します。
3xxxxx_列数 (行数は 1 であるものと見なされます)。
3xxxxx_行数_列数。

Word、Short、BCD 配列の場合、ベースアドレス + (行数 * 列数) が 65536 を超えてはなりません。
Float、DWord、Long、および Long BCD 配列の場合、ベースアドレス + (行数 * 列数 * 2) が 65535 を超えてはなりません。
すべての配列で、要求されるレジスタの総数が、このデバイスに指定された内部レジスタのブロックサイズを超えてはなりません。

保持レジスタ

デフォルトのデータ型を**太字**で示しています。

保持レジスタの位置では Boolean と String 以外のすべてのデータ型で配列がサポートされています。

注記: サーバーデバイスの場合、読み取り専用の位置は読み取り/書き込みが可能です。

アドレス	範囲	データ型	アクセス	ファンクションコード
4xxxxx	1-65536 1-65535 1-65533	**Word**、Short、BCD Float、DWord、 Long、LBCD Double	読み取り/書き込み	03, 06, 16
4xxxxx.bb	xxxxx=1-65536 bb=0/1-15/16*	**Boolean**	読み取り/書き込み	03, 06, 16, 22
4xxxxx:Xbb	xxxxx=0-65535 bb=0/1-15/16*	**Boolean**	読み取り/書き込み	03, 06, 16, 22
D4xxxxx	1-65535	**DWord**	読み取り/書き込み	03, 06, 16
F4xxxxx	1-65535	**Float**	読み取り/書き込み	03, 06, 16
4xxxxx_S	1-65536	**Short**	読み取り/書き込み	03, 06, 16
4xxxxx_B	1-65536	**BCD**	読み取り/書き込み	03, 06, 16
4xxxxx_A**	1-65536	**String**	読み取り専	03, 16

アドレス	範囲	データ型	アクセス	ファンクションコード
			用	
4xxxxx_X<1, 2, 3>***	1-65536 1-65535	**Word**、Short、BCD Float、DWord、 Long、LBCD	読み取り/書き込み	03, 06, 16
D4xxxxx_S	1-65535	**Long**	読み取り/書き込み	03, 06, 16
D4xxxxx_B	1-65535	**LBCD**	読み取り/書き込み	03, 06, 16
D4xxxxx_X<1, 2, 3>***	1-65535	**DWord**	読み取り/書き込み	03, 06, 16
F4xxxxx_X<1, 2, 3>***	1-65535	**Float**	読み取り/書き込み	03, 06, 16
M_4xxxxx_n(H) (String)、HiLo バイトオーダー (H はオプション)	xxxxx=1-65536 n は文字列長 範囲は 1 から 120 Word	**String**	読み取り/書き込み	03, 16
M_4xxxxx_nL (String)、LoHi バイトオーダー	xxxxx=1-65536 n は文字列長 範囲は 1 から 120 Word	**String**	読み取り/書き込み	03, 16

*詳細については、**設定**の「ゼロベースのビットアドレス指定」を参照してください。
**文字列の長さは 2 バイトです。
詳細については、バイト交換のサフィックス***を参照してください。

配列のサポート
保持レジスタアドレスでは配列がサポートされています。10 進アドレス指定を使用して配列を宣言する構文を次に示します。

4xxxxx_列数 (行数は 1 であるものと見なされます)。
4xxxxx_行数_列数。

Word、Short、BCD 配列の場合、ベースアドレス + (行数 * 列数) が 65536 を超えてはなりません。
Float、DWord、Long、および Long BCD 配列の場合、ベースアドレス + (行数 * 列数 * 2) が 65535 を超えてはなりません。
すべての配列で、要求されるレジスタの総数が、このデバイスに指定された保持レジスタのブロックサイズを超えてはなりません。

文字列のサポート
Applicom モデルでは保持レジスタメモリを ASCII 文字列として読み書きできます。文字列データに保持レジスタを使用している場合、各レジスタに 2 バイトの ASCII データが格納されます。文字列の長さは 1 から 120 Word の範囲です。
*文字列タグに対してブロック読み取りを実行する方法については、**ブロックサイズ**を参照してください。*

注記: デバイスで許可される書き込み要求の最大サイズによって文字列の長さが制限されることがあります。サーバーのイベントウィンドウで「デバイス <デバイス> のアドレス <アドレス> に書き込めません: デバイスは例外コード 3 を返しました」というエラーメッセージを受信した場合、そのデバイスはこの文字列長をサポートしていません。これを解決するには、サポートされている長さまで文字列を短くしてください。

バイト交換のサフィックス
これらのサフィックスは 16 ビット Word、32 ビット DWord、または 32 ビット Float のデータを構成するバイトの交換に使用されます。バイト交換は、デバイスレベルの設定である「Modbus バイトオーダー」と「最初の Word を下位とする」が適用された後で適用されます。詳細については、**設定**を参照してください。

バイト交換のサフィックスは内部レジスタおよび保持レジスタでのみ使用できます。アイテムのサフィックスとデータ型に依存する各種交換については、次の表を参照してください。

サフィックス	16 ビットデータ型 (Word、Short、BCD)	32 ビットデータ型 (DWord、Long、LBCD、Float)
_X1	O1 O2 -> O2 O1 (バイト交換)	O1 O2 O3 O4 -> O4 O3 O2 O1 (バイト交換)
_X2	O1 O2 -> O2 O1 (バイト交換)	O1 O2 O3 O4 -> O3 O4 O1 O2 (Word 交換)
_X3	O1 O2 -> O2 O1 (バイト交換)	O1 O2 O3 O4 -> O2 O1 O4 O3 (Word 内のバイト交換)

TSX Premium

すべてのファンクションコードが 10 進数で表示されます。詳細については、ファンクションコードの説明を参照してください。

出力コイル

アドレス	範囲	データ型	アクセス	ファンクションコード
%MXxxxxx	0-65535	Boolean	読み取り/書き込み	01, 05, 15
%Mxxxxx	0-65535	Boolean	読み取り/書き込み	01, 05, 15

配列のサポート
出力コイルのアドレスでは配列がサポートされています。配列を宣言する構文を次に示します。
%MXxxxxx_列数 (行数は 1 であるものと見なされます)。
%MXxxxxx_行数_列数。

ベースアドレス + (行数 * 列数) が 65535 を超えてはなりません。要求されるコイルの総数が、このデバイスに指定された出力コイルのブロックサイズを超えてはなりません。

保持レジスタ

デフォルトのデータ型を**太字**で示しています。

保持レジスタの位置では Boolean と String 以外のすべてのデータ型で配列がサポートされています。

注記: サーバーデバイスの場合、読み取り専用の位置は読み取り/書き込みが可能です。

アドレス	範囲	データ型	アクセス	ファンクションコード
%MWxxxxx	0-65535 0-65534 0-65532	**Word**、Short、BCD Float、DWord、Long、 LBCD Double	読み取り/書き込み	03, 06, 16
%MWxxxxx.bb	xxxxx=0-65535 bb=0/1-15/16*	**Boolean**	読み取り/書き込み	03, 06, 16, 22
%MWxxxxx:Xbb	xxxxx=0-65535 bb=0/1-15/16*	**Boolean**	読み取り/書き込み	03, 06, 16, 22
%DWxxxxx または %MDxxxxx	0-65534	**DWord**	読み取り/書き込み	03, 06, 16
%FWxxxxx または %MFxxxxx	0-65534	**Float**	読み取り/書き込み	03, 06, 16
%MWxxxxx_S	0-65535	**Short**	読み取り/書き込み	03, 06, 16
%MWxxxxx_B	0-65535	**BCD**	読み取り/書き込み	03, 06, 16
%MWxxxxx_A**	0-65535	**String**	読み取り専用	03, 16
%MWxxxxx_X<1, 2, 3>***	0-65535 0-65534	**Word**、Short、BCD Float、DWord、Long、 LBCD	読み取り/書き込み	03, 06, 16
%DWxxxxx_S	0-65534	**Long**	読み取り/書き込み	03, 06, 16

アドレス	範囲	データ型	アクセス	ファンクションコード
%DWxxxxx_B	0-65534	**LBCD**	読み取り/書き込み	03, 06, 16
%DWxxxxx_X<1, 2, 3>*** または %MDxxxxx_X<1, 2, 3>***	0-65534	**DWord**	読み取り/書き込み	03, 06, 16
%FWxxxxx_X<1, 2, 3>*** または %MFxxxxx_X<1, 2, 3>***	0-65534	**Float**	読み取り/書き込み	03, 06, 16
M_%MWxxxxx_n(H) String、HiLo バイトオーダー (H はオプション)	xxxxx=0-65535 n は文字列長 範囲は 1 から 120 Word	**String**	読み取り/書き込み	03, 16
M_%MWxxxxx_nL (String)、LoHi バイトオーダー	xxxxx=0-65535 n は文字列長 範囲は 1 から 120 Word	**String**	読み取り/書き込み	03, 16

*詳細については、**設定**の「ゼロベースのビットアドレス指定」を参照してください。
**文字列の長さは 2 バイトです。
***詳細については、**バイト交換のサフィックス**を参照してください。

配列のサポート
保持レジスタアドレスでは配列がサポートされています。10 進アドレス指定を使用して配列を宣言する構文を次に示します。
%MWxxxxx_列数 (行数は 1 であるものと見なされます)。
%MWxxxxx_行数_列数。

Word、Short、BCD 配列の場合、ベースアドレス + (行数 * 列数) が 65535 を超えてはなりません。
Float、DWord、Long、および Long BCD 配列の場合、ベースアドレス + (行数 * 列数 * 2) が 65534 を超えてはなりません。
すべての配列で、要求されるレジスタの総数が、このデバイスに指定された保持レジスタのブロックサイズを超えてはなりません。

文字列のサポート
Applicom モデルでは保持レジスタメモリを ASCII 文字列として読み書きできます。文字列データに保持レジスタを使用している場合、各レジスタに 2 バイトの ASCII データが格納されます。文字列の長さは 1 から 120 Word の範囲です。文字列タグに対してブロック読み取りを実行する方法については、**ブロックサイズ**を参照してください。

注記: デバイスで許可される書き込み要求の最大サイズによって文字列の長さが制限されることがあります。サーバーのイベントウィンドウで「デバイス <デバイス> のアドレス <アドレス> に書き込めません: デバイスは例外コード 3 を返しました」というエラーメッセージを受信した場合、そのデバイスはこの文字列長をサポートしていません。これを解決するには、サポートされている長さまで文字列を短くしてください。

バイト交換のサフィックス
これらのサフィックスは 16 ビット Word、32 ビット DWord、または 32 ビット Float のデータを構成するバイトの交換に使用されます。バイト交換は、デバイスレベルの設定である「Modbus バイトオーダー」と「最初の Word を下位とする」が適用された後で適用されます。詳細については、**設定**を参照してください。

バイト交換のサフィックスは内部レジスタおよび保持レジスタでのみ使用できます。アイテムのサフィックスとデータ型に依存する各種交換については、次の表を参照してください。

サフィックス	16 ビットデータ型 (Word、Short、BCD)	32 ビットデータ型 (DWord、Long、LBCD、Float)
_X1	O1 O2 -> O2 O1 (バイト交換)	O1 O2 O3 O4 -> O4 O3 O2 O1 (バイト交換)
_X2	O1 O2 -> O2 O1 (バイト交換)	O1 O2 O3 O4 -> O3 O4 O1 O2 (Word 交換)
_X3	O1 O2 -> O2 O1 (バイト交換)	O1 O2 O3 O4 -> O2 O1 O4 O3 (Word 内のバイト交

サフィックス	16 ビットデータ型 (Word、Short、BCD)	32 ビットデータ型 (DWord、Long、LBCD、Float)
		換)

CEG のアドレス指定

CEG デバイスモデルのアドレス指定は Modbus デバイスモデルのアドレス指定と同じです。

詳細については、Modbus のアドレス指定を参照してください。

Fluenta のアドレス指定

デフォルトのデータ型を**太字**で示しています。

アドレス	範囲	データ型	アクセス
システム	400000-409999	**Float**、Double	読み取り/書き込み
出力	410000-410999 420000-420999 430000-430999	**Float**、Double	読み取り専用
ユーザー	411000-411999 421000-421999 431000-431999	**Float**、Double	読み取り/書き込み
サービス	412000-412999 422000-422999 432000-432999	**Float**、Double	読み取り/書き込み
累積	413000-413999 423000-423999 433000-433999	**Float**、Double	読み取り専用

Instromet のアドレス指定

デフォルトのデータ型を**太字**で示しています。

アドレス	範囲	データ型	アクセス
短整数	400000-400199	**Word**、Short	読み取り専用
長整数	400200-400399	**DWord**、Long	読み取り専用
単精度実数	400400-400599	**Float**	読み取り専用

メールボックスのアドレス指定

デフォルトのデータ型を**太字**で示しています。

10 進アドレス指定

アドレス	範囲	データ型	アクセス
4xxxxx	1-65536	Word、Short、BCD	読み取り/書き込み
4xxxxx.bb	xxxxx=1-65536 bb=0-15	**Boolean**	読み取り/書き込み
4xxxxx	1-65535	Float、DWord、Long、LBCD	読み取り/書き込み

16 進アドレス指定

アドレス	範囲	データ型	アクセス
H4yyyyy	1-10000	Word、Short、BCD	読み取り/書き込み
H4yyyyy.c	yyyyy=1-10000 c=0-F	**Boolean**	読み取り/書き込み
H4yyyy	1-FFFF	Float、DWord、Long、LBCD	読み取り/書き込み

注記: Modbus メールボックスではファンクションコード 22 (0x16) はサポートされていません。0x10 (複数の保持レジスタへの書き込み) と 0x6 (単一の保持レジスタへの書き込み) はサポートされています。デバイスのプロパティの「設定」タブにある**「保持レジスタのビットマスク」**をオフにすることで、個々のビットに書き込むことが可能です。これにより、ビットに直接書

き込む代わりに、読み取り/修正/書き込みシーケンスが使用されます。これを機能させるためには、(メールボックスではなく) クライアント Modbus デバイスの設定だけを変更する必要があります。

配列
保持レジスタアドレスでは配列もサポートされています。(10 進アドレス指定を使用して) 配列を宣言する構文を次に示します。

4xxxx[列数] (行数は 1 であるものと見なされます)。
4xxxx[行数][列数]。

Word、Short、BCD 配列の場合、ベースアドレス + (行数 * 列数) が 65536 を超えてはなりません。

Float、DWord、Long、および Long BCD 配列の場合、ベースアドレス + (行数 * 列数 * 2) が 65535 を超えてはなりません。

すべての配列で、要求されるレジスタの総数が、このデバイスに指定された保持レジスタのブロックサイズを超えてはなりません。

Modbus のアドレス指定
このドライバーでは、「サーバー」と「非送信請求」という用語は同じ意味になります。

5 桁のアドレス指定と 6 桁のアドレス指定
Modbus のアドレス指定では、アドレスの最初の桁はプライマリテーブルを示します。以降の桁はデバイスのデータアイテムを表します。データアイテムの最大値は 2 バイトの符号なし整数 (65,535) です。内部では、このドライバーはアドレステーブルとアイテム全体を表すのに 6 桁を必要とします。Modbus デバイスの多くはすべてのデータアイテムをサポートしていないことに注意してください。そのようなデバイスのアドレスを入力する際の混乱を回避するため、このドライバーはアドレスフィールドに入力されたものに従ってアドレスに "パディング" (桁を追加) します。プライマリテーブルタイプの後ろに最大 4 桁ある場合 (例: 4x、4xx、4xxx、4xxxx)、アドレスはそのままになるか、5 桁までゼロが追加されます。プライマリテーブルタイプの後ろに 5 桁ある場合 (例: 4xxxxx)、アドレスは変わりません。内部では、41、401、4001、40001、または 400001 として入力されたアドレスはすべて、プライマリテーブルタイプ 4 とデータアイテム 1 を示すアドレスを表します。

プライマリテーブル	説明
0	出力コイル
1	入力コイル
3	内部レジスタ
4	保持レジスタ

Modbus のアドレス指定 (10 進フォーマット)
ファンクションコードは 10 進数で表示されます。詳細については、ファンクションコードの説明を参照してください。

アドレスタイプ	範囲	データ型	アクセス*	ファンクションコード
出力コイル	000001-065536	Boolean	読み取り/書き込み	01, 05, 15
入力コイル	100001-165536	Boolean	読み取り専用	02
内部レジスタ	300001-365536 300001-365535 300001-365533 xxxxx=1-65536 bb=0/1-15/16** 300001.2H-365536.240H*** 300001.2L-365536.240L***	**Word**、Short、BCD Float、DWord、Long、LBCD Double **Boolean** String String	読み取り専用 読み取り専用 読み取り専用 読み取り専用 読み取り専用	04 04 04 04 04 04

アドレスタイプ	範囲	データ型	アクセス*	ファンクションコード
			読み取り専用 読み取り専用	
保持レジスタ	400001-465536 400001-465535 400001-465533 xxxxx=1-65536 bb=0/1-15/16* 400001.2H-465536.240H*** 400001.2L-465536.240L***	**Word**、Short、BCD Float、DWord、Long、LBCD Double **Boolean** **String** **String**	読み取り/書き込み 読み取り/書き込み 読み取り/書き込み 読み取り/書き込み 読み取り/書き込み 読み取り/書き込み	 03, 06, 16 03, 06, 16 03, 06, 16 03, 06, 16, 22 03, 16 03, 16

*サーバーデバイスの場合、読み取り専用の位置は読み取り/書き込みが可能です。
***詳細については、**設定**のゼロベースアドレス指定を参照してください。*
***ピリオドの後ろのビット番号は 2 から 240 バイトの範囲の文字列長を示します。

Modbus のアドレス指定 (16 進フォーマット)

アドレスタイプ	範囲	データ型	アクセス*
出力コイル	H000001-H010000	Boolean	読み取り/書き込み
入力コイル	H100001-H110000	Boolean	読み取り専用
内部レジスタ	H300001-H310000 H300001-H30FFFF H300001-H30FFFD yyyyy=1-10000 cc=0/1-F/10 H300001.2H-H3FFFF.240H** H300001.2L-H3FFFF.240L**	**Word**、Short、BCD Float、DWord、Long、LBCD Double **Boolean** String String	読み取り専用 読み取り専用 読み取り専用 読み取り専用 読み取り専用 読み取り専用
保持レジスタ	H400001-H410000 H400001-H40FFFF H400001-H40FFFD yyyyy=1-10000 cc=0/1-F/10 H400001.2H-H4FFFF.240H H400001.2L-H4FFFF.240L	**Word**、Short、BCD Float、DWord、Long、LBCD Double **Boolean** **String** **String**	読み取り/書き込み 読み取り/書き込み 読み取り/書き込み 読み取り/書き込み 読み取り/書き込み 読み取り/書き込み

アドレスタイプ	範囲	データ型	アクセス*
			読み取り/書き込み

*サーバーデバイスの場合、読み取り専用の位置は読み取り/書き込みが可能です。
**ピリオドの後ろのビット番号は2から240バイトの範囲の文字列長を示します。

パックコイル

パックタイプのコイルアドレスでは、複数の連続するコイルにアナログ値としてアクセスできます。この機能は入力コイルと出力コイルの両方で、ポーリングモードでのみ使用できます。非送信請求メモリマップにアクセスするよう設定されているデバイスやメールボックスモードのデバイスでは使用できません。10進の構文は0*xxxxx#nn*であり、ここで:
*サーバーデバイスの場合、読み取り専用の位置は読み取り/書き込みが可能です。
**ピリオドの後ろのビット番号は2から240バイトの範囲の文字列長を示します。

パックコイル

パックタイプのコイルアドレスでは、複数の連続するコイルにアナログ値としてアクセスできます。この機能は入力コイルと出力コイルの両方で、ポーリングモードでのみ使用できます。非送信請求メモリマップにアクセスするよう設定されているデバイスやメールボックスモードのデバイスでは使用できません。10進の構文は0*xxxxx#nn*であり、ここで:

- *xxxxx*は1つ目のコイルのアドレスです(範囲は000001-065521)。
- *nn*はアナログ値にパックされるコイルの数です(範囲は01-16)。

16進の構文はH0*yyyyy#nn*であり、ここで:

- *yyyyy*は1つ目のコイルのアドレスです(範囲はH000001-H000FFF1)。
- *nn*はアナログ値にパックされるコイルの数です(範囲は01-16)。

注記:

1. 有効な唯一のデータ型がWordです。出力コイルでは読み取り/書き込みのアクセスが可能であり、入力コイルでは読み取り専用のアクセスが可能です。10進アドレス指定では、出力コイルはファンクションコード01と15をサポートするのに対し、入力コイルはファンクションコード02をサポートします。
2. 開始アドレスがアナログ値の最下位ビット(LSB)となるビットオーダーになります。

書き込み専用アクセス

"W40001"などのように、アドレスの先頭に"W"を付けることによって、すべての読み取り/書き込み可能アドレスを書き込み専用として設定でき、これによってドライバーはレジスタの指定したアドレスを読み取れなくなります。クライアントが書き込み専用タグを読み取ろうとすると、指定したアドレスへの最後に成功した書き込みの値が取得されます。成功した書き込みがない場合、クライアントは数値/文字列値の初期値である0/NULLを受信します。
警告: 書き込み専用タグのクライアントアクセス権限を読み取り専用に設定した場合、これらのタグへの書き込みは失敗し、クライアントは数値/文字列値として必ず0/NULLを受信します。

メールボックスモード

メールボックスモードでは保持レジスタのみがサポートされます。クライアントから読み取る場合、データは物理デバイスからではなく、キャッシュからローカルに読み取られます。クライアントから書き込む場合、データはデバイスIDのルーティングパスによって指定されている物理デバイスとローカルキャッシュの両方に書き込まれます。
注記: Double データ型はサポートされません。

文字列のサポート

Modbusモデルでは保持レジスタメモリをASCII文字列として読み書きできます。文字列データに保持レジスタを使用している場合、各レジスタに2バイトのASCIIデータが格納されます。文字列を定義する際に、そのレジスタにおけるASCIIデータの順序を選択できます。文字列の長さは2から240バイトの範囲で指定でき、ビット番号の位置に入力します。この長さは偶数として入力する必要があります。"H"または"L"をアドレスに追加することでバイトオーダーが指定されます。
Modbusモデルの文字列タグに対してブロック読み取りを実行する方法については、ブロックサイズを参照してください。

例

1. 40200 で開始し、長さが 100 バイト、HiLo バイトオーダーの文字列をアドレス指定するには、"40200.100H" と入力します。
2. 40500 で開始し、長さが 78 バイト、LoHi バイトオーダーの文字列をアドレス指定するには、"40500.78L" と入力します。

注記: デバイスで許可される書き込み要求の最大サイズによって文字列の長さが制限されることがあります。サーバーのイベントウィンドウで「デバイス <デバイス> のアドレス <アドレス> に書き込めません: デバイスは例外コード 3 を返しました」というエラーメッセージを受信した場合、文字列の長さがそのデバイスに適していませんでした。可能な場合、文字列を短くしてみてください。

配列のサポート

内部レジスタと保持レジスタの位置 (Boolean と String を除くすべてのデータ型) および入力コイルと出力コイル (Boolean データ型) では配列がサポートされています。配列のアドレスを指定するには 2 つの方法があります。次の例は保持レジスタに当てはまります。
4*xxxx* [行数] [列数]
4*xxxx* [列数] (行数は 1 であるものと見なされます)。

Word、Short、BCD 配列の場合、ベースアドレス + (行数 * 列数) が 65536 を超えてはなりません。Float、DWord、Long、および Long BCD 配列の場合、ベースアドレス + (行数 * 列数 * 2) が 65535 を超えてはなりません。すべての配列で、要求されるレジスタの総数が、このデバイスに指定された保持レジスタのブロックサイズを超えてはなりません。

Roxar のアドレス指定

デフォルトのデータ型を**太字**で示しています。

アドレス	範囲	データ型	アクセス
短整数	403000-403999	**Word**、Short	読み取り/書き込み
単精度実数	407000-407999	**Float**	読み取り/書き込み
単精度実数	409000-409999	**Float**	読み取り専用

イベントログメッセージ

次の情報は、メインユーザーインタフェースの「イベントログ」枠に記録されたメッセージに関するものです。「イベントログ」詳細ビューのフィルタリングとソートについては、OPC サーバーのヘルプを参照してください。 サーバーのヘルプには共通メッセージが多数含まれているので、これらも参照してください。通常は、可能な場合、メッセージのタイプ (情報、警告) とトラブルシューティングに関する情報が提供されています。

Winsock 通信を開始できませんでした。

エラータイプ:
エラー

非送信請求通信を開始できませんでした。

エラータイプ:
エラー

考えられる原因:
ドライバーは非送信請求通信用のリッスンソケットを作成できませんでした。

解決策:
チャネルレベルで定義されているポートがシステム上の別のアプリケーションによって使用されていないことを確認してください。

注記:
このドライバーでは、「Modbus サーバー」と「非送信請求」という用語は同義で用いられています。

未定義のデバイスに対して非送信請求メールボックスアクセスが行われました。ソケットを閉じています。| IP アドレス = '<アドレス>'。

エラータイプ:
エラー

考えられる原因:

1. 指定されている IP アドレスを持つデバイスがサーバーに対してメールボックスメッセージを送信しようとしました。その IP を持つデバイスがメールボックスプロジェクトで設定されていないため、メッセージは検証を通過しませんでした。
2. 指定されている IP アドレスを持つデバイスがサーバーに対してメールボックスメッセージを送信しようとしました。デバイスは設定されていますが、そこからデータを要求しているクライアントがないため、メッセージは検証を通過しませんでした。

解決策:
サーバーがメールボックスメッセージを受け入れるためには、指定されているデバイス IP がプロジェクトで設定されている必要があります。デバイスの少なくとも 1 つのデータアイテムがクライアントによって要求されている必要があります。

受信した要求は非送信請求メールボックスでサポートされていません。| IP アドレス = '<アドレス>'。

エラータイプ:
エラー

考えられる原因:
指定されているデバイス IP からサポートされていない要求を受信しました。要求のフォーマットは無効であり、Modbus の仕様の範囲内にありません。

解決策:
メールボックスデータの送信用に設定されているデバイスが有効な要求を送信していることを確認してください。

非送信請求メールボックスのメモリ割り当てエラー。| IP アドレス = '<アドレス>'。

エラータイプ:
エラー

考えられる原因:

1. 指定されている IP アドレスを持つデバイスがサーバーに対してメールボックスメッセージを送信しようとしました。その IP を持つデバイスがメールボックスプロジェクトで設定されていないため、メッセージは検証を通過しませんでした。
2. 指定されている IP アドレスを持つデバイスがサーバーに対してメールボックスメッセージを送信しようとしました。デバイスは設定されていますが、そこからデータを要求しているクライアントがないため、メッセージは検証を通過しませんでした。

解決策:
サーバーがメールボックスメッセージを受け入れるためには、指定されているデバイス IP がプロジェクトで設定されている必要があります。デバイスの少なくとも 1 つのデータアイテムがクライアントによって要求されている必要があります。

ソケット接続を作成できません。

エラータイプ:
エラー

考えられる原因:
サーバーは指定されたデバイスとの TCP/IP ソケット接続を確立できませんでしたが、引き続き接続を試みます。

解決策:

1. デバイスがオンラインであることを確認してください。
2. デバイス IP が、サーバーがバインドされている IP のサブネット内であることを確認してください。ほかのネットワークへの接続が可能な有効なゲートウェイが使用可能であることを確認してください。

タグデータベースのインポート用のファイルを開くときにエラーが発生しました。| OS エラー = '<エラー>'。

エラータイプ:
エラー

不良配列。| 配列範囲 = <開始> ～ <終了>。

エラータイプ:
エラー

考えられる原因:
アドレス空間の末端を超えてアドレスの配列が定義されています。

解決策:
デバイスのメモリ空間のサイズを確認し、配列長を適切に再定義してください。

ブロックに不良アドレスがあります。| ブロック範囲 = <アドレス> から <アドレス>。

エラータイプ:
エラー

考えられる原因:
ドライバーは、おそらく範囲外である、PLC 内の存在しない位置を読み取ろうとしました。たとえば、保持レジスタ 40001 ～ 41400 を持つ PLC でアドレス 41405 を要求した場合にこのエラーが生成されます。このエラーが生成された場合、ド

ライバーは指定されたデータブロックをその PLC から再び要求しません。この同じブロックから要求されているその他のアドレスはすべて無効と見なされます。

解決策:
デバイスの範囲内にあるアドレスを要求するようクライアントアプリケーションを更新してください。

関連項目:
エラー処理

ホストの解決に失敗しました。| ホスト名 = '<名前>'。

エラータイプ:
エラー

考えられる原因:
このデバイスは IP アドレスではなく DNS ホスト名を使用するよう設定されています。このホスト名をサーバーによって IP アドレスに解決することはできません。

解決策:
デバイスがオンラインでありドメインに登録されていることを確認してください。

指定された出力コイルブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (コイル)、最大ブロックサイズ = <数値> (コイル)。

エラータイプ:
エラー

指定された入力コイルブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (コイル)、最大ブロックサイズ = <数値> (コイル)。

エラータイプ:
エラー

指定された内部レジスタブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (レジスタ)、最大ブロックサイズ = <数値> (レジスタ)。

エラータイプ:
エラー

指定された保持レジスタブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (レジスタ)、最大ブロックサイズ = <数値> (レジスタ)。

エラータイプ:
エラー

ブロック要求で例外が返されました。| ブロック範囲 = <アドレス> から <アドレス>、例外 = <コード>。

エラータイプ:
警告

考えられる原因:
デバイスが例外コードを返しました。

解決策:
例外コードのドキュメントを参照してください。

関連項目:
Modbus 例外コード

ブロック要求で例外が返されました。| ブロック範囲 = <アドレス> から <アドレス>、関数コード = <コード>、例外 = <コード>。

エラータイプ:
警告

考えられる原因:
デバイスが例外コードを返しました。

解決策:
例外コードのドキュメントを参照してください。

関連項目:
Modbus 例外コード

受信したブロックの長さは不適切です。| ブロック範囲 = <開始> ～ <終了>。

エラータイプ:
警告

考えられる原因:
ドライバーは PLC 内のメモリブロックを読み取ろうとしました。PLC はエラーなしで応答しましたが、要求されたブロックサイズのデータをドライバーに返しませんでした。

解決策:
その範囲のメモリが PLC に存在することを確認してください。

メモリリソース量の低下によりタグインポートが失敗しました。

エラータイプ:
警告

考えられる原因:
ドライバーは変数インポートファイルの処理に必要なメモリを割り当てることができませんでした。

解決策:
不要なアプリケーションをすべて終了してから、再試行してください。

タグのインポート中にファイル例外が発生しました。

エラータイプ:
警告

考えられる原因:
変数インポートファイルを読み取れませんでした。

解決策:
変数インポートファイルを再生成してください。

インポートファイルのレコードの解析でエラーが発生しました。| レコード番号 = <数値>、フィールド = <field>。

エラータイプ:
警告

考えられる原因:
変数インポートファイルの指定されたフィールドが予想より長いか無効なため、解析できませんでした。

解決策:
可能な場合、変数インポートファイルを編集して、問題のあるフィールドを変更してください。

インポートファイルのレコードの説明が切り詰められました。| レコード番号 = <数値>。

エラータイプ:
警告

考えられる原因:
指定されたレコードのタグの説明が長すぎます。

解決策:
ドライバーは必要に応じて説明を切り詰めます。このエラーを防止するには、変数インポートファイルを編集して、説明を短くしてください。

インポートされたタグ名が無効のため変更されました。| タグ名 = '<タグ>'、変更後のタグ名 = '<タグ>'。

エラータイプ:
警告

考えられる原因:
変数インポートファイル内のタグ名に無効な文字が含まれていました。

解決策:
ドライバーは変数インポートファイルに基づいて有効な名前を構築します。このエラーを防止し、名前の一貫性を維持するには、エクスポートされた変数の名前を変更してください。

データ型がサポートされていないため、タグをインポートできませんでした。| タグ名 = '<タグ>'、サポートされていないデータ型 = '<タイプ>'。

エラータイプ:
警告

考えられる原因:
変数インポートファイルで指定されたデータ型は、このドライバーでサポートされている型ではありません。

解決策:
変数インポートファイルで指定されているデータ型を、サポートされているいずれかの型に変更してください。構造体の変数である場合、ファイルを手動で編集して構造体に必要な各タグを定義するか、サーバーで必要なタグを手動で設定してください。

関連項目:
Concept からの変数のエクスポート

アドレスに書き込めません。デバイスは例外を返しました。| アドレス = '<アドレス>'、例外 = <コード>。

エラータイプ:
警告

考えられる原因:
デバイスが例外コードを返しました。

解決策:
例外コードのドキュメントを参照してください。

関連項目:
Modbus 例外コード

イーサネットマネージャが開始されました。

エラータイプ:
情報

イーサネットマネージャが停止しました。

エラータイプ:
情報

タグデータベースをインポートしています。| ソースファイル = '<ファイル名>'。

エラータイプ:
情報

クライアントアプリケーションはシステムタグ _CEGExtension を介して CEG 拡張を変更しました。| 拡張 = '<拡張>'。

エラータイプ:
情報

考えられる原因:
サーバーに接続しているクライアントアプリケーションは指定されたデバイスでの CEG 拡張を 0 (Modbus) または 1 (CEG) に変更しました。

解決策:
このデバイスプロパティは CEG モデルのデバイスのみに適用されます。変更はその他のモデルには影響しません。クライアントアプリケーションがこのプロパティを変更できないようにするには、OPC DA 設定を介してシステムレベルのタグに対するクライアントの書き込み権限を無効にします。

非送信請求通信を開始しています。| プロトコル = '<名前>'、ポート = <数値>。

エラータイプ:
情報

Modbus サーバーデバイス用のメモリが作成されました。| Modbus サーバーデバイス ID = <デバイス>。

エラータイプ:
情報

すべてのチャネルが仮想ネットワークを利用しているか、すべてのデバイスがリモートアドレスを受信しているため、非送信請求通信を停止しています。

エラータイプ:
情報

チャネルは仮想ネットワーク内にあるため、すべてのデバイスがデバイスにつき 1 つのソケットを使用する設定に戻りました。

エラータイプ:
情報

接続されているクライアントでデバイス ID を Modbus クライアントモードからサーバーモードに変更できません。

エラータイプ:
情報

接続されているクライアントでデバイス ID を Modbus サーバーモードからクライアントモードに変更できません。

エラータイプ:
情報

チャネルが仮想ネットワーク内にある場合、Modbus サーバーモードは許可されません。デバイス ID にループバックアドレスまたはローカル IP アドレスが含まれていてはなりません。

エラータイプ:
情報

チャネルが仮想ネットワーク内にある場合、メールボックスモデルは許可されません。

エラータイプ:
情報

Modbus 例外コード

以下のデータは Modbus Application Protocol Specifications ドキュメントからのものです。

コード 10 進 /16 進	名前	意味
01/0x01	ILLEGAL FUNCTION (不正なファンクション)	クエリーで受信したファンクションコードを、サーバーに対する操作として実行することはできません。このファンクションコードは新しいデバイスにだけ適用できるか、選択したユニットに実装されていないことが原因である可能性があります。また、サーバーが、このタイプの要求を処理する状態になっていない可能性もあります (レジスタ値を返す必要があるにもかかわらず、サーバーがそのように設定されていない場合など)。
02/0x02	ILLEGAL DATA ADDRESS (不正なデータアドレス)	クエリーで受信したデータアドレスを、サーバーに対するアドレスとして使用することはできません。具体的には、参照番号と転送長さの組み合わせが無効です。レジスタが 100 個あるコントローラの場合、オフセット 96 と長さ 4 の要求では成功します。オフセット 96 と長さ 5 の要求では例外 02 が生成されます。
03/0x03	ILLEGAL DATA VALUE (不正なデータ値)	クエリーデータフィールドに含まれている値を、サーバーに対する値として使用することはできません。これは、示された長さが正しくないなど、複合型要求の残りの構造体に誤りがあることを示しています。Modbus プロトコルでは個々のレジスタのそれぞれの値の有意性は認識されないので、これはレジスタのストレージにサブミットされたデータアイテムの値がアプリケーションプログラムでの予想の範囲外であることを必ずしも意味しません。
04/0x04	SERVER DEVICE FAILURE (サーバーデバイスの障害)	サーバーが要求された操作を実行しようとしたときに回復不可能なエラーが発生しました。
05/0x05	ACKNOWLEDGE	サーバーは要求を受け入れて処理していますが、処理が完了するまで時間がかかります。クライアントでタイムアウトエラーが発生しないようにするために、この応答が返されます。クライアントは次にプログラム完了ポーリングメッセージを送信して、処理が完了したかどうかを判断します。
06/0x06	SERVER DEVICE BUSY (サーバーデバイスがビジー状態)	サーバーは、時間がかかるプログラムコマンドを処理しています。クライアントは、サーバーによる処理の完了後にメッセージを再送信する必要があります。
07/0x07	NEGATIVE ACKNOWLEDGE (否定応答)	サーバーは、クエリーで受信したプログラムファンクションを実行できません。このコードはファンクションコード 13 または 14 (10 進) を使用したプログラミング要求が成功しなかった場合に返されます。クライアントは、サーバーに対して診断情報またはエラー情報を要求する必要があります。
08/0x08	MEMORY PARITY ERROR (メモリパリティエラー)	サーバーが拡張メモリを読み取ろうとしたときに、メモリ内でパリティエラーが検出されました。クライアントはこの要求を再試行できますが、サーバーデバイス上でサービスが必要になる場合があります。
10/0x0A	GATEWAY PATH UNAVAILABLE (ゲートウェイパスを使用できません)	ゲートウェイが使用されている場合、ゲートウェイが要求を処理するために入力ポートから出力ポートへの内部通信パスを割り当てることができなかったことを示します。これは通常、ゲートウェイの設定に誤りがあるかオーバーロードされていることを意味します。
11/0x0B	GATEWAY TARGET DEVICE FAILED TO RESPOND (ゲートウェイのターゲットデバイスが応答しませんでした)	ゲートウェイが使用されている場合、ターゲットデバイスから応答がなかったことを示します。これは通常、デバイスがネットワーク上に存在しないことを意味します。

注記: このドライバーでは、「サーバー」と「非送信請求」という用語は同じ意味になります。

Modbus Ethernet チャネルのプロパティ

以下はすべての Modbus Ethernet チャネルレベルのプロパティの完全なリストです。


```
{ "common.ALLTYPES_NAME": "MyChannel", "common.ALLTYPES_DESCRIPTION": "",
"servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Modbus TCP/IP Ethernet",
"servermain.CHANNEL_DIAGNOSTICS_CAPTURE": false, "servermain.CHANNEL_UNIQUE_ID":
721923342, "servermain.CHANNEL_ETHERNET_COMMUNICATIONS_NETWORK_ADAPTER_STRING": "",
"servermain.CHANNEL_WRITE_OPTIMIZATIONS_METHOD": 2, "servermain.CHANNEL_WRITE_
OPTIMIZATIONS_DUTY_CYCLE": 10, "servermain.CHANNEL_NON_NORMALIZED_FLOATING_POINT_
HANDLING": 0, "servermain.CHANNEL_COMMUNICATIONS_SERIALIZATION_VIRTUAL_NETWORK": 0,
"servermain.CHANNEL_COMMUNICATIONS_SERIALIZATION_TRANSACTIONS_PER_CYCLE": 1,
"servermain.CHANNEL_COMMUNICATIONS_SERIALIZATION_NETWORK_MODE": 0, "modbus_
ethernet.CHANNEL_USE_ONE_OR_MORE_SOCKETS_PER_DEVICE": 1, "modbus_ethernet.CHANNEL_
MAXIMUM_SOCKETS_PER_DEVICE": 1 }
```


Modbus Ethernet デバイスのプロパティ

以下はすべての Modbus Ethernet デバイスレベルのプロパティの完全なリストです。


```
{ "common.ALLTYPES_NAME": "MyDevice", "common.ALLTYPES_DESCRIPTION": "",
"servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Modbus TCP/IP Ethernet",
"servermain.DEVICE_MODEL": 0, "servermain.DEVICE_UNIQUE_ID": 70949968,
"servermain.DEVICE_CHANNEL_ASSIGNMENT": "MyChannel", "servermain.DEVICE_ID_FORMAT":
0, "servermain.DEVICE_ID_STRING": "<0.0.0.0>.0", "servermain.DEVICE_ID_HEXADECIMAL":
0, "servermain.DEVICE_ID_DECIMAL": 0, "servermain.DEVICE_ID_OCTAL": 0,
"servermain.DEVICE_DATA_COLLECTION": true, "servermain.DEVICE_SIMULATED": false,
"servermain.DEVICE_SCAN_MODE": 0, "servermain.DEVICE_SCAN_MODE_RATE_MS": 1000,
"servermain.DEVICE_SCAN_MODE_PROVIDE_INITIAL_UPDATES_FROM_CACHE": false,
"servermain.DEVICE_CONNECTION_TIMEOUT_SECONDS": 3, "servermain.DEVICE_REQUEST_
TIMEOUT_MILLISECONDS": 1000, "servermain.DEVICE_RETRY_ATTEMPTS": 3,
"servermain.DEVICE_INTER_REQUEST_DELAY_MILLISECONDS": 0, "servermain.DEVICE_AUTO_
DEMOTION_ENABLE_ON_COMMUNICATIONS_FAILURES": false, "servermain.DEVICE_AUTO_DEMOTION_
DEMOTE_AFTER_SUCCESSIVE_TIMEOUTS": 3, "servermain.DEVICE_AUTO_DEMOTION_PERIOD_MS":
10000, "servermain.DEVICE_AUTO_DEMOTION_DISCARD_WRITES": false, "servermain.DEVICE_
TAG_GENERATION_ON_STARTUP": 0, "servermain.DEVICE_TAG_GENERATION_DUPLICATE_HANDLING":
0, "servermain.DEVICE_TAG_GENERATION_GROUP": "", "servermain.DEVICE_TAG_GENERATION_
ALLOW_SUB_GROUPS": true, "modbus_ethernet.DEVICE_VARIABLE_IMPORT_FILE": "normal.txt",
"modbus_ethernet.DEVICE_VARIABLE_IMPORT_INCLUDE_DESCRIPTIONS": 1, "modbus_
ethernet.DEVICE_DEACTIVATE_TAGS_ON_ILLEGAL_ADDRESS": 1, "modbus_ethernet.DEVICE_SUB_
MODEL": 1, "modbus_ethernet.DEVICE_ETHERNET_PORT_NUMBER": 502, "modbus_
ethernet.DEVICE_ETHERNET_IP_PROTOCOL": 1, "modbus_ethernet.DEVICE_ETHERNET_CLOSE_TCP_
SOCKET_ON_TIMEOUT": true, "modbus_ethernet.DEVICE_ZERO_BASED_ADDRESSING": true,
"modbus_ethernet.DEVICE_ZERO_BASED_BIT_ADDRESSING": true, "modbus_ethernet.DEVICE_
HOLDING_REGISTER_BIT_MASK_WRITES": true, "modbus_ethernet.DEVICE_MODBUS_FUNCTION_06":
true, "modbus_ethernet.DEVICE_MODBUS_FUNCTION_05": true, "modbus_ethernet.DEVICE_
MODBUS_BYTE_ORDER": true, "modbus_ethernet.DEVICE_FIRST_WORD_LOW": true, "modbus_
ethernet.DEVICE_FIRST_DWORD_LOW": true, "modbus_ethernet.DEVICE_MODICON_BIT_ORDER":
false, "modbus_ethernet.DEVICE_TREAT_LONGS_AS_DOUBLE_PRECISION_UNSIGNED_DECIMAL":
false, "modbus_ethernet.DEVICE_OUTPUT_COILS": 32, "modbus_ethernet.DEVICE_INPUT_
COILS": 32, "modbus_ethernet.DEVICE_INTERNAL_REGISTERS": 32, "modbus_ethernet.DEVICE_
HOLDING_REGISTERS": 32, "modbus_ethernet.DEVICE_PERFORM_BLOCK_READ_ON_STRINGS": 0 }
```


注記: servermain.DEVICE_MODEL パラメータのデフォルト値は、ジェネリック Modbus モデルです。このデフォルト値を使用しない場合は、このパラメータに適切な値を指定してください。

Modbus Ethernet タグのプロパティ

以下はすべての Modbus Ethernet タグのプロパティの完全なリストです。

{ "common.ALLTYPES_NAME": "MyTag", "common.ALLTYPES_DESCRIPTION": "",
"servermain.TAG_ADDRESS": "400001", "servermain.TAG_DATA_TYPE": 5, "servermain.TAG_
READ_WRITE_ACCESS": 1, "servermain.TAG_SCAN_RATE_MILLISECONDS": 100, "servermain.TAG_
AUTOGENERATED": false, "servermain.TAG_SCALING_TYPE": 0, "servermain.TAG_SCALING_RAW_
LOW": 0, "servermain.TAG_SCALING_RAW_HIGH": 1000, "servermain.TAG_SCALING_SCALED_
DATA_TYPE": 9, "servermain.TAG_SCALING_SCALED_LOW": 0, "servermain.TAG_SCALING_
SCALED_HIGH": 1000, "servermain.TAG_SCALING_CLAMP_LOW": false, "servermain.TAG_
SCALING_CLAMP_HIGH": false, "servermain.TAG_SCALING_NEGATE_VALUE": false,
"servermain.TAG_SCALING_UNITS": "" }

索引

1

10 進アドレス指定 43
16 進アドレス指定 43

5

5 桁のアドレス指定 44

6

6 桁のアドレス指定 44

A

Applicom 28
Applicom のアドレス指定 33

B

BCD 31
BOOL 29
Boolean 31
BYTE 29

C

CEG 28
CEG のアドレス指定 43
CEG 拡張 22
CEGExtension 32
CSV 29

D

DINT 29
Double 31
DWord 31

F

Float 31
Fluenta 6
Fluenta FGM 28
Fluenta のアドレス指定 43

H

HoldingRegisterBlockSize 32

I

ID 13
InputCoilBlockSize 32
Instromet 6, 28
Instromet のアドレス指定 43
INT 29
InternalRegisterBlockSize 32
IP プロトコル 12, 20

L

LBCD 31
Long 31
Long を 10 進数として扱う 23

M

Modbus イーサネット通信の最適化 29
Modbus クライアント 6
Modbus クライアントとModbus サーバーに関する考慮事項 19
Modbus サーバーデバイス用のメモリが作成されました。| Modbus サーバーデバイス ID = <デバイス>。 53
Modbus のアドレス指定 44
Modbus バイトオーダー 22
Modbus メールボックス 44
Modbus 関数 05 22
Modbus 関数 06 21
Modbus 非送信請求 6
Modbus 例外コード 55
Modicon ビットオーダー 22

O

OPC 品質を不良に設定 18
OutputCoilBlockSize 32

R

REAL 29
Roxar 7
Roxar RFM 28
Roxar のアドレス指定 47

S

Short 31
String 31
STRING 29

T

TIME 29
TSX Premium 40
TSX Quantum 36

U

UDINT 29
UINT 29

W

Winsock 通信を開始できませんでした。 48
Word 31
WORD 29

あ

アドレス 29
アドレスに書き込めません。デバイスは例外を返しました。| アドレス = '<アドレス>'、例外 = <コード>。 52
アドレスの説明 32

い

イーサネット 11, 20
イーサネットから Modbus Plus へのブリッジ 6
イーサネットマネージャが開始されました。 53
イーサネットマネージャが停止しました。 53
イーサネット設定 8
イベントログメッセージ 48
インポートされたタグ名が無効のため変更されました。| タグ名 = '<タグ>'、変更後のタグ名 = '<タグ>'。 52
インポートファイルのレコードの解析でエラーが発生しました。| レコード番号 = <数値>、フィールド = <field>。 51
インポートファイルのレコードの説明が切り詰められました。| レコード番号 = <数値>。 52

え

エラー時に格下げ 15
エラー処理 19

か

カスタムアプリケーションからのインポート 29
カスタムタグ 29

き

キャッシュからの初回更新 14

く

クライアントアプリケーションはシステムタグ _CEGExtension を介して CEG 拡張を変更しました。| 拡張 = '<拡張>'。 53
グローバル設定 10

こ

コイルのステータスを読み取り 33
コメント 29

さ

サービス 43
サイクルあたりのトランザクション数 10

サブグループを許可 17
サポートされる 6

し

ジェネリック Modbus のアドレス指定 33
システム 43
システムタグ 32
シミュレーション 13

す

スキャンしない、要求ポールのみ 14
スキャンモード 14
すべてのタグのすべての値を書き込み 9
すべてのタグの最新の値のみを書き込み 9
すべてのチャネルが仮想ネットワークを利用しているか、すべてのデバイスがリモートアドレスを受信しているため、非送信請求通信を停止しています。 53

せ

ゼロで置換 9
ゼロベースアドレス指定 21
ゼロベースのビットアドレス指定 21

そ

ソケット使用 11
ソケット接続を作成できません。 49
ソケット利用 11

た

タイミング 14
タイムアウト時にソケットを閉じる 20
タイムアウト前の試行回数 15
タグデータベースのインポート用のファイルを開くときにエラーが発生しました。| OS エラー = '<エラー>'。 49
タグデータベースをインポートしています。| ソースファイル = '<ファイル名>'。 53
タグに指定のスキャン速度を適用 14
タグのインポート中にファイル例外が発生しました。 51
タグ数 8
タグ生成 15

ち

チャネルが仮想ネットワーク内にある場合、Modbus サーバーモードは許可されません。デバイス ID にループバックアドレスまたはローカル IP アドレスが含まれていてはなりません。 54
チャネルが仮想ネットワーク内にある場合、メールボックスモデルは許可されません。 54
チャネルのプロパティ - イーサネット通信 8
チャネルのプロパティ - 一般 7
チャネルのプロパティ - 書き込み最適化 8
チャネルのプロパティ - 詳細 9
チャネルのプロパティ - 通信シリアル化 10
チャネルは仮想ネットワーク内にあるため、すべてのデバイスがデバイスにつき 1 つのソケットを使用する設定に戻りました。 53
チャネルレベルの設定 10
チャネル割り当て 13

て

データアクセス 21
データエンコーディング 22
データコレクション 13
データ型がサポートされていないため、タグをインポートできませんでした。| タグ名 = '<タグ>'、サポートされていないデータ型 = '<タイプ>'。 52
データ型の説明 31
デバイスあたりの最大ソケット数 12
デバイスのプロパティ - タイミング 14
デバイスのプロパティ - タグ生成 15
デバイスのプロパティ - 自動格下げ 15
デバイスのプロパティ - 冗長 25
デバイス間遅延 10
デバイス起動時 16
デューティサイクル 9

と

ドライバー 13
ドライバーのシステムタグのアドレス指定 32

ね

ネットワーク 1 - ネットワーク 500 10
ネットワークアダプタ 8
ネットワークモード 10

は

バイト交換のサフィックス 36, 41

ふ

ファンクションコードの説明 33
ブロックサイズ 23
ブロックに不良アドレスがあります。| ブロック範囲 = <アドレス> から <アドレス>。 49
ブロック要求で例外が返されました。| ブロック範囲 = <アドレス> から <アドレス>、関数コード = <コード>、例外 = <コード>。 51
ブロック要求で例外が返されました。| ブロック範囲 = <アドレス> から <アドレス>、例外 = <コード>。 50
プロパティ変更時 16

へ

ヘルプの目次 5

ほ

ポート 12, 20, 32
ホストの解決に失敗しました。| ホスト名 = '<名前>'。 50

め

メールボックス 6, 28
メールボックスクライアントの権限 22
メールボックスのアドレス指定 43
メールボックスモード 46
メモリリソース量の低下によりタグインポートが失敗しました。 51

も

モデル 6, 13

ゆ

ユーザー 43

れ

レコード 29
レジスタへのマスク書き込み 33

漢字

一般 12
仮想ネットワーク 10
概要 6
格下げまでのタイムアウト回数 15
格下げ期間 15
格下げ時に要求を破棄 15
最初の DWord を下位とする 22
最初の Word を下位とする 22
最適化方法 9
作成 17
削除 16
指定された出力コイルブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (コイル)、最大ブロックサイズ = <数値> (コイル)。 50
指定された内部レジスタブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (レジスタ)、最大ブロックサイズ = <数値> (レジスタ)。 50
指定された入力コイルブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (コイル)、最大ブロックサイズ = <数値> (コイル)。 50
指定された保持レジスタブロックサイズは最大ブロックサイズを超えています。| 指定されたブロックサイズ = <数値> (レジスタ)、最大ブロックサイズ = <数値> (レジスタ)。 50
自動タグデータベース生成 29
自動格下げ 15
識別 7
受信したブロックの長さは不適切です。| ブロック範囲 = <開始> ～ <終了>。 51
受信した要求は非送信請求メールボックスでサポートされていません。| IP アドレス = '<アドレス>'。 48
重複タグ 16
出力 43
出力コイル 24, 33, 36, 40, 44
書き込み専用アクセス 46
上書き 16
冗長 25
親グループ 17
診断 8
生成 16
接続されているクライアントでデバイス ID を Modbus クライアントモードからサーバーモードに変更できません。 53
接続されているクライアントでデバイス ID を Modbus サーバーモードからクライアントモードに変更できません。 54
接続のタイムアウト 14
設定 7, 20
設定値 29

説明 13
説明を含める 17
単一コイルを適用 33
単一レジスタをプリセット 33
単精度実数 43, 47
短整数 43, 47
長整数 43
通信タイムアウト 14, 18
内部タグ 32
内部レジスタ 24, 34, 37, 44
内部レジスタを読み取り 33
入力コイル 24, 33, 37, 44
入力ステータスを読み取り 33
配列 44
配列のサポート 33-36, 40-41, 47
非 Boolean タグの最新の値のみを書き込み 9
非正規化浮動小数点処理 9
非送信請求 17
非送信請求メールボックスのメモリ割り当てエラー。| IP アドレス = '<アドレス>'。 49
非送信請求通信を開始しています。| プロトコル = '<名前>'、ポート = <数値>。 53
非送信請求通信を開始できませんでした。 48
不正なアドレスでタグを無効化 20
不良配列。| 配列範囲 = <開始> ～ <終了>。 49
負荷分散 10
複数コイルを適用 33
複数レジスタをプリセット 33
文字列のサポート 36, 41, 46
文字列のブロック読み取り 24
変数 29
変数のインポート設定 17
保持レジスタ 24, 35, 38, 40, 45
保持レジスタのビットマスク 21
保持レジスタを読み取り 33
未修正 9
未定義のデバイスに対して非送信請求メールボックスアクセスが行われました。ソケットを閉じています。| IP アドレス = '<アドレス>'。 48
名前 13
優先順位 10
要求のタイムアウト 14
累積 43
列挙 28