
RS-232C Monitor and Analyzer Series

プロトコル アナライザー for Modbus ASCII

Model: AKM-RSM-FM0

ユーザーズ マニュアル

Version 1.3.0.0

目次

ご使用になる前に.....	- 3 -
はじめに.....	- 4 -
詳細分析.....	- 5 -
概要分析.....	- 6 -
注意事項.....	- 7 -
操作条件.....	- 8 -
起動方法.....	- 9 -
操作方法.....	- 11 -
モニター結果ファイルを開く.....	- 11 -
オプション設定を行う.....	- 11 -
(1) サーバー オプション.....	- 11 -
(2) プレフィックス オプション.....	- 12 -
(3) その他 オプション.....	- 13 -
(4) 初期設定ボタン.....	- 13 -
プロトコル アナライザーを実行する.....	- 14 -
その他の機能.....	- 17 -
終了方法.....	- 18 -
分析結果ファイルについて.....	- 19 -
ヘッダーについて.....	- 20 -
分析結果について.....	- 21 -
(1) 分析結果ファイルの書式の概要.....	- 21 -
(2) 分析結果 [×異常] の例.....	- 22 -
(3) 分析結果 [△警告] の例.....	- 23 -
(4) 分析結果 [!!例外] の例.....	- 24 -
付録A: 分析結果ファイルでのメッセージ一覧.....	- 25 -
エラー メッセージ.....	- 25 -
警告メッセージ.....	- 27 -
付録B: 各ファンクションでの分析結果出力例.....	- 28 -
付録C: 対訳.....	- 41 -
ファンクション名.....	- 41 -
用語.....	- 42 -

ご使用になる前に

【登録商標】

- Modbus は Schneider Electric, Inc. の登録商標です。
- CANopen は、CAN in Automation の登録商標です。
- Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標です。
- その他、このマニュアルに記載されている会社名、商品名は、各社の登録商標または商標です。

ご注意

- ・このマニュアルの内容の一部または全部を無断転載することは禁止されています。
- ・このマニュアルの内容に関しては、将来予告なしに変更することがあります。

参考文献

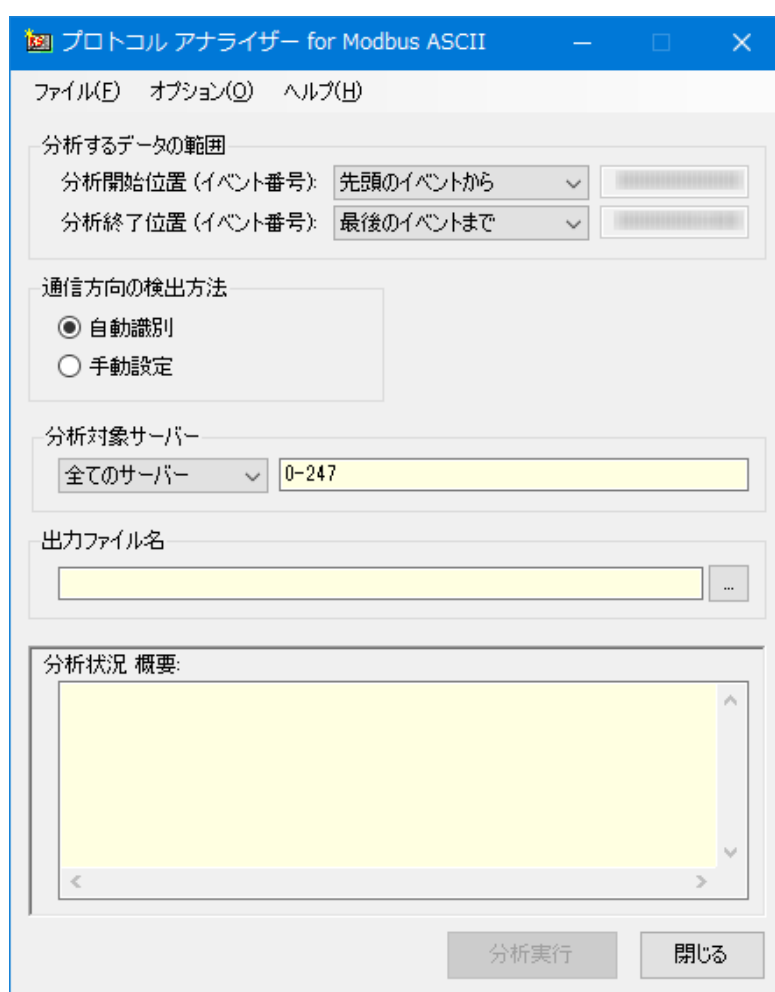
MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3

はじめに

プロトコル アナライザー for Modbus ASCII (Model: AKM-RSM-FM0) (以下、プロトコル アナライザー) は、RS-232C Monitor and Analyzer (Model: AKM-RSM-100) (以下、AKM-RSM-100) のモニター結果のデータ列を分析し、Modbus ASCII のデータとしてテキストファイルへ出力するツールです。

分析内容は、**Modbus の V1.1b3** に準拠しています。

モニター結果のデータ列を分析することで、Modbus ASCII の通信の流れや異常などを視覚的に確認することができます。



画面イメージ

なお、プロトコル アナライザーは、AKM-RSM-100 のアドオン機能です。
AKM-RSM-100 の Professional Edition をご利用いただけます。
(Basic Edition および Standard Edition ではご利用いただけません。)

プロトコル アナライザーでは、「詳細分析」と「概要分析」の2種類の分析方法があります。

■ 詳細分析

データ列を全て詳細に分析して、その内容を出力します。

加えて、クライアントからの「要求」と、サーバーからの「応答」との関係も確認し、異変を見つけた場合には、警告情報を出力します。

```

      ⋮
---- [O正常] [要求] (0x000000000001) (xxxx/xx/xx xx:xx:xx) ----->
:010100000001FD<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し

先頭アドレス: 0x0001, 読み出し数: 1 (0x0001)
----->

<--- [O正常] [応答] (0x000000000013) (xxxx/xx/xx xx:xx:xx) -----
:01010105F8<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し

出力バイト数: 1 (0x01)
ステータス : ----- 下記 -----
0x0001:0N
(要求イベント番号: 0x000000000001)
<-----

---- [O正常] [要求] (0x000000000021) (xxxx/xx/xx xx:xx:xx) ----->
:010100130013D8<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し

先頭アドレス: 0x0014, 読み出し数: 19 (0x0013)
----->

<--- [O正常] [応答] (0x000000000032) (xxxx/xx/xx xx:xx:xx) -----
:010103CD6B05BE<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し

出力バイト数: 3 (0x03)
ステータス : ----- 下記 -----
0x0014:0N , 0x0015:0FF, 0x0016:0N , 0x0017:0N , 0x0018:0FF, 0x0019:0FF, 0x001A:0N , 0x001B:0N ,
0x001C:0N , 0x001D:0N , 0x001E:0FF, 0x001F:0N , 0x0020:0FF, 0x0021:0N , 0x0022:0N , 0x0023:0FF,
0x0024:0N , 0x0025:0FF, 0x0026:0N
(要求イベント番号: 0x000000000021)
<-----

      ⋮

```

ファイル出力イメージ (詳細分析)

■ 概要分析

サーバーアドレスとファンクション番号だけを分析して出力します。

```

      ⋮
---- [○正常] [要求] (0x000000000001) (xxxx/xx/xx xx:xx:xx) ----->
:010100000001FD<cr><lf>
サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し
----->

<--- [○正常] [応答] (0x000000000013) (xxxx/xx/xx xx:xx:xx) -----
:01010105F8<cr><lf>
サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し
<-----

---- [○正常] [要求] (0x000000000021) (xxxx/xx/xx xx:xx:xx) ----->
:010100130013D8<cr><lf>
サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し
----->

<--- [○正常] [応答] (0x000000000032) (xxxx/xx/xx xx:xx:xx) -----
:010103CD6B05BE<cr><lf>
サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し
<-----

---- [○正常] [要求] (0x000000000044) (xxxx/xx/xx xx:xx:xx) ----->
:0101000007D027<cr><lf>
サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し
----->
      ⋮

```

ファイル出カイメージ (概要分析)

注意事項

ご使用にあたっては、あらかじめ次の事項にご注意ください。

- (1) プロトコル アナライザーは、AKM-RSM-100 の Professional Edition の [ツール] メニューから起動することができます。
(Basic Edition および Standard Edition ではご利用いただけません。)
- (2) プロトコル アナライザーは、「Modbus Application Protocol Specification V1.1b3」に基づいて分析を行います。
他のバージョンの Modbus のデータ列では、正しい分析結果が得られない場合があります。
- (3) モニター結果データ上、クライアントからの [要求] と、サーバーからの [応答] が交錯している場合、正しい分析結果が得られないことがあります。
(特に、2 線の RS-485 で、ポート 1 またはポート 2 のどちらかのみ、[要求] と [応答] が両方入っている場合)
- (4) 分析するデータ範囲の中に [ストップ マーク] があった場合、その [ストップ マーク] まで分析は止まります。
- (5) プロトコル アナライザーは、.NetFramework 4 に基づくアプリケーション ソフトウェアです。
AKM-RSM-100 の Version 4 シリーズをご使用のお客様は、PC に .NetFramework 4 が適用されているか確認してください。
.NetFramework 4 が適用されていない場合には、プロトコル アナライザーを起動する前にあらかじめ適用しておいてください。
- (6) モニター結果データのボリュームによっては、膨大な分析結果ファイルが出力される可能性があります。
出力先は、NTFS のストレージを指定するようにしてください。

[分析するデータの範囲] を使って、分析する範囲を絞ることで、余分な情報を出力しないようにすることも、効果的な使い方です。

操作条件

- (1) プロトコル アナライザーは、後述の「起動方法」に従って AKM-RSM-100 の [ツール] メニューから起動してください。
- (2) プロトコル アナライザーは、AKM-RSM-100 から、複数起動することができますが、起動数に応じて PC (Windows) のメモリ リソースを占有していきますので、起動しすぎにご注意ください。
- (3) 起動時、次の内容は AKM-RSM-100 の設定が引き継がれます。
 - ・ 先頭のイベント番号
 - ・ 現在のカーソル位置
 - ・ 最後のイベント番号
 - ・ 年月日表現
 - ・ ツール チップの有無
- (4) AKM-RSM-100 で、Professional Edition の**試用版 ID**をお使いの場合、**10 プロトコルデータ列まで分析した時点で、処理が停止**します。
- (5) このマニュアルでは、各種手順や画面表示について、Windows 10 をベースに記述しておりますが、他の OS の場合でも概ね同様の手順および画面です。

起動方法

プロトコル アナライザは、AKM-RSM-100 のアドオンです。
AKM-RSM-100 の [ツール] メニューから起動しますが、あらかじめメニュー登録する必要があります。

プロトコル アナライザをインストールした時、自動的に「アドオン メニュー 制御」が起動され、プロトコル アナライザを AKM-RSM-100 のアドオン メニューにマージしようとしていますが、下記の要因により、マージできなかった場合には、AKM-RSM-100 の [ツール] - [アドオン管理] で、プロトコル アナライザを手動登録してください。

- ・ AKM-RSM-100 がインストールされていない場合
- ・ すでに 10 種類のアドオンが登録されている場合

※ 同名のアドオンが登録されていた場合は、データを上書きで更新します。

手動での登録内容は、下記の通りです。

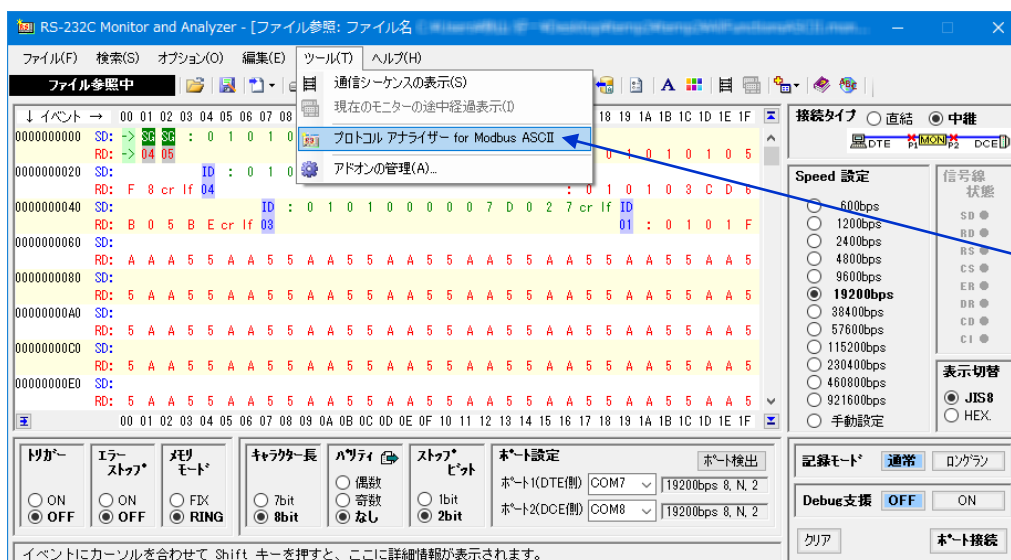
- ・ [メニュー名] : プロトコル アナライザ for Modbus ASCII
- ・ [実行ファイル指定] : [インストール先フォルダ]¥AKM-RSM-FMO.exe
- ・ [コマンド ライン引数] : /MULTI <%datafile%>.aod
- ・ [データ ファイル名] : AKM-RSM-FMO_<%date%>

※ インストール先フォルダの規定値は、通常、下記です。


[OS のドライブ]:¥Program Files (x86)¥RS-232C Monitor and Analyzer¥Add-on¥AKM-RSM-FMO¥

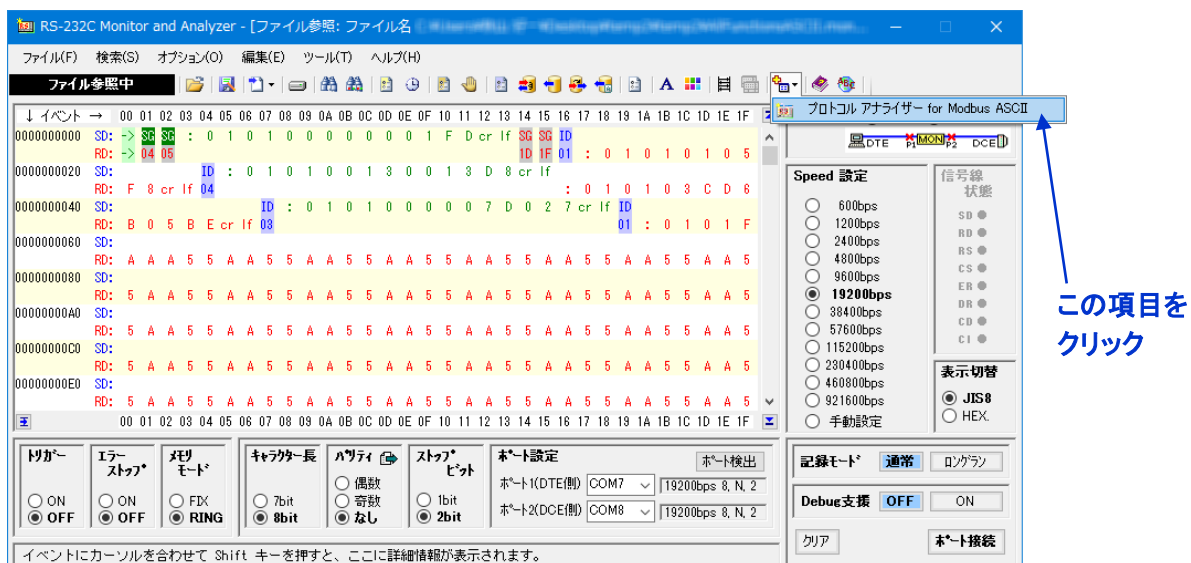
以下、アドオン メニューへの登録が完了しているものとして、説明します。
プロトコル アナライザは、AKM-RSM-100 から、次のように起動します。

起動方法：メニューバーの [ツール] - [プロトコル アナライザ for Modbus ASCII] をクリック



このメニューを
クリック

または、ツールバーの  アイコンをドロップダウンして、[プロトコル アナライザ for Modbus ASCII] をクリック



なお、プロトコル アナライザが使えるのは、AKM-RSM-100 のステータスが下記の場合です。

モニター結果参照中	ファイル参照中
モニター結果参照中	ファイル参照中

操作手法

■ モニター結果ファイルを開く

もし、AKM-RSM-100 のステータスが次のステータスだった場合には、あらかじめ AKM-RSM-100 でモニター結果ファイルを開く必要があります。

表示なし、未接続

表示なし、接続中

AKM-RSM-100 のメニューバーの [ファイル] - [保存されているモニター結果を開く] をクリックして、モニター結果ファイルを開いてください。

その後、プロトコル アナライザーを起動してください。

■ オプション設定を行う

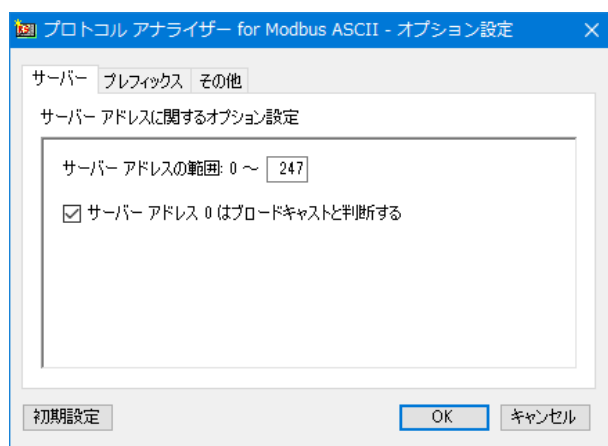
最初に、プロトコル アナライザーの画面上でオプション設定を行います。

プロトコル アナライザーの メニューから、[オプション] - [オプション] で、[オプション設定] 画面を開きます。

全ての設定が完了したら、[OK] ボタンで [オプション設定] 画面を閉じてください。

(1) サーバー オプション

ここでは、サーバー アドレスに関する設定を行います。



[サーバー アドレスの範囲]: (初期値: 0-247)

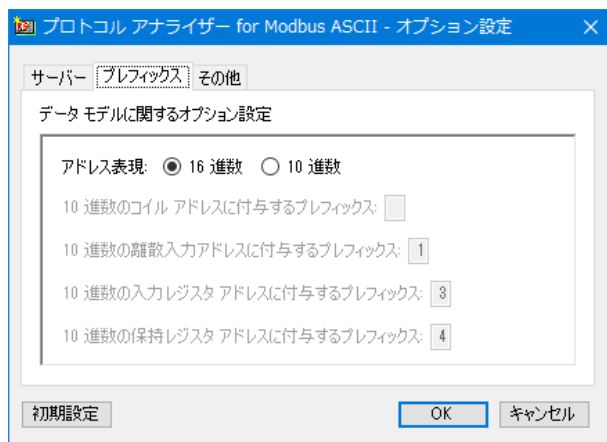
使用しているシステムのサーバー アドレスの範囲を設定します。
最大 255 まで設定できます。

[サーバー アドレス 0 はブロードキャストと判断する]: (初期値: チェックあり)

チェックすることにより、クライアントからサーバー アドレス 0 の [要求] が送信された場合、その [要求] はブロードキャストであると判断して分析処理が実行されます。
具体的には、サーバーからのレスポンスが返らなくても、[警告] を出力しません。
サーバー アドレス 0 を他のサーバーと同様に扱う場合には、このチェックを外してください。

(2) プレフィックス オプション

ここでは、データ モデルに関する設定を行います。



[アドレス表現]: (初期値: 16 進数)

分析結果ファイルにおいて、下記の各アドレスを 16 進数で出力するか、10 進数で出力するか、を選択します。

- (a) コイル アドレス
- (b) 離散入力アドレス
- (c) 入力レジスタ アドレス
- (d) 保持レジスタ アドレス

[10 進数のコイル アドレスに付与するプレフィックス]: (初期値: "" プレフィックスなし)

コイル アドレスを 10 進数で出力する場合、アドレスの先頭に付与するプレフィックスを設定します。
"" (プレフィックスなし), および "0" から "9" までの数字が設定できます。

[10 進数の離散入力アドレスに付与するプレフィックス]: (初期値: "1")

離散入力アドレスを 10 進数で出力する場合、アドレスの先頭に付与するプレフィックスを設定します。
"" (プレフィックスなし), または "0" から "9" までの数字が設定できます。

[10 進数の入力レジスタ アドレスに付与するプレフィックス]: (初期値: "3")

入力レジスタ アドレスを 10 進数で出力する場合、アドレスの先頭に付与するプレフィックスを設定します。
"" (プレフィックスなし), または "0" から "9" までの数字が設定できます。

[10 進数の保持レジスタ アドレスに付与するプレフィックス]: (初期値: "4")

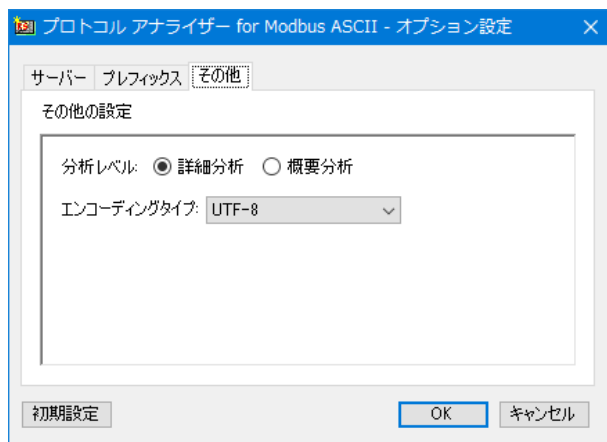
保持レジスタ アドレスを 10 進数で出力する場合、アドレスの先頭に付与するプレフィックスを設定します。
"" (プレフィックスなし), または "0" から "9" までの数字が設定できます。

(注) 各アドレスは、10 進数で 1 から 65536 まで存在しますが、プレフィックスを付与した場合、P0001 から P65536 (P: プレフィックス) と表現されます。

例) プレフィックスが "3" の場合: 30001~365536

(3) その他 オプション

ここでは、その他の設定を行います。



[分析レベル]: (初期値: 詳細分析)

分析レベルを選択します。

[詳細分析]: モニター結果データの全てのデータを詳細に分析します。

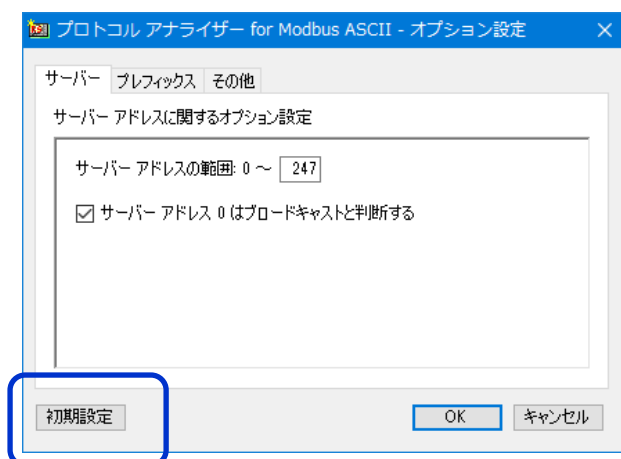
[概要分析]: モニター結果データの内、サーバー アドレスとファンクション番号（およびサブ ファンクション番号、MEI タイプ）だけを分析します。

[エンコーディングタイプ]: (初期値: UTF-8)

分析結果ファイルのエンコーディングタイプを選択します。

(4) 初期設定ボタン

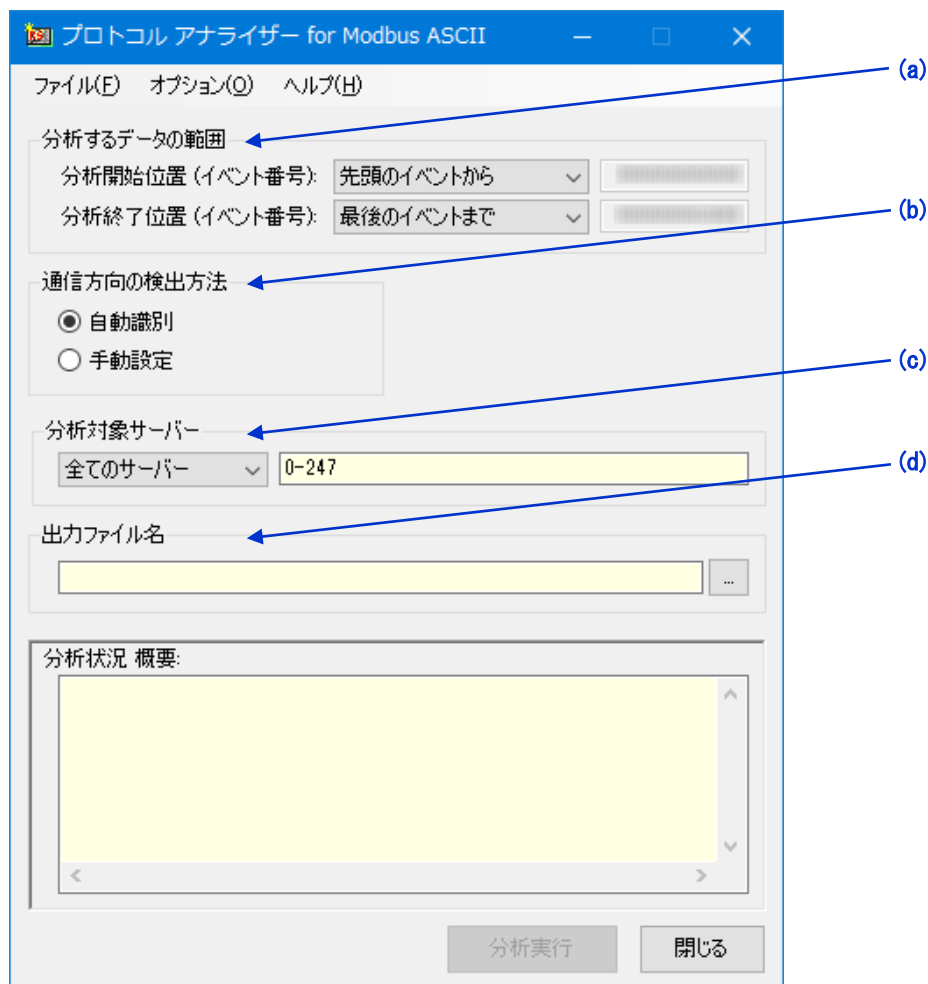
このボタンをクリックすることにより、全てのオプション設定が初期値に戻ります。



(注) このボタンをクリックしても、その後、[OK] ボタンをクリックしないと、変更は反映されません。

■ プロトコル アナライザを実行する

各種オプションを設定後、プロトコル アナライザの画面の設定を行って、プロトコル アナライザを実行します。



(a) [分析するデータの範囲]:

[分析開始位置]: (初期値: 先頭のイベントから)

分析を開始するモニター結果データのイベント番号を選択 (または設定) します。
下記の 3 種類から選択してください。

- ・ [先頭のイベントから]
- ・ [現在のカーソル位置から]
- ・ [手動設定] (イベント番号を手入力で設定します)

[分析終了位置]: (初期値: 最後のイベントまで)

分析を終了するモニター結果データのイベント番号を選択 (または設定) します。
下記の 3 種類から選択してください。

- ・ [最後のイベントまで]
- ・ [現在のカーソル位置まで]
- ・ [手動設定] (イベント番号を手入力で設定します)

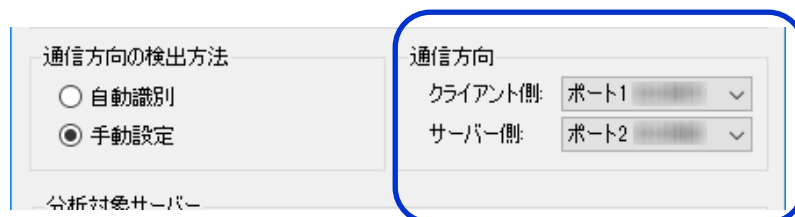
(b) [通信方向の検出方法]: (初期値: 自動識別)

通信方向の検出方法を選択します。

[自動識別] は、分析実行時に、引き渡されたモニター結果データからクライアント側とサーバー側を自動的に識別します。

[自動識別] で、クライアント側とサーバー側の識別ができない場合、[手動設定] で、クライアント側とサーバー側のポート番号を指定してください。

[手動設定] を選択した場合、下図のような [通信方向] 設定が現れますので、ここで通信方向を手動設定します。



クライアント側とは、Modbus の Function を送信する側のポートことです。

サーバー側とは、クライアント側からのファンクションに対してレスポンスを送信する側のポートのことです。

(c) [分析対象サーバー]: (初期値: 全てのサーバー 0-247)

分析結果ファイルに出力する対象のサーバー アドレスを設定します。

- ・ [全てのサーバー]: [オプション] で設定した [サーバー アドレスの範囲] になります。
- ・ [サーバーを選択]: [オプション] で設定した [サーバー アドレスの範囲] の内、分析結果ファイルに出力する対象のサーバー アドレスを選択できます。
[サーバーを選択] を選択すると下図のような [分析対象 サーバー アドレス設定] 画面が表示されますので、分析対象とするサーバーをチェックして、[OK] ボタンをクリックしてください。



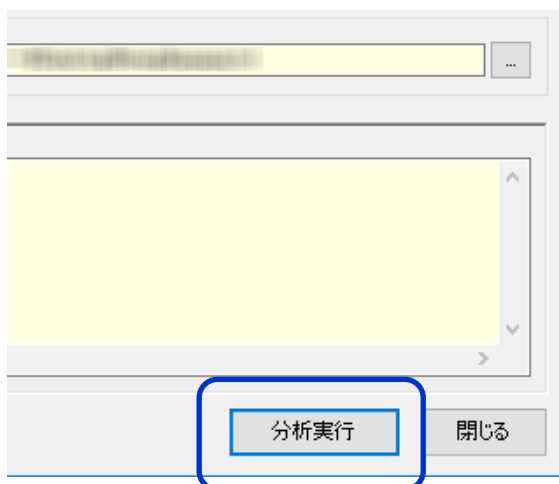
(d) [出力ファイル名]: (初期値: "" (Null))

分析結果ファイルの出力先とファイル名を設定します。

[テキスト ボックス] または [...] ボタンをクリックして、出力先とファイル名を設定してください。

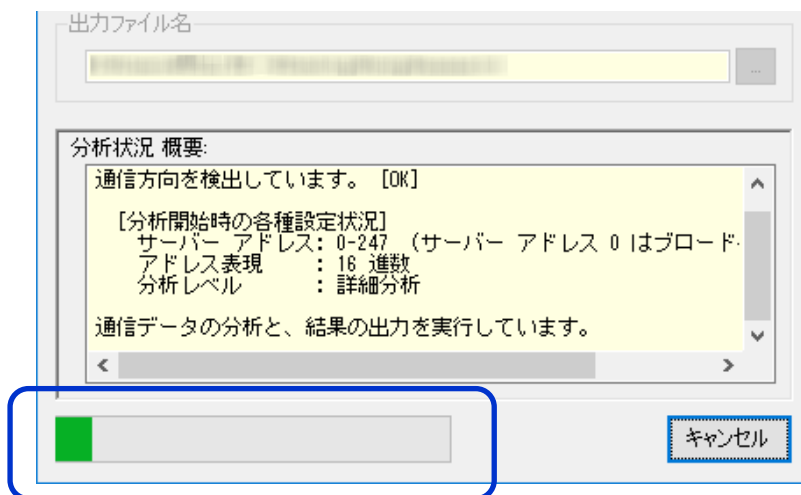
(注) [テキスト ボックス] への直接入力はできません。

[出力ファイル名] を設定すると、[分析実行] ボタンが有効になります。



[分析実行] ボタンをクリックすることで、分析が開始されます。

分析実行中は、下図のような [プログレス バー] で進捗状況を表示しますので、分析が完了するまでお待ちください。



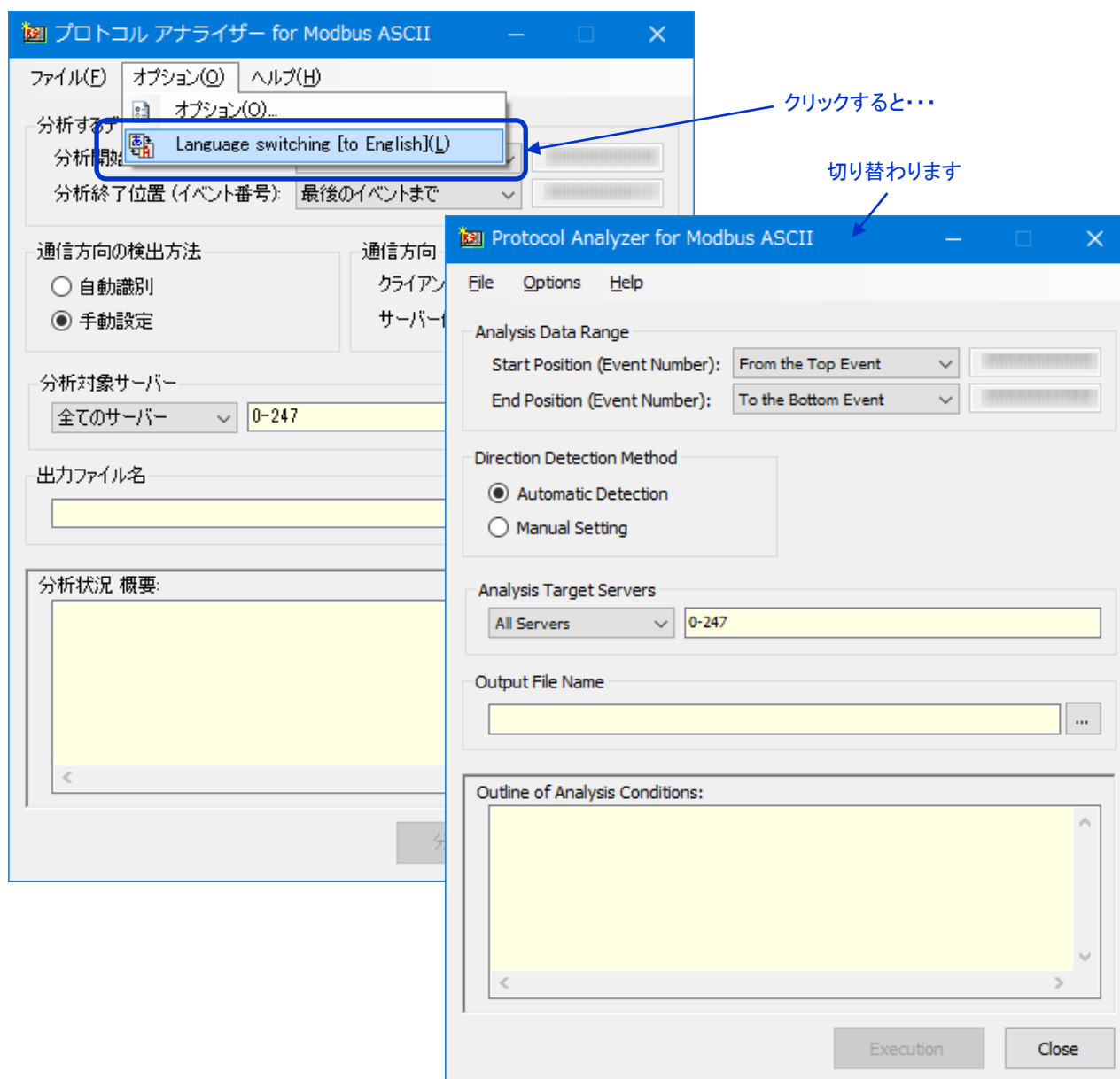
分析実行中、[キャンセル] ボタンをクリックすることで、分析を中止することができます。

その他の機能

プロトコル アナライザーは、日本語と英語の両方に対応しており、起動元の AKM-RSM-100 の表示言語と同じ言語で立ち上がります。

分析結果ファイルは、表示言語と同じ言語で出力されます。

なお、プロトコル アナライザーの [オプション] メニューから表示言語を変更することができます。海外の技術者様へ分析結果ファイルを提供する場合などに、ご活用ください。



※ 英語版に切り替えた後、同じ操作で日本語版に戻ります。

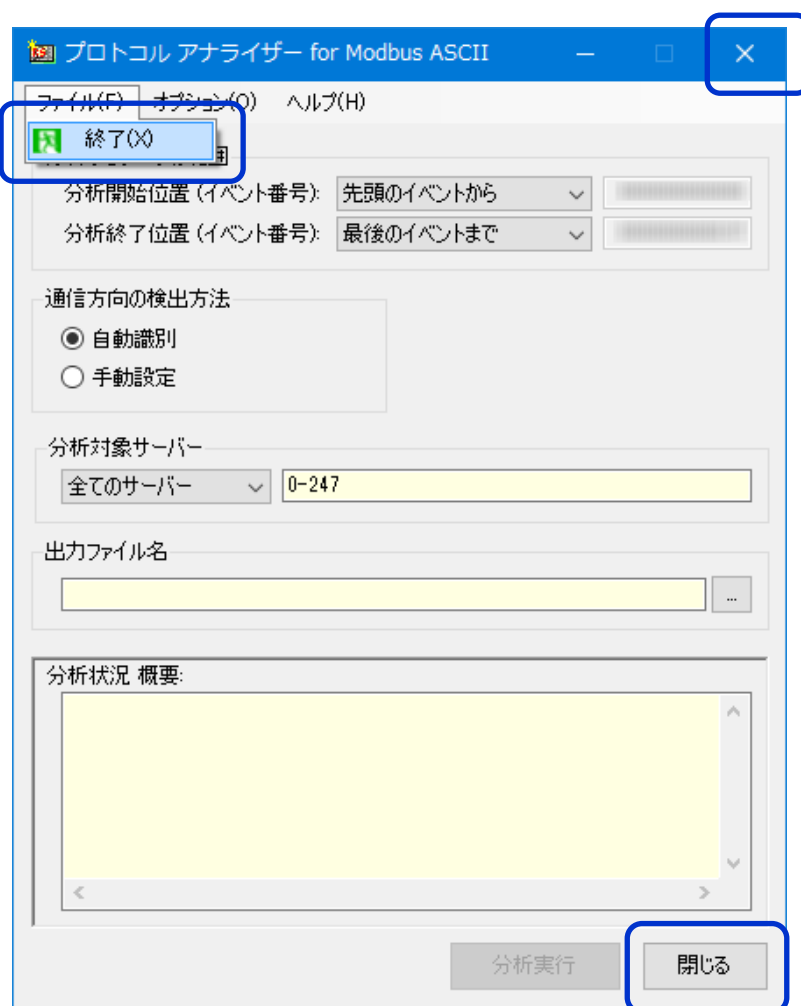
終了方法

プロトコル アナライザーは、次のいずれかの方法で終了してください。

終了方法 1: 画面右下の [閉じる] をクリック

終了方法 2: メニューバーの [ファイル] - [終了] をクリック

終了方法 3: タイトルバーの [×] をクリック



分析結果ファイルについて

分析結果ファイルは、テキストファイルです。

分析結果ファイルでは、左側にクライアント、右側にサーバー、があるものと想定して出力しています。

下記は、ファンクション 7 の分析結果ファイルの例です。

```

RS-232C Monitor and Analyzer (Model: AKM-RSM-100)
プロトコル アナライザー for Modbus ASCII (Ver. x. x. x. x) [based on 1.1b3] (Dic. Rev. xxxx/xx/xx)

ソース ファイル名: xxxxxxxx.mon

[分析開始時の各種設定状況]
サーバー アドレス: 0-247 (サーバー アドレス 0 はブロードキャスト: はい)
アドレス表現      : 16 進数
分析レベル        : 詳細分析

分析対象サーバー : 0-247

分析データ範囲   : 0x000000000000-0x000000000019

==== 分析開始 (xxxx/xx/xx xx:xx:xx) =====
---- [○正常] [要求] (0x000000000003) (xxxx/xx/xx xx:xx:xx) ----->
:1007E9<cr><lf>

サーバー アドレス: 16 (0x10), ファンクション番号: 7 (0x07)
ファンクション名 : 例外ステータスの読み出し
----->

<---- [○正常] [応答] (0x00000000000F) (xxxx/xx/xx xx:xx:xx) -----
:10076D7C<cr><lf>

サーバー アドレス: 16 (0x10), ファンクション番号: 7 (0x07)
ファンクション名 : 例外ステータスの読み出し

ステータス: OFF ON  ON  OFF ON  ON  OFF ON  (109 (0x6D))
(要求イベント番号: 0x000000000003)
<-----

処理は正常終了しました。

==== 分析終了 (xxxx/xx/xx xx:xx:xx) =====

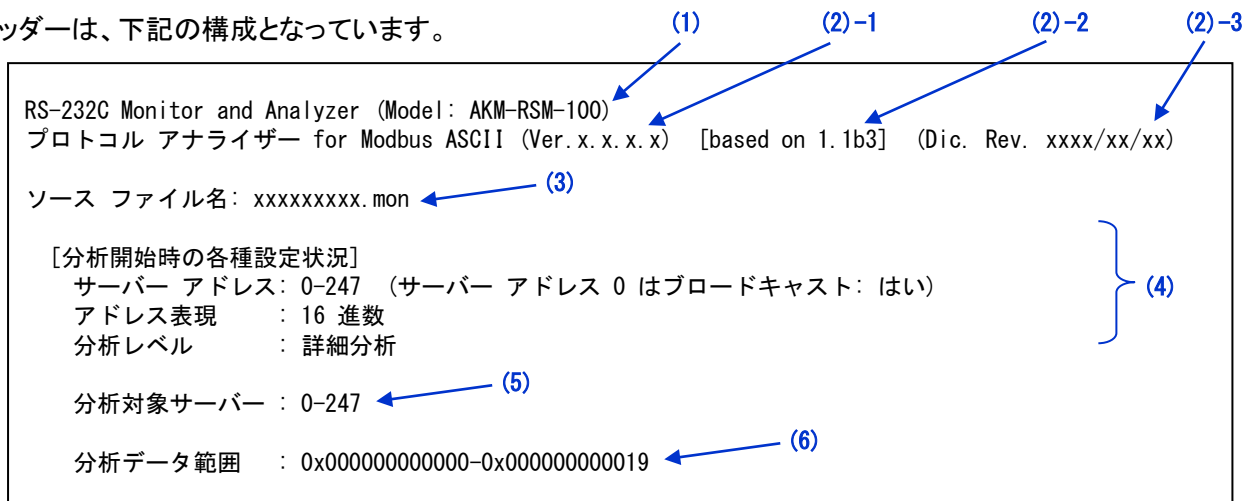
```

(a) は、ヘッダーで、分析結果ファイルを後で参照する際に役立つ情報です。

(b) が分析結果です。(モニター結果データのボリュームにより、この部分が膨大なボリュームになる可能性があります。)

■ ヘッダーについて

ヘッダーは、下記の構成となっています。

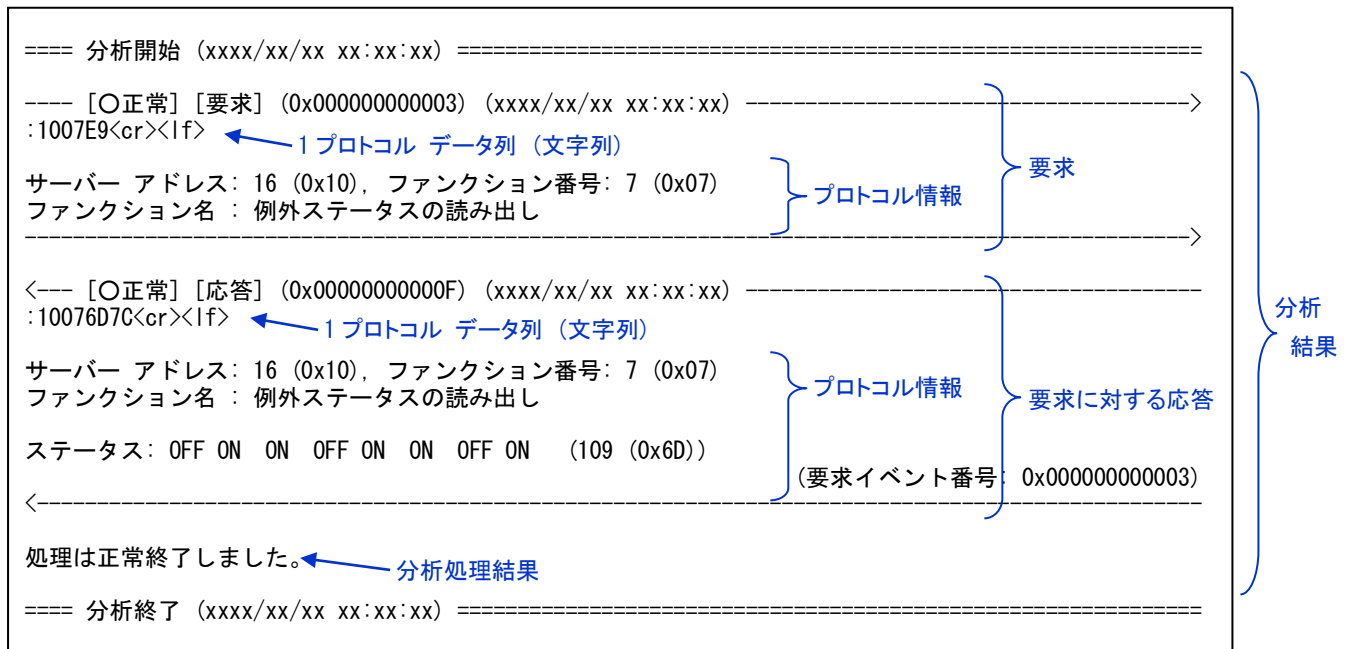


- (1) : プロトコル アナライザーの起動元アプリケーション名
- (2)-1: プロトコル アナライザーの名称とバージョン
- (2)-2: 準拠している標準プロトコルのバージョン
- (2)-3: プロトコル アナライザーの辞書ファイルのレビジョン
- (3) : 分析したモニター結果ファイルのファイル名 (メモリの場合は [メモリ] と出力されます)
- (4) : 分析を実行した時の [オプション] 設定状況
- (5) : 分析を実行した時の [分析対象サーバー]
- (6) : 分析を実行した時の [分析するデータの範囲]

これらの情報により、分析結果ファイルを後で参照した際に、何を使って、どのように分析したのか、明確になります。

■ 分析結果について

(1) 分析結果ファイルの書式の概要



クライアントからの [要求] は、矢印 (----->) で囲まれます。
 また、サーバーからの [応答] は、矢印 (<-----) で囲まれます。

[要求] および [応答] の 1 行目には、下記の情報が出力されます。

```

---- [分析結果] [要求] (先頭イベント番号) (通信日時) ----->
<--- [分析結果] [応答] (先頭イベント番号) (通信日時) -----
    
```

分析結果:

○正常	データ列が正常で、データの内容にも特に矛盾がなかった場合。
×異常	データ列は正常だが、データの内容に、標準プロトコル上、異常があった場合。
	分析できないデータ列だった場合。
△警告	クライアントが、サーバーからの [応答] を待たずに [要求] を送信した場合。 (注) 下記の場合、警告は出力されません。 ・元々 [応答] がないファンクション (ファンクション 8 サブファンクション 4, 等) ・サーバー アドレス 0 をブロードキャストと判断している場合
	クライアントが [要求] を送信していない状態で、サーバーが [応答] を返した場合。
	クライアントからの [要求] に対するサーバーからの [応答] が間違っている場合。 (ファンクション番号が合わない、等)
!!例外	データ長が正常で、サーバーが、標準プロトコルで定義されている「例外」を返した場合。 (注) !!例外 は、サーバーからの [応答] の分析結果の場合のみで、あり得ます。

先頭イベント番号:

このプロトコル データ列の先頭イベント番号 (16 進数) です。

通信日時:

このプロトコル データ列の先頭イベント番号のタイムスタンプです。

(注) モニター結果データにタイムスタンプが含まれていない場合は ???/?/?/? ??:?:?? になります。

(2) 分析結果 [×異常] の例

データ列は正常で、データの内容に標準プロトコル上の異常があった場合、次のよう出力されます。

```

---- [×異常] [要求] (0x000000000275) (xxxx/xx/xx xx:xx:xx) ----->
:0101FFFF0002FE<cr><l f>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し

先頭アドレス: 0x10000, 読み出し数: 2 (0x0002)

*** 先頭アドレスまたは読み出し数が間違っています。 ***
----->

```

← エラー メッセージ

この例は、コイル アドレス 65536 から 2つのステータスを読み出そうとしている例です。

分析できないデータ列は、次のよう出力されます。

```

---- [×異常] [要求] (0x000000000207) (xxxx/xx/xx xx:xx:xx) ----->
:01080013A0B094<cr><l f>
*** 分析できなかったデータ列 (原因: xxxxxxxxxx) ***
----->

```

← エラー メッセージ

この例は、ファンクション 8 (サブ ファンクション 19 (=RESERVED)) を [要求] した例です。

[原因] には、以下のいずれかが出力されます。

- ADU に使用できない文字が含まれています
- PDU の長さが奇数バイトです
- ヘッダー または フッターの異常です
- 未サポートのファンクションです
- データ長に関する異常です
- 予期しないエラーが発生しました

なお、分析できないデータ列が ADU の最大長 (513 バイト) より長かった場合、データ列は、次のような簡易的な出力になります。

```

---- [×異常] [不明] (0x0000000000FC) (xxxx/xx/xx xx:xx:xx) ----->
3A-30-31-30-31-30-30-30-30-30-30-30-30-31-46-44-3A-30-31-30-31-30-30-30-30-30-30-30-30-31-46-44-3A-30-
:
: (827 Bytes)
:
-3A-30-31-30-31-30-30-30-30-30-30-30-30-30-31-46-44-3A-30-31-30-31-30-30-30-30-30-30-30-30-31-46-44-0D-0A
*** 分析できなかったデータ列 (原因: xxxxxxxxxx) ***
----->

```

← エラー メッセージ

} データ列の簡易的な出力

(注) 各種エラー メッセージについては、「付録A: 分析結果ファイルでのメッセージ一覧」を参照してください。

(3) 分析結果 [△警告] の例

クライアントが、サーバーからの [応答] を待たずに [要求] を送信した場合、次のよう出力されます。

```

---- [○正常] [要求] (0x000000000001) (xxxx/xx/xx xx:xx:xx) ----->
:010100000001FD<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し

先頭アドレス: 1, 読み出し数: 1 (0x0001)
----->

<--- [△警告] [応答] -----
*** サーバー 1 (0x01) は、前回の要求に対して、応答を返していません。 *** ← 警告メッセージ
    前回要求: コイルの読み出し
    イベント: 000000000001   要求日時: xxxx/xx/xx xx:xx:xx
---- [○正常] [要求] (0x000000000014) (xxxx/xx/xx xx:xx:xx) ----->
:010100130013D8<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し
} 警告対象

先頭アドレス: 20, 読み出し数: 19 (0x0013)
----->

```

クライアントが [要求] を送信していない状態で、サーバーが [応答] を返した場合、次のよう出力されます。

```

---- [△警告] [要求] ----->
*** クライアントは、次の応答に対する要求を送信していません。 *** ← 警告メッセージ
<--- [○正常] [応答] (0x00000000001E) (xxxx/xx/xx xx:xx:xx) -----
:010103CD6B05BE<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し

出力バイト数: 3 (0x03)
ステータス : ----- 下記 -----
????+0:ON , ???+1:OFF, ???+2:ON , ???+3:ON , ???+4:OFF, ???+5:OFF, ???+6:ON , ???+7:ON ,
????+8:ON , ???+9:ON , ???+10:OFF, ???+11:ON , ???+12:OFF, ???+13:ON , ???+14:ON , ???+15:OFF,
???+16:ON , ???+17:OFF, ???+18:ON , ???+19:OFF, ???+20:OFF, ???+21:OFF, ???+22:OFF, ???+23:OFF
<-----
} 警告対象

```

この例では、[要求] がなく、コイル アドレスが不明のため ????+0~???+23 と出力されます。

クライアントが [要求] を送信していない状態で、サーバーが [応答] を返した場合、次のように出力されます。

```

---- [○正常] [要求] (0x000000000001) (xxxx/xx/xx xx:xx:xx) ----->
:01010000001FD<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し

先頭アドレス: 1, 読み出し数: 1 (0x0001)
----->

---- [△警告] [要求] ----->
*** クライアントは、イベント番号 0x000000000001 でファンクション 1 を要求しています。 ***
<--- [○正常] [応答] (0x000000000014) (xxxx/xx/xx xx:xx:xx) ----->
:01020105F7<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 2 (0x02)
ファンクション名 : 離散入力を読み出し

出力バイト数: 1 (0x01)
ステータス : ----- 下記 -----
????+0:ON , ???+1:OFF, ???+2:ON , ???+3:OFF, ???+4:OFF, ???+5:OFF, ???+6:OFF, ???+7:OFF
<----->
    
```

警告メッセージ

警告対象

この例では、ファンクション 1 の [要求] に対して、サーバーがファンクション 2 の [応答] を返した例です。

(4) 分析結果 [!!例外] の例

```

<--- [!!例外] [応答] (0x000000000286) (xxxx/xx/xx xx:xx:xx) -----
:0181017D<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 + 128 (0x01 + 0x80)
ファンクション名 : コイルの読み出し

例外コード: 1 (0x01) 許容されないファンクションです。
                                                    (要求イベント番号: 0x000000000275)
<----->
    
```

[!!例外] は、例外コード毎に、次のように出力されます。

例外コード	出力メッセージ
01	許容されないファンクションです。
02	許容されないデータ アドレスです。
03	許容されないデータ値です。
04	サーバー装置が故障しました。
05	要求は受け入れられ、処理中です。しばらくお待ちください。
06	サーバー装置はビジーです。
08	メモリのパリティ エラーを検出しました
0A	ゲートウェイは、内部のコミュニケーション パスを割り当てられませんでした。
0B	ゲートウェイは、ターゲット装置から回答が得られませんでした。

(注) 上表以外の例外コードの場合、(未知の例外) と出力されます。

付録A：分析結果ファイルでのメッセージ一覧

■ エラー メッセージ

Function	方向	エラー メッセージ
01	要求	先頭アドレスまたは読み出し数が間違っています。 コイルの読み出し数が範囲外です。
	応答	要求と応答の間で、コイルの数に矛盾があります。
02	要求	先頭アドレスまたは読み出し数が間違っています。 離散入力の読み出し数が範囲外です。
	応答	要求と応答の間で、離散入力の数に矛盾があります。
03	要求	先頭アドレスまたは読み出し数が間違っています。 保持レジスタの読み出し数が範囲外です。
	応答	出力バイト数が異常（奇数）です。 要求と応答の間で、保持レジスタの数に矛盾があります。
04	要求	先頭アドレスまたは読み出し数が間違っています。 入力レジスタの読み出し数が範囲外です。
	応答	出力バイト数が異常（奇数）です。 要求と応答の間で、入力レジスタの数に矛盾があります。
05	要求	ステータスの値が異常です
	応答	書き込んだアドレスが要求のアドレスと異なっています。 ステータスの値が異常です。
06	応答	書き込んだアドレスが要求のアドレスと異なっています。
08-00	応答	要求データと応答データが異なっています。
08-01	応答	処理タイプの値が異常です。
08-02	要求	データの値が異常です。
08-03	要求	データ（2 バイト目）の値が異常です。
	応答	要求の区切り文字と応答の区切り文字が異なっています。 データ（2 バイト目）の値が異常です。
08-04	要求	データの値が異常です。
08-10	要求	データの値が異常です。
	応答	データの値が異常です。
08-11	要求	データの値が異常です。
08-12	要求	データの値が異常です。
08-13	要求	データの値が異常です。
08-14	要求	データの値が異常です。
08-15	要求	データの値が異常です。
08-16	要求	データの値が異常です。
08-17	要求	データの値が異常です。
08-18	要求	データの値が異常です。
08-20	要求	データの値が異常です。
	応答	データの値が異常です。

Function	方向	エラー メッセージ
12	応答	プロトコル データのデータ長が異常です。
		出力バイト数が異常です。
15	要求	書き込み数と出力バイト数が合っていません。
		先頭アドレスまたは書き込み数が間違っています。
		コイルの書き込み数が範囲外です。
	応答	書き込んだアドレスが要求のアドレスと異なっています。 書き込んだコイルの数が要求の数と異なっています。
16	要求	先頭アドレスまたは書き込み数が間違っています。
		出力バイト数が異常（奇数）です。
		書き込み数と出力バイト数が合っていません。
	応答	書き込んだアドレスが要求のアドレスと異なっています。 書き込んだ保持レジスタの数が要求の数と異なっています。
20	要求	出力バイト数が異常（7 の倍数ではない）です。
		出力バイト数が異常（7 より小さい）です。
		ファイル番号が異常（0）です。（グループ x）
		レコードの読み出し数が範囲外です。（グループ x）
		先頭レコード番号が範囲外です。（グループ x）
	参照タイプが間違っています。（グループ x）	
	応答	出力バイト数が異常なグループがあるようです。（通信データ不足）
		要求と応答の間で、レコード数に矛盾があります。（グループ x）
出力バイト数が異常（偶数）です。（グループ x） 参照タイプが間違っています。（グループ x）		
21	要求	出力バイト数が異常なグループがあるようです。（通信データ不足）
		リクエスト データ長が異常（偶数）です。
		レコードの書き込み数が範囲外です。（グループ x）
		先頭レコード番号が範囲外です。（グループ x）
		ファイル番号が異常（0）です。（グループ x）
		参照タイプが間違っています。（グループ x）
	応答	出力バイト数が異常なグループがあるようです。（通信データ不足）
		書き込んだレコード データが要求のレコード データと異なっています。（グループ x）
		書き込んだレコード長が要求のレコード長と異なっています。（グループ x）
		書き込んだ先頭レコード番号が要求の先頭レコード番号と異なっています。（グループ x）
		書き込んだファイル番号が要求のファイル番号と異なっています。（グループ x）
		リクエスト データ長が異常（偶数）です。
		レコードの書き込み数が範囲外です。（グループ x）
		先頭レコード番号が範囲外です。（グループ x） ファイル番号が異常（0）です。（グループ x） 参照タイプが間違っています。（グループ x）
22	応答	OR マスクが要求の OR マスクと異なっています。
		AND マスクが要求の AND マスクと異なっています。
		参照アドレスが要求の参照アドレスと異なっています。

Function	方向	エラー メッセージ
23	要求	出力バイト数が異常です。
		レジスタの書き込み数が範囲外です。
		レジスタの読み出し数が範囲外です。
		レジスタの書き込み数が範囲外です。
		レジスタの読み出し数が範囲外です。
	応答	要求と応答の間で、読み出し数に矛盾があります。 出力バイト数が異常（奇数）です。
24	応答	プロトコル データのデータ長が異常です。
		出力バイト数が異常 です。（FIFO 数と合わない）
		出力バイト数が異常（奇数）です。
43-14	要求	装置 ID コードが異常です。
	応答	プロトコル データのデータ長が異常です。
		次のオブジェクト ID の値が異常です。
		後続のオブジェクト データが異常です。
		適合レベルの値が異常です。
		装置 ID コードが異常です。

Function 共通	このプロトコル データのデータ長は、標準仕様の PDU より長いです。
	LRC の値が異常です。

例外	応答	標準仕様上、この例外は、このファンクションでは発生しないはずです。
----	----	-----------------------------------

■ 警告メッセージ

方向	警告メッセージ
要求	サーバー x (0xXX) は、前回の要求に対して、応答を返していません。
応答	クライアントは、次の応答に対する要求を送信していません。
	クライアントは、イベント番号 0XXXXXXXXXXXXX でファンクション x を要求しています。

付録B：各ファンクションでの分析結果出力例

ファンクション1:

```

---- [○正常] [要求] (0x000000000021) (xxxx/xx/xx xx:xx:xx) ----->
:010100130013D8<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し

先頭アドレス: 0x0014, 読み出し数: 19 (0x0013)
----->

<--- [○正常] [応答] (0x000000000032) (xxxx/xx/xx xx:xx:xx) -----
:010103CD6B05BE<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 1 (0x01)
ファンクション名 : コイルの読み出し

出力バイト数: 3 (0x03)
ステータス : ----- 下記 -----
0x0014:ON , 0x0015:OFF, 0x0016:ON , 0x0017:ON , 0x0018:OFF, 0x0019:OFF, 0x001A:ON , 0x001B:ON ,
0x001C:ON , 0x001D:ON , 0x001E:OFF, 0x001F:ON , 0x0020:OFF, 0x0021:ON , 0x0022:ON , 0x0023:OFF,
0x0024:ON , 0x0025:OFF, 0x0026:ON
(要求イベント番号: 0x000000000021)
<-----

```

ファンクション2:

```

---- [○正常] [要求] (0x000000000025) (xxxx/xx/xx xx:xx:xx) ----->
:010200C4001623<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 2 (0x02)
ファンクション名 : 離散入力の読み出し

先頭アドレス: 0x00C5, 読み出し数: 22 (0x0016)
----->

<--- [○正常] [応答] (0x000000000037) (xxxx/xx/xx xx:xx:xx) -----
:010203ACDE353B<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 2 (0x02)
ファンクション名 : 離散入力の読み出し

出力バイト数: 3 (0x03)
ステータス : ----- 下記 -----
0x00C5:OFF, 0x00C6:OFF, 0x00C7:ON , 0x00C8:ON , 0x00C9:OFF, 0x00CA:ON , 0x00CB:OFF, 0x00CC:ON ,
0x00CD:OFF, 0x00CE:ON , 0x00CF:ON , 0x00D0:ON , 0x00D1:ON , 0x00D2:OFF, 0x00D3:ON , 0x00D4:ON ,
0x00D5:ON , 0x00D6:OFF, 0x00D7:ON , 0x00D8:OFF, 0x00D9:ON , 0x00DA:ON
(要求イベント番号: 0x000000000025)
<-----

```

ファンクション 3:

```

---- [○正常] [要求] (0x000000000027) (xxxx/xx/xx xx:xx:xx) ----->
:0103006B00038E<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 3 (0x03)
ファンクション名 : 保持レジスタの読み出し

先頭アドレス: 0x006C, 読み出し数: 3 (0x0003)
----->

<--- [○正常] [応答] (0x000000000039) (xxxx/xx/xx xx:xx:xx) -----
:010306022B0000006465<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 3 (0x03)
ファンクション名 : 保持レジスタの読み出し

出力バイト数: 6 (0x06)
レジスタ値 : ----- 下記 -----
0x006C:  555 (0x022B), 0x006D:  0 (0x0000), 0x006E:  100 (0x0064)
                                                (要求イベント番号: 0x000000000027)
<----->

```

ファンクション 4:

```

---- [○正常] [要求] (0x000000000001) (xxxx/xx/xx xx:xx:xx) ----->
:0104FFFD0003FC<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 4 (0x04)
ファンクション名 : 入力レジスタの読み出し

先頭アドレス: 0xFFFE, 読み出し数: 3 (0x0003)
----->

<--- [○正常] [応答] (0x000000000013) (xxxx/xx/xx xx:xx:xx) -----
:010406123456789ABC8B<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 4 (0x04)
ファンクション名 : 入力レジスタの読み出し

出力バイト数: 6 (0x06)
レジスタ値 : ----- 下記 -----
0xFFFE:  4660 (0x1234), 0xFFFF: 22136 (0x5678), 0x10000: 39612 (0x9ABC)
                                                (要求イベント番号: 0x000000000001)
<----->

```

ファンクション 5:

```

---- [○正常] [要求] (0x0000000000028) (xxxx/xx/xx xx:xx:xx) ----->
:010500ACFF004F<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 5 (0x05)
ファンクション名 : コイルの書き込み (シングル)

アドレス: 0x00AD, ステータス: ON (0xFF00)
----->

<---- [○正常] [応答] (0x000000000003A) (xxxx/xx/xx xx:xx:xx) -----
:010500ACFF004F<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 5 (0x05)
ファンクション名 : コイルの書き込み (シングル)

アドレス: 0x00AD, ステータス: ON (0xFF00)

(要求イベント番号: 0x0000000000028)
<-----

```

ファンクション 6:

```

---- [○正常] [要求] (0x0000000000025) (xxxx/xx/xx xx:xx:xx) ----->
:010600010003F5<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 6 (0x06)
ファンクション名 : 保持レジスタの書き込み (シングル)

アドレス: 0x0002, 保持レジスタ値: 3 (0x0003)
----->

<---- [○正常] [応答] (0x0000000000036) (xxxx/xx/xx xx:xx:xx) -----
:010600010003F5<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 6 (0x06)
ファンクション名 : 保持レジスタの書き込み (シングル)

アドレス: 0x0002, 保持レジスタ値: 3 (0x0003)

(要求イベント番号: 0x0000000000025)
<-----

```

ファンクション 7:

```

---- [○正常] [要求] (0x0000000000003) (xxxx/xx/xx xx:xx:xx) ----->
:1007E9<cr><lf>

サーバー アドレス: 16 (0x10), ファンクション番号: 7 (0x07)
ファンクション名 : 例外ステータスの読み出し
----->

<---- [○正常] [応答] (0x000000000000F) (xxxx/xx/xx xx:xx:xx) -----
:10076D7C<cr><lf>

サーバー アドレス: 16 (0x10), ファンクション番号: 7 (0x07)
ファンクション名 : 例外ステータスの読み出し

ステータス: OFF ON ON OFF ON ON OFF ON (109 (0x6D))

(要求イベント番号: 0x0000000000003)
<-----

```

ファンクション 8-0:

```

---- [○正常] [要求] (0x000000000001) (xxxx/xx/xx xx:xx:xx) ----->
:01080000414274<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-0 (0x08-0x0000)
ファンクション名 : 診断 - 送信したデータ フィールドのエコーバック

要求データ: 65 66 (0x41 0x42)
----->

<---- [○正常] [応答] (0x000000000013) (xxxx/xx/xx xx:xx:xx) -----
:01080000414274<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-0 (0x08-0x0000)
ファンクション名 : 診断 - 送信したデータ フィールドのエコーバック

応答データ: 65 66 (0x41 0x42)

(要求イベント番号: 0x000000000001)
<-----

```

ファンクション 8-1:

```

---- [○正常] [要求] (0x000000000025) (xxxx/xx/xx xx:xx:xx) ----->
:010800010000F6<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-1 (0x08-0x0001)
ファンクション名 : 診断 - 通信ポートの初期化と再起動

処理タイプ: 0 (0x0000) 一般的な初期化と再起動
----->

<---- [○正常] [応答] (0x000000000037) (xxxx/xx/xx xx:xx:xx) -----
:010800010000F6<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-1 (0x08-0x0001)
ファンクション名 : 診断 - 通信ポートの初期化と再起動

処理タイプ: 0 (0x0000) 一般的な初期化と再起動

(要求イベント番号: 0x000000000025)
<-----

```

ファンクション 8-2:

```

---- [○正常] [要求] (0x00000000006C) (xxxx/xx/xx xx:xx:xx) ----->
:010800020000F5<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-2 (0x08-0x0002)
ファンクション名 : 診断 - 診断レジスタの取得

データ: 0 (0x0000)
----->

<---- [○正常] [応答] (0x00000000007E) (xxxx/xx/xx xx:xx:xx) -----
:010800021020C5<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-2 (0x08-0x0002)
ファンクション名 : 診断 - 診断レジスタの取得

診断レジスタの内容: 4128 (0x1020)

(要求イベント番号: 0x00000000006C)
<-----

```

ファンクション 8-3:

```

---- [○正常] [要求] (0x000000000090) (xxxx/xx/xx xx:xx:xx) ----->
:010800030A00EA<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-3 (0x08-0x0003)
ファンクション名: 診断 - ASCII 入力の区切り文字 (デフォルト LF) の変更

メッセージの区切り文字: "lf" (10 (0x0A))
----->

<---- [○正常] [応答] (0x0000000000A2) (xxxx/xx/xx xx:xx:xx) -----
:010800030A00EA<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-3 (0x08-0x0003)
ファンクション名: 診断 - ASCII 入力の区切り文字 (デフォルト LF) の変更

メッセージの区切り文字: "lf" (10 (0x0A))

(要求イベント番号: 0x000000000090)
<-----

```

ファンクション 8-4:

```

---- [○正常] [要求] (0x0000000000B4) (xxxx/xx/xx xx:xx:xx) ----->
:010800040000F3<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-4 (0x08-0x0004)
ファンクション名: 診断 - Force Listen Only Mode

データ: 0 (0x0000)
----->

```

※ ファンクション 8-4 には [応答] がありません。

ファンクション 8-10:

```

---- [○正常] [要求] (0x0000000000C6) (xxxx/xx/xx xx:xx:xx) ----->
:0108000A0000ED<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-10 (0x08-0x000A)
ファンクション名: 診断 - 全てのカウンターと診断レジスタのクリア

データ: 0 (0x0000)
----->

<---- [○正常] [応答] (0x0000000000D7) (xxxx/xx/xx xx:xx:xx) -----
:0108000A0000ED<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-10 (0x08-0x000A)
ファンクション名: 診断 - 全てのカウンターと診断レジスタのクリア

データ: 0 (0x0000)

(要求イベント番号: 0x0000000000C6)
<-----

```


ファンクション 8-11:

```

----- [○正常] [要求] (0x0000000000E9) (xxxx/xx/xx xx:xx:xx) ----->
:0108000B0000EC<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-11 (0x08-0x000B)
ファンクション名 : 診断 - バス メッセージ数の取得

データ: 0 (0x0000)
----->

<----- [○正常] [応答] (0x0000000000FB) (xxxx/xx/xx xx:xx:xx) -----
:0108000B20309C<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-11 (0x08-0x000B)
ファンクション名 : 診断 - バス メッセージ数の取得

バス メッセージ総数: 8240 (0x2030)
(要求イベント番号: 0x0000000000E9)
<-----

```

ファンクション 8-12:

```

----- [○正常] [要求] (0x00000000010D) (xxxx/xx/xx xx:xx:xx) ----->
:0108000C0000EB<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-12 (0x08-0x000C)
ファンクション名 : 診断 - バス CRC エラー数の取得

データ: 0 (0x0000)
----->

<----- [○正常] [応答] (0x00000000011F) (xxxx/xx/xx xx:xx:xx) -----
:0108000C30407B<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-12 (0x08-0x000C)
ファンクション名 : 診断 - バス CRC エラー数の取得

バス CRC エラー数: 12352 (0x3040)
(要求イベント番号: 0x00000000010D)
<-----

```

ファンクション 8-13:

```

----- [○正常] [要求] (0x000000000131) (xxxx/xx/xx xx:xx:xx) ----->
:0108000D0000EA<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-13 (0x08-0x000D)
ファンクション名 : 診断 - バス例外エラー数の取得

データ: 0 (0x0000)
----->

<----- [○正常] [応答] (0x000000000143) (xxxx/xx/xx xx:xx:xx) -----
:0108000D40505A<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-13 (0x08-0x000D)
ファンクション名 : 診断 - バス例外エラー数の取得

バス例外エラー数: 16464 (0x4050)
(要求イベント番号: 0x000000000131)
<-----

```

ファンクション 8-14:

```

---- [○正常] [要求] (0x00000000155) (xxxx/xx/xx xx:xx:xx) ----->
:0108000E0000E9<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-14 (0x08-0x000E)
ファンクション名 : 診断 - サーバー メッセージ数の取得

データ: 0 (0x0000)
----->

<---- [○正常] [応答] (0x00000000167) (xxxx/xx/xx xx:xx:xx) -----
:0108000E506039<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-14 (0x08-0x000E)
ファンクション名 : 診断 - サーバー メッセージ数の取得

サーバー メッセージ数: 20576 (0x5060)
(要求イベント番号: 0x00000000155)
<-----

```

ファンクション 8-15:

```

---- [○正常] [要求] (0x00000000179) (xxxx/xx/xx xx:xx:xx) ----->
:0108000F0000E8<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-15 (0x08-0x000F)
ファンクション名 : 診断 - サーバーが応答しなかったメッセージ数の取得

データ: 0 (0x0000)
----->

<---- [○正常] [応答] (0x0000000018A) (xxxx/xx/xx xx:xx:xx) -----
:0108000F607018<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-15 (0x08-0x000F)
ファンクション名 : 診断 - サーバーが応答しなかったメッセージ数の取得

サーバーが応答しなかった数: 24688 (0x6070)
(要求イベント番号: 0x00000000179)
<-----

```

ファンクション 8-16:

```

---- [○正常] [要求] (0x0000000019C) (xxxx/xx/xx xx:xx:xx) ----->
:010800100000E7<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-16 (0x08-0x0010)
ファンクション名 : 診断 - サーバーが NAK を返した数の取得

データ: 0 (0x0000)
----->

<---- [○正常] [応答] (0x000000001AE) (xxxx/xx/xx xx:xx:xx) -----
:010800107080F7<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-16 (0x08-0x0010)
ファンクション名 : 診断 - サーバーが NAK を返した数の取得

サーバーが NAK を返した数: 28800 (0x7080)
(要求イベント番号: 0x0000000019C)
<-----

```

ファンクション 8-17:

```

----- [○正常] [要求] (0x0000000001C0) (xxxx/xx/xx xx:xx:xx) ----->
:010800110000E6<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-17 (0x08-0x0011)
ファンクション名: 診断 - サーバー・ビジー, 例外レスポンスを返した数の取得

データ: 0 (0x0000)
----->

<----- [○正常] [応答] (0x0000000001D1) (xxxx/xx/xx xx:xx:xx) -----
:010800118090D6<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-17 (0x08-0x0011)
ファンクション名: 診断 - サーバー・ビジー, 例外レスポンスを返した数の取得

サーバー・ビジーおよび例外を返した数: 32912 (0x8090)
(要求イベント番号: 0x0000000001C0)
<-----

```

ファンクション 8-18:

```

----- [○正常] [要求] (0x0000000001E3) (xxxx/xx/xx xx:xx:xx) ----->
:010800120000E5<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-18 (0x08-0x0012)
ファンクション名: 診断 - サーバーでオーバーラン エラーが発生した数の取得

データ: 0 (0x0000)
----->

<----- [○正常] [応答] (0x0000000001F5) (xxxx/xx/xx xx:xx:xx) -----
:0108001290A0B5<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-18 (0x08-0x0012)
ファンクション名: 診断 - サーバーでオーバーラン エラーが発生した数の取得

オーバーラン エラー発生数: 37024 (0x90A0)
(要求イベント番号: 0x0000000001E3)
<-----

```

ファンクション 8-20:

```

----- [○正常] [要求] (0x00000000022B) (xxxx/xx/xx xx:xx:xx) ----->
:010800140000E3<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-20 (0x08-0x0014)
ファンクション名: 診断 - オーバーラン エラー カウンターのクリアと、エラー フラグのリセット

データ: 0 (0x0000)
----->

<----- [○正常] [応答] (0x00000000023D) (xxxx/xx/xx xx:xx:xx) -----
:010800140000E3<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 8-20 (0x08-0x0014)
ファンクション名: 診断 - オーバーラン エラー カウンターのクリアと、エラー フラグのリセット

データ: 0 (0x0000)
(要求イベント番号: 0x00000000022B)
<-----

```

ファンクション 11:

```

---- [○正常] [要求] (0x000000000003) (xxxx/xx/xx xx:xx:xx) ----->
:100BE5<cr><lf>

サーバー アドレス: 16 (0x10), ファンクション番号: 11 (0x0B)
ファンクション名 : Comm イベント数の取得
----->

<--- [○正常] [応答] (0x00000000000E) (xxxx/xx/xx xx:xx:xx) -----
:100B0000FFFE7<cr><lf>

サーバー アドレス: 16 (0x10), ファンクション番号: 11 (0x0B)
ファンクション名 : Comm イベント数の取得

ステータス: 0 (0x0000) リモート装置は、処理を行っていません。
イベント数: 65535 (0xFFFF)
(要求イベント番号: 0x000000000003)
<-----

```

ファンクション 12:

```

---- [○正常] [要求] (0x000000000023) (xxxx/xx/xx xx:xx:xx) ----->
:110CE3<cr><lf>

サーバー アドレス: 17 (0x11), ファンクション番号: 12 (0x0C)
ファンクション名 : Comm イベントログの取得
----->

<--- [○正常] [応答] (0x00000000002D) (xxxx/xx/xx xx:xx:xx) -----
:110C08000001080121200090<cr><lf>

サーバー アドレス: 17 (0x11), ファンクション番号: 12 (0x0C)
ファンクション名 : Comm イベントログの取得

出力バイト数: 8 (0x08)
ステータス : 0 (0x0000) リモート装置は、処理を行っていません。
イベント数 : 264 (0x0108), メッセージ数: 289 (0x0121)
イベントログ: ----- 下記 (イベント 0 が最新の情報) -----
イベント 0: 32 (0x20)
イベント 1: 0 (0x00) リモート装置の通信ポートが初期化され再起動されました
(要求イベント番号: 0x000000000023)
<-----

```

ファンクション 15:

```

---- [○正常] [要求] (0x00000000002C) (xxxx/xx/xx xx:xx:xx) ----->
:010F0013000A02CD0103<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 15 (0x0F)
ファンクション名 : コイルの書き込み (マルチ)

先頭アドレス: 0x0014, 書き込み数: 10 (0x000A)
出力バイト数: 2 (0x02)
ステータス : ---- 下記 -----
0x0014:0N , 0x0015:0FF, 0x0016:0N , 0x0017:0N , 0x0018:0FF, 0x0019:0FF, 0x001A:0N , 0x001B:0N ,
0x001C:0N , 0x001D:0FF
----->

<--- [○正常] [応答] (0x000000000044) (xxxx/xx/xx xx:xx:xx) -----
:010F0013000AD3<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 15 (0x0F)
ファンクション名 : コイルの書き込み (マルチ)

先頭アドレス: 0x0014, 書き込み数: 10 (0x000A)
(要求イベント番号: 0x00000000002C)
<----->

```

ファンクション 16:

```

---- [○正常] [要求] (0x00000000002E) (xxxx/xx/xx xx:xx:xx) ----->
:01100001000204000A0102DB<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 16 (0x10)
ファンクション名 : 保持レジスタの書き込み (マルチ)

先頭アドレス: 0x0002, 書き込み数: 2 (0x0002)
出力バイト数: 4 (0x04)
レジスタ値 : ---- 下記 -----
0x0002: 10 (0x000A), 0x0003: 258 (0x0102)
----->

<--- [○正常] [応答] (0x00000000004A) (xxxx/xx/xx xx:xx:xx) -----
:011000010002EC<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 16 (0x10)
ファンクション名 : 保持レジスタの書き込み (マルチ)

先頭アドレス: 0x0002, 書き込み数: 2 (0x0002)
(要求イベント番号: 0x00000000002E)
<----->

```

ファンクション 17:

```

---- [○正常] [要求] (0x00000000001D) (xxxx/xx/xx xx:xx:xx) ----->
:1111DE<cr><lf>

サーバー アドレス: 17 (0x11), ファンクション番号: 17 (0x11)
ファンクション名 : サーバー ID レポート
----->

<--- [○正常] [応答] (0x000000000027) (xxxx/xx/xx xx:xx:xx) -----
:1111050123FF45670A<cr><lf>

サーバー アドレス: 17 (0x11), ファンクション番号: 17 (0x11)
ファンクション名 : サーバー ID レポート

出力バイト数: 5 (0x05)
サーバー ID ~ 追加のデータ: ---- 下記 -----
1 (0x01), 35 (0x23), 255 (0xFF), 69 (0x45), 103 (0x67)
(要求イベント番号: 0x00000000001D)
<----->

```

ファンクション 20:

```

----- [○正常] [要求] (0x00000000002F) (xxxx/xx/xx xx:xx:xx) ----->
:01140E0600040001000206000300090002BC<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 20 (0x14)
ファンクション名 : ファイル レコードの読み出し

出力バイト数: 14 (0x0E)
グループ番号: 1 (0x0001), 参照タイプ: 6 (0x06)
ファイル番号: 4 (0x0004), 先頭レコード番号: 1 (0x0001), レコード長: 2 (0x0002)
グループ番号: 2 (0x0002), 参照タイプ: 6 (0x06)
ファイル番号: 3 (0x0003), 先頭レコード番号: 9 (0x0009), レコード長: 2 (0x0002)
----->

<----- [○正常] [応答] (0x000000000056) (xxxx/xx/xx xx:xx:xx) -----
:01140C05060DFE0020050633CD00405E<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 20 (0x14)
ファンクション名 : ファイル レコードの読み出し

レスポンス データ長: 12 (0x0C)
グループ番号: 1 (0x0001), 参照タイプ: 6 (0x06), 出力バイト数: 5 (0x05)
ファイル番号: 4 (0x0004), 先頭レコード番号: 1 (0x0001), レコード長: 2 (0x0002)
0x0001: 3582 (0x0DFE), 0x0002: 32 (0x0020)
グループ番号: 2 (0x0002), 参照タイプ: 6 (0x06), 出力バイト数: 5 (0x05)
ファイル番号: 3 (0x0003), 先頭レコード番号: 9 (0x0009), レコード長: 2 (0x0002)
0x0009: 13261 (0x33CD), 0x000A: 64 (0x0040)

(要求イベント番号: 0x00000000002F)
<-----

```

ファンクション 21:

```

----- [○正常] [要求] (0x00000000003D) (xxxx/xx/xx xx:xx:xx) ----->
:01150D0600040007000306AF04BE100D35<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 21 (0x15)
ファンクション名 : ファイル レコードの書き込み

リクエスト データ長: 13 (0x0D)
グループ番号: 1 (0x0016), 参照タイプ: 6 (0x06)
ファイル番号: 4 (0x0004), 先頭レコード番号: 7 (0x0007), レコード長: 3 (0x0003)
0x0007: 1711 (0x06AF), 0x0008: 1214 (0x04BE), 0x0009: 4109 (0x100D)
----->

<----- [○正常] [応答] (0x000000000063) (xxxx/xx/xx xx:xx:xx) -----
:01150D0600040007000306AF04BE100D35<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 21 (0x15)
ファンクション名 : ファイル レコードの書き込み

レスポンス データ長: 13 (0x0D)
グループ番号: 1 (0x0016), 参照タイプ: 6 (0x06)
ファイル番号: 4 (0x0004), 先頭レコード番号: 7 (0x0007), レコード長: 3 (0x0003)
0x0007: 1711 (0x06AF), 0x0008: 1214 (0x04BE), 0x0009: 4109 (0x100D)

(要求イベント番号: 0x00000000003D)
<-----

```

ファンクション 22:

```

----- [○正常] [要求] (0x000000000030) (xxxx/xx/xx xx:xx:xx) ----->
:1116000400F20025BE<cr><lf>

サーバー アドレス: 17 (0x11), ファンクション番号: 22 (0x16)
ファンクション名 : 保持レジスタのマスク

参照アドレス: 0x0005, AND マスク: 242 (0x00F2), OR マスク: 37 (0x0025)
----->

<----- [○正常] [応答] (0x000000000045) (xxxx/xx/xx xx:xx:xx) -----
:1116000400F20025BE<cr><lf>

サーバー アドレス: 17 (0x11), ファンクション番号: 22 (0x16)
ファンクション名 : 保持レジスタのマスク

参照アドレス: 0x0005, AND マスク: 242 (0x00F2), OR マスク: 37 (0x0025)
(要求イベント番号: 0x000000000030)
<-----

```

ファンクション 23:

```

----- [○正常] [要求] (0x000000000035) (xxxx/xx/xx xx:xx:xx) ----->
:01170003000600E00030600FF00FF00FCB<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 23 (0x17)
ファンクション名 : 保持レジスタの書き込みと読み出し (マルチ)

読み出し先頭アドレス: 0x0004, 読み出し数: 6 (0x0006)
書き込み先頭アドレス: 0x000F, 書き込み数: 3 (0x0003),
出力バイト数: 6 (0x06)
レジスタ値 : ----- 下記 -----
0x000F: 255 (0x00FF), 0x0010: 255 (0x00FF), 0x0011: 255 (0x00FF)
----->

<----- [○正常] [応答] (0x00000000005D) (xxxx/xx/xx xx:xx:xx) -----
:01170C00FE0ACD00010003000D00FFF7<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 23 (0x17)
ファンクション名 : 保持レジスタの書き込みと読み出し (マルチ)

出力バイト数: 12 (0x0C)
レジスタ値 : ----- 下記 -----
0x0004: 254 (0x00FE), 0x0005: 2765 (0x0ACD), 0x0006: 1 (0x0001), 0x0007: 3 (0x0003),
0x0008: 13 (0x000D), 0x0009: 255 (0x00FF)
(要求イベント番号: 0x000000000035)
<-----

```

ファンクション 24:

```

----- [○正常] [要求] (0x000000000029) (xxxx/xx/xx xx:xx:xx) ----->
:011804DE05<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 24 (0x18)
ファンクション名 : FIFO Queue の読み出し

FIFO ポインター アドレス: 1246 (0x04DE)
----->

<----- [○正常] [応答] (0x000000000037) (xxxx/xx/xx xx:xx:xx) -----
:01180006000201B8128490<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 24 (0x18)
ファンクション名 : FIFO Queue の読み出し

出力バイト数: 6 (0x0006), FIFO 数: 2 (0x0002)
FIFO Value レジスタ: ----- 下記 -----
0x04DE: 440 (0x01B8), 0x04DF: 4740 (0x1284)
(要求イベント番号: 0x000000000029)
<-----

```

ファンクション 43-13:

```

----- [○正常] [要求] (0x000000000001) (xxxx/xx/xx xx:xx:xx) ----->
:112B0D123456789ABCDEF0123456789ABCDEF01234567833<cr><lf>

サーバー アドレス: 17 (0x11), ファンクション番号: 43-13 (0x2B-0x0D)
ファンクション名 : カプセル化されたインタフェースの伝送 - CANopen General Ref. Req. and Res. PDU

MEI タイプ: 13 (0x0D)
MEI タイプの特定データ: ----- 下記 -----
0x12, 0x34, 0x56, 0x78, 0x9A, 0xBC, 0xDE, 0xF0, 0x12, 0x34, 0x56, 0x78, 0x9A, 0xBC, 0xDE, 0xF0,
0x12, 0x34, 0x56, 0x78
----->

<----- [○正常] [応答] (0x0000000000035) (xxxx/xx/xx xx:xx:xx) -----
:112B0D123456789ABCDEF0123456789ABCDEF01234567833<cr><lf>

サーバー アドレス: 17 (0x11), ファンクション番号: 43-13 (0x2B-0x0D)
ファンクション名 : カプセル化されたインタフェースの伝送 - CANopen General Ref. Req. and Res. PDU

MEI タイプ: 13 (0x0D)
MEI タイプの特定データ: ----- 下記 -----
0x12, 0x34, 0x56, 0x78, 0x9A, 0xBC, 0xDE, 0xF0, 0x12, 0x34, 0x56, 0x78, 0x9A, 0xBC, 0xDE, 0xF0,
0x12, 0x34, 0x56, 0x78

(要求イベント番号: 0x000000000001)
<-----

```

ファンクション 43-14:

```

----- [○正常] [要求] (0x000000000003) (xxxx/xx/xx xx:xx:xx) ----->
:012B0E0100C5<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 43-14 (0x2B-0x0E)
ファンクション名 : カプセル化されたインタフェースの伝送 - 装置識別の読み出し

MEI タイプ: 14 (0x0E)
装置 ID コード : 1 (0x01) 基本的な装置識別の取得を要求 (連続アクセス)
オブジェクト ID : 0 (0x00) ベンダー名 (分類: 基本)
----->

<----- [○正常] [応答] (0x000000000015) (xxxx/xx/xx xx:xx:xx) -----
:012B0E01010000030016436F6D70616E79206964656E74696669636174696F6E010D50726F6475637420636F64655802
0556322E3131C9<cr><lf>

サーバー アドレス: 1 (0x01), ファンクション番号: 43-14 (0x2B-0x0E)
ファンクション名 : カプセル化されたインタフェースの伝送 - 装置識別の読み出し

MEI タイプ: 14 (0x0E)
装置 ID コード : 1 (0x01) 基本的な装置識別の取得を要求 (連続アクセス)
適合レベル : 1 (0x01) 基本的な識別 (連続アクセス)
後続のオブジェクト: 0 (0x00) 後続のオブジェクトは、ありません。
オブジェクト数 : 3 (0x03)
オブジェクトデータ: ----- 下記 -----
オブジェクト ID : 0 (0x00) ベンダー名 (分類: 基本)
  オブジェクト長: 22 (0x16)
  0x43, 0x6F, 0x6D, 0x70, 0x61, 0x6E, 0x79, 0x20, 0x69, 0x64, 0x65, 0x6E, 0x74, 0x69, 0x66, 0x69,
  0x63, 0x61, 0x74, 0x69, 0x6F, 0x6E
  "Company identification"
オブジェクト ID : 1 (0x01) 製品コード (分類: 基本)
  オブジェクト長: 13 (0x0D)
  0x50, 0x72, 0x6F, 0x64, 0x75, 0x63, 0x74, 0x20, 0x63, 0x6F, 0x64, 0x65, 0x58
  "Product codeX"
オブジェクト ID : 2 (0x02) メジャーマイナーリビジョン (分類: 基本)
  オブジェクト長: 5 (0x05)
  0x56, 0x32, 0x2E, 0x31, 0x31
  "V2.11"

(要求イベント番号: 0x000000000003)
<-----

```


付録C：対訳

■ ファンクション名

Func.	英語	日本語
01	Read Coils	コイルの読み出し
02	Read Discrete Inputs	離散入力の読み出し
03	Read Holding Registers	保持レジスタの読み出し
04	Read Input Registers	入力レジスタの読み出し
05	Write Single Coil	コイルの書き込み (シングル)
06	Write Single Register	保持レジスタの書き込み (シングル)
07	Read Exception Status	例外ステータスの読み出し
08-00	Diagnostics - Return Query Data	診断 - 送信したデータ フィールドのエコーバック
08-01	Diagnostics - Restart Communications Option	診断 - 通信ポートの初期化と再起動
08-02	Diagnostics - Return Diagnostic Register	診断 - 診断レジスタの取得
08-03	Diagnostics - Change ASCII Input Delimiter	診断 - ASCII 入力の区切り文字 (デフォルト LF) の変更
08-04	Diagnostics - Force Listen Only Mode	診断 - Force Listen Only Mode
08-10	Diagnostics - Clear Counters and Diagnostic Register	診断 - 全てのカウンターと診断レジスタのクリア
08-11	Diagnostics - Return Bus Message Count	診断 - バス メッセージ数の取得
08-12	Diagnostics - Return Bus Communication Error Count	診断 - バス CRC エラー数の取得
08-13	Diagnostics - Return Bus Exception Error Count	診断 - バス例外エラー数の取得
08-14	Diagnostics - Return Server Message Count	診断 - サーバー メッセージ数の取得
08-15	Diagnostics - Return Server No Response Count	診断 - サーバーが応答しなかったメッセージ数の取得
08-16	Diagnostics - Return Server NAK Count	診断 - サーバーが NAK を返した数の取得
08-17	Diagnostics - Return Server Busy Count	診断 - サーバー・ビジー, 例外レスポンスを返した数の取得
08-18	Diagnostics - Return Bus Character Overrun Count	診断 - サーバーでオーバーラン エラーが発生した数の取得
08-20	Diagnostics - Clear Overrun Counter and Flag	診断 - オーバーラン エラー カウンターのクリアと、エラー フラグのリセット
11	Get Comm Event Counter	Comm イベント数の取得
12	Get Comm Event Log	Comm イベントログの取得
15	Write Multiple Coils	コイルの書き込み (マルチ)
16	Write Multiple Registers	保持レジスタの書き込み (マルチ)
17	Report Server ID	サーバー ID レポート
20	Read File Record	ファイル レコードの読み出し
21	Write File Record	ファイル レコードの書き込み
22	Mask Write Register	保持レジスタのマスク
23	Read/Write Multiple Registers	保持レジスタの書き込みと読み出し (マルチ)
24	Read FIFO Queue	FIFO Queue の読み出し
43-13	Encapsulated Interface Transport - CANopen General Ref. Req. and Res. PDU	カプセル化されたインタフェースの伝送 - CANopen General Ref. Req. and Res. PDU
43-14	Encapsulated Interface Transport - Read Device Identification	カプセル化されたインタフェースの伝送 - 装置識別の読み出し

■ 用語

(注) 内容により、対訳とは別に、柔軟に表現を変更している場合があります。

英語	日本語
Acknowledge	受け入れ
Address	アドレス
Alert	警告
Basic	基本 または 基本的な
Bus	バス
Busy	ビジー
Byte Count	出力バイト数
Category	分類
Client	クライアント
Coil (s)	コイル
Communications	通信
Conformity Level	適合レベル
Count	数
Counter (s)	カウンター
Data	データ
Delimiter	区切り文字
Dependant	依存
Device	装置
Diagnostic Register	診断レジスタ
Diagnostics	診断
Discrete Inputs	離散入力
Error	異常エラー
Even Number	偶数
Event	イベント
Exception	例外
Extended	拡張
Failure	故障
FIFO Value Register	FIFO Value レジスタ
File	ファイル
File Number	ファイル番号
Following	下記
Function Name	ファンクション名
Function Number	ファンクション番号
Gateway	ゲートウェイ
Get	取得
Group Number	グループ番号
Holding Registers	保持レジスタ
Identification	識別
Illegal	異常 または 許容されない
Individual Access	個別アクセス

英語	日本語
Input Registers	入力レジスタ
Issue	発行
Last Request	前回要求
Length	長
Log	ログ
MajorMinorRevision	メジャーマイナーリビジョン
Memory	メモリ
Message	メッセージ
ModelName	モデル名
More Follows	後続のオブジェクト
Multiple	マルチ
Normal	正常 または 一般的な
Number of xxxx	xxxx 数
Object	オブジェクト
Obscurity	不明
Occur	発生
Odd Number	奇数
Overrun	オーバーラン エラー
Parity	パリティ
Pointer	ポインタ
Previously	以前に
Process	処理
ProductCode	製品コード
ProductName	製品名
Quantity of xxxx	読み出し数 または 書き込み数
Read	読み出し
Receive	受信
Record	レコード
Reference Type	参照タイプ
Register (s)	レジスタ
Regular	通常
Remote	リモート
Request	要求 または リクエスト
Request Date	要求日時
Reserved	予約
Respond	回答
Response	応答
Restart	再起動
Return	取得
Send	送信

英語	日本語
Server	サーバー
Single	シングル
Specific	特定
Specification	仕様
Standard	標準
Starting Address	先頭アドレス
Starting Record	先頭レコード番号
Status	ステータス
Stream Access	連続アクセス
Total	総数
Transport	伝送
Type	タイプ
Unknown	未知の
UserApplicationName	ユーザー アプリケーション名
Value	値
VenderName	ベンダー名
VendorUrl	ベンダー URL
Write	書き込み

RS-232C Monitor and Analyzer Series

プロトコル アナライザー for Modbus ASCII (Model: AKM-RSM-FM0) ユーザーズ マニュアル

2016年11月	0.3.0.0版 発行
2016年12月	0.4.0.0版 発行
2017年1月	1.0.0.0版 発行
2019年7月	1.1.0.0版 発行
2019年11月	1.1.1.0版 発行
2022年2月	1.2.0.0版 発行
2023年7月	1.3.0.0版 発行

Copyright (C) 2016-2023 秋山製作所
発行所 秋山製作所

ご注意

- ・このマニュアルの内容の一部または全部を無断転載することは禁止されています。
- ・このマニュアルの内容に関しては、将来予告なしに変更することがあります。