

# MdobusRTUログ支援ソフトの使い方

## ――スクリプトの自動生成とログファイル変換ソフト――

【LogFriendForModbusRTU】

2020年9月15日 (株)データ・テクノ

### ■■ 概要

SDロガー4とラインコンバータで、Modbusからデータを取得し、記録することを支援する、パソコン上のプログラムです。設定された条件で、SDロガー4のスクリプトを自動生成する機能と、SDロガー4で記録されたファイルをCSVに変換する機能があります。

生成と変換は、ドラッグアンドドロップの、簡単な操作で行うことができます。

何れも画面上で条件を設定します。設定した条件はプログラムが終了しても保存されます。また、個別のファイルに保存することもできます。

### ■■ インストール

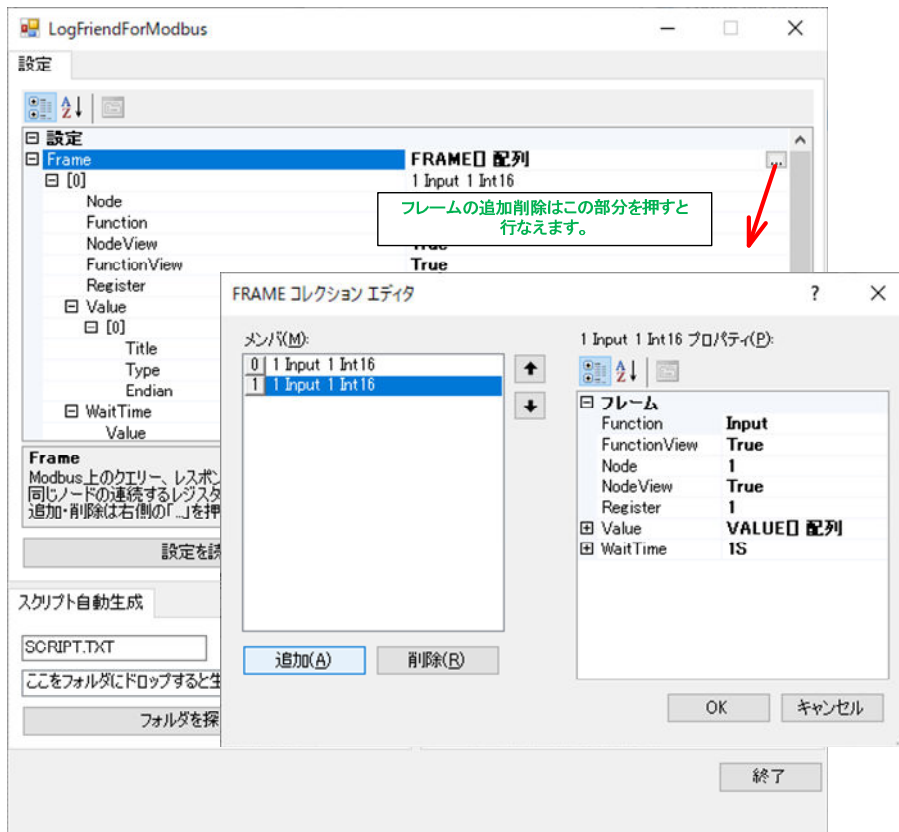
インストール作業は特に必要ありません、ダウンロードしたファイルを適当なフォルダに展開し、「LogFriendForModbusRTU.exe」をダブルクリックなどで起動してください。

実行には、「.NET Framework 3.5」以降を必要とします。

|                  |                           |
|------------------|---------------------------|
| MdobusRTUログ支援ソフト | LogFriendForModbusRTU.exe |
|------------------|---------------------------|

### ■■ 操作

●フレーム(配列要素)の追加削除



※その他の配列要素(値)も同様にして追加、削除が行なえます。

■ ■ 設定

● スクリプトの自動生成、ログファイルの変換は、この設定をもとに行なわれます。

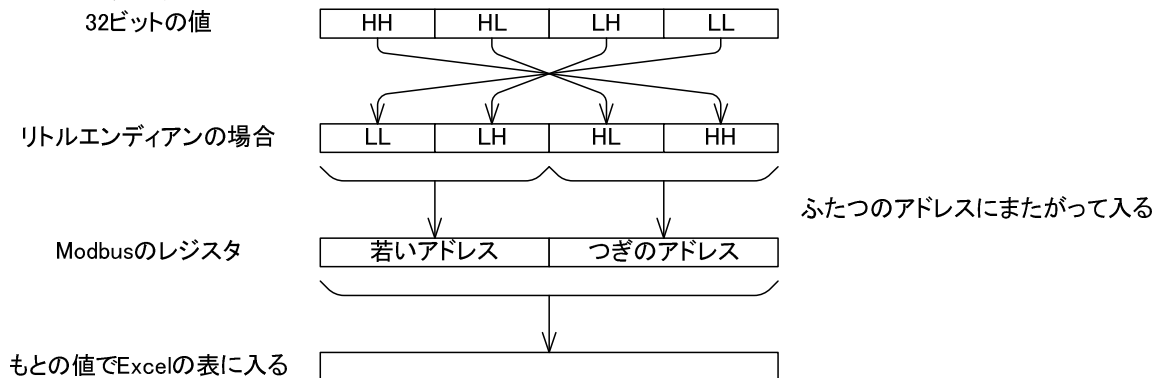
|              |                                                                                                                             |
|--------------|-----------------------------------------------------------------------------------------------------------------------------|
| Frame...     | 複数のフレームを設定できます。<br>Modbus 上のクエリー、レスポンスの1回のやり取りを、フレームという単位で扱います。同じノードの連続するレジスタがひとつのフレームで取得できます。                              |
| Node         | 1~247 ノードアドレス(スレーブアドレス、デバイスアドレス)                                                                                            |
| Function     | Coil, Status, Input, Holding<br>取得する、Modbus 上のデータの種類を指定します。                                                                 |
| NodeView     | Y/N ノードアドレスを出力項目に含めるかどうか。                                                                                                   |
| FunctionView | Y/N ファンクションコードを出力項目に含めるかどうか。                                                                                                |
| Register     | 1~9999 取得する、Modbus 上のレジスタの開始アドレスを1から始まるオフセット値で指定してください。                                                                     |
| Value...     | 複数の値を設定できます。値の変換方法などを設定します。                                                                                                 |
| Title        | <文字列> タイトル行に表示する文字列。<br>ひとつも指定が無ければ、タイトル行は出力されません。                                                                          |
| Type         | Int16, UInt16, Float, Double, Int32, UInt32, Bit 値の種類<br>Coil, Status は Bit を指定してください。<br>Input, Holding は Bit 以外を指定してください。 |
| Endian       | Big, Little エンディアン(Bit では無視されます)                                                                                            |
| WaitTime...  | 1 ミリ秒~999 分 クエリーを送信するまでの待ち時間<br>この時間内に前のレスポンスの受信が完了している必要があります。                                                             |
| Value        | 1~60000                                                                                                                     |
| Unit         | MS, S, M                                                                                                                    |

● サポートしているデータの種類の種類と Register に記述する値

| Function symbol | データの種類の種類 | Modbus 上のアドレス範囲 | Register に記述する値 |
|-----------------|-----------|-----------------|-----------------|
| Coil            | コイル       | 1~9999          | 1~9999          |
| Status          | 入力ステータス   | 10001~19999     | 1~9999          |
| Input           | 入力レジスタ    | 30001~39999     | 1~9999          |
| Holding         | 保持レジスタ    | 40001~49999     | 1~9999          |

● エンディアンの説明

(リトルエンディアン)



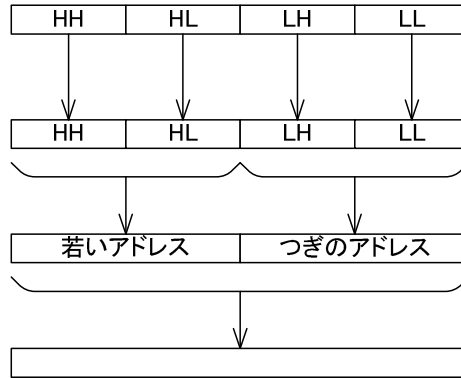
(ビッグエンディアン)

32ビットの値

ビッグエンディアンの場合

Modbusのレジスタ

もとの値でExcelの表に入る



ふたつのアドレスにまたがって入る

## ■ 生成されるスクリプト

### ● 生成されるスクリプト例

|                                                                                                                 |                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>#f:ENCODE *<br/><br/>#LOOP<br/><br/>#WAIT TIME 1S<br/>#LOG *_<br/>:010400000008F1CC<br/>...<br/>#END</pre> | <p>受信データに「*」が含まれた場合、ログデータは「**」に置き換えて記録する(エンコード機能を設定)。</p> <p>#END までを繰り返す。</p> <p>設定された時間待つ。<br/>「_」はフレーム番号に置き換える。<br/>設定されたクエリーを送信する。</p> <p>フレームの数だけ続く。</p> <p>#LOOP の終了。</p> |
|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### ● フレームの分離について

クエリーの送信に先立って、開始マーク「\*\_」が記録されます(「\_」はフレーム番号)。  
レスポンス中に「\*」が含まれた場合は「\*\*」に置き換えられます(エンコード機能)。

Modbus RTU では一定時間受信が無いことで、フレームの区切りを判断しなければなりません。  
しかしSDロガーで記録されたデータには、時間経過は失われているので、フレームの開始の判断が難しくなります。  
ロギングの開始から全く冗長な受信が無い場合は、受信バイト数で判断できないこともありませんが、ひとたび狂うと後を続けるのが難しくなります。

そこでクエリーの送信前に開始マーク「\*\_」をログデータに記録します。  
データを解析する側は、「\*\_」を見つけることでフレームの開始を知ることが出来ます。

しかしそれをする、Modbus 機器が返すレスポンス自身に「\*\_」が含まれると区別がつかなくなります。  
そのためエンコード機能を使います。エンコード機能を使い、本来のデータに「\*」が含まれる場合はふたつの「\*\*」に置き換えて記録します。  
データを解析する側は、「\*\_」は開始マークとして認識しますが、「\*\*\_」は本来のデータ(「\*\_」)だと認識します。

## ■ ■ 設定例(入力レジスタ)

※つぎのように定義された入力レジスタ値を取得する例を示します。

※ノードアドレス=1、待ち時間=1 秒の場合。

### ● 入力レジスタの定義

| アドレス  | 説明                 |
|-------|--------------------|
| 30001 | 値A: 符号あり整数値(16ビット) |
| 30002 | 値B: 符号なし整数値(16ビット) |
| 30003 | 値C: 浮動小数点値(単精度)    |
| 30004 |                    |
| 30005 | 値D: 浮動小数点値(倍精度)    |
| 30006 |                    |
| 30007 |                    |
| 30008 |                    |
| 30009 | 値E: 符号あり整数値(32ビット) |
| 30010 |                    |
| 30011 | 値F: 符号なし整数値(32ビット) |
| 30012 |                    |

### ● 設定例

|              |        |
|--------------|--------|
| Frame[0]     |        |
| Node         | 1      |
| Function     | Input  |
| NodeView     | Y      |
| FunctionView | Y      |
| Register     | 1      |
| Value[0]     |        |
| Title        | 値A     |
| Type         | Int16  |
| Endian       | Big    |
| Value[1]     |        |
| Title        | 値B     |
| Type         | UInt16 |
| Endian       | Big    |
| Value[2]     |        |
| Title        | 値C     |
| Type         | Float  |
| Endian       | Big    |
| Value[3]     |        |
| Title        | 値D     |
| Type         | Double |
| Endian       | Big    |
| Value[4]     |        |
| Title        | 値E     |
| Type         | Int32  |
| Endian       | Big    |
| Value[5]     |        |
| Title        | 値F     |
| Type         | UInt32 |
| Endian       | Big    |
| WaitTime...  |        |
| Value        | 1      |
| Unit         | S      |

## ■■設定例2(コイル)

※つぎのように定義されたコイルの値を取得する例を示します。  
※ノードアドレス=1、待ち時間=1 秒の場合。

### ●コイルの定義

| アドレス | 説明   |
|------|------|
| 1    | コイルA |
| 2    | コイルB |
| 3    | コイルC |
| 4    | コイルD |

### ●設定例

|              |      |
|--------------|------|
| Frame[0]     |      |
| Node         | 1    |
| Function     | Coil |
| NodeView     | Y    |
| FunctionView | Y    |
| Register     | 1    |
| Value[0]     |      |
| Title        | コイルA |
| Type         | Bit  |
| Endian       | Big  |
| Value[1]     |      |
| Title        | コイルB |
| Type         | Bit  |
| Endian       | Big  |
| Value[2]     |      |
| Title        | コイルC |
| Type         | Bit  |
| Endian       | Big  |
| Value[3]     |      |
| Title        | コイルD |
| Type         | Bit  |
| Endian       | Big  |
| WaitTime...  |      |
| Value        | 1    |
| Unit         | S    |

※Endian の設定は無視される。

## ■■その他

ModbusRTU ログ支援ソフト(LogFriendForModbusRTU)は、弊社製品でデータ収集を行なう目的でのみ、無償でご使用いただけます。

その他の目的には、ご使用にならないようにお願いいたします。

### 株式会社データ・テクノ

京都市下京区西七条東御前田町48番地

〒600-8898 TEL:(075)313-3275 FAX:(075)314-0576

<http://www.datatecno.co.jp/>

・本取扱説明書の内容は、改良のため予告なく変更することがあります。