

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】

2022年1月7日 (株)データ・テクノ (改1版)

2020年9月15日 (株)データ・テクノ

■■ 概要

SDロガー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ロガーのスクリプトを自動生成する機能と、SDロガー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。

Ver. 0.02 から、ロガーで日付時刻を付加する機能を、支援する機能が追加されました。

■■ 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

The screenshot shows the LogFriendForModbus application window. The main area is titled '設定' (Settings) and contains a tree view on the left and a table on the right. The table is titled 'FRAME 配列' (FRAME Array) and has columns for Node, Function, NodeView, FunctionView, Register, Value, Title, Type, Endian, and WaitTime. The first row shows '1 Input 1 Int16' under Node, 'Input' under Function, and 'True' under NodeView. Below the table is a 'Frame' section with explanatory text. At the bottom, there are two tabs: 'スクリプト自動生成' (Script Auto-Generation) and 'ログファイル変換' (Log File Conversion). The 'スクリプト自動生成' tab has a text box 'SCRIPT.TXT' and a button 'フォルダを探して生成' (Generate by searching for folder). The 'ログファイル変換' tab has a text box and a button 'ファイルを探して変換する' (Convert by searching for file). A '終了' (End) button is at the bottom right. Red arrows point from various parts of the interface to external file icons: '0000.LOG', '0000.csv', and 'sample.xml'. Green text boxes provide instructions: 'スクリプトの生成方法や、ログファイルの変換方法について設定します。' (Set the script generation method and log file conversion method.); '設定はプログラムを終了しても保存されます。' (Settings are saved even if the program is closed.); 'ログファイルをウィンドウのどこかにドラッグ&ドロップすると変換されます。' (Log files are converted by dragging&dropping them anywhere in the window.); '設定はファイルに保存し、読み出せます。' (Settings are saved to a file and can be loaded.); 'この部分をドラッグ&ドロップすると、スクリプトファイルが自動生成されます。' (Dragging&dropping this part automatically generates a script file.); and '設定を読み込む' (Load settings) and '設定を保存' (Save settings) buttons.

設定

FRAME 配列

Node	Function	NodeView	FunctionView	Register	Value	Title	Type	Endian	WaitTime
1 Input 1 Int16	Input	True							1

Frame
Modbus上のクエリ、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。
追加・削除は右側の「+」を押してください。

設定を読み込む 設定を保存

スクリプト自動生成 ログファイル変換

SCRIPT.TXT
ここをフォルダにドロップすると生成されます。
フォルダを探して生成

ログギングしたファイルを、このウィンドウのどこかにドラッグ・アンド・ドロップすると変換します。
ファイルを探して変換する

終了

0000.LOG

0000.csv

sample.xml

SCRIPT.TXT

スクリプトの生成方法や、ログファイルの変換方法について設定します。

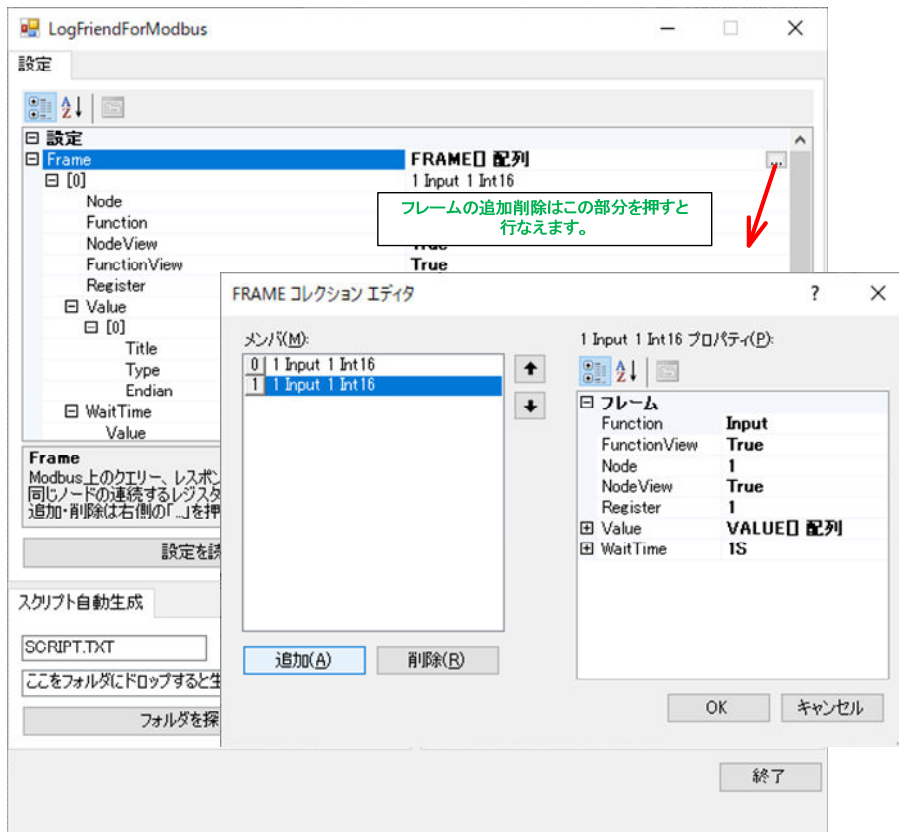
設定はプログラムを終了しても保存されます。

ログファイルをウィンドウのどこかにドラッグ&ドロップすると変換されます。

設定はファイルに保存し、読み出せます。

この部分をドラッグ&ドロップすると、スクリプトファイルが自動生成されます。

●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。

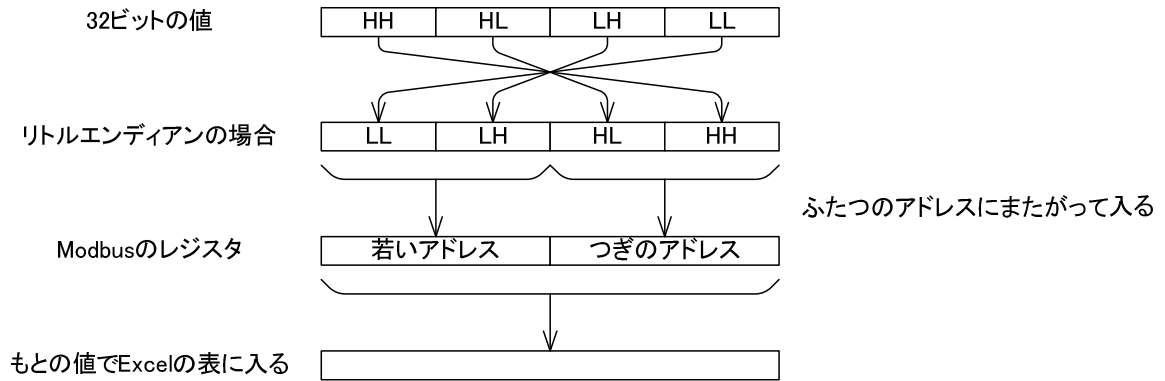
Frame...	複数のフレームを設定できます。 Modbus 上のクエリー、レスポンスの1回のやり取りを、フレームという単位で扱います。同じノードの連続するレジスタがひとつのフレームで取得できます。
Node	1~247 ノードアドレス(スレーブアドレス、デバイスアドレス) ロガーで日付時刻を付加する場合は無視されます。
Function	Coil, Status, Input, Holding, DateTime 取得する、Modbus 上のデータの種別を指定します。 ロガーで日付時刻を付加する場合は DateTime を指定してください。
NodeView	True, False ノードアドレスを出力項目に含めるかどうか。 ロガーで日付時刻を付加する場合は False を指定してください。
FunctionView	True, False ファンクションコードを出力項目に含めるかどうか。 ロガーで日付時刻を付加する場合は False を指定してください。
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。 ロガーで日付時刻を付加する場合は無視されます。
Format	<文字列> ロガーで日付時刻を付加する場合は、その書式をロガーの LOG 文の形式で指定してください。 その他では無視されます。
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。 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, Through 値の種類 Coil, Status は Bit を指定してください。 Input, Holding は Bit, Through 以外を指定してください。 DateTime は Through を指定してください。
Endian	Big, Little エンディアン(Bit, DateTime 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 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 Unit	1~60000 MS, S, M

● サポートしているデータの種別と Register に記述する値

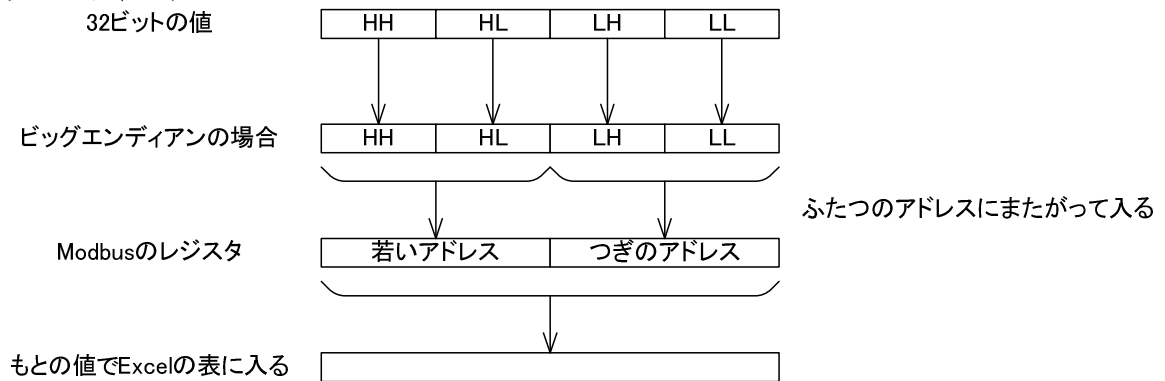
Function symbol	データの種別	Modbus 上の アドレス範囲	Register に 記述する値
Coil	コイル	1~9999	1~9999
Status	入力ステータス	10001~19999	1~9999
Input	入力レジスタ	30001~39999	1~9999
Holding	保持レジスタ	40001~49999	1~9999

●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)



■生成されるスクリプト

●生成されるスクリプト例

<pre>#f:ENCODE * #LOOP #WAIT TIME 1S #LOG *120@Y/@M/@D,@h:@m:@s #WAIT TIME 1S #LOG *_ :010400000008F1CC ... #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。</p> <p>設定された時間待つ。</p> <p>「_」はフレーム番号に置き換える。</p> <p>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	True
FunctionView	True
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	True
FunctionView	True
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。

■■設定例3(日付時刻の付加)

ロガーで日付時刻を付加させる場合は、Function に DateTime を指定してください。
Value はひとつだけ設け、Type に Through を指定してください。

Format にはロガーのスキプトの、LOG 文の書式で指定してください。「20@Y/@M/@D,@h:@m:@s」と記述すると「2022/01/01,10:10:10」のような書式で付加されます。

この例では日付と時刻はふたつの列に分かれて付加されます。

Format には LOG 文の書式ならば任意のものが指定できます。従って日付時刻に限らず LOG 文で記録できるものなら、他のものも記録できます。

タイトル行を設ける場合は Title に任意の文字列が指定できます。日付時刻が複数列に渡る場合は、列の区切りに「,(カンマ)」も記述してください。

NodeView, FunctionView は False を指定してください。

日付時刻の付加には WaitTime は必要ありません。DateTime の、つぎのフレームの WaitTime には 0 を指定できます。

DateTime を指定したフレーム自身の WaitTime の必要性は、直前のフレームに依存します。

Node, Register の設定は DateTime が指定された場合は無視されます。Endian の設定は、Trough が指定された場合は無視されます。

この機能は Ver. 0.02 から使用できます。

●設定例 「2022/01/01,10:10:10」のような書式で、ふたつの列に分かれて付加される例です。

Frame[0]	
Node	1 ※
Function	DateTime
NodeView	False
FunctionView	False
Register	1 ※
Format	20@Y/@M/@D,@h:@m:@s
Value[0]	
Title	Date, Time タイトルに表示したい任意の文字列を指定してください
Type	Through
Endian	Big ※
WaitTime...	
Value	1
Unit	S

※これらの設定は無視されます。

●設定例 「10:10:10」のような書式で時刻だけ付加される例です。

Frame[0]	
Node	1 ※
Function	DateTime
NodeView	False
FunctionView	False
Register	1 ※
Format	@h:@m:@s
Value[0]	
Title	Time タイトルに表示したい任意の文字列を指定してください
Type	Through
Endian	Big ※
WaitTime...	
Value	1
Unit	S

※これらの設定は無視されます。

■ ■ 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。