

RsFile64-D Ver2.30

ヘルプファイル 目次

- ソフトの使用での注意点
- ソフトの概要
- 動作環境
- インストールの方法
- アンインストールの方法
- ファイルの構成
- ソフトウェア使用許諾契約
- ソフトの名前について
- シリアル通信のデバック操作方法
- ラインモニターの操作方法
- 使用する前に設定する事
- メイン画面の説明
- ツールバーの説明
- ステータスバーの説明
- メニューバーの説明
 - ファイル
 - LOGファイルの保存
 - LOGファイルの読込
 - LOGファイルの読込
- 表示
 - 再表示
 - 画面クリアへ
 - スクロールアップ
 - スクロールダウン
 - 最上位
 - 最下位
 - チェックサム
 - A S C I I
- 送信
 - 入力文字送信
 - 設定文字送信
 - マクロ実行
 - Modbus送信
- 送信文字編集
 - 設定文字の編集
 - 新規マクロ作成
 - マクロ文の編集
- COM設定
 - ポートの設定
 - O P E N
 - C L O S E
 - シリアル設定
 - B C D T I M E の設定
 - 状態表示

環境設定

環境設定

ダーク&ライトモード

setup of language

色設定

スタート文字・ストップ文字・END文字・チェックサムについて
Macro命令について

このソフトについて
バージョンについて
ご意見など連絡先

ソフトの使用でのご注意点

【ソフトの使用でのご注意点】

このソフトを起動した時点で上記を同意したとします。
同意されない方は、このソフトやダウンロードした内容を削除して下さい。

ご使用の前に、必ずこのページをお読みください。

このソフトは、企業ではなく個人で作成した物です。
個人で使用していましたが、期間限定で公開します。
動作保証はありません。各自で動作確認するようにお願いします。

使用条件及び免責について

御使用されるための契約として

本プログラムの使用にあたっては、使用者自身の責任の下に行ってください。

作者は何の保証もしませんし、本プログラムを使用したことによって生じた損害を補償する義務も負いません。

このソフトウェアを使用して個人または団体が直接あるいは、間接的な損害を受けたとしても作者は一切の責任を負いません。

このソフトを起動した時点で上記を同意したとします。
同意されない方は、このソフトやダウンロードした内容を削除して下さい。

ソフトの概要

「RsFile64-D」は、シリアル（RS232C, RS422, RS485）通信専用のデバッグソフト
シリアルは2CH使用可能でラインモニターにもなります。

このソフトの概略説明

他の装置とシリアル通信でデバックするのに役に立つソフトです。

特徴は、表示がラインモニター風で送信文字と受信文字を表示します。

表示する文字は上段にASC文字（0x20以下は制御コードを省略文字で表示）下段にHEX文字（2文字で表示）です。

送受信から次の送受信間の待機時間も表示するようになっています。

Ver2.1にあったテキスト表示は撤去しました。

「RsFile64-D」で必要な物

当たり前ですがパソコン（シリアルポートがある物）

デスクトップPCは、付いているPCもあれば、付いていないPCもあります。

ノートPCは、最近ではついていません。

付いていないPCについては、USBからRS232Cに変換する機器があるのでそれを使用するしかありません。

USBからRS422や485に変換する機器で動作します。

（Windows上で認識するもの）

例として下記のような製品です



PCにRS232Cのポートが無い方は、専門店・量販店やネット通販で購入してください。

* 御使用されるための契約として

本ソフトはフリーソフトです。

本プログラムの使用にあたっては、**使用者自身の責任の下に行ってください。**

作者は何の保証もしませんし、本プログラムを使用したことによって生じた損害を補償する義務も負いません。

動作環境

- 1・マシンはWindowsパソコンを使用する事
- 2・OSは下記のWindowsである事
Windows10以上の64bitOS（Windows11は動作確認済み）
- 3・記憶媒体に空きがある事
- 4・シリアルポートが付いている物

インストールの方法

インストールはありません、解凍して実行ファイルを起動するだけです。

一般的作業内容

- 1・管理しやすいところで新規にフォルダを作成する
- 2・そのフォルダ内に解凍
解凍ソフトは各自で準備する事。（Zipで圧縮しています）
- 3・ショートカットの作成
使いやすい場所にショートカットを作ってデスクトップなどに置いてください。

アンインストールの方法

アンインストールもありません、実行ファイルや設定ファイルを削除するだけです。

一般的作業内容

- 1・インストールで作成したフォルダを削除する
- 2・フォルダ以外に設定ファイルを保存しているのなら、それも削除する
- 3・インストール時に作成したショートカットを削除する

ファイルの構成

RsFile64-D.exe	実行ファイル
rsfile64d.ini	設定ファイル（環境設定）
rsfile64d_char2.ini	設定ファイル（文字列テーブル）
dkm.ini	設定ファイル（ダークモード用）
color.ini	設定ファイル（色テーブル）
mdata.txt	設定ファイル（送信文字列）
RsFile64-D_manual.pdf	マニュアル
RsFile64-Dドキュメント.txt	ドキュメント

ソフトウェア使用許諾契約

1・ライセンス

- 1) このソフトはフリーソフトとして公開しています。ご自由にご利用ください。

2・免責

- 1) 本ソフトウェアの使用にあたっては、使用者自身の責任の下に行ってください。作者は何の保証もしません。
- 2) 本ソフトウェアを使用したことによって生じた損害を補償する義務も負いません。
- 3) このソフトウェアを使用して個人または団体が直接あるいは間接的な損害を受けたとしても作者には一切の責任を負いません。
- 4) 本ソフトウェアをダウンロード、インストール、使用又は利用した結果、ハードウェア又はデータに支障が生じた場合等、本ソフトウェアに起因し又は関連して損害が発生した場合であっても、作者は一切責任を負いません。本ソフトウェアを複製し、組み込み又は改変したソフトウェア及びこれらを使用又は利用して作成されたソフトウェアについても同様とします。

3・対象ソフトウェア

- 1) 対象となるソフトウェアは、このダウンロードしたソフトウェアです。

4・権利

- 1) 本ソフトウェア及びマニュアルやドキュメントの著作権はBitDeviserにあります。

5・禁止事項

- 1) 本ソフトウェアの使用権を第三者に貸与、譲渡、リース、レンタル、サブライセンスすること。
- 2) 本ソフトウェアを改変したものをネットワーク上で配信すること。
- 3) 本作者名を名乗り勝手に配信やサービスを行うこと。
- 4) 無許可で他ネットへの転載，再配布，雑誌のCD-ROM 等への収録に関しては禁止します。

ソフトの名前について

「RsFile64-D」という名について

Rsは、RS232CやRS422やRS485のRs

Fileは、RS232CのWindows-API関数は、ファイルアクセスの関数と同じでFile

64はx64で構築したという意味です。

-DはデバッグのDという意味です

「RsFile64-D」は、**RS232C, RS422, RS485通信用のソフトです。**

詳細はRS232C通信で**デバッグが便利**になるソフトで、また**ラインモニター**の機能を持っています。

RS232C通信とは

パソコンと外部機器を通信を行うで、1 B i t の情報を時間差シリアルで転送する方法です。
有線接続で差動でなく電圧のON/OFF信号でクロック用の配線もないので伝送距離が短く
伝送速度も遅くノイズに弱いです。

但し、産業用の機器間通信でよく使っています。

RS232C通信とよく似たもの

RS422通信

RS232Cを差動式に変えたもので伝送速度・伝送距離が良くなっています

RS485通信

RS232CとRS422は1対1で通信が目的ですが、RS485はn対nを目的として使用します。

RS422でも1対nは可能ですよ。

RS422は4〜5本の線に対して、RS485は2〜3本の線になります。

半2重方式しか通信出来ません。

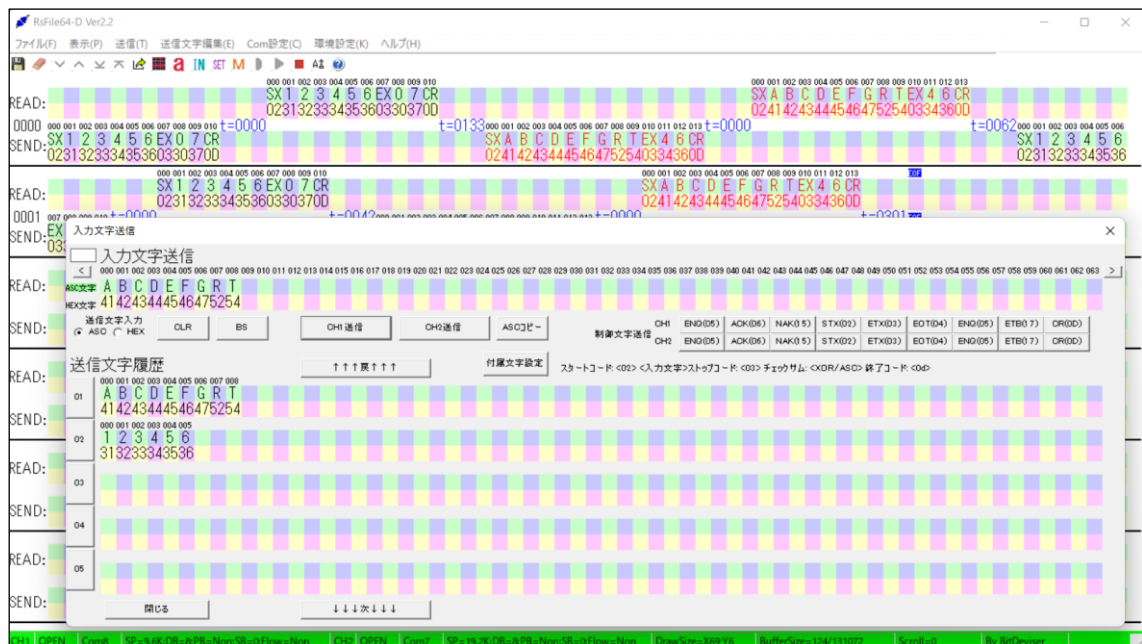
LAN通信

差動式で、全2重式でクロック情報があるので、伝送速度は飛躍的に早くなります。

光ファイバー通信

1本であれば、半2重式でクロック情報がなく、光の残像問題があり、思ったより
早くはないが伝送距離やノイズに関しては飛躍的よい。

シリアル通信のデバック操作方法



<送信操作>

送信する方法は、下記の3パターンがあります。

- ① キーボードで入力した文字を送信する方法
- ② あらかじめテキストで設定された文字を送信する方法
- ③ 制御文字の送信する方法
- ④ マクロ命令による自動で送受信方法

① 入力文字送信

キーボードで入力した文字を送信する方法

操作方法

メニューバーの送信の入力文字送信をクリックします。

またはツールバーの「IN」をクリックします。



0～9・A～Zのキーを押すと一番上のシマシマのところに文字が入ります。

入力文字はASC入力とHEX入力がありRadioButtonで切替が可能です。

HEX入力は2文字で1文字になります。例31と入力したらASC文字の1になる。

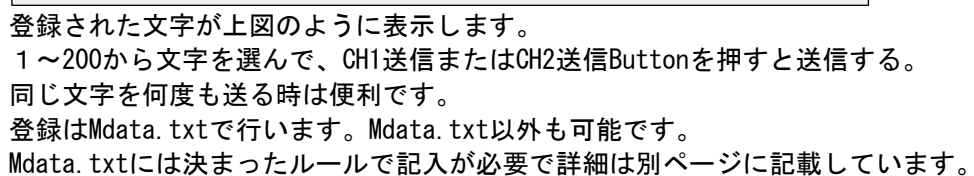
CH1送信またはCH2送信を押すとそのCHからその文字を送信します。

一度送信したら下の段の1～20にその送信文字が格納し履歴表示します。

1～20のボタンを押せば、その文字が上の段に移行し再送信が便利です。

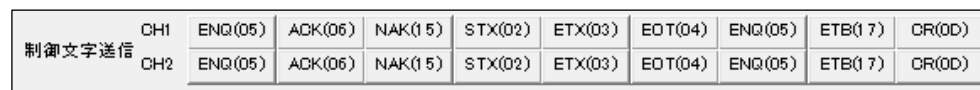
あらかじめテキストで設定された文字を送信する方法

あらかじめメモ帳などのテキストエディタで文字を登録します。
記入（登録）には決まったルールがあります。詳細は別ページに記載済み
メニューバーの送信の設定文字送信をクリックします。
またはツールバーの「**SET**」をクリックします。



制御文字の送信する方法

メニューバーの送信の制御文字送信をクリックします。
またはツールバーの「**IN**」をクリックします。



0x00～0x1Fの文字は制御文字でよく使う物をボタンで用意しました。
制御文字のボタンを押せ、その制御文字を送信します。
CH1とCH2はそれぞれのボタン毎に分かれています。

④ マクロ実行

マクロ命令による自動送受信方法

あらかじめメモ帳などのテキストエディタでマクロ文を作って登録します。
その登録した内容で受信や送信や判断・比較などを行い自動で送受信します。

操作方法

あらかじめメモ帳などのテキストエディタでマクロ文を作って登録します。
マクロ文には命令文があり、それにはルールがあります。
マクロ文の詳細は別ページに記載があります。

メニューバーの送信のマクロ実行をクリックします。

またはツールバーの「**M**」をクリックします。



「・」のボタンでマクロ文のテキストファイルを選びます。

マクロ実行のボタンを押すとマクロが実行します。

中断する時は中断のボタンを押します。

受信中（受信命令の設定時間）は操作が無効になります。

受信命令の時間設定は長くしないでほしい。

マクロ文には命令文があり、それにはルールがあります。

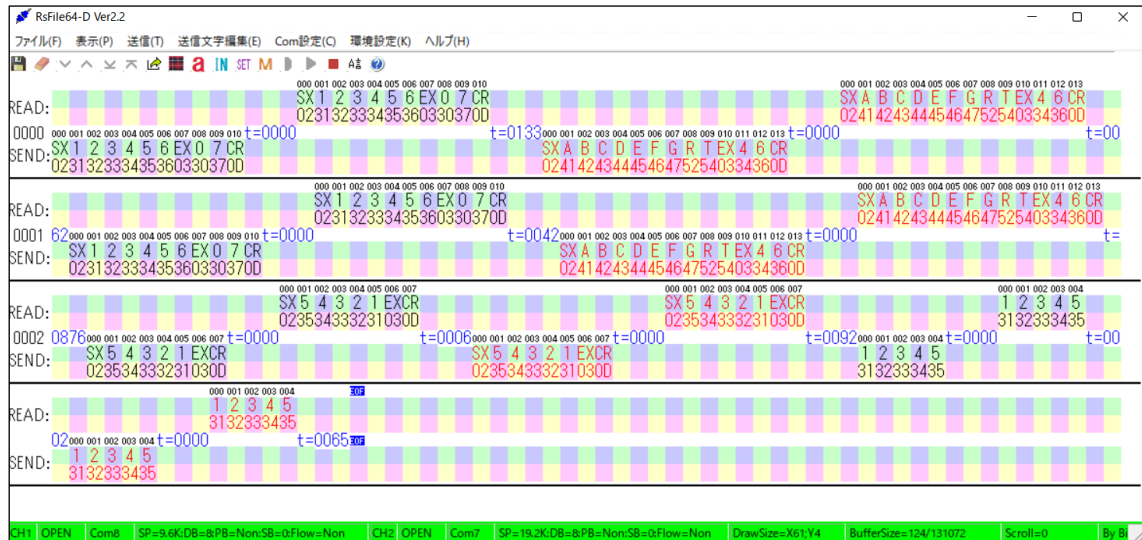
マクロ文の詳細は別ページに記載があります。

<受信操作>

特に操作はありません。

受信されれば、勝手にメイン画面の「READ」覧に表示します。

画面表示（ラインモニタ風）下図参照



READ(上)とは、受信した文字を表示します。

SEND(下)とは、送信した文字を表示します。

黒色の横ラインは行の区切りです。

READとSENDの間の4桁に数値はスクロール番号です。

黒文字は、CH1のポートで送受信した文字を表示します。

赤文字は、CH2のポートで送受信した文字を表示します。

上段（小さなフォント）の3桁の数値は送受信の文字数を記載

中段（大きなフォント）の1文字はASC文字で表示

中段（大きなフォント）の2文字は制御文字を省略文字で表示

下段（大きなフォント）の2文字はHEX文字で表示

送受信から次の送受信間の待機時間はt=0000で表示（単位は0.1秒）

「EOF」最後であることを表示しています。

COMポートがOPEN中はスクロールしないで右下まで文字表示したら、その続きは左上に表示します。

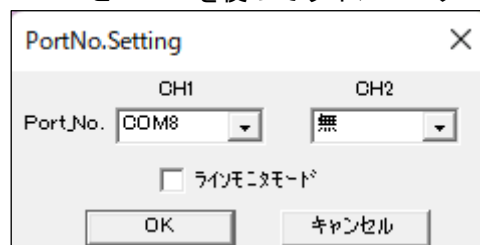
OPEN中は最新の送受信から決まった数量（1画面分）しか表示しません。

COMポートがCLOSEは送受信した全体が表示可能で、スクロールして表示します。

送受信した文字はバッファに記憶し、CLOSE中はそれを読出して表示します。

ラインモニターの操作方法

CH 1 と CH 2 を使ってラインモニターになります。

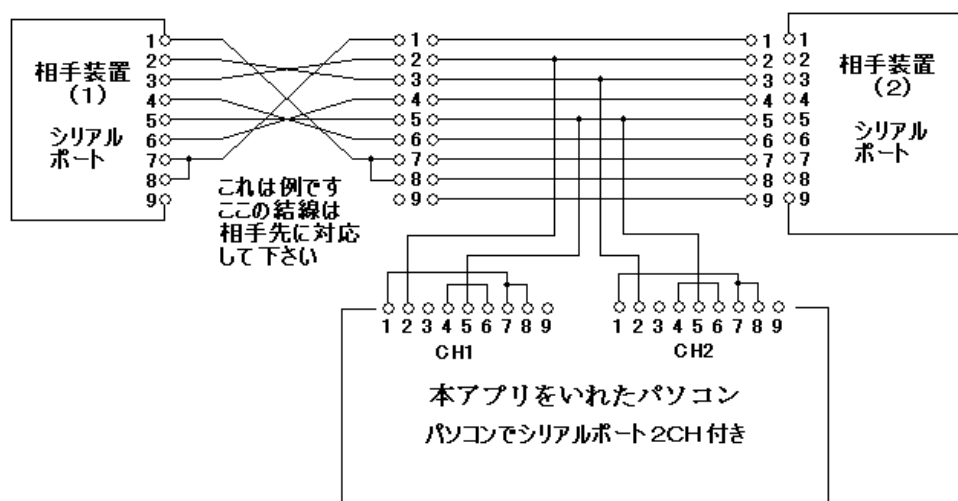


上記のようにCH 1 と CH 2 のポート番号を設定後、ラインモニターにチェックを入れます。

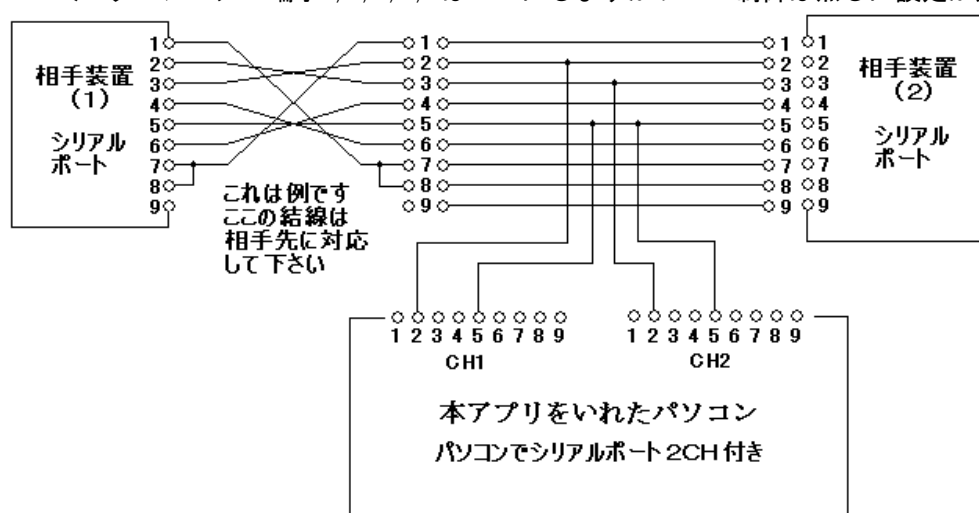
下記の T 型中継ケーブルを準備します。

結線図は下記のように 2 パターンあります。

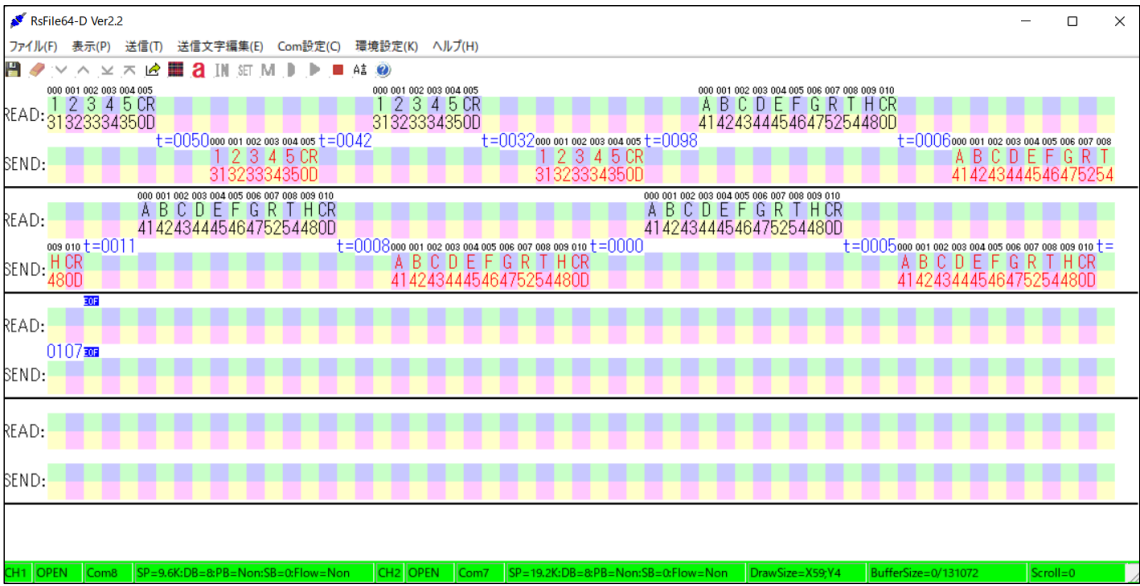
<パターン 1> 端子 1, 4, 6, 7, 8 はジャンパします。



<パターン 2> 端子 1, 4, 6, 7, 8 は OPEN にしますがフロー制御は無しに設定が必要。



画面のイメージで下記のようにラインモニターを行います。



ラインモニターへ切替、またはラインモニターからの切替を行う場合は、画面クリアを実行して下さい。
画面クリアをしなかった場合は、スクロール操作で受信文字と送信文字の表示がおかしくなります。

使用する前に設定する事

設定する前に、以下の事を行ってください。

「RsFile64-D」で必要な物

- ① 当たり前ですがパソコン（シリアルポートがある物）
- ② COMポートの付いていないPCはUSBシリアル変換器を準備する
COMポートはWindowsの標準APIで動作する物です。

<ポートの準備>



最近のノートPCにはRS232Cは付いていません。

上図のような、USB-RS232C変換器を準備して下さい。（市販品であります）

USB-RS232C変換器には専用のドライバーが必要になると思います。

USB-RS232C変換器のメーカーからダウンロードしてインストールを行ってください。

<ポートの設定>

メニューバーのCOM設定のポートの設定で設定を行います。

またはツールバーの「」をクリックします。

<OPEN処理>

メニューバーのCOM設定のOPENで行います。

またはツールバーの「」をクリックします。

<シリアルの設定>

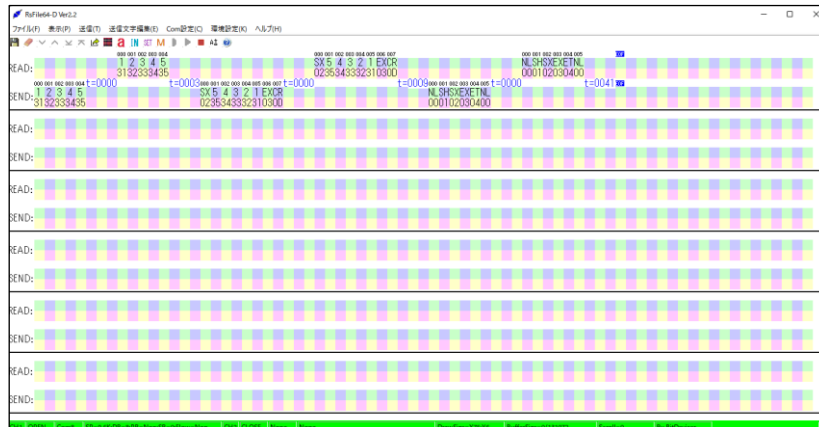
ボーレート・データビット・ストップビット・チェックサムなどの設定はシリアル設定で行い、COM1～255などの設定はポートの設定で行います。

CH1とCH2それぞれ設定する必要があります。

メニューバーのCOM設定のシリアル設定で行います。

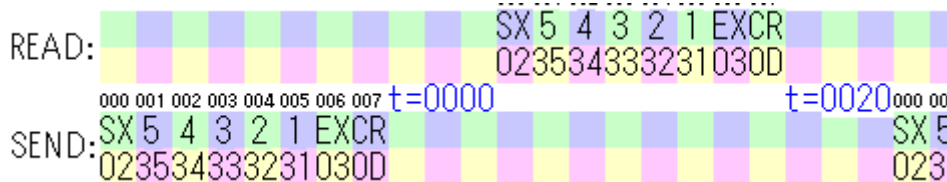
これで準備完了

メイン画面の説明



メニューバー・ツールバー・ステータスバーを準備しています。

文字の表示



上部：READは受信した内容を表示します。

下部：SENDは送信した内容を表示します。

READ部の上段の小文字は受信した文字数を表示します。

READ部の中段の大文字は受信した文字をASCで表示します。

READ部の下段の大文字は受信した文字をHEXで表示します。

SEND部の上段の小文字は送信した文字数を表示します。

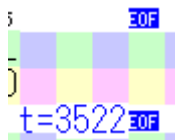
SEND部の中段の大文字は送信した文字をASCで表示します。

SEND部の下段の大文字は送信した文字をHEXで表示します。

T=0000は送受信から次の送受信間の待機時間を表示します。単位は0.1秒単位です。

バックの色で1文字分の区切りをおこなっています。

0x20以下の制御文字のASC文字は制御文字省略文字を使用しています。



のEOFは送受信の最終位置を示します。

行の罫は横の線で区別しています。

OPEN時は送受信した文字を左から右へ表示します。

その行がいっぱいなら下の行へ表示が移ります。

一番下の行でその行がいっぱいになったら、左上の位置から表示します。

絵画している行の文字は消されますが、その他の行はそのまま残ります。


CLOSE時はスクロール機能が使えて、すべての送受信した文字の表示が可能です。


メニューバーから、またはツールバーのボタンを押してスクロールして下さい


スクロールは↑・↓キーでも対応しています。


ツールバーの説明





 LOGファイルの保存：送受信のバッファ内容をLOGファイルとして保存する


 LOGファイルの保存：送受信のバッファ内容のLOGファイルを読み出す


 画面クリア：画面表示のクリアと送受信のバッファ内容もクリアします。


 スクロールダウン：画面スクロール

 スクロールアップ：画面スクロール


 最下限：画面表示スクロールで最下限位置へ移動


 最上位：画面表示スクロールで最上位位置へ移動


 再表示：画面の再表示します。

 チェックサム：文字列をチェックサムの計算する


 ASCII：ASCII文字の一覧表

 入力文字送信：入力した文字を送信する


 設定文字送信：登録した文字を送信する

 マクロ実行：マクロ文にしたがって自動送受信する


 Modbus 送信：Modbus通信に特化した機能


 ポートの設定：CH1/2にCOM番号を割り当てる


 OPEN：シリアルポートのオープン処理


 CLOSE：シリアルポートのクローズ処理


 CH1シリアル設定：CH1のボーレートなどの設定


 CH2シリアル設定：CH2のボーレートなどの設定


 設定文字送信の編集：mdata.txtをメモ帳で編集


 新規マクロ作成：メモ帳を起動してマクロ文を作成する

 マクロ文の編集：メモ帳を起動してマクロ文を編集する

 Setup of language：日本語・英語・韓国語・中国語に切替

 ダークモードとライトモードの切替

 ヘルプ

 アプリケーション終了

各詳細については、別ページを参照

ステータスバーの説明

CH1	CLOSE	Com8	SP=9.6K;DB=8;PB=Non;SB=0;Flow=Non	CH2	CLOSE	None	None	DrawSize=X67;Y4	BufferSize=0/131072	Scroll=0	By BitDeviser
-----	-------	------	-----------------------------------	-----	-------	------	------	-----------------	---------------------	----------	---------------

- 左から 1 番め
CH1 （固定文字）
- 左から 2 番め
OPEN/CLOSEを表示 CH1のOPEN/CLOSEの状態を表示
- 左から 3 番め
CH1のシリアル設定の状態を表示
SP=ボーレート（速度） DB=データビットの長さ
PB=パリティビット SB=ストップビットの長さ
Flow=フロー制御方法
- 左から 5 番め
CH2（固定文字）
- 左から 6 番め
OPEN/CLOSEを表示 CH2のOPEN/CLOSEの状態を表示
- 左から 7 番め
CH2シリアル設定の状態を表示
SP=ボーレート（速度） DB=データビットの長さ
PB=パリティビット SB=ストップビットの長さ
Flow=フロー制御方法
- 左から 8 番め
画面で文字が表示できる横の文字数と縦の文字数の表示
- 左から 9 番め
バッファサイズで使用数／最大数の表示
- 左から 1 0 番め
スクロールの位置で表示開始するバッファポイントの表示
- 左から 1 1 番め
作者名の表示

CH1とCH2両方がCLOSE時のステータスバーの色

CH1	CLOSE	Com8	SP=9.6K;DB=8;PB=Non;SB=0;Flow=Non	CH2	CLOSE	None	None	DrawSize=X67;Y4	BufferSize=0/131072	Scroll=0	By BitDeviser
-----	-------	------	-----------------------------------	-----	-------	------	------	-----------------	---------------------	----------	---------------

CH1のみOPEN時のステータスバーの色

CH1	OPEN	Com5	SP=9.6K;DB=8;PB=Non;SB=0;Flow=Non	CH2	CLOSE	None	None	DrawSize=X75;Y6	BufferSize=0/131072	Scroll=0	By BitDeviser
-----	------	------	-----------------------------------	-----	-------	------	------	-----------------	---------------------	----------	---------------

CH2のみOPEN時のステータスバーの色

CH1	CLOSE	None	None	CH2	OPEN	Com5	SP=19.2K;DB=8;PB=Non;SB=0;Flow=Non	DrawSize=X75;Y6	BufferSize=0/131072	Scroll=0	By BitDeviser
-----	-------	------	------	-----	------	------	------------------------------------	-----------------	---------------------	----------	---------------

CH1とCH2両方がOPEN時のステータスバーの色

CH1	OPEN	Com6	SP=9.6K;DB=8;PB=Non;SB=0;Flow=Non	CH2	OPEN	Com5	SP=19.2K;DB=8;PB=Non;SB=0;Flow=Non	DrawSize=X75;Y6	BufferSize=0/131072	Scroll=0	By BitDeviser
-----	------	------	-----------------------------------	-----	------	------	------------------------------------	-----------------	---------------------	----------	---------------

CH1とCH2両方がOPENでラインモニタの時のステータスバーの色

CH1	OPEN	Com5	SP=9.6K;DB=8;PB=Non;SB=0;Flow=Non	CH2	OPEN	Com6	SP=19.2K;DB=8;PB=Non;SB=0;Flow=Non	DrawSize=X75;Y6	BufferSize=0/131072	Scroll=0	By BitDeviser
-----	------	------	-----------------------------------	-----	------	------	------------------------------------	-----------------	---------------------	----------	---------------

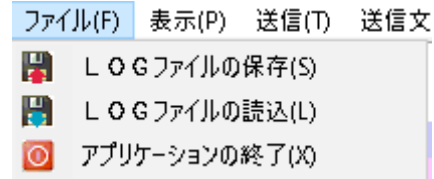
メニューバーの説明

<メニューバー>



メニューはファイル・表示・送信・送信文字編集・Com設定・環境設定・ヘルプがあります。

<ファイル>



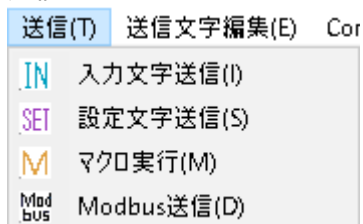
ファイルの中にはLOGファイルの保存・LOGファイルの読込・アプリケーション終了があります。

<表示>



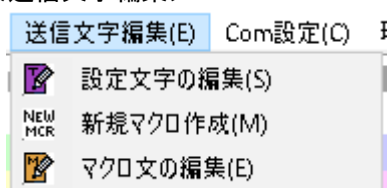
表示の中には再表示・画面クリア・スクロールアップ・スクロールダウン・最上位・最下位・チェックサム・ASCIIがあります。

<送信>



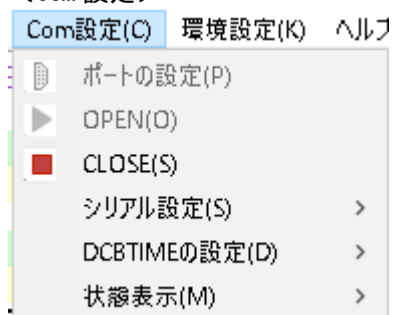
送信の中には入力文字送信と設定文字送信とマクロ実行とModbus通信があります。

<送信文字編集>



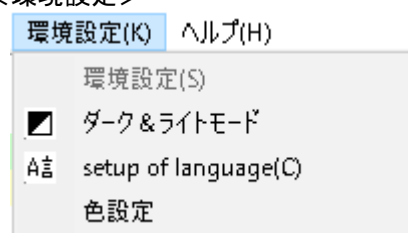
送信文字編集の中には設定文字の編集と新規マクロ作成とマクロ文の編集があります。

<Com 設定>



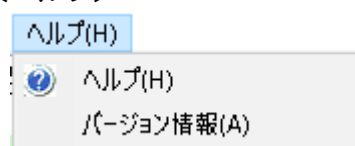
Com 設定の中にはポートの設定・OPEN・CLOSE・シリアル設定・DCB TIME の設定・状態表示・があります。

<環境設定>



環境設定の中には環境設定・Setup of languageがあります。

<ヘルプ>



ヘルプの中にはヘルプ・バージョン情報があります。

<ファイル>

LOGファイルの保存

この機能は送受信した内容（画面の内容）をバッファに記憶しています。
そのデータをファイルに保存する機能です。
ファイルの拡張子はLOGで保存します。

LOGファイルの読出

この機能はLOGファイルを開く機能です。
LOGファイルを保存した内容を読み出します。
ファイルの拡張子がLOGを選択して読み出します。

アプリケーションの終了

メニューのファイルの中にあります。
このアプリを終了します。

<表示>

再表示

これを実行すると画面が再表示します。

表示クリア

これを実行すると「はい／いいえ」メッセージBOXが出て、「はい」をクリックすると、画面の送受信の文字がクリアされます。
また、記憶したバッファもクリアします。

スクロールアップ

これをクリックすると画面がスクロールアップします。
スクロールする幅は環境設定で出来ます。
OPEN中は操作出来ません、CLOSEしてから操作出来ます。

スクロールダウン

これをクリックすると画面がスクロールダウンします。
スクロールする幅は環境設定で出来ます。
OPEN中は操作出来ません、CLOSEしてから操作出来ます。

最上位

これをクリックすると最初の行へスクロールします。
OPEN中は操作出来ません、CLOSEしてから操作出来ます。

最下位

これをクリックすると最後の行へスクロールします。
OPEN中は操作出来ません、CLOSEしてから操作出来ます。

チェックサム

この機能は設定文字送信などで文字列を設定するときに便利です。

- 1・キーボードでキー入力して文字を入れます。
上段にはASC文字、下段にはHEX文字（2文字で表現）
ペーストも出来ます。
- 2・チェックサムの種類を選びます。B U T T O Nを押して切り替えてください。
 - 1) XOR
 - 2) ADD
 - 3) RCR16
 - 4) LRC
- 3・計算のB U T T O Nで入力された文字がチェックサムします。
結果はボタンの右横に表示します。
結果はASCコピーまたはHEXコピーボタンでコピーして、別の場所にペースト出来ます。

ASCII

アスキー表を表示します。

ASCII表							
00: NUL	10: DIE	20: SP	30: 0	40: @	50: P	60: `	70: p
01: SOH	11: DC1	21: !	31: 1	41: A	51: Q	61: a	71: q
02: STX	12: DC2	22: ~	32: 2	42: B	52: R	62: b	72: r
03: ETX	13: DC3	23: #	33: 3	43: C	53: S	63: c	73: s
04: EOT	14: DC4	24: \$	34: 4	44: D	54: T	64: d	74: t
05: ENQ	15: NAK	25: %	35: 5	45: E	55: U	65: e	75: u
06: ACK	16: SYN	26: &	36: 6	46: F	56: V	66: f	76: v
07: BEL	17: ETB	27: '	37: 7	47: G	57: W	67: g	77: w
08: BS	18: CAN	28: (38: 8	48: H	58: X	68: h	78: x
09: TAB	19: EM	29:)	39: 9	49: I	59: Y	69: i	79: y
0A: LF	1A: SUB	2A: *	3A: :	4A: J	5A: Z	6A: j	7A: z
0B: VT	1B: ESC	2B: +	3B: ;	4B: K	5B: [6B: k	7B: {
0C: FF	1C: FS	2C: ,	3C: <	4C: L	5C: ¥	6C: l	7C:
0D: CR	1D: GS	2D: -	3D: =	4D: M	5D:]	6D: m	7D: }
0E: SO	1E: RS	2E: .	3E: >	4E: N	5E: ^	6E: n	7E: ~
0F: SI	1F: US	2F: /	3F: ?	4F: O	5F: _	6F: o	7F: ¨

ちょっと見る時に便利だと思います。

0x00~0x20までのところをマウスでクリックすると説明文が表示します。

01: SOH	11: DC1	21: !	31
02: STX	12: DC2	22: ~	32
03: ETX	13: DC3	23: #	33

例 02のところをマウスでクリック「SX: テキスト開始」と表示します。
SXはこのアプリのASC表示で省略文字です。

入力文字送信

このダイアログは、入力した文字を送信します。

送信方法の手受信

1・文字の入力

0~9・A~Zのキーを押すと一番上のシマシマのところに文字が入ります。
0x00~0x1Fまでの制御文字を入れる場合は、送信文字入力でHEXにします。
ASC入力とHEX入力はRadioButtonで切替が可能です。
HEX入力は2文字で1文字になります。例31と入力したらASC文字の1になる。
64文字以上の文字を入力した場合は「<」「>」のボタンで表示をスライド出来ます。
1文字削除する時は「BS」ボタンを押します。
すべての文字を削除する時は「CLR」ボタンを押します。
入力した文字はASCコピーボタンを押すとコピーされ、別の場所にペースト出来ます。
画面のFocusがないと文字入力出来ません。
自動（一定周期）でFocusを合わすようにしています。
Focus合わない時は画面の左上の口をクリックして下さい。

2・付属文字設定

入力文字の前後にヘッダーや制御文字や終了コードを付けることが出来ます。
「付属文字設定」ボタンを押して設定します。
設定した結果はボタンの右側に表示します。
スタートコード: <>の中が設定した内容
<入力文字>キー入力した文字の内容
ストップコード: <>の中が設定した内容
チェックサム: <>の中が設定した内容 (HEXまたはASCで選べます)
終了コード: <>の中が設定した内容

付属文字設定の設定方法

必要な物はチェックBOXにチェックを入れて下さい。
コードはそのEditboxに入れます。HEXで必ず2文字入力します。
チェックサムはComboboxでXOR・ADD・CRC16・LRCを選びます。

チェックサムは結果はASC文字、HEX文字を選ぶ事が出来ます。

3・送信

「CH1送信」または「CH2送信」のボタンを押すと入力文字+付属文字で送信します。

送信した結果は、メイン画面のラインモニタ風画面に表示します。

又、送信した文字は下の画面に履歴として表示します。

例：入力文字「123」でスタート文字が02、ストップ文字が03で終了コード無しの場合
HEXで 02, 31, 32, 33, 03 を送信します。

4・制御文字の送信

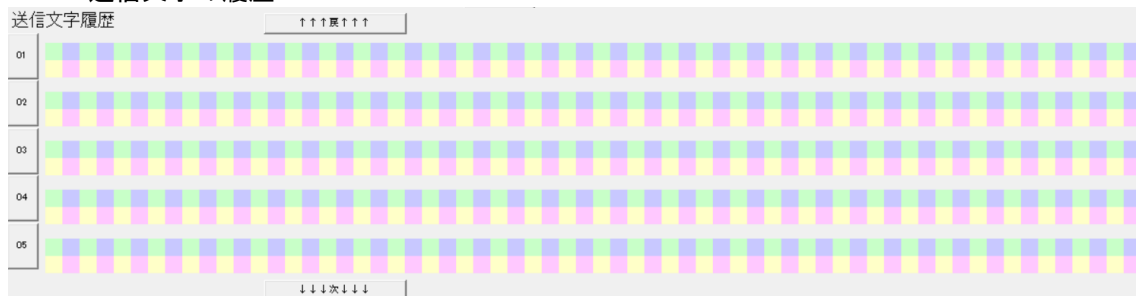
制御文字送信	CH1	ENQ(05)	ACK(06)	NAK(15)	STX(02)	ETX(03)	EOT(04)	ENQ(05)	ETB(17)	CR(0D)
	CH2	ENQ(05)	ACK(06)	NAK(15)	STX(02)	ETX(03)	EOT(04)	ENQ(05)	ETB(17)	CR(0D)

上記のボタンを押すとその制御文字が送信します。

この時は付属文字は付きません。

例：ACK(06)のボタンを押すとHEXで 06 を送信します。

5・送信文字の履歴



送信した文字は20個まで履歴として残します。

20件以上は古い文字から切り捨てます。

画面上は5件までしか表示しませんので、↑・↓ボタンで5件毎にスライドして下さい。

長い文字の場合は先頭からの文字を表示して画面からはみ出す文字は見えません。

長い文字のスライド機能はありません。

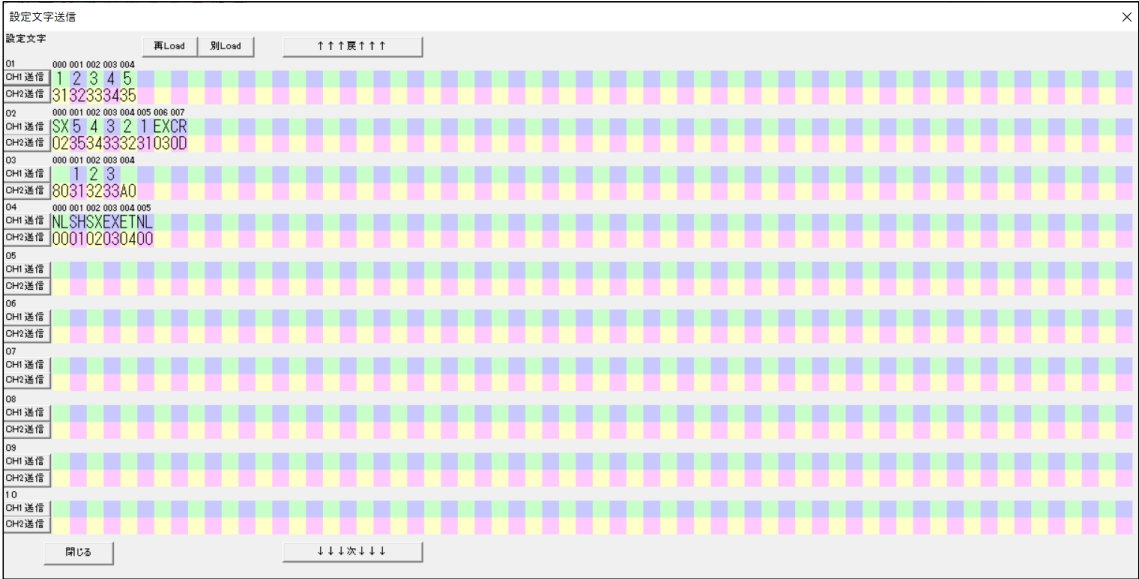
履歴文字を再度送信する場合は

01～20のボタンを押すと、上の入力文字とのところにその文字が入ります。

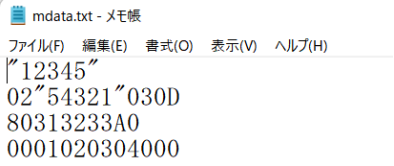
「CH1送信」または「CH2送信」のボタンを押すと入力文字+付属文字で送信します。

設定文字送信

このダイアログは、登録した文字を送信します。



登録する文字をMdata.txtというファイル名で、メモ帳で作成します。



ASC文字は「 “ ” 」ダブルコーテーションで前後で括ってください。

HEX文字は2文字で1文字になります。

例 1 “123” の場合ASC文字の123で、HEXでは31, 32, 33になります。

例 2 02” 54321” 030Dは、HEXでは02, 35, 34, 33, 32, 31, 03, 0Dになります。
(” 54321” はASC文字です)

例 3 80313233A0は、そのままHEXで80, 31, 32, 33, A0になります。

Mdata.txtのファイルを読んで文字が上図のように表示します。

画面には10件しか表示しません。それ以上は↓・↑ボタンで10件毎に切り替えてください。

最大256文字で200件まで登録出来ます。

この画面とメモ帳を両方開き、メモ帳を編集保存した後に「再LOAD」ボタンを押すと画面に反映します。

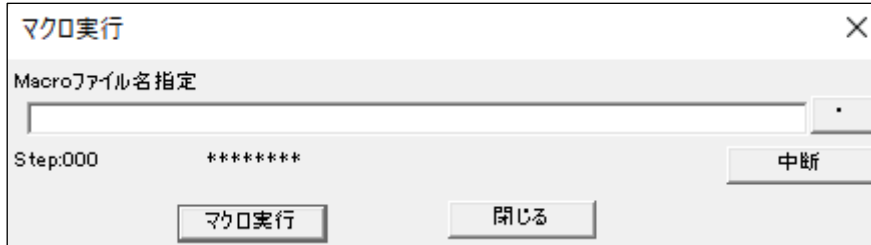
また、Mdata.txt以外のファイルを選ぶ時は「別LOAD」ボタンを押すと他のファイルも選べます。
基本はMdata.txtです。起動時はMdata.txtを読み込みます。

「CH1送信」または「CH2送信」のボタンを押すとその文字が送信します。

Macro実行

メニューの送信の中にあります。

このダイアログは、あらかじめ設定されたマクロ文を選択し実行させるところです。



ファイルを実行する度に選択してそのマクロファイルを実行する方法です。

<画面の説明>

上段部分はマクロファイルを選択したパスを表示と選択

中段部分はマクロ文のステップ情報と命令文を表示します

下段部分はマクロ実行時のボタン操作部分

- 1・あらかじめマクロを作成します。（環境設定のMacro新規／編集で登録して下さい）
- 2・「ファイルを開く」のボタンを押してマクロファイルを選んで下さい。
- 3・マクロファイルが選ばれたらMacro Fileのところ選ばれたファイル名が表示します。
- 4・マクロ実行のボタンを押します。
- 5・マクロが実行されているときは、Macro Stepが変化し、処理状態を表示します。
- 6・マクロが実行が終了すれば、処理完と表示されます。

<注意事項>

マクロ文の組み方によっては無限ループに入ることがあります。

十分にデバックしてからご使用してください。

特に受信コマンドは受信されないと次へ行きません。

<Step表示>

Macro Stepに数字が表示するのはMacro実行中のステップ番号です。

<閉じる>

このBUTTONを押すとのダイアログが閉じます。

Macro実行中はこのダイアログを閉じる事は出来ません。Macroを終了にしてから閉じて下さい。

<clock>

このBUTTONを押すとマクロ実行時の1命令の時間設定する画面に行こうします。

<注意事項>

マクロの一度に使える最大文字数は256文字です。

マクロ文は256×100文字以内で作成してください。

Modbus送信

メニューの送信の中にあります。

このダイアログは、Modbus ASCIIとModbus RTUに基づいて送信するものです。

Modbus(ASCII)送信

開始文字: 3A スレーブアドレス: 01 ファンクション: 0x08:強制の診断 アドレス: 0123 点数: 0001 データ: FFFF

組立 CH1送信 CH2送信 閉じる

<画面の説明>

Modbus ASCIIとModbus RTUを切り替えるRADIOボタン

Modbus RTUを選択すると開始文字と終了コードの文字が灰色になります。

下部の開始文字・スレーブアドレス・ファンクション・アドレス・点数・データ・終了コードなどの設定を行います。

上部のシマシマ模様での部分は、下部の設定に基づき組立された送信文字の表示

<Modbus ASCII時の操作の説明>

- 1・上部のModbus ASCIIのRADIOボタンの押して設定します。
- 2・開始文字は3A「:」で固定です。
- 3・スレーブアドレスを文字で設定します。
- 4・ファンクションはComboBoxで選びます。
- 5・アドレスを数字で設定します。(ファンクションによって使用不可になります)
- 6・点数を数字で設定します。(ファンクションによって使用不可になります)
- 7・データを数字で設定します。(ファンクションによって使用不可になります)
- 8・終了コードは0D0A「CRLF」で固定です。
- 9・組み立てのボタンを押すと上記の内容を反映してシマシマ部分に送信文字が表示します。
- 10・CH1送信またはCH2送信のボタンを押すと送信します。

例 1

Modbus(ASCII)送信

開始文字: 3A スレーブアドレス: 01 ファンクション: 0x04:入力レジスタ読込/Word アドレス: 0123 点数: 0001 データ: FFFF

組立 CH1送信 CH2送信 閉じる

例 2

Modbus(ASCII)送信

開始文字: 3A スレーブアドレス: 01 ファンクション: 0x10:連続レジスタ書込/Word アドレス: 0123 点数: 0001 データ: FFFF

組立 CH1送信 CH2送信 閉じる

<M Modbus RTU時の操作の説明>

- 1・上部のModbus RTUのRADIOボタンの押して設定します。
- 2・開始文字は使用不可になります。
- 3・スレーブアドレスを文字で設定します。
- 4・ファンクションはComboBoxで選びます。
- 5・アドレスを数字で設定します。（ファンクションによって使用不可になります）
- 6・点数を数字で設定します。（ファンクションによって使用不可になります）
- 7・データーを数字で設定します。（ファンクションによって使用不可になります）
- 8・終了コードは使用不可になります。
- 9・組み立てのボタンを押すと上記の内容を反映してシマシマ部分に送信文字が表示します。
- 10・CH1送信またはCH2送信のボタンを押すと送信します。

例 1

Modbus

Modbus(RTU)送信

Modbus ☐ ASC ☒ RTU

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036

ASCII文字 SHETNL[NLSHA 0

HEX文字 0104007B000141D3

開始文字: []
HEX文字: []

スレーブアドレス: 01
ASCII文字: []

ファンクション: 0x04:入力レジスタ読込/Word

アドレス: 0123
数値入力: []

点数: 0001
数値入力: []

データー: FFFF
数値入力: []

☒ ASC ☐ HEX

組立 CH1送信 CH2送信 閉じる

例 2

Modbus

Modbus(RTU)送信

Modbus ☐ ASC ☒ RTU

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036

ASCII文字 SHDENL[NLSHSH y y] <<

HEX文字 0110007B000101FFFF5DAB

開始文字: []
HEX文字: []

スレーブアドレス: 01
ASCII文字: []

ファンクション: 0x10:連続レジスタ書込/Word

アドレス: 0123
数値入力: []

点数: 0001
数値入力: []

データー: FFFF
数値入力: []

☒ ASC ☐ HEX

組立 CH1送信 CH2送信 閉じる

<注意事項>

- スレーブアドレスはASCII文字で2文字設定してください。
- 0～Fの16進数で設定してください。（自動修正しません）
- アドレスは数値で入力してください。0～9の10進数で4桁以内に設定してください。
- 点数は数値で入力してください。0～9の10進数で4桁以内に設定してください。
- 送信する前に組立のボタンを押して送信文字を確認してから送信してください。

Com Port 設定

COMポートの設定

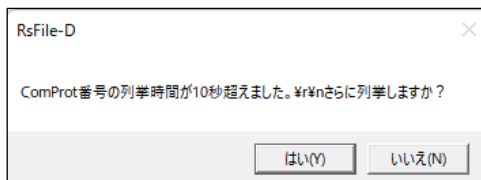


このダイアログでCH1とCH2のシリアルポートの番号を設定します。

CH1とCH2のComboBoxには使用可能なCOM番号が表示します。

このダイアログを開いた状態でUSBシリアル変換器をつないだ場合は、再度このダイアログを開き直してください。ダイアログが開くときに使用可能なシリアル番号を調査します。

使用可能なシリアルポートがない場合は5秒間列挙し下記のメッセージが出ます。

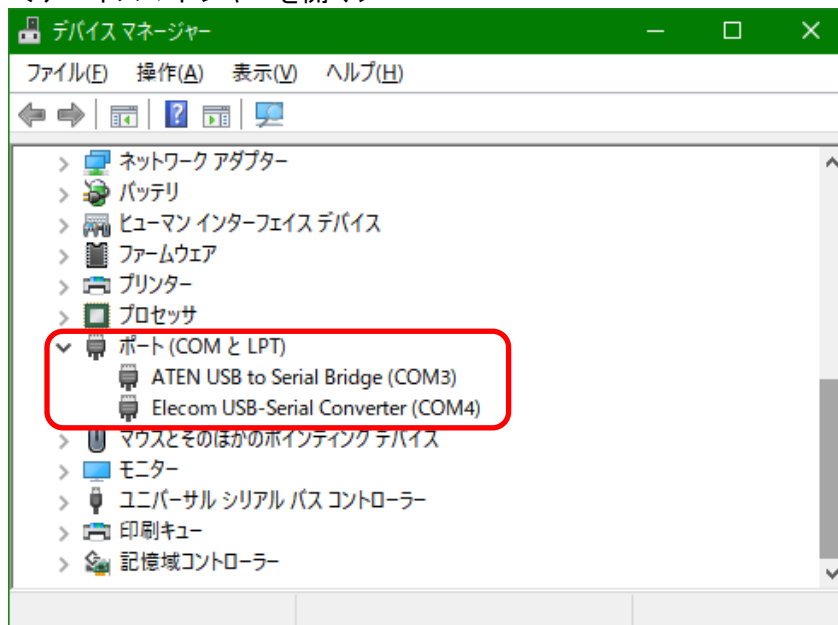


下記のメッセージが出たら「いいえ」をクリックして下さい。

使用可能なシリアルポートがない場合は何回列挙してもないと思います。

どのポートにするかは、システムのデバイスマネジャーで調べることも出来ます。

<デバイスマネジャーを開く>



設定方法

- 1・CH 1のシリアルポートを上図のCH 1のコンボボックスで設定します。
無を設定した場合はCH 1は使用しないことになります。
- 2・CH 2のシリアルポートを上図のCH 1のコンボボックスで設定します。
無を設定した場合はCH 1は使用しないことになります。
CH 1とCH 2で同じシリアルポートの番号は指定出来ません。
- 3・ラインモニタ機能
CH 1とCH 2を使用してラインモニターになることが出来ます。
ラインモニターにするときは「ラインモニタモード」にチェックを入れて下さい。
ラインモニターするにはT接続のケーブルを準備して下さい。
T接続のケーブルは各自で制作して下さい。
詳細はラインモニターの操作方法を参照して下さい。
- 3・「OK」BUTTONで設定された内容が反映されこのダイアログを閉じます。
- 4・「閉じる」のボタンでこのダイアログ閉じます。

CH1 Serial 設定

このダイアログは、CH1のボーレート・データビット・ストップビット・パリティ・制御方法を設定出来ます。

また、OPENしていないと開きません。



The image shows a Windows-style dialog box titled "COM8のプロパティ" (COM8 Properties). It has a tab labeled "ポートの設定" (Port Settings). Inside the dialog, there are five dropdown menus for configuring serial port settings:

- ビット/秒(B): 9600
- データビット(D): 8
- パリティ(P): なし
- ストップビット(S): 1
- フロー制御(F): なし

Below these settings is a button labeled "既定値に戻す(R)" (Restore Defaults). At the bottom of the dialog are three buttons: "OK", "キャンセル" (Cancel), and "適用(A)" (Apply).

- 1・ボーレートはCOMBOBOXで設定します。
設定は75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400ボーレートです。
- 2・データビットもCOMBOBOXで設定します。
設定は8ビットと7ビットの2種類です。
6ビット・5ビットは対応していません。
- 3・ストップビットもCOMBOBOXで設定します。
設定は1ビットと1.5ビットと2ビットの3種類です。
0ビットは対応していません。
- 4・パリティもCOMBOBOXで設定します。
設定は「無し」と「奇数ビット」と「偶数ビット」の3種類です。
これ以外の設定は対応していません。
- 5・制御方法もCOMBOBOXで設定します。
設定はOFFとXONXOFF制御と制御線で制御するDTR制御の3種類です。
- 6・OKのBUTTONで設定された内容が反映します。
- 7・キャンセルのBUTTONで設定された内容が無効になり以前に設定された内容になります。
- 8・ヘルプのBUTTONを押すときっとこの画面が出ているはずですが。

CH2 Serial 設定

このダイアログは、CH2のボーレート・データビット・ストップビット・パリティ・制御方法を設定出来ます。

また、OPENしていないと開きません。



- 1・ボーレートはCOMBOBOXで設定します。
設定は75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400ボーレートです。
- 2・データビットもCOMBOBOXで設定します。
設定は8ビットと7ビットの2種類です。
6ビット・5ビットは対応していません。
- 3・ストップビットもCOMBOBOXで設定します。
設定は1ビットと1.5ビットと2ビットの3種類です。
0ビットは対応していません。
- 4・パリティもまたまたCOMBOBOXで設定します。
設定は「無し」と「奇数ビット」と「偶数ビット」の3種類です。
これ以外の設定は対応していません。
- 5・制御方法もCOMBOBOXで設定します。
設定はOFFとXONXOFF制御と制御線で制御するDTR制御の3種類です。
- 6・OKのBUTTONで設定された内容が反映します。
- 7・キャンセルのBUTTONで設定された内容が無効になり以前に設定された内容になります。
- 8・ヘルプのBUTTONを押すときっとこの画面が出ているはずです。

DCB設定（CH1）

Property	Value	Property	Value
<input checked="" type="checkbox"/> fBinary	1	fDtrControl	19200
<input type="checkbox"/> fOutxCtsFlow	0	fRtsControl	8
<input type="checkbox"/> fOutxDsrFlow	0	fDummy2	
<input type="checkbox"/> fDsrSensitivity	2048	XonLim	0
<input type="checkbox"/> fTXContinueOnXoff	512	XoffLim	0
<input type="checkbox"/> fOutX	11	XonChar	100
<input type="checkbox"/> fInX	13	XoffChar	2
<input type="checkbox"/> fErrorChar	0	ErrorChar	100
<input type="checkbox"/> fNull	0	EofChar	10
<input type="checkbox"/> fAbortOnError	d	EvtChar	1000

9600,8,n,1,off

初期状態 OK キャンセル

このダイアログでCH1のDCB設定を行います。
また、OPENしていないと開きません。

DCB設定とはWindows API関数のDCB構造体の事です。
通常はSerialの設定を行えば良いのですが、さらに詳細な設定を行うときに設定します。
たとえば、XONの文字を指定したいときやバッファの大きさを変えたい時に設定します。
設定内容についての説明は省きます。
詳細はWindows のAPI関数のDCB構造体について説明されている物を参照して下さい。
わからないときは、むやみに設定しないで下さい。

DCB設定 (CH2)

Property	Value	Property	Value
<input checked="" type="checkbox"/> fBinary	1	fDtrControl	19200
<input type="checkbox"/> fOutxCtsFlow	0	fRtsControl	8
<input type="checkbox"/> fOutxDsrFlow	0	fDummy2	
<input type="checkbox"/> fDsrSensitivity	2048	XonLim	0
<input type="checkbox"/> fTXContinueOnXoff	512	XoffLim	0
<input type="checkbox"/> fOutX	11	XonChar	100
<input type="checkbox"/> fInX	13	XoffChar	2
<input type="checkbox"/> fErrorChar	0	ErrorChar	100
<input type="checkbox"/> fNull	0	EofChar	10
<input type="checkbox"/> fAbortOnError	d	EvtChar	1000

9600,8,n,1,off

初期状態 OK キャンセル

このダイアログでCH2のDCB設定を行います。
また、OPENしていないと開きません。

DCB設定とはWindows API関数のDCB構造体の事です。
通常はSerialの設定を行えば良いのですが、さらに詳細な設定を行うときに設定します。
たとえば、XONの文字を指定したいときやバッファの大きさを変えたい時に設定します。
設定内容についての説明は省きます。
詳細はWindows のAPI関数のDCB構造体を説明されている物を参照して下さい。
わからないときは、むやみに設定しないで下さい。

状態表示（CH1）

状態表示 (CH1)

<input type="radio"/> フレーウ状態	<input type="radio"/> 送信はGTS待ち	<input type="radio"/> GTS_ON
<input type="radio"/> フレーミングエラー	<input type="radio"/> 送信はDSR待ち	<input checked="" type="radio"/> DSR_ON
<input type="radio"/> オーバーランエラー	<input type="radio"/> 送信はRLS(CD)待ち	<input type="radio"/> RING_ON
<input type="radio"/> モードエラー	<input type="radio"/> 送信停止XOFFを受信	<input checked="" type="radio"/> RLSD_ON
<input type="radio"/> L/Oエラー	<input type="radio"/> 送信停止XOFFを送信	強制ON/OFF
<input type="radio"/> 受信BFオーバー	<input type="radio"/> EOF文字を送出	<input type="button" value="DTR ON"/>
<input type="radio"/> 受信パリティエラー	<input type="radio"/> 文字の送信待ち	<input type="button" value="DTR OFF"/>
<input type="radio"/> 送信BF満		<input type="button" value="RTS ON"/>
		<input type="button" value="RTS OFF"/>

このダイアログでCH1の状態表示の画面です。
この画面ではエラー表示と状態表示とハードウェア制御を行っています。
また、OPENしていないと開きません。

状態表示とはWindows API関数のClearCommErrorで取得した内容を表示します。
又、強制出力は、Windows API関数のEscapeCommFunctionで行っています。
たとえば、エラーの内容を表示したり、現在の状態表示したり、ハードウェアを制御します。
詳細はWindowsのAPI関数を説明されている物を参照して下さい。
見るのは問題ありませんが、わからないときは、むやみにクリックしないで下さい。

ハードウェア制御の出力の強制ON/OFF
出力の各ボタンを押せば、内容に応じてその処理を行います。

状態表示（CH2）

状態表示 (CH2)

<input type="radio"/> フレーウ状態	<input type="radio"/> 送信はGTS待ち	<input type="radio"/> GTS_ON
<input type="radio"/> フレーミングエラー	<input type="radio"/> 送信はDSR待ち	<input checked="" type="radio"/> DSR_ON
<input type="radio"/> オーバーランエラー	<input type="radio"/> 送信はRLS(CD)待ち	<input type="radio"/> RING_ON
<input type="radio"/> モードエラー	<input type="radio"/> 送信停止XOFFを受信	<input checked="" type="radio"/> RLSD_ON
<input type="radio"/> L/Oエラー	<input type="radio"/> 送信停止XOFFを送信	強制ON/OFF
<input type="radio"/> 受信BFオーバー	<input type="radio"/> EOF文字を送出	<input type="button" value="DTR ON"/>
<input type="radio"/> 受信パリティエラー	<input type="radio"/> 文字の送信待ち	<input type="button" value="DTR OFF"/>
<input type="radio"/> 送信BF満		<input type="button" value="RTS ON"/>
		<input type="button" value="RTS OFF"/>

このダイアログでCH2の状態表示の画面です。
この画面ではエラー表示と状態表示とハードウェア制御を行っています。
また、OPENしていないと開きません。

状態表示とはWindows API関数のClearCommErrorで取得した内容を表示します。
又、強制出力は、Windows API関数のEscapeCommFunctionで行っています。
たとえば、エラーの内容を表示したり、現在の状態表示したり、ハードウェアを制御します。
詳細はWindowsのAPI関数を説明されている物を参照して下さい。
見るのは問題ありませんが、わからないときは、むやみにクリックしないで下さい。

ハードウェア制御の出力の強制ON/OFF
出力の各ボタンを押せば、内容に応じてその処理を行います。

設定文字の編集

ここは、設定文字送信用の文字を設定するところで、mdata, txtをメモ帳で編集します。

操作方法はメモ帳のヘルプを参照してください。

又、設定文字として認識できるのは、256文字列×200件の文字のみで、それ以上は無効です。

メモ帳でなくても他のE D I Tで編集してもかまいません。

新規マクロ作成

ここは、マクロファイルを新規に作成します。

と言っても、ただ、メモ帳が無題で立ち上がるだけです。

マクロ分の作成が終われば、Macroのフォルダーに収納して下さい。

Macro命令についてはMacroを参照してください。

Macroファイルはテキストファイルなので他のエディターでも作る事が出来ます。

Macroとして認識できるのは、256×100文字列の文字のみで、それ以上は無効です。

マクロ文の編集

ここは、マクロファイルを編集します。

Macroのフォルダー内のTxtファイルを選べば、メモ帳で編集します。

マクロ分の編集が終われば、Macroのフォルダーに収納して下さい。

Macro命令についてはMacroを参照してください。

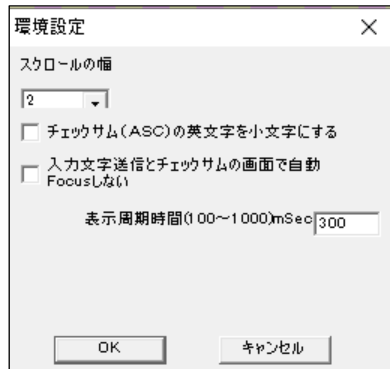
Macroファイルはテキストファイルなので他のエディターでも作る事が出来ます。

Macroとして認識できるのは、256×100文字列の文字のみで、それ以上は無効です。

環境設定

メニューの環境設定の中にあります。

このダイアログは、画面の表示の設定を行います。



スクロールの幅

1～21行の設定が可能です。

私は2がちょうどいいと思います。

「チェックサムの英文字を小文字にする」にチェックを入れると
チェックサムの結果の英文字が小文字に対応します。

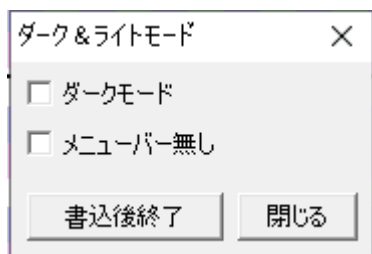
「入力文字送信とチェックサムの画面で自動Focusしない」にチェックを入れると
入力文字送信とチェックサムの画面でSetFocusが自動でしない。

表示周期時間でバッファに書き込まれた内容を画面に反映する周期時間の設定です。
通常300mSECに設定するとよいでしょう

「閉じる」のBUTTONでこのダイアログが閉じます。

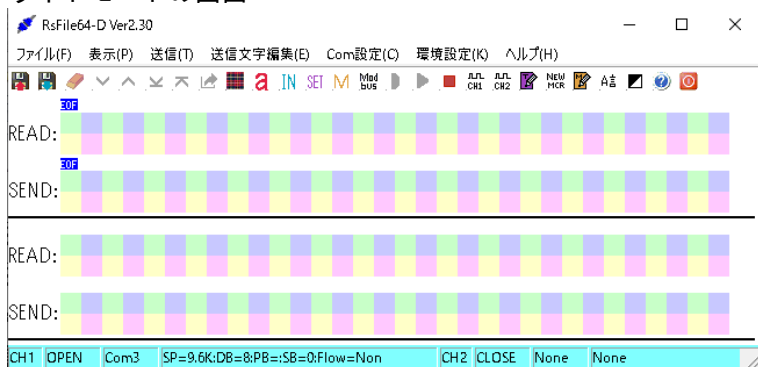
ダーク & ライトモード

メニューの環境設定の中にあります。

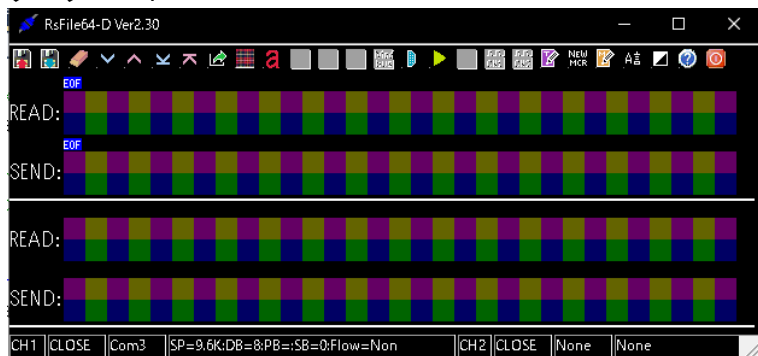


ダークモード時はチェックを入れる。ライトモード時はチェックを外す。
ダークモードにしてもメニューバーはライトのままです。
メニューバーを消す時は、メニューバー無しにチェックを入れてください。
書込後終了を押すとアプリは自動で終了します。
閉じるのボタンはそのまま画面を閉じる
モードを切り替えるにはアプリの再起動をお願いします。

ライトモードの画面



ダークモード



下記の物はダークモードになりません。（簡単に出来なかった）
メニューバーはライトモードです。

よって、メニューバー有・無の機能を追加しました。
Dialogの押しボタンもライトモードです。

Set of language

メニューの環境設定の中にあります。

このダイアログは、画面の表示の設定を行います。



このアプリの言語を変更が出来ます。

日本語・英語・韓国語・中国語の4種類選択出来ます。

日本語について

日本人でありながら国語が苦手です。誤字脱字や意味違っていたらごめんなさい。

英語について

日本人で英語はまったく出来ません。よって翻訳サイトで変換しました。
意味が違っていたらごめんなさい。

韓国語について

日本人でハングルはまったく出来ません。よって翻訳サイトで変換しました。
意味が違っていたらごめんなさい。

中国語について

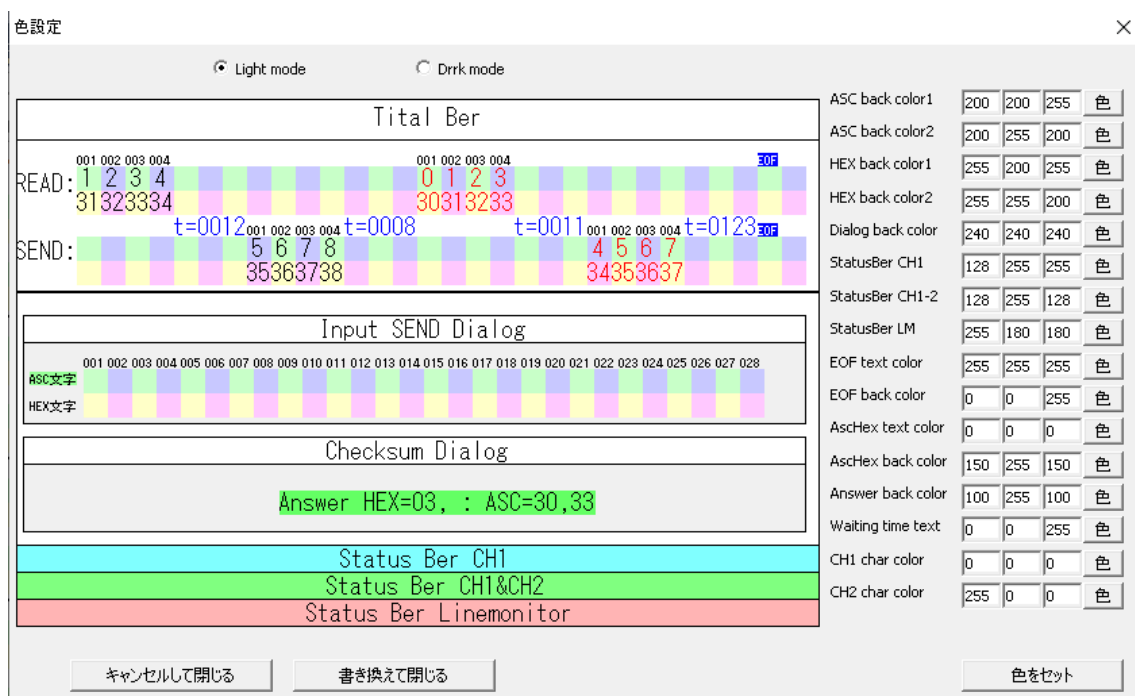
日本人で中国語はまったく出来ません。よって翻訳サイトで変換しました。
意味が違っていたらごめんなさい。

この言語は「rsfile64d_char2.ini」のファイルで設定しています。

意味が違っていてもどうしても修正するなら、「rsfile64d_char2.ini」の内容を変更すれば可能です。

色設定

メニューの環境設定の中にあります。



各色の設定が出来る機能です

色はRGB単位で設定します。

「色」のボタンを押すと下記の画面が表示します。



RGBをEditboxで数値にて設定した場合は「色をセット」のボタンを押せば左側の図に反映します。

「キャンセルして閉じる」のボタンを押したら、設定した内容を反映せず画面を閉じます
「書き換えて閉じる」のボタンを押したら、設定した内容を色テーブルに保存して画面を閉じます

スタート・ストップ・E N D文字とチェックサムについて

A S C文字送信・H E X文字送信時に設定された文字の前後に決められた文字をあらかじめ設定知ることが出来ます。

<スタート文字>

スタート文字とは送信入力文字の先頭に1 B Y T E付けて送信します。使用時はチェックボックスをO Nして文字をE D I Tで設定してください。文字はH E X文字で設定して下さい。

<ストップ文字>

ストップ文字とは送信入力文字の最後に1 B Y T E付けて送信します。使用時はチェックボックスをO Nして文字をE D I Tで設定してください。文字はH E X文字で設定して下さい。

<チェックサム>

チェックサムとは送信入力文字の全文字をチェックサム計算した結果の文字を付けて送信します。

チェックサムはストップ文字の次に付いて送信します。

<チェックサムの詳細設定>

「詳細設定」のボタンは、チェックサムの詳細設定のDialogを表示する

1・チェックサムの結果をH E X文字で1文字するか。又はA S C文字で2文字にするか設定します。各ラジオボタンを押して設定して下さい。

2・チェックサムの計算方法の設定

「Byte単位でXOR」とはByteづつをXORする方法です。

例) 30Hと14Hと31HをXORすると15Hとなります。

「Byte単位で加算」とはByteづつを加算した結果の下1Byteを使用する方法です。

例) 30Hと14Hと31Hを加算すると75Hとなります。

<E N D文字>

E N D文字とはこの送信文字の一番最後に1 B Y T E付けて送信します。(通常終了コードといいます)

使用時はチェックボックスをO Nして文字をE D I Tで設定してください。文字はH E X文字で設定して下さい。

<F O R M A T>

以上を送信全文字で表現すると以下のようになります。

[スタート文字] + [入力文字] + [ストップ文字] + [チェックサム] + [E N D文字]

MACRO文について

MACROは4文字のコマンド文字とパラメーター文字により送信・受信・入出力などのシリアル・I/O操作を自動でできるものです。

又、記憶できる文字列メモリとBITメモリがあります。

文字列メモリはMDATAと呼び、0～9番地まであり、一つの番地で256文字記憶できます。

BITメモリはMBITと呼び、00～99ポートで設定でき、1ポートで7ビット記憶できます。

MACRO文はCR（改行）でひとつのコマンドになっています。

コメント一覧表

送信コマンド

SEND 文字を指定してCH1で送信する

MSED M_DATAの内容をCH1で送信する

SED2 文字を指定してCH2で送信する

MSD2 M_DATA内容をCH2で送信する

受信コマンド

READ CH1で文字を受信してM_DATAへ格納する

ZRED CH1で文字を受信してM_DATAへ格納する（以前も取得）

RED2 CH2で文字を受信してM_DATAへ格納する

ZRD2 CH2で文字を受信してM_DATAへ格納する（以前も取得）

タイマーコマンド

TIME 設定した時間待つ

比較コマンド

CMPM M_DATAとM_DATAを比較して条件ジャンプ

CMP L M_DATAと指定文字を比較して条件ジャンプ

チェックサム コマンド

CHKA M_DATAの内容をXORのチェックサムしてASCでM_DATAへ格納する

CHKH M_DATAの内容をXORのチェックサムしてHEXでM_DATAへ格納する

CK1A M_DATAの内容を加算のチェックサムしてASCでM_DATAへ格納する

CK1H M_DATAの内容を加算のチェックサムしてHEXでM_DATAへ格納する

制御コマンド

GOTO ジャンプ命令

PEND マクロ終了命令

文字列制御コマンド

MOVE 指定した文字列をM_DATAへ格納する

MADD M_DATAの文字とM_DATA文字を加算する

MMID M_DATAの文字の中から指定した文字を取りだしてM_DATAへ格納

MINC M_DATAの文字を10進法でインクリメントする

HINC M_DATAの文字を16進法でインクリメントする

MCLR M_DATAをクリアする

ラベル

ラベル ジャンプの飛び先

記憶メモリ

MDATA 文字列を格納するエリア

SEND

CH 1 の送信コマンド

概要：パラメーターに設定された文字をCH 1 で送信します。

コマンド：SEND

パラメーター：送信文字（先頭から 6 番めに記入）

送信文字は 0 ～ F の HEX 数値 2 文字で一組の ASC に変換して送信します。

文字をそのまま送るときは “ ” 又は ‘ ’ を前後に付ければ HEX でなく

ASC そのままで送信します。

例 SEND 02"123455667"030D

パラメーター 2 に設定された文字「SX “1 2 3 4 5 5 6 6 7” EX CR」をCH 1 から送信します。

MSED

CH 1 の送信コマンド

概要：パラメーターに設定されたMDATAの番号の文字をCH 1 で送信します。

コマンド：MSED

パラメーター 1：MDATAの番号（先頭から 6 番めに 2 文字記入）

例 MOVE 00 02"ABC"030D

MSED 00

MDATAの番地 00 にパラメーター文字「SX"ABC" EX CR」をセットし、

MDATAの 00 番地の文字列をCH 1 から送信します。

SED 2

CH 2 の送信コマンド

概要：パラメーターに設定された文字をCH 2 で送信します。

コマンド：SEND

パラメーター：送信文字（先頭から 6 番めに記入）

送信文字は 0 ～ F の HEX 数値 2 文字で一組の ASC に変換して送信します。

文字をそのまま送るときは “ ” 又は ‘ ’ を前後に付ければ HEX でなく

ASC そのままで送信します。

例 SEND 02"123455667"030D

パラメーター 2 に設定された文字「SX “1 2 3 4 5 5 6 6 7” EX CR」をCH 2 から送信します。

MSD 2

CH 2の送信コマンド

概要：パラメーターに設定されたMDATAの番号の文字をCH 2で送信します。

コマンド：MSED

パラメーター 1： MDATAの番号（先頭から 6 番めに 2 文字記入）

例 MOVE 00 02"ABC"030D
MSED 00

MDATAの番地 00 にパラメーター文字「SX"ABC"EX CR」をセットし、
MDATAの 00 番地の文字列をCH 2から送信します。

READ

CH 1の受信コマンド

概要：このコマンドが実行されたから、CH 1で受信するまで待ちつづけ、受信された文字は、
パラメーターに設定されたMDATAの番号に格納されます。（このコマンドを実行された後の
文字を受信します）

コマンド：READ

パラメーター 1： MDATAの番号（先頭から 6 番めに 2 文字記入）

パラメーター 2： 受信数（先頭から 9 番めに 3 文字記入）

受信数が設定されている数までこのコマンドは実行します。
受信文字を 000 にすると受信時間が優先になり受信時間まで
の文字を受信します。

パラメーター 3： 受信時間（先頭から 1 3 番めに 3 文字記入）

受信時間が設定されている時間までこのコマンドは実行します。
受信文字を 000 にすると受信時間が無効になり受信時間まで
の文字を受信します。
単位は 0. 1 S です。

*パラメーター 2 とパラメーター 3 両方 0 のときは一回受信するまで実行します。

例 READ 00 010 100
MSED 00

CH 1 で受信数 10 文字又は、10. 0 秒まで受信待ちし、その文字列は、MDATA
の 00 番地に記憶され、その文字をそのまま送信します。

ZRED

CH1の受信コマンド

概要：CH1で受信するまで待ちつづけ、受信された文字は、パラメーターに設定されたMDATAの番号に格納されます。（但し受信文字は送信後から記憶された文字です。タイミングによってはREADコマンドが読めない時使います。）

コマンド：READ

パラメーター1：MDATAの番号（先頭から6番めに2文字記入）

パラメーター2：受信数（先頭から9番めに3文字記入）

受信数が設定されている数までこのコマンドは実行します。
受信文字を000にすると受信時間が優先になり受信時間までの文字を受信します。

パラメーター3：受信時間（先頭から13番めに3文字記入）

受信時間が設定されている時間までこのコマンドは実行します。
受信文字を000にすると受信時間が無効になり受信時間までの文字を受信します。
単位は0.1Sです。

*パラメーター2とパラメーター3両方0のときは一回受信するまで実行します。

例 ZED 00 010 100
MSD 00

CH1で受信数10文字又は、10.0秒まで受信待ちし、その文字列は、MADATAの00番地に記憶され、その文字をそのまま送信します。

RED2

CH2の受信コマンド

概要：このコマンドが実行されたから、CH2で受信するまで待ちつづけ、受信された文字は、パラメーターに設定されたMDATAの番号に格納されます。（このコマンドを実行された後の文字を受信します）

コマンド：READ

パラメーター1：MDATAの番号（先頭から6番めに2文字記入）

パラメーター2：受信数（先頭から9番めに3文字記入）

受信数が設定されている数までこのコマンドは実行します。
受信文字を000にすると受信時間が優先になり受信時間までの文字を受信します。

パラメーター3：受信時間（先頭から13番めに3文字記入）

受信時間が設定されている時間までこのコマンドは実行します。
受信文字を000にすると受信時間が無効になり受信時間までの文字を受信します。
単位は0.1Sです。

*パラメーター2とパラメーター3両方0のときは一回受信するまで実行します。

例 READ 00 010 100
MSD 00

CH2で受信数10文字又は、10.0秒まで受信待ちし、その文字列は、MADATAの00番地に記憶され、その文字をそのまま送信します。

Z R D 2

CH 2の受信コマンド

概要：CH 2で受信するまで待ちつづけ、受信された文字は、パラメーターに設定されたMDATAの番号に格納されます。（但し受信文字は送信後から記憶された文字です。タイミングによってはREADコマンドが読めない時使います。）

コマンド：READ

パラメーター1：MDATAの番号（先頭から6番めに2文字記入）

パラメーター2：受信数（先頭から9番めに3文字記入）

受信数が設定されている数までこのコマンドは実行します。
受信文字を000にすると受信時間が優先になり受信時間までの文字を受信します。

パラメーター3：受信時間（先頭から13番めに3文字記入）

受信時間が設定されている時間までこのコマンドは実行します。
受信文字を000にすると受信時間が無効になり受信時間までの文字を受信します。

単位は0.1Sです。

*パラメーター2とパラメーター3両方0のときは一回受信するまで実行します。

例 ZRAD 00 010 100
MSRD 00

CH 2で受信数10文字又は、10.0秒まで受信待ちし、その文字列は、MDATAの00番地に記憶され、その文字をそのまま送信します。

T I M E

時間待ちコマンド

概要：設定された時間まで待ちつづけます。

コマンド：TIME

パラメーター：待ち時間（先頭から6番めに3文字記入）

単位は0.1Sです。

例 SEND "123"0D
TIME 100
SEND "ABC"0D

“123”を送信して10.0秒後にもう一度“ABC”を送信します。

GOTO

ジャンプコマンド

概要：パラメーターにかかれたラベルまで飛びます。

コマンド：GOTO

パラメーター： ジャンプ先ラベル（先頭から6番めに3文字記入）
ラベルはASC文字です。

例 GOTO ABC
 TIME 100
 :ABC
 PEND

TIMEコマンドを実行せず、ラベルABC（:ABC）まで飛んで終了します。

CMPM

メモリ同志の比較コマンド

概要：MDATAとMDATAの内容を比較し一致すれば指定したラベルまで飛びます。

パラメーター1： MDATAの番号（先頭から6番めに2文字記入）

パラメーター2： MDATAの番号（先頭から9番めに2文字記入）

パラメーター3： ジャンプ先ラベル（先頭から12番めに3文字記入）

例 READ 00 010 100
 MOVE 01 "123"OD
 CMPM 00 01 ABC
 SEND "NG"
 GOTO EEE
 :ABC
 SEND "OK"
 :EEE
 PEND

受信した文字をMDATAの番地00に入れ、文字“123”+CRをMDATAの番地01にいれ、MDATA00とMDATA01を比較し一致すればラベルABCへ飛びん“OK”と送信し。一致しないときは次行へ行き“NG”と送信します。

C M P L

メモリとの比較コマンド

概要:MDATAとパラメーターの文字内容进行比较し一致すれば指定したラベルまで飛びます。

パラメーター1:MDATAの番号(先頭から6番めに2文字記入)

パラメーター2:ジャンプ先ラベル(先頭から9番めに3文字記入)

パラメーター3:比較文字(先頭から13番めに記入)

送信文字は0~FのHEX数値2文字で一組のASCに変換し、文字をその物は“ ”又は‘ ’を前後に付ければよいです。

```
例  READ 00 010 100
      CMPL 00 ABC "123"
      SEND "NG"
      GOTO EEE
      :ABC
      SEND "OK"
      :EEE
      PEND
```

受信した文字をMDATAの番地00に入れ、MDATA00と文字“123”+CRを比較し一致すればラベルABCへ飛びん“OK”と送信し。一致しないときは次行へ行き“NG”と送信します。

M O V E

メモリへ転送コマンド

概要:MDATAに文字をセットします。

パラメーター1:MDATAの番号(先頭から6番めに2文字記入)

パラメーター2:文字(先頭から9番めに記入)

送信文字は0~FのHEX数値2文字で一組のASCに変換し、文字をその物は“ ”又は‘ ’を前後に付ければよいです。

```
例  MOVE 00 02"ABC"030D
      MSRD 00
```

MDATAの番地00にパラメーター文字「SX"ABC"EX CR」をセットし、MDATAの00番地の文字列を送信します。

MADD

メモリ同志の加算コマンド

概要：MDATAとMDATAの文字を加算します。

パラメーター１：MDATAの番号（先頭から６番めに２文字記入）加算先

パラメーター２：MDATAの番号（先頭から９番めに２文字記入）加算元

例 MOVE 00 "ABC"
 MOVE 01 "123"
 MADD 00 01

MDATAの番地００にパラメーター文字「"ABC"」をセットし、
MDATAの番地０１にパラメーター文字「"123"」をセットし、
MDATAの番地００の最終文字からMDATAの番地０１の文字を加算します。
結果としてMDATAの００番地は「"ABC123"」になっています。

MMID

メモリ内の取り出しコマンド

概要：MDATAの中から先頭から何文字目から何文字分をMDATAへ格納します。

パラメーター１：MDATAの番号（先頭から６番めに２文字記入）元のDATA

パラメーター２：先頭文字を指定する２桁の数値（先頭から９番めに２文字記入）

パラメーター３：取得する文字数を指定する２桁の数値（先頭から１２番めに２文字記入）

パラメーター４：MDATAの番号（先頭から９番めに２文字記入）格納先

例 MOVE 00 "1234567890"
 MMID 00 03 04 01

MDATAの番地００にパラメーター文字「"1234567890"」をセットし、
MDATAの番地００の文字の先頭から３番目から４文字分をMDATAの番地０１へ
コピーします。
結果としてMDATAの０１番地は「"4567"」になっています。

MINC

メモリでのインクリメントしコマンド

概要：MDATAの数文字を１０進法でインクリメントします。

パラメーター１：MDATAの番号（先頭から６番めに２文字記入）

例 MOVE 00 "000"
 MINC 00

MDATAの番地００にパラメーター文字「"000"」をセットし、
MDATAの番地００をインクリメントします。
結果としてMDATAの００番地は「"001"」になっています。

HINC

メモリでのHEXでインクリメントしコマンド

概要：MDATAの数文字を16進法でインクリメントします。

パラメーター1：MDATAの番号（先頭から6番めに2文字記入）

例 MOVE 00 "000"
 MINC 00

MDATAの番地00にパラメーター文字「"000"」をセットし、
MDATAの番地00をインクリメントします。
結果としてMDATAの00番地は「"001"」になっています。

CHKA

チェックサム結果をASCに出力するコマンド

概要：MDATAの文字をXORでチェックサムし、その結果MDATAにASC文字として格納します。

パラメーター1：MDATAの番号（先頭から6番めに2文字記入）

パラメーター2：MDATAの番号（先頭から9番めに2文字記入）

例 MOVE 00 "123"
 CHKA 00 01

MDATAの番地00にパラメーター文字「"123"」をセットし、
MDATAの番地00をチェックサムします。
その結果をASC文字（2文字）をMDATAの番地01に格納します。
結果としてMDATAの01番地は「"30"」になっています。

XORとは

スタート文字・ストップ文字を含まない送信文字をXORします。

例) "123456"の場合

31 XOR 32 XOR 33 XOR 34 XOR 35 XOR 36 = 07(HEX)

CHKH

チェックサム結果をHEXに出力するコマンド

概要: MDATAの文字をXORでチェックサムし、その結果MDATAにHEX文字として格納します。

パラメーター1: MDATAの番号 (先頭から6番めに2文字記入)

パラメーター2: MDATAの番号 (先頭から9番めに2文字記入)

例 MOVE 00 "123"
 CHKA 00 01

MDATAの番地00にパラメーター文字「" 1 2 3"」をセットし、
MDATAの番地00をチェックサムします。
その結果をHEXC文字 (1文字) をMDATAの番地01に格納します。
結果としてMDATAの01番地は「"0"」になっています。

XORとは

スタート文字・ストップ文字を含まない送信文字をXORします。

例) "1 2 3 4 5 6" の場合

31 XOR 32 XOR 33 XOR 34 XOR 35 XOR 36 = 07 (HEX)

CK1A

加算タイプのチェックサム結果をASCに出力するコマンド

概要: MDATAの文字を加算タイプでチェックサムし、その結果MDATAにASC文字として格納します。

パラメーター1: MDATAの番号 (先頭から6番めに2文字記入)

パラメーター2: MDATAの番号 (先頭から9番めに2文字記入)

例 MOVE 00 "123"
 CHKA 00 01

MDATAの番地00にパラメーター文字「" 1 2 3"」をセットし、
MDATAの番地00をチェックサムします。
その結果をASC文字 (2文字) をMDATAの番地01に格納します。
結果としてMDATAの01番地は「"30"」になっています。

加算タイプとは

スタート文字・ストップ文字を含まない送信文字をBYTE単位で加算して下位8bitを使う。

例) "1 2 3 4 5 6" の場合

31 + 32 + 33 + 34 + 35 + 36 = 135 → 35 (HEX)

CK1H

加算タイプのチェックサム結果をHEXに出力するコマンド

概要：MDATAの文字を加算タイプでチェックサムし、その結果MDATAにHEX文字として格納します。

パラメーター1：MDATAの番号（先頭から6番めに2文字記入）

パラメーター2：MDATAの番号（先頭から9番めに2文字記入）

例 MOVE 00 "123"
CHKA 00 01

MDATAの番地00にパラメーター文字「" 1 2 3"」をセットし、MDATAの番地00をチェックサムします。
その結果をHEXC文字（1文字）をMDATAの番地01に格納します。
結果としてMDATAの01番地は「"0"」になっています。

加算タイプとは

スタート文字・ストップ文字を含まない送信文字をBYTE単位で加算して下位8bitを使う。

例) "1 2 3 4 5 6" の場合

$31 + 32 + 33 + 34 + 35 + 36 = 135 \rightarrow 35(\text{HEX})$

MCLR

メモリ内をクリアするコマンド

概要：MDATAの内容をクリアします。

（MDATAとは文字列を記憶するメモリの事）

パラメーター1：メモリ番号00～99（先頭から6番め2文字記入）

例 MCLR 05

この行を実行されたときMDATAの05の内容をクリアします。

備考 MDATAの詳細について

MDATAは以下の内容のエリアがあります。

1・256文字入るエリア

2・何文字有効であるかという文字数が入るエリア

* 文字の中でNULLを有効にするため、有効文字数の設定があります。

PEND

マクロ命令の終了コマンド

概要：この自動送受信マクロを終了させます。

コマンド：PEND

例 PEND

マクロを終了させ「MACROが終了しました。」とメッセージBOXが出ます。

ラベル

概要：GOTOやCMPM・CMPLのジャンプ先に使うラベルに使用します。

書式：「：」を先頭に書きその後ろ3文字がラベルになります。

例 :ABC

M DATA

文字列を記憶するメモリでM DATAと呼びます。

概要：

文字を記憶するためのメモリで、00～99の100個あります。

ひとつのメモリで256文字分の文字が記憶できます。

使用するコマンド

MSD・READ・ZRED・CMPM・CMPL・MOVE・MINC・HINC・CHKA・CHKH・PUEM

このソフトについて

以前に作成したソフトを64Bit化にして表現も変更しました。
このソフトでRS232C通信のデバックに役に立つと思います。
産業用機器ではいまでもRS232C通信を使用しています。
まだまだ、このソフトの使い道はあると思います。
是非、ご利用下さい。

バージョンについて

- Ver1.91 RsFile-D.exeはWin32で構築したもの
- Ver2.10 RsFile-D.exeから×64で構築して名前はRsFile64-D.exeに変更。
- Ver2.20 2024-4-6
大幅に改善しリニューアルしました。（わかりやすくした）
画面表示はASC文字HEX文字を同時に表示し切換しなくてよい
待機時間の表示の追加
送受信文字数の表示の追加
チェックサムにCRC16を追加
テキスト表示は撤去した
数値変換機能も撤去した
- Ver2.22 2024-5-4
日本語以外でのメニューの項目が一部なかったので修正した
表示方法は送受信毎に行うのではなく時間周期で表示するように修正
環境設定にて表示周期時間の変更出来るようにした。
時計表示でバッファ方式から記憶方式に変更
チェックサムにLRCを追加
ModBus送信Dialog画面を追加
ステータルバーの色を変更
入力文字の終了コードで「OD0A」CRLFの設定可能した
- Ver2.30 2025-5-4
画面でダークモードとライトモードを追加した。
但しメニューはダークモードに出来ないため
メニューバー表示・非表示の設定を追加。
画面の色テーブルをダークモードとライトモードに用意し
文字やバックの色を自由に設定可能にした。
メニューバーにアイコンを表示した。
ツールバーのアイテムを増やしてコメント表示を付けた。

ご意見など連絡先

そのソフトに関するご意見、ご感想、ご要望は下記のメールアドレスでご連絡するようにお願いします。

bit deviser @ yah oo .co. jp