

# Windows 用ターミナル ソフト

## あじゃた〜む (ユニコード版)

### 操作説明書

( 1.6.2.2 版 )

## - 目次 -

1.	概要	1
2.	メインウインド	5
2.1.	ファイル出力	7
2.2.	文字コード	7
2.3.	キー入力内容の送信とローカルエコー	7
2.4.	テキストの送信	8
2.5.	バイナリデータの送信	8
2.6.	受信エラーカウンタ表示	8
2.7.	ストロボ機能	9
2.8.	数値の認識と受信テキストの文字変換	10
2.9.	自動スクロール	11
2.10.	ANSIエスケープコード	12
2.11.	言語設定	12
2.12.	ポップアップメニュー	13
3.	通信回線の設定	14
4.	タイムチャート(波形グラフ)の表示	16
4.1.	受信データの選択と数値認識方法の設定	17
4.1.1.	受信テキストデータから数値を抽出する場合	17
4.1.2.	パケットデータから数値を抽出する場合	20
4.2.	ポップアップメニュー	22
4.3.	レンジ設定	22
4.4.	ワンタッチでレンジ設定	22
4.5.	レンジ自動調整	23
4.6.	ドラッグ操作によるレンジ設定	23
4.7.	オフセット設定	24
4.8.	横線の描画	25
4.9.	テキストファイルを入力し波形グラフ表示	26
4.10.	描画時間情報表示	26
4.11.	波形の補間表示	27
4.12.	波形補間表示の設定	28
4.13.	プロット周期表示	29
4.14.	時間計測	29
5.	2Dプロットグラフの表示	30
5.1.	受信データの選択と数値認識方法の設定	31
5.1.1.	受信テキストデータから数値を抽出する場合	31
5.1.1.	受信パケットデータから数値を抽出する場合	34
5.2.	ポップアップメニュー	36
5.3.	レンジ設定	36

5.4.	ワンタッチでレンジ設定 .....	36
5.5.	レンジ自動調整 .....	37
5.6.	アスペクト比の設定 .....	37
5.7.	フィルタ機能 .....	37
5.8.	スケールの表示 .....	37
5.9.	グラフィック上のプロット点の座標値表示 .....	38
5.10.	テキストファイルを入力し2D プロット・グラフ表示 .....	39
1.1.1.	描画時間情報表示 .....	39
6.	<b>3Dプロットグラフの表示</b> .....	40
6.1.	受信データの選択と数値認識方法の設定 .....	41
6.1.1.	受信テキストデータから数値を抽出する場合 .....	41
6.1.2.	受信パケットデータから数値を抽出する場合 .....	44
6.2.	ポップアップメニュー .....	46
6.3.	レンジ設定 .....	47
6.4.	ワンタッチでレンジ設定 .....	47
6.5.	レンジ自動調整 .....	47
6.6.	アスペクト比の設定 .....	47
6.7.	フィルタ機能 .....	48
6.8.	スケールの表示 .....	48
1.1.2.	描画時間情報表示 .....	49
1.1.3.	奥行き表現 .....	49
6.9.	グラフィック上のプロット点の座標値表示 .....	50
6.10.	テキストファイルを入力し3D プロット・グラフ表示 .....	51
7.	<b>マクロ機能</b> .....	52
8.	<b>ファイル送受信</b> .....	53
8.1.	テキストファイル送信 .....	53
8.2.	バイナリファイル送信 .....	54
8.3.	ファイル転送(XMODEM) .....	55
8.3.1.	XMODEMによるファイル送信 .....	55
8.3.2.	XMODEMによるファイル受信 .....	55
8.4.	ファイル転送(YMODEM) .....	56
8.4.1.	YMODEMによるファイル送信 .....	56
8.4.2.	YMODEMによるファイル受信 .....	57
9.	<b>中継機能</b> .....	58
10.	<b>初期化ファイル</b> .....	60
11.	<b>テストデータ用サンプルプログラム (Test32.exe / Test64.exe)</b> .....	61
12.	<b>アドイン</b> .....	62
12.1.	ベクトルデータの構造体 .....	63
12.2.	アドイン関数の仕様 .....	63
12.2.1.	アドインDLL有効化の処理 .....	63

12.2.2.	アドインDLL無効化の処理	63
12.2.3.	「設定」→「アドインの設定」→「アドインのプロパティ」メニュー選択時の処理	63
12.2.4.	通信ポートのオープン時の処理	64
12.2.5.	通信ポートのクローズ時の処理	64
12.2.6.	タイムチャートデータの値変更／表示中止	64
12.2.7.	2Dプロットデータの値変更／表示中止	65
12.2.8.	3Dプロットデータの値変更／表示中止	65
12.2.9.	タイムチャート・ウインドのオープン／クローズ通知	66
12.2.10.	タイムチャート・ウインドのデータクリアー通知	66
12.2.11.	2Dプロット・ウインドのオープン／クローズ通知	66
12.2.12.	2Dプロット・ウインドのデータクリアー通知	67
12.2.13.	3Dプロット・ウインドのオープン／クローズ通知	67
12.2.14.	3Dプロット・ウインドのデータクリアー通知	67
12.3.	アドインから あじやた〜むへのメッセージ	68
12.4.	アドイン関数の DLL エクスポート名称	68
12.5.	アドイン・サンプルプログラム	69
13.	問い合わせ先	78
14.	変更履歴	79

## 1. 概要

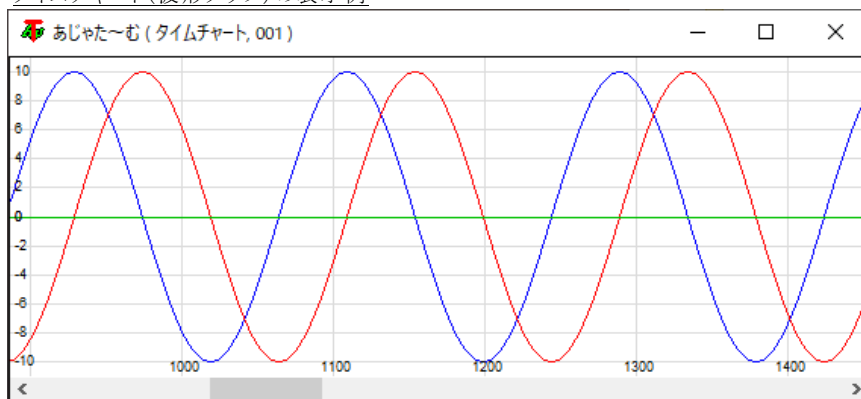
このソフトは、データ・ビューアをメインとした、ターミナルソフトです。(以降「あじやた〜む」と言う)

主に、組み込み系ソフトウェア(ファームウェア)の評価/デバッグを目的としています。

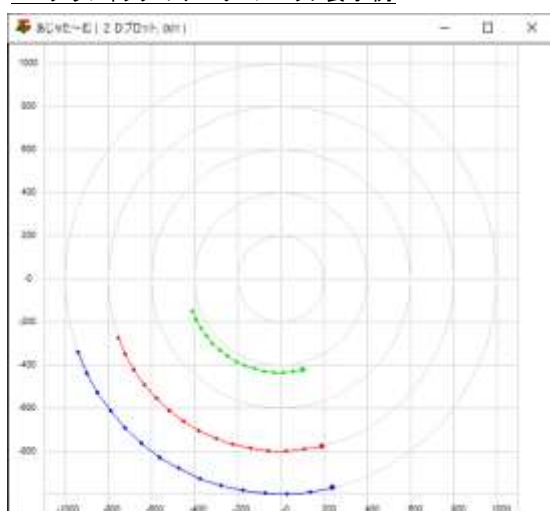
シリアル回線(COMポート)と、メールスロット、ソケット(TCPIPクライアント)による通信が可能です。

受信したデータの中から数値を取り出して、タイムチャート(波形グラフ)を表示したり、2D/3Dグラフィック上にデータをプロット表示することができます。

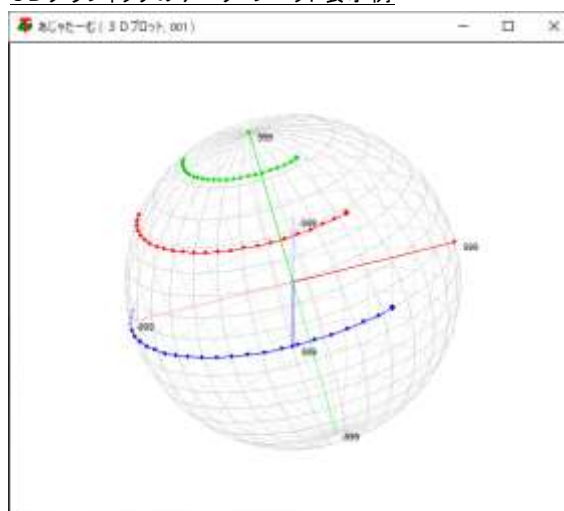
タイムチャート(波形グラフ)の表示例



2Dグラフィックのデータ・プロット表示例



3Dグラフィックのデータ・プロット表示例



受信データは、テキストデータの他に、透過型パケットフレーム (DLE・STX~DLE・ETX で区切られたバイナリ・データ) を扱うことができます。テキストデータは行単位で認識します。テキストデータ行の末尾は CR(0x0D), LF(0x0A) あるいは CR・LF の2バイトです。透過型パケットフレーム(以降パケットと言う)は、DLE・STX で始まり、DLE・ETX で終了するバイトストリームです。

パケット・フレーム 

DLE	STX	パケットデータ	DLE	ETX
-----	-----	---------	-----	-----

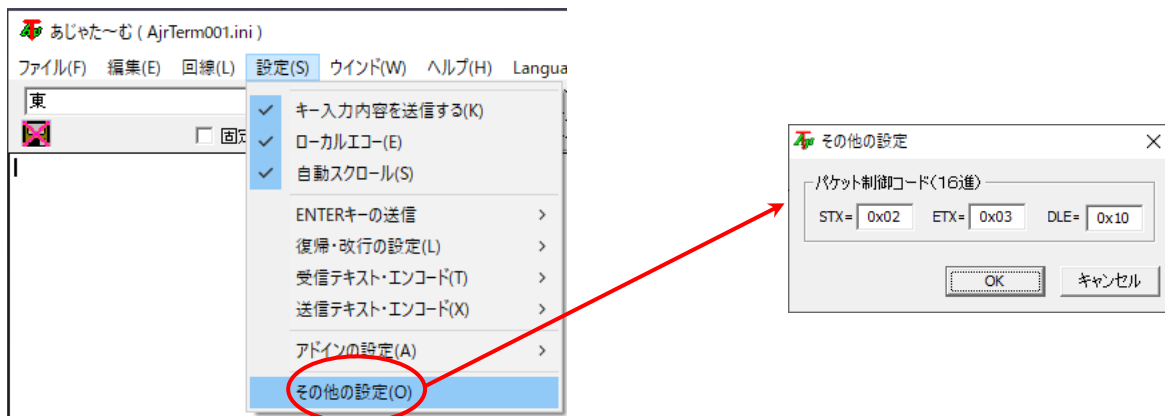
 デフォルトでは、STX=0x02, ETX=0x03, DLE=0x10

STX,ETX や DLE の実際のコード値は、自由に設定可能です。(デフォルトでは、STX=0x02, ETX=0x03, DLE=0x10)

データ受信時、パケットデータ中の、2つの連続する DLE は、1つの DLE に変換されます。

(つまり、DLEと同一値のバイトをパケットデータに含める場合は、DLE を1つ挿入して、DLE・DLE の2バイトに変換して送信します)

STX.ETX や DLE の実際のコード値を変更するには、「設定」メニューから「その他の設定」を選択します。



「あじやた〜む」では、データ・ビューア以外にも、以下のような機能があります。

#	機能	内容
1	VT100エミュレーション	テキスト受信において、一部のエスケープシーケンスをサポートしています。
2	受信データのログ表示機能	受信したテキストの表示の他に、受信データを16進ダンプ表示したり、パケットデータの単位で16進ダンプ表示することができます。
3	受信データの保存機能	表示した内容をファイルにも記録する機能です。
4	テキスト送信機能	キー入力した内容や、テキストボックスに入力した内容を送信します。
5	ファイル送信機能	テキストファイルや、バイナリファイルを送信します。
6	XMODEMファイル転送機能	XMODEMプロトコルによる、単一ファイルの送受信を行います。
7	YMODEMファイル転送機能	YMODEMプロトコルによる、単一ファイルの送受信や、フォルダ下のファイル群を一括して送受信することができます。

### UNICODE

バージョン 1.6.0.0 以降の「あじやた〜む」は、UNICODE対応のアプリケーションです。さまざまな多国間の文字（韓国語、中国語、日本語、ロシア語・・・）を混在して扱うことができます。

バージョン 1.6.0.0 より前の「あじやた〜む」は、バイト文字アプリケーションです。扱える文字は、規定のバイトコードページ（日本の場合はコードページ932、日本語シフトJIS）に限定されます。

UNICODEバージョンとバイト文字バージョンでは、メインウインドに表示する際の1行当たりの全角文字数に違いがあります。バイト文字バージョンでは、半角文字は1桁、全角文字は2桁として計算します。UNICODEバージョンでは、全角／半角の区別なく、全て1桁として計算します。例えば、描画メモリの文字数=8を設定した場合は、以下のようになります。

#### UNICODE バージョン

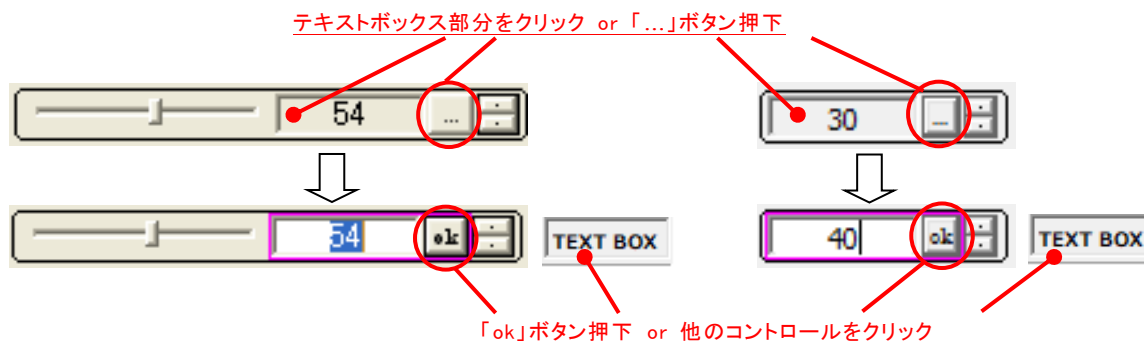


#### バイト文字バージョン



## データ入力に関して

以下のような外観のコントロールでデータを入力する場合、スライダやスピンのボタン以外に、テキスト部分をクリック(あるいは、「...」ボタン)で直接数値を入力することができます。



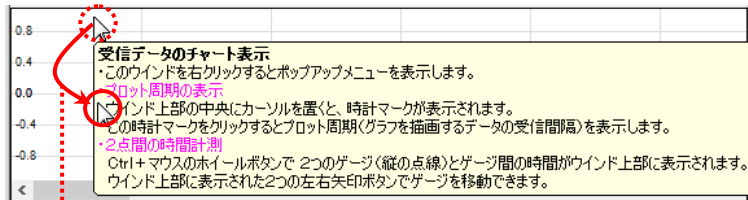
ここで数値を入力し、「ok」ボタンを押すか、他のコントロール(ボタンやテキストボックス等)をクリックすると、入力した数値が設定されます。16進数を入力する場合は、先頭に「0x」を付加してください。

## ツールチップについて

様々なボタンや、テキストボックス等にカーソルを置くとツールチップが表示され一定時間が経過するとツールチップが消えますが、このアプリケーションではツールチップを単に表示するだけでなく、以下の機能を持ちます。

### ・ツールチップの表示継続

ツールチップ上にマウスカーソルを移動すると、ツールチップの表示を継続します。



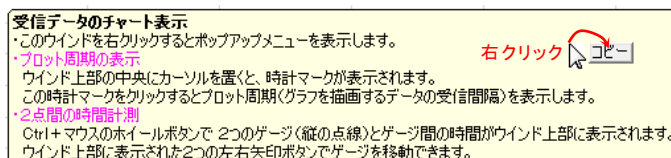
マウスカーソルを素早くツールチップ上に移動するとツールチップの表示を継続します。その後、マウスカーソルをツールチップ外に移動すると、ツールチップは消えます。

### ・ツールチップのコピー

ツールチップ上を右クリックすると「コピー」ボタンが表示されます。

「コピー」ボタンを押すと、ツールチップ・テキストがクリップボードにコピーされます。

SHIFT キーを押しながら「コピー」ボタンを押した場合は、ツールチップのイメージがコピーされます。



### ・ツールチップの非表示

ツールチップ上をクリックすると、ツールチップは消えます。

「コピー」ボタンが表示されている場合は、「コピー」ボタンが消えます。

### ・ツールチップの表示禁止

「設定」メニューで、「ツールチップ」→「表示禁止」を選択すると、全てのツールチップは表示されなくなります。

「設定」メニューで、「ツールチップ」→「表示許可」を選択すると、再度、全てのツールチップを表示するようになります。

## モーダルダイアログボックスとモードレスダイアログボックス

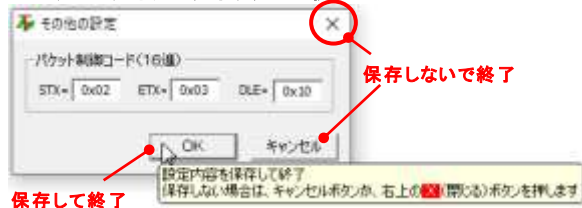
モーダルダイアログボックスとは、当該ダイアログボックスを(OKやキャンセルボタンで)閉じないと他の操作を行うことができないタイプのダイアログを意味します。

このタイプのダイアログには、「OK」ボタンが表示され、「OK」ボタンを押すと設定内容が保存され、「キャンセル」ボタンやウインド右上の「X」ボタンを押すと設定内容は保存されません。

モードレスダイアログボックスとは、当該ダイアログと同時に他の操作が行えるタイプのダイアログを意味します。

このタイプのダイアログには「OK」ボタンは表示されず、ウインド右上の「X」ボタンを押すと設定内容が保存され、「キャンセル」ボタンを押すと設定内容は保存されません。

モーダル ダイアログボックスの例



モードレス ダイアログボックスの例

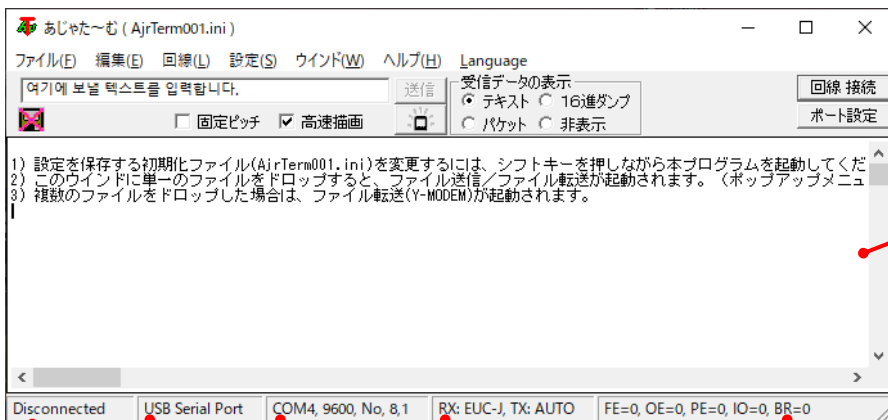


モーダルダイアログボックスと、モードレスダイアログボックスでは、ウインド右上の「X」ボタンでウインドを閉じた場合の動作が異なるので注意が必要です。



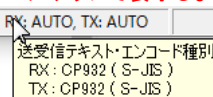
## 2. メインウインド

プログラムを起動すると、以下のメインウインドが表示されます。



ログウインド

※1: AUTO 設定時の実際のエンコードはツールチップで表示します



接続状態

送受信テキストエンコード種別(※1) 受信エラーカウンタ

通信パラメタ(メールスロット時は接続先メールスロット名)  
デバイス名(メールスロット時は自メールスロット名)

メインウインドでは、受信したデータの表示を行います。

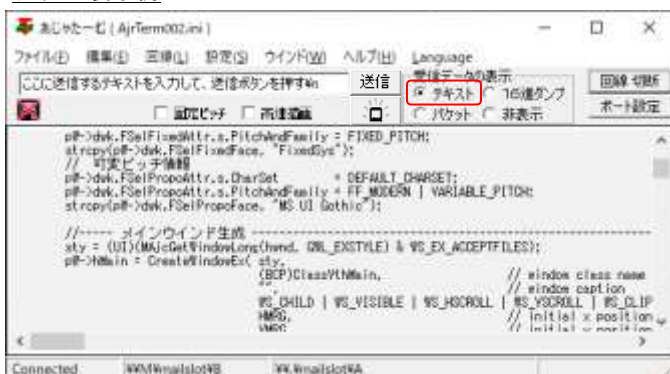
「受信データの表示」で「テキスト」が選択されている場合は、受信したテキストデータをリアルタイムに表示します。

「16進ダンプ」が選択されている場合は、全受信データを16進ダンプ形式でリアルタイムに表示します。

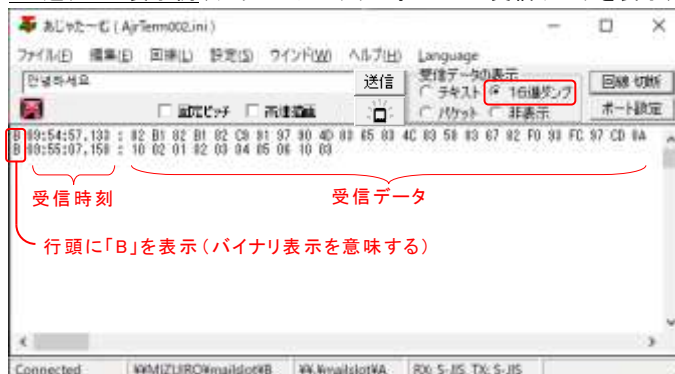
「パケット」が選択されている場合は、タイムスタンプに続いて、受信したパケットデータを16進ダンプ形式で表示します。

「非表示」が選択されている場合は、何も表示しません。

### テキスト表示例



### 16進ダンプ表示例(テキストやパケット等 全ての受信データを表示)

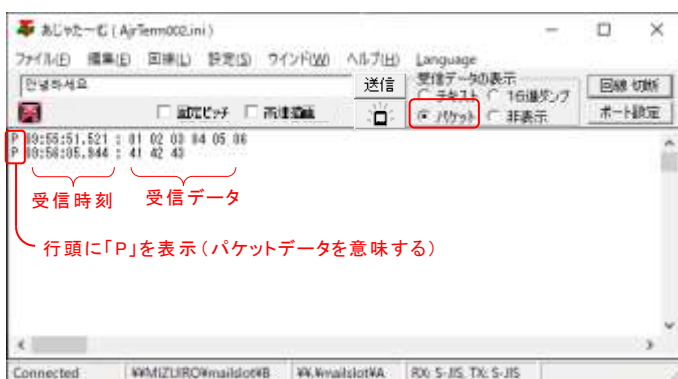


受信時刻

受信データ

行頭に「B」を表示(バイナリ表示を意味する)

### パケット表示例(パケット中のパケットデータ部分だけを16進ダンプ表示します)



受信時刻

受信データ

行頭に「P」を表示(パケットデータを意味する)

**チェックボックス**(「 固定ピッチ」と「 高速描画」)

「固定ピッチ」をチェックすると、強制的に固定ピッチで表示します。

文字によっては、固定ピッチフォントを設定しても固定ピッチとならない場合があります。

「固定ピッチ」をチェックすると、全ての文字を固定ピッチで表示します。(但し、可変ピッチフォントの場合は適用されません)

固定ピッチなし

◆おはようございます・こんにちは・こんばんは  
안녕하세요  
안녕하세요  
안녕하세요

◆さようなら(自分が見送るとき)  
안녕히 가세요  
안녕히 가세요  
안녕히 가세요

◆さようなら(自分が去る時)  
안녕히 계세요  
안녕히 계세요  
안녕히 계세요

固定ピッチあり

◆おはようございます・こんにちは・こんばんは  
안녕하세요  
안녕하세요  
안녕하세요

◆さようなら(自分が見送るとき)  
안녕히 가세요  
안녕히 가세요  
안녕히 가세요


◆さようなら(自分が去る時)  
안녕히 계세요  
안녕히 계세요  
안녕히 계세요

「高速描画」をチェックすると、ログウインドへのテキスト描画を高速に行います。

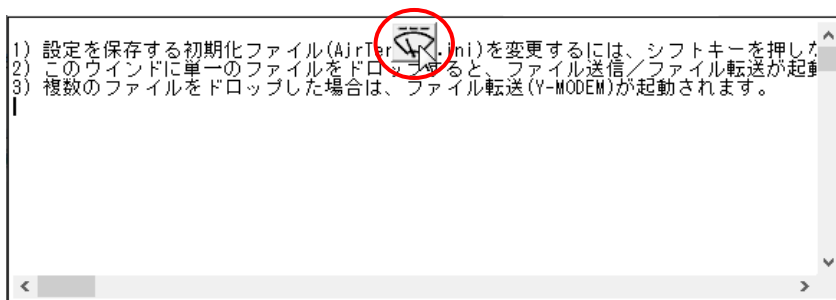
高速に受信しログ表示する場合に有効です。

この場合、文字色は単色(最後に設定された文字色)に、背景色は白に固定されます。



**画面クリアーボタン**


ログウインド中央の上部にカーソルを置くと、画面クリアーボタン「」が表示されます。

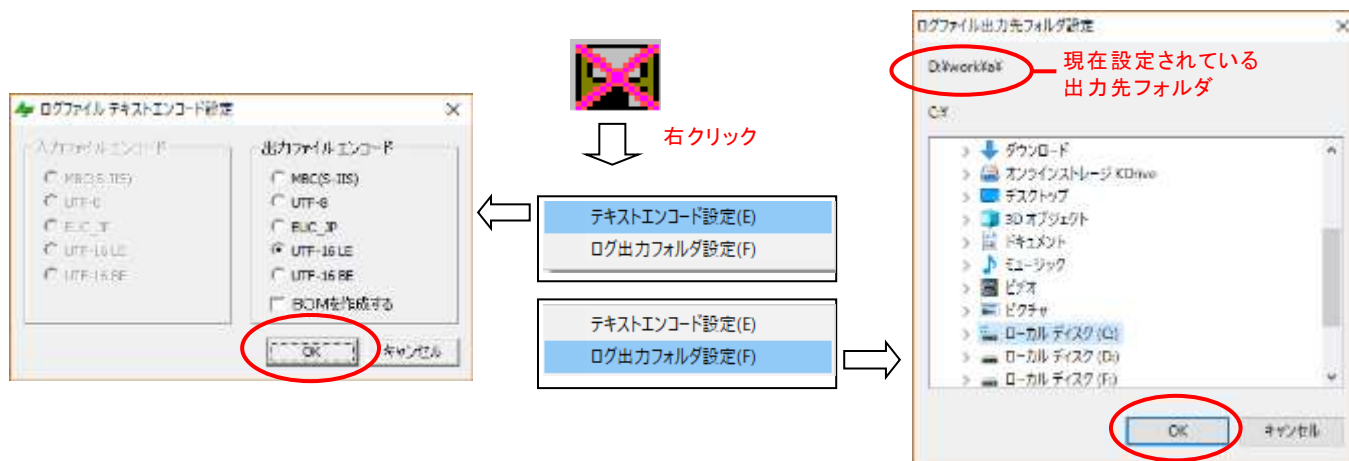
このボタンをクリックすると、画面がクリアーされます。




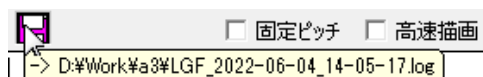
## 2.1. ファイル出力

 は、ファイル出力が停止していることを示します。このアイコンをクリックすると表示が  に変わり、表示内容がファイルにも出力されるようになります。(ファイル名は「 LGF\_yyyy-mm-dd\_hh-mm-ss.log 」となります)

 を右クリックすると、ポップアップメニューにより、出力テキストファイルのエンコードやファイルの書き込み先フォルダを設定できます。

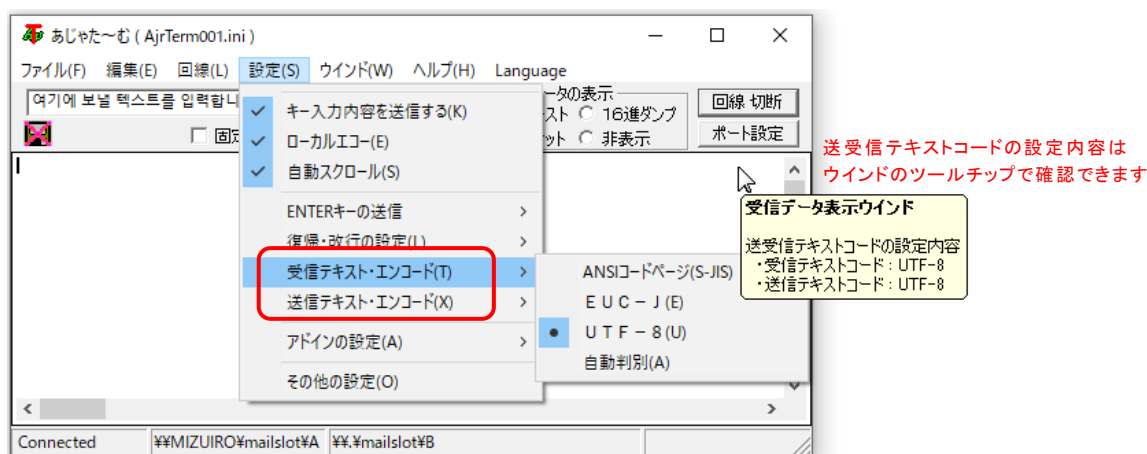


 にカーソルを置くと、ツールチップテキストで、出力ファイルのパス名が、表示されます。



## 2.2. 文字コード

送受信する文字コードは、「設定」メニューの「送信／受信テキスト・エンコード」から「ANSIコードページ(S-JIS)」、「EUC-J」、「UTF-8」から選択できます。「自動判別」を選択した場合は、実際に受信したテキストから、最も確率が高いと思われる文字コードが選択されます。

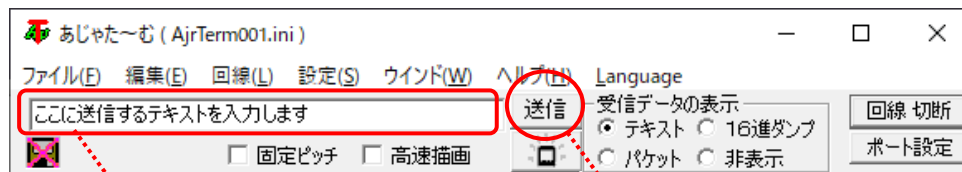


## 2.3. キー入力内容の送信とローカルエコー

「設定」メニューで、「キー入力内容を送信する」を選択した場合は、メインウインド上で入力したキーを送信します。また、「ローカルエコー」が選択されている場合は、送信したキーの内容をメインウインド上に表示します。

## 2.4. テキストの送信

テキストボックスに送信するテキストを入力し、「送信」ボタンを押す(あるいはEnterキーを押す)ことにより、入力したテキストを送信することができます。



ここにテキストを入力します

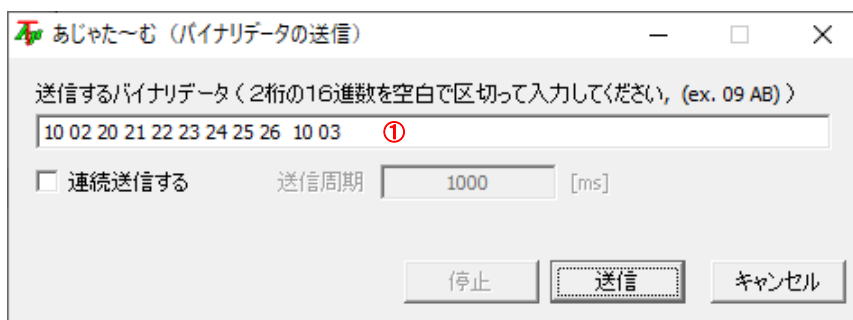
「送信」ボタンを押すと、入力したテキストを送信します。

テキストに以下のエスケープ文字を含めることができます。(ex. 末尾に LF(0x0A)を付加する場合は「xxxx¥n」と入力します。)

¥a - 0x07 (BEL)	¥v - 0x0B (VT)	¥' - 0x27 (')
¥b - 0x08 (BS)	¥f - 0x0C (FF)	¥? - 0x3F (?)
¥t - 0x09 (TAB)	¥r - 0x0D (CR)	¥¥ - 0x5C (¥)
¥n - 0x0A (LF)	¥" - 0x22 (")	
¥xhhh - 16進数 (h = 0-9 / A-F)		
¥ooooo - 8進数 (o = 0-7)		

## 2.5. バイナリデータの送信

「回線」メニューで「バイナリデータの送信」を選択すると、以下のウインドが表示されます。



①へテキストボックスに送信するバイナリデータを2桁の16進数を空白で区切って入力します。( ex. 41 42 OD OA )

「送信」ボタンを押すと、入力したバイナリデータを送信します。

「連続送信」をチェックし、送信ボタンを押すと、「送信周期」で指定された周期で送信を繰り返します。

連続送信中に「停止」ボタンを押すと、送信を停止します。

## 2.6. 受信エラーカウンタ表示

通信回線を COM ポートに設定した場合、メインウインドの右下に受信エラーカウンタを表示します。


FE=7, OE=0, PE=0, IO=0, BR=2

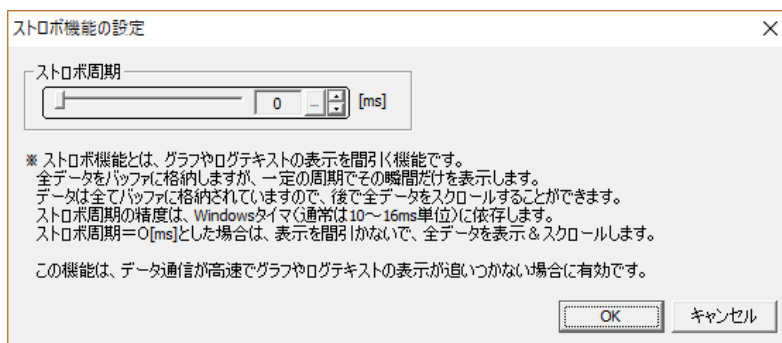
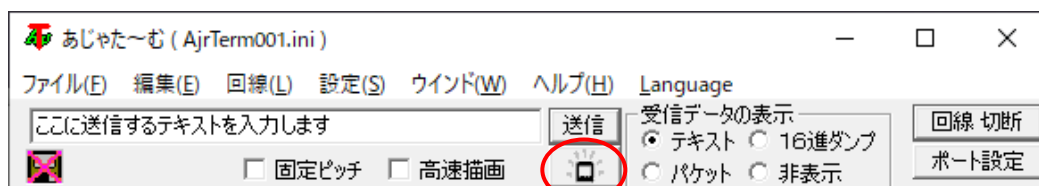
※受信エラーカウンタを表示するには、メインウインドを横に広げてください  
 ※エラーカウンタ表示部分をダブルクリックすると、カウンタがクリアされます

各表示内容は以下の通りです。

表示	内容
FE=n	フレーミングエラー
OE=n	オーバーランエラー
PE=n	パリティエラー
IO=n	I/Oエラー
BR=n	ブレーク信号検出

## 2.7. ストロボ機能

 ボタンを押すと、ストロボ機能の設定ダイアログが表示されます。



ストロボ機能とは、グラフ(タイムチャート、2D/3D プロット)や、ログの表示を間引く機能です。

通常は、データが更新される毎にグラフやログの表示が更新されて必要ならばスクロールしますが、ストロボ機能を有効にすると、データが更新されても表示を更新しないで、指定した周期毎に、その瞬間だけを表示します。

ストロボ周期を 0[ms]とした場合は、ストロボ機能は無効となり、データが更新される度に表示も更新されます。

ストロボ周期に 0[ms]以外を設定した場合は、ストロボ機能が有効となり、指定した周期で表示が更新されます。

ストロボ機能が有効な場合でも、データは更新され続けていますので、後ですべてのデータをスクロールバーによりスクロールして見ることができます。

ストロボ周期は、Windows のタイム精度 (10~16ms 程度) に切り上げられます。

ストロボ機能は、データ通信が高速でグラフやログの表示が追いつかない(表示が重くなる)場合に有効です。

## 2.8. 数値の認識と受信テキストの文字変換

受信テキストから数値を抽出する際は、受信テキストを字句に分解し、「数値定数」と認識されたものだけを数値として認識します。字句の分解方法は、C 言語と同じで、アンダバー(\_)は、英字と同じとみなされます。

尚、数値定数の直前に「+」や「-」がある場合は、当該数値の符号とみなします。

(例)

テキスト	分解される字句	字句の種別	認識される数値
123_456_789	123	数値定数	123
	_456_789	シンボル	
123, 456, 789	123	数値定数	123
	,	デリミタ	
	456	数値定数	456
	,	デリミタ	
-7.89	-	デリミタ	
	7.89	数値定数	-7.89
3.14ABC999	3.14	数値定数	3.14
	ABC999	シンボル	

例えば、「123\_456\_789」という文字列は、「123」(数値定数)と「\_456\_789」(シンボル)という2つの字句に分解され、数値として認識されるのは、「123」だけになります。

このように、都合の悪い文字は、別の文字に変換することにより、全てを数値として認識させることができます。

上記の「123\_456\_789」は「123,456,789」のように、アンダバー(\_)をカンマ(.)に変換することにより「123」「456」「789」の3つの数値を認識させるようにできます。

### 受信テキストの文字変更

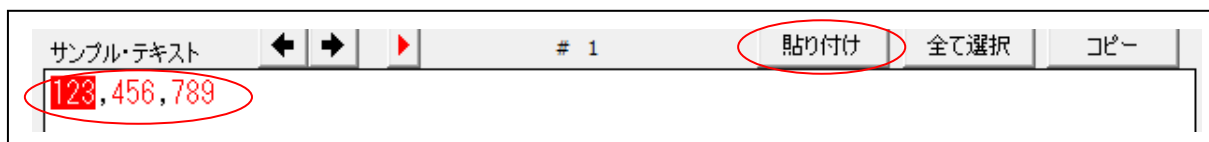
「ウインド」メニューで「数値抽出テキストの文字変更」を選択すると、以下のダイアログが表示されます。



数値抽出用テキストの貼り付け時に、テキスト中のアンダバー(\_)をカンマ(.)に変換します  
変換する文字は、変換前と返還後の2つを入力しなければなりません。  
両方の文字を入力していない場合は矢印( --> )がグレー表示されます。

「文字を変更する」をチェックし、変更する文字のペア(最大4つ)を設定します。

数値抽出用のサンプルテキストで「貼り付け」ボタンを押した場合、テキスト中の文字が変更されます。

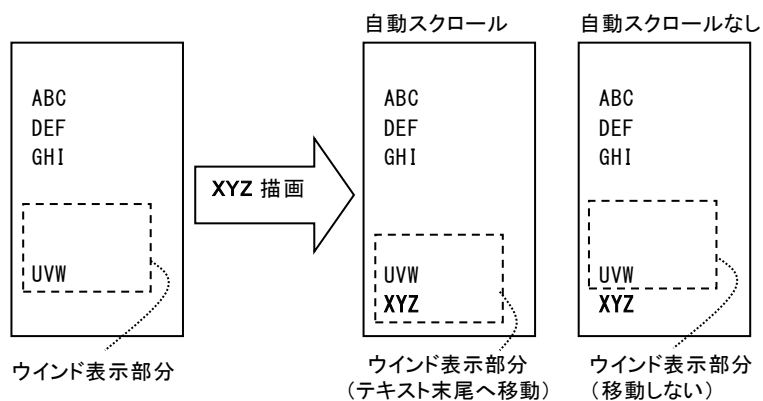


## 2.9. 自動スクロール

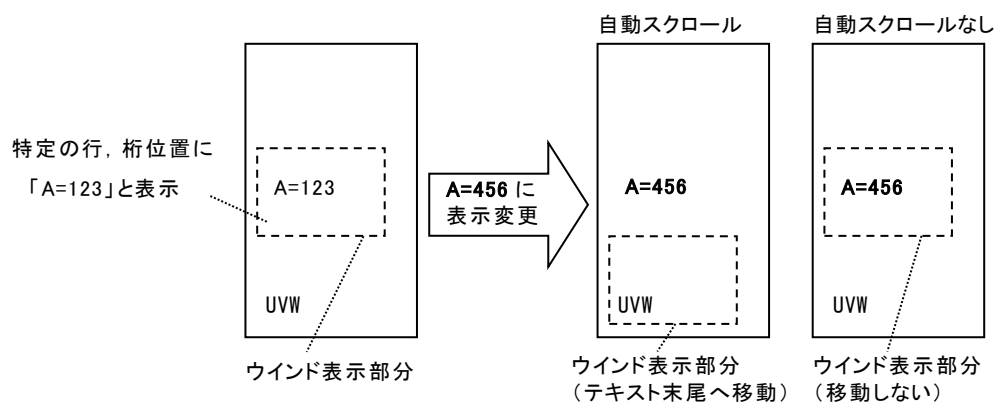
「設定」メニューで、「自動スクロール」が選択されている場合は、データを受信した際に、データ末尾位置まで自動的にスクロールします。

「自動スクロール」が選択されていない場合は、手動でスクロールしない限り、スクロールを行いません。

「自動スクロール」は、ログ表示等で、常に最新の表示部分へ移動する場合に有効です。



「自動スクロールなし」は、例えばスクリーンの固定位置(固定の行、桁位置)に情報を表示する場合に有効です。



## 2.10. ANSIエスケープコード

メインウインドのテキスト表示に関して、以下のANSIエスケープコードをサポートしています。

### ANSIエスケープコード

#	ESCシーケンス	内容
1	ESC・[0J	カーソル位置～最終行の右端までクリアー
2	ESC・[1J	先頭行の左端～カーソル位置までクリアー
3	ESC・[2J ESC・*	画面をクリアーし、カーソルをホームへ移動
4	ESC・[0K ESC・[K	カーソル位置～同行右端までをクリアー
5	ESC・[1K	カーソル行の左端～カーソル位置までをクリアー
6	ESC・[2K	カーソル行をクリアー
7	ESC・[s	カーソル位置を退避
8	ESC・[u	カーソル位置を回復
9	ESC・[>5l	カーソル表示
10	ESC・[>5h	カーソル非表示
11	ESC・[pl;pcH	カーソル位置設定 (pl は行位置 (1～), pc は桁位置 (1～)) ※ 桁位置は全角/半角を問わず、いずれも1桁として計算します。
12	ESC・[pnA	カーソルを上方向に移動 (pn は移動行数であり、省略時は1を仮定)
13	ESC・[pnB	カーソルを下方向に移動 ( " " )
14	ESC・[pnC	カーソルを右方向に移動 (pn は移動桁数であり、省略時は1を仮定)
15	ESC・[pnD	カーソルを左方向に移動 ( " " )
16	ESC・[pnM	カーソル行以降をスクロールアップ (pn はスクロール行数)
17	ESC・[pnL	カーソル行以降をスクロールダウン ( " " )
18	ESC・[psm	描画属性設定 (ps=属性値、0=デフォルト属性、7=反転、30～37(文字色)=黒、赤、緑、黄、青、紫、水色、白、 40～47=背景色)
19	ESC・D	カーソルを1行下へ移動
20	ESC・E	カーソルを1行下の左端へ移動
21	ESC・M	カーソルを1行上へ移動

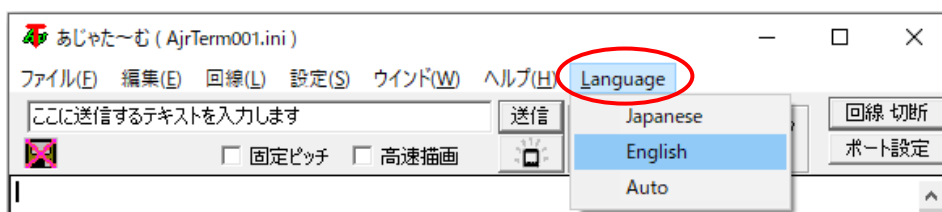
[注] 「ESC・」はエスケープコード(0x1B)を意味します。

## 2.11. 言語設定

「あじやた〜む」は2つの言語(日本語と英語)に対応しています。

デフォルトの言語は、日本語 Windows の場合は日本語、その他の Windows では英語となります。

言語の設定は、「Language」メニューで切り替えることもできます。(言語設定は、次回起動時から有効となります)

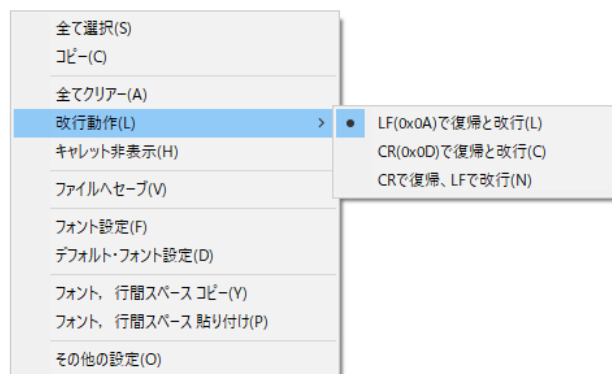


※ 英文は、一般の翻訳ソフトで、日本語→英語変換したものですので、あまり読みやすい英文とはなっていませんが、日本以外でも、とりあえずのアルファベットで読めればとの目的でサポートしています。



## 2.12. ポップアップメニュー

コントロール・ウインド上で右クリックすると、以下のポップアップメニューが表示されます。



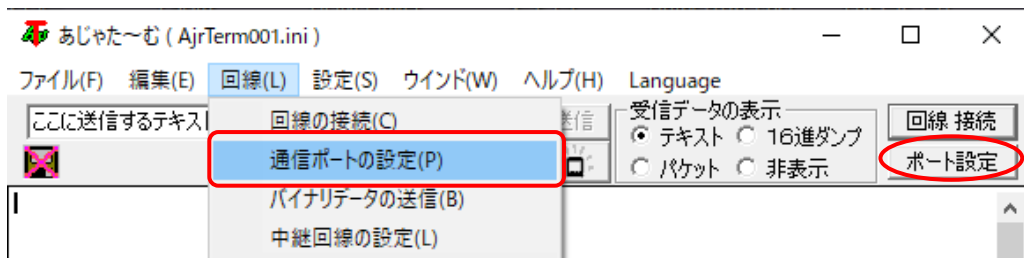
各メニューの内容は、以下のとおりです。

#	メニュー	内容
1	全て選択	全てのテキストを選択状態にします
2	コピー	選択されているテキストをクリップボードへコピーします (テキストが選択されていない場合、このメニューは表示されません)
3	全てクリアー	全てのテキストを破棄し、画面を消去します。
4	改行動作	CR、LF での改行動作を選択します
5	キャレット非表示 (※1)	キャレット(点滅文字カーソル)を非表示にします。 次回は「キャレット表示」メニューに変わります
6	ファイルへセーブ	現在表示／選択しているテキストをファイルへ出力します
7	フォント設定	フォントの設定ダイアログを表示します
8	デフォルトフォント設定	デフォルトフォントを設定します。(設定に際して確認メッセージが表示されます)
9	フォント、行間スペースコピー	設定されているフォントと行間スペース情報をクリップボードへコピーします
10	フォント、行間スペース貼り付け	クリップボード上のフォントと行間スペース情報を設定します
11	その他の設定	各種設定ダイアログを表示します

### 3. 通信回線の設定

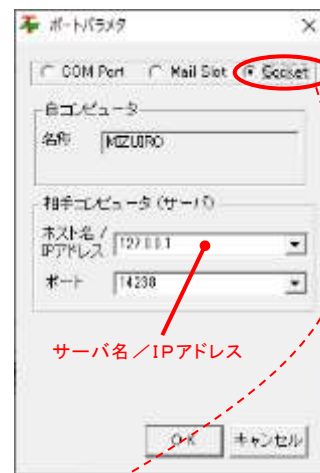
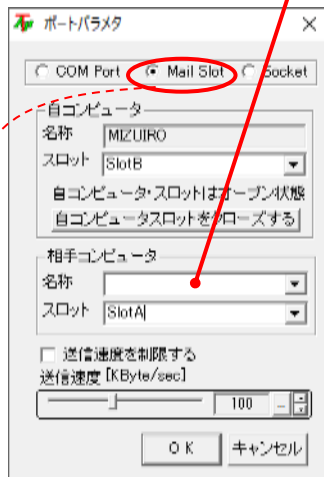
「あじやた〜む」では、シリアル回線(COMポート)、LAN(メールスロット)とソケット(TCP/IP)による通信が可能です。

COMポート/メールスロット/ソケットの選択は、「ポート設定ボタン」か、あるいは、「回線メニュー」→「通信ポートの設定」からラジオボタン(「Com Port」、「Mail Slot」、「Socket」、「Socket」)で選択します。



COMポートのデバイス名(デバイスマネージャで表示される名称)

相手コンピュータ名を空白とした場合は、  
自コンピュータ内のプロセス間通信となります。

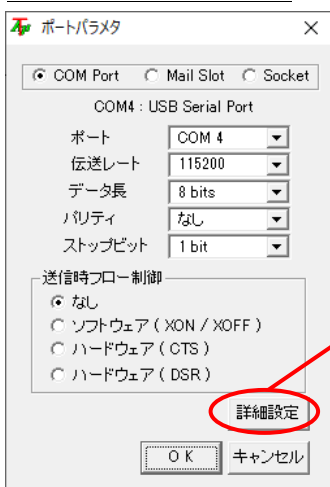


サーバ名/IPアドレス

ラジオボタンで「ComPort」、「MailSlot」 or 「Socket」を切り替える

COMポート設定ダイアログ中で「詳細設定」ボタンを押すとCOMポートの詳細設定ダイアログが表示されます。

COMポートの設定ダイアログ



COMポートの詳細設定ダイアログ



「伝送レート」は、コンボボックスから目的の値を選択するか、あるいは(選択肢に希望する値が無い場合)コンボボックスに直接値を入力してください。

「OK」ボタンを押すと入力した内容が設定され、「キャンセル」ボタンを押すと入力した内容が破棄されます。

「Mail Slot」を選択した場合、相手コンピュータのコンピュータ名と両スロットを入力し、「OK」ボタンを押します。

尚、自コンピュータ上で他のプロセスと通信する場合は、相手コンピュータ名を空白とし、自コンピュータ(自プロセス)のスロット名と相手コンピュータ(相手プロセス)のスロット名を入力してください。この場合、スロット名は異なる名称でなければなりません。

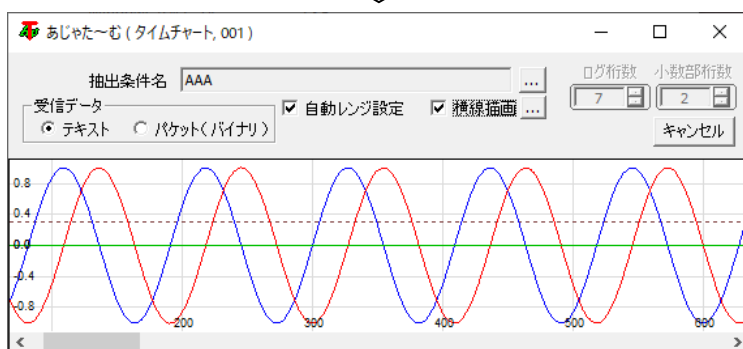
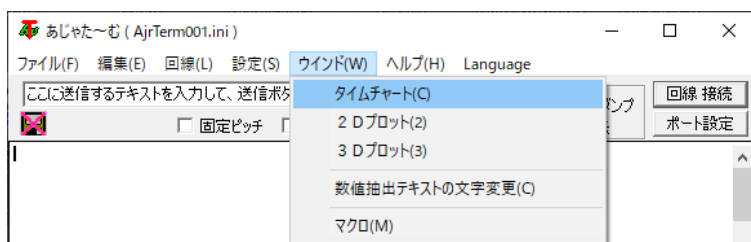
「通信速度を制限する」をチェックした場合、擬似的に、「送信速度」で設定した速度(キロバイト/秒)でデータの送信を行います。(ウェイトを挿入し、平均的な転送速度を、指定した値となるように制御します)

「OK」ボタンを押すと入力した内容が設定され、「キャンセル」ボタンを押すと入力した内容が破棄されます。

## 4. タイムチャート（波形グラフ）の表示

受信したテキストやパケットから数値を抽出し、タイムチャート・グラフを表示することができます。



タイムチャート・グラフを表示するには、「ウインド」メニューから「タイムチャート」を選択します。



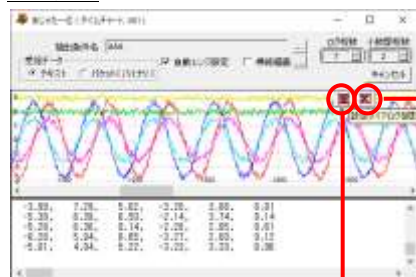
「自動レンジ設定」をチェックすると、受信データの最小値～最大値となるようにグラフのレンジを自動的に設定します。

「抽出した数値をログ表示」をチェックすると(下左図のように)ウインド下部に数値がログ表示されます。

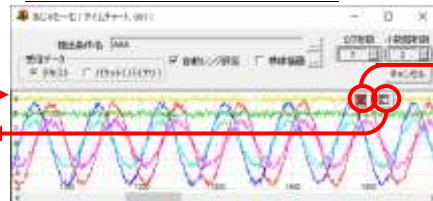
ログ表示する数値の表示桁数は、「ログ桁数」と「小数部桁数」で設定します。

カーソルをチャートウインドの右上部分に置くとボタン(   )が表示され、ダイアログやログウインドを表示／非表示できます。

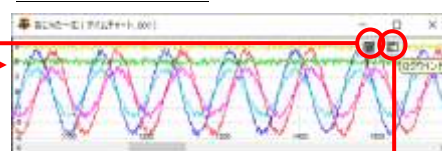
全て表示



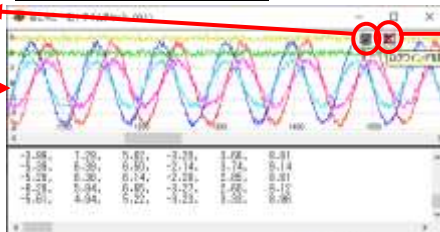
設定ダイアログとチャート・グラフ



チャート・グラフのみ



チャート・グラフとログ表示



#### 4.1. 受信データの選択と数値認識方法の設定

数値を抽出する受信データは、テキストデータとパケットデータのいずれかを選択できます。

##### テキストデータを選択



##### パケットデータを選択



「テキスト」を選択した場合は、テキストデータ行中の、何番目の数値語句を抽出するかを設定します。

「パケット(バイナリ)」を選択した場合は、パケットデータ中のバイト位置と、数値のタイプで指定します。

##### 4.1.1. 受信テキストデータから数値を抽出する場合

既に数値抽出条件が設定されている場合は、「…」ボタンから、コンボボックスで設定されている条件を選択し「OK」ボタンを押してください。



コンボボックスから目的の数値抽出条件を選択します



新たに数値抽出条件を設定する場合は、以下の手順で行います。

### 1) サンプル・テキスト行の設定

まず最初に、数値を取り出すサンプル・テキスト行を設定します。

何らかの方法でサンプルテキスト行をクリップボードへコピーしておきます



「貼り付け」ボタンを押すと、サンプルテキスト行が設定されます。

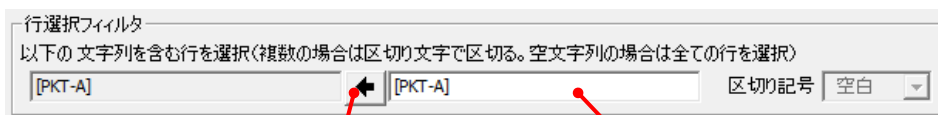


赤字は認識された数値項目を意味します。

赤反転は現在選択されている項目で、←→ボタンで移動できます。

### 2) 行選択フィルタ

何種類かあるテキスト行から、目的のテキスト行を選択する場合は、「行選択フィルタ」を設定します。



上記の例では、“[PKT-A]”という文字列が含まれるテキスト行だけが処理対象となります。

フィルタ文字列を複数設定する場合は、「区切り記号」で区切って複数の文字列を入力します(ex. “[PKT-A] [TYP1]”)

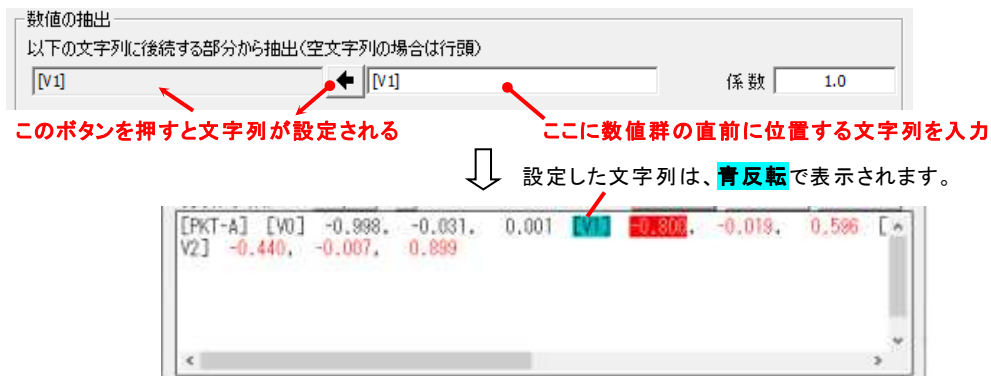
この場合は、指定した複数の文字列が全て含まれるテキスト行が処理対象となります。

行選択フィルタを空文字列とした場合は、全てのテキスト行が処理対象となります。

### 3) 抽出する数値の位置を指定

抽出する数値群の直前に位置する文字列を指定することにより、当該文字列以降から数値を認識することができます。

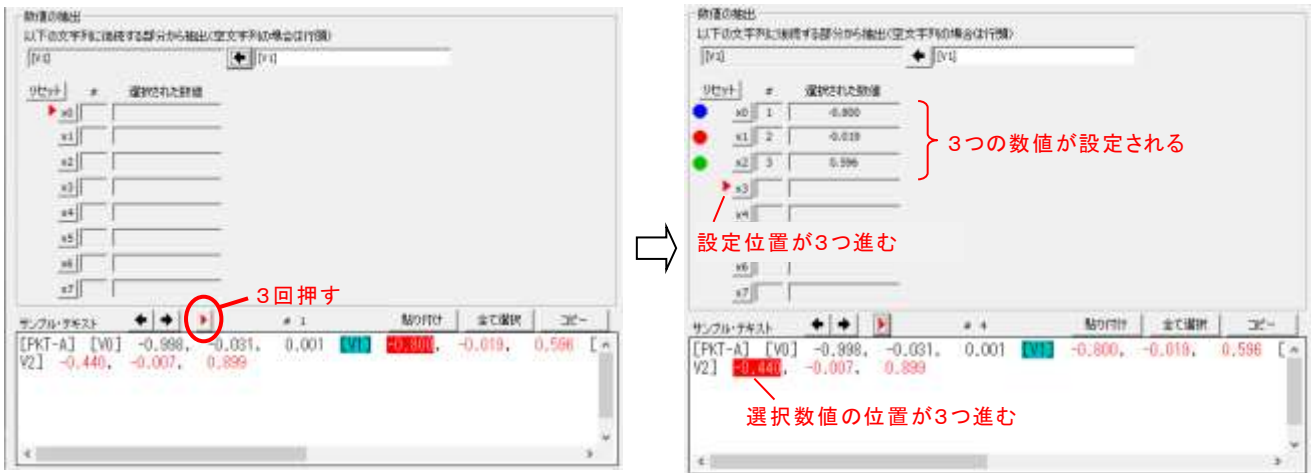
以下の例では、文字列「[V1]」以降の数値を認識するように設定しています。



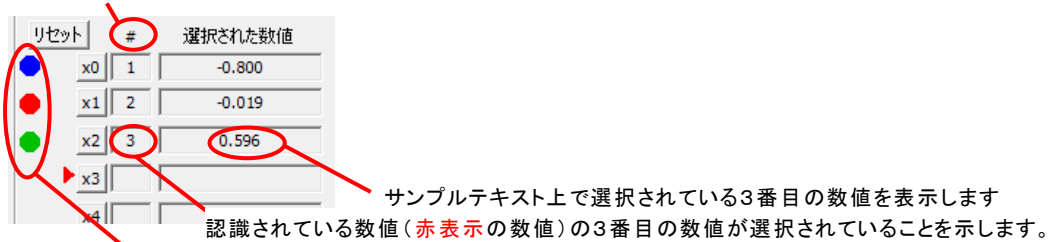
#### 4) 抽出する数値の設定

対象とする数値は、**←** **→** ボタンで選択し、**x0** ~ **x7** ボタンで当該ボタンの位置へ設定します。  
 または、**▶** ボタンを押すと、**▶** で示される位置へ設定し、設定位置と数値の選択が1つ進みます。

例えば、下左の状態では **▶** ボタンを3回押すと、下右図のようになります



「#」は、何番目の数値が選択されているかを示します。



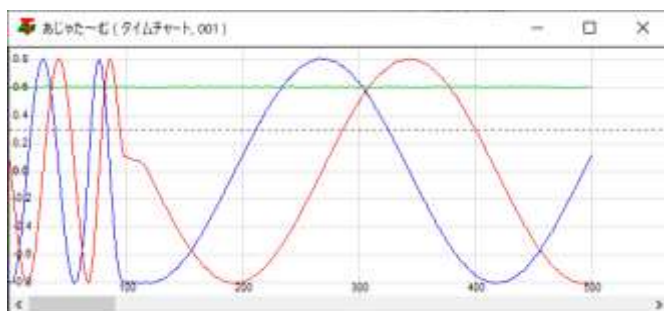
サンプルテキスト上で選択されている3番目の数値を表示します  
 認識されている数値 (赤表示の数値) の3番目の数値が選択されていることを示します。

有効な数値が設定されている場合のインジケータです。この円の色は、グラフィック上での描画色となります。

「リセット」ボタンを押すと、全ての数値選択設定がクリアされます。(上記、左図のようになります)

設定が完了したら「OK」ボタンで上書きするか、「保存」ボタンで名前を付けて保存を行ってください。

上記の設定例では、受信したテキスト行から3つの数値が抽出されて、以下のようなタイムチャート・グラフ (波形) が表示されます。



## 4.1.2. パケットデータから数値を抽出する場合

既に数値抽出条件が設定されている場合は、「...」ボタンから、コンボボックスで設定されている条件を選択し「OK」ボタンを押してください。





新たに数値抽出条件を設定する場合は、以下のように設定します。

バイナリ・パケットデータの「パケット選択フィルタ」と、抽出する数値の位置とタイプを指定します。

例えば、以下のようなバイナリデータの場合、下図に示すように設定します。

**バイナリデータ**

Loc=0	1	2	3	4	8	12
0x01	-	-	-	x0 (INT-32)	x1 (INT-32)	x2 (INT-32)



**ダイアログの設定**

抽出条件名 AAA [削除] [ヘルプ]

パケット選択フィルタ

位置	タイプ	値	位置	タイプ	値	位置	タイプ	値
0x0	Hex8	0x1	0x0	*****	0	0x0	*****	0

数値の抽出

	位置	タイプ
X0	0x8	Int32
X1	0xC	Int32
X2	0x10	Int32
X3	0x0	*****
X4	0x0	*****
X5	0x0	*****
X6	0x0	*****
X7	0x0	*****

演算式: N/2

係数: 0.001  ビッグエンディアン

[リセット] [保存] [OK] [キャンセル]

ローケーション 0 が、0x01 (8ビット整数)であるバイナリ・パケットデータを対象とします

「リセット」ボタンを押すと、全ての数値選択設定がクリアされます。

各々、ローケーション4, 8, 12の位置の符号付32ビット整数を指定します

有効な数値が設定されている場合のインジケータです。この円の色は、グラフィック上での描画色となります。

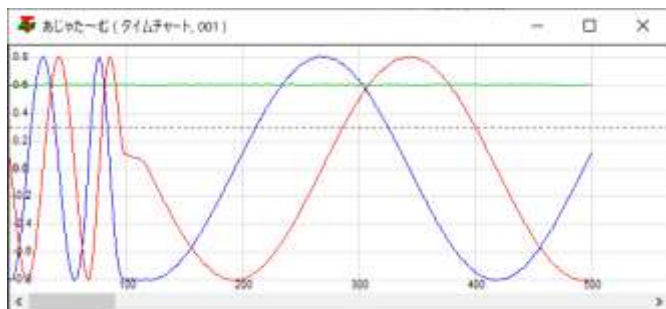
抽出した数値を「N」として、計算式を指定できます。

数値がビッグエンディアン形式である場合にチェックします。

「係数」はグラフ表示時、抽出したすべての数値に乗算する値です

設定が完了したら「OK」ボタンで上書きするか、「保存」ボタンで名前を付けて保存を行ってください。

上記の設定例では、受信したバイナリ・パケット・データ行から3つ数値が抽出されて、以下のようなタイムチャート(波形)グラフが表示されます。



## 4.2. ポップアップメニュー

グラフ上で右クリックすると、以下のポップアップメニューが表示されます。

グラフ上で右クリックすると、以下のポップアップメニューが表示されます。



- ストップ : グラフ表示を停止します。次回は「スタート」メニューとなります。
- コピー : グラフ表示内容(ビットマップ)をクリップボードへコピーします。
- レンジ設定 : グラフのレンジ(上限値, 下限値)を設定します。
- レンジ自動調整 : プロットデータからグラフのレンジを自動算出して設定します。
- オフセット設定 : プロットデータに加算するオフセット値を設定します。
- その他の設定 : 平均化個数, タイムスケール幅, バッファに格納するデータ数を設定します。
- フィルタ非表示 : コントロール左上のフィルタ(チェックボックス)を非表示にします。  
次回は「フィルタ表示」メニューに変わります。
- スケールライン非表示 : 目盛り線(薄いグレーの線)を非表示にします。  
次回は「スケールライン表示」メニューに変わります。
- スケール値非表示 : 目盛り数値を非表示にします。  
次回は「スケール値表示」メニューに変わります。
- 波形の補間表示設定 : 波形の補間表示に関するパラメタを設定します
- 波形の補間表示ウインド: 波形を補間して表示するウインドを開きます。
- データクリアー : バッファリングされているデータを全て破棄し、画面をクリアーします。
- 描画速時間情報表示 : グラフィックイメージの描画時間を計測し表示します

## 4.3. レンジ設定

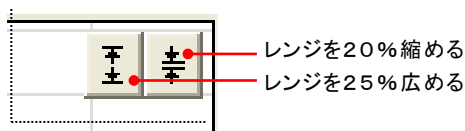
ポップアップメニューで「レンジ設定」を選択すると、以下のダイアログボックスが表示されます。



ここで、レンジ値を入力し、「OK」ボタンを押すと、グラフのレンジが設定されます。「Cancel」ボタンを押すと設定を中止します。

## 4.4. ワンタッチでレンジ設定

マウスマウスカーソルをタイムチャートグラフの右上隅に置くと、2つのボタンが表示されます。これらのボタンで、レンジを25%広めたり、20%縮めたりすることができます。



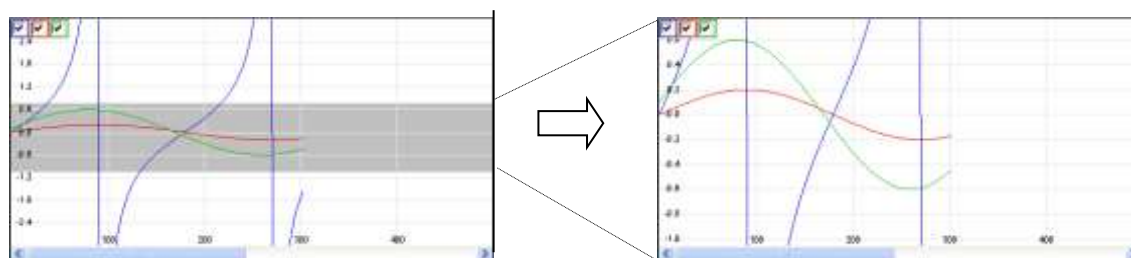
#### 4.5. レンジ自動調整

ポップアップメニューで「レンジ自動調整」を選択すると、(フィルタで非表示となっているデータ項目を除く)全てのデータから最小値と最大値を算出し、±5%のマージンを持ってレンジ設定を行います。



#### 4.6. ドラッグ操作によるレンジ設定

CTRL キーを押しながら、マウス左ボタンで、レンジ設定したい部分をドラッグすることにより、レンジの設定を行うことができます。



レンジ設定する部分を、CTRL キーを押しながらマウスでドラッグ  
(ドラッグされている部分はグレー表示されます)

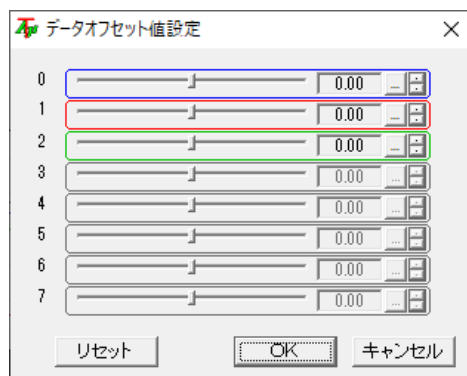
CTRL キーを押したまま、マウス左ボタンを離すと、  
ドラッグした部分がレンジ設定されます。

グラフの上端／下端を越えた部分までドラッグしても、当該ドラッグ範囲がレンジとして設定されます。

CTRL キーを先に離して、マウス左ボタンを離した場合は、レンジ設定は行われません。

#### 4.7. オフセット設定

ポップアップメニューで「オフセット設定」を選択すると、以下のダイアログが表示されます。



各0～7の項目は、表示されているデータ項目に対応します。(外枠の表示色がグラフ表示色と同じになっています)

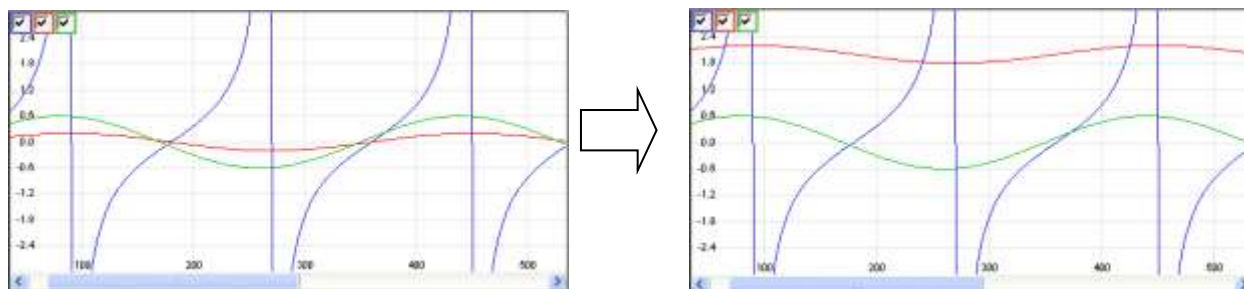
ここで、値を設定すると、当該データ項目のデータ値に、設定値を加算した値でグラフが表示されます。

値の設定に追従して設定したオフセット値がグラフに反映されます。

「OK」ボタンを押すと設定内容が確定されます。「Cancel」ボタンを押すと設定内容は破棄され、元のオフセット値に戻ります。

「リセット」ボタンを押すと、全てのオフセット値が「0」に設定されます。

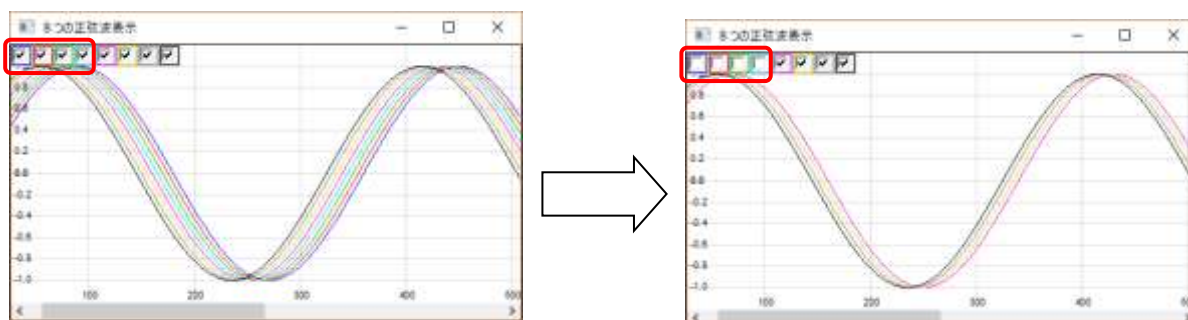
以下の例は、赤色表示のデータに、オフセット値として「+2.0」を設定したものです。



#### フィルタ機能

マウスマウスカーソルをタイムチャートグラフの左上隅に置くと、チェックボックスが表示されます。

ウインド左上のチェックボックスは、表示フィルタであり、チェックを外すと当該表示色のデータは非表示となります。



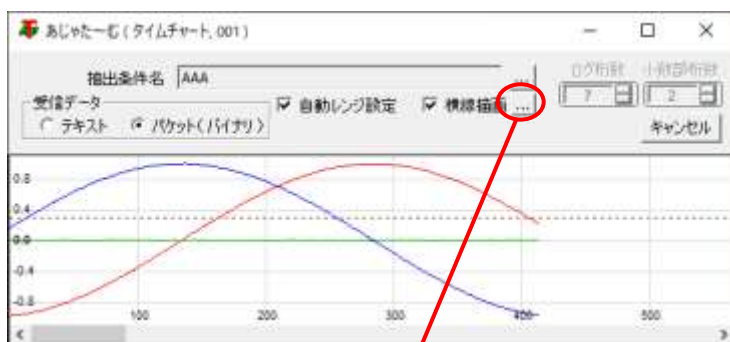
#### 4.8. 横線の描画

波形グラフ上で、最大8個の横線を描画することができます。

横線を描画することで、ボーダーライン(例えば、許容値の最小値や最大値)を視覚的に表示できます。

グラフ上に横線を描画するには、「横線描画」チェックボックスをチェックし、「...」ボタンにより横線の描画条件を設定します。

以下の例では、-0.3と+0.3の位置に緑色の点線を表示しています。



横線の表示位置(レベル)を入力

横線の種類を選択(実践, 点線, 1点鎖線, 2点鎖線)

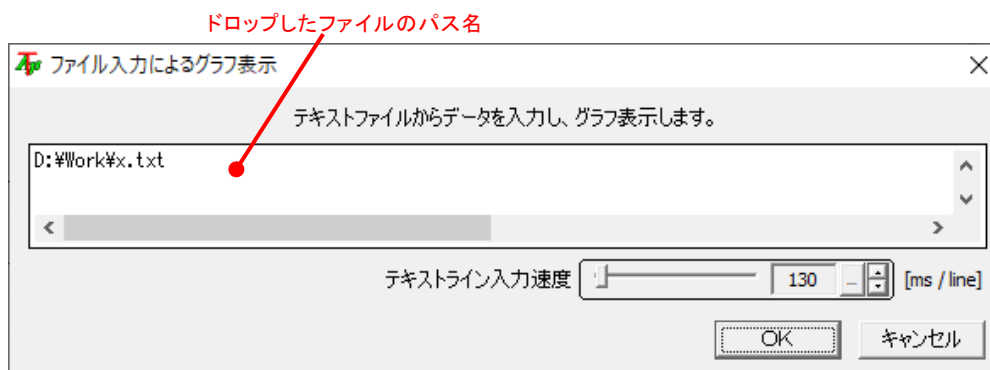
線の太さ(実線以外の場合は「1」に設定すること)

横線の描画色設定ボタン

横線の表示/非表示のチェック

#### 4.9. テキストファイルを入力し波形グラフ表示

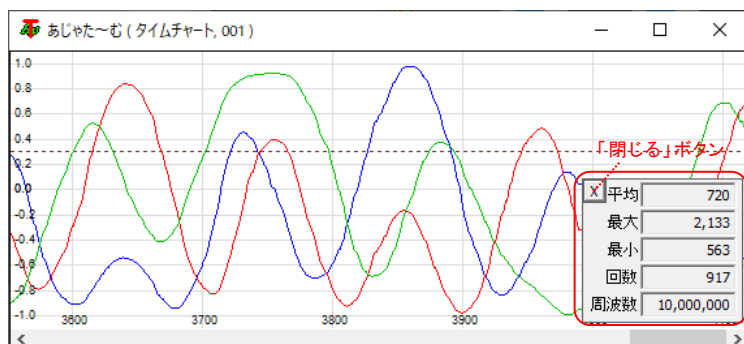
波形グラフウインドにテキストファイルをドロップすることにより、テキストファイルからデータを抽出して波形グラフ表示ができます。  
波形グラフウインドにテキストファイルをドロップすると、以下のダイアログが表示されます。



「テキストライン入力速度」は、テキストファイルから1行を読み出す速度です。  
ここで「OK」ボタンを押すと、テキストファイルからデータを抽出し、波形グラフを表示します。  
「キャンセル」ボタンを押すとテキストファイルの読み出しを中止します。

#### 4.10. 描画時間情報表示

タイムチャートグラフを右クリックし、ポップアップメニューから「描画時間情報表示」を選択すると、タイムチャート・イメージの描画時間を計測し、右下に以下のように表示します。



平均: 1回の描画に要する時間の平均[ $\mu$ s]  
最大: 最大描画時間[ $\mu$ s]  
最小: 最小描画時間[ $\mu$ s]  
回数: 計測回数  
周波数: 計測周波数[Hz] (PCで固定な値)

ウインドのサイズを変更した場合は、計測をやり直します。

短い周期でデータを投与すると、処理が重くなり、タイムチャートイメージをスムーズに表示できなくなります。

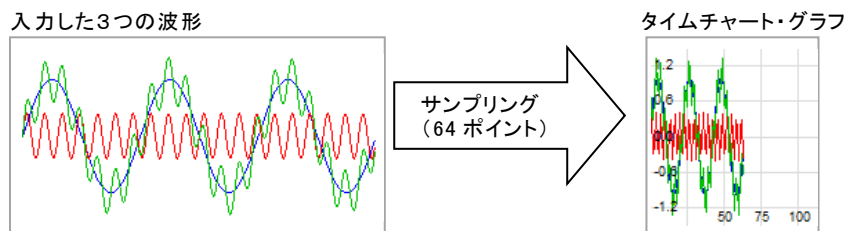
処理時間は、描画時間だけではありませんが、データを周期的に表示する場合は、少なくとも、平均値よりも長い周期で表示する必要があります。

#### 4.11. 波形の補間表示

サンプル数が少なく、グラフに波形を正常に表示できない場合、サンプリングしたデータの間を補間することにより、本来の波形を再現して表示することができます。(3次スプライン曲線(サンプリング点を通る曲線)による補間表示)

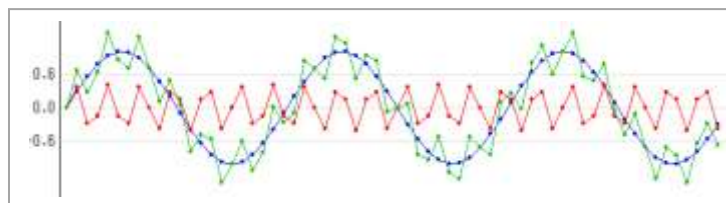
以下に、波形補間表示の例を示します。

下図は、3つの波形を等間隔にサンプリングし、タイムチャートグラフで表示したものです。



ここでは、(タイムチャート・グラフから)元の波形を見ることはできません。

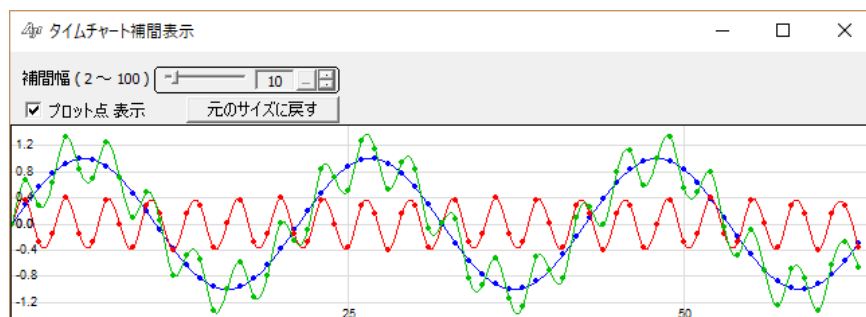
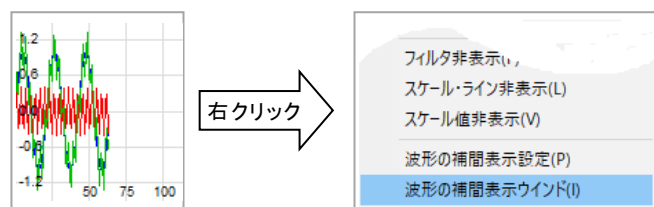
そこで、グラフを横に引き伸ばして、プロット点の間を線で結んでみます。(下図)



低い周波数の波形は、それなりに表示できますが、高い周波数の波形は依然として表示できていません。

(タイムチャート・コントロールには、横に引き伸ばしてプロット点を線で結ぶ機能はありません。上図は別途作成したものです。)

今度は、タイムチャートグラフを右クリックし、ポップアップメニューから「波形の補間表示ウインド」を選択します。



グラフを横に引き伸ばして、プロット点の間を(直線ではなく)曲線で補間したグラフが表示されます。

グラフ上の点は、プロットしたデータ(サンプリングデータ)を示します。

「プロット点表示」のチェックを外すと、プロット点を消去したグラフを表示します。(線だけの表示となります)

「補間幅」はプロット点の表示間隔(ピクセル数)です。(つまり、グラフを横に引き延ばす倍率となります)

「補間幅」を変更すると、プロット点の表示間隔を変更したグラフを再表示します。

「タイムチャート補間表示」ウインドは、初回表示時は(なるべく)元のグラフと同じサイズになるように表示し、以降、自由にサイズを変更することができます。

「元のサイズに戻す」ボタンを押すと、ウインドのサイズを初回に表示した時のサイズに設定し直します。

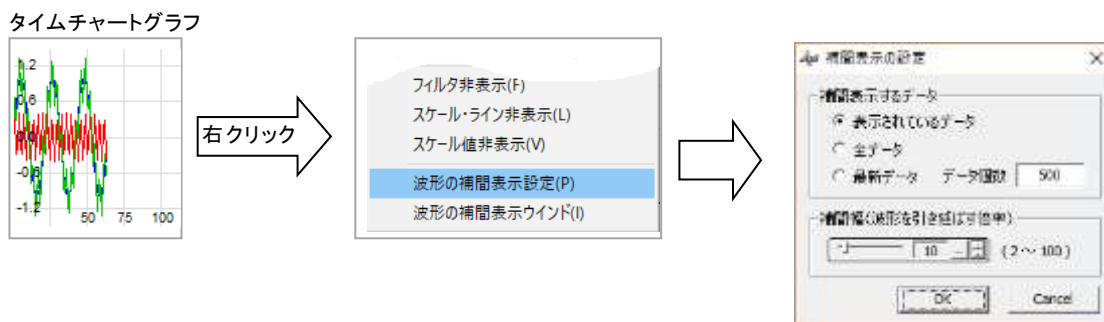
「タイムチャート補間表示」ウインドは、ポップアップメニューから「波形の補間表示ウインド」を選択した時点のデータを切り取って補間表示します。元のタイムチャートグラフを更新しても、補間表示は更新されません。

補間表示を更新するには、「タイムチャート補間表示」ウインドを一旦閉じて、再度表示し直してください。

尚、補間表示対象データが少ない(8個未満)の場合は、補間表示できません。

#### 4.12. 波形補間表示の設定

波形の補間表示に関するパラメタを設定するには、タイムチャートグラフを右クリックし、ポップアップメニューから「波形の補間表示設定」を選択します。



設定内容は、以下のとおりです。


項目		内容
補間表示するデータ (補間表示対象とするデータの選択)	表示されているデータ	タイムチャート・コントロールで表示しているデータ
	全データ	バッファに格納されている全データ
	最新データ	最新のプロットデータ群
データ個数		「最新データ」選択時の、データ個数
補間幅		プロット点の表示間隔(ピクセル数)



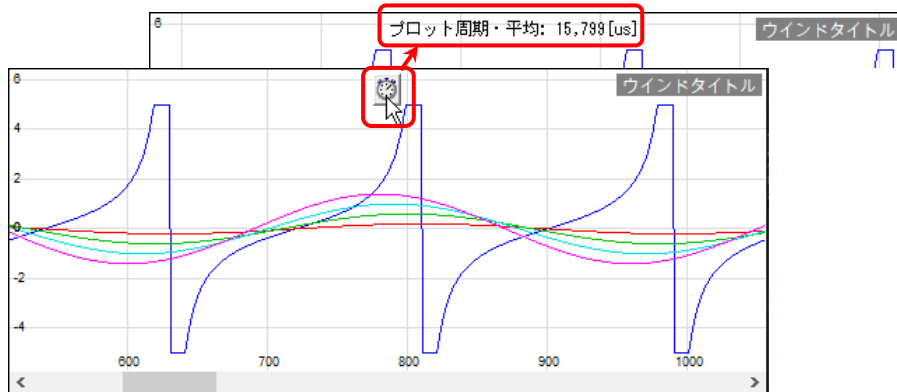
#### 4.13. プロット周期表示

チャートグラフでは、常時プロット周期を計測しています。

プロット周期とは、グラフを描画するデータの受信間隔を意味します。

ウインド中央上部にカーソルを置くと現れる「」ボタンを押すと、ボタンを押した時点のプロット周期(平均値)が表示されます。

プロット周期は5秒間だけ表示後、自動的に消えます。




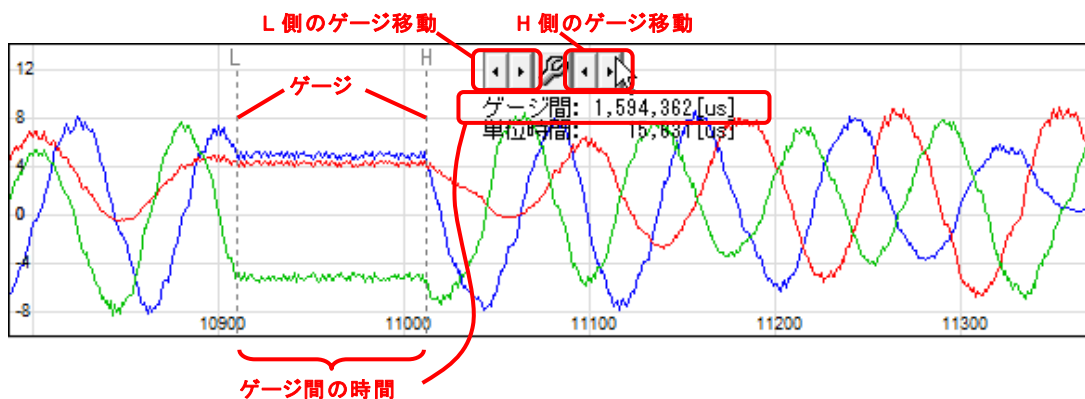
マウスのホイールボタンを押すと、プロット周期の計測をリセットします。


#### 4.14. 時間計測

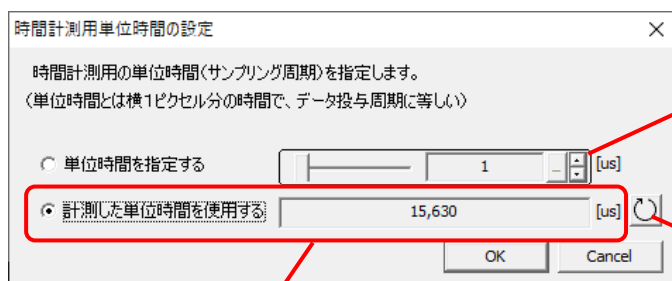
2つのゲージを表示し、ゲージ間の時間を計測することができます。

Ctrl キーを押しながら、マウスのホイールボタンを押すと2つのゲージ(縦の点線)が表示されます。

ここで、ウインド中央上部にカーソルを置くと現れる2つの「」ボタンでゲージを移動します。



「」ボタンを押すと、以下のダイアログにより単位時間(プロット周期)を設定することができます。



計測された単位時間を使用しない場合は「単位時間を指定する」をチェックし、独自の単位時間を指定します

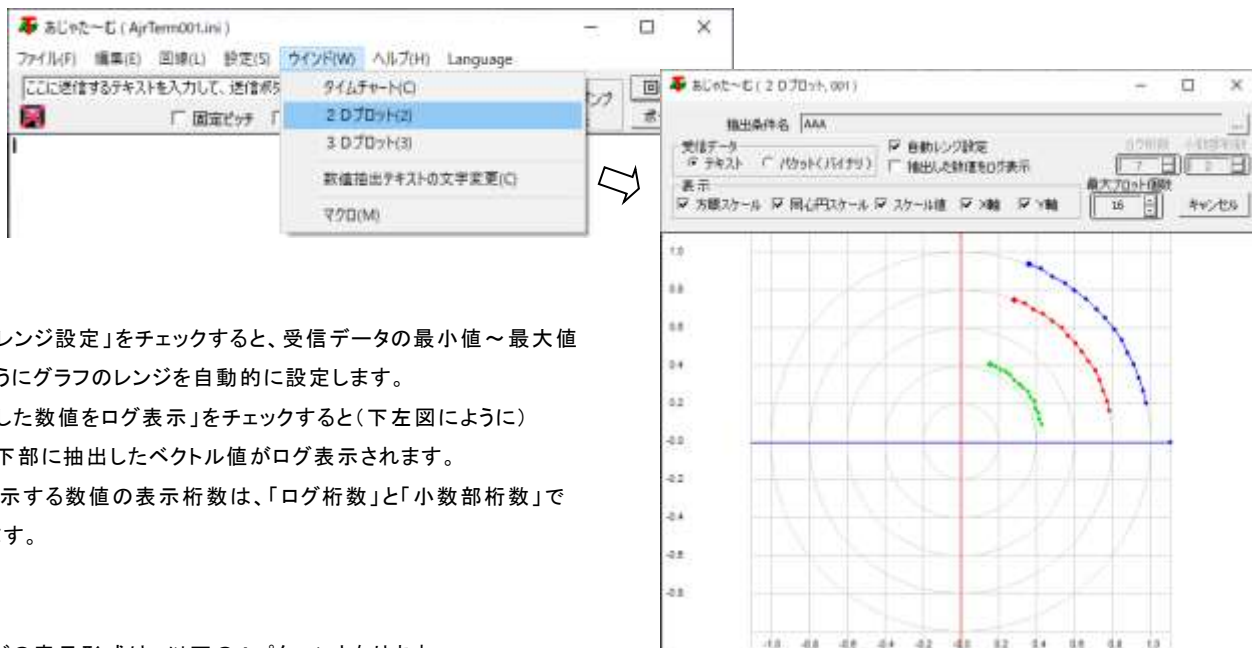
単位時間を計測し直す

自動的に計測された単位時間

## 5. 2Dプロットグラフの表示

受信したテキストやパケットから2Dベクトル値を抽出し、2Dプロット・グラフを表示することができます。

2Dプロット・グラフを表示するには、「ウインド」メニューから「2Dプロット」を選択します。





「自動レンジ設定」をチェックすると、受信データの最小値～最大値となるようにグラフのレンジを自動的に設定します。

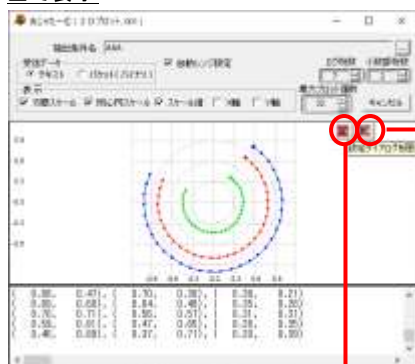
「抽出した数値をログ表示」をチェックすると(下左図のように)ウインド下部に抽出したベクトル値がログ表示されます。

ログ表示する数値の表示桁数は、「ログ桁数」と「小数部桁数」で設定します。

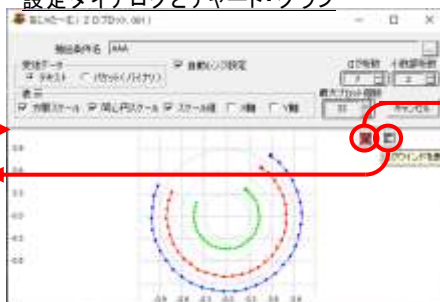
ウインドの表示形式は、以下の4パターンとなります。

カーソルをチャートウインドの右上部分に置くとボタン(   )が表示され、ダイアログやログウインドを表示／非表示できます。

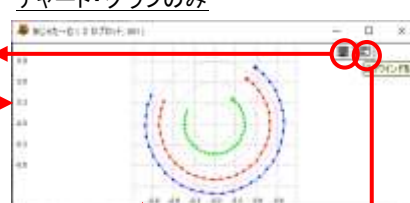
全て表示



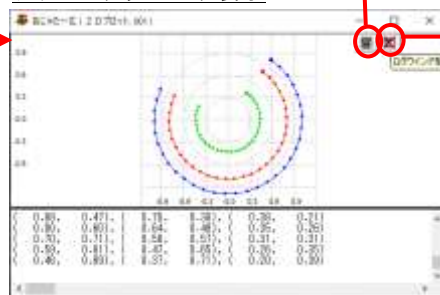
設定ダイアログとチャート・グラフ



チャート・グラフのみ



チャート・グラフとログ表示



## 5.1. 受信データの選択と数値認識方法の設定

数値を抽出する受信データは、テキストデータとパケットデータのいずれかを選択できます。

### テキストデータを選択



### パケットデータを選択

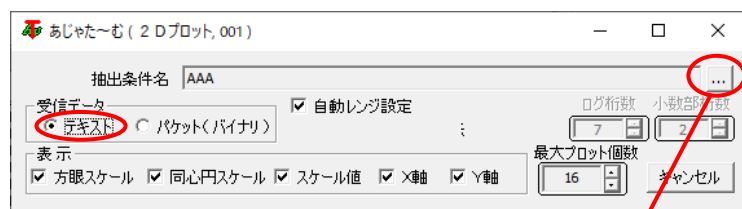


「テキスト」を選択した場合は、テキストデータ行中の、何番目の数値語句を抽出するかを設定します。

「パケット(バイナリ)」を選択した場合は、パケットデータ中のバイト位置と、数値のタイプで指定します。

### 5.1.1. 受信テキストデータから数値を抽出する場合

既に数値抽出条件が設定されている場合は、「…」ボタンから、コンボボックスで設定されている条件を選択し「OK」ボタンを押してください。



コンボボックスから目的の数値抽出条件を選択します



新たに数値抽出条件を設定する場合は、以下の手順で設定します。

### 1) サンプル・テキスト行の設定

まず最初に、数値を取り出すサンプル・テキスト行を設定します。

何らかの方法でサンプルテキスト行をクリップボードへコピーしておきます



「貼り付け」ボタンを押すと、サンプルテキスト行が設定されます。

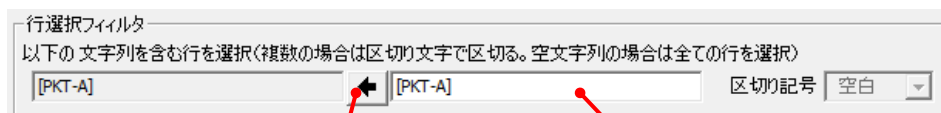


赤字は認識された数値項目を意味します。

赤反転は現在選択されている項目で、 $\leftarrow$  $\rightarrow$ で移動できます。

### 2) 行選択フィルタ

何種類かあるテキスト行から、目的のテキスト行を選択する場合は、「行選択フィルタ」を設定します。



このボタンを押すとフィルタ文字列が設定される

ここにフィルタ文字列を入力

上記の例では、“[PKT-A]”という文字列が含まれるテキスト行だけが処理対象となります。

フィルタ文字列を複数設定する場合は、「区切り記号」で区切って複数の文字列を入力します(ex. “[PKT-A] [TYP1]”)

この場合は、指定した複数の文字列が全て含まれるテキスト行が処理対象となります。

行選択フィルタを空文字列とした場合は、全てのテキスト行が処理対象となります。

### 3) 抽出する数値の位置を指定

抽出する数値群の直前に位置する文字列を指定することにより、当該文字列以降から数値を認識することができます。

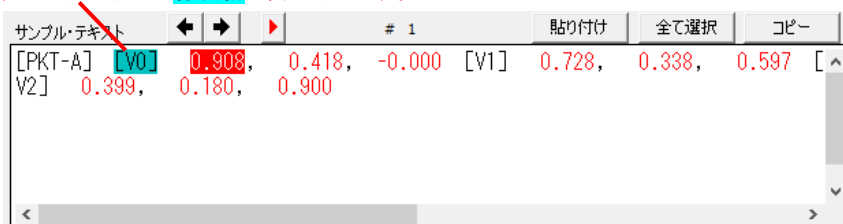
以下の例では、文字列「[V1]」以降から数値を認識するように設定しています。



このボタンを押すと文字列が設定される

ここに数値群の直前に位置する文字列を入力

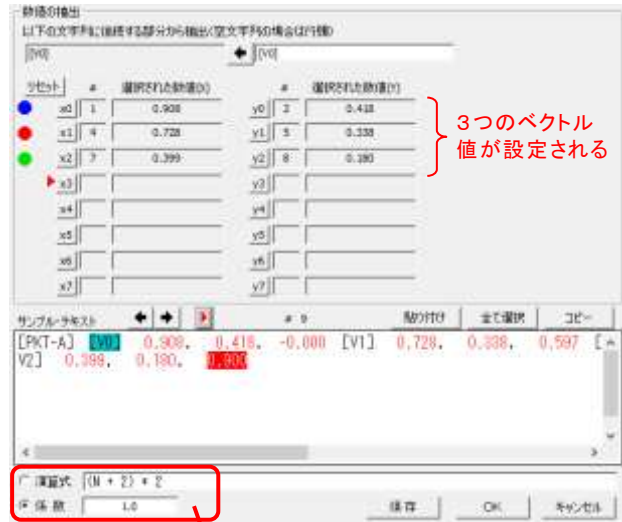
設定した文字列は、青反転で表示されます。



4) 抽出する数値の設定

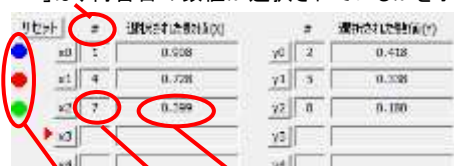
対象とする数値は、 $\leftarrow \rightarrow$  ボタンで選択し、 $x0 \sim y7$  ボタンで当該ボタンの位置へ設定します。  
 または、 $\triangleright$  ボタンを押すと、 $\triangleright$  で示される位置へ設定し、設定位置と数値の選択が1つ進みます。

例えば、下左の状態  $\triangleright \triangleright \rightarrow \triangleright \triangleright \rightarrow \triangleright \triangleright$  の順にボタンを押すと下右図のように設定されます。



「演算式」は、抽出した数値をNとして、グラフ表示の値を算出します  
 「係数」はグラフ表示時、抽出した全ての数値に乗算する値です

「#」は、何番目の数値が選択されているかを示します。



サンプルテキスト上で選択されている7番目の数値を表示します

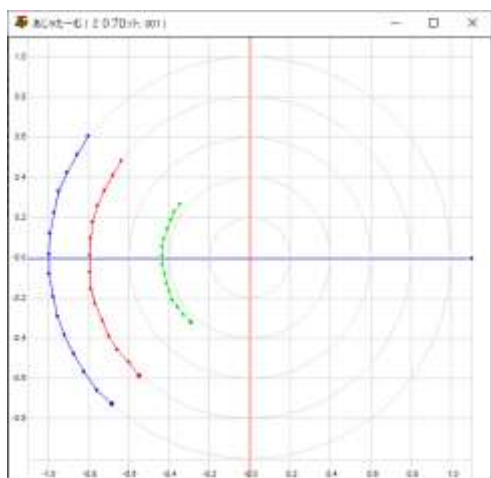
認識されている数値 (赤表示の数値) の7番目の数値が選択されていることを示します。

有効な数値が設定されている場合のインジケータです。この円の色は、グラフィック上での描画色となります。

「リセット」ボタンを押すと、全ての数値選択設定がクリアされます。(上記、左図のようになります)

設定が完了したら「OK」ボタンで上書きするか、「保存」ボタンで名前を付けて保存を行ってください。

上記の設定例では、受信したテキスト行から3つのベクトル値が抽出されて、以下のような2Dプロット・グラフが表示されます。



## 5.1.1. 受信パケットデータから数値を抽出する場合

既に数値抽出条件が設定されている場合は、「…」ボタンから、コンボボックスで設定されている条件を選択し「OK」ボタンを押してください。



コンボボックスから目的の数値抽出条件を選択します



新たに数値抽出条件を設定する場合は、以下のように設定します。

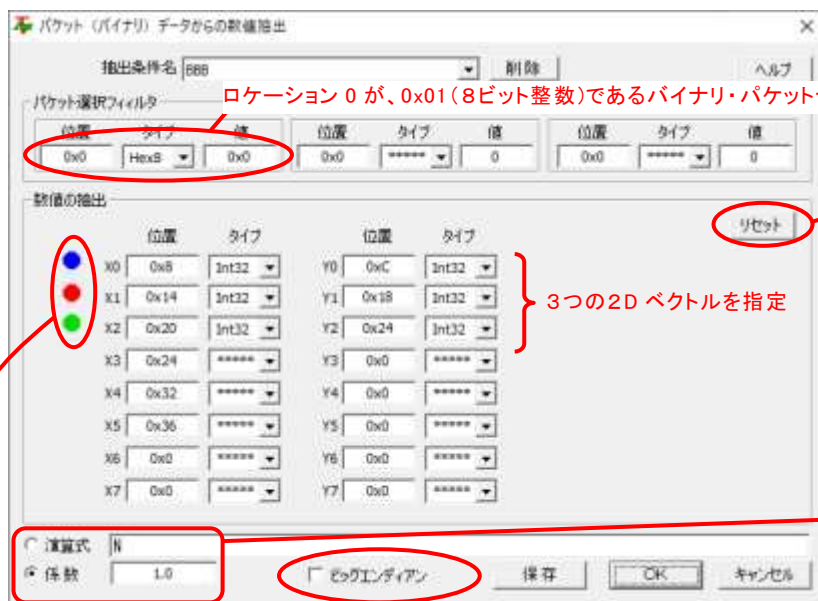
バイナリ・パケットデータの「パケット選択フィルタ」と、抽出する数値の位置とタイプを指定します。

例えば、以下のようなバイナリデータの場合、下図に示すように設定します。

**バイナリデータ**

Loc=0	1	2	3	4	5	6	7	8	12 (0xC)	16 (0x10)			
0x01	-	-	-	-	-	-	-	x0 (INT-32)	y0 (INT-32)	-			
20 (0x14)	24 (0x18)			28 (0x1C)			32 (0x20)		36 (0x24)		40 (0x28)		
x1 (INT-32)				y1 (INT-32)				-	x2 (INT-32)		y2 (INT-32)		-

**ダイアログの設定**



ロケーション 0 が、0x01 (8ビット整数)であるバイナリ・パケットデータを対象とします

「リセット」ボタンを押すと、全ての数値選択設定がクリアされます。

3つの2Dベクトルを指定

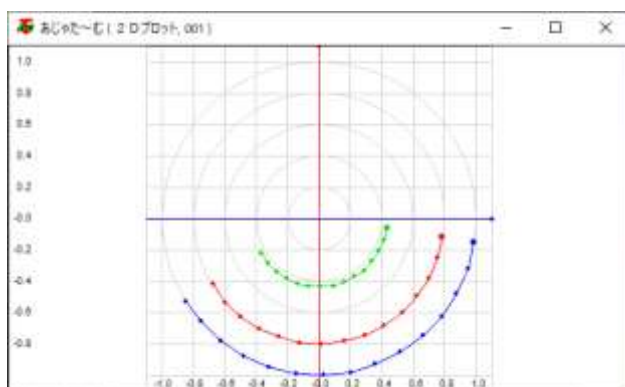
「演算式」は、抽出した数値をNとして、グラフ表示の値を算出します  
「係数」はグラフ表示時、抽出した全ての数値に乗算する値です

数値がビッグエンディアン形式である場合にチェックします。

有効な数値が設定されている場合のインジケータです。この円の色は、グラフィック上での描画色となります。

設定が完了したら「OK」ボタンで上書きするか、「保存」ボタンで名前を付けて保存を行ってください。

上記の設定例では、受信したバイナリ・パケット・データ行から3つベクトル値が抽出されて、以下のような2Dプロット・グラフが表示されます。



## 5.2. ポップアップメニュー

グラフ上で右クリックすると、以下のポップアップメニューが表示されます。



各メニューの内容は、以下のとおりです。

#	メニュー	内容
1	コピー	グラフ表示内容(ビットマップ)をクリップボードへコピーします
2	レンジ設定	グラフのレンジ設定ダイアログを表示します
3	レンジ自動調整	データから最大値/最小値を検索し、レンジを自動的に設定します。
4	各軸のレンジを同一サイズにする	中心位置は変更せずに、各軸のレンジ幅を同一にします。 (最大のレンジ幅に合わせる)
5	アスペクト比をウインドサイズに合わせる アスペクト比を1固定にする	図形描画におけるアスペクト比を、1固定にするか、ウインドサイズに合わせるかの選択。
6	フィルタ表示/非表示	ウインド左上のフィルタ(チェックボックス)を表示/非表示します
7	データクリアー	全てのプロットデータと描画データを破棄します
8	描画時間情報表示	グラフィックの描画に要する時間を計測し表示します

## 5.3. レンジ設定

ポップアップメニューで「レンジ設定」を選択すると、右図のダイアログボックスが表示されます。

ここで、各軸の中心位置と半径/範囲を設定し、OKボタンを押すと、グラフレンジが設定されます。

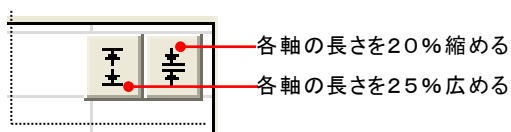
「Cancel」ボタンを押すと、設定を中止します。



## 5.4. ワンタッチでレンジ設定

マウスカーソルを2Dプロットグラフ右上隅に置くと、2つのボタンが表示されます。

これらのボタンで、各軸の長さ(直径)を25%広めたり、20%縮めたりすることができます。



各軸の長さを20%縮める

各軸の長さを25%広める



### 5.5. レンジ自動調整

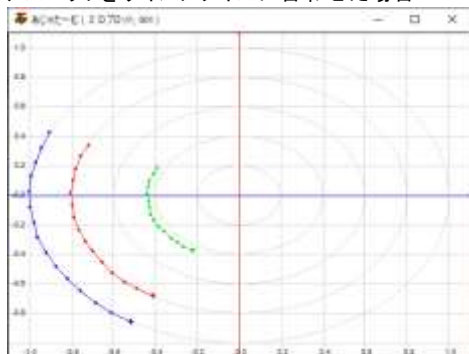
ポップアップメニューで「レンジ自動調整」を選択すると、(フィルタで非表示となっている項目を除く)全てのデータから、最大値と最小値を算出し、(中心位置は変わらないように)±5%のマージンを持ってレンジ設定を行います。

つまり、描画されている全データが視界に入るようにレンジを設定します。

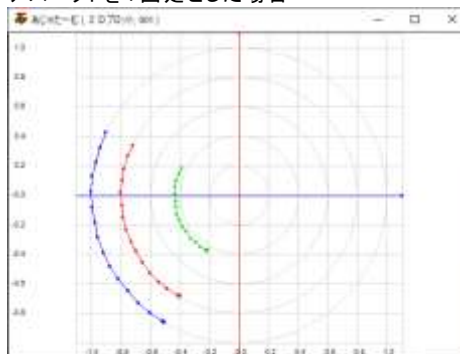
### 5.6. アスペクト比の設定

アスペクト比を可変にする(ウインドサイズに合わせる)か、アスペクト比を1固定にするかを設定できます。

アスペクトをウインドサイズに合わせた場合



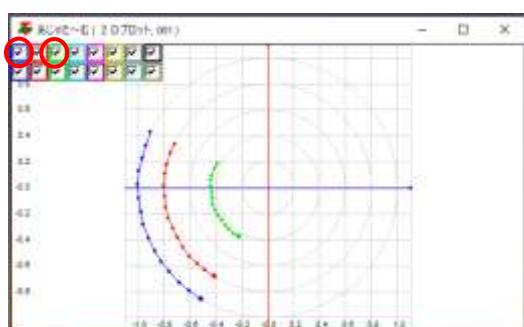
アスペクトを1固定とした場合



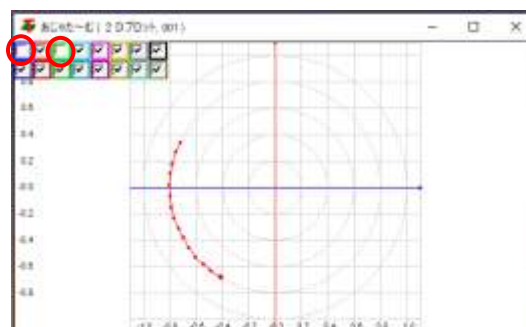
### 5.7. フィルタ機能

カーソルを、2Dプロットグラフの左上隅に置くとチェックボックスが表示されます。

ウインド左上のチェックボックスは、表示フィルタであり、チェックを外すと当該表示色のデータは非表示となります。



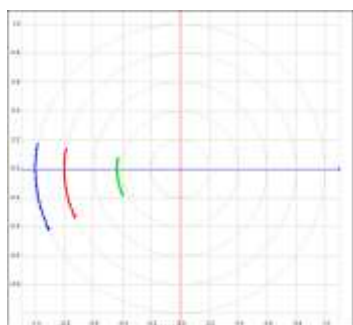
青緑を非表



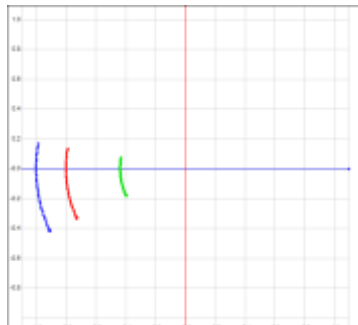
### 5.8. スケールの表示

ダイアログ(「表示」グループ)により、「方眼スケール」「同心(楕)円スケール」を選択できます。また、スケール値やX軸、Y軸の表示／非表示も選択可能です。

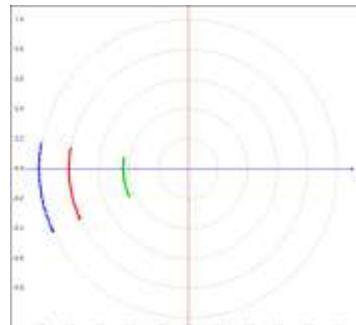
同心円+方眼スケール



方眼スケールのみ

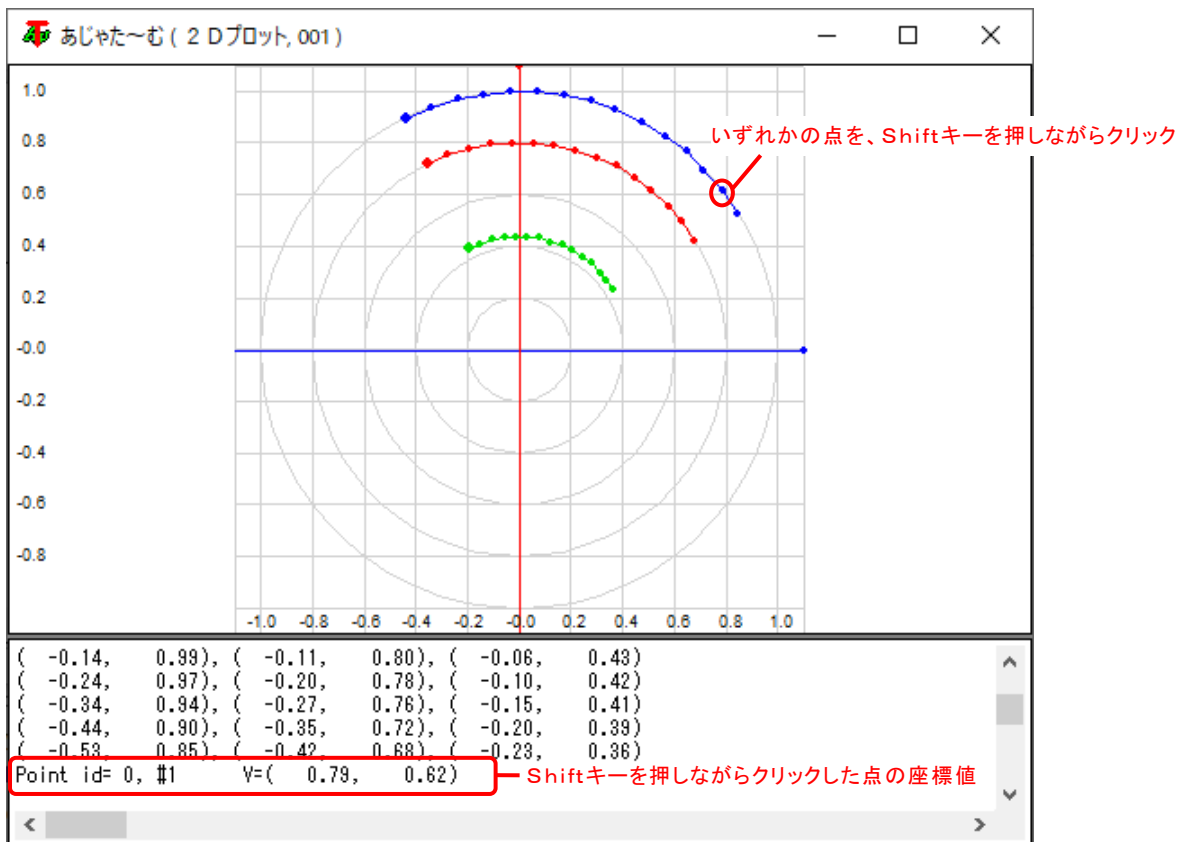


同心円スケールのみ



## 5.9. グラフィック上のプロット点の座標値表示

グラフィック上で、プロット点を、Shiftキーを押しながらクリックすることにより、当該プロット点の座標値を表示することができます。プロット点の座標は、ログウインドに以下のように表示されます。

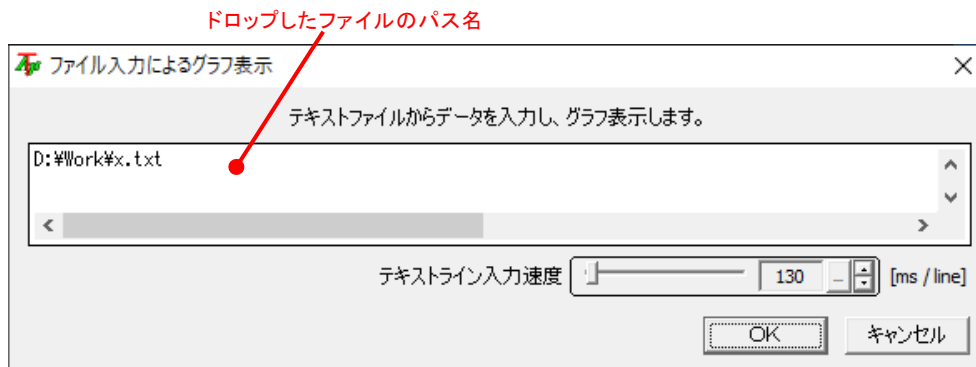


「id」はデータの識別(青=0, 赤=1...)を、「#nnn」はデータの位置(最古のプロット点=0)を、「V=(...)」は座標値を示します。クリックした位置に複数の点が重なっている場合は、複数の点の座標値が表示されます。

## 5.10. テキストファイルを入力し2Dプロット・グラフ表示

2Dプロット・グラフウインドにテキストファイルをドロップすることにより、テキストファイルからデータを抽出して2Dプロット・グラフ表示ができます。

2Dプロット・グラフウインドにテキストファイルをドロップすると、以下のダイアログが表示されます。



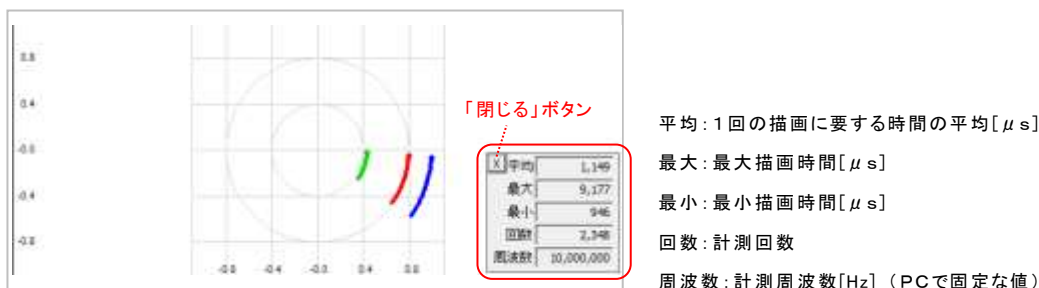
「テキストライン入力速度」は、テキストふぁいるから1行を読み出す速度です。

ここで「OK」ボタンを押すと、テキストファイルからデータを抽出し、2Dプロット・グラフを表示します。

「キャンセル」ボタンを押すとテキストファイルの読み出しを中止します。

### 1.1.1. 描画時間情報表示

2Dプロットグラフを右クリックし、ポップアップメニューから「描画時間情報表示」を選択すると、2Dプロット・イメージの描画時間を計測し、右下に以下のように表示します。



ウインドのサイズを変更した場合は、計測をやり直します。

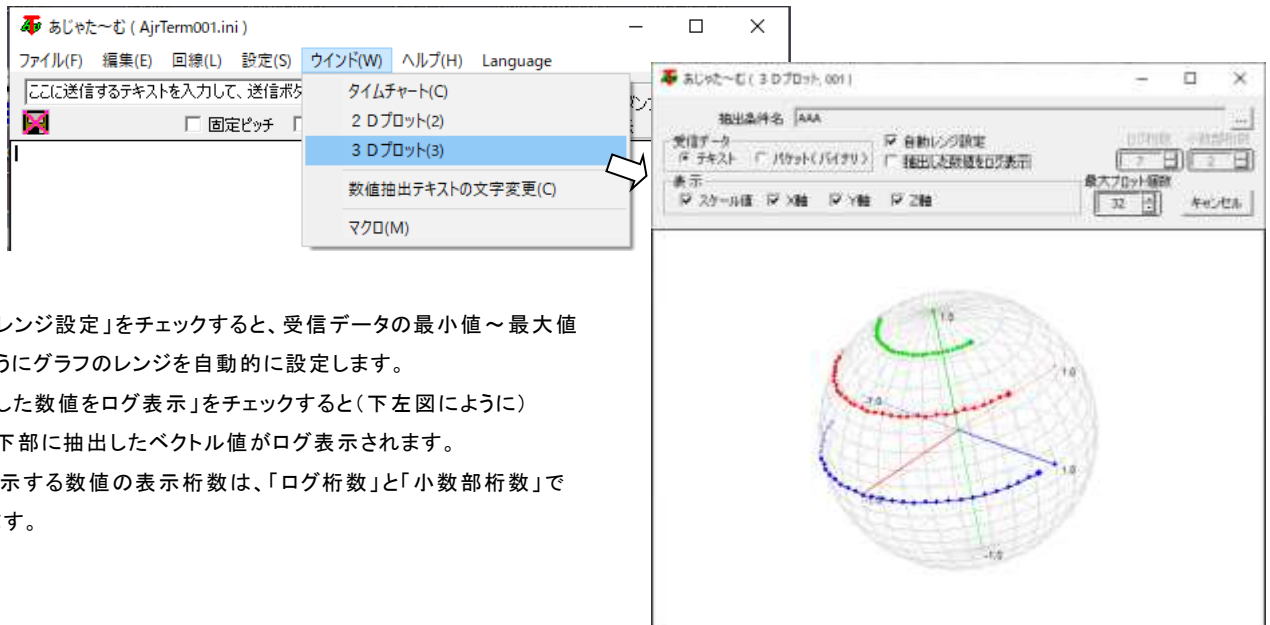
短い周期でデータを投与すると、処理が重くなり、2Dプロット・イメージをスムーズに表示できなくなります。

処理時間は、描画時間だけではありませんが、少なくとも、データを周期的に投与する場合は、平均値よりも長い周期で投与する必要があります。

## 6. 3Dプロットグラフの表示

受信したテキストやパケットから3Dベクトル値を抽出し、3Dプロット・グラフを表示することができます。



3Dプロット・グラフを表示するには、「ウインド」メニューから「3Dプロット」を選択します。

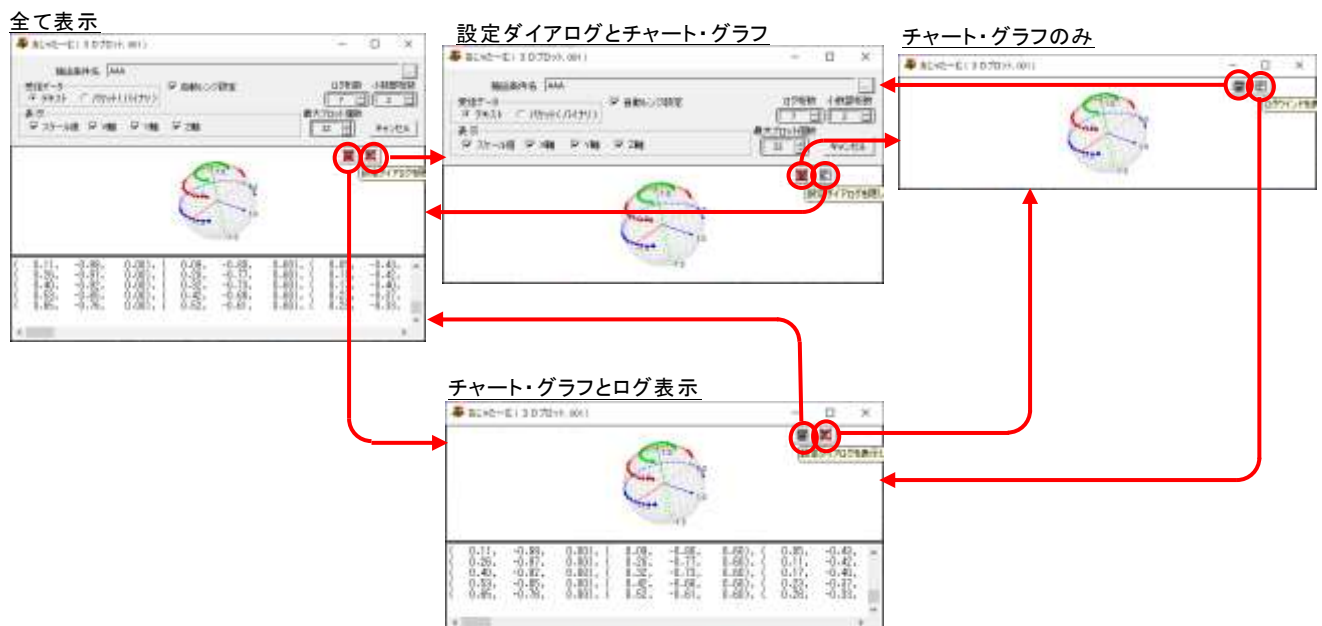


「自動レンジ設定」をチェックすると、受信データの最小値～最大値となるようにグラフのレンジを自動的に設定します。

「抽出した数値をログ表示」をチェックすると(下左図のように)ウインド下部に抽出したベクトル値がログ表示されます。

ログ表示する数値の表示桁数は、「ログ桁数」と「小数部桁数」で設定します。

カーソルをチャートウインドの右上部分に置くとボタン(   )が表示され、ダイアログやログウインドを表示／非表示できます。



## 6.1. 受信データの選択と数値認識方法の設定

数値を抽出する受信データは、テキストデータとパケットデータのいずれかを選択できます。

### テキストデータを選択



### パケットデータを選択

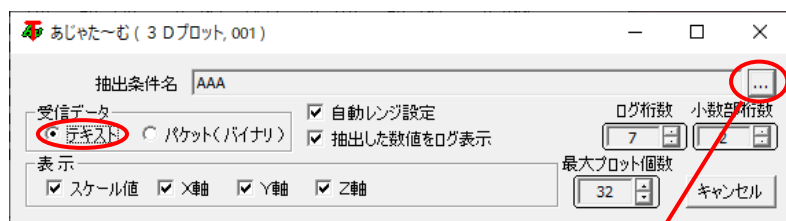


「テキスト」を選択した場合は、テキストデータ行中の、何番目の数値語句を抽出するかを設定します。

「パケット(バイナリ)」を選択した場合は、パケットデータ中のバイト位置と、数値のタイプで指定します。

### 6.1.1. 受信テキストデータから数値を抽出する場合

既に数値抽出条件が設定されている場合は、「…」ボタンから、コンボボックスで設定されている条件を選択し「OK」ボタンを押してください。



コンボボックスから目的の数値抽出条件を選択します



新たに数値抽出条件を設定する場合は、以下の手順で行います。

### 1) サンプル・テキスト行の設定

まず最初に、数値を取り出すサンプル・テキスト行を設定します。

何らかの方法でサンプルテキスト行をクリップボードへコピーしておきます

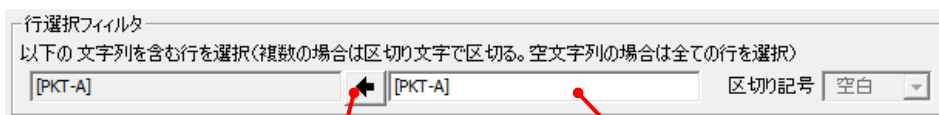


赤字は認識された数値項目を意味します。

赤反転は現在選択されている項目で、←→ボタンで移動できます。

### 2) 行選択フィルタ

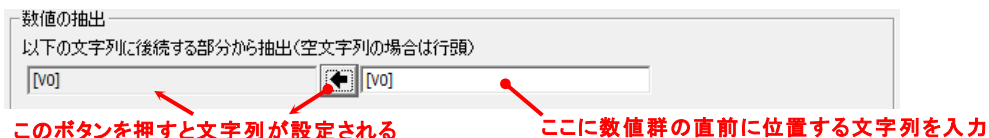
何種類かあるテキスト行から、目的のテキスト行を選択する場合は、「行選択フィルタ」を設定します。



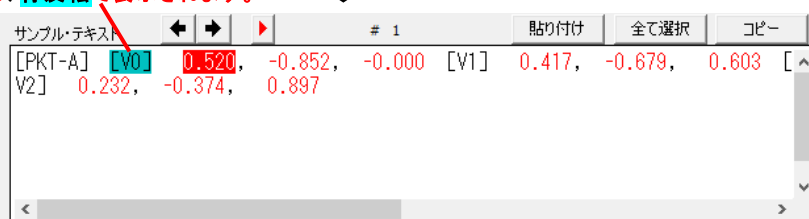
上記の例では、“[PKT-A]”という文字列が含まれるテキスト行だけが処理対象となります。  
 フィルタ文字列を複数設定する場合は、「区切り記号」で区切って複数の文字列を入力します(ex. “[PKT-A] [TYP1]”)  
 この場合は、指定した複数の文字列が全て含まれるテキスト行が処理対象となります。  
 行選択フィルタを空文字列とした場合は、全てのテキスト行が処理対象となります。

### 3) 抽出する数値の位置を指定

抽出する数値群の直前に位置する文字列を指定することにより、当該文字列以降から数値を認識することができます。  
 以下の例では、文字列「[V1]」以降から数値を認識するように設定しています。



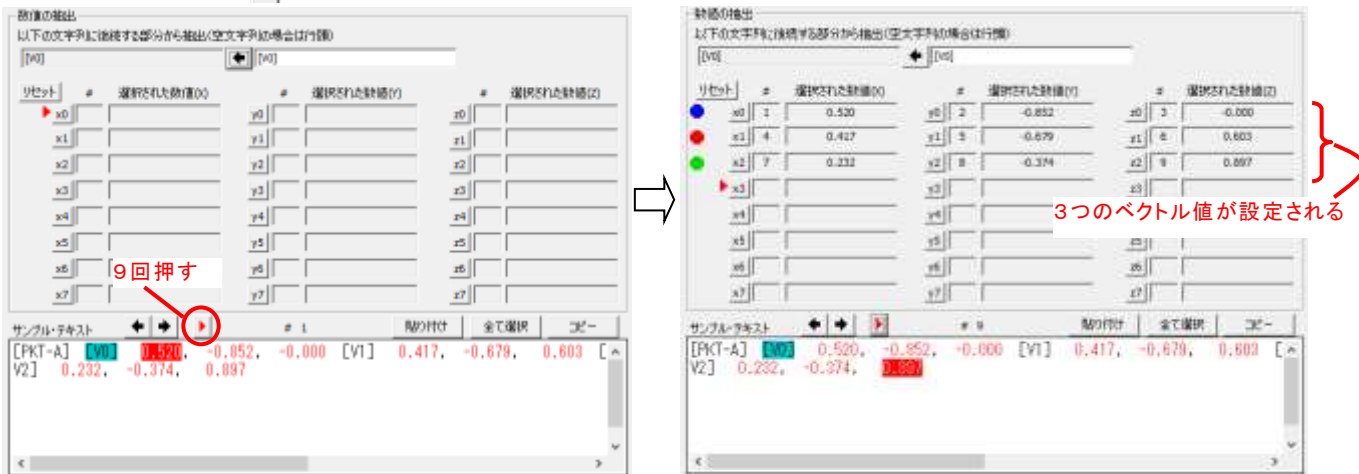
設定した文字列は、青反転で表示されます。



#### 4) 抽出する数値の設定

対象とする数値は、**←→** ボタンで選択し、**x0** ~ **z7** ボタンで当該ボタンの位置へ設定します。  
 または、**▶** ボタンを押すと、**▶** で示される位置へ設定し、設定位置と数値の選択が1つ進みます。

例えば、下左図の状態では **▶** ボタンを9回押すと下右図のように設定されます。



「#」は、何番目の数値が選択されているかを示します。



サンプルテキスト上で選択されている7番目の数値を表示します

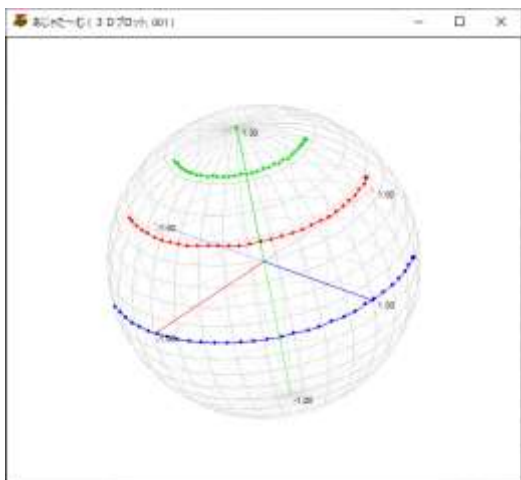
認識されている数値(赤表示の数値)の7番目の数値が選択されていることを示します。

有効な数値が設定されている場合のインジケータです。この円の色は、グラフィック上での描画色となります。

「リセット」ボタンを押すと、全ての数値選択設定がクリアされます。(上記、左図のようになります)

設定が完了したら「OK」ボタンで上書きするか、「保存」ボタンで名前を付けて保存を行ってください。

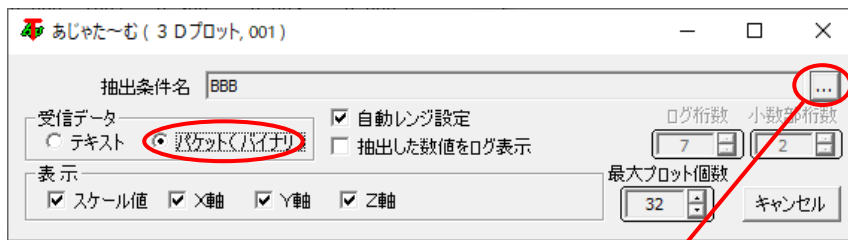
上記の設定例では、受信したテキスト行から3つのベクトル値が抽出されて、以下のような3Dプロット・グラフが表示されます。



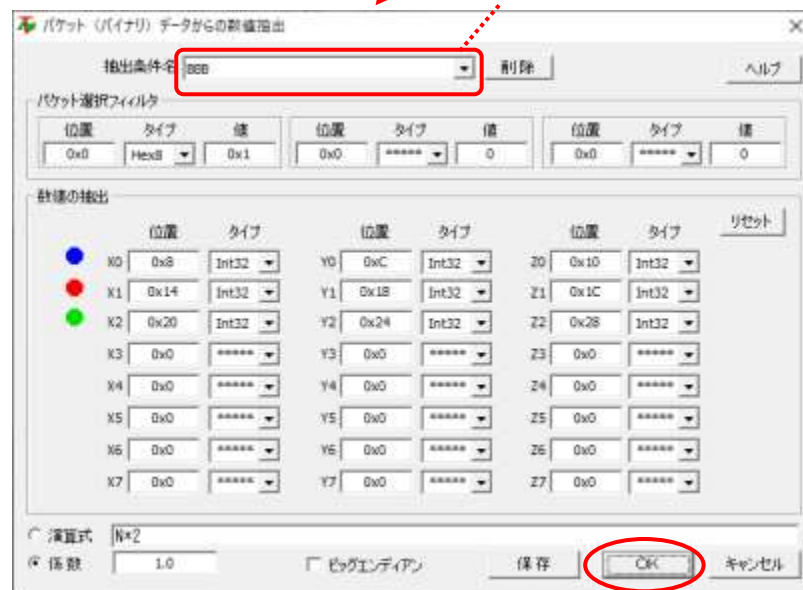
**視点の設定**  
 視点の設定は、ポップアップメニューの「XY座標面」「XZ座標面」「YZ座標面」「3D座標面」で特定の視点を設定するか、あるいは、グラフィックウィンドウ上を、マウスの左ボタンでドラッグすることにより、任意の視点を設定することができます。  
 マウスで横方向にドラッグすると、表示物体がX軸回りに回転します。  
 マウスで縦方向にドラッグすると、表示物体がY軸回りに回転します。  
 CTRL キーを押しながら横方向にドラッグすると、表示物体がZ軸回りに回転します。

## 6.1.2. 受信パケットデータから数値を抽出する場合

既に数値抽出条件が設定されている場合は、「…」ボタンから、コンボボックスで設定されている条件を選択し「OK」ボタンを押してください。



コンボボックスから目的の数値抽出条件を選択します





新たに周知抽出条件を設定する場合は、以下のように設定します。

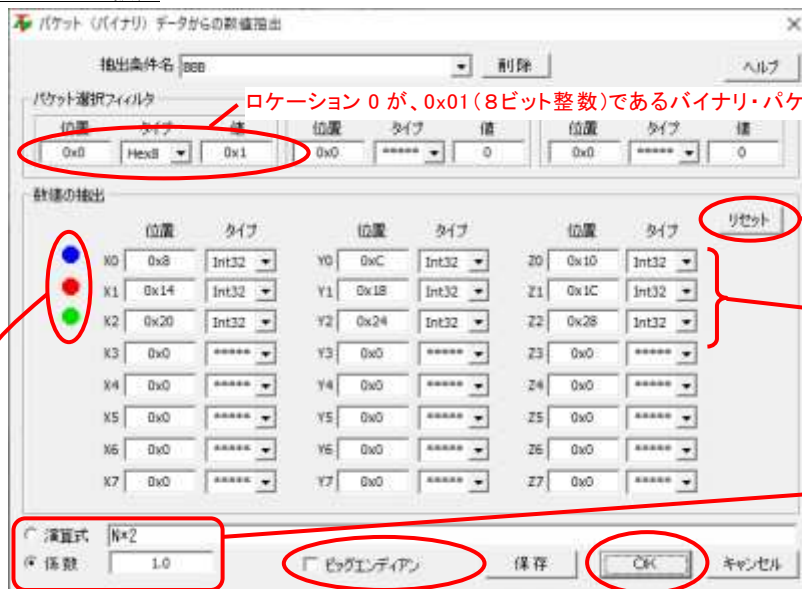
バイナリ・パケットデータの「パケット選択フィルタ」と、抽出する数値の位置とタイプを指定します。

例えば、以下のようなバイナリデータの場合、下図に示すように設定します。

**バイナリデータ**

Loc=0								12 (0xC)				16 (0x10)											
0x01	—	—	—	—	—	—	—	x0 (INT-32)				y0 (INT-32)				z0 (INT-32)							
20 (0x14)				24 (0x18)				28 (0x1C)				32 (0x20)				36 (0x24)				40 (0x28)			
x1 (INT-32)				y1 (INT-32)				z1 (INT-32)				x2 (INT-32)				y2 (INT-32)				z2 (INT-32)			

**ダイアログの設定**



ロケーション 0 が、0x01 (8ビット整数)であるバイナリ・パケットデータを対象とします

「リセット」ボタンを押すと、全ての数値選択設定がクリアされます。

3つの 3D ベクトルを指定

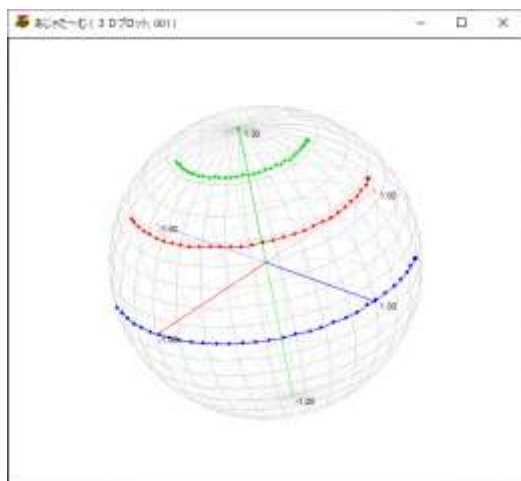
「演算式」は、抽出した数値をNとして、グラフ表示の値を算出します  
「係数」はグラフ表示時、抽出した全ての数値に乗算する値です

数値がビッグエンディアン形式である場合にチェックします。

有効な数値が設定されている場合のインジケータです。この円の色は、グラフィック上での描画色となります。

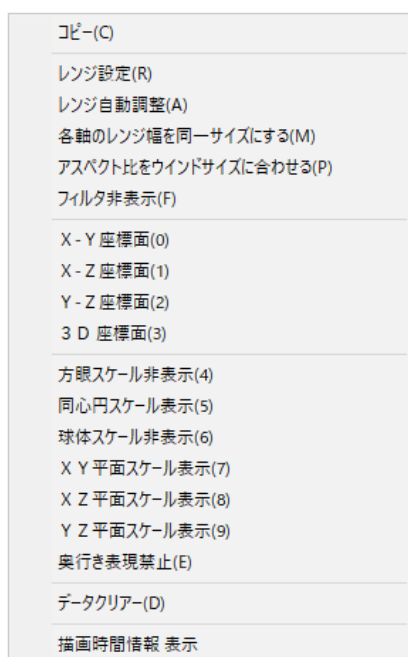
設定が完了したら「OK」ボタンで上書きするか、「保存」ボタンで名前を付けて保存を行ってください。

上記の設定例では、受信したバイナリ・パケット・データ行から3つベクトル値が抽出されて、以下のような 3D プロット・グラフが表示されます。



## 6.2. ポップアップメニュー

グラフ上で右クリックすると、以下のポップアップメニューが表示されます。



各メニューの内容は、以下のとおりです。

#	メニュー	内容
1	コピー	グラフ表示内容(ビットマップ)をクリップボードへコピーします
2	レンジ設定	グラフのレンジ設定ダイアログを表示します
3	レンジ自動調整	データから最大値/最小値を検索し、レンジを自動的に設定します。
4	各軸のレンジを同一サイズにする	中心位置は変更せずに、各軸のレンジ幅を同一にします。 (最大のレンジ幅に合わせる)
5	アスペクト比をウインドサイズに合わせる アスペクト比を1固定にする	図形描画におけるアスペクト比を、1固定にするか、ウインドサイズに合わせるかの選択。
6	フィルタ表示/非表示	ウインド左上のフィルタ(チェックボックス)を表示/非表示します
7	X-Y座標面	X-Y平面を表示するように視点を設定します
8	X-Z座標面	X-Z平面を表示するように視点を設定します
9	Y-Z座標面	Y-Z平面を表示するように視点を設定します
10	3D座標面	3Dイメージを表示するように視点を設定します
11	方眼スケール表示	#13~15で表示設定された平面に、方眼スケールの表示/非表示
12	同心円スケール表示	#13~15で表示設定された平面に、同心円スケールの表示/非表示
13	球体スケール表示	球体スケールの表示/非表示
14	XY平面スケール表示	XY平面に方眼/同心円スケールの表示を許可/禁止
15	XZ平面スケール表示	XZ平面に方眼/同心円スケールの表示を許可/禁止
16	YZ平面スケール表示	YZ平面に方眼/同心円スケールの表示を許可/禁止
17	奥行き表現禁止	奥行き表現(原点より前側と向こう側で表示色を変える)の許可/禁止
18	データクリアー	全てのプロットデータと描画データを破棄します
19	描画時間情報表示	グラフィックの描画に要する時間を計測し表示します

### 6.3. レンジ設定

ポップアップメニューで「レンジ設定」を選択すると、右図のダイアログボックスが表示されます。

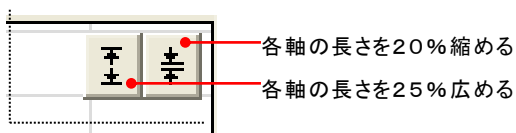
ここで、各軸の中心位置と半径／範囲を設定し、OKボタンを押すと、グラフレンジが設定されます。

「Cancel」ボタンを押すと、設定を中止します。

### 6.4. ワンタッチでレンジ設定

マウスカーソルを3Dプロットグラフの右上隅に置くと、2つのボタンが表示されます。

これらのボタンで、各軸の長さ(直径)を30%広めたり、縮めたりすることができます。



### 6.5. レンジ自動調整

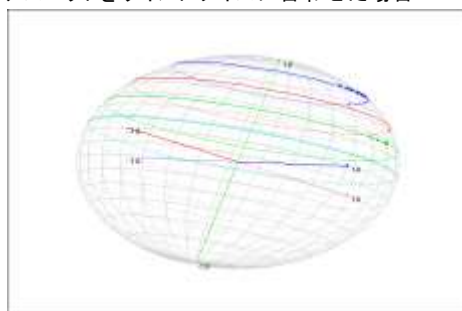
ポップアップメニューで「レンジ自動調整」を選択すると、(フィルタで非表示となっている項目を除く)全てのデータから、最大値と最小値を算出し、(中心位置は変わらないように)±5%のマーヅンを持ってレンジ設定を行います。

つまり、描画されている全データが視界に入るようにレンジを設定します。

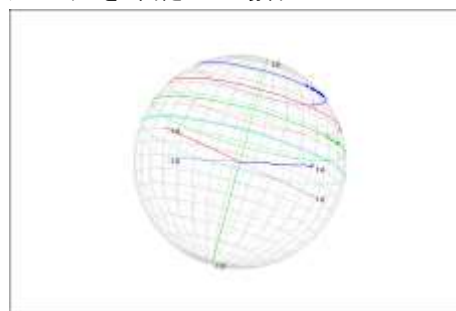
### 6.6. アスペクト比の設定

アスペクト比を可変にする(ウインドサイズに合わせる)か、アスペクト比を1固定にするかを設定できます。

アスペクトをウインドサイズに合わせた場合



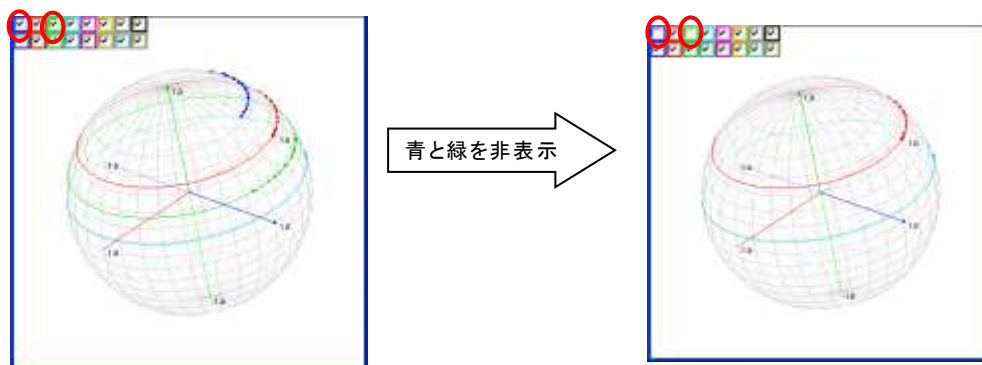
アスペクトを1固定とした場合



### 6.7. フィルタ機能

カーソルを、3Dプロットグラフの左上隅に置くとチェックボックスが表示されます。

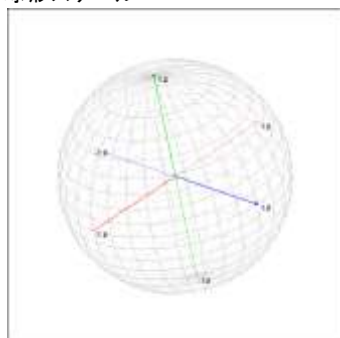
このチェックボックスは、表示フィルタであり、チェックを外すと当該表示色のデータは非表示となります。



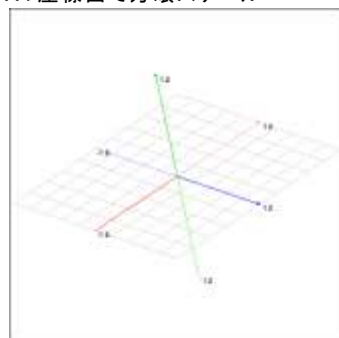
### 6.8. スケールの表示

ポップアップメニュー／設定ダイアログにより、「方眼スケール」「同心(楕)円スケール」「(楕)球形スケール」を選択できます。また、スケール値の表示／非表示も選択可能です。

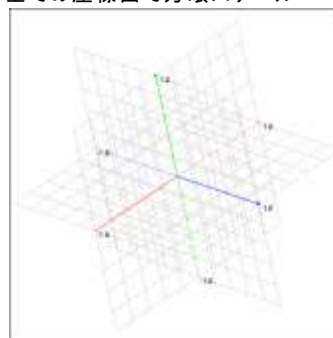
球形スケール



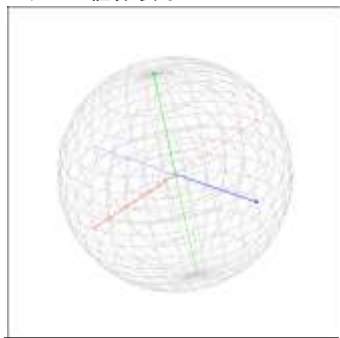
XY座標面で方眼スケール



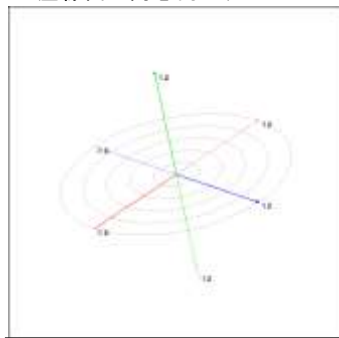
全ての座標面で方眼スケール



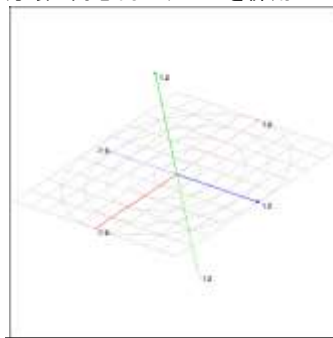
スケール値非表示



XY座標面で同心円スケール

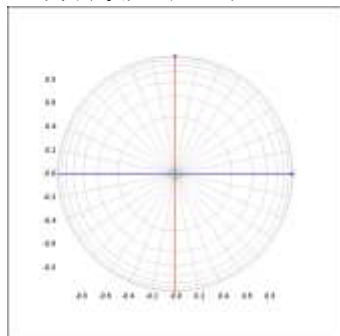


方眼と同心円スケールを併用

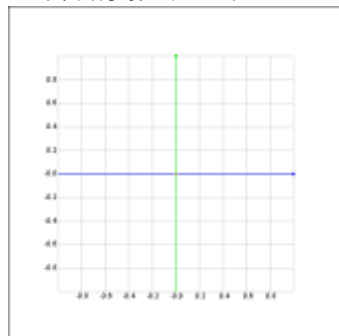


スケール値は、視点がいずれかの座標軸と一致した(つまり、いずれか2軸の平面を見る)場合に限り、中間値も表示されます。その他の視点設定では、各軸の先端値だけを表示します。

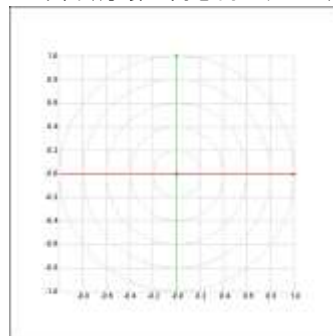
XY平面(球形スケール)



XZ平面(方眼スケール)



YZ平面(方眼+同心円スケール)



### 1.1.2. 描画時間情報表示

3Dプロットグラフを右クリックし、ポップアップメニューから「描画時間情報表示」を選択すると、3Dプロット・イメージの描画時間を計測し、右下に以下のように表示します。



ウインドのサイズを変更した場合は、計測をやり直します。

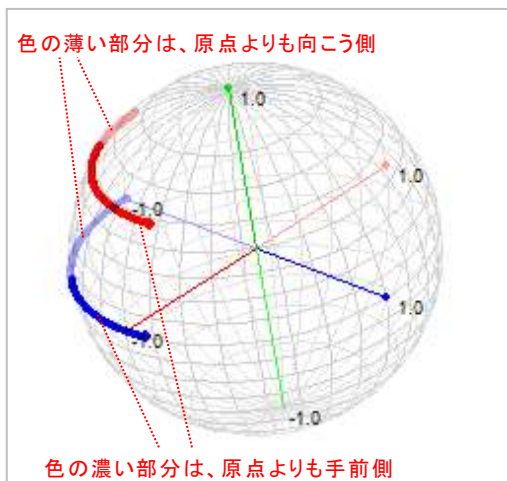
短い周期でデータを投与すると、処理が重くなり、3Dプロット・イメージをスムーズに表示できなくなります。

処理時間は、描画時間だけではありませんが、少なくとも、データを周期的に投与する場合は、平均値よりも長い周期で投与する必要があります。

### 1.1.3. 奥行き表現

原点より向う側のイメージを少し薄く描画することで、原点より手前側か、あるいは、向こう側にあるイメージかを表現します。

1.1.2. Depth representation ちょうど、原点に垂直な「すりガラス」を立てたようなイメージとなります。



ポップアップメニューで「奥行き表現禁止」を選択した場合は、原点より手前側でも、向こう側でも、同じ描画(濃い色)となります。

## 6.9. グラフィック上のプロット点の座標値表示

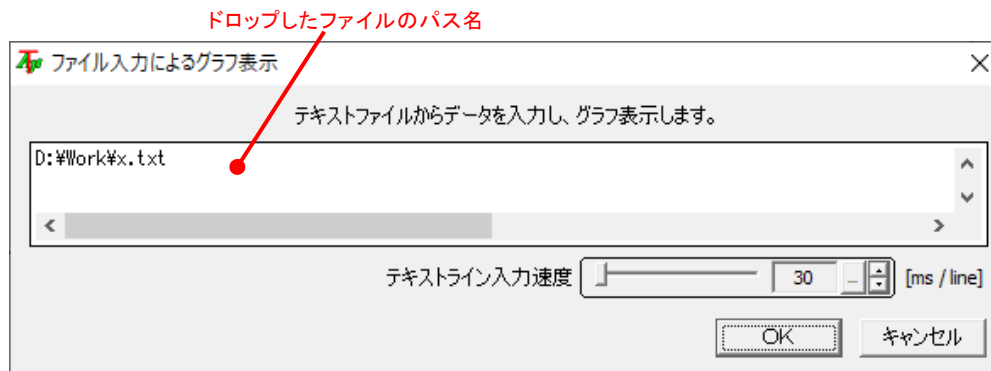
グラフィック上で、プロット点を、Shiftキーを押しながらクリックすることにより、当該プロット点の座標値を表示することができます。プロット点の座標は、ログウインドに以下のように表示されます。

「id」はデータの識別(青=0, 赤=1・・・)を、「#nnn」はデータの位置(最古のプロット点=0)を、「V=(・・・)」は座標値を示します。クリックした位置に複数の点が重なっている場合は、複数の点の座標値が表示されます。

## 6.10. テキストファイルを入力し3Dプロット・グラフ表示

3Dプロット・グラフウインドにテキストファイルをドロップすることにより、テキストファイルからデータを抽出して3Dプロット・グラフ表示ができます。

3Dプロット・グラフウインドにテキストファイルをドロップすると、以下のダイアログが表示されます。



「テキストライン入力速度」は、テキストファイルから1行を読み出す速度です。

ここで「OK」ボタンを押すと、テキストファイルからデータを抽出し、3Dプロット・グラフを表示します。

「キャンセル」ボタンを押すとテキストファイルの読み出しを中止します。

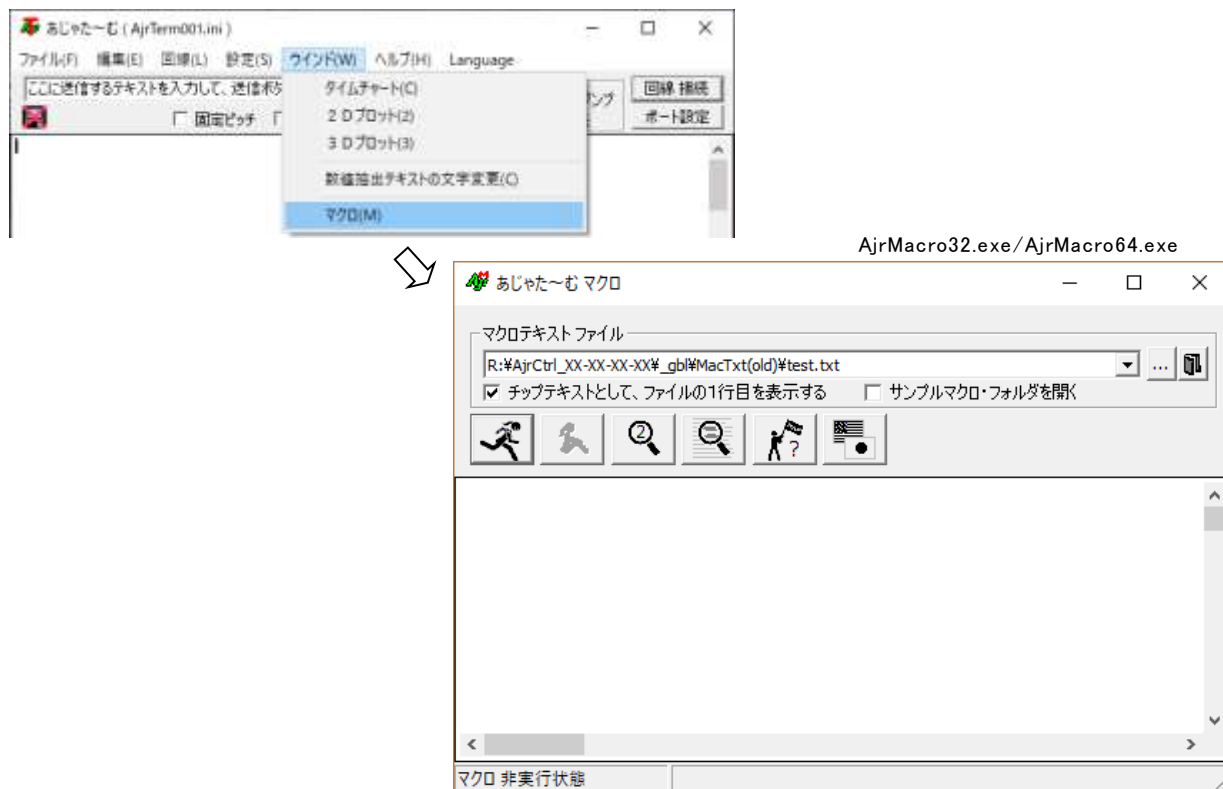
## 7. マクロ機能

マクロ機能により、「あじゃた〜む」が接続している回線を利用して、データの送受信を行うことができます。

また、「あじゃた〜む」とは接続しないで、マクロ単独で回線の送受信を行うこともできます。

(「あじゃた〜む(マクロ機能)」実行のプログラムは、AjrMacro32.exe/AjrMacro64.exe です )

「ウインド」メニューから「マクロ」を選択すると、以下のマクロ・ウインドが表示されます。



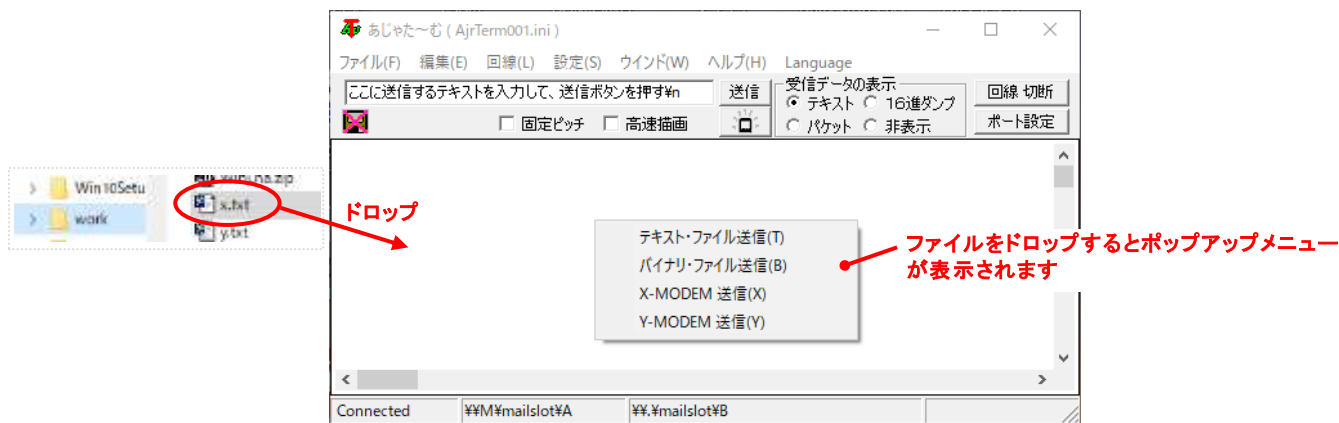
「あじゃた〜む」マクロ機能の詳細な内容については、「[AjmMacroJ.pdf](#)」を参照してください。



## 8. ファイル送受信

「あじやた〜む」では、単純なファイル送信に加え、XMODEMおよびYMODEMプロトコルによるファイル転送機能をサポートしています。ファイル送信は「ファイル」メニューから選択しますが、メインウインドに送信するファイルをドロップし、ポップアップメニューから選択することもできます。

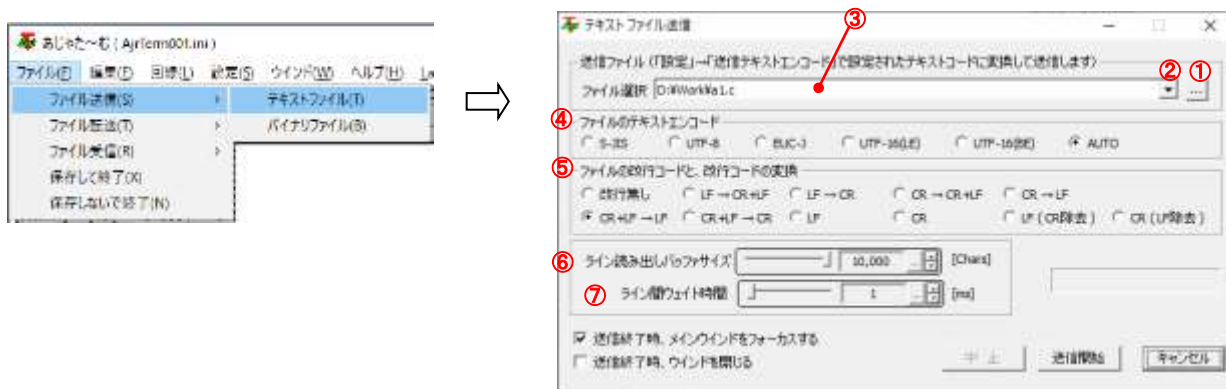
複数のファイルをドロップした場合は、ポップアップメニューは表示されず YMODEM 送信が選択されます。



### 8.1. テキストファイル送信

テキストファイルを、送信する機能です。

「ファイル」メニューから「ファイル送信」→「テキストファイル」を選択（あるいは、ポップアップメニューから「テキスト・ファイル送信」を選択）すると、以下のダイアログが表示されます。



①の「...」ボタンにより送信ファイルを選択します。(あるいは、送信するファイルを③の部分へドロップします)

または、②のコンボボックスから、送信するファイルを選択します。

「送信開始」ボタンを押すと、当該ファイルの送信を開始します。

④は、ファイルのテキストエンコード種別を指定します。「AUTO」を指定した場合はエンコード種別を自動判別します。(UTF-8 や UTF-16 ファイルで、BOM がある場合は、指定を無視し、BOM で指定されたエンコード種別となります)

⑤は、テキストファイルの改行コードと、改行コードの変換を指定します。

⑥は、ファイルから 1 行読み出す際のバッファサイズを文字数で指定します。

1 行のサイズがバッファサイズより大きい場合は、途中までを 1 行とみなし、後続を次行とします。

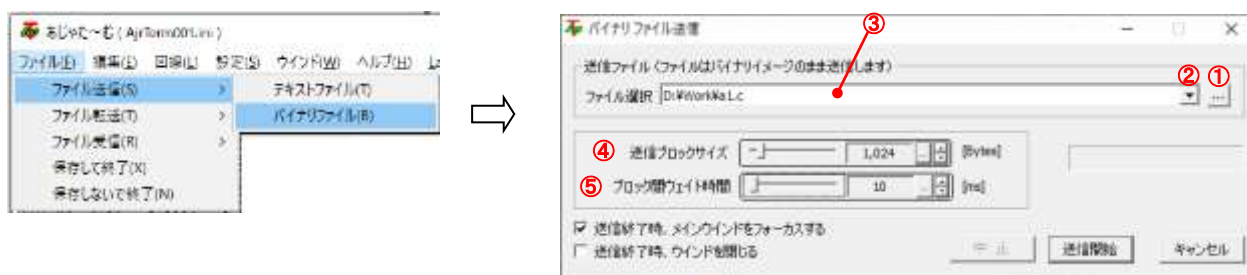
⑦は、行間に挿入するウェイト時間を指定します。

実際のウェイト時間は、システムタイマの精度(15ms 程度)の倍数に切り上げられます)

## 8.2. バイナリファイル送信

ファイルを、バイナリイメージで送信する機能です。

「ファイル」メニューから「ファイル送信」→「バイナリファイル」を選択(あるいは、ポップアップメニューから「バイナリ・ファイル送信」を選択)すると、以下のダイアログが表示されます。



①の「...」ボタンにより送信ファイルを選択します。(あるいは、送信するファイルを③の部分へドロップします)

または、②のコンボボックスから、送信するファイルを選択します。

「送信開始」ボタンを押すと、当該ファイルの送信を開始します。

④は、1回に送信するバイナリデータのバイト数を指定します。

⑤は、送信ブロック間に挿入するウェイト時間を指定します。

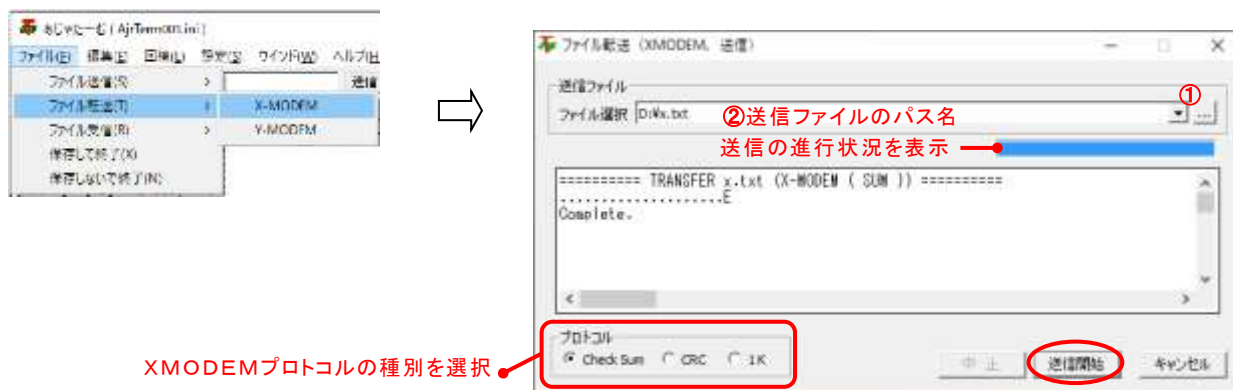
実際の、ウェイト時間はシステムタイマの精度(15ms程度)の倍数に切り上げられます。

### 8.3. ファイル転送(XMODEM)

XMODEMプロトコルによる単一ファイルの送受信ができます。

#### 8.3.1. XMODEMによるファイル送信

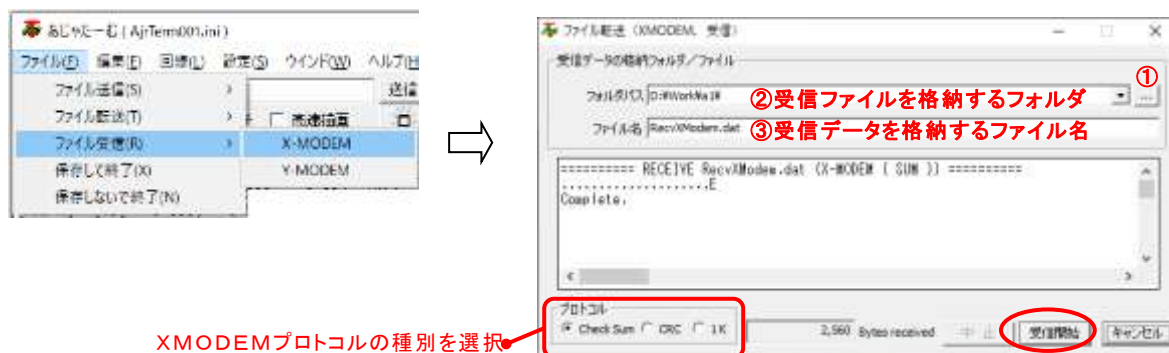
「ファイル」メニューから「ファイル転送」→「XMODEM」を選択(あるいは、ポップアップメニューから「XMODEM 送信」を選択)すると、以下のダイアログが表示されます。



- ①の「...」ボタンにより送信ファイルを選択します。(あるいは、送信するファイルを②の部分へドロップします)  
 または、②のコンボボックスから、送信するファイルを選択します。  
 「送信開始」ボタンを押すと、当該ファイルの送信を開始します。

#### 8.3.2. XMODEMによるファイル受信

「ファイル」メニューから「ファイル受信」→「XMODEM」を選択すると、以下のダイアログが表示されます。



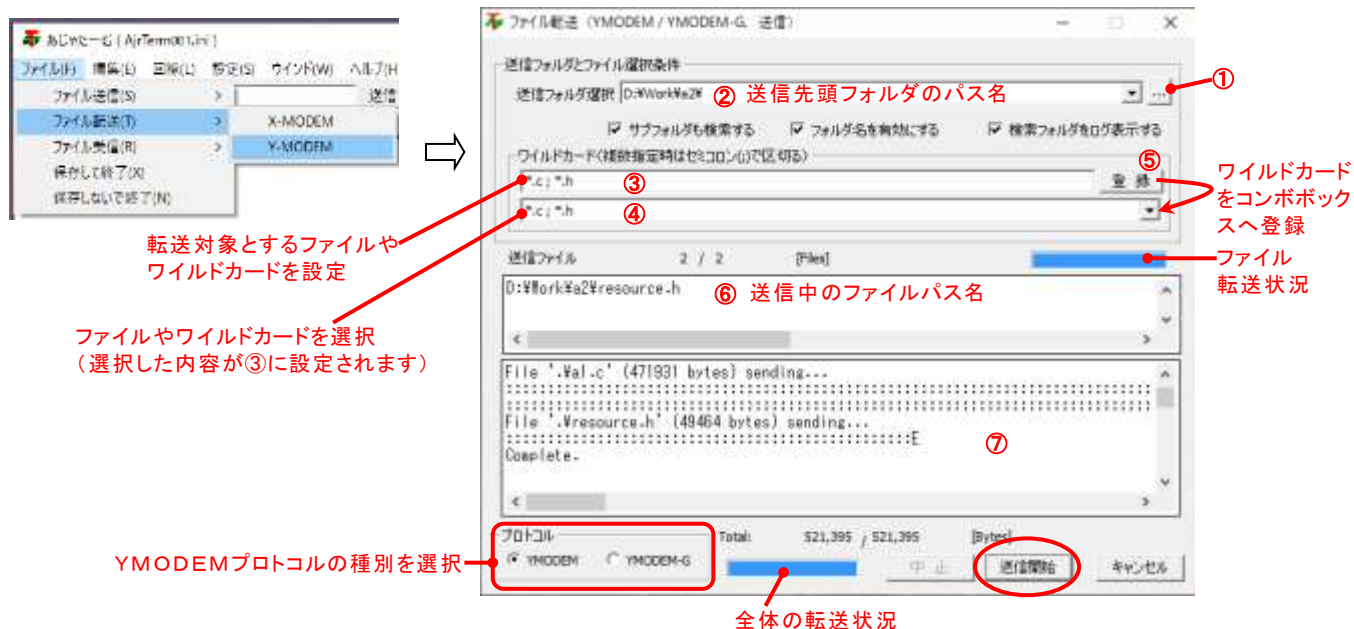
- ①の「...」ボタンにより受信ファイルを格納するフォルダを設定します。(あるいは、送信するファイルを②の部分へドロップする)  
 または、②のコンボボックスから、格納するするフォルダを選択します。  
 ③に受信データを格納するファイル名を設定します。  
 「受信開始」ボタンを押すと、当該ファイルの受信を開始します。

## 8.4. ファイル転送(YMODEM)

YMODEMプロトコルによる単一ファイル、あるいは、複数のファイルを一括して送受信ができます。

### 8.4.1. YMODEMによるファイル送信

「ファイル」メニューから「ファイル転送」→「YMODEM」を選択(あるいは、ポップアップメニューから「YMODEM 送信」を選択)すると、以下のダイアログが表示されます。



①の「...」ボタンにより転送対象とするフォルダを設定します。(あるいは、送信するフォルダを②の部分へドロップする)

または、②のコンボボックスから、送信対象のフォルダを選択します。

次に、③へ転送対象とするファイルのワイルドカード／ファイル名を入力します。(あるいは、④のコンボボックスから選択します)

ワイルドカード／ファイル名を複数指定する場合は、セミicolon (;) で区切って入力します。

入力したワイルドカード／ファイル名の前後の空白は無視されます。(「Temp.txt;\*.png」と「Temp.txt ; \*.png」は同じ)

⑤の「登録」ボタンを押すと、入力したワイルドカード／ファイル名が④のコンボボックスへ登録され、後から選択できるようになります。

②の部分へ単一(あるいは複数)のファイルをドロップすることにより、当該ファイルの送信をセットアップすることができます。

(③へ、ドロップしたファイル(群)が設定されます)

「送信開始」ボタンを押すと、指定されたファイル(群)の送信を開始します。

「サブフォルダも検索する」をチェックした場合、指定したフォルダだけでなく、さらに下のサブフォルダも検索します。

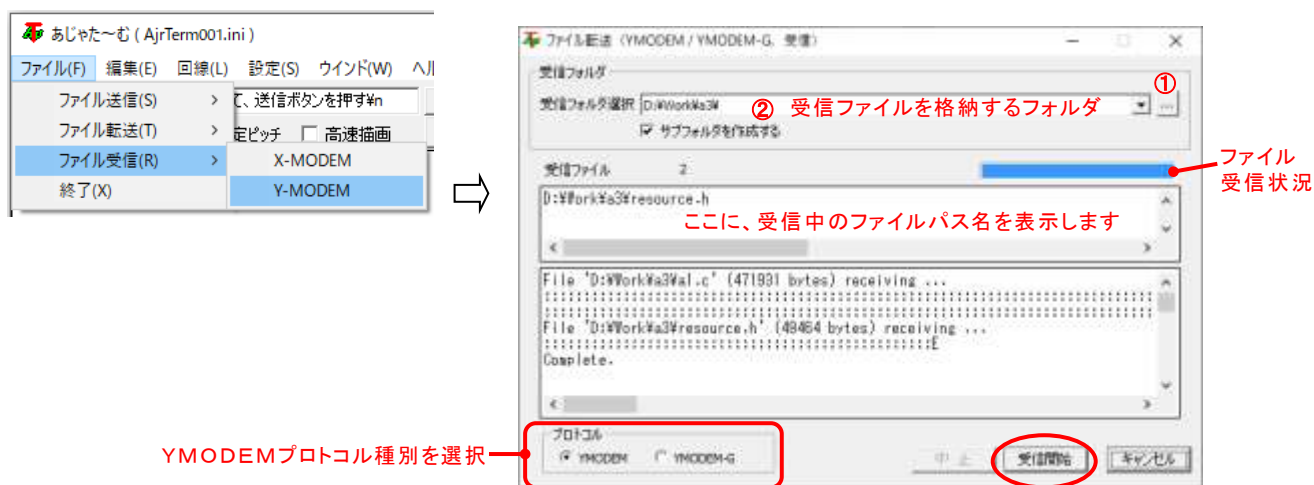
「フォルダ名を有効にする」をチェックした場合、YMODEMプロトコルで送信するファイル情報に、トップフォルダからの相対アドレスを付加します。(ex . .¥Sub1¥Sub2¥sample.txt)

「フォルダ名を有効にする」をチェックしない場合は、YMODEMプロトコルで送信するファイル情報に、トップフォルダからの相対アドレスを付加せずに、ファイル名部分だけとなります。(ex . sample.txt)

「検索フォルダをログ表示する」をチェックした場合、送信ファイルの検索を行ったフォルダを全て⑦のウインドヘログ表示します。

## 8.4.2. YMODEMによるファイル受信

「ファイル」メニューから「ファイル受信」→「YMODEM」を選択すると、以下のダイアログが表示されます。



①の「...」ボタンにより受信したファイルを格納するフォルダを指定します。(あるいは、フォルダを②の部分へドロップする) または、②のコンボボックスから、フォルダを選択します。

「受信開始」ボタンを押すとファイルの受信を開始します。

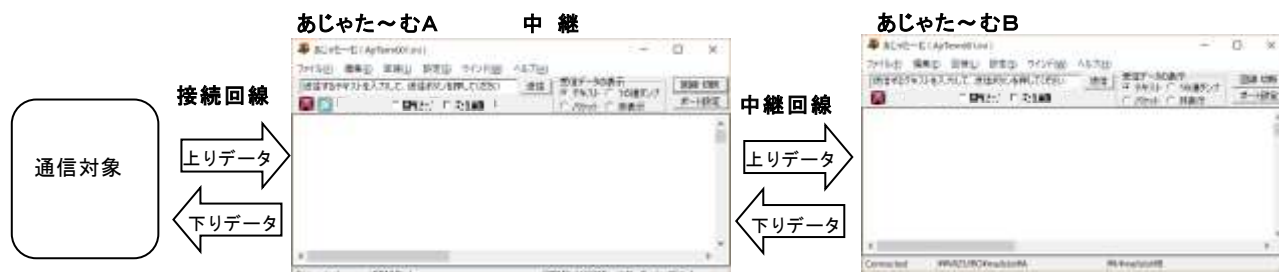
「サブフォルダを作成する」をチェックした場合、YMODEMプロトコルで受信したファイル情報に、フォルダ情報が付加されていれば、トップフォルダからの相対アドレスとみなし、当該フォルダを作成します。

「サブフォルダを作成する」をチェックしない場合は、受信したファイルは全てトップフォルダ内に格納されます。

## 9. 中継機能

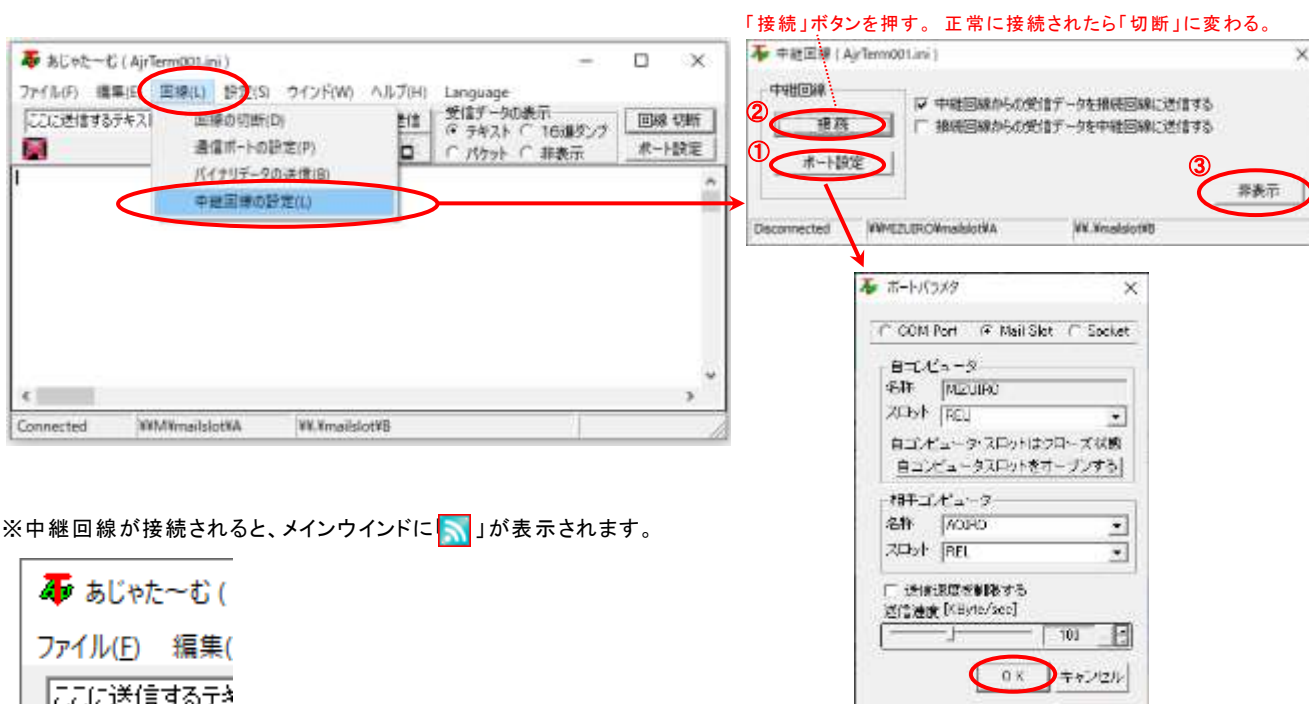
あじやた〜む間で、回線を中継することができます。


つまり、あじやた〜むAが接続回線から受信したデータを、あじやた〜むBに転送したり、あじやた〜むBから受信したデータをあじやた〜むAの接続回線に送信させることができます。

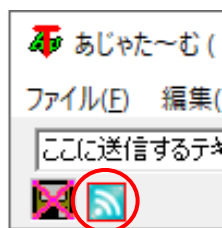


データの中継するには、「回線」メニューから「中継回線の設定」を選択し、以下の手順で設定します。

- 1) 「ポート設定」ボタンにより、自メールスロットと、相手コンピュータ名、スロット名を設定します。
- 2) 「接続」ボタンで、中継先と接続します。(ボタンフェースが「切断」に変わります)
- 3) 「非表示」ボタンを押すと、ウインドを非表示にします。(中継回線の接続状態は維持されます)



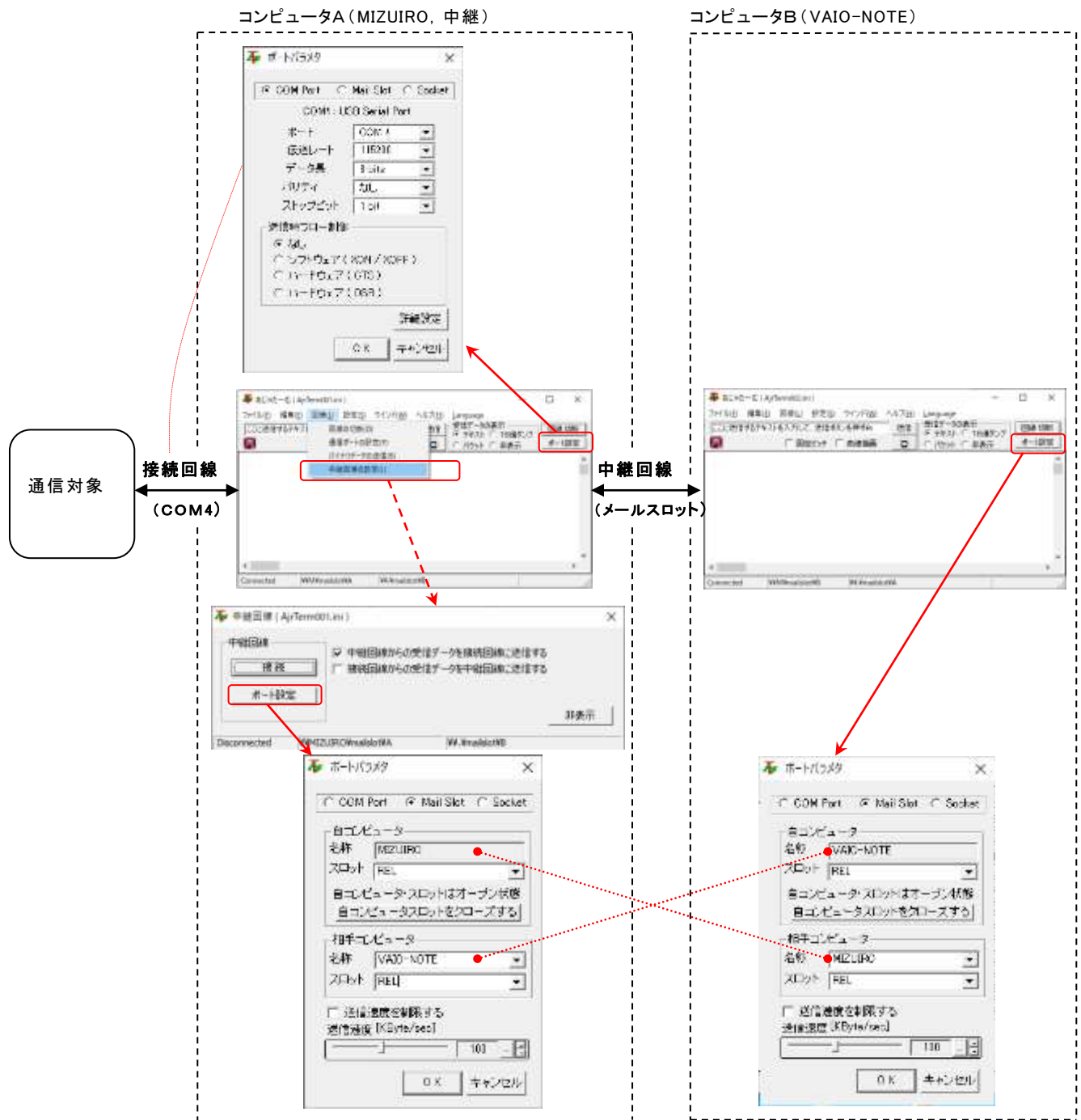
※中継回線が接続されると、メインウインドにが表示されます。



「中継回線からの受信データを接続回線に送信する」をチェックすると、中継先(上図の場合「あじやた〜むB」)から受信したデータを「通信対象」に送信します。(この受信データはウインドに表示されません)

「接続回線からの受信データを中継回線に送信する」をチェックすると、通信対象から受信したデータを中継先(上図の場合「あじやた〜むB」)に送信します。

例えば、次のようにポートを設定すると、コンピュータAにCOMポートで接続した通信対象を、コンピュータBから遠隔操作することができます。



## 10. 初期化ファイル

「あじやた〜む」で設定した内容は全て初期化ファイルとして記録されます。

初期化ファイルの格納フォルダは、デフォルトで「C:¥Ajara¥AjrTerm¥Init」です。

初期化ファイルは、(複数起動した場合は)「あじやた〜む」の起動順に「AjrTerm001.ini」「AjrTerm002.ini」・・・と割り振られますが、以下の方法で初期化ファイルを直接指定することもできます。

### 1) コマンドラインで初期化ファイルを指定する

コマンドラインの第1パラメタで初期化ファイルを指定します。

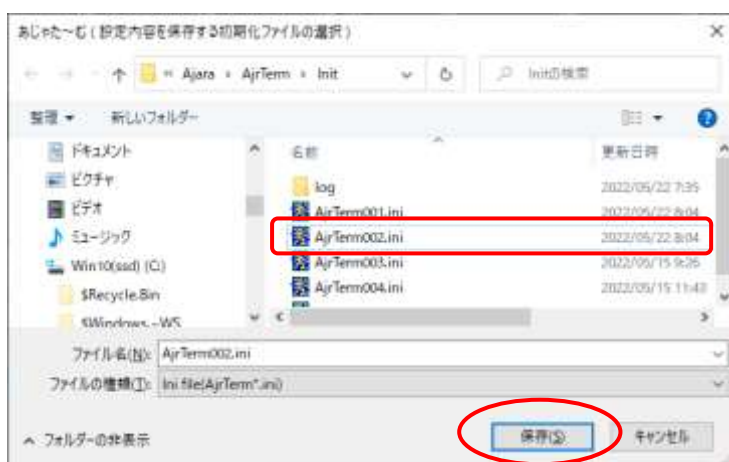
(例)

```
d:>AjrTerm.exe C:¥Ajara¥AjrTerm¥Init¥AjrInit001.ini
```

### 2) ショートカット「」へ初期化ファイルをドロップする

### 3) シフトキーを押したままで「あじやた〜む」を起動する

シフトキーを押したまま「あじやた〜む」を起動した場合は、以下のダイアログで初期化ファイルを選択できます。



上記ダイアログで、初期化ファイル(AjrTerm $nnn$ .ini)を指定し、「保存」ボタンを押します。

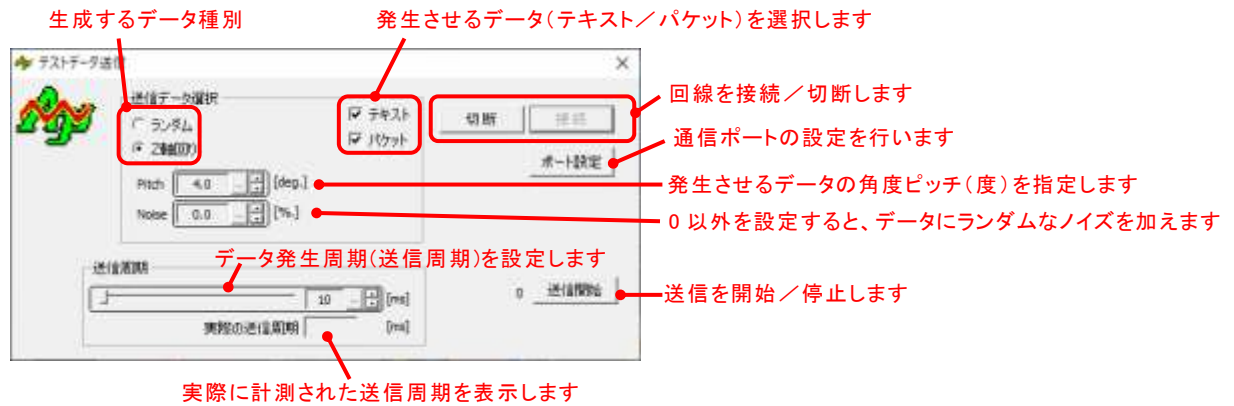
※ 初期化ファイルが Ver1.6.0.0 より前のバージョンで作成されている場合 (UNICODE ファイルでない場合)、初期化ファイルを UNICODE ファイルへ変換します。

この時、オリジナルの初期化ファイルは、初期化ファイルのフォルダ下に「bkup」フォルダを作成し、退避します。



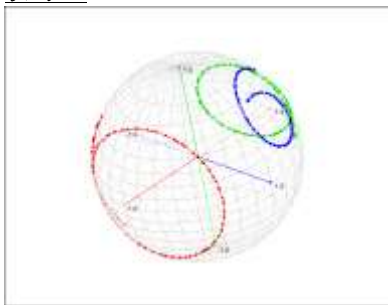
## 11. テストデータ用サンプルプログラム (Test32.exe / Test64.exe)

「あじやた〜む」の動作サンプル用として、テストデータを発生するプログラムを用意しています。  
 このプログラムは、「すべてのプログラム」→「あじやた〜む」→「テストデータプログラム」から起動します。  
 このプログラムの外観は、以下のとおりです。

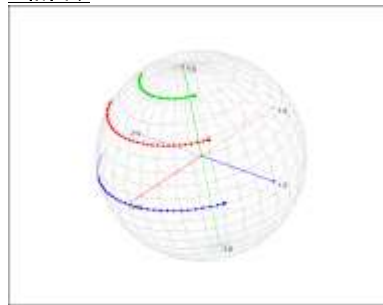


ランダム(ラジオボタン)をチェックした場合、ランダムに球面を回る、3種類の3Dベクトルデータを発生して送信します。  
 Z軸回り(ラジオボタン)をチェックした場合、球面をZ軸回りに回る、3種類の3Dベクトルデータを発生して送信します。

ランダム



Z軸回り



テキストデータで送信する場合は、以下の形式で送信します。

[PKT-A]	[V0]	-0.977,	0.213,	-0.005	[V1]	-0.785,	0.173,	0.601	[V2]	-0.423,	0.088,	0.900	'¥n'
[PKT-B]	[V0]	-0.992,	0.157,	0.005	[V1]	-0.789,	0.130,	0.600	[V2]	-0.433,	0.073,	0.904	'¥n'

行頭は「PKT-A」「PKT-B」の繰り返し、x0~z2, x0'~z2' は全て、-1.000~+1.000の実数です。

パケットデータ(バイナリデータ)で送信する場合は、以下の形式で送信します。

Loc=0	4	8	12 (0xC)	16 (0x10)	
1 固定	2 固定	x0	y0	z0	
20 (0x14)	24 (0x18)	28 (0x1C)	32 (0x20)	36 (0x24)	40 (0x28)
x1	y1	z1	x2	y2	z2

各データ(x0~z2)は全て、リトルエンディアン形式の32ビット整数(-1000~+1000)です。

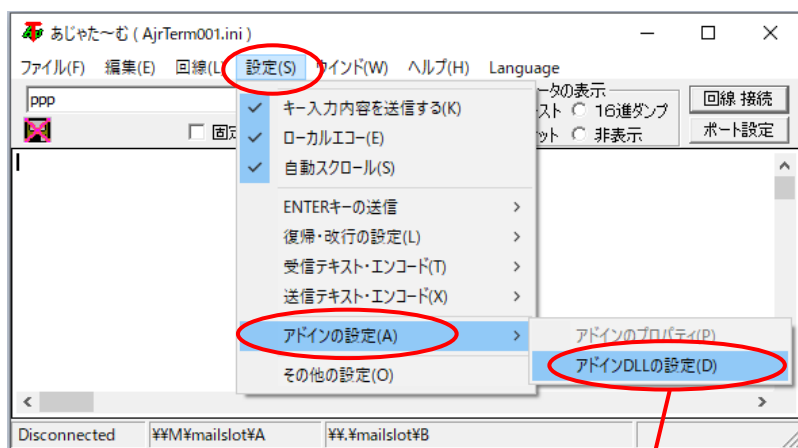
## 12. アドイン

アドイン機能により、「あじやた〜む」の機能を拡張することができます。

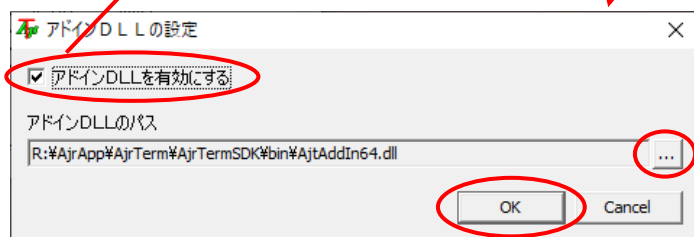
アドインを設定すると、以下のタイミングでDLL内の特定の関数が呼び出されます。

#	関数名	呼び出しタイミング	備考
1	AjtAddinValid	アドインDLLが有効となった時	
2	AjtAddinInvalid	アドインDLLが無効となった時	
3	AjtAddinSetProp	「設定」→「アドインの設定」→「アドインのプロパティ」メニュー選択時	
4	AjtAddinOpened	通信回線がオープンされた時	
5	AjtAddinClosed	通信回線がクローズされた時	
6	AjtAddinChartData	タイムチャートにデータを表示する直前	表示するデータ値を変更したり、当該データの表示を中止することができます。
7	AjtAddin2DPlotData	2Dプロットにデータを表示する直前	
8	AjtAddin3DPlotData	3Dプロットにデータを表示する直前	
9	AjtAddinChartOpen AjtAddinChartClose	タイムチャート・ウインドのオープン時／クローズ時	
10	AjtAddinChartClear	タイムチャート・ウインドのクリアー通知	
11	AjtAddin2DPlotOpen AjtAddin2DPlotClose	2Dプロット・ウインドのオープン時／クローズ時	
12	AjtAddin2DPlotClear	2Dプロット・ウインドのクリアー通知	
13	AjtAddin3DPlotOpen AjtAddin3DPlot	3Dプロット・ウインドのオープン時／クローズ時	
14	AjtAddin3DPlotClear	3Dプロット・ウインドのクリアー通知	

アドイン機能を使用するには、アドイン用DLLを作成し、「設定」→「アドインの設定」→「アドインDLLの設定」メニューで、当該DLLを指定します。



ここをチェックするとアドインDLLが有効となります。



このボタンを押すと、DLL選択ダイアログが表示されます

## 12.1. ベクトルデータの構造体

2D/3D ベクトルデータ構造体の形式は、以下のとおりです。

### 2Dベクトル

```
typedef struct {
    double x, y;
} AJC2DVEC, *PAJC2DVEC;
typedef const AJC2DVEC *PCAJC2DVEC;
```

### 3Dベクトル

```
typedef struct {
    double x, y, z;
} AJC3DVEC, *PAJC3DVEC;
typedef const AJC3DVEC *PCAJC3DVEC;
```

## 12.2. アドイン関数の仕様

アドイン DLL には、以下の関数を実装してください。(未使用のものは実装する必要はありません)

### 12.2.1. アドインDLL有効化の処理

**形式** : \_\_declspec(dllexport) void CALLBACK AjtAddinValid(HWND hwnd);

**引数** : hwnd - 「あじやた〜む」のメイン・ウインド・ハンドル

**説明** : この関数は、アドインDLLをロードし、有効となった場合に呼び出されます。

**戻り値** : なし

### 12.2.2. アドインDLL無効化の処理

**形式** : \_\_declspec(dllexport) void CALLBACK AjtAddinInvalid(HWND hwnd);

**引数** : hwnd - 「あじやた〜む」のメイン・ウインド・ハンドル

**説明** : この関数は、アドインDLLがロード済みで、無効となった場合に呼び出されます。

**戻り値** : なし

### 12.2.3. 「設定」→「アドインの設定」→「アドインのプロパティ」メニュー選択時の処理

**形式** : \_\_declspec(dllexport) void CALLBACK AjtAddinSetProp(HWND hwnd);

**引数** : hwnd - 「あじやた〜む」のメイン・ウインド・ハンドル

**説明** : この関数は、「あじやた〜む」のメインウインドで、「設定」「アドインの設定」「アドインのプロパティ」メニューが選択された時に呼び出されます。

ダイアログボックス等により、アドイン機能のパラメタを設定することができます。

**戻り値** : なし

## 12.2.4. 通信ポートのオープン時の処理

**形 式** : `__declspec(dllexport) void CALLBACK AjtAddinOpen (int com);`

**引 数** : `com` - オープンした通信ポートの種類(1~255:COM1~COM255, 256:メールスロット, 257:ソケット)

**説 明** : この関数は、通信ポートがオープンされたときに呼び出されます。

**戻り値** : なし

## 12.2.5. 通信ポートのクローズ時の処理

**形 式** : `__declspec(dllexport) void CALLBACK AjtAddinClose (void);`

**引 数** : なし

**説 明** : この関数は、通信ポートがクローズされたときに呼び出されます。

**戻り値** : なし

## 12.2.6. タイムチャートデータの値変更/表示中止

**形 式** : `__declspec(dllexport) void CALLBACK AjtAddinExcChart(HWND hwnd, int seq, int n, double dat[]);`

**引 数** : `hwnd` - タイムチャートウインドのハンドル  
`seq` - 識別番号(タイムチャートが複数表示されている場合の識別に使用します)  
`n` - データの個数(1~8)  
`dat` - タイムチャートデータ配列の先頭アドレス

**説 明** : この関数は、タイムチャートへデータを表示する直前に呼び出されます。  
「seq」は、当該タイムチャートウインドに表示されているタイトル(タイムチャート, nnn)の数値(nnn)を示します。  
「dat」はタイムチャートに表示しようとしているデータ値の配列です。  
「n」はデータ配列の要素数です。  
表示する値を変更するには、「dat」で示される配列内の値を書き換えます。  
戻り値に「FALSE」を指定すると、このデータはタイムチャートに表示されずに破棄されます。

**戻り値** : `TRUE` - このデータをタイムチャートに表示する  
`FALSE` - このデータをタイムチャートに表示せずに破棄します。

## 12.2.7. 2D プロットデータの値変更／表示中止

**形式** : `__declspec(dllexport) void CALLBACK AjtAddin2DPlotData (HWND hwnd, int seq, int n, AJC2DVEC vec[]);`

**引数** : `hwnd` - 2D プロットウインドのハンドル  
`seq` - 識別番号 (2D プロットが複数表示されている場合の識別に使用します)  
`n` - データの個数 (1 ~ 16)  
`dat` - 2D プロットデータ配列の先頭アドレス

**説明** : この関数は、2D プロットヘデータを表示する直前に呼び出されます。  
「seq」は、当該2Dプロットウインドに表示されているタイトル(2Dプロット, nnn)の数値(nnn)を示します。  
「dat」は2Dプロットに表示しようとしているデータ値(x, y)の配列です。  
「n」はデータ配列の要素数です。  
表示する値を変更するには、「dat」で示される配列内の値を書き換えます。  
戻り値に「FALSE」を指定すると、このデータは2Dプロットに表示されずに破棄されます。

**戻り値** : TRUE - このデータを2Dプロットに表示する  
FALSE - このデータを2Dプロットに表示せずに破棄します。

## 12.2.8. 3D プロットデータの値変更／表示中止

**形式** : `__declspec(dllexport) void CALLBACK AjtAddin3DPlotData(HWND hwnd, int seq, int n, AJC3DVEC vec[]);`

**引数** : `hwnd` - 3D プロットウインドのハンドル  
`seq` - 識別番号 (3D プロットが複数表示されている場合の識別に使用します)  
`n` - データの個数 (1 ~ 16)  
`dat` - 3D プロットデータ配列の先頭アドレス

**説明** : この関数は、3D プロットヘデータを表示する直前に呼び出されます。  
「seq」は、当該3Dプロットウインドに表示されているタイトル(3Dプロット, nnn)の数値(nnn)を示します。  
「dat」は3Dプロットに表示しようとしているデータ値(x, y, z)の配列です。  
「n」はデータ配列の要素数です。  
表示する値を変更するには、「dat」で示される配列内の値を書き換えます。  
戻り値に「FALSE」を指定すると、このデータは3Dプロットに表示されずに破棄されます。

**戻り値** : TRUE - このデータを3Dプロットに表示する  
FALSE - このデータを3Dプロットに表示せずに破棄します。

## 12.2.9. タイムチャート・ウインドのオープン/クローズ通知

**形式** : `__declspec(dllexport) void CALLBACK AjtAddinChartOpen(HWND hwnd, int seq, HWND hCtrl);`  
`__declspec(dllexport) void CALLBACK AjtAddinChartClose(HWND hwnd, int seq);`

**引数** : `hwnd` - タイムチャートウインドのハンドル(ダイアログやコントロールを含むコンテナウインド)  
`seq` - 識別番号(タイムチャートが複数表示されている場合の識別に使用します)  
`hCtrl` - タイムチャート・コントロールのハンドル

**説明** : この関数は、タイムチャートウインドがオープン/クローズされた時にコールされます。  
「seq」は、当該タイムチャートウインドに表示されているタイトル(タイムチャート, nnn)の数値(nnn)を示します。

**戻り値** : なし

## 12.2.10. タイムチャート・ウインドのデータクリアー通知

**形式** : `__declspec(dllexport) void CALLBACK AjtAddinChartClear(HWND hwnd, int seq);`  
`__declspec(dllexport) void CALLBACK AjtAddinChartClear(HWND hwnd, int seq);`

**引数** : `hwnd` - タイムチャートウインドのハンドル(ダイアログやコントロールを含むコンテナウインド)  
`seq` - 識別番号(タイムチャートが複数表示されている場合の識別に使用します)

**説明** : この関数は、タイムチャートウインドの表示データをクリアーした際ににコールされます。  
「seq」は、当該タイムチャートウインドに表示されているタイトル(タイムチャート, nnn)の数値(nnn)を示します。

**戻り値** : なし

## 12.2.11. 2Dプロット・ウインドのオープン/クローズ通知

**形式** : `__declspec(dllexport) void CALLBACK AjtAddin2DPlotOpen(HWND hwnd, int seq, HWND hCtrl);`  
`__declspec(dllexport) void CALLBACK AjtAddin2DPlotClose(HWND hwnd, int seq);`

**引数** : `hwnd` - 2Dプロットウインドのハンドル(ダイアログやコントロールを含むコンテナウインド)  
`seq` - 識別番号(2Dプロットが複数表示されている場合の識別に使用します)  
`hCtrl` - 2Dプロット・コントロールのハンドル

**説明** : この関数は、2Dプロットウインドの表示データをクリアーした際ににコールされます。  
「seq」は、当該2Dプロットウインドに表示されているタイトル(2Dプロット, nnn)の数値(nnn)を示します。

**戻り値** : なし

## 12.2.12. 2Dプロット・ウィンドのデータクリアー通知

**形 式** : `__declspec(dllexport) void CALLBACK AjtAddin2DPlotClear(HWND hwnd, int seq, int kind);`

**引 数** : `hwnd` - 2Dプロットウィンドのハンドル(ダイアログやコントロールを含むコンテナウィンド)  
`seq` - 識別番号(2Dプロットが複数表示されている場合の識別に使用します)  
`kind` - クリアーしたデータの種別を示します(0x01:プロットデータ, 0x02:描画データ, 0x03:全データ)

**説 明** : この関数は、2Dプロットウィンドがクリアーされた時にコールされます。  
「seq」は、当該2Dプロットウィンドに表示されているタイトル(2Dプロット, nnn)の数値(nnn)を示します。

**戻り値** : なし

## 12.2.13. 3Dプロット・ウィンドのオープン/クローズ通知

**形 式** : `__declspec(dllexport) void CALLBACK AjtAddin3DPlotOpen(HWND hwnd, int seq, HWND hCtrl);`  
`__declspec(dllexport) void CALLBACK AjtAddin3DPlotClose(HWND hwnd, int seq);`

**引 数** : `hwnd` - 3Dプロットウィンドのハンドル(ダイアログやコントロールを含むコンテナウィンド)  
`seq` - 識別番号(3Dプロットが複数表示されている場合の識別に使用します)  
`hCtrl` - 3Dプロット・コントロールのハンドル

**説 明** : この関数は、3Dプロットウィンドがオープン/クローズされた時にコールされます。  
「seq」は、当該3Dプロットウィンドに表示されているタイトル(3Dプロット, nnn)の数値(nnn)を示します。

**戻り値** : なし

## 12.2.14. 3Dプロット・ウィンドのデータクリアー通知

**形 式** : `__declspec(dllexport) void CALLBACK AjtAddin3DPlotClear(HWND hwnd, int seq, int kind);`

**引 数** : `hwnd` - 3Dプロットウィンドのハンドル(ダイアログやコントロールを含むコンテナウィンド)  
`seq` - 識別番号(3Dプロットが複数表示されている場合の識別に使用します)  
`kind` - クリアーしたデータの種別を示します(0x01:プロットデータ, 0x02:描画データ, 0x03:全データ)

**説 明** : この関数は、3Dプロットウィンドがオープン/クローズされた時にコールされます。  
「seq」は、当該3Dプロットウィンドに表示されているタイトル(3Dプロット, nnn)の数値(nnn)を示します。

**戻り値** : なし

### 12.3. アドインから あじやた〜むへのメッセージ

アドインから、ウインド(タイムチャートや2D/3Dプロット)へメッセージを送信することにより、以下の操作が可能です。

名称	文字タイプ	値	内容	wParam	lParam
WM_AJT_CLEAR_DATA	—	WM_USER + 102	全データクリアー	0	0
WM_AJT_SHOW_DATA	—	WM_USER + 110	データ投与	0	double 型データ配列のアドレス
WM_AJT_SET_TOOLTEXT	ANSI	WM_USER + 160	ツールテキスト設定	0	ツールチップ文字列のアドレス
	UNICODE	WM_USER + 161			
WM_AJT_SET_TITLETEXT	ANSI	WM_USER + 162	タイトル表示	0	タイトル文字列のアドレス
	UNICODE	WM_USER + 163			

※上記シンボルは AjtAddinDef.h で定義されています。

※「WM\_AJT\_SHOW\_DATA」は、タイムチャートでのみ可能です。

### 12.4. アドイン関数の DLL エクスポート名称

アドイン関数は、DLL から以下の名称でエクスポートしてください。

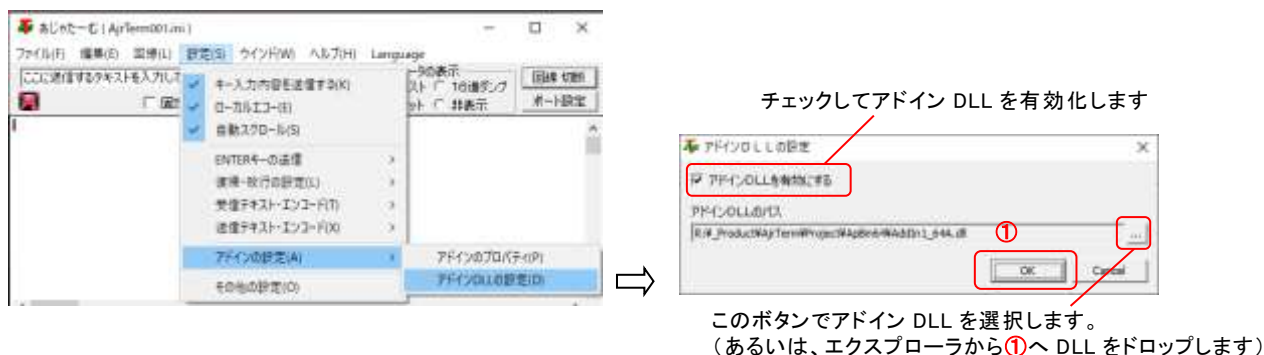
#	関数名	エクスポート名称	
		32ビット(WIN32)	64ビット(WIN64)
1	AjtAddinValid	_AjtAddinValid@4	AjtAddinValid
2	AjtAddinInvalid	_AjtAddinInvalid@4	AjtAddinInvalid
3	AjtAddinSetProp	_AjtAddinSetProp@4	AjtAddinSetProp
4	AjtAddinComOpen	_AjtAddinComOpen@4	AjtAddinComOpen
5	AjtAddinComClose	_AjtAddinComClose@0	AjtAddinComClose
6	AjtAddinChartOpen	_AjtAddinChartOpen@12	AjtAddinChartOpen
7	AjtAddinChartData	_AjtAddinChartData@16	AjtAddinChartData
8	AjtAddinChartClose	_AjtAddinChartClose@8	AjtAddinChartClose
9	AjtAddinChartClear	_AjtAddinChartClear@8	AjtAddinChartClear
10	AjtAddin2DPlotOpen	_AjtAddin2DPlotOpen@12	AjtAddin2DPlotOpen
11	AjtAddin2DPlotData	_AjtAddin2DPlotData@16	AjtAddin2DPlotData
12	AjtAddin2DPlotClear	_AjtAddin2DPlotClear@12	AjtAddin2DPlotClear
13	AjtAddin2DPlotClose	_AjtAddin2DPlotClose@8	AjtAddin2DPlotClose
14	AjtAddin3DPlotOpen	_AjtAddin3DPlotOpen@12	AjtAddin3DPlotOpen
15	AjtAddin3DPlotData	_AjtAddin3DPlotData@16	AjtAddin3DPlotData
16	AjtAddin3DPlotClose	_AjtAddin3DPlotClose@8	AjtAddin3DPlotClose
17	AjtAddin3DPlotClear	_AjtAddin3DPlotClear@12	AjtAddin3DPlotClear



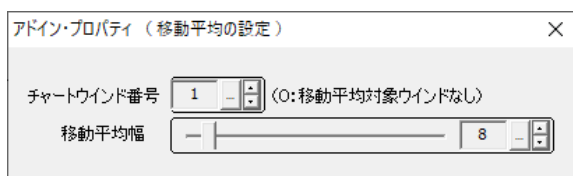
## 12.5. アドイン・サンプルプログラム

以下のサンプルプログラムは、あじやた〜むのアドインとして、指定タイムチャートウィンドで移動平均波形の表示を行います。(2D/3D プロットに関しては、何も処理を行っていません)

まず、「設定」→「アドインの設定」→「アドイン DLL の設定」で、アドイン DLL を指定します。



アドイン DLL を有効にすると、以下の移動平均の設定ダイアログが表示されます。



ダイアログを閉じて、再度表示するには「設定」→「アドインの設定」→「アドインのプロパティ」を選択します。

チャートウィンド番号は、移動平均するタイムチャートウィンドの番号(タイトルバーに表示される番号)です。

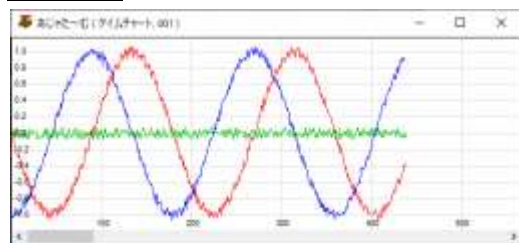


チャートウィンド番号にゼロ(0)を指定した場合は、移動平均を行いません。

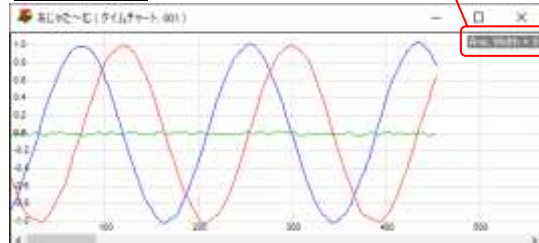
移動平均幅は、移動平均に使用するデータの個数を設定します。

指定チャートウィンドにおいて、以下のように移動平均されたチャートが表示されます。

移動平均なし



移動平均あり



※このサンプルプログラムでは、「カスタムライブラリ」(AjrCst32.dll/AjrCst64.dll)を使用しています。

AjrCst32.dll/AjrCst64.dll については、「<http://www.ne.jp/asahi/ajara/kojara/ajrct132.htm>」を参照してください。

AjtAddinDef.h

```

1 :
2 : #ifndef AJTADDINDEF_H_
3 : #define AJTADDINDEF_H_
4 :
5 : /*****
6 :
7 : #ifdef UNICODE
8 :     #define WM_AJT_SET_TOOLTEXT    WM_AJT_SET_TOOLTEXTW
9 :     #define WM_AJT_SET_TITLETEXT  WM_AJT_SET_TITLETEXTW
10 : #else
11 :     #define WM_AJT_SET_TOOLTEXT    WM_AJT_SET_TOOLTEXTA
12 :     #define WM_AJT_SET_TITLETEXT  WM_AJT_SET_TITLETEXTA
13 : #endif
14 :
15 : #define WM_AJT_CLEAR_DATA          (WM_USER + 102) // アドインからのデータクリア          (wParam = 0, lParam = 0          )
16 : #define WM_AJT_SHOW_DATA           (WM_USER + 110) // アドインからのデータ表示メッセージ (wParam = 0, lParam = データアドレス )
17 :
18 : #define WM_AJT_SET_TOOLTEXTA       (WM_USER + 160) // アドインからのツールテキスト設定 (wParam = 0, lParam = ANSI テキストアドレス )
19 : #define WM_AJT_SET_TOOLTEXTW      (WM_USER + 161) // アドインからのツールテキスト設定 (wParam = 0, lParam = UNICODE Text アドレス )
20 :
21 : #define WM_AJT_SET_TITLETEXTA      (WM_USER + 162) // アドインからのタイトルテキスト設定 (wParam = 0, lParam = ANSI テキストアドレス )
22 : #define WM_AJT_SET_TITLETEXTW     (WM_USER + 163) // アドインからのタイトルテキスト設定 (wParam = 0, lParam = UNICODE Text アドレス )
23 :
24 : /*****
25 : #endif

```

AjtAddin1.c

```

1 : /*****
2 : //
3 : // あじやた〜む用アドインDLL (タイムチャートの移動平均処理) //
4 : // //
5 : /*****
6 : #pragma warning(disable:4996)
7 : #include <AjrCstXX.h>
8 : #include <AjtAddinDef.h>
9 : #include <mbctype.h>
10 : #include "resource.h"
11 :
12 : #define MYSECT TEXT("MySect")
13 : #define PROPBIF TEXT("AjtAddinPropBif")
14 : #define PROPSEQ TEXT("AjtAddinPropSeq")
15 : #define DLLEXPORT __declspec ( dllexport )
16 :
17 : //-----//
18 : // 移動平均データバッファエントリ //
19 : //-----//
20 : #define MAX_CH 8
21 :
22 : typedef struct {
23 :     double dat[MAX_CH];
24 : } AVEDAT, *PAVEDAT;
25 : //-----//
26 : // 移動平均データバッファ情報 //
27 : //-----//
28 : typedef struct {
29 :     int num; // バッファサイズ (データ数)
30 :     int ix; // データ格納インデクス
31 :     int cnt; // データ数カウンタ
32 :     PAVEDAT pBuf; // バッファアドレス
33 : } BUFINFO, *PBUFINFO;
34 : //-----//
35 : // 作業領域 //
36 : //-----//
37 : static HINSTANCE hDllInst = NULL; // DLLインスタンス ハンドル
38 : static HWND hDlgAve = NULL; // ダイアログ・ハンドル

```

```

39 : static UI      AveWndSeq   = 0;          // 移動平均対象チャートウインド番号 (0 : 移動平均なし)
40 : static UI      AveWidth    = 10;         // 移動平均幅
41 :
42 : //-----//
43 : // 内部サブ関数 //
44 : //-----//
45 : AJC_DLGPROC_DEF(SetAddinProp);
46 : static BOOL     IsChartWindow(HWND hwnd);
47 : static PBUFINFO GetAveBuf (HWND hwnd);
48 : static PBUFINFO AllocAveBuf(HWND hwnd);
49 : static VOID     FreeAveBuf (HWND hwnd);
50 : static BOOL CALLBACK cbEnumShowTitle(HWND hwnd, LPARAM lParam);
51 : static BOOL CALLBACK cbEnumAlloc (HWND hwnd, LPARAM lParam);
52 : static BOOL CALLBACK cbEnumFree (HWND hwnd, LPARAM lParam);
53 :
54 : //=====//
55 : // //
56 : // DllMain //
57 : // //
58 : //=====//
59 : int WINAPI DllMain (HINSTANCE hInstance, DWORD fdwReason, PVOID pvReserved)
60 : {
61 :     int rc = TRUE;
62 :
63 :     //----- 初期設定処理 -----//
64 :     if (fdwReason == DLL_PROCESS_ATTACH) { // プロセス起動?
65 :         hDllInst = hInstance;
66 :         //----- プロパティ値ロード -----//
67 :         AJCGETPF_UINT(MYSECT, AveWndSeq, 0);
68 :         AJCGETPF_UINT(MYSECT, AveWidth, 10);
69 :     }
70 :     //----- 終了時後処理 -----//
71 :     else if (fdwReason == DLL_PROCESS_DETACH) { // プロセス終了?
72 :         //----- プロパティ値セーブ -----//
73 :         AJCPUTPF_UINT(MYSECT, AveWndSeq);
74 :         AJCPUTPF_UINT(MYSECT, AveWidth);
75 :     }
76 :     return rc;
77 : }
78 : //=====//
79 : // アドインDLL有効化 //
80 : // //
81 : // 引数 : hwnd - あじやた〜むメインウインドのハンドル //
82 : // //
83 : // 戻り値 : なし //
84 : //=====//
85 : DLLEXPORT VOID CALLBACK AjtAddinSetProp(HWND hwnd);
86 :
87 : DLLEXPORT VOID CALLBACK AjtAddinValid(HWND hwnd)
88 : {
89 :     // ダイアログ表示
90 :     AjtAddinSetProp(hwnd);
91 :     // チャートウインドを検索し、移動平均バッファ割り当て
92 :     EnumThreadWindows(GetCurrentThreadId(), (WNDENUMPROC) cbEnumAlloc, 0);
93 : }
94 : //=====//
95 : // アドインDLL無効化 //
96 : // //
97 : // 引数 : hwnd - あじやた〜むメインウインドのハンドル //
98 : // //
99 : // 戻り値 : なし //
100 : //=====//
101 : DLLEXPORT VOID CALLBACK AjtAddinInvalid(HWND hwnd)
102 : {
103 :     // チャートウインドを検索し、移動平均用バッファ破棄
104 :     EnumThreadWindows(GetCurrentThreadId(), (WNDENUMPROC) cbEnumFree, 0);
105 :
106 :     // 移動平均設定ダイアログクローズ
107 :     if (hDlgAve != NULL) {
108 :         DestroyWindow(hDlgAve);
109 :         hDlgAve = NULL;
110 :     }
111 : }
112 : //=====//
113 : // アドインプロパティの設定 //
114 : // //
115 : // 引数 : hwnd - あじやた〜むメインウインドのハンドル //
116 : // //

```

```

117 : // 戻り値 : なし //
118 : //=====//
119 : DLLEXPORT VO CALLBACK AjtAddinSetProp(HWND hwnd)
120 : {
121 :     if (hDlgAve == NULL) {
122 :         // ダイアログ表示
123 :         hDlgAve = CreateDialogParam(hDllInst, MAKEINTRESOURCE(IDD_ADDIN_PROP), hwnd,
124 :                                     AJC_DLGPROC_NAME(SetAddinProp), (LPARAM)hwnd);
125 :         ShowWindow(hDlgAve, SW_SHOWNORMAL);
126 :     }
127 : }
128 : //-----//
129 : // アドイン・プロパティ設定ダイアログ・プロシージャ //
130 : //-----//
131 : //---- ダイアログ初期化 -----//
132 : AJC_DLGPROC(SetAddinProp, WM_INITDIALOG )
133 : {
134 :     // ダイアログにあじやた〜むウインドハンドルを関連付け
135 :     SetProp(hDlg, TEXT("hWndAjrTerm"), (HANDLE)IParam);
136 :     // 日英テキスト
137 :     SetWindowText(hDlg, AJCLNGSEL(TEXT("アドイン・プロパティ (移動平均の設定)"),
138 :                                     TEXT("Add-in properties (setting of moving average)")));
139 :     AjcSetDlgItemStr(hDlg, IDC_LBL_AVEWNDSEQ, AJCLNGSEL(TEXT("チャートウインド番号"), TEXT("Chart window number")));
140 :     AjcSetDlgItemStr(hDlg, IDC_LBL_AVEWIDTH, AJCLNGSEL(TEXT("移動平均幅"), TEXT("Moving average width")));
141 :     AjcSetDlgItemStr(hDlg, IDC_LBL_NOTE, AJCLNGSEL(TEXT(" (0 : 移動平均対象ウインドなし) "),
142 :                                     TEXT("(0: No window for moving average)")));
143 :     // ダイアログ項目の設定
144 :     AjcSetDlgItemUInt(hDlg, IDC_INP_AVEWNDSEQ, AveWndSeq);
145 :     AjcSetDlgItemUInt(hDlg, IDC_INP_AVEWIDTH, AveWidth);
146 :
147 :     return TRUE;
148 : }
149 : //---- ダイアログ破棄 -----//
150 : AJC_DLGPROC(SetAddinProp, WM_DESTROY )
151 : {
152 :     hDlgAve = NULL;
153 :     return TRUE;
154 : }
155 : //---- 移動平均対象の設定 -----//
156 : AJC_DLGPROC(SetAddinProp, IDC_INP_AVEWNDSEQ )
157 : {
158 :     if (HIWORD(wParam) == AJCIVN_INTVALUE) {
159 :         if (AveWndSeq != (UI)IParam) {
160 :             AveWndSeq = (UI)IParam;
161 :             // チャートウインドを検索し、タイトル表示
162 :             EnumThreadWindows(GetCurrentThreadId(), (WNDENUMPROC)cbEnumShowTitle, 0);
163 :         }
164 :     }
165 :     return TRUE;
166 : }
167 : //---- 移動平均幅の設定 -----//
168 : AJC_DLGPROC(SetAddinProp, IDC_INP_AVEWIDTH )
169 : {
170 :     HWND hwnd = (HWND)GetProp(hDlg, TEXT("hWndAjrTerm"));
171 :
172 :     if (HIWORD(wParam) == AJCIVN_INTVALUE) {
173 :         if (AveWidth != (UI)IParam) {
174 :             // 移動平均幅設定
175 :             AveWidth = (UI)IParam;
176 :             // チャートウインドを検索し、移動平均バッファ再割り当て
177 :             EnumThreadWindows(GetCurrentThreadId(), (WNDENUMPROC)cbEnumAlloc, 0);
178 :         }
179 :     }
180 :     return TRUE;
181 : }
182 : //---- キャンセル -----//
183 : AJC_DLGPROC(SetAddinProp, IDCANCEL )
184 : {
185 :     DestroyWindow(hDlg);
186 :     return TRUE;
187 : }
188 : //---- ウインド通知 (タイトル表示) -----//
189 : static BOOL CALLBACK cbEnumShowTitle(HWND hwnd, LPARAM IParam)
190 : {
191 :     UI seq = 0;
192 :     UT txt[64] = {0};
193 :
194 :     if (IsChartWindow(hwnd)) { // チャートウインド?

```

```

195 :         seq = (UI)((UX)GetProp(hwnd, PROPSEQ));           //   チャートウインド番号取得
196 :         if (seq == AveWndSeq) {
197 :             AjsnPrintf(txt, AJCTSIZE(txt), TEXT("Ave. Width = %d"), AveWidth);
198 :         }
199 :         SendMessage(hwnd, WM_AJT_SET_TITLETEXT, 0, (LPARAM)txt);
200 :     }
201 :     return TRUE;
202 : }
203 : //----- ウインド通知 (移動平均バッファ割り当て) -----//
204 : static BOOL CALLBACK cbEnumAlloc(HWND hwnd, LPARAM lParam)
205 : {
206 :     if (IsChartWindow(hwnd)) {                             //   チャートウインド?
207 :         AllocAveBuf(hwnd);                                 //   移動平均バッファ割り当て
208 :         cbEnumShowTitle(hwnd, lParam);                    //   タイトル表示
209 :     }
210 :     return TRUE;
211 : }
212 : //----- ウインド通知 (移動平均バッファ解放) -----//
213 : static BOOL CALLBACK cbEnumFree(HWND hwnd, LPARAM lParam)
214 : {
215 :     PBUFINFO    pW;
216 :
217 :     if (IsChartWindow(hwnd)) {                             //   チャートウインド?
218 :         if (pW = (PBUFINFO)GetProp(hwnd, PROPBIF)) {     //   移動平均バッファ有り?
219 :             FreeAveBuf(hwnd);                             //   移動平均バッファ破棄
220 :             cbEnumShowTitle(hwnd, lParam);                //   タイトル消去
221 :         }
222 :     }
223 :     return TRUE;
224 : }
225 : //-----//
226 : AJC_DLGMAP_DEF(SetAddinProp)
227 :     AJC_DLGMAP_MSG(SetAddinProp, WM_INITDIALOG    )
228 :     AJC_DLGMAP_MSG(SetAddinProp, WM_DESTROY      )
229 :     AJC_DLGMAP_CMD(SetAddinProp, IDC_INP_AVEWNDSEQ )
230 :     AJC_DLGMAP_CMD(SetAddinProp, IDC_INP_AVEWIDTH )
231 :     AJC_DLGMAP_CMD(SetAddinProp, IDCANCEL        )
232 : AJC_DLGMAP_END
233 :
234 : //=====//
235 : // 通信ポートのオープン通知                                     //
236 : //                                                             //
237 : // 引 数   : com      - オープンしたCOMポート番号 (1~255)   //
238 : //                                                             //
239 : // 戻り値  : なし                                           //
240 : //=====//
241 : DLLEXPORT V0 CALLBACK AjtAddinComOpen(int com)
242 : {
243 : }
244 : //=====//
245 : // 通信ポートのクローズ通知                                   //
246 : //                                                             //
247 : // 引 数   : なし                                           //
248 : //                                                             //
249 : // 戻り値  : なし                                           //
250 : //=====//
251 : DLLEXPORT V0 CALLBACK AjtAddinComClose(V0)
252 : {
253 : }
254 : //=====//
255 : // タイムチャートウインドのオープン                           //
256 : //                                                             //
257 : // 引 数   : hwnd     - タイムチャートウインドのハンドル   //
258 : //           seq      - 識別番号 (1~, タイムチャートが複数表示されている場合の識別に使用します) //
259 : //           hCtrl    - タイムチャート・コントロールのハンドル //
260 : //                                                             //
261 : // 戻り値  : なし                                           //
262 : //=====//
263 : DLLEXPORT V0 CALLBACK AjtAddinChartOpen(HWND hwnd, int seq, HWND hCtrl)
264 : {
265 :     PBUFINFO    pW = NULL;
266 :
267 :     // 移動平均対象チャートならば、バッファ情報ワークを確保
268 :     if (AveWndSeq == seq) {
269 :         // 移動平均バッファ割り当て
270 :         AllocAveBuf(hwnd);
271 :         // タイトル表示
272 :         cbEnumShowTitle(hwnd, 0);

```

```

273 :     }
274 : }
275 : //=====//
276 : // タイムチャートウインドのクローズ //
277 : // //
278 : // 引数 : hwnd - タイムチャートウインドのハンドル //
279 : //      seq - 識別番号 (1~, タイムチャートが複数表示されている場合の識別に使用します) //
280 : // //
281 : // 戻り値 : なし //
282 : //=====//
283 : DLLEXPORT VOID CALLBACK AjtAddinChartClose(HWND hwnd, int seq)
284 : {
285 :     // 移動平均バッファを破棄
286 :     FreeAveBuf(hwnd);
287 : }
288 :
289 : //=====//
290 : // タイムチャートデータの通知/変更 //
291 : // //
292 : // 引数 : hwnd - タイムチャートウインドのハンドル //
293 : //      seq - 識別番号 (1~, タイムチャートが複数表示されている場合の識別に使用します) //
294 : //      n   - データの個数 (1~8) //
295 : //      dat - タイムチャートデータ配列の先頭アドレス (この配列の要素数は「n」です) //
296 : // //
297 : // 戻り値 : TRUE - このデータをタイムチャートに表示します //
298 : //          FALSE - このデータをタイムチャートへ表示せずに破棄します //
299 : //=====//
300 : DLLEXPORT BOOL CALLBACK AjtAddinChartData(HWND hwnd, int seq, int n, double dat[])
301 : {
302 :     BOOL rc = TRUE; // 戻り値
303 :     int i, j;
304 :     PBUFINFO pW; // バッファ情報ポインタ
305 :     PAVEDAT p; // バッファデータ項目ポインタ
306 :     double ave[MAX_CH]; // 移動平均算出ワーク
307 :
308 :     // 移動平均対象チャートならば、移動平均処理実行
309 :     if (AveWndSeq == seq) {
310 :         // 移動平均バッファ情報取得 (バッファ無しならば割り当て)
311 :         if ((pW = GetAveBuf(hwnd)) != NULL) {
312 :             // バッファにデータを格納
313 :             memcpy(&pW->pBuf[pW->ix], dat, sizeof(double) * n);
314 :             pW->ix = ((pW->ix + 1) % pW->num);
315 :             if (pW->cnt < pW->num) pW->cnt++;
316 :             // バッファ満了ならば、移動平均計算
317 :             if (pW->cnt >= pW->num) {
318 :                 for (i = 0; i < n; i++) {
319 :                     ave[i] = 0.0;
320 :                     for (j = 0, p = pW->pBuf; j < pW->num; j++, p++) {
321 :                         ave[i] += p->dat[i];
322 :                     }
323 :                     ave[i] /= (double)pW->num;
324 :                 }
325 :                 // データを移動平均値で書き換え
326 :                 memcpy(dat, ave, sizeof(double) * n);
327 :             }
328 :             // バッファ未了ならば、データ非表示
329 :             else rc = FALSE;
330 :         }
331 :     }
332 :     return rc;
333 : }
334 : //=====//
335 : // タイムチャートウインドのクリアー //
336 : // //
337 : // 引数 : hwnd - タイムチャートウインドのハンドル //
338 : //      seq - 識別番号 (1~, タイムチャートが複数表示されている場合の識別に使用します) //
339 : // //
340 : // 戻り値 : なし //
341 : //=====//
342 : DLLEXPORT VOID CALLBACK AjtAddinChartClear(HWND hwnd, int seq)
343 : {
344 :     PBUFINFO pW; // バッファ情報ポインタ
345 :
346 :     // 移動平均バッファ情報リセット
347 :     if (pW = (PBUFINFO)GetProp(hwnd, TEXT(PROPBIF))) {
348 :         pW->num = AveWidth;
349 :         pW->ix = 0;
350 :         pW->cnt = 0;

```

```

351 :     }
352 : }
353 : //=====//
354 : // 2Dプロットウィンドのオープン //
355 : // //
356 : // 引数 : hwnd - 2Dプロットウィンドのハンドル //
357 : //      seq - 識別番号(1~, タイムチャートが複数表示されている場合の識別に使用します) //
358 : //      hCtrl - 2Dプロット・コントロールのハンドル //
359 : // //
360 : // 戻り値 : なし //
361 : //=====//
362 : DLLEXPORT VOID CALLBACK AjtAddin2DPlotOpen(HWND hwnd, int seq, HWND hCtrl)
363 : {
364 : }
365 : //=====//
366 : // 2Dプロットウィンドのクローズ //
367 : // //
368 : // 引数 : hwnd - タイムチャートウィンドのハンドル //
369 : //      seq - 識別番号(1~, タイムチャートが複数表示されている場合の識別に使用します) //
370 : // //
371 : // 戻り値 : なし //
372 : //=====//
373 : DLLEXPORT VOID CALLBACK AjtAddin2DPlotClose(HWND hwnd, int seq)
374 : {
375 : }
376 : //=====//
377 : // 2Dプロットデータの通知/変更 //
378 : // //
379 : // //
380 : // 引数 : hwnd - 2Dプロットウィンドのハンドル //
381 : //      seq - 識別番号(2Dプロットが複数表示されている場合の識別に使用します) //
382 : //      n - データの個数(1~16) //
383 : //      dat - 2Dベクトルデータ配列の先頭アドレス(この配列の要素数は「n」です) //
384 : // //
385 : // 戻り値 : TRUE - このデータを2Dプロットに表示します //
386 : //          FALSE - このデータを2Dプロットへ表示せずに破棄します //
387 : //=====//
388 : DLLEXPORT BOOL CALLBACK AjtAddin2DPlotData(HWND hwnd, int seq, int n, AJC2DVEC vec[])
389 : {
390 :     return TRUE;
391 : }
392 : //=====//
393 : // 2Dプロットウィンドのクリアー //
394 : // //
395 : // 引数 : hwnd - タイムチャートウィンドのハンドル //
396 : //      seq - 識別番号(1~, タイムチャートが複数表示されている場合の識別に使用します) //
397 : //      kind - クリアー種別(0x01:プロットデータ, 0x02:描画データ, 0x03:全データ) //
398 : // //
399 : // 戻り値 : なし //
400 : //=====//
401 : DLLEXPORT VOID CALLBACK AjtAddin2DPlotClear(HWND hwnd, int seq, int kind)
402 : {
403 : }
404 : //=====//
405 : // 3Dプロットウィンドのオープン //
406 : // //
407 : // 引数 : hwnd - 3Dプロットウィンドのハンドル //
408 : //      seq - 識別番号(1~, タイムチャートが複数表示されている場合の識別に使用します) //
409 : //      hCtrl - 3Dプロット・コントロールのハンドル //
410 : // //
411 : // 戻り値 : なし //
412 : //=====//
413 : DLLEXPORT VOID CALLBACK AjtAddin3DPlotOpen(HWND hwnd, int seq, HWND hCtrl)
414 : {
415 : }
416 : //=====//
417 : // 3Dプロットウィンドのクローズ //
418 : // //
419 : // 引数 : hwnd - タイムチャートウィンドのハンドル //
420 : //      seq - 識別番号(1~, タイムチャートが複数表示されている場合の識別に使用します) //
421 : // //
422 : // 戻り値 : なし //
423 : //=====//
424 : DLLEXPORT VOID CALLBACK AjtAddin3DPlotClose(HWND hwnd, int seq)
425 : {
426 : }
427 : //=====//
428 : // 3Dプロットデータの通知/変更 //

```

```

429 : // //
430 : // 引 数 : hwnd - 3Dプロットウインドのハンドル //
431 : // seq - 識別番号(3Dプロットが複数表示されている場合の識別に使用します) //
432 : // n - データの個数(1~16) //
433 : // dat - 3Dベクトルデータデータ配列の先頭アドレス(この配列の要素数は「n」です) //
434 : // //
435 : // 戻り値 : TRUE - このデータを3Dプロットに表示します //
436 : // FALSE - このデータを3Dプロットへ表示せずに破棄します //
437 : //=====//
438 : DLLEXPORT BOOL CALLBACK AjtAddin3DPlotData(HWND hwnd, int seq, int n, AJC3DVEC vec[])
439 : {
440 :     return TRUE;
441 : }
442 : //=====//
443 : // 3Dプロットウインドのクリア //
444 : // //
445 : // 引 数 : hwnd - タイムチャートウインドのハンドル //
446 : // seq - 識別番号(1~, タイムチャートが複数表示されている場合の識別に使用します) //
447 : // kind - クリアー種別( ) 0x01:プロットデータ, 0x02:描画データ, 0x03:全データ //
448 : // //
449 : // 戻り値 : なし //
450 : //=====//
451 : DLLEXPORT VOID CALLBACK AjtAddin3DPlotClear(HWND hwnd, int seq, int kind)
452 : {
453 : }
454 : //-----//
455 : // タイムチャートウインドかチェック //
456 : // //
457 : // 引 数 : hwnd - ウインドのハンドル //
458 : // //
459 : // 戻り値 : TRUE - タイムチャートウインドである //
460 : // FALSE - タイムチャートウインド以外 //
461 : // //
462 : //-----//
463 : static BOOL IsChartWindow(HWND hwnd)
464 : {
465 :     UT cname[256];
466 :     GetClassName(hwnd, cname, 256);
467 :     return (MAJcStrCmp(cname, TEXT("CBackTmc")) == 0);
468 : }
469 : //-----//
470 : //-----//
471 : // 移動平均バッファ取得(無ければ新規作成) //
472 : // //
473 : // 引 数 : hwnd - タイムチャートウインドのハンドル //
474 : // //
475 : // 戻り値 : ≠NULL - 成功 //
476 : // =NULL - 失敗 //
477 : // //
478 : //-----//
479 : static PBUFINFO GetAveBuf(HWND hwnd)
480 : {
481 :     PBUFINFO pW = NULL;
482 : // バッファ管理インスタンス取得
483 :     if ((pW = (PBUFINFO)GetProp(hwnd, TEXT("PROPBUF"))) == NULL) {
484 : // バッファ管理インスタンス無しならば割り当て
485 :         if (pW = (PBUFINFO)malloc(sizeof(BUFINFO))) {
486 :             memset(pW, 0, sizeof(BUFINFO));
487 :             // 移動平均バッファ確保
488 :             if (pW->pBuf = (PAVEDAT)malloc(sizeof(AVEDAT) * AveWidth)) {
489 :                 // バッファ管理インスタンスをウインドに登録
490 :                 SetProp(hwnd, TEXT("PROPBUF"), (HANDLE)pW);
491 :                 // バッファ情報初期化
492 :                 pW->num = AveWidth;
493 :                 pW->ix = 0;
494 :                 pW->cnt = 0;
495 :             }
496 :         }
497 :     }
498 :     else {
499 :         free(pW);
500 :         pW = NULL;
501 :     }
502 : }
503 : return pW;
504 : }
505 : //-----//
506 : // 移動平均バッファ割り当て //

```



```

507 : // //
508 : // 引数 : hwnd - タイムチャートウインドのハンドル //
509 : // //
510 : // 戻り値 : ≠NULL - 成功 //
511 : // =NULL - 失敗 //
512 : //-----//
513 : static PBUFINFO AllocAveBuf(HWND hwnd)
514 : {
515 :     PBUFINFO pW = NULL;
516 :
517 :     // バッファ管理インスタンス取得 (無ければ新規作成)
518 :     if ((pW = (PBUFINFO)GetProp(hwnd, PROPBIF)) == NULL) {
519 :         if (pW = (PBUFINFO)malloc(sizeof(BUFINFO))) {
520 :             memset(pW, 0, sizeof(BUFINFO));
521 :         }
522 :     }
523 :     // 移動平均バッファ作成
524 :     if (pW != NULL) {
525 :         // 旧バッファ解放
526 :         if (pW->pBuf != NULL) {
527 :             free(pW->pBuf);
528 :             pW->pBuf = NULL;
529 :         }
530 :         // 移動平均バッファ確保
531 :         if (pW->pBuf = (PAVEDAT)malloc(sizeof(AVEDAT) * AveWidth)) {
532 :             // バッファ管理インスタンスをウインドに登録
533 :             SetProp(hwnd, PROPBIF, (HANDLE)pW);
534 :             // チャート番号をウインドに登録
535 :             SetProp(hwnd, PROPSEQ, (HANDLE)((UX)AveWndSeq));
536 :             // バッファ情報初期化
537 :             pW->num = AveWidth;
538 :             pW->ix = 0;
539 :             pW->cnt = 0;
540 :         }
541 :         // 移動平均バッファ確保失敗・・・
542 :         else {
543 :             // バッファ管理インスタンスをウインドから削除
544 :             RemoveProp(hwnd, PROPBIF);
545 :             // バッファ管理インスタンス解放
546 :             free(pW);
547 :             pW = NULL;
548 :         }
549 :     }
550 :     return pW;
551 : }
552 : //-----//
553 : // 移動平均バッファ破棄 //
554 : // //
555 : // 引数 : hwnd - タイムチャートウインドのハンドル //
556 : // //
557 : // 戻り値 : TRUE - 成功 //
558 : // FALSE - 失敗 //
559 : //-----//
560 : static VOID FreeAveBuf(HWND hwnd)
561 : {
562 :     PBUFINFO pW;
563 :
564 :     if (pW = (PBUFINFO)GetProp(hwnd, PROPBIF)) { // 移動平均バッファ有り?
565 :         if (pW->pBuf != NULL) free(pW->pBuf); // 移動平均データバッファ解放
566 :         free(pW); // バッファ情報解放
567 :         RemoveProp(hwnd, PROPBIF); // ウインドプロパティ解除 (バッファ情報)
568 :         RemoveProp(hwnd, PROPSEQ); // ウインドプロパティ解除 (チャート番号)
569 :     }
570 : }

```

## 13. 問い合わせ先

本ソフトウェアに関するお問い合わせは、件名の先頭を「Ajara:」として、以下のメールアドレスに送付してください。

xxxajarakojara@kk. email. ne. jpxxx

[注] 先頭と末尾の「xxx」は削除してください。

「@」は、全角となっていますので、半角に訂正してください。

メールアドレスは変更される場合がありますので、以下の URL で確認してください。

<http://www.ne.jp/asahi/ajara/kojara/>

## 14. 変更履歴

Ver	変更内容	備考
1.6.0.0	<p>ユニコード版 初版発行</p> <p>バージョン 1.6.0.0 以降は、全面的に UNICODE をサポートしたバージョンです。</p> <p>UNICODE バージョンでは、多国間の文字が混在できます。 (例えば、"안녕하세요 (おはようございます)" のように日本語と韓国語を混在といった文字が扱えます)</p> <p>バージョン 1.6.0.0 より前の旧バージョンは ANSI バージョンです。</p> <p>ANSI バージョンでは、既定のコードページ(日本の場合シフトJIS)に限定されます。</p>	<p>UNICODE 版では、表面的(機能的)には大きな変更はありませんが、内部的には全面的な変更となるためバージョンを 1.6.0.0 としてリセットしました。</p>
1.6.1.0	受信データの「16進ダンプ表示」の表示形式を変更	