

**Model 701991
DL シリーズ用 MATLAB ツールキット
ユーザーズマニュアル**

ユーザー登録のお願い

今後の新製品情報を確実にお届けするために、お客様にユーザー登録をお願いしております。下記 URL の「製品のユーザー登録」のページで、ご登録いただけます。

<http://www.yokogawa.com/jp-ymi/tm/Bu/>

計測相談のご案内

当社では、お客様に正しい計測をしていただけるよう、当社計測器製品の仕様、機種を選定、および応用に関するご相談を下記カスタマサポートセンターにて承っております。なお、価格や納期などの販売に関する内容については、最寄りの営業、代理店にお問い合わせください。

横河計測株式会社 カスタマサポートセンター

フリーダイヤル
 0120-137046
tmi-cs@csv.yokogawa.co.jp

ファクシミリ
 FAX 0422-52-6013

【フリーダイヤル受付時間：祝祭日を除く月～金曜日の 9：00 ～ 12：00、13：00 ～ 17：00】

はじめに

このたびは、DL シリーズ用 MATLAB ツールキット (形名 701991) をご購入いただきましてありがとうございます。

このユーザーズマニュアルは、DL シリーズ用 MATLAB ツールキットの取り扱い上の注意 / 機能 / 操作方法などについて説明したものです。ご使用前にこのマニュアルをよくお読みいただき、正しくお使いください。

お読みになったあとは、ご使用時にすぐにご覧になれるところに、大切に保存してください。ご使用中に操作がわからなくなったときなどにきつとお役に立ちます。

なお、Windows, MATLAB、および当社製 DL シリーズの取り扱い上の注意 / 機能 / 操作方法などについては、それぞれの説明書をご覧ください。

ご注意

- 本書の内容は、性能 / 機能の向上などにより、将来予告なしに変更することがあります。また、実際の画面表示内容は、本書に記載の画面表示内容と多少異なることがあります。
- 本書の内容に関しては万全を期しておりますが、万一ご不審の点や誤りなどお気づきのことがありましたら、お手数ですが、当社支社 / 支店 / 営業所までご連絡ください。
- 本書の内容の全部または一部を無断で転載、複製することは禁止されています。
- 本ソフトウェアは、下記の DL シリーズに対応しています。

SL1000	DLM2000 シリーズ
DL6000/DLM6000 シリーズ	DL850 シリーズ
DLM4000 シリーズ	
- 本書は **MATLAB R2011b** 以降に対応しています。
- 本製品の TCP/IP ソフトウェア、および TCP/IP ソフトウェアに関するドキュメントは、カリフォルニア大学からライセンスされた BSD Networking Software、Release 1 をもとに当社で開発 / 作成したものです。

商標

- MATLAB は、米国 The MathWorks, Inc. の登録商標です。
- Microsoft、MS-DOS、および Windows は、米国 Microsoft Corporation の、米国およびその他の国における登録商標または商標です。
- Adobe および Acrobat は、アドビシステムズ社の米国およびその他の国における登録商標または商標です。
- 本文中の各社の登録商標または商標には、®、TM マークは表示していません。
- その他、本文中に使われている会社名 / 商品名は、各社の登録商標または商標です。

履歴

- 2004 年 7 月 初版発行
- 2004 年 9 月 2 版発行
- 2004 年 10 月 3 版発行
- 2006 年 2 月 4 版発行
- 2007 年 4 月 5 版発行
- 2010 年 10 月 6 版発行
- 2011 年 11 月 7 版発行
- 2013 年 3 月 8 版発行
- 2014 年 1 月 9 版発行
- 2016 年 6 月 10 版発行
- 2017 年 10 月 11 版発行

ソフトウェア使用許諾契約書

お客様が本ソフトウェア使用許諾契約書（以下、「本契約」といいます）に同意することを条件として、横河電機株式会社および横河メータ&インスツルメンツ株式会社（以下、「当社」といいます）は、包装されたソフトウェア製品（以下、「横河ソフトウェア」といいます）の使用権をお客様に許諾します。なお、当社は、横河ソフトウェアの使用権をお客様に許諾するものであり、横河ソフトウェアを販売するものではありません。

製品：701991 DL シリーズ用 MATLAB ツールキット
ライセンス数：1 ライセンス

第 1 条（適用範囲）

- 本契約は、当社がお客様に提供する横河ソフトウェア製品に適用するものとします。
- 横河ソフトウェアは、それに含まれる一切の技術、アルゴリズム、およびプロセスを包含するものとします。

第 2 条（使用権の許諾）

- お客様は、横河ソフトウェアについて、別途合意した使用料を対価として、前文に定めるライセンス数に対応する台数のコンピュータに限りインストールできるものとし、当社は、お客様の自己使用を目的とした、非独占的かつ譲渡不能の使用権（以下「使用権」といいます）を許諾します。
- お客様は、当社の事前の書面による承諾なしに、横河ソフトウェアを第三者に頒布、転貸、複製、譲渡、質入、伝送もしくは再使用権を許諾しないものとします。
- お客様は、バックアップ目的として一組のみ横河ソフトウェアを複製する以外は、横河ソフトウェアの全部または一部を複製しないものとします。また当該複製物の保管および管理については厳重な注意を払うものとします。
- お客様は、いかなる理由においても横河ソフトウェアをダンプ、逆アセンブル、逆コンパイル、リバースエンジニアリングなどによるソースプログラムその他人間が読み取り可能な形式への変換もしくは複製または横河ソフトウェアの修正もしくは他の言語への翻訳など、提供された形式以外に改変しないものとします。また、当社は、別に同意しない限り、お客様にソースプログラムを提供しないものとします。
- 横河ソフトウェアおよびそれらに含まれる一切の技術、アルゴリズム、およびプロセスなどのノウハウは、当社または当社に対し再使用許諾を含む使用許諾権を付与している第三者の固有財産であり、当社または当社に対し再使用許諾権を付与している第三者が権利を有しているものであり、お客様に権利の移転や譲渡を一切行うものではありません。
- 当社は、横河ソフトウェアに保護の機構（コピープロテクト）を使用または付加することがあります。当該コピープロテクトを除去したり、除去を試みることは認められないものとします。
- 横河ソフトウェアには、当社が第三者から再使用許諾を含む使用許諾権を付与されているソフトウェアプログラム（以下「第三者プログラム」といい、当社の関連会社が独自に製作・販売しているソフトウェアプログラムもこれに含みます）を含む場合があります。かかる第三者プログラムに関し、当社が当該第三者より本契約と異なる再使用許諾条件を受け入れている場合には、別途書面により通知される当該条件を遵守していただきます。

第 3 条（特定用途に関する制限）

- 横河ソフトウェアは、下記の各号を目的として、製作または頒布されるものではありません。
(a) 航空機の運行または船舶の航行や、これらを地上でサポートする機器の立案、設計、開発、保守、運用および使用されること。(b) 原子力施設の立案、設計、開発、建設、保守、運用および使用されること。(c) 核兵器、化学兵器または生物兵器の立案、設計、開発、保守、運用および使用されること。(d) 医療機器などの人身に直接関わるような状況下で使用されることを目的に立案、設計、開発、保守、運用および使用されること。
- お客様が前項の目的で横河ソフトウェアを使用する場合には、当社は当該使用により発生するいかなる請求および損害に対しても責任を負わないものとし、お客様は、お客様の責任においてこれを解決するものとし、当社を免責するものとします。

第 4 条（保証）

- 横河ソフトウェアは、当該製品完成時または出荷時の現状のままでお客様に提供されるものとし、お客様は、これに同意するものとします。横河ソフトウェアの記録媒体に破損、損傷が発見された場合は、開封後 7 日間に限り無償で交換をいたします（お客様の費用で当社の指定するサービス拠点に当該ソフトウェア製品の記憶媒体を送付していただくものとします）が、いかなる場合であっても横河ソフトウェアに瑕疵のないこと、的確性、正確性、信頼性もしくは最新性などの品質上または性能上の明示または黙示の保証をするものではありません。また、横河ソフトウェアが他のソフトウェアとの間で不整合、相互干渉などの影響のないことを保証するものではありません。
- 前項の規定に関わらず、横河ソフトウェアに第三者プログラムが存在する場合の保証期間、保証条件については、かかるプログラムの供給者の定めるところによるものとします。
- 当社は、自己の判断により必要と認めた場合、横河ソフトウェアに関するレビジョンアップおよびバージョンアップ（以下、アップデートサービスといいます）を実施することがあります。
- 前項の定めにも拘らず、当社は、いかなる場合であってもお客様により改変または修正された横河ソフトウェアに関するアップデートサービスについては、第三者により改変・修正された場合を含め、一切対応しないものとします。

第 5 条（特許権、著作権の侵害に関する損害賠償責任）

- お客様は、横河ソフトウェアについて、第三者から特許権、商標権、著作権その他の権利に基づき使用の差し止め、損害賠償請求などが行われた場合は、書面に速やかに請求の内容を当社に通知するものとします。
- 前項の請求などが当社の責に帰すべき事由による場合は、その防御および和解交渉について、お客様から当社に防御、交渉に必要なすべての権限を与えていただき、かつ必要な情報および援助をいただくことを条件に、当社は自己の費用負担で当該請求などの防御および交渉を行い、前項記載の第三者に対して最終的に認められた責任を負うものとします。
- 当社は第 1 項における請求またはその恐れがあると判断した場合は、当社の選択により、当社の費用で下記のいずれかの処置を取るものとします。
(a) 正当な権利を有する者からかかる横河ソフトウェアの使用を継続する権利を取得する。(b) 第三者の権利の侵害を回避できるようなソフトウェア製品と交換する。
(c) 第三者の権利を侵害しないようにかかる横河ソフトウェアを改造する。
- 前項各号の処置がとれない場合、当社は、お客様から当社にお支払い頂いた第 2 条第 1 項に定める使用料の対価を限度として損害を賠償するものとします。

第 6 条（責任の制限）

本契約に基づいて当社がお客様に提供した横河ソフトウェアによって、当社の責に帰すべき事由によりお客様が損害を被った場合は、当社は、本契約の規定に従って対応するものとなりますが、いかなる場合においても、派生損害、結果損害、その他の間接損害（営業上の利益の損失、業務の中断、営業情報の喪失などによる損害その他）については一切責任を負わないものとし、かつ当社の損害賠償責任は、かかる横河ソフトウェアについてお客様からお支払いを受けた第 2 条第 1 項に定める使用料の対価を限度とします。なお、当社が納入した製品をお客様が当社の書面による事前の承諾なく改造、改変、他のソフトウェアとの結合を行い、またはその他基本仕様書または機能仕様書との相違を生ぜしめた場合は、当社は一部または全ての責任を免れることができるものとします。

第 7 条（輸出規制）

お客様は、事前に当社の同意を得た場合を除き、横河ソフトウェアを、直接、間接を問わず輸出または他国に伝送しないものとします。

第 8 条（本契約の期間）

本契約は、お客様が横河ソフトウェアを受領した日から、契約解除されない限り、お客様または当社が相手方に対し、1 ヶ月前に書面による通知によって当該ソフトウェア製品の使用を終了させるまで、またはお客様の横河ソフトウェアの使用終了時まで、有効とします。

第 9 条（使用の差止め）

横河ソフトウェアの使用許諾後とはいえども、使用環境の変化または許諾時には見出せなかった不適切な環境条件が見られる場合、その他横河ソフトウェアを使用するに著しく不適切であると当社が判断した場合には、当社はお客様に対して当該使用を差止めることができるものとします。

第 10 条（解除）

当社は、お客様が本契約に違反した場合には、何ら催告を要することなく通知をもって本契約を解除できます。ただし、本契約終了または解除後とはいえども第 5 条、第 6 条ならびに第 11 条は効力を有するものとします。

第 11 条（管轄裁判所）

本契約に関して生じた紛争、疑義については、両者誠意を持って協議解決するものとします。ただし、一方当事者が他方当事者に協議解決をしたい旨の通知後 90 日以内に両当事者間で協議が整わない場合は東京地方裁判所（本庁）を第一審の専属的管轄裁判所とします。

以上

目次

	ソフトウェア使用許諾契約書	ii
	製品概要	iv
	動作に必要なシステム環境	vi
	ご使用にあたっての注意	viii
第 1 章	本ソフトウェアを使用するための準備	
	1.1 DL シリーズ用 MATLAB ツールキットのインストール	1-1
	1.2 MATLAB の初期設定	1-2
第 2 章	MATLAB から DL をコントロールするための準備	
	2.1 DL シリーズの通信インターフェース設定	2-1
	2.2 DL コントロール用 MEX-Function の実行	2-2
	2.3 DL コントロール用 MEX-Function 使用例 (サンプルプログラム)	2-4
第 3 章	WDF ファイルの波形データを MATLAB に取り込む方法	
	3.1 ファイル用 MEX-Functions を使用するには	3-1
	3.2 ファイル用 MEX-Function の実行例	3-2
	3.3 サンプルプログラム	3-3
第 4 章	関数一覧	
	4.1 DL コントロール用 MEX-Functions	4-1
	4.2 WDF ファイルアクセス用 MEX-Functions	4-16
	索引	

製品概要

本ソフトウェアは DL コントロール用 MEX-Functions、WDF ファイル用 MEX-Functions、DL シリーズライブラリから構成されています。

DL コントロール用 MEX-Functions は、当社製品デジタルオシロスコープ (以降、DL 本体と省略します) をコントロールしたり、データ取得するための関数群です。MATLAB 上から DL 本体の測定レンジなどを変更したり、DL 本体のデータを MATLAB の行列に取り込むことができます。

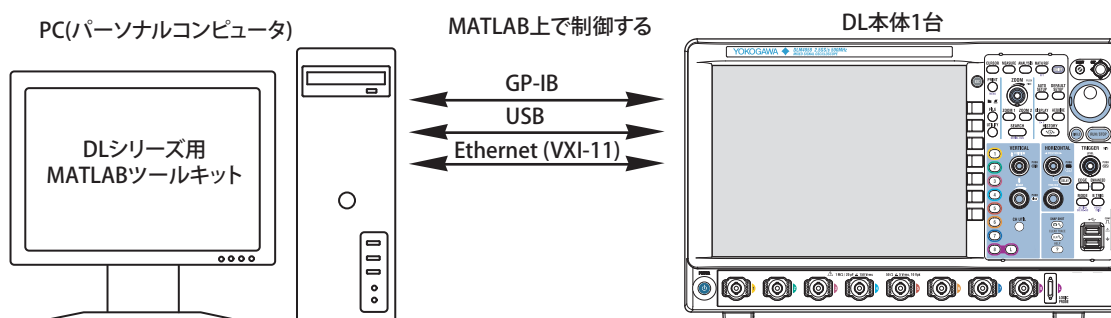
DL シリーズライブラリは、上記 DL コントロール用 MEX-Functions で DL 本体にアクセスするときに使用されます。

WDF ファイル用 MEX-Functions は、DL 本体で保存された波形データファイル (拡張子: wdf) に MATLAB 環境からアクセスするための関数群です。

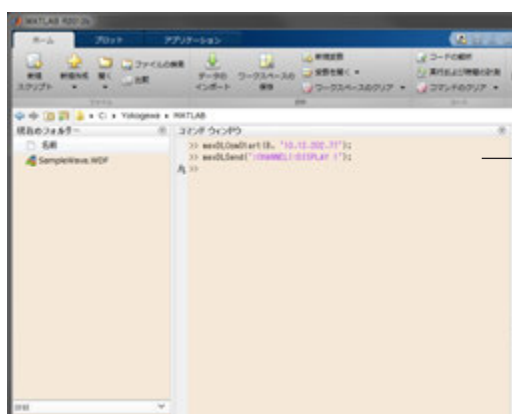
Note

本ソフトウェアでは、DL 本体のロジック入力で捕捉したデータは取り込みません。

本ソフトウェアは、PC と当社製品 DL 本体を接続し、MATLAB 上で DL コントロール用 MEX-Functions や DL 専用通信コマンドを使って DL 本体をコントロールします。



MATLAB上でDLをコントロールする画面



DLコントロール用MEX-FunctionsやDL専用通信コマンドを使ってDL本体をコントロールする。または波形データやヒストリデータを取得する。

Figure 1 shows the MATLAB R2012b interface. The top menu bar includes 'ホーム', 'プロット', and 'アプリケーション'. The 'アプリケーション' menu is open, showing options like '新規 スクリプト', '新規作成', '開く', '比較', 'ファイルの検索', 'データのインポート', 'ワークスペースの保存', '新規変数', '変数を開く', and 'ワークスペースのクリア'. The main workspace area shows the file path 'C:\Yokogawa\MATLAB' and a list of files. The 'コマンド ウィンドウ' (Command Window) is visible on the right.

DL 専用通信コマンドを使って DL 本体をコントロールする

例：構文 `>mexDLSend(Msg);` の場合、
`>mexDLSend('CHANNEL1:DISPLAY 1')`

動作に必要なシステム環境

OS

Windows 7(32 ビット版、64 ビット版)
Windows 8(32 ビット版、64 ビット版)
Windows 8.1(32 ビット版、64 ビット版)
Windows 10(32 ビット版、64 ビット版)

CPU

MATLAB の動作環境に準ずる

メモリ

MATLAB の動作環境に準ずる

CRT

MATLAB の動作環境に準ずる

Color

MATLAB の動作環境に準ずる

通信インタフェース

GP-IB、Ethernet(VXI-11)、USB のいずれかの通信機能

コントロール可能な DL シリーズ

SL1000 DLM2000 シリーズ
DL6000/DLM6000 シリーズ DL850 シリーズ
DLM4000 シリーズ

MATLAB

R2011b 以降

DL 側で必要な通信機能 (以下のいずれか)

GP-IB

当社 GP-IB 搭載 IEEE St'd 488.2 準拠製品

Note

当社製品で通信を行なう場合、ターミネータの設定は、通常時は、LF および EOI、バイナリデータ転送時は EOI をご使用ください。

USB

当社製デジタルオシロスコープ、SL1000、DLM2000 シリーズ、DL6000/DLM6000 シリーズ、DL850 シリーズ、DLM4000 シリーズで、USB インターフェースが搭載されているもの。

Ethernet

SL1000、DLM2000 シリーズ、DL6000/DLM6000 シリーズ、DL850 シリーズ、DLM4000 シリーズで、Ethernet インターフェースが搭載されているもの。

その他

CD-ROM ドライブ (インストール用)

ご使用にあたっての注意

- 本ソフトウェア使用中は、DL シリーズ本体での操作はしないでください。誤動作の原因になります。
- PC に装備されているスタンバイモードに入ると、本ソフトウェアの動作を継続できなくなるものがあります。
スタンバイモードをオフにしてご使用ください。
- イーサネットインターフェースを使って、本ソフトウェアを実行した場合、回線負荷は最大で 800K バイト /s、通常で 400K バイト /s 以下です。
ご使用にあたっては、ネットワーク管理者にご相談ください。
- 本ソフトウェアで、DL シリーズ本体のネットワークや通信の設定をしないでください。接続が切断される恐れがあります。
- 本ソフトウェアで、セルフテストを実行しないでください。
- 本ソフトウェアでコントロールできる DL シリーズは、1 台だけです。また、複数の PC から同じ DL シリーズに対して、同時には接続できません。
- 1 つの PC で本ソフトウェアを複数起動して、複数の DL シリーズをコントロールできますが、PC の仕様や回線の状態によっては、動作が遅くなることがあります。また、本ソフトウェアを複数起動した場合、PC の CPU やメモリサイズによっては、正しく動作しなくなることがあります。
- DL に接続中に接続エラーが出て接続できないときは、DL 本体の電源スイッチを OFF/ON してください。

1.1 DL シリーズ用 MATLAB ツールキットのインストール

DL シリーズ用 MATLAB ツールキットのインストールについては、本ソフトウェアに同梱されている「インストールの前にお読みください (DL シリーズ用 MATLAB ツールキット)」(IM701991-71) をご覧ください。

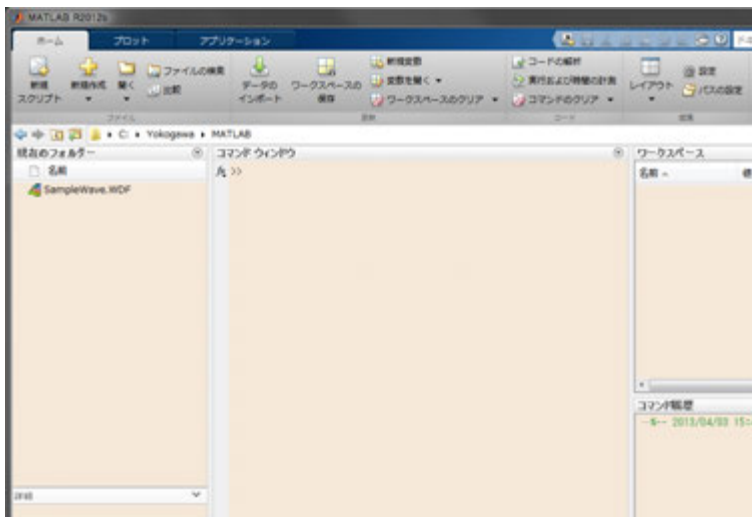
Note

本ソフトウェアをインストールすると、DL コントロール用 MEX-Functions(4.1 節参照) と WDF ファイルアクセス用 MEX-Functions(4.2 節 /4.3 節参照) も同時にインストールされます。

1.2 MATLAB の初期設定

以下の操作は、本ソフトウェアをインストールしたあと、一度だけ操作します。

1. MATLAB を起動します。



MATLAB のパスの設定

2. ツールバーのパスの設定をクリックします。「パスを設定」ダイアログボックスが表示されます。



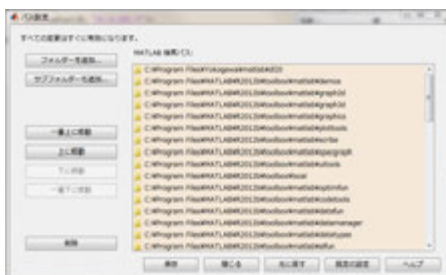
3. フォルダを追加ボタンをクリックしたあと、DL シリーズ用 MATLAB ツールキットをインストールしたフォルダを選択します。

Note

デフォルトのインストール先は、下記のフォルダです。

C:\Program Files\Yokogawa\matlab\dl20

4. 保存ボタンをクリックしたあと、閉じるボタンをクリックします。



2.1 DL シリーズの通信インタフェース設定

本ソフトウェアをインストールすると、DL コントロール用 MEX-Functions がソフトウェアと同時にインストールされます。以下の設定を行うことで、この DL コントロール用 MEX-Functions を使って DL シリーズをコントロールできます。

使用するインタフェースの設定は DL 本体で行ってください。

DLM2000 シリーズ、DL6000/DLM6000 シリーズ、 DL850 シリーズ、DLM4000 シリーズの場合	UTILITY > Remote Control > Device
---	-----------------------------------

USB で接続する場合

DL シリーズと USB 接続する場合には、専用の USB ドライバが必要です。USB ドライバは、MATLAB ツールキットのインストールディスクに入っています。

最新の USB ドライバは、下記のホームページからダウンロードできます。

<http://www.yokogawa.com/jp-ymi/>

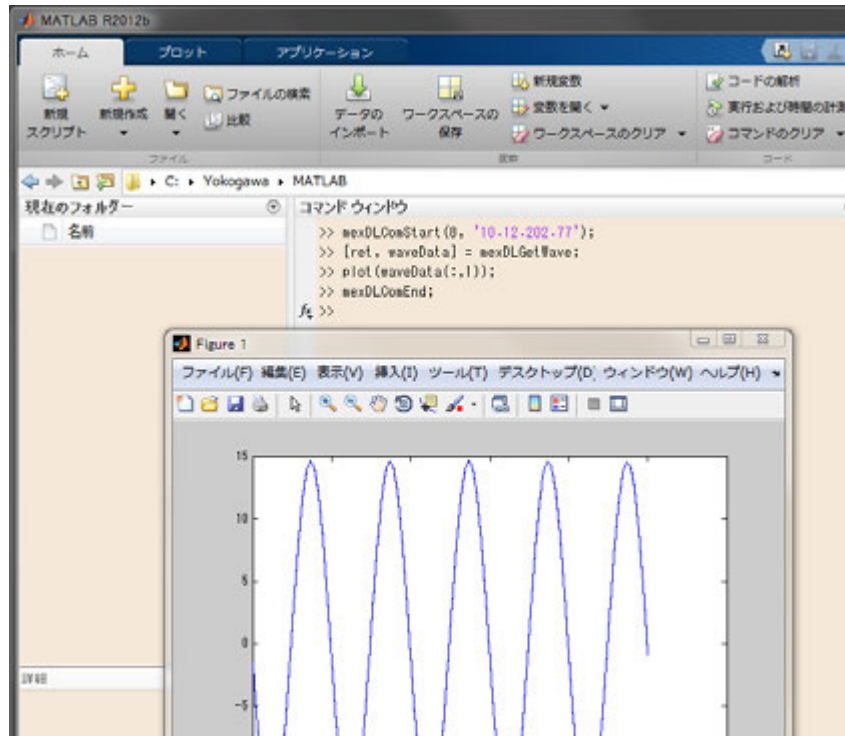
USB ドライバのインストール

YKMUSB フォルダ内の Setup.exe を実行してください。インストールウィザードが起動します。インストール方法の詳細については、YKMUSB フォルダ内のマニュアル (IM B9852UT-01) をご覧ください。

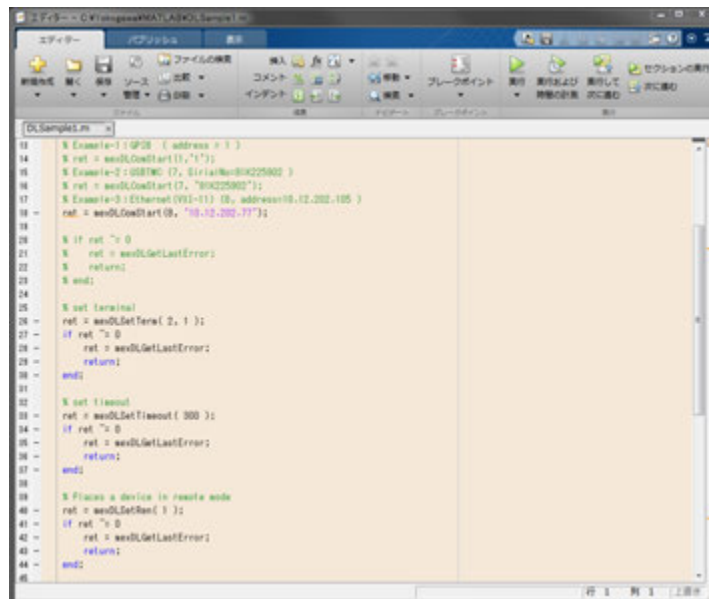
2.2 DL コントロール用 MEX-Function の実行

DL コントロール用 MEX-Functions は、MATLAB の特長である対話形式を使ってプログラミングします。MATLAB のコマンドウィンドウおよび m ファイルでプログラムを実行することで、DL をコントロールしたり、波形データを MATLAB に取り込むことができます。

コマンドウィンドウ



m ファイルエディタ



コマンドウィンドウでの実行例

```
>> ret = mexDLComStart(8,'10.0.236.100');  
>> ret = mexDLSend( 'STOP' );  
>> ret = mexDLSend( 'CHANNEL:VDIV 500mV' );  
>> ret = mexDLSend( 'ACQUIRE:MODE NORMAL; RLENGTH 1000' );  
>> ret = mexDLComEnd;
```

2.3 DL コントロール用 MEX-Function 使用例 (サンプルプログラム)

各サンプルプログラムは、DL シリーズ用 MATLAB ツールキットをインストールしたフォルダにあります。

サンプルプログラム 1

以下は、通信回線を接続し、コマンドを送信するプログラムです。

接続する通信方法を選択 (% をはずし) し、ご使用ください。

サンプルプログラムは、DLSample1.m です。

使用例 **[ret, Buf, Size] = DLSample1**

```
function [ret, Buf, Size] = DLSample1()

% DL Sample 1
%
% Basic communication verification.
% Open communication > inquire ID > close communication
%
% Choose an interface type by removing '%' in front of ret
% from the following lines ( between line no. 13 and 18 )
% You may need to adjust parameters of the interface.

% Example-1: GPIB ( address = 1 )
% ret = mexDLComStart(1,'1');
% Example-2: USBTMC (7, SerialNo=91K225902 )
% ret = mexDLComStart(7, '91K225902');
% Example-3: Ethernet (VXI-11) (8, address=10.12.202.105 )
ret = mexDLComStart(8, '10.12.202.105');

% if ret ~= 0
%     ret = mexDLGetLastError;
%     return;
% end;

% set terminal
ret = mexDLSetTerm( 2, 1 );
if ret ~= 0
    ret = mexDLGetLastError;
    return;
end;

% set timeout
ret = mexDLSetTimeout( 300 );
if ret ~= 0
    ret = mexDLGetLastError;
    return;
end;

% Places a device in remote mode
ret = mexDLSetRen( 1 );
if ret ~= 0
    ret = mexDLGetLastError;
    return;
end;

% sending IDN? & receiving query
ret = mexDLSend( '*IDN?' );
if ret ~= 0
    ret = mexDLGetLastError;
    return;
end;

[ret, Buf, Size] = mexDLReceive( 1000 );
if ret ~= 0
    ret = mexDLGetLastError;
    return;
end;

% Places a device in local mode
ret = mexDLSetRen( 0 );
if ret ~= 0
    ret = mexDLGetLastError;
    return;
end;

Ret = mexDLComEnd;
if ret ~= 0
    ret = mexDLGetLastError;
    return;
end;
```

サンプルプログラム 2

以下は、通信回線を接続し、トリガーが掛かったら波形データを取得し、plot 関数でその波形を表示するサンプルです。

接続する通信方法を選択 (% をはずし) し、ご使用ください。

サンプルプログラムは、DLSample2.m です。

```

使用例          [ ret, WaveData ] = DLSample2

function [ ret, WaveData ] = DLSample2()

%
% DL Sample 2
%
% Recieve data as soon as triggering conditions are met.
% Captured data is stored into 'WaveData' matrix.
% Open communication > start signal acquisition >
% transfer DL data to MATLAB matrix > close communication
%
% Choose an interface type by removing '%' in front of ret
% from the following lines ( between line no. 15 and 20 )
% You may need to adjust parameters of the interface.

% Example-1: GPIB ( address = 1 )
% ret = mexDLComStart(1, '1');
% Example-2: USBTMC (7, SerialNo=91K225902 )
% ret = mexDLComStart(7, '91K225902');
% Example-3: Ethernet (VXI-11) (8, address=10.12.202.105 )
ret = mexDLComStart(8, '10.12.202.105');

% if ret ~= 0
%     ret = mexDLGetLastError;
% return;
% end;

% set terminal
ret = mexDLSetTerm( 2, 1 );
if ret ~= 0
    ret = mexDLGetLastError;
    return;
end;

% set timeout
%ret = mexDLSetTimeout( 300 );
ret = mexDLSetTimeout( 3000 );
if ret ~= 0
    ret = mexDLGetLastError;
    return;
end;

WaveData = 0;

% Start signal acquisition
Ret = mexDLSend('sstart? 100');
if Ret ~= 0
    Ret = mexDLGetLastError;
    return;
end;

[Ret, Buf, Size] = mexDLReceive( 10 );
if Ret ~= 0
    Ret = mexDLGetLastError;
    return;
end;

% Receive DL data into WaveData matrix if a
% signal is triggered
if strcmp( deblank( Buf(1,:)), ':SST 0' ) == 1
    [Ret, WaveData] = mexDLGetWave;
    plot(WaveData);
    if Ret ~= 0
        Ret = mexDLGetLastError;
        return;
    end;
end;

% Set DL trigger mode to AUTO
mexDLSend(':TRIG:MODE AUTO');

%Close communication port
Ret = mexDLComEnd;
if ret ~= 0
    ret = mexDLGetLastError;
    return;
end;

```

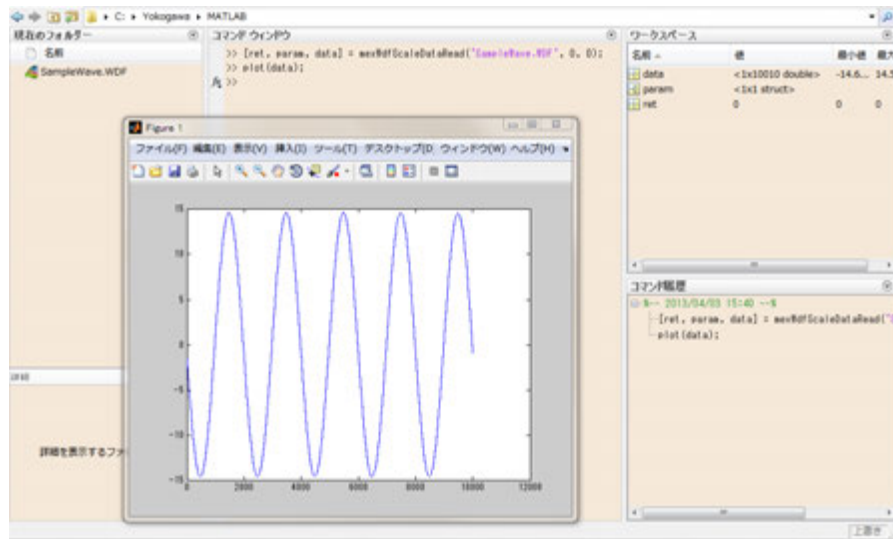
3.1 ファイル用 MEX-Functions を使用するには

本ソフトウェアをインストールすると、DL コントロール用 MEX-Functions と同時に、同じ場所に WDF ファイル用 MEX-Functions もインストールされます。

3.2 ファイル用 MEX-Function の実行例

WDF ファイル用 MEX-Functions は、MATLAB の特長である対話形式を使ってプログラミングします。MATLAB のコマンドウィンドウおよび m ファイル上での WDF ファイル用 MEX-Functions の使用例は次のとおりです。

MATLAB の画面



コマンドウィンドウでの実行例

```
>> [ret, param, data] = mexWdfScaleDataRead('SampleWave.WDF', 0, 0);  
>> plot(data);  
>>
```

3.3 サンプルプログラム

WDF サンプルプログラム

以下は、WDF ファイルのヘッダー情報を取得してから、波形データを plot 関数で表示するサンプルです。

サンプルプログラムは、WdfSample1.m ～ WdfSample4.m です。

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  MATLAB M-File : WdfSample4.m
%  WDF file access sample script- 4
%
%  Copyright (C) Yokogawa Test & Measurement Corporation
%  Software Japan. All rights reserved.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
%   Input the Filename
%
filename = input( 'filename = ', 's' );
[ ret, chNum ] = mexWdfGetChNum( filename );
block = 0;

for ch = 0 : double( chNum ) - 1

    %
    %   Get file parameters
    %
    [ ret, traceName ] = mexWdfItemRead ( filename, 'TraceName', ch, 0 );
    [ ret, vScaleUpper ] = mexWdfItemRead ( filename, 'VScaleUpper', ch, block );
    [ ret, vScaleLower ] = mexWdfItemRead ( filename, 'VScaleLower', ch, block );
    [ ret, hResolution ] = mexWdfItemRead ( filename, 'HResolution', ch, block );
    [ ret, hOffset ] = mexWdfItemRead ( filename, 'HOffset', ch, block );
    [ ret, vUnit ] = mexWdfItemRead ( filename, 'VUnit', ch, block );
    [ ret, hUnit ] = mexWdfItemRead ( filename, 'HUnit', ch, block );

    %
    %   Get waveform data
    %
    clear data x;
    [ ret, param, data ] = mexWdfScaleDataRead( filename, ch, block );
    x( 1 : param.cntOut ) = [ hOffset : hResolution : hResolution * ( param.cntOut -
1 ) + hOffset ];

    %
    %   Display waveform data to a graph
    %
    subplot( double( chNum ), 1, ch + 1);
    plot( x, data );
    title( traceName );
    axis( [ x( 1 ) x( param.cntOut ) vScaleLower vScaleUpper] );
    ylabel( strcat( 'Amplitude [', vUnit, ']' ) );

end
xlabel( strcat( 'Time [', hUnit, ']' ) );

```

4.1 DL コントロール用 MEX-Functions

MATLAB の「Command Window」に DL コントロール用 MEX-Functions や DL 専用通信コマンドを入力すると、DL 本体をコントロールしたり、波形データを取得することが可能です。

DL コントロール用 MEX-Function 一覧

mex 関数名	機能	ページ
[ret] = mexDLComStart(wire, Adr);	回線を初期化し、指定されたデバイスとの回線を接続します。	4-3
[ret] = mexDLDeviceClear();	選択されたデバイスのクリア (SDC) を実行します。GP-IB 専用命令です。	4-4
[ret] = mexDLSend(Msg);	デバイスへメッセージを送信します。	4-4
[ret] = mexDLSendByLength(Msg, len);	デバイスへメッセージを指定されたバイト数送信します。	4-5
[ret] = mexDLSendSetup();	デバイスへメッセージを送信する準備をします。	4-5
[ret] = mexDLSendOnly(Msg, len, end);	デバイスへメッセージを指定されたバイト数送信します。	4-5
[ret,buf,size] = mexDLReceive(blen);	デバイスからメッセージを受信します。	4-6
[ret] = mexDLReceiveSetup();	デバイスからメッセージを受信する準備をします。	4-6
[ret,buf,size] = mexDLReceiveOnly(blen);	デバイスから (受信準備後の) メッセージを受信します。	4-7
[ret,length] = mexDLReceiveBlockHeader();	デバイスから送られてくる Block data のヘッダ部分を受信し、そのあとに続くデータバイト数を返します。	4-7
[ret,buf,size,endFlag] = mexDLReceiveBlockData(blen);	デバイスから送られてくる Block Data のデータ部分を受信します。	4-8
[endFlag] = mexDLCheckEnd();	デバイスからのメッセージが終了かどうかを返します。GP-IB、USB、Ethernet インタフェースで使用できます。	4-8
[ret] = mexDLSetRen(flag);	デバイスをリモート / ローカル状態にします。GP-IB 以外での使用は、当社製品に限定されます。	4-9
[errorID] = mexDLGetLastError();	最後に MATLAB のコマンドで発生したエラーのエラー番号を返します。	4-10
[ret] = mexDLSetTerm(eos, eot);	メッセージの送受信におけるターミネータを設定します。	4-11
[ret] = mexDLSetTimeout(tmo);	通信のタイムアウト時間を設定します。	4-11
[ret,WaveData,Time] = mexDLGetWave();	波形データを取得します。	4-12
[ret,WaveData,Time] = mexDLGetWave(traceNum, subtraceNum, waveStart, waveEnd);	トレース番号、波形の範囲を指定し、波形データを取得します。	4-13
[ret,WaveData,Time] = mexDLGetWave(traceNum, subTraceNum, waveStart, waveEnd);	DL850 シリーズのサブチャネルデータ取得専用の関数です。サブトレース (チャネル) 番号、サブトレース (サブチャネル) 番号、波形の範囲を指定し、波形データを取得します。	4-14
[ret,HistoryWave,Time] = mexDLGetHistoryWave(rec);	ヒストリ波形データを取得します。	4-15
[ret,HistoryWave,Time] = mexDLGetHistoryWave(rec, traceNum, Sub traceNum, waveStart, waveEnd);	レコード番号、トレース番号、波形の範囲を指定し、ヒストリ波形データを取得します。	4-15
[ret,HistoryWave,Time] = mexDLGetHistoryWave(rec, traceNum, subTraceNum, waveStart, waveEnd);	DL850 シリーズのサブチャネルデータ取得専用の関数です。サブトレース (チャネル) 番号、サブトレース (サブチャネル) 番号、波形の範囲を指定し、ヒストリ波形データを取得します。	4-17
[ret] = mexDLComEnd();	DL 本体のコントロール画面を起動します。	4-18
mexDLToolkit;	本ソフトウェアのバージョン情報を表示します。	4-18

4.1 DL コントロール用 MEX-Functions

[ret] = mexDLComStart(wire, Adr);

機能 : 回線を初期化し、指定されたデバイスとの回線を接続します。

引数 : wire 回線の種類
Adr 回線固有のアドレスのキャラクタ配列

戻り値 : ret (0 = OK, 1 = ERROR)

詳細 : 引数説明

wire 制御対象デバイスが繋がっている回線の種類を指定します。
各インターフェースの設定値は、以下のとおりです。

GP-IB : wire = 1
USB-TMC : wire = 7
VXI-11 : wire = 8

Adr 制御対象デバイスのアドレスを文字列として設定します。
インターフェースによる設定の仕方は、以下のとおりです。

GP-IB : Adr = "1" ~ "30" (デバイスの GP-IB アドレス値)
USB-TMC : Adr = "シリアル番号 "
シリアル番号は、機器のシリアル番号
ETHERNET (VXI-11) : Adr = "IP アドレス "

使用例 : ret = mexDLComStart(1, '1');
ret = mexDLComStart(7, '27E000001');
ret = mexDLComStart(8, '10.0.222.111');

[ret] = mexDLDeviceClear();

機能 : 選択されたデバイスのクリア (SDC) を実行します。GP-IB 専用命令です。

引数 : なし

戻り値 : ret (0 = OK, 1 = ERROR)

詳細 : この関数は、GP-IB に繋がっているデバイス専用で、他のインターフェース経由のデバイスには、何も行ないません。
USB-TMC では、USB-TMC 規格の InitiateClear を発行します。InitiateClear に成功すると、ClearFeature を発行して処理を終了します。

使用例 : ret = mexDLDeviceClear;

[ret] = mexDLSend(Msg);

機能 : デバイスへメッセージを送信します。

引数 : Msg DL 専用通信コマンドのキャラクタ配列

戻り値 : ret (0 = OK, 1 = ERROR)

詳細 : 引数説明

Msg DL 専用通信コマンド文字列を設定します。
1 つの送信 DL 専用通信コマンドを分割して送信する場合は、
"mexDLSendSetup"、"mexDLSendOnly" を使用してください。

使用例 : ret = mexDLSend(':CANNEL1:DISPLAY 1');
ret = mexDLSend('start');
ret = mexDLSend('stop');

[ret] = mexDLSendByLength(Msg, len);

機能 : デバイスへメッセージを指定されたバイト数送信します。

引数 : Msg DL 専用通信コマンドのキャラクタ配列
len DL 専用通信コマンドの送信するバイト数

戻り値 : ret (0 = OK, 1 = ERROR)

詳細 : 引数説明
Msg DL 専用通信コマンドを設定します。
len DL 専用通信コマンドの送信バイト数を設定します。
DL 専用通信コマンドに、Binary データを含む場合でも、送信できます。
また、1つの送信メッセージを分割して送信する場合は、
"mexDLSendSetup"、"mexDLSendOnly" を使用してください。

使用例 : ret = mexDLSendByLength(':CHANNEL1:DISPLAY 1', 19);

[ret] = mexDLSendSetup();

機能 : デバイスへメッセージを送信する準備をします。

引数 : なし

戻り値 : ret (0 = OK, 1 = ERROR)

詳細 : デバイスへメッセージを送信する準備をします。
1メッセージを数回に分けて送信するときに送信前に1回実行します。
実際のメッセージの送信は、"mexDLSendOnly" を使用します。

使用例 : ret = mexDLSendSetup;

[ret] = mexDLSendOnly(Msg, len, end);

機能 : デバイスへメッセージを指定されたバイト数送信します。

引数 : Msg DL 専用通信コマンドのキャラクタ配列
len DL 専用通信コマンドの送信するバイト数
end 終了フラグ

戻り値 : ret (0 = OK, 1 = ERROR)

詳細 : 引数説明
Msg メッセージを設定します。
len メッセージの送信バイト数を設定します。
end 今回の送信で送信終了かどうかを設定します。
終了の場合は1を、まだ送信を続ける場合は0を設定します。
指定されたデバイスへDL 専用通信コマンドを送信します。
DL 専用通信コマンドに、Binary データを含む場合でも、送信できます。
終了フラグを1に設定して送信したときのみ、ターミネータをDL 専用通信コマンドの最後に送信します。
そのため、終了フラグが0のうちは、デバイス側は一連のメッセージと判断します。

使用例 : ret = mexDLSendOnly(':CANNEL1:DISPLAY 1', 80,0);

4.1 DL コントロール用 MEX-Functions

[ret, buf, size] = mexDLReceive(blen);

機能 : デバイスから、メッセージを受信します。

引数 : blen 受信サイズ (バイト単位)

戻り値 : ret (0 = OK, 1 = ERROR)

buf 受信データ用バッファ

size 実受信バイト数

詳細 : 引数説明

blen メッセージの最大受信バイト数 (通常はバッファのバイト数) を設定します。

戻り値説明

buf 受信メッセージを格納するバッファを設定します。

size 実際に受信したバイト数を返します。

デバイスからメッセージを受信します。ターミネータを検出した場合は、そこまでのデータを、検出なかった場合は、blen で指定されたバイト数までのデータを受信します。当社デジタルオシロスコープとの通信で、"WAVEform:SEND?"、"IMAGe:SEND?" 等のメッセージのデータを受信する場合は、"mexDLReceiveBlockHeader"、"mexDLReceiveBlockData" を使用してください。

使用例 : [ret,buf,size] = mexDLReceive(80);

[ret] = mexDLReceiveSetup();

機能 : デバイスから、メッセージを受信する準備をします。

引数 : なし

戻り値 : ret (0 = OK, 1 = ERROR)

詳細 : デバイスから、大量データを分割して受信する場合に、受信準備をするために実行します。実際のデータは、"mexDLReceiveOnly" を使用して受信します。

使用例 : ret = mexDLReceiveSetup;

[ret, buf, size] = mexDLReceiveOnly(blen);	
機能	デバイスから、(受信準備後の)メッセージを受信します。
引数	blen 受信サイズ (バイト単位)
戻り値	ret (0 = OK, 1 = ERROR) buf 受信データ用バッファ size 実受信バイト数
詳細	引数説明 blen メッセージの最大受信バイト数 (通常はバッファのバイト数) を設定します。 戻り値説明 buf 受信メッセージを格納するバッファを設定します。 size 実際に受信したバイト数を返します。 大量データを分割して受信する場合に使用します。 "mexDLReceiveSetup" で受信準備後に、指定されたデバイスからメッセージを受信します。 ターミネータを検出した場合は、そこまでのデータを、検出しなかった場合は、Blen で指定されたバイト数までのデータを受信します。
使用例	[ret,buf,size] = mexDLReceiveOnly(80);

[ret, length] = mexDLReceiveBlockHeader();	
機能	デバイスから送られてくる Block Data のヘッダ部分を受信し、そのあとに続くデータバイト数を返します。
引数	なし
戻り値	ret (0 = OK, 1 = ERROR) length Block Data のデータバイト数
詳細	戻り値説明 length Block Data のバイト数を返します。 Block Data を受信するときに最初に使用します。 length に、あとに続くデータバイト数が返ってきますので、そのバイト数+1 (ターミネータ) 分を "mexDlReceiveBlockData" を使用してデータを受信します。
使用例	[ret,length] = mexDLReceiveBlockHeader;

4.1 DL コントロール用 MEX-Functions

[ret, buf, size, endFlag] = mexDLReceiveBlockData(blen);

機能 : デバイスから送られてくる Block Data のデータ部分を受信します。

引数 : blen 受信サイズ (バイト単位)

戻り値 : ret (0 = OK, 1 = ERROR)

buf 受信データ用バッファ

size 実受信バイト数

endFlag 終了フラグ

詳細 : 引数説明

blen メッセージの最大受信バイト数 (通常はバッファのバイト数) を設定します。

戻り値説明

buf 受信メッセージを格納するバッファを設定します。

size 実際に受信したバイト数を返します。

endFlag "mexDLReceiveBlockHeader" で受信したデータバイト数をすべて受信し終わったかどうかを返します。

終わっている場合は 1 を、まだ残っている場合は 0 を返します。

ブロックデータ (# ~ で始まるメッセージ) を受信する場合に使用します。

"mexDLReceiveBlockHeader" で受信準備後に、指定されたデバイスからメッセージを受信します。

ターミネータを検出した場合は、そこまでのデータを、検出しなかった場合は、blen で指定されたバイト数までのデータを受信します。

使用例 : [ret,buf,size,endFlag] = mexDLReceiveBlockData(80);

[endFlag] = mexDLCheckEnd();

機能 : デバイスからのメッセージが終了したかどうかを返します。GP-IB、USB、Ethernet インターフェイスで使用できます。

引数 : なし

戻り値 : end 1 = メッセージあり、0 = メッセージ終了

詳細 : 一連の受信メッセージを分割して受信したとき、"mexDLReceiveOnly" ですべて受信し終わったかどうかを返します。

使用例 : endFlag = mexDLCheckEnd;

[ret] = mexDLSetRen(flag);

機能 : デバイスをリモート / ローカル状態にします。GP-IB 以外での使用は、当社製品に限定されます。

引数 : flag リモート (1) / ローカル (0)

戻り値 : ret (0 = OK, 1 = ERROR)

詳細 : 引数説明

flag リモートにする場合は 1、ローカルにする場合は 0 を送ります。
 インターフェースの種類によって、動作が若干異なります。
 GP-IB の場合は、REN ラインを TRUE/FALSE にします。
 そのため、実際にリモートにする場合は、そのあと、そのデバイスに対して、なんらかのメッセージを送信した場合に、初めてリモートとなります。(デバイス個別のリモート / ローカル操作は行ないません。)
 USB、Ethernet の場合は、当社 488.2 準拠品で、通信メッセージに COMMunicate グループをサポートしているものに限り、この場合は、デバイス個別に操作できます。

使用例 : ret = mexDLSetRen(1);

[errorID] = mexDLGetLastError();

機能 : 最後に MATLAB のコマンドで発生したエラーのエラー番号を返します (DL 側のエラーコードではありません)。

引数 : int id デバイス ID 値

戻り値 : エラー番号

詳細 : 戻り値説明

ErrorID そのデバイスの最後に発生したエラー番号を返します。

0x00000000(0)	No error
0x00000001(1)	Timeout
0x00000002(2)	Device Not Found
0x00000004(4)	Open Port Error
0x00000008(8)	Device Not Open
0x00000010(16)	Device Already Open
0x00000020(32)	Controller Not Found
0x00000040(64)	Parameter is illegal
0x00000100(256)	Send Error
0x00000200(512)	Receive Error
0x00000400(1024)	Data is not Block Data
0x00001000(4096)	System Error
0x00002000(8192)	Device ID is Illegal

初期化関数を含め、関数の戻り値 0 (= OK) 以外の場合に、この関数を使用して、実際のエラー番号を取得します。

使用例 : errorID = mexDLGetLastError;

4.1 DL コントロール用 MEX-Functions

[ret] = mexDLSetTerm(eos, eot);

機能 : メッセージの送受信における、ターミネータを設定します。

引数 : eos ターミネータ

eot EOI

戻り値 : ret (0 = OK, 1 = ERROR)

詳細 : 引数説明

eos ターミネータを設定します。設定値は、以下のとおりです。

eos = 0 : CR+LF

eos = 1 : CR

eos = 2 : LF

eos = 3 : EOI(GPIB)、またはなし (RS-232、USB、Ethernet)

インターフェースが GP-IB で、eos が 3 のときに EOI を使用するかどうかは、eot で設定します。

eot ターミネータに EOI を使用するかどうか設定します。

GP-IB 専用です。

当社製品との通信を行なう場合には、どのインターフェースでも、基本的には、以下の設定をしてください。

```
mexDLSetTerm( 2,1 ); /* eos = LF, eot = TRUE */
```

binary データ受信時には、eos = LF の設定のままですと、binary コードに LF コードが含まれている場合、そこで終了と判断してしまいます。

ただし、当社製品でブロックデータを受信するときは、"mexDLReceiveBlockHeader"、"mexDLReceiveBlockData" を使用した場合は、ターミネータを切り替える必要はありません。

使用例 : ret = mexDLSetTerm(1,0);

[ret] = mexDLSetTimeout(tmo);

機能 : 通信のタイムアウト時間を設定します。

引数 : tmo タイムアウト時間 (100 ~ 6553600ms)

戻り値 : ret (0 = OK, 1 = ERROR)

詳細 : 引数説明

tmo タイムアウト時間を設定します。100ms 単位です。

tmo = 0 の場合は、

GP-IB : タイムアウト無限

その他 : タイムアウトなし

Note

「無限」は、待ち時間が無限時間という意味で、「なし」は待ち時間はなく、すぐに応答するという意味です。

通信のタイムアウト時間を設定します。

当社製品の場合は、通常 30s 以上を設定してください。

(タイムアウト時間を長くしても、全体のパフォーマンスには影響ありません。)

使用例 : ret = mexDLSetTimeout(300);

[ret, WaveData, Time] = mexDLGetWave();

機能 : 波形データを取得します。
引数 : なし
戻り値 : ret (0 = OK, 1 = ERROR)
WaveData 波形データの行列
Time トリガ点からの時間 (double 型、省略可)
詳細 : 戻り値説明
WaveData 設定されたレコード長のすべての波形データが、表示されているチャネルごとの行に格納されます。

HistoryWave =
$$\begin{bmatrix} W_{1,1} & \cdot & \cdot & \cdot & W_{1,Ch} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \\ \cdot & & & \cdot & \\ W_{Len,1} & \cdot & \cdot & \cdot & W_{Len,Ch} \end{bmatrix}$$

Time WaveData に対応するトリガ点からの時間情報が、配列に格納されます。
Time は省略可能です。

レコード長の長い波形データを取得する場合は、タイムアウト時間を長めに設定してください。
> ret = mexDLSetTimeout(1000);
DL 本体のロジック入力で捕捉したデータは格納されません。
使用例 : [ret, WaveData] = mexDLGetWave;
plot(Wavedata)
plot(Wavedata(:,2)); %Ch2 の波形を表示

Note

- 大容量の波形データを本コマンドにより MATLAB の行列に格納するときに、データを格納するために必要なメモリ領域が連続して確保できない場合は "Out of Memory" メッセージが表示されます。確保可能な連続したメモリ領域の容量はご使用の PC 環境に依存します。メッセージが表示された場合は、次の方法でデータを格納してください。
 - 仮想メモリサイズを変更する。
 - 行列のサイズを小さくする

波形の範囲を指定してデータを取得する関数

[ret, WaveData] = mexDLGetWave(traceNum, waveStart, waveEnd); を使用して、大きな行列をいくつかの小さい行列に分割し、MATLAB で一度に扱うデータ量を小さくします。

- DL850 シリーズのリアルタイム演算波形データは取得できません。

4.1 DL コントロール用 MEX-Functions

[ret, WaveData, Time] = mexDLGetWave(traceNum, waveStart, waveEnd);	
機能	： トレース番号、波形の範囲を指定し、波形データを取得します。
引数	： traceNum データを取得する波形のトレース番号 waveStart 取得する波形データの始点（省略可） waveEnd 取得する波形データの終点（省略可）
戻り値	： ret (0 = OK, 1 = ERROR) WaveData 波形データの行列（各要素は double 型） Time トリガ点からの時間（double 型、省略可）
詳細	： 引数説明 traceNum トレース番号を指定します。最小値は 1、最大値は接続する機種種のチャンネル数に依存します。0 を入力した場合、表示されている全チャンネルの波形を取得します。また、非表示となっているチャンネルのトレース番号は指定できません。 waveStart, waveEnd 取得する波形データの範囲を指定します。 waveStart、waveEnd の入力範囲は表示レコード長の範囲内です。また、waveStart、waveEnd は省略可能です（片方のみを省略することはできません）。省略した場合、表示レコード長の全波形データを取得します。
戻り値説明	WaveData 表示レコード内の指定された範囲で、指定されたトレース番号の波形データが行列に格納されます。画面上で非表示になっているチャンネルのデータを取得するとはできません。
$\text{HistoryWave} = \begin{bmatrix} \mathbf{W}_{1,1} & \cdot & \cdot & \cdot & \mathbf{W}_{1,\text{Ch}} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ \mathbf{W}_{\text{Len},1} & \cdot & \cdot & \cdot & \mathbf{W}_{\text{Len},\text{Ch}} \end{bmatrix}$	
Time	WaveData に対応するトリガ点からの時間情報が、配列に格納されます。Time は省略可能です。
使用例	： CH1、CH2、CH4 が画面に表示されている場合（表示レコード長は 10k ワードとする） [ret, WaveData] = mexDLGetWave(0); %CH1、CH2、CH4 の波形データをすべて取得 [ret, WaveData] = mexDLGetWave(1); %CH1 の波形データをすべて取得 [ret, WaveData] = mexDLGetWave(1,1000,4000); %CH1 の、1k ～ 4k ワード部分の波形データを取得

[ret, WaveData, Time] = mexDLGetWave(traceNum, subTraceNum, waveStart, waveEnd)	
機能	: DL850 シリーズのサブチャンネルデータ取得専用の関数です。サブトレース（チャンネル）番号、サブトレース（サブチャンネル）番号、波形の範囲を指定し、波形データを取得します。DL850 シリーズの多チャンネルモジュール（サブチャンネルを持つモジュール）に対してのみ使用可能です。サブチャンネルを持たないモジュールに対して指定するとエラーになります。
引数	: traceNum データを取得する波形のチャンネル番号 subTraceNum データを取得する波形のサブチャンネル番号 waveStart 取得する波形データの始点（省略可） waveEnd 取得する波形データの終点（省略可）
戻り値	: ret (0 = OK, 1 = ERROR) WaveData 波形データの行列（各要素は double 型） Time トリガ点からの時間（double 型、省略可）
詳細	: 引数説明 traceNum チャンネル番号を指定します。最小値は 1、最大値は接続する機種 のチャンネル数に依存します。また、非表示となっているチャンネルは指定出来ません。 subTraceNum サブチャンネル番号を指定します。最大値は装着された多チャンネルモジュールのサブチャンネル数に依存します。最小値は 1 です。非表示になっているサブチャンネルは指定できません。 waveStart, waveEnd 取得する波形データの範囲を指定します。 waveStart、waveEnd の入力範囲は表示レコード長の範囲内です。また、waveStart、waveEnd は省略可能です（片方のみを省略することはできません）。省略した場合、表示レコード長の全波形データを取得します。
戻り値説明	: WaveData 表示レコード内の指定された範囲で、指定されたサブチャンネル番号の波形データが行列に格納されます。画面上で非表示になっているチャンネルのデータを取得するとはできません。
HistoryWave = $\begin{bmatrix} \mathbf{W}_{1,1} & \cdot & \cdot & \cdot & \mathbf{W}_{1,\text{Ch}} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ \mathbf{W}_{\text{Len},1} & \cdot & \cdot & \cdot & \mathbf{W}_{\text{Len},\text{Ch}} \end{bmatrix}$	
Time	: WaveData に対応するトリガ点からの時間情報が、配列に格納されます。Time は省略可能です。
使用例	: CH1 の SubCH1、SubCH2、SubCH3 が画面に表示されている場合（表示レコード長は 10k ワードとする） [ret, WaveData] = mexDLGetWave(0); %CH1 の SubCH1、SubCH2、SubCH3 の波形データをすべて取得 [ret, WaveData] = mexDLGetWave(1, 1); %CH1 の SubCH1 の波形データをすべて取得 [ret, WaveData] = mexDLGetWave(1, 1, 1000, 4000); %CH1 の SubCH1 の、1k ～ 4k ワード部分の波形データを取得

[ret, HistoryWave, Time] = mexDLGetHistoryWave (rec);

機能 : ヒストリ波形データを取得します。
 引数 : rec レコード No. 最新の波形を 0 として、過去の波形に戻るに従い、-1、-2、-3... と番号付けします。設定範囲はそれぞれの DL のユーザーズマニュアルをご覧ください。

戻り値 : ret (0 = OK, 1 = ERROR)
 HistoryWave ヒストリ波形データの行列
 Time トリガ点からの時間 (double 型、省略可)

詳細 : 戻り値説明
 HistoryWave 選択されたレコード No. の波形データが、表示されているチャンネルごとの列に格納されます。

$$\text{HistoryWave} = \begin{bmatrix} \mathbf{W}_{1,1} & \cdot & \cdot & \cdot & \mathbf{W}_{1,\text{Ch}} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{W}_{\text{Len},1} & \cdot & \cdot & \cdot & \mathbf{W}_{\text{Len},\text{Ch}} \end{bmatrix}$$

Time HistoryWave に対応するトリガ点からの時間情報が、配列に格納されます。Time は省略可能です。

レコード長の長い波形データを取得する場合は、タイムアウト時間を長めに設定してください。

```
> ret = mexDLSetTimeout(1000);
```

DL 本体のロジック入力で捕捉したデータは格納されません。

使用例 : [ret, HistoryWave] = mexDLGetHistoryWave(-20);
 -20 番目の History 波形データをすべて取得

Note

大容量の波形データを本コマンドにより MATLAB の行列に格納するときに、データを格納するために必要なメモリ領域が連続して確保できない場合は "Out of Memory" メッセージが表示されます。確保可能な連続したメモリ領域の容量はご使用の PC 環境に依存します。メッセージが表示された場合は、次の方法でデータを取得してください。

- メモリのスワップスペースを拡大する
 1. マイコンピュータ (My Computer) アイコンを右クリックし、プロパティ (Properties) を選択します。
 2. 詳細 (Advanced) タブを選択し、パフォーマンスオプション (Performance Options) を選択します。
 3. 変更 (Change) ボタンをクリックし、仮想メモリのサイズを変更します。
- 行列のサイズを小さくする
 波形の範囲を指定してデータを取得する関数

```
[ret, HistoryWave] = mexDLGetHistoryWave( rec, traceNum, waveStart, waveEnd );
```

 を使用して、大きな行列をいくつかの小さい行列に分割し、MATLAB で一度に扱うデータ量を小さくします。

[ret, HistoryWave, Time] = mexDLGetHistoryWave (rec, traceNum, waveStart, waveEnd);

機能 : レコード番号、トレース番号、波形の範囲を指定し、ヒストリ波形データを取得します。
引数 : rec レコード番号
traceNum データを取得する波形のトレース番号
waveStart 取得する波形データの始点 (省略可)
waveEnd 取得する波形データの終点 (省略可)
Time トリガ点からの時間 (double 型、省略可)

戻り値 : ret (0 = OK, 1 = ERROR)
HistoryWave ヒストリ波形データの行列 (各要素は double 型)
Time トリガ点からの時間 (double 型、省略可)

詳細 : 引数説明

rec レコード No. 最新の波形を 0 として、過去の波形に戻るに従い、-1、-2、-3... と番号付けします。設定範囲はそれぞれの DL のユーザズマニュアルをご覧ください。

traceNum トレース番号を指定します。最小値は 1、最大値は接続する機種種のチャンネル数に依存します。0 を入力した場合、表示されている全チャンネルの波形を取得します。また、非表示となっているチャンネルのトレース番号は指定できません。

waveStart, waveEnd 取得する波形データの範囲を指定します。waveStart、waveEnd の入力範囲は表示レコード長の範囲内です。また、waveStart、waveEnd は省略可能です (片方のみを省略することはできません)。省略した場合、表示レコード長の全波形データを取得します。

戻り値説明

HistoryWave 選択されたレコード No. の波形データが、表示されているチャンネルごとの行列に格納されます。画面上で非表示になっているチャンネルデータは取得しません。

$$\text{HistoryWave} = \begin{bmatrix} \mathbf{W}_{1,1} & \cdot & \cdot & \cdot & \mathbf{W}_{1,\text{Ch}} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \\ \cdot & & & \cdot & \\ \mathbf{W}_{\text{Len},1} & \cdot & \cdot & \cdot & \mathbf{W}_{\text{Len},\text{Ch}} \end{bmatrix}$$

Time HistoryWave に対応するトリガ点からの時間情報が、配列に格納されます。Time は省略可能です。

使用例 : CH1、CH2、CH4 が画面に表示されている場合 (表示レコード長は 10k ワードとする)

```
[ret, HistryWave] = mexDLGetHistoryWave (-20, 0);  
%CH1、CH2、CH4 の、-20 番目の History 波形データをすべて取得  
[ret, HistryWave] = mexDLGetHistoryWave (-20, 4);  
%CH4 の、-20 番目の History 波形データをすべてを取得  
[ret, HistryWave] = mexDLGetHistoryWave (-20, 4, 2000, 5000);  
%CH4 の、-20 番目の History 波形データの 2k ~ 5k ワード部分を取得
```

[ret, HistoryWave, Time] = mexDLGetHistoryWave(rec, traceNum, subTraceNum, waveStart, waveEnd)

機能 : DL850 シリーズのサブチャンネルデータ取得専用の関数です。サブトレース (チャンネル) 番号、サブトレース (サブチャンネル) 番号、波形の範囲を指定し、ヒストリ波形データを取得します。DL850 シリーズの多チャンネルモジュール (サブチャンネルを持つモジュール) に対してのみ使用可能です。サブチャンネルを持たないモジュールに対して指定するとエラーになります。

引数 : rec レコード番号
 traceNum データを取得する波形のチャンネル番号
 subTraceNum データを取得する波形のサブチャンネル番号
 waveStart 取得する波形データの始点 (省略可)
 waveEnd 取得する波形データの終点 (省略可)

戻り値 : ret (0 = OK, 1 = ERROR)
 HistoryWave ヒストリ波形データの行列 (各要素は double 型)
 Time トリガ点からの時間 (double 型、省略可)

詳細 : 引数説明

rec レコード No. 最新の波形を 0 として、過去の波形に戻るに従い、-1、-2、-3... と番号付けします。設定範囲はそれぞれの DL のユーザズマニュアルをご覧ください。

traceNum チャンネル番号を指定します。最小値は 1、最大値は接続する機種種のチャンネル数に依存します。また、非表示となっているチャンネルは指定出来ません。

subTraceNum サブチャンネル番号を指定します。最大値は装着された多チャンネルモジュールのサブチャンネル数に依存します。最小値は 1 です。非表示になっているサブチャンネルは指定できません。

waveStart, waveEnd 取得する波形データの範囲を指定します。waveStart、waveEnd の入力範囲は表示レコード長の範囲内です。また、waveStart、waveEnd は省略可能です (片方のみを省略することはできません)。省略した場合、表示レコード長の全波形データを取得します。

戻り値説明

HistoryWave 選択されたレコード No. の波形データが、表示されているチャンネル毎の行列に格納されます。画面上で非表示になっているチャンネルのデータは取得しません。

$$\text{HistoryWave} = \begin{bmatrix} \mathbf{W}_{1,1} & \cdot & \cdot & \cdot & \mathbf{W}_{1,\text{Ch}} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \\ \cdot & & & \cdot & \cdot \\ \mathbf{W}_{\text{Len},1} & \cdot & \cdot & \cdot & \mathbf{W}_{\text{Len},\text{Ch}} \end{bmatrix}$$

Time HistoryWave に対応するトリガ点からの時間情報が、配列に格納されます。Time は省略可能です。

使用例 : CH1 の SubCH1、SubCH2、SubCH4 が画面に表示されている場合 (表示レコード長は 10k ワードとする)

```
[ret, WaveData] = mexDLGetWave(-20, 0);
%CH1 の SubCH1、SubCH2、SubCH4 の、-20 番目の History 波形データをすべて取得
[ret, WaveData] = mexDLGetWave(-20, 1, 4);
%CH1 の SubCH4 の、-20 番目の History 波形データをすべて取得
[ret, WaveData] = mexDLGetWave(-20, 1, 4, 2000, 5000);
%CH1 の SubCH4 の、-20 番目の History 波形データの 2k ~ 5k ワード部分を取得
```

[ret] = mexDLComEnd();

機能 : デバイスと 接続している回線を閉じます。
引数 : なし
戻り値 : ret (0 = OK, 1 = ERROR)
詳細 : "mexDLComStart" (初期化関数) で開いた回線を閉じます。
この関数は、通信を終了するときには、必ず実行してください。
使用例 : ret = mexDLComEnd;

mexDLToolkit;

機能 : 本ソフトウェアのバージョン情報を表示します。
引数 : なし
戻り値 : なし
使用例 : help mexDLToolkit;
Model 701991
MATLAB ToolKit for DL Series
Version *.*

All Rights Reserved,
Copyright (c) (西暦) Yokogawa Meters & Instruments Corporation

4.2 WDF ファイルアクセス用 MEX-Functions

4.2.1 WDF ファイルアクセス用 MEX-Function 一覧

ファイル情報アクセス

mex 関数名	機能	ページ
mexWdfItemRead	ファイル情報の読み込み	4-20
mexWdfGetBlockNum	ブロック数の読み込み	4-20
mexWdfGetChNum	チャンネル数の読み込み	4-20

データ操作

mex 関数名	機能	ページ
mexWdfDataRead	生波形データの読み込み	4-21
mexWdfDataReadEx	生波形データの読み込み (拡張版)	4-21
mexWdfScaleDataRead	物理値波形データの読み込み	4-22
mexWdfScaleDataReadEx	物理値波形データの読み込み (拡張版)	4-22

4.2.2 ファイル情報アクセス

[ret, data] = mexWdfItemRead(filename, itemName, ch, block)	
機能	ファイル情報の読み込み
引数	filename : 読み込まれる拡張子付きファイル名 itemName : 読み込まれるアイテム名 (パラメータ参照) ch : 読み込まれるチャンネル番号 (0 ~) block : 読み込まれるブロック番号 (0 ~)
戻り値	ret : 正常終了したときは「0」、異常終了したときはエラーコードを返します。 data : 読み込まれたデータ
説明	指定した WDF ファイルから、指定したチャンネル、ブロック、項目のファイル情報を取得します。

[ret, blockNum] = mexWdfGetBlockNum(filename)	
機能	ブロック数の読み込み
引数	filename : 読み込まれる拡張子付きファイル名
戻り値	ret : 正常終了したときは「0」、異常終了したときはエラーコードを返します。 blockNum : 読み込まれたブロック数
説明	指定した WDF ファイルのブロック数を取得します。

[ret, chNum] = mexWdfGetChNum(filename)	
機能	チャンネル数の読み込み
引数	filename : 読み込まれる拡張子付きファイル名
戻り値	ret : 正常終了したときは「0」、異常終了したときはエラーコードを返します。 chNum : 読み込まれたチャンネル数
説明	指定した WDF ファイルのチャンネル数を取得します。

4.2.3 データ操作

[ret, param, data] = mexWdfDataRead(filename, ch, block)

機能	生波形データの読み込み
引数	filename: 読み込まれる拡張子付きファイル名
	ch: 読み込まれるチャンネル番号 (0 ~)
	block: 読み込まれるブロック番号 (0 ~)
戻り値	ret: 正常終了したときは「0」、異常終了したときはエラーコードを返します。
	param: 読み込まれた情報の構造体 (構造体参照)
	data: 読み込まれたデータ (配列)
説明	指定した WDF ファイルから、指定したチャンネル、ブロックのデータを取得します。

[ret, param, data] = mexWdfDataReadEx(filename, ch, block, start, length)

機能	生波形データの読み込み (拡張版)
引数	filename: 読み込まれる拡張子付きファイル名
	ch: 読み込まれるチャンネル番号 (0 ~)
	block: 読み込まれるブロック番号 (0 ~)
	start: 読み込まれるデータの開始点
	length: 読み込まれるデータの点数
戻り値	ret: 正常終了したときは「0」、異常終了したときはエラーコードを返します。
	param: 読み込まれた情報の構造体 (構造体参照)
	data: 読み込まれたデータ (配列)
説明	指定したファイルから、指定したチャンネル、ブロックのデータを指定範囲分のみ取得します。

[ret, param, data] = mexWdfScaleDataRead(filename, ch, block)

機能	物理値波形データの読み込み
引数	filename: 読み込まれる拡張子付きファイル名 ch: 読み込まれるチャンネル番号 (0 ~) block: 読み込まれるブロック番号 (0 ~)
戻り値	ret: 正常終了したときは「0」、異常終了したときはエラーコードを返します。 param: 読み込まれた情報の構造体 (構造体参照) data: 読み込まれたデータ (配列)
説明	指定した WDF ファイルから、指定したチャンネル、ブロックの物理値データを取得します。ファイルの生データ (符号付き 16 ビット整数) に VResolution を乗じて、VOffset を加えます。

[ret, param, data] = mexWdfScaleDataReadEx(filename, ch, block, start, length)

機能	物理値波形データの読み込み (拡張版)
引数	filename: 読み込まれる拡張子付きファイル名 ch: 読み込まれるチャンネル番号 (0 ~) block: 読み込まれるブロック番号 (0 ~) start: 読み込まれるデータの開始点 length: 読み込まれるデータの点数
戻り値	ret: 正常終了したときは「0」、異常終了したときはエラーコードを返します。 param: 読み込まれた情報の構造体 (構造体参照) data: 読み込まれたデータ (配列)
説明	指定した WDF ファイルから、指定したチャンネル、ブロックの物理値データを指定範囲のみ取得します。ファイルの生データ (符号付き 16 ビット整数) に VResolution を乗じて、VOffset を加えます。

4.2.4 パラメータと構造体

パラメータ

itemName エリアに入れる項目	dataType	パラメータ		内容
		ch	block	
Comment	string	×	×	コメント文字列
Version	string	×	×	バージョン文字列
Model	string	×	×	モデル名
TraceNumber	UINT	×	×	チャンネル数
BlockNumber	UINT	×	×	ブロック数
TraceName	string	○	×	チャンネル名
BlockSize	UINT	○	○	ブロックサイズ
VDataType	UINT	○	○	データの型
VUnit	string	○	○	縦軸の単位文字列
VResolution	double	○	○	縦軸の分解能
VOffset	double	○	○	縦軸のオフセット
VScaleUpper	double	○	○	縦軸の上限スケール値
VScaleLower	double	○	○	縦軸の下限スケール値
HResolution	double	○	○	横軸の分解能
HOffset	double	○	○	横軸のオフセット
HUnit	string	○	○	横軸の単位文字列
Date	string	○	○	日付
Time	string	○	○	時刻
DateTime	string	○	○	日付時刻
VIllegalData	double	○	○	欠損値
VMaxData	double	○	○	最大値
VMinData	double	○	○	最小値
SplitNumMain	UINT	×	×	Main の表示分割数
SplitNumZ1	UINT	×	×	Zoom-1 の表示分割数
SplitNumZ2	UINT	×	×	Zoom-2 の表示分割数
TraceColor0	UINT	○	×	波形色 (通常時)
TraceColor1	UINT	○	×	波形色 (中間色)

* ○：必要、×：無視

構造体

パラメータ	意味	値
ch	読み込みチャンネル	0 ～
block	読み込みブロック	0 ～
start	読み込み開始点	0 ～
count	読み込み点数	1 ～
waveType	出力波形の種類	0：測定生波形 (AD 値) 1：物理値変換波形
dataType	出力波形のデータ型	0：符号なし 8 ビット整数 1：符号付き 8 ビット整数 4：8 ビット論理値 16：符号なし 16 ビット整数 17：符号付き 16 ビット整数 20：16 ビット論理値 32：符号なし 32 ビット整数 33：符号付き 32 ビット整数 36：32 ビット論理値 48：符号なし 64 ビット整数 49：符号付き 64 ビット整数 52：64 ビット論理値 34：単精度実数 50：倍精度実数 256：なし
cntOut	出力点数	読み込み成功点数

4.2.5 エラーコード

エラーコード	内容
0	正常終了
100	ファイルオープン失敗
101	アロケーション失敗
102	ファイルアクセスエラー
103	DLL のリンクに失敗
200	未対応なバージョン
201	未対応なフォーマット
202	不明な関数
300	範囲指定エラー
301	ファイルハンドルが未取得
900	リアルタイム測定に失敗しているデータ
901	データ値が異常
902	読み込めないデータ (PPsave、Z1/Z2save)
1000	その他のエラー

索引

D ページ

DL コントロール用 MEX-Function	
mexDLCheckEnd	4-6
mexDLComEnd	4-15
mexDLComStart.....	4-2
mexDLDeviceClear.....	4-2
mexDLGetHistoryWave.....	4-12,4-13,4-14
mexDLGetLastError.....	4-7
mexDLGetWave	4-9,4-10,4-11
mexDLReceive	4-4
mexDLReceiveBlockData.....	4-6
mexDLReceiveBlockHeader	4-5
mexDLReceiveOnly.....	4-5
mexDLReceiveSetup	4-4
mexDLSend.....	4-2
mexDLSendByLength	4-3
mexDLSendOnly.....	4-3
mexDLSendSetup.....	4-3
mexDLSetRen.....	4-7
mexDLSetTerm.....	4-8
mexDLSetTimeout	4-8
mexDLToolkit.....	4-15
DL コントロール用 MEX-Function 一覧.....	4-1

W ページ

WDF サンプルプログラム	3-3
WDF ファイルアクセス用 MEX-Functions	
mexWdfDataRead	4-18
mexWdfDataReadEx.....	4-18
mexWdfGetBlockNum.....	4-17
mexWdfGetChNum	4-17
mexWdfItemRead	4-17
mexWdfScaleDataRead.....	4-19
mexWdfScaleDataReadEx.....	4-19
WDF ファイルアクセス用 MEX-Function 一覧.....	4-16

サ ページ

サンプルプログラム	3-3
サンプルプログラム 1	2-4
サンプルプログラム 2	2-5

ソ ページ

ソフトウェア使用許諾契約書	ii
---------------------	----