

MATLAB 対応データ収録用ライブラリ ML-DAQ セットアップ&リファレンスガイド



1. はじめに.....	3
2. ML-DAQのセットアップ.....	4
2-1. デバイスドライバのインストール.....	4
2-2. 診断プログラムによるテスト.....	4
2-3. ML-DAQのセットアップ.....	5
3. ML-DAQリファレンス.....	6
3-1. 第10章 関数リファレンス.....	6
3-2. 第11章 基本プロパティリファレンス.....	7
3-3. 第12章 デバイス固有のプロパティリファレンス.....	8
4. 同期制御の方法.....	9
4-1. AI開始トリガ(マスタ)とAO開始トリガ(スレーブ)の同期.....	9
4-2. AO開始トリガ(マスタ)とAI開始トリガ(スレーブ)の同期.....	9
4-3. AIクロック(マスタ)とAOクロック(スレーブ)の同期.....	10
4-3. AOクロック(マスタ)とAIクロック(スレーブ)の同期.....	10
5. ML-DAQバージョンアップ履歴.....	11

1. はじめに

ML-DAQ は、弊社のアナログ入出力ボードを The MathWorks 社の MATLAB で使用するためのライブラリです。各機能は、MATLAB の Data Acquisition Toolbox で統一されたインターフェイスに合わせて提供されます。本セットアップガイドは、ML-DAQ をご使用いただくまでの手順と、Data Acquisition Toolbox User's Guide Version 2 日本語版 (daqug.pdf) の補足説明が記述されています。

2. ML-DAQのセットアップ

2-1. デバイスドライバのインストール

MATLAB で ML-DAQ を使用する前に、デバイスドライバのインストールを行い、ハードウェアを使用可能な状態にしておく必要があります。なお、デバイスドライバのインストールは、管理者権限でログインして行ってください。

1. <http://www.contec.co.jp/apipac/>より、
Windows版 高機能アナログ入出力ドライバ API-AIO(WDM) をダウンロードします。
ダウンロードするファイルは実行環境(ランタイム)です。
2. ダウンロードしたファイルを、適当な場所で解凍します。
3. <解凍先フォルダ>\Inf\Wdm\Aio\Setup.exe を実行して、デバイスドライバをインストールします。
4. OS をシャットダウンし、ハードウェアを PCI スロットまたはカードバススロットに実装します。カードバスや PC カードの場合は、OS を起動したままデバイスをプラグインします。
5. パソコンの電源を ON にして、Windows を起動します。
6. Windows が起動すると、自動的にハードウェアウィザードが起動します。
7. Windows XP の場合、ウィザードの開始画面で[ソフトウェアを自動的にインストールする(推奨)]を選択します。Windows Vista や Windows 2000 の場合、インストールは自動的に行われます。

2-2. 診断プログラムによるテスト

デバイスのインストールが正しく行われているかどうかを、診断プログラムにより確認できます。

1. スタートメニューから、[ファイル名を指定して実行]を選択し、CAIODIAG.EXE と入力し診断プログラムを起動します。
2. 診断プログラム中で[診断レポート]をクリックすると、診断結果がレポートで表示されます。
3. 診断レポートで、API-AIO(WDM) のバージョンが最新かどうか、エラー等が発生していないかを確認してください。

2-3. ML-DAQのセットアップ

1. ダウンロードした MWCONTEC.DLL を以下の場所へコピーします。
\$MATLAB¥toolbox¥daq¥daq¥private¥
(\$MATLAB は、MATLAB のインストールディレクトリになります)
2. MATLAB を起動します。
3. コマンドウィンドウから以下のコマンドを実行し、アダプタの登録を行います。
>>> rehash toolboxcache
>> daqregister('contec')
'mwcontec.dll' successfully registered. と表示されていれば、セットアップ完了です。
4. コマンドウィンドウから以下のコマンドを実行し、デバイスが使用可能である事を確認します。
>> daqhwinfo('contec')
.
.
BoardNames: {'ADA16-32/2 (PCI) F'}
.
使用可能なデバイス一覧が、BoardNames に表示されます。

3. ML-DAQリファレンス

Data Acquisition Toolbox の使い方については、MATLAB 統合環境内の HELP もしくは、PDF 形式のユーザーズガイドを参考にしてください。ここでは、Data Acquisition Toolbox User's Guide Version 2 日本語版 (daqug.pdf) についての補足説明を行います。

3-1. 第10章 関数リファレンス

P10-10. addchannel

備考 チャンネル追加時の規則

- ・ hwch に与える数値はアクセスするハードウェアに依存します。
Contec のハードウェアに対してはチャンネルは“ゼロベース” (0 で始まる) です。
- ・ 一部の Contec デバイスにおいて、追加されるチャンネルは必ず昇順でなければなりません。 (AI0-160802L-LPE, AI-1616L-LPE, AO-1604L-LPE, ADA16-8/2(LPCI)L, ADA16-8/2(LPCI)L, ADA16-8/2(CB)L, AD16-64(LPCI)LA, AD16-16(LPCI)L, AD16-16(LPCI)L, AI0-160802AY-USB, AI-1608AY-USB)

P10-15. addline

備考 ラインを追加するときの規則

- ・ Contec のハードウェアに対しては、ライン ID は“ゼロベース” (0 で始まる) です。

P10-18. analoginput

引数 'adaptor' Contec 製アダプタでは、'contec' を指定します。

ID Contec 製アダプタでは、'AI0000' 等のデバイス名文字列を指定します。

P10-21. analogoutput

引数 'adaptor' Contec 製アダプタでは、'contec' を指定します。

ID Contec 製アダプタでは、'AI0000' 等のデバイス名文字列を指定します。

P10-34. daqhwinfo

引数 'adaptor' Contec 製アダプタでは、'contec' を指定します。

P10-47. daqregister

引数 'adaptor' Contec 製アダプタでは、'contec' を指定します。

説明 daqregister('adaptor') は、adaptor によって指定されたハードウェアドライバアダプタを登録します。Contec 製のアダプタを指定する場合は、adaptor にフルパス

名を与えなければなりません。

`daqregister('adaptor','unload')` は、`adaptor` によって指定されたハードウェアドライバアダプタを抹消します。Contec 製のアダプタを指定する場合は、`adaptor` にフルパス名を与えなければなりません。

P10-55. digitalio

引数 `'adaptor'` Contec 製アダプタでは、`'contec'` を指定します。

ID Contec 製アダプタでは、`'AI0000'` 等のデバイス名文字列を指定します。

3-2. 第11章 基本プロパティリファレンス

P11-23. ChannelSkewMode

値 Contec

{Minimum} チャンネルスキューはサポートされている最小値。

Equisample チャンネルスキューは $[(\text{サンプリングレート} \times \text{チャンネル数})] - 1$ で与えられます。

Manual チャンネルスキューは ChannelSkew で与えられます。

P11-24. ClockSource

値 Contec

{Internal} ハードウェア内部のクロックを使用。

External 外部サンプルクロック。

Software コンピュータクロックを使用。

P11-24. InputType

値 Contec

Differential チャンネルは差動入力に設定される。

{SingleEnded} チャンネルはシングルエンド入力に設定される。

P11-94. TriggerCondition

值 Contec

つぎのトリガ条件は、TriggerType が HwDigital の時に使用されます。

[PositiveEdge] デジタル信号の正のエッジ（立ち上がり）を検出したときにトリガが発生。

NegativeEdge デジタル信号の負のエッジ（立ち下がり）を検出したときにトリガが発生。

つぎのトリガ条件は、TriggerType が HwAnalog の時に使用されます。

[Both] アナログ信号が指定した値の範囲を通過するとき、
正または負の勾配を有している場合にトリガが発生。

Rising	アナログ信号が指定した値の範囲を通過するとき、 正の勾配を有している場合にトリガが発生。
--------	---

Falling	アナログ信号が指定した値の範囲を通過するとき、負の勾配を有している場合にトリガが発生。
---------	---

Leaving アナログ信号が指定した値の範囲から外れるときにトリガが発生。

Entering アナログ信号が指定した値の範囲に入るときにトリガが発生。

P11-104. TriggerType

值 Contec

HwDigital トリガソースは外部デジタル信号。プリトリガデータは確保されません。

HwAnalog トリガソースは外部アナログ信号 (AI のみ)。

P11-101. TriggerRepeat

AI-1608AY-USB では、TriggerType が HwDigital または HwAnalog の時にトリガを複数回設定することはできません。TriggerRepeat = 0 としてください。

3-3. 第12章 デバイス固有のプロパティリファレンス

P12-3. プロパティ - ベンダ別

Contec プロパティ

プロパティ名 説明

OutOfDataMode	アナログ出力サブシステムによって定義された値の定義方法を指定。 (デバイスオブジェクト: A0)
---------------	---

4. 同期制御の方法

CONTEC 製 F シリーズマルチファンクションデバイスをご使用の場合、アナログ入力とアナログ出力の同期スタート、アナログ入力とアナログ出力の同期クロック機能を使用することができます。

サポートデバイス：AI0-163202F-PE、ADA16-32/2 (PCI)F、ADA16-32/2 (CB)F

4-1. AI開始トリガ（マスタ）とAO開始トリガ（スレーブ）の同期

AI の開始トリガ信号を、ハードウェア内部で AO の開始トリガに接続し、AI と AO を同時にスタートさせる方法です。

＜サンプルコード＞

ai = analoginput('contec', 'AI0000')	'AI オブジェクトの生成'
ao = analogoutput('contec', 'AO0000')	'AO オブジェクトの生成'
addchannel(ai, 0)	'チャンネルの追加'
addchannel(ao, 0)	'チャンネルの追加'
data = sin(linspace(0.2*pi, 8000))';	'出力データの作成'
putdata(ao, data)	'出力データのセット'
<u>ao.TriggerType = 'HwSync'</u>	<u>'ハードウェア同期トリガを使用'</u>
<u>ao.TriggerCondition = 'AiStart'</u>	<u>'AI の開始信号に同期'</u>
start(ao)	'先に AO を開始'
start(ai)	'AI の開始'

4-2. AO開始トリガ（マスタ）とAI開始トリガ（スレーブ）の同期

AO の開始トリガ信号を、ハードウェア内部で AI の開始トリガに接続し、AO と AI を同時にスタートさせる方法です。

＜サンプルコード＞

ai = analoginput('contec', 'AI0000')	'AI オブジェクトの生成'
ao = analogoutput('contec', 'AO0000')	'AO オブジェクトの生成'
addchannel(ai, 0)	'チャンネルの追加'
addchannel(ao, 0)	'チャンネルの追加'
data = sin(linspace(0.2*pi, 8000))';	'出力データの作成'
putdata(ao, data)	'出力データのセット'
<u>ai.TriggerType = 'HwSync'</u>	<u>'ハードウェア同期トリガを使用'</u>
<u>ai.TriggerCondition = 'AoStart'</u>	<u>'AO の開始信号に同期'</u>

start(ai)	' 先に AI を開始
start(ao)	' A0 の開始

4-3. AIクロック（マスタ）とAOクロック（スレーブ）の同期

AI のクロック信号を、ハードウェア内部で A0 のクロック信号に接続し、AI と A0 変換タイミングを同期させる方法です。

<サンプルコード>

ai = analoginput('contec', 'AI0000')	' AI オブジェクトの生成
ao = analogoutput('contec', 'AI0000')	' A0 オブジェクトの生成
addchannel(ai, 0)	' チャンネルの追加
addchannel(ao, 0)	' チャンネルの追加
data = sin(linspace(0.2*pi, 8000))';	' 出力データの作成
putdata(ao, data)	' 出力データのセット
<u>ao.ClockSource = 'HwSync'</u>	<u>' 同期クロック</u>
<u>ao.ClockCondition = 'AiClock'</u>	<u>' AI のクロックと同期</u>
start(ao)	' 先に A0 を開始
start(ai)	' AI の開始

4-3. AOクロック（マスタ）とAIクロック（スレーブ）の同期

A0 のクロック信号を、ハードウェア内部で AI のクロック信号に接続し、A0 と AI 変換タイミングを同期させる方法です。

<サンプルコード>

ai = analoginput('contec', 'AI0000')	' AI オブジェクトの生成
ao = analogoutput('contec', 'AI0000')	' A0 オブジェクトの生成
addchannel(ai, 0)	' チャンネルの追加
addchannel(ao, 0)	' チャンネルの追加
data = sin(linspace(0.2*pi, 8000))';	' 出力データの作成
putdata(ao, data)	' 出力データのセット
<u>ai.ClockSource = 'HwSync'</u>	<u>' 同期クロック</u>
<u>ai.ClockCondition = 'AoClock'</u>	<u>' A0 のクロックと同期</u>
start(ai)	' 先に AI の開始
start(ao)	' A0 を開始

5. ML-DAQバージョンアップ履歴

- ・ Ver1.44 2009/10/09

AI-1204Z-PCI で、10MHz のサンプルレートが設定できない不具合を修正しました。
特定の PC (Intel Core 2 Quad CPU Q9650 3.00GHz) で動作しない問題を修正しました。

- ・ Ver1.41 2009/07/03

trigger (ao) 関数でエラーが発生することがある不具合を修正しました。

- ・ Ver1.40 2008/11/14

アナログ入力とアナログ出力の同期スタート、
アナログ入力とアナログ出力の同期クロック機能に対応しました。(F シリーズ専用機能)
タイムアウトメッセージが発生する事がある不具合を修正しました。

- ・ Ver1.30 2008/03/28

AI-1204Z-PCI に対応しました。

- ・ Ver1.22c 2007/10/19 (mwcontec.dll の変更はありません)

AI0-121601E3-PE, AI0-121601UE3-PE, AI0-161601E3-PE, AI0-161601UE3-PE に対応しました。
Windows Vista に対応しました。

- ・ Ver1.22 2007/4/9

外部トリガ+リピート使用時に、取得データが異常値となる不具合を修正しました。
Simulink を使用して、アナログ入力とデジタル入力を同時に使用すると正常に動作しない不
具合を修正しました。

- ・ Ver1.21 2006/11/13

USB ターミナル使用時、条件によりサンプリングクロックエラーが発生する不具合を
修正しました。

- ・ Ver1.20b 2006/7/20 (mwcontec.dll の変更はありません)

AI0-160802L-LPE, AI-1616L-LPE, AO-1604L-LPE, AI0-160802AY-USB
に対応しました。

- ・ Ver1.20a 2006/4/14 (mwcontec.dll の変更はありません)

AI0-163202F-PE, AI0-1608AY-USB, DA16-16 (LPCI) L, DA16-8 (LPCI) L

に対応しました。

- ・ Ver1. 20 2005/11/11

AD12-16 (PCI) EV, AD12-16U (PCI) EV, AD16-16 (PCI) EV, AD16-16U (PCI) EV, AD16-64 (LPCI) LA
に対応しました。

- ・ Ver1. 10 2005/09/09

ADA16-8/2 (LPCI) L, AD16-16 (LPCI) L, DA16-4 (LPCI) L に対応しました。

- ・ Ver1. 00 2005/05/24

ADA16-32/2 (PCI) F, ADA16-8/2 (LPCI) L, AD16-16 (LPCI) L, DA16-4 (LPCI) L, AD16-32/2 (PCI) F,
ADA16-8/2 (LPCI) に対応しました。