Data Acquisition Library for MATLAB

ML-DAQ Setup & Reference Guide

1. Preface

ML-DAQ is the library for using CONTEC Analog Input/Output boards in MATLAB of The MathWorks. All the functions are supported in the same interface with the Data Acquisition Toolbox of MATLAB. In this setup guide, the usage of ML-DAQ and the supplementary explanation for Data Acquisition Toolbox User's Guide Version 2 (English edition) are presented.

2. ML-DAQ Setup

2-1. Device Driver Installation

Before using ML-DAQ with MATLAB, it is necessary to install the device driver and make the hardware valid. Moreover, please login with administrator to install the device driver.

1. Download the high functionality Analog I/O driver for Windows, i.e. API-AIO(WDM), from
   http://www.contec.com/apipac/index.html
   The file to download is for Run-Time.
2. Extract the downloaded file in a suitable place.
3. Run <the folder where the file was extracted>/Inf/Wdm/Aio/Setup.exe to install the device driver.
4. Shutdown the OS, and insert the hardware to PCI slot or Card-Bus slot. You can plug in devices such as card bus and PC card with OS started.
5. Turn on PC and start Windows.
6. While Windows is started, the Hardware Wizard is launched automatically.
7. For Windows XP, please select [Install the software automatically (Recommended)] at beginning of wizard. For Windows Vista and Windows 2000, installation completes automatically.

2-2. diagnosing the device

It can be confirmed by using Diagnosis Program whether the installation of device is performed correctly.

1. Select [Run…] from Start menu and input CAIODIAG.EXE to run the Diagnosis Program.
2. Click [Diagnosis Report] in Diagnosis Program to display diagnosis result in report.
3. Please confirm in diagnosis report whether the version of API-AIO(WDM) is latest and whether there are some errors occurred.

2-3. ML-DAQ Installation

1. Copy the downloaded file MWCONTEC.DLL to the following location.
   $MATLAB/toolbox/daq/daq/private/
   ($MATLAB is the directory where the MATLB is installed.)
2. Start the MATLAB.
3. Execute the following commands in Command Window to register the adaptor.
   >> rehash toolboxcache
   >> daqregister('contec')
   While ['mwcontec.dll' successfully registered.] is displayed, it means the installation succeeded.
4. Execute the following command in Command Window to confirm whether the device is valid.
   >> daqhwinfo('contec')
   .
   .
   BoardNames: {'ADA16-32/2(PCI)F'}
   .
   The list of valid devices is displayed in [BoardNames].

3. ML-DAQ Reference

Please refer to the User Guide in PDF format or Help of MATLAB IDE for how to use Data Acquisition Toolbox. The following contents are just the supplementary explanation for Data Acquisition Toolbox User's Guide Version 2 (English edition).

3-1. Chapter 11, Functions – Alphabetical List

P11-2.addchannel

Remarks Rules for Adding Channels

The numeric values you supply for *hwch* depend on the hardware you access.

For Contec hardware, channels are “zero-based” (begin at zero).

For some Contec hardware, added channels must be increasing order.

(AIO-160802L-LPE, AI-1616L-LPE, AO-1604L-LPE, ADA16-8/2(LPCI)L,ADAI16-8/2(LPCI)L, ADA16-8/2(CB)L, AD16-64(LPCI)LA, AD16-16(LPCI)L, ADI16-16(LPCI)L, AIO-160802AY-USB, AI-1608AY-USB)

P11-8.addline

Remarks Rules for Adding Lines

The numeric values you supply for *hwline* depend on the hardware you access.

For Contec hardware, line IDs are “zero-based” (begin at zero).

P11-14.analoginput

Arguments	'adaptor'	For Contec adaptor, 'contec' is specified.
	ID	For Contec adaptor, the string indicating Device-Name is specified such as 'AIO000'.

P11-17.analogoutput

Arguments	'adaptor'	For Contec adaptor, 'contec' is specified.
	ID	For Contec adaptor, the string indicating Device-Name is specified such as 'AIO000'.

P11-31.daqhwinfo

Arguments	'adaptor'	For Contec adaptor, 'contec' is specified.

P11-41.daqregister

Arguments	'adaptor'	For Contec adaptor, 'contec' is specified.
Description		*daqregister*(‘adaptor’) registers the hardware driver adaptor specified by adaptor. For Contec adaptors, adaptor must include the full pathname. *daqregister*('adaptor','unload’) unregisters the hardware driver adaptor specified by adaptor. For Contec adaptors, adaptor must include the full pathname.

P11-49.digitalio

Arguments	'adaptor'	For Contec adaptor, 'contec' is specified.
	ID	For Contec adaptor, the string indicating Device-Name is specified such as 'AIO000'.

3-2. Chapter 13, Base Properties – Alphabetical List

P13-12.ChannelSkewMode

Values	Contec
{Minimum}	The channel skew is set to the minimum supported value.
Equisample	The Channel skew is given by [(sampling rate)(number of channels)]$^{-1}$.
Manual	The channel skew is given by ChannelSkew.

P13-15.ClockSource

Values	Contec
{Internal}	The internal hardware clock is used.
External	The external sample clock.
Software	The computer clock is used.

P13-39.InputType

Values	Contec
Differential	Channels are configured for differential input.
{SingleEnded}	Channels are configured for single-ended input.

P13-94.TriggerCondition

Values	Contec

The following trigger conditions are available when TriggerType is HwDigital.

{PositiveEdge}	The trigger occurs when the positive (rising) edge of a digital signal is detected.
NegativeEdge	The trigger occurs when the negative (falling) edge of a digital signal is detected.

The following trigger conditions are available when TriggerType is HwAnalog.

{Both}	The trigger occurs when the analog signal has a positive or negative slope when passing through the specified range of values.
Rising	The trigger occurs when the analog signal has a positive slope when passing through the specified range of values.
Falling	The trigger occurs when the analog signal has a negative slope when passing through the specified range of values.
Leaving	The trigger occurs when the analog signal leaves the specified range of values.
Entering	The trigger occurs when the analog signal enters the specified range of values.

P13-106.TriggerRepeat

When using AI-1608AY-USB, TriggerRepeat cannot be set to a positive integer value while TriggerType is seto to HwDigital or HwAnalog. TriggerRepeat must be set to 0.

P13-108.TriggerType

Values	Contec
HwDigital	The trigger source is an external digital signal. Pretrigger data cannot be captured.
HwAnalog	The trigger source is an external analog signal (AI only).

3-3. Chapter 14, Device-Specific Properties – By Vendor

Contec	Properties	
Property Name	Description	Device Object
OutOfDataMode	Specify how the value held by the analog output subsystem is determined.	AO

4. Synchronization Method

By using CONTEC F-Series Multi-Function-Device, you can use synchronized start of analog input and analog output, and synchronized clock of analog input and analog output.
Supported Devices : AIO-163202F-PE, ADA16-32/2(PCI)F, ADA16-32/2(CB)F

4-1. Synchronization of AI start Trigger (master) and AO start trigger (slave)

This method is to start AI and AO at the same time by connecting AI start trigger to AO start trigger within a hardware circuit.

<sample code>

```
ai = analoginput('contec', 'AIO000')     ' create AI  object
ao = analogoutput('contec', 'AIO000')    ' create AO  object
addchannel(ai, 0)                        ' add AI channel
addchannel(ao, 0)                        ' add AO channel
data = sin(linspace(0.2*pi, 8000))';     ' create output data
putdata(ao, data)                        ' register output data
ao.TriggerType = 'HwSync'                ' use hardware synchronization trigger
ao.TriggerCondition = 'AiStart'          ' synchronize to AI start trigger
start(ao)                                ' start AO first
start(ai)                                ' start AI
```

4-2. Synchronization of AO start Trigger (master) and AI start trigger (slave)

This method is to start AO and AI at the same time by connecting AO start trigger to AI start trigger within a hardware circuit.

```
ai = analoginput('contec', 'AIO000')     ' create AI  object
ao = analogoutput('contec', 'AIO000')    ' create AO  object
addchannel(ai, 0)                        ' add AI channel
addchannel(ao, 0)                        ' add AO channel
data = sin(linspace(0.2*pi, 8000))';     ' create output data
putdata(ao, data)                        ' register output data
ai.TriggerType = 'HwSync'                ' use hardware synchronization trigger
ai.TriggerCondition = 'AoStart'          ' synchronize to AO start trigger
```

```
start(ai)                          ' start AI first
start(ao)                          ' start AO
```

4-3. Synchronization of AI Clock (master) and AO Clock (slave)

This method is to synchroniza AI clock and AO clock by connecting AI clock to AO clock within a hardware circuit.

<sample code>

```
ai = analoginput('contec', 'AIO000')    ' create AI  object
ao = analogoutput('contec', 'AIO000')   ' create AO  object
addchannel(ai, 0)                       ' add AI channel
addchannel(ao, 0)                       ' add AO channel
data = sin(linspace(0.2*pi, 8000))';    ' create output data
putdata(ao, data)                       ' register output data
ao.ClockSource = 'HwSync'               ' use hardware synchronization clock
ao.ClockCondition = 'AiClock'           ' synchronize to AI start clock
start(ao)                               ' start AO first
start(ai)                               ' start AI
```

4-4. Synchronization of AO Clock (master) and AI Clock (slave)

This method is to synchroniza AO clock and AI clock by connecting AO clock to AI clock within a hardware circuit.

<sample code>

```
ai = analoginput('contec', 'AIO000')    ' create AI  object
ao = analogoutput('contec', 'AIO000')   ' create AO  object
addchannel(ai, 0)                       ' add AI channel
addchannel(ao, 0)                       ' add AO channel
data = sin(linspace(0.2*pi, 8000))';    ' create output data
putdata(ao, data)                       ' register output data
ai.ClockSource = 'HwSync'               ' use hardware synchronization clock
ai.ClockCondition = 'AoClock'           ' synchronize to AO start clock
start(ai)                               ' start AI first
start(ao)                               ' start AO
```

5. ML-DAQ Version Upgrade History

Ver1.44 2009/10/09

Modified the problem that ML-DAQ could not be used on certain PC (Intel Core 2 Duo CPU Q9650 3.00GHz)

Modified the problem that 10MHz sample rate couldn't be specified with Modified the problem that 10MHz sample rate couldn't be specified with AI-1204Z-PCI.

Ver1.41 2009/07/03

Modified the problem that the error might occur when using trigger(ao).

Ver1.40 2008/11/14

Synchronized start of analog input and analog output, and synchronized clock of analog input and analog output are supported. (F-Series only)

Modified the problem that timeout error might occur.

Ver1.30 2008/03/28

AI-1204Z-PCI is supported.

Ver1.22c 2007/10/19 (There are no changes in mwcontec.dll)

AIO-121601E3-PE, AIO-121601UE3-PE, AIO-161601E3-PE, AIO-161601UE3-PE are supported.

Windows Vista is supported.

Ver1.22 2007/4/9

Modified the problem that exceptional data is acquired when using external trigger and repeat function.

Modified the problem about unusual operation when using analog input and digital 2input on Simulink.

Ver1.21 2006/11/13

Modified the problem about sampling clock error when using USB terminal.

Ver1.20b 2006/7/20 (There are no changes in mwcontec.dll)

AIO-160802L-LPE, AI-1616L-LPE, AO-1604L-LPE, AIO-160802AY-USB are supported.

Ver1.20a 2006/4/14 (There are no changes in mwcontec.dll)

AIO-163202F-PE, AIO-1608AY-USB, DA16-16(LPCI)L, DA16-8(LPCI)L are supported.

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 are supported.

Ver1.10 2005/09/09

ADAI16-8/2(LPCI)L, ADI16-16(LPCI)L, DAI16-4(LPCI)L are supported.

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) are supported.