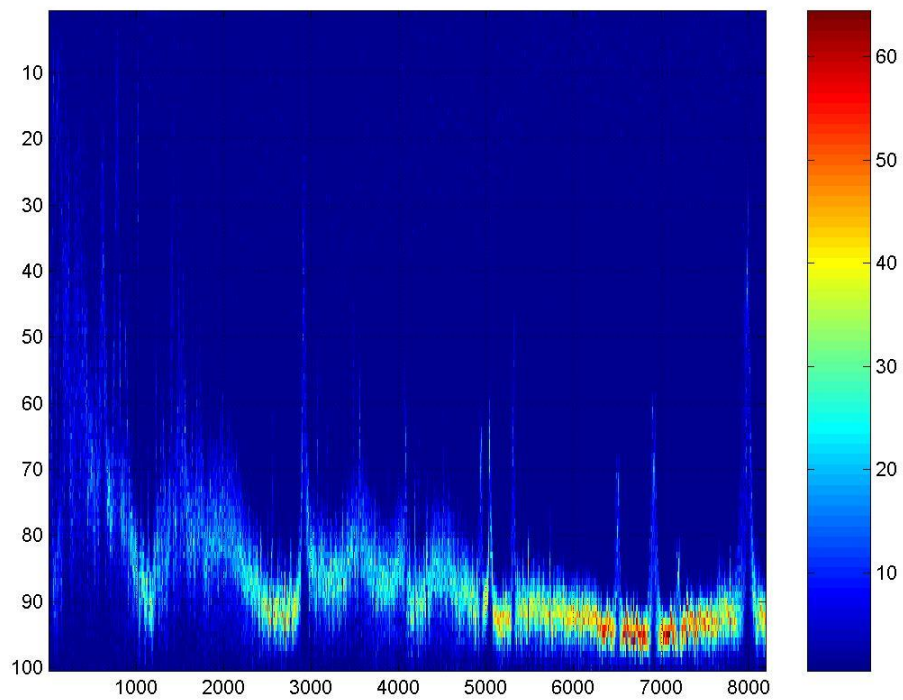


Sergio Frasca

Laboratorio di Sistemi e Segnali II

Esercitazioni

Dipartimento di Fisica
“Sapienza” Università di Roma



Versione 13 febbraio 2018

Versione aggiornata in <http://grwavsf.roma1.infn.it/lssii/lssii.pdf>

Contents

Esercitazioni di laboratorio	4
Usò della scheda Analog Discovery (AD) come acquisitore e generatore di segnali	4
Comandi base del Data Acquisition Toolbox	8
Le funzioni usate per il corso.....	10
Scheda AD – Test base	12
Scheda AD - Identificazione di dipoli	13
Scheda AD - Identificazione di sistemi	14
Scheda AD – Il rumore	15
Interoperabilità con Arduino.....	16
Analisi dati gravitazionali	17
Spettro e spettrogramma: studio del rumore.....	17
Filtro adattato: ricerca di segnali.....	18
Altre analisi.....	19
Esercitazioni a casa	20
Acquisizione e analisi dei segnali della scheda audio del computer: spettrogramma	20
Filtraggio dei segnali della scheda audio del computer	21
Acquisizione ed elaborazione dati da cellulari Android e iPhone	22
Appendice	23
Funzioni del Data Acquisition Toolbox.....	23

Esercitazioni di laboratorio

Uso della scheda Analog Discovery (AD) come acquirente e generatore di segnali

Il sistema di acquisizione che utilizzeremo in queste esercitazioni è la scheda Analog Discovery (https://reference.digilentinc.com/media/analog_discovery:analog_discovery_rm.pdf) della Digilent.

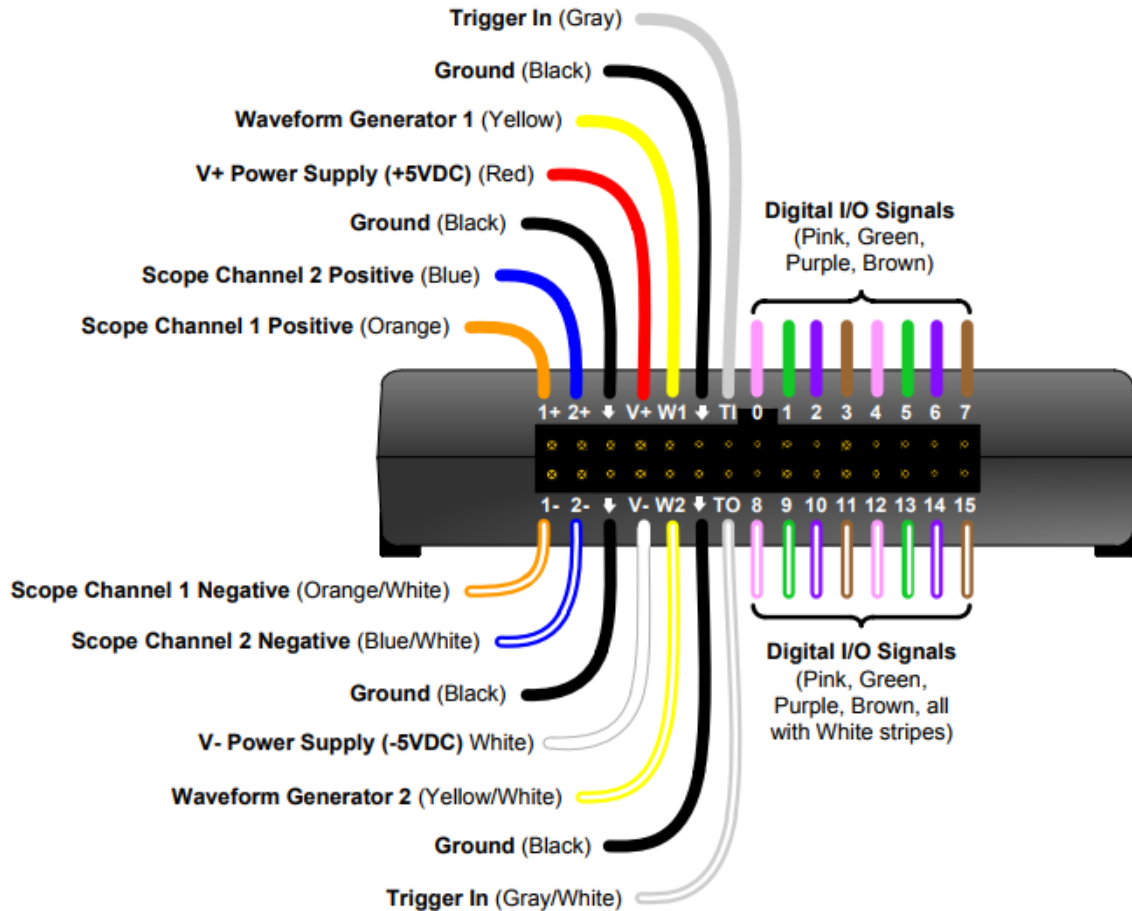
Essa ha due canali di acquisizione (ADC) ($1M\Omega$, $\pm 25V$, differenziali, 14 bit, 100Msample/s, banda di 5MHz) e due canali generatori di funzione arbitraria (DAC) (22Ω , $\pm 5V$, 14 bit, 100Msample/sec, banda di 5MHz), oltre a vari canali digitali di input-output che non utilizzeremo.



La scheda viene connessa al computer con un cavo USB.

I cavetti di connessione hanno il seguente schema:

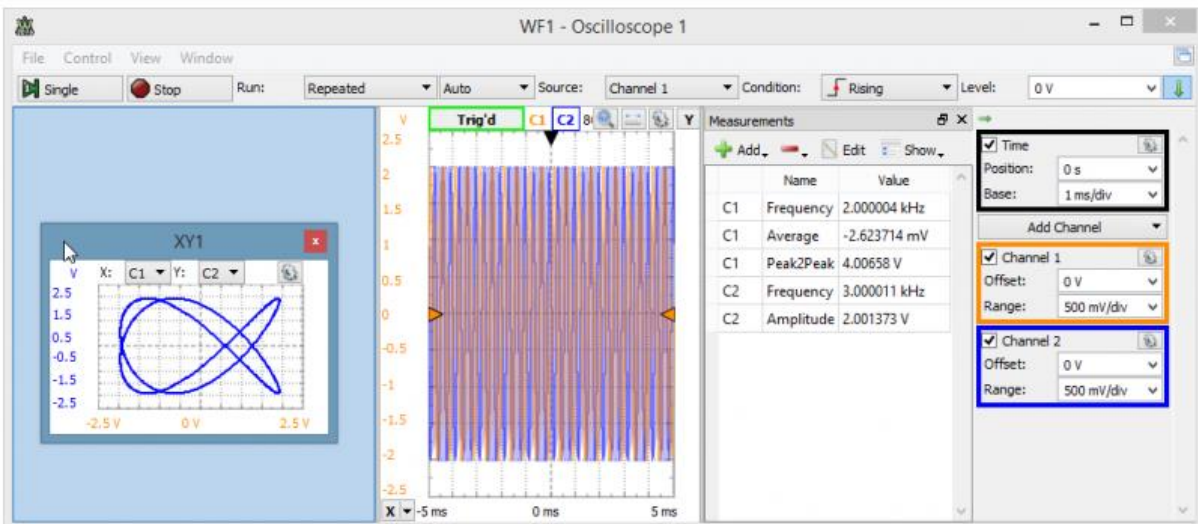
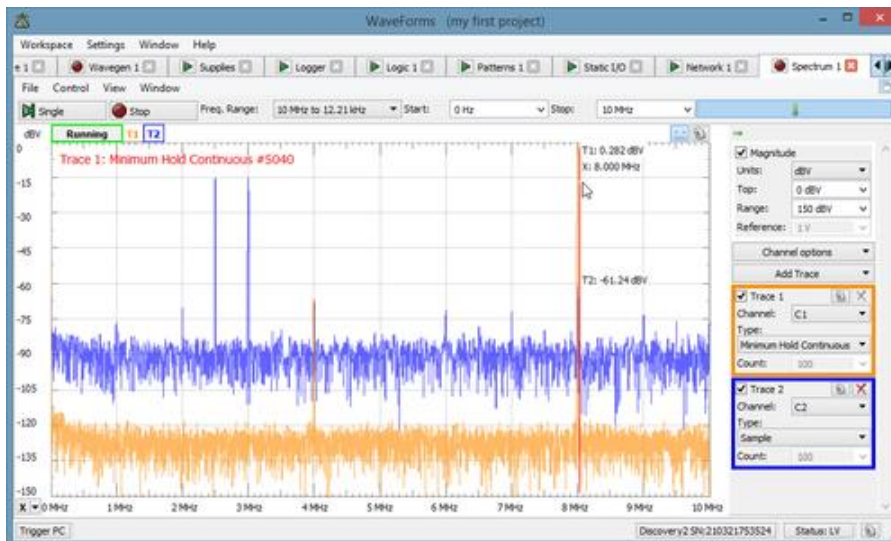
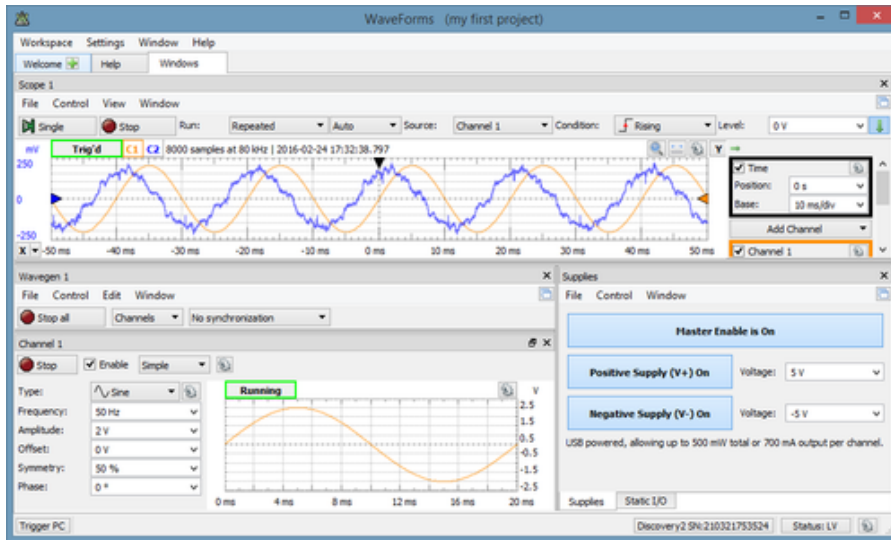
The Analog Discovery paired with the included 2x15 colored fly-wire:

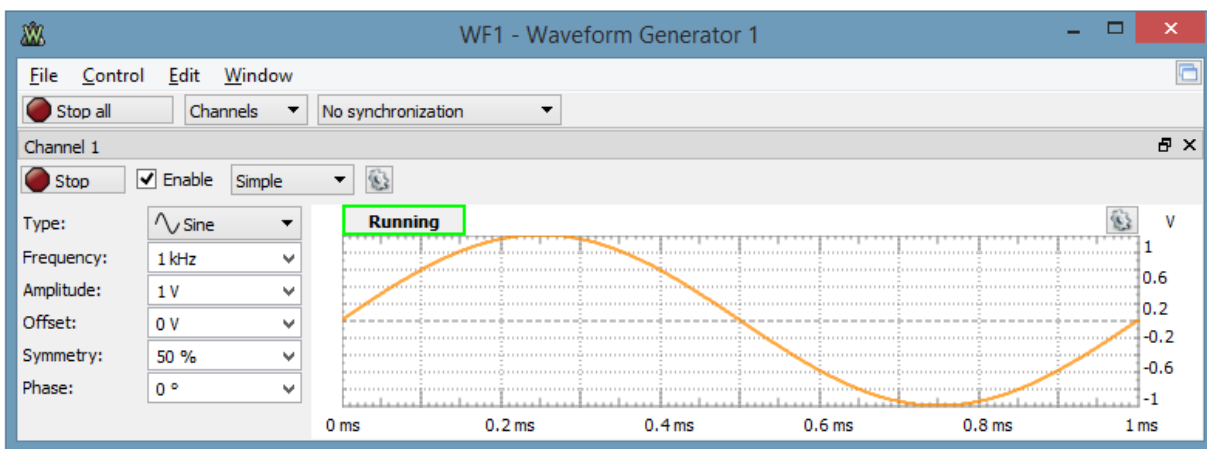
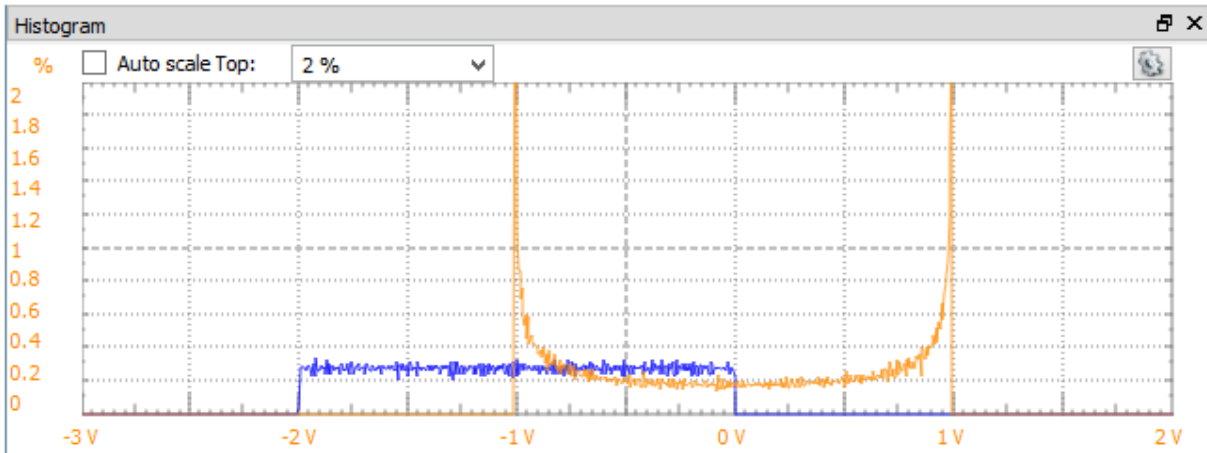


Notare che i due input sono differenziali (pin 1+ e 1- e pin 2+ e 2-), mentre le due uscite (W1 e W2) sono rispetto a massa (Ground, ce ne sono quattro equivalenti con cavetti neri). Gli input “-” possono essere messi a massa, se occorre.

Col software Digilent Waveforms si ottiene un oscilloscopio a due canali con molte interessanti features, come l’analizzatore di spettro, lo spettrogramma, xy, l’istogramma e inoltre un generatore di segnali.

In <https://reference.digilentinc.com/reference/software/waveforms/waveforms-3/start?redirect=1#documentation> c’è la documentazione.





Se si installa da Matlab (dal pulsante add-ons, per esempio) il **Data Acquisition Toolbox Support Package for Digilent Analog Discovery Hardware**, si possono gestire i dati in input e in output alla scheda direttamente da Matlab. Potremo quindi utilizzare le funzioni del Matlab toolbox Data Acquisition (vedere in <https://it.mathworks.com/products/daq.html>).

Comandi base del Data Acquisition Toolbox

Il Data Acquisition Toolbox permette di sviluppare applicazioni anche molto complesse con una varietà di sistemi di acquisizione.

Il manuale (user guide) si può scaricare da

https://it.mathworks.com/help/pdf_doc/daq/daqug.pdf , ed è piuttosto corposo. In appendice c'è l'elenco di tutte le funzioni, tuttavia, per il nostro uso elementare solo poche funzioni sono essenziali.

Vediamoli:

>> **daq.getDevices** che elenca i vari sistemi di acquisizione, gestibili da Matlab, presenti nel computer

ans =

digilent: Digilent Inc. Analog Discovery Kit Rev. C (Device ID: 'AD1')

Analog input subsystem supports:

-2.5 to +2.5 Volts,-25 to +25 Volts ranges

Rates from 0.1 to 1000000.0 scans/sec

2 channels ('1','2')

'Voltage' measurement type

Analog output subsystem supports:

-5.0 to +5.0 Volts range

Rates from 0.0 to 1000000.0 scans/sec

2 channels ('1','2')

'Voltage' measurement type

Function generator subsystem supports:

-5.0 to +5.0 Volts range

Rates from 0.0 to 25000000.0 scans/sec

2 channels ('1','2')

'Sine','Square','Triangle','RampUp','RampDown','DC','Arbitrary' measurement types

>> **s = daq.createSession('digilent')** % create a data acquisition session

s =

Data acquisition session using Digilent Inc. hardware:

Will run for 1 second (10000 scans) at 10000 scans/second.

No channels have been added.

>> **s.DurationInSeconds = 10** % Configure session properties

s =

Data acquisition session using Digilent Inc. hardware:
Will run for 10 seconds (100000 scans) at 10000 scans/second.
Number of channels: 1

index	Type	Device	Channel	MeasurementType	Range	Name
1	ai	AD1	1	Voltage (Diff)	-25 to +25 Volts	

>> **ch = addAnalogInputChannel(s,'AD1', 1, 'Voltage')** % Add channels to the session

ch =

Data acquisition analog input voltage channel '1' on device 'AD1':

Coupling: DC
TerminalConfig: Differential
Range: -25 to +25 Volts
Name: ''
ID: '1'
Device: [1x1 daq.di.DeviceInfo]
MeasurementType: 'Voltage'

>> **daqreset**

Le funzioni usate per il corso

Al fine di semplificare l'uso del toolbox sono state create alcune funzioni:

Funzioni base

- **daq_set** che crea la sessione e si aggiungono i canali di input e output. C'è anche un "work-around" per risolvere un problema del sistema (un artefatto iniziale dell'acquisizione)

```
function [s,chin,chout]=daq_set(typ,nch)
% setup daq
%
% typ    1 = 'digilent'
% nch    [nchin nchout] number of channels input (ADC) and output (DAC)
```

- **daq_input** per acquisire dati per un certo intervallo di tempo

```
function [data,times,s,triggerTime,trigDate]=daq_input(s,sf,tt,rg)
% data acquisition
%
% s      session (set by daq_set)
% sf     sampling frequency
% tt     total time (in seconds)
% rg     range [min max] (def [-25 25] V)
```

- **daq_output** per mandare in output dei DAC dei segnali

```
function s=daq_output(s,datout,sf,rg)
% data acquisition
%
% s      session (set by daq_set)
% sf     sampling frequency
% rg     range [min max] (def [-25 25] V)
```

- **daq_inpout** per mandare in output e contemporaneamente acquisire dati per tutto il periodo dell'output

```
function [data,times,s,triggerTime,trigDate]=daq_inpout(s,datout,sf)
% data acquisition
%
% s      session (set by daq_set)
% datout data in output
% sf     sampling frequency
```

- **daq_cont_inp** continuazione di un'acquisizione iniziata

```
function [data,times]=daq_cont_inp(typ,sf,tt,n,rg)
% continuing data acquisition
```

```

%
% typ    1 = 'diligent'
% sf     sampling frequency
% tt     total time (in seconds)
% n      number of chunks
% rg     range [min max] (def [-25 25] V

```

- **daq_cont_inpout** continuazione di input-output di dati

```

function [data,times]=daq_cont_inpout(typ,datout,sf,n,rg)
% continuing data input-output
%
% typ    1 = 'diligent'
% datout length of datout must be even
% sf     sampling frequency
% n      number of chunks
% rg     range [min max] (def [-25 25] V

```

- **daq_reset** simply resetta l'acquisizione
- **daq_close** deleta (cancella)

Alcune funzioni applicative

- **daq_calib** per la calibrazione
- **daq_charac** e **daq_charac2** usate per le curve caratteristiche
- **daq_diode** per le caratteristiche dei diodi
- **daq_noise** per osservare l'uscita a un sistema eccitato da rumore
- **daq_sweep** per osservare l'uscita a un sistema eccitato da segnale con frequenza variabile
- **daq_testfr** per misurare la risposta in frequenza e i ritardi

Scheda AD – Test base

Quando si usa un sistema di acquisizione è bene aver presente quali sono gli eventuali errori sistematici e i disturbi che possono essere aggiunti al segnale dal sistema stesso.

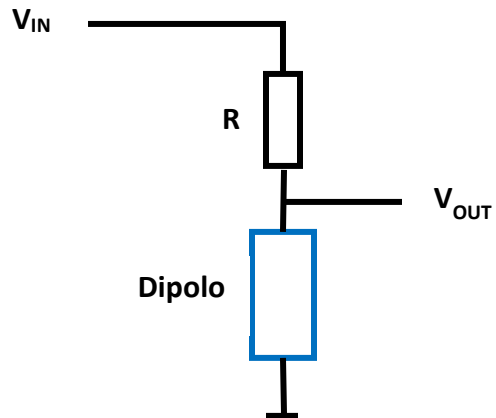
In questa esperienza cercheremo di caratterizzare questi aspetti del sistema Analog Discovery.

Procederemo osservando l'output con l'oscilloscopio, quindi vedremo come viene acquisito il segnale prodotto, come ne viene eventualmente distorta l'ampiezza, se ci sono distorsioni in frequenza e valuteremo gli eventuali ritardi.

Scheda AD - Identificazione di dipoli

In questa esperienza cercheremo di caratterizzare dipoli, come per esempio diodi, resistenze, capacità, o semplici reti dipolari.

Lo schema tipico è il seguente:



Scheda AD - Identificazione di sistemi

Scheda AD – Il rumore

Interoperabilità con Arduino

Analisi dati gravitazionali

Spettro e spettrogramma: studio del rumore

Filtro adattato: ricerca di segnali

Altre analisi

Esercitazioni a casa

Acquisizione e analisi dei segnali della scheda audio del computer: spettrogramma

Filtraggio dei segnali della scheda audio del computer

Acquisizione ed elaborazione dati da cellulari Android e iPhone

Appendice

Funzioni del Data Acquisition Toolbox

Hardware Discovery and Session Setup

<code>daq.createSession</code>	Create data acquisition session for specific vendor hardware
<code>daq.getDevices</code>	Display available data acquisition devices
<code>daq.getVendors</code>	Display available vendors
<code>release</code>	Release session resources
<code>prepare</code>	Prepare session for operation
<code>daqreset</code>	Reset Data Acquisition Toolbox
<code>daqhelp</code>	Help for toolbox interface

Analog Input and Output

Analog Data Acquisition

<code>addAnalogInputChannel</code>	Add analog input channel
<code>inputSingleScan</code>	Acquire single scan from all input channels
<code>startForeground</code>	Start foreground operations
<code>startBackground</code>	Start background operations
<code>removeChannel</code>	Remove channel from session object
<code>addListener</code>	Create event listener
<code>stop</code>	Stop background operation
<code>wait</code>	Block MATLAB until background operation completes

Digital Input and Output

<code>addDigitalChannel</code>	Add digital channel
<code>inputSingleScan</code>	Acquire single scan from all input channels
<code>outputSingleScan</code>	Generate single scan on all output channels
<code>startForeground</code>	Start foreground operations
<code>startBackground</code>	Start background operations
<code>addClockConnection</code>	Add clock connection
<code>stop</code>	Stop background operation
<code>wait</code>	Block MATLAB until background operation completes
<code>decimalToBinaryVector</code>	Convert decimal value to binary vector
<code>binaryVectorToDecimal</code>	Convert binary vector value to decimal value
<code>hexToBinaryVector</code>	Convert hexadecimal value to binary vector
<code>binaryVectorToHex</code>	Convert binary vector value to hexadecimal

Counter and Timer Input and Output

<code>addCounterInputChannel</code>	Add counter input channel
<code>addCounterOutputChannel</code>	Add counter output channel
<code>inputSingleScan</code>	Acquire single scan from all input channels
<code>resetCounters</code>	Reset counter channel to initial count
<code>startForeground</code>	Start foreground operations
<code>startBackground</code>	Start background operations
<code>removeChannel</code>	Remove channel from session object
<code>stop</code>	Stop background operation
<code>wait</code>	Block MATLAB until background operation completes

Multichannel Audio Input and Output

<code>addAudioInputChannel</code>	Add audio input channel
<code>addAudioOutputChannel</code>	Add audio output channel
<code>startForeground</code>	Start foreground operations
<code>startBackground</code>	Start background operations
<code>removeChannel</code>	Remove channel from session object
<code>stop</code>	Stop background operation
<code>wait</code>	Block MATLAB until background operation completes

Periodic Waveform Generation

<code>addFunctionGeneratorChannel</code>	Add function generator channel
--	--------------------------------

Simultaneous and Synchronized Operations

<code>addTriggerConnection</code>	Add trigger connection
<code>addClockConnection</code>	Add clock connection
<code>removeConnection</code>	Remove clock or trigger connection

Build Custom Adaptors

<code>buildAdaptor</code>	Build adaptor for third-party data acquisition interface
<code>enableDemoAdaptorDiscovery</code>	Allow SDK demo adaptor to be enabled for device discovery and usage