

ADLINK Technical Document

Abstract	How to use the Dynamic Signal Analyzer in MATLAB		
OS	Windows		
Keyword	MAPS, DSA-DASK		
Related Products	PCI-9527, PCI-9527L, PXI-9527, PCI-9529, PXIe-9529		
Date	2021-09-10	No.	202110006

- Overview:

The DAQ module requires a third-party compiler installed on your system to control the DAQ card from MATLAB[®] correctly. This document outlines the compiler setup process and how to download sample code for MATLAB.

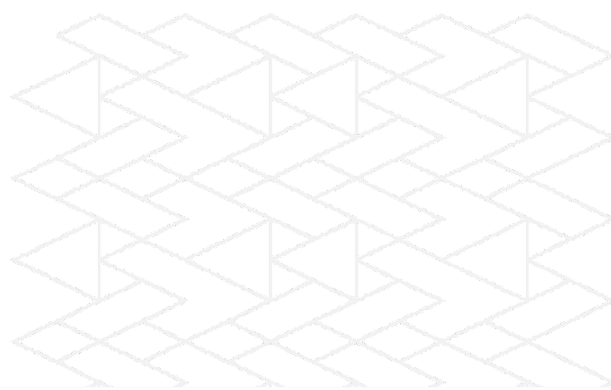
NOTE: The MathWorks[®] Data Acquisition Toolbox[™] is widely used to connect to data acquisition hardware and read data into MATLAB (also a MathWorks product). ADLINK does not provide this tool, so it requires an alternative compiler to install the DLLs needed to control the DAQ card with MATLAB.

- Prerequisites:

Install MAPS Core or DSA-DASK.

- Solution:

The steps below show how to set up the correct environment and use the code samples in MATLAB.



Step 1:

Go to this link: http://www.mathworks.com/support/sysreq/previous_releases.html

Step 2:

In the “Release” column, find the correct MATLAB version installed on your system.

Previous Releases: System Requirements and Supported Compilers

Release	Windows	Linux	Mac	Solaris/UNIX	Supported Compilers	Platform Availability
R2021a (MATLAB 9.10)	Details	Details	Details	N/A	Details	Details
R2020b (MATLAB 9.9)	Details	Details	Details	N/A	Details	Details
R2020a (MATLAB 9.8)	Details	Details	Details	N/A	Details	Details
R2019b (MATLAB 9.7)	Details	Details	Details	N/A	Details	Details
R2019a (MATLAB 9.6)	Details	Details	Details	N/A	Details	Details
R2018b (MATLAB 9.5)	Details	Details	Details	N/A	Details	Details
R2018a (MATLAB 9.4)	Details	Details	Details	N/A	Details	Details
R2017b (MATLAB 9.3)	Details	Details	Details	N/A	Details	Details
R2017a (MATLAB 9.2)	Details	Details	Details	N/A	Details	Details
R2016b (MATLAB 9.1)	Details	Details	Details	N/A	Details	Details
R2016a (MATLAB 9.0)	Details	Details	Details	N/A	Details	Details
R2015b (MATLAB 8.6)	Details	Details	Details	N/A	Details	N/A

Step 3:

In the “Supported Compilers” column, click the “Details” link in the corresponding row to your MATLAB version.

Previous Releases: System Requirements and Supported Compilers

Release	Windows	Linux	Mac	Solaris/UNIX	Supported Compilers	Platform Availability
R2021a (MATLAB 9.10)	Details	Details	Details	N/A	Details	Details
R2020b (MATLAB 9.9)	Details	Details	Details	N/A	Details	Details
R2020a (MATLAB 9.8)	Details	Details	Details	N/A	Details	Details
R2019b (MATLAB 9.7)	Details	Details	Details	N/A	Details	Details
R2019a (MATLAB 9.6)	Details	Details	Details	N/A	Details	Details
R2018b (MATLAB 9.5)	Details	Details	Details	N/A	Details	Details
R2018a (MATLAB 9.4)	Details	Details	Details	N/A	Details	Details
R2017b (MATLAB 9.3)	Details	Details	Details	N/A	Details	Details
R2017a (MATLAB 9.2)	Details	Details	Details	N/A	Details	Details
R2016b (MATLAB 9.1)	Details	Details	Details	N/A	Details	Details
R2016a (MATLAB 9.0)	Details	Details	Details	N/A	Details	Details

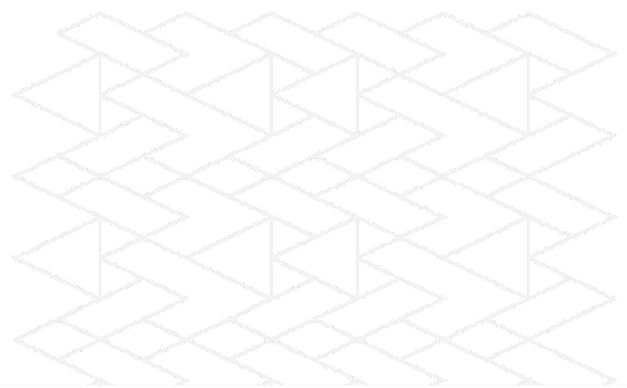
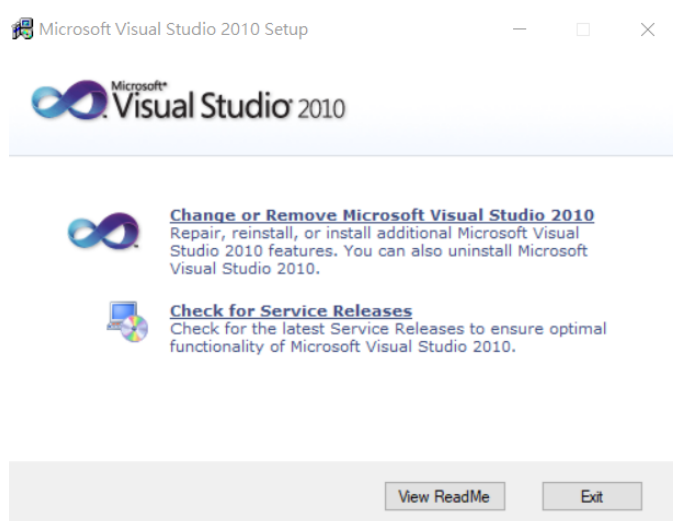
Step 4:

Install a recommended compiler according to your MATLAB version.

MATLAB Product Family – Release 2021a									
Compiler	MATLAB For MEX-file compilation, loadlibrary, C++ interface, and external usage of MATLAB Engine and MATLAB APIs	MATLAB Coder For all features	GPU Coder For all features	SimBiology For accelerated computation	Fixed-Point Designer For accelerated computation	HDL Coder For accelerated testbench simulation	HDL Verifier For DPI and TLM component generation	Audio Toolbox For validating and generating audio plugins	ROS Toolbox For custom messages and code generation
MinGW 6.3 C/C++ (Distributor: <i>mingw-w64</i>) Download Now Available at no charge	✓	✓		✓	✓	✓	✓		
Microsoft Visual C++ 2019 product family	✓	✓	✓	✓	✓			✓	
Microsoft Visual C++ 2017 product family ¹⁰	✓	✓	✓	✓	✓	✓	✓	✓	✓
Microsoft Visual C++ 2015 Professional ⁹	✓	✓	✓	✓	✓	✓	✓	✓	
Intel Parallel Studio XE 2020 for C/C++ ⁵	✓	✓		✓	✓				
Intel Parallel Studio XE 2019 for C/C++ ⁵	✓	✓		✓	✓				
Intel Parallel Studio XE 2018 for C/C++ ⁵	✓	✓		✓	✓				
Intel Parallel Studio XE 2020 for Fortran ⁵	✓			✓					

Step 5:

Install the compiler if not installed already. In this case, Microsoft Visual C++ 2010.



Step 6:

Launch MATLAB. Enter the “mex-setup” command to begin the default compiler setup process. Press “y” to automatically locate the installed compilers.

```
>> mex -setup
```

```
Welcome to mex -setup. This utility will help you set up  
a default compiler. For a list of supported compilers, see  
http://www.mathworks.com/support/compilers/R2013b/win64.html
```

```
Please choose your compiler for building MEX-files:
```

```
fx Would you like mex to locate installed compilers [y]/n?
```

Step 7:

A numbered list of available compilers is displayed. Type the number of the preferred compiler, e.g., “1”. Press “y” to confirm.

```
Would you like mex to locate installed compilers [y]/n? y
```

```
Select a compiler:
```

```
[1] Microsoft Visual C++ 2010 in C:\Program Files (x86)\Microsoft Visual Studio 10.0
```

```
[2] Microsoft Visual C++ 2008 SP1 in C:\Program Files (x86)\Microsoft Visual Studio 9.0
```

```
[0] None
```

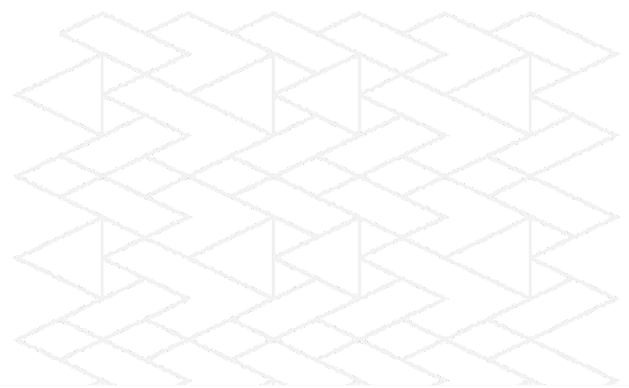
```
Compiler: 1
```

```
Please verify your choices:
```

```
Compiler: Microsoft Visual C++ 2010
```

```
Location: C:\Program Files (x86)\Microsoft Visual Studio 10.0
```

```
fx Are these correct [y]/n? y
```



Step 8:

Check the results after the updates are complete.

```
Are these correct [y]/n? y

*****
Warning: MEX-files generated using Microsoft Visual C++ 2010 require
        that Microsoft Visual Studio 2010 run-time libraries be
        available on the computer they are run on.
        If you plan to redistribute your MEX-files to other MATLAB
        users, be sure that they have the run-time libraries.
*****

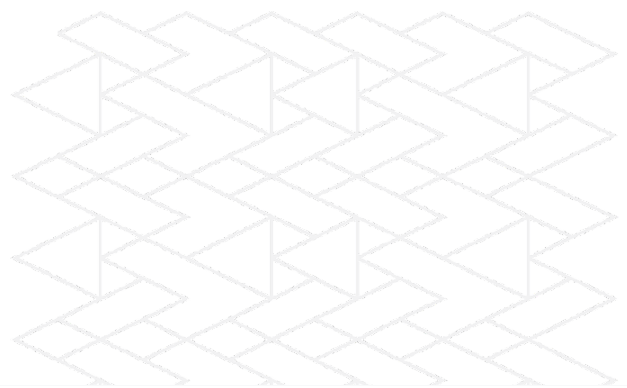
Trying to update options file: C:\Users\TEST\AppData\Roaming\MathWorks\MATLAB\R2013b\mexopts.bat
From template:                C:\PROGRA~1\MATLAB\R2013b\bin\win64\mexopts\msvc100opts.bat

Done . . .
```

Step 9:

Download additional ADLINK MATLAB samples from the link below.

Link: https://ftp.adlinktech.com/daq/dsa_dask_matlab.zip

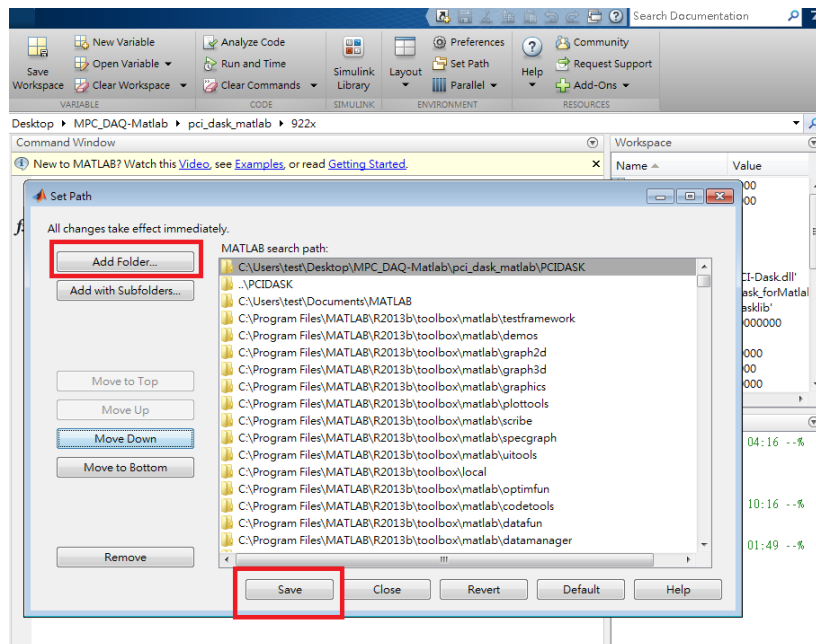


Step 10:

Extract the contents of the zip file to a folder. To set up this new folder in MATLAB:

1. Open *setpath*
2. Click “Add Folder”
3. Select the folder in the MATLAB search path area
4. Click “Save”

The M file in the selected folder is now accessible from within MATLAB.



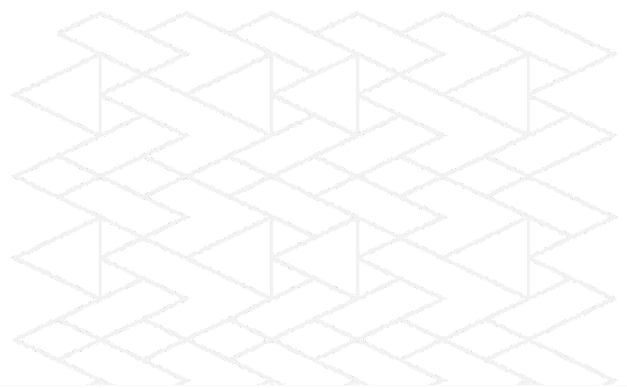
Step 11:

View the contents of the sample M file through the Editor. Modify the contents as needed. The contents of the sample M file are shown below.

```

error = calllib(LIB, 'DSA_AI_ContReadChannel', card, Channel, 0, bufferID0, AI_ReadCount, 0, SyncMode);
if error < 0
    calllib(LIB, 'DSA_AI_ContBufferReset', card);
    calllib(LIB, 'DSA_Release_Card', card);
    unloadlibrary(LIB);
    fprintf('DSA_AI_ContReadChannel failed with error code %d\n', error);
    return;
end
tic; % Set the Time
margin = 2; % margin in seconds for the TimeOut
TimeOut = double(AI_ReadCount)/SampleRate + margin; % Acquisition time in seconds (plus margin)
TimeLeft = TimeOut;
fprintf('Start AI, press anykey on figure to stop\n');
index = 0;
%Here is like kbhit() in C code , press anykey to exit loop
figh = figure('KeyPressFcn', @(obj,ev) set(obj, 'userdata', 1));
while isempty(get(figh, 'userdata')) && TimeLeft >= 0
    TimeLeft = TimeOut - toc;
    [error, HalfReady, Stopped] = calllib(LIB, 'DSA_AI_AsyncDblBufferHalfReady', card, HalfReady, Stopped);
    if error < 0
        calllib(LIB, 'DSA_AI_AsyncClear', card, AccessCnt);
        calllib(LIB, 'DSA_AI_ContBufferReset', card);
        calllib(LIB, 'DSA_Release_Card', card);
        unloadlibrary(LIB);
        fprintf('DSA_AI_AsyncDblBufferHalfReady failed with error code %d\n', error);
        return;
    end
    if HalfReady == 1
        tic;
        TimeLeft = TimeOut; %reset TimeLeft for next buffer
    end
end

```



Step 12:

Type the filename to run the M file, e.g., “PCI_9527_AI_DMA_DB.m”, at the MATLAB command window prompt. The result is shown below. The card worked correctly and returned the data into MATLAB.

