ADLINK Technical Document

Abstract	How to use USB-DAQ in MATLAB		
OS	Windows		
Keyword	MAPS, USB -DAQ, UD-DASK		
Related Products	USB-1901, USB-1902, USB-1903, USB-2401, USB-2405, USB-7230, USB-7250		
Date	2021-07-16	No.	202110004

- 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 UD-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 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, in the row corresponding to the MATLAB version, click the "Details" link.

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 4:

Install a recommended compiler according to your MATLAB version.

MATLAB Product Family – Release 2021a									
Compiler	MATLAB	MATLAB Coder	GPU Coder	SimBiology	Fixed-Point Designer	HDL Coder	HDL Verifier	Audio Toolbox	ROS Toolbox
	For MEX-file compilation, loadlibrary, C++ interface, and external usage of MATLAB Engine and MAT-file APIs	For all features	For all features	For accelerated computation	For accelerated computation	For accelerated testbench simulation	For DPI and TLM component generation	For validating and generating audio plugins	For custom messages and code generation
MinGW 6.3 C/C++ (Distributor: *mingw-w64*) *Download Now* Available at no charge	✔	✔		✔	✔	✔	✔		
Microsoft Visual C++ 2019 product family	✔	✔	✔	✔	✔			✔	
Microsoft Visual C++ 2017 product family [10]	✔	✔	✔	✔	✔	✔	✔	✔	✔
Microsoft Visual C++ 2015 Professional [8]	✔	✔	✔	✔	✔	✔	✔	✔[1]	
Intel Parallel Studio XE 2020 for C/C++ [3]	✔	✔		✔	✔				
Intel Parallel Studio XE 2019 for C/C++ [3]	✔	✔		✔	✔				
Intel Parallel Studio XE 2018 for C/C++ [3]	✔	✔		✔	✔				
Intel Parallel Studio XE 2020 for Fortran [3]	✔			✔					

Step 5:

Install the compiler if not installed already. In this case, Microsoft Visual C++ 2010.
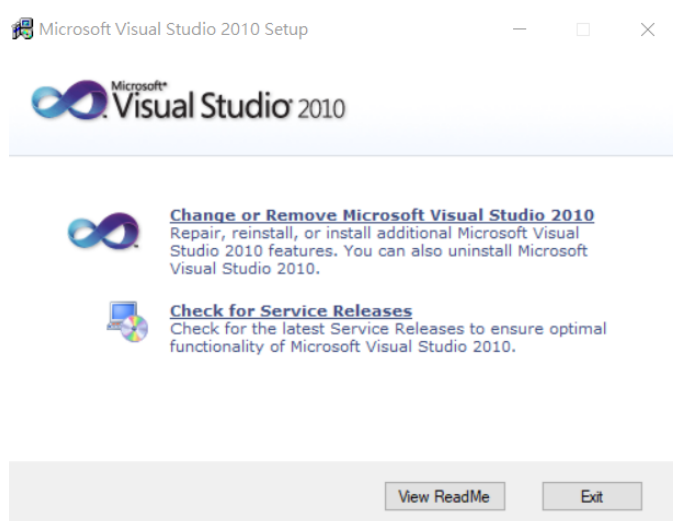
Step 6:

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

Step 7:

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

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.

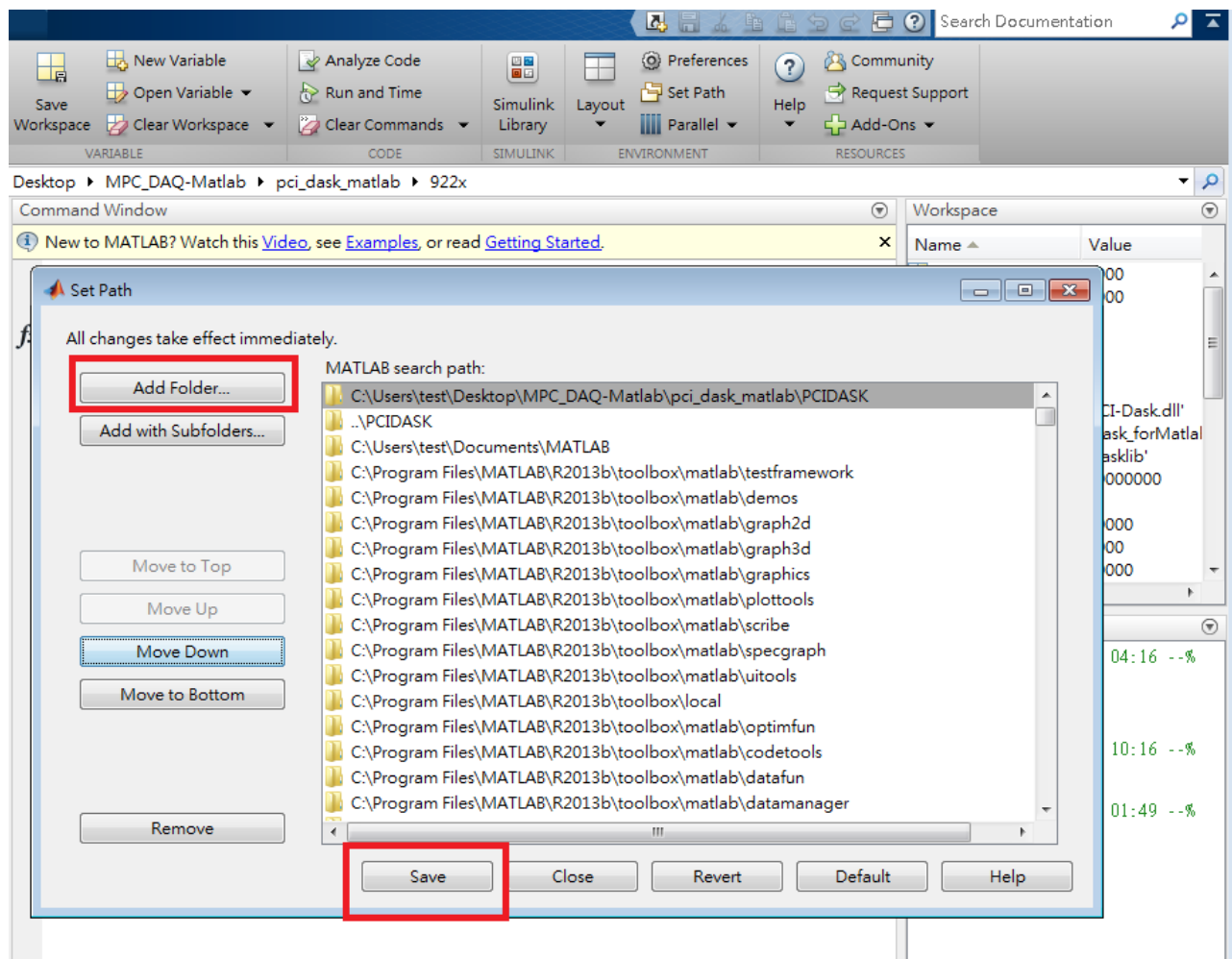Link: https://ftp.adlinktech.com/daq/ud_dask_matlab.zip

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 for your test if necessary. The contents of the sample M file are shown below.

```
error = calllib(LIB,'UD_AI_1902_Config',card,ConfigCtrl,TrigCtrl,TriggerLvel,ReTriggerCount,DelayCount);
if error < 0
    calllib(LIB,'UD_Release_Card',card);
    unloadlibrary(LIB);
    fprintf('UD_AI_1902_Config failed with error code %d\n',error);
    return;
end

error = calllib(LIB,'UD_AI_AsyncDblBufferMode',card,bEnable);
if error < 0
    calllib(LIB,'UD_Release_Card', card);
    unloadlibrary(LIB);
    fprintf('UD_AI_AsyncDblBufferMode failed with error code %d\n',error);
    return;
end
```

Step 12:

Type the filename to run the M file, e.g., "USB_1902_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.