Arduino MATLAB Interface Made Easy

In this project, we will show you how the Arduino MATLAB Interface works and all the steps involved in interfacing Arduino with MATLAB.

Arduino is one of the favorite prototyping platform among makers and hobbyists. MATLAB, on the other hand, is a professional numerical computing tool used by engineers and scientists. Interfacing Arduino with MATLAB makes things interesting as one is aimed at an entry and hobbyist level of users while the other is a very complex tool used by professionals.

How Arduino MATLAB Interface Works?

We normally use Arduino IDE to write and upload programs (or Sketches) to Arduino. This is fine for regular use but if you want to plot the data obtained from sensors, then the programming becomes a time consuming task.

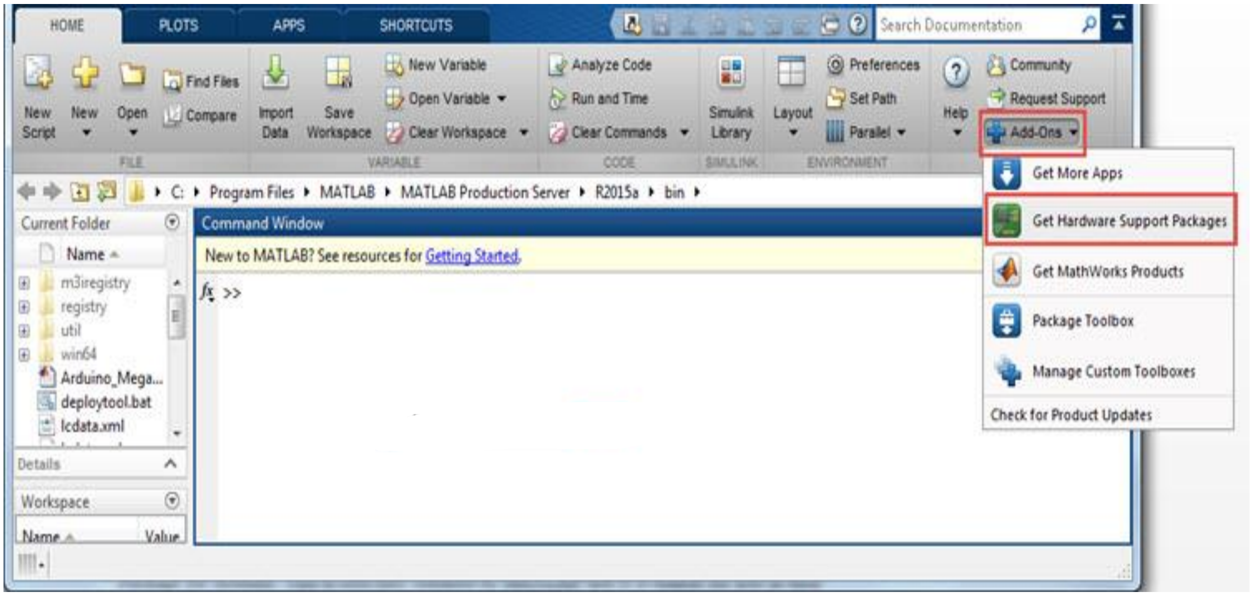
MATLAB addresses several such problems in Arduino Programming with the help of two types of processes:

- Reading, writing and analyzing data from Sensors
- Developing Algorithms that run as standalone program on Arduino

In order to achieve this, we need an extra package for MATLAB called MATLAB Support Package for Arduino. This is officially released by MATLAB and is available for free in their website (you need to have an account to download this package).

Installing MATLAB Support Package for Arduino

Now we'll see how to install the MATLAB Support package for Arduino. First, start the MATLAB Application and in the RESOURCES Tab, go to Add-ons and select Get Hardware Support Packages option.



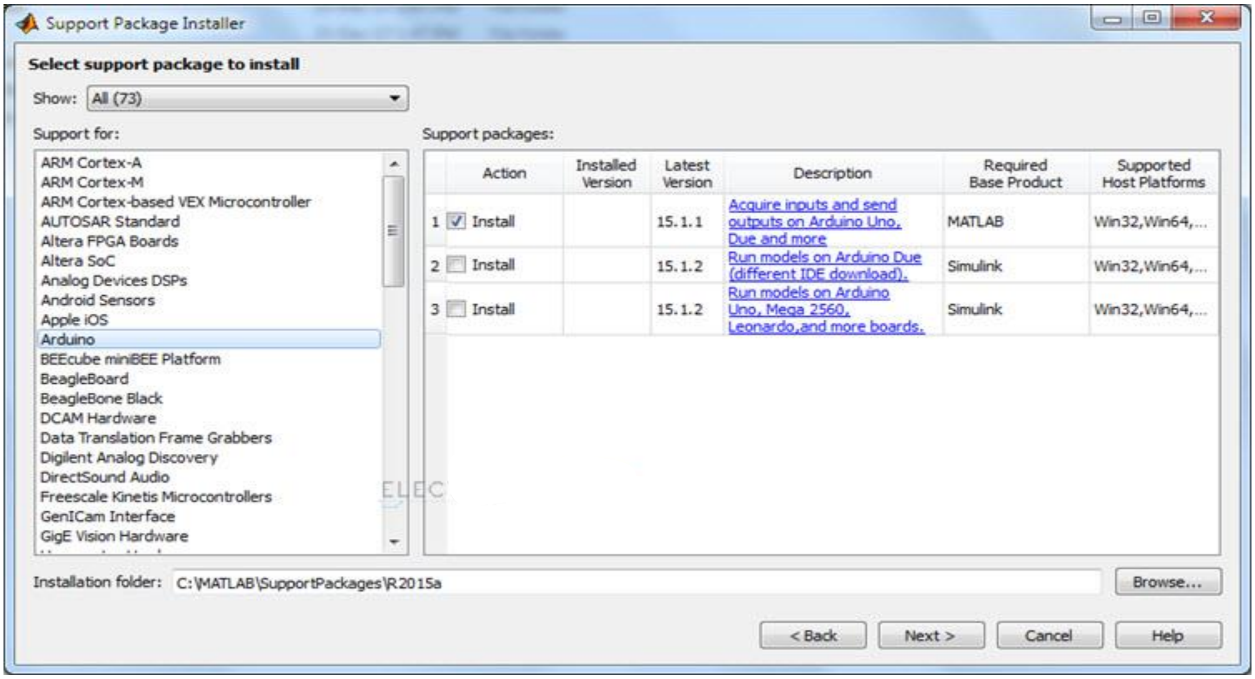
C: ▸ Program Files ▸ MATLAB ▸ MATLAB Production Server ▸ R2015a ▸ bin ▸

New to MATLAB? See resources for Getting Started.

```
fx >>
```

After initializing the add-on, you will then get an options of how to install the package i.e. from the internet or from a file already that is already downloaded. We have chosen to install from the internet.

You will then get a list of all the available Support Packages. Select Arduino and check the MATLAB Support Package. There are Arduino Support Packages for Simulink as well but we will leave it for later.



	Action	Installed Version	Latest Version	Description	Required Base Product	Supported Host Platforms
1	☑ Install		15.1.1	Acquire inputs and send outputs on Arduino Uno, Due and more	MATLAB	Win32,Win64,...
2	□ Install		15.1.2	Run models on Arduino Due (different IDE download).	Simulink	Win32,Win64,...
3	□ Install		15.1.2	Run models on Arduino Uno, Mega 2560, Leonardo,and more boards.	Simulink	Win32,Win64,...

Installation folder: C:\MATLAB\SupportPackages\R2015a

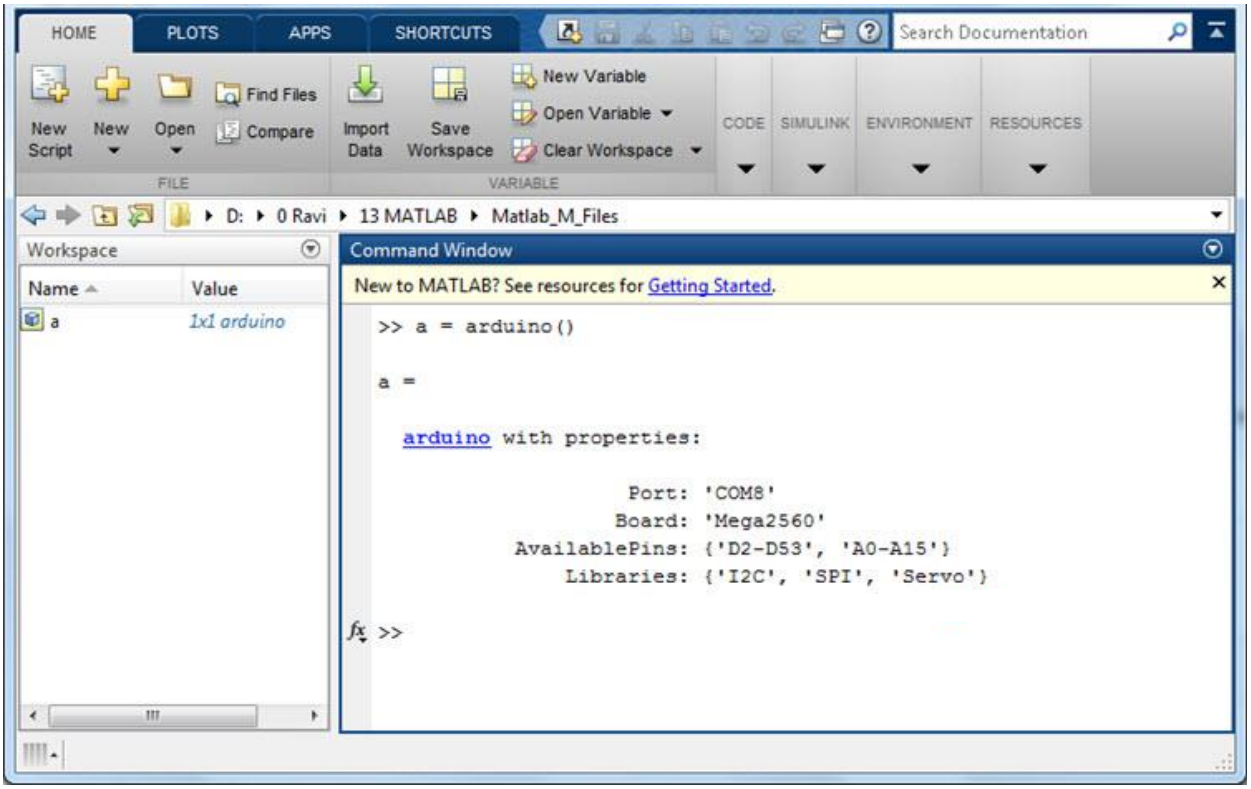
Now, the MATLAB Application will ask you to login to you MathWorks Account. If you have an account, use those credentials and login or you can create an account during installation. Next, accept the license terms and continue to download the package.

MATLAB will now download and install the selected packages. Sit back and relax until it is done.

Interfacing Arduino with MATLAB and Testing the Support Package
After installing the Arduino Package for MATLAB, we will now see how to interface Arduino and MATLAB. Connect the Arduino board to the computer that is running MATLAB and open MATLAB Application.

In the command window, type the following and hit enter.

a=arduino()
This step will create a connection between MATLAB and Arduino and returns with the properties of the Arduino Board like COM Port, Name, etc. as shown in the screenshot below.



Name	Value
a	1x1 arduino

New to MATLAB? See resources for Getting Started.

```
>> a = arduino()

a =

  arduino with properties:

                  Port: 'COM8'
                 Board: 'Mega2560'
         AvailablePins: {'D2-D53', 'A0-A15'}
             Libraries: {'I2C', 'SPI', 'Servo'}

>>
```

If more than one Arduino board is connected to the computer and if you want to communicate with a specific board, use the following command.

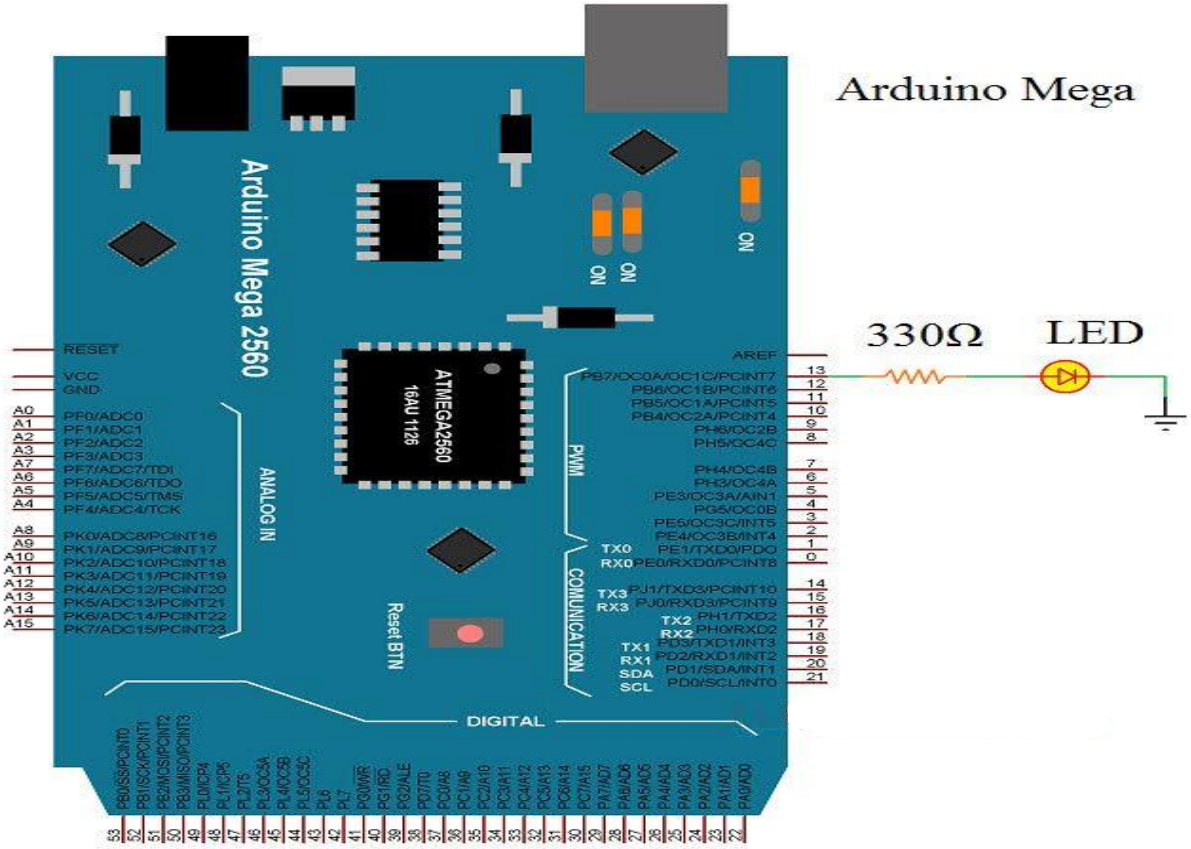
```
a=arduino(‘com8’,’Mega2560’)
```

NOTE: You have to clear the variable ‘a’ using the command ‘clear a’.

Programming Arduino using MATLAB

We will now see how to program the Arduino Board using MATLAB Programming. For this, we will be using the Arduino Mega Board. It has an on-board LED connected to Digital Pin 13.

We will write a MATLAB Program in order to blink this LED for 10 times with a delay of 1 second. Alternatively, you can connect an external LED with a current limiting resistor to the Digital Pin 13 and the following image shows the circuit diagram for this connection.



In the file tab, go to New and select Script option. A blank editor will be opened. Paste the code mentioned in the following section and save the file. It will be save as a .m file, which is the extension for MATLAB Programs.

NOTE: Change the COM Port and Arduino Board according to your needs in the code.

After saving the file, click on Run option. The MATLAB program will interact with the Arduino Board and will perform the desired option.

An important point to be noted is that the program will not be uploaded to Arduino and hence if you reset or restart the Arduino Board, the program will be lost and you need to perform the above steps once again.

Code

```
% create an arduino object
a = arduino('com8', 'Mega2560');
% start the loop to blink led for 10 seconds
for i = 1:5
    writeDigitalPin(a, 'D13', 1);
    pause(1);
    writeDigitalPin(a, 'D13', 0);
    pause(1);
end
%end communication with arduino
clear a
```

Summary of Arduino MATLAB Interface

- Download and Install the Arduino Hardware Support Package for MATLAB.
- Connect the Arduino board to the computer running MATLAB and check if the Arduino can be communicated through MATLAB.
- Make a simple circuit for blinking an LED connected to one of the Digital Pins of Arduino.
- Write a MATLAB program for blinking the LED and save it as a .m file.
- Run the MATLAB Program and the LED connected to Arduino should start blinking as per the program.

Applications of Arduino MATLAB Interface

- By interfacing Arduino with MATLAB, we can acquire Sensor Data from Arduino and start plotting various graphs.
- We can create GUI (Graphical User Interface) for controlling our Arduino with the help of MATLAB.
- A wide range of projects related to image processing and signal processing can be implemented using Arduino and MATLAB.
- Some of the projects can be Color Detection, Object Tracking, Face Detection, etc.