

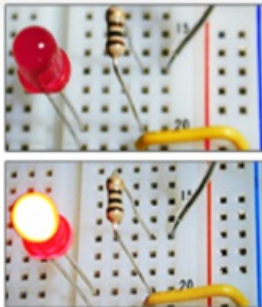
Set up and Blink - MATLAB and Simulink with Arduino

Created by Anuja Apte

```
>> a = arduino('COM5')
>> configureDigitalPin(a,9,'output')

>> writeDigitalPin(a,9,0) → LED off

>> writeDigitalPin(a,9,1) → LED on
```

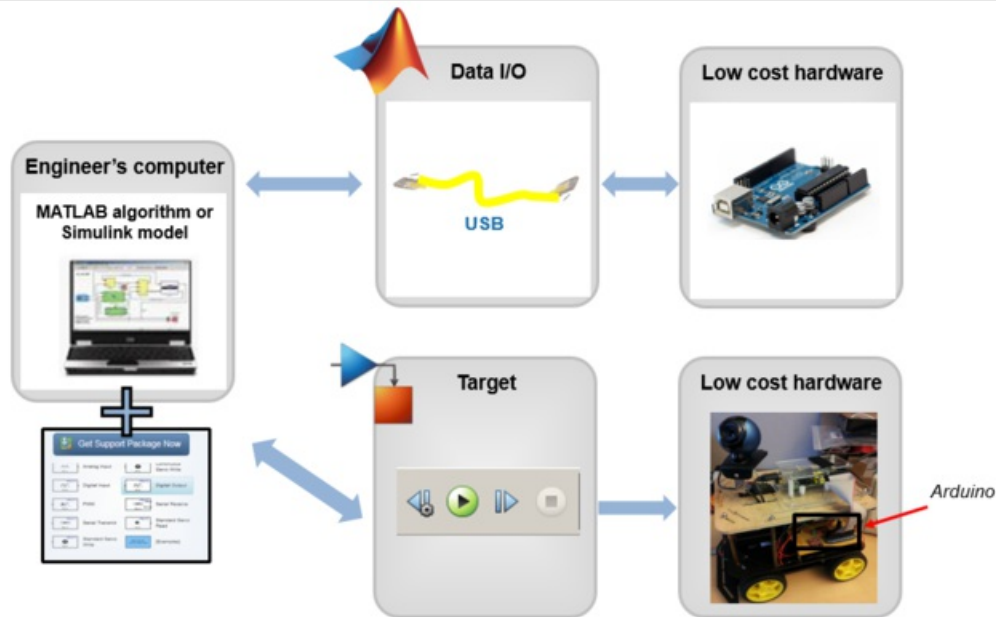


Last updated on 2018-08-22 03:36:56 PM UTC

Guide Contents

Guide Contents	2
Overview	3
Parts and Software	4
Build the circuit	5
Set up MATLAB and Simulink support package for Arduino Due	6
Blink an LED Using MATLAB Support Package and Arduino	9
Common Error Messages	9
Set up compiler support	11
Simulink model	12
Generate code, Load and Run	14
Resources	16

Overview



You can use MATLAB to communicate with an Arduino board using MATLAB Support Package for Arduino.

You can program an Arduino board using Simulink Support Package for Arduino. The support package automatically generates code from your Simulink model that then runs on the Arduino board (in a click of a button).

This tutorial covers a step-by-step guide to:

MATLAB Support Package

1. Set up the MATLAB support package for Arduino
2. Send commands from MATLAB to Arduino to blink an LED

Simulink Support Package

1. Set up the Simulink support package for Arduino
2. Build a simple Simulink model for controlling pins on the Arduino
3. Generate, download and run code on the Arduino to blink an LED

This is the first tutorial in a series on using MATLAB and Simulink to program an Arduino. In this tutorial Arduino Due is used as an example, however the same steps can be used for other boards like Uno, Mega 2560 etc.

Parts and Software

Software

MATLAB and Simulink Student Suite Release 2015a - now available for \$99

- [MathWorks \(https://adafru.it/d5E\)](https://adafru.it/d5E)
- [Amazon US \(https://adafru.it/eoc\)](https://adafru.it/eoc)
- [Amazon UK \(https://adafru.it/eod\)](https://adafru.it/eod)

If you are not a student, you can purchase the MATLAB Home-Use license for personal use:

- [MATLAB home-use \(https://adafru.it/eoe\)](https://adafru.it/eoe)

Hardware

1. [Small breadboard \(http://adafru.it/64\)](http://adafru.it/64)
2. 220 Ohm resistor
3. [LED \(http://adafru.it/845\)](http://adafru.it/845)
4. [Connecting wires \(http://adafru.it/153\)](http://adafru.it/153)
5. [Arduino Due \(http://adafru.it/50\)](http://adafru.it/50)
6. [USB cable \(http://adafru.it/2008\)](http://adafru.it/2008)

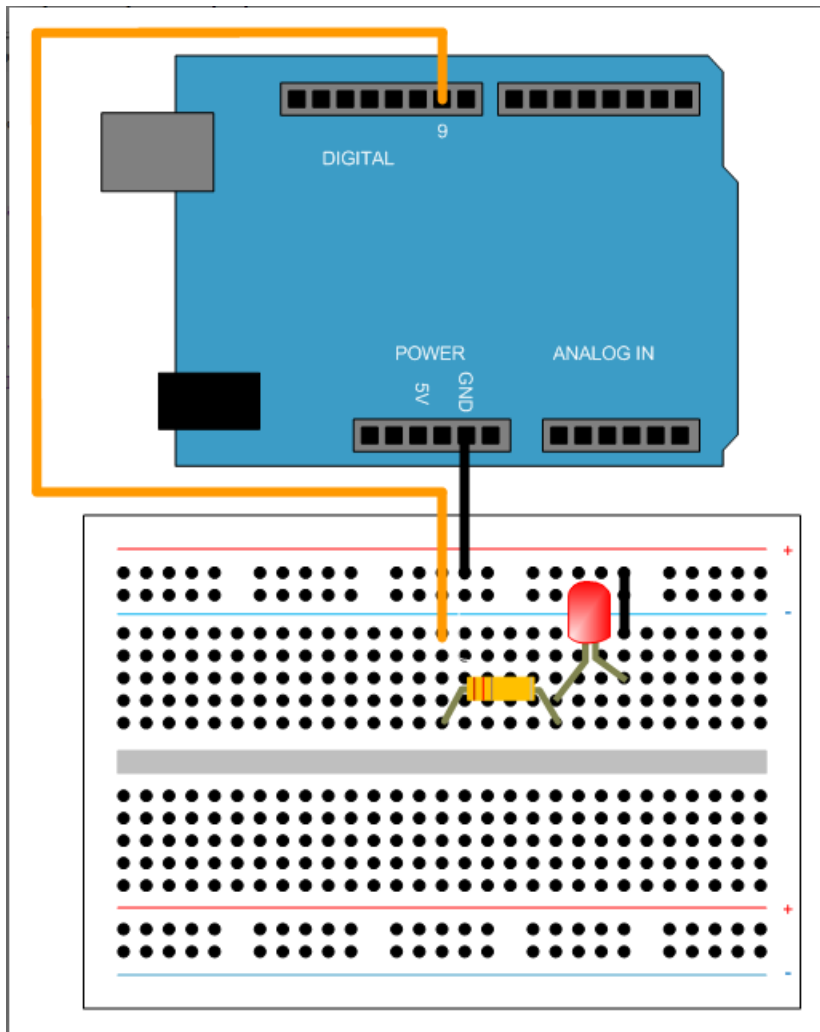
Build the circuit

Circuit

Use the following image to build a simple circuit that connects an LED to pin 9 of Arduino Due.

Connect the hardware

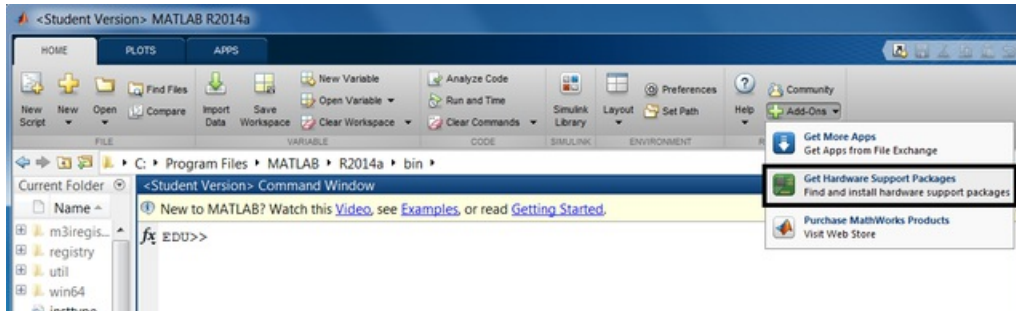
Using the USB connector, connect Arduino Due to the computer that has a MATLAB & Simulink installation.



Set up MATLAB and Simulink support package for Arduino Due

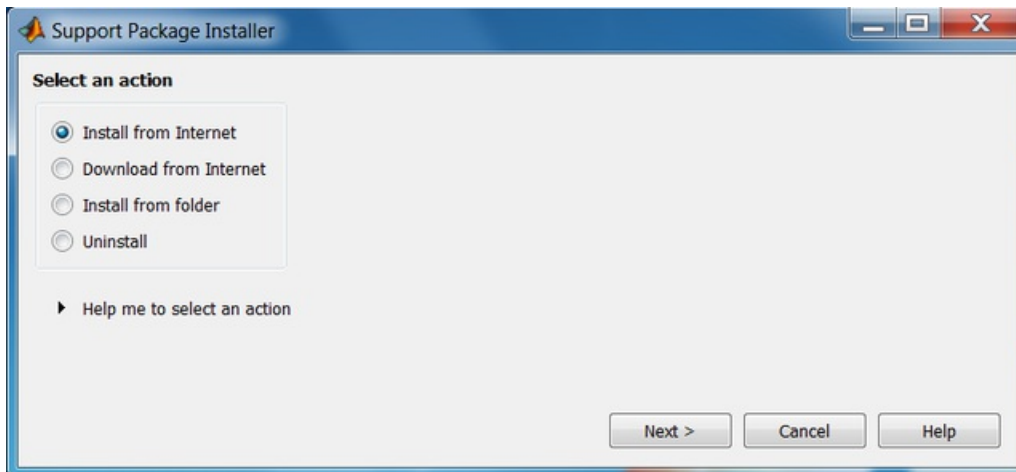
Start MATLAB

Open MATLAB and click the Add-Ons drop down menu on the top right



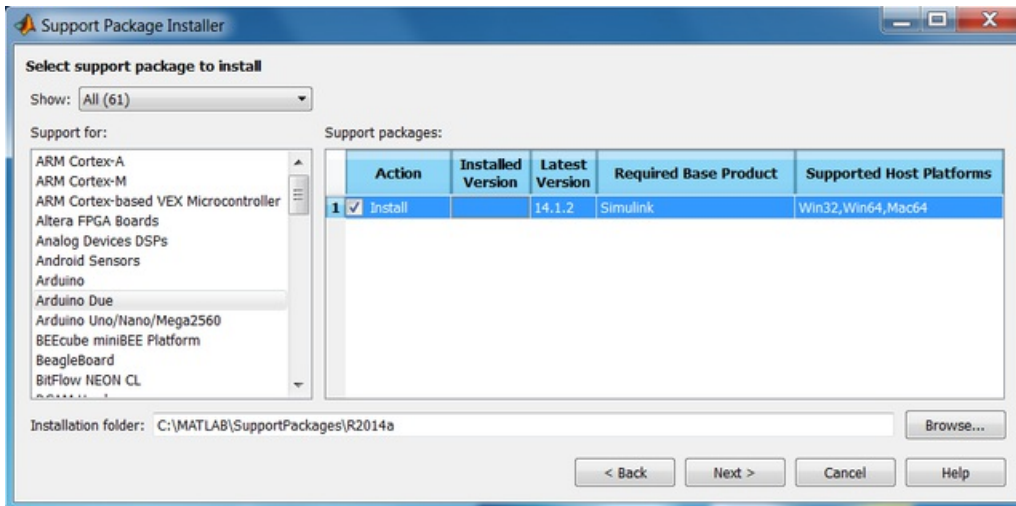
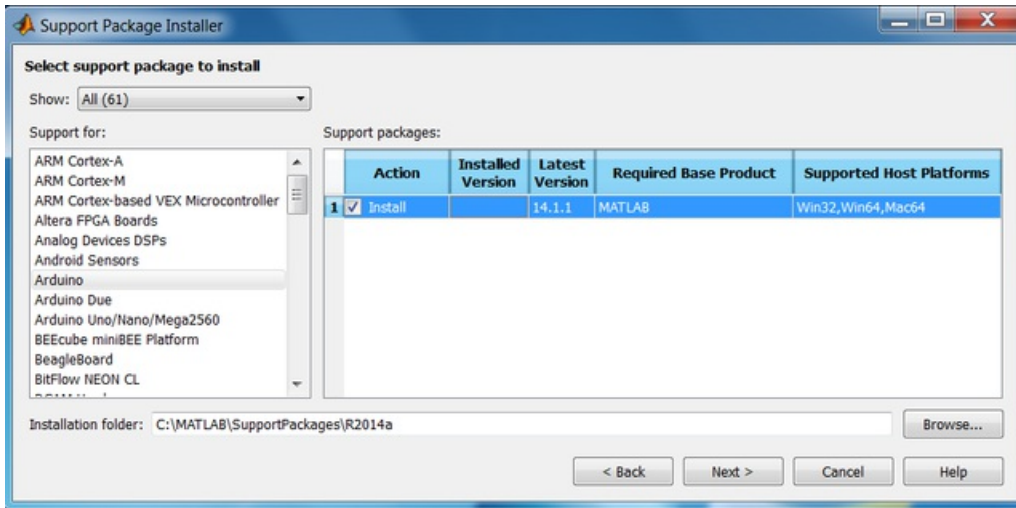
Start Support Package Installer

Click on Get Hardware Support Packages in the drop down menu to start the installer. Select 'Install from Internet' as the source for installing the support package



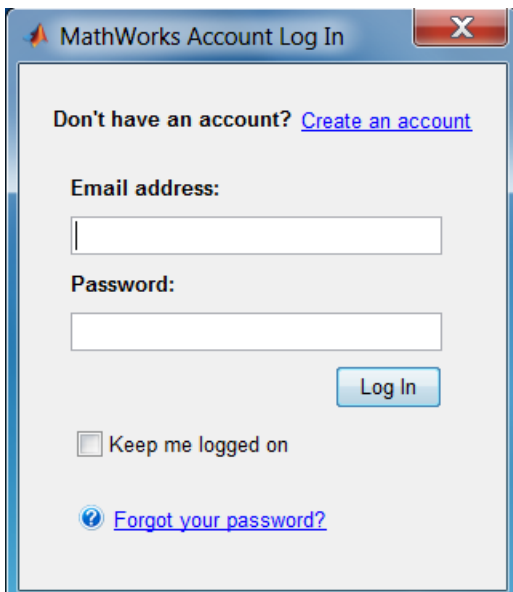
Select Arduino from a list of support packages

Click Next to see a list of support packages and select Arduino from the list for MATLAB Support Package and Arduino Due for the Simulink Support Package.



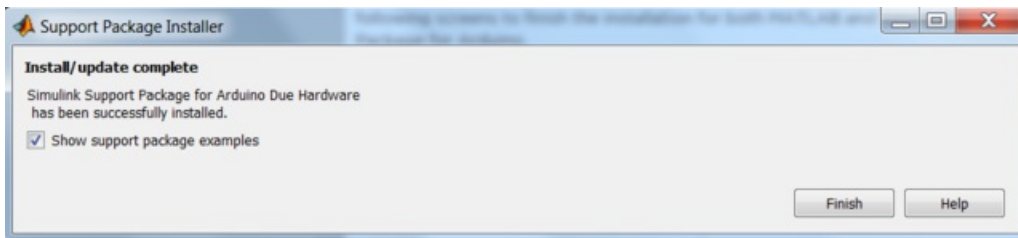
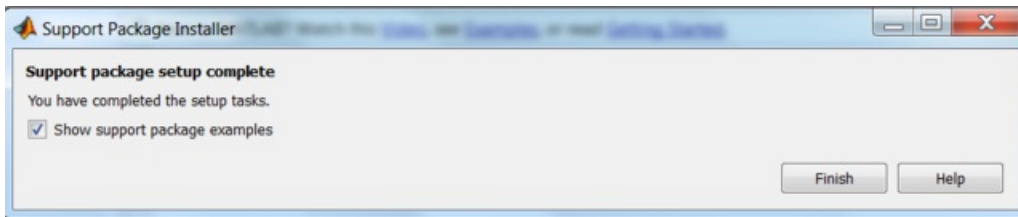
MathWorks Account

Click next and log in to your MathWorks account. If you don't have a MathWorks account, you can create one during the install process or by [visiting this page on the MathWorks website \(https://adafruit.it/eoA\)](https://adafruit.it/eoA). (<https://adafruit.it/d5K>)



Continue and Complete the Installation

Accept the license agreement on the next screen and click Next through the following screens to finish the installation for both MATLAB and Simulink Support Package for Arduino.



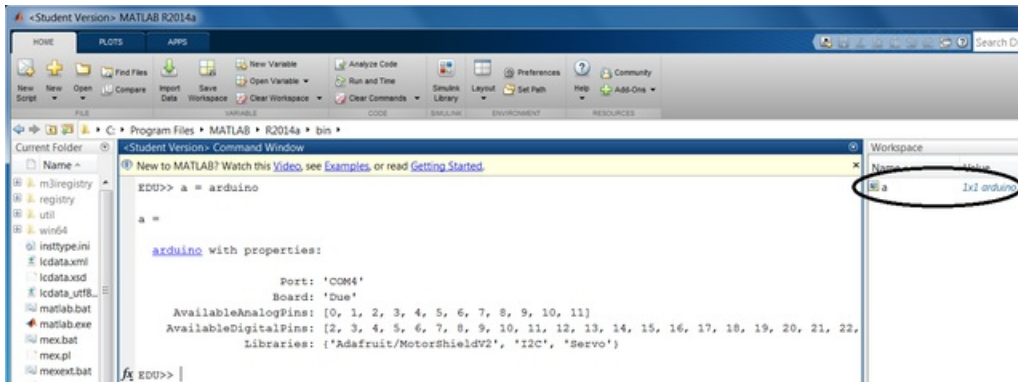
Blink an LED Using MATLAB Support Package and Arduino

Now that we have the necessary software installed, it's time to blink an LED. In this example, we will use the MATLAB support package.

To establish a connection between MATLAB and Arduino boards, the following command can be used from MATLAB.

```
>> a = arduino
```

This command connects MATLAB to the first Arduino detected on the system. A MATLAB variable by the name 'a' should now appear on the workspace or an error message indicating what went wrong.



To switch on the LED connected to Pin 9 on the Due, execute the following command

```
>> writeDigitalPin(a, 9, 1)
```

Here we are sending a request from MATLAB to Arduino, to write on the Digital Pin 9 the value of 1. This should switch ON your LED.

To blink the LED, we have to switch it ON and OFF periodically.

```
for i = 1:10
    writeDigitalPin(a, 9, 1);
    pause(0.5);
    writeDigitalPin(a, 9, 0);
    pause(0.5);
end
```

Other functions that are related to the MATLAB Support Package for Arduino can also be used along with the arduino object 'a'.

To take a look at the featured examples from MATLAB Support Package for Arduino type `arduinoExamples` in MATLAB Command Window.

Common Error Messages

Cannot detect Arduino hardware. Make sure original Arduino hardware is properly plugged in. If using unofficial(clone) hardware, specify port and board type. For more information, see this user's guide for additional troubleshooting steps.

As mentioned in the error message, it is good practice to mention the COM port at which the Due is connected to the computer. For Windows Users, **Control Panel > Hardware and Sound > Devices and Printers**.

```
>> a = arduino('COM4','Due')
```

If for some reason the connection between MATLAB and Arduino board is lost, the following error message appears.

The host and client connection is lost. Make sure the board is plugged in and/or recreate arduino and its related objects.

The best practices to follow to resolve this error -

1. Disconnect the USB cable from both Arduino and computer
2. Clean up MATLAB by typing `clear` in MATLAB Command Window
3. Plug in the USB cable to computer and Arduino
4. Try connecting to the Arduino from MATLAB by using `arduino` command

The other common error message that one can encounter during the stage of reconnection is -

Failed to open serial port COM4 to communicate with Arduino board Due. Make sure there is no other MATLAB arduino object for this board. For troubleshooting, see Arduino Hardware Troubleshooting.

To resolve this error message, repeat all the steps from before after typing `clear all` in MATLAB Command Window. In the event that this also does not help, close MATLAB and reopen before proceeding further.

Set up compiler support

Install a supported compiler

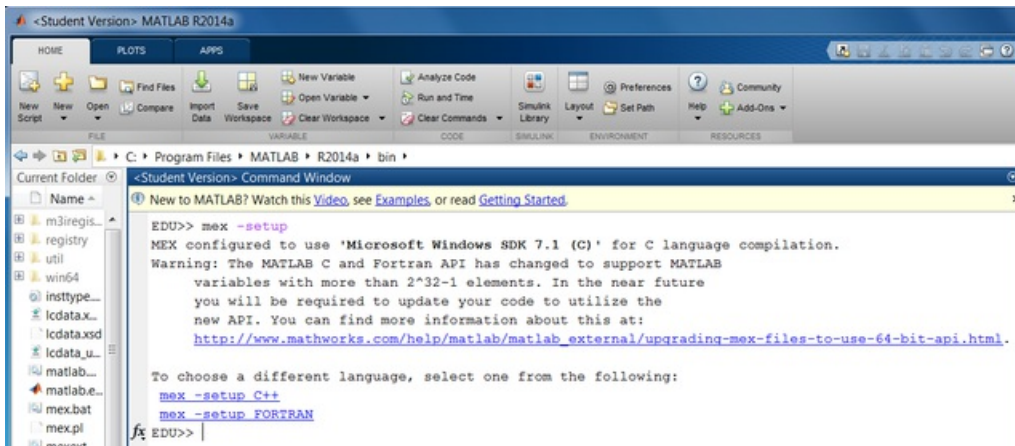
Simulink support package for Arduino is supported on 32-bit and 64-bit Microsoft Windows and Mac OS X. For the code generation process to work, a supported compiler installation is required. Refer to the following page on MathWorks website for a list of supported compilers on Windows and Mac OS:

[List of supported compilers \(https://adafru.it/eof\)](https://adafru.it/eof)

For example, you can download a Simulink supported compiler from the [Microsoft website \(https://adafru.it/d5J\)](https://adafru.it/d5J) for free.

Compiler set up in MATLAB

Enter `mex -setup` at the MATLAB command prompt to start the compiler set up. When the compiler set up is complete, you should see on MATLAB Command Window the compiler being used.



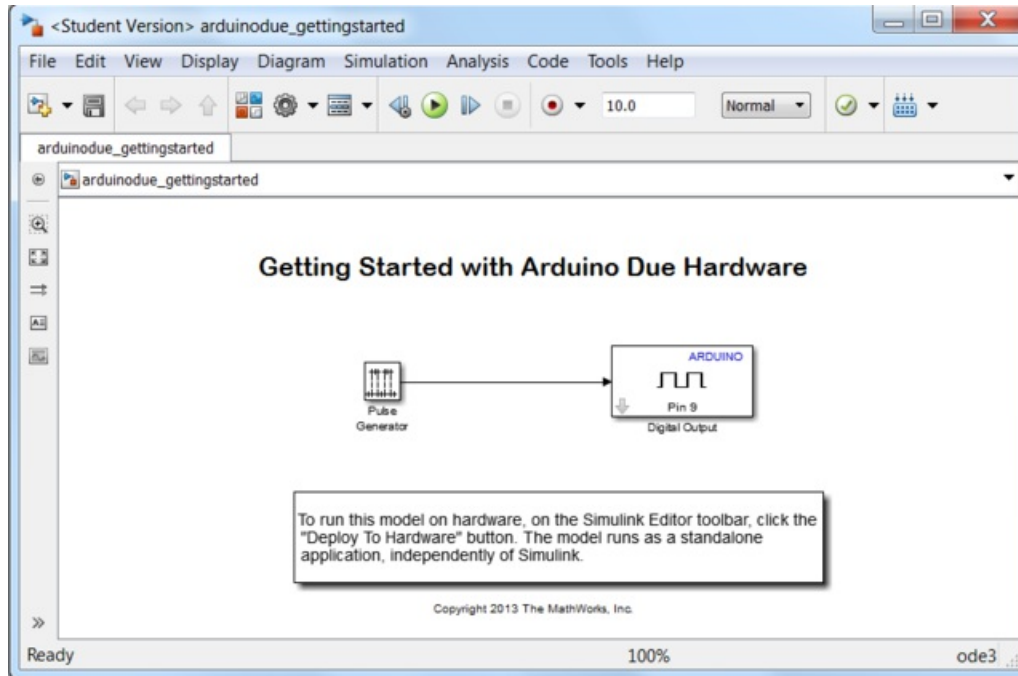
Simulink model

Open a Simulink demo model

Enter `arduinodue_gettingstarted` at the MATLAB Command Window to open a demo model. Again, in this tutorial Arduino Due is used as an example but the same steps can be used for other boards like Uno, Mega 2560 etc.

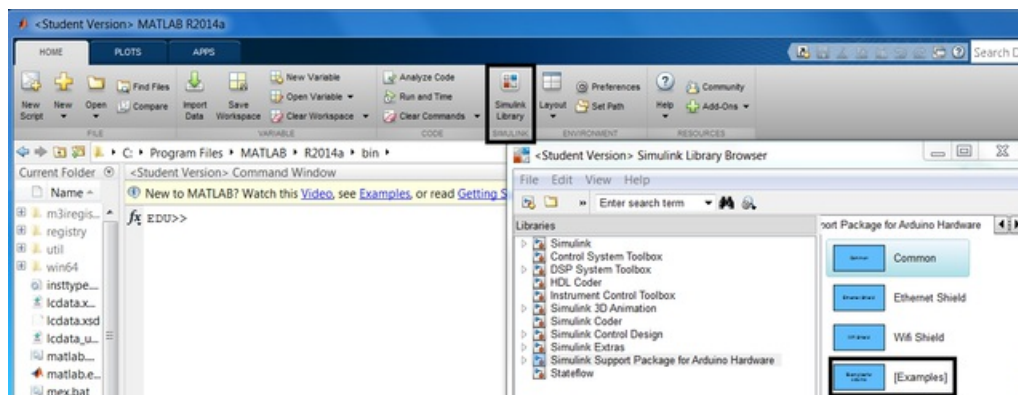
Double-click the Pulse Generator block to explore its pre-configured parameters. The Digital Output block is set up to send the output of the Pulse Generator to pin 9 of the Arduino Due. Double-click the block to explore its pre-configured parameters.

You are now ready to program Arduino boards!



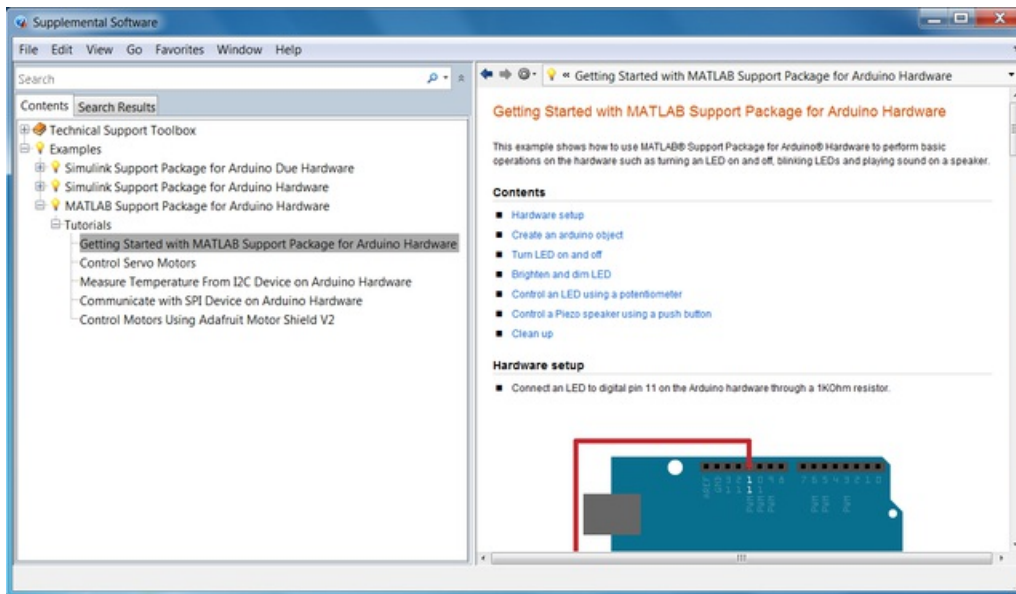
Optional - Create the model on your own

- Click on the Simulink Library icon to open the Simulink library browser
- In the left navigation pane, select Simulink Support Package for Arduino Hardware
- On the right navigation pane, double-click the [Examples] block to open the documentation



Open step-by-step guide

Click 'Getting started with Arduino Due Hardware' in the list of demos to open the step-by-step guide to build this model on your own.



Generate code, Load and Run

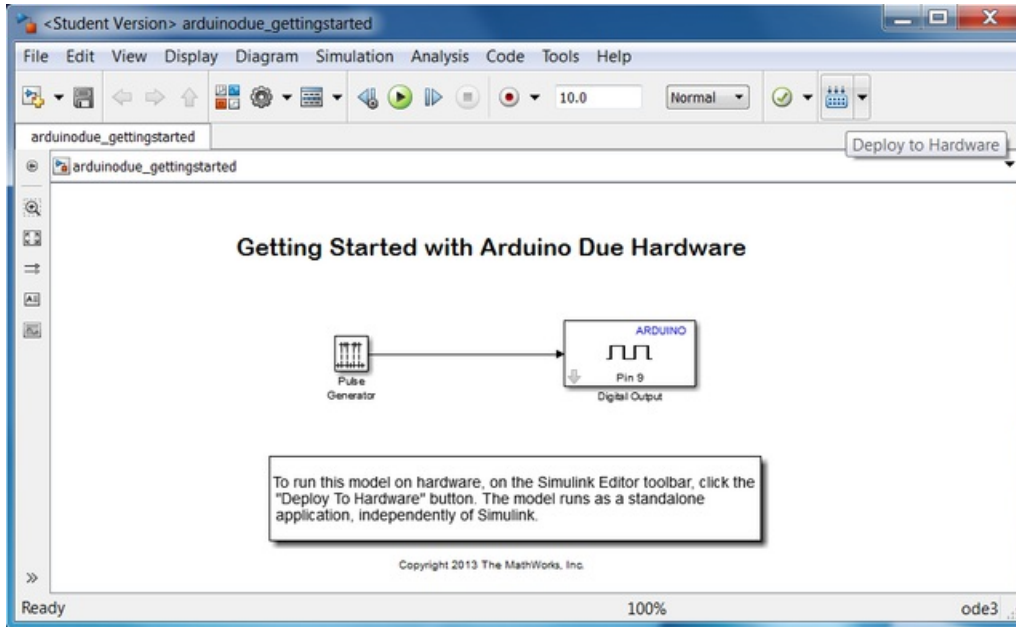
If you opened the demo model, you can go straight to Run the model on Arduino section. If you followed the steps to create your own model, then to configure the model for Arduino boards follow this section.

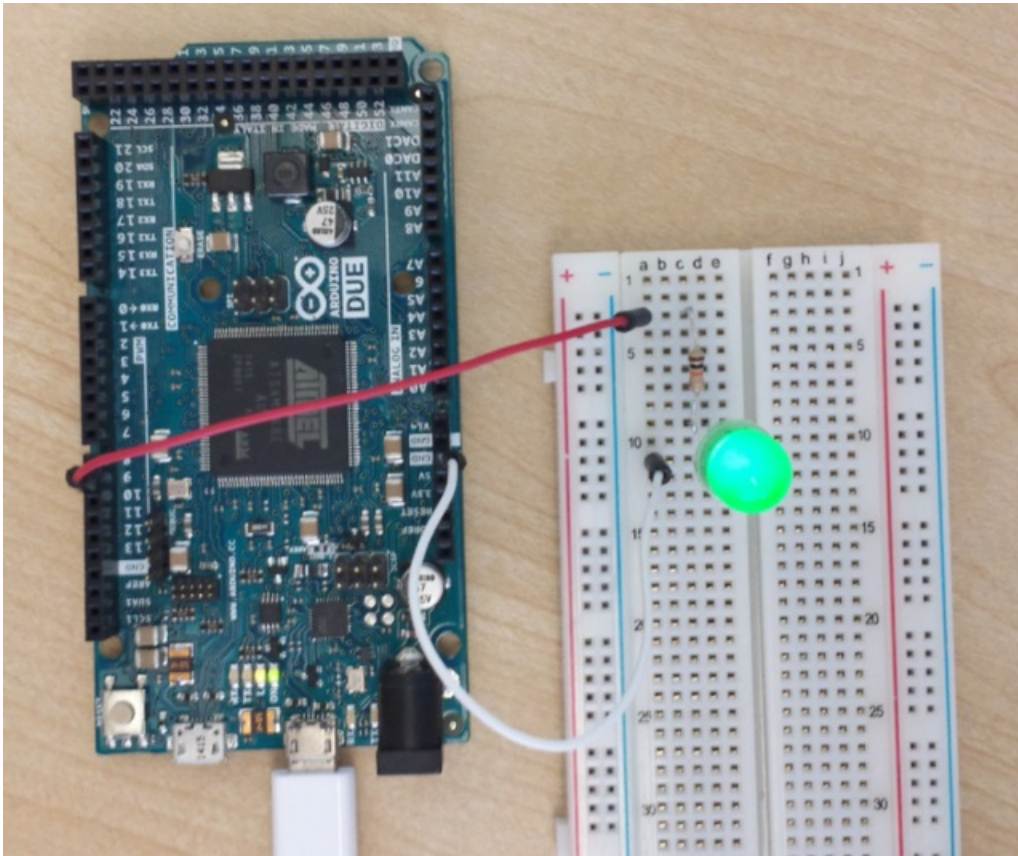
Prepare the model to run on Arduino boards

- Connect the Arduino board to your computer with a USB cable
- Select **Tools > Run on Target Hardware > Prepare to Run...**
- Review the parameters in the dialog that opens
- Unless already set, set the Target hardware parameter to the Arduino board being used (Arduino Due here)
- Do not change any other settings
- Click OK
- Save the model

Run the model on Arduino

- Click on the 'Deploy to Hardware' button as shown in the figure below
- You should see the LED on Pin 9 blinking



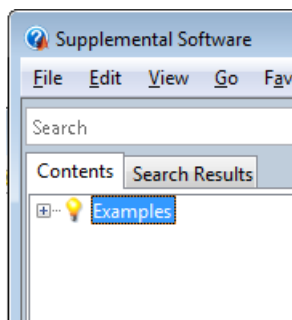
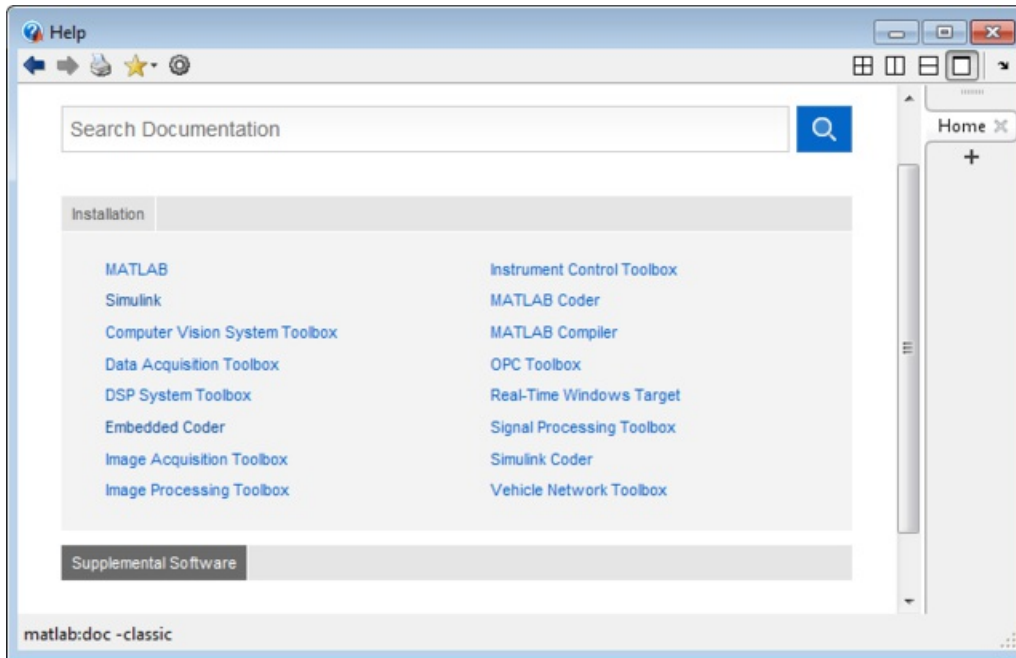


Resources

Examples for getting started

- [MATLAB examples \(https://adafru.it/d5M\)](https://adafru.it/d5M)
- [Simulink examples \(https://adafru.it/eUL\)](https://adafru.it/eUL)

Open Examples for the Support Package



Full tutorials

- [MATLAB Getting Started Tutorial \(https://adafru.it/d5N\)](https://adafru.it/d5N) (3 hrs 15 mins)
- [Simulink Getting Started Tutorial \(https://adafru.it/d5N\)](https://adafru.it/d5N) (2 hrs 15 mins - 10 mins per module)