

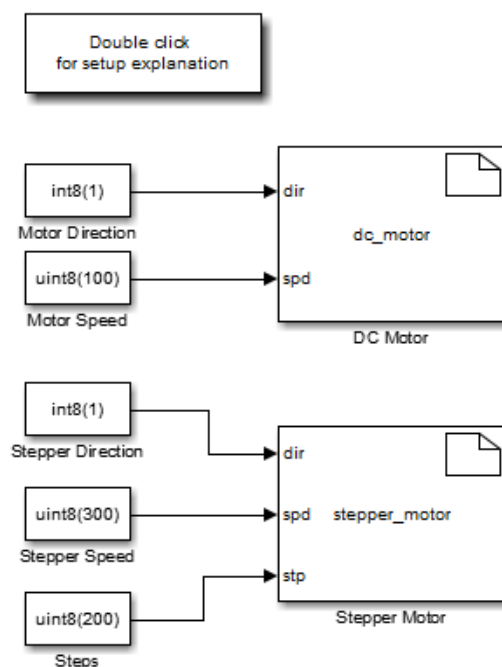
# Adafruit Motorshield V2 Simulink Blocks

This document provides step-by-step instructions on how to use the DC and Stepper Motor blocks for the Adafruit Motor Shield V2 with the Simulink Support package for Arduino. These blocks have been built following the guidelines in the [Simulink Device Drivers Guide](#).

**NOTE:** for MATLAB 2013b you will first need to download a fix for a known issue with the S-function builder block. To do this:

- 1) Login to the MathWorks [bug reports page](http://www.mathworks.com/support/bugreports/) (http://www.mathworks.com/support/bugreports/) (or go directly to <http://www.mathworks.com/support/bugreports/1006532/>).
- 2) Look for record number 1006532, on the resulting page click on the 1006532 hyperlink at the bottom, and follow instructions therein in the “workaround” section. It basically comes down to saving the zip file, opening winzip as administrator, and unzipping the fix in the MATLAB folder (e.g. C:\Program Files\MATLAB\R2013b).

## Adafruit Motor Shield V2 Driver Blocks



### DC MOTOR BLOCK:

Every time this block is executed it sets direction and speed of a given DC motor.

The first input is the direction (mode) of the motor (positive=forward, 0=release, negative=backward), while the second input is the desired speed (0 to 255). The parameter in the mask is the number of the motor (1 to 4).

### STEPPER MOTOR BLOCK:

Every time this block is executed it rotates a given stepper motor by a given number of steps. Execution is suspended while the motor is running and is resumed when the motor completes its rotation.

The first input is the direction (mode) of the motor (positive=forward, 0=release, negative=backward), the second input is the desired speed (0 to 255), and the third is the number of desired steps (0 to 255). The first parameter in the mask is the number of the stepper motor (1 or 2). The second parameter is the motion style (single, double, interleave, microstep).

### SEE ALSO:

See AFMotorV2.pdf file for instructions on how to set up the blocks before use. Also note that many of the block parameters are the same as the ones used in the MATLAB Support Package for Arduino, so you might also have a look at that package for additional explanations: <http://www.mathworks.com/matlabcentral/fileexchange/32374>

Copyright 2013, The MathWorks, Inc.

**Figure 1: DC and Stepper Motor Blocks for Adafruit MotorShield V2**

Also note that another known issue in MATLAB 2013b causes high memory usage on the Arduino side, and thus prevents the upload of models that have many blocks and/or high memory requirements. This might sometimes be a problem for the Uno (not in general for the Mega).

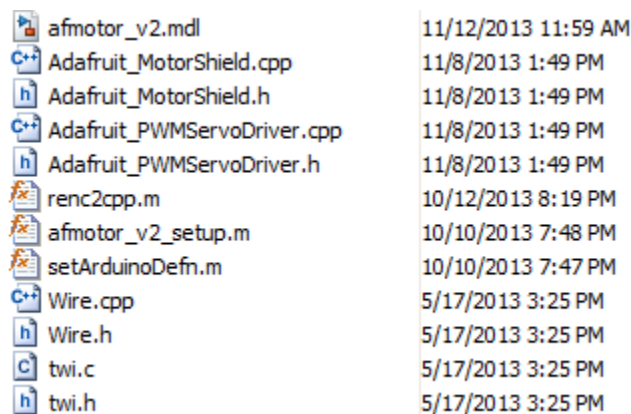
## General requirements for using the blocks

1. MATLAB and Simulink installation with release R2013a or later.
2. [Arduino support package for Simulink](#) downloaded and installed, to manage and install support packages (see also this [YouTube video](#) for step-by-step instructions), either:
  - a. From the Toolstrip, click “add-ons”, and select “get hardware support packages”
  - b. Type “supportPackageInstaller” at the MATLAB command line”
  - c. From a Simulink Model click on “Tools -> Run on Target hardware -> Install/Update Support Package ...”
3. Have the necessary hardware to use the drivers:
  - a. Arduino (Uno or Mega) board
  - b. Adafruit Motorshield V2 (<http://www.adafruit.com/products/1438>)
  - c. Motors (Stepper, DC and/or Servo)

## Files and folders required in the current directory

First of all, it is important to remember that due to the GCC compiler, the current directory should not contain any space in its name or path.

Building and using the blocks requires the following 12 files in the current directory (note, make sure to **copy** these files from their original locations, **don't move** them) See the following figure for a snapshot of how the current directory should look like.



afmotor_v2.mdl	11/12/2013 11:59 AM
Adafruit_MotorShield.cpp	11/8/2013 1:49 PM
Adafruit_MotorShield.h	11/8/2013 1:49 PM
Adafruit_PWMServoDriver.cpp	11/8/2013 1:49 PM
Adafruit_PWMServoDriver.h	11/8/2013 1:49 PM
renc2cpp.m	10/12/2013 8:19 PM
afmotor_v2_setup.m	10/10/2013 7:48 PM
setArduinoDefn.m	10/10/2013 7:47 PM
Wire.cpp	5/17/2013 3:25 PM
Wire.h	5/17/2013 3:25 PM
twi.c	5/17/2013 3:25 PM
twi.h	5/17/2013 3:25 PM

Figure 2: Current Directory in MATLAB after copying the required files

### Where to get the files:

From this file-exchange submission (the motorshields.zip file).

- afmotor\_v2.mdl
- setArduinoDefn.m
- AFMotorV2Setup.m
- renc2cpp.m

From Arduino installation/ libraries/Wire/ directory:

- Wire.cpp
- Wire.h

From the Adafruit library [Adafruit Motor Shield V2 Library](#):

- Adafruit\_MotorShield.cpp
- Adafruit\_MotorShield.h

From the utility directory in the [Adafruit Motor Shield V2 Library](#):

- Adafruit\_PWMServoDriver.cpp
- Adafruit\_PWMServoDriver.h

From the <Arduino installation>/libraries/Wire/utility directory:

- twi.c
- twi.h

The Arduino installation directory is typically included in the MATLAB support package (e.g. on windows this is “C:\MATLAB\SupportPackages\R2013b\arduino-1.0”).

## Modify an include line in Adafruit\_MotorShield.h

The next step is opening the Adafruit\_MotorShield.h file and editing line #22 to get rid of the redirection to the “utility” folder. That is replace line 22:

```
#include "utility/Adafruit_PWMServoDriver.h"
```

With the line:

```
#include "Adafruit_PWMServoDriver.h"
```

## Setting up the blocks

You need to follow these steps before the blocks are ready for use:

1. Open the model afmotor\_v2.mdl
2. Double-click on the dc\_motor block to open the S-function interface
3. Click on the Build button in right top corner to build the S-function
4. Double-click on the stepper\_motor block to open the S-function interface
5. Click on the Build button in right top corner to build the S-function
6. Run the following commands at the MATLAB command prompt:  

```
>> renc2cpp('dc_motor');  
>> renc2cpp('stepper_motor');
```
7. Click on “Tools > Run on Target Hardware > Prepare to run” (or, if you don’t see the “Prepare to run” option click on “Tools > Run on Target Hardware > Options ...”), and make sure that the right board (e.g. Arduino Uno) and the right serial ports are selected.
8. Click OK to save the settings.
9. Run the following command at the MATLAB command prompt:  

```
>> AFMotorV2Setup('afmotor_v2');
```

10. Save the model and simulate it to make sure there are no errors (the model should not do anything in simulation).
11. Make sure the Arduino is connected to the right serial port, that a DC motor is connected to port 1 and a Stepper motor is connected to port 3 and 4, and that no other program is using the serial port. The best way to make sure everything is good on the Arduino side is to test the motors with a small example sketch like the one provided on the Adafruit site.
12. Click Tools > Run on Target Hardware > Run
13. If everything goes well you should see both DC and Stepper motor running now.

## Using the blocks

Note that every time you change a parameter in either the `dc_motor` or `stepper_motor` blocks you will need to rebuild the s-function and re-issue the respective `ren2cpp` call (step 6). Also, if you change board or serial port, you'll need to redo steps 7 to 10 above.

If you create a new model and drag and drop either the `dc_motor` or `stepper_motor` blocks you will need to redo steps 7 to 10 above, where in step 9 “`afmotor_v2`” must be replaced with the name of the new model.

## DC Motor Block usage

Every time this block is executed it sets direction and speed of a given DC motor. The first input is the direction (mode) of the motor (positive=forward, 0=release, negative=backward), while the second input is the desired speed (0 to 255). The parameter in the mask is the number of the motor (1 to 4).

## Stepper Motor block usage

Every time this block is executed it rotates a given stepper motor by a given number of steps. Note that execution is suspended while the motor is running and is resumed when the motor completes its rotation. The first input is the direction (mode) of the motor (positive=forward, 0=release, negative=backward), the second input is the desired speed (0 to 255), and the third is the number of desired steps (0 to 255). The first parameter in the mask is the number of the stepper motor (1 or 2). The second parameter is the motion style (single, double, interleave, microstep).

## More Info

The meaning of many of these parameters is also described both in the Adafruit Motor Shield documentation and tutorials, as well as in the [MATLAB Support Package for Arduino](#), which also supports both V1 and V2 versions of the Motor Shield.