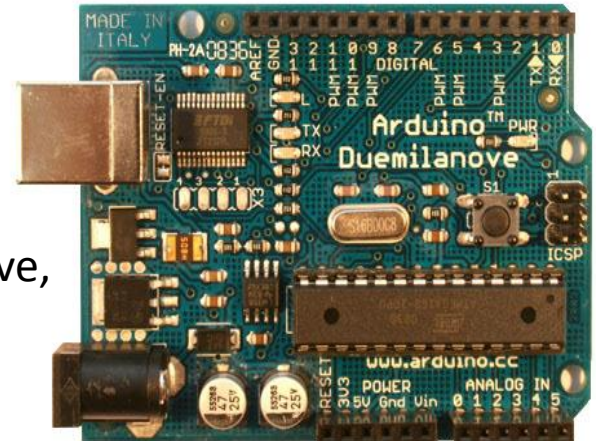


# Arduino Development Board

## The Arduino board

- The Arduino family consists of popular, low-cost development boards based on various Atmel AVR microcontrollers
- The particular model used in this class is the Duemilanove, which uses an ATmega328P microcontroller with a simplified USB interface provided by an FT232 bridge
- The board incorporates 5v and 3.3v LDO regulators to provide regulated voltage sources using either USB bus power or an external power supply of 7-12 volts
- There are also indicator LED's for power and serial activity, as well as a single user LED on pin 13
- The microcontroller pin-nodes are made accessible by 0.10" female headers on the sides of the board
- Pins are grouped according to function and are labeled at the foot of the pins



# Arduino Duemilanove

**ATMega328p** 8-Bit Microcontroller

Clock speed: 16 MHz

## Memory

32 KB flash memory (including 2 KB for bootloader)

2 KB SRAM for holding variables

1 KB EEPROM for saving data

**Operating Voltage:** 5 V

## Power Supply

5V regulated supply via USB, host PC

( Range: 7-12 V, Limits: 6-20 V )

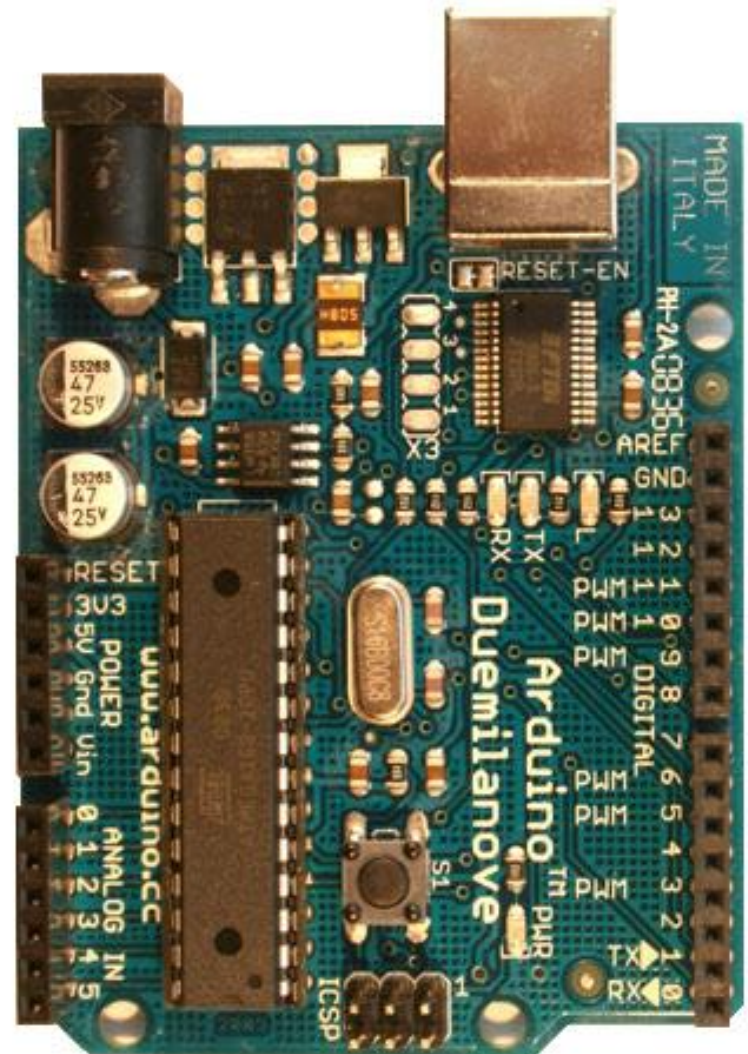
$V_{in}$ : regulated external power, or Battery (9V)

## Power Output Pins:

5V: regulated 5V at 40 mA max

3V3: regulated 3.3 V, 50 mA max via on board FTDI

GND: ground



# Arduino Pin-outs

## I/O Pins

20 I/O pins (each as input or output)

- 14 Digital, 6 Analog
- Operate at 5V, 40mA max, with pull-up resistor of 20-50 K $\Omega$  (disconnected by default)

Digital I/O: 14 pins (6 PWM outputs)

Analog I/O: 6 pins (PWM),

- 10-bit resolution as inputs (0-5V)

## Specialized Pins:

**Serial I/O:** 0 (RX), 1 (TX), connected via FTDI

**External Interrupts:** pins 2 and 3 (configurable)

**PWM:** on pin 3, 5, 6, 9, 10, 11 (8-bit output)

**SPI** (serial peripheral interface):

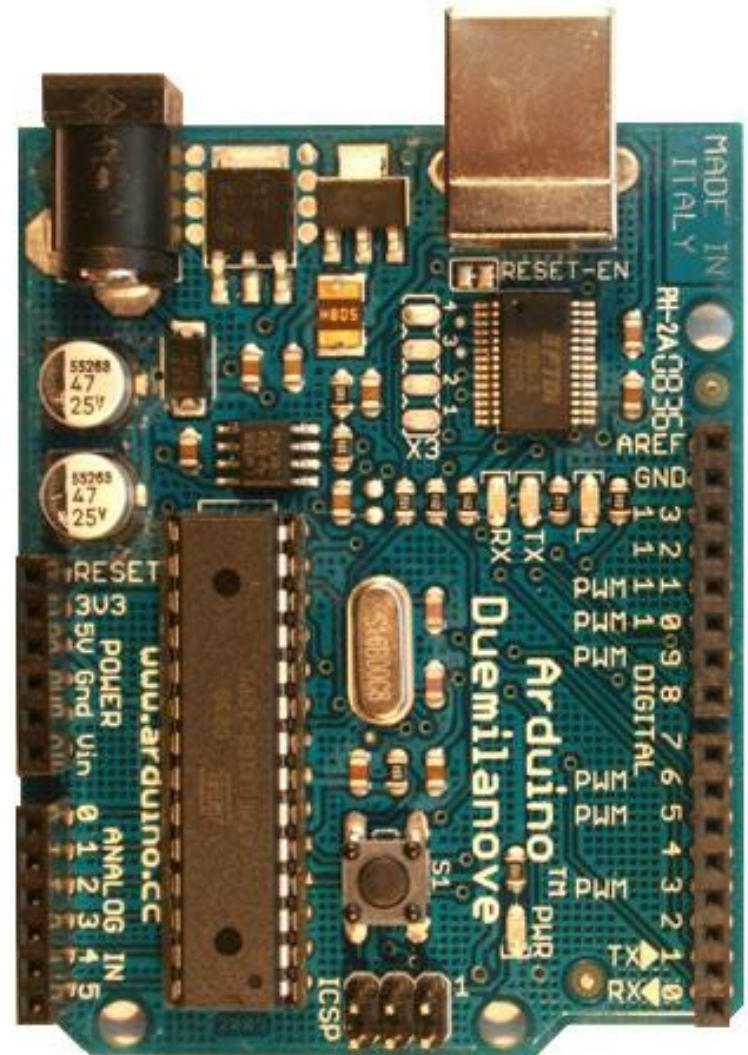
- Pins: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)

**LED:** built-in LED on pin 13

**I<sup>2</sup>C (TWI) Comm:** 4 (SDA) and 5 (SCL)

**AREF:** reference voltage for analog inputs

**Reset:** resets the microcontroller when pulled low



# Arduino Family of Products

The ATMEGA328P datasheet along with other app notes can be found at:

- [http://www.atmel.com/dyn/products/product\\_card.asp?part\\_id=4198&category\\_id=163&family\\_id=607&subfamily\\_id=760](http://www.atmel.com/dyn/products/product_card.asp?part_id=4198&category_id=163&family_id=607&subfamily_id=760)

Other sources of information are available from the AVR development community:

- <http://www.avrfreaks.net/>
- <http://forums.adafruit.com/viewforum.php?f=37&sid=a1565239d19ca21175c6a411f47b0cc1>
- <http://www.freeduino.org/>
- <http://forum.sparkfun.com/viewforum.php?f=32>

## **Note:**

- Aside from the supporting circuitry and an installed bootloader, the Arduino board is just an ATMEGA328 processor
- Most codes written for any ATMEGAxx8x microcontroller is likely to work for Arduino with little modification



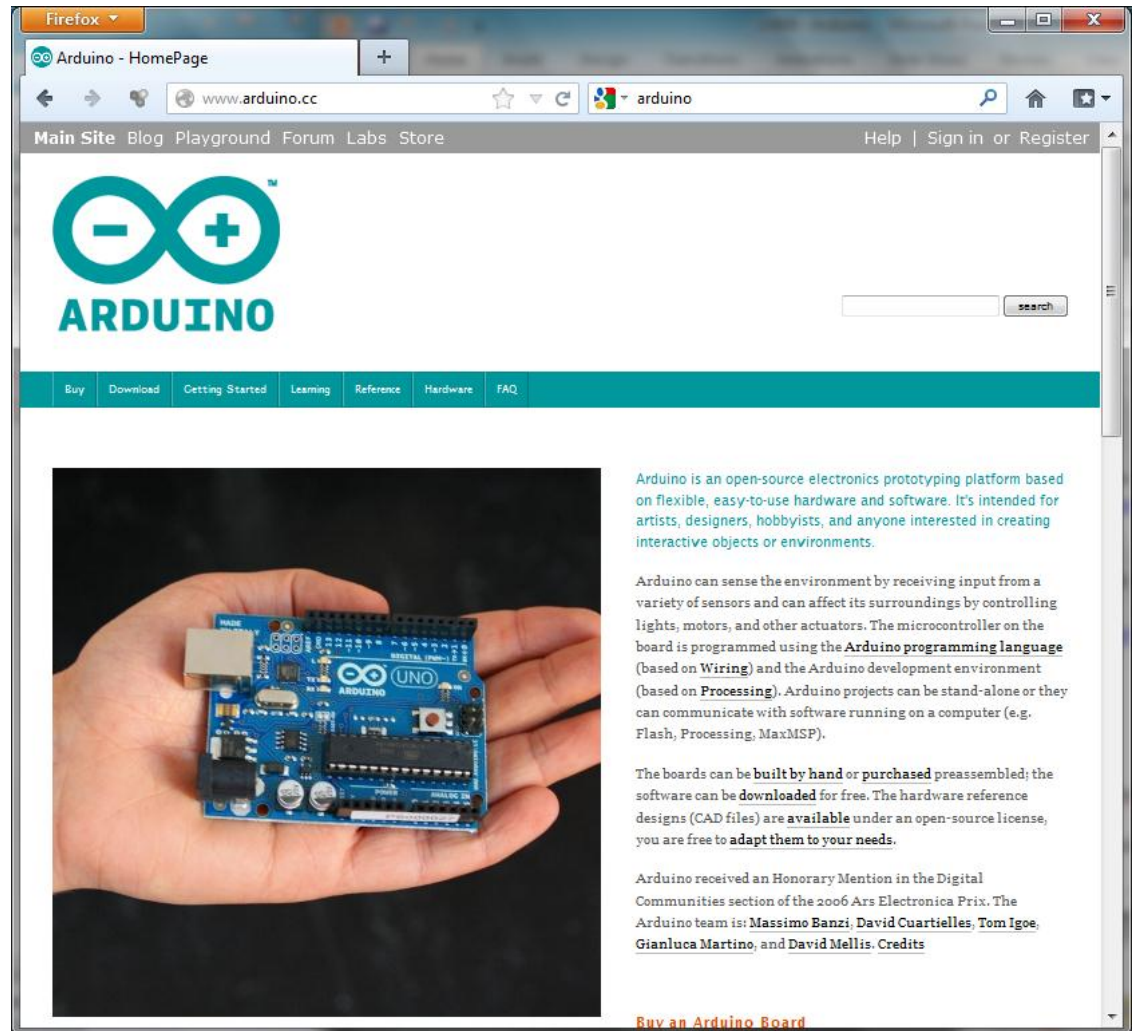
# Arduino Information

## Arduino - HomePage:

- <http://arduino.cc/>

It provides links to:

- [Buy](#)
- [Download](#)
- [Getting Started](#)
- [Learning](#)
- [Reference](#)
- [Hardware](#)
- [FAQ](#)



# Programming the Arduino

Arduino can be programmed in several ways:

- The Arduino IDE uses a subset of C/C++, which runs on Windows, Mac, or Linux.

<http://arduino.cc/en/Main/Software>

- AVR Studio is provided by Atmel and runs in Windows or a simulated Windows environment

[http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=2725](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725)

- On Linux, libavr-gcc and avrdude can be used for programming outside a GUI IDE Refer to the appropriate package management system to acquire these
- With the Matlab support package installed, MATLAB can be used for direct or scripted control of Arduino via a maintained USB connection and a server binary on the microcontroller

<http://www.mathworks.com/academia/arduino-software/arduino-matlab.html>

- Arduino programs can also be easily developed as Simulink models, which can be automatically converted to C/C++, compiled, and downloaded into Arduino via Arduino Target toolbox, Real-Time Workshop, and RTW Embedded Coder

<http://www.mathworks.com/academia/arduino-software/arduino-simulink.html>

# Arduino Programming

Programs are called sketches

Written in a C language subset

Every sketch requires 2 structural commands

```
void setup() {  
}
```

// command runs once at the beginning of operation

```
void loop() {  
}
```

// command runs a repeating loop after setup

// One-line comment

/\* Multi-line comment \*/

For a language reference and a list of libraries, check:

<http://arduino.cc/en/Reference/HomePagePrograms>

# Arduino Code Library

## Useful functions

pinMode() -set pin as INPUT/OUTPUT  
digitalWrite() -set a digital pin (HIGH/LOW)  
digitalRead() -read a digital pins state (HIGH/LOW)  
analogRead() -read analog pin  
analogWrite() -write an analog value  
delay() -wait specific amount of time (ms)  
millis() -get current counter time

## Serial Commands

serial.begin(baud rate) -start serial communication with PC  
serial.println(value) -print value to PC  
serial.print("...", value) -print value without line  
serial.read("...", value) -read value from PC with concatenation

## Examples:

```
pinMode(2, INPUT);  
digitalWrite(2, LOW);  
val=digitalRead(2);  
val=analogRead(2);  
analogWrite(2,val); //val=0-255  
delay(value); //e.g., 1000 samples
```



# Example Arduino Sketch

**/\* Blink**

Turns on an LED for one second, then off for one second, repeatedly.

This example code is in the public domain.

**\*/**

**void setup() {**

// initialize the digital pin as an output

// Pin 13 has an LED connected on most Arduino boards:

pinMode(13, OUTPUT);

**}**

**void loop() {**

digitalWrite(13, HIGH); // set the LED on

delay(1000); // wait for a second (1000 ms)

digitalWrite(13, LOW); // set the LED off

delay(1000); // wait for a second

**}**

**More tutorials found at**

**<http://arduino.cc/en/Tutorial/HomePage>**

# Programming Arduino with Matlab

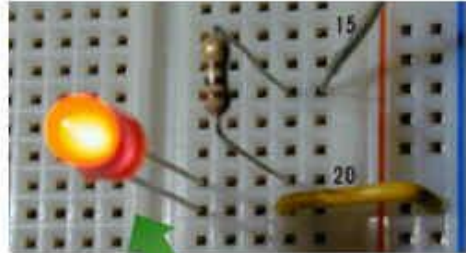
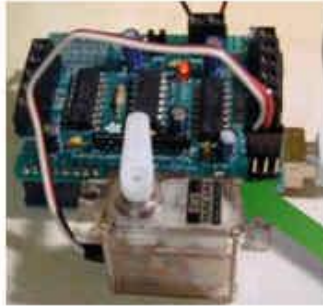
Arduino can be interactively programmed in Matlab

- Arduino support from Matlab:
  - <http://www.mathworks.com/academia/arduino-software/arduino-matlab.html>
  - [http://www.mathworks.com/matlabcentral/fileexchange/32374-matlab-support-package-for-arduino-aka-arduinoio-package/all\\_files](http://www.mathworks.com/matlabcentral/fileexchange/32374-matlab-support-package-for-arduino-aka-arduinoio-package/all_files)
  - <http://www.mathworks.com/matlabcentral/fileexchange/24638-arduino-meets-matlab>

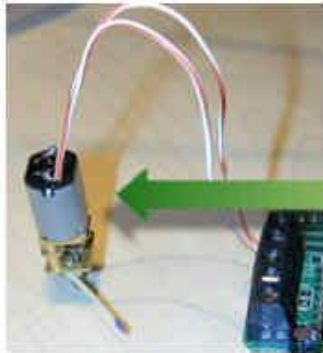
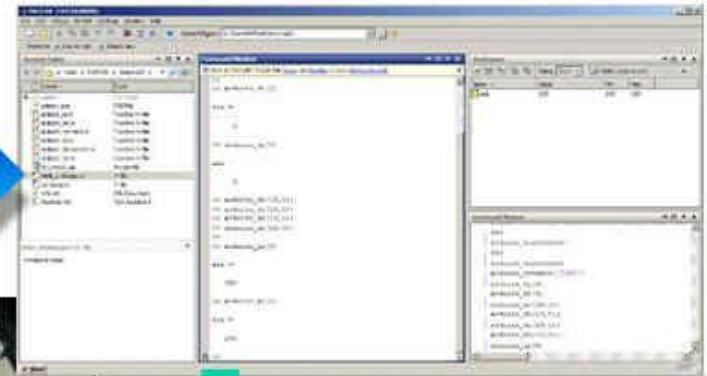
For more information, see:

- MATLAB webinar and interactive tutorial:
  - <http://www.mathworks.com/matlabcentral/fileexchange/27843-arduino-io-package-slides-and-examples>
  - <http://www.mathworks.com/company/events/webinars/wbnr43537.html>

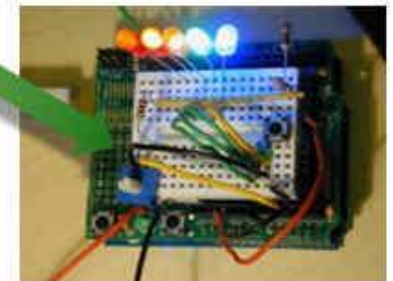
# Arduino Application in Servo Motor Control



```
>> a=arduino('COM5');  
>> a.servoWrite(1,45);  
>> a.motorSpeed(3,100);  
>> a.motorRun(3,'backward');
```



```
>> a.digitalRead(4)  
>> a.digitalWrite(5,0)  
>> a.analogRead(6)  
>> a.analogWrite(9,50)
```



# Programming Arduino in Simulink

Arduino programs can be easily developed as Simulink models

- Using Arduino Target toolbox one needs to only set the necessary parameters for the corresponding function blocks
- The Arduino Simulink program can be automatically converted to the appropriate machine code and downloaded into the Arduino board, via USB link, using Arduino Target toolbox, Real-Time Workshop, and RTW Embedded Coder
  - <http://www.mathworks.com/academia/arduino-software/arduino-simulink.html>
  - Also see our “lab guide for Arduino”

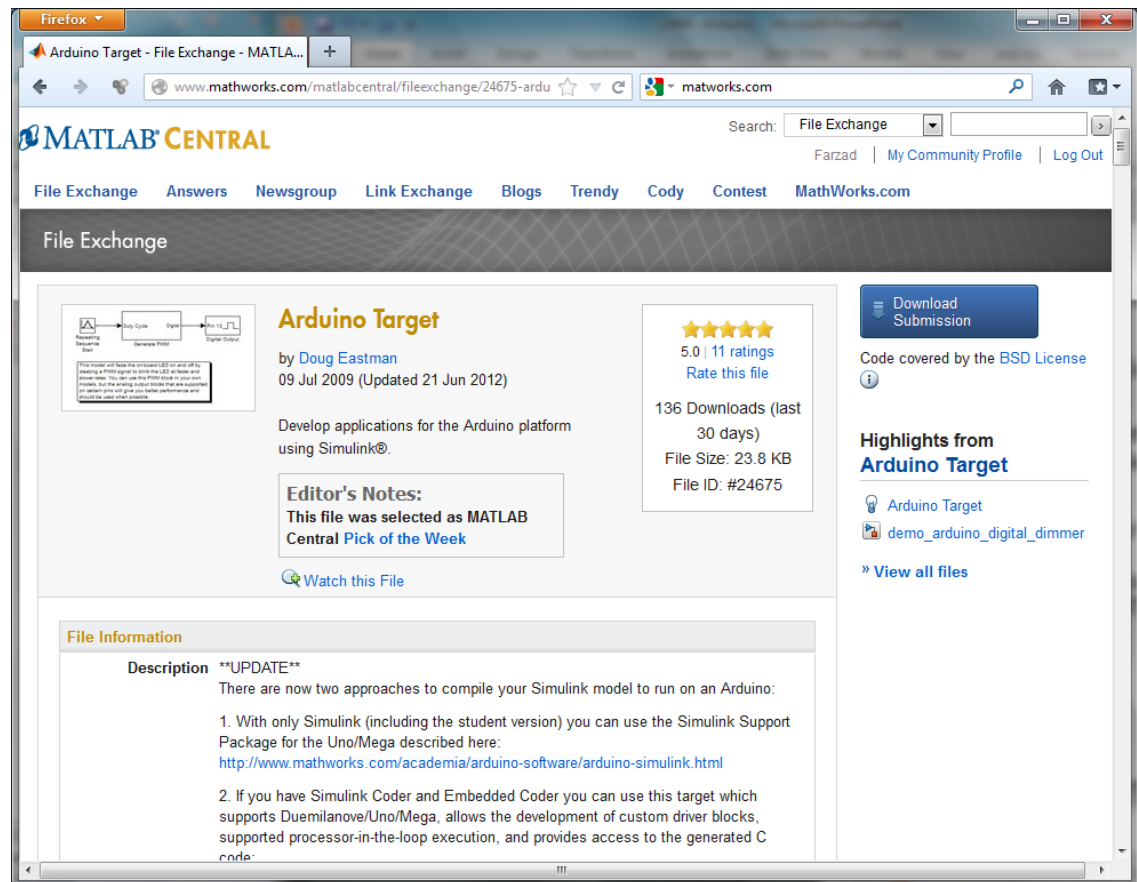
# Arduino Target Tools Information

## “Arduino Target” tools for Simulink:

- URL: <http://www.mathworks.com/matlabcentral/fileexchange/24675-arduino-target>

Two approaches to compile your Simulink model to run on an Arduino:

1. With only Simulink (including the student version) you can use the Simulink Support Package for the Uno/Mega described here:  
<http://www.mathworks.com/academia/arduino-software/arduino-simulink.html>
2. If you have Simulink Coder and Embedded Coder you can use this target which supports Duemilanove/Uno/Mega, allows the development of custom driver blocks, supported processor-in-the-loop execution, and provides access to the generated C code:  
<http://www.mathworks.com/matlabcentral/fileexchange/30277>





# Arduino Target Programming

## Arduino Target blocks:

[Arduino Analog Input](#)

[Arduino PWM](#)

[Arduino Digital Input](#)

[Arduino Digital Output](#)

[Arduino Serial Receive](#)

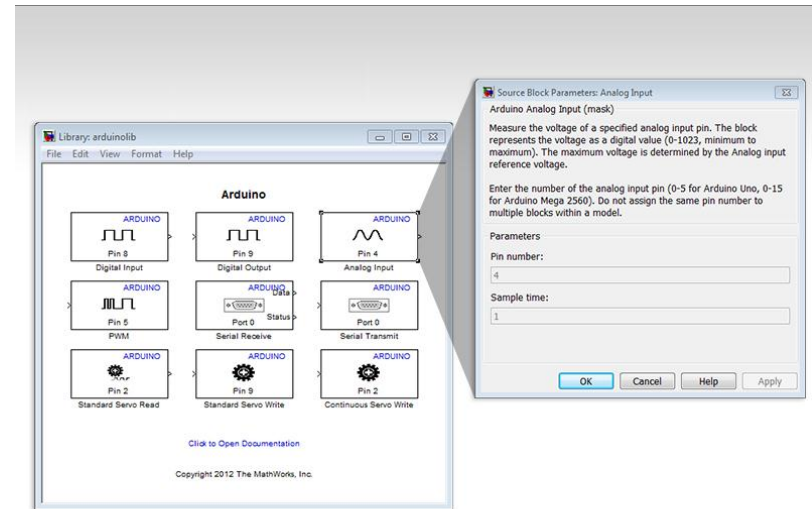
[Arduino Serial Transmit](#)

[Arduino Standard Servo Read](#)

[Arduino Continuous Servo Write](#)

[Arduino Standard Servo Write](#)

Measure voltage of analog input pin  
Generate PWM on analog output pin  
Get logical value of digital input pin  
Set logical value of digital output pin  
Get one byte of data from serial port  
Send buffered data to serial port  
Get shaft position of servo motor  
Set shaft speed of servo motor  
Set shaft position of servo motor



## Example:

A Simulink model that solves the Arduino Blink Challenge. The controlled circuit switches between four modes: all LEDs off, all LEDs on, LEDs blinking simultaneously, and LEDs cycling at a specified frequency. This model is included with [Arduino Blink Challenge Simulink Models](#).

