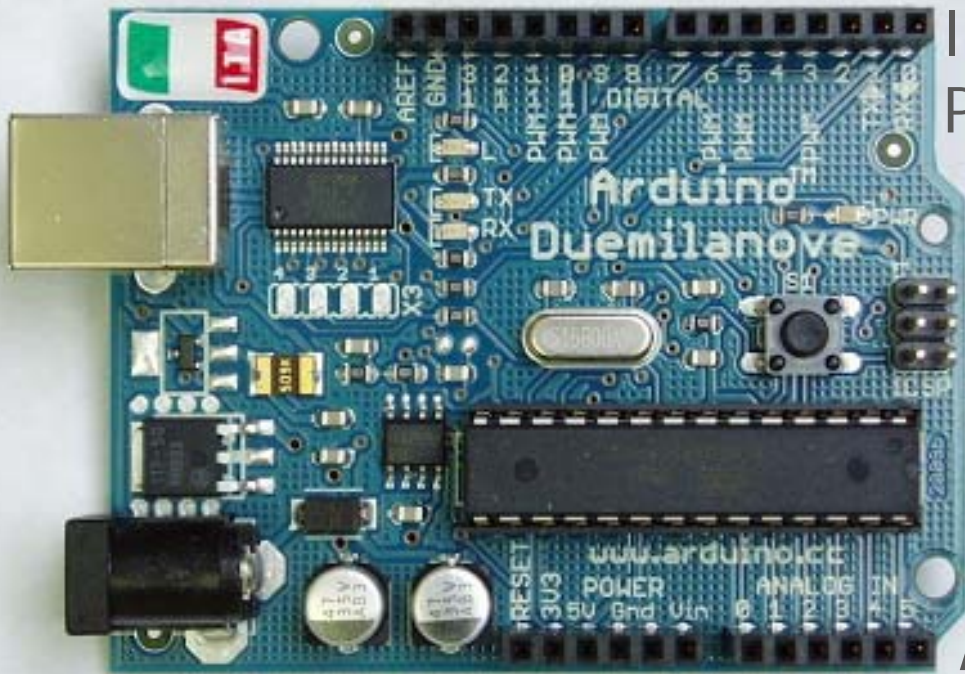


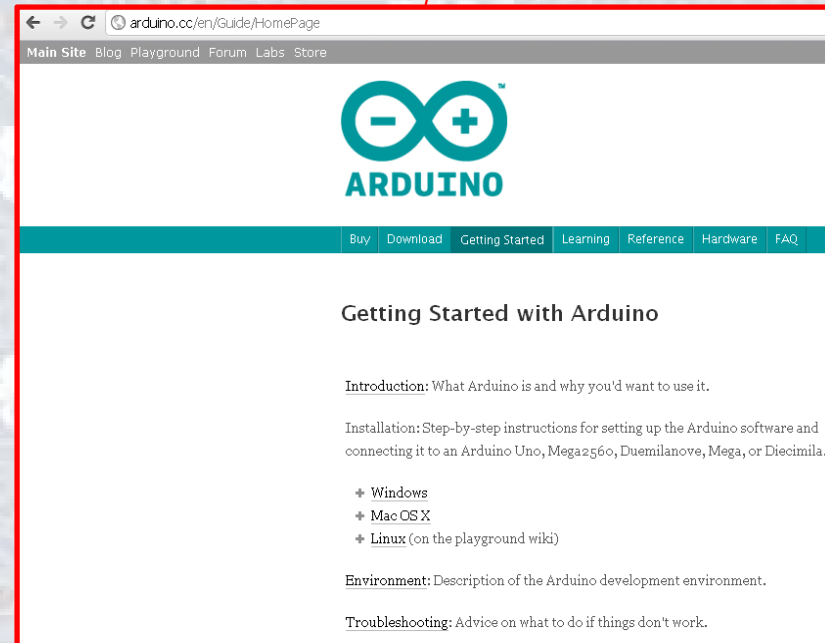
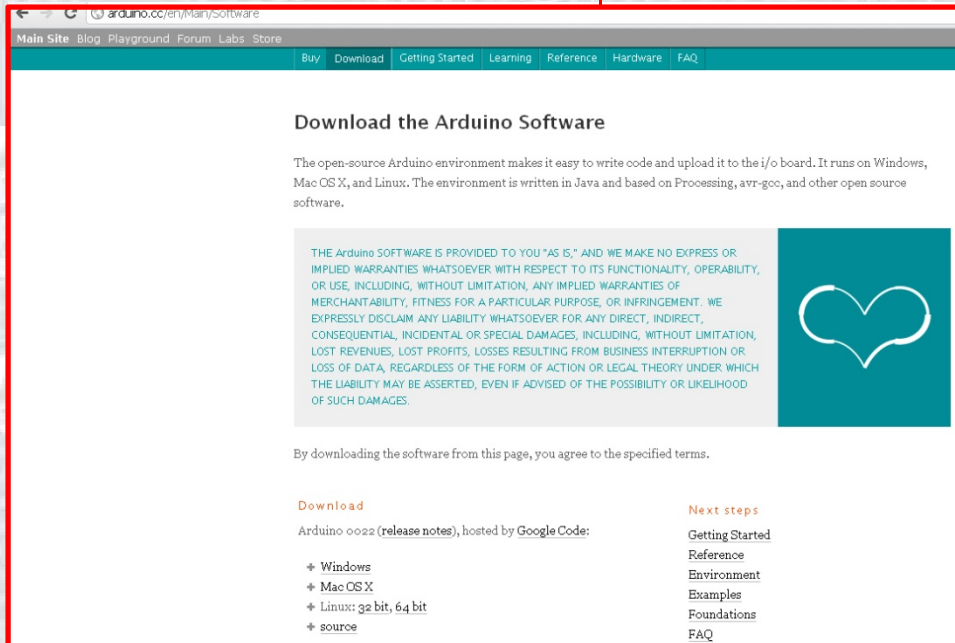
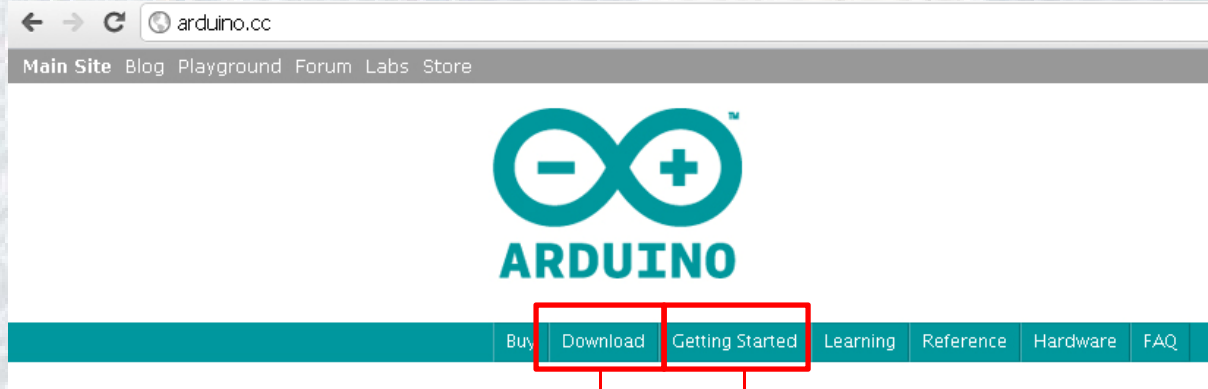
Introduction to Arduino



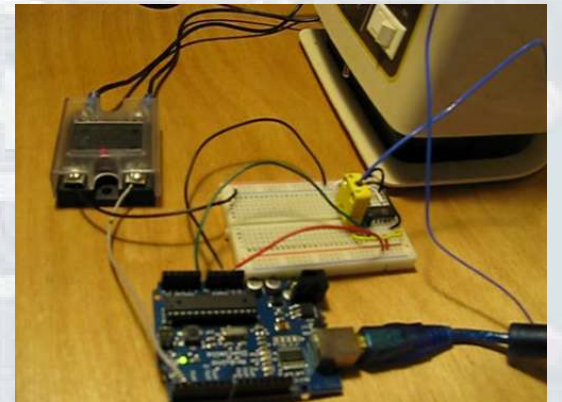
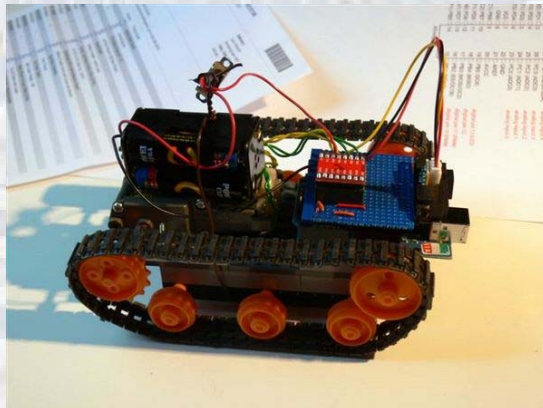
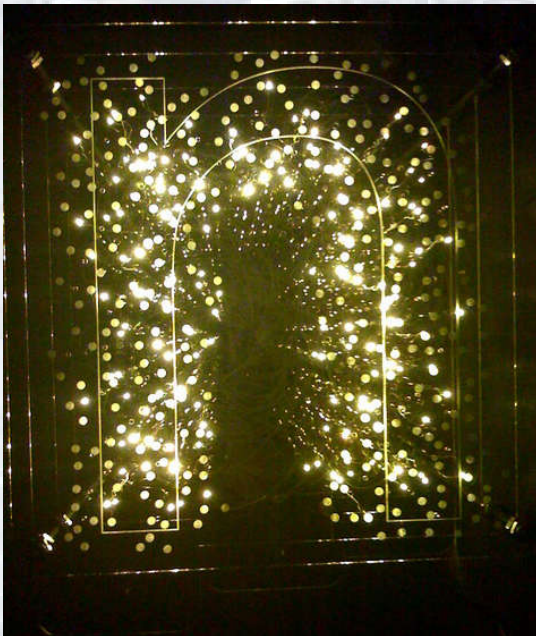
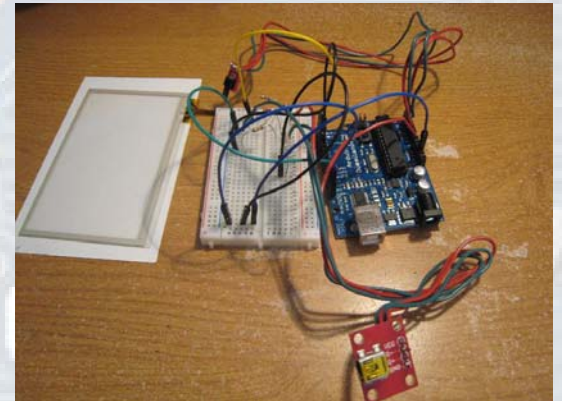
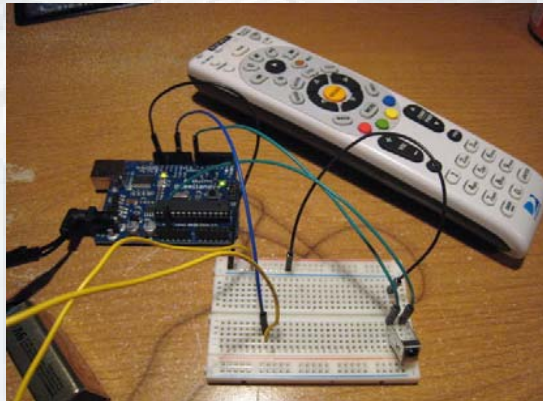
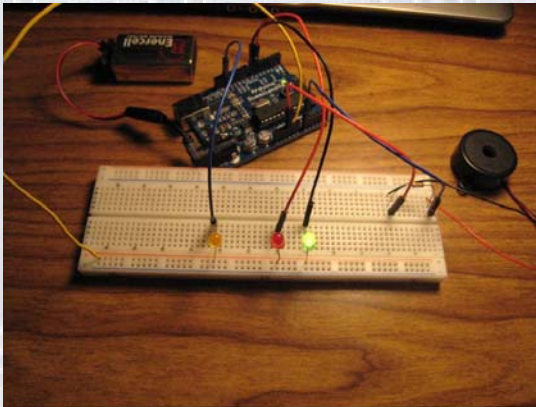
Instructor:
Plamena Milusheva

August 23, 2011

Set up: Drivers and IDE

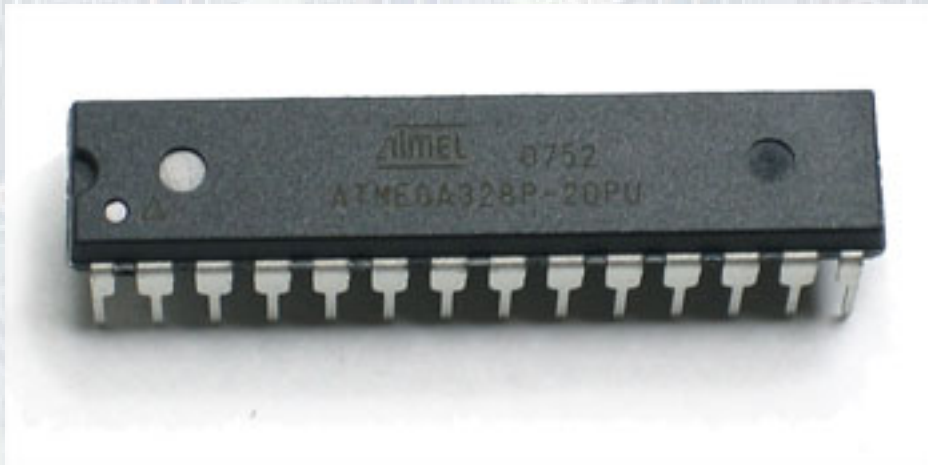


Some Arduino applications:



...and so much more!

Microcontroller: The brain of your Arduino



ATmega328p

Miniature computer

- ➔ processor
- ➔ long term memory (program storage)
- ➔ short term memory (calculations)

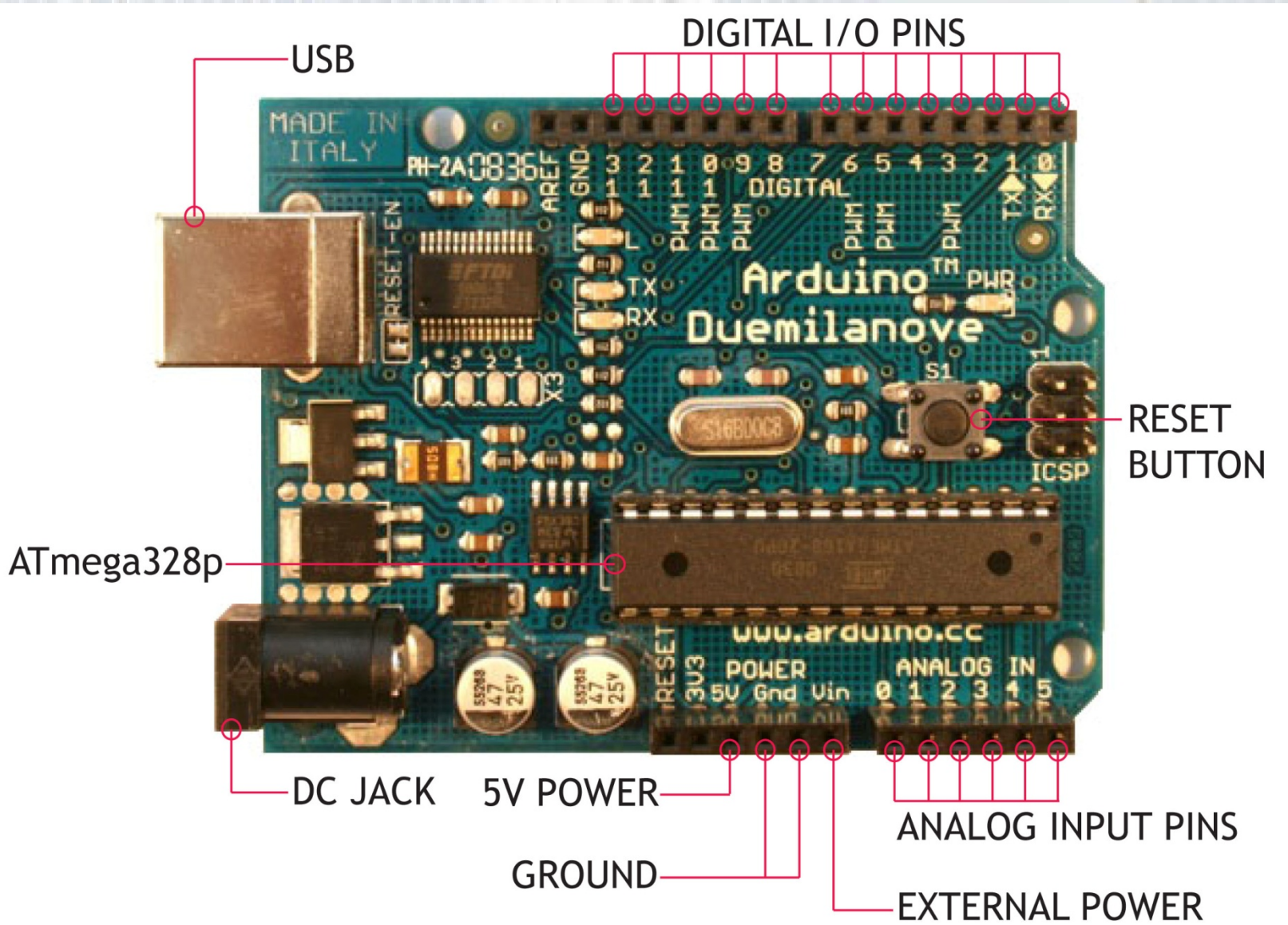
Input and Output

- ➔ digital (binary: High, Low)
- ➔ analog (data sequence)

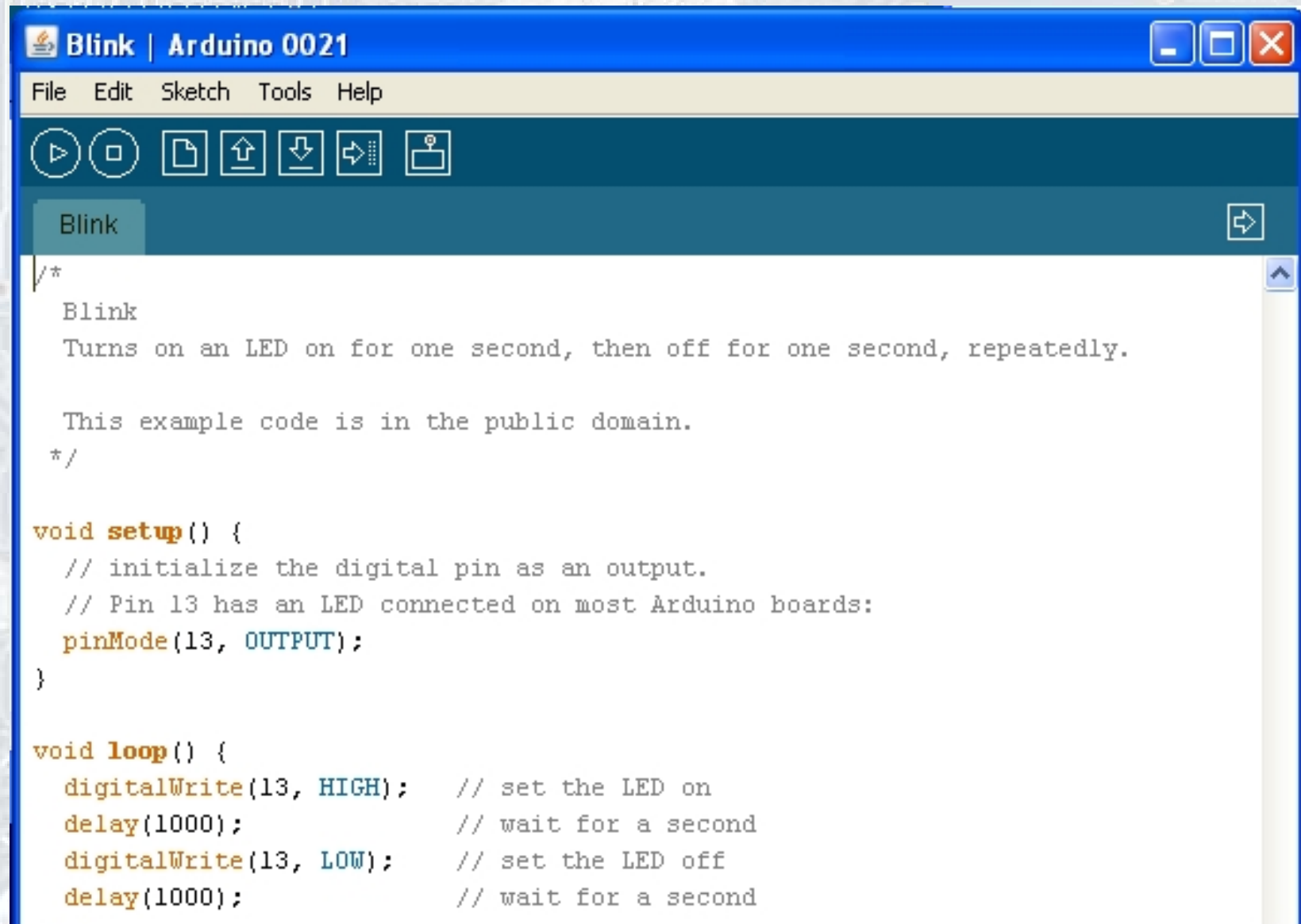
What makes your Arduino Duemilanove board work:

- ➔ Runs at 16 MHz
- ➔ 32 KB Flash memory
- ➔ 2 KB RAM (for calculations)
- ➔ 1 KB EEPROM (re-writable read-only memory)
- ➔ 14 Digital Input/Output pins
- ➔ 6 Analog Output pins

The essentials of the Arduino board



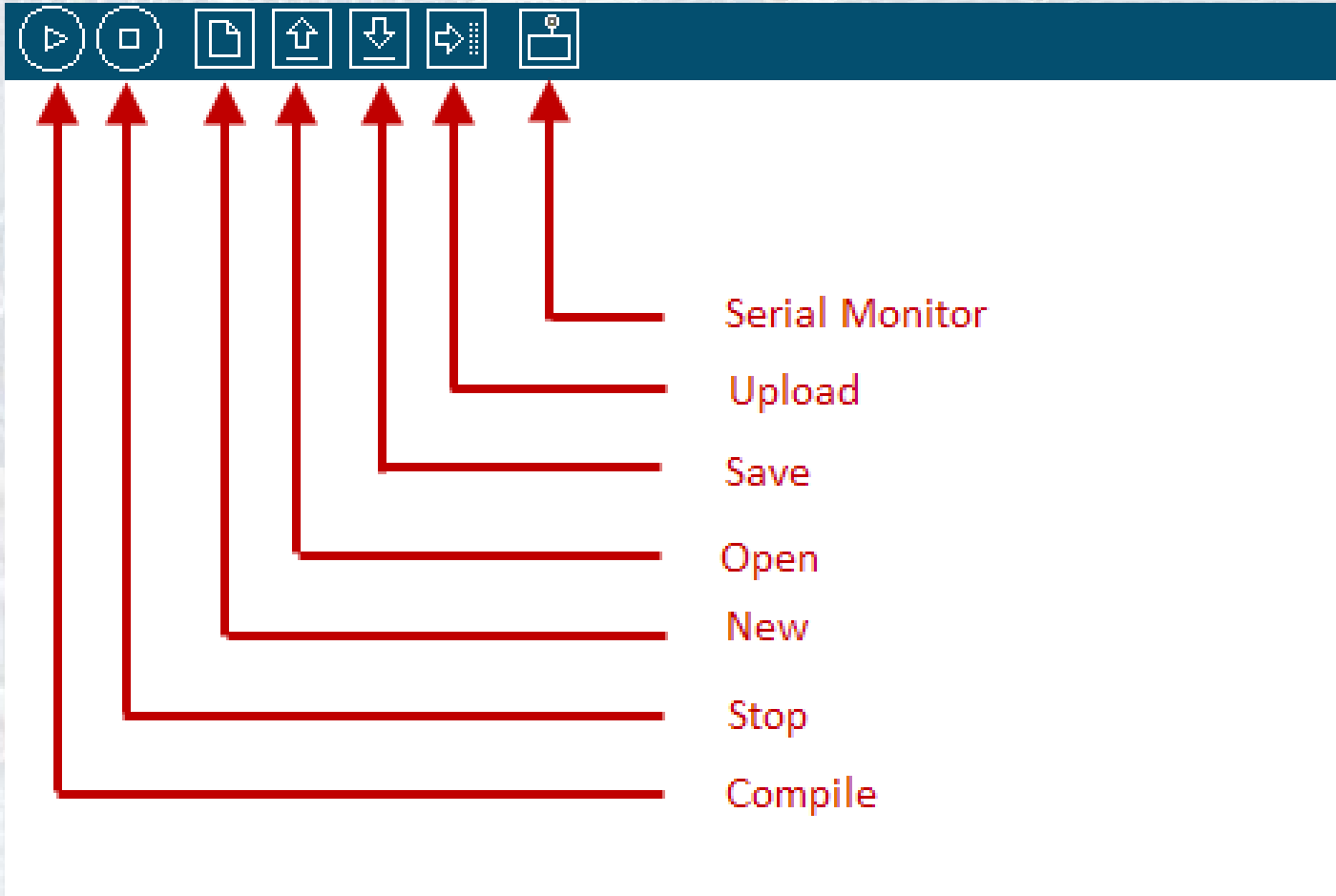
The Arduino IDE



The screenshot shows the Arduino IDE window titled "Blink | Arduino 0021". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for running, stopping, saving, opening, and other sketch-related actions. The main editor area displays the following code:

```
/*  
  Blink  
  Turns on an LED on 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  
  digitalWrite(13, LOW); // set the LED off  
  delay(1000);           // wait for a second
```

Basic button controls



Library of Arduino code

← → ↻ arduino.cc/en/Reference/HomePage

[Main Site](#) [Blog](#) [Playground](#) [Forum](#) [Labs](#) [Store](#)

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

[Reference](#) [Language](#) | [Libraries](#) | [Comparison](#) | [Changes](#)

Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

Structure

- + [setup\(\)](#)
- + [loop\(\)](#)

Control Structures

- + [if](#)
- + [if...else](#)
- + [for](#)
- + [switch case](#)
- + [while](#)
- + [do... while](#)
- + [break](#)
- + [continue](#)
- + [return](#)
- + [goto](#)

Further Syntax

- + [;](#) (semicolon)
- + [{}](#) (curly braces)
- + [//](#) (single line comment)

Variables

Constants

- + [HIGH](#) | [LOW](#)
- + [INPUT](#) | [OUTPUT](#)
- + [true](#) | [false](#)
- + [integer constants](#)
- + [floating point constants](#)

Data Types

- + [void](#)
- + [boolean](#)
- + [char](#)
- + [unsigned char](#)
- + [byte](#)
- + [int](#)
- + [unsigned int](#)
- + [word](#)
- + [long](#)
- + [unsigned long](#)

Functions

Digital I/O

- + [pinMode\(\)](#)
- + [digitalWrite\(\)](#)
- + [digitalRead\(\)](#)

Analog I/O

- + [analogReference\(\)](#)
- + [analogRead\(\)](#)
- + [analogWrite\(\)](#) - *PWM*

Advanced I/O

- + [tone\(\)](#)
- + [noTone\(\)](#)
- + [shiftOut\(\)](#)
- + [shiftIn\(\)](#)
- + [pulseIn\(\)](#)

Time

- + [millis\(\)](#)

Getting your Arduino to do things...

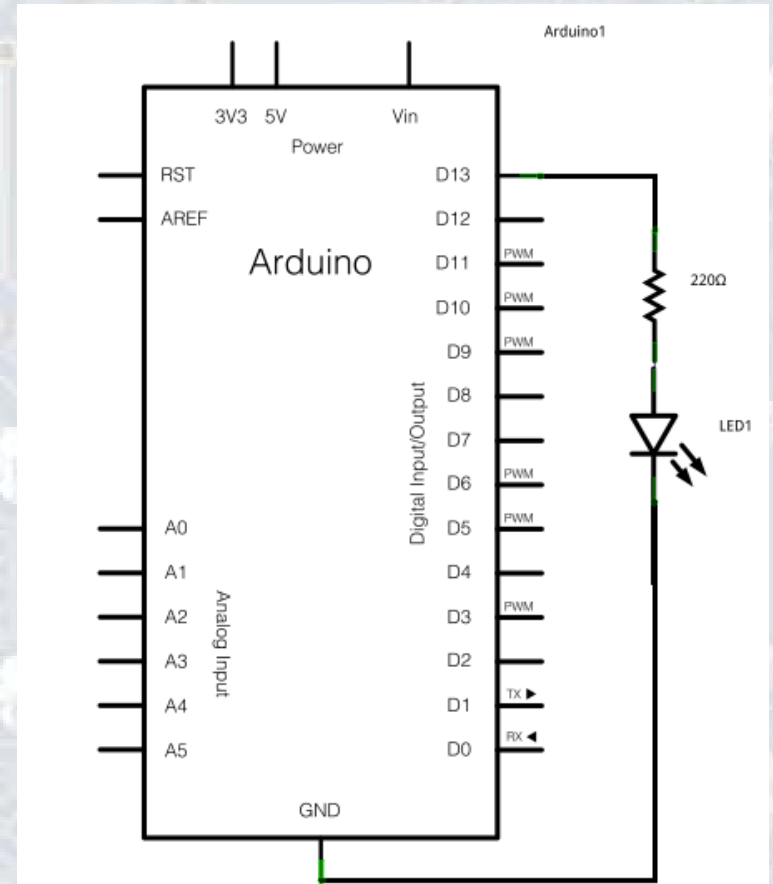
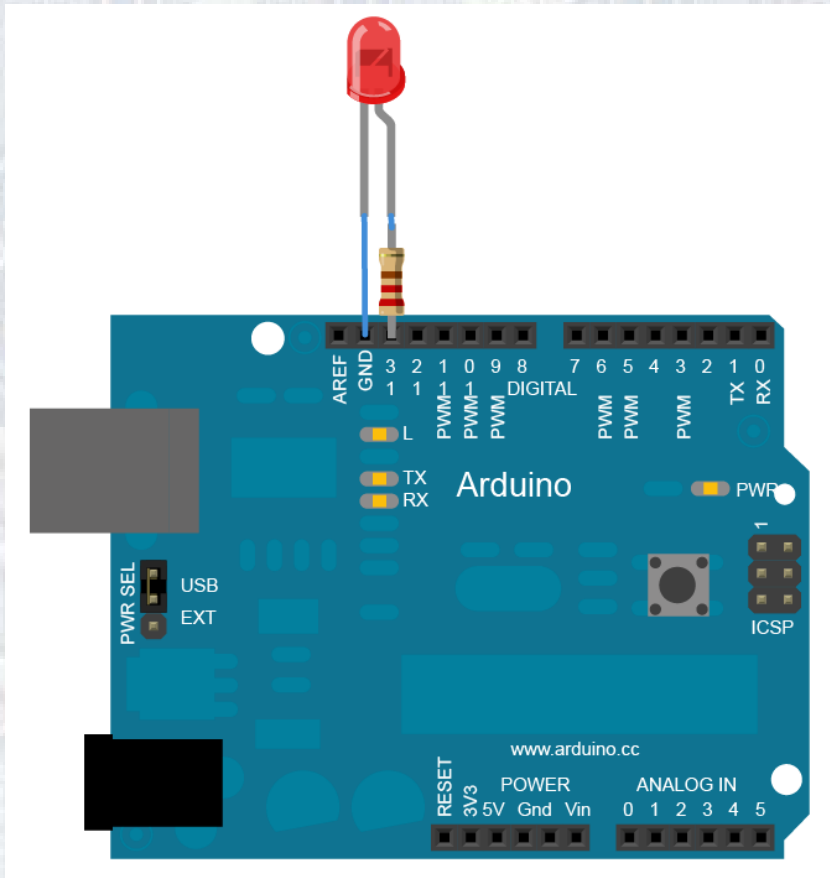
```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("success!");  
}
```

Setup () function:
Runs once at beginning of
program execution

Loop () function:
Loops the dynamic
elements of the program

*You may need to change the rate of the serial monitor to 57600 in order to get a read.

Digital Output: Make and LED blink



Digital Output: Code

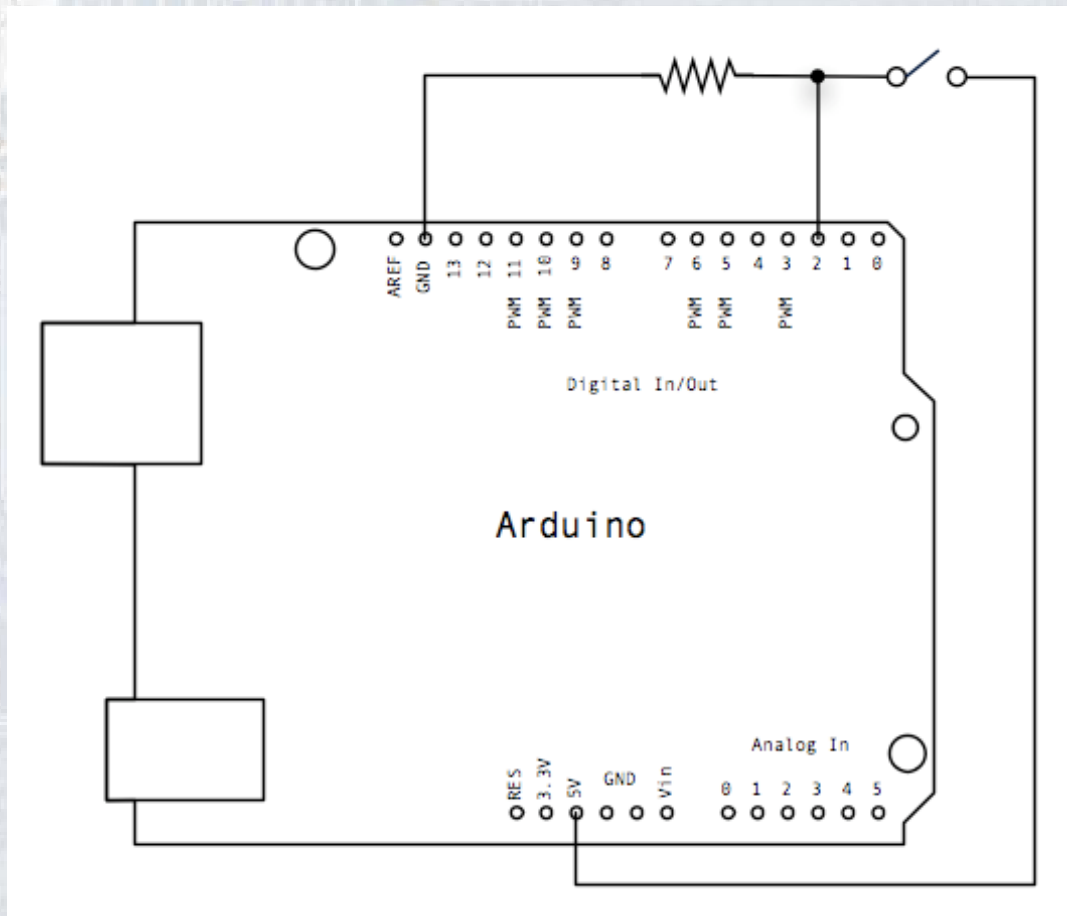
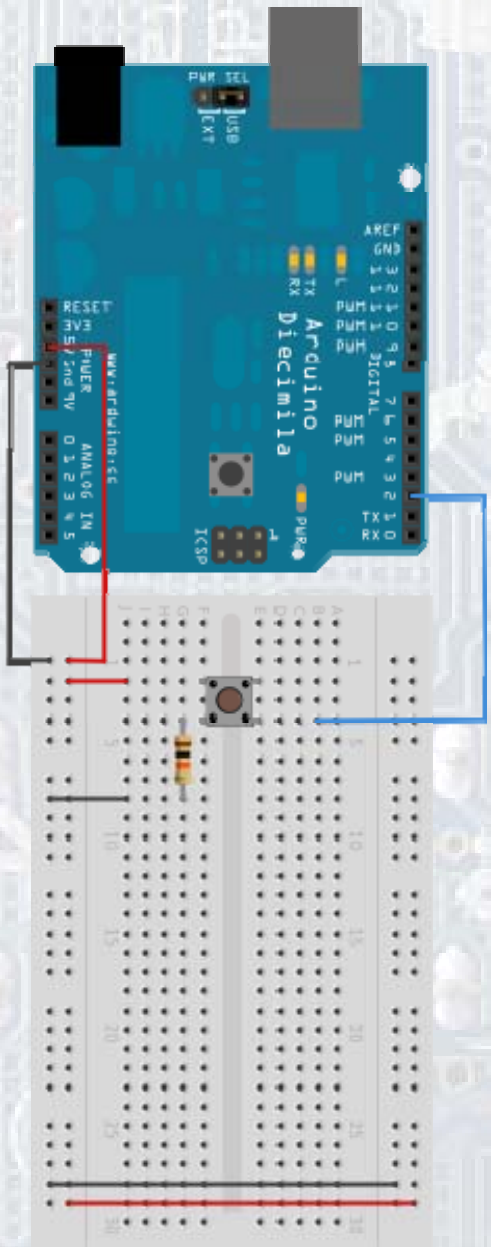
```
void setup() {  
  pinMode(3, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(3, HIGH);  
  delay(500);  
  digitalWrite(3, LOW);  
  delay(500);  
}
```

Play around with the delay length

Create a variable blinking sequence

Add more LEDs

Digital Input: Controlling your LED with a button



Digital Output: Code

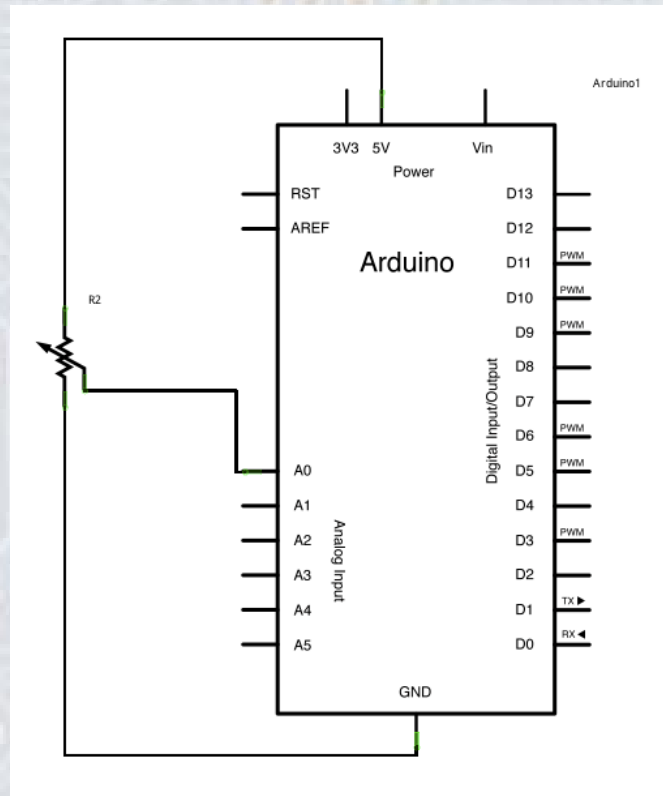
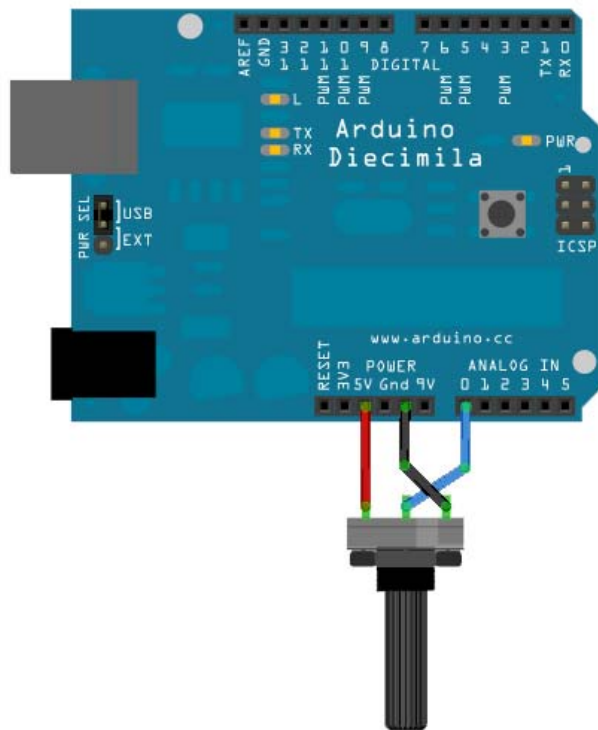
```
void setup() {  
  pinMode(3, OUTPUT);  
  pinMode(2, INPUT);  
}  
  
void loop() {  
  if (digitalRead(2) == HIGH)  
  {  
    digitalWrite(3, HIGH);  
  } else  
  {  
    digitalWrite(3, LOW);  
  }  
}
```

How IF statements work

Switch the result produced by pressing the button

== VS. =

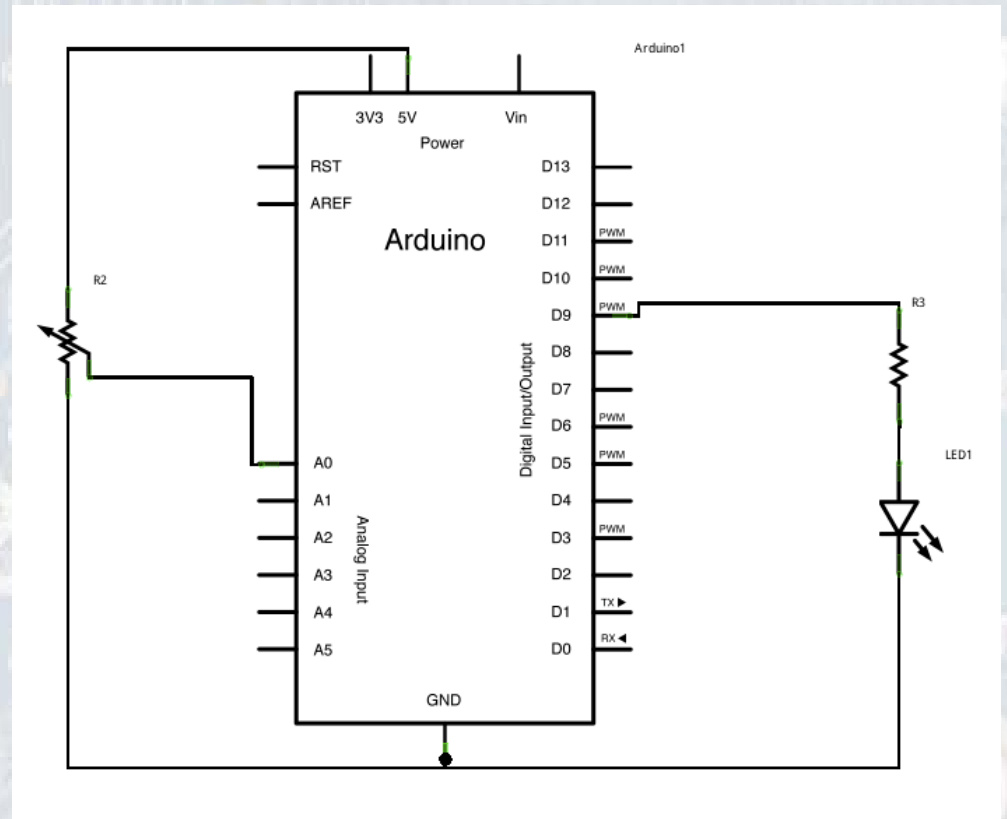
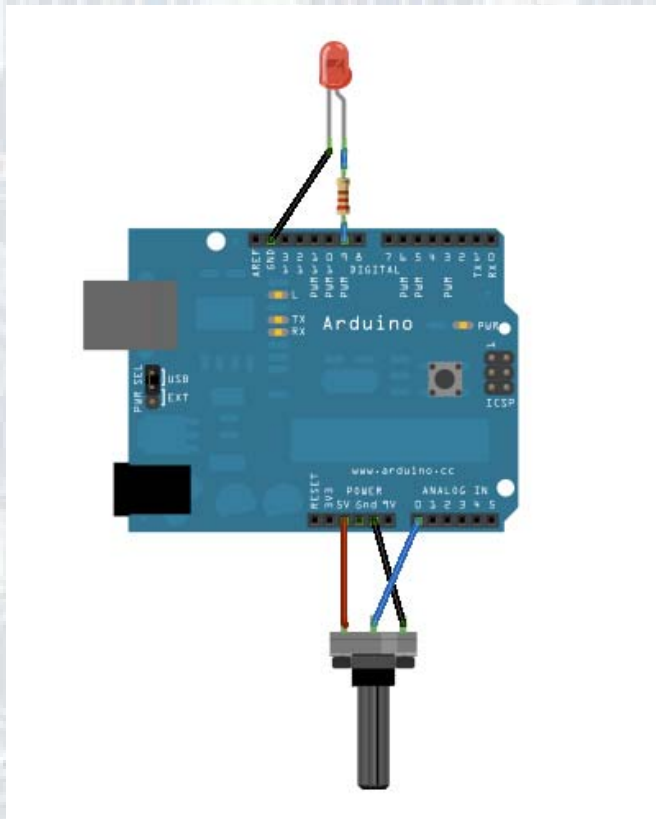
Analog Input: Reading a variable resistor using Serial



Analog Input serial read: Code

```
void setup() {  
  //no setup required for analog inputs  
  //need to set up Serial  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println(analogRead(0));  
  delay(30);  
}
```

Analog Input: Controlling LED with a potentiometer



Analog Input: Code

```
int sensorValue = 0;      // value read from the pot
int outputValue = 0;     // value output to the PWM (analog out)

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  // change the analog out value:
  analogWrite(analogOutPin, outputValue);

  delay(10);
}
```

Expanding the capabilities of the Arduino board

Using external power supplies

- ➔ USB vs. DC adapter
- ➔ Vin pin vs. 5V pin

Using shields: modular pre-made circuit boards for various functions (touch screen, ethernet, blue tooth, and much more)

