

Arduino

Introduction to the interaction
between software and hardware

{coding&&community}

Thanks to Arduino Education for the awesome content!

Lesson adapted from:

<https://www.arduino.cc/en/Guide/Introduction>

{coding&&community}

1

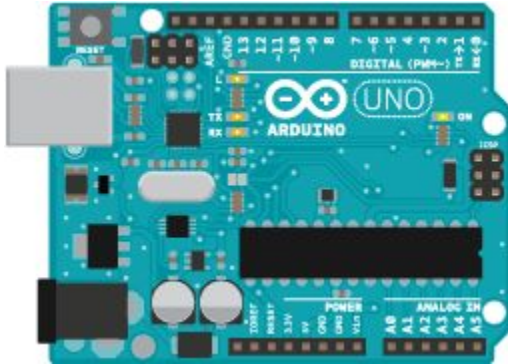
Hardware Introduction



What is Arduino?

Open-source electronics platform for hardware and software.

Hardware



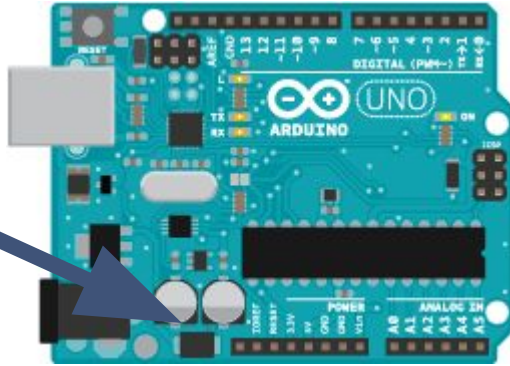
Arduino board

Software



Arduino IDE

User enters
input --
pressing a
button



Hardware sends
input to software

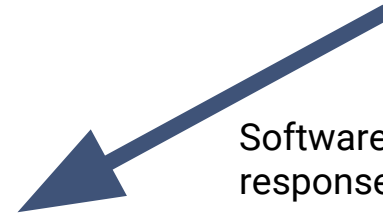


Software processes
input

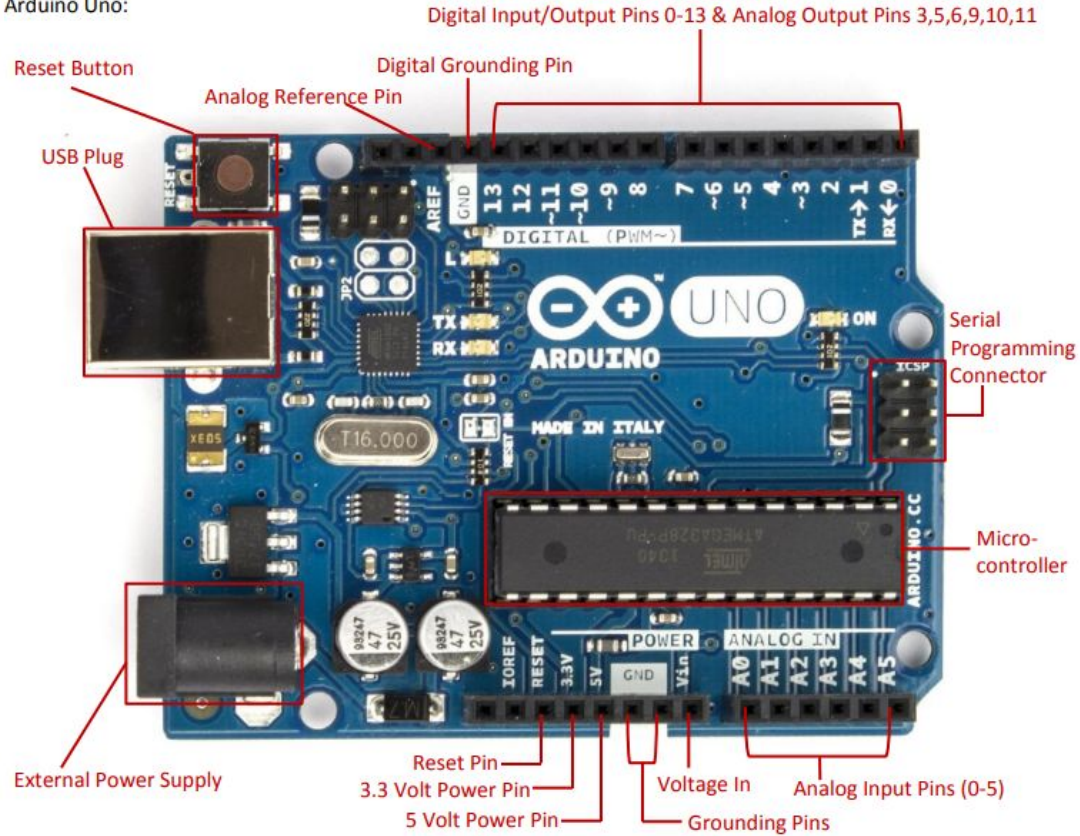
Hardware displays
output -- light blinks



Software sends
response to hardware



Arduino Uno:



Circuits

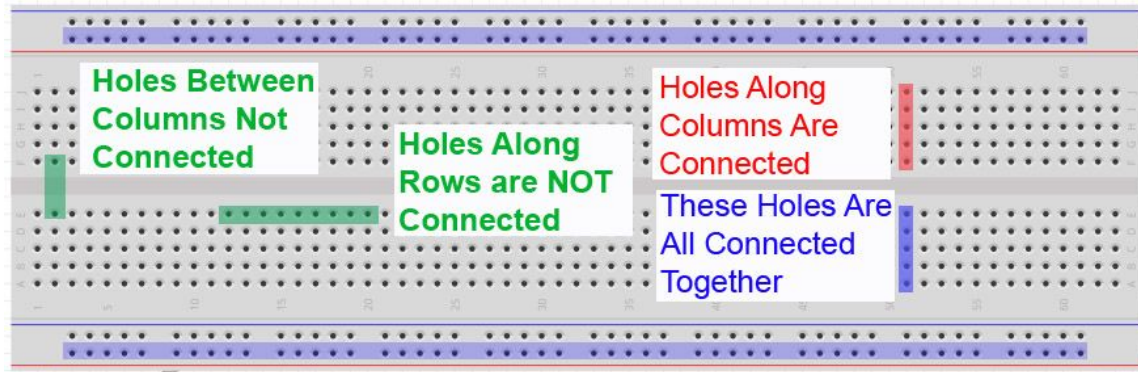
- A **circuit** is a complete loop which connects a **power source**, through a device/load, and back to the the power source.
- All circuits have basic parts, called **components**.
 - **Power Source/Voltage Source**: pushes the electricity through the circuit.
 - **Connectors**: connect all the parts of the circuit and create the path or loop that the electricity travels through (eg. wires)
 - **Load**: The thing being powered by the electricity in a circuit. (eg. LED)





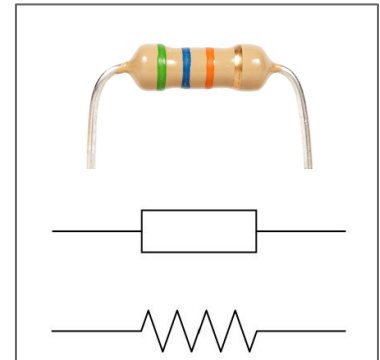
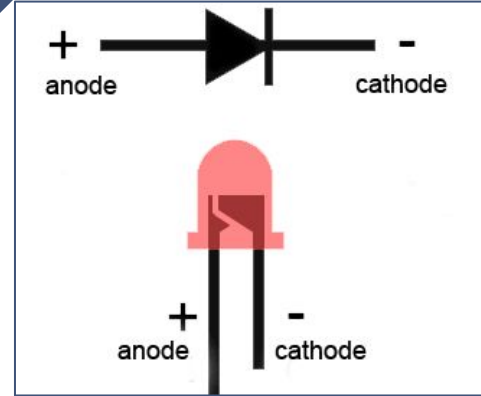
Breadboards

- A breadboard lets you build circuits without soldering
- **Components** are pushed into the sockets on the breadboard and then extra **jumper wires** are used to make connections.



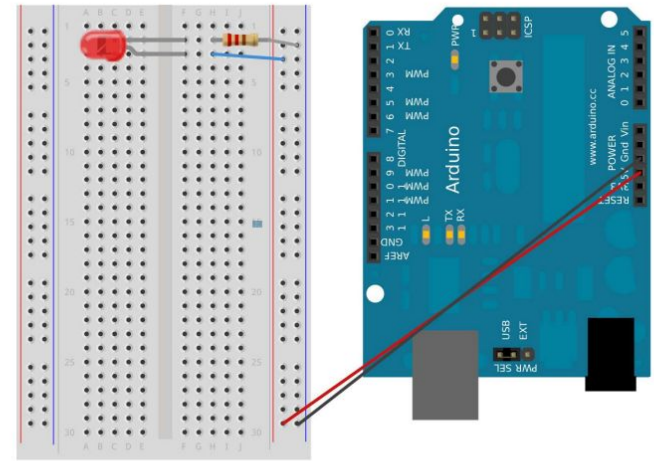
LEDs and Resistors

- **LED** stands for **Light Emitting Diode**. LEDs are **Polarized**:
 - Only allows electricity to flow through it one way (from + to -)
- If you connect the LED directly to power and ground, too much current will go through the diode and destroy it
 - Please don't do this
- **Resistors** are used to limit current
 - The **higher the value** of the resistor, **less electricity "flows"** through.



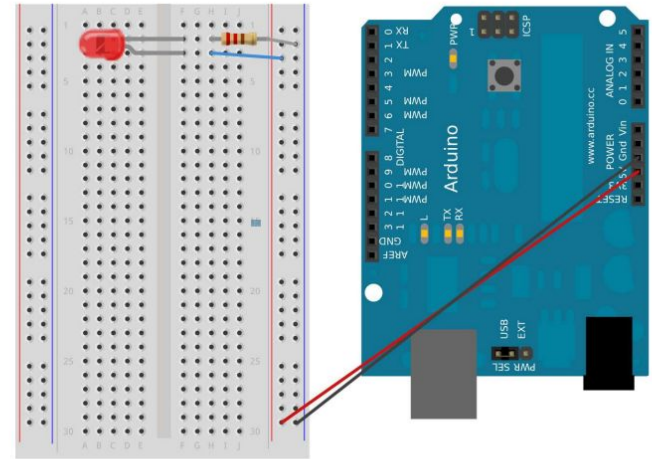
Our First Circuit!

- With a wire, connect ground from the Arduino (labeled GND) to the bottom row of the farthest right column of the breadboard.
- With a wire, connect power from where it says 5V (the V stands for voltage and this is where the electric power comes from.) on the Arduino to the bottom row of the next to right column.
- Connect the resistor with one end in h2 and the other end on the far right column (ground).



Our First Circuit!

- Connect the LED cathode (shorter leg) to f2. (This makes it connect to the resistor through the breadboard because they are on the same row.)
- Connect the LED anode (longer leg) to f3.
- Connect a wire from h3 to the next to right column (+5V).
- Plug power into the Arduino .
- The LED should light up. If it doesn't, unplug power from the Arduino, check all of your connections and make sure you have not plugged the LED in backwards. Then try power again.

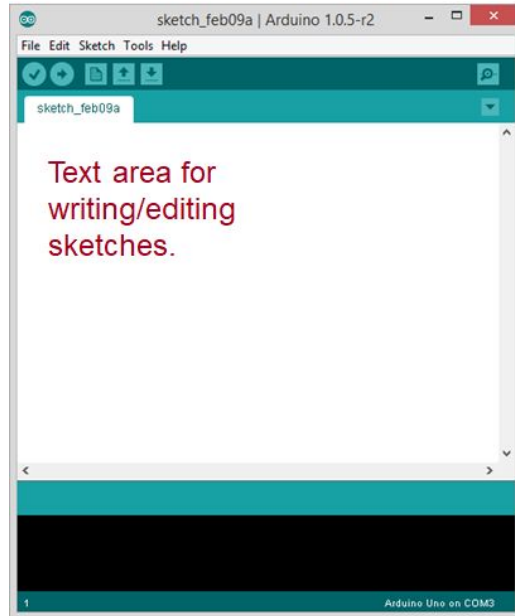


2

Programming an Arduino









Arduino IDE



← Name of current sketch

← Main menus

← Action buttons/icons

-  Verify (AKA compile)
-  Upload (send to Arduino)
-  Start a new sketch
-  Open a sketch (from a file)
-  Save current sketch (to a file)
-  Open Serial Monitor window

← Error messages and other feedback show up here.



Arduino Sketches

- A **sketch** is the name that Arduino uses for a program. It's the unit of code that is uploaded to and run on an Arduino board. Sketches are written in a programming language similar to C and C++
- It has two main functions:
 - **setup** and **loop**
 - You need both functions

```
void setup(){  
    //This code runs once when the sketch starts.  
    //It's a good place to do setup tasks like  
    //setting pin modes or initializing libraries.  
}  
  
void loop() {  
    //This is the main code. This function is called  
    //over and over after setup() is called.  
}
```

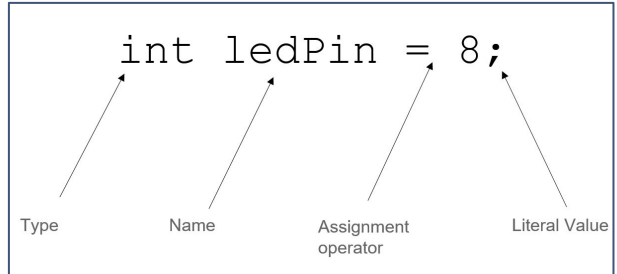


Basic Arduino Commands

- **pinMode(pin, mode)**
 - Sets pin to either INPUT or OUTPUT
- **digitalRead(pin)**
 - Reads HIGH or LOW from a pin
 - input
- **digitalWrite(pin, value)**
 - Writes HIGH or LOW to a pin
 - output
- **delay(value)**
 - Pauses the program for the amount of time in milliseconds (1/1000th of a second)

Constants:

- **HIGH:** 5V or on
- **LOW:** 0V or off



Our First Arduino Program!

- Write a sketch to make the LED blink

```
//declare an variable for the pin number 13  
  
void setup()  
{  
    //set pin 13 (input or output?)  
}  
  
void loop()  
{  
    //add code to make the LED blink on and off  
    //every second  
}
```


Extra Exercises

- Change the amount of time the LED is off to 1 second. (Leaving the amount of time the LED is on at $\frac{1}{2}$ second.)
- Change the pin to which the LED is connected from pin 13 to pin 2. (Note that both the circuit AND the program must be changed.)
- Hook up 8 LEDs to pins 2 through 9 (**with resistors**) Modify the code to turn on each one in order and then extinguish them in order.
 - **HINT:** hook them up one additional LED at a time and make sure the new one works before you add the next one.