00 - Intro to Arduino

Ardu-what?! An Arduino is a small computer that you can program to do a number of things: blink lights, buzz, turn motors, etc. It can be used to build a robot, setup a home burglar alarm, or send you a tweet when someone opens your refrigerator at home.

ARDUINO™

SKILL REQUIREMENTS

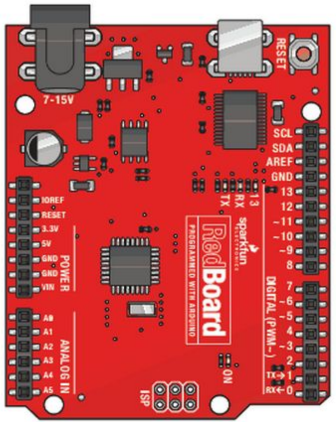
ELECTRICAL PROTOTYPING	■□□□□
ROBOTICS	□□□□□
SOLDERING	□□□□□
PROGRAMMING	■□□□□
DIY	□□□□□

MATERIALS LIST

- Computer
- Arduino UNO
- USB Cable
- LEDs

STEP 1: What is Arduino?

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment), that runs on your computer, and is used to write and upload computer code to the physical board.



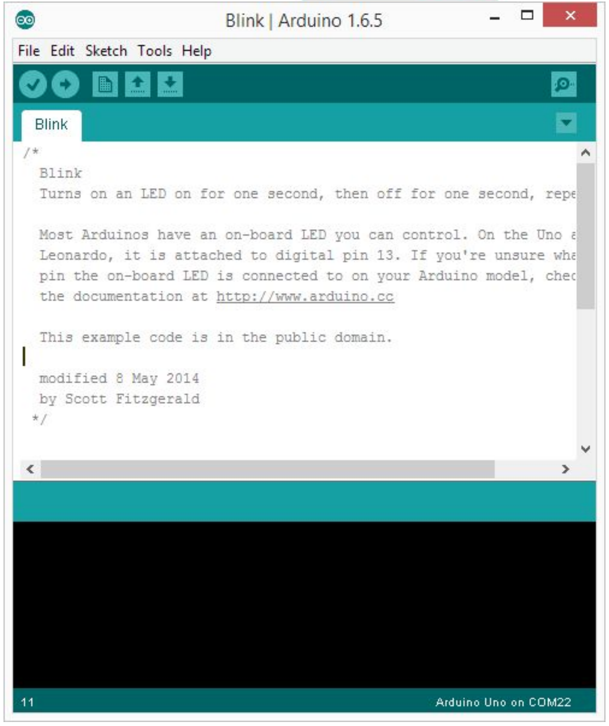
STEP 2: Pins?

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire). They usually have black plastic **headers** that allow you to plug a wire right into the board. The Arduino has several different kinds of pins; each is labeled on the board and used for different functions.

- Pins 0-13 Digital Pins (3,5,6,9,10,11 are PWM)
- Analog Input Pins (A0-A5)
- Power (5V, 3.3V, GND)

STEP 3: The IDE

The Arduino **IDE** (integrated development environment) uses a simplified version of C++, making it easier to learn to program. The Arduino IDE is a free text-based programming software! The IDE is where you write all of the instructions (code) for the Arduino to do something like blink a Light Emitting Diode (LED).



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repe

  Most Arduinos have an on-board LED you can control. On the Uno a
  Leonardo, it is attached to digital pin 13. If you're unsure wha
  pin the on-board LED is connected to on your Arduino model, chec
  the documentation at http://www.arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
 */
```

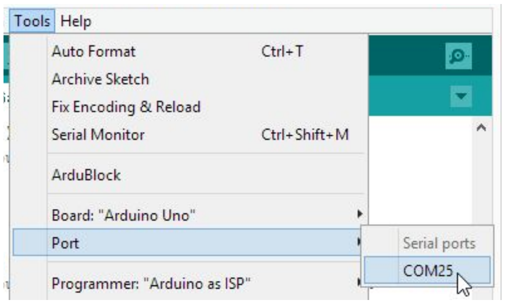
STEP 4: The Hookup

The first thing you are going to do with your Arduino to hook it up to a USB cable and plug the cable into your computer. When this is completed successfully the green “ON” LED. If your Arduino is new out of the box the “13” LED will also be blinking once a second. Note: If the LED doesn’t light up, that’s OK, it just means it still has some older code on it!

DRAFT- UNRELEASED

STEP 5: Boards and PORTs

We are almost there! Open the Arduino IDE on the computer you are using. Before you do anything else, go up to the Tools pull down menu at the top. Click on it and select the “Board” option. Select which model of Arduino you have...chances are it will be an UNO! Next, Click on Tools -> Port and select the highest COM # that you see. Chance are that will be the Arduino that you plugged in. Now you are ready to code!



STEP 6: Hello World!

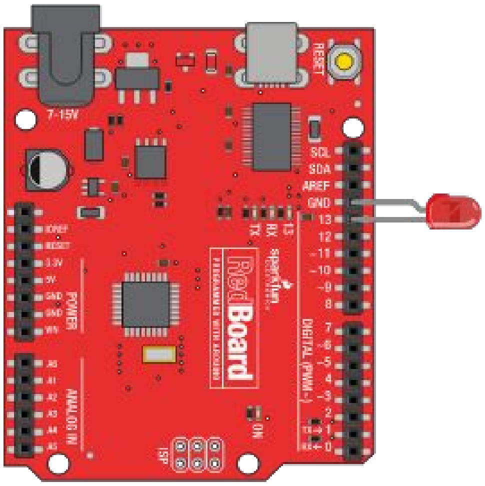
Now we are now going to *write* some basic code! But, before we write anything from scratch, let’s take a look at some of the examples in Arduino. In the IDE, go to **File>Examples>01.Basics>Blink.** Once you have opened this code, click on the Arrow button at the top to upload it to your board. If all goes well you will see a number of LEDs blinking like crazy. After things settle down, do you notice anything different? Check your LED 13. Is it blinking? What is the timing of the blink?

In your code, change the numbers inside of the `delay()` functions to 1500 and hit the arrow again. What happens? What changed?

STEP 7: Play!

Now that you got the onboard LED to bend to your will, let’s try adding your own LED to the board. Take an LED, find digital pin 13 and insert the longer leg into pin 13 and the shorter leg into GND, which is right next to 13. What happens?

What happens if you flip the LED around? Each “circuit” must have a path for electricity to go from a power source, through a device, and back to ground (which is the negative side of the power). Want to add more LEDs? Ah - just wait to our next activity.



Made with Fritzing.org

STEP 8:

Take what you know so far, take a piece of code that transforms the simple blink above and rewrite it to portray your interpretation of a heartbeat. ***Clue:*** *You may have to write more digitalWrite functions; HIGH means ‘on’ and LOW means…???*

TAKING IT FURTHER

- ... --- ... : Do you know Morse Code? No? Well, that’s okay. Look up some hints online and see if you can get the LED to blink your name. If you have a long name - pick a word or shorter name. Translate it to “dots” (short blips) and “dashes” (long blips). See if your friend across the room can decipher your message!
- Knowing what you know about breadboards, how would you add multiple LEDs?
- Create a different link pattern over multiple LEDs.
- SparkFun Getting Started with Arduino Video

sparkfun ELECTRONICS

Visit us at: learn.sparkfun.com