Getting to know your Arduino:

Basic electronics

You can’t do much with Arduino unless you have the basic skills needed for building circuits. Every circuit needs a power source, connections, and working components. When you’re working with Arduino, your power source will often be the Arduino, hooked up to a battery pack or computer. Your connections will usually be jumper wires (and sometimes breadboards), and your components will often be things such as LEDs/lights, chips, sensors, and other basic components such as resistors and capacitors.

For these projects you’ll be using LEDs, a button, a photoresistor, and a tilt ball switch as your components. You’ll also need your Arduino (preferably powered by connecting it to your computer with the USB Cable suited for that purpose, a breadboard, and jumper wires.

There are a few forms in which you’ll see the circuits for these projects: photos, circuit schematics, and circuit diagrams. I’ll probably try to stick to using photos and circuit diagrams, but it’s still important to know how to read circuit schematics.

# Interpreting Circuits

## Reading Circuit Schematics

A circuit schematic is a diagram that shows what components a circuit consists of and how they’re connected. It consists of symbols representing components, connected with lines, representing wires or connections.

Figure 1, shown below, depicts some common schematic symbols:

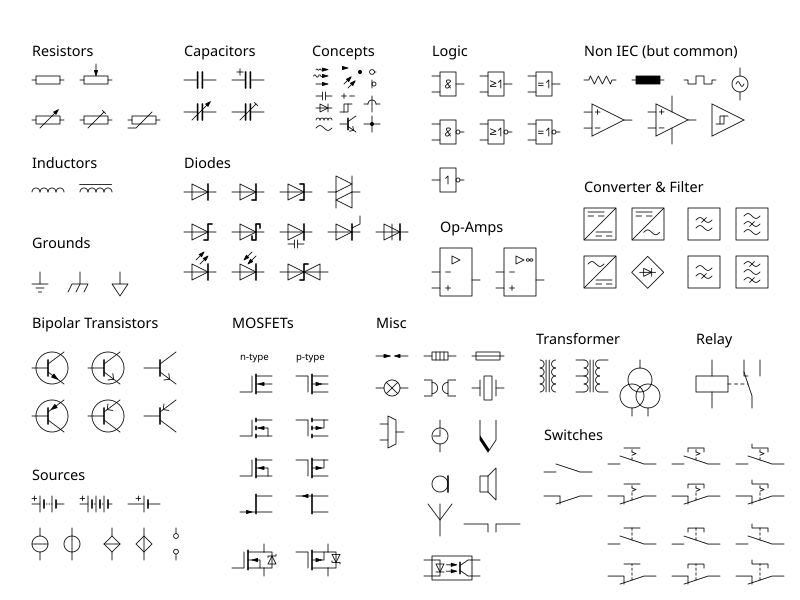


Figure : common schematic symbols

Another thing you’ll often see with circuit schematics is abstraction, where a component is either depicted with its own special symbol (Figure 2) or as a box with its connections labeled (Figure 3)

A diagram of a circuit

Description automatically generated

Figure : Alternative schematics for an OR gate

A close-up of a circuit board

Description automatically generated

Figure : an abstracted Arduino

In addition to showing components, schematics need to show connections for their respective circuits:

A circuit board with wires and a few other wires

Description automatically generated with medium confidence

Figure : schematic for a simple circuit

Note that a circuit schematic often looks quite different from the circuit it represents. The schematic for the circuits under a car’s hood often looks relatively simple or well-organized, but looking under the hood of most cars yields more confusing results: wires running here and there, with hardware twisting this way and that.

## Interpreting Circuit Diagrams

Circuit diagrams can be a lot simpler, because they show simplified versions of the components and how they’re wired. These diagrams are often helpful because unlike circuit schematics, they often show how a circuit might actually be laid out. They also often show the actual components’ colors and/or markings, as opposed to schematics, which only tend to use coloring for color-coding, which isn’t that common in the first place. Remember that circuit diagrams still don’t always show circuits as they are built- they just show the circuits in a more human-friendly fashion.

A blue circuit board with wires connected to it

Description automatically generated

Figure : Diagram for circuit in Figure 4

## Circuit Photos

As long as they’re well done, photos can be an excellent source for seeing how a circuit is built by showing you how the finished product looks. They’re not usually too complicated to interpret, but they can often be confusing still. It’s good to have a photo showing a finished circuit as an extra reference, but it’s important to have other methods of showing circuits alongside the photos.

# Project 1: building a simple circuit

Now that you (hopefully) understand circuitry a little better, it’s time to try building a circuit yourself!

## Materials

You won’t need too much for this project. For now you’ll only need the following materials:

* Arduino Uno R3 microcontroller, hooked up to a power source
* 3 jumper wires
* 1 breadboard
* 1 button
* 1 LED
* 1 220-Ω (220-Ohm) resistor

## Building Instructions

For this project you’ll be building the circuit shown in Figure 5. Pay attention to the direction of the LED: if you arrange your components as shown in the image, the long pin of your LED should be on the right side, not the left. If you put it in backwards, it won’t turn on.

## Operating Instructions

If you haven’t already, plug in the Arduino board. If everything works correctly, the LED should turn on whenever you press the button. If it does, then congratulations! You’ve created a working circuit using an Arduino!