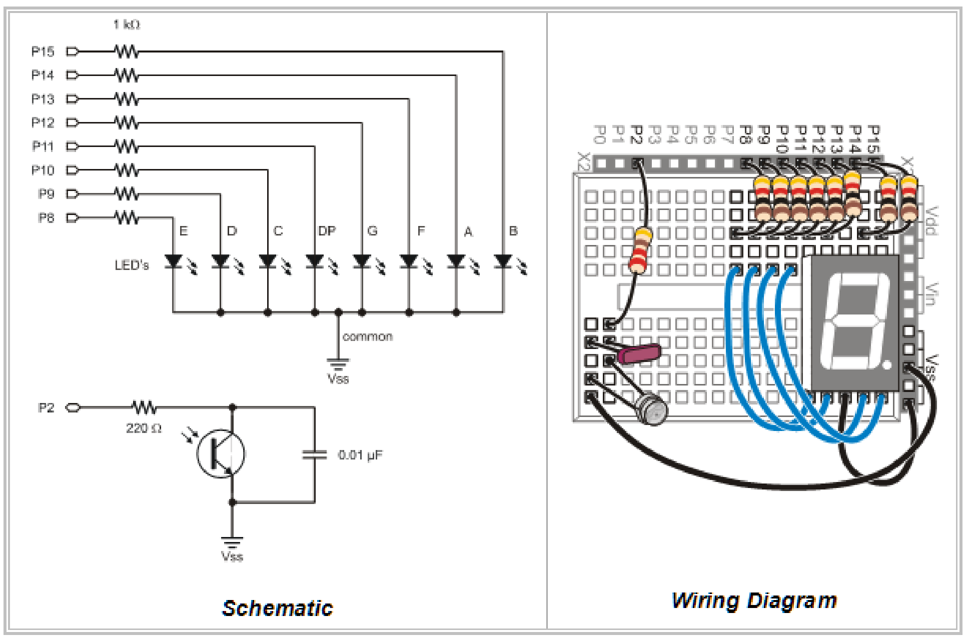
How to Wire Circuits from Schematics

View Video Introduction (**YouTube**)

In electronics, the schematic is a drawing that represents a circuit, and is used to communicate designs and ideas. Knowing how to draw and read a schematic is very important as you venture into the world of electronics, particularly if you want to share your design with fellow electronic enthusiasts or if you want to use a circuit designed by someone else.

If you have gone through any of our Stamps in Class texts, you've probably come across various schematics and wiring diagrams showing you how to build the circuits needed for your program to run. Until now you may have relied solely on the wiring diagram, or the picture that shows you exactly which breadboard sockets to connect your electronic components to. However, in the "real world" of electronics, wiring diagrams aren't generally used to portray circuits; so you'll have to learn how to read the schematic instead.



At first glance, a schematic might seem somewhat daunting, and you might be confused by all of the symbols and lines. But have faith! Even the most complex schematics can be easily wired if broken into small pieces. Today, we'll get you up to speed on how to read schematics, and how to wire circuits from them on a breadboard.

Getting Started

This tutorial demonstrates how to wire circuits from schematics on a breadboard. If you have never worked with a breadboard before, or just aren't completely sure how one works, you can view a tutorial video on the basics of breadboarding here.

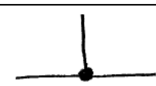
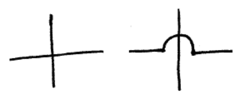
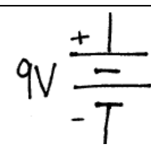
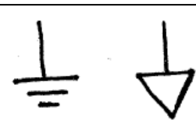
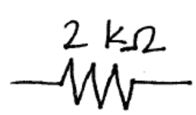
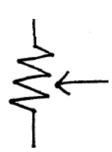
Parts Required

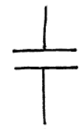
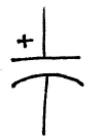
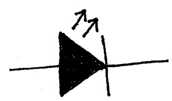
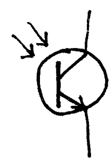
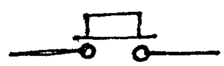
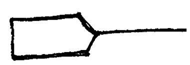
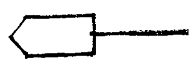
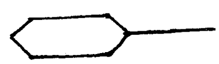
(1) 9V Battery Clip
(1) LED (green)
(1) 470Ω resistor
(1) Tact Switch (aka pushbutton)
(1) Potentiometer

<u>Schematic Symbols</u>

Every schematic is made up of different symbols, and these symbols represent particular electronic components. Lines are then drawn to show how each component connects to others in the circuit.

The table below lists some common electronic components, their function, and the symbols used to represent them in schematics. Unfortunately, the best way to get familiar with these symbols is simply through route memorization, but once you start working with these components more often, you'll get the hang of things in no time!

Component Name	Schematic Symbol	Function
Wire		Represents an electrical connection
Wires (Connected)		Represents a shared electrical connection between two components
Wires (Not Connected)		Represents wires that cross in a schematic for aesthetic purposes, but do not connect in the circuit
DC Supply Voltage (Battery)		Represents how much voltage to supply to your circuit, where: Long lines = Positive battery terminals Short lines = Negative battery terminals
DC Supply Voltage (General)		Represents how much voltage to supply to your circuit, may also be labeled as Vcc, Vdd, or Vin
Ground		Zero Volts, may also be labeled as Gnd, Vss, or Vee
Resistor		Limits electrical current, value is usually included near the symbol
Variable Resistor		Also called a potentiometer, where the value of the resistor depends on the position of an internal wiper

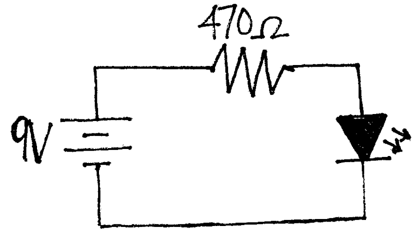
Capacitor		Stores electrical energy
Polarized (Electrolytic) Capacitor		Stores electrical energy and can only be connected in a circuit one way. The positive lead of a polarized capacitor is represented by the plus sign.
Light Emitting Diode (LED)		Converts electrical energy to light, commonly used to indicate the status of a circuit. The positive terminal (anode) is the flat spot of the triangle on the left side.
Transistor (NPN)		Controls current flow
Phototransistor		Restricts or allows current to flow proportional to the amount of light detected
Tact Switch		Also known as a pushbutton, allows current to flow through a circuit only when button is pressed
Input Pin		Represents a circuit that inputs signals to a microcontroller pin
Output Pin		Represents a microcontroller pin that outputs signals to a circuit
Bi-Directional Pin		Represents a microcontroller pin that can function as an input or output

Wiring Circuits from Schematics

Now that we've got a general understanding of the schematic symbols for many common electronic components, it's time to see how everything is put together. Each of the examples below will include components that will allow you to see instantly if you've wired the circuit correctly, and will also take you step-by-step through each step of the wiring process.

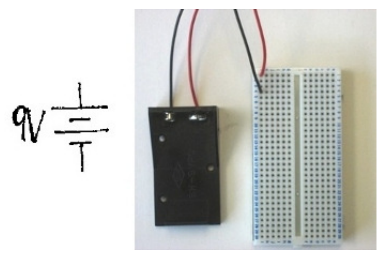
Starting Small: Turning on an LED

Let's start out with a simple LED circuit. Below is the schematic we will be using:

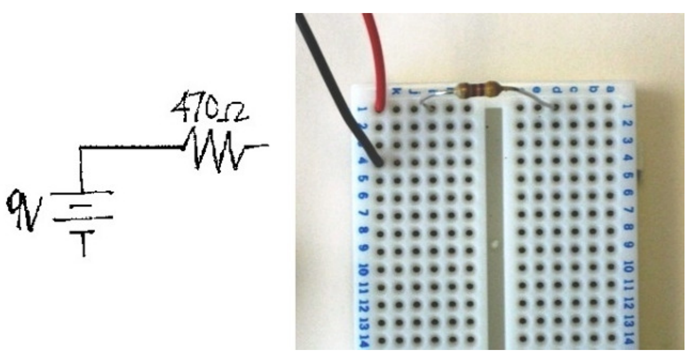


By looking at the schematic we can identify a few symbols from our list: a battery supplying 9V, one 470Ω resistor, and one LED. In order to wire this on a breadboard, it's easiest if you just take each component one at time.

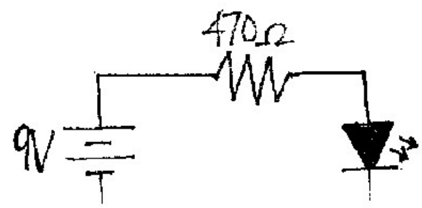
So let's start with the positive and negative terminals of the battery. If using a 9V battery clip, the positive terminal will be the red wire, and the negative terminal is the black wire. You can connect these to any to breadboard sockets that you desire:

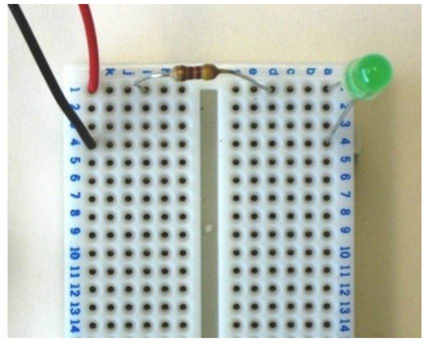


Next, we have one end of the 470Ω connected to the positive terminal (red wire) of the battery, like so:

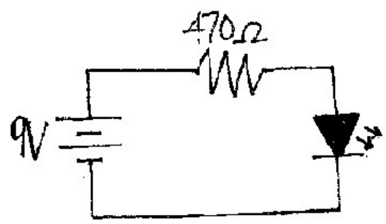


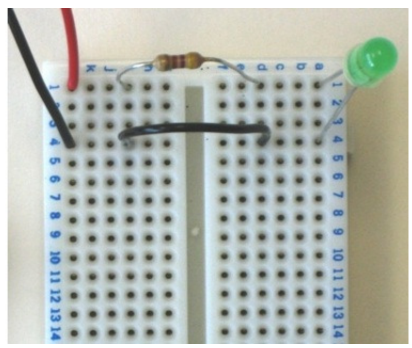
Then, the other end of the resistor is connected to the anode (or long lead) of the LED:





Finally, the cathode of the LED is connected to the negative terminal of the battery, like so:





And congratulations, you've successfully wired a circuit from a schematic! That wasn't so horrible, was it? Now if you plug a battery into the clip, the LED should light up.

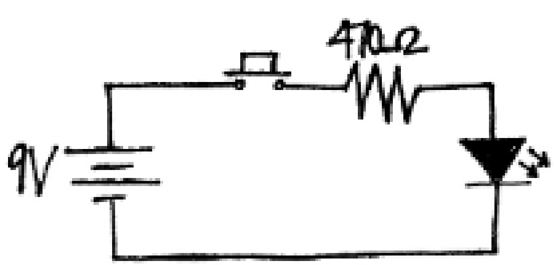
Where's the ground connection?

You may have noticed that there was no ground symbol in this circuit, and that may have struck you as odd especially if you are under the impression that every circuit needs a ground connection.

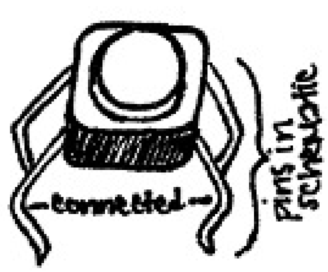
Well this is true, every circuit does need a ground connection, and in this case, the negative terminal of the battery serves that purpose. Ground connections generally allow a place for electric current to return to, so it's not necessary to draw an additional ground symbol when the negative terminal of the battery already does this job.

Controlling an LED with a Pushbutton

Now let's add a pushbutton to our original LED circuit, which will enable us to control when the LED will turn on or off. Here is the schematic we will be using:

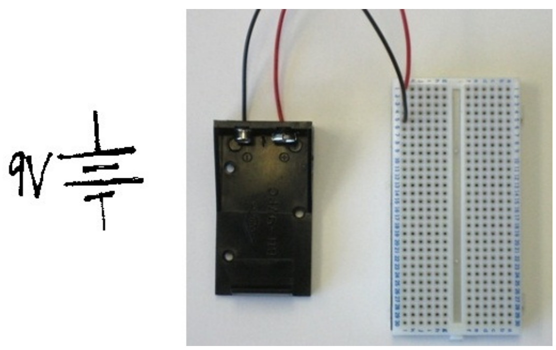


The pushbutton is interesting in that it has four pins, but is only represented in a schematic by two. This is because the pushbutton needs all four pins for stability, but only two for functionality. The two pins not shown in the schematic symbol are electrically connected to the two pins that are shown and therefore not necessary to picture.

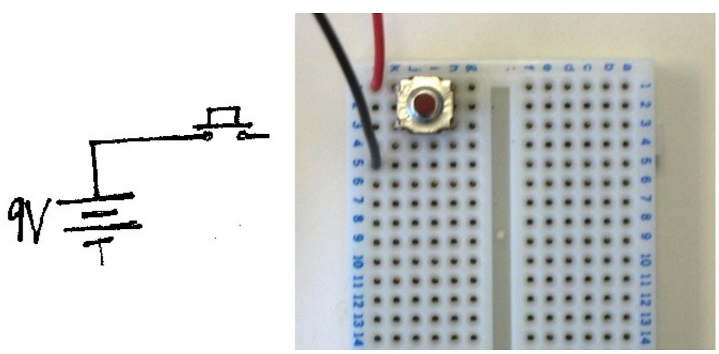


But how does this relate physically to the pushbutton itself? If you take a look at it from one side, you will see that two of the terminals sort of bend toward one another (see the picture on the left). These are the connected pins. Relating this back to the symbol drawn in the schematic – think of the pushbutton symbol being as drawn in 2D, and you're only being shown a side where the two leads aren't connected.

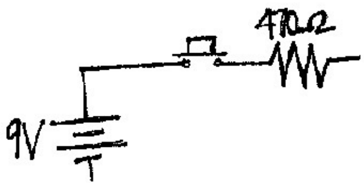
Now how do we go about wiring the above schematic on a breadboard? Just like before, we'll start with the connection from our battery terminals:

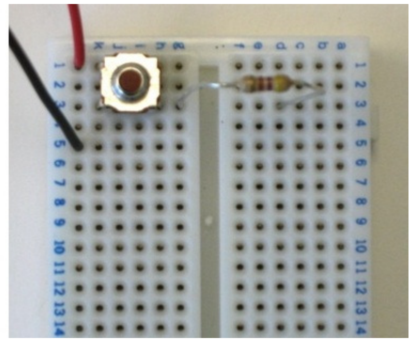


Next, we have one terminal of the pushbutton connected to the positive terminal of the battery, like so:

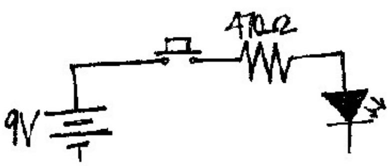


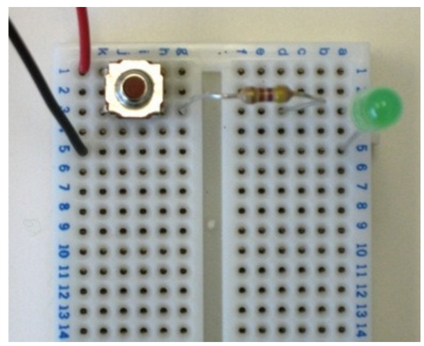
Then, the other terminal of the pushbutton is connected to one end of the 470Ω resistor:



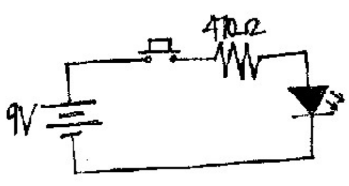


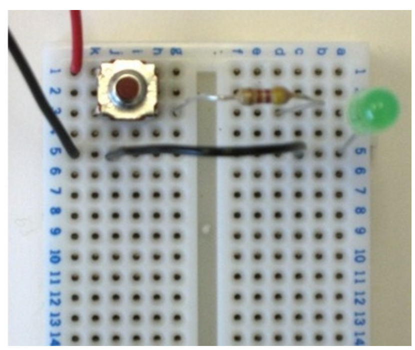
Next, the other end of the resistor is connected to the anode of the LED:





And finally, the cathode of the LED is connected to the negative terminal of the battery:





And you've successfully wired your second circuit! Just connect your battery and verify that when you press the button, the LED lights up.

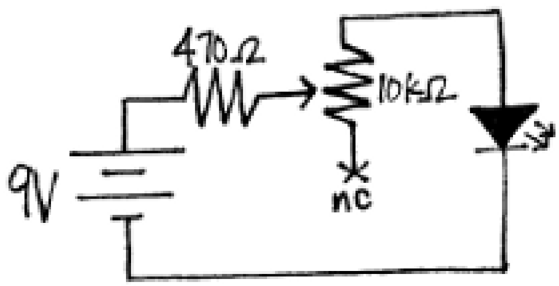
Still confused about the pushbutton?

The pushbutton is a tricky little electronic component to fully understand, and it's OK if you still don't fully "get" the correlation between the component and its schematic symbol at first.

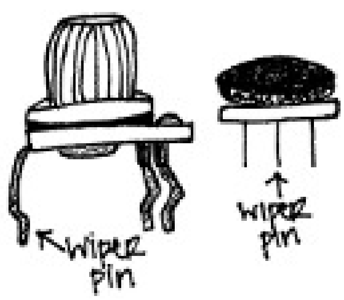
If this is the case, you can also check out Chapter 3, Activity #1 in [What's a Microcontroller] for some more diagrams and explanations. Or, even better, you can mess around with the pushbutton on the breadboard yourself! Try rotating it, or wiring your circuit in different ways so you can really get a feel for what's going on. After all, the best way to learn is through doing!

Controlling LED Brightness with a Potentiometer

In this next example, let's replace the pushbutton from the last circuit with a potentiometer. Here's what the schematic will look like:

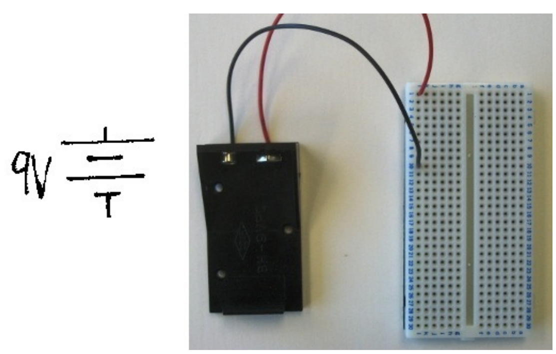


The potentiometer is also known as a variable resistor which, as its name implies, has a resistance that changes. The 10kΩ marking next to the schematic symbol indicates the maximum resistance value the potentiometer has.

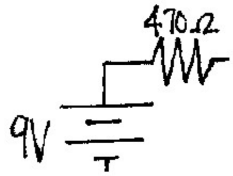


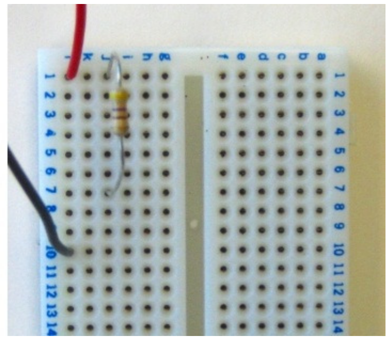
Potentiometers come in all sorts of shapes and sizes, but their functionality remains the same across most types. They have three pins, two of which make up the total resistor value, and a third called the wiper. The wiper is the actual mechanism that changes the resistance value as you rotate the dial. This is normally the center pin on most potentiometers, and an example drawing of two commonly found potentiometers can be seen to the left.

So let's get started and wire this bad boy up! First, as always, we'll start with connecting the two leads of the battery terminal.

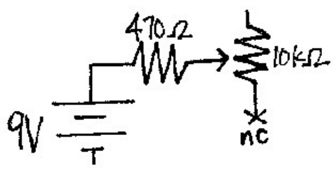


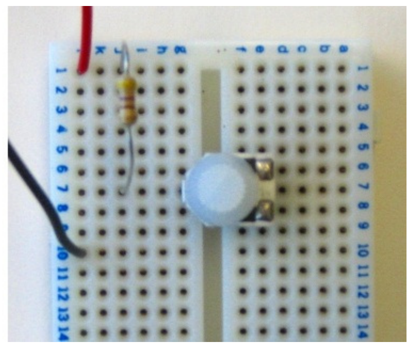
Next, we've got one end of a 470Ω resistor connected to the positive terminal of the battery. This will ensure that there is always at least 470Ω between the battery and the LED.



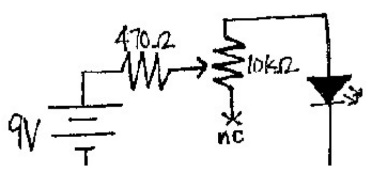


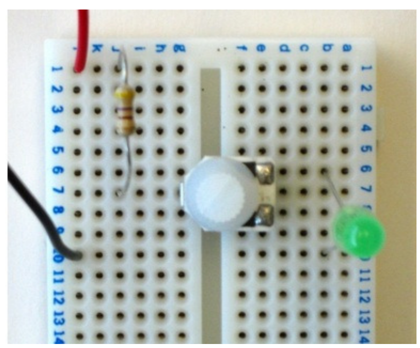
Now it's time to connect the potentiometer. According to the schematic, the wiper pin of the potentiometer is connected to the other end of the resistor.



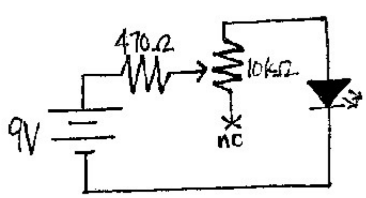


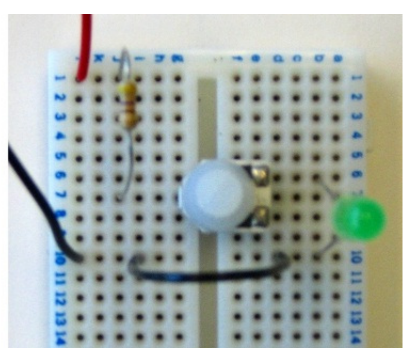
The LED's anode is then connected to the top pin of the potentiometer, like so:





Lastly, the LED's cathode is connected to the negative terminal of the battery to complete the circuit.





Now, when you plug the battery into the clip, the LED should become brighter or dimmer depending on the direction of rotation.

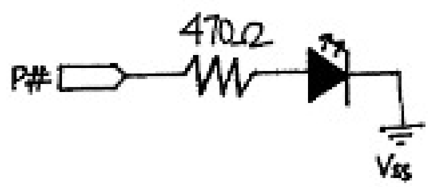
Replacing the Battery with Microcontroller Pins

Chances are that for most applications, you'll want to make your circuit "smart." In other words, tell an LED when to turn on and off, determine whether a pushbutton has been pressed or not, or measure the decay time of a potentiometer. In these cases, we will no longer be using a battery as our power source, but the pins of a microcontroller. It will be very important know which symbol to use, so that other people reading the schematic understand how the circuit operates.

Controlling an LED with a Microcontroller

For example, let's take another look at our original LED circuit. Which I/O pin symbol would be appropriate, and how would the circuit change?

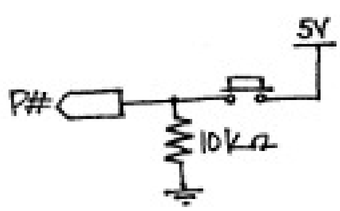
In this instance, the microcontroller pin would be *outputting* a voltage to turn the LED on and off. So, we need the symbol for an output pin. The rest of the circuit will remain the same for the most part: we'll still have our resistor and LED, but we no longer have the negative terminal of the battery to close our circuit, so we just need to add a ground symbol instead.



Reading a Pushbutton State with a Microcontroller

In some applications, it's helpful to know whether a pushbutton has been pressed or not. In these cases, your microcontroller pin would be detecting an *input* voltage. So we'll need to use the symbol for an input pin.

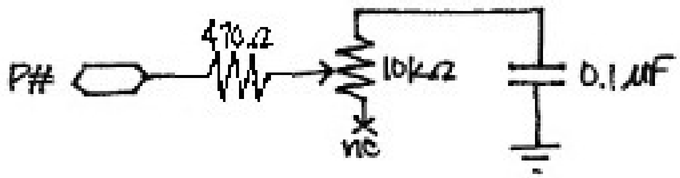
Our circuit will also change a bit compared to the original pushbutton circuit shown above, since we will now need to wire the pushbutton so the pin detects 0V if the button is not pressed and 5V if the button is pressed. The schematic is shown below; can you wire it on a breadboard yourself?



Measuring Decay Time of a Potentiometer

Still other applications may require measuring the resistance of a potentiometer to determine the position of a dial (see What's a Microcontroller Chapter 5). In these cases, your microcontroller pin would be outputting a voltage to charge a capacitor, and then detecting an input voltage to determine the time it takes for the voltage to decay to a certain amount. In these applications, we'll need to use the symbol to a bi-directional pin.

The rest of the schematic will remain similar to the one shown above, except that the LED will be replaced with a capacitor. Once again, the schematic is shown below, and can you wire it on a breadboard yourself?



Going Forward

This tutorial may not make you the world's foremost expert on wiring circuits from schematics, but hopefully it helped build your circuit wiring confidence. Next time you come across an activity with both a schematic and a wiring diagram, challenge yourself and try to wire your circuit only from the schematic. Before you know it, you'll be a schematic wiring machine.

Happy Developing!