Lab 01 - Digital Logic and Gates

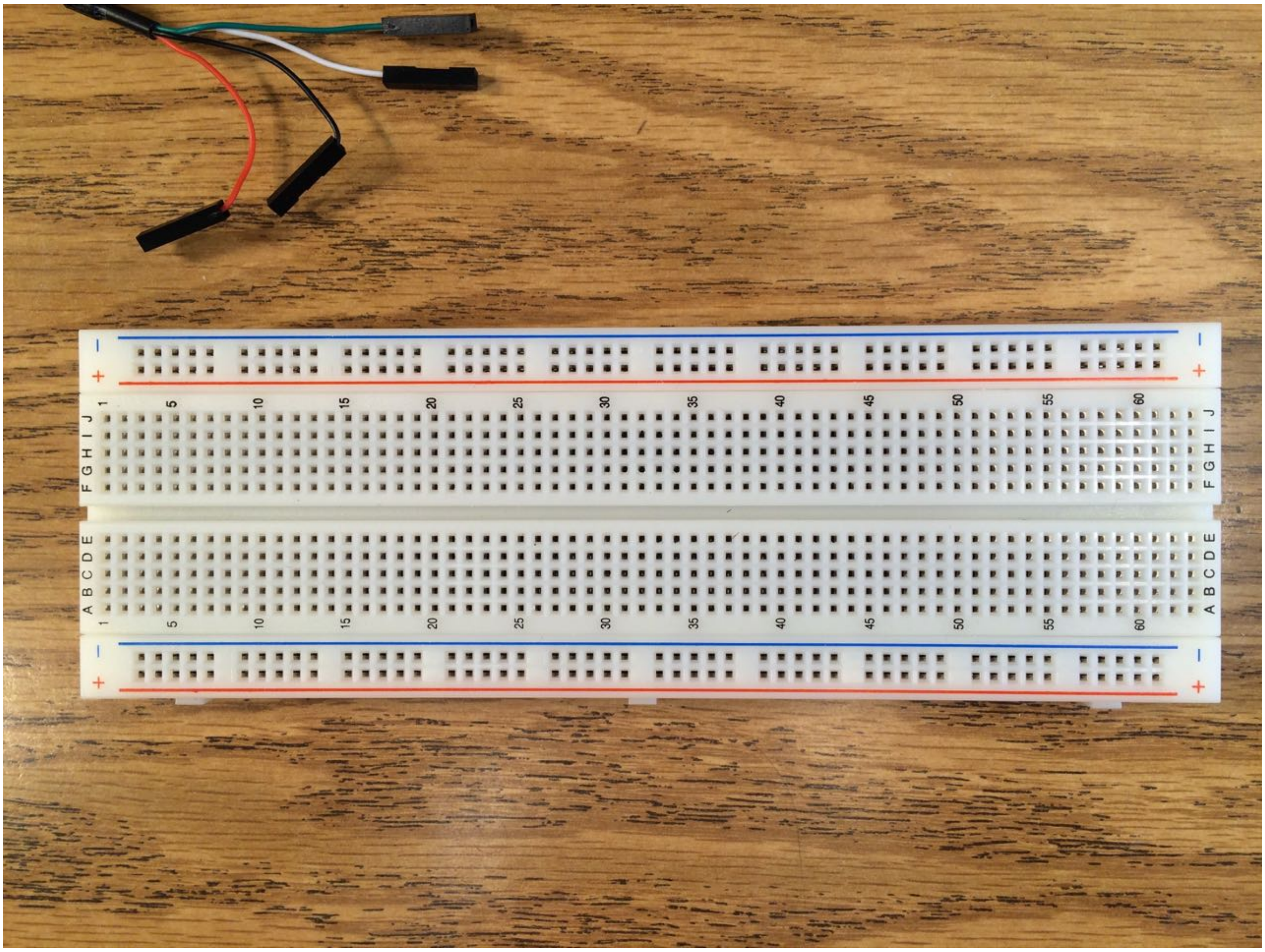
In this lab, you will build a circuit using multiple two-input NAND gates to implement the two-input Boolean algebra XNOR function. The following lab documentation provides instructions about how to build and test this circuit.

Step 1- Preparing Your Breadboard

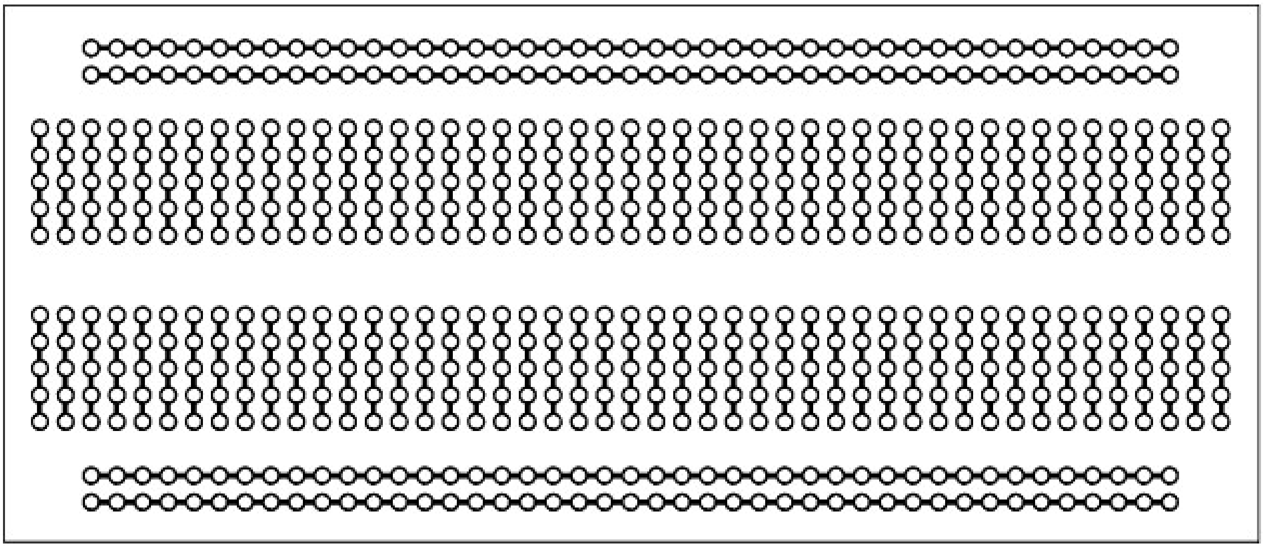
- Add wires to connect the two power buses on either long edge of the breadboard to each other.
- To prevent delivery of excessive current to lab circuits on your breadboard, install the 10 Ohm, 2 Watt resistor from your kit onto the breadboard in series with the USB console cable.
- Then test that power is being supplied to the breadboard using a light-emitting diode (LED) as a probe.

Internal structure of the breadboard

Here is a picture of the breadboard and the breadboard end of the USB console cable used to bring 5 volt power from a USB port to the breadboard.



Across its top surface, the breadboard presents a matrix of sockets into which electrical components can be inserted and removed easily. The breadboard both holds the components of the circuits you build, but it also connects these components electrically when you insert them into the sockets. Sockets are grouped and electrically connected in the pattern shown in the following figure.

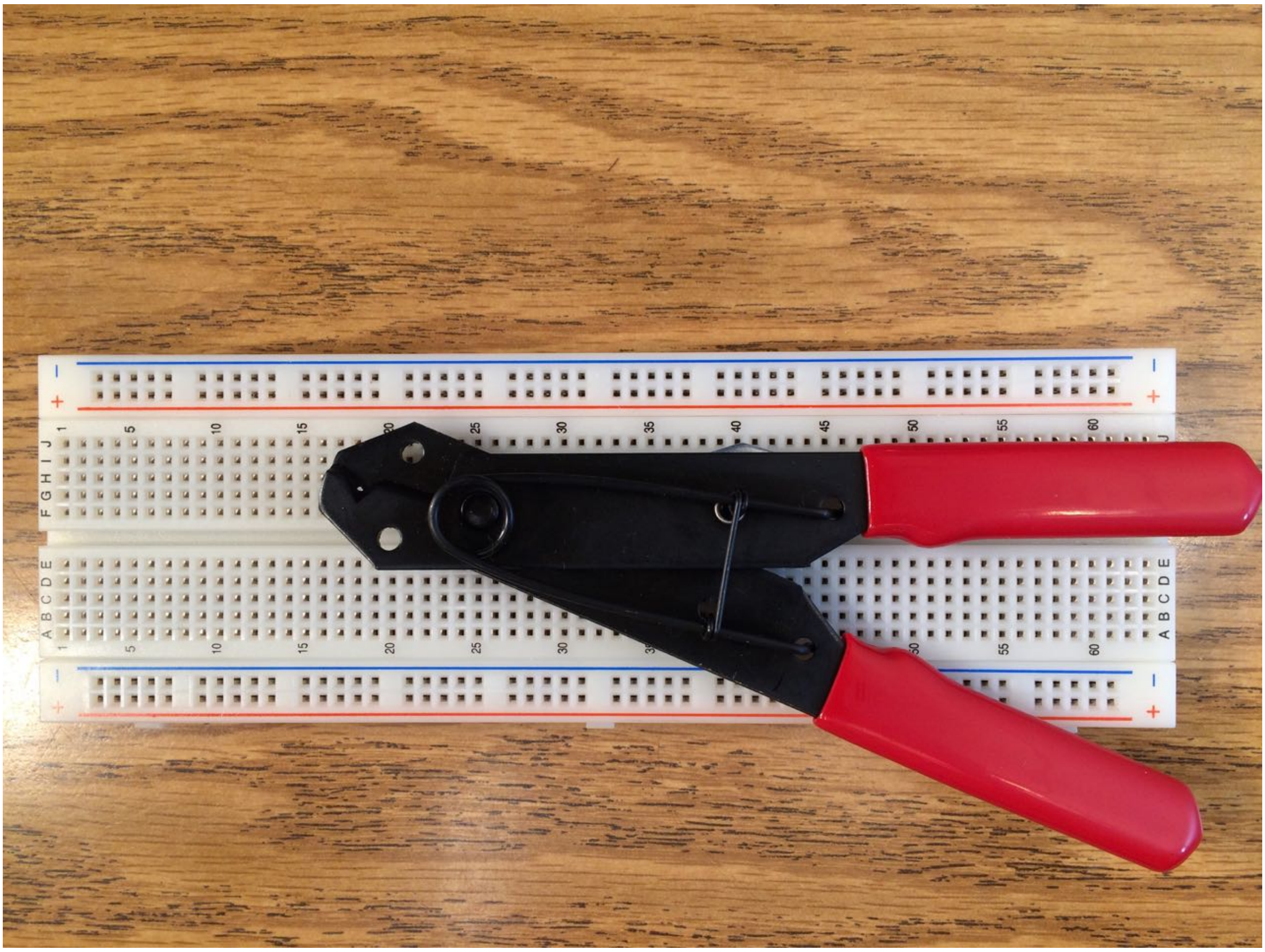


The breadboard is designed specifically for circuits built from integrated circuits (ICs) of the dual-inline package (DIP) family of parts, plus discrete (individual) components such as wires, switches, resistors, and capacitors. To make it easy to build such circuits, the breadboard provides two rows of sockets on either long side intended to be used for power and ground. The columns of sockets and the dividing gutter down the center are for DIP and discrete components. DIP devices are to be inserted parallel to the power rows and spanning the gutter. This provides each pin on either side of the DIP with a column with four sockets available for wires and other devices to connect to that DIP pin.

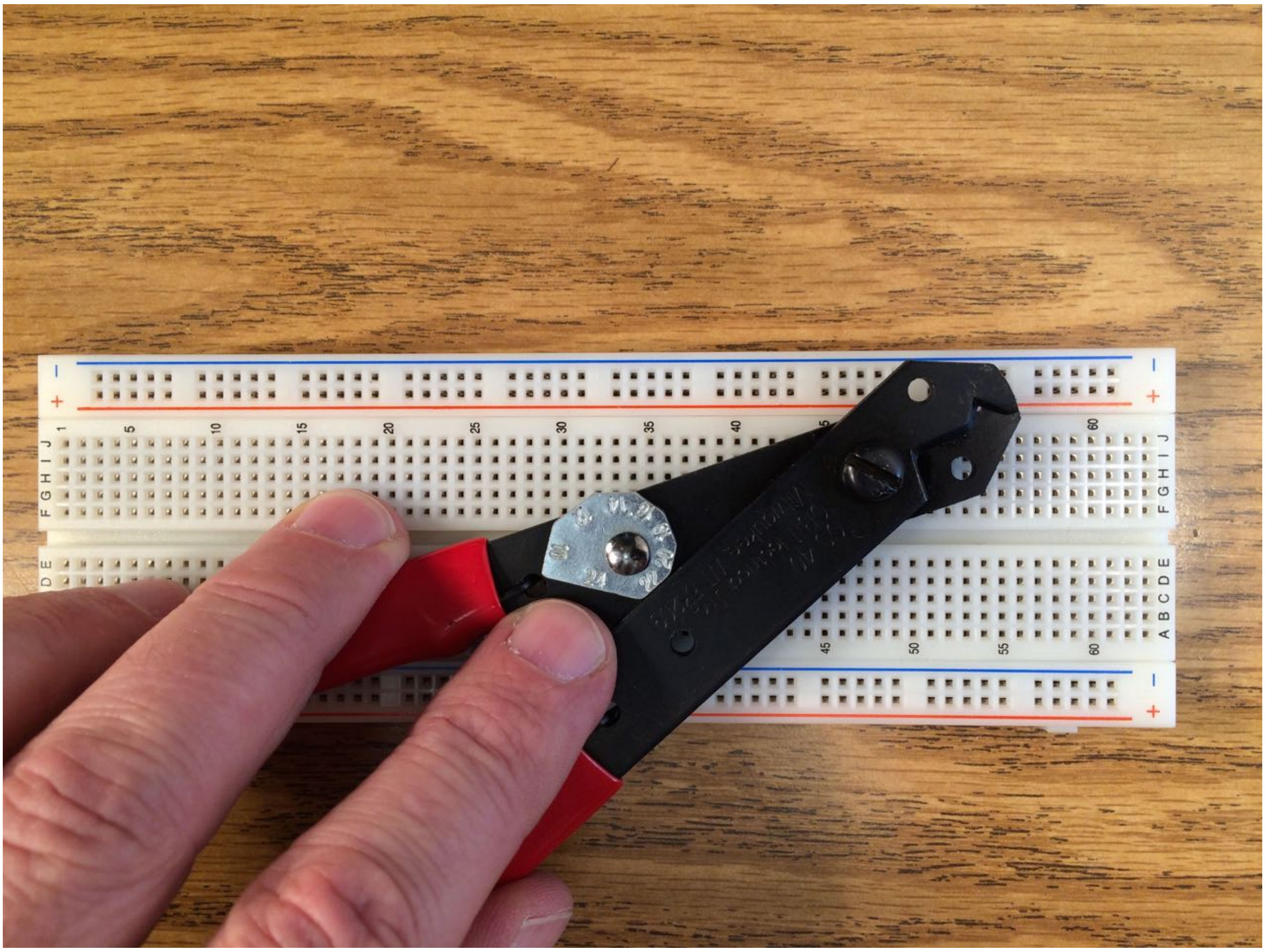
You will need to use the wire stripper to prepare wires to connect the power buses on your breadboard.

How to use the wire stripper

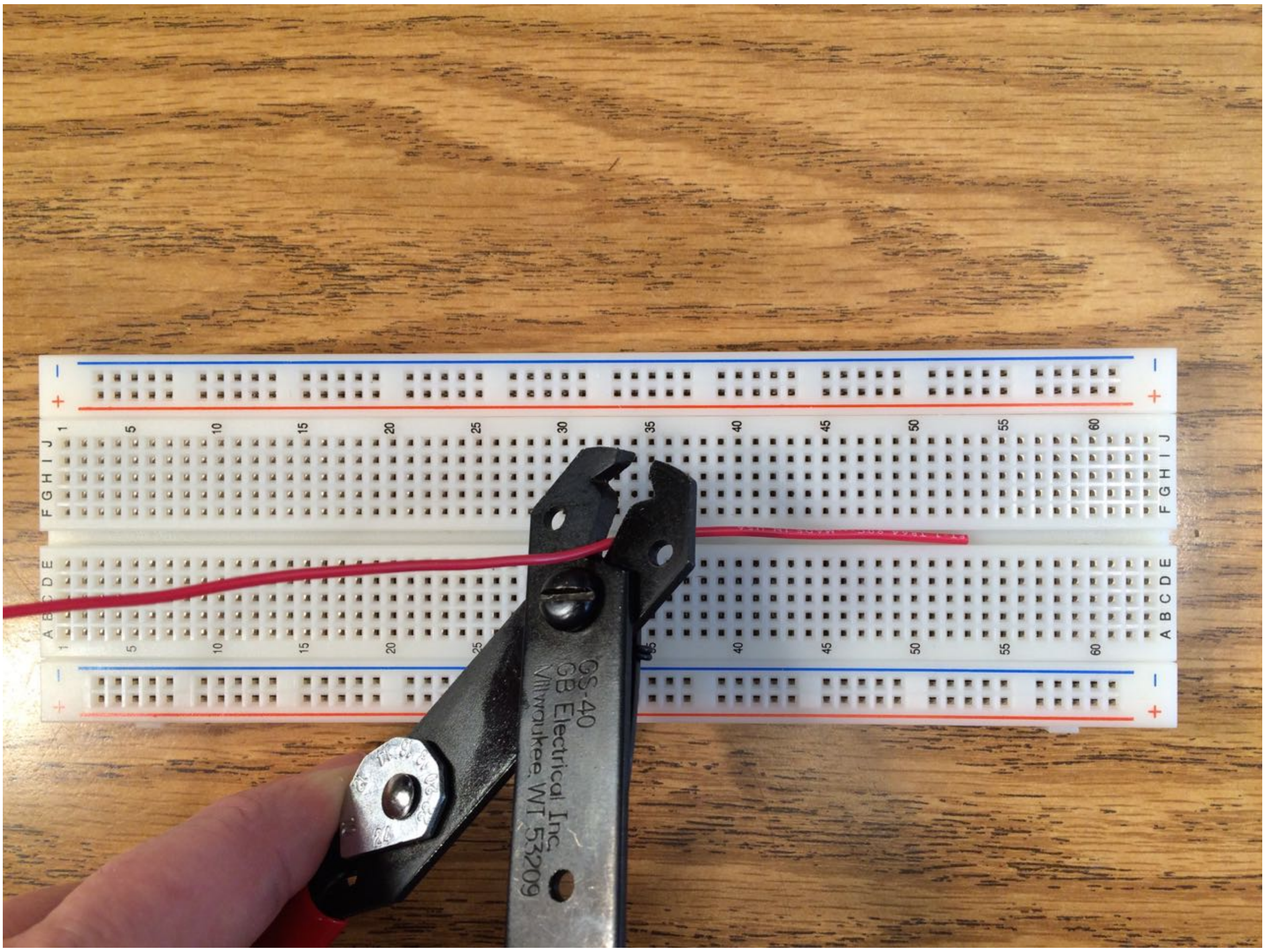
First, turn the wire stripper so that you can see the spring that spreads the handles apart and look to find the wire bale on that spring holding it closed.



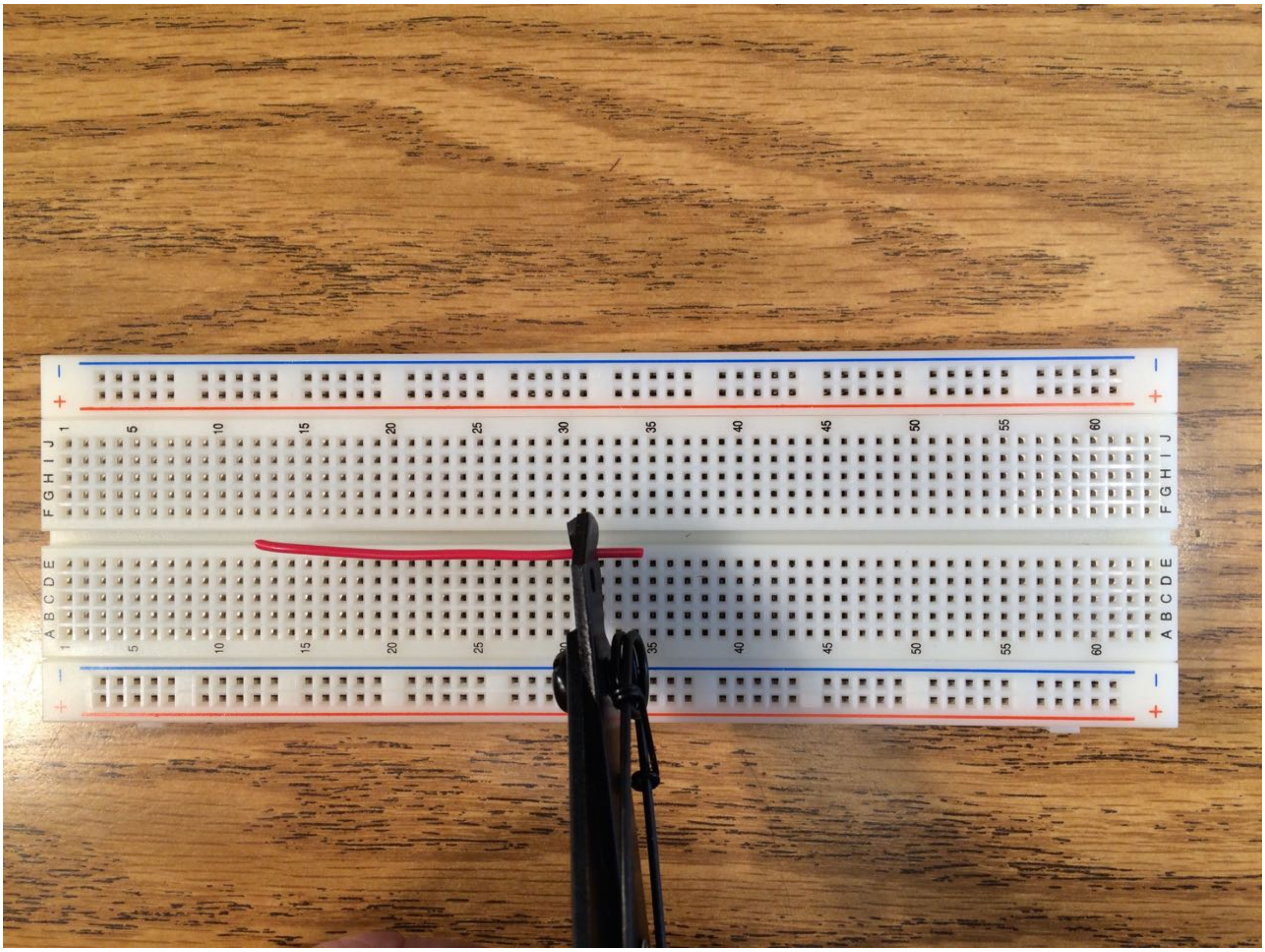
Slide the bale towards the pivot to unlock the tool. Next, adjust the wire gauge cam on the other side of the tool to 22, which is the gauge of the wire in your kit. Make sure that when you sneeze the handles closed the side of the handle that comes into contact with the cam rests squarely on the flat cam face. While squeezing the tool handles, jiggle the cam a bit to make sure it is aligned with the side of the handle. This sets the V-shaped jaws to the correct closing distance to cut deeply into the plastic insulation without cutting the wire.



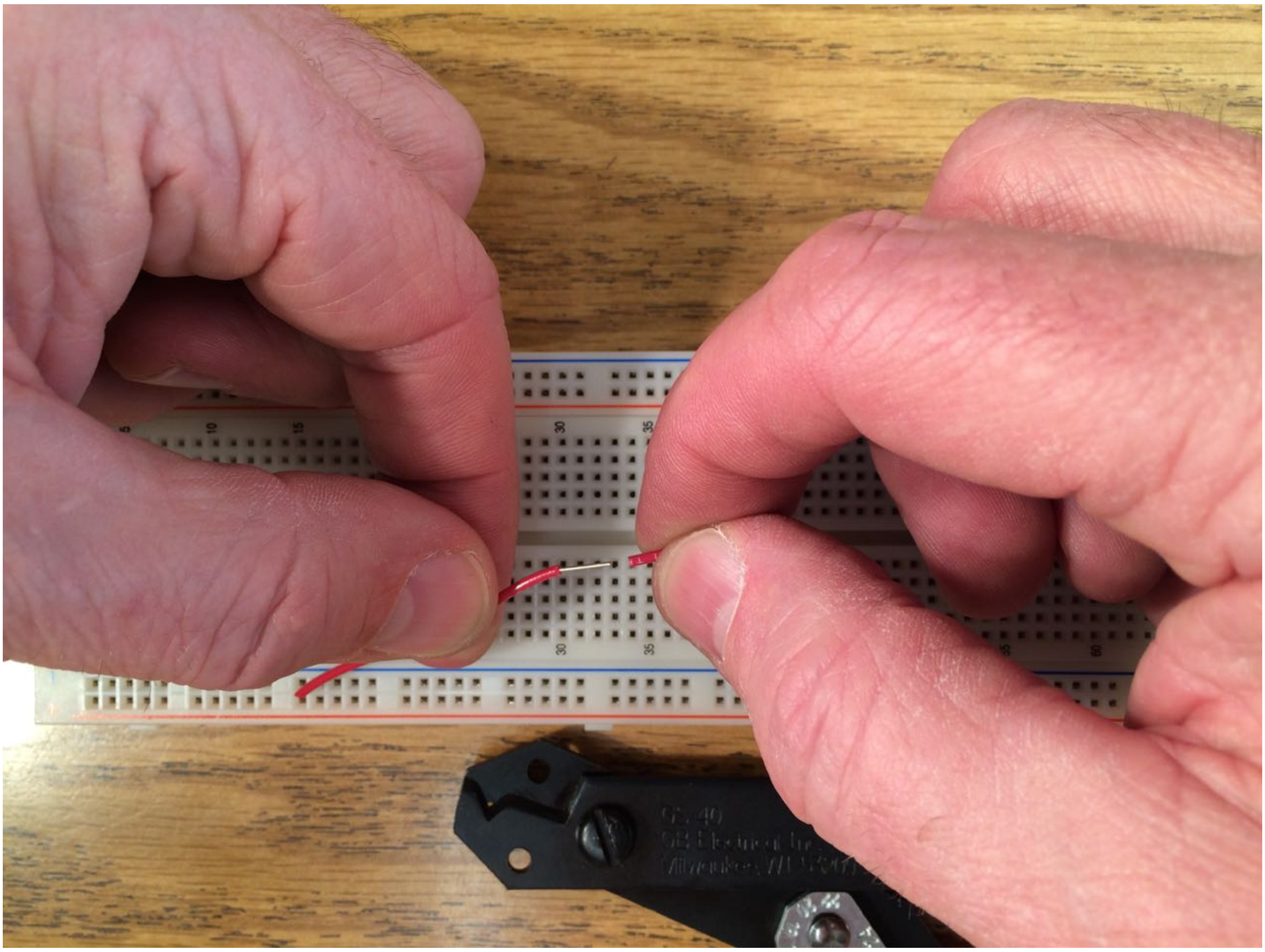
To cut the 22-guage hookup wire in your kit to a desired length, use the cutter next to the pivot point of the tool as shown.



A good rule of thumb when stripping wire for the breadboard is to expose about 1/4 inch or 6 mm of metal conductor. Place the wire in the V-shaped jaws and squeeze all the way to the stop. Flip the wire, cut the insulation at the other end.

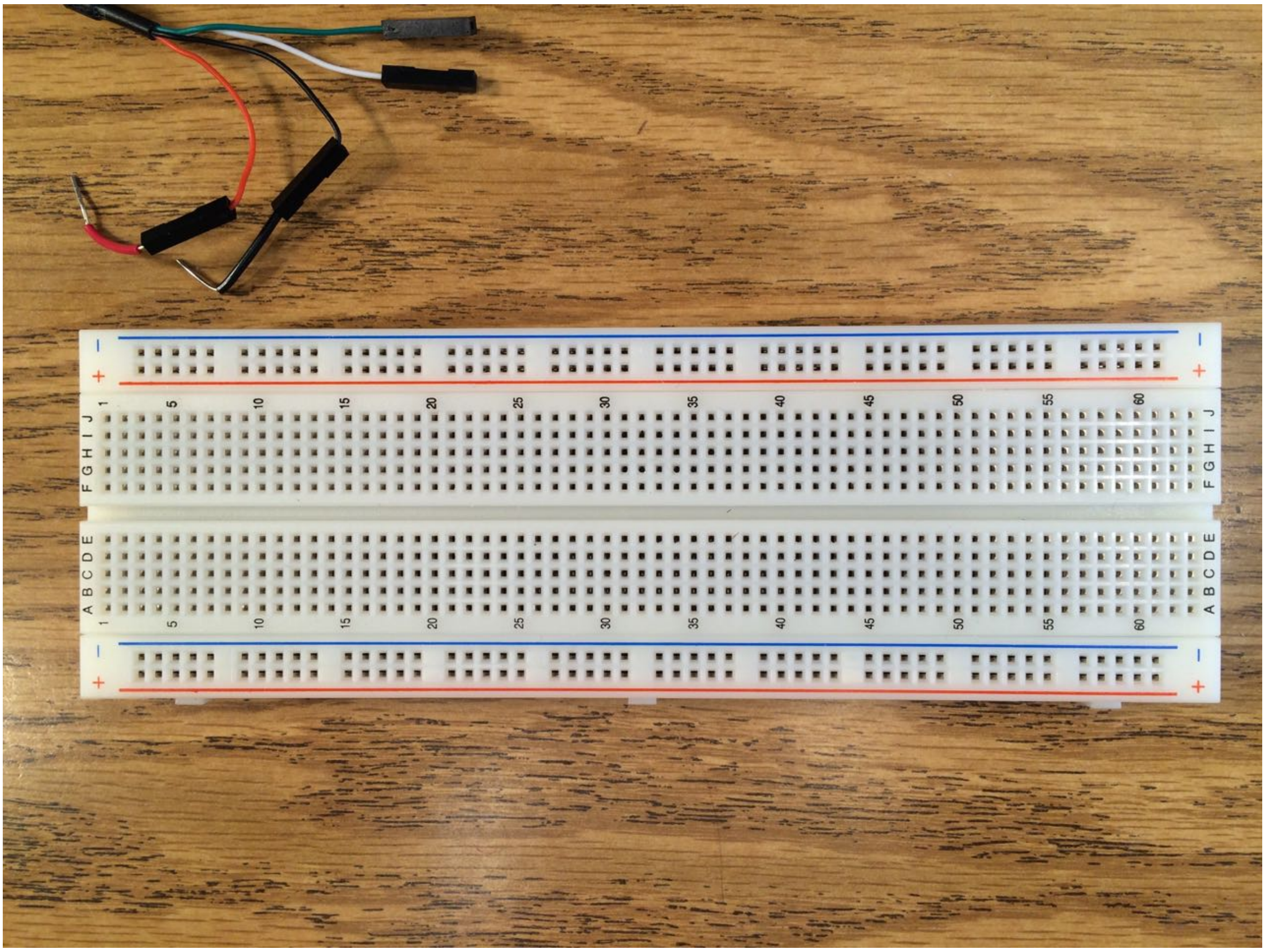


With a little twist between your fingers, you should now be able to slide the insulation off the wire. If the insulation was not sufficiently cut, try setting your tool to 24 on the cam (smaller diameter wire). If that does a better job, then use that setting on your kit's 22 gauge wire anyway.

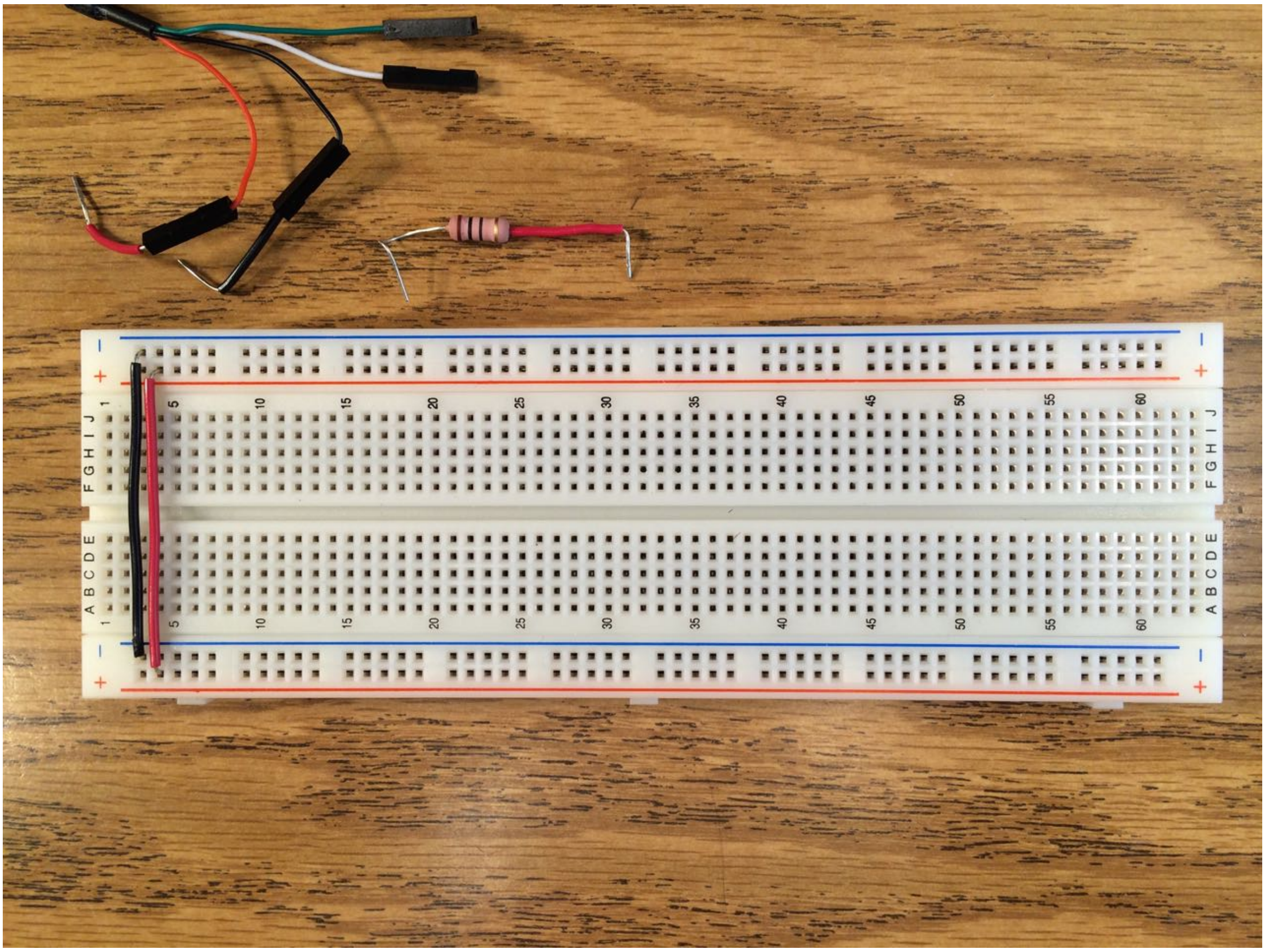


Preparing the USB Console cable to connect to the breadboard

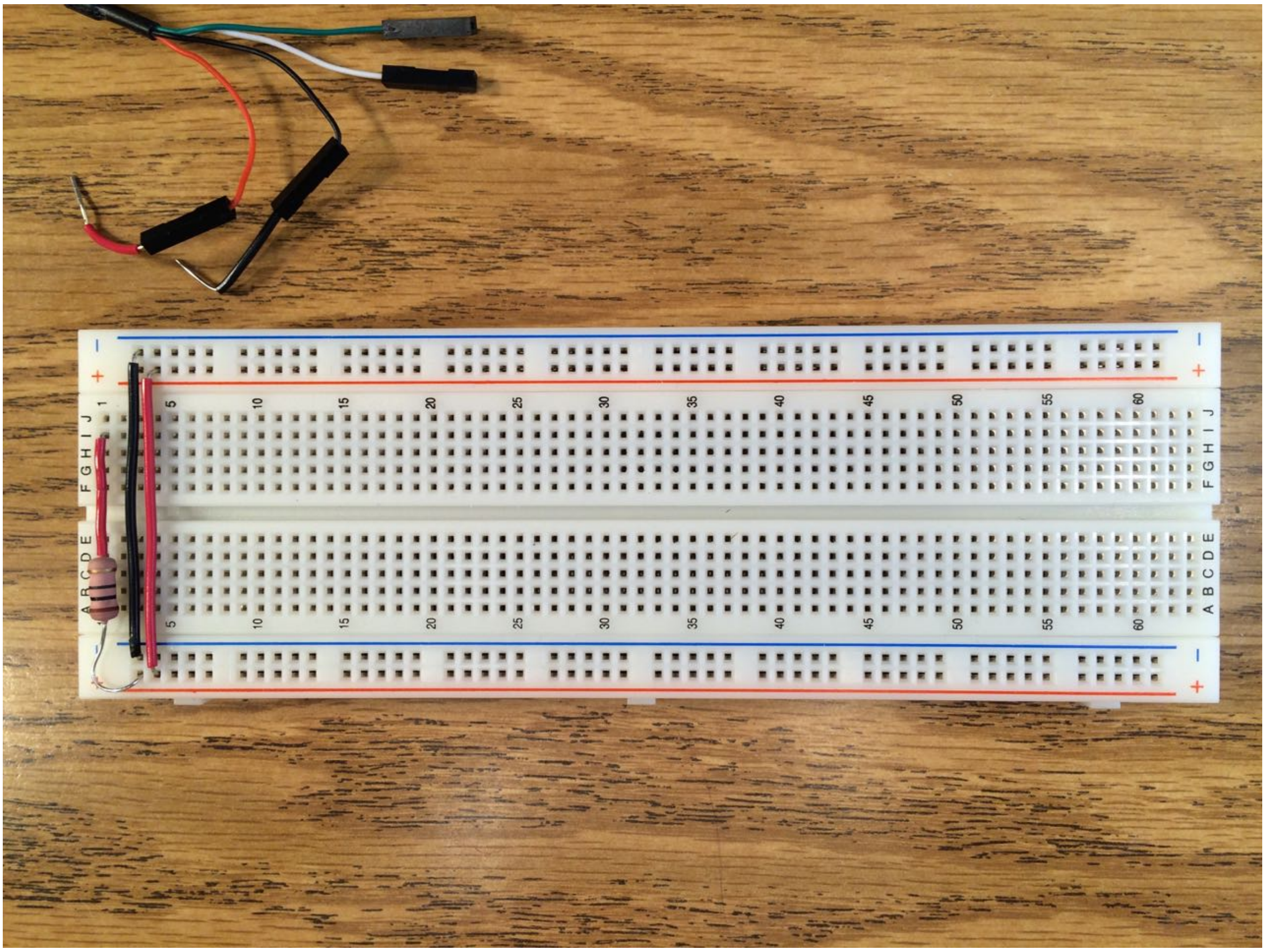
When plugged into a USB port, the USB console cable delivers +5 volts to the red output wire measured from it to the black ground wire. (The white and green wires together carry data.) Short lengths of hookup wire, called jumpers, are needed to connect the USB Console cable with the breadboard. Cut 1 inch lengths of red and black wire, strip their ends, and insert the red wire into the red connector and the black wire into the black wire connector. The result should look like this.



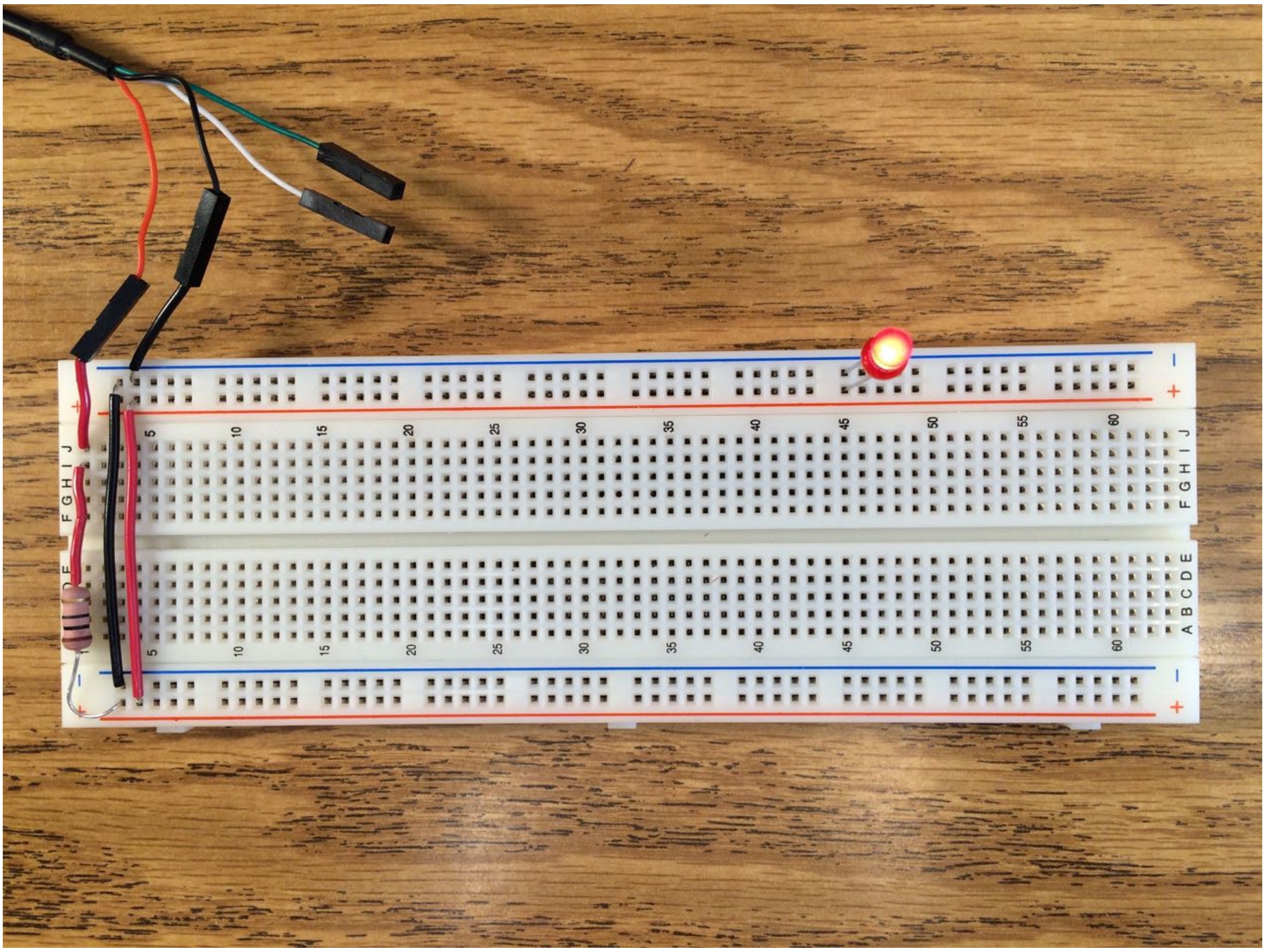
Next, prepare and install wires to link the + power bus on one side of your breadboard to the other + power bus. Repeat for the – power bus. Find your 10 ohm 2 Watt resistor (color band code: brown black black then a space and gold to indicate a tolerance of +-5% from the stated resistance). We suggest that you take some stripped insulation pieces and slide them over one of the leads on the 10 ohm resistor. This will protect your circuits from contacting the USB console cable ahead of the current-limiting action of the resistor. Your result should be equivalent to this.



One way to place the 10 ohm resistor on the breadboard is shown here. *IMPORTANT: Do not directly connect the USB Console cable to the 10 ohm resistor. The diameter of the resistor leads is greater than the cable terminals can handle without damage.*



Prepare to test power to your breadboard by inserting the USB Console cable jumpers and connecting a red LED into the upper power buses as shown in the picture below. Be SURE to insert the longer LED lead into the positive (red) bus. Finally, connect the USB Console cable to an active USB port on you computer or USB hub. The LED should glow.



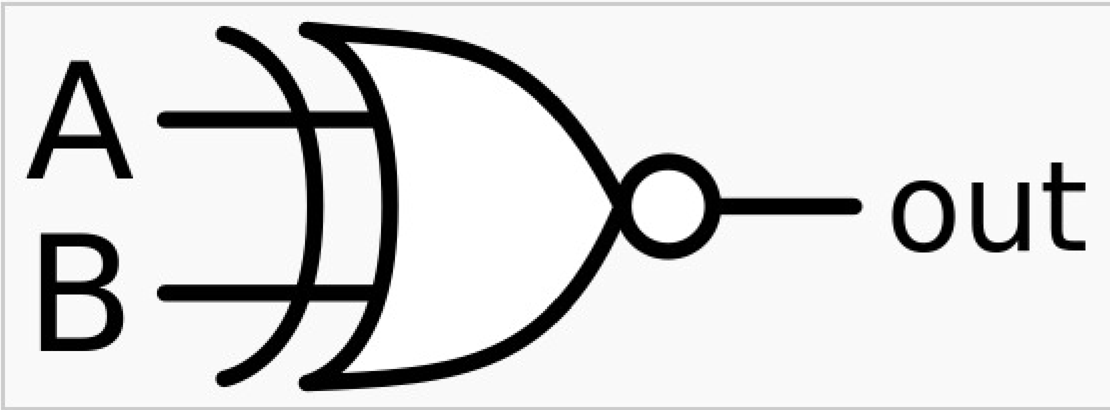
When you are done testing, in fact whenever you are not actively using a breadboard circuit or are revising the circuit, it is a good idea to disconnect the USB Console cable from the USB port. This will prevent any accidental short circuit from possibly damaging the USB port on the computer.

Step 2- Building "A XNOR B" Using Components from Your Lab Kit

There are many distinct ways to write equivalent algebraic expressions. For example, f(y) = y+3 = 7-5(y/(1+4))+10. Which of the two expressions for f(y) is preferred depends on the particular situation you find yourself in. If you have no threes, you might prefer the later form.

Similarly, Boolean algebra admits to many distinct expressions that are equivalent. In particular, it can be shown that any Boolean function of any number of variables is equivalent to an expression comprised entirely of the 2-input NAND function. While there are DIP-packaged devices that directly implement XNOR (for example, 74LS266 is a 14-pin DIP with four 2-input XNOR gates), in this lab we will design and build a 2-input XNOR gate using NAND gates with the result that the circuit is slightly cheaper than buying a 74LS266 chip.

Start with the schematic symbol and truth table for A XNOR B (in the schematic “out” stands for the same logic as shown in the A XNOR B column of the truth table).



A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

Mnemonically, A XNOR B is true only when A and B are both true or both false.

Express XNOR as a composition of 2-input NAND functions.

To implement XNOR using NAND gates, we need to an expression for XNOR in terms of NANDs. We can start with an expression for A XNOR B in terms of AND and OR obtained by looking at the truth table: since A XNOR B is true when both A and B are 0 or both A and B are 1, then A XNOR B = A'B' + AB. To convert to an expression in terms of NAND, play around with algebraic manipulation. In this case, starting by applying double negation, then use De Morgan's Law and look for factors of the form (AB)', which is A NAND B. These factors may be nested.

So, in terms of just 2-input NAND, the expression A XNOR B = ——————————————————–

Step 3- Translate the NAND form into a circuit

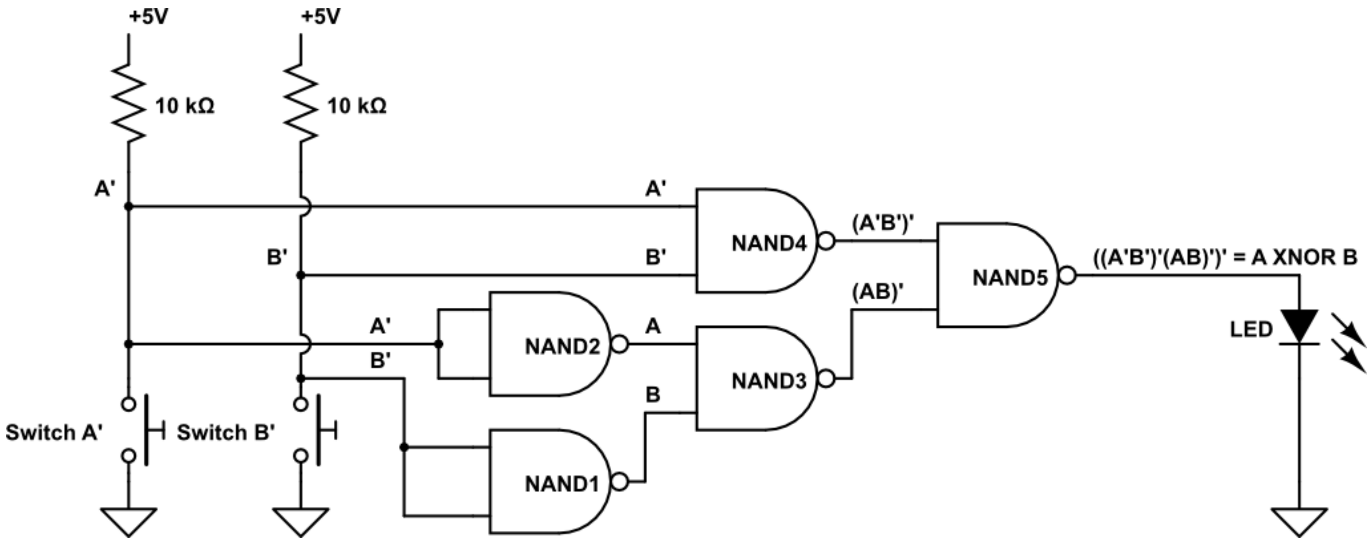
Now that we have A XNOR B expressed in terms of NANDs, we need to design a circuit that corresponds to the expression and meet any other goals. Our other goals are two: the input values A and B should be generated manually (we want to control this circuit personally) and the output should be displayed visually. The push button switches in the lab kit are one way to accomplish the first goal. The red LEDs in the kit can take care of the second goal.

What does the circuit need? First, we must have logic inputs A, A', B, and B'. We can use the voltage divider concept to create input. The design for this pairs a fixed resistor from the kit (10 Kohms) and a “variable resistor” from the kit, the push button switch. The push button switch transitions between nearly infinite resistance when open to nearly zero resistance when closed. This specific voltage divider circuit will consist of our power supply voltage connected to 10 Kohms in series with the switch that is then connected to zero volts (called ground). The output will be the node between the resistor and switch. The voltage at the output will alternate between a essentially zero when the switch is closed and essentially the supply voltage when the switch is open. This circuit is similar to the two-MOSFET circuit for an inverter in this way: the 10 Kohm resistor serves as “poor quality” MOSFET that is stuck in a somewhat ON state while the push button switch plays the role of an “ideal” MOSFET that alternates between really, really OFF and really, really ON. The combination of qualities delivers an acceptable quality signal to use an an input and does it manually, rather than electronically. Because this circuit is similar to the inverter, it is important to realize that pushing the button generates a low-voltage input, corresponding to logic zero or false.

Next, the circuit needs NAND gates. Three are needed for the alternative expression for XNOR, but we also need the full complement of input variables and their complements. A pair of resistor/switch voltage dividers can provide inputs A' and B' so the question is how to create A and B. Examination of the NAND truth table reveals that when a NAND is presented with identical inputs, its output is the opposite logic value. Thus, we will use two additional NAND gates in the circuit we design to perform the NOT logic function.

Finally, the circuit needs visual output. The LEDs in the lab kit are compatible with the outputs of the gates in the DIP packages. So, if we conned the anode (longer lead of a red or green LED in the kit) to the output of a gate and connect the shorter lead to breadboard ground, the LED will light when the gate output is high (near power supply voltage).

Summing this all up, here is the circuit. Examine it and be able to point out the voltage divider input, the NAND gates serving as inverters, and the three NAND gates combining to perform the XNOR function.



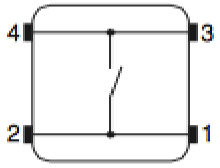
When you have built the circuit and it is working, show it to your TA. If you need help debugging your circuit, ask your TA.

Step 4- Construct a circuit corresponding to the design schematic in Step 3

The following is a step-by-step guide to implementing the circuit we designed above. *If this is your first electronics lab experience, take your time! If you rush, errors such as improperly placed components or mis-routed wires are often more time consuming to debug than the time needed for deliberate assembly.*

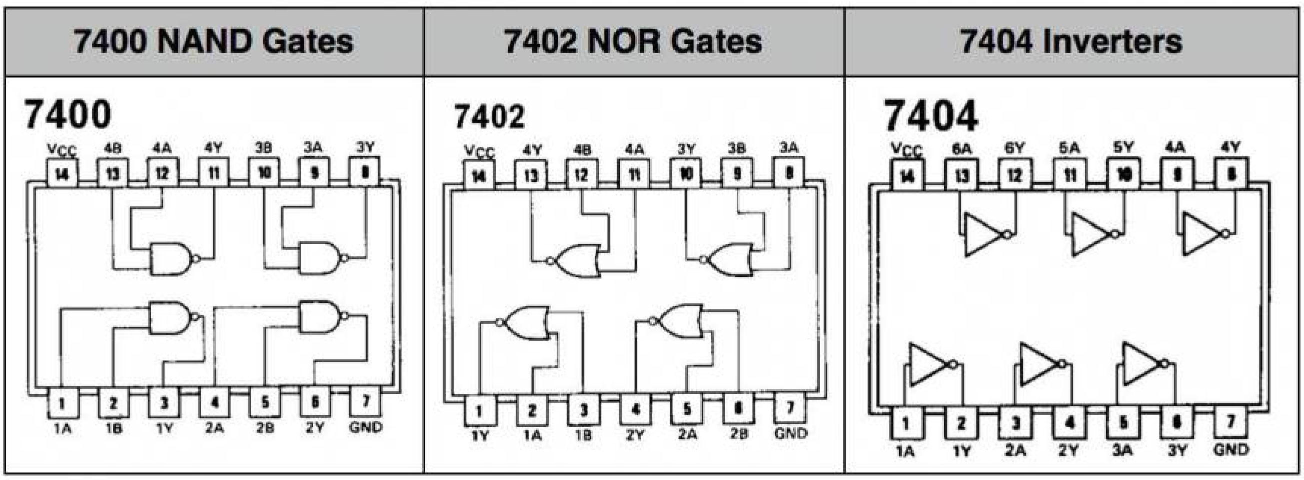
The pushbutton switches in the kit are built in a way that is both very convenient and not at all intuitive. First, the switches are intended to span the breadboard gutter just like the DIPs, and you will notice that there are two rotational positions that fit fine and two that do not span the gutter. Unlike the DIPs the two rotational positions for the switch are equivalent. Second, when the switch is positioned across the gutter, the pins on left side are electrically connected at all times and, thus, link the two breadboard columns on opposite sides of the gutter together. The same is true for the two pins on the right side. Finally, when the button is pushed the internal mechanism of the switch connects the left and right sides electrically. When the button is released a spring pops the switch back to an electrically open condition. Switches look like this.

A pushbutton switch schematic that looks a bit like the actual switch is this. Rotate the schematic 30 degrees counterclockwise if you would like to align it with the image above.



Note that this switch schematic looks different from the one in the circuit schematic in Step 3, but mathematically, logically, and topologically the two are equivalent. Each is a view of our switch from a different level of abstraction. And the image of the switch is yet another level of abstraction. Practice moving among the various levels of abstraction that we will use in CS250.

The DIP packages containing NAND gates (a.k.a. chips) are the ones labeled SN74HCT00N. Here are pin diagrams for three chips in the 7400 series, including NAND. These diagrams show you which pins on the chips correspond to which portion of a particular gate (each chip has multiple gates on it).

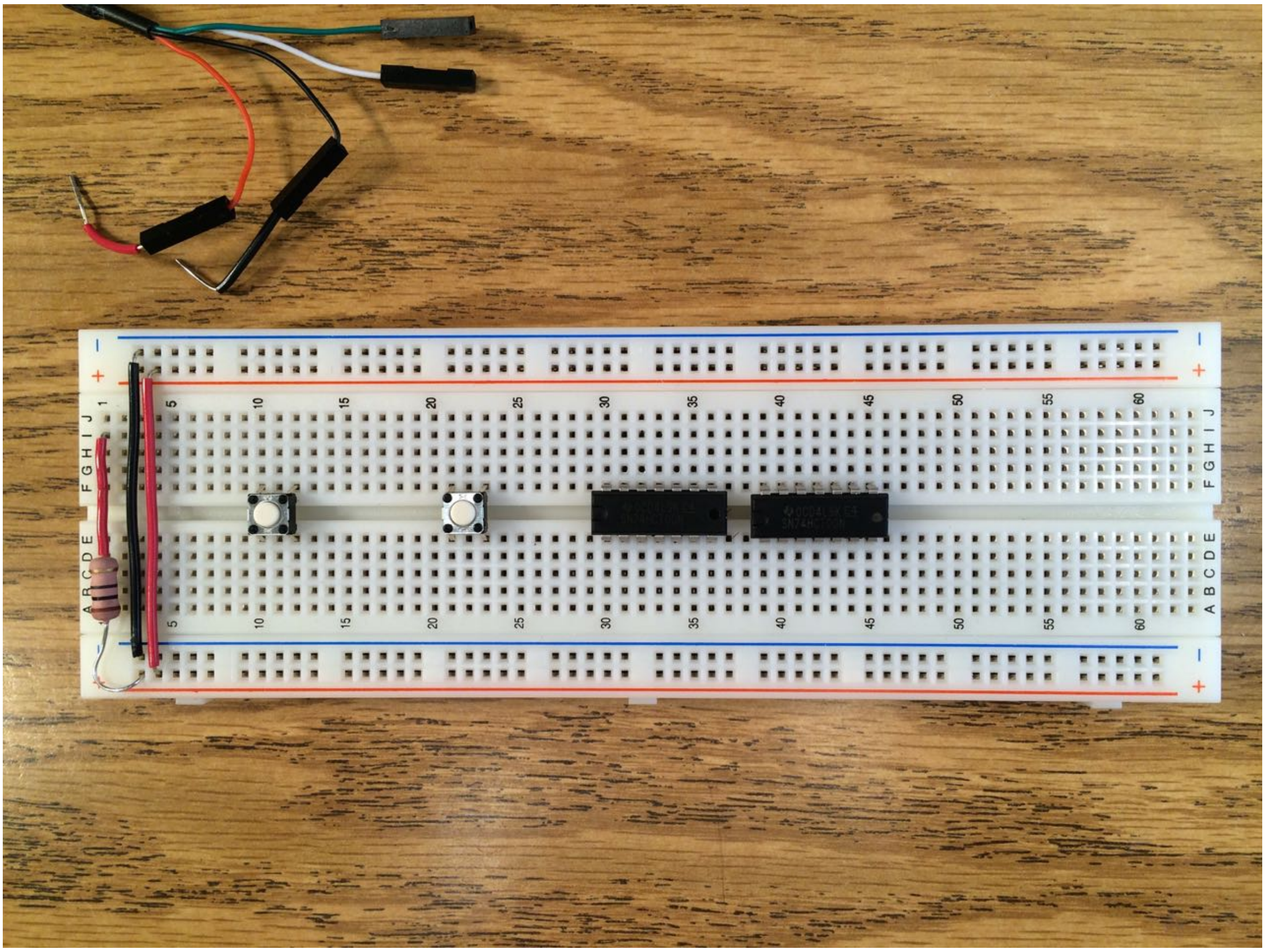


Make sure you can read the tiny label print on the top of the chip chip when referencing the diagram: if the part number on the chip is upside down, then you are holding the chip upside down.

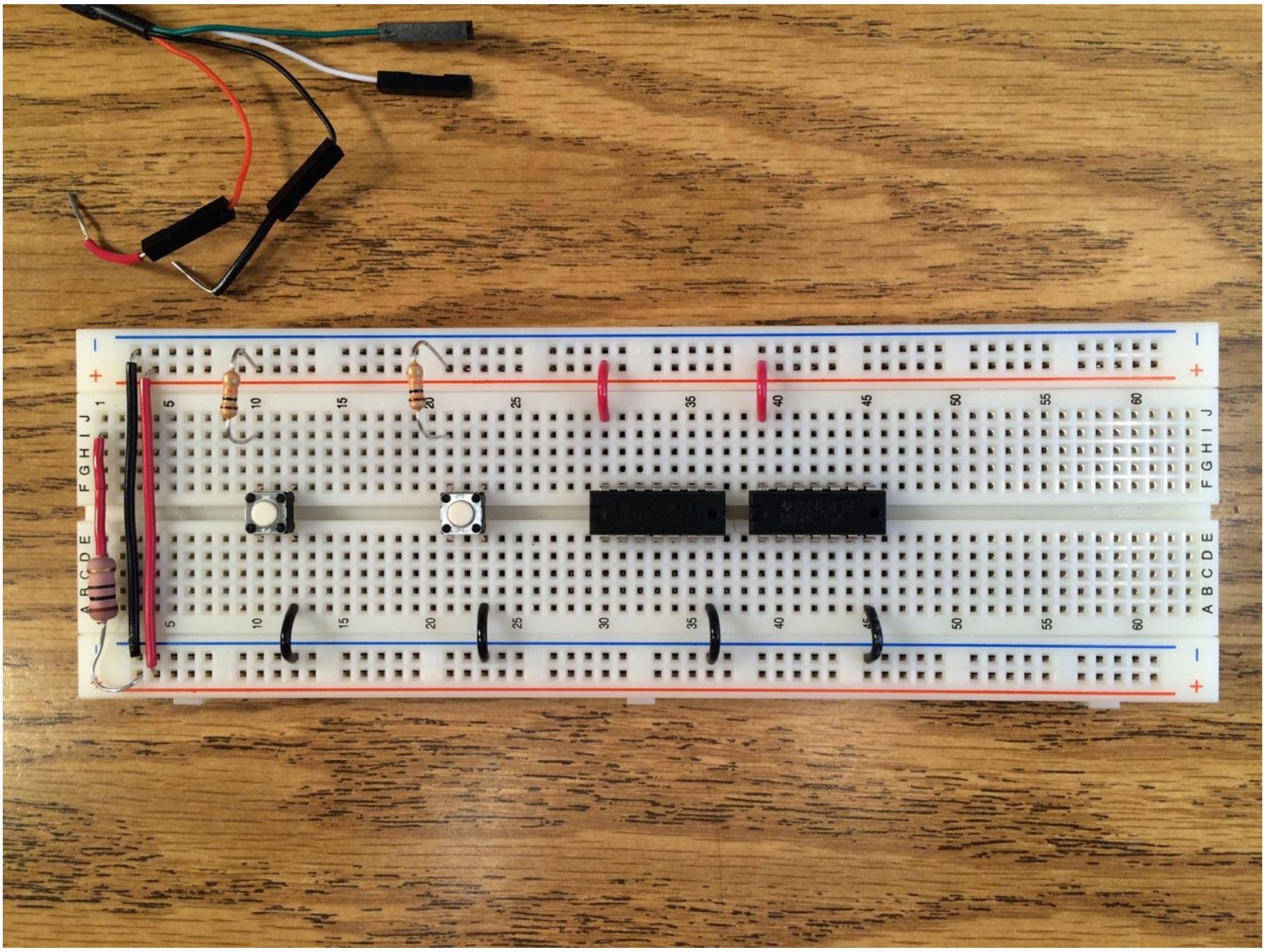
At any point if you want to test your construction, you can use an LED as a visual probe. just connect the LED anode (longer lead) to the circuit node you wish to interrogate and connect the other LED lead to breadboard ground. The LED will light when the circuit node is at logic 1 (high voltage) and not light if the circuit node is at low voltage or is not connected to anything (called an open circuit).

Step 1. A good starting strategy is to place the main components of your circuit on the breadboard first, particularly switches and DIP devices. Place switches where you can easily reach them. If there will be several, space the switches to give room for a finger to reach each switch simultaneously. This will allow you easily press the switches in any combination that may be needed. DIP packages are large, which is a good reason to site them next.

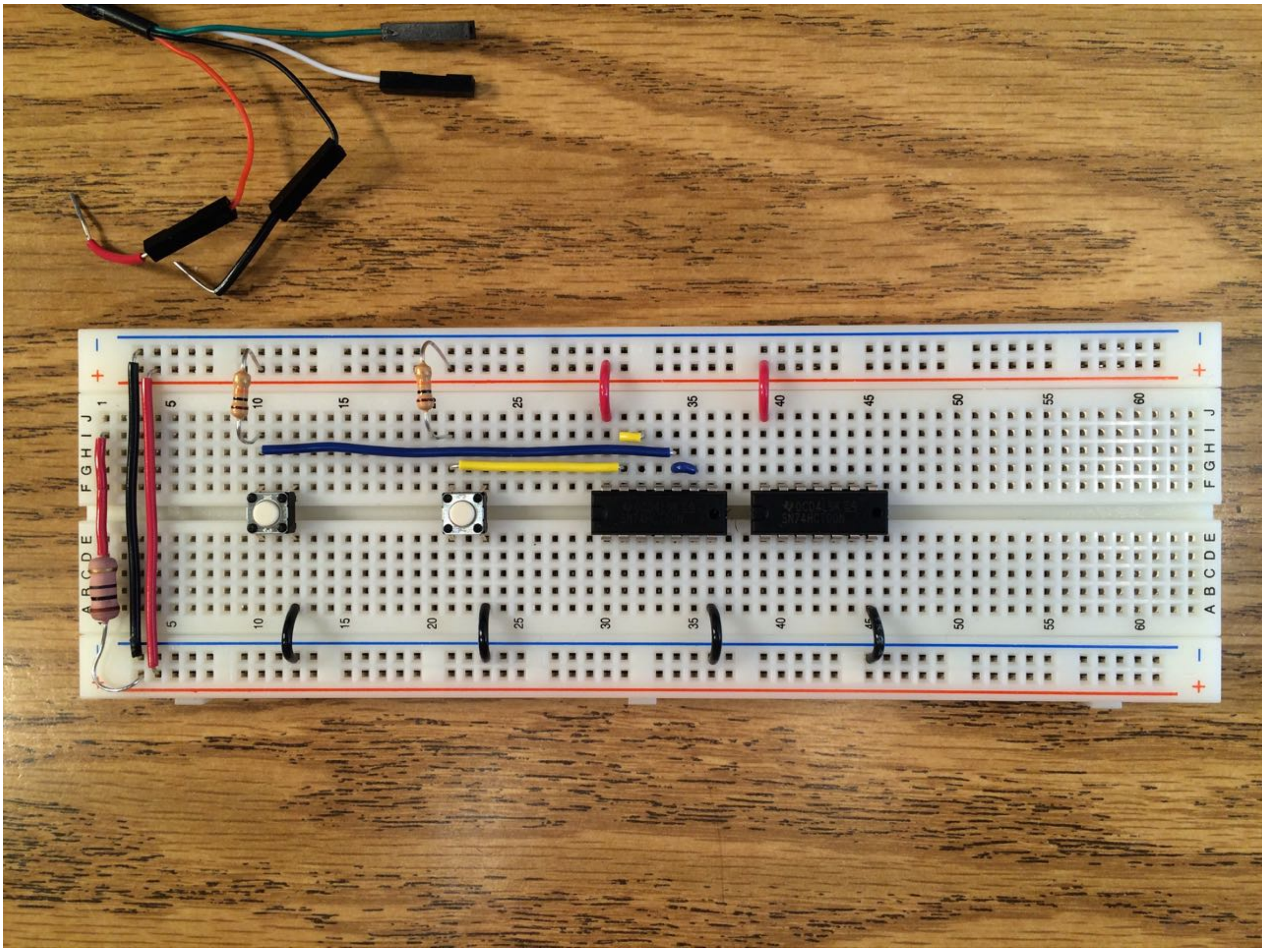
Here is how the switches and DIP packages might be placed. The photo does not show the tiny print on the DIP packages, but they have both been placed with that print right side up in this view. Both the switches and the DIP packages need to be pushed into the sockets until they are flush with the breadboard surface. They tend to “snap” into place when fully seated in the sockets, especially the DIP packages.



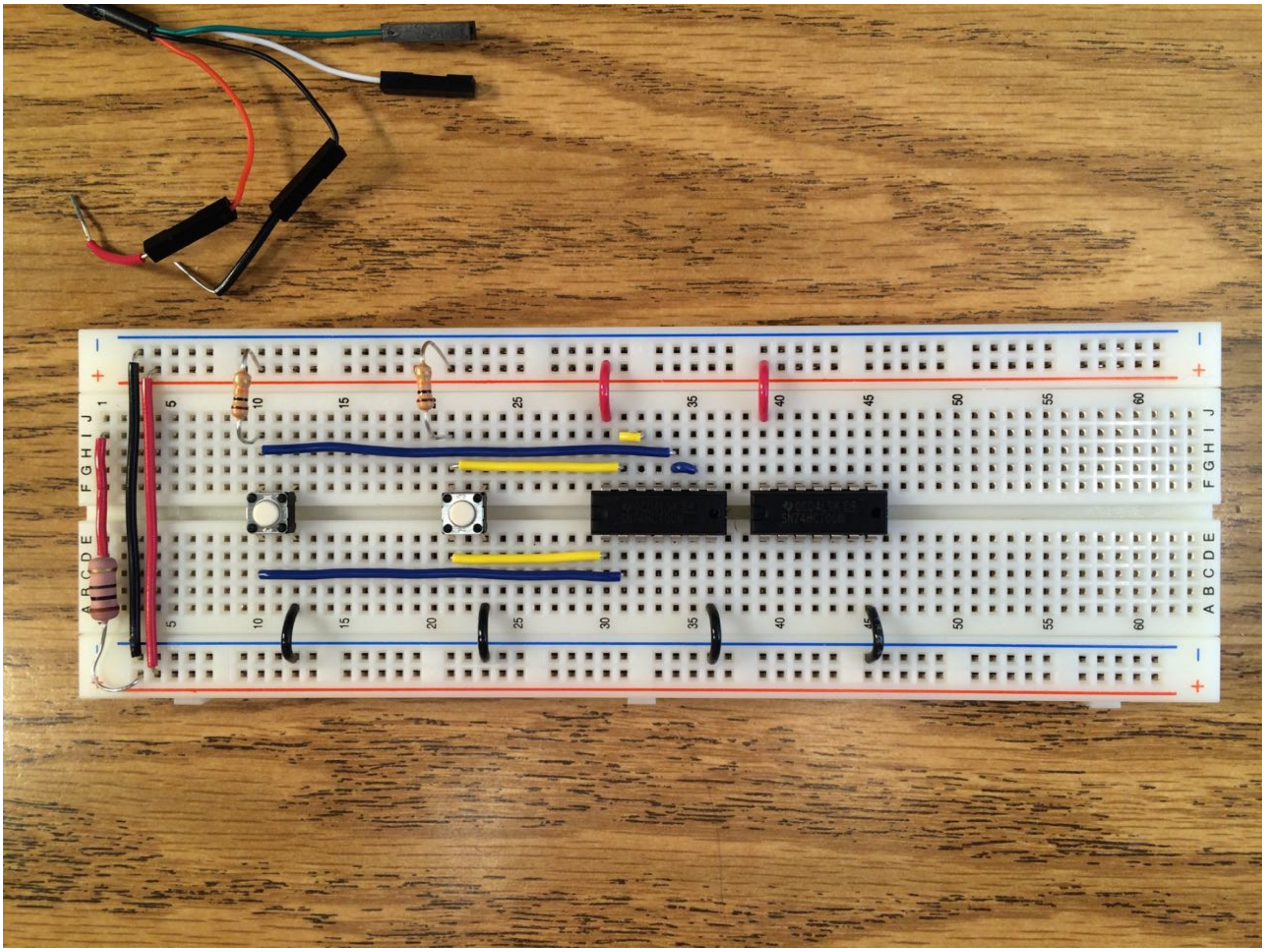
Step 2. Add connections to power and ground.



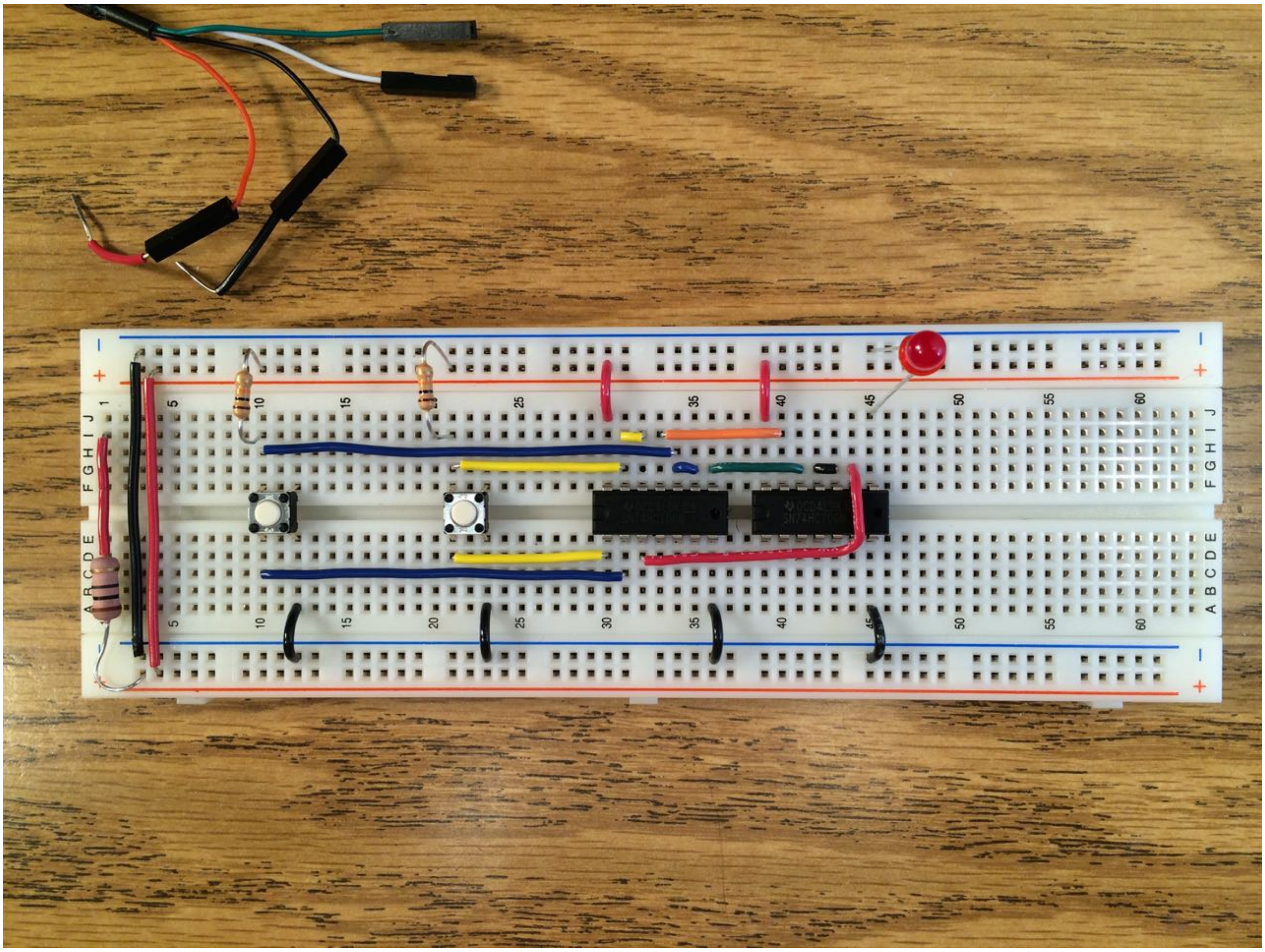
Step 3. Bring A' and B' to the inverting NANDs. In the photo, A' is the blue wire, and B' is the yellow wire.



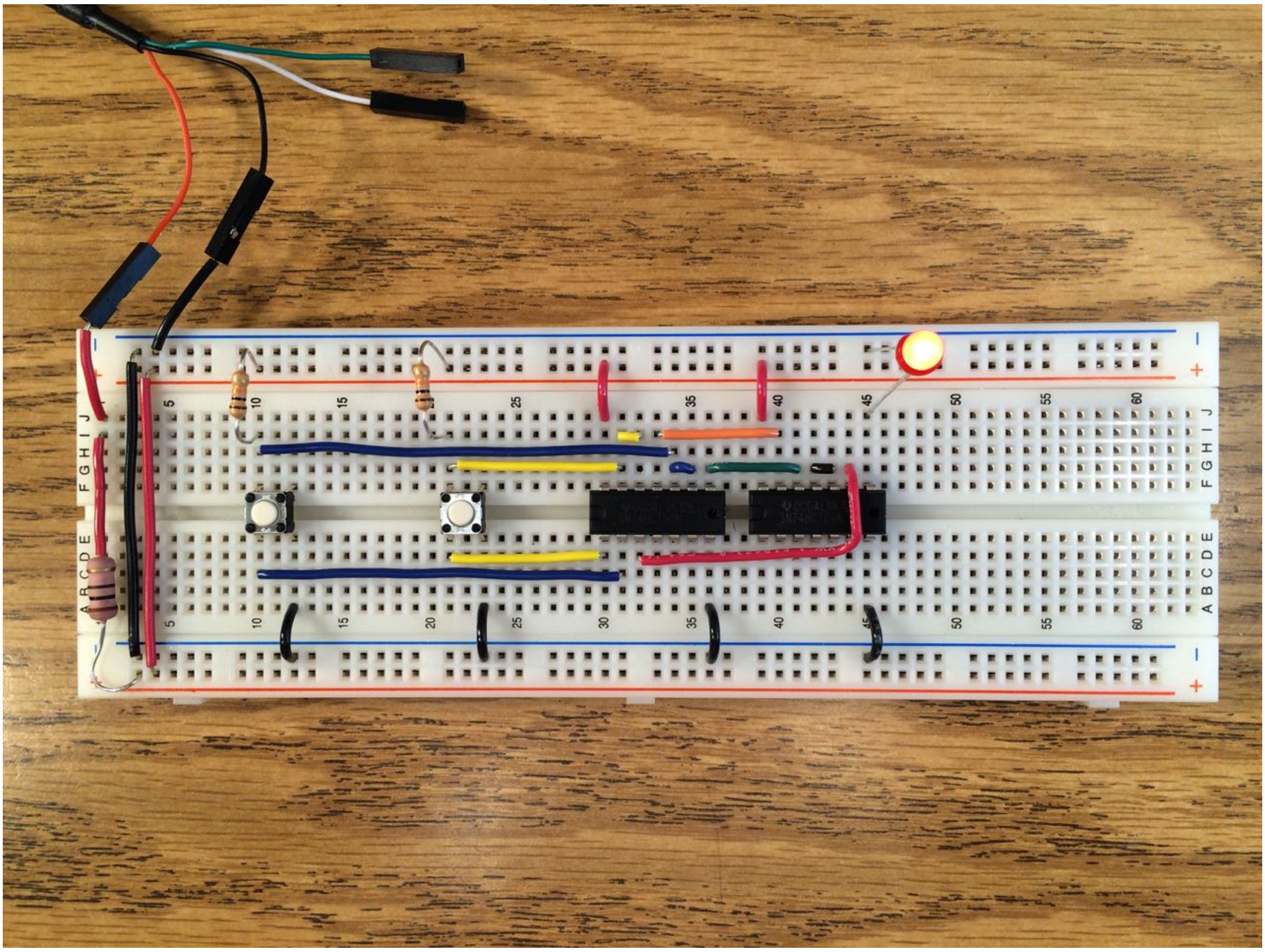
Step 4. Complete the wiring to the other first-level NAND, NAND3. This gate also takes A' (blue) and B' (yellow) as input. Note the convenience of having the voltage divider output node available on both sides of the gutter because of the design of the switch.



Step 5. Wire NAND4, NAND5, and the LED per the schematic. Here, orange and green wire carry inputs to NAND4; red and black wire serve NAND5. The LED anode is connected to the output of NAND5 and its other lead (the cathode) is connected to breadboard ground (the blue bus along the edge).



Step 6. Review your work one last time, then connect the console cable jumpers as shown, and finally connect the USB end of the console cable to an active USB port. This will deliver 5 volt DC power to the breadboard. If the circuit is correct the LED will light with neither button pushed as shown here.



Try all combinations of A and B input values to verify that your circuit does indeed implement the 2-input XNOR function.

Show your TA your working circuit before you disassemble it.

When you disassemble a circuit, keep the various pieces of hookup wire to reuse in your next circuit.

Take Home Assignment

See Blackboard for the take home assignment for Lab 01.