

# Institute for STEM Education



Stony Brook University  
College of Engineering  
and Applied Sciences

## *Night Light*

Student Lab Guide



Engineering Teaching Laboratory

Name \_\_\_\_\_ Date \_\_\_\_\_

Lab Partner(s) \_\_\_\_\_

## **NEW TERMS**

**Voltage:** is an electric force that causes free electrons to move from one atom to another and creates a current in a closed circuit. Voltage (V) is measured in **Volts**.

**Current:** is the physical movement of the charges in a system, it is defined as the amount of charge that moves past some point per second. In other words, it is the rate of the charge movement. Current (I) is measured in **Amperes or Amps**.

$$\text{Current (I)} = \text{Amount of charge/Time.}$$

**Resistors:** a device that is designed to resist the passage of an electric current. Resistance (R) is measured in **Ohms or  $\Omega$** .

**Electric Circuit:** Electric circuits are paths for transmitting electric current, or moving electricity. Such circuits allow electricity to be used to provide power to lights, appliances, and many other devices.

**Potentiometers:** a device that can act as a variable resistor. It is a resistor that can change its value.

**Light Emitting Diode (LED):** an electronic device that emits light when an electrical current is passed through it.

**Breadboard:** A thin plastic board used to hold electronic components (resistors, LED chips, etc.) that are wired together.

**Integrated Circuit (IC):** An IC is a collection of electronic components (resistors, transistors, capacitors, etc.) all stuffed into a tiny chip, and connected to achieve a common goal. In this design, the IC will function as a timer.

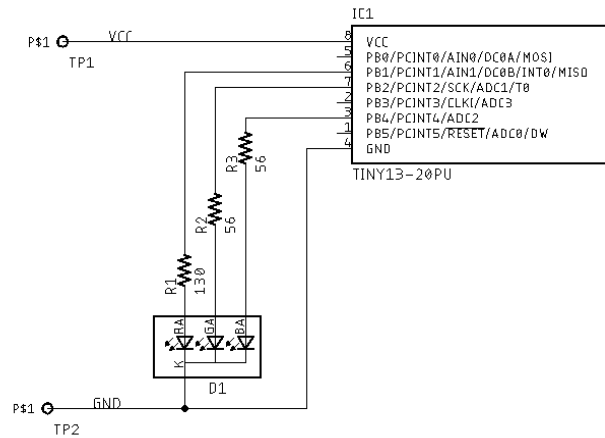
**Microcontroller:** an electrical component that controls output either “on” or “off.”

### **Terms to be familiar with**

Conductors  
Insulators  
Electricity  
Electrons

## INTRODUCTION

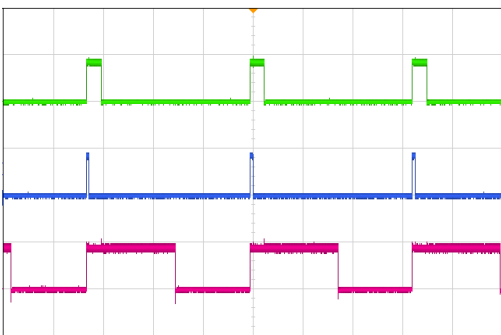
Current, voltage, and resistance are related through Ohm's Law ( $V = I * R$ ). Simply put, a current is defined as a voltage across a known resistance and a voltage is defined as the current through a known resistance. Putting together voltage, current, and resistance develops a circuit. Circuits are very powerful tools used all throughout our daily lives. They are used to turn on our lights in the house and power our electronic devices. Engineers use simplified drawings called schematic diagrams to show how a circuit is connected. The circuit diagrams have a code to show different components, like resistors (zigzag) and LEDs (triangles). A schematic for the circuit being built in this lab is shown here.



Circuits are composed of various components that contribute to its functionality. They all need a power source to function and have different combinations of resistors and capacitors to determine what the output signal will be. The output can be regulated by how many resistors or how much resistance they have to make different signals.

In this lab, you will build a circuit for a nightlight that can be powered by any USB charger. Light will come from a red, green, and blue light emitting diode (LED). You will control the intensity and color of the light using a potentiometer. You will also learn and experiment with combining these colors together.

Later in the experiment, a microcontroller chip is used to control the light color and permit color shifting. With this, the appearance of variable brightness is portrayed by switching the output on and off very fast. If the output is on most of the time, the LED appears bright. If the output is off most of the time, the LED appears dim. The output is flashing, but it is flashing about 150 times each second - too fast for your eye to notice.



The graph shows the current through the green LED (top trace), the blue LED (middle trace), and the red LED (bottom trace). The horizontal axis is time – the whole graph represents 0.02 seconds. The blue LED is hardly on at all, the green is on a bit, and the red is on a lot. This would look to your eye as a slightly yellowish red.

## **PROCEDURE:**

### **A) Building the Night Light**

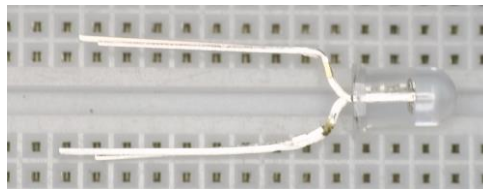
1. Find the RGB LED. Note the legs are different lengths.



2. Hold the LED as shown above. Bend legs #2 and #4 towards you.
3. Bend legs #1 and #3 away from you. When you are done, the LED should look like this when viewed from the side:



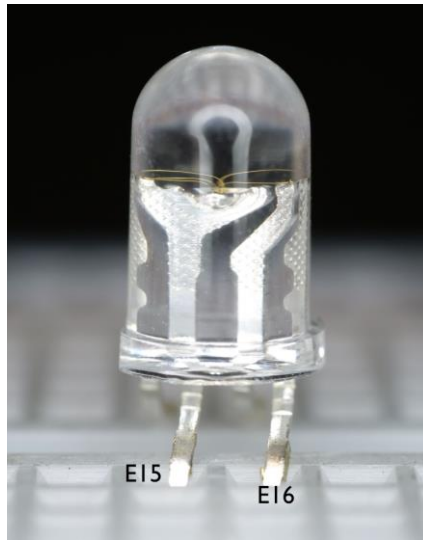
4. Bend the LED legs again so that they are spaced to go into columns E and F of the breadboard.



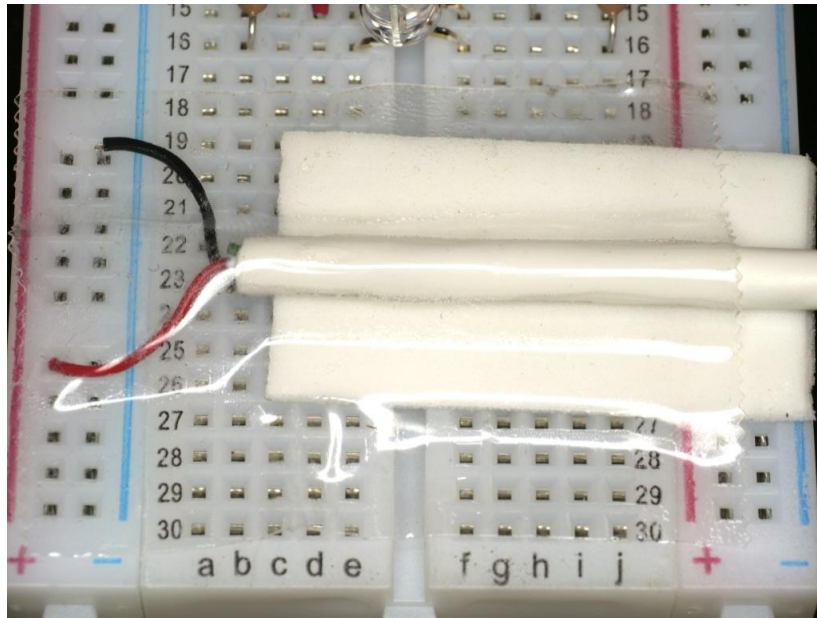
5. Cut the legs to the length shown.



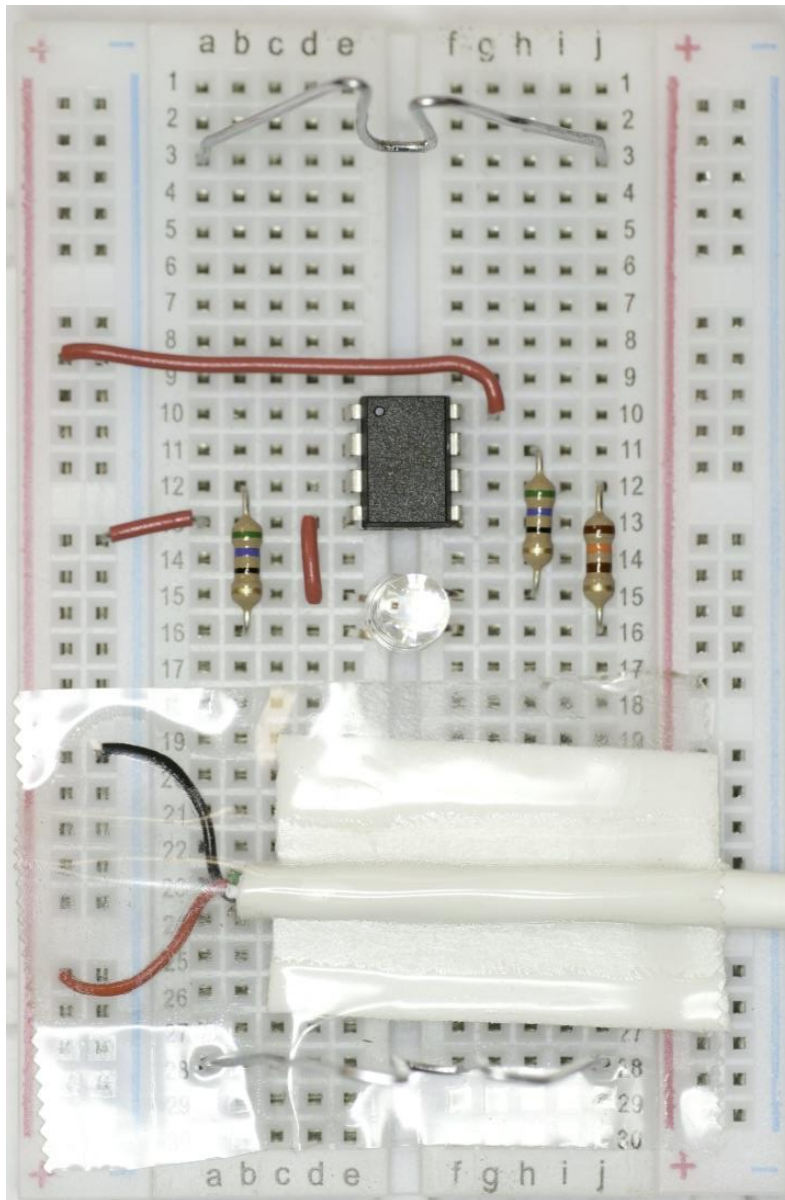
6. Look inside the LED for the 'T' shape. Install the LED into the breadboard at locations E15, E16, F15, and F16 as shown below. The 'T' lead goes into E15.



7. Install a 130  $\Omega$  (**brown, orange, brown, gold**) resistor between J12 and J16.
8. Install a 56  $\Omega$  (**green, blue, black, gold**) resistor between H11 and H15.
9. Install a 56  $\Omega$  (**green, blue, black, gold**) resistor between B12 and B16.
10. Install a jumper wire between D13 and D15.
11. Install a jumper wire between A13 and a nearby hole in the negative (–) column nearest the A column.
12. Install a jumper wire between G10 and a hole in the positive (+) column nearest the A column. Make this wire 6mm ( $\frac{1}{4}$  inch) longer than it needs to be – you’ll be installing another component later and might need the room.
13. Locate the USB cable. Cut off the white and green wires without damaging any of the other wires.
14. Insert one of the black wires into the negative (–) column hole nearest to A19.
15. Insert the other black wire into a neighboring negative (–) column hole.
16. Insert the red wire into the positive (+) column hole nearest to A25.
17. The USB cable is not very secure yet. Place a piece of double stick foam tape under the USB cable as shown below. Stick the cable into the tape being certain that the wires are still in their holes.



18. Place a piece of clear tape over the USB cable all the way across the board. The cable will still be a bit fragile, but it'll stay on if you handle it with care.



Completed Bread Board for Night Light.

## B) Project Testing: Setting #1- One Color

In this experiment, you will use variable resistors to control the amount of light from each of the LEDs to create different colors.

1. Obtain a resistor module. Look at the end of the resistor module cable and notice that there is a white mark on one corner of the connector. The pin near that white mark will be going into breadboard hole E10; the other pins on that side will go into E11, E12, and E13. The pins on the other side will end up in holes F10 through F13. Press the connector into your board.
  2. Place a tissue over the LED. This will decrease the brightness and make it easier to see the colors.
  3. Plug the USB cable into a charger. Some LEDs might light up.
  4. Each of the red, green, and blue LEDs has a resistor to control its brightness and a switch to turn it off completely. Play with the controls to see how they work. Notice that the LEDs have slightly different beam patterns which causes color fringes.
  5. Adjust the knobs to get cyan light. What colors did you use to make cyan?
- 
6. Now, find a way to generate yellow light.
  7. Magenta can be generated, too. What colors did you use to make magenta?
- 
8. **CHALLENGE:** The hardest color to generate is white. Your eye is very sensitive to small errors near white and the color fringes make a crisp white impossible, but try and see how close you can get.

### **C) Project Testing: Setting #2- Changing Colors**

In this experiment, we will replace the resistor module with a microcontroller that will automatically generate shifting colors.

1. Disconnect the USB cable from the charger and remove the resistor module used in Experiment #1.
2. Obtain a microcontroller chip. Look carefully to see that the chip has a dimple on one corner. The pin near that corner goes into hole E10, the other pins on that side go into E11, E12, and E13. The pins on the other side go into F10 through F13. Press the chip into the board.
3. Place the tissue back over the LED and plug in the USB cable. The LED will blink red and blue faster and faster until you can no longer see the individual flashes and you see a solid magenta color.
4. After the flashing is finished, the LED will begin to trace out the bright, or saturated, colors. It will light up with the colors of the rainbow: red to orange to yellow to green to blue, then it will go to magenta and back to red.
5. Disconnect the USB cable.
6. Install a jumper from A11 to a nearby hole in the (–) column. Your microcontroller has been programmed to recognize this jumper and produce a different color pattern.
7. Plug the USB cable into the charger. The LED will blink like before, but then the RGB LED will begin a “random walk” through the colors. The previous pattern only did saturated colors, this pattern includes non-saturated, or pastel, colors, as well as whitish colors.

### **D) Choose your Setting**

1. Decide which setting (1 or 2) you liked the best.
2. If you like Setting 1 better, remove the jumper from A11 to (–).
3. If you like Setting 2 better, leave everything as it is.

### **E) Decorating your light**

1. You can decorate the panel by painting on it with transparent matte paint. The paint starts out whitish, but it will turn clear when it dries. The dried paint will be hard to see with the lights on, but it will show up well in the dark. The LED light will be dim in the lower corners of the panel – so anything there might not show up. If you get paint on yourself, it will wash out with water. When you are done, set the panel aside to dry.
2. Paperclips will be used to support the panel. Use your pliers to straighten a paperclip.
3. Place the straightened paperclip on the line below. The triangle marks the center. Grab the wire in the center with your pliers.



4. Use the pliers to bend the wire in half. Don't make the bend too tight – you will need enough room to fit the plastic panel.



5. Use the pliers to bend the wire into the shape shown below. It doesn't have to be exact.



6. Bend another paperclip in the same way.
7. Insert one paperclip into your breadboard at locations A3 and J3.
8. Insert the other paperclip into your board at locations A28 and J28. You will have to poke it through the tape.

### Questions:

1. What is the function of the Potentiometer in the circuit?

---

---

2. What is the function of the light emitting diode?

---

---

3. What is the function of the IC chip in your design?

---

---

4. How did the change in the potentiometer value effect the light coming out?

---

---

5. Was energy conserved in our design?

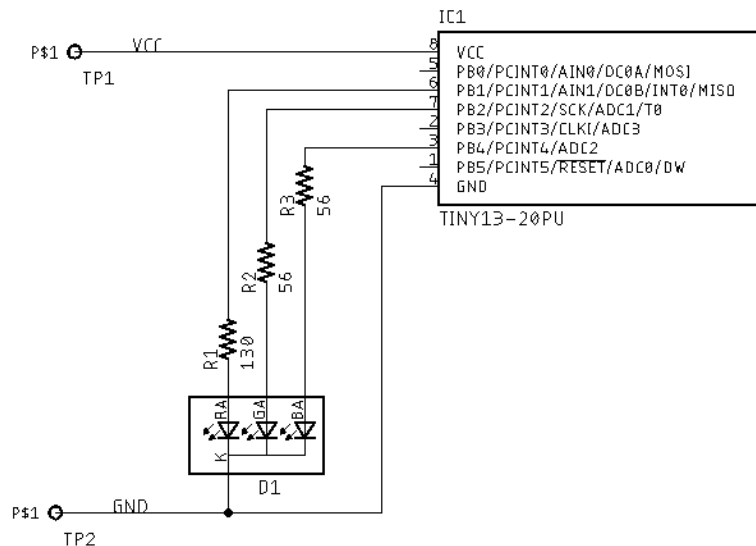
---

---

6. Electrical energy was transformed into \_\_\_\_\_ energy.

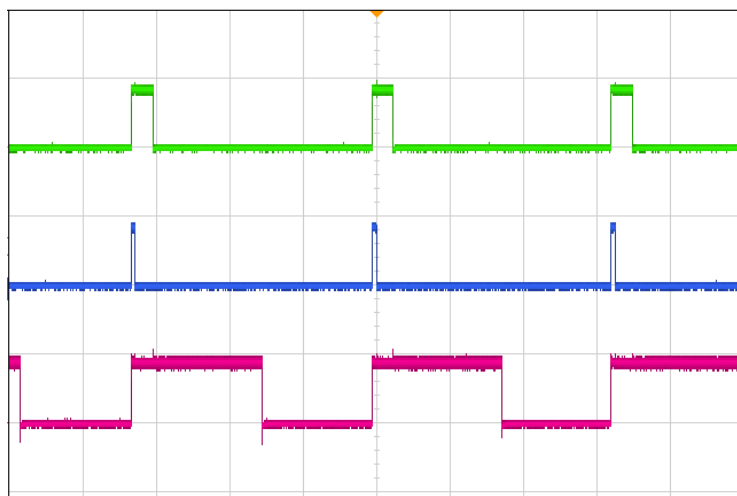
## Appendix

### Circuit Description:



The RGB LED (labeled D1) contains three individual LEDs (triangle looking things) in one package (rectangular box). Resistors R1, R2, and R3 (squiggly lines) limit the current that flows through the LEDs. The resistor for the red LED has a different value (130 ohms) compared to the resistors for the green and blue LEDs (56 ohms). This is because the red LED is made from different stuff and we need a bigger resistor to control the current.

IC1 is an ATtiny13A, an 8-bit microcontroller. Unlike the resistor module, its outputs are either “on” or “off”. We create the appearance of variable brightness by switching the output on and off very fast. If the output is on most of the time, the LED appears bright. If the output is off most of the time, the LED appears dim. The output is flashing, but it is flashing about 150 times each second - too fast for your eye to notice. There is an instrument called an oscilloscope which can reveal these signals.



The graph shows the current through the green LED (top trace), the blue LED (middle trace), and the red LED (bottom trace). The horizontal axis is time – the whole graph represents 0.02

seconds. The blue LED is hardly on at all, the green is on a bit, and the red is on a lot. This would look to your eye as a slightly yellowish red.

The microcontroller can be programmed as described in the next section. The program is stored in memory on the chip.

#### Program description

The microcontroller program is written in C. The ATtiny13A chip can be programmed using the free Microchip Studio software, but a programmer such as a Microchip Dragon is required to program the chip. The chip has a pretty small program memory, there is only about 9% free after loading this code.