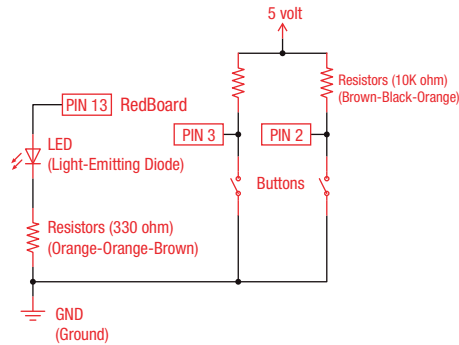


CIRCUIT #5

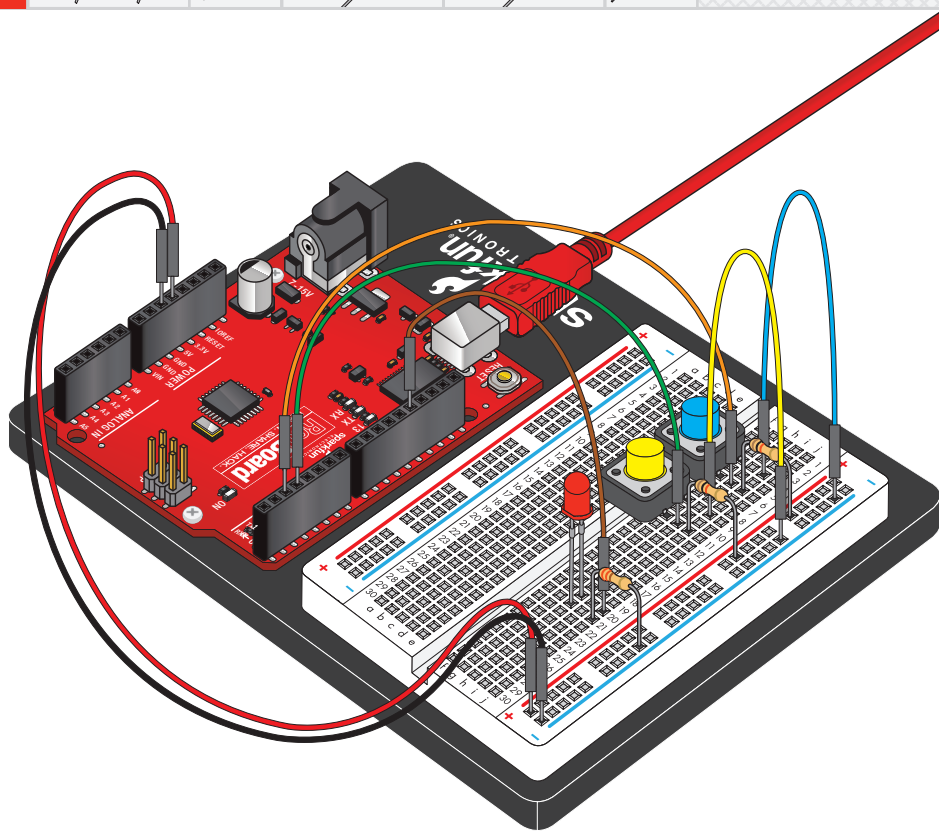
5

Push Buttons

Up until now, we've focused solely on outputs. Now we're going to go to the other end of spectrum and play around with inputs. In this circuit, we'll be looking at one of the most common and simple inputs – a push button. The way a push button works with RedBoard is that when the button is pushed, the voltage goes LOW. The RedBoard reads this and reacts accordingly. In this circuit, you will also use a pull-up resistor, which keeps the voltage HIGH when you're not pressing the button.



PARTS:	Push Button	LED	10KΩ Resistor	330Ω Resistor	Wire
	 x2	 x1	 x2	 x1	 x7





Component:	Image Reference:		
Push Button			<div>d4g4</div> <div>d6g6</div>
Push Button			<div>d9g9</div> <div>d11g11</div>
LED (5mm)			<div>h20h21</div> <div>+</div> <div>-</div>
10KΩ Resistor			<div>i6</div> <div>+</div>
10KΩ Resistor			<div>i11</div> <div>+</div>
330Ω Resistor			<div>j21</div> <div>-</div>
Jumper Wire			<div>i4</div> <div>-</div>
Jumper Wire			<div>i9</div> <div>-</div>
Jumper Wire			<div>Pin 2</div> <div>h6</div>
Jumper Wire			<div>Pin 3</div> <div>h11</div>
Jumper Wire			<div>Pin 13</div> <div>j20</div>
Jumper Wire			<div>5V</div> <div>+</div>
Jumper Wire			<div>GND</div> <div>-</div>

How to use logic like a Vulcan:

One of the things that makes the RedBoard so useful is that it can make complex decisions based on the input it's getting. For example, you could make a thermostat that turns on a heater if it gets too cold, a fan if it gets too hot, waters your plants if they get too dry, etc.

In order to make such decisions, the Arduino environment provides a set of logic operations that let you build complex "if" statements. They include:

==	EQUIVALENCE	A == B is true if A and B are the SAME .
!=	DIFFERENCE	A != B is true if A and B are NOT THE SAME .
&&	AND	A && B is true if BOTH A and B are TRUE .
 	OR	A B is true if A or B or BOTH are TRUE .
!	NOT	!A is TRUE if A is FALSE !A is FALSE if A is TRUE

You can combine these functions to build complex if() statements.

For example:

```
if ((mode == heat) && ((temperature < threshold) || (override == true)))
{
  digitalWrite(HEATER, HIGH);
}
```

...will turn on a heater if you're in heating mode **AND** the temperature is low, **OR** if you turn on a manual override. Using these logic operators, you can program your RedBoard to make intelligent decisions and take control of the world around it!

5

Arduino Code:



Open Arduino IDE // File > Examples > SIK Guide > **Circuit # 5**

Code to Note:



`pinMode(button2Pin, INPUT);`



The digital pins can be used as inputs as well as outputs. Before you do either, you need to tell the RedBoard which direction you're going.

`button1State = digitalRead(button1Pin);`



To read a digital input, you use the `digitalRead()` function. It will return HIGH if there's 5V present at the pin, or LOW if there's 0V present at the pin.

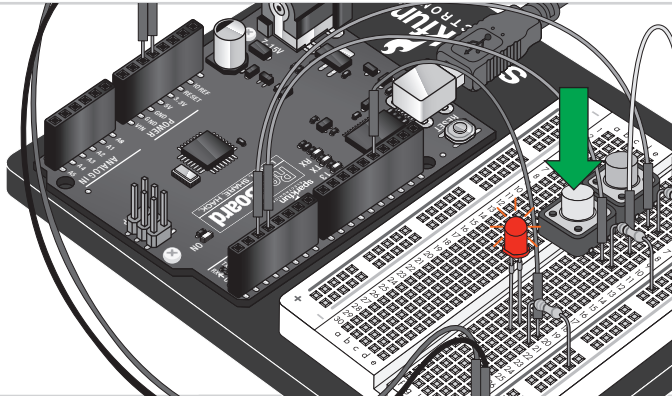
`if (button1State == LOW)`



Because we've connected the button to GND, it will read LOW when it's being pressed. Here we're using the "equivalence" operator ("==") to see if the button is being pressed.

What You Should See:

You should see the LED turn on if you press either button, and off if you press both buttons. (See the code to find out why!) If it isn't working, make sure you have assembled the circuit correctly and verified and uploaded the code to your board or see the troubleshooting tips below.



Troubleshooting:

Light Not Turning On

The pushbutton is square, and because of this it is easy to put it in the wrong way. Give it a 90 degree twist and see if it starts working.

Underwhelmed

No worries, these circuits are all super stripped down to make playing with the components easy, but once you throw them together the sky is the limit.

Real World Application:

The buttons we used here are similar to the buttons in most video game controllers.

