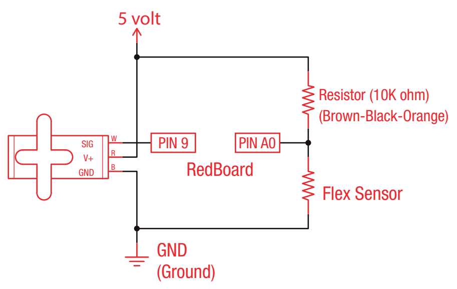
CIRCUIT #9

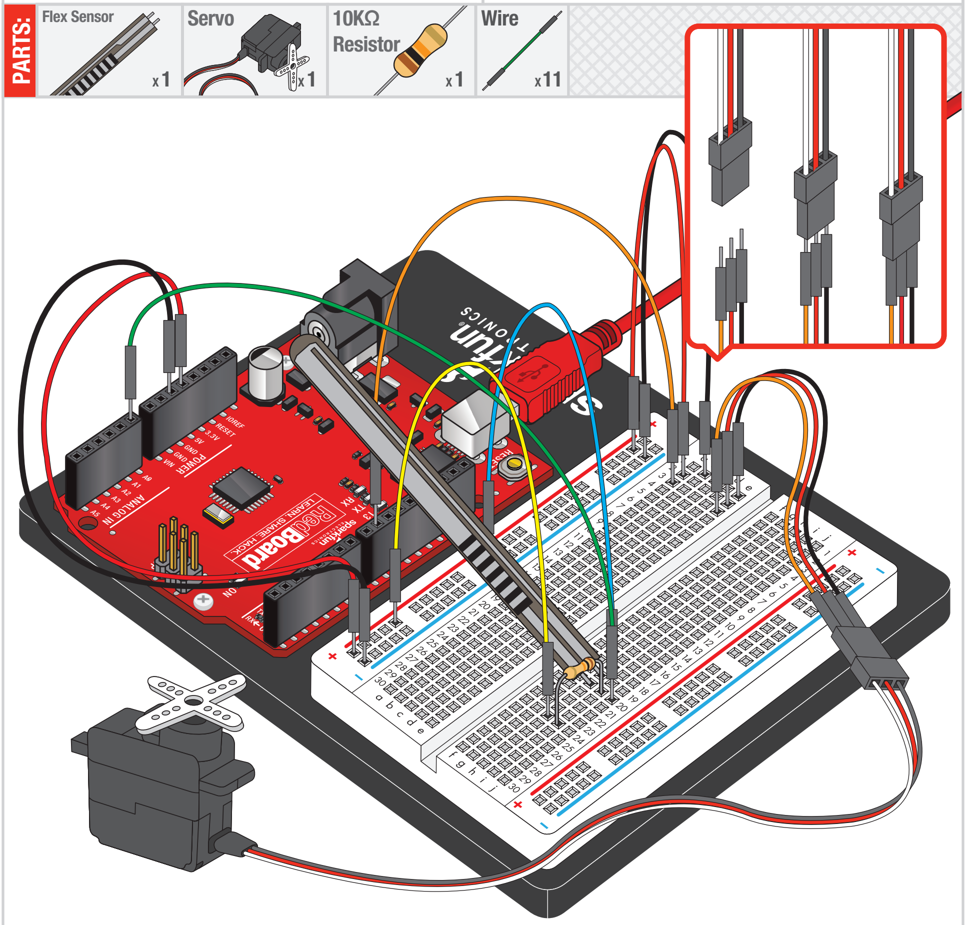
9

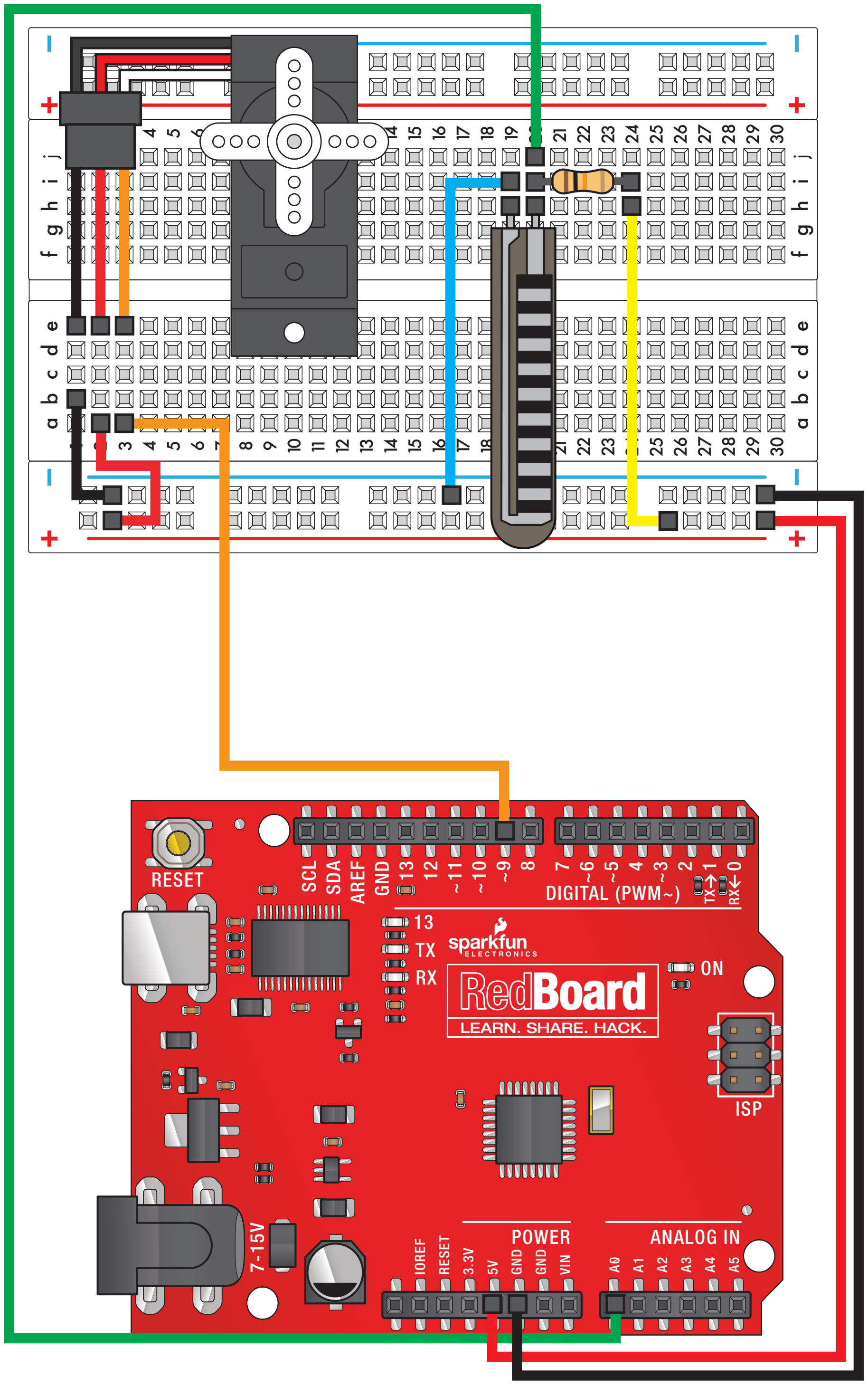


Flex Sensor

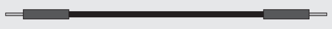
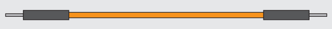
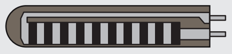
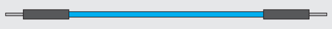
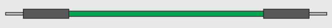
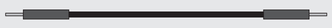
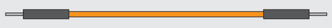
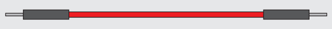
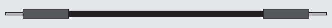
In this circuit, we will use a flex sensor to measure, well, flex! A flex sensor uses carbon on a strip of plastic to act like a variable resistor, but instead of changing the resistance by turning a knob, you change it by flexing (bending) the component. We use a "voltage divider" again to detect this change in resistance. The sensor bends in one direction and the more it bends, the higher the resistance gets; it has a range from about 10K ohm to 35K ohm. In this circuit we will use the amount of bend of the flex sensor to control the position of a servo.

PARTS:				
	Flex Sensor x1	Servo x1	10KΩ Resistor x1	Wire x11





Circuit 9: Flex Sensor

Component:	Image Reference:		
Servo			e1 e2 e3
Jumper Wire			e1 ●
Jumper Wire			e2 🔴
Jumper Wire			e3 ○
Flex Sensor			h19 h20
10KΩ Resistor			i20 i24
Jumper Wire			i19 -
Jumper Wire		A0	j20
Jumper Wire			h24 +
Jumper Wire			b1 -
Jumper Wire			a2 +
Jumper Wire		Pin 9	a3
Jumper Wire		5V	+
Jumper Wire		GND	-

Debugging your sketches using the Serial Monitor:

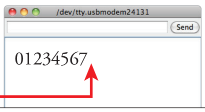
It happens to everyone - you write a sketch which successfully compiles and uploads, but you can't figure out why it's not doing what you want it to. Larger computers have screens, keyboards, and mice that you can use to debug your code, but tiny computers like the RedBoard have no such things.

The key to visibility into a microcontroller is output. This can be almost anything, including LEDs and buzzers, but one of the most useful tools is the serial monitor. Using **Serial.print()** and **println()**, you can easily output human-readable text and data from the RedBoard to a window back on the host computer. This is great for your sketch's final output, but it's also incredibly useful for debugging.

```
for (x = 0; x < 8; x++)
{
    Serial.print(x);
}
```

Let's say you wanted a **for()** loop from 1 to 8, but your code just doesn't seem to be working right. Just add **Serial.begin(9600);** to your **setup()** function, and add a **Serial.print()** or **println()** to your loop:

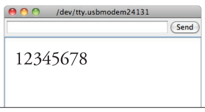
You wanted 1 to 8, but the loop is actually giving you 0 to 7. Whoops! Now you just need to fix the loop.



```
01234567
```

```
for (x = 1 ; x < 9 ; x++)
{
    Serial.print(x);
}
```

And if you run the code again, you'll see the output you wanted:



```
12345678
```

Arduino Code:

Open Arduino IDE // File > Examples > SIK Guide > Circuit # 9

Code to Note:

```
servoposition = map(flexposition, 600, 900, 0, 180);
map(value, fromLow, fromHigh, toLow, toHigh)
```

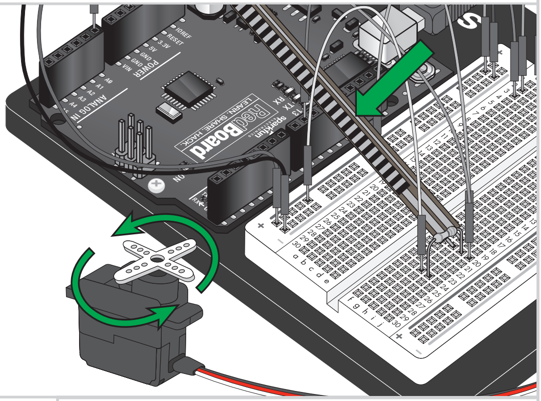
Because the flex sensor / resistor combination won't give us a full 0 to 5 volt range, we're using the map() function as a handy way to reduce that range. Here we've told it to only expect values from 600 to 900, rather than 0 to 1023.

```
servoposition = constrain(servoposition, 0, 180);
constrain(x, a, b)
```

Because map() could still return numbers outside the "to" range, we'll also use a function called constrain() that will "clip" numbers into a range. If the number is outside the range, it will make it the largest or smallest number. If it is within the range, it will stay the same.

What You Should See:

You should see the servo motor move in accordance with how much you are flexing the flex sensor. 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:

Servo Not Twisting
Even with colored wires it is still shockingly easy to plug a servo in backwards. This might be the case.

Servo Not Moving as Expected
The sensor is only designed to work in one direction. Try flexing it the other way (where the striped side faces out on a convex curve).

Servo Doesn't Move very Far
You need to modify the range of values in the call to the map() function.

Real World Application:

Controller accessories for video game consoles like Nintendo's "Power Glove" use flex-sensing technology. It was the first video game controller attempting to mimic hand movement on a screen in real time.