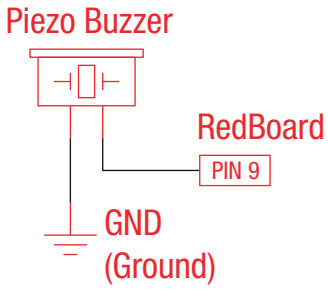
CIRCUIT #11

11

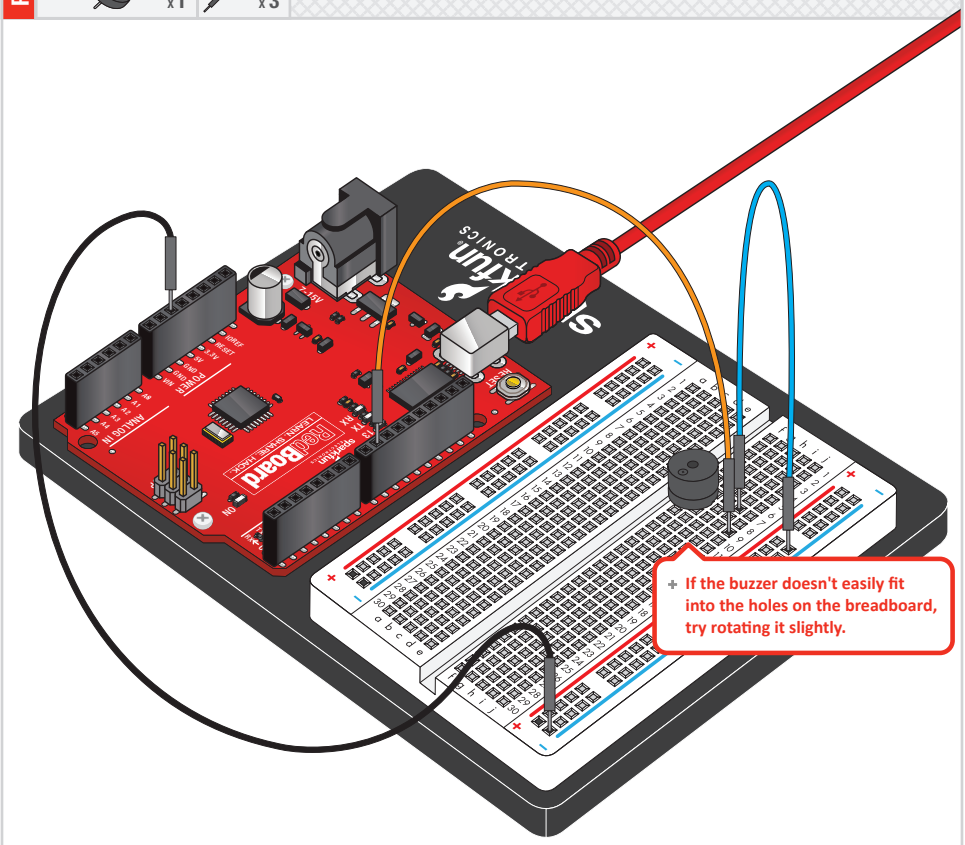
Piezo Buzzer



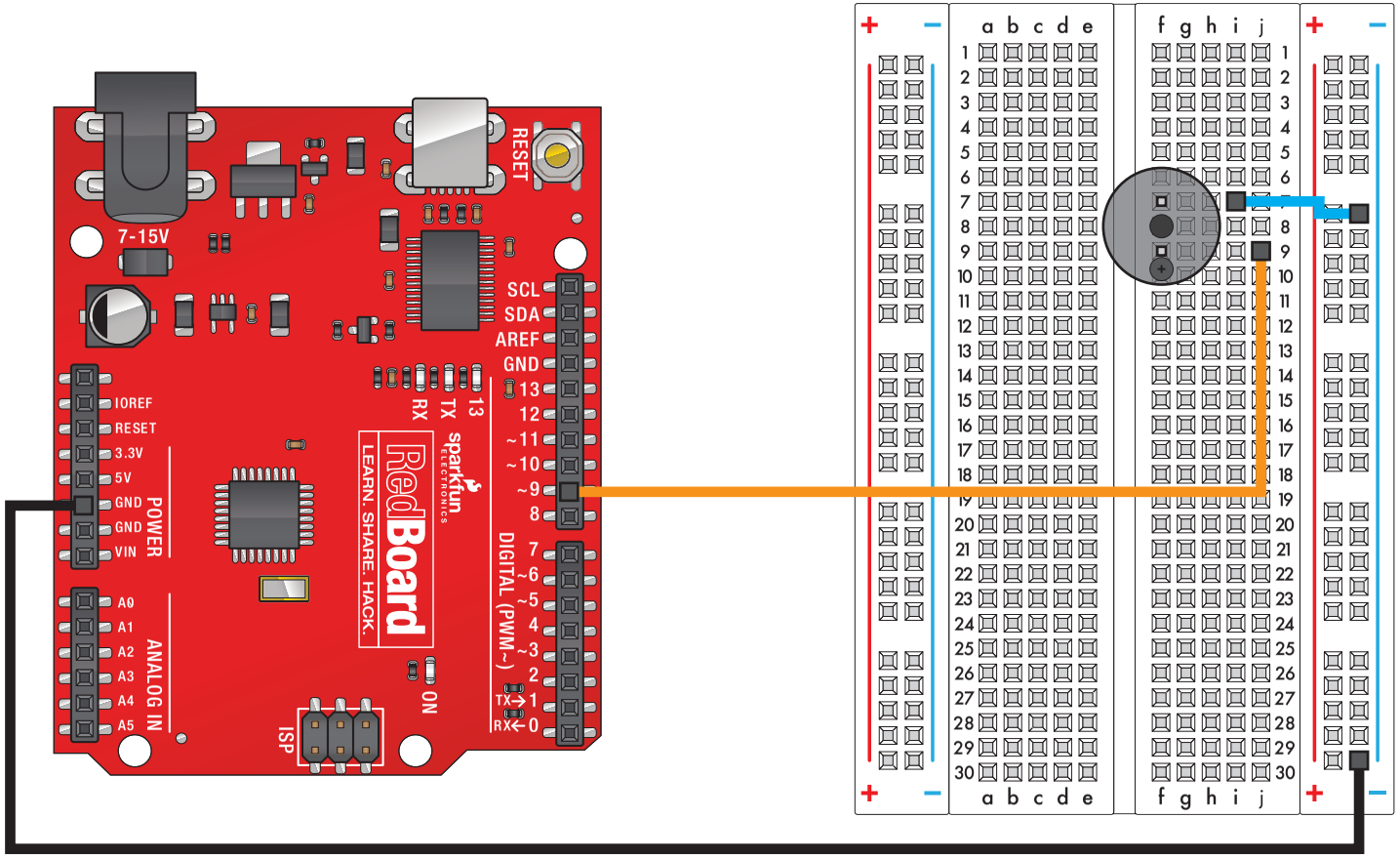
In this circuit, we'll again bridge the gap between the digital world and the analog world. We'll be using a buzzer that makes a small "click" when you apply voltage to it (try it!). By itself that isn't terribly exciting, but if you turn the voltage on and off hundreds of times a second, the buzzer will produce a tone. And if you string a bunch of tones together, you've got music! This circuit and sketch will play a classic tune. We'll never let you down!

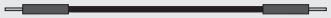
PARTS:

Piezo Buzzer	Wire
 x1	 x3



Circuit 11: Piezo Buzzer



Component:	Image Reference:		
Piezo Buzzer			f9 (+) – f7 (-)
Jumper Wire			i7 – -
Jumper Wire		Pin 9	j9
Jumper Wire		GND	-

Creating your own functions:

Arduino contains a wealth of built-in functions that are useful for all kinds of things. (See **http://arduino.cc/en/reference** for a list). But you can also easily create your own functions. First, we need to declare a function. Here's a simple example named "add," which adds two numbers together and returns the result. Let's break it down.

```
int add(int parameter1, int parameter2)
{
        int x;

        x = parameter1 + parameter2;

        return(x);
}
```

Your functions can take in values ("parameters"), and return a value, as this one does.

If you'll be passing parameters to your function, put them (and their types) in the parentheses after the function name. If your function is not using any parameters, just use an empty parenthesis () after the name.

If your function is returning a value from your function, put the type of the return value in front of the function name. Then in your function, when you're ready to return the value, put in a return(value) statement. If you won't be returning a value, put "void" in front of the function name (similar to the declaration for the setup() and loop() functions).

When you write your own functions, you make your code neater and easier to re-use. See **http://arduino.cc/en/reference/functiondeclaration** for more information about functions.

11

Arduino Code:

Open Arduino IDE // File > Examples > SIK Guide > Circuit # 11

Code to Note:

```
char notes[] = "cdfda ag cdfdg gf ";

char names[] = {'c','d','e','f','g','a','b','C'};
```

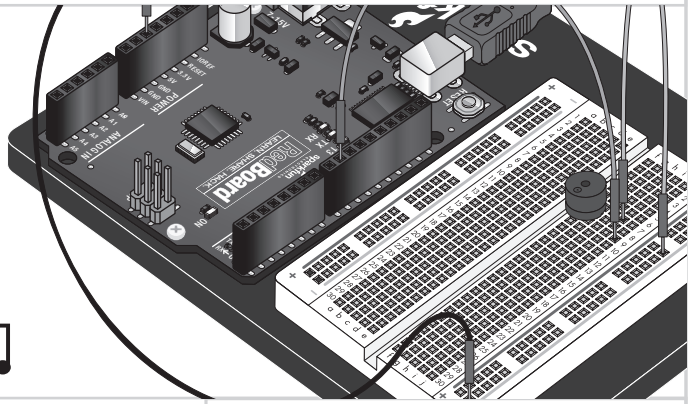
Up until now we've been working solely with numerical data, but the Arduino can also work with text. Characters (single, printable, letters, numbers and other symbols) have their own type, called "char". When you have an array of characters, it can be defined between double-quotes (also called a "string"), OR as a list of single-quoted characters.

```
tone(pin, frequency, duration);
```

One of Arduino's many useful built-in commands is the tone() function. This function drives an output pin at a certain frequency, making it perfect for driving buzzers and speakers. If you give it a duration (in milliseconds), it will play the tone then stop. If you don't give it a duration, it will keep playing the tone forever (but you can stop it with another function, noTone()).

What You Should See:

You should see - well, nothing! But you should be able to hear a song. 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:

No Sound
Given the size and shape of the piezo buzzer it is easy to miss the right holes on the breadboard. Try double checking its placement.

Can't Think While the Melody is Playing
Just pull up the piezo buzzer whilst you think, upload your program then plug it back in.

Feeling Let Down and Deserted
The code is written so you can easily add your own songs.

Real World Application:

Many modern megaphones have settings that use a loud amplified buzzer. They are usually very loud and quite good at getting people's attention.