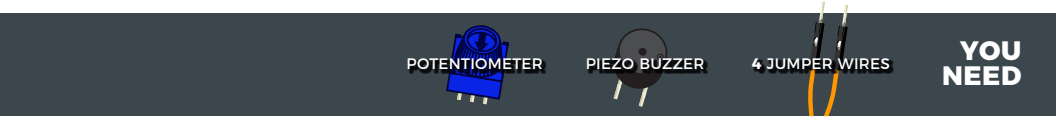
Circuit 2A: Buzzer

In this circuit, you'll use the RedBoard and a small buzzer to make music, and you'll learn how to program your own songs using arrays.

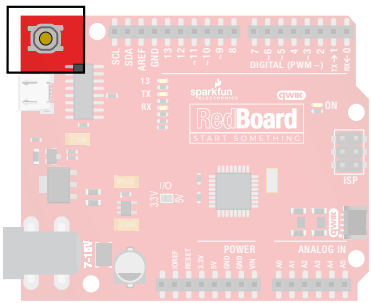


NEW COMPONENTS

BUZZER: The buzzer uses a small magnetic coil to vibrate a metal disc inside a plastic housing. By pulsing electricity through the coil at different rates, different frequencies (pitches) of sound can be produced. Attaching a potentiometer to the output allows you to limit the amount of current moving through the buzzer and lower its volume.

NEW CONCEPTS

RESET BUTTON: The RedBoard has a built-in reset button. This button will reset the board and start the code over from the beginning, running `setup()` then `loop()`.



TONE FUNCTION: To control the buzzer, you will use the `tone()` function. This function is similar to PWM in that it generates a wave that is of a certain frequency on the specified pin. The frequency and duration can both be passed to the `tone()` function when calling it. To turn the tone off, you need to call `noTone()` or pass a duration of time for it to play and then stop. Unlike PWM, `tone()` can be used on any digital pin.

ARRAYS are used like variables, but they can store multiple values. The simplest array is just a list. Imagine that you want to store the frequency for each note of the C major scale. We could make seven variables and assign a frequency to each one, or we could use an array and store all seven in the same list. To refer to a specific value in the array, an index number is used. Arrays are indexed from 0. For example, to call the first element in the array, use `array_name[0];` to call the second element, use `array_name[1];` and so on.

MUSICAL NOTE	FREQUENCY (HZ)	USING VARIABLES	USING AN ARRAY
A	220	A_FREQUENCY	FREQUENCY[0]
B	247	B_FREQUENCY	FREQUENCY[1]
C	261	C_FREQUENCY	FREQUENCY[2]
D	294	D_FREQUENCY	FREQUENCY[3]
E	330	E_FREQUENCY	FREQUENCY[4]
F	349	F_FREQUENCY	FREQUENCY[5]
G	392	G_FREQUENCY	FREQUENCY[6]