

AZIA Robotics

Lesson 6 - Create a BUZZ!

website: aziarobotics.github.io

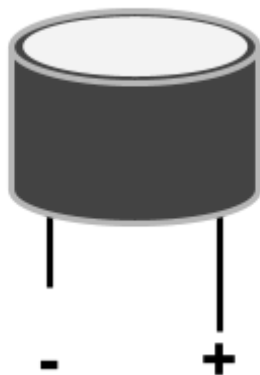
tinkercad.com class code: ALD 5WB V3J

Your Arduino kit comes with two buzzers. They both can be used to make sound output, but they operate differently.

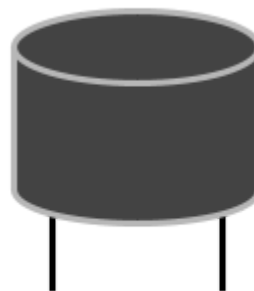
In your kit, the active buzzer has a sticker on it, but since stickers can get removed, you can also tell apart the two types of buzzers by their leads (legs):

The leads on the passive buzzer are the same length, but in the active buzzer the positive (+) lead is longer than the negative (-) lead.

Active Buzzer

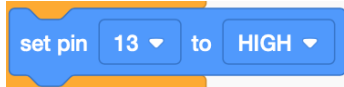


Passive Buzzer

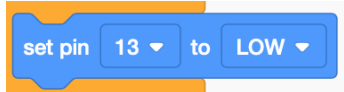


Activity 1 - Create a Buzz using the Active Buzzer

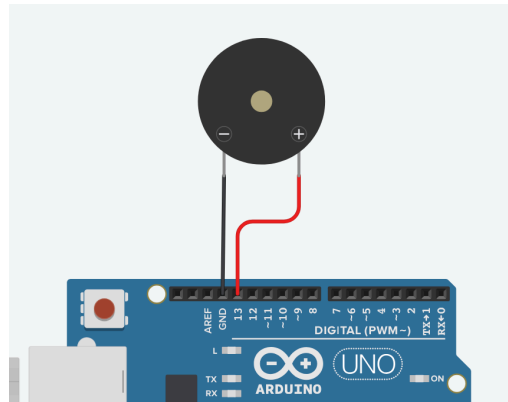
The **active buzzer** will make a buzz when it has power. So all you need to do is to connect its negative lead to GND, and its positive lead to a digital pin and set the pin to HIGH to make it buzz.



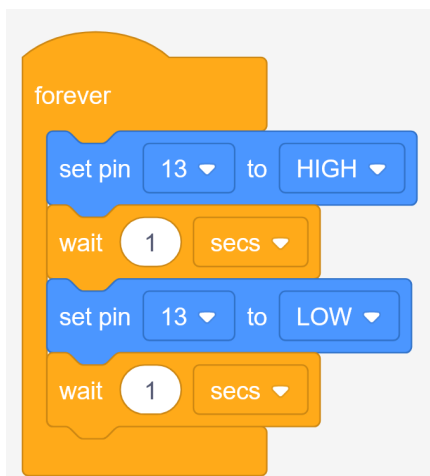
Set the pin to LOW to make the buzz stop.



Now, go ahead and build the circuit using your active buzzer on your breadboard.



Then write your code and then upload it to your arduino board. The code is exactly the same code that you had used for blinking LED, except that you now have an active speaker instead of LED in your board.



```
// C++ code
//
void setup()
{
  pinMode(13, OUTPUT);
}

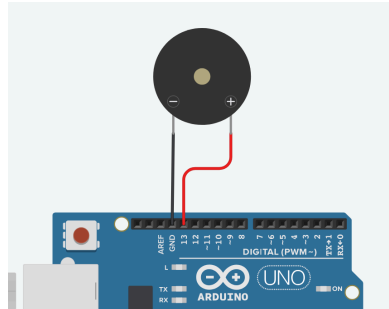
void loop()
{
  digitalWrite(13, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Activity 2 - Play some tones using the passive buzzer

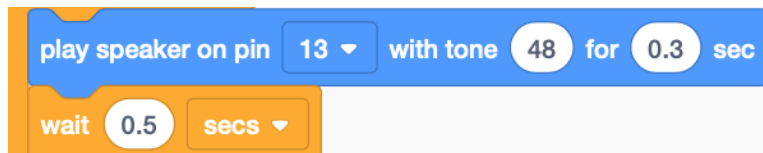
The active buzzer is easy to use, but it is used to make only one single sound.

To create more sound tones, you should use the passive buzzer. But you also need to do a little bit more coding.

The circuit for the passive buzzer is similar to the circuit for active buzzer: Just connect any one of the leads to GND and the other lead to one of the digital pins from 2 to 13.



Then you can then use the play speaker code block to play a tone:



This code block takes three inputs:

1. the pin you have connected your passive speaker to. for example 13
2. the tone number: for example, the tone number for note C is 48.
3. the duration to play the tone in seconds. So 0.5 is half a second.

For long songs, it will be easier and faster to use the C++ code instead of using blocks. Because in the C++ code, you can copy and paste the commands and change them much quicker than the blocks.

In the C++ code, you need to use the **tone** command to play a tone and use the **delay** command to wait. The tone command takes 3 parameters:

1. the pin number: for example 13.
2. the tone C++ code: for example, the tone code for note C is 262.
3. the tone duration in milliseconds. For example 300 milliseconds

```
tone(13, 262, 300);
```

```
delay(500);
```

This is the table for the some of the notes used in simple songs:

note	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
tone number	48	49	50	51	52	53	54	55	56	57	58	59
tone C++ code	262	277	297	311	330	349	370	392	415	440	466	494

Let's play the notes for the first part of the ABC song:

C C G G A A G

Using the table, you can see that you need to play the tones:

48 - 48 - 55 - 55 - 57 - 57 - 55

Play each tone for 0.3 seconds, then add a wait for 0.5 seconds:



```
// C++ code
//
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  tone(13, 262, 300); // play tone 48 (C4 = 262 Hz)
  delay(500); // Wait for 500 millisecond(s)
  tone(13, 262, 300); // play tone 48 (C4 = 262 Hz)
  delay(500); // Wait for 500 millisecond(s)
  tone(13, 392, 300); // play tone 55 (G4 = 392 Hz)
  delay(500); // Wait for 500 millisecond(s)
  tone(13, 392, 300); // play tone 55 (G4 = 392 Hz)
  delay(500); // Wait for 500 millisecond(s)
  tone(13, 440, 300); // play tone 57 (A4 = 440 Hz)
  delay(500); // Wait for 500 millisecond(s)
  tone(13, 440, 300); // play tone 57 (A4 = 440 Hz)
  delay(500); // Wait for 500 millisecond(s)
  tone(13, 392, 300); // play tone 55 (G4 = 392 Hz)
  delay(500); // Wait for 500 millisecond(s)
}
```

Now try these and see if you can guess what song they are:

ABC Song: C C G G A A G F F E E D D D C

Song 2: G G G D E E D B B A A G

Song 3: D D E D G F# D D E D A G