

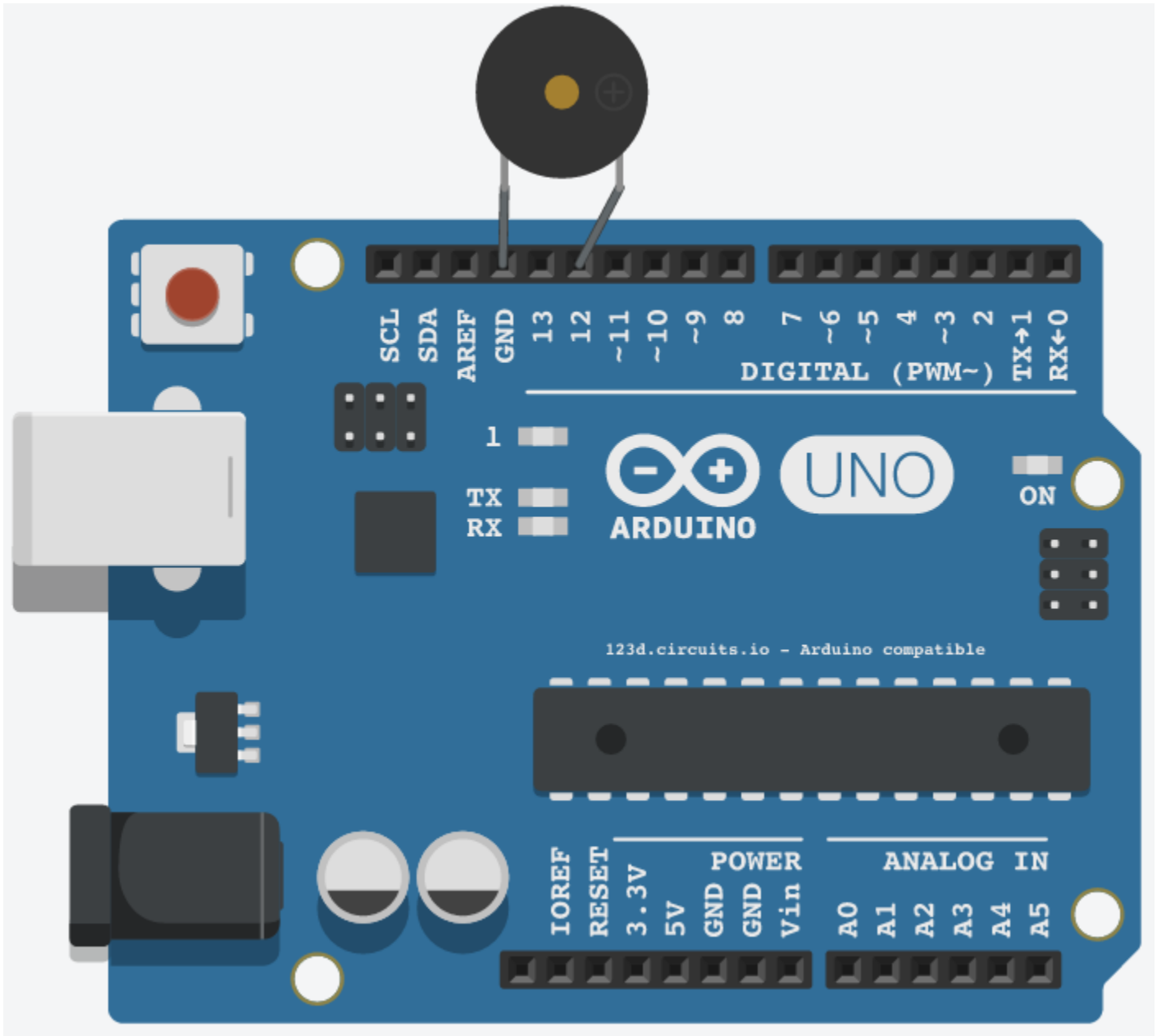
Arduino: Piezo Diagrams & Code

Brown County Library

Projects 01 & 02: Scale and Playing a Tune

Components needed:

- Arduino Uno board
- piezo



```

/*
Piezo 01 : Play a scale
Code adapted from Adafruit Arduino Lesson 10 (learn.adafruit.com/adafruit-arduino-lesson-10-making-sounds)
*/

int speakerPin = 12; // piezo is connected to digital PIN 12

int numTones = 10; // the number of tones in the scale
int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440}; // the frequency for each tone
//      mid C C# D  D# E  F  F# G  G# A

void setup(){
  for (int i = 0; i < numTones; i++) // counts through tones 0-9 using the variable i
  {
    tone(speakerPin, tones[i]); // play the tone in each array position
    delay(500); // wait 500 milliseconds
  }
  noTone(speakerPin); // stop playing
}

void loop(){
}

```

Create your own tune!

```

261 middle C
277 C#
294 D
311 D#
330 E
349 F
370 F#
392 G
415 G#
440 A
494 B
523 C

```

```

/*
Piezo 02 : Play a song
Code adapted from Sparkfun Inventor's Kit Sketch 11 (learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v32/experiment-11-using-a-piezo-buzzer)
*/

const int speakerPin = 12; // piezo is connected to digital PIN 12

// We'll set up an array with the notes we want to play. Change these values to make different
songs!
const int songLength = 18; // Length must equal the total number of notes and spaces

char notes[] = "cdfda ag cdfdg gf "; // Notes is an array of text characters corresponding to the
notes in your song. A space represents a rest (no tone).
int beats[] = {1,1,1,1,1,1,4,4,2,1,1,1,1,1,1,4,4,2}; // Beats is an array of values for each note
and rest.

// A "1" represents a quarter-note, 2 a half-note, etc. Don't
forget that the rests (spaces) need a length as well.

int tempo = 150; // The tempo is how fast to play the song - to make the song play faster,
decrease this value.

void setup() {
  pinMode(speakerPin, OUTPUT); // sets the piezo as an output
}

void loop() {
  int i, duration;

  for (i = 0; i < songLength; i++) { // step through the song arrays
    duration = beats[i] * tempo; // length of note/rest in milliseconds

    if (notes[i] == ' ') { // if it is a rest
      delay(duration); // then pause for a moment
    }
    else { // otherwise, play the note
      tone(speakerPin, frequency(notes[i]), duration);
      delay(duration); // wait for tone to finish
    }
    delay(tempo/10); // brief pause between notes
  }
}

```

```

}

while(true){} // we only want to play the song once, so we'll pause forever (remove this line
to play repeatedly)
}

int frequency(char note) { // this function takes a note character (a-g), and returns the
corresponding frequency in Hz for the tone() function.
    int i;
    const int numNotes = 8; // number of notes we're storing

    // The following arrays hold the note characters and their corresponding frequencies. The last
"C" note is uppercase
    // to separate it from the first lowercase "c". If you want to add more notes, you'll need to
use unique characters.
    // For the "char" (character) type, we put single characters in single quotes.

    char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
    int frequencies[] = {262, 294, 330, 349, 392, 440, 494, 523};

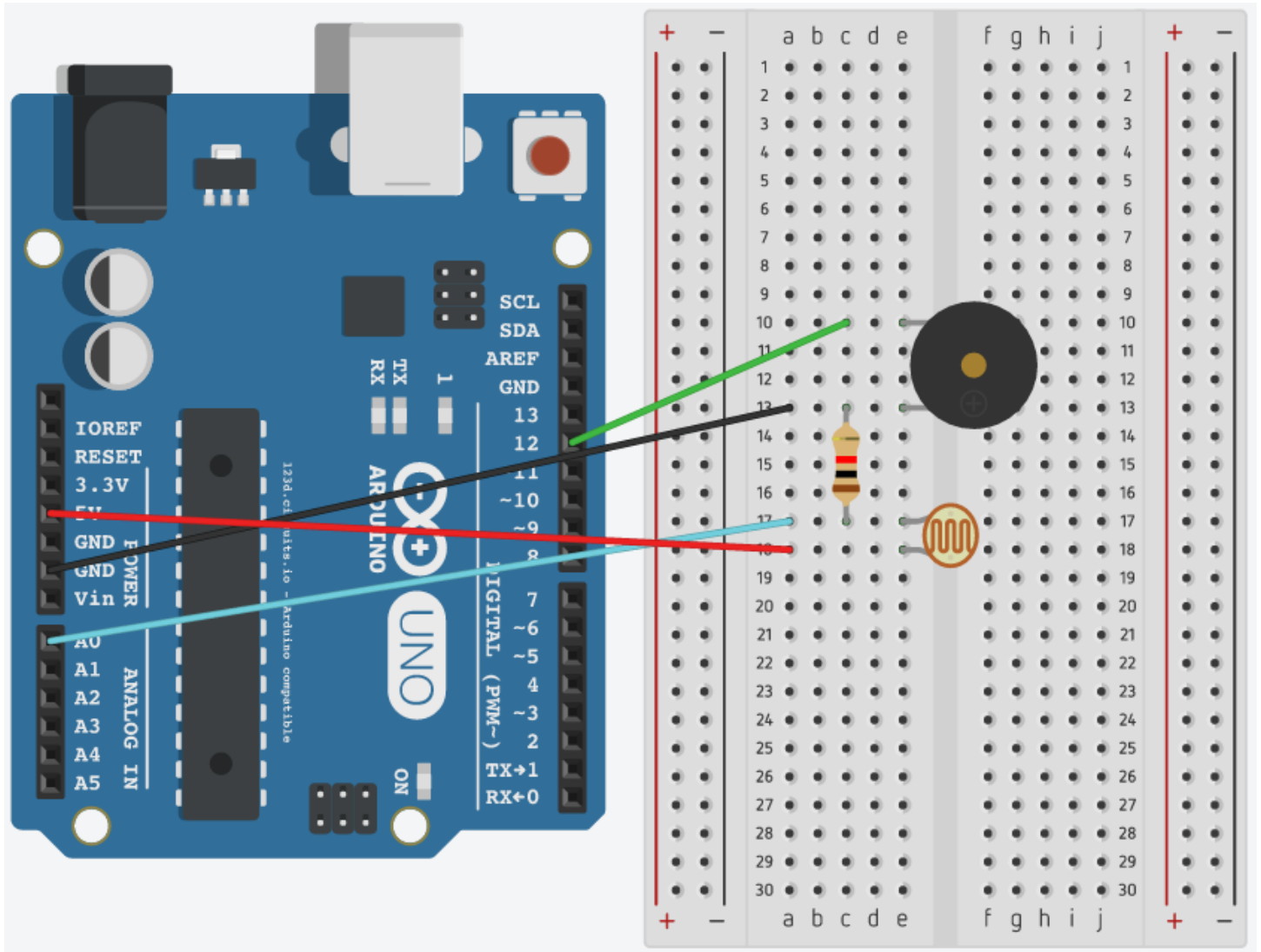
    // Now we'll search through the letters in the array, and if we find it, we'll return the
frequency for that note.
    for (i = 0; i < numNotes; i++) { // step through the notes
        if (names[i] == note) { // is this the one?
            return(frequencies[i]); // if yes, return the frequency
        }
    }
    return(0); // we looked through everything and didn't find it, but we still need to return a
value, so return 0.
}

```

Projects 03: Theramin

Components needed:

- Arduino Uno board
- breadboard
- piezo
- 4 jumper wires
- 1 1K ohm resistor
- light dependent resistor (sometimes called a *photoresistor*)



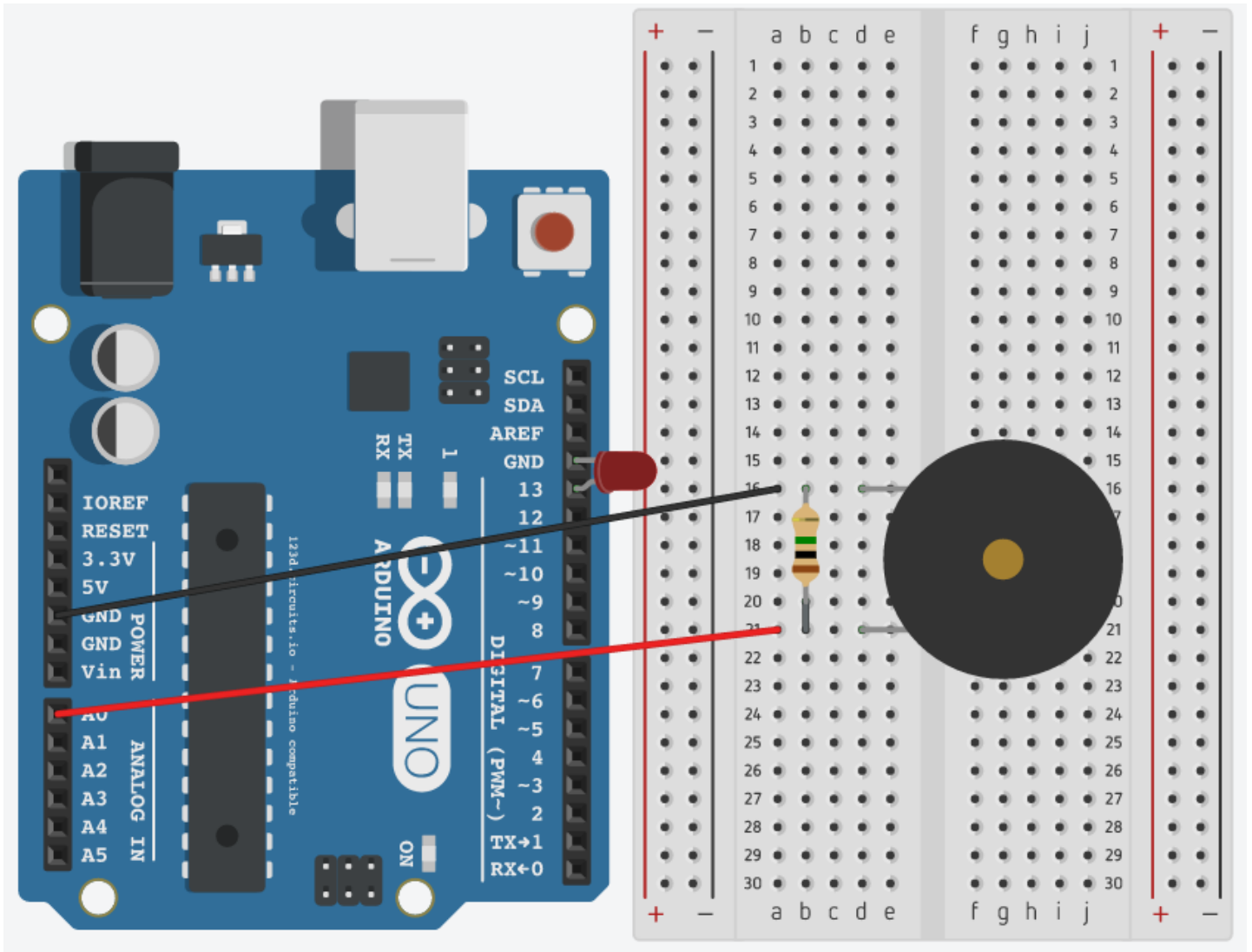
```
/*  
Piezo 03 : Theremin  
Code adapted from Adafruit Arduino Lesson 10 (learn.adafruit.com/adafruit-arduino-lesson-10-making-sounds)  
*/  
  
int speakerPin = 12;    // piezo is connected to digital PIN 12  
int photoresistorPin = 0; // photoresistor is connected to analog PIN 0  
  

```

Projects 04: Knock Sensor

Components needed:

- Arduino Uno board
- breadboard
- piezo
- 2 jumper wires
- 1 million ohm resistor
- 1 LED



```

/*
Piezo 04 : Knock Sensor
Code adapted from Arduino Knock Sensor Tutorial (arduino.cc/en/Tutorial/KnockSensor)
and Learning About Electronics (learningaboutelectronics.com/Articles/Piezo-knock-sensor-circuit.php)
*/

int LED = 13;      // LED connected to digital PIN 13
int knockSensor = 0; // variable used to store the value coming from the sensor
int threshold = 300; // minimum signal from the piezo to register as a knock (adjust this as necessary)

void setup() {
  pinMode(LED, OUTPUT); // sets the digital PIN (LED) as an output
}

void loop() {
  int val= analogRead(knockSensor); // listen for a knock

  if (val >= threshold) { // if a knock that is higher than the set threshold occurs...
    digitalWrite(LED, HIGH); // turn the LED on
    delay(500); // wait for half a second
    digitalWrite(LED, LOW); // turn the LED off
  } else {
    digitalWrite(LED, LOW); // if there is no knock, leave the LED off
  }
}

```


Ideas to Build On

Add code to the Knock Sensor you can track the knocks using the Serial Monitor:

<https://www.arduino.cc/en/Tutorial/KnockSensor>

Find more musical notes:

Search for tables that list the frequencies for additional notes to create your own song with new notes that weren't included in the first project. Search the Internet ("simple song notes") to find easy songs and write the appropriate code with the notes you find.

Find simple tunes:

Search the Internet for other simple tunes that you can play with piezo ("arduino simple piezo song" or similar) using code that others have already written.

Can you get a tune to play when the piezo is knocked?

- Hint: You'll need to combine the code for projects #2 & #4

Learn More

Want to learn more about how piezos work? Try these resources:

Adafruit Arduino Lesson 10. Making Sounds.

<https://learn.adafruit.com/adafruit-arduino-lesson-10-making-sounds>

Sparkfun Inventor's Kit Sketch 11.

<https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v33/experiment-11-using-a-piezo-buzzer>

Adafruit Arduino Lesson 10. Making Sounds.

<https://learn.adafruit.com/adafruit-arduino-lesson-10-making-sounds>

Arduino Knock Sensor Tutorial

[arduino.cc/en/Tutorial/KnockSensor](https://www.arduino.cc/en/Tutorial/KnockSensor)

Learning About Electronics

learningaboutelectronics.com/Articles/Piezo-knock-sensor-circuit.php