Project 3: Music with Piezo and Arduino

Description:

The Piezo speaker is a small metal plate enclosed in a round case that flexes and 'clicks' when current current is passed through the plate. By quickly switching the current on and off the plate 'clicks' rapidly we hear this as a pitch. This project will have us use the 'Tone' block to create musical patterns using the Arduino and the Piezo.

Building and Wiring:

You need the following:

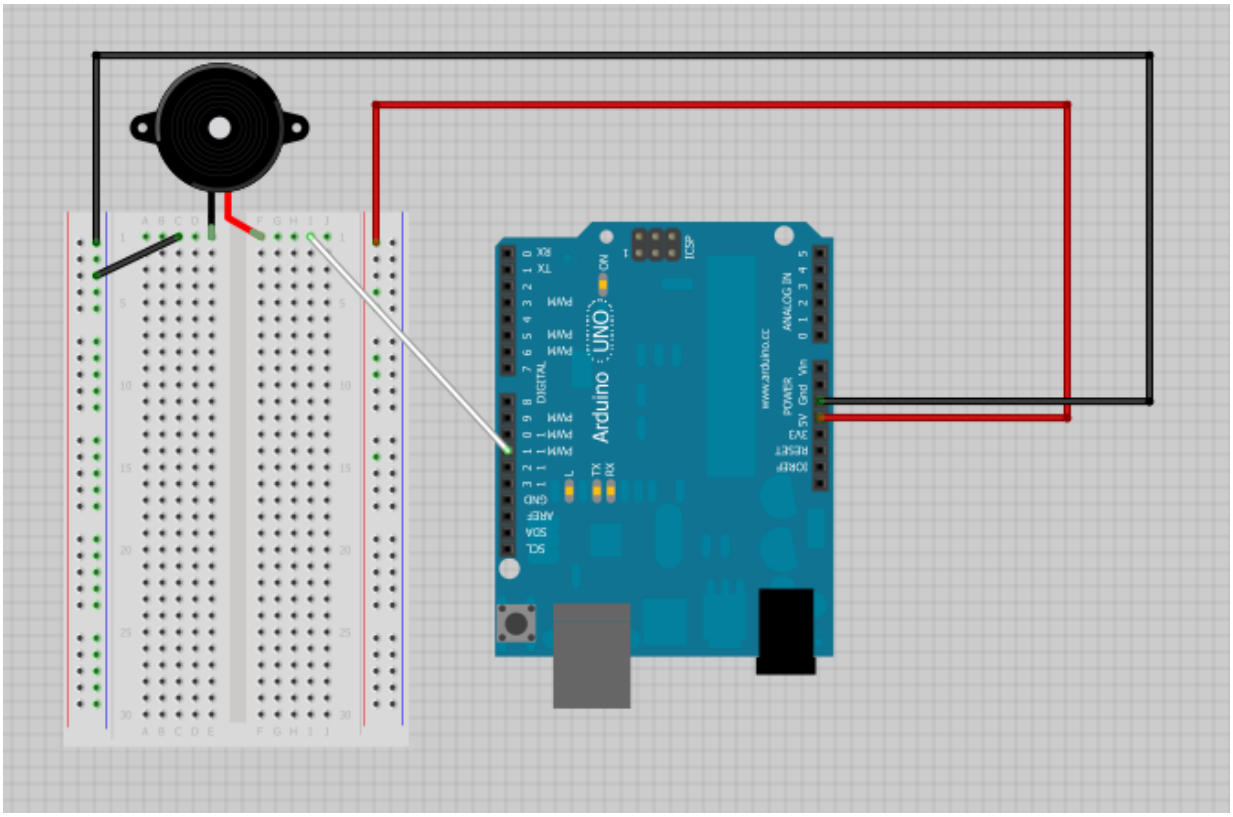
1. Arduino Board and Breadboard set up from Project 2 (8 LED lights)
2. Piezo Speaker

3. White or light colored wire for signal

4. Dark colored wire for ground. (Black)

Process:

You may keep the same design from the LED Row of lights project. The diagram below shows the wiring without the LED lights.



A. White wire from Arduino Pin 11 to I1 on Breadboard
B. Piezo + on F1
C. Piezo – on E1
D. Black wire from C1 to Blue Rail (Ground)

Encoding Music:

Description of Blocks:

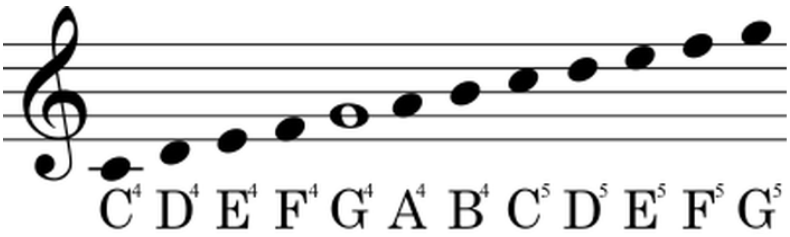
The Tone function takes three parameters; PIN#, Pitch, and Delay in milliseconds (Duration of the Pitch).

```
tone(7, 256, 500);
```

PIN#: Pin on Arduino board for signal
Pitch: The note you want the Piezo to Play
Delay: How many milliseconds to play the pitch.

Music notation:

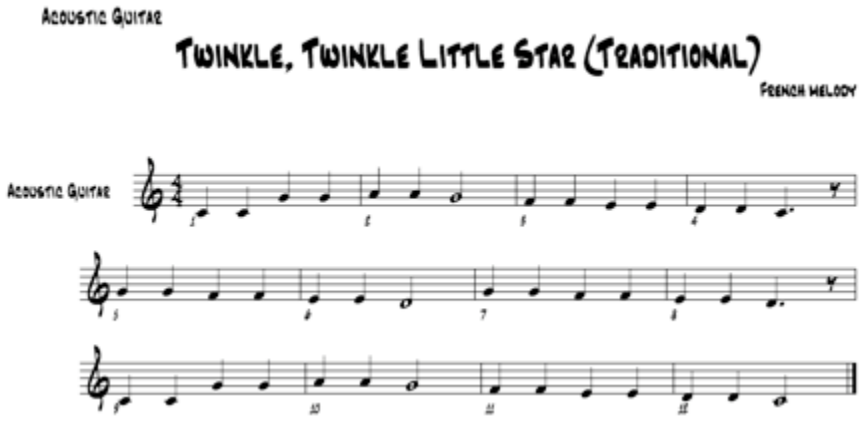
Music notation conveys two parameters: Which Pitch to play and how long to play the pitch. The following chart shows how pitches are related to the vertical location of the notes:



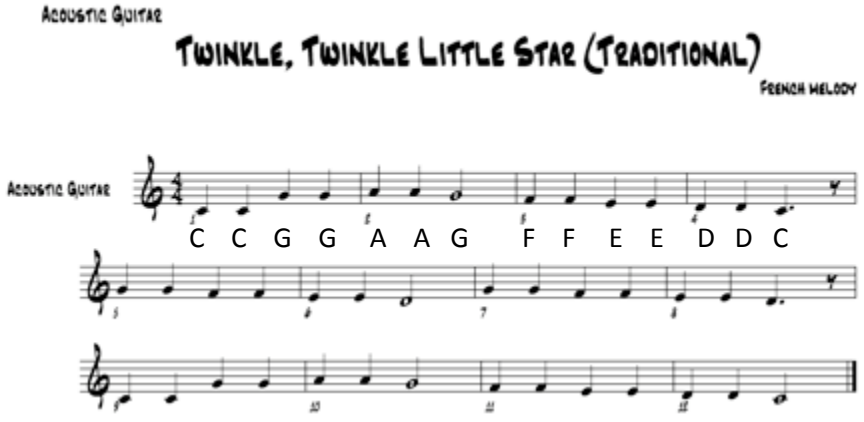
The shape of the notes shows the rhythm (duration).

Note	Name	Beats
	Whole note	4 beats
	Half note	2 beats
	Quarter note	1 beat
	Eighth note	½ beat
	Sixteenth note	¼ beat

For this lesson we will program the melody for 'Twinkle Twinkle Little Star. Here is the notation below:

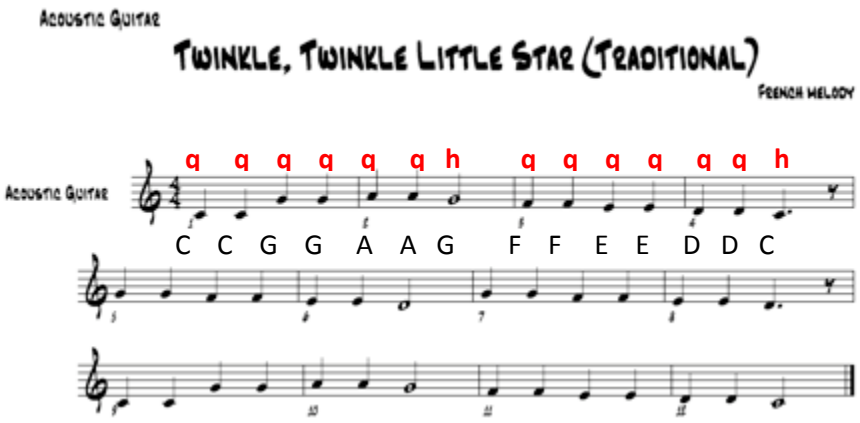


First, label the pitches beneath each note. The first line is done here as an example:



Now, label the rhythm values for each note above the line:

q = Quarter note
h = Half note



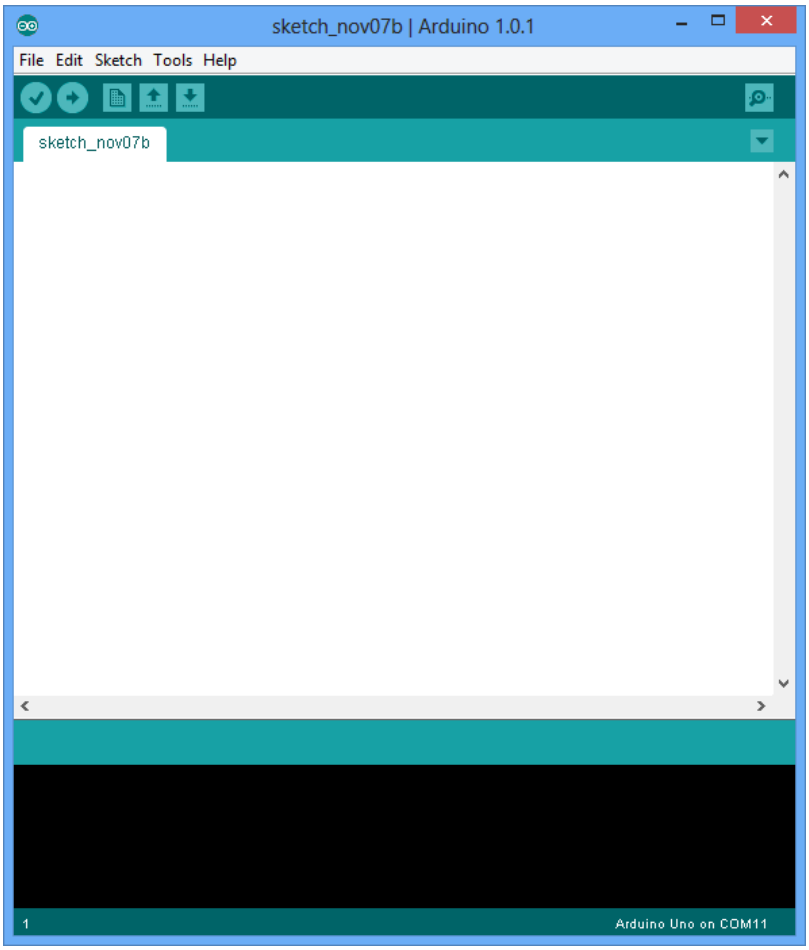
Here is a blank copy of Twinkle Twinkle to label:

Arduino Programming

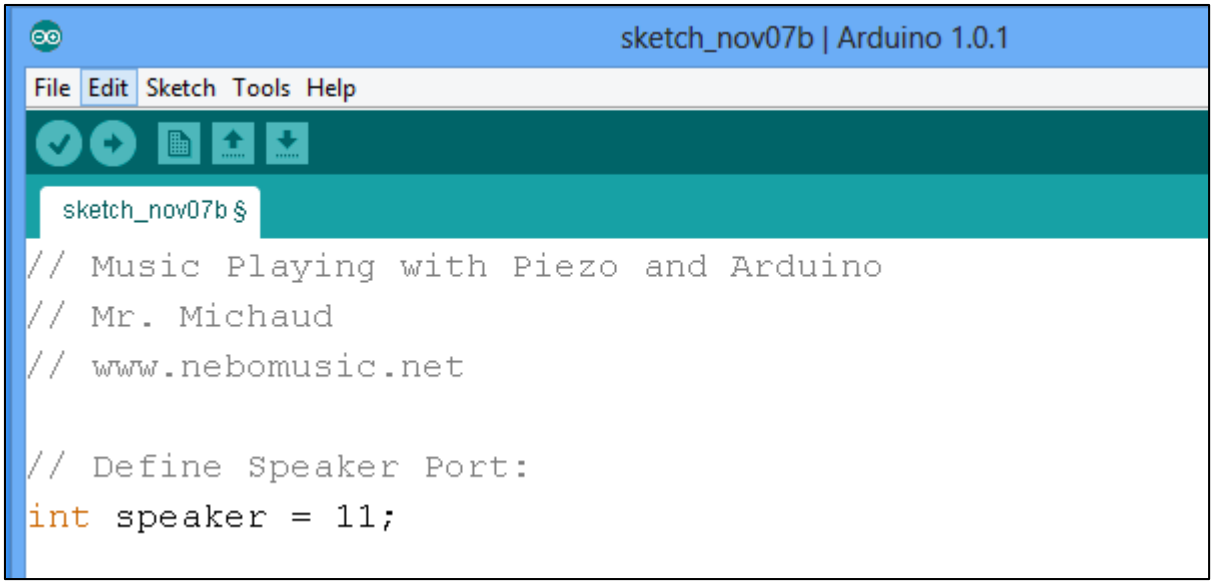
For this program we will define variables for rhythm notation and tempo. These will help us control the speed and rhythm of the music.

Process:

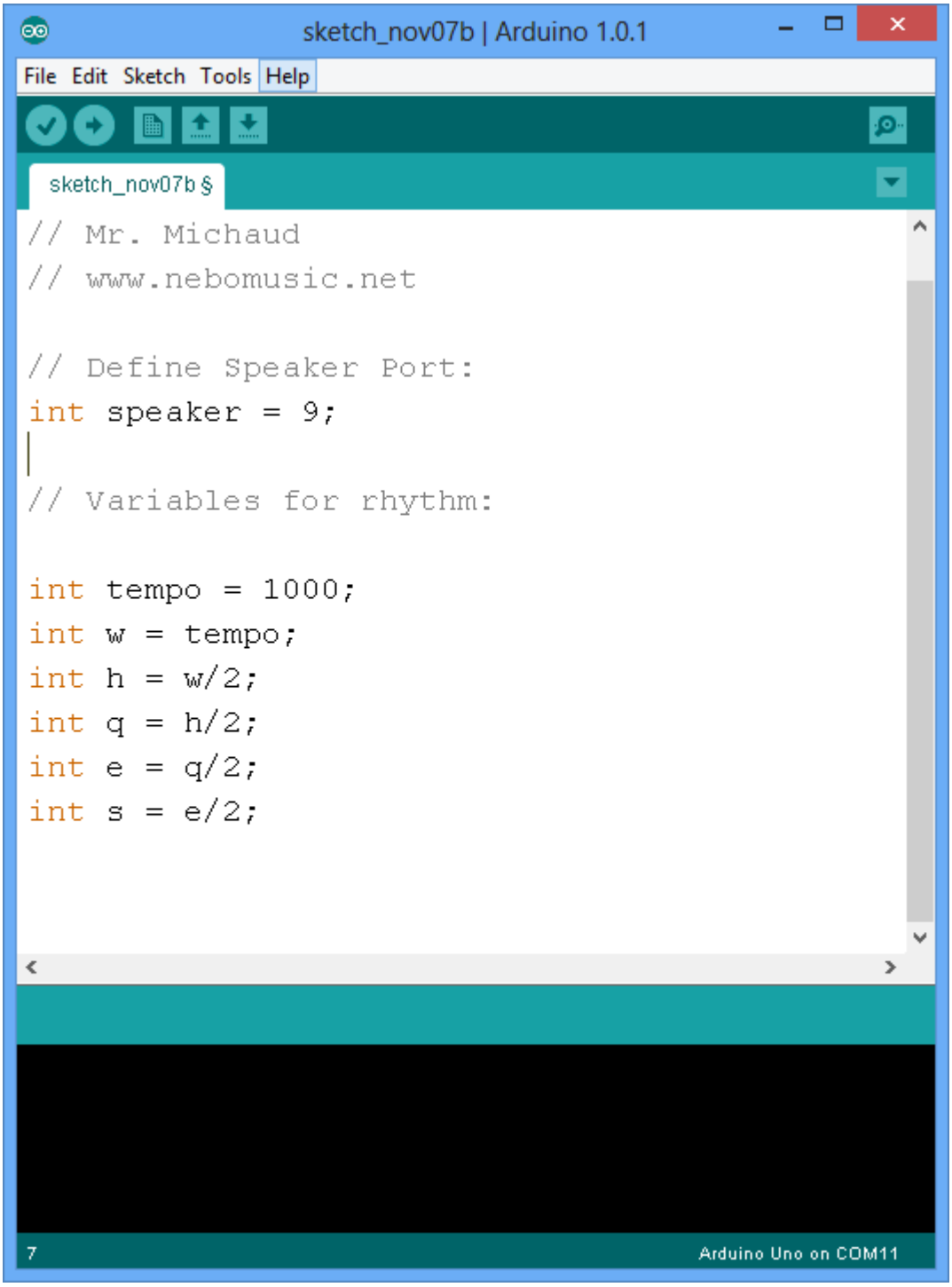
1. Start Arduino Sketch Program



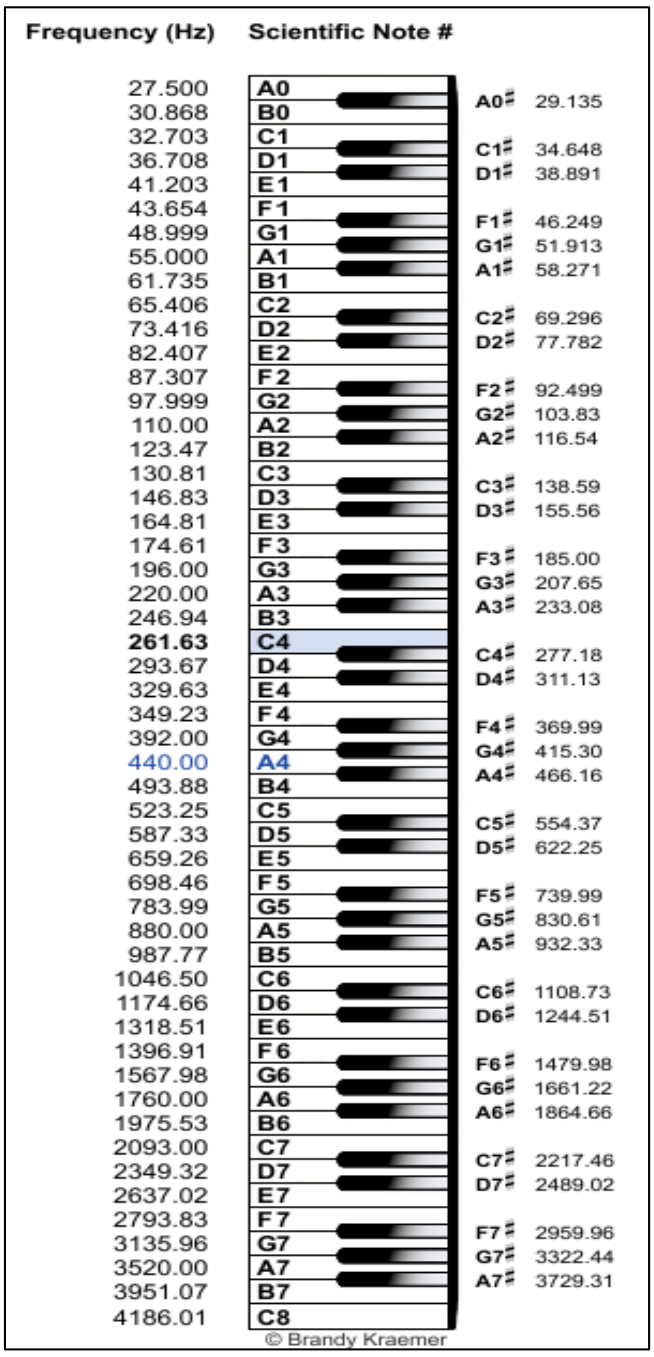
2. We need to declare a port for the Piezo. Type in the following code to define the pin for the Piezo speaker:



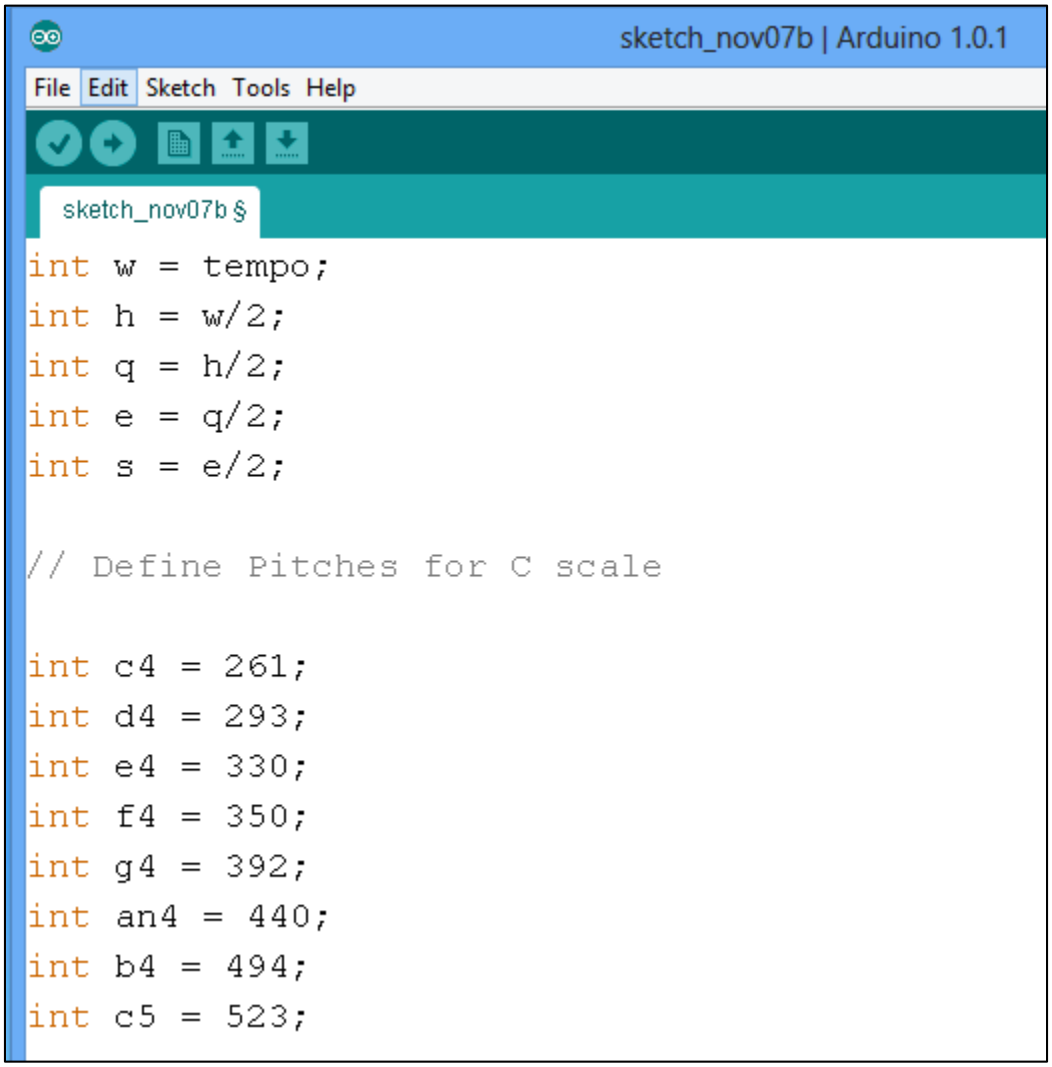
3. We now need to define the values for rhythm. Type the following rhythm values into the program.



4. We now will define values for pitches. Below is a chart representing pitch values for notes on the Piano Keyboard:



5. Define the following pitches for a C Major Scale:




```
int w = tempo;
int h = w/2;
int q = h/2;
int e = q/2;
int s = e/2;

// Define Pitches for C scale

int c4 = 261;
int d4 = 293;
int e4 = 330;
int f4 = 350;
int g4 = 392;
int an4 = 440;
int b4 = 494;
int c5 = 523;
```

6. We will now write a function to play the scale in ascending note order. Outline the function as follows:

```
// ascending Scale Function
void scaleUp() {

} // end scaleUp()
```

7. Inside the scale function, define an Array of integers listing the notes of the scale:

```
// ascending Scale Function
void scaleUp() {
  int scale [] = {c4, d4, e4, f4, g4, an4, b4, c5};

} // end scaleUp()
```

8. Write a for loop to move through the scale array and play each pitch:

```
// ascending Scale Function
void scaleUp() {
  int scale [] = {c4, d4, e4, f4, g4, an4, b4, c5};

  for (int n = 0; n < 8; n++) {
    tone(speaker, scale[n], q);
    delay(q);
  }

} // end scaleUp()
```

9. Now we will test the function. Write the void setup() function:

```
void setup() {

} // end setup
```

10. Set the pin mode of the speaker pin to OUTPUT.

```
void setup() {
  // Set Speaker Pin to Output
  pinMode(speaker, OUTPUT);

} // end setup
```

11. Write the void loop() function:

```
void loop() {

} // end loop
```

12. Call the scaleUp() function in loop():

```
void loop() {
  // Play the scale
  scaleUp();

} // end loop
```

13. Download the program to the Arduino and test – you should hear a beautiful scale! (Over and over!)

14. Now that we have a experimented with scaleUp, lets create functions to play Twinkle Twinkle. Declare a new function named 'twinkle1'

```
// Function Twinkle 1
void twinkle1() {

}
```

15. Define an integer array for the first line of Twinkle pitches:

```
// Function Twinkle 1
void twinkle1() {
  // Notes for Twinkle Line 1
  int lineA [] = {c4, c4, g4, g4, an4, an4, g4, f4, f4, e4, e4, d4, d4, c4};

}
```

16. Define an array for the rhythm values of Twinkle:

```
// Function Twinkle 1
void twinkle1() {
  // Notes for Twinkle Line 1
  int lineANotes [] = {c4, c4, g4, g4, an4, an4, g4, f4, f4, e4, e4, d4, d4, c4};
  int lineARhythm [] = {q, q, q, q, q, q, h, q, q, q, q, q, h};

}
```

17. Write the for loop that moves through the arrays and plays the notes:

```
// Function Twinkle 1
void twinkle1() {
  // Notes for Twinkle Line 1
  int lineANotes [] = {c4, c4, g4, g4, an4, an4, g4, f4, f4, e4, e4, d4, d4, c4};
  int lineARhythm [] = {q, q, q, q, q, q, h, q, q, q, q, q, h};

  for (int n = 0; n < 14; n++) {
    tone(speaker, lineANotes[n], lineARhythm[n]);
    delay(lineARhythm[n]);
  }

} // end twinkle1
```

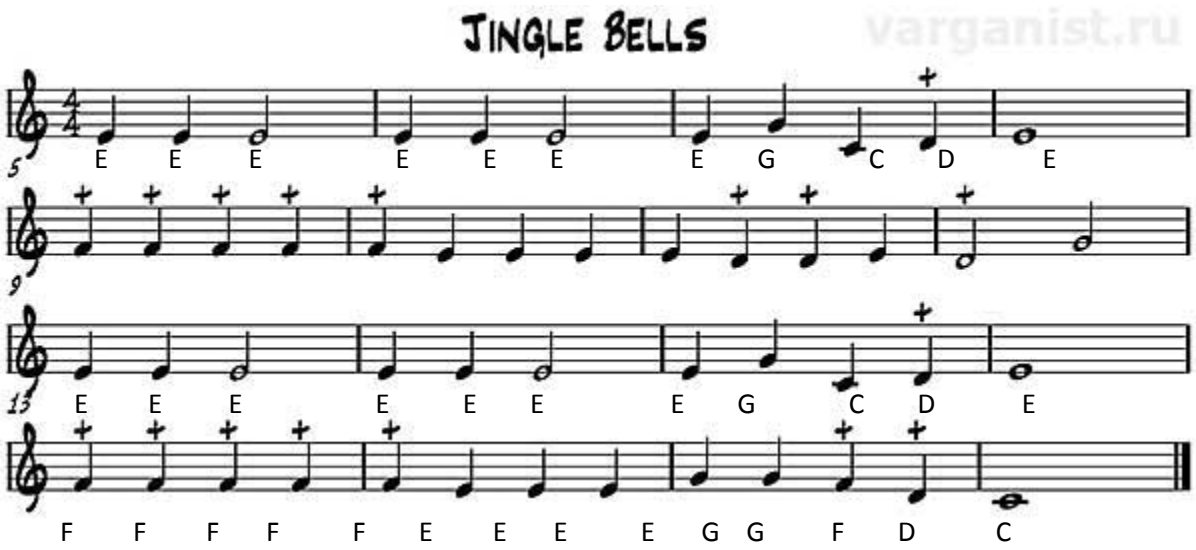
18. Call the twinkle1() function in the void loop and download to the Arduino:

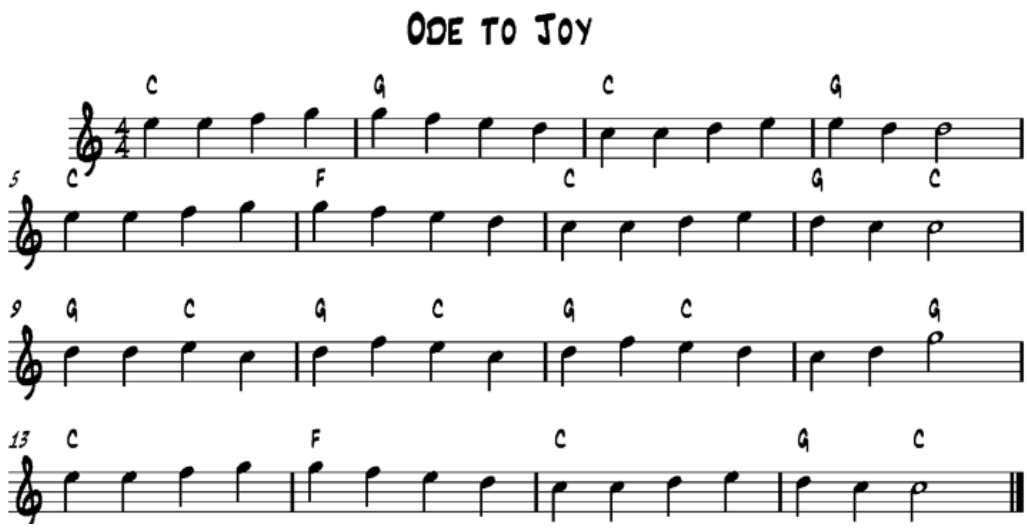
```
void loop() {
  // Play the scale
  scaleUp();
  twinkle1();

} // end loop
```

19. Complete functions twinkle2() and twinkle3() to play the rest of the song. Note that the third line of Twinkle is the same as the first (hint . . .)

20. Try to write functions for some of the other songs listed.

Additional Songs:





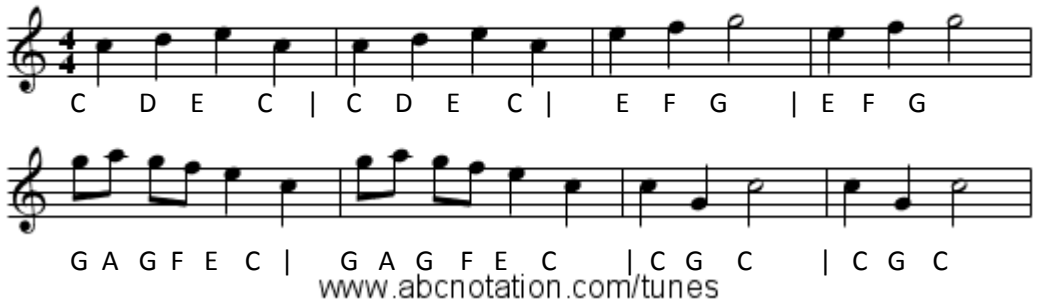
E E F G | G F E D | C C D E | E D D

E E F G | G F E D | C C D E | D C C

D D E C | D F E C | D F E D | C D G

E E F G | G F E D | C C D E |D C C

Frere Jacques



Vocabulary:

Piezo: Electric device with a thin plate of metal that quickly changes position when current is changed. By quickly changing the current to on and off the Piezo can emit a single tone. Piezo's are used for the 'beeping' sounds in smoke alarms, alarms, greeting cards, and other simple electronic devices.

Pulse Width Modulation (PWM): Rapidly changing the 'on and off' signal thousands of times a second. This allows for the Arduino to produce pitches on a Piezo or drive servos.