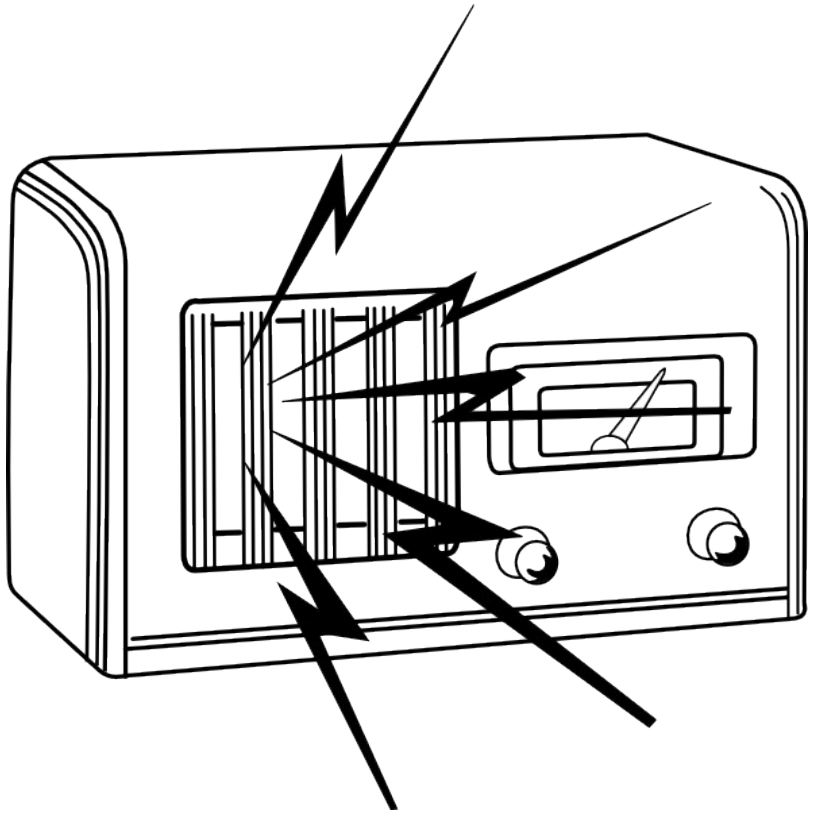
Making Noise using Analogue Input



Created by Thomas Preece for the Technology Volunteers

Warwick Volunteers

Task

1. Create circuit diagram as shown below
2. Open the Arduino IDE program on your computer
3. In the menu at the top of the arduino program go to
   File → Examples → 02.Digital → tonePitchFollower
4. Upload the code to your arduino
5. You will notice turning the potentiometer doesn't affect the buzzer pitch much, that is because the map command isn't correct. By using the Serial Monitor figure out what the range of values the potentiometer gives is and correct the map command so that the initial range it takes is what you found and not 800-1000
6. Upload your code and check that the buzzer has a greater range

Extensions

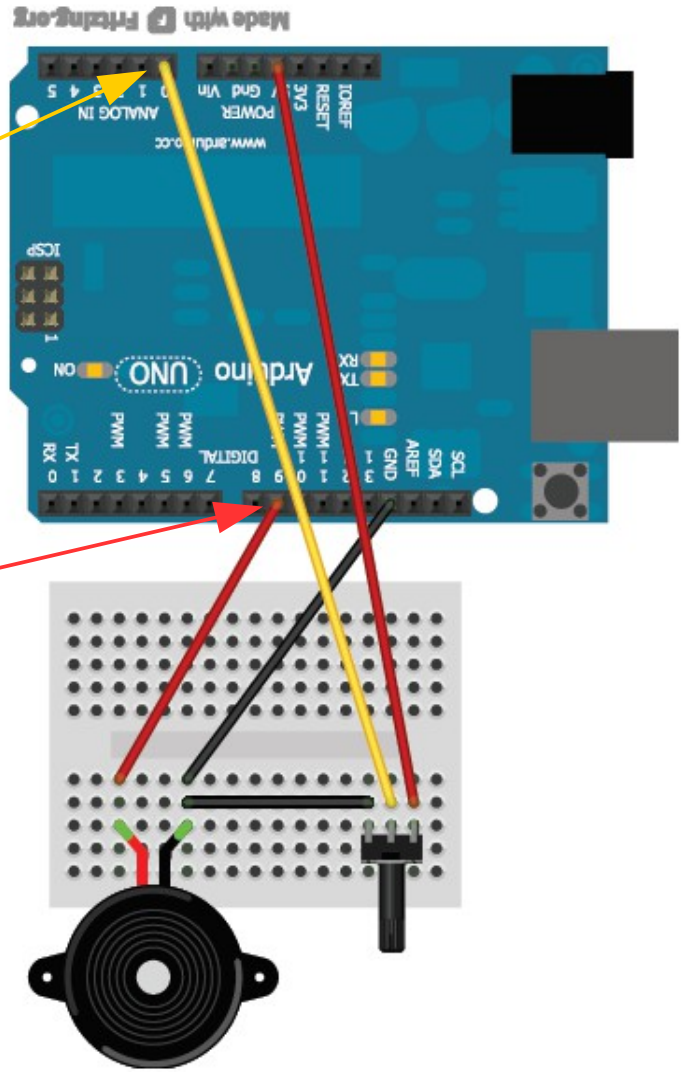
1. Add another tone command to make a variable pitch ambulance noise.
2. Load up File → Examples → 02.Digital → toneMelody, move buzzer from pin 9 to pin 8, upload code and edit code to make your own song.

Program Code: (File → Examples → 02.Digital → tonePitchFollower)

```
void setup() {
  // initialize serial communications (for debugging only):
  Serial.begin(9600);
}

void loop() {
  // read the sensor:
  int sensorReading = analogRead(A0);
  // print the sensor reading so you know its range
  Serial.println(sensorReading);
  // map the pitch to the range of the analog input.
  // change the minimum and maximum input numbers
below
  // depending on the range your sensor's giving:
  int thisPitch = map(sensorReading, 800, 1000, 100, 900);

  // play the pitch:
  tone(9, thisPitch, 10);
  delay(1);        // delay in between reads for stability
}
```



Created by Thomas Preece for the Technology Volunteers

Warwick Volunteers

map()

The map(range1value,range1min,range1max,range2min,range2max) command is an easy way to scale values from one range of number to another. We give it two ranges range1min-range1max and range2min-range2max and a value within the first range, range1value. It then converts that value using maths to a value within the second range. In the above example the range 800-1000 is scaled to 100-900 so if sensorReading (our value range1value) is 800 map returns 100, if it is 1000 it returns 900. If sensorReading is 900 it returns 500 as 900 is half way in range1: 800-1000 so it returns the half way value of range2: 100-900.
We need to use this command so the buzzer gives out a nicer range of sounds that we can hear easier.

tone()

tone(buzzerPin,pitch,length) is a built in function that causes a buzzer on pin number 'buzzerPin' to output a beep at a specified frequency 'pitch' for a certain length of time 'length'.