

```
1/* This sketch was written by the Daria team, with help from SparkFun Electronics, and the
2  Arduino community. This code is completely free for any use.
3 */
4
5 // We'll be controlling the motor from pin 3.
6 // This must be one of the PWM-capable pins. e.g., a pin with a ~ in front of it.
7
8 const int motorPin = 3;
9
10 // We'll use pin 9 and 10 for the sensor.
11 const int trigPin = 9;
12 const int echoPin = 10;
13
14 // defines variables for sensor
15 long duration;
16 int distance;
17 int previousDistance;
18
19
20 void setup()
21 // this is the part of the code that runs one time, to set up.
22 {
23   // Set up the motor pin to be an output
24   pinMode(motorPin, OUTPUT);
25
26 // Set up sensor
27   pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
28   pinMode(echoPin, INPUT); // Sets the echoPin as an Input
29
30 // Set up the serial port
31   Serial.begin(9600); // Starts the serial communication
32 }
33
34
35 void loop()
36 // this part of the code runs constantly
37 {
38
39 //SENSOR
40 // Clear the trigPin
41 digitalWrite(trigPin, LOW);
42 delayMicroseconds(2);
43
44 // Set the trigPin on HIGH state for 10 micro seconds
45 digitalWrite(trigPin, HIGH);
46 delayMicroseconds(10);
47 digitalWrite(trigPin, LOW);
48
49 // Read the echoPin, returns the sound wave travel time in
50   microseconds.
51
52 duration = pulseIn(echoPin, HIGH);
53
54 // Calculate the distance in cm
55 distance = duration * 0.034/2;
56
57 /* We now set the condition for when Daria should switch to the active mode. We want Daria
58   to switch to the active mode when the distance sensor senses something within less than
59   1 meter = 100 cm. However, we do not want Daria to stay in the active mode when an object
60   stands in front of the sensor. e.g. when the measured distance stays the same. That's why
61   we only want to switch to the active mode when the measured distance (distance) is at least
62   1 cm bigger or smaller than the previous distance that activated Daria (previousDistance).
63 */
64
65 if((distance < 100) && ((distance < previousDistance-1) || (distance > previousDistance+1))) {
66
67   int speed;
68   int delayTimeStart = 5;           // milliseconds between each speed step
69   int delayTimeEnd = 20;            // milliseconds between each speed step
70
71 //breath 1
72
73 // decelerate the motor, makes the airpump create less bubbles till it is producing no bubbles.
74 for(speed = 150; speed >= 0; speed--)
75 {
76   analogWrite(motorPin,speed);      // set the new speed
77   delay(delayTimeEnd);               // delay between speed steps
78 }
79   delay(2000);
80
81 // accelerate the motor, makes the airpump create more bubbles.
82 for(speed = 0; speed <= 150; speed++)
83 {
84   analogWrite(motorPin,speed);      // set the new speed
85   delay(delayTimeStart);            // delay between speed steps
86 }
87   delay(500);
88
89 //breath 2
90
91 for(speed = 150; speed >= 0; speed--)
92 { analogWrite(motorPin,speed)
93   delay(delayTimeEnd);
94 } delay(2000);
95 for(speed = 0; speed <= 150; speed++)
96 { analogWrite(motorPin,speed);
97   delay(delayTimeStart);
98 } delay(500);
99
100 //breath 3
101
102 for(speed = 150; speed >= 0; speed--)
103 { analogWrite(motorPin,speed)
104   delay(delayTimeEnd);
105 } delay(2000);
106 for(speed = 0; speed <= 150; speed++)
107 { analogWrite(motorPin,speed);
108   delay(delayTimeStart);
109 } delay(500);
110
111 //breath 4
112
113 for(speed = 150; speed >= 0; speed--)
114 { analogWrite(motorPin,speed)
115   delay(delayTimeEnd);
116 } delay(2000);
117 for(speed = 0; speed <= 150; speed++)
118 { analogWrite(motorPin,speed);
119   delay(delayTimeStart);
120 } delay(500);
121
122 //breath 5
123
124 for(speed = 150; speed >= 0; speed--)
125 { analogWrite(motorPin,speed)
126   delay(delayTimeEnd);
127 } delay(2000);
128 for(speed = 0; speed <= 150; speed++)
129 { analogWrite(motorPin,speed);
130   delay(delayTimeStart);
131 } delay(500);
132
133 //breath 6
134
135 for(speed = 150; speed >= 0; speed--)
136 { analogWrite(motorPin,speed)
137   delay(delayTimeEnd);
138 } delay(2000);
139 for(speed = 0; speed <= 150; speed++)
140 { analogWrite(motorPin,speed);
141   delay(delayTimeStart);
142 } delay(500);
143
144 }
145 // Now we want to remember the distance the sensor measures right now as previousDistance.
146   previousDistance = distance;
147
148 }
```