```
01
02  const int inPins[] = {A0, 8};  // The pins we will be reading in
03  const int outPins[] = {9, 10, 11, 12, 13};  // The pins we will be outputting to (The LEDs)
04
05  // Initialize all our variables
06  int prevState = LOW;
07  int state = LOW;
08  int pressCount = 0;
09  float ratio = 1.0;
10  long currTime = 0;
11
12  // the setup function runs once when you press reset or power the board
13  void setup() {
14    // Initialize all the input and output pins
15    for (int i = 0; i < sizeof(inPins); i++) {
16      pinMode(inPins[i], INPUT);
17    }
18    for (int i = 0; i < sizeof(outPins); i++) {
19      pinMode(outPins[i], OUTPUT);
20    }
21  }
22
23  // Turn on all the pins
24  void allOn() {
25    for (int i = 0; i < sizeof(outPins); i++) {
26      digitalWrite(outPins[i], HIGH);
27    }
28  }
29
30  // Turn off all the pins
31  void allOff() {
32    for (int i = 0; i < sizeof(outPins); i++) {
33      digitalWrite(outPins[i], LOW);
34    }
35  }
36
37  // Turn off every pin except one pin, which we turn on
38  void allOffExcept(int pin) {
39    for (int i = 0; i < sizeof(outPins); i++) {
40      // If the current output pin is the argument pin, turn it on
41      if (pin == outPins[i]) {
42        digitalWrite(outPins[i], HIGH);
43      }
44      // Otherwise, turn it off
45      else {
46        digitalWrite(outPins[i], LOW);
47      }
48    }
49  }
50
51  // Turn off every pin except two pins, which we turn on
52  void allOffExcept(int pin1, int pin2) {
53    for (int i = 0; i < sizeof(outPins); i++) {
54      // If the current output pin is either of the argument pins, turn it on
55      if (pin1 == outPins[i] || pin2 == outPins[i]) {
56        digitalWrite(outPins[i], HIGH);
57      }
58      else {
59        digitalWrite(outPins[i], LOW);
60      }
61    }
62  }
63
64  // Calculate which "phase" we're currently in. For example, if we have 4 phases, a wait
65  time of 1000, and ratio is, then calcCurrentPhase returns 0, 1, 2, or 3, switching values
66  every second.
67  int calcCurrentPhase(int phases, int wait) {
68    int scaledWait = wait * ratio;
69    return (currTime % (scaledWait * phases)) / scaledWait;
70  }
71
72  // Bounce the LED light back and forth
73  void bounce() {
74
75    int currentPhase = calcCurrentPhase(8, 50);
```

```
  switch(currentPhase)
  {

    case 0 :
      allOffExcept(outPins[0]);
      break;
    case 1 :
      allOffExcept(outPins[1]);
      break;
    case 2 :
      allOffExcept(outPins[2]);
      break;
    case 3 :
      allOffExcept(outPins[3]);
      break;
    case 4 :
      allOffExcept(outPins[4]);
      break;
    case 5 :
      allOffExcept(outPins[3]);
      break;
    case 6 :
      allOffExcept(outPins[2]);
      break;
    case 7 :
      allOffExcept(outPins[1]);
      break;

  }

}
// Cycle the LED light
void wheel() {

  int currentPhase = calcCurrentPhase(5, 100);

  switch(currentPhase)
  {
    case 0 :
      allOffExcept(outPins[0]);
      break;
    case 1 :
      allOffExcept(outPins[1]);
      break;
    case 2 :
      allOffExcept(outPins[2]);
      break;
    case 3 :
      allOffExcept(outPins[3]);
      break;
    case 4 :
      allOffExcept(outPins[4]);
      break;
  }

}
// Move the LED from inside to outside
void zigzag()
{
  int currentPhase = calcCurrentPhase(4, 100);

  switch(currentPhase)
  {
    case 0 :
      allOffExcept(outPins[0], outPins[4]);
      break;
    case 1 :
      allOffExcept(outPins[1], outPins[3]);
      break;
    case 2 :
      allOffExcept(outPins[2]);
      break;
    case 3 :
      allOffExcept(outPins[1], outPins[3]);
      break;
  }
}
```

```
105
106 // Flash all LED lights
107 void allFlash() {
108
109   int currentPhase = calcCurrentPhase(2, 100);
110
111   switch(currentPhase)
112   {
113     case 0 :
114       allOn();
115       break;
116     case 1 :
117       allOff();
118       break;
119   }
120 }
121
122 // Read in the button pin. If It has just gone from Low To High, then register a button
123 press.
124 void checkButton() {
125   state = digitalRead(inPins[1]);
126
127   if (state != prevState) {
128     if (state == HIGH) {
129       pressCount++;
130     }
131   }
132
133   prevState = state;
134 }
135
136 // Based on the total number of presses, do a different LED display.
137 void cycle() {
138
139   checkButton();
140
141   if (pressCount % 6 == 0) {
142     allOn();
143   }
144   if (pressCount % 6 == 1) {
145     allOff();
146   }
147   if (pressCount % 6 == 2) {
148     bounce();
149   }
150   if (pressCount % 6 == 3) {
151     zigzag();
152   }
153   if (pressCount % 6 == 4) {
154     allFlash();
155   }
156   if (pressCount % 6 == 5) {
157     wheel();
158   }
159 }
160
161 //// the loop function runs over and over again forever
162 void loop() {
163   currTime = millis();  // read in the current time in milli seconds
164   int val = analogRead(inPins[0]); // read in the value of the distance sensor
165   float voltage= val * (5.0 / 1023.0); // convert it to a voltage from 0-5
166   ratio = 5.0 / (3.6 * voltage - 0.8); // convert it to a ratio where 0.5 volts corresponds
167  to a ratio of 5, and 3 volts corresponds to a ratio of 0.5.
168   ratio = ((int)((ratio + 0.25) / 0.5)) * 0.5; // round it to the nearest 0.5 volts
169   ratio = min(max(ratio, 0.5), 2.5); // coerce it inside the bounds [0.5, 2.5]
170   cycle(); // display the correct LED configuration
171 }
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
```

159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212

```
213
214
215
216
217
218
```

code formatter