## **Touchless Door Sensor | Version 2.0**

Russell MacGregor

```
#include <Wire.h>
    #include <LiquidCrystal_I2C.h>
    LiquidCrystal_I2C lcd(0x27, 16, 2);
    //****Pin initializations****
 5
   int buzzer = 4;
    int led = 9;
    int button = 7;
 9
10
    int trigPin = 6;
    int echoPin = 5;
11
12
    //****Variables****
13
    //Ultrasonic Sensor
15
    long duration;
    int distance;
16
17
18
    //Program
19
    int mode = 0;
  bool distanceSet = false;
   bool boundsExceeded = false;
21
   int closedDoorDistance;
23
    int upperBound;
    int lowerBound;
24
25
    int timeRemaining;
    int index = 0;
26
27
    //****Settings****
28
29
    int tolerance = 20;
    int countdownTime = 10; //time in seconds it takes for the alert to get set off after distance
    exceeds acceptable bounds
31
    void setup() {
32
      //LCD Initialization
33
34
      lcd.init();
35
      lcd.backlight();
37
      //Serial monitor initialization
38
      Serial.begin(9600);
39
      //pinMode Initialziation
40
41
      pinMode(trigPin, OUTPUT);
42
      pinMode(echoPin, INPUT);
43
      pinMode(led, OUTPUT);
      pinMode(button, INPUT PULLUP);
44
45
      //Code to run once:
46
47
      mode = 1;
48
      timeRemaining = countdownTime;
49
    }
50
51
    void loop() {
52
      getDistance();
53
      if (mode == 1) {
54
55
        runConfig();
56
57
```

about:blank 1/4

```
11/2/22, 9:25 PM
    58
           if (mode == 2) {
    59
             runSurveillance();
    60
    61
           if (mode == 3) {
    62
             runCountdown();
    63
    64
    65
           if (mode == 4) {
    66
    67
             runAlert();
           }
    68
    69
         }
    70
    71
         void returnToConfig(){
           mode = 1;
    72
    73
           ledOff();
    74
           buzzerOff();
    75
           lcd.clear();
           timeRemaining = countdownTime;
    76
    77
           distanceSet = false;
    78
           boundsExceeded = false;
    79
           delay(100);
    80
         }
    81
         void returnToSurveillance() {
    82
           mode = 2;
    83
           ledOff();
    84
           buzzerOff();
    85
           lcd.clear();
    86
    87
           timeRemaining = countdownTime;
           delay(10);
    88
         }
    89
    90
    91
         void displayDistance(){
           lcd.home();
    92
    93
           lcd.print("Distance: ");
           lcd.print(distance);
    94
    95
           lcd.print("
                          ");
    96
           delay(500);
    97
         }
    98
         void displayButtonNotPushed(){
    99
   100
           lcd.setCursor(0,1);
   101
           lcd.print("distance not set");
   102
         }
   103
         void displaySurveillanceScreen(){
   104
           lcd.setCursor(0,1);
   105
           lcd.print("Set, monitoring ");
   106
   107
   108
         void displayCountdownScreen(){
   109
   110
           lcd.setCursor(11,1);
   111
           lcd.print("
                           ");
   112
           delay(50);
           lcd.setCursor(0,1);
   113
           lcd.print("Countdown: ");
   114
           lcd.print(timeRemaining);
   115
           Serial.print("Countdown: ");
   116
   117
           Serial.print(timeRemaining);
           Serial.print("\n");
   118
   119
         }
   120
   121
         void displayAlertScreen(){
   122
           lcd.setCursor(0,1);
```

about:blank

2/4

```
lcd.print("Close door");
123
124
     }
125
126
     //Configuration mode
     void runConfig(){
127
128
        displayDistance();
129
        displayButtonNotPushed();
130
        if (digitalRead(button) == LOW){
131
132
          closedDoorDistance = distance;
133
          upperBound = closedDoorDistance + tolerance;
134
          lowerBound = closedDoorDistance - tolerance;
135
          mode = 2;
136
          lcd.clear();
137
          delay(10);
138
        }
      }
139
140
141
      //Surveillance
142
      void runSurveillance() {
143
        displayDistance();
144
145
        displaySurveillanceScreen();
        checkDistanceBounds();
146
147
        if (boundsExceeded == true) {
148
149
          mode = 3;
150
          lcd.clear();
151
          delay(10);
152
        }
153
        if (digitalRead(button) == LOW){
154
155
          returnToConfig();
156
        }
157
      }
158
159
     //Countdown
     void runCountdown() {
160
161
162
        ledOn();
163
        displayDistance();
164
        displayCountdownScreen();
        checkDistanceBounds();
165
166
        if (boundsExceeded == true) {
167
168
          if (timeRemaining == 0) {
169
            mode = 4;
170
            lcd.clear();
171
            delay(10);
172
173
174
          if (timeRemaining != 0) {
175
            timeRemaining = timeRemaining - 1;
176
            delay(1000);
177
          }
178
        }
179
180
        if (boundsExceeded == false) {
181
          returnToSurveillance();
182
        }
183
        if (digitalRead(button) == LOW){
184
185
          returnToConfig();
186
187
```

about:blank 3/4

```
188
189
     //Alert
     void runAlert() {
190
191
       buzzerOn();
192
       displayDistance();
       displayAlertScreen();
193
194
       checkDistanceBounds();
195
       if (boundsExceeded == false) {
196
197
          returnToSurveillance();
198
199
200
       if (digitalRead(button) == LOW) {
201
          returnToConfig();
202
       }
203
     }
204
205
      void ledOn() {
       digitalWrite(led, HIGH);
206
207
208
209
     void ledOff() {
210
       digitalWrite(led, LOW);
211
212
213
     void buzzerOn() {
214
       tone(buzzer, 1000);
       delay(50);
215
216
217
       noTone(buzzer);
218
       delay(50);
219
220
     }
221
     void buzzerOff() {
222
223
       noTone(buzzer);
224
225
226
     void getDistance() {
227
       // Clears the trigPin
228
       digitalWrite(trigPin, LOW);
229
       delayMicroseconds(2);
230
231
       digitalWrite(trigPin, HIGH);
232
       delayMicroseconds(10);
233
       digitalWrite(trigPin, LOW);
234
235
       duration = pulseIn(echoPin, HIGH);
236
237
       distance = duration * 0.034 / 2;
238
      }
239
240
     void checkDistanceBounds(){
       if ( (distance < lowerBound) || (distance > upperBound) ){
241
242
          boundsExceeded = true;
243
       }
244
       else{
245
          boundsExceeded = false;
246
       }
247
     }
```

about:blank 4/4