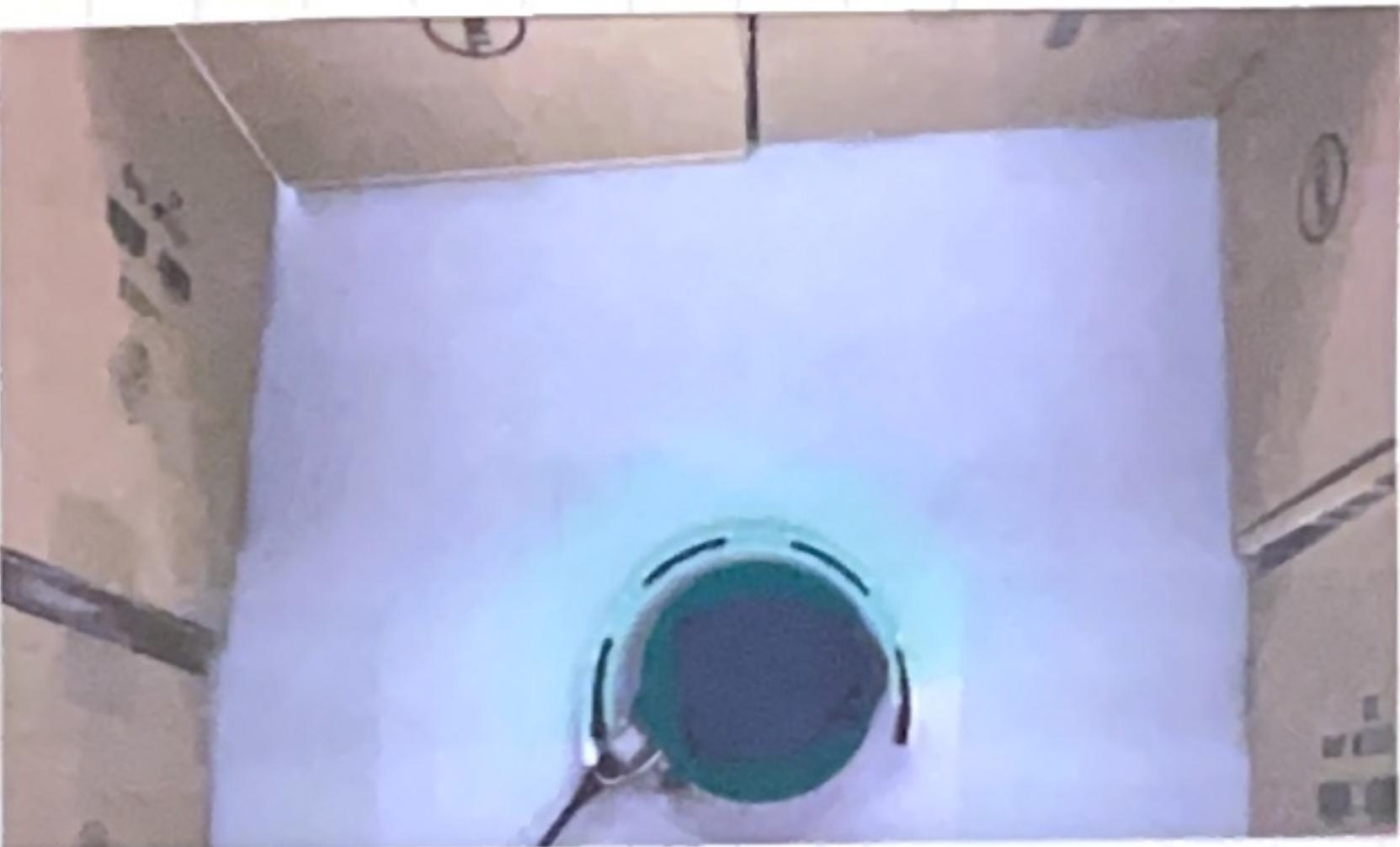


Install the robo peak arduino library:
<https://github.com/robopeak/rplidar-arduino>

Sketch → include Library → Add .ZIP

Objective

Detect objects that are close or far away and light up an LED that relates to the distance.



Results

~~ext~~ Results for objects far away



Results for objects at a medium distance



Results for objects very close.

Code

```

1 #include <RPLidar.h>
2 #include <Adafruit_NeoPixel.h>
3
4 #define LED_PIN 8
5 #define LED_COUNT 19 // These variables are for the neo pixels.
6
7 Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB+NEO_KHZ800);
8
9 RPLidar lidar;
10 #define RPLIDAR_MOTOR 3 // PWM Pin for motor control.
11
12 // Let's keep an array that has a distance for every integer angle.
13 float distance[360]; // Can access indexes 0-359.
14 // Let's look at 10 rotations before we average our data.
15 int numRotations = 0;
16 int targetRotations = 10;
17
18 void setup()
19 {
20   Serial.begin(115200);
21   lidar.begin(Serial);
22   pinMode(RPLIDAR_MOTOR, OUTPUT);
23
24   strip.begin();
25   strip.setBrightness(50);
26 }
27 void printData()
28 {
29   // This method shows each angle and their corresponding distance
30   for (int i=0; i<=360; i++)
31   {
32     Serial.print(i);
33     Serial.print(" ");
34     Serial.println(distance[i]);
35   }
36 }
37
38 void loop()
39 {
40   if (IS_OK(lidar.waitPoint()))
41   {
42     RPLidarMeasurement currentMeasure = lidar.getCurrentPoint();
43     if (currentMeasure.startBit == 1)
44     {
45       numRotations++;            ↗ looking for 10 rotations.
46       if (numRotations >= targetRotations)
47       {
48         printData();
49         numRotations = 0;
50         lightLED(); *→ all the cool stuff is in here
51       }
52     }
53     else
54     {
55       distance[int(currentMeasure.angle)] = currentMeasure.distance;
56     }
57   }
58   else
59   {
60     analogWrite(RPLIDAR_MOTOR, 0); // Stop the motor
61     rplidar_response_device_info_t info;
62     Serial.println("----");
63     Serial.println(IS_OK(lidar.waitPoint()));
64     Serial.println(IS_OK(lidar.getDeviceInfo(info, 100)));
65     if (IS_OK(lidar.getDeviceInfo(info, 100)))
66     {
67       lidar.startScan();
68       analogWrite(RPLIDAR_MOTOR, 255);
69       delay(1000);
70     }
71   }
72 }
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137

```

} handle an
error state
and restart.

```
50 void lightLED()
51 {
52
53     // We know we need 19 averages. -> i= (0-19) 10 degrees for each pixel.
54     for(int i=0; i<19; i++)
55     {
56
57         // We keep a sum and how many value we added to our average.
58         float sum = 0;
59         float averageNum = 0;
60         float averageDistance = 0.00;
61
62         // Here, we need to look at the proper subset of our data.
63         // For iteration one, we want to look at indexes 0-9 (degrees)
64         // For iteration two, we want to look at indexes 10-19 (degrees)
65
66         // So we start by multiplying which average we are on by 10. -> i*10
67         // we go up to an index of i*10 + 9. So that we get 10 total values.
68
69         for(int a = i*10; a<=i*10+9; a++) *→ look at proper subset.
70         {
71
72             // We are ignoring 0 values since we don't know
73             // this 0 is from being too close, or that the lidar
74             // didn't get data there.
75             if (distance[a] != 0.00)
76             {
77                 sum += distance[a];
78                 averageNum++;
79             }
80
81             // If we actually added a non zero value to our sum.
82             if (averageNum != 0)
83             {
84
85                 // Get the average distance
86                 averageDistance = sum/averageNum;
87             }
88             else
89             {
90
91                 // Otherwise, set it to 0 to say either
92                 // 1. We didn't actually get distances at those angles.
93                 // 2. Something is way too close to the sensor.
94                 averageDistance = 0.00;
95             }
96
97             // Set the strip color based on the average distance.
98             Serial.println(averageDistance);
99             if (averageDistance == 0.00) // White for 0's
100             {
101                 strip.setPixelColor(i, strip.Color(255,255,255));
102             }
103             else if (averageDistance < 500) // Red for too close!
104             {
105                 strip.setPixelColor(i, strip.Color(255,0,0));
106             }
107             else if (averageDistance < 1000) // Yellow for getting close
108             {
109                 strip.setPixelColor(i, strip.Color(255,165,0));
110             }
111             else // Green for not close enough
112             {
113                 strip.setPixelColor(i, strip.Color(0,255,0));
114             }
115         }
116         strip.show();
117     }
118 }
```

} inner for
loop explanation.

} Light up
the LED's