

```

1  # 11 Sistema d'Alarma Completa amb PIR, Buzzer i LEDs (Amb Threading)
2  # Crear una alarma completa que inclogui el sensor PIR, el sensor ultrasònic,
   diversos LEDs i
3  # un bronzidor per senyalitzar la proximitat i el moviment en paral·lel utilitzant
   *threading*.
4  # Solució en codi:
5  # Solució en codi Python
6  import RPi.GPIO as GPIO
7  import time
8  import threading
9
10 GPIO.setmode(GPIO.BCM)
11 PIR_PIN = 4
12 TRIG = 23
13 ECHO = 24
14 LED_PIN = 17
15 BUZZER_PIN = 18
16
17 GPIO.setup(TRIG, GPIO.OUT)
18 GPIO.setup(ECHO, GPIO.IN)
19 GPIO.setup(PIR_PIN, GPIO.IN)
20 GPIO.setup(LED_PIN, GPIO.OUT)
21 GPIO.setup(BUZZER_PIN, GPIO.OUT)
22
23 def distance():
24     GPIO.output(TRIG, True)
25     time.sleep(0.00001)
26     GPIO.output(TRIG, False)
27     start, stop = time.time(), time.time()
28
29     while GPIO.input(ECHO) == 0:
30         start = time.time()
31     while GPIO.input(ECHO) == 1:
32         stop = time.time()
33
34     return (stop - start) * 34300 / 2
35
36 def proximity_alert():
37     while True:
38         dist = distance()
39         if dist < 5:
40             GPIO.output(LED_PIN, True)
41             GPIO.output(BUZZER_PIN, True)
42         else:
43             GPIO.output(LED_PIN, False)
44             GPIO.output(BUZZER_PIN, False)
45             time.sleep(0.5)
46
47 def movement_alert():
48     while True:
49         if GPIO.input(PIR_PIN):
50             GPIO.output(LED_PIN, True)
51             time.sleep(1)
52         else:
53             GPIO.output(LED_PIN, False)
54             time.sleep(0.5)
55
56 try:
57     proximity_thread = threading.Thread(target=proximity_alert)
58     movement_thread = threading.Thread(target=movement_alert)
59
60     proximity_thread.start()
61     movement_thread.start()
62     proximity_thread.join()
63     movement_thread.join()
64
65 except KeyboardInterrupt:
66     GPIO.cleanup()

```