**Raspberry Pi Kullanarak Akıllı Otopark Sistemleri**

**Giriş**

Bu proje akıllı otopark tasarımıdır. Raspberry pi zero w, IR alıcı-verici ve LCD Display kullanılarak otoparklarda park sorununa çözüm bulmak için tasarlanmıştır. Basit ve ucuz bir yapı olmaktadır. Katlı otoparklarda otoparka girmeden dışarıdan LCD Display aracılığıyla hangi katta kaç araçlık boş yer olduğunu belirtmekte ve otoparkın içinde park yerlerinin üzerinde Led’ler yardımıyla belirtmeyi sağlamaktadır.

Projede IR alıcı-verici ile park alanında araç olup olmadığı öğrenilecek. Araç yok ise Led yardımıyla buranın boş olduğu bilgisi verilecek. Aynı anda da dışarıda yer alan LCD Dispaly de kattaki boş araç sayısı yazacak.

**Gerekli Donanım Bileşenleri**

1. 1 adet Raspberry Pi Zero W
2. Park yeri sayısınca IR alıcı-verici
3. Park yeri sayısınca Led
4. Her iki kat için 1 adet 2x16 LCD Display

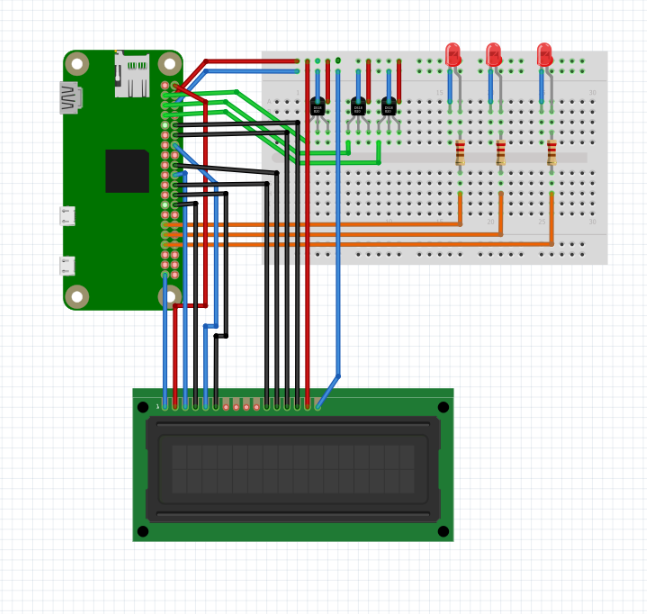
**Gerekli Yazılım Bileşenleri**

1. Raspbian Jessie OS([www.raspbian.org](http://www.raspbian.org))

**Kullanılan Bileşenlerin Özellikleri**

1. Raspberry Pi Zero W; Raspberry Pi Zero ile aynı yapıya sahip olup ek olarak dahili Wifi ve Bluetooth özelliklerine sahiptir.
2. IR alıcı-verici; Sensör modülü bir çift ir alıcı verici tüpe sahiptir, ortam ışığına uyum sağar. Verici tüpler IR bandında belli bir frekansta ışık verir, ışık huzmesinin yoluna bir engel çıktığında, söz konusu engelin yüzeyine çarpıp yansıyan ışık alıcı sensör tarafından okunur. Oluşan elektriksel sinyal bir komprador devresince işlenir ve yeşil indikatör ışığı yanar, dijital çıkıştan da LOW durumu okunur. Algılama uzaklığı potansiyometre ile ayarlanır. Verimli mesafe 2-30cm, çalışma voltaj aralığı ise 3.3V-5V arasındadır. (<https://www.direnc.net/ir-alici-verici-modul>)
3. Led; yarı-iletken, diyot temelli, ışık yayan bir elektronik devre elemanıdır.
4. LCD Display; Liquid Crystal Display yani Sıvı Kristal Ekran elektrikle kutuplanan sıvının ışığı tek fazlı geçirmesi ve önüne eklenen bir kutuplanma filtresi ile gözle görülebilmesi ilkesine dayanan bir görüntü teknolojisidir.

**Şematik Çizimi**



schema.png

(Şematik çizimde IR alıcı-verici olmadığı için transistör kullanıldı.)

**Yapım Aşamaları**

İlk olarak Raspberry Pi Zero W içerisindeki Rasbian OS’yi güncellemek için aşağıdaki komutu çalıştırın:

1. sudo apt - getupload

Sonra RPLCD kütüphanesini PIP dizininden kuruyoruz:

1. sudo apt - get install python - pip

Daha sonra RPLCD kütüphanesini yüklüyoruz:

1. sudo pip3 install RPLCD

LCD kodların düzenli çalışabilmesi için <https://github.com/leon-anavi/raspberrypi-lcd.git> adresinden kod yardımı almak için kodları klonluyoruz:

1. git clone  <https://github.com/leon-anavi/raspberrypi-lcd>

**Python Kodu**

1. # The wiring for the LCD is as follows:
2. # 1 : GND
3. # 2 : 5V
4. # 3 : Contrast (0-5V)\*
5. # 4 : RS (Register Select)
6. # 5 : R/W (Read Write) - GROUND THIS PIN
7. # 6 : Enable or Strobe
8. # 7 : Data Bit 0 - NOT USED
9. # 8 : Data Bit 1 - NOT USED
10. # 9 : Data Bit 2 - NOT USED
11. # 10: Data Bit 3 - NOT USED
12. # 11: Data Bit 4
13. # 12: Data Bit 5
14. # 13: Data Bit 6
15. # 14: Data Bit 7
16. # 15: LCD Backlight +5V\*\*
17. # 16: LCD Backlight GND
18. import RPi.GPIO as GPIO
19. import os
20. import socket
21. import fcntl
22. import struct
23. import time
24. from time import gmtime, strftime
25. # LCD and sensor pins
26. LCD\_RS = 7
27. LCD\_E = 8
28. LCD\_D4 = 25
29. LCD\_D5 = 24
30. LCD\_D6 = 23
31. LCD\_D7 = 18
32. katbir1=2
33. katbir2=3
34. katiki1=4
35. ledbir1=5
36. ledbir2=6
37. lediki1=13
38. # Define some device constants
39. LCD\_WIDTH = 16 # Maximum characters per line
40. LCD\_CHR = True
41. LCD\_CMD = False
42. LCD\_LINE\_1 = 0x80 # LCD RAM address for the 1st line
43. LCD\_LINE\_2 = 0xC0 # LCD RAM address for the 2nd line
44. # Timing constants
45. E\_PULSE = 0.0005
46. E\_DELAY = 0.0005
47. def main():
49. GPIO.setwarnings(False)
50. GPIO.setmode(GPIO.BCM) # Use BCM GPIO numbers
51. GPIO.setup(LCD\_E, GPIO.OUT) # E
52. GPIO.setup(LCD\_RS, GPIO.OUT) # RS
53. GPIO.setup(LCD\_D4, GPIO.OUT) # DB4
54. GPIO.setup(LCD\_D5, GPIO.OUT) # DB5
55. GPIO.setup(LCD\_D6, GPIO.OUT) # DB6
56. GPIO.setup(LCD\_D7, GPIO.OUT) # DB7
57. GPIO.setup(katbir1, GPIO.IN) # First floor first sensor
58. GPIO.setup(katbir2, GPIO.IN) # Fisrt floor second sensor
59. GPIO.setup(katiki1, GPIO.IN) # Second floor first sensor
60. GPIO.setup(ledbir1, GPIO.OUT)# First floor first led
61. GPIO.setup(ledbir2, GPIO.OUT)# First floor second led
62. GPIO.setup(lediki1, GPIO.OUT)# Second floor first led
63. # Initialise display
64. lcd\_init()
65. while True:
66. while True :
67. if katbir1==0:
68. ledbir1=1
69. else :
70. ledbir1=0
72. if katbir2==0:
73. ledbir2=1
75. else:
76. ledbir2=0
77. if katiki1==0:
78. lediki1=1
79. else :
80. lediki1=0
81. break
82. if (katbir1==1 || katbir2==1) && katiki1==0:
84. lcd\_string("1.kat 1 bos",LCD\_LINE\_1)
85. lcd\_string("2.kat 1 bos",LCD\_LINE\_2)
86. time.sleep(3)
87. if katbir1==1 && katbir2==1 && katiki1==0:
89. lcd\_byte(lcd.LCD\_LINE\_1, lcd.LCD\_CMD)
90. lcd\_string("1.kat dolu",LCD\_LINE\_1)
91. lcd\_byte(lcd.LCD\_LINE\_2, lcd.LCD\_CMD)
92. lcd\_string("2.kat 1 bos",LCD\_LINE\_2)
93. time.sleep(3)
94. if katiki1==1 && katbir1==0 && katbir2==0:
96. lcd\_byte(lcd.LCD\_LINE\_1, lcd.LCD)
97. lcd\_string("1.kat 2 bos",LCD\_LINE\_1)
98. lcd\_byte(lcd.LCD\_LINE\_2, lcd.LCD\_CMD)
99. lcd\_string("2.kat dolu",LCD\_LINE\_2)
100. time.sleep(3)
102. if katiki1==1 && katbir1==1 && katbir2==1:
104. lcd\_byte(lcd.LCD\_LINE\_1, lcd.LCD\_CMD)
105. lcd\_string("1.kat dolu",LCD\_LINE\_1)
106. lcd\_byte(lcd.LCD\_LINE\_2, lcd.LCD\_CMD)
107. lcd\_string("2.kat dolu",LCD\_LINE\_2)
108. time.sleep(3)
109. if katiki1==1 && (katbir1==1|| katbir2==1):
111. lcd\_byte(lcd.LCD\_LINE\_1, lcd.LCD\_CMD)
112. lcd\_string("1.kat 1 bos",LCD\_LINE\_1)
113. lcd\_byte(lcd.LCD\_LINE\_2, lcd.LCD\_CMD)
114. lcd\_string("2.kat dolu",LCD\_LINE\_2)
115. time.sleep(3)
116. def lcd\_init():
117. # Initialise display
118. lcd\_byte(0x33,LCD\_CMD) # 110011 Initialise
119. lcd\_byte(0x32,LCD\_CMD) # 110010 Initialise
120. lcd\_byte(0x06,LCD\_CMD) # 000110 Cursor move direction
121. lcd\_byte(0x0C,LCD\_CMD) # 001100 Display On,Cursor Off, Blink Off
122. lcd\_byte(0x28,LCD\_CMD) # 101000 Data length, number of lines, font size
123. lcd\_byte(0x01,LCD\_CMD) # 000001 Clear display
124. time.sleep(E\_DELAY)
125. def lcd\_byte(bits, mode):
126. # Send byte to data pins
127. # bits = data
128. # mode = True for character
129. # False for command
130. GPIO.output(LCD\_RS, mode) # RS
131. # High bits
132. GPIO.output(LCD\_D4, False)
133. GPIO.output(LCD\_D5, False)
134. GPIO.output(LCD\_D6, False)
135. GPIO.output(LCD\_D7, False)
136. if bits&0x10==0x10:
137. GPIO.output(LCD\_D4, True)
138. if bits&0x20==0x20:
139. GPIO.output(LCD\_D5, True)
140. if bits&0x40==0x40:
141. GPIO.output(LCD\_D6, True)
142. if bits&0x80==0x80:
143. GPIO.output(LCD\_D7, True)
144. # Toggle 'Enable' pin
145. lcd\_toggle\_enable()
146. # Low bits
147. GPIO.output(LCD\_D4, False)
148. GPIO.output(LCD\_D5, False)
149. GPIO.output(LCD\_D6, False)
150. GPIO.output(LCD\_D7, False)
151. if bits&0x01==0x01:
152. GPIO.output(LCD\_D4, True)
153. if bits&0x02==0x02:
154. GPIO.output(LCD\_D5, True)
155. if bits&0x04==0x04:
156. GPIO.output(LCD\_D6, True)
157. if bits&0x08==0x08:
158. GPIO.output(LCD\_D7, True)
159. # Toggle 'Enable' pin
160. lcd\_toggle\_enable()
161. def lcd\_toggle\_enable():
162. # Toggle enable
163. time.sleep(E\_DELAY)
164. GPIO.output(LCD\_E, True)
165. time.sleep(E\_PULSE)
166. GPIO.output(LCD\_E, False)
167. time.sleep(E\_DELAY)
168. def lcd\_string(message,line):
169. # Cast to string
170. message = str(message)
171. # Send string to display
172. message = message.ljust(LCD\_WIDTH," ")
173. lcd\_byte(line, LCD\_CMD)
174. for i in range(LCD\_WIDTH):
175. lcd\_byte(ord(message[i]),LCD\_CHR)

**Kaynak Kodu**

Buradaki proje resimlerine, videolarına (kısa bir video koyunuz) ve kaynak koduna

<https://github.com/Tzcan> adresinden erişilebilir.

**Nasıl Kullanılır?**

Akıllı Otopark Sistemlerinin kullanımı çok basittir. Sistem park alanlarına kurulduktan sonra Raspberry Pi’nin komut satırına aşağıdaki kod yazılıp çalıştırılır.

1. python otopark.py

Ardından Led’ler ve LCD Display çalışmaya başlayacaktır.

**Tezcan Bilgiç 170215004**

**Okan Ertürk 171214009**