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)

Kullanılan Bileşenlerin Özellikleri

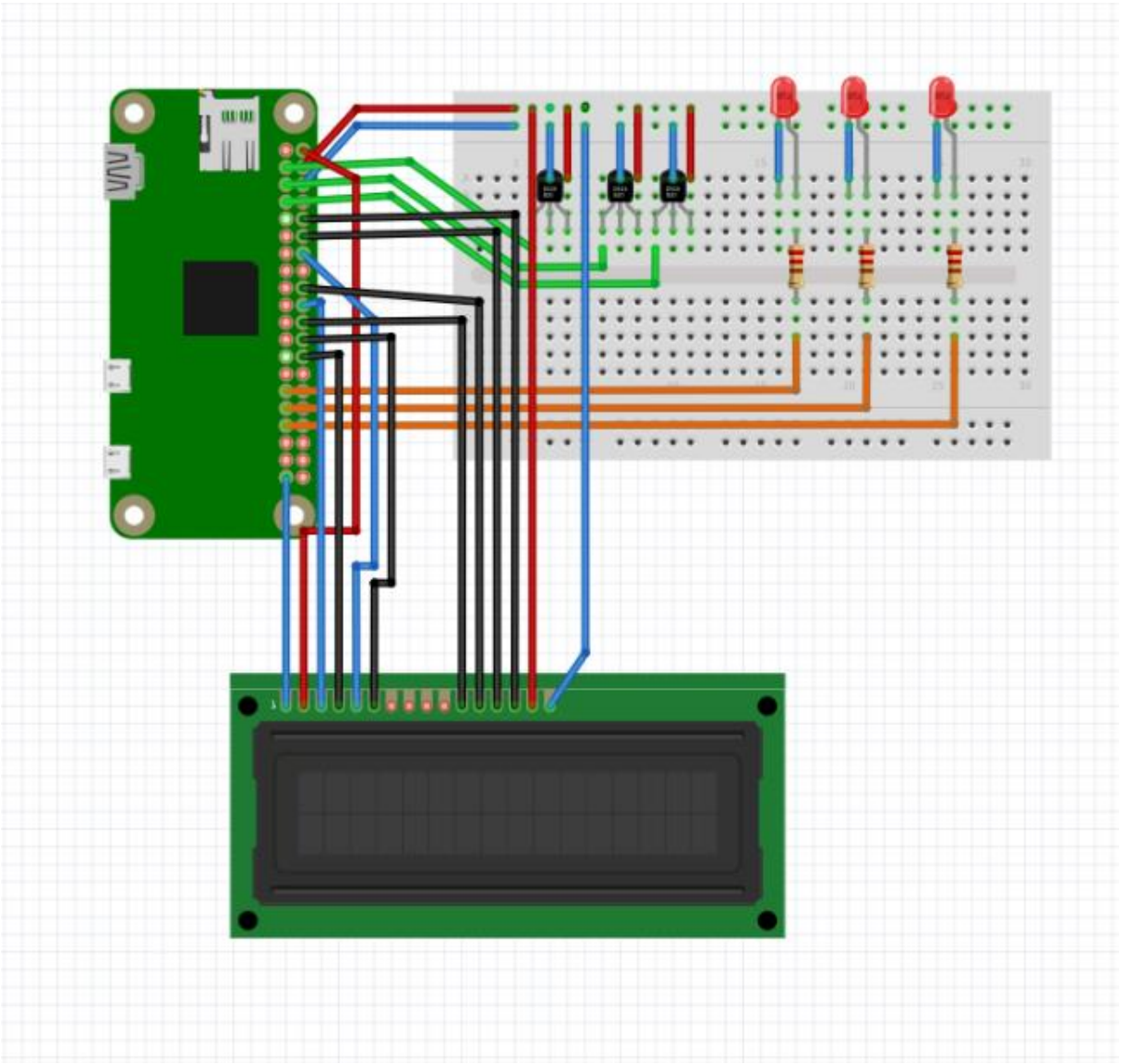
1. Raspberry Pi Zero W; Raspberry Pi Zero ile aynı yapıya sahip olup ek olarak dahili Wifi ve Bluetooth özelliklerine sahiptir.

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:

```
sudo apt - get install python - pip
```

Daha sonra RPLCD kütüphanesini yüklüyoruz:

```
sudo pip3 install RPLCD
```

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

```
git clone  https://github.com/leon-anavi/raspberrypi-lcd
```

Python Kodu

```
# The wiring for the LCD is as follows:
# 1 : GND
# 2 : 5V
# 3 : Contrast (0-5V)*
# 4 : RS (Register Select)
# 5 : R/W (Read Write)       - GROUND THIS PIN
# 6 : Enable or Strobe
# 7 : Data Bit 0             - NOT USED
# 8 : Data Bit 1             - NOT USED
# 9 : Data Bit 2             - NOT USED
# 10: Data Bit 3             - NOT USED
# 11: Data Bit 4
# 12: Data Bit 5
# 13: Data Bit 6
# 14: Data Bit 7
# 15: LCD Backlight +5V**
# 16: LCD Backlight GND

import RPi.GPIO as GPIO
import os
import socket
import fcntl
import struct
import time
from time import gmtime, strftime


# LCD and sensor pins
LCD_RS = 7
LCD_E  = 8
LCD_D4 = 25
LCD_D5 = 24
LCD_D6 = 23
LCD_D7 = 18

katbir1=2
katbir2=3
katiki1=4

ledbir1=5
ledbir2=6
lediki1=13


```

```
# Define some device constants
LCD_WIDTH = 16    # Maximum characters per line
LCD_CHR = True
LCD_CMD = False

LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line
LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line

# Timing constants
E_PULSE = 0.0005
E_DELAY = 0.0005

def main():

  GPIO.setwarnings(False)
  GPIO.setmode(GPIO.BCM)       # Use BCM GPIO numbers
  GPIO.setup(LCD_E, GPIO.OUT)  # E
  GPIO.setup(LCD_RS, GPIO.OUT) # RS
  GPIO.setup(LCD_D4, GPIO.OUT) # DB4
  GPIO.setup(LCD_D5, GPIO.OUT) # DB5
  GPIO.setup(LCD_D6, GPIO.OUT) # DB6
  GPIO.setup(LCD_D7, GPIO.OUT) # DB7
  GPIO.setup(katbir1, GPIO.IN) # First floor first sensor
  GPIO.setup(katbir2, GPIO.IN) # Fisrt floor second sensor
  GPIO.setup(katiki1, GPIO.IN) # Second floor first sensor
  GPIO.setup(ledbir1, GPIO.OUT)# First floor first led
  GPIO.setup(ledbir2, GPIO.OUT)# First floor second led
  GPIO.setup(lediki1, GPIO.OUT)# Second floor first led

  # Initialise display
  lcd_init()

  while True:
    while True :
        if katbir1==0:
            ledbir1=1

        else :
            ledbir1=0


        if katbir2==0:
            ledbir2=1

        else:
            ledbir2=0


        if katiki1==0:
            lediki1=1

        else :
            lediki1=0

        break

           if (katbir1==1 || katbir2==1) && katiki1==0:

               lcd_string("1.kat 1 bos",LCD_LINE_1)
               lcd_string("2.kat 1 bos",LCD_LINE_2)
               time.sleep(3)

           if katbir1==1 && katbir2==1 && katiki1==0:

               lcd_byte(lcd.LCD_LINE_1, lcd.LCD_CMD)
               lcd_string("1.kat dolu",LCD_LINE_1)
```

```
111.              lcd_byte(lcd.LCD_LINE_2, lcd.LCD_CMD)
112.              lcd_string("2.kat 1 bos",LCD_LINE_2)
113.              time.sleep(3)
114.
115.          if katiki1==1 && katbir1==0 && katbir2==0:
116.
117.              lcd_byte(lcd.LCD_LINE_1, lcd.LCD)
118.              lcd_string("1.kat 2 bos",LCD_LINE_1)
119.              lcd_byte(lcd.LCD_LINE_2, lcd.LCD_CMD)
120.              lcd_string("2.kat dolu",LCD_LINE_2)
121.              time.sleep(3)
122.
123.          if katiki1==1 && katbir1==1 && katbir2==1:
124.
125.              lcd_byte(lcd.LCD_LINE_1, lcd.LCD_CMD)
126.              lcd_string("1.kat dolu",LCD_LINE_1)
127.              lcd_byte(lcd.LCD_LINE_2, lcd.LCD_CMD)
128.              lcd_string("2.kat dolu",LCD_LINE_2)
129.              time.sleep(3)
130.
131.          if katiki1==1 && (katbir1==1|| katbir2==1):
132.
133.              lcd_byte(lcd.LCD_LINE_1, lcd.LCD_CMD)
134.              lcd_string("1.kat 1 bos",LCD_LINE_1)
135.              lcd_byte(lcd.LCD_LINE_2, lcd.LCD_CMD)
136.              lcd_string("2.kat dolu",LCD_LINE_2)
137.              time.sleep(3)
138.
139.
140.      def lcd_init():
141.        # Initialise display
142.        lcd_byte(0x33,LCD_CMD) # 110011 Initialise
143.        lcd_byte(0x32,LCD_CMD) # 110010 Initialise
144.        lcd_byte(0x06,LCD_CMD) # 000110 Cursor move direction
145.        lcd_byte(0x0C,LCD_CMD) # 001100 Display On,Cursor Off, Blink Off
146.        lcd_byte(0x28,LCD_CMD) # 101000 Data length, number of lines, font size
147.        lcd_byte(0x01,LCD_CMD) # 000001 Clear display
148.        time.sleep(E_DELAY)
149.
150.      def lcd_byte(bits, mode):
151.        # Send byte to data pins
152.        # bits = data
153.        # mode = True  for character
154.        #        False for command
155.
156.        GPIO.output(LCD_RS, mode) # RS
157.
158.        # High bits
159.        GPIO.output(LCD_D4, False)
160.        GPIO.output(LCD_D5, False)
161.        GPIO.output(LCD_D6, False)
162.        GPIO.output(LCD_D7, False)
163.        if bits&0x10==0x10:
164.          GPIO.output(LCD_D4, True)
165.        if bits&0x20==0x20:
166.          GPIO.output(LCD_D5, True)
167.        if bits&0x40==0x40:
168.          GPIO.output(LCD_D6, True)
169.        if bits&0x80==0x80:
170.          GPIO.output(LCD_D7, True)
171.
172.        # Toggle 'Enable' pin
173.        lcd_toggle_enable()
174.
175.        # Low bits
176.        GPIO.output(LCD_D4, False)
```

```
177.          GPIO.output(LCD_D5, False)
178.          GPIO.output(LCD_D6, False)
179.          GPIO.output(LCD_D7, False)
180.          if bits&0x01==0x01:
181.            GPIO.output(LCD_D4, True)
182.          if bits&0x02==0x02:
183.            GPIO.output(LCD_D5, True)
184.          if bits&0x04==0x04:
185.            GPIO.output(LCD_D6, True)
186.          if bits&0x08==0x08:
187.            GPIO.output(LCD_D7, True)
188.
189.          # Toggle 'Enable' pin
190.          lcd_toggle_enable()
191.
192.        def lcd_toggle_enable():
193.          # Toggle enable
194.          time.sleep(E_DELAY)
195.          GPIO.output(LCD_E, True)
196.          time.sleep(E_PULSE)
197.          GPIO.output(LCD_E, False)
198.          time.sleep(E_DELAY)
199.
200.        def lcd_string(message,line):
201.          # Cast to string
202.          message = str(message)
203.          # Send string to display
204.          message = message.ljust(LCD_WIDTH," ")
205.
206.          lcd_byte(line, LCD_CMD)
207.
208.          for i in range(LCD_WIDTH):
209.            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